**A MINOR PROJECT REPORT**

**ON**

**SHOPPING APPLICATION**

***Submitted by***

**PRUDVI RAO ERRABELLI (21CSX79L)**

***Under the Guidance of***

**Mr. Abhishek Gupta**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



**Department of Computer Science & Engineering**

**LINGAYAS VIDYAPEETH, FARIDABAD**

**December, 2023**

**Acknowledgement**

I would like to express my sincere gratitude to my supervisor **Mr.Abhishek Gupta, Department of Computer Science and Engineering,** for giving me the opportunity to work on this topic. It would never be possible for us to take this project to this level without their innovative ideas and their relentless support and encouragement.

I express my deepest thanks to **other faculties who have contributed** for taking part in useful decision & giving necessary advices and guidance and arranged all facilities to make life easier. I choose this moment to acknowledge his/her contribution gratefully.

I take immense pleasure in thanking **Dr. Ritu Sindhu**, Dean & Head of Department of Computer Science & Engineering. Her tolerances of having discussions, sacrificing her personal time, are extremely insightful and greatly appreciated.

Prudvi Rao Errabelli, 21CSX79L

**Declaration**

We hereby declare that this project report entitled “**SHOPPING APPLICATION**” *by* **Prudvi Errabelli (21CSX79L),** being submitted in partial fulfillment of the requirements for the degreeof Bachelor of Technology in **Computer Science & Engineering** under the Department of Computer Science and Engineering of Lingayas Vidyapeeth Faridabad, during the academic year \_\_\_\_\_\_2023\_, is a bonafide record of our original work carried out under the guidance of **NAME OF PROJECT\_GUIDE, DESIGNATION , DEPARTMENT.**

We further declare that we have not submitted the matter presented in this Project for the award of any other Degree/Diploma of this University or any other University/Institute.

1. Prudvi Rao Errabelli , 21CSX79L

**TABLE OF CONTENTS**

**Chapter 1. Introduction 1**

* 1. Introduction
  2. Problem Definition
  3. Feasibility Study
  4. Motivation
  5. Project Overview/Specifications
  6. Hardware Specification
  7. Software Specification
  8. Overview of the report

**Chapter 2. System Analysis & Design 6**

2.1 Requirement Specification

2.2 Flowcharts / DFDs / ERDs

2.3 Design and Test Steps / Criteria

2.4 Algorithms and Pseudo Code

2.4.1

2.4.2…

2.5 Testing and Validation

2.5.1 Unit Testing

2.5.2 Integrity Testing

2.5.3 Software Maintenance

**Chapter 3. Implementation and Results 9**

**Chapter 4. Conclusion and Future Enhancements**

4.1 Summary of work done

4.2 Proposal/scope of future enhancement

**References/Bibliography**

**Appendices**

* Project Code
* Progress Report by Guide

**\*\*Chapter 1. Introduction 1\*\***

**\*\*1.1 Introduction\*\***

The Terminal-Based Shopping Application project is a response to the need for a simplified and efficient online shopping experience using a command-line interface (CLI). In an era dominated by graphical user interfaces, this project explores the possibilities of a tex—based approach, offering users an alternative way to interact with an e-commerce platform.

**\*\*1.2 Problem Definition\*\***

Traditional online shopping interfaces often come with graphical complexities, potentially overwhelming some users. The CLI shopping application aims to address this by providing a minimalist and intuitive interface, appealing to users who seek simplicity and functionality over elaborate visuals.

**\*\*1.3 Feasibility Study\*\***

The feasibility study assessed the practicality of developing a terminal-based shopping application. Technical considerations, economic factors, and potential user acceptance were evaluated. The findings indicate a favorable environment for the project, suggesting that it aligns with current trends and demands.

**\*\*1.4 Motivation\*\***

The primary motivation behind this project is to fill a gap in the market for users who prefer a command-line approach to online shopping. Additionally, the project serves as an educational tool for developers interested in exploring and understanding the intricacies of command-line applications.

**\*\*1.5 Project Overview/Specifications\*\***

The project encompasses essential features, including product listing, cart management, and order tracking, within the CLI. Specifications detail user interactions, system behavior, and the overall functionality of the application, guiding the development process.

**\*\*1.6 Hardware Specification\*\***

The application is designed to run on standard computing hardware, emphasizing compatibility with various devices. This approach ensures broad accessibility, allowing users to engage with the CLI application across different platforms.

**\*\*1.7 Software Specification\*\***

To build the CLI application, a suitable programming language for terminal-based environments, such as Python or Go, will be employed. File-based storage simplifies data management, while encryption libraries enhance the security of user information during transactions.

**\*\*1.8 Overview of the Report\*\***

The report’s structure includes subsequent chapters on system analysis and design, implementation and results, and a conclusion with future enhancements. Each chapter provides in-depth insights into the respective stages of the development process, aiming to offer a comprehensive understanding of the Terminal-Based Shopping Application.

**\*\*Chapter 2. System Analysis & Design \*\***

**\*\*2.1 Requirement Specification\*\***

User Interaction

- Users should be able to browse products.

- Users must have the ability to add items to the cart.

- The application should provide a seamless checkout process.

System Functionality

- The application must handle user authentication securely.

- Product information should be stored and retrieved efficiently.

- Cart management functionalities, including add, remove, and view operations, must be implemented.

**\*\*2.2 Flowcharts / DFDs / ERDs\*\***

Flowchart: User Authentication

* Describes the process of user login and authentication.

Entity-Relationship Diagram (ERD): Database Structure

* Illustrates the relationships between entities like users, products, and orders.

**\*\*2.3 Design and Test Steps / Criteria\*\***

Design Principles

- Follow SOLID principles for modular and maintainable code.

- Implement a user-friendly interface using ASCII art for visual appeal.

Test Criteria

- Ensure each function within the application performs as intended.

- Verify that error handling is effective in preventing unexpected behaviors.

**\*\*2.4 Algorithms and Pseudo Code\*\***

\*\*2.4.1 User Authentication Algorithm\*\*

* Describes the steps involved in authenticating a user.

\*\*2.4.2 Product Listing Algorithm\*\*

* Details the process of fetching and displaying products.

\*\*2.5 Testing and Validation\*\*

**\*\*2.5.1 Unit Testing\*\***

- Test individual components, such as authentication and cart management.

- Use testing frameworks to automate and streamline the testing process.

**\*\*2.5.2 Integration Testing\*\***

- Verify the interaction between different modules.

- Ensure seamless integration of user authentication, product listing, and cart management.

**\*\*2.5.3 Software Maintenance\*\***

\*\*Bug Tracking\*\*

- Establish a system for tracking and resolving software bugs.

- Regularly update the application to address security vulnerabilities and enhance functionality.

**\*\*Chapter 3. Implementation and Results \*\***

**\*\*3.1 Implementation\*\***

\*\*Programming Language Selection\*\*

* Chose Python for its simplicity and versatility in terminal-based applications.

\*\*File-Based Storage\*\*

* Implemented a file system to store product and user information.

\*\*Encryption Implementation\*\*

* Utilized encryption libraries to secure sensitive user data during transactions.

\*\*CLI Interface Design\*\*

* Designed an ASCII art-based interface for a visually appealing user experience.

**\*\*3.2 Results\*\***

\*\*User Feedback\*\*

- Conducted user testing to gather feedback on the application’s usability.

- Incorporated user suggestions to enhance the overall user experience.

\*\*Performance Metrics\*\*

- Measured response times for key functions such as product retrieval and checkout.

- Optimized code to improve application responsiveness.

\*\*Security Audit\*\*

- Conducted a security audit to identify and address potential vulnerabilities.

- Ensured secure transmission of data and robust authentication mechanisms.

\*\*Offline Mode Testing\*\*

- Validated the application’s functionality in offline mode.

- Implemented synchronization mechanisms for offline transactions.

**\*\*Chapter 4. Conclusion and Future Enhancements\*\***

**\*\*4.1 Conclusion\*\***

\*\*Achievements\*\*

- Successfully developed a functional and user-friendly terminal-based shopping application.

- Addressed the initial problem of providing a streamlined alternative to graphical interfaces.

\*\*User Adoption\*\*

- Monitored user adoption and received positive feedback on the CLI approach.

- Identified a niche market of users who appreciate the simplicity of the application.

\*\*Lessons Learned\*\*

- Gained insights into designing and implementing command-line applications.

- Acknowledged the importance of user feedback in refining the user interface.

**\*\*4.2 Future Enhancements\*\***

\*\*Feature Expansion\*\*

- Consider incorporating additional features like user reviews and ratings.

- Explore options for integrating external APIs to expand the product catalog dynamically.

\*\*Multi-Platform Support\*\*

- Investigate the possibility of creating versions for different operating systems.

- Ensure compatibility with a broader range of terminal environments.

\*\*Accessibility Improvements\*\*

- Implement accessibility features to cater to users with diverse needs.

- Enhance support for screen readers and keyboard navigation.

\*\*Performance Optimization\*\*

- Continue optimizing code for improved response times.

- Explore options for parallel processing to handle increased user traffic.

\*\*Security Updates\*\*

- Stay vigilant for emerging security threats and update encryption protocols.

- Regularly audit and patch vulnerabilities to maintain a secure application.

**\*\*4.3 Closing Remarks\*\***

The Terminal-Based Shopping Application project has been a valuable exploration into the realm of command-line interfaces for online shopping. The positive reception from users and the insights gained throughout the development process provide a strong foundation for future enhancements and potential expansions. This project stands as a testament to the adaptability of technology and the importance of considering diverse user preferences in software design.

```

**Introduction**

The mini-world we’ve chosen is an eCommerce website, which provides clothing for men,

Women, and kids. This application allows users to shop fashionable clothing online. There

Would be different categories of products like clothes for men, women, kids. Users will be

Able to order the products of their preference and add desired products to their cart. This is

An overview of the miniworld.

**Purpose**

The purpose of the database is to store details of users, keep track of their orders,

Payments, transactions.Beyond the user, It also stores details of suppliers for various

Products, delivery agents, shipping details etc.

**Users**

● Suppliers

● users(customers)

● Organizers

● Delivery agents

**Applications**

● Managers will be able to know the orders and ensure delivery of products.

● We can analyse brands’ performance based on various reviews.

● Marketing teams can analyse potential users and create targeting lists for different

Advertisements.

● Customers will be able to keep track of previous orders, payments, review products

Etc.

**Database Requirements**

Entities

1.User (key attribute-User ID or Email ID)

➢ User ID [10 digits]

➢ Name(First Name,Last Name) [<=20 char] \*composite

➢ Email ID [<=30 char]

➢ Phone no [<=12 digits] \*multivalued

➢ Address(line1,line2,city,district,state,country,pincode) [<=200char]\*composite

➢ Password [<=8 char]

➢ Age[<=100]

➢ Gender [M/F/O]

2.Product (key attribute-Product code)

➢ Product code [10 digits]

➢ Name [<=20 char]

➢ Colour [<=20 char]

➢ Stock [<=10000]

➢ Description [<=500 char]

➢ Product availability [Yes/No]

➢ Manufacturer details(Company name,Address) \*composite

[<=200 char] -composite

➢ Brand name [<=15 char]

➢ Available Sizes(some of XS/S/M/L/XL/XXL) \* multivalued

➢ Original Price

➢ Current price → derived attribute

➢ No of products sold out.

➢ discount

● Sub classes-> men wear, women wear, kids wear

3.Payment key attribute-(Transaction ID)

➢ Transaction ID [12-18 digits]

➢ Mode of payment [<=10 char]

➢ Status [ <=10 char]

➢ Total price (derived attribute)

4.Supplier key attribute-(supplier ID)

➢ Seller ID

➢ Name (First name,Last name) [<=20 char]

➢ Address (line1,line2,city,district,state,country,pincode)- [<=200char]\*composite

1. Order key-Order ID

➢ Order ID

➢ Time stamp

➢ Expected date of delivery (derived attribute)

➢ Status [<=10 char]

➢ Date

6.Shipper key-Shipper ID

➢ Shipper ID

➢ Company name [<= 50 char]

➢ Address (line1,line2,city,district,state,country,pincode)- [<=200char]\*composite

Weak Entity:

1.Review key –(userID+product ID)

➢ Rating [1 to 5 ]

➢ Quality [1 to 5]

➢ Fitness [1 to 5]

➢ Length [1 to 5]

➢ Transparency [1 to 5]

➢ Text [1 to 5]

2. Cards key – (user ID+Card No)?

➢ Card type [ Credit/debit ]

➢ Card No [ 16 digits]

➢ Expiry Month [ 2 digits ]

➢ Expiry year [4 digits ]

3. Order details key-(customer ID + order ID + product code)

➢ Quantity [<=100 ]

➢ size [ <5 char]

➢ price to be paid

**Relationships**

* + - 1. **Relation between supplier and product (relation name:supply)**

A.Binary Relationship

b.Related in 1:n ratio

c.Participation of supplier is total and participation of product is total

d.Structural constraint (1,n) for supplier and (1,1) for product

* + - 1. **Relation between User and Cards (relation name: used)**

A.Binary Relationship

b.Related in 1:n ratio

c.Participation of the user is partial and participation of the card is total.

d.Structural constraint (0,n) for User and (1,1) for card

1. **Relation between order and payment (relation name:has)**

a.Binary relationship

b.Related in 1:1 ratio

c.Participation of order is total and participation of payment is total

d.Structural constraint(1,1) for order and (1,1) for payment

1. .**Relation between order and shipper(relation name ships)**

a.Binary relationship

b.Related in 1:n ratio

c.Participation of order is total and participation of payment is total

d.Structural constraint(1,1) for order and (1,n) for shipper

1. **.Relation between user,product and review**

a.Ternary relationship

b.An user can give only one review to one product and an user can give

review to multiple products and a product can be reviewed by multiple

users

**n>3 Relationships:**

**Relation between user,product,order,order details**

a.fourth degree relationship

b.An order ordered by a user contains some products each having order details

**Functional Requirements**

**Insert**:

➢Insert users info when a new account is created and insert card details if they want.

➢Insert product info when new stock arrives.

➢Insert order details when the user places an order.

➢Insert reviews to products given by users.

➢Insert info of new suppliers,shippers when they make deal with them

**Delete**:

➢Delete an order info when a person cancels his order

➢Delete a user info when he removes his account.

➢Delete Shippers info when they cancel their deal with them.

**Update**:

➢Update users info when they change their details such as password,phone no etc.

➢Update stock available each time a product is sold out.

➢Update status of payment when payment is done and status of order.

**Retrievals**

1.selection:

➢Retrieve complete data tuples of products belonging to a particular brand or particular

category.

➢Retrieve complete data tuples of male users.

➢Retrieve complete data of orders shipped by a company.

2.Projection:

➢List out the users whose age>25.

➢Products whose price is >=2000.

3.Aggregate:

➢Total amount to be paid by user for an order.

➢Average rating of a product totally or in terms of quality,transparency etc.

4.Search:

➢List out the products whose name partially matches with text.

5.Analysis:

➢Brand which has the highest average rating of its product.

➢Total no of 5 star ratings given to a product.

**REFERENCES**

**Project Code**:

<https://github.com/errabelliprudvi/Shopping-application-.git>