**A Modern Approach to Ration Distribution**

**A project report submitted by**

**Name:-**

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

**BACHELOR OF COMPUTER APPLICATIONS**

**Under the guidance of**

**Faculty Name**

Designation, Department of Computer Science,

College Name

**Dec 2023**

**LIST OF FIGURES**

* Fig 1. Data Flow Diagram (Level-0) 6
* Fig 2. Data Flow Diagram(Level-1) 7
* Fig 3. Data Flow Diagram (level-2) 7
* Fig 4. Use Case Diagram 8
* Fig 5. System Architecture 8

**LIST OF TABLES**

* Test Case 1 14
* Test Case 2 15
* Online Testing 15

**TABLE OF CONTENTS**

**1. INTRODUCTION**

1.1 Motivation 1

1.2 Problem definition 1

1.3 Objective of Project 1

1.4 Limitations of Project 1

**2. ANALYSIS**

2.1 Introduction 3

2.2 Software Requirement Specification 3

2.2.1 User requirement 3

2.2.2 Software requirement 4

2.2.3 Hardware requirement 4

2.3 Content diagram of Project 5

2.4 Conclusion 5

**3. DESIGN**

3.1 Introduction 6

3.2 Diagrams 6

3.3 Module design and organization 9

3.4 Conclusion 9

**4. IMPLEMENTATION & RESULTS**

4.1 Introduction 10

4.2 Explanation of Key functions 10

4.3 Result Analysis 11

4.4 Method of Implementation 13

4.5 Conclusion 13

**5. TESTING & VALIDATION**

5.1 Introduction 14

5.2 Design of test cases and scenarios 14

5.3 Validation 15

5.4 Conclusion 16

**6. CONCLUSION** 17

6.1Future Enhancements 17

**REFERENCES** 18

**APPENDICES**

1. Screenshots 19

**ABSTRACT**

“The Driving School Management System” is a web application software solution that streamlines administrative tasks in driving schools. We wanted to create a better system for ration distribution, which is user friendly and easy to use so that no matter how a little a person knows about technology they would still be to use the web application. We are using MERN Stack, React.js for the frontend, Express and Node.js for the backend of the system and using MongoDB as the database. The system optimizes operations by automating processes and providing real-time insights. This abstract outlines DSMS's key functionalities and benefits in revolutionizing driving school management. We were finally able to create an application which was efficient, responsive as well as user friendly and complied with all of our initial goals for the application.

**CHAPTER 1**

**INTRODUCTION**

**1.1 Motivation**

By creating a Driving School Management System, we aim to unlock the full potential of driving schools, freeing them from the burdens of administrative tasks and empowering them to focus entirely on their core mission: shaping confident, skilled drivers for safer roads and brighter futures.

**1.2 Problem definition**

Manual administrative tasks burden driving schools, leading to inefficiencies in enrollment, scheduling, billing, and compliance. Without real-time insights, decision-making is hindered. A Driving School Management System is urgently needed to automate operations, streamline processes, ensure compliance, and provide actionable data for informed decision-making, revolutionizing driving school management.

**1.3 Objective of Project**

The main objectives of this project are:

* Student Management
* Tutor Management
* Resource Allocation
* Scheduling
* Communication

**1.4 Limitations of Project**

* **Scope**

The project may not address every possible scenario or requirement within the driving school

Management domain due to scope constraints.

* **Resource Limitation**

Constraints on time, budget, and human resources may limit the depth and breadth of features and functionalities.

* **Technology Constraints**

The project's implementation may be limited by the available technology stack, leading to potential constraints on scalability or compatibility.

* **External Factors**

External factors such as regulatory changes, market dynamics, or unforeseen events may impact the project's timeline, scope, or success.

**CHAPTER 2**

**ANALYSIS**

**2.1 Introduction**

Analysis is the process of collecting and interpreting facts, understanding problems and using the information to suggest improvements on the system. This will help to understand the existing system and determine how computers make its operation more effective. The aim of this analysis is to collect the detailed information on the system and the feasibility study of the proposed system. This analysis focuses on the flow of the system module by module and the efficiency of each. To design the proposed system, we need the exact processing logic as well as the extended features of the existing system such as reliability, consistency, storage capacity etc. The analysis will concentrate on the information gathering for the efficient, user friendly and reliable system, which will carry forward the features of the existing system.

**2.2 Software requirement specification**

**2.2.1 User Requirement**

1. User Registration and Login:

* Users should be able to create accounts easily.
* Secure login functionality with username/email and password.

1. Profile Management:

* Users should be able to edit and update their profiles.
* Option to reset password in case of forgetfulness.

1. Product Catalog:

* A well-organized catalog displaying available ration items.
* Search and filter options for users to find specific products.

1. Shopping Cart:

* Users should be able to add items to a virtual shopping cart.
* View and modify the cart before checkout.

1. Order Placement:

* Easy and intuitive order placement process.
* Confirmation message after successful order placement.

1. Payment Integration:

* Secure payment gateway integration for online transactions.
* Support for multiple payment methods like credit/debit cards, UPI, etc.

1. Order History:

* Users should have access to their order history.
* Details of past orders, including date and items purchased.

1. Feedback and Ratings:

* Users should be able to leave feedback and ratings for products.
* Option to provide feedback on the overall shopping experience.

1. Customer Support:

* Access to customer support through chat, email, or a helpline.
* FAQs section to address common queries.

**2.2.2 Software Requirement**

* Operating System: Windows
* Front End: React.js
* Back End: Express and Node.js
* Programming Language: JavaScript, HTML, CSS
* Database: MongoDB

**2.2.3 Hardware Requirement**

* Processor: Intel i3 or Higher
* RAM: 1GB or Above
* Speed: Above 1.5GHz
* Hard Disk: 40GB or Above

**2.3 Conclusion**

In conclusion, the analysis aimed to understand the existing system, emphasizing module-by-module efficiency and user-friendliness. The software requirement specification outlines key features for an advanced online ration shopping system, including streamlined user registration, comprehensive profile management, and an organized product catalog. The system prioritizes a seamless shopping experience with features like a user-friendly shopping cart, intuitive order placement, secure payment integration, and accessible order history. User engagement is promoted through feedbacks. Collectively paving the way for the development of an innovative and user-centric online ration shopping platform.

**CHAPTER 3**

**DESIGN**

**3.1 Introduction**

The design of the Driving School Management System aims to address the complex needs of driving schools by providing a user-friendly, efficient, and comprehensive solution. By leveraging modern technologies and intuitive interfaces, the system seeks to streamline administrative tasks, enhance communication, and improve decision-making processes for administrators, tutors, and students alike. This introduction outlines the key principles guiding the design process, including scalability, usability, security, and interoperability, to ensure the system meets the diverse needs of driving schools while remaining adaptable to future challenges and opportunities.

**3.2 Diagrams**

1. DFD (0th Level)

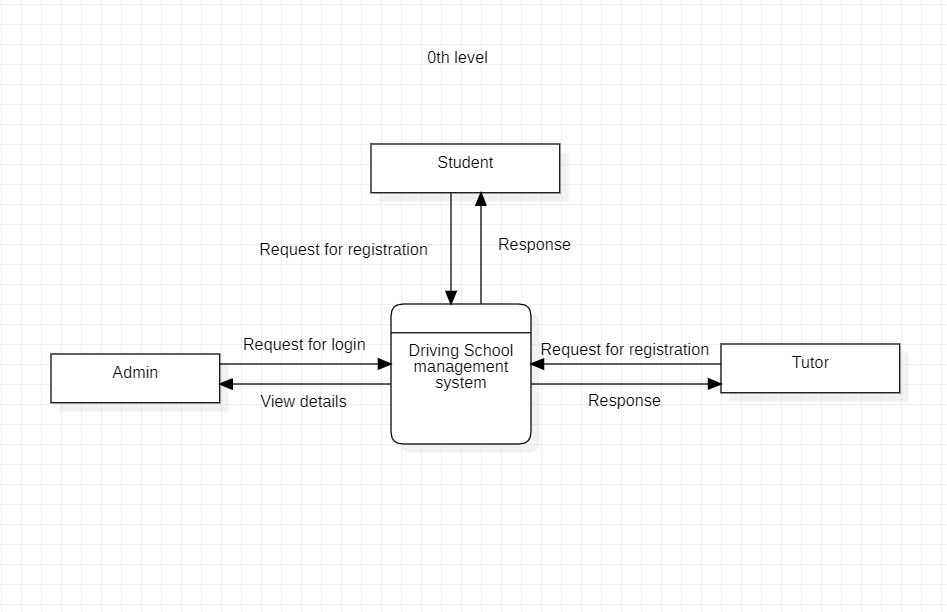


Fig 1: DFD 0th level

1. DFD (1st Level)

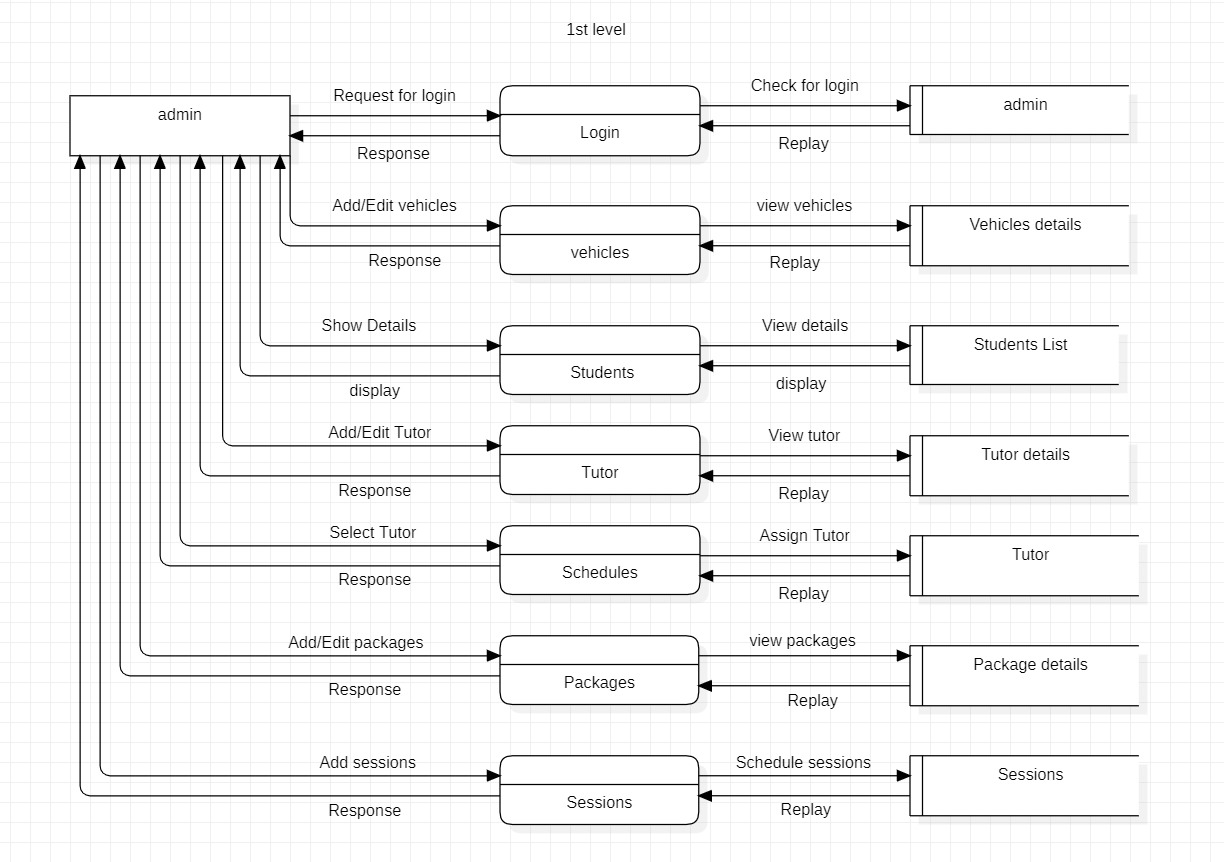


Fig 2: DFD 1st level

1. DFD (2nd Level)

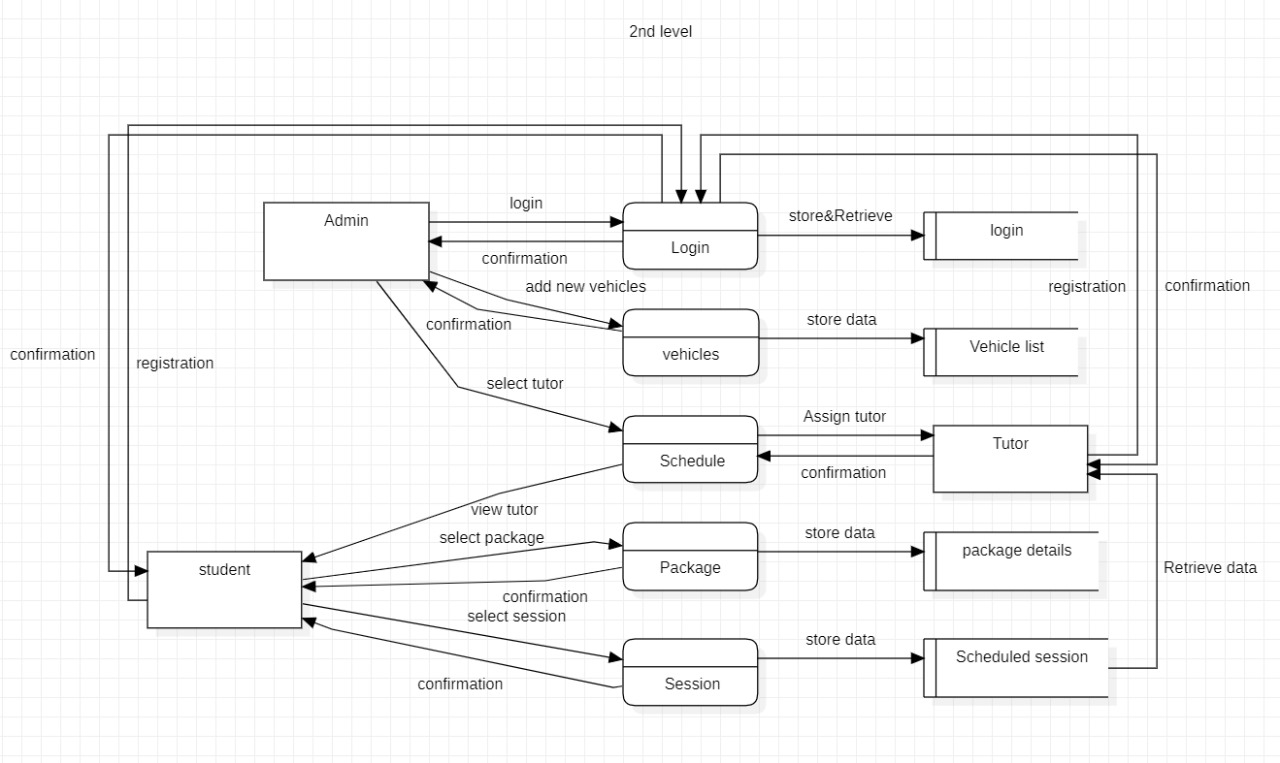


Fig 3: DFD 2nd level

1. Use Case Diagram for Remote Learning

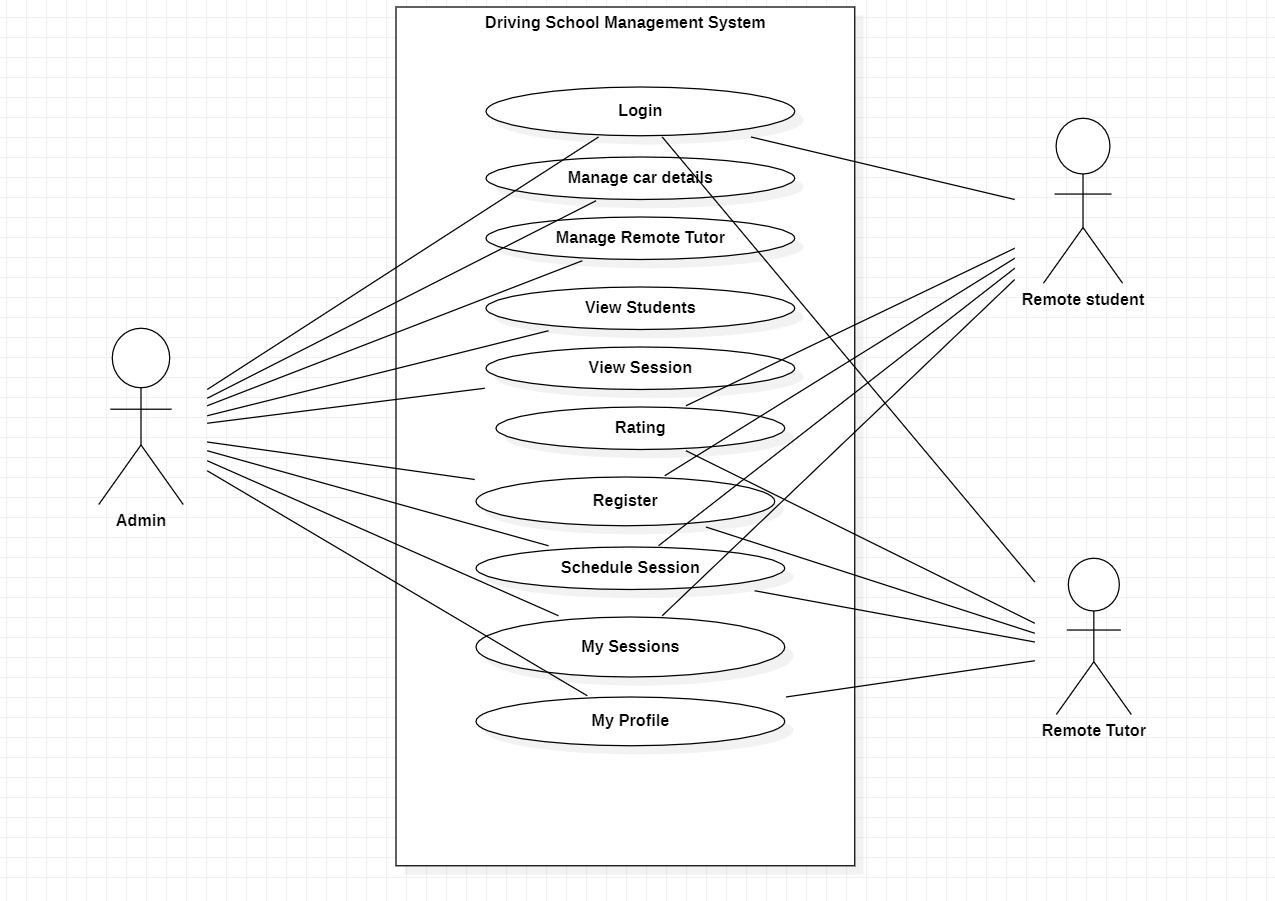


Fig 4: Use Case Diagram

1. Use Case Diagram for Local Learning

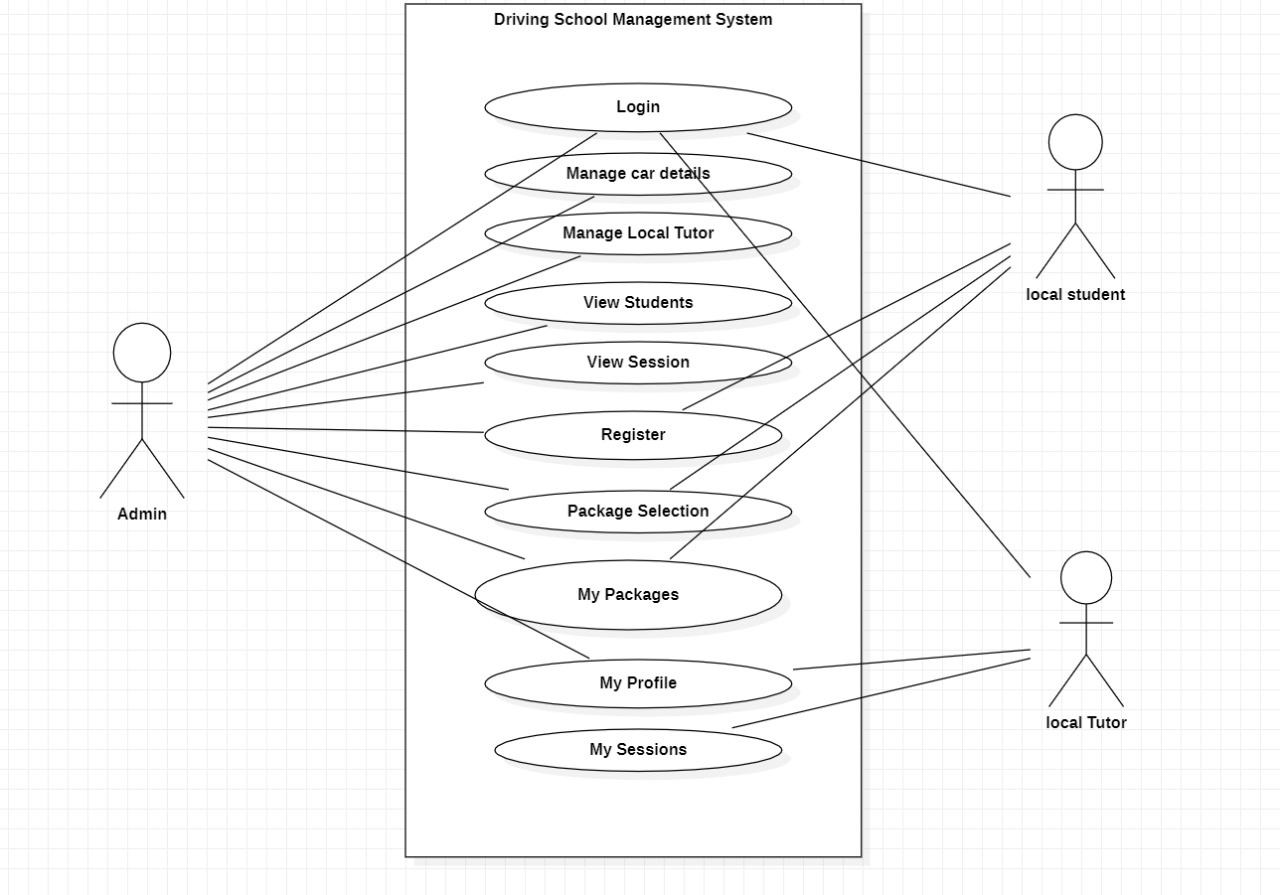


Fig 5: Use Case Diagram

1. System Architecture

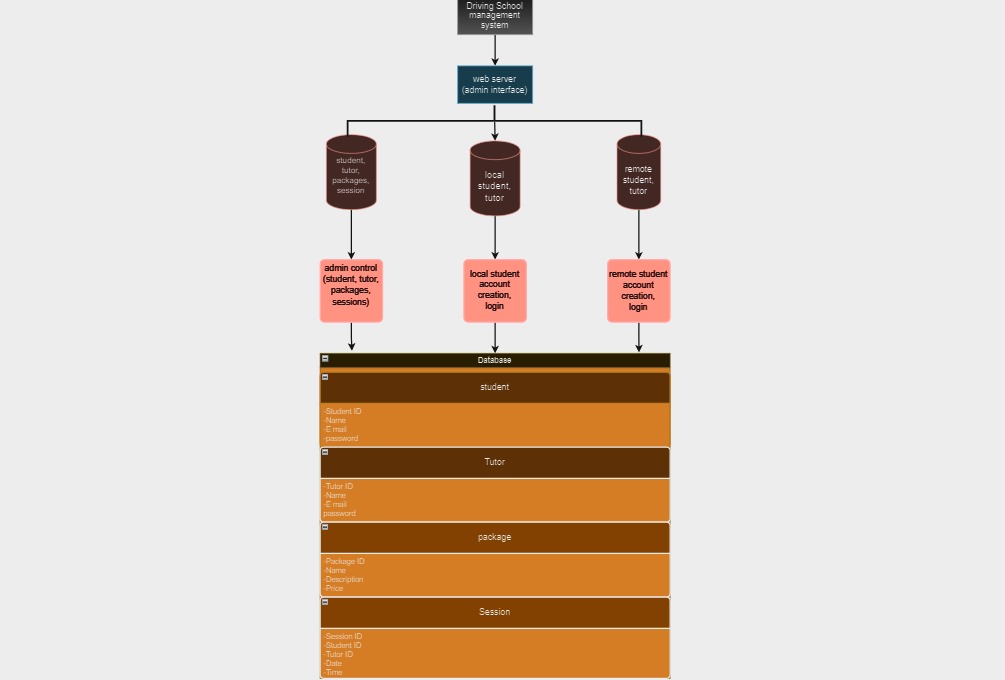


Fig 6: System Architecture

**3.3 Module design and organization**

* **Admin**

The admin can have unique registration id to maintain details in database. Admin logged in to their page, they will verify the stocks are available or anything to order from the government and maintain the product, employee and customer records.

* **Stock**

In this module, the product details are available to the customer to buy the product. Product details include its id, name, price and available number of products.

* **Customer**

Customer can view the available products and book the particular products from their home page. By choosing their desire product they can pay amount by cash or credit card.

**3.4 Conclusion**

In conclusion, the design phase is a vital component in the system lifecycle, involving careful analysis, integration, and boundary definition. Whether dealing with simple or complex systems, this approach is particularly pertinent in computer science, providing a means to comprehend organizational structures and address challenges. Design is not only a solution to current issues but also provides foresight into future projects. This dynamic and creative phase includes specification, development, and testing, focusing on both structural and physical design aspects. It acts as a crucial bridge, translating conceptualization into effective implementation, ensuring the resulting system is not only well-conceived but also optimally functional.

**CHAPTER 4**

**IMPLEMENTATION & RESULTS**

**4.1 Introduction**

Online ration shop marks a significant milestone in the integration of technology to streamline and enhance the distribution of essential goods. Developed using a robust technological stack including HTML, CSS, JavaScript, and Python Django, with a SQLite database, this web application represents a modern solution for managing and facilitating the distribution of rationed items. The amalgamation of these technologies ensures a user-friendly interface, efficient data processing, and a secure and scalable framework. Through careful planning and execution, the implementation phase will transition the theoretical design into a fully functional and accessible system, providing users with a seamless and responsive experience while optimizing the administration of the ration shop.

**4.2 Explanation of Key functions**

**Admin Side:**

**•** Add products

• Remove products

• Add information of ration shops

• Checking the price of stock

• View allotted stock

• View customers

• View payment reports

• View product shortage report

• View complaints

**Customer Side:**

• Add members

• Check available ration shop

• Able to view stock information

• Able to give complaints

• Able to book items online for later collection

**4.3 Result Analysis**

The implementation of the Online Ration Shop web application has yielded positive outcomes, aligning with the project's objectives and enhancing the efficiency of ration distribution. The following key results and analysis points highlight the success of the implementation process:

* **Functional Integration:**

The HTML, CSS, JavaScript, and Python Django components seamlessly integrated into a cohesive web application, providing a unified and intuitive interface for users.

* **Database Stability:**

The migration to the SQLite database was successful, ensuring data integrity and reliability in both development and production environments. The database proved to be stable, handling real-world data scenarios effectively.

* **Code Deployment Efficiency:**

The deployment process was executed efficiently, with proper organization and configuration of files on the web server. Static files, including CSS and JavaScript, were served optimally for enhanced performance.

* **Thorough Testing and Debugging:**

Rigorous testing and debugging procedures resulted in the identification and resolution of potential issues, contributing to a robust and error-free web application. Cross-browser compatibility and responsiveness were addressed, ensuring a consistent user experience across different devices.

* **User Training Success:**

User training sessions facilitated a smooth transition for end-users, administrators, and support staff. The availability of comprehensive documentation further empowered users to navigate and utilize the features of the web application effectively.

* **Parallel Run Validation:**

The parallel run phase successfully identified and addressed discrepancies, confirming the accuracy of data processing and validating the reliability of the new system. Users experienced minimal disruption during the transition.

* **Data Migration Accuracy:**

The final data migration from the existing system to the new web application was executed with precision, ensuring the integrity and consistency of data. The transition was seamless, minimizing data-related issues.

* **Monitoring and Evaluation:**

Ongoing monitoring of system performance revealed stability and responsiveness. User interactions in the live environment were analyzed, and feedback was collected to make real-time adjustments and enhancements.

* **Deployment Completion Satisfaction:**

The deployment completion phase marked a successful transition, with users directed to the new Online Ration Shop web application. Any post-deployment adjustments were promptly implemented based on user feedback, contributing to overall user satisfaction.

* **Overall System Reliability:**

The implemented web application stands as a reliable, secure, and user-friendly platform, streamlining the ration distribution process. The results indicate that the Online Ration Shop effectively meets the organizational requirements and enhances the accessibility of essential goods for users.

**4.4 Method of Implementation**

The front end is created using HTML and CSS. HTML is used for structuring the content of your web pages, while CSS is used for styling and layout. It's responsible for presenting the content and visual design to the user, including elements like text, images, forms, buttons, and the overall look and feel of your web pages. The front end interacts with the user's web browser. The backend is built using Python and Django. Python is a programming language used to write the server-side code, and Django is a web framework for Python that simplifies web development by providing tools for building web applications. The backend manages user authentication, database operations (in your case, SQLite3), and any other server-side tasks required to support the functionality of your web application. It responds to user requests from the front end, processes data, and generates dynamic content that is sent back to the front end for display. The database is SQLite and works by seamlessly integrating into the online ration shop application, managing data storage, retrieval, and modification efficiently. It requires no separate database server, uses SQL queries for data operations, supports transactions for data integrity, and ensures a secure and scalable data management solution.

**4.5 Conclusion**

The development of the online ration shop represents a significant advancement in the use of technology to enhance the distribution of essential goods. With a robust technological stack, including HTML, CSS, JavaScript, and Python Django, coupled with a SQLite database, the web application promises a modern and efficient solution for managing rationed item distribution. The amalgamation of these technologies ensures a user-friendly interface, streamlined data processing, and a secure, scalable framework. As the implementation phase approaches, the careful planning and execution are poised to transform the conceptual design into a fully functional system. This system is anticipated to provide users with a seamless and responsive experience, ultimately optimizing the administration of the ration shop. The online ration shop stands as a testament to the potential of technology in improving the accessibility and efficiency of essential goods distribution systems.

**CHAPTER 5**

**TESTING & VALIDATION**

**5.1 Introduction**

In the rapidly evolving landscape of digital services, the importance of robust testing and validation processes cannot be overstated, especially when it comes to essential services like online ration shops. As societies transition to digital platforms to streamline and enhance the distribution of essential goods, ensuring the reliability, security, and functionality of online ration shop web applications becomes paramount.

The primary objective of testing and validation in the context of an online ration shop web application is to identify and rectify any potential issues that may compromise the system's functionality, security, or usability. This process is essential to instill confidence in users, government authorities, and other stakeholders regarding the effectiveness and reliability of the online ration distribution system.

**5.2 Design of test cases and scenarios**

**Test cases:**

Table 1. Test cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Scenario** | **Expected Result** | **Observed Result** | **Result** |
| 1 | Enter wrong user name and pass word. | Display login form again with a warning message. | Message displayed. | Pass |
| 2 | Enter correct user name and wrong password. | Display login form again with a warning message. | Message displayed. | Pass |
| 3 | Enter correct user name and password. | Users can login into the system. | Appropriate home page is displayed. | Pass |
| 4 | Press login button without filling the user’s name and password. | Display a warning message to fill the fields. | Warning message is displayed. | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Scenario** | **Expected Result** | **Observed Result** | **Result** |
| 1 | Form displayed. | Display the registration form. | Form loaded | Pass |
| 2 | Enter the name in integers. | Display an invalid message. | Invalid message displayed | Pass |
| 3 | Enter the mobile number in characters. | Display an invalid message. | Invalid message displayed. | Pass |
| 4 | Enter the mobile number more than and less than 10 integers. | Display an invalid message. | Invalid message displayed. | Pass |
| 5 | Click the save button without filling the details | Display a warning message to fill the details. | Warning message displayed. | Pass |
| 6 | Click on save button with filled fields | Accept the details. | Registration successfully done. | Pass |
| 7 | Click cancel button | Clear all fields to blank | All fields cleared. | Pass |

Table 2. Test cases

* **Output Testing**

It is the section where we check whether the required output is being displayed or not. If the desired output is not displayed, then that means that there are still some errors to fix in the code.

Table 3. Output Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Input/procedure** | **Expected result** | **Actual result** | **Result** |
| 1 | Check whether intended output is obtained | All operations are carried out properly | Same as expected | Pass |

**5.3 Validation**

At the conclusion of integration testing, software is completely assembled as a package, interfacing errors have been uncovered and corrected and a final series of software tests begins validation test has been conducted one of the two possible conditions exists. One is the function or performance characteristics confirm to specification and are accepted and the other is deviation from specification is uncovered and a deficiency list is created.

**5.4 Conclusion**

In conclusion, the testing and validation process have instilled confidence in the reliability and effectiveness of our online ration shop web application. However, we recognize that continuous monitoring and improvement are essential to address evolving user needs and technological advancements. As we move forward, we commit to ongoing testing and maintenance to uphold the highest standards of quality, security, and user satisfaction.The testing and validation phase for an online ration shop web application is a critical step to ensure its reliability, security, and functionality. The entire process aims to identify and address potential issues, validate user requirements, and ensure a seamless experience for end-users.

**CHAPTER 6**

**CONCLUSION**

Online ration shop management system has been developed, tested, documented, and implemented successfully. The main objective of the system was brought in to effect. The system is developed in Python Django as front-end tool and SQLite as backend tool. This application is currently an open one, which promises any number of modules to be integrated along with it. Considering the current trends and the developments the future might offer, this is considered as an excellent system with a promising bright future in the coming Years.

The important benefits that have been found out through the implemented system are:

1. User friendly.

2. Simplified operation.

3. Reduced processing time.

4. Accurate result providing.

5. More reliability.

6. Increases accuracy.

Any system that has been used for several years gradually decays and becomes less effective because of the changes in environment to which it must adopt. For a time, it is possible to overcome problems by amending and minor modifications to acknowledgement the need of fundamental change.

**6.1 Future Enhancements**

There is no online system to buy ration product and stock maintenance. Government will provide product to buy that waiting for queuing, bargaining and products are robbed by making wrong entries without knowledge of the ration card holder. So, government meets money loss, time, and cost. The project “A modern approach to ration distribution” help the people who use the public distribution system. In this system the user can apply for the ration card via application and once the admin sees this, if the applicant is a valid one then he can issue a ration card. Also, this user can view the status of his or her purchase, the price of the commodities, and the quantity of each item for the user. There is also a dealer in a who directly deals with the user and he or she enter the details of the available stock in the shop and the purchase and sales details of the customer so that the user can view it online. This application uploads the data directly to the server by verifying the data, since it is direct process done by online ration shop keeper cannot do nothing in these transactions as they do in paper work.

**REFERENCES**

[**https://ahara.kar.nic.in/**](https://ahara.kar.nic.in/)

[**https://cleartax.in/s/ration-cards-types-benefits**](https://cleartax.in/s/ration-cards-types-benefits)

[**https://github.com/MaharaniThiraviyam/E-Ration/blob/master/consumerhome.php**](https://github.com/MaharaniThiraviyam/E-Ration/blob/master/consumerhome.php)

[**https://github.com/ROHITAUTI/E-RationShop**](https://github.com/ROHITAUTI/E-RationShop)

[**https://github.com/naveenrajceg13/Automated-Ration-Shop-System-**](https://github.com/naveenrajceg13/Automated-Ration-Shop-System-)

**APPENDICES**

1. **Screenshots**

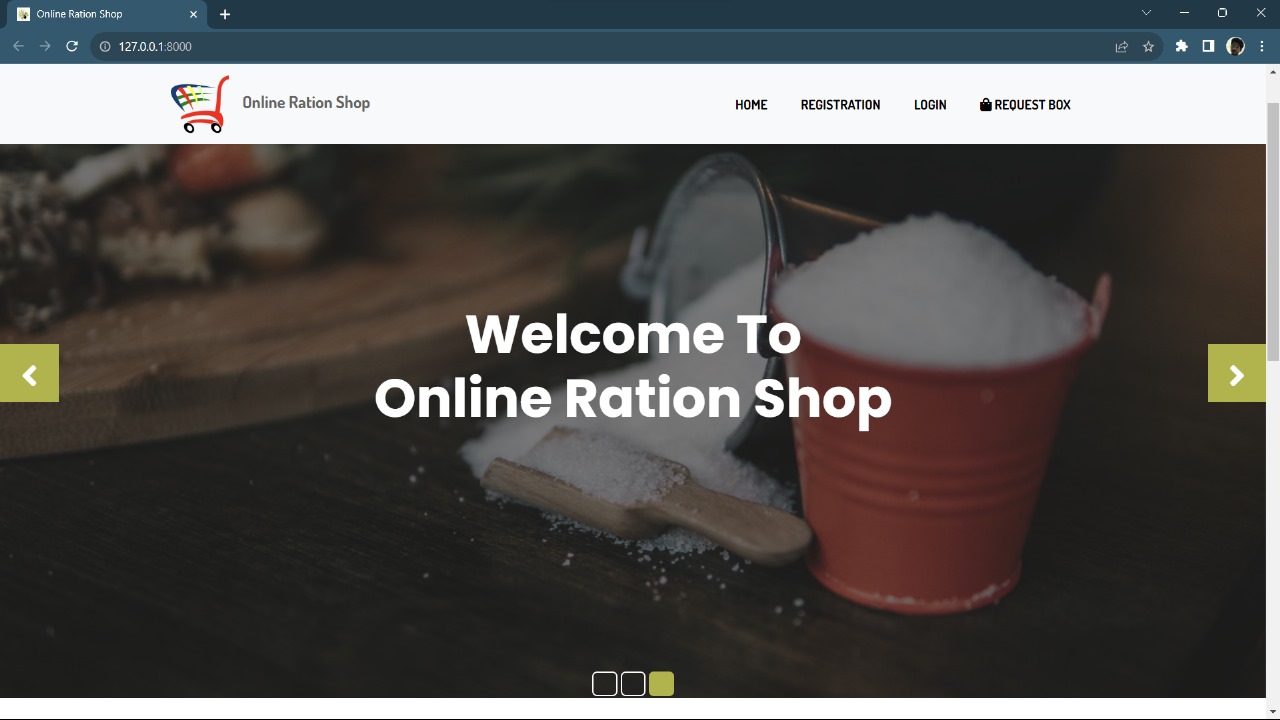
****

Fig 1: System Interface

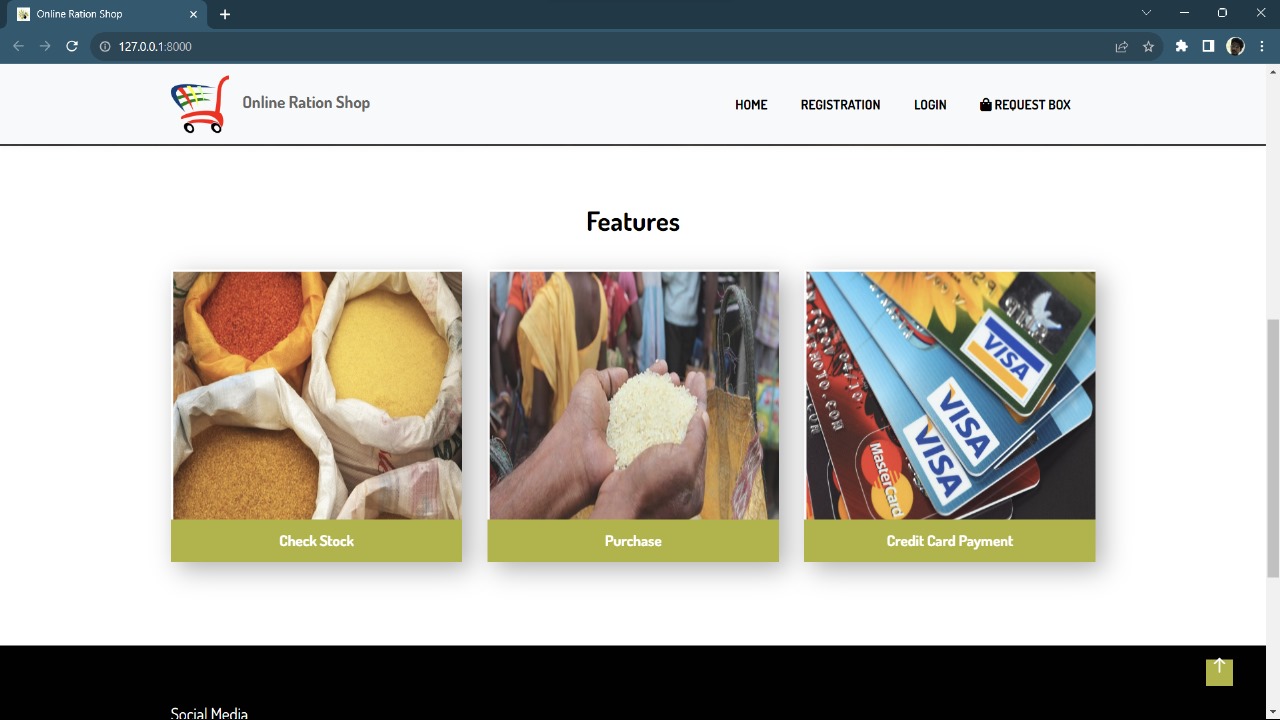
****

Fig 2: System Interface

****

Fig 3: Admin Interface

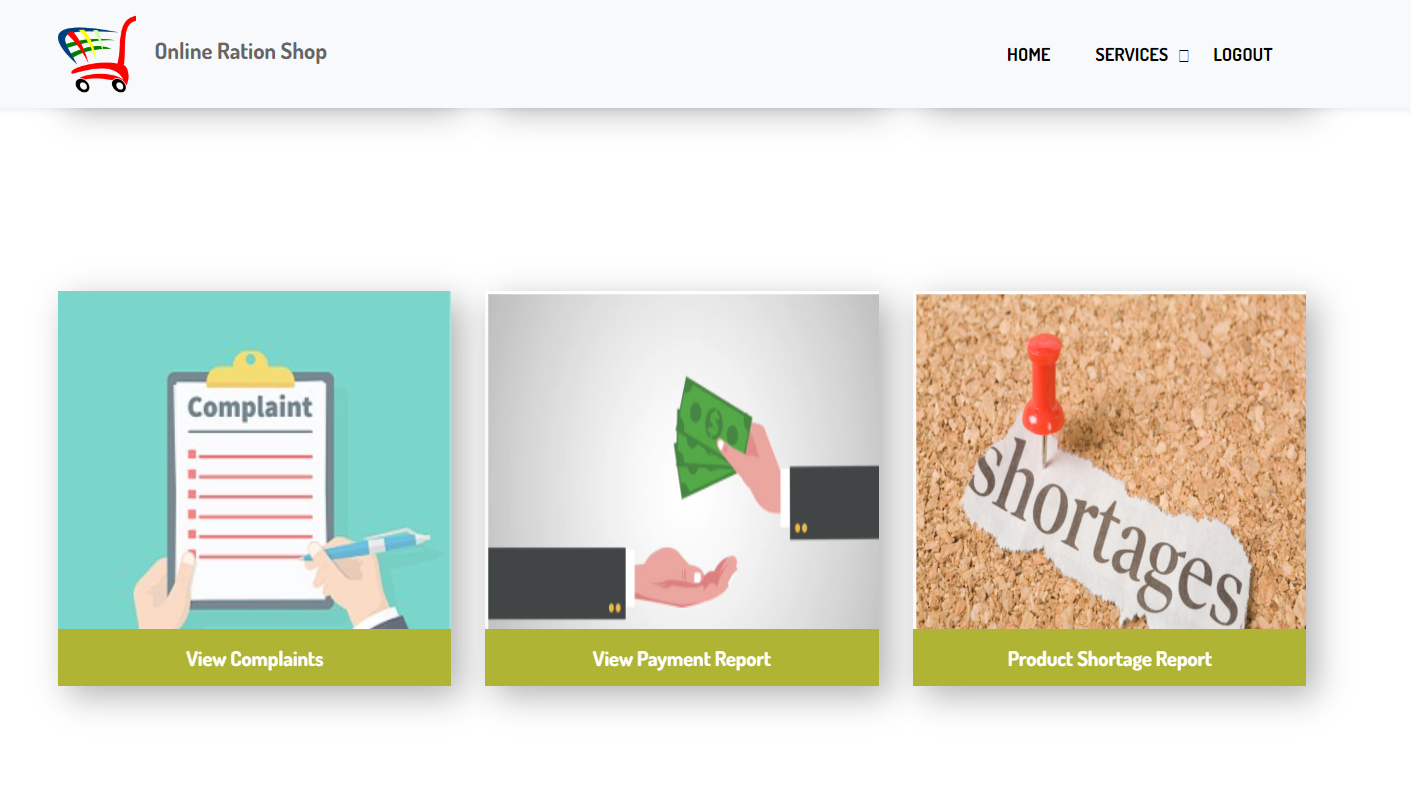
****

Fig 4: Admin Interface

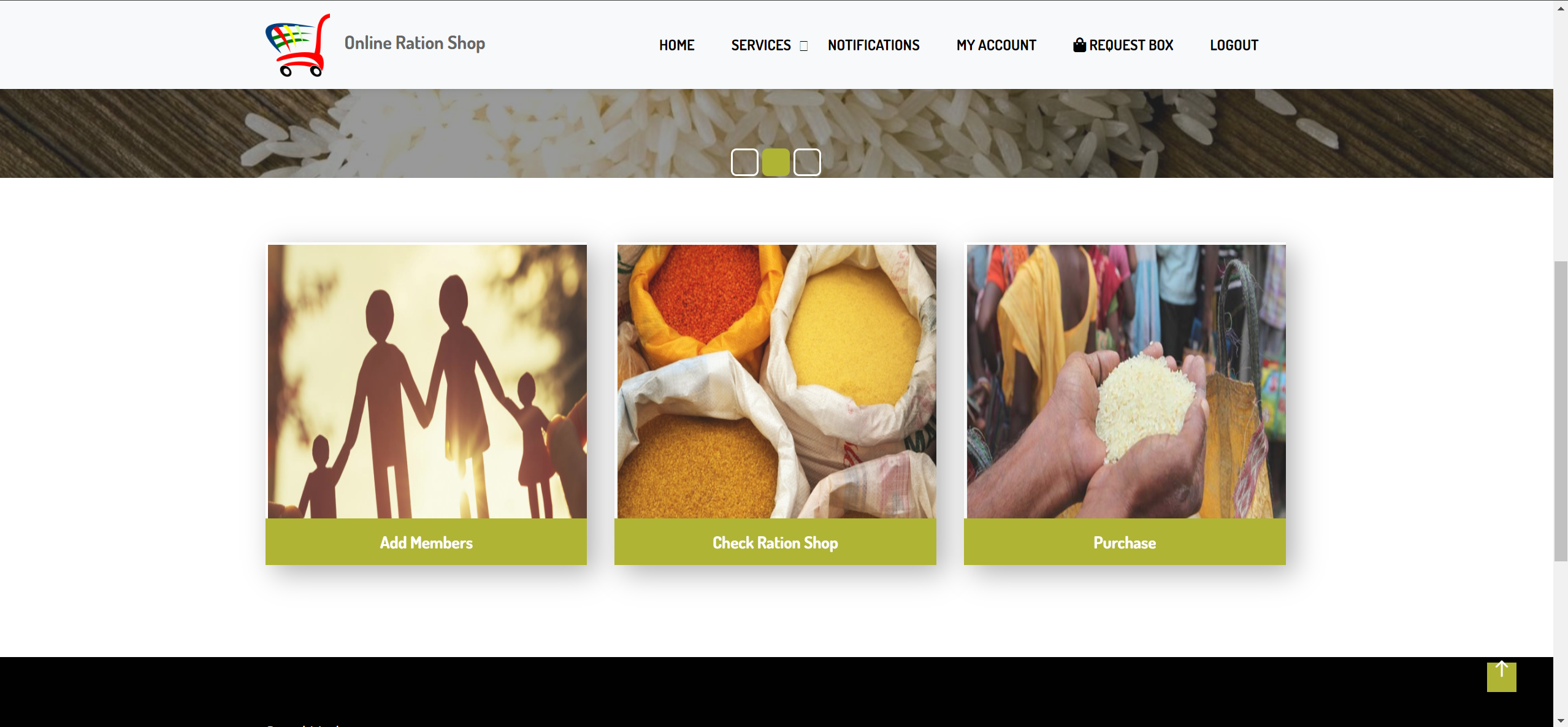


Fig 5: Customer Interface

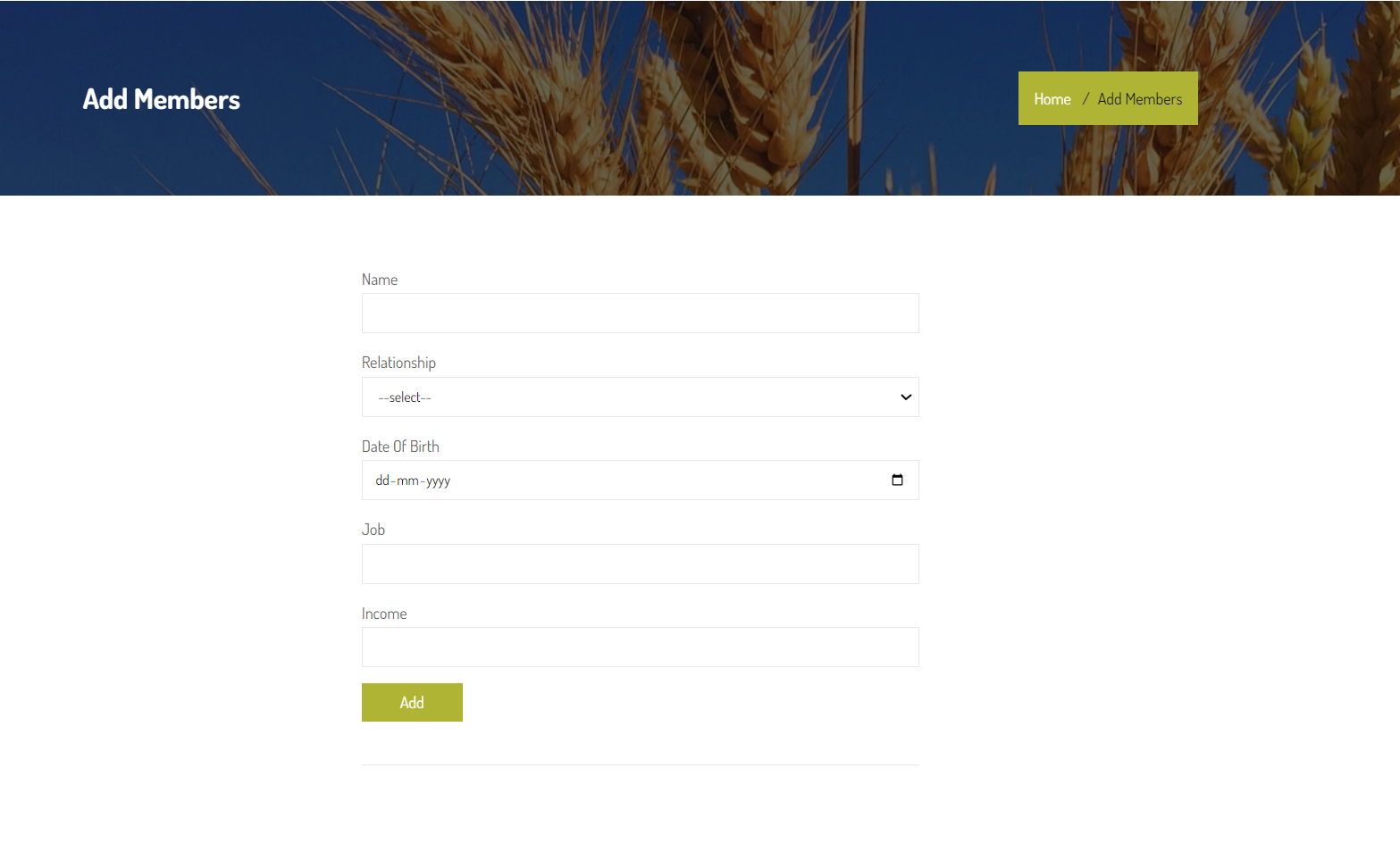


Fig 6: Add Members

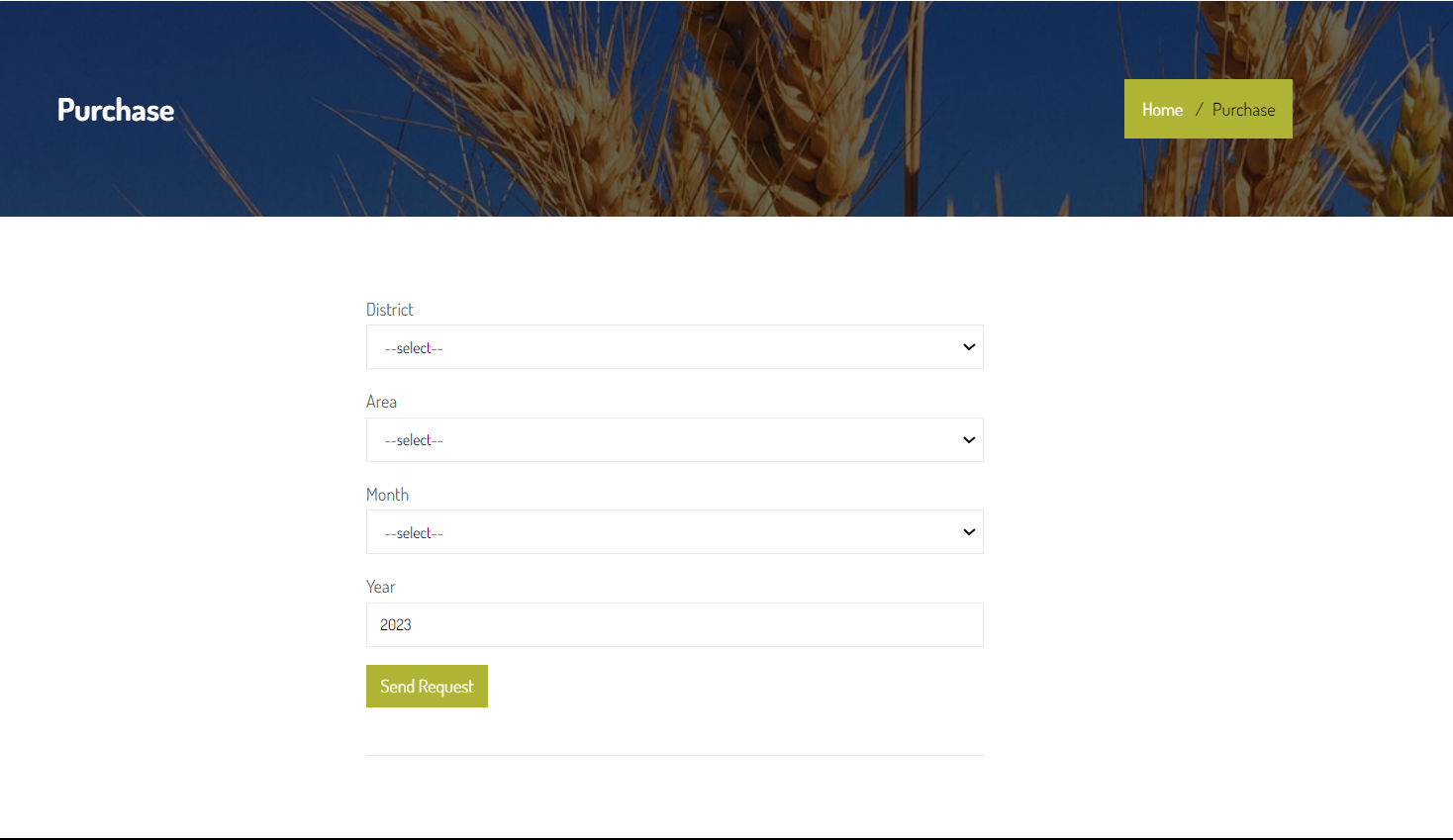


Fig 7: Product Request

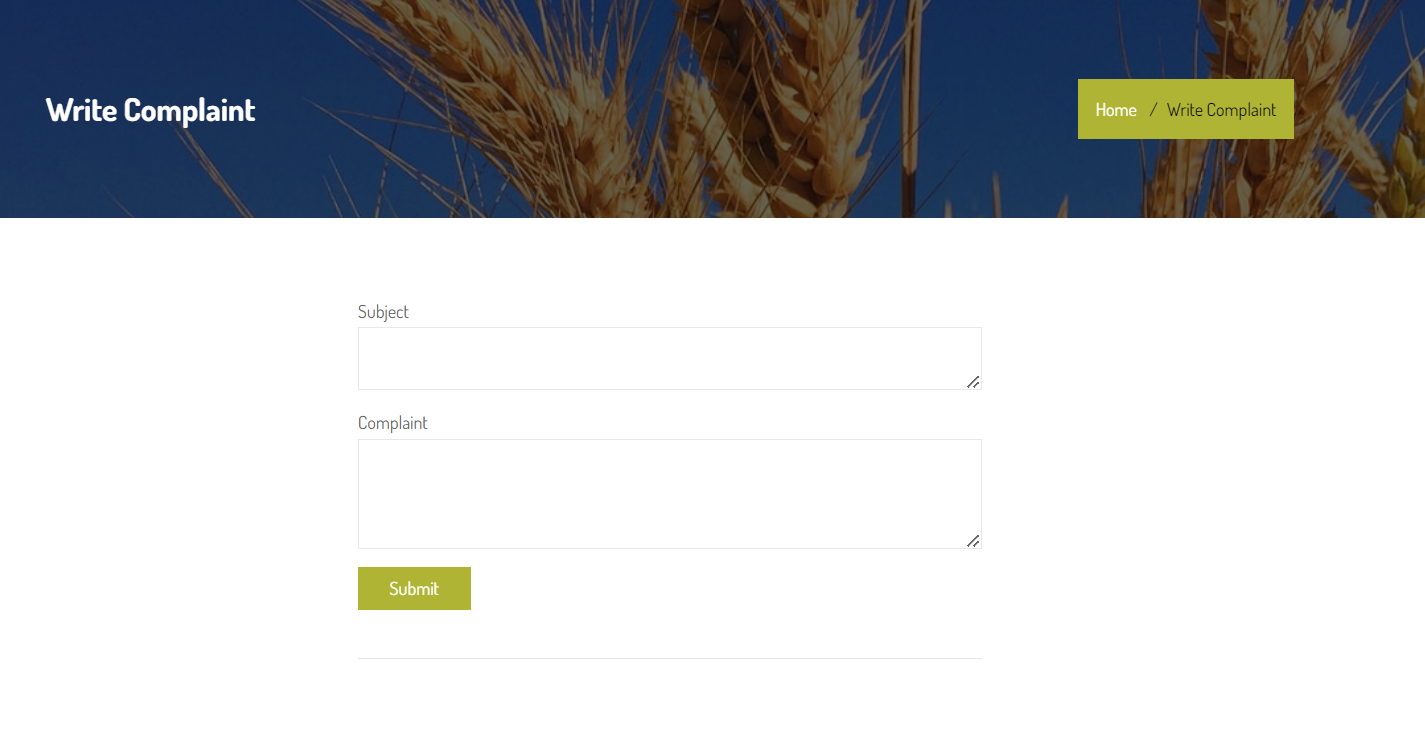


Fig 8: Complaint Forum