

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет
вычислительной техники

Кафедра
МОиПЭВМ

Направление подготовки 09.03.02 «Информационные системы и технологии»

Направленность Информационные системы и технологии

БАКАЛАВРСКАЯ РАБОТА

на тему

**Web-платформа подбора журналистов для
тематических исследований**

Студент Бурмистрова Ирина Андреевна
(подпись, дата) (ФИО полностью)

Руководитель Михалев А.Г.
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр Такташкин Д.В.
(подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол заседания кафедры от _____ № _____)

Заведующий кафедрой _____ Козлов А.Ю.
(подпись) (фамилия, инициалы)

Работа защищена с отметкой _____ (протокол заседания ГЭК от _____ № _____)

Секретарь ГЭК _____ Попова Н.А.
(подпись) (фамилия, инициалы)

Пенза, 2025

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

Вычислительной техники

Кафедра

МО и ПЭВМ

«Утверждаю»
Заведующий кафедрой
А.Ю. Козлов «20»
декабря 2024 г

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
БАКАЛАВРА**

1. Студент Бурмистрова Ирина Андреевна гр. 21ВИ1 факультета Вычислительной техники

<Фамилия, имя, отчество полностью>

2. Тема работы Web-платформа подбора журналистов для тематических исследований

Тема утверждена приказом ПГУ № _____ от " _____ " _____ 20 _____

2. Руководитель работы доцент кафедры «МОиПЭВМ» Михалев А.Г.

3. Задание на работу (назначение разработки, исходные данные и т.п.)

1) Подбор журналистов по указанным критериям работодателем

2) Поиск статей по указанным критериям работодателем

3) Поиск доступных вакансий для журналистов

4) Редактирование профилей, вакансий и резюме пользователями

6) Генерация резюме журналистов

Технология разработки: ООП

Язык программирования: MySQL, php, html, JavaScript

Среда программирования: Visual Studio Code

4. Перечень подлежащих разработке вопросов _____

1) Анализ предметной области и постановка задачи

2) Проектирование программных средств

3) Реализация программных средств

4) Тестирование программных средств

5) Оформление пояснительной записки

5. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Анализ предметной области и требований к системе	10%	08.04.2025 – 14.04.2025	
2	Проектирование программных средств	20%	15.04.2025 – 25.04.2025	
3	Реализация программных средств	30%	26.04.2025 – 12.05.2025	
4	Тестирование программных средств	20%	13.05.2025 – 22.05.2025	
5	Оформление пояснительной записки	20%	23.05.2025 – 03.06.2025	

Дата выдачи задания «20» декабря 2024г.

Руководитель бакалаврской работы _____
(подпись, дата)

Задание к исполнению принял студент _____
(подпись, дата)

Работу к защите допустить
Декан ФВТ _____ Л.Р. Фионова
(подпись, дата)

Реферат

Пояснительная записка содержит: 116 листов, 53 рисунка , 12 таблиц, 5 приложений

WEB-ПЛАТФОРМА, MYSQL, PHPMYADMIN, HTML, SPRINTHOST, PHP, ЖУРНАЛИСТ, РАБОТОДАТЕЛЬ

Объект разработки: Web-платформа подбора журналистов для тематических исследований.

Цель работы: разработать Web-платформу подбора журналистов для тематических исследований с помощью написания базы данных с использованием MySQL и phpMyAdmin, создания веб-страниц с помощью языка гипертекстовой разметки - HTML, PHP, JavaScript и CMS, а также с разворачиваемостью платформы на хостинге Sprinthost.

Результаты работы: в результате проекта представлена Web-платформа подбора журналистов для тематических исследований.

Средства разработки: MySQL Server, phpMyAdmin, HTML, Sprinthost. JavaScript, PHP,

					ПГУ 09.03.02 – 08. ВР 211.07 ПЗ							
Изм.	Лист	№ докум.	Подпись	Дат	«Web-платформа подбора журналистов для тематических исследований» Пояснительная записка			Лит.	Лист	Листов		
Разраб.	Бурмистрова И.А									7		
Провер.	Михалев А. Г.							группа 21ВИ1				
Реценз.												
Н. Контр.	Такташкин Д.В.											
Утверд.	Козлов А.Ю.											

Содержание

Введение	7
1 Анализ предметной области и постановка задачи	9
1.1 Анализ предметной области	10
1.2 Сравнительный анализ средств подбора журналистов по выбранной тематике.....	11
1.3 Анализ средств реализации web-платформы.....	15
1.4 Анализ требований к Web-платформе	17
1.4.1 Анализ функциональных требований	17
1.4.2 Анализ нефункциональных требований	26
1.5 Постановка задачи.....	27
1.6 Планирование и расчет бюджета.....	28
2 Проектирование Web-платформы подбора журналистов	31
2.1 Проектирование архитектуры приложения	31
2.2 Проектирование базы данных	34
2.2.1 Концептуальное проектирование базы данных	37
2.2.2 Логическое проектирование базы данных	40
2.2.3 Физическое проектирование базы данных.....	42
2.4 Проектирование интерфейса пользователя	44
2.5 Проектирование алгоритма подбора журналистов	47
3 Реализация Web-платформы подбора журналистов.....	51
3.1 Реализация базы данных и заполнение тестовыми данными	53
3.2 Реализация Web –платформы.....	58
3.2.1 Реализация авторизации пользователей	58
3.2.2 Реализация подсистемы администратора.....	60
3.2.3 Реализация подсистемы работодателя	61
3.2.4 Реализация подсистемы журналиста.....	63
4 Тестирование Web-платформы подбора журналистов	66
4.1 Тестирование «черным ящиком»	66
4.2 Метрики кода.....	67

Заключение	70
Список использованных источников	72
Приложение А(обязательное)Текст программы.....	76
Приложение Б(обязательное)Текст программы	84
Приложение В(обязательное)Текст программы.....	100
Приложение Г Диаграммы	107
Приложение Д Результат тестирования	112

Введение

В настоящее время на современном рынке труда постоянно возрастает потребность в эффективных программных средствах для подбора соответствующих специалистов, особенно узкоспециализированной направленности, как журналистика. В условиях стремительного развития информационных и цифровых технологий, специалисты по направлениям исследований и работодатели отмечают нехватку инструментальных средств, которые бы позволили быстро и точно находить квалифицированных журналистов, обладающих необходимыми навыками и опытом работы в сфере проведения тематических исследований.

Из-за этого, Web-платформы для подбора персонала становятся более востребованными и популярными, благодаря своим возможностям объединять многомасштабные данные и предоставлять удобный и простой в использовании интерфейс для взаимодействия между работодателями и кандидатами. В частности, для журналистских исследований необходимо не только наличие определенных навыков претендентов, но и их способности работать с определенными темами и направлениями.

Целью данной работы является разработка Web-платформы подбора журналистов для тематических исследований, которая облегчит процесс подбора журналистов работодателями. Реализация платформы будет основана с помощью современных технологий Web-разработки, включающих в себя использование баз данных MySQL и инструментов управления базами данных phpMyAdmin. Для создания пользовательского интерфейса будут использованы язык гипертекстовой разметки – HTML, PHP, JavaScript и CMS (система управления контентом). Развертывание платформы будет осуществлено на бесплатной версии хостинга Sprinthost, что поможет обеспечить её доступность и надежность.

В ходе проектирования будут рассмотрены основные аспекты разработки подобных систем, включая проектирование базы данных, создание

пользовательского интерфейса и интеграцию различных компонентов в единую функциональную платформу. Результатом проекта станет готовая Web-платформа подбора журналистов для тематических исследований, которая поспособствует работодателям эффективно находить специалистов в сфере журналистики с целью найма на работу, а также проведения тематических исследований, а журналистам – находить подходящие вакансии и проекты.

1 Анализ предметной области и постановка задачи

Web-платформа подбора журналистов для тематических исследований предназначена для автоматизирования и оптимизирования процессов поиска и подбора журналистов, обладающих необходимыми навыками и опытом работы не только для деятельности по найму в организациях и издательствах, но и для проведения тематических исследований в разных областях. Платформа должна предоставлять удобный и доступный интерфейс для взаимодействия между работодателями и журналистами, а также предоставлять инструменты для эффективного поиска фильтрации кандидатов.

Основными компонентами для проектирования web-платформы являются реализация базы данных, создание доступного пользовательского интерфейса и использование различных функциональных возможностей.

База данных должна отображать и хранить информацию обо всех зарегистрированных на платформе журналистах, их навыках, опыте работы, образовании и других характеристиках.

Пользовательский интерфейс должен предоставлять возможность регистрации, авторизации, аутентификации поиска и фильтрации кандидатов, а также управления вакансиями.

Функциональные возможности платформы должны реализовывать сопоставления кандидатов с вакансиями, уведомлений, аналитикой и отчетами.

Основные процессы платформы должны включать в себя:

- регистрацию и аутентификацию пользователей;
- создание и управление профилями журналистов;
- поиск и фильтрацию кандидатов по различным критериям;
- сопоставление журналистов с вакансиями;
- управление вакансиями и уведомлениями.

Таким образом, разрабатываемая Web-платформа будет представлять собой эффективное решение для автоматизации и оптимизации процессов подбора журналистов для тематических исследований, способствуя

повышению эффективности взаимодействия между работодателями и специалистами в области журналистики.

1.1 Анализ предметной области

В настоящее время СМИ играют ключевую роль в жизни каждого человека, предоставляя свежую информацию, анализируя различные события и этим самым формируя общественное мнение. Журналисты являются главными действующими лицами в данном процессе, они выполняют важные функции по сбору, обработке и распространению информации.

Роли и обязанности журналистов заключаются в:

- сборе и обработке информации – сбор информации из различных источников, государственных структур, документов, социальных сетей и др., избегая при этом фейковых сообщений;
- анализе и интерпретации – анализируя информацию, журналисты перерабатывают текстовый материал, чтобы предоставить более доступные сведения по поводу событий;
- написании и редактировании – специалисты в СМИ пишут статьи, заметки на разные тематики, снимают репортажи, а также проводят ещё большое количество времени за обработкой и редактированием и корректированием информации, чтобы впоследствии распространить качественные материалы;
- распространении информационного материала – распространение информации происходит по разным источникам – радио, пресса, телевидение, интернет и др.

Немаловажными факторами для журналиста становятся его образование и навыки и компетенции, а также опыт работы.

Необходимые навыки и компетенции представляют собой:

- навыки написания – четкое, доступное и информативное преподнесение новостного материала;

- навыки интервьюирования – умение проводить интервью с интересными и правильными вопросами;
- навыки исследования – умение проводить специализированные исследования и анализировать полученные данные;
- технические навыки – умение работать с различными видами инструментов и технологий, аудио- и видеозапись, фотографирование, а также использование цифровых платформ.

Необходимым базовым образованием для специалиста является образование в сфере журналистики, где обучают различным видам коммуникаций, специализированным и смежным дисциплинам. Также в подготовку журналистов, помимо обучения, включена стажировка, и по желанию, курсы по повышению квалификации.

Все эти характеристики учитываются при проектировании базы данных о журналистах на MySQL и phpMyAdmin.

1.2 Сравнительный анализ средств подбора журналистов по выбранной тематике

В Российской Федерации поиск работы для журналистов через службу занятости населения (Центра занятости) происходит в несколько этапов.

Регистрация в центре занятости:

- журналист ищущий работу должен обратиться в местный Центр занятости населения по месту жительства;
- при себе необходимо иметь следующие документы, а именно: паспорт, трудовую книжку, документы об образовании и квалификации, а также дополнительные документы, подтверждающие профессиональные навыки и опыт работы;

Консультация с специалистом:

- после регистрации журналист встречается с консультантом по трудоустройству, который помогает определить направления поиска работы, оценить квалификацию и опыт работы;

- консультант предлагает пройти профессиональное тестирование или переподготовку;

Поиск вакансий:

- центр занятости предоставляет доступ к базе данных вакансий, где можно найти предложения работы для журналистов;
- консультант помогает подобрать подходящую вакансию, исходя из опыта и квалификации соискателя;

Помощь в трудоустройстве:

- центр занятости может предложить участие в программах поддержки занятости, например: стажировки, подработки и так далее;
- консультант может помочь с подготовкой резюме и сопроводительным письмом;

Содействие в трудоустройстве:

- если журналист находит подходящую вакансию, консультант Центра занятости может оказать содействие в организации собеседования с работодателем;
- в некоторых случаях Центр занятости может предложить финансовую поддержку на период поиска работы;

Поддержка безработных:

- журналист может получать пособие по безработице, если не может сразу найти работу;
- консультант Центра занятости продолжает оказывать поддержку в поиске работы, предлагая новые вакансии и программы.

Таким образом, процесс поиска работы для журналистов в РФ через Службу занятости населения представляет собой многослойную систему, направленную на максимальную поддержку соискателей.

Поиск работы для журналистов на hh.ru (крупнейшей российской компании интернет-рекрумента) происходит в несколько этапов:

Регистрация и создание профиля:

- соискатель должен зарегистрироваться на сайте hh.ru и создать личный профиль;

- в профиле нужно указать личные данные, такие как: контактные данные, опыт работы, образование, навыки и другие профессиональные характеристики.

Заполнение резюме:

- необходимо детально заполнить резюме, указав весь опыт работы, ключевые навыки, достижения и профессиональные цели;
- рекомендуется добавить портфолио с примерами работ, для оценки качества навыков работодателями.

Поиск вакансий:

- используя поиск и фильтр на сайте, журналист может подобрать подходящую под свои требования вакансию.

Отклик на вакансии:

- найдя подходящую вакансию, соискатель может отправить отклик работодателю, предварительно приложив своё резюме и сопроводительное письмо.

Дополнительные функции платформы:

- hh.ru предлагает дополнительные функции, такие как подписка hh PRO (позволяет получить отклик быстрее), готовое резюме (составление резюме специалистами вместо соискателя), сетка (соцсеть нетворкинга в ИТ) и другие.

Обратная связь с работодателями:

- работодатели могут просматривать резюме и отправлять приглашения на собеседования;
- соискатели могут получать сообщения от работодателей через внутреннюю систему сообщений на платформе.

Подготовка к собеседованиям:

- сайт также предлагает помощь в данном вопросе, через дополнительные услуги;
- на платформе журналист может ознакомиться с некоторыми данными компании, выдвигающей свои вакансии.

Редактирование профиля и резюме:

- регулярное обновление профиля и резюме помогает соискателям оставаться видимыми для работодателей и увеличивает шансы на получение интересных вакансий.

Следовательно, платформа hh.ru предлагает удобные инструменты для поиска работы, позволяя журналистам эффективно взаимодействовать с потенциальными работодателями и находить подходящие вакансии в интересующей сфере.

Также, для журналистов ищущих интересные предложения в своей сфере деятельности доступны следующие платформы:

1) MediaJobs – специализированная площадка для размещения вакансий и поиска сотрудников в сферах медиа, диджитала и других: журналисты могут зарегистрироваться на сайте, создать профиль и загрузить свое резюме, после чего им доступны функции просмотра вакансий, отклика на них и получение приглашений на собеседования;

2) LinkedIn – профессиональная иностранная социальная сеть для поиска и установления деловых контактов, ориентированная на поиск работы, развитие бизнеса и других направлений: не является специализированной платформой для поиска и подбора журналистов, но специалисты в данной области могут зарегистрироваться на данном сайте и внести данные в свой профиль, указав необходимые критерии, такие как: опыт работы, навыки и другие; они могут добавляться в группы обсуждений по определенной тематике и принимать в них участие.

3) Союз журналистов России – общероссийская общественная организация, ориентирована на защиту прав и интересов журналистов России:

на сайте для журналистов доступны функции регистрации, поиска работы, повышения квалификации, участия в различных проектах и других проектов;

4) Reddit – представляет собой сайт, сочетающий в себе черты социальной сети и форума: на данном форуме пользователи могут пройти регистрацию, размещать ссылки на какую-либо информацию, а также обсуждать её; журналисты могут участвовать в определенных обсуждениях, просматривать вакансии, публиковать свои резюме; сайт хорошо подходит для нетворкинга и обмена опытом.

Рассмотренные выше платформы, форумы и сайты предоставляют журналистам хорошие возможности не только для поиска работы, но и подходят для установления профессиональных отношений, обмена опытом и возможности быть в курсе последних тенденций и событий в медиа сфере.

1.3 Анализ средств реализации web-платформы

В настоящее время в Web-разработке существует разнообразие инструментов и технологий, которые применяются в создании и управлении Web-приложениями.

Для создания Web-платформы будут использоваться такие инструменты, как MySQL, phpMyAdmin, HTML, Sprinthost, а также WordPress.

MySQL – быстрая, легко настраиваемая, в использовании и администрировании, система управления базами данных, – описывает в своем сборнике Дюбуа П. [Дюбуа П., 2006: 1056с., 15], [1].

Существуют аналоги MySQL, такие как:

- PostgreSQL – сложная система управления реляционными б/д с открытым исходным кодом;
- Microsoft SQL Server – система управления базами данных от Microsoft с широким функционалом;
- Oracle Database – надежная и производительная система управления базами данных от Oracle.

phpMyAdmin – представляет собой Web-приложение с удобным интерфейсом для управления базами данных MySQL.

Аналоги данного Web-приложения:

- Adminer – простое Web-приложение для управления базами данных;
- DBeaver – универсальный инструмент для управления базами данных;
- HeidiSQL – инструмент для управления базами данных;

WordPress представляет собой систему управления контентом - CMS, которая применяется для создания и управления Web-сайтами.

Аналоги CMS:

- Joomla – система управления контентом с большим функционалом для создания Web-сайтов;
- Drupal – сложная и одновременно гибкая и масштабируемая система управления контентом с открытым кодом;
- Wix – платформа для создания Web-сайтов с широким функционалом для дизайна.

Sprinthost – хостинг-провайдер для Web-сайтов и приложений. В разработке Web-платформы используется бесплатный тариф, что позволяет сэкономить и не переплачивать за содержание сайта на хостинге.

Аналоги хостинга:

- Google Cloud – облачная платформа от Google предлагающая различные решения для хостинга;
- Azure – облачная платформа от Microsoft с интегрированными решениями для хостинга и Web-приложений.

HTML – язык гипертекстовой разметки для создания Web-страниц.

Аналогом данного языка является язык разметки Markdown, который используется для создания Web-страниц и документов.

1.4 Анализ требований к Web-платформе

Технические требования представляют собой серверную, клиентскую часть и инфраструктуру.

Серверная часть подразумевает под собой использование базы данных, таких как MySQL (для хранения и управления данными о пользователях, вакансиях и прочих данных) и phpMyAdmin (для управления базами данных MySQL, с удобным интерфейсом для выполнения SQL-запросов) [2].

Клиентская часть представляет собой использование удобного плагина и языка разметки, для структурирования Web-страниц – WordPress (для создания адаптивного и удобного дизайна интерфейса с использованием различных тем) и HTML (для обеспечения качественного дизайна интерфейса с использованием CSS).

Под инфраструктурой понимается развертывание системы на надежном и масштабируемом сервере, таком как Sprinthost, а также использование в нём систем мониторинга и логирования, что обеспечивает стабильность и отказоустойчивость, а также предоставляет инструменты для работы с доменами.

1.4.1 Анализ функциональных требований

Функциональные требования к системе представлены в виде:

- 1) регистрация и аутентификация - возможность регистрации и входа в систему для журналистов и работодателей; восстановление паролей и управления учетными записями;
- 2) управление профилей - создание, редактирование и просмотр профилей журналистов; добавление и обновление информации о навыках, опыте образовании и других характеристик;
- 3) поиск и фильтрация - поиск журналистов по различным критериям; фильтрация результатов поиска по заданным параметрам;

4) управление вакансиями - создание, редактирование и удаление вакансий; просмотр списка вакансий и деталей каждой вакансии;

5) сопоставление кандидатов - автоматическое сопоставление журналистов с вакансиями на основе их навыков, опыта и смежных критериев; настройка параметров сопоставления вручную;

6) уведомления - отправка уведомлений журналистам о новых вакансиях; отправка уведомлений работодателям о новых кандидатах;

7) аналитика и отчетность - генерация отчетов о процессе подбора и эффективности платформы; анализ данных для улучшения процессов подбора.

В UML–диаграмме вариантов использования кратко отображается информация о пользователях разрабатываемой системы и их взаимодействиях с ней. Она демонстрирует возможные действия пользователей с созданной базой данных, например: создание, изменение или удаление документов, исправление данных, добавление новых элементов или справочников. Диаграмма вариантов использования представлена на рисунке 1.

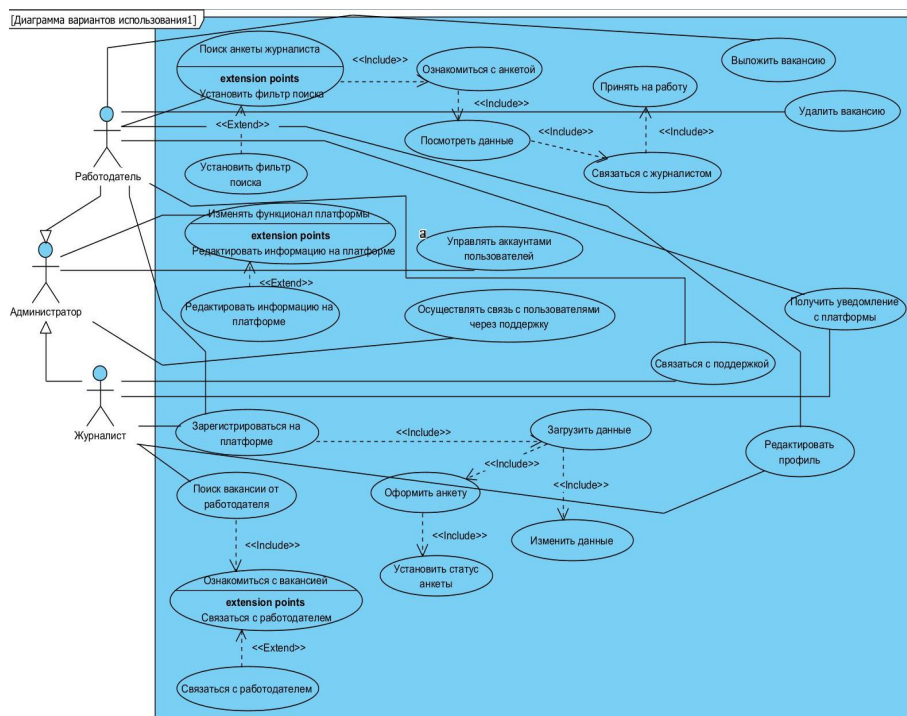


Рисунок 1 – Диаграмма вариантов использования

Описание актеров диаграммы вариантов использования представлено в таблице 1.

Таблица 1 – Описание актеров

Название	Профиль, подготовка и навыки
Администратор	Пользователь информационной системы с обширным опытом работы с программным обеспечением и полным доступом ко всем функциям системы.
Работодатель	Пользователь имеет опыт работы с узким функционалом, доступ к справочной информации платформы и документам, отвечающих за найм журналистов.
Журналист	Пользователь имеет доступ к информации о работодателях, лично загруженных документах.

Краткое описание прецедентов представлено в таблице 2.

Таблица 2 – Описание прецедентов

Действующее лицо	Цель	Краткое описание
Работодатель	Зарегистрироваться на платформе	Должен зарегистрироваться на платформе, чтобы увидеть анкеты журналистов и выкладывать вакансии.
	Поиск анкеты журналиста	Может осуществлять поиск анкет журналистов на платформе.
	Установить фильтр поиска	Может воспользоваться специальным фильтром для нахождения более точного результата поиска анкет.
	Ознакомиться с анкетой	Может ознакомиться с анкетами журналистов.
	Посмотреть данные	Может просмотреть загруженные файлы журналистом на сайт.
	Связаться с журналистом	Может связаться с журналистом после просмотра анкеты, на усмотрение работодателя.
	Принять на работу	Может принять журналиста на работу, на усмотрение работодателя.
	Выложить вакансию	Может выложить вакансию на платформу.
	Удалить вакансию	Может удалить вакансию с платформы.
	Редактировать профиль	Может изменять данные своего профиля.
	Связаться с поддержкой	Может связаться с поддержкой по возникшим проблемам работы с платформой.
Журналист	Получить уведомление с платформы	Может настроить получение уведомлений с платформы.
	Зарегистрироваться на платформе	Должен зарегистрироваться на платформе, чтобы создать анкету.
	Загрузить данные	Может загрузить необходимые данные в формате PDF файла.

Продолжение таблицы 2

Действующее лицо	Цель	Краткое описание
	Оформить анкету	Может оформить анкету на платформе для поиска работы.
	Изменить данные	Может изменить данные анкеты.
	Установить статус анкеты	Может установить статус анкеты.
	Поиск вакансии от работодателя	Может осуществлять поиск вакансий, размещенных на платформе работодателями.
	Ознакомиться с вакансией	Может ознакомиться с интересующей вакансией.
	Связаться с работодателем	Может связаться с работодателем.
	Редактировать профиль	Может редактировать свой профиль.
	Связаться с поддержкой	Может связаться с поддержкой, если возникнут трудности в обращении с платформой.
	Получить уведомление с платформы	Может настроить получение уведомлений с платформы.
Администратор	Изменять функционал платформы	Может изменять функционал платформы.
	Редактировать информацию на платформе	Может редактировать информацию на платформе.
	Управлять аккаунтами пользователей	Может управлять аккаунтами пользователей на платформе.
	Осуществлять связь с пользователями через поддержку	Может осуществлять связь с пользователями через поддержку платформы.

Полное описание актеров и сценариев использования представлено в таблицах 3.

Таблица 3 – Описание актера «Работодатель» и сценариев использования

Действующее лицо	Описание актера	Сценарий использования	Описание сценария использования
Работодатель	Актер работодатель представляет собой пользователя, которому доступны определенные наборы функций на Web-платформе.	Зарегистрироваться на платформе	Сценарий использования представляет собой функцию, позволяющую пользователю пройти регистрацию на платформе. В последствии окно регистрации сменится на окно входа на платформу.
		Поиск анкеты журналиста	Сценарий использования представляет собой функцию, позволяющую пользователю искать опубликованные анкеты журналистов на сайте.

Продолжение таблицы 3

Действующее лицо	Описание актера	Сценарий использования	Описание сценария использования
		Установить фильтр поиска	Сценарий использования представляет собой функцию, позволяющую пользователю устанавливать фильтр для более удобного и качественного поиска анкет журналистов. В фильтрах можно указать необходимые критерии город, стаж работы и образование журналиста. Фильтр поиска поможет вывести более точные анкеты, которые задаст пользователь с помощью указанных критериев.
		Ознакомиться с анкетой	Сценарий использования представляет собой функцию, позволяющую пользователю ознакомиться с опубликованными анкетами журналистов на платформе.
		Посмотреть данные	Сценарий использования представляет собой функцию, позволяющую пользователю ознакомиться с данными опубликованных анкет журналистов на платформе, а также посмотреть и изучить приложенные ими документы в формате PDF.
		Связаться с журналистом	Сценарий использования представляет собой функцию, позволяющую пользователю связаться с журналистом с помощью платформы, либо осуществить связь через предоставленные контакты журналистом.
		Принять на работу	Сценарий использования представляет собой функцию, позволяющую пользователю осуществить прием на работу журналиста, предварительно ознакомившись с его анкетой и связавшись с ним.
		Выложить вакансию	Сценарий использования представляет собой функцию, позволяющую пользователю выложить вакансию на

Окончание таблицы 3

Действующее лицо	Описание актера	Сценарий использования	Описание сценария использования
			платформу по поиску и найму журналиста.
		Удалить вакансию	Сценарий использования представляет собой функцию, позволяющую пользователю удалить вакансию с платформы по поиску и найму журналиста.
		Редактировать профиль	Сценарий использования представляет собой функцию, позволяющую пользователю редактировать свой профиль и вносить какие-либо необходимые изменения.
		Связаться с поддержкой	Сценарий использования представляет собой функцию, позволяющую пользователю обратиться в поддержку платформы для решения возникших проблем.
		Получить уведомление с платформы	Сценарий использования представляет собой функцию, позволяющую пользователю настроить получение уведомлений с платформы в личном кабинете.

Полное описание актеров и сценариев использования представлено в таблицах 4.

Таблица 4 – Описание актера «Журналист» и сценариев использования

Действующее лицо	Описание актера	Сценарий использования	Описание сценария использования
Журналист	Актер журналист представляет собой пользователя, которому доступны определенные наборы функций на Web-платформе.	Зарегистрироваться на платформе	Сценарий использования представляет собой функцию, позволяющую пользователю пройти регистрацию на платформе. В последствии окно регистрации сменится на окно входа на платформу.
		Загрузить данные	Сценарий использования представляет собой функцию, позволяющую пользователю загрузить необходимые данные формате PDF на платформу.
		Оформить анкету	Сценарий использования представляет собой функцию, позволяющую пользователю оформить свою анкету перед

Продолжение таблицы 4

Действующее лицо	Описание актера	Сценарий использования	Описание сценария использования
			публикацией на платформе, предварительно указав необходимые характеристики, такие как: навыки, опыт работы, образование, ФИО и другие.
		Изменить данные	Сценарий использования представляет собой функцию, позволяющую пользователю изменять данные в личном профиле и анкете, удалять и загружать документы в формате PDF на платформу и так далее.
		Установить статус анкеты	Сценарий использования представляет собой функцию, позволяющую пользователю устанавливать статус анкеты – в поиске вакансии, анкета заморозена/удалена или скрывать данные анкеты, если необходимая вакансия и трудоустройство было осуществлено.
		Поиск вакансии от работодателя	Сценарий использования представляет собой функцию, позволяющую пользователю осуществлять поиск выложенной вакансии от работодателя на платформе.
		Ознакомиться с вакансией	Сценарий использования представляет собой функцию, позволяющую пользователю ознакомиться с опубликованными вакансиями от работодателей на платформе.
		Связаться с работодателем	Сценарий использования представляет собой функцию, позволяющую пользователю связаться с работодателем через контактную информацию, предоставленную работодателем, либо через платформу.
		Редактировать профиль	Сценарий использования представляет собой функцию, позволяющую пользователю редактировать свой профиль на платформе, вносить необходимые правки.

Окончание таблицы 4

Действующее лицо	Описание актера	Сценарий использования	Описание сценария использования
		Связаться с поддержкой	Сценарий использования представляет собой функцию, позволяющую пользователю связаться с поддержкой платформы, при возникновении каких-либо технических проблем.
		Получить уведомление с платформы	Сценарий использования представляет собой функцию, позволяющую пользователю настраивать уведомления на платформе.

Полное описание актеров и сценариев использования представлено в таблицах 5.

Таблица 5 – Описание актера «Администратор» и сценариев использования

Действующее лицо	Описание актера	Сценарий использования	Описание сценария использования
Администратор	Актер администратор представляет собой пользователя, которому доступны все наборы функций на Web-платформе и в работе с ней.	Изменять функционал платформы	Сценарий использования представляет собой функцию, позволяющую пользователю изменять функционал платформы, расширять её возможности и модернизировать.
		Редактировать информацию на платформе	Сценарий использования представляет собой функцию, позволяющую пользователю редактировать информацию на платформе – изменять текст, блоки и другую структуру сайта, также изменять различные отделы и описания страниц.
		Управлять аккаунтами пользователей	Сценарий использования представляет собой функцию, позволяющую пользователю управлять аккаунтами всех пользователей платформы, а также удалять/блокировать личные кабинеты (замороженные, заброшенные аккаунты), производить чистку пользователей, добавлять новых пользователей (в виде других администраторов).

Заключение таблицы 5

Действующее лицо	Описание актера	Сценарий использования	Описание сценария использования
		Осуществлять связь с пользователями через поддержку	Сценарий использования представляет собой функцию, позволяющую пользователю осуществлять связь с пользователями, работая от имени поддержки платформы – помогать с техническими вопросами платформы, правил пользования платформой, доступом к личным кабинетам пользователей и другими организационными моментами.

Таким образом, в вышеперечисленных таблицах были раскрыты полный функционал и возможности каждого пользователя с работой на Web-платформе.

Администратор обладает полным доступом ко всем функциям системы и имеет обширный опыт работы с программным обеспечением платформы. У данного пользователя представлен самый широкий функционал, что позволяет обеспечивать стабильную и эффективную работу платформы, а также поддерживать её актуальность и безопасность.

Работодатель наделен более узким функционалом, чем администратор платформы, что, в свою очередь, позволяет работодателю эффективно пользоваться платформой без лишней нагрузки, и, способствует сохранению безопасности данных и функционала платформы.

Журналист наделен примерно равными правами с работодателем, что также эффективно сказывается на его работе с функционалом платформы. Все указанные выше функции способствуют более понятному и безопасному использованию платформы, без вмешательства в её техническую часть.

Таким образом, Web-платформа предоставляет комплексное решение для взаимодействия между работодателями и журналистами, обеспечивая удобные и эффективные инструменты поиска работы и подбора персонала.

1.4.2 Анализ нефункциональных требований

Нефункциональные требования должны быть представлены и реализованы в виде:

- 1) производительность: быстрая загрузка и обработка данных; минимальное время отклика при выполнении фильтрации и выполнении поиска вакансий;
- 2) безопасность: защита данных пользователей и обеспечение конфиденциальности; аутентификация и авторизация пользователей;
- 3) масштабируемость: возможность масштабирования системы обработки большого количества пользователей и данных;
- 4) удобство эксплуатации: доступный и удобный интерфейс; поддержка различных устройств и браузеров.

Встроенные системы должны быть разработаны с учетом потребностей и особенностей пользователей.

Учет остроты и цветовой избирательности зрения:

- цветовая схема – интерфейс должен поддерживать различные цветовые схемы, включая высококонтрастные режимы для пользователей с нарушениями зрения; это поможет пользователям с дальтонизмом и другими дефектами зрения.

Поддержка различных языков:

- многоязычность – интерфейс должен поддерживать несколько языков, чтобы пользователи могли выбрать свой родной язык или какие-либо другие (на свой выбор);
- локализация – все элементы интерфейса, включая текст, дату, время и числовые форматы, должны быть локализованы в соответствии с выбранным языком.

Удобство и простота использования:

- интуитивный дизайн – интерфейс должен быть интуитивно понятным и простым в использовании, чтобы пользователи могли легко находить и выполнять необходимые действия;
- навигация – четкая и логичная навигация по платформе, чтобы пользователи могли быстро находить нужные разделы и функции;
- обратная связь – предоставление пользователям обратной связи на их действия, чтобы они были в курсе изменений и действий в системе.

Доступность:

- соблюдение стандартов доступности – интерфейс должен соответствовать стандартам доступности, чтобы обеспечить доступность для пользователей с ограниченными возможностями;
- тестирование с участием пользователей – проведение тестирования интерфейса с участием пользователей с различными потребностями и особенностями для выявления и устранения проблем доступности [7].

Таким образом, реализация этих нефункциональных требований является важным аспектом разработки Web-платформы, обеспечивая её эффективность, безопасность, масштабируемость и удобство использования. Это способствует созданию надежной и удобной системы, которая удовлетворяет потребности как работодателей, так и журналистов.

1.5 Постановка задачи

Разрабатываемая Web-платформа подбора журналистов для тематических исследований предназначена для подбора журналистов по запросам работодателей. Платформа должна облегчить процесс поиска и подбора специалистов, обладающих необходимыми навыками, опытом работы в запрашиваемых сферах. Основной задачей Web-платформы является предоставление доступного и удобного интерфейса для работодателей и специалистов в области исследования, помогающей находить подходящих кандидатов для выполнения специализированных задач.

Платформа должна предоставить возможность работодателям проводить поиск и подбор журналистов, используя любой браузер и интернет соединение. Также, необходимо, чтобы данный сайт мог отображать вакансии журналистов, их опыт, образование, навыки и примеры работ в формате PDF. Платформа должна получать с сервера запросы с описанием требований к кандидатам, которые будут отображать все необходимые данные для подбора.

Платформа должна вести отчетность о произведенных действиях по подбору журналистов, осуществлять анализ навыков и опыта специалистов, после чего опрашивать данные на сервер. Это позволило бы работодателям эффективно находить и подбирать работников по своим запросам, а журналистам – находить подходящие вакансии.

1.6 Планирование и расчет бюджета

При разработке программного обеспечения необходимо начать с проектирования, то есть полностью спланировать необходимую разработку, установить сроки выполнения задачи, определить исходные данные и ожидаемый результат.

Современные методики разработки программного обеспечения включают разнообразные подходы, которые помогают структурировать процесс и сделать все этапы более доступными для понимания.

Проектное планирование подразумевает под собой определение ключевых задач, установление сроков их выполнения и назначение ответственных сотрудников. Перечень задач представлен на рисунке 2.

#ID	Task Name	Start	End	Duration	Resources
1	Начало проекта	2025-05-01	2025-05-01	1 day	
2	▼ Анализ требований и постановка задачи	2025-05-02	2025-05-07	4 days	Аналитик
3	Анализ предметной области	2025-05-02	2025-05-06	3 days	Аналитик
4	Сравнение программ-аналогов	2025-05-05	2025-05-06	2 days	Аналитик
5	Постановка задачи	2025-05-06	2025-05-07	2 days	Аналитик
6	▼ Проектирование программных средств	2025-05-06	2025-05-13	6 days	
7	Проектирование архитектуры системы	2025-05-06	2025-05-08	3 days	Программист 1, Аналитик
8	Проектирование базы данных	2025-05-07	2025-05-09	3 days	Программист 1, Аналитик
9	Проектирование пользовательского интерфейса	2025-05-12	2025-05-13	2 days	Программист 2
10	▼ Реализация	2025-05-14	2025-05-20	5 days	
11	Реализация серверной части приложения	2025-05-14	2025-05-20	5 days	Программист 1
12	Реализация клиентской части приложения	2025-05-14	2025-05-20	5 days	Программист 2
13	▼ Тестирование	2025-05-20	2025-05-29	8 days	
14	Тестирование серверной части	2025-05-20	2025-05-29	8 days	Программист 1, Аналитик
15	Тестирование клиентской части	2025-05-20	2025-05-29	8 days	Программист 2, Аналитик
16	Составление и оформление документации	2025-05-26	2025-05-29	4 days	Аналитик
17	Конец проекта	2025-05-30	2025-05-30	1 day	

Рисунок 2 – Главные задачи проекта

Для моделирования программных решений был использован язык UML – унифицированный язык моделирования, предназначенный для описания, визуализации и документирования и объектно-ориентированных систем на этапах их анализа и проектирования.

Диаграмма Ганта представляет собой инструмент управления проектами, который используется для визуализации графика выполнения задач в рамках проекта. Эта диаграмма наглядно визуализирует временные рамки, что упрощает восприятие проекта, помогает координировать работу между различными этапами проекта и обеспечивает своевременное выполнение задач.

На рисунке 3 представлена Диаграмма Ганта, полученная в результате проектирования.

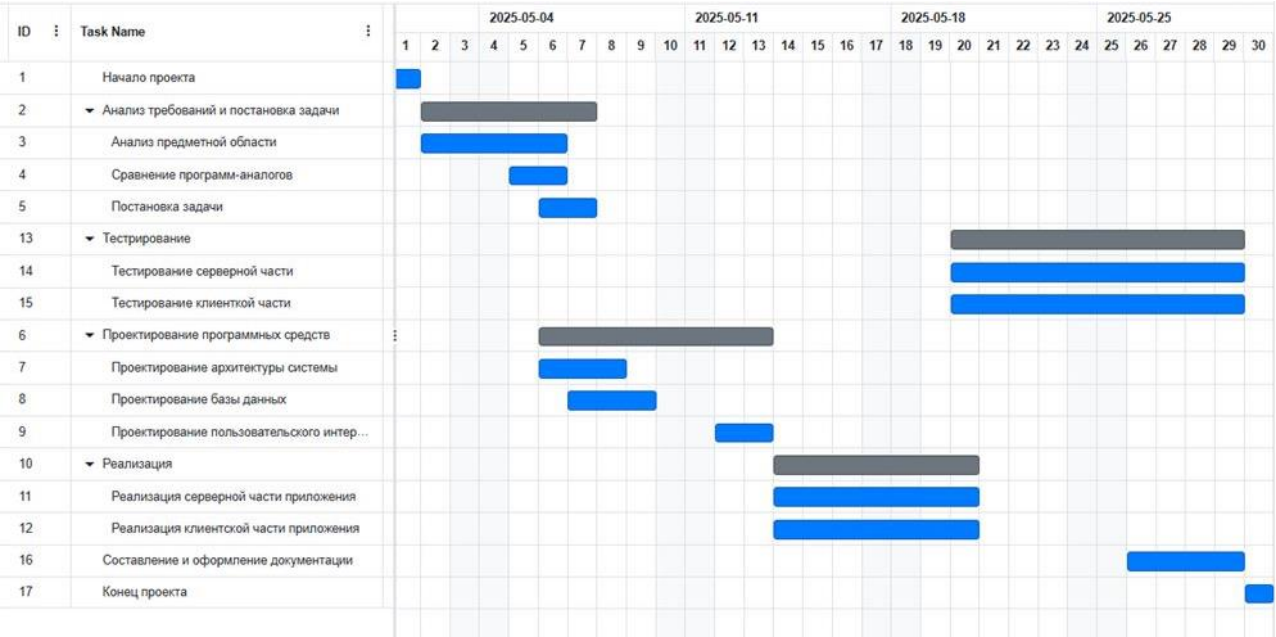


Рисунок 3 - Диаграмма Ганта

На диаграмме Ганта, представленной на рисунке 4, отображены задачи и даты. Задачи представлены горизонтальной полосой, которая начинается с даты начала производственной (преддипломной) практики и завершается датой её окончания. Благодаря этому, можно отчётливо увидеть, как задачи распределены во времени, какие задачи выполняются параллельно, а какие последовательно.

Таблица 6 – Список ресурсов

Наименование ресурсов	Тип	Макс. един.	Ставки		Начисление
			Стандартная	Сверхурочная	
Аналитик	Трудовой	100%	1000.00р/час	2000.00р/час	Пропорциональн ое
Программист	Трудовой	100%	1800.00р/час	3600.00р/час	Пропорциональн ое

Расчет для аналитика:

- 1) Стоимость стандартных часов – $70\text{ч} \cdot 1000\text{руб/час} = 70\text{тыс.руб.}$
- 2) Стоимость сверхурочных часов – $10\text{ч} \cdot 2000\text{руб/ч} = 20\text{тыс.руб.}$
- 3) Общая стоимость для аналитика - $70\text{тыс.руб.} + 20\text{тыс.руб.} = 90\text{тыс.руб.}$

Расчет для программиста:

- 1) Стоимость стандартных часов – $100\text{ч} \cdot 1800\text{руб/ч} = 180\text{тыс.руб.}$
- 2) Стоимость сверхурочных часов – $20\text{ч} \cdot 3600\text{руб/ч} = 72\text{тыс.руб.}$
- 3) Общая стоимость для программиста – $180\text{тыс.руб.} + 72\text{тыс.руб.} = 252\text{тыс.руб.}$

Общая стоимость проекта составила 342тыс.рублей, общее количество часов работы составило 200 единиц.

2 Проектирование Web-платформы подбора журналистов

2.1 Проектирование архитектуры приложения

Концептуальное проектирование базы данных является начальным этапом проектирования базы данных, на котором определяется общая структура и содержание базы данных без привязки к конкретной системе управления базами данных (СУБД) или техническим деталям реализации.

Этот этап включает в себя несколько ключевых задач и процессов:

1) Определение требований – сбор и анализ требований к базе данных от заинтересованных сторон, таких как пользователи, разработчики и бизнес-аналитики, и, постановка целей и задач, которые должна решать база данных.

2) Идентификация сущностей – определение основных объектов (сущностей), которые будут представлены в базе данных, а также атрибутов (характеристики/свойства сущностей) для каждой сущности.

3) Определение связей между сущностями – определение и описание отношений между сущностями, могут быть представлены различные связи («многие ко многим», «один ко многим», «часть-целое» и другие).

4) Построение концептуальной модели – создание концептуальной модели данных, которая визуально представляет сущности, их атрибуты и связи между ними.

5) Нормализация данных – применение правил нормализации для устранения избыточности данных и аномалий обновления, включает в себя декомпозицию таблиц для минимизации дублирования данных и обеспечения целостности данных.

6) Документирование – создание документации, описывающей концептуальную модель данных, включая описание сущностей, атрибутов и связей, а также любые предложения и ограничения, которые были сделаны в процессе проектирования.

Поэтому концептуальное проектирование базы данных является важным этапом, так как оно закладывает основу для последующих этапов проектирования, таких как логическое и физическое проектирование, и обеспечивает четкое понимание структуры данных и их взаимосвязей.

В данной дипломной работе было использовано проектирование на концепции клиент-серверной архитектуры. Для лучшего представления архитектуры была реализована схема клиент-серверной архитектуры, представленная на рисунке 4.

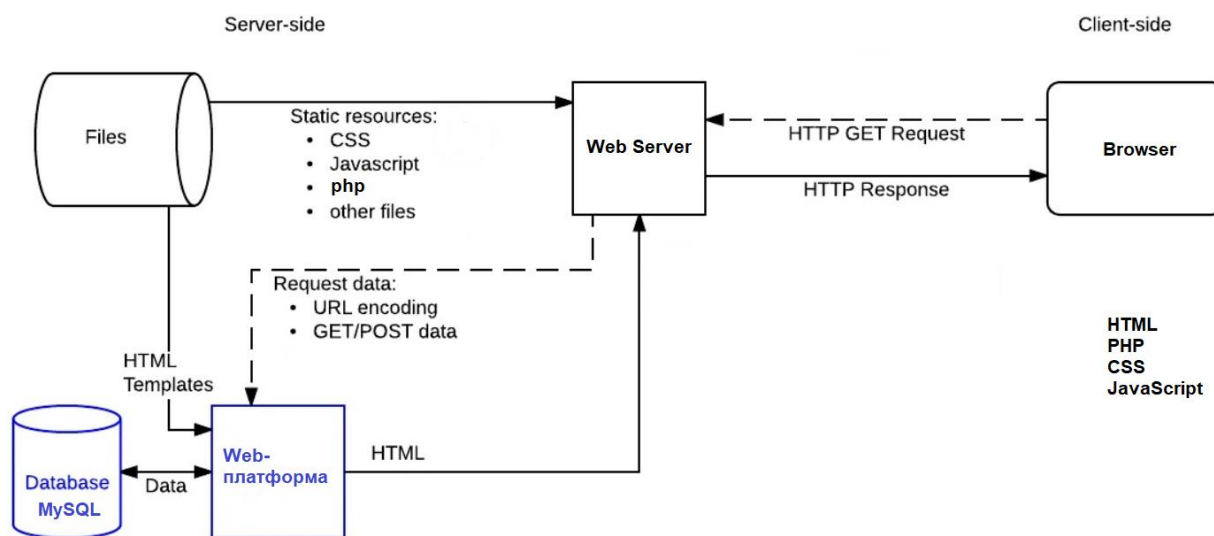


Рисунок 4 – Схема клиент-серверной архитектуры

В этом случае клиентом является Web-платформа (frontend), а сервером (backend) – база данных и логика обработки запросов.

На этапе проектирования Web-платформы используется бесплатный тариф хостинга Sprinthost, движок WordPress и загрузка базы данных на phpMyAdmin.

Sprinthost является хостинг-провайдером для Web-платформы.

WordPress представляет собой систему управления контентом - CMS, и используется для подгрузки шаблона темы на Web-платформу.

phpMyAdmin подразумевает под собой Web-приложение с удобным интерфейсом для управления базами данных MySQL.

Диаграмма классов является одним из ключевых элементов объектно-ориентированного проектирования и используется в основном для

визуализации структуры системы. Она помогает разработчикам и аналитикам понять, как различные компоненты системы взаимодействуют друг с другом.

Для лучшего понимания, проектирования и реализации системы была создана диаграмма классов, которая представлена на рисунке 5.

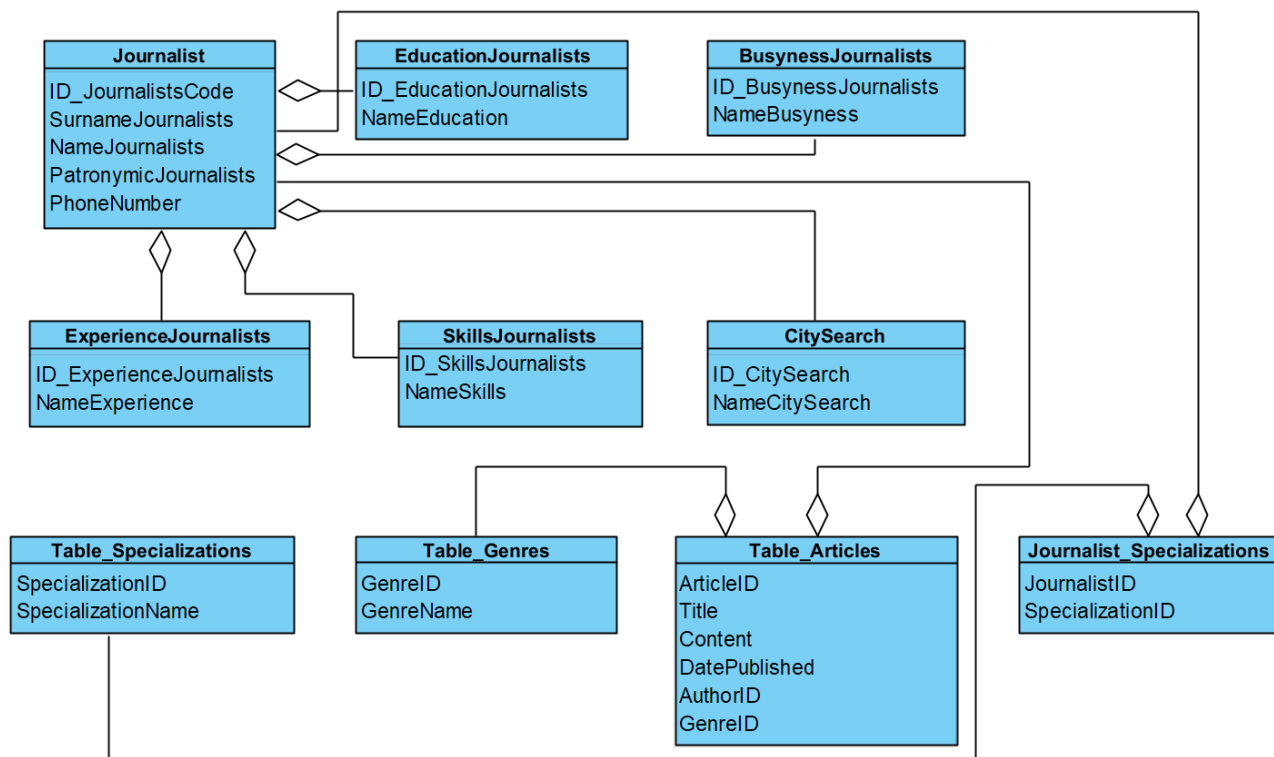


Рисунок 5 – Диаграмма классов

На диаграмме классов представлены несколько основных классов, каждый из которых соответствует таблице в базе данных. Эти классы будут иметь атрибуты, соответствующие столбцам таблиц, а также связи между классами, отражающие отношения между таблицами.

В UML-диаграммах классов связь агрегация используется для обозначения «часть-целое» отношения между классами. Агрегация показывает, что один класс (агрегатор) содержит и включает другие классы (агрегируемые), но при этом эти классы могут существовать независимо друг от друга. Это означает, что жизнь агрегируемых объектов не зависит от агрегатора.

Таким образом, агрегация позволяет моделировать структуры, где объекты имеют частичное взаимодействие и взаимозависимость, но при этом могут существовать.

2.2 Проектирование базы данных

Проектирование базы данных на MySQL Server для Web-платформы включает в себя несколько основных этапов, обеспечивающие создание структуры базы данных, которая, в свою очередь, будет эффективно поддерживать функциональность платформы.

Анализ требований:

- определение целей и задач – этап, который поможет понять, какие данные будут храниться и обрабатываться (для данной базы данных характерны данные о журналистах, их навыках, опыте, образовании, занятости и другие);
- сбор требований – взаимодействие с заинтересованными сторонами для определения функциональных и нефункциональных требований к системе.

Концептуальное проектирование:

- определение сущностей – идентификация основных объектов, таких как «Журналист», «Навыки журналиста», «Образование журналиста», «Опыт журналиста», «Опыт журналиста», «Занятость журналиста» и «Город»;
- определение атрибутов – для сущности «Журналист» это ФИО, опыт работы, навыки, образование, номер телефона и другие;
- определение связей – определение отношений между сущностями (для данной базы данных используется агрегация, то есть связь «часть-целое»).

Логическое проектирование:

- создание логической модели данных – преобразование концептуальной модели в логическую модель, которая включает таблицы, столбцы и связи между ними;
- нормализация данных – применение правил нормализации для минимизации избыточности данных и обеспечения целостности данных (включает декомпозицию таблиц и создание между ними связей).

Физическое проектирование:

- определение структуры таблиц – создание таблиц в базе данных с учетом требований к производительности и хранению данных;
- определение индексов – создание индексов для ускорения выполнения запросов;
- определение ограничений – настройка ограничений целостности данных, такие как первичные (Primary key) и внешние ключи (Foreign key), уникальные ограничения (Unique), проверки значений (Check) и ограничение NOT NULL (гарантирует столбцы без значений NULL, которая используется в данной базе данных).

Реализация базы данных:

- создание базы данных и таблиц – использование SQL-запросов для создания таблиц в базе данных на MySQL Server;
- настройка связей – определение и настройка связей между таблицами с использованием внешних ключей;
- наполнение базы данных – загрузка начальных данных и настройка процедур для управления данными.

Выше были описаны необходимые этапы проектирование для создания удобной и надежной базы данных на MySQL Server для Web-платформы по подбору журналистов и тематических исследований.

Для более подробного описания процесса создания базы данных была спроектирована таблица атрибутов сущностей.

Таблица 7 – Атрибуты сущностей

№ п/п	Наименование атрибута	Тип	Тип ключа	Ограничения	
				Nullable	Unique
1	2	3	4	5	
ЖУРНАЛИСТ					
1	ID_JournalistsCode	integer	PK	NOT NULL	НЕТ
2	SurnameJournalists	char(30)	-	NOT NULL	НЕТ
3	NameJournalists	char(30)	-	NOT NULL	НЕТ
4	PatronymicJournalists	char(30)	-	NOT NULL	НЕТ
5	PhoneNumber	char(11)	-	NOT NULL	НЕТ
НАВЫКИ ЖУРНАЛИСТА					
1	ID_SkillsJournalists	integer	PK	NOT NULL	НЕТ
2	NameSkills	char(30)	-	NOT NULL	НЕТ

Продолжение таблицы 7

№ п/п	Наименование атрибута	Тип	Тип ключа	Ограничения	
				Nullable	Unique
1	2	3	4	5	
ОБРАЗОВАНИЕ ЖУРНАЛИСТА					
1	ID_EducationJournalists	integer	PK	NOT NULL	НЕТ
2	NameEducation	char(30)	-	NOT NULL	НЕТ
ОПЫТ ЖУРНАЛИСТА					
1	ID_ExperienceJournalists	integer	PK	NOT NULL	НЕТ
2	NameExperience	char(30)	-	NOT NULL	НЕТ
ЗАНЯТОСТЬ ЖУРНАЛИСТА					
1	ID_BusynessJournalists	integer	PK	NOT NULL	НЕТ
2	NameBusyness	char(30)	-	NOT NULL	НЕТ
ГОРОД					
1	ID_CitySearch	integer	PK	NOT NULL	НЕТ
2	NameCitySearch	char(30)	-	NOT NULL	НЕТ
СПЕЦИАЛИЗАЦИЯ					
1	SpecializationID	integer	PK	NOT NULL	НЕТ
2	SpecializationName	char(100)	-	NOT NULL	ДА
ЖАНРЫ					
1	GenreID	integer	PK	NOT NULL	НЕТ
2	GenreName	char(100)	-	NOT NULL	ДА
СТАТЬИ					
1	ArticleID	integer	PK	NOT NULL	НЕТ
2	Title	char(255)	-	NOT NULL	НЕТ
3	Content	char(255)	-	NOT NULL	НЕТ
4	DatePublished	date	-	NOT NULL	НЕТ
5	AuthorID	integer	FK	NOT NULL	НЕТ
6	GenreID	integer	FK	NOT NULL	НЕТ
СПЕЦИАЛИЗАЦИЯ И ЖУРНАЛИСТЫ					
1	JournalistID	integer	FK	NOT NULL	НЕТ
2	SpecializationID	integer	FK	NOT NULL	НЕТ

В представленной выше таблице описываются атрибуты сущностей базы данных, предназначенной для управления информацией о журналистах и вакансиях. Эта таблица содержит данные о различных аспектах, связанных с журналистами, включая их навыки, опыт, образование, занятость и местоположение (город проживания).

Колонка наименование атрибута представляет собой имя поля в базе данных.

Колонка тип – тип данных атрибутов, который хранится в этом поле.

Колонка тип ключа указывает, является ли атрибут первичным ключом (PK) или внешним ключом (FK).

Колонка ограничения представляет собой ограничения, которые наложены на атрибуты, такие как NOT NULL – не может быть пустым.

В строках заглавными буквами указаны сущности (названия таблиц) базы данных.

Благодаря таблице атрибутов сущностей можно сделать вывод о том, что база данных спроектирована для хранения и управления информацией о журналистах и их анкетах. Каждая сущность имеет уникальные идентификаторы, что обеспечивает целостность данных, их согласованность и возможность выполнения сложных запросов.

2.2.1 Концептуальное проектирование базы данных

(Entity Relationship) – это диаграмма отношений сущностей, которая используется для моделирования данных в базе данных, показывает логическую структуру данных через определение сущностей, их атрибутов и отношений между ними.

Диаграмма «сущность-связь» является визуальным представлением концептуальной информационной модели, которое используется для описания структуры данных и взаимосвязей между ними в проектировании баз данных.

На основе анализа предметной области были определены классы объектов предметной области.

Были определены свойства (поля, атрибуты) классов информационной модели предметной области, методы классов уровня бизнес-процессов и процессов управления, ассоциации между классами.

Была построена диаграмма сущность-связь концептуальной информационной модели Web-платформы в терминах, которая представлена на рисунке 6.

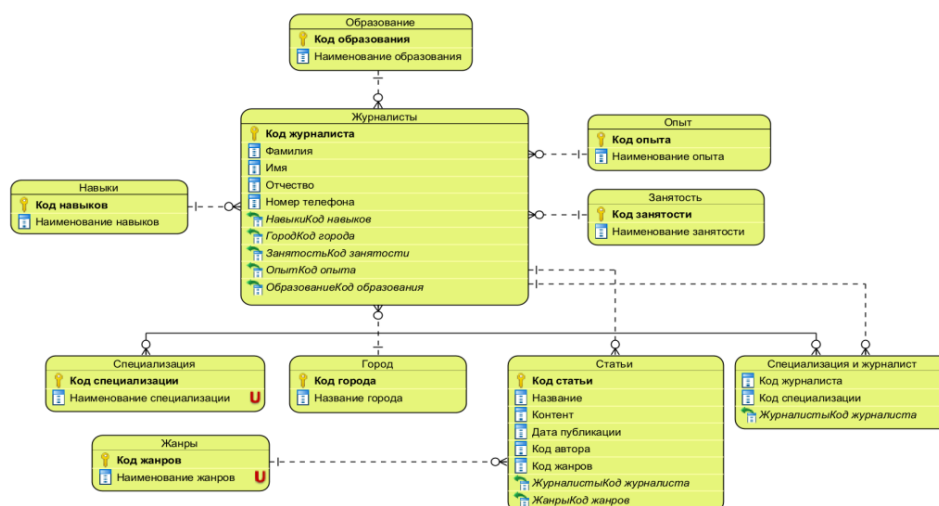


Рисунок 6 – Диаграмма сущность-связь концептуальной информационной модели Web-платформы

В данной ER-диаграмме присутствует связь между таблицами «многие к одному» и означает, что множество экземпляров одного класса связаны с одним экземпляром другого класса. Данный вид связи чаще всего используется при проектировании реляционных баз данных.

Для наглядности представления концептуальной информационной модели была составлена таблица спецификации классов.

Таблица 8 – Спецификация классов

№ п/п	Наименование класса	Тип	Тип ключа	Ограничения	
				Nullable	Unique
1	2	3	4	5	
ЖУРНАЛИСТЫ					
1	Код журналиста	integer	PK	NOT NULL	HET
2	Фамилия	char(30)	-	NOT NULL	HET
3	Имя	char(30)	-	NOT NULL	HET
4	Отчество	char(30)	-	NOT NULL	HET
5	Номер телефона	char(11)	-	NOT NULL	HET
6	НавыкиКод навыков	integer	FK	NULL	HET
7	ГородКод города	integer	FK	NULL	HET
8	ЗанятостьКод занятости	integer	FK	NULL	HET
9	ОпытКод опыта	integer	FK	NULL	HET
10	ОбразованиеКод образования	char(11)	FK	NULL	HET
НАВЫКИ					
1	Код навыков	integer	PK	NOT NULL	HET
2	Наименование навыков	char(30)	-	NOT NULL	HET
ОБРАЗОВАНИЕ					
1	Код образования	integer	PK	NOT NULL	HET
2	Наименование образования	char(30)	-	NOT NULL	HET

Продолжение таблицы 8

№ п/п	Наименование класса	Тип	Тип ключа	Ограничения	
				Nullable	Unique
1	2	3	4	5	
ОПЫТ					
1	Код опыта	integer	PK	NOT NULL	НЕТ
2	Наименование опыта	char(30)	-	NOT NULL	НЕТ
ЗАНЯТОСТЬ					
1	Код занятости	integer	PK	NOT NULL	НЕТ
2	Наименование занятости	char(30)	-	NOT NULL	НЕТ
ГОРОД					
1	Код города	integer	PK	NOT NULL	НЕТ
2	Название города	char(30)	-	NOT NULL	НЕТ
СПЕЦИАЛИЗАЦИЯ					
1	Код специализации	integer	PK	NOT NULL	НЕТ
2	Наименование специализации	char(100)	-	NOT NULL	ДА
ЖАНРЫ					
1	Код жанров	integer	PK	NOT NULL	НЕТ
2	Наименование жанров	char(100)	-	NOT NULL	ДА
СТАТЬИ					
1	Код статьи	integer	PK	NOT NULL	НЕТ
2	Название	char(255)	-	NOT NULL	НЕТ
3	Контент	char(255)	-	NOT NULL	НЕТ
4	Дата публикации	date	-	NOT NULL	НЕТ
5	Код автора	integer	FK	NOT NULL	НЕТ
6	Код жанров	integer	FK	NOT NULL	НЕТ
СПЕЦИАЛИЗАЦИЯ И ЖУРНАЛИСТЫ					
1	Код журналиста	integer	FK	NOT NULL	НЕТ
2	Код специализации	integer	FK	NOT NULL	НЕТ

Колонка наименование класса в данной таблице представляет собой имя поля в диаграмме сущность-связь концептуальной информационной модели Web-платформы.

Столбец тип является типом данных классов, которые хранятся в этом поле.

Колонка тип ключа указывает, является ли класс первичным ключом (PK) или внешним ключом (FK).

Колонка ограничения представляет собой ограничения, которые наложены на классы, такие как NOT NULL – не может быть пустым и NULL, который в свою очередь является противоположным значением ограничения NOT NULL, то есть значения могут быть пустыми, а также Unique – уникальный (со значениями ДА или НЕТ).

2.2.2 Логическое проектирование базы данных

Логическая схема базы данных представляет собой абстрактное представление структуры базы данных, которое, в свою очередь, описывает организацию данных и связи между ними, без привязки к конкретной системе управления базами данных. Также она является промежуточным этапом между концептуальной моделью данных и физической реализацией базы данных. Логическая диаграмма представлена на рисунке 7.

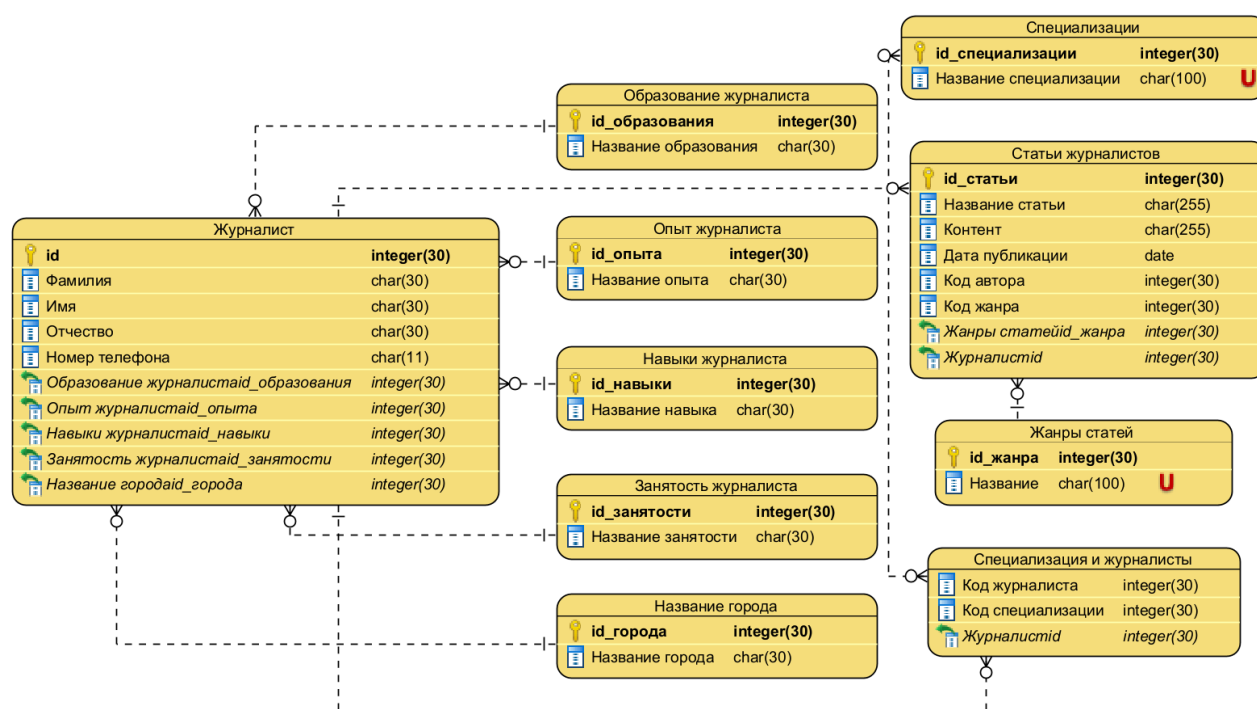


Рисунок 7 – Логическая схема базы данных информационной модели Web-платформы

В логической схеме базы данных присутствует связь между таблицами «многие к одному» и означает, что множество экземпляров одного класса связаны с одним экземпляром другого класса. Данный вид связи чаще всего используется при проектировании реляционных баз данных.

Также, в данной ER-диаграмме указываются типы и размерности типов атрибутов, что помогает наглядно ознакомиться с компонентами базы данных.

Для наглядности представления логической схемы базы данных информационной модели Web-платформы была создана таблица - Атрибуты классов логической схемы базы данных.

Таблица 9 – Атрибуты классов логической схемы базы данных

№ п/п	Наименование классов	Тип	Тип ключа	Ограничения	
				Nullable	Unique
1	2	3	4	5	
ЖУРНАЛИСТ					
1	id	integer(30)	PK	NOT NULL	НЕТ
2	Фамилия	char(30)	-	NOT NULL	НЕТ
3	Имя	char(30)	-	NOT NULL	НЕТ
4	Отчество	char(30)	-	NOT NULL	НЕТ
5	Номер телефона	char(11)	-	NOT NULL	НЕТ
6	Навыки журналиста id_навыки	integer(30)	FK	NULL	НЕТ
7	Опыт журналиста id_опыта	integer(30)	FK	NULL	НЕТ
8	Образование журналиста id_образования	char(11)	FK	NULL	НЕТ
НАВЫКИ ЖУРНАЛИСТА					
1	id_навыки	integer(30)	PK	NOT NULL	НЕТ
2	Наименование навыков	char(30)	-	NOT NULL	НЕТ
ОБРАЗОВАНИЕ ЖУРНАЛИСТА					
1	id_образования	integer(30)	PK	NOT NULL	НЕТ
2	Наименование образования	char(30)	-	NOT NULL	НЕТ
ОПЫТ ЖУРНАЛИСТА					
1	id_опыта	integer(30)	PK	NOT NULL	НЕТ
2	Наименование опыта	char(30)	-	NOT NULL	НЕТ
ЗАНЯТОСТЬ ЖУРНАЛИСТА					
1	id_занятости	integer(30)	PK	NOT NULL	НЕТ
2	Наименование занятости	char(30)	-	NOT NULL	НЕТ
ГОРОД					
1	id_города	integer(30)	PK	NOT NULL	НЕТ
2	Название города	char(30)	-	NOT NULL	НЕТ
СПЕЦИАЛИЗАЦИЯ					
1	id_специализации	integer(30)	PK	NOT NULL	НЕТ
2	Название специализации	char(100)	-	NOT NULL	ДА
ЖАНРЫ					
1	id_жанров	integer(30)	PK	NOT NULL	НЕТ
2	Название	char(100)	-	NOT NULL	ДА
СТАТЬИ					
1	id_статьи	integer(30)	PK	NOT NULL	НЕТ
2	Название статьи	char(255)	-	NOT NULL	НЕТ
3	Контент	char(255)	-	NOT NULL	НЕТ
4	Дата публикации	date	-	NOT NULL	НЕТ
5	Код автора	integer(30)	FK	NOT NULL	НЕТ
6	Код жанра	integer(30)	FK	NOT NULL	НЕТ
СПЕЦИАЛИЗАЦИЯ И ЖУРНАЛИСТЫ					
1	Код журналиста	integer(30)	FK	NOT NULL	НЕТ
2	Код специализации	integer(30)	FK	NOT NULL	НЕТ

В колонках таблицы приводятся названия сущностей, типы данных классов, типы ключей (PK или FK), а также виды ограничений Nullable (со

значениями NOT NULL или NULL) и Unique - уникальный (со значениями ДА или НЕТ).

Колонка тип ключа указывает, является ли класс первичным ключом (PK) или внешним ключом (FK).

Колонка ограничения представляет собой ограничения, которые наложены на классы, такие как NOT NULL – не может быть пустым и NULL – значения могут быть пустыми.

Также одной из важных значений при проектировании базы данных является её нормальная форма.

Третья нормальная форма (3NF) представляет собой уровень нормализации базы данных, устраняющий транзитивные зависимости. База данных является 3NF, если она находится во второй нормальной форме 2NF (уровень нормализации, устраняющий частичные зависимости в данных), и, если ни один неключевой атрибут не зависит транзитивно от первичного ключа.

2.2.3 Физическое проектирование базы данных

Физическая схема базы данных – это детальное описание данных, которые будут фактически храниться в СУБД. В неё входят конкретные детали реализации, такие как определение таблиц, столбцов, индексов, типов данных и других различных аспектов, которые связаны с физическим хранением данных.

Для лучшего представления физического проектирования базы данных была создана физическая схема базы данных, представленная на рисунке 8.

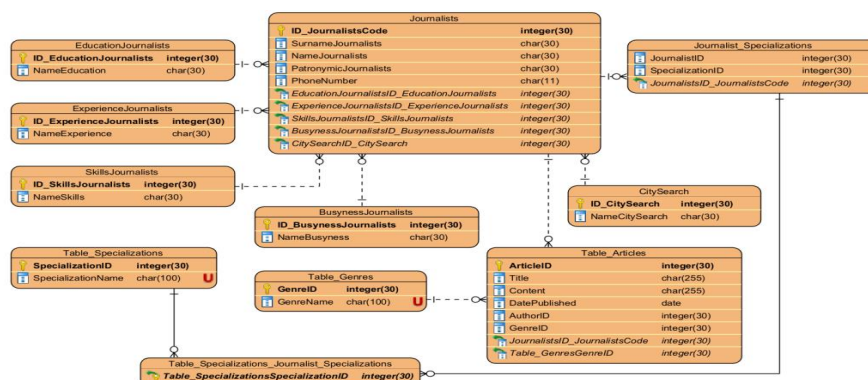


Рисунок 8 – Физическая схема базы данных

В физической схеме базе данных изображены таблицы, которые будут непосредственно указаны в базе данных MySQL Server. Каждая таблица соответствует сущности из логической модели данных.

Помимо этого, для каждой таблицы происходит определение столбцов, включая их имена, типы данных и соответствующие ограничения. Также идет определение первичных и внешних ключей, которые уникально идентифицируют каждую запись и обеспечивают связи между таблицами.

В данной схеме базы данных присутствует связь между таблицами «многие к одному» - множество экземпляров одного класса связаны с одним экземпляром другого класса. Для более понятного описания физической схемы была спроектирована таблица атрибутов объектов физической схемы, представленная ниже.

Таблица 10 – Атрибуты объектов физической схемы

№ п/п	Наименование атрибута	Тип	Тип ключа	Ограничения	
				Nullable	Unique
1	2	3	4	5	
JOURNALISTS					
1	ID_JournalistsCode	integer(30)	PK	NOT NULL	HET
2	ID_ExperienceJournalists	integer(30)	FK	NULL	HET
3	ID_SkillsJournalists	integer(30)	FK	NULL	HET
4	ID_BusynessJournalists	integer(30)	FK	NULL	HET
5	ID_CitySearch	integer(30)	FK	NULL	HET
6	ID_EducationJournalists	integer(30)	FK	NULL	HET
7	SurnameJournalists	char(30)	-	NOT NULL	HET
8	NameJournalists	char(30)	-	NOT NULL	HET
9	PatronymicJournalists	char(30)	-	NOT NULL	HET
10	PhoneNumber	char(11)	-	NOT NULL	HET
SKILLS JOURNALISTS					
1	ID_SkillsJournalists	integer(30)	PK	NOT NULL	HET
2	NameSkills	char(30)	-	NOT NULL	HET
EDUCATION JOURNALISTS					
1	ID_EducationJournalists	integer(30)	PK	NOT NULL	HET
2	NameEducation	char(30)	-	NOT NULL	HET
EXPERIENCE JOURNALISTS					
1	ID_ExperienceJournalists	integer(30)	PK	NOT NULL	HET
2	NameExperience	char(30)	-	NOT NULL	HET
BUSYNESS JOURNALISTS					
1	ID_BusynessJournalists	integer(30)	PK	NOT NULL	HET
2	NameBusyness	char(30)	-	NOT NULL	HET
CITY SEARCH					
1	ID_CitySearch	integer(30)	PK	NOT NULL	HET
2	NameCitySearch	char(30)	-	NOT NULL	HET

Продолжение таблицы 10

№ п/п	Наименование классов	Тип	Тип ключа	Ограничения	
				Nullable	Unique
1	2	3	4	5	
TABLE_SPECIALIZATIONS					
1	SpecializationID	integer	PK	NOT NULL	HET
2	SpecializationName	char(100)	-	NOT NULL	ДА
TABLE_GENRES					
1	GenreID	integer	PK	NOT NULL	HET
2	GenreName	char(100)	-	NOT NULL	ДА
TABLE_ARTICLES					
1	ArticleID	integer	PK	NOT NULL	HET
2	Title	char(255)	-	NOT NULL	HET
3	Content	char(255)	-	NOT NULL	HET
4	DatePublished	date	-	NOT NULL	HET
5	AuthorID	integer	FK	NOT NULL	HET
6	GenreID	integer	FK	NOT NULL	HET
JOURNALIST_SPECIALIZATIONS					
1	JournalistID	integer	FK	NOT NULL	HET
2	SpecializationID	integer	FK	NOT NULL	HET

В колонках таблицы приводятся наименования атрибутов, типы данных атрибутов, типы ключей (первичным - PK или внешним - FK), а также виды ограничений Nullable (со значениями NOT NULL – значение не может быть пустым или NULL – значение может быть пустым) и Unique - уникальный (со значениями ДА или НЕТ).

2.4 Проектирование интерфейса пользователя

Пользовательский интерфейс (user interface) представляет собой определенный канал для коммуникаций, с помощью которого пользователь взаимодействует с компьютером, программой и разными другими системами [5].

При анализе предметной области дипломной работы «Web-платформа подбора журналистов для тематических исследований» были определены необходимые элементы для создания пользовательского интерфейса:

- на платформе должен быть реализован доступ к странице регистрации и входа на сайт для разных пользователей;
- возможность поиска, просматривания вакансий и анкет для журналистов и работодателей;

– методы редактирования аккаунтов, а также информации, вакансий и анкет;

В ходе разработки Web-платформы был спроектирован пользовательский интерфейс. Пример интерфейса показан на рисунке 9.

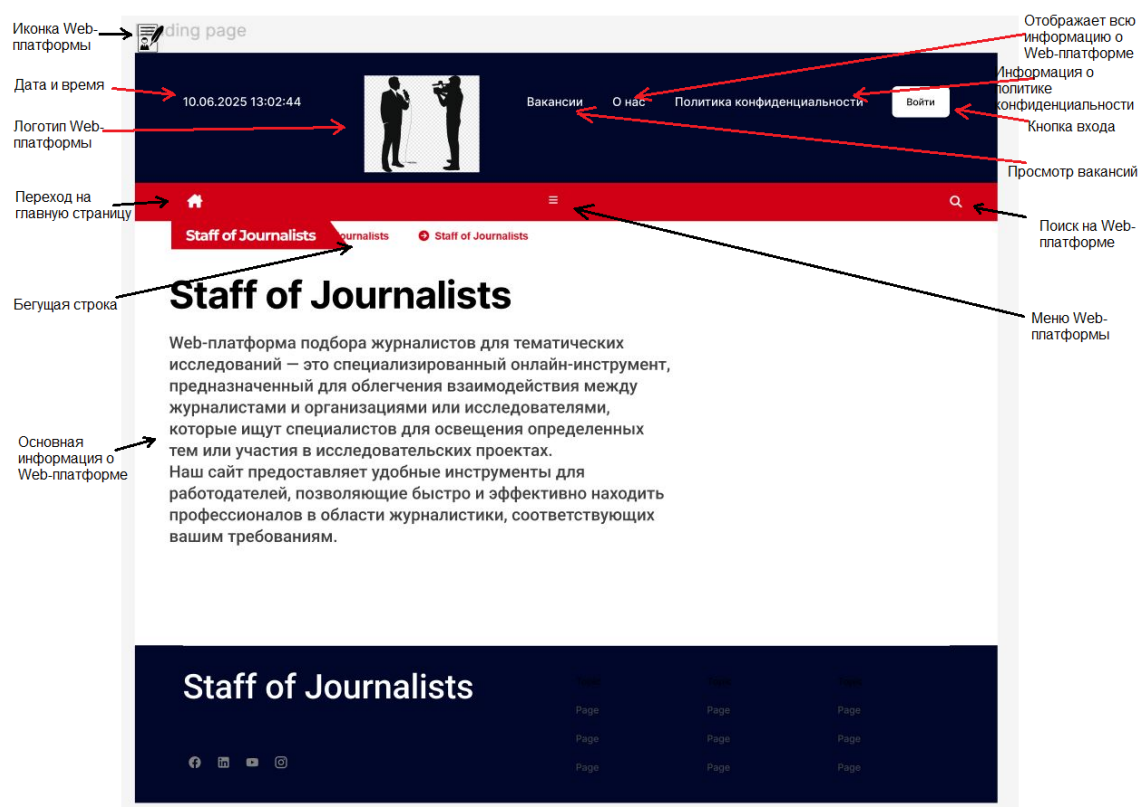


Рисунок 9 – Пример интерфейса главной страницы

«Существует свод базовых правил по комбинации цветов для достижения желаемого результата. К примеру, сочетание красного и желтого цветов побудит к активности. Психологический эффект усиливается, когда эти цвета присутствуют в оформлении или рекламе предметов, связанных с энергией (топливный насос или коробка спичек); сочетание красного и зеленого ассоциируется с безопасностью и надежностью. Часто используется в рекламе технического оборудования. Сочетание красного и белого создает ощущение чистоты и здоровья (эмблема медицинского креста). Сочетание синего и зеленого способно вызвать ощущение покоя. Комбинация белого и черного, считающегося в нашей стране официальным, придает ощущение оттенок торжественности и официальности; многоцветные яркие комбинации маркетологи используют для рекламы товаров для детей.

Для печатных средств массовой информации существуют свои особенности использования цвета. Цвет в основном выполняет в изданиях коммуникативную, символическую и выразительную функции. Прежде всего проявляется самая элементарная коммуникативная функция цвета — различительная. Взглянув на обложку или колонтитулы с названиями разделов, выделенные различными цветами, читатель понимает, что в этом издании существует рубрикация. Постоянный читатель данного издания, пролистывая журнал, будет ассоциировать цвет с конкретной рубрикой. В рекламных модулях, для привлечения внимания, используют цветовой контраст. Цветовые коммуникативные обозначения помогают восприятию основного содержания, активно управляют вниманием читателя и обеспечивают удобную навигацию по изданию.

Для телевизионных СМИ есть свои критерии выбора цветового решения. Очень важен не только выбор цвета, но и настройка света. Блики от темного экрана или наоборот слишком яркое освещение приведут к тому, что зритель быстро устанет от телеэкрана, а с помощью правильно настроенного света на телезрителя можно воздействовать эмоционально. Интерьеры в новостных телепередачах на разных телеканалах выполнены по единому стандарту - в синих и красных цветах. Это связано с влиянием цветов национального флага страны. Для телеведущих так же есть установленный регламент выбора цвета в костюме. Не стоит быть полностью одетым в белое, черное и красное. Белый будет засвечен на телеэкране, черный может испортить тон лица или придать уставший вид, красный - слишком яркий и будет отвлекать от происходящего на экране. Это же касается неоновых цветов. Оптимальным выбором будет одеться в оттенки синего, за исключением случаев, когда фон этого цвета.

Работая с цветом нужно проявлять осторожность. Знание языка цвета необходимо при создании тех или иных источников информации, ведь с помощью цветовых сочетаний можно влиять на подсознание и поступки людей. Выбор цвета в различных СМИ, будь то телепрограмма, печатное издание или

реклама будет различаться, но все они преследуют единую функцию - корректно донести информацию до зрителя» [8].

Поэтому цветовая гамма Web-платформы представлена в основных трёх цветах: темно-синем, красном и белом. Синий цвет способствует концентрации внимания и ассоциируется с небом, а также считается более спокойным оттенком из всех. Белый цвет преобладает на Web-платформе, что дает возможность пользователю «не утонуть» в ярких красках, тем самым успокаивая аудиторию. Красный цвет, в свою очередь, является неким раздражителем нервной системы, заставляющим обратить внимание на конкретный материал, может ассоциироваться с важной информацией, опасностью, кровью, властью и так далее.

С главной страницы Web-платформы пользователь может перейти на другие страницы – входа, просмотра вакансий, политике конфиденциальности и другие. Иконки помогают пользователям ознакомиться с меню платформы, перейти на главную страницу и осуществлять поиск по запросу работодателей и журналистов.

Интерфейс спроектирован так, чтобы позволить пользователям интуитивно находить необходимые данные о платформе и интересующей информации на ней.

Без регистрации на платформе, доступ к данным и информации ограничен. Это сделано для того, чтобы обезопасить данные регистрирующихся пользователей и содержать их в сохранности на платформе.

Чтобы зарегистрироваться на платформе пользователю необходимо перейти в окно входа, которое впоследствии предложит войти на платформу или зарегистрироваться на ней.

2.5 Проектирование алгоритма подбора журналистов

Проектирование алгоритма платформы является важным этапом создания детального плана, который определяет взаимодействие определенных компонентов друг с другом для выполнения различных задач. Этот процесс

включает в себя разработку логики и последовательности действий, которая система должна выполнять для нормального функционирования платформы.

Основные этапы проектирования алгоритма платформы приведены ниже.

1) Определение требований – платформа для работодателя должна предоставлять возможность поиска и подбора журналистов на основе различных характеристик, таких как опыт работы, навыки, образование; для журналистов – возможность поиска и подбора лучших вакансий опубликованных работодателями.

2) Анализ и проектирование – представляет собой разработку архитектуры платформы, включая базу данных для хранения информации о журналистах, пользовательского интерфейса для взаимодействия и работой с платформой, а также обработкой запросов.

3) Разработка алгоритмов – создание алгоритмов поиска резюме журналистов, который включает ввод критериев и поиска, выполнение запросов к базе данных, фильтрацию результатов и отображение списка подходящий кандидатов и их резюме.

4) Тестирование и валидация – проверка работы алгоритмов поиска на различных наборах данных для обеспечения его корректности и эффективности.

5) Оптимизация – улучшение алгоритма поиска для ускорения выполнения запросов и повышения удобства использования и интерфейса.

6) Документирование – написание документации, описывающей алгоритмы поиска, его шаги, логику работы и взаимодействия с различными компонентами платформы.

Система поддержки принятия решения представляет собой программный инструмент, который помогает при анализе данных для выбора оптимальных решений в многоуровневых ситуациях.

Экспертные системы представляют собой компьютерные программы, имитирующие процесс принятия решений экспертов в определенной области.

Экспертные системы и система поддержки принятия решений отличаются по назначению, структуре, методам логического вывода и примерам применения.

Критерии на Web-платформе проектировались с использованием системы поддержки принятия решения.

Для подробного раскрытия критериев была спроектирована карта знаний критериев выбора, представленная на рисунке 10.

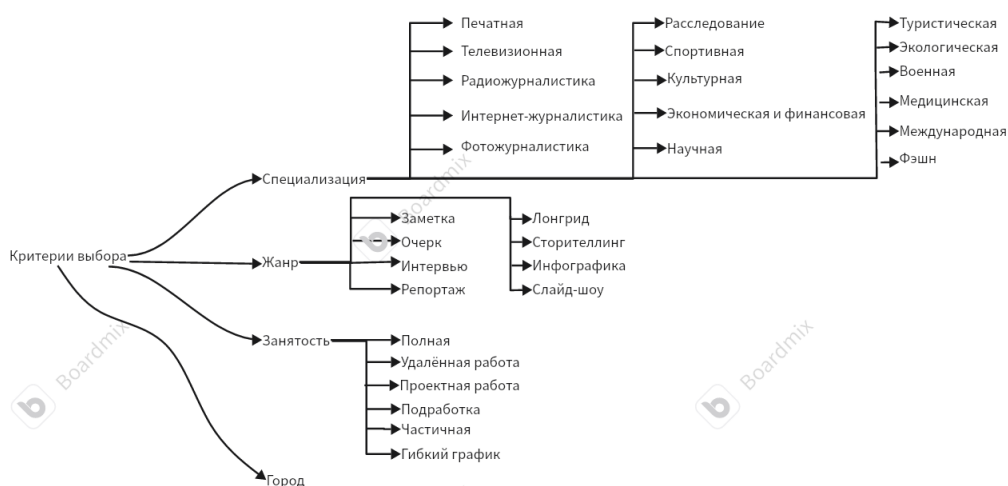


Рисунок 10 – Карта знаний критериев выбора

Карта знаний критериев выбора представляет собой визуальное представление структуры знаний в определенной области, которое помогает организовать, понять и передать информацию. Она используется для демонстрации связей между различными концепциями, идеями или элементами знаний для лучшего понимания взаимосвязи и структуры.

Критерии выбора на карте знаний подразделяются на несколько значений: специализация, жанр, занятость и город.

Специализация в данной схеме представляет собой направление деятельности в журналистике, которые журналисты и работодатели могут указывать при написании резюме или вакансии.

Жанр как критерий в карте знаний – устойчивая форма журналистского произведения, собранного материала, который отличается методом литературной подачи, стилем изложения и другими качествами, которые

указываются при загрузке статьи в формате PDF на Web-платформу при написании резюме.

Занятость в данной схеме является формой критериев профессиональной деятельности журналиста, которая подразумевает под собой сбор, анализ и распространение информации в определенном формате трудоустройства. Этот критерий также используется для написания резюме на Web-платформе.

Критерий город – это локация, в которой находится журналист. На Web-платформе будет реализован поиск по разным городам, а в дальнейшем платформа может быть усовершенствована с добавлением поиска по ближайшим районам и областям одного города.

Была спроектирована диаграмм деятельности, которая представлена на рисунке 11.

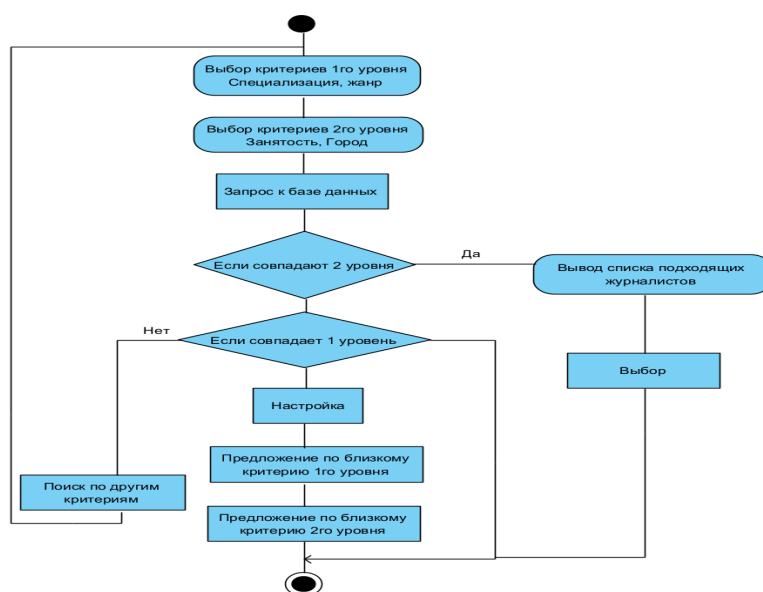


Рисунок 11 – Диаграмма деятельности подбора резюме журналистов

Уровни 1 и 2, в диаграмме деятельности, представляют собой уровни необходимых критериев для подбора журналистов. Основные рассматриваемые критерии представлены в виде – специализации журналиста и жанра статей, а также занятости и города (местоположения) журналиста.

Благодаря этапу проектирования алгоритма платформы можно обеспечить эффективную и надежную работу системы, а впоследствии Web-платформы.

3 Реализация Web-платформы подбора журналистов

Для реализации Web-платформы были учтены все необходимые этапы, начиная от проектирования и заканчивая развертыванием и поддержкой.

Основные шаги процесса представлены ниже.

Анализ требований:

- сбор требований – представляет собой определение задач и целей платформы, а также потребностей пользователей, что включает взаимодействие с заинтересованными сторонами для понимания их требований и ожиданий;
- анализ конкурентности – изучение существующих решений, платформ, приложений на рынке для выявления сильных и слабых сторон.

Проектирование архитектуры:

- архитектура системы – определение общей архитектуры платформы, включая клиентскую и серверную части, базу данных и взаимодействие между ними;
- выбор средств для реализации платформы – определение технологий, которые будут использоваться для разработки (СУБД, фреймворки и другие).

Разработка платформы включает в себя такие этапы как дизайн интерфейса, frontend-разработка клиентская часть, backend-разработка серверная, КЛИЕНТ_СЕРВЕРНАЯ ЧАСТЬ проектирование базы данных и её реализацию. Схема взаимодействия модулей представлена на рисунке 12.

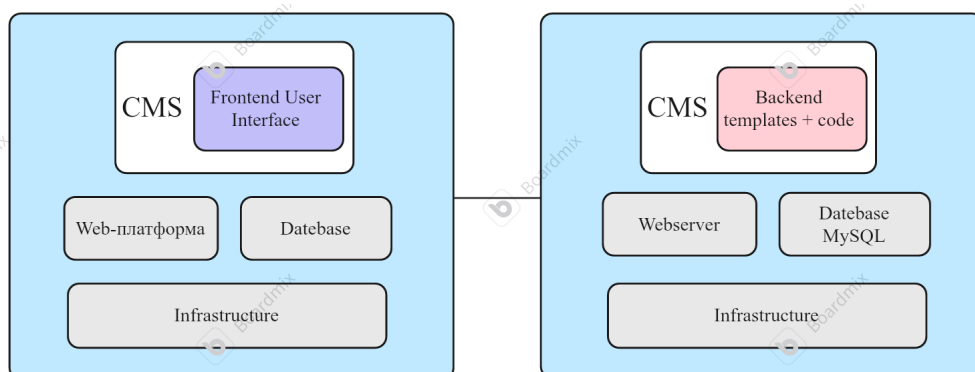


Рисунок 12 – Схема взаимодействия модулей

Клиентская часть:

- дизайн интерфейса – создание пользовательского интерфейса, который позволит пользователям интуитивно находить необходимую информацию на платформе, также включает в себя разработку макетов и прототипов;
- frontend-разработка – реализация клиентской части с использованием технологий – HTML, CSS, JavaScript и других.

Серверная часть:

- backend-разработка – реализация серверной логики, обработка запросов, взаимодействие с базой данных и управление бизнес-логикой;
- API – разработка API для взаимодействия между клиентской и серверной частями.

База данных:

- проектирование базы данных – подразумевает под собой создание логической и физической схемы базы данных, включая определение таблиц, столбцов, связей и ограничений;
- реализация базы данных – настройка и наполнение базы данных с использованием СУБД (MySQL Server и другие).

Интеграция и тестирование подразумевает под собой интеграцию компонентов и тестирование функциональной части платформы.

Интеграция компонентов:

- связывание компонентов – обеспечение взаимодействия между клиентской и серверной частями, а также интеграция с внешними сервисами и API;
- настройка окружения – подготовка и настройка окружения для развертывания, включая серверы, базы данных и сетевые компоненты.

Тестирование:

- функциональное тестирование – представляет из себя проверку функциональности платформы для обеспечения соответствия требованиям;

- тестирование производительности – оценка производительности платформы и выявление узких мест;
- тестирование безопасности – проверка уязвимостей и обеспечение безопасности данных и системы.

Развертывание и поддержка включает в себя настройку серверов, баз данных и других компонентов для развертывания платформы, а также установку и настройку платформы на производственных серверах. Помимо этого, данный этап реализации платформы включает в себя мониторинг работы системы и поддержку пользователей, а также обновление и улучшение платформы (добавление новых функций, улучшение производительности и безопасности).

Реализация Web-платформы подбора журналистов является комплексным процессом, этапы которого необходимы для создания эффективной и надежной платформы, которая обеспечит удобство её эксплуатации и удовлетворит потребности пользователей.

Для реализации Web-платформы подбора журналистов для тематических исследований были выбраны следующие вспомогательные средства: MySQL Server и phpMyAdmin – для написания и наполнения СУБД, язык гипертекстовой разметки HTML для создания веб-страниц и CMS – WordPress (вспомогательный движок), а также бесплатная версия хостинга Sprinthost для развертываемости на нем Web-платформы.

3.1 Реализация базы данных и заполнение тестовыми данными

Для реализации базы данных был выбран MySQL Server, представляющий собой реляционную систему по управлению базами данных MySQL, который выполняет основную работу по обработке запросов, управления данными и взаимодействию с клиентами. А также, веб-приложение phpMyAdmin с открытым кодом, написанное на языке программирования PHP, которое является графическим интерфейсом для администрирования системы

управления базами данных MySQL. На рисунке 13 представлены запросы по созданию, выборочному подключению базы данных.

```
create database jour;
use jour;
```

Рисунок 13 – Запросы по созданию и выборочному подключению базы данных

Запрос Create database в MySQL Server позволяет создать новую базу данных. Запрос Use позволяет выбрать конкретную базу данных для реализации последующих запросов и операций.

На рисунке 14 представлена созданная новая база данных под наименованием jour.

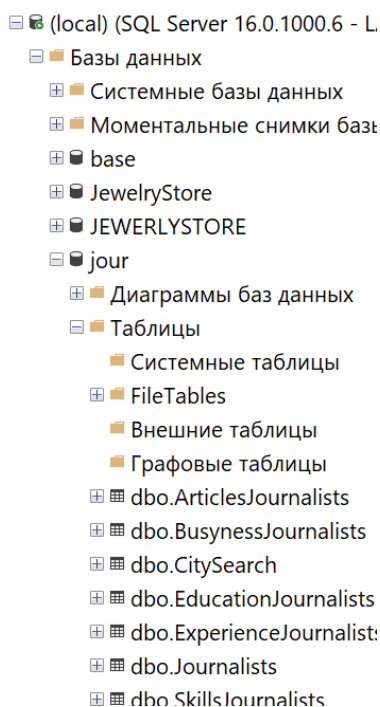


Рисунок 14 – Результат создания объектов базы данных

Следующим шагом стало создание запросов по созданию необходимых таблиц и наполнение их тестовыми данными. Результаты написания запросов представлены на рисунке 15.

```
-- Таблица Журналисты
CREATE TABLE Journalists (
    ID_JournalistsCode INTEGER PRIMARY KEY CHECK (ID_JournalistsCode >=1)
    SurnameJournalists CHAR(30) NOT NULL CHECK (SurnameJournalists LIKE R
    NameJournalists CHAR(30) NOT NULL CHECK (NameJournalists LIKE REPLIC
    PatronymicJournalists CHAR(30) NOT NULL CHECK (PatronymicJournalists
    PhoneNumber CHAR(11) NOT NULL CHECK (PhoneNumber LIKE REPLICATE('[0-9

-- Внешние ключи
ID_ExperienceJournalists INTEGER NOT NULL FOREIGN KEY REFERENCES Expe
ID_SkillsJournalists INTEGER NOT NULL FOREIGN KEY REFERENCES SkillsJo
ID_BusynessJournalists INTEGER NOT NULL FOREIGN KEY REFERENCES Busyne
ID_CitySearch INTEGER NOT NULL FOREIGN KEY REFERENCES CitySearch(ID_C
ID_EducationJournalists INTEGER NOT NULL FOREIGN KEY REFERENCES Educa
);

-- Таблица Образование
CREATE TABLE EducationJournalists (
    ID_EducationJournalists INTEGER PRIMARY KEY CHECK (ID_EducationJourna
    NameEducation CHAR(30) NOT NULL CHECK (NameEducation LIKE REPLICATE('
);

-- Таблица Опыт журналистов
CREATE TABLE ExperienceJournalists (
```

Рисунок 15 – Результат создания таблиц

На данном этапе были созданы все необходимые таблицы базы данных jour, а именно – «Журналисты», «Образование», «Опыт», «Навыки», «Занятость», «Город», «Специализация», «Жанры», «Статьи» и «Специализация и журналисты». Результат заполнения таблиц представлен на рисунке 16.

jour.sql - (local)j...-BJQVL47A\Пк-1 (52))* ✕

```
-----Заполнение таблиц-----
INSERT INTO Journalists (ID_JournalistsCode, SurnameJournalists, NameJournalists, PatronymicJournalists,
VALUES
(1, 'Агафонова', 'Наталья', 'Ирьевна', 3, '89053599689', 1, 1, 1, 1),
(2, 'Арехина', 'Дарья', 'Петровна', 1, '89093357667', 2, 2, 1, 2),
(3, 'Босов', 'Андрей', 'Витальевич', 8, '89092228666', 2, 4, 3, 1),
(4, 'Борисов', 'Григорий', 'Геннадьевич', 2, '89055558226', 3, 3, 4, 6),
(5, 'Быстрова', 'Анна', 'Алексеевна', 8, '89088786661', 2, 4, 2, 1),
(6, 'Гришина', 'Ольга', 'Алексеевна', 7, '89093888666', 2, 1, 1, 4),
(7, 'Зотов', 'Георгий', 'Владимирович', 7, '89083546667', 4, 3, 1, 3),
(8, 'Карпов', 'Василий', 'Иванович', 3, '89093568676', 5, 5, 4, 1),
(9, 'Кирилов', 'Андрей', 'Борисович', 5, '89043548666', 2, 2, 1, 6),
(10, 'Кирилова', 'Анна', 'Александровна', 6, '89053558666', 3, 2, 3, 2),
(11, 'Коробина', 'Наталья', 'Андреевна', 10, '89053556653', 4, 5, 2, 2),
(12, 'Курилова', 'Александра', 'Сергеевна', 9, '89053558677', 2, 1, 1, 5),
(13, 'Латов', 'Сергей', 'Сергеевич', 7, '89051158666', 1, 1, 2, 3),
(14, 'Никифорова', 'Ольга', 'Владимировна', 4, '89093782234', 3, 3, 1, 4),
(15, 'Савичев', 'Алексей', 'Алексеевич', 4, '89093444666', 2, 2, 4, 5);

INSERT INTO ExperienceJournalists (NameExperience)
VALUES
('Стажер/ассистент'),
('Высший/старший специалист');
```

67 %

Результаты					
ID_JournalistsCode	SurnameJournalists	NameJournalists	PatronymicJournalists	Phone	
3	Босов	Андрей	Витальевич	89092	
4	Борисов	Григорий	Геннадьевич	89055	
5	Быстрова	Анна	Алексеевна	89088	
6	Гришина	Ольга	Алексеевна	89093	
7	Зотов	Георгий	Владимирович	89083	
8	Карпов	Василий	Иванович	89093	
9	Кирилов	Андрей	Борисович	89043	
10	Кирилова	Анна	Александровна	89053	
11	Коробина	Наталья	Андреевна	89053	
12	Курилова	Александра	Сергеевна	89053	
13	Латов	Сергей	Сергеевич	89051	
14	Никифорова	Ольга	Владимировна	89093	
15	Савичев	Алексей	Алексеевич	89093	

Рисунок 16 – Результат заполнения таблиц данными

На рисунке 16 показан результат вывода данных заполненных таблиц базы данных jour с помощью синтаксиса оператора Select*From в MySQL, который выводит все выбранные столбцы из необходимой таблицы.

На рисунке 17 представлена реализация и загрузка базы данных в Web-приложении phpMyAdmin.

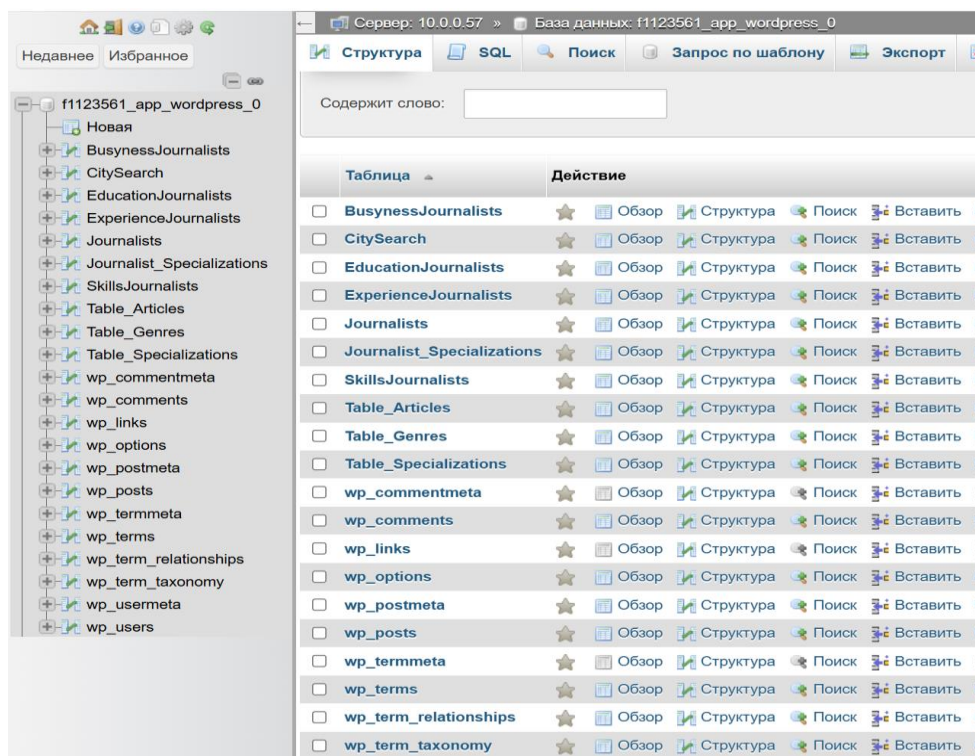


Рисунок 17 – База данных в phpMyAdmin

Для загрузки данных необходимо было видоизменить код в консоли phpMyAdmin из-за возникших различий и ограничений, работа с консолью и исправление кода представлены на рисунке 18.

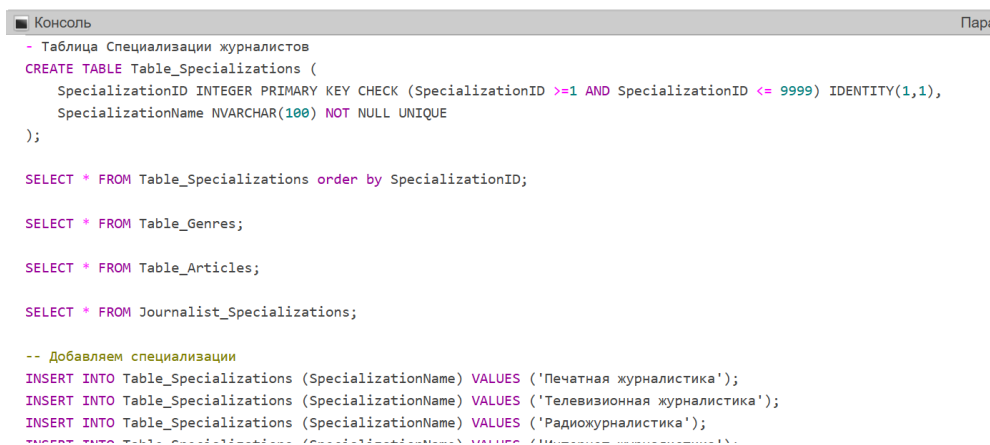


Рисунок 18 – Работа с консолью в phpMyAdmin

3.2 Реализация Web –платформы

Для реализации Web-платформы журналистов для тематических исследований были использованы язык гипертекстовой разметки HTML для создания веб-страниц и CMS – WordPress (вспомогательный движок).

Все загруженные и подключенные данные хранятся на хостинге Sprinthost в файловой системе.

Была создана диаграмма компонентов, представленная на рисунке 20.

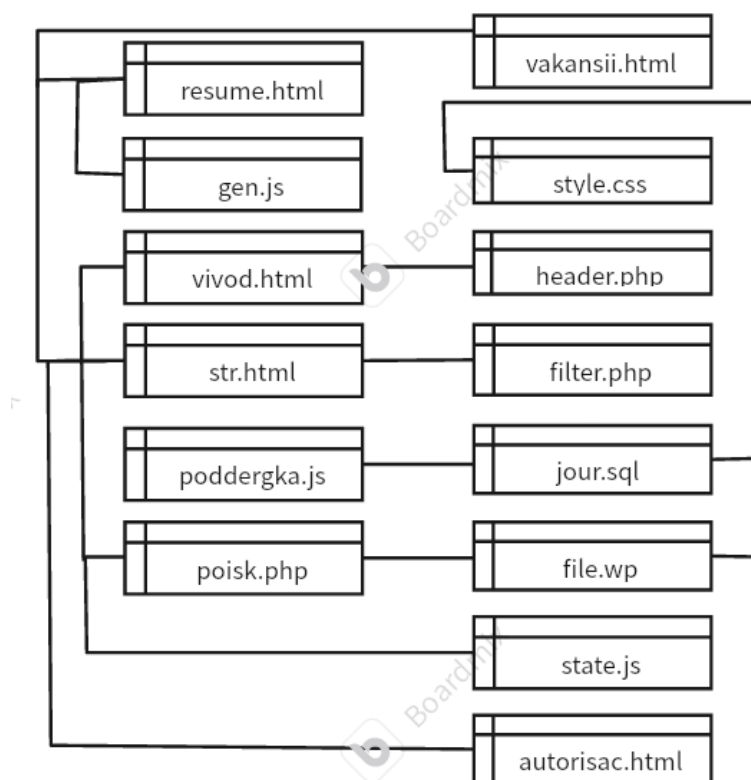


Рисунок 20 – Диаграмма компонентов

На диаграмме компонентов показаны все файлы в форматах – .sql, .js, .wp, .php, .html, хранящиеся на хостинге Sprinthost в файловой системе.

3.2.1 Реализация авторизации пользователей

Далее происходит разработка и написание страниц кода на HTML. На рисунке 21 представлен процесс написания страниц кода и их первоначальное тестирование перед загрузкой на хостинг.

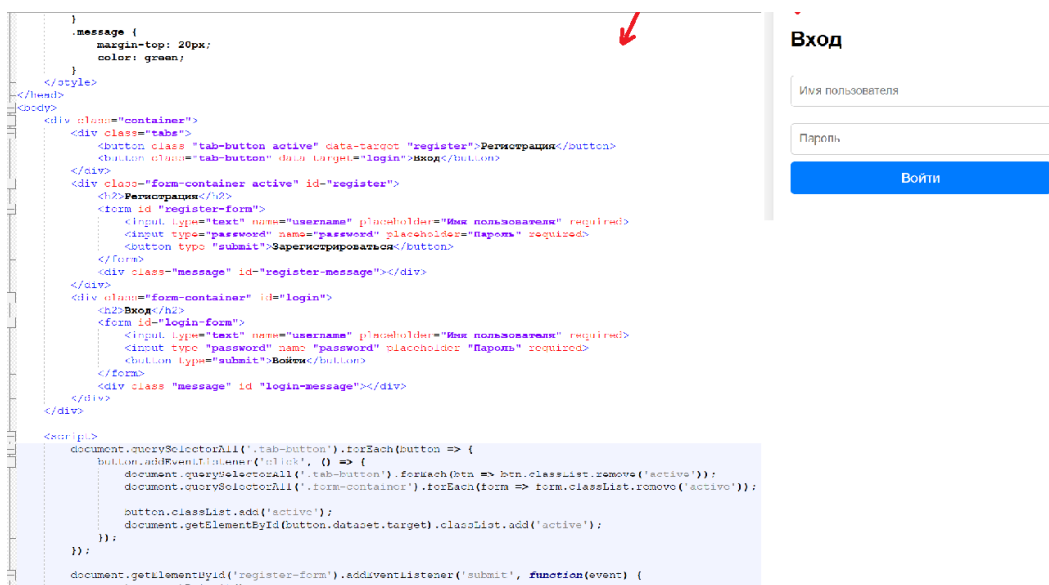


Рисунок 21 – Написание и предварительное тестирование страницы
«Вход/Регистрация»

С помощью данного редактора и языка HTML были написаны и протестированы все основные страницы Web-платформы, такие как: «Вход/Регистрация», «Вакансии», «Резюме», «Политика конфиденциальности», «Личный кабинет», «Профиль» и другие. Также был написан основной фильтр поиска по выбранным критериям.

Для создания фильтра поиска по подбору журналистов был реализован код на скриптовом языке программирования php с открытым исходным кодом. Работа с кодом представлена на рисунке 22.

```

6 // Проверка соединения
7 if ($conn->connect_error) {
8     die("Connection failed: " . $conn->connect_error);
9 }
10
11 // Получение данных из формы
12 $specialization = isset($_POST['specialization']) ? $_POST['specialization'] : null;
13 $city = isset($_POST['city']) ? $_POST['city'] : null;
14 $skill = isset($_POST['skill']) ? $_POST['skill'] : null;
15 $experience = isset($_POST['experience']) ? $_POST['experience'] : null;
16 $employment = isset($_POST['employment']) ? $_POST['employment'] : null;
17 $genre = isset($_POST['genre']) ? $_POST['genre'] : null;
18
19 // Подготовка SQL-запроса
20 $sql = "SELECT DISTINCT j.*
21     FROM Journalists j
22     LEFT JOIN Journalist_Specializations js ON j.ID_JournalistsCode = js.JournalistID
23     LEFT JOIN Table_Specializations s ON js.SpecializationID = s.SpecializationID
24     LEFT JOIN CitySearch c ON j.ID_CitySearch = c.ID_CitySearch
25     LEFT JOIN SkillsJournalists sk ON j.ID_SkillsJournalists = sk.ID_SkillsJournalists
26     LEFT JOIN ExperienceJournalists e ON j.ID_ExperienceJournalists = e.ID_ExperienceJournalists
27     LEFT JOIN BusynessJournalists b ON j.ID_BusynessJournalists = b.ID_BusynessJournalists
28     LEFT JOIN Table_Articles a ON j.ID_JournalistsCode = a.AuthorID
29     LEFT JOIN Table_Genres g ON a.GenreID = g.GenreID
30     WHERE 1=1";
31
32 if ($specialization) {
33     $sql .= " AND s.SpecializationName = '" . $conn->real_escape_string($specialization) . "'";
34 }
35 if ($city) {
36     $sql .= " AND c.NameCitySearch = '" . $conn->real_escape_string($city) . "'";

```

Рисунок 22 – Код фильтра по подбору журналистов

Фильтр представляет собой своеобразную выборку данных по указанным критериям, что впоследствии помогает осуществлять поиск резюме журналистов и статей на Web-платформе для работодателей. SQL-запрос выполняется с помощью метода query.

После этого страницы подгружаются на хостинг в формате txt файлов и подключаются к вспомогательному движку WordPress, для последующего настраивания.

Все данные хранятся на хостинге Sprinthost в файловой системе и частично на движке WordPress в разделе «Страницы», хранение данных и страниц представлено на рисунке 23.

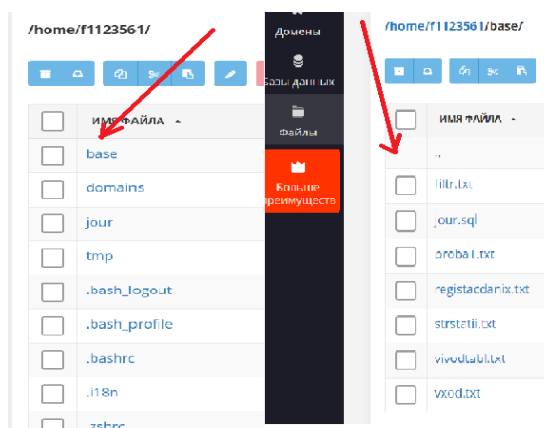


Рисунок 23 – Хранение данных и страниц на хостинге

Такая система хранения позволяет не перегружать хостинг и способствует лучшей работе Web-платформы.

3.2.2 Реализация подсистемы администратора

Реализация подсистемы администратора на Web-платформе подбора журналистов для тематических исследований представляет собой создание интерфейсов и функциональности, которые позволяют администраторам управлять пользователями, их аккаунтами, контентом, настройками платформы, модернизацией платформы, осуществлять деятельности поддержки платформы и помощи пользователям, а также обеспечивать безопасность данных пользователей на платформе.

Реализованная страница администратора для работы с пользователями представлена на рисунке 24.

Управление пользователями

Добавить пользователя

Имя:

Email:

Роль:

Добавить

Список пользователей

ID	Имя	Email	Роль	Действия
1	Иван Иванов	ivan@example.com	Журналист	<div>Редактировать</div> <div>Удалить</div>
2	Мария Петрова	maria@example.com	Редактор	<div>Редактировать</div> <div>Удалить</div>

Рисунок 24 – Страница администратора для работы с пользователями

Разработка интерфейса подсистемы администратора происходила с использованием языка гипертекстовой разметки HTML.

3.2.3 Реализация подсистемы работодателя

Реализация подсистемы работодателя на Web-платформе подбора журналистов для тематических исследований представляет собой создание интерфейсов и функциональности, которые позволяют работодателям управлять вакансиями, а также редактировать информацию в своем профиле. Помимо этого, на Web-платформе был настроен поиск резюме и статей по указанным работодателем критериям.

Реализованная страница работодателя для работы с вакансиями представлена на рисунке 25.

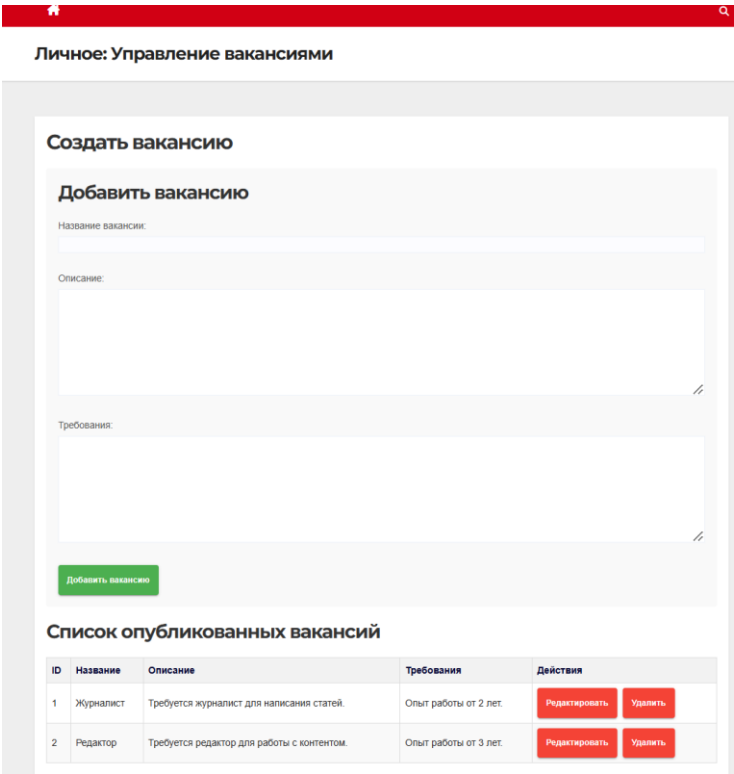


Рисунок 25 – Страница работодателя для работы с вакансиями

На данной странице Web-платформы работодатель может создать новую вакансию для поиска журналиста, а также ознакомиться и произвести редактирование с уже опубликованными ранее вакансиями своей организации.

Реализованная страница работодателя для работы с профилем представлена на рисунке 26.

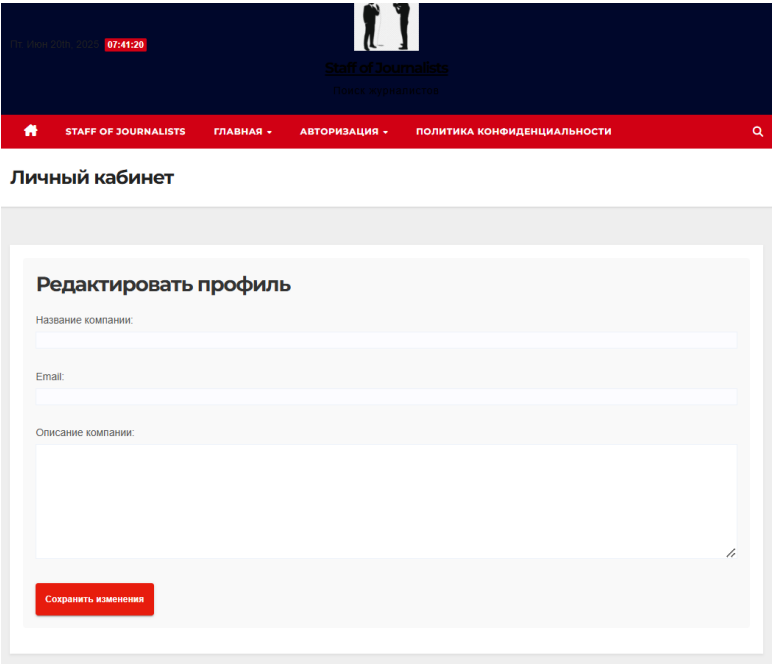


Рисунок 26 – Страница работодателя для редактирование профиля

На данной странице Web-платформы работодатель может редактировать свой профиль.

Рисунок 27 – Реализация поиска по критериям

Благодаря данному поиску с фильтром на Web-платформе работодатель может осуществлять поиск резюме журналистов и статей для ознакомления.

Разработка интерфейса подсистемы работодателя происходила с использованием языка гипертекстовой разметки HTML и скриптовым языком программирования с открытым кодом - PHP.

3.2.4 Реализация подсистемы журналиста

Реализация подсистемы журналиста на Web-платформе подбора журналистов для тематических исследований представляет собой создание интерфейсов и функциональности, которые позволяют журналистам управлять резюме, а также редактировать информацию в своем профиле и осуществлять поиск вакансий.

Реализованная страница журналиста для работы с резюме представлена на рисунке 28.

Рисунок 28 – Страница журналиста для работы с резюме

Помимо стандартного заполнения резюме для подсистемы журналиста на Web-платформе была реализована автоматическая генерация резюме с помощью интерпретируемого языка программирования JavaScript. Пример генерации представлен на рисунке 29.

Рисунок 29 – Генерация резюме пользователя

Работа с кодом представлена на рисунке 30.

```

23 try {
24   // Получение данных из базы данных
25   const [journalists] = await pool.query('SELECT * FROM Journalists
26     WHERE id = ?', [req.query.id]);
27
28   if (journalists.length === 0) {
29     return res.status(404).send('Журналист не найден');
30   }
31
32   const journalist = journalists[0];
33
34   // Генерация резюме на основе данных
35   const resume = `
36     <h1>Резюме Журналиста</h1>
37     <p><strong>Имя:</strong> ${journalist.name}</p>
38     <p><strong>Специализация:</strong> ${journalist.specialization}</p>
39     <p><strong>Опыт работы:</strong> ${journalist.experience}</p>
40     <p><strong>Навыки:</strong> ${journalist.skills}</p>
41     <p><strong>Город:</strong> ${journalist.city}</p>
42   `;
43
44   res.send(resume);
45 } catch (error) {
46   console.error('Ошибка при генерации резюме:', error);
47   res.status(500).send('Ошибка сервера');
48 });
49
50 // Запуск сервера
51 app.listen(port, () => {
52   console.log('Сервер запущен на http://localhost:${port}');

```

Рисунок 30 – Генерация резюме пользователя

Благодаря данной странице Web-платформы журналист может производить работу с резюме и видоизменять его.

Реализованная страница журналиста для работы с профилем показана на рисунке 31.

Рисунок 31 – Страница журналиста для работы с профилем

С помощью данной страницей Web-платформы журналист может редактировать свой профиль.

Реализованная страница журналиста для поиска вакансий представлена на рисунке 32.

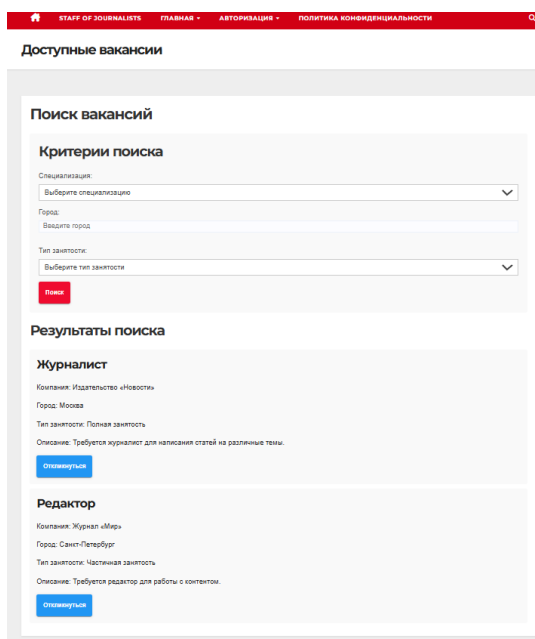


Рисунок 32 – Страница журналиста для поиска вакансий

Благодаря реализации данной страницы платформы журналист может осуществлять поиск вакансий от работодателей по выбранным вакансиям.

4 Тестирование Web-платформы подбора журналистов

4.1 Тестирование «черным ящиком»

Функциональная диаграмма помогает в визуализации и понимании взаимодействия различных компонентов Web-платформы подбора журналистов для тематических исследований, представленная на рисунке 33.

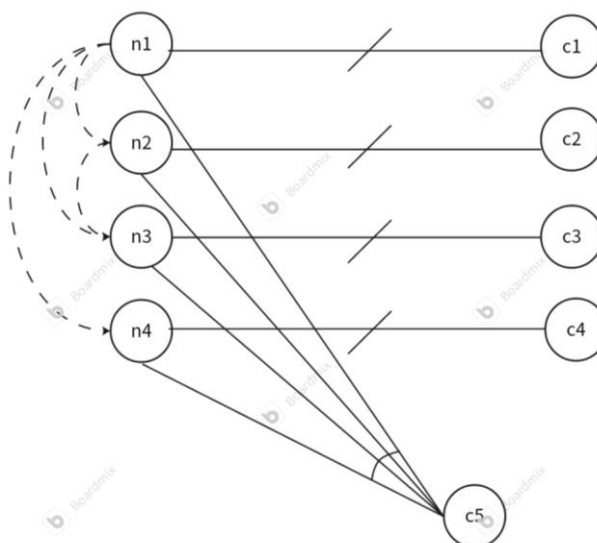


Рисунок 33 – Функциональная диаграмма

На основе функциональной диаграммы была составлена таблица решений причин и следствий, представленная ниже.

Таблица 11 – Таблица решений причин и следствий

№ п/п	Причины				Следствия				
	n1	n2	n3	n4	c1	c2	c3	c4	c5
	2				3				
1	0	-	-	-	1	-	-	-	-
2	1			0	0	-	-	1	0
3	1	0	0	1	0	1	1	0	1
4	1	1	0	1	0	0	1	0	1
5	1	0	1	1	0	1	0	0	1
6	1	1	1	1	0	0	0	0	1

Первая причина – пользователь авторизован. Вторая причина – нет резюме. Третья причина – не выбраны критерии. Четвертая причина – выбор критериев 1го уровня. Пятая причина – выбор критериев 2го уровня. Шестая причина – выбрано все - показать резюме.

Тестирование методом «черный ящик» представляет собой метод тестирования программного обеспечения, при котором тестирующий не имеет доступа к внутренней структуре и коду системы. Вместо этого тестирование происходит с помощью проверки функциональности системы на основе спецификаций и требований.

Основной целью данного тестирования является проверка правильной работы Web-платформы от лица подсистем пользователей.

4.2 Метрики кода

Рассмотрели исходный код функции materials для получения файлов, представленный на рисунке 34, для которого будут рассчитаны метрики Холстеда и выполнен комплексный анализ кода. Анализ кода будет выполнен согласно формулам метрики Холстеда.

```
        cfg.get_message(key="materials_data", default="Имен материалы для вашей группы.")
    )

    # Список всех файлов (только обычные файлы, без папок) в group_dir
    files = []
    if group_dir.exists() and group_dir.is_dir():
        # Сортируем только файлы (не папки), без скрытых
        for entry in sorted(group_dir.iterdir()):
            if entry.is_file() and not entry.name.startswith("."):
                files.append(entry.name)

    # Если файлов нет
    if not files:
        await status_msg.edit_text(
            cfg.get_message(
                key="no_materials",
                default="Преподаватель для вашей группы ничего не загрузил."
            )
        )
        return

    # Иначе: формируем Inline-кнопки по каждому файлу
    keyboard = []
    for filename in files:
        # Здесь encode необходим, чтобы в callback_data не было пробелов и специальных символов
        callback_data = f"material:{quote(group_name)}:{quote(filename)}"
        keyboard.append(InlineKeyboardButton(text=filename, callback_data=callback_data))

    reply_markup = InlineKeyboardMarkup(keyboard)

    # Отправляем сообщение с кнопками
    await status_msg.edit_text(
        cfg.get_message(
            key="materials_list",
            default="Список материалов для вашей группы. Нажмите на название, чтобы получить файл."
        ),
        reply_markup=reply_markup
    )

    return CommandHandler(commands="materials", materials)
```

Рисунок 34 – Код функции materials

Таблица 12 – Таблица решений причин и следствий

Переменная	№	Встречается	Оператор	№	Встречается
update context	1	7	<!DOCTYPE html>	1	5
context	2	6	<html>	2	2
user	3	3	<head>	3	4
document	4	8	<title>	4	4
podd	5	10	<body>	5	4
state	6	4	<p>	6	3
vak	7	7		7	3
pois	8	8		8	3
jour	9	8	delete	9	4
prof	10	6	in	10	2
filt	11	4	instanceof	11	3
gen	12	8	typeof	12	2
spec	13	3			
id	14	3			
Итого:	14	85		12	39
	n2	N2		n1	N1

Расчет метрик Холстеда выполнен с использованием математического пакета Mathcad. Результат расчета приведен ниже.

Исходные данные:

$mn1:=12, mN1:=39, mn2:=14, mN2:=39.$

Расчет словаря, длины и объема программы производится по формуле:

$n = mn1 + mn2 =26; N = mN1 + mN2 =124; V=N*\log(2,n)=582,50.$

Расчет уровня качества программирования производится по формуле:

$L=2mn2 /(mn1*mN2)=0,0275.$

Расчет сложности понимания программы производится по формуле:

$$EC=V/2L=10538,18.$$

Расчет уровня языка производится по формуле:

$$Y = V/D2=0,439.$$

Расчет оценки необходимых интеллектуальных усилий производится по формуле:

$$E=N*\log(2,n/L)=1218,16.$$

Расчет трудоемкости кодирования производится по формуле:

$$D = 1/L = 36,36.$$

Расчет информационного содержания программы производится по формуле:

$$I=V/D=16,00.$$

Подход, приведенный в расчете метрик Холстеда, значительно усложняется при увеличении объема кода проекта. В таком случае целесообразно использовать встроенные в среду разработки или отдельно поставляемые инструменты для автоматического анализа качества кода.

Заключение

В рамках выполнения выпускной квалификационной работы была разработана комплексная Web-платформы подбора журналистов для тематических исследований. На первом этапе было проведено исследование предметной области, в ходе которого проанализированы современные решения для подбора журналистов, такие как используемые сайты и платформы по схожей тематике. Также, сравнительный анализ существующих сайтов, платформ позволил выявить их сильные и слабые стороны, впоследствии повлиявших на формирование функциональных и нефункциональных требований Web-платформы.

Архитектурное проектирование Web-платформы включило создание концептуальной, логической и физической моделей данных, обеспечивающих эффективное хранение и обработку данных о журналистах, их навыках, опыте и специализации. Было уделено особое внимание проектированию пользовательского интерфейса, который должен был быть доступным, интуитивным, для нахождения необходимой информации работодателями и журналистами.

Реализация Web-платформы была осуществлена с использованием современных технологий и инструментов разработки. База данных была разработана на MySQL Server, что впоследствии обеспечило её надежное хранение информации. Серверная часть разворачивалась на хостинге Sprinthost. Клиентская часть создана с использованием HTML, PHP, JavaScript, что позволило добиться высокой производительности и отзывчивости интерфейса. Отдельное внимание было уделено реализации взаимодействия между всеми компонентами системы через API. Контроль качества разработанной Web-платформы включил комплексное тестирование функционала методом черного ящика.

В результате выполнения выпускной квалификационной работы была разработана Web-платформы подбора журналистов для тематических исследований, отвечающая необходимым заявленным требованиям.

Список использованных источников

1. Дюбуа П. MySQL. Сборник рецептов. – Пер. с англ. – СПб: Символ Плюс, 2006 – 1056 с., ил. ISBN 5 93286 070 7 [Электронный ресурс] URL: <https://docs.yandex.ru/docs/view?tm=1747769455&tld=ru&lang=ru&name=Dubua.pdf&text=mysql%20это%20что%20определение%20из%20учебника&url=https%3A.>
2. Мотев А. А. М85 Уроки MySQL. Самоучитель. — СПб.: БХВ-Петербург, 2006. — 208 с.: ил. ISBN 5-94157-658-7 [Электронный ресурс] URL: https://books.4nmv.ru/books/uroki_mysql_samouchitel_3642745.pdf
3. Author: Marc Delisle, Reviewers: Garvin Hicking Alexander Turek, Development Editor: Louay Fatoohi, Technical Editor: Saurabh Singh, Editorial Manager: Dipali Chittar, Indexer: Mithil Kulkarni, Proofreaders: Martin Brooks, Chris Smith, Layouts and Illustrations: Shantanu Zagade, Cover Designer: Shantanu Zagade / Mastering phpMyAdmin 2.8 for Effective MySQL Management / First edition: April 2004 Second edition: October 2004 Third edition: October 2006 Production Reference: 2290906 Published by Packt Publishing Ltd. 32 Lincoln Road Olton Birmingham, B27 6PA, UK. ISBN 1-847191-60-6 [Электронный ресурс] URL: <https://docs.yandex.ru/docs/view?tm=1747772914&tld=ru&lang=en&name=Mastering-Phpmyadmin-for-Effective-MySQL-Management.pdf&text=phpmyadmin%20из%20учебника&u-rl=https%3A%2F%2Fwww.programmer...>
4. Грачев А. Г21 Создаем свой сайт на WordPress: быстро, легко и бесплатно. Работа с CMS WordPress 3.— СПб.: Питер, 2011. — 288 с.: ил. ISBN 978-5-49807-939-4 [Электронный ресурс] URL: https://kki.tuva.muzkult.ru/media/2020/11/15/1242224593/Grachev_A._Sozdaem_svoj_sajt_na_WordPres_S_WordPress_3._SPb._Piter_s._il.pdf
5. Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство, 6 е издание. – Пер. с англ. – СПб: Символ Плюс, 2008 – 752 с., ил. ISBN 10: 5 93286 104 5 / ISBN 13: 978 5 93286 104 2 [Электронный ресурс] URL: https://docs.yan-dex.ru/docs/view?tm=1747773833&tld=ru&lang=ru&name=-html_i_xhtml_podrobnoe_rukovodstvo_6e_izdanie_fail_pdf_614056.pdf&text=учеб

ник%20html&url=https%3A%2F%2Fbooks.4nmv.ru%2Fbooks%2Fhtml_i_xhtml_p
odrobnое_rukovodstvo_6-e_izdanie...

6. Вершинин А. Создаём сайт самостоятельно, 2018 ISBN 978-5-4493-3006-2 Создано в интеллектуальной издательской системе Ridero [Электронный ресурс] URL: <http://uchcom7.botik.ru/L/web/selfmade.pdf>

7. Пользовательский интерфейс: назначение, принципы разработки и практические рекомендации – тема научной статьи по компьютерным и информационным наукам читайте бесплатно текст научно-исследовательской работы в электронной библиотеке КиберЛенинка URL: <https://cyberleninka.ru/article/n/polzovatelskiy-interfeys-naznachenie-printsipy-razrabotki-i-prakticheskie-rekomendatsii>

8. Волкова К.Э., Бартенева Ю.В. Психологические науки. Особенности использования цветовой гаммы в средствах массовой информации, ФГБОУ ВО «Курский государственный университет» (КГУ) г. Курск, ул. Кирова, 5, 305000, Российская Федерация URL: <https://cyberleninka.ru/article/n/osobennosti-ispolzovaniya-tsvetovoy-gammy-v-sredstvah-massovoy-informatsii>

9. Создание сайтов: HTML, CSS, PHP, MySQL, учебное пособие - Ставрополь 2023 г., 103с. URL: <https://old.stgau.ru/company/personal/user/8958/files/lib/Web-программирование/учебник.pdf>

10. Петин В. А. П29 Сайт на AJAX под ключ. Готовое решение для интернет-магазина. — СПб.: БХВ-Петербург, 2011. — 432 с.: ил. + CD-ROM — (Профессиональное программирование) URL: https://wapinet.ru/build/resources/textbook/php_petin.pdf

11. Осетрова И.С. Администрирование MS SQL Server 2014 - СПб: Университет ИТМО, 2016. – 90 с. URL: <https://books.ifmo.ru/file/pdf/2149.pdf>

12. Бен-Ган, И. Б46 Microsoft® SQL Server® 2012. Создание запросов. Учебный курс Microsoft: Пер. с англ. / И. Бен-Ган, Д. Сарка, Р. Талмейдж. — М.: Издательство «Русская редакция», 2014. — 720 с.: ил. + CD-ROM URL: ht-

[tps://minyurov.wordpress.com/wp-content/uploads/2014/10/microsoft-sql-server-2012-t-sql.pdf](https://minyurov.wordpress.com/wp-content/uploads/2014/10/microsoft-sql-server-2012-t-sql.pdf)

13. Наместников С. М. Основы языка гипертекстовой разметки HTML и CSS: Учебное пособие – Ульяновск: УлГТУ, 2014. – с URL: <https://tk.ulstu.ru/lib/books/html.pdf>

14. Магомедова Д.С. Учебное пособие дисциплины «Создание и продвижение Web-сайта» для направления подготовки 38.03.05 Бизнес-информатика, профиль «Электронный бизнес». – Махачкала: ДГУНХ, 2019г., 58 с. URL: https://dgunh.ru/content/glavnay/ucheb_deyatel/uposob/up-it_ib-fgos-90.pdf

15. Дудник Е.А. Проектирование пользовательских интерфейсов: Учебно-методическое пособие для студентов всех форм обучения/ Рубцовский индустриальный институт. – Рубцовск, 2022 – 71 с. URL: [https://edu.rubinst.ru/resources/books/Dudnik_E.A._Proektirovanie_pol'zovatel'skikh_interpheysov_\(UP\)_2022_g.pdf](https://edu.rubinst.ru/resources/books/Dudnik_E.A._Proektirovanie_pol'zovatel'skikh_interpheysov_(UP)_2022_g.pdf)

16. Шилдс Уолтер Ш57 SQL: быстрое погружение. — СПб.: Питер, 2022. — 224 с.: ил. — (Серия «Библиотека программиста»). ISBN 978-5-4461-1835-9 URL: <https://library.tsilikin.ru/Техника/Информационные%20технологии/SQL/Шилдс%20Уолтер%20SQL.pdf>

17. Скляр Д., Трахтенберг А. PHP. Сборник рецептов.– Пер. с англ.– СПб: СимволПлюс, 2005.– 672 с., ил. ISBN 5932860596 URL: <https://hostadmina.ru/books/PHP%20sbornik%20retseptov%20-%202005.pdf>

18. Суханов, Владимир Иванович. С91 Разработка веб-приложений на платформе Spring : учебно-методическое пособие / В.И. Суханов ; М-во науки и высшего образования РФ.— Екатеринбург : Изд-во Урал. ун-та, 2023.— 180 с. — ISBN 978-5-7996-3735-4. — Текст : непосредственный. URL: https://elar.urfu.ru/bitstream/10995/127898/1/978-5-7996-3735-4_2023.pdf

19. Государев И.Б., Основы разработки веб-приложений на платформах Node.js и Deno— СПб: Университет ИТМО, 2023. – 177 с. URL: <https://books.ifmo.ru/file/pdf/3205.pdf>

20. Киргизова Е.В. К43 Web-технологии: от теории к практике: учеб. пособие / Е.В. Киргизова, А. В. Рубцов. – Красноярск: Сибирский федеральный университет, 2018. – 160 с. URL: http://oldlpi.sfu-kras.ru/images/stories/library/Учебной_пособие_WEB-ТЕХНОЛОГИИ_от_теории_к_практике.pdf

Приложение А
(обязательное)
Текст программы

MySQL

```

create database jour;
use jour;
-----Создание таблиц-----
-- Таблица Журналисты
CREATE TABLE Journalists (
    ID_JournalistsCode INTEGER PRIMARY KEY CHECK
    (ID_JournalistsCode >=1 AND ID_JournalistsCode <= 9999)
    IDENTITY(1,1),
    SurnameJournalists CHAR(30) NOT NULL CHECK (SurnameJournalists
    LIKE REPLICATE('[а-яА-Я- ]', LEN(SurnameJournalists))),
    NameJournalists CHAR(30) NOT NULL CHECK (NameJournalists LIKE
    REPLICATE('[а-яА-Я- ]', LEN(NameJournalists))),
    PatronymicJournalists CHAR(30) NOT NULL CHECK
    (PatronymicJournalists LIKE REPLICATE('[а-яА-Я- ]',
    LEN(PatronymicJournalists))),
    PhoneNumber CHAR(11) NOT NULL CHECK (PhoneNumber LIKE
    REPLICATE('[0-9]', LEN(PhoneNumber))),
    -- Внешние ключи
    ID_ExperienceJournalists INTEGER NOT NULL FOREIGN KEY
    REFERENCES ExperienceJournalists(ID_ExperienceJournalists),
    ID_SkillsJournalists INTEGER NOT NULL FOREIGN KEY REFERENCES
    SkillsJournalists(ID_SkillsJournalists),
    ID_BusynessJournalists INTEGER NOT NULL FOREIGN KEY REFERENCES
    BusynessJournalists(ID_BusynessJournalists),
    ID_CitySearch INTEGER NOT NULL FOREIGN KEY REFERENCES
    CitySearch(ID_CitySearch),
    ID_EducationJournalists INTEGER NOT NULL FOREIGN KEY REFERENCES
    EducationJournalists(ID_EducationJournalists),
    ID_Articles INTEGER NOT NULL FOREIGN KEY REFERENCES
    Table_Articles(ArticleID));
-- Таблица Образование
CREATE TABLE EducationJournalists (
    ID_EducationJournalists INTEGER PRIMARY KEY CHECK
    (ID_EducationJournalists >=1 AND ID_EducationJournalists <= 9999)
    IDENTITY(1,1),
    NameEducation CHAR(30) NOT NULL CHECK (NameEducation LIKE
    REPLICATE('[а-яА-Я-0-9,а-зА-З."/\\ -]', LEN(NameEducation))));
-- Таблица Опыт журналистов
CREATE TABLE ExperienceJournalists (
    ID_ExperienceJournalists INTEGER PRIMARY KEY CHECK
    (ID_ExperienceJournalists >=1 AND ID_ExperienceJournalists <=
    9999) IDENTITY(1,1),
    NameExperience CHAR(30) NOT NULL CHECK (NameExperience LIKE
    REPLICATE('[а-яА-Я-\\/ ]', LEN(NameExperience))));
-- Таблица Навыки журналистов
CREATE TABLE SkillsJournalists (
    ID_SkillsJournalists INTEGER PRIMARY KEY CHECK
    (ID_SkillsJournalists >=1 AND ID_SkillsJournalists <= 9999)
    IDENTITY(1,1),
    NameSkills CHAR(30) NOT NULL CHECK (NameSkills LIKE
    REPLICATE('[а-яА-Я- ]', LEN(NameSkills))));
-- Таблица Занятость журналистов

```

```

CREATE TABLE BusynessJournalists (
    ID_BusynessJournalists INTEGER PRIMARY KEY CHECK
    (ID_BusynessJournalists >=1 AND ID_BusynessJournalists <= 9999)
    IDENTITY(1,1),
    NameBusyness CHAR(30) NOT NULL CHECK (NameBusyness LIKE
    REPLICATE('[а-яА-Я- ]', LEN(NameBusyness))));
-- Город
CREATE TABLE CitySearch (
    ID_CitySearch INTEGER PRIMARY KEY CHECK (ID_CitySearch >=1 AND
    ID_CitySearch <= 9999) IDENTITY(1,1),
    NameCitySearch CHAR(30) NOT NULL CHECK (NameCitySearch LIKE
    REPLICATE('[а-яА-Я-0-9,а-зА-З." -]', LEN(NameCitySearch))));
-- Удаление таблицы (для очистки перед созданием новой структуры)
DROP TABLE IF EXISTS Journalists;
DROP TABLE IF EXISTS ExperienceJournalists;
DROP TABLE IF EXISTS SkillsJournalists;
DROP TABLE IF EXISTS BusynessJournalists;
DROP TABLE IF EXISTS EducationJournalists;
DROP TABLE IF EXISTS Table_Articles;
DROP TABLE IF EXISTS Journalist_Specializations;
DROP TABLE IF EXISTS Table_Genres;
DROP TABLE IF EXISTS Table_Specializations;
ALTER TABLE Table_Articles
ADD CONSTRAINT FK_Table_Articles_Journalists
FOREIGN KEY (AuthorID) REFERENCES Journalists(ID_JournalistsCode);
-- Заполнение данных
SET IDENTITY_INSERT Journalists ON;
SELECT * FROM Journalists;
SELECT * FROM ExperienceJournalists;
SELECT * FROM SkillsJournalists;
SELECT * FROM BusynessJournalists;
SELECT * FROM CitySearch;
SELECT * FROM EducationJournalists;
SELECT * FROM Table_Specializations order by SpecializationID;
SELECT * FROM Table_Genres;
SELECT * FROM Table_Articles;
SELECT * FROM Journalist_Specializations;
-- Выводит таблицу с теми данными, которые задаются
Select j.SurnameJournalists, c.NameCitySearch, s.NameSkills
From Journalists j join CitySearch c on j.ID_CitySearch =
c.ID_CitySearch join SkillsJournalists s on j.ID_SkillsJournalists
= s.ID_SkillsJournalists;
-----Заполнение таблиц-----
INSERT INTO Journalists (ID_JournalistsCode, SurnameJournalists,
NameJournalists, PatronymicJournalists, ID_CitySearch,
PhoneNumber, ID_EducationJournalists, ID_ExperienceJournalists,
ID_SkillsJournalists, ID_BusynessJournalists)
VALUES
(1, 'Агафонова', 'Наталья', 'Юрьевна', 3, '89053599689', 1, 1, 1,
1),
(2, 'Арехина', 'Дарья', 'Петровна', 1, '89093357667', 2, 2, 1, 2),
(3, 'Босов', 'Андрей', 'Витальевич', 8, '89092228666', 2, 4, 3,
1),

```

```

(4, 'Борисов', 'Григорий', 'Геннадьевич', 2, '89055558226', 3, 3,
4, 6),
(5, 'Быстрова', 'Анна', 'Алексеевна', 8, '89088786661', 2, 4, 2,
1),
(6, 'Гришина', 'Ольга', 'Алексеевна', 7, '89093888666', 2, 1, 1,
4),
(7, 'Зотов', 'Георгий', 'Владимирович', 7, '89083546667', 4, 3, 1,
3),
(8, 'Карпов', 'Василий', 'Иванович', 3, '89093568676', 5, 5, 4,
1),
(9, 'Кириелов', 'Андрей', 'Борисович', 5, '89043548666', 2, 2, 1,
6),
(10, 'Кириелова', 'Анна', 'Александровна', 6, '89053558666', 3, 2,
3, 2),
(11, 'Коробина', 'Наталья', 'Андреевна', 10, '89053556653', 4, 5,
2, 2),
(12, 'Курилова', 'Александра', 'Сергеевна', 9, '89053558677', 2, 1,
1, 5),
(13, 'Латов', 'Сергей', 'Сергеевич', 7, '89051158666', 1, 1, 2, 3),
(14, 'Никифорова', 'Ольга', 'Владимировна', 4, '89093782234', 3, 3,
1, 4),
(15, 'Савичев', 'Алексей', 'Алексеевич', 4, '89093444666', 2, 2, 4,
5);
INSERT INTO ExperienceJournalists (NameExperience)
VALUES
('Стажер/ассистент'),
('Внештатный корреспондент'),
('Корреспондент'),
('Редактор'),
('Главный редактор');
INSERT INTO SkillsJournalists (NameSkills)
VALUES
('Писательство'),
('Редактирование'),
('Интервьюирование'),
('Исследование');
INSERT INTO BusynessJournalists (NameBusyness)
VALUES
('Полная занятость'),
('Удалённая работа'),
('Проектная работа'),
('Подработка'),
('Частичная занятость'),
('Гибкий график');
INSERT INTO CitySearch (NameCitySearch)
VALUES
('Архангельск'),
('Владивосток'),
('Волгоград'),
('Вологда'),
('Камчатка'),
('Нижний Новгород'),
('Пенза'),

```

```

('Санкт-Петербург'),
('Саратов'),
('Сочи');
INSERT INTO EducationJournalists (NameEducation)
VALUES
('СПО - бакалавриат'),
('ВПО - бакалавриат/специалитет'),
('2ВПО - магистратура'),
('3ВПО - аспирантура'),
('Докторская степень');
CREATE PROCEDURE GetJournalistsByCity (@cityName NVARCHAR(100))
AS
BEGIN
    SELECT
        j.ID_JournalistsCode AS 'Идентификатор',
        j.SurnameJournalists AS 'Фамилия',
        j.NameJournalists AS 'Имя',
        j.PatronymicJournalists AS 'Отчество',
        j.PhoneNumber AS 'Телефон',
        e.NameExperience AS 'Опыт работы',
        sk.NameSkills AS 'Навыки',
        b.NameBusyness AS 'Уровень занятости',
        c.NameCitySearch AS 'Город',
        ed.NameEducation AS 'Образование'
    FROM
        Journalists j
    INNER JOIN
        CitySearch c ON j.ID_CitySearch = c.ID_CitySearch
    INNER JOIN
        SkillsJournalists sk ON j.ID_SkillsJournalists =
sk.ID_SkillsJournalists
    INNER JOIN
        ExperienceJournalists e ON j.ID_ExperienceJournalists =
e.ID_ExperienceJournalists
    INNER JOIN
        BusynessJournalists b ON j.ID_BusynessJournalists =
b.ID_BusynessJournalists
    INNER JOIN
        EducationJournalists ed ON j.ID_EducationJournalists =
ed.ID_EducationJournalists
    WHERE
        c.NameCitySearch = @cityName;
END;
GO
EXEC GetJournalistsByCity @cityName = Пенза
DROP PROCEDURE IF EXISTS GetJournalistsByCity;
SELECT
    j.ID_JournalistsCode AS 'Идентификатор',
    j.SurnameJournalists AS 'Фамилия',
    j.NameJournalists AS 'Имя',
    j.PatronymicJournalists AS 'Отчество',
    j.PhoneNumber AS 'Телефон',
    e.NameExperience AS 'Опыт работы',

```

```

        sk.NameSkills AS 'Навыки',
        b.NameBusyness AS 'Уровень занятости',
        c.NameCitySearch AS 'Город',
        ed.NameEducation AS 'Образование'
FROM
    Journalists j
INNER JOIN
    CitySearch c ON j.ID_CitySearch = c.ID_CitySearch
INNER JOIN
    SkillsJournalists sk ON j.ID_SkillsJournalists =
sk.ID_SkillsJournalists
INNER JOIN
    ExperienceJournalists e ON j.ID_ExperienceJournalists =
e.ID_ExperienceJournalists
INNER JOIN
    BusynessJournalists b ON j.ID_BusynessJournalists =
b.ID_BusynessJournalists
INNER JOIN
    EducationJournalists ed ON j.ID_EducationJournalists =
ed.ID_EducationJournalists
JOIN CitySearch c ON j.ID_CitySearch = c.ID_CitySearch
JOIN ExperienceJournalists e ON j.ID_ExperienceJournalists =
e.ID_ExperienceJournalists
JOIN EducationJournalists ed ON j.ID_EducationJournalists =
ed.ID_EducationJournalists
WHERE c.NameCitySearch = 'Указанный город'
AND e.NameExperience = 'Указанный опыт'
AND ed.NameEducation = 'Указанное образование';
-----Создание новых таблиц-----
-- Таблица Специализации журналистов
CREATE TABLE Table_Specializations (
    SpecializationID INTEGER PRIMARY KEY CHECK (SpecializationID
>=1 AND SpecializationID <= 9999) IDENTITY(1,1),
    SpecializationName NVARCHAR(100) NOT NULL UNIQUE);
SELECT * FROM Table_Specializations order by SpecializationID;
SELECT * FROM Table_Genres;
SELECT * FROM Table_Articles;
SELECT * FROM Journalist_Specializations;
-- Добавляем специализации
INSERT INTO Table_Specializations (SpecializationName) VALUES
('Печатная журналистика');
INSERT INTO Table_Specializations (SpecializationName) VALUES
('Телевизионная журналистика');
INSERT INTO Table_Specializations (SpecializationName) VALUES
('Радиожурналистика');
INSERT INTO Table_Specializations (SpecializationName) VALUES
('Интернет-журналистика');
INSERT INTO Table_Specializations (SpecializationName) VALUES
('Фотожурналистика');
INSERT INTO Table_Specializations (SpecializationName) VALUES
('Журналистика расследования');
INSERT INTO Table_Specializations (SpecializationName) VALUES
('Спортивная журналистика');

```

```

INSERT INTO Table_Specializations (SpecializationName) VALUES
('Культурная журналистика');
INSERT INTO Table_Specializations (SpecializationName) VALUES
('Экономическая и финансовая журналистика');
INSERT INTO Table_Specializations (SpecializationName) VALUES
('Научная журналистика');
INSERT INTO Table_Specializations (SpecializationName) VALUES
('Туристическая журналистика');
INSERT INTO Table_Specializations (SpecializationName) VALUES
('Экологическая журналистика');
INSERT INTO Table_Specializations (SpecializationName) VALUES
('Военная журналистика');
INSERT INTO Table_Specializations (SpecializationName) VALUES
('Медицинская журналистика');
INSERT INTO Table_Specializations (SpecializationName) VALUES
('Международная журналистика');
INSERT INTO Table_Specializations (SpecializationName) VALUES
('Фэшн-журналистика');
-- Таблица жанров
CREATE TABLE Table_Genres (
    GenreID INTEGER PRIMARY KEY CHECK (GenreID >=1 AND GenreID <=
9999) IDENTITY(1,1),
    GenreName NVARCHAR(100) NOT NULL UNIQUE
);
-- Заполняем таблицу жанров
INSERT INTO Table_Genres (GenreName) VALUES ('Заметка');
INSERT INTO Table_Genres (GenreName) VALUES ('Очерк');
INSERT INTO Table_Genres (GenreName) VALUES ('Интервью');
INSERT INTO Table_Genres (GenreName) VALUES ('Репортаж');
INSERT INTO Table_Genres (GenreName) VALUES ('Лонгрид');
INSERT INTO Table_Genres (GenreName) VALUES ('Сторителлинг');
INSERT INTO Table_Genres (GenreName) VALUES ('Инфографика');
INSERT INTO Table_Genres (GenreName) VALUES ('Слайд-шоу');
-- Главная таблица статей
CREATE TABLE Table_Articles (
    ArticleID INTEGER PRIMARY KEY CHECK (ArticleID >=1 AND
ArticleID <= 9999) IDENTITY(1,1),
    Title NVARCHAR(255) NOT NULL,
    Content NTEXT,
    DatePublished DATETIME DEFAULT GETDATE(),
    AuthorID INTEGER NOT NULL FOREIGN KEY REFERENCES
Journalists(ID_JournalistsCode),
    GenreID INTEGER NOT NULL FOREIGN KEY REFERENCES
Table_Genres(GenreID));
-- Вспомогательная таблица для связи статей и специализированных
направлений журналистов
CREATE TABLE Journalist_Specializations (
    JournalistID INTEGER NOT NULL FOREIGN KEY REFERENCES
Journalists(ID_JournalistsCode),
    SpecializationID INTEGER NOT NULL FOREIGN KEY REFERENCES
Table_Specializations(SpecializationID),
    PRIMARY KEY(JournalistID, SpecializationID));

```

```

INSERT INTO Journalist_Specializations (JournalistID,
SpecializationID) --журналист может писать в разных жанрах
VALUES
(1, 1), (1, 2), (2, 3), (2, 4), (3, 5),
(4, 1), (4, 3), (5, 2), (5, 4), (6, 5),
(7, 1), (8, 2), (9, 3), (10, 4), (11, 5);
INSERT INTO Table_Articles (Title, Content, DatePublished,
AuthorID, GenreID)
VALUES
('Новейшие технологии', 'Статья посвящена последним достижениям в
области технологий.', '2023-01-01', 1, 1),
('История древнего мира', 'Исследования древних цивилизаций.',
'2023-02-01', 2, 2),
('Репортаж из Парижа', 'Очерк о жизни французской столицы.',
'2023-03-01', 3, 3),
('Проблемы экологии', 'Обзор экологических проблем
современности.', '2023-04-01', 4, 4),
('Искусство эпохи Возрождения', 'Анализ искусства периода
Ренессанса.', '2023-05-01', 5, 5),
('Путешествие вокруг света', 'Описание кругосветного
путешествия.', '2023-06-01', 6, 1),
('Политика XXI века', 'Современная политическая ситуация.', '2023-
07-01', 7, 2),
('Культура Древней Руси', 'Исторические аспекты русской
культуры.', '2023-08-01', 8, 3),
('Экономика будущего', 'Прогнозы развития мировой экономики.',
'2023-09-01', 9, 4),
('Литературные шедевры', 'Классическая литература разных эпох.',
'2023-10-01', 10, 5),
('Наука и техника', 'Современные научные открытия.', '2023-11-01',
11, 1),
('Мировые события', 'Важнейшие международные происшествия.',
'2023-12-01', 12, 2),
('Жизнь и искусство', 'Биографии известных художников.', '2024-01-
01', 13, 3),
('Спортивные достижения', 'Рекорды и выдающиеся спортсмены.',
'2024-02-01', 14, 4),
('Фантастическое будущее', 'Научная фантастика и её влияние.',
'2024-03-01', 15, 5);

```

**Приложение Б
(обязательное)
Текст программы**

```

registrvxod.html
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Регистрация</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }
    .register-container {
      background-color: #fff;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }
    .register-container h2 {
      margin-bottom: 20px;
    }
    .register-container input {
      width: 100%;
      padding: 10px;
      margin: 10px 0;
      border: 1px solid #ccc;
      border-radius: 4px;
    }
    .register-container button {
      width: 100%;
      padding: 10px;
      background-color: #007BFF;
      border: none;
      border-radius: 4px;
      color: #fff;
      font-size: 16px;
      cursor: pointer;
    }
    .register-container button:hover {
      background-color: #0056b3;
    }
  </style>
</head>
<body>
  <div class="register-container">
    <h2>Регистрация</h2>

```

```

        <form action="/register" method="post">
            <input type="text" name="surname"
placeholder="Фамилия" required>
            <input type="text" name="name" placeholder="Имя"
required>
            <input type="text" name="patronymic"
placeholder="Отчество" required>
            <input type="text" name="phone" placeholder="Телефон"
required>
            <input type="text" name="city" placeholder="Город"
required>
            <input type="text" name="education"
placeholder="Образование" required>
            <input type="text" name="experience"
placeholder="Опыт" required>
            <input type="text" name="skills" placeholder="Навыки"
required>
            <input type="text" name="busyness"
placeholder="Занятость" required>
            <input type="password" name="password"
placeholder="Пароль" required>
            <button type="submit">Зарегистрироваться</button>
        </form>
        <p>Уже есть аккаунт? <a href="login.html">Войдите</a></p>
    </div>
</body>
</html>
vxodperecluch.html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Регистрация и Вход</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
        }
        .container {
            background-color: #fff;
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            width: 300px;
        }
        .tabs {
            display: flex;

```

```

        border-bottom: 1px solid #ccc;
    }
    .tabs button {
        background-color: #fff;
        border: none;
        border-bottom: 2px solid transparent;
        padding: 10px 20px;
        cursor: pointer;
        font-size: 16px;
    }
    .tabs button.active {
        border-bottom: 2px solid #007BFF;
        color: #007BFF;
    }
    .form-container {
        display: none;
    }
    .form-container.active {
        display: block;
    }
    .form-container h2 {
        margin-bottom: 20px;
    }
    .form-container input {
        width: 100%;
        padding: 10px;
        margin: 10px 0;
        border: 1px solid #ccc;
        border-radius: 4px;
    }
    .form-container button {
        width: 100%;
        padding: 10px;
        background-color: #007BFF;
        border: none;
        border-radius: 4px;
        color: #fff;
        font-size: 16px;
        cursor: pointer;
    }
    .form-container button:hover {
        background-color: #0056b3;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="tabs">
            <button class="tab-button active" data-
target="register">Регистрация</button>
            <button class="tab-button" data-
target="login">Вход</button>
        </div>

```

```

        <div class="form-container active" id="register">
            <h2>Регистрация</h2>
            <form action="/register" method="post">
                <input type="text" name="surname"
placeholder="Фамилия" required>
                <input type="text" name="name" placeholder="Имя"
required>
                <input type="text" name="patronymic"
placeholder="Отчество" required>
                <input type="text" name="phone"
placeholder="Телефон" required>
                <input type="text" name="city" placeholder="Город"
required>
                <input type="text" name="education"
placeholder="Образование" required>
                <input type="text" name="experience"
placeholder="Опыт" required>
                <input type="text" name="skills"
placeholder="Навыки" required>
                <input type="text" name="busyness"
placeholder="Занятость" required>
                <input type="password" name="password"
placeholder="Пароль" required>
                <button type="submit">Зарегистрироваться</button>
            </form>
        </div>
        <div class="form-container" id="login">
            <h2>Вход</h2>
            <form action="/login" method="post">
                <input type="text" name="username"
placeholder="Имя пользователя" required>
                <input type="password" name="password"
placeholder="Пароль" required>
                <button type="submit">Войти</button>
            </form>
        </div>
    </div>
    <script>
        document.querySelectorAll('.tab-button').forEach(button =>
        {
            button.addEventListener('click', () => {
                document.querySelectorAll('.tab-
button').forEach(btn => btn.classList.remove('active'));
                document.querySelectorAll('.form-
container').forEach(form => form.classList.remove('active'));
                button.classList.add('active');

                document.getElementById(button.dataset.target).classList.add('acti
ve');

            });
        });
    </script>
</body>

```

```

</html>
registracdanix.html
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Регистрация и Вход</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }
    .container {
      background-color: #fff;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
      width: 300px;
    }
    .tabs {
      display: flex;
      border-bottom: 1px solid #ccc;
    }
    .tabs button {
      background-color: #fff;
      border: none;
      border-bottom: 2px solid transparent;
      padding: 10px 20px;
      cursor: pointer;
      font-size: 16px;
    }
    .tabs button.active {
      border-bottom: 2px solid #007BFF;
      color: #007BFF;
    }
    .form-container {
      display: none;
    }
    .form-container.active {
      display: block;
    }
    .form-container h2 {
      margin-bottom: 20px;
    }
    .form-container input {
      width: 100%;
      padding: 10px;
    }
  </style>

```

```

        margin: 10px 0;
        border: 1px solid #ccc;
        border-radius: 4px;
    }
    .form-container button {
        width: 100%;
        padding: 10px;
        background-color: #007BFF;
        border: none;
        border-radius: 4px;
        color: #fff;
        font-size: 16px;
        cursor: pointer;
    }
    .form-container button:hover {
        background-color: #0056b3;
    }
    .message {
        margin-top: 20px;
        color: green;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="tabs">
            <button class="tab-button active" data-
target="register">Регистрация</button>
            <button class="tab-button" data-
target="login">Вход</button>
        </div>
        <div class="form-container active" id="register">
            <h2>Регистрация</h2>
            <form id="register-form">
                <input type="text" name="username"
placeholder="Имя пользователя" required>
                <input type="password" name="password"
placeholder="Пароль" required>
                <button type="submit">Зарегистрироваться</button>
            </form>
            <div class="message" id="register-message"></div>
        </div>
        <div class="form-container" id="login">
            <h2>Вход</h2>
            <form id="login-form">
                <input type="text" name="username"
placeholder="Имя пользователя" required>
                <input type="password" name="password"
placeholder="Пароль" required>
                <button type="submit">Войти</button>
            </form>
            <div class="message" id="login-message"></div>
        </div>
    </div>

```

```

</div>
<script>
    document.querySelectorAll('.tab-button').forEach(button =>
{
    button.addEventListener('click', () => {
        document.querySelectorAll('.tab-
button').forEach(btn => btn.classList.remove('active'));
        document.querySelectorAll('.form-
container').forEach(form => form.classList.remove('active'));
        button.classList.add('active');

document.getElementById(button.dataset.target).classList.add('acti
ve');

    });
});
document.getElementById('register-
form').addEventListener('submit', function(event) {
    event.preventDefault();
    const username = event.target.username.value;
    const password = event.target.password.value;
    localStorage.setItem('username', username);
    localStorage.setItem('password', password);
    document.getElementById('register-
message').textContent = 'Вы зарегистрировали аккаунт';
});
document.getElementById('login-
form').addEventListener('submit', function(event) {
    event.preventDefault();
    const username = event.target.username.value;
    const password = event.target.password.value;
    const storedUsername =
localStorage.getItem('username');
    const storedPassword =
localStorage.getItem('password');
    if (username === storedUsername && password ===
storedPassword) {
        document.getElementById('login-
message').textContent = 'Вы вошли';
    } else {
        document.getElementById('login-
message').textContent = 'Неверное имя пользователя или пароль';
    }
});
</script>
</body>
</html>
vivod.html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

```

```

<title>Журналисты</title>
<style>
    body {
        font-family: Arial, sans-serif;
        margin: 20px;
    }
    h1 {
        text-align: center;
    }
    table {
        width: 100%;
        border-collapse: collapse;
    }
    th, td {
        border: 1px solid #ddd;
        padding: 8px;
        text-align: left;
    }
    th {
        background-color: #f2f2f2;
    }
    tr:nth-child(even) {
        background-color: #f9f9f9;
    }
</style>
</head>
<body>
    <h1>Информация о журналистах</h1>
    <table>
        <thead>
            <tr>
                <th>Идентификатор</th>
                <th>Фамилия</th>
                <th>Имя</th>
                <th>Отчество</th>
                <th>Телефон</th>
                <th>Опыт работы</th>
                <th>Навыки</th>
                <th>Уровень занятости</th>
                <th>Город</th>
                <th>Образование</th>
            </tr>
        </thead>
        <tbody>
            <?php
                $servername = "localhost";
                $username = "username";
                $password = "password";
                $dbname = "jour";
                // Создание подключения
                $conn = new mysqli($servername, $username, $password,
$servername);
                // Проверка подключения

```

```

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT
        j.ID_JournalistsCode AS 'Идентификатор',
        j.SurnameJournalists AS 'Фамилия',
        j.NameJournalists AS 'Имя',
        j.PatronymicJournalists AS 'Отчество',
        j.PhoneNumber AS 'Телефон',
        e.NameExperience AS 'Опыт работы',
        sk.NameSkills AS 'Навыки',
        b.NameBusyness AS 'Уровень занятости',
        c.NameCitySearch AS 'Город',
        ed.NameEducation AS 'Образование'
    FROM
        Journalists j
    INNER JOIN
        CitySearch c ON j.ID_CitySearch =
c.ID_CitySearch
    INNER JOIN
        SkillsJournalists sk ON
j.ID_SkillsJournalists = sk.ID_SkillsJournalists
    INNER JOIN
        ExperienceJournalists e ON
j.ID_ExperienceJournalists = e.ID_ExperienceJournalists
    INNER JOIN
        BusynessJournalists b ON
j.ID_BusynessJournalists = b.ID_BusynessJournalists
    INNER JOIN
        EducationJournalists ed ON
j.ID_EducationJournalists = ed.ID_EducationJournalists";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        echo "<tr>
            <td>" . $row['Идентификатор'] . "</td>
            <td>" . $row['Фамилия'] . "</td>
            <td>" . $row['Имя'] . "</td>
            <td>" . $row['Отчество'] . "</td>
            <td>" . $row['Телефон'] . "</td>
            <td>" . $row['Опыт работы'] . "</td>
            <td>" . $row['Навыки'] . "</td>
            <td>" . $row['Уровень занятости'] .
            <td>" . $row['Город'] . "</td>
            <td>" . $row['Образование'] . "</td>
        </tr>";
    }
} else {
    echo "<tr><td colspan='10'>Нет данных</td></tr>";
}
$conn->close();
?>

```

```

        </tbody>
    </table>
</body>
</html>
strstatii.html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Статьи журналистов</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
        }
        h1 {
            text-align: center;
        }
        .article-list {
            list-style-type: none;
            padding: 0;
        }
        .article-list li {
            margin-bottom: 10px;
        }
        .article-list a {
            text-decoration: none;
            color: #0066cc;
            font-weight: bold;
        }
        .article-list a:hover {
            text-decoration: underline;
        }
    </style>
</head>
<body>
    <h1>Статьи журналистов</h1>
    <ul class="article-list">
        <li><a href="articles/1.pdf" target="_blank">Статья
1</a></li>
        <li><a href="articles/2.pdf" target="_blank">Статья
2</a></li>
        <li><a href="articles/3.pdf" target="_blank">Статья
3</a></li>
        <li><a href="articles/4.pdf" target="_blank">Статья
4</a></li>
        <li><a href="articles/5.pdf" target="_blank">Статья
5</a></li>
        <li><a href="articles/6.pdf" target="_blank">Статья
6</a></li>
    </ul>
</body>
</html>

```

```

        <li><a href="articles/7.pdf" target="_blank">Статья
7</a></li>
        <li><a href="articles/8.pdf" target="_blank">Статья
8</a></li>
        <li><a href="articles/9.pdf" target="_blank">Статья
9</a></li>
        <li><a href="articles/10.pdf" target="_blank">Статья
10</a></li>
        <li><a href="articles/11.pdf" target="_blank">Статья
11</a></li>
        <li><a href="articles/12.pdf" target="_blank">Статья
12</a></li>
        <li><a href="articles/13.pdf" target="_blank">Статья
13</a></li>
        <li><a href="articles/14.pdf" target="_blank">Статья
14</a></li>
        <li><a href="articles/15.pdf" target="_blank">Статья
15</a></li>
    </ul>
</body>
</html>
filtr.html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Журналисты</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
        }
        h1 {
            text-align: center;
        }
        .filter-form {
            margin-bottom: 20px;
            padding: 20px;
            background-color: #f9f9f9;
            border-radius: 5px;
        }
        .filter-form label {
            margin-right: 10px;
        }
        .filter-form select {
            margin-right: 20px;
            padding: 5px;
        }
        .filter-form button {
            padding: 5px 10px;
            background-color: #4CAF50;

```

```

        color: white;
        border: none;
        border-radius: 3px;
        cursor: pointer;
    }
    table {
        width: 100%;
        border-collapse: collapse;
    }
    th, td {
        border: 1px solid #ddd;
        padding: 8px;
        text-align: left;
    }
    th {
        background-color: #f2f2f2;
    }
    tr:nth-child(even) {
        background-color: #f9f9f9;
    }
</style>
</head>
<body>
    <h1>Информация о журналистах</h1>
    <div class="filter-form">
        <form method="get" action="">
            <label for="experience">Опыт работы:</label>
            <select name="experience" id="experience">
                <option value="">Все</option>
                <?php
                    $servername = "localhost";
                    $username = "username";
                    $password = "password";
                    $dbname = "jour";

                    // Создание подключения
                    $conn = new mysqli($servername, $username,
$password, $dbname);
                    // Проверка подключения
                    if ($conn->connect_error) {
                        die("Connection failed: " . $conn-
>connect_error);
                    }
                    $sql = "SELECT DISTINCT NameExperience FROM
ExperienceJournalists";
                    $result = $conn->query($sql);
                    if ($result->num_rows > 0) {
                        while($row = $result->fetch_assoc()) {
                            echo "<option value='" .
$row['NameExperience'] . "'>" . $row['NameExperience'] .
"</option>";
                        }
                    }
                }
            </select>
        </form>
    </div>
</body>
</html>

```

```

        ?>
    </select>
    <label for="city">Город:</label>
    <select name="city" id="city">
        <option value="">Все</option>
        <?php
            $sql = "SELECT DISTINCT NameCitySearch FROM
CitySearch";
            $result = $conn->query($sql);
            if ($result->num_rows > 0) {
                while($row = $result->fetch_assoc()) {
                    echo "<option value='" .
$row['NameCitySearch'] . "'>" . $row['NameCitySearch'] .
"</option>";
                }
            }
        ?>
    </select>

    <label for="education">Образование:</label>
    <select name="education" id="education">
        <option value="">Все</option>
        <?php
            $sql = "SELECT DISTINCT NameEducation FROM
EducationJournalists";
            $result = $conn->query($sql);
            if ($result->num_rows > 0) {
                while($row = $result->fetch_assoc()) {
                    echo "<option value='" .
$row['NameEducation'] . "'>" . $row['NameEducation'] .
"</option>";
                }
            }
        ?>
    </select>

    <label for="busyness">Уровень занятости:</label>
    <select name="busyness" id="busyness">
        <option value="">Все</option>
        <?php
            $sql = "SELECT DISTINCT NameBusyness FROM
BusynessJournalists";
            $result = $conn->query($sql);
            if ($result->num_rows > 0) {
                while($row = $result->fetch_assoc()) {
                    echo "<option value='" .
$row['NameBusyness'] . "'>" . $row['NameBusyness'] . "</option>";
                }
            }
        ?>
    </select>

    <label for="skills">Навыки:</label>
    <select name="skills" id="skills">
        <option value="">Все</option>

```

```

        <?php
            $sql = "SELECT DISTINCT NameSkills FROM
SkillsJournalists";
            $result = $conn->query($sql);
            if ($result->num_rows > 0) {
                while($row = $result->fetch_assoc()) {
                    echo "<option value='" .
$row['NameSkills'] . "'>" . $row['NameSkills'] . "</option>";
                }
            }
        ?>
    </select>
    <button type="submit">Применить фильтр</button>
</form>
</div>
<table>
    <thead>
        <tr>
            <th>ФИО</th>
            <th>Телефон</th>
            <th>Опыт работы</th>
            <th>Город</th>
            <th>Образование</th>
            <th>Уровень занятости</th>
            <th>Навыки</th>
        </tr>
    </thead>
    <tbody>
        <?php
            $experience = isset($_GET['experience']) ?
$_GET['experience'] : '';
            $city = isset($_GET['city']) ? $_GET['city'] : '';
            $education = isset($_GET['education']) ?
$_GET['education'] : '';
            $busyness = isset($_GET['busyness']) ?
$_GET['busyness'] : '';
            $skills = isset($_GET['skills']) ? $_GET['skills'] :
'';

            $sql = "SELECT
                CONCAT(j.SurnameJournalists, ' ',
j.NameJournalists, ' ', j.PatronymicJournalists) AS 'ФИО',
                j.PhoneNumber AS 'Телефон',
                e.NameExperience AS 'Опыт работы',
                c.NameCitySearch AS 'Город',
                ed.NameEducation AS 'Образование',
                b.NameBusyness AS 'Уровень занятости',
                sk.NameSkills AS 'Навыки'
            FROM
                Journalists j
            INNER JOIN
                CitySearch c ON j.ID_CitySearch =
c.ID_CitySearch
            INNER JOIN

```

```

        SkillsJournalists sk ON
j.ID_SkillsJournalists = sk.ID_SkillsJournalists
        INNER JOIN
        ExperienceJournalists e ON
j.ID_ExperienceJournalists = e.ID_ExperienceJournalists
        INNER JOIN
        BusynessJournalists b ON
j.ID_BusynessJournalists = b.ID_BusynessJournalists
        INNER JOIN
        EducationJournalists ed ON
j.ID_EducationJournalists = ed.ID_EducationJournalists
        WHERE
        (e.NameExperience = '$experience' OR
'$experience' = '')
        AND (c.NameCitySearch = '$city' OR '$city'
= '')
        AND (ed.NameEducation = '$education' OR
'$education' = '')
        AND (b.NameBusyness = '$busyness' OR
'$busyness' = '')
        AND (sk.NameSkills = '$skills' OR
'$skills' = '');"
    $result = $conn->query($sql);
    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            echo "<tr>
                <td>" . $row['ФИО'] . "</td>
                <td>" . $row['Телефон'] . "</td>
                <td>" . $row['Опыт работы'] . "</td>
                <td>" . $row['Город'] . "</td>
                <td>" . $row['Образование'] . "</td>
                <td>" . $row['Уровень занятости'] .
"</td>

                <td>" . $row['Навыки'] . "</td>
            </tr>";
        }
    } else {
        echo "<tr><td colspan='7'>Нет данных</td></tr>";
    }
    $conn->close();
?>
</tbody>
</table>
</body>
</html>

```

Приложение В
(обязательное)
Текст программы

```

phpMyAdmin
-- Удаление таблиц (для очистки перед созданием новой структуры)
DROP TABLE IF EXISTS Journalists;
DROP TABLE IF EXISTS ExperienceJournalists;
DROP TABLE IF EXISTS SkillsJournalists;
DROP TABLE IF EXISTS BusynessJournalists;
DROP TABLE IF EXISTS EducationJournalists;
DROP TABLE IF EXISTS CitySearch;
-- Таблица Образование
CREATE TABLE EducationJournalists (
    ID_EducationJournalists INT AUTO_INCREMENT PRIMARY KEY,
    NameEducation VARCHAR(30) NOT NULL);
-- Таблица Опыт журналистов
CREATE TABLE ExperienceJournalists (
    ID_ExperienceJournalists INT AUTO_INCREMENT PRIMARY KEY,
    NameExperience VARCHAR(30) NOT NULL);
-- Таблица Навыки журналистов
CREATE TABLE SkillsJournalists (
    ID_SkillsJournalists INT AUTO_INCREMENT PRIMARY KEY,
    NameSkills VARCHAR(30) NOT NULL);
-- Таблица Занятость журналистов
CREATE TABLE BusynessJournalists (
    ID_BusynessJournalists INT AUTO_INCREMENT PRIMARY KEY,
    NameBusyness VARCHAR(30) NOT NULL);
-- Город
CREATE TABLE CitySearch (
    ID_CitySearch INT AUTO_INCREMENT PRIMARY KEY,
    NameCitySearch VARCHAR(30) NOT NULL);
-- Таблица Журналисты
CREATE TABLE Journalists (
    ID_JournalistsCode INT AUTO_INCREMENT PRIMARY KEY,
    SurnameJournalists VARCHAR(30) NOT NULL,
    NameJournalists VARCHAR(30) NOT NULL,
    PatronymicJournalists VARCHAR(30) NOT NULL,
    PhoneNumber VARCHAR(11) NOT NULL,
    ID_ExperienceJournalists INT NOT NULL,
    ID_SkillsJournalists INT NOT NULL,
    ID_BusynessJournalists INT NOT NULL,
    ID_CitySearch INT NOT NULL,
    ID_EducationJournalists INT NOT NULL,
    FOREIGN KEY (ID_ExperienceJournalists) REFERENCES
ExperienceJournalists(ID_ExperienceJournalists),
    FOREIGN KEY (ID_SkillsJournalists) REFERENCES
SkillsJournalists(ID_SkillsJournalists),
    FOREIGN KEY (ID_BusynessJournalists) REFERENCES
BusynessJournalists(ID_BusynessJournalists),
    FOREIGN KEY (ID_CitySearch) REFERENCES
CitySearch(ID_CitySearch),
    FOREIGN KEY (ID_EducationJournalists) REFERENCES
EducationJournalists(ID_EducationJournalists));
-- Заполнение данных в таблицы, на которые ссылаются внешние ключи
INSERT INTO ExperienceJournalists (NameExperience)
VALUES

```

```

('Стажер/ассистент'),
('Внештатный корреспондент'),
('Корреспондент'),
('Редактор'),
('Главный редактор');
INSERT INTO SkillsJournalists (NameSkills)
VALUES
('Писательство'),
('Редактирование'),
('Интервьюирование'),
('Исследование');
INSERT INTO BusynessJournalists (NameBusyness)
VALUES
('Полная занятость'),
('Удалённая работа'),
('Проектная работа'),
('Подработка'),
('Частичная занятость'),
('Гибкий график');
INSERT INTO CitySearch (NameCitySearch)
VALUES
('Архангельск'),
('Владивосток'),
('Волгоград'),
('Вологда'),
('Камчатка'),
('Нижний Новгород'),
('Пенза'),
('Санкт-Петербург'),
('Саратов'),
('Сочи');
INSERT INTO EducationJournalists (NameEducation)
VALUES
('СПО - бакалавриат'),
('ВПО - бакалавриат/специалитет'),
('2ВПО - магистратура'),
('3ВПО - аспирантура'),
('Докторская степень');
-- Заполнение данных в таблицу Journalists
INSERT INTO Journalists (SurnameJournalists, NameJournalists,
PatronymicJournalists, PhoneNumber, ID_ExperienceJournalists,
ID_SkillsJournalists, ID_BusynessJournalists, ID_CitySearch,
ID_EducationJournalists)
VALUES
('Агафонова', 'Наталья', 'Юрьевна', '89053599689', 1, 1, 1, 3, 1),
('Арехина', 'Дарья', 'Петровна', '89093357667', 2, 1, 2, 1, 2),
('Босов', 'Андрей', 'Витальевич', '89092228666', 4, 3, 1, 8, 2),
('Борисов', 'Григорий', 'Геннадьевич', '89055558226', 3, 4, 6, 2,
3),
('Быстрова', 'Анна', 'Алексеевна', '89088786661', 4, 2, 1, 8, 2),
('Гришина', 'Ольга', 'Алексеевна', '89093888666', 1, 1, 4, 7, 2),
('Зотов', 'Георгий', 'Владимирович', '89083546667', 3, 1, 3, 7,
4),

```

```

('Карпов', 'Василий', 'Иванович', '89093568676', 5, 4, 1, 3, 5),
('Кириелов', 'Андрей', 'Борисович', '89043548666', 2, 1, 6, 5, 2),
('Кириелова', 'Анна', 'Александровна', '89053558666', 2, 3, 2, 6,
3),
('Коробина', 'Наталья', 'Андреевна', '89053556653', 5, 2, 2, 10,
4),
('Курилова', 'Александра', 'Сергеевна', '89053558677', 1, 1, 5, 9,
2),
('Латов', 'Сергей', 'Сергеевич', '89051158666', 1, 2, 3, 7, 1),
('Никифорова', 'Ольга', 'Владимировна', '89093782234', 3, 1, 4, 4,
3),
('Савичев', 'Алексей', 'Алексеевич', '89093444666', 2, 4, 5, 4,
2);
-- Процедура для поиска журналистов по городу
DELIMITER //
CREATE PROCEDURE GetJournalistsByCity (IN cityName VARCHAR(100))
BEGIN
    SELECT
        j.ID_JournalistsCode AS 'Идентификатор',
        j.SurnameJournalists AS 'Фамилия',
        j.NameJournalists AS 'Имя',
        j.PatronymicJournalists AS 'Отчество',
        j.PhoneNumber AS 'Телефон',
        e.NameExperience AS 'Опыт работы',
        sk.NameSkills AS 'Навыки',
        b.NameBusyness AS 'Уровень занятости',
        c.NameCitySearch AS 'Город',
        ed.NameEducation AS 'Образование'
    FROM
        Journalists j
    INNER JOIN
        CitySearch c ON j.ID_CitySearch = c.ID_CitySearch
    INNER JOIN
        SkillsJournalists sk ON j.ID_SkillsJournalists =
sk.ID_SkillsJournalists
    INNER JOIN
        ExperienceJournalists e ON j.ID_ExperienceJournalists =
e.ID_ExperienceJournalists
    INNER JOIN
        BusynessJournalists b ON j.ID_BusynessJournalists =
b.ID_BusynessJournalists
    INNER JOIN
        EducationJournalists ed ON j.ID_EducationJournalists =
ed.ID_EducationJournalists
    WHERE
        c.NameCitySearch = cityName;
END //
DELIMITER ;
-- Вызов процедуры
CALL GetJournalistsByCity('Пенза');
-- Удаление процедуры
DROP PROCEDURE IF EXISTS GetJournalistsByCity;
-- Запрос для вывода данных

```

```

SELECT
    j.ID_JournalistsCode AS 'Идентификатор',
    j.SurnameJournalists AS 'Фамилия',
    j.NameJournalists AS 'Имя',
    j.PatronymicJournalists AS 'Отчество',
    j.PhoneNumber AS 'Телефон',
    e.NameExperience AS 'Опыт работы',
    sk.NameSkills AS 'Навыки',
    b.NameBusyness AS 'Уровень занятости',
    c.NameCitySearch AS 'Город',
    ed.NameEducation AS 'Образование'
FROM
    Journalists j
INNER JOIN
    CitySearch c ON j.ID_CitySearch = c.ID_CitySearch
INNER JOIN
    SkillsJournalists sk ON j.ID_SkillsJournalists =
sk.ID_SkillsJournalists
INNER JOIN
    ExperienceJournalists e ON j.ID_ExperienceJournalists =
e.ID_ExperienceJournalists
INNER JOIN
    BusynessJournalists b ON j.ID_BusynessJournalists =
b.ID_BusynessJournalists
INNER JOIN
    EducationJournalists ed ON j.ID_EducationJournalists =
ed.ID_EducationJournalists;
-- Таблица Специализации журналистов
CREATE TABLE Table_Specializations (
    SpecializationID INT AUTO_INCREMENT PRIMARY KEY,
    SpecializationName VARCHAR(100) NOT NULL UNIQUE);
-- Таблица Жанров
CREATE TABLE Table_Genres (
    GenreID INT AUTO_INCREMENT PRIMARY KEY,
    GenreName VARCHAR(100) NOT NULL UNIQUE);
-- Главная таблица статей
CREATE TABLE Table_Articles (
    ArticleID INT AUTO_INCREMENT PRIMARY KEY,
    Title VARCHAR(255) NOT NULL,
    Content TEXT,
    DatePublished DATETIME DEFAULT CURRENT_TIMESTAMP,
    AuthorID INT NOT NULL,
    GenreID INT NOT NULL,
    FOREIGN KEY (AuthorID) REFERENCES
Journalists(ID_JournalistsCode),
    FOREIGN KEY (GenreID) REFERENCES Table_Genres(GenreID));
-- Вспомогательная таблица для связи статей и специализированных
направлений журналистов
CREATE TABLE Journalist_Specializations (
    JournalistID INT NOT NULL,
    SpecializationID INT NOT NULL,
    PRIMARY KEY (JournalistID, SpecializationID),

```

```

        FOREIGN KEY (JournalistID) REFERENCES
Journalists(ID_JournalistsCode),
        FOREIGN KEY (SpecializationID) REFERENCES
Table_Specializations(SpecializationID));
-- Добавляем специализации
INSERT INTO Table_Specializations (SpecializationName) VALUES
('Печатная журналистика'),
('Телевизионная журналистика'),
('Радиожурналистика'),
('Интернет-журналистика'),
('Фотожурналистика'),
('Журналистика расследования'),
('Спортивная журналистика'),
('Культурная журналистика'),
('Экономическая и финансовая журналистика'),
('Научная журналистика'),
('Туристическая журналистика'),
('Экологическая журналистика'),
('Военная журналистика'),
('Медицинская журналистика'),
('Международная журналистика'),
('Фэшн-журналистика');
-- Заполняем таблицу жанров
INSERT INTO Table_Genres (GenreName) VALUES
('Заметка'),
('Очерк'),
('Интервью'),
('Репортаж'),
('Лонгрид'),
('Сторителлинг'),
('Инфографика'),
('Слайд-шоу');
-- Вставка данных в таблицу статей
INSERT INTO Table_Articles (Title, Content, DatePublished,
AuthorID, GenreID) VALUES
('Новейшие технологии', 'Статья посвящена последним достижениям в
области технологий.', '2023-01-01', 1, 1),
('История древнего мира', 'Исследования древних цивилизаций.',
'2023-02-01', 2, 2),
('Репортаж из Парижа', 'Очерк о жизни французской столицы.',
'2023-03-01', 3, 3),
('Проблемы экологии', 'Обзор экологических проблем
современности.', '2023-04-01', 4, 4),
('Искусство эпохи Возрождения', 'Анализ искусства периода
Ренессанса.', '2023-05-01', 5, 5),
('Путешествие вокруг света', 'Описание кругосветного
путешествия.', '2023-06-01', 6, 1),
('Политика XXI века', 'Современная политическая ситуация.', '2023-
07-01', 7, 2),
('Культура Древней Руси', 'Исторические аспекты русской
культуры.', '2023-08-01', 8, 3),
('Экономика будущего', 'Прогнозы развития мировой экономики.',
'2023-09-01', 9, 4),

```

```
( 'Литературные шедевры', 'Классическая литература разных эпох.',
'2023-10-01', 10, 5),
( 'Наука и техника', 'Современные научные открытия.', '2023-11-01',
11, 1),
( 'Мировые события', 'Важнейшие международные происшествия.',
'2023-12-01', 12, 2),
( 'Жизнь и искусство', 'Биографии известных художников.', '2024-01-
01', 13, 3),
( 'Спортивные достижения', 'Рекорды и выдающиеся спортсмены.',
'2024-02-01', 14, 4),
( 'Фантастическое будущее', 'Научная фантастика и её влияние.',
'2024-03-01', 15, 5);
```

```
-- Вставка данных в таблицу связи журналистов и специализаций
INSERT INTO Journalist_Specializations (JournalistID,
SpecializationID) VALUES
(1, 1), (1, 2), (2, 3), (2, 4), (3, 5),
(4, 1), (4, 3), (5, 2), (5, 4), (6, 5),
(7, 1), (8, 2), (9, 3), (10, 4), (11, 5);
```

**Приложение Г
(обязательное)
Диаграммы**

[illegible]

Рисунок 2 - Диаграмма Ганта

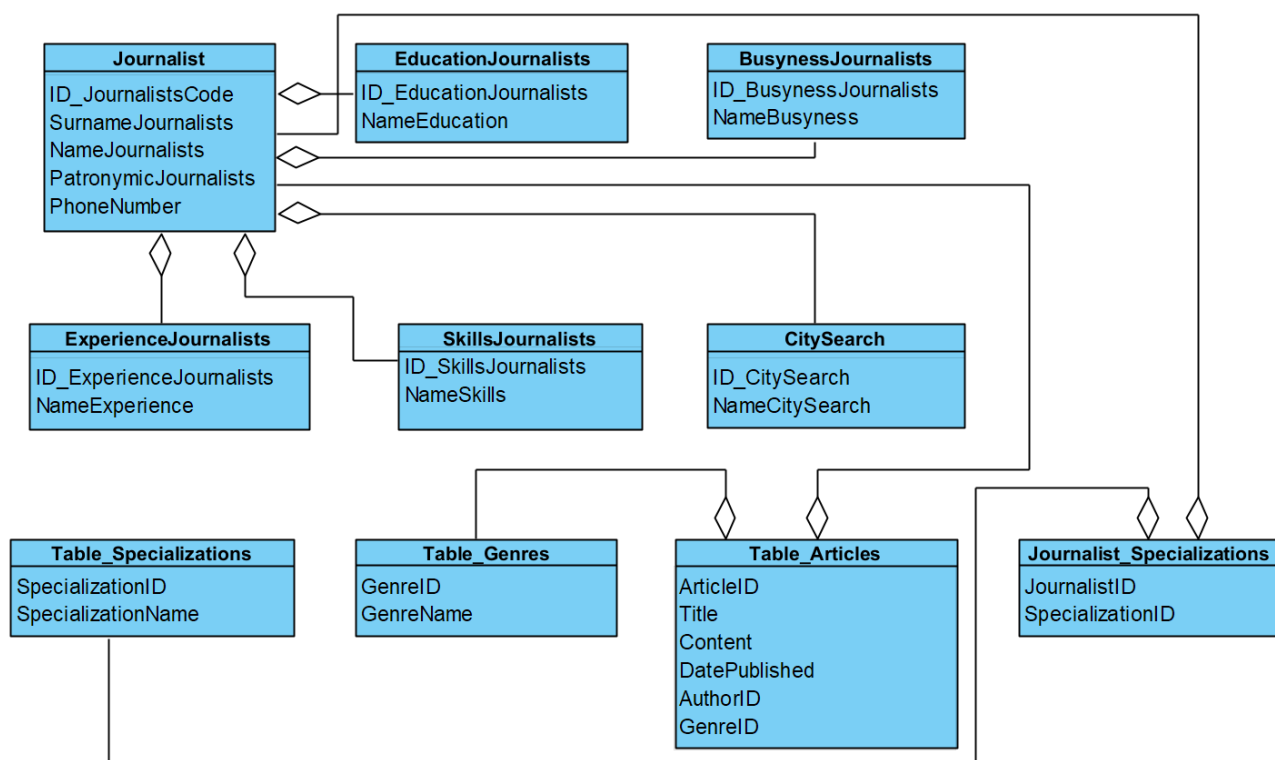


Рисунок 3 – Диаграмма классов

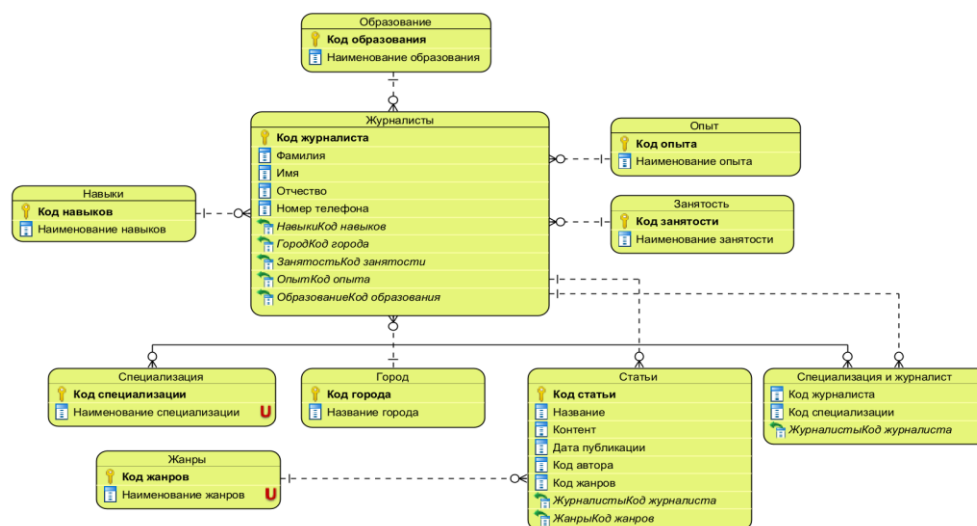


Рисунок 4 – Диаграмма сущность-связь концептуальной информационной модели Web-платформы

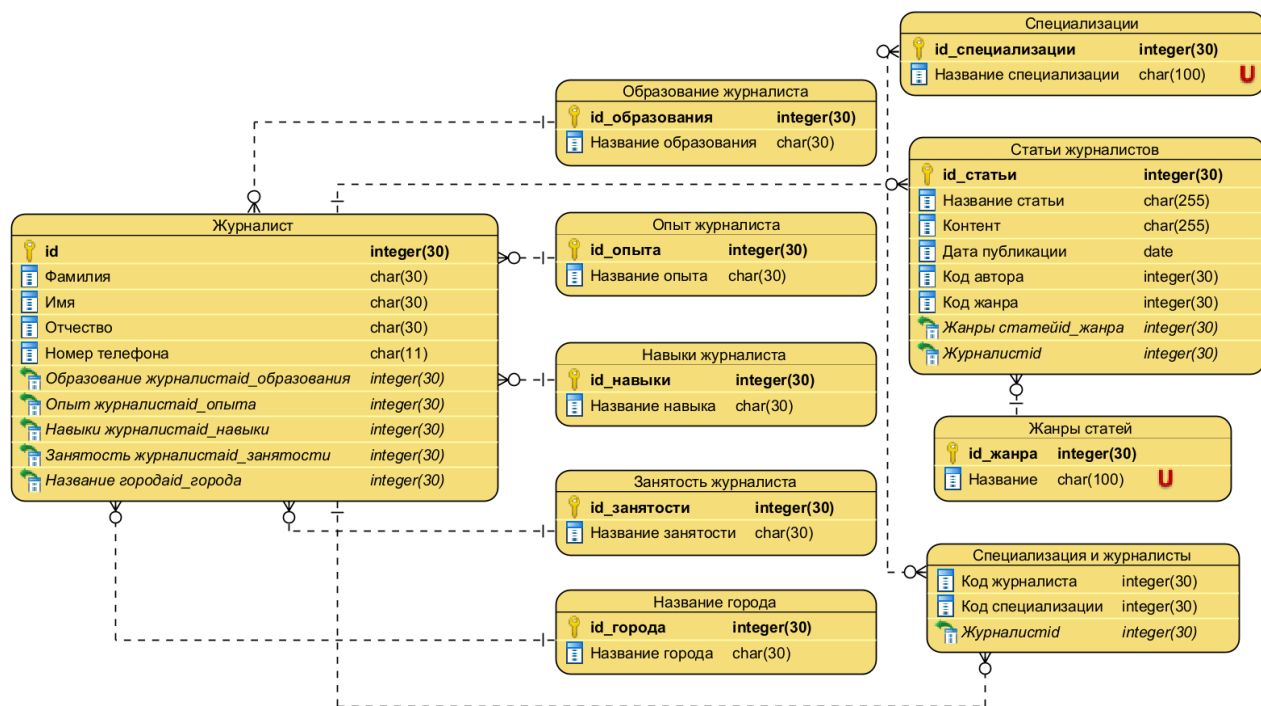


Рисунок 5 – Логическая схема базы данных информационной модели Web-платформы

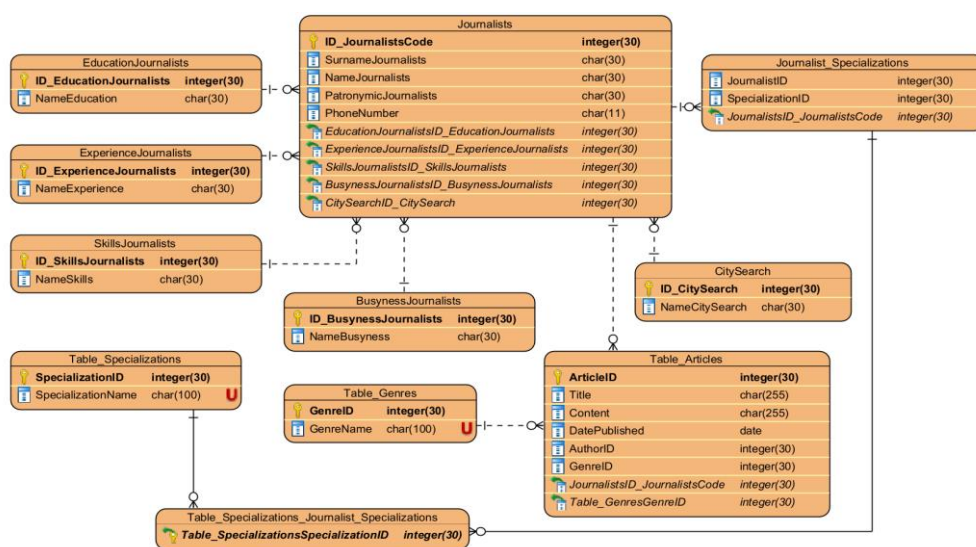


Рисунок 6 – Физическая схема базы данных

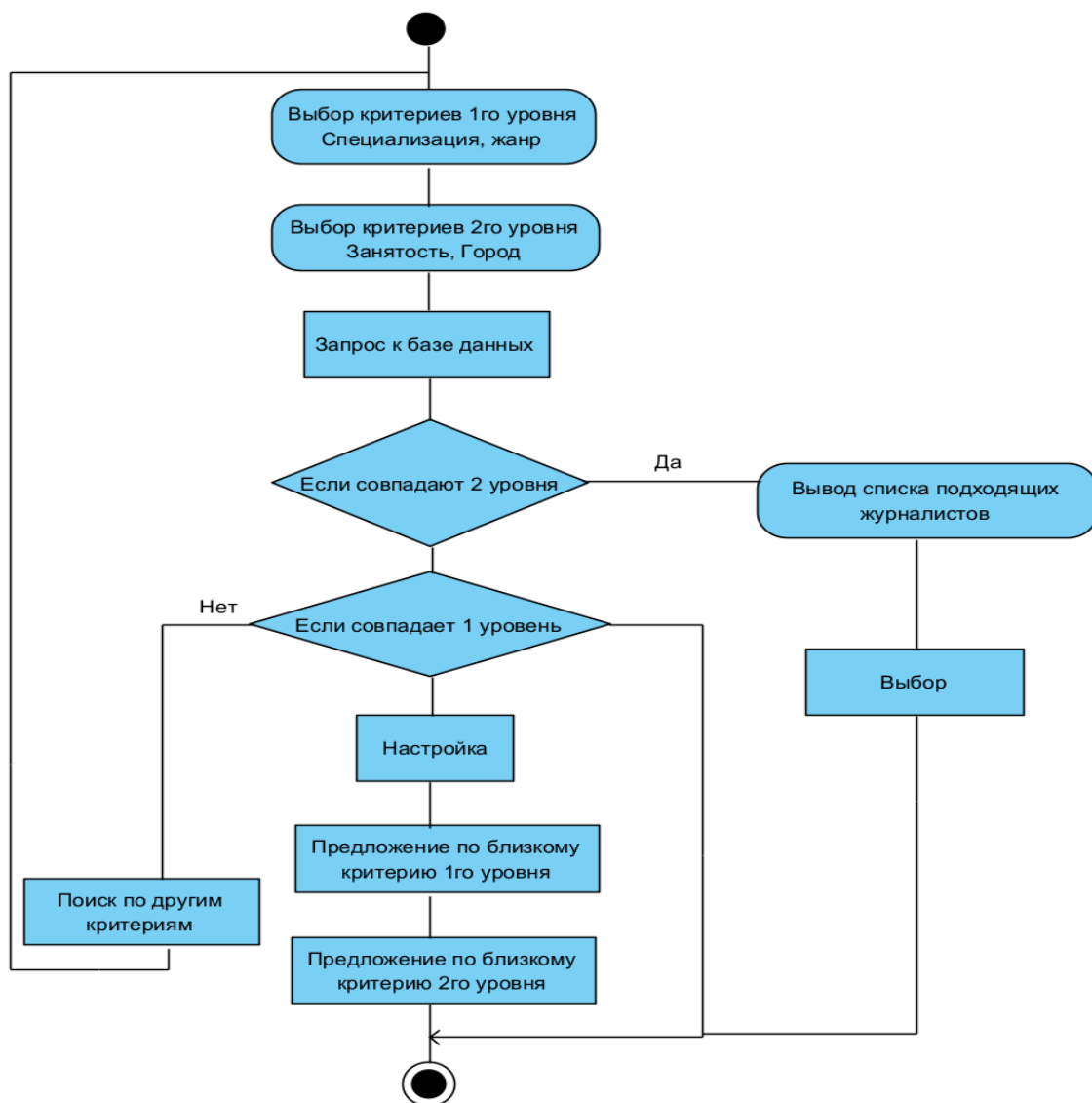


Рисунок 7 – Диаграмма деятельности подбора резюме журналистов

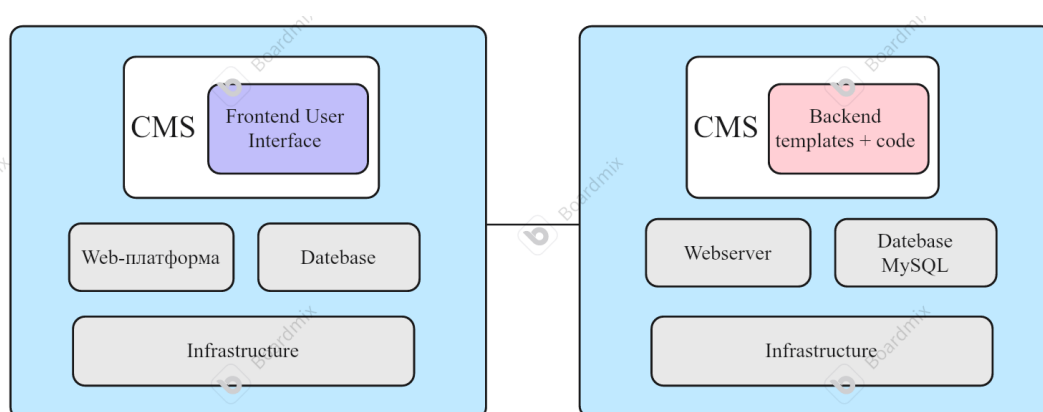


Рисунок 8 – Диаграмма компонентов

**Приложение Д
(рекомендуемое)
Результат тестирования**

Регистрация

Имя:
Ира

Email:
pruSCA903aKL@yandex.ru

Пароль:

Зарегистрироваться

[Уже есть аккаунт? Войти](#)

Рисунок 1 – Авторизация

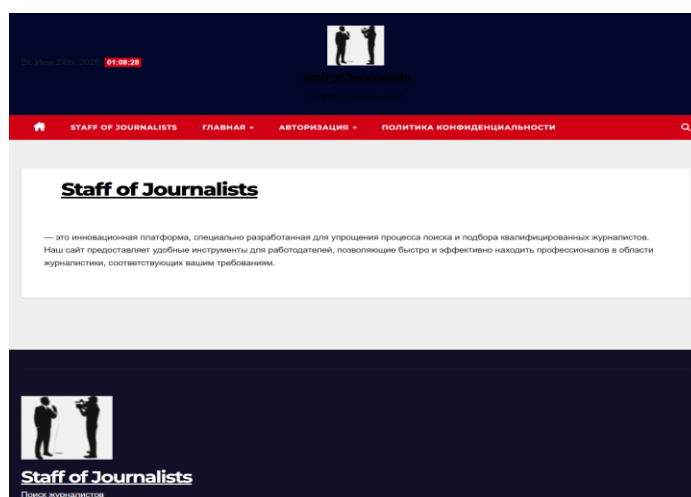


Рисунок 2 – Главная страница сайта

Поиск

Резюме

Фильтры

Специализация журналиста:
Печатная журналистика

Занятость журналиста:
Полная занятость

Город:
Москва

Иван Иванов

Специализация: Печатная журналистика

Занятость: Полная занятость

Город: Москва

Опыт работы: 5 лет в области журналистики, работа в различных изданиях.

Навыки: Написание статей, Редактирование, Интервьюирование

Проекты: Проект 'Городские истории', Колонка 'Культурный обзор'

Рисунок 3 – Поиск резюме

Статьи

Жанр статьи:

Репортаж

Поиск

Экологические вызовы современности

Жанр: Репортаж

Автор: [Анна Кузнецова](#)

Содержание: Репортаж о текущих экологических проблемах и путях их решения.

Рисунок 4 – Поиск статей

Резюме

Управление резюме

Создать резюме

Имя:

Специализация:

Выберите специализацию

Занятость:

Выберите тип занятости

Город:

Выберите город

Опыт работы:

Навыки:

Создать резюме

Опубликованные резюме

Ирина

Специализация: Культурная журналистика

Занятость: Полная занятость

Город: Пенза

Опыт работы: 1 год

Навыки: Писательство

Скрыть

Редактировать

Рисунок 5 – Управление резюме

Генерация резюме

Иван Иванов

Сгенерировать резюме

Резюме: Иван Иванов

Специализация: Печатная журналистика

Город: Москва

Образование: Московский Государственный Университет, факультет журналистики

Опыт работы: 5 лет в области журналистики, работа в различных изданиях.

Навыки: Написание статей, Редактирование, Интервьюирование, Анализ данных

Языки: Русский, Английский, Немецкий

Проекты:

- Проект "Городские истории" - серия статей о жизни в мегаполисе.
- Участие в создании еженедельной колонки "Культурный обзор".

Достижения:

- Лауреат премии "Лучший журналист года" в 2020 году.
- Публикация более 200 статей в ведущих изданиях.

Рисунок 6 – Генерация резюме

СТАТЬИ

Уведомление от сайта f1123561.xsph.ru

Файл "4.pdf" загружен.

Закреть

Статьи

Управление ста

Загрузить статью (PDF):

Выберите файл 4.pdf

Загрузить

Пример статьи.pdf

Открыть

Скачать

Удалить

Изменить

Рисунок 7 – Управление статьями

Поиск вакансий

Критерии поиска

Специализация:

Печатная журналистика

Город:

Москва

Тип занятости:

Полная занятость

Поиск

Журналист

Компания:

Издательство "Новости"

Город:

Москва

Тип занятости:

Полная занятость

Описание:

Требуется журналист для написания статей на различные темы.

Откликнуться

Рисунок 8 – Поиск вакансий

Управление вакансиями

Добавить вакансию

Название вакансии:

Описание:

Требования:

Сохранить вакансию

Список вакансий

ID	Название	Описание	Требования	Статус	Действия
1	Журналист	Требуется журналист по интервьюированию, зарплата 30тыс. р.	Стаж не меньше 3х лет	Скрыто	<div>Опубликовать</div> <div>Удалить</div>

Рисунок 9 – Управление вакансиями

JOURNALISTS

ГЛАВНАЯ

АВТОРИЗАЦИЯ

ПОЛИТИКА КОНФИДЕНЦИАЛЬНОСТИ

Техническая поддержка

Администратор: Добро пожаловать в службу поддержки! Как мы можем вам помочь?

Вы: Не могу загрузить файл

Администратор: Спасибо за ваше сообщение! Мы рассмотрим ваш вопрос и свяжемся с вами в ближайшее время.

Введите ваше сообщение...

Отправить

Рисунок 10 – Техническая поддержка

Профиль

Редактировать профиль

Имя:
Ира

Фамилия:
Евр

Email:
pruSCA903aKL@yandex.ru

Город:
Пенза

Специализация:
Печатная журналистика

Опыт работы:
1 год

Навыки:
Писательство

Рисунок 11 – Профиль

Личный кабинет

Редактировать профиль

Название компании:
ООО

Email:
pruSCA903aKL@yandex.ru

Описание компании:
ООО

[Сохранить изменения](#)

Рисунок 12 – Личный кабинет

Политика конфиденциальности

Кто мы

Настоящая Политика конфиденциальности описывает, как мы собираем, используем, раскрываем и защищаем информацию, полученную от пользователей нашего сайта. Мы уважаем вашу конфиденциальность и стремимся защитить ваши личные данные.
Наш адрес сайта: <http://it1123561.xsph.ru/>.

Сбор информации

Личная информация:

- Мы можем собирать личную информацию, такую как имя, адрес электронной почты, номер телефона и другие контактные данные, когда вы регистрируетесь на нашем сайте, подписываетесь на рассылку или заполняете формы обратной связи.

Неперсональная информация:

- Мы также можем собирать неперсональную информацию, такую как IP-адрес, тип браузера, операционная система, страницы, которые вы посещаете на нашем сайте, и время посещения.

1.

Использование информации

Личная информация:

- Мы используем личную информацию для предоставления вам услуг, поддержки клиентов, отправки уведомлений и обновлений, а также для улучшения нашего сайта и услуг.

Рисунок 12 – Политика конфиденциальности