Ministry of Science and Higher Education of the Russian Federation

Penza State University

Department of Computer Engineering

REPORT on the course work

Topic «Development of a program of complex structure using the top-down programming method»

Technical executor:

student of group ~~Saad Hmaidi,~~

group 23VVia1 FIKRI MOHAMED

Checker: Fedyunin R.N.

3.10.2024

Penza

2024

CamScanner

PENZA STATE UNIVERSITY

Faculty of Computer Science Department of Computer Science

APPROVED

Head Department of VT

_____ M.A. Mitrokhin

«____»_____ 2024 year

THE TASK LIST

for the course task

Programming

Student Mohamed Fikri. Group 23ВВив1

Project topic: **Development of a program of complex structure using the top-down programming method.**

Initial data (technical requirements) for design Develop a **Grocery Store Managment System** where the user (owner) can add and delete products and orders and control orders and total price of all orders or a specific order

Mandatory program requirements:

1. Multimodularity.
2. Use of complex data types – functions, structures, lists, Dicts.
3. Video system operating mode – text / graphic.
4. Information input device – keyboard, mouse.
5. Availability of a screensaver.
6. MySQL – database.
7. UI – HTML, CSS, JS.
8. Server – Flask.
9. Operating system MS Windows.
10. Programming language – Python.
11. Software development environment – Pycharm community.

CamScanner

Scope of work for the course

Calculation part

Program development.

The grafical part

Data diagram, system resource diagram, system operation diagram, hierarchical structure of the program, program interaction diagram.

Experimental part

Debugging the program.

Project completion time by section

According to schedule.

May 5 2024

The date of issue of the task is "~~October 1~~", ~~2024~~.

Teacher

Project defense date "____"_____

~~Head Golotenkov N. O.~~ Fedicunin R.N.

Received the task on ~~October 3, 2024.~~ May 5 2024

StudentMohamed Fikri.

CamScanner

Content

1. **Introduction**..... 5
   - System Architecture.....
   - Key Features.....
   - Objective.....
2. **Algorithmic**.....8
   - UML Diagrams.....
     - UML Class Diagram.....
     - UML Sequence Diagram.....
     - UML Case Diagram.....
   - WorkFlow Chart.....
3. **Program part**..... 11
   - Explination of the work.....
4. **Program listing**..... 24
   - Source Code.....
   - Database.....
   - Database Diagram.....
   - Output.....
5. **Bibliography**..... 35
   - References.....
6. **Conclusions on the work**..... 36
   - Conclusion.....
   - Key Features and Highlights.....
   - Personal Learning and Skill Development..
   - Final Thoughts.....

CamScanner

Bibliography

References:

https://chatgpt.com/

https://www.youtube.com/@codebasics

https://www.w3schools.com/python/

https://www.w3schools.com/sql/

https://dev.mysql.com/doc/relnotes/connector-python/en/news-9-0-0.html

https://www.programiz.com/python-programming

https://www.programiz.com/sql

https://github.com/jonecir/grocerystore

https://github.com/bugaboo22/grocerystore

CamScanner

Conclusions on the work

Conclusion

The Grocery Store Management System developed with Flask serves as a comprehensive solution for efficiently managing various operational aspects of grocery retail. This web-based application facilitates product management, order processing, and unit of measurement handling through a structured and user-friendly API, significantly enhancing the productivity of grocery store operations.

Key Features and Highlights:

1. Robust API Architecture:

- The application employs a RESTful API architecture that enables seamless interaction between the client and server. This design allows users to perform essential operations such as retrieving product information, adding new products, managing orders, and accessing unit of measurement data via straightforward HTTP requests (GET and POST methods).
- Each endpoint is clearly defined, making it easy for front-end applications or third-party integrations to communicate with the backend system.

2. Modular Design with Data Access Objects (DAOs):

- The system utilizes a modular design pattern by implementing DAOs for database interactions. Each DAO (e.g., products_dao, orders_dao, uom_dao, and vieworders_dao) encapsulates the logic for handling specific database operations, such as retrieving, inserting, updating, and deleting records.
- This separation of concerns enhances code maintainability, facilitates testing, and allows for scalability, enabling future enhancements without significant restructuring of the existing codebase.

3. Cross-Origin Resource Sharing (CORS) Support:

- The inclusion of CORS in the response headers allows the application to be accessed from different domains. This feature is crucial for building modern web applications, where front-end and back-end components may be hosted on separate servers or domains, ensuring a smooth user experience.

4. Efficient Data Handling with JSON:

CamScanner