

MAMASPHERE

(Track, Nourish, Shop, Connect!)

Main Project Report

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

BACHELOR OF COMPUTER APPLICATIONS

BASELIOS POULOSE II CATHOLICOSE COLLEGE

BASELIOS MOUNT, PIRAVOM

Re-accredited with 'A' Grade by NAAC

(Affiliated to Mahatma Gandhi University)

2024-25



Submitted by:

ADHYA K S

(Reg No: 220021090185)

MAMASPHERE

(Track, Nourish, Shop, Connect!)

Main-Project Report

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

BACHELOR OF COMPUTER APPLICATIONS

**BASELIOS POULOSE II CATHOLICOSE COLLEGE
BASELIOS MOUNT, PIRAVOM**

**Re-accredited with 'A' Grade by NAAC
(Affiliated to Mahatma Gandhi University)**

2024-25



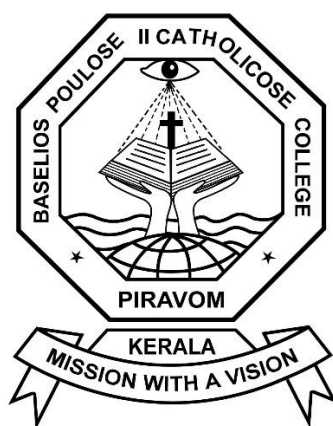
Submitted by:

Ms. ADHYA K S

(S6 BCA, Reg No: 220021090185)

Guided by:

Dr. SHEBA K U

BASELIOS POULOSE II CATHOLICOSE COLLEGE**Affiliated to Mahatma Gandhi University****(Re-accredited with 'A' Grade by NAAC)****PIRAVOM****2024-25****DEPARTMENT OF COMPUTER APPLICATIONS****Certificate**

This is to certify that the project entitled "MAMASPHERE" submitted in partial fulfilment for the award of the degree of BACHELOR OF COMPUTER APPLICATION is a bonafide report of the project done by Adhya K S (Reg no: 220021090185) during the year 2024-25.

Internal Guide:**Dr. Sheba K U****Head of the Department:****Dr. Anu Paul****Examiner: 1****College Seal****Department Seal**



Ph: +91 9072488881
Second Floor, RKV Arcade, Near Private Bus Stand, Muvattupuzha
progressiveofc@gmail.com • www.pcplglobal.com

Date: 22/03/2025

CERTIFICATE

This is to certify that **Ms. ADHYA K S** (Reg.No:220021090185) 6th semester BCA student of **Baselios Poulouse II Catholicose College, Piravom** has been completed her main project in partial fulfillment of the requirement for the award of BCA under M.G University as detailed below.

Project Name : MAMASPHERE
Technology : Flutter
Duration : 14th December 2024 to 20th March 2025

We found her as quick learner and good team player throughout the period.

Progressive Software Solutions & Training,


Project Manager



DECLARATION

*I hereby declare that this project work entitled “MAMASPHERE” is a record of original work done by me under the guidance of **Dr. Sheba K U**, Assistant Professor, Department of Computer Applications and the work has not formed the basis for the award of any degree or diploma or similar title to any candidate of any university subject.*

Internal Guide:

Dr. Sheba K U

Signature of Student

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

At the outset, I thank God Almighty for making endeavor a success.

I express my gratitude to **Dr. Baby Paul**, Principal, Baselios Poulse II Catholicos College, for providing me with adequate facilities, ways and means by which I was able to complete the project work. I express my sincere thanks to my internal guide **Dr. Sheba K U**, who guide me properly from the beginning to the end of my project and examining the draft of this project, suggestions and modifications. With immense pleasure I take this opportunity to record out sincere thanks to Head of the Department **Dr. Anu Paul**, Associate Professor, Department of Computer Applications for her motivation throughout this project.

Last but not the least, I also express my gratitude to all other members of the faculty and well-wishers who assisted me in various occasions during the project work.

- Adhya K S

ABSTRACT

ABSTRACT

Pregnancy is a crucial phase in a woman's life that requires proper health monitoring, guidance, and support. Our maternity app aims to provide a comprehensive solution for expectant mothers by integrating essential features such as weight and pregnancy tracking, personalized diet plans, guided exercises, community engagement through user posts, and a shopping section for maternity products.

The app enables users to track their weight gain and pregnancy progress with insightful analytics and reminders. It offers customized diet plans based on nutritional needs, ensuring a healthy pregnancy. Additionally, safe and effective exercises are provided to promote well-being. The user post feature fosters a supportive community where expectant mothers can share experiences, seek advice, and interact with experts. To further ease maternity care, the app includes a shopping section for essential maternity products, offering convenience at their fingertips.

By combining health monitoring, guidance, and social interaction, our app serves as a one-stop solution for expectant mothers, ensuring a healthier and more informed pregnancy journey.

TABLE OF CONTENTS

| | |
|---|-----------|
| 1.Introduction..... | 12 |
| 1.1. Background and motivation | 13 |
| 1.2. The proposed system | 14 |
| 1.3. Project Scope..... | 14 |
| 2. System Analysis..... | 15 |
| 2.1. Introduction | 16 |
| 2.2. Stake holders of this project..... | 17 |
| 2.2.1. Admin..... | 17 |
| 2.2.2. User..... | 17 |
| 2.2.3. Shop..... | 17 |
| 2.3. Software requirement specifications | 17 |
| 2.3.1. System Features..... | 17 |
| 2.3.1.1. Admin..... | 17 |
| 2.3.1.2. User..... | 18 |
| 2.3.1.3. Shop..... | 18 |
| 2.3.2. Non-functional Requirements..... | 19 |
| 2.4. Feasibility study | 20 |
| 2.5. Software Development lifecycle model | 21 |
| 2.6. Hardware and software requirements | 22 |
| 2.6.1. Software specifications..... | 22 |
| 2.6.1.1.Flutter..... | 23 |
| 2.6.1.2. Dart | 23 |
| 2.6.1.3. Supabase..... | 23 |
| 2.6.1.4.Windows 11 | 23 |
| 2.6.2. Hardware requirements..... | 24 |

| | |
|--|-----------|
| 3. System Design | 25 |
| 3.1. System Architecture..... | 26 |
| 3.2. Module Design..... | 26 |
| 3.3. Database Design..... | 27 |
| 3.3.1. Normalization..... | 28 |
| 3.3.2. Table Structure..... | 29 |
| 3.3.3. Data Flow Diagram..... | 36 |
| 3.3.3.1. Introduction to data flow diagrams..... | 36 |
| 3.3.3.2. Data flow diagram..... | 37 |
| 3.4. Interface Design..... | 45 |
| 3.4.1. User interface screen design | 45 |
| 3.4.2. Output design..... | 45 |
| 4.System Testing And Implementation..... | 47 |
| 4.1. System Testing..... | 48 |
| 4.1.1. Unit testing..... | 49 |
| 4.1.2. Integration testing..... | 49 |
| 4.1.3. Black box testing..... | 49 |
| 4.1.4. White box testing..... | 50 |
| 4.1.5. Validation Testing | 50 |
| 4.1.6. User acceptance testing..... | 50 |
| 4.2. Implementation..... | 50 |
| 4.2.1.Coding Standards..... | 51 |
| 5. Conclusion..... | 52 |
| 5.1.Future Enhancement..... | 53 |
| 6.Bibliography..... | 55 |
| 7. Appendix..... | 56 |
| 7.1. Screenshots..... | 57 |

INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

Pregnancy is one of the most significant phases in a woman's life, marked by profound physical, emotional, and lifestyle changes. It is crucial for expectant mothers to have access to proper health monitoring, nutritional guidance, and emotional support throughout their journey. However, many women face challenges in managing their pregnancy due to a lack of reliable information, personalized care, and a strong support system.

In today's digital era, mobile applications have transformed healthcare by providing accessible and real-time solutions. Recognizing the need for a comprehensive maternity support system, our project aims to bridge the gap by offering an all-in-one mobile application that caters to the diverse needs of pregnant women. This app integrates essential features such as pregnancy tracking, weight monitoring, personalized diet plans, exercises, community engagement, and a shopping section for maternity products.

The motivation behind this project stems from the realization that many expectant mothers experience difficulties in keeping track of their health and well-being. While healthcare professionals provide guidance, continuous monitoring and timely interventions can be challenging. Our app seeks to address these challenges by empowering women with the right tools and knowledge, ensuring a healthier and more informed pregnancy journey. Moreover, the inclusion of a community feature allows expectant mothers to share their experiences, seek advice, and interact with experts, fostering a sense of belonging and support. The shopping section further enhances convenience by providing easy access to essential maternity products.

By integrating technology into maternity care, our project aspires to create a user-friendly and reliable application that enhances the pregnancy experience, promotes well-being, and ensures that expectant mothers receive the necessary guidance and support at every stage of their journey.

1.2 PROPOSED SYSTEM

Our maternity app is designed to provide a seamless and user-friendly experience for expectant mothers by integrating essential health monitoring, guidance, and support features. The system will include pregnancy tracking with milestone updates, weight monitoring with insightful analytics, and personalized diet plans tailored to individual nutritional needs. Additionally, it will offer exercises designed for different pregnancy stages, ensuring safe and effective physical activity. A community engagement feature will enable users to share experiences, seek advice, and interact with healthcare professionals, fostering a supportive network. To further enhance convenience, the app will incorporate a shopping section for maternity essentials, allowing users to purchase necessary products directly from the platform. By combining technology with maternity care, our system aims to provide a comprehensive, accessible, and informative solution that supports expectant mothers throughout their pregnancy journey.

1.3 PROJECT SCOPE

Limitations of existing system

- Existing systems do not provide customized tracking for weight, diet, and pregnancy progress.
- Expectant mothers may struggle to get timely medical advice without frequent hospital visits.
- Many systems lack a platform for mothers to share experiences and seek advice.
- Users need multiple apps for tracking, diet, exercise, and shopping, causing inconvenience.
- Traditional maternity care lacks digital integration for easy access to essential services.

Advantages of proposed system

- The app provides tailored tracking for weight, pregnancy progress, and overall well-being.
- Expectant mothers can access medical advice, diet plans, and safe exercises anytime.
- Users can share experiences, seek advice, and interact with experts in a dedicated forum.
- The app integrates tracking, diet planning, exercise guidance, community support, and shopping in a single platform.
- Expectant mothers can easily monitor their health and purchase maternity essentials anytime, anywhere.

SYSTEM ANALYSIS

2.1 INTRODUCTION

Software Engineering is the analysis, design, construction, verification and management of technical or social entities. To engineer software accurately, a software engineering process must be defined. System analysis is a detailed study of the various operations performed by the system and their relationship within and module of the system. It is a structured method for solving the problems related to the development of a new system. The detailed investigation of the present system is the focal point of system analysis. This phase involves the study of parent system and identification of system objectives. Information has to be collected from all people who are affected by or who use the system. During analysis, data are collected on the variable files, decision point and transactions handled by the present system. The main aim of system is to provide the efficient and user friendly automation. So the system analysis process should be performed with extreme precision, so that an accurate picture of existing system, its disadvantages and the requirements of the new system can be obtained.

System analysis involves gathering the necessary information and using the structured tool for analysis. This includes the studying existing system and its drawback, designing a new system and conducting cost benefit analysis. System analysis is a problem-solving activity that requires intensive communication between the system users and system developers. The system is studied to the minute detail and analyzed. The system is viewed as a whole and the inputs to the system are identified. The outputs from the organization are traced through various phases of processing of inputs. There are a number of different approaches to system analysis. When a computer based information system is developed, systems analysis (according to the Waterfall model) would constitute the following steps:

- The development of a feasibility study, involving determining whether a project is economically, technologically and operationally feasible.
- Conducting fact-finding measures, designed to ascertain the requirements of the system's end-users. These typically span interviews, questionnaires, or visual observations of work on the existing system.
- Gauging how the end-users would operate the system (in terms of general experience in using computer hardware or software), what the system would be used for and so on.

Techniques such as interviews, questionnaires etc. can be used for the detailed study of these processes. The data collected by these sources must be scrutinized to arrive at a conclusion.

The conclusion is an understanding of how the system functions. This system is called the Existing System. The Existing system is then subjected to close observation and the problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as a proposal which is the Proposed System. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is then presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is a loop that ends as soon as the user is satisfied with the proposal.

2.2 STAKE HOLDERS OF THIS PROJECT

2.2.1. ADMIN

The Admin Panel of MamaSphere provides comprehensive control over the platform, ensuring smooth management of maternity-related services. The admin can manage districts and places, organizing location-based data efficiently. They can also handle categories and subcategories, streamlining content organization for diet plans, exercises, and maternity products. Additionally, the admin has the authority to add and update diet plans tailored to pregnancy needs and upload exercises for safe physical activity. The system allows the admin to view new shops list, as well as verify or reject shop applications, ensuring only trusted sellers offer maternity products. This centralized management ensures a well-structured and reliable platform for expectant mothers.

2.2.2. USER

The User Panel of MamaSphere offers a seamless and interactive experience for expectant mothers. Users can create and update their profiles, change passwords, and track their weight for a healthy pregnancy journey. They can view personalized diet plans and watch guided exercise videos to ensure safe fitness routines. The shopping section allows users to browse maternity products, add items to their cart, place bookings, and track orders for a hassle-free shopping experience. Additionally, the platform features a social media-style community where moms can create posts, interact with each other, share experiences, and ask queries, fostering a supportive and engaging environment.

2.2.3. SHOP

The Maternishop Panel in MamaSphere serves as a dedicated marketplace where multiple sellers can register and sell maternity-related products. Sellers can create and manage their profiles, list new products, and categorize items for better visibility. They can update stock, modify prices, and track product performance to enhance sales. The system allows sellers to process orders, manage bookings, and track shipments, ensuring smooth transactions. The admin oversees shop registrations, verifying sellers before approval. Additionally, sellers can view sales reports, customer reviews, and inquiries, helping them improve their services. Maternishop provides a reliable and convenient platform for both sellers and expectant mothers to access quality maternity products.

2.3.SOFTWARE REQUIREMENTS SPECIFICATIONS

2.3.1. SYSTEM FEATURES

2.3.1.1.ADMIN

- The admin has the supreme power of the system.
- It has dashboard to view platform stats and manage activities.
- Manage districts and places by adding and updating locations.

- Create and manage categories and subcategories.
- Add, update, and delete diet plans and exercise videos.
- The admin should verify and approve or reject seller that register shop.
- Monitor user posts and remove inappropriate content.
- Track and manage orders, bookings, and transactions.
- Generate reports and analytics on users, shops, and engagement.
- Admin is responsible for maintaining and updating the whole system.
- Ensure high performance and scalability for future growth.

2.3.1.2.USER

- The user can browse around create,update and manage user profiles.
- Mamasphere shall allow the user to login to the system.
- The user enters valid username and password then goes to the user homepage. Otherwise display the error message like invalid username and password.
- The user can track weight and monitor pregnancy progress.
- View personalized diet plans and exercise videos.
- Browse, add to cart, and purchase maternity products.
- The user can see updates and details on bookings ,track orders, payments, and deliveries.
- Post pictures, ask queries, and interact with other moms.
- Like, comment, and share posts in the community.
- Receive notifications for reminders, updates, and offers.
- Access expert advice and maternity-related resources.
- The user can add reviews and ratings about the shop.

2.3.1.3.SHOP

- Maternishop allows sellers to sign up and wait for the approval of admin.
- Once approved by admin shall allow the Seller to login to the system.
- The Seller enters valid username and password then goes to the seller home page otherwise display the error message like invalid username and password.
- Add, update, and categorize maternity products.
- Manage stock, prices, and product details.

- View and process customer orders and bookings.
- Track order status, payments, and deliveries.
- Receive notifications for new orders and updates.
- The Seller should be able to view the details of the user who made an order.
- Sellers can view report on their most sold products and the categories with most sales.
- Seller can view products that are sold in particular periods.
- Seller is able to edit/update their personal details whenever they need.

2.3.2. NON-FUNCTIONAL REQUIREMENTS

- Usability: The user interface of Mamasphere is simple, user-friendly, and easy-to-navigate interface.
-
- Performance: The app should run smoothly with minimal lag.
-
- Security: The system should implement robust security measures to protect user data, including secure authentication and data encryption.
-
- Reliability: Maintain consistent functionality without crashes or errors.
- Scalability: Support growing users, shops, and data efficiently.
-
- Compatibility: Work seamlessly across various devices and screen sizes.

2.4. FEASIBILITY STUDY

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards. Various other objectives of feasibility study are listed below.

- To analyse whether the software will meet organizational requirements.
- To determine whether the software can be implemented using the current technology and within the specified budget and schedule.
- To determine whether the software can be integrated with other existing software.

Various types of feasibility that we checked include technical feasibility, operational feasibility, and economic feasibility.

Technical Feasibility

Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements. Technical feasibility also performs the following tasks.

- Analyses the technical skills and capabilities of the software development team members.
- Determines whether the relevant technology is stable and established.
- Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

From our perspective, MamaSphere is both a website and a mobile app, developed using Flutter with Dart for the front end and Supabase as the backend database. Since Flutter is a widely adopted cross-platform framework and Supabase is a scalable and reliable backend solution, our project is technically feasible. It can be accessed through web browsers as well as mobile devices, ensuring a smooth and efficient user experience.

Operational Feasibility

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks.

- Determines whether the problems anticipated in user requirements are of high priority.
- Determines whether the solution suggested by the software development team is acceptable.
- Analyses whether users will adapt to a new software.
- Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.

We found that our project will be satisfied for the client since we were discussing every detail about the software with the client at every step. The most important part of operational feasibility study is the input from client. So the software is built completely according to the requirements of the client. We have used the current industry standards for the software. Hence we can say that this software is operationally feasible.

Economic Feasibility

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases (such as hardware purchase) and activities required to carry out software development. In addition, it is necessary to consider the benefits that can be achieved by developing the software. Software is said to be economically feasible if it focuses on the issues listed below.

- Cost incurred on software development to produce long-term gains for an organization.
- Cost required to conduct full software investigation (such as requirements elicitation and requirements analysis).
- Cost of hardware, software, development team, and training.

It is estimated that our project is economically feasible as development cost is very minimal since the tools and technologies used are available online. It's a group student project so there are no personnel costs. Development time is well planned and will not affect other operations and activities of the individuals. Once the system has been developed, the companies purchasing the system will be providing with a manual for training purposes. There is no need to purchase new hardware since the existing computers can still be used to implement the new system.

2.5.SOFTWARE DEVELOPMENT LIFECYCLE MODEL

One of the basic notions of the software development process is SDLC models which stand for Software Development Life Cycle models. SDLC – is a continuous process, which starts from the moment, when it's made a decision to launch the project, and it ends at the moment of its full remove from the exploitation. Software development lifecycle (SDLC) is a framework that defines the steps involved in the development of software. It covers the detailed plan for building, deploying and maintaining the software. SDLC defines the complete cycle of development i.e. all the tasks involved in gathering a requirement for the maintenance of a Product.

Some of the common SDLC models are Waterfall Model, V-Shaped Model, Prototype Model, Spiral Model, Iterative Incremental Model, Big Bang Model, Agile Model. We used Agile Model for our Project.

Agile Model

Agile Model is a combination of the Iterative and incremental model. This model focuses more on flexibility while developing a product rather than on the requirement. In the agile methodology after every development iteration, the client is able to see the result and understand if he is satisfied with it or he is not. Extreme programming is one of the practical use of the agile model.

The basis of this model consists of short meetings where we can review our project. In Agile, a product is broken into small incremental builds. It is not developed as a complete product in one go. At the end of each sprint, the project guide verifies the product and after his approval, it is finalized. Client feedback is taken for improvement and his suggestions and enhancement are worked on in the next sprint. Testing is done in each sprint to minimize the risk of any failures.

Advantages of Agile Model:

- It allows more flexibility to adapt to the changes.
- The new feature can be added easily.
- Customer satisfaction as the feedback and suggestions are taken at every stage.
- Risks are minimized thanks to the flexible change process.

Disadvantages:

- Lack of documentation.
- If a customer is not clear about how exactly they want the product to be, then the project would fail.
- With all the corrections and changes there is possibility that the project will exceed expected time.

2.6.HARDWARE AND SOFTWARE REQUIREMENTS

2.6.1.SOFTWARE SPECIFICATION

This project is developed using modern cross-platform technologies to ensure efficiency, scalability, and a seamless user experience across both web and mobile applications.

| | | |
|----------------------|---|------------|
| Front end | : | Flutter |
| Development tool | : | Flutter |
| Database | : | Supabase |
| Programming Language | : | Dart |
| Operating System | : | Windows 11 |

2.6.1.1.FLUTTER

Flutter, developed by Google, is an open-source UI toolkit designed for building high-performance, natively compiled applications for mobile, web, and desktop from a single codebase. It provides a reactive framework with customizable widgets, enabling developers to create visually appealing and responsive user interfaces.

Flutter's **hot reload** feature allows instant updates during development, improving productivity. It eliminates the need to maintain separate codebases for different platforms, ensuring faster development cycles and consistent UI/UX across devices.

2.6.1.2. DART

Dart is the programming language used in Flutter development. Created by Google, Dart is an object-oriented, class-based language designed for building fast and scalable applications.

Key features of Dart:

- JIT (Just-in-Time) and AOT (Ahead-of-Time) Compilation: Ensures fast development cycles and high-performance execution.
- Null Safety: Reduces runtime errors by preventing null reference issues.
- Expressive and Concise Syntax: Simplifies development while maintaining readability and maintainability.

2.6.1.3.SUPABASE

Supabase is an open-source backend-as-a-service (BaaS) that provides scalable database management and authentication solutions. Built on PostgreSQL, it offers:

- Real-time Database: Enables live synchronization of data across clients.
- Authentication: Supports multiple login methods, including email/password and OAuth.
- Storage Services: Secure cloud storage for files, images, and media.

2.6.1.4.Windows 11

Windows 11 is the development environment for this project, offering improved performance, security, and support for modern development tools. Its compatibility with Flutter and Supabase ensures a smooth development process.

2.6.2. Hardware requirements

The selection of hardware configuring is a very task related to the software development, particularly inefficient RAM may affect adversely on the speed and corresponding on the efficiency of the entire system. The processor should be powerful to handle all the operations. The hard disk should have the sufficient to solve the database and the application.

Hardware used for development:

| | |
|------------------|--------------------------------------|
| CPU | : Intel i5 |
| Processor Memory | : 16 GB |
| Cache | :12 MB |
| SSD | : 512 TB |
| Monitor | : 14.6" |
| Monitor Keyboard | : Standard108 keys Enhanced Keyboard |
| Mouse | : Optical Mouse |

Minimum Hardware Required For Implementation:

| | |
|-----------|------------------------|
| CPU | : Pentium IV Processor |
| Memory | : 256MB Above |
| Cache | : 512 KB Above |
| Hard Disk | : 20 GB Above |
| Monitor | : Any |
| Keyboard | : Any |
| Mouse | :Any |

SYSTEM DESIGN

3.1.SYSTEM ARCHITECTURE

A system architecture or system's architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

The system architecture can best be thought of as a set of representations of an existing (or to be created) system. It is used to convey the informational content of the elements comprising a system, the relationships among those elements, and the rules governing those relationships. The architectural components and set of relationships between these components that architecture describes may consist of hardware, software, documentation, facilities, manual procedures, or roles played by organizations or people.

System architecture is primarily concerned with the internal interfaces among the system's components or subsystems, and the interface between the system and its external environment, especially the user. The structural design reduces complexity, facilitates change and result in easier implementation by encouraging parallel development of different parts of the system. The procedural design transforms structural elements of program architecture into a procedural description of software components. The architectural design considers architecture as the most important functional requirement. The system is based on the three-tier architecture.

The first level is the user interface (presentation logic), which displays controls, receives and validates user input. The second level is the business layer (business logic) where the application specific logic takes place. The third level is the data layer where the application information is stored in files or database. It contains logic about to retrieve and update data. The important feature about the three-tier design is that information only travels from one level to an adjacent level.

3.2.MODULE DESIGN

Modular programming is a software design technique that emphasizes separating the functionality of a program into independent, interchangeable modules, such that each contains everything necessary to execute only one aspect of the desired functionality. Conceptually, modules represent a separation of concerns, and improve maintainability by enforcing logical boundaries between components. Different modules of this project include.

1. User Authentication

This module ensures secure authentication for all stakeholders. Admins, sellers, and users can log in using their respective username and password. Admins can manage the system only after logging in. Sellers of shop log in using credentials assigned by the admin and can access their

shop details. Users can register themselves and log in to track their diet plans, weight progress, shop for maternity products, and interact with the community.

2. Registration

The registration module manages all stakeholder sign-ups. Admins register basic details like name and email. Sellers provide business information, including name, contact, address, profile photo, and verification documents. Users register with their name, email, contact, and address to access shopping, diet plans, exercise videos, and community features.

3. Activities

This module handles all activities within MamaSphere. Admins verify or reject seller registrations, manage districts, places, categories, subcategories, diet plans, exercises, and posts. Sellers add and manage products, process orders, edit shop details, and submit complaints or feedback to the admin. Users can update profiles, browse and purchase maternity products, track weight, follow diet plans, interact in the community, and review products.

4. Reports

The reporting module provides valuable insights for effective management. Admins generate reports on popular products, active sellers, most-followed shops, and order trends. Sellers can view sales reports for their products, helping them track revenue and performance. The system uses charts, tables, and PDFs to present order analytics, user engagement, and sales insights, eliminating the need for manual record-keeping.

3.3.DATABASE DESIGN

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive and flexible for the users. The general theme behind a database is to integrate all information. Database design is recognized as a standard of management information system and is available virtually for every computer system. In database design several specific objectives are considered:

- Ease of learning and use
- Controlled redundancy
- Data independence
- More information at low cost
- Accuracy and integrity
- Recovery from failure
- Privacy and security
- Performance

A database is an integrated collection of data and provides centralized access to the data. Usually the centralized data managing the software is called RDBMS. The main significant difference between RDBMS and other DBMS is the separation of data as seen by the program and data has in direct access to stores device. This is the difference between logical and physical data.

3.3.1.Normalization

Designing a database is complete task and the normalization theory is a useful aid in the design process. The process of normalization is concerned with transformation of conceptual schema into computer representation form. There will be need for most databases to grow by adding new attributes and new relations. The data will be used in new ways. Tuples will be added and deleted. Information stored may undergo updating also. New association may also be added. In such situations the performance of a database is entirely depend upon its design. A bad database design may lead to certain undesirable things like:

- Repetition of information
- Inability to represent certain information
- Loss of information

To minimize these anomalies, Normalization may be used. If the database is in a normalized form, the data can be growing without, in most cases, forcing the rewriting application programs. This is important because of the excessive and growing cost of maintaining an organization's application programs and its data from the disrupting effects of database growth. As the quality of application programs increases, the cost of maintaining the without normalization will rise to prohibitive levels. A normalized database can also encompass many related activities of an organization thereby minimizing the need for rewriting the applications of programs. Thus, normalization helps one attain a good database design and there by ensures continued efficiency of database.

Normalization theory is built around the concept of normal forms. A relation is said to be in normal form if it satisfies a certain specified set of constraints. For example, a relation is said to be in first normal form (1NF) if it satisfies the constraint that it contains atomic values only. Thus every normalized relation is in 1NF. Numerous normal forms have been defined. Codd defined the first three normal forms. All normalized relations are in 1NF, some 1NF relations are also in 2NF and some 2NF relations are also in 3NF. 2NF relations are more desirable than 1NF and 3NF are more desirable than 2NF. That is, the database designer should prefer 3NF than 1NF or 2NF. Normalization procedure states that a relation that is in some given normal form can be converted into a set of relations in a more desirable form. We can define this procedure as the successive reduction of a given collection of relations to some more desirable form. This procedure is reversible. That is, it is always possible to take the output from the procedure and convert them back into input. In this process, no information is lost. So it is also called "no loss decomposition".

First Normal Form

A relation is in first normal form (1NF) if and all its attributes are based on single domain. The objective of normalizing a table is to remove its repeating groups and ensure that all entries of the resulting table have at most single value.

Second Normal Form

A table is said to be second Normal Form (2NF), when it is in 1NF and every attribute in record is functionally dependent upon the whole key, and not just a part of the key.

Third Normal Form

A table is in third Normal Form (3NF), when it is in 2NF and every non-key attribute is functionally dependent on just the primary key.

3.3.2. Table Structure

Table is a collection of complete details about a particular subject. These data are saved in rows and Columns. The data of each Row are different units. Hence, rows are called RECORDS and Columns of each row are called FIELDS.

Data is stored in tables, which is available in the backend the items and data, which are entered in the input, form id directly stored in this table using linking of database. We can link more than one table to input forms. We can collect the details from the different tables to display on the output. There are mainly 19 tables in the project. They are:

1. tbl_admin
2. tbl_category
3. tbl_subcategory
4. tbl_district
5. tbl_place
6. tbl_user
7. tbl_weighttracker
8. tbl_exercise
9. tbl_dietplan
10. tbl_post
11. tbl_like
12. tbl_comment
13. tbl_shop
14. tbl_product
15. tbl_stock
16. tbl_cart
17. tbl_booking
18. tbl_complaint
19. tbl_review

TABLE DESIGN

1.tbl_admin

Description : to store the admin's login details.

Primary key : admin_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|----------------|-----------|-------------|--------------------|
| 1. | admin_id | UUID | PRIMARY KEY | Unique ID of Admin |
| 2. | admin_name | TEXT | NOT NULL | Name of admin |
| 3. | admin_email | TEXT | NOT NULL | Email of admin |
| 4. | admin_password | TEXT | NOT NULL | Password of admin |

2.tbl_category

Description : to store the category details.

Primary key : category_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|---------------|-----------|-------------|-----------------------|
| 1. | category_id | INT(8) | PRIMARY KEY | Unique ID of Category |
| 2. | category_name | TEXT | NOT NULL | Name of Category |

3.tbl_subcategory

Description : to store the subcategory details.

Primary key : subcategory_id

Foreign key:category_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|------------------|-----------|-------------|--------------------------|
| 1. | subcategory_id | INT(8) | PRIMARY KEY | Unique ID of subcategory |
| 2. | subcategory_name | TEXT | NOT NULL | Name of subcategory |
| 3. | category_id | INT(8) | FOREIGN KEY | ID of category |

4.tbl_district

Description : to store the district details.

Primary key : district_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|---------------|-----------|-------------|-----------------------|
| 1. | district_id | INT(8) | PRIMARY KEY | Unique ID of district |
| 2. | district_name | TEXT | NOT NULL | Name of district |

5.tbl_place

Description : to store the place details.

Primary key : place_id

Foreign key:district_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|-------------|-----------|-------------|--------------------|
| 1. | place_id | INT(8) | PRIMARY KEY | Unique ID of place |
| 2. | place_name | TEXT | NOT NULL | Name of place |
| 3. | district_id | INT(8) | FOREIGN KEY | ID of district |

6.tbl_user

Description : to store the user details.

Primary key : user_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|--------------------|-----------|-------------|-----------------------|
| 1. | user_id | UUID | PRIMARY KEY | Unique ID of user |
| 2. | user_name | TEXT | NOT NULL | Name of user |
| 3. | user_email | TEXT | NOT NULL | Email of user |
| 4. | user_password | TEXT | NOT NULL | Password of user |
| 5. | user_pregnancydate | DATE | NOT NULL | Pregnancydate of user |
| 6. | user_dob | DATE | NOT NULL | Date of birth of user |
| 7. | user_contact | TEXT | NOT NULL | Contact of user |

7.tbl_weighttracker

Description : to store the weight tracking details.

Primary key : weighttracker_id

Foreign key:user_weight,user_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|----------------------|------------|-------------|---|
| 1. | weighttracker_id | INT (8) | PRIMARY KEY | Unique ID of pregnancy tracking record. |
| 2. | weighttracker_date | TIMESTAMPZ | NOT NULL | Weighttracker Date |
| 3. | weighttracker_weight | FLOAT(8) | FOREIGN KEY | Weight of user |
| 4. | user_id | UUID | FOREIGN KEY | ID of user |

8.tbl_exercise

Description : to store the exercise details.

Primary key : exercise_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|----------------------|-----------|-------------|-----------------------------|
| 1. | exercise_id | INT(8) | PRIMARY KEY | Unique ID for each exercise |
| 2. | exercise_title | TEXT | NOT NULL | Title of exercise |
| 3. | exercise_description | TEXT | NOT NULL | Description of the exercise |
| 4. | exercise_file | TEXT | NOT NULL | Files of exercise |
| 5. | exercise_trimester | INT(8) | NOT NULL | Trimester for exercise |

9.tbl_dietplan

Description : to store the details of diet.

Primary key : dietplan_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|----------------------|-----------|-------------|------------------------------|
| 1. | dietplan_id | INT(8) | PRIMARY KEY | Unique ID for each dietpaln. |
| 2. | dietplan_title | TEXT | NOT NULL | Title of dietplan |
| 3. | dietplan_description | TEXT | NOT NULL | Description of diet |
| 4. | dietplan_breakfast | TEXT | NOT NULL | Details of breakfast |
| 5. | dietplan_lunch | TEXT | NOT NULL | Details of lunch |
| 6. | dietplan_dinner | TEXT | NOT NULL | Details of dinner |
| 7. | dietplan_month | TEXT | NOT NULL | Month |

10.tbl_post

Description : to store the details of post.

Primary key : post_id

Foreign key: user_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|---------------|------------|-------------|---------------------|
| 1. | post_id | INT(8) | PRIMARY KEY | Unique ID of post |
| 2. | post_content | TEXT | NOT NULL | Content of post |
| 3. | post_file | TEXT | NOT NULL | File of post |
| 4. | post_datetime | TIMESTAMPZ | NOT NULL | Date & time of post |
| 5. | user_id | UUID | FOREIGN KEY | ID of user |

11.tbl_like

Description : to store the details of like

Primary key : like_id

Foreign key: post_id ,user_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|---------|-----------|-------------|-----------------------|
| 1. | like_id | INT(8) | PRIMARY KEY | Unique ID of the like |
| 2. | post_id | INT (8) | FOREIGN KEY | ID of the post |
| 3. | user_id | UUID | FOREIGN KEY | ID of the user |

12.tbl_comment

Description : to store comments

Primary key : comment_id

Foreign key : user_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|-----------------|-----------|-------------|----------------------|
| 1. | comment_id | INT(8) | PRIMARY KEY | Unique ID of comment |
| 2. | comment_content | TEXT | NOT NULL | Content of comment |
| 3. | post_id | INT(8) | FOREIGN KEY | Date of comment |
| 4. | user_id | UUID | FOREIGN KEY | ID of the user |

13.tbl_shop

Description : to store the details of shop

Primary key : shop_id

Foreign key: place_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|---------------|-----------|-------------|---------------------|
| 1. | shop_id | UUID | PRIMARY KEY | Unique ID of shop |
| 2. | shop_name | TEXT | NOT NULL | Name of shop |
| 3. | shop_email | TEXT | NOT NULL | Email of shop |
| 4. | shop_password | TEXT | NOT NULL | Password of shop |
| 5. | shop_contact | TEXT | NOT NULL | Contact of shop |
| 6. | shop_address | TEXT | NOT NULL | Address of shop |
| 7. | shop_logo | TEXT | NOT NULL | Logo of shop |
| 8. | shop_proof | TEXT | NOT NULL | Proof of shop |
| 9. | place_id | INT (8) | FOREIGN KEY | ID of place |
| 10. | shop_vstatus | INT(8) | NOT NULL | Shop visited status |

14.tbl_product

Description : to store the details of product.

Primary key : product_id

Foreign key: subcategory_id,shop_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|---------------------|-----------|-------------|------------------------|
| 1. | product_id | INT(8) | PRIMARY KEY | Unique ID of product |
| 2. | product_name | TEXT | NOT NULL | Name of product |
| 3. | product_description | TEXT | NOT NULL | Description of product |
| 4. | product_price | INT(8) | NOT NULL | Price of product |
| 5. | product_image | TEXT | NOT NULL | Image of product |
| 6. | shop_id | UUID | FOREIGN KEY | ID of shop |
| 7. | subcategory_id | INT(8) | FOREIGN KEY | ID of subcategory |

15.tbl_stock

Description : to store the details of stock

Primary key : stock_id

Foreign key: product_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|----------------|-------------|-------------|--------------------|
| 1. | stock_id | INT(8) | PRIMARY KEY | Unique ID of stock |
| 2. | stock_quantity | INT(8) | NOT NULL | Quantity of stock |
| 3. | stock_date | TIMESTAMPTZ | NOT NULL | Date of stock |
| 4. | product_id | INT (8) | FOREIGN KEY | ID of product |

16.tbl_cart

Description : to store the details of cart

Primary key : cart_id

Foreign key: product_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|-------------|-----------|-------------|--------------------|
| 1. | cart_id | INT(8) | PRIMARY KEY | Unique ID of cart |
| 2. | cart_qty | INT (8) | NOT NULL | Quantity of cart |
| 3. | cart_status | INT (8) | NOT NULL | Status of cart |
| 4. | booking_id | INT (8) | FOREIGN KEY | Booking ID of cart |
| 5. | product_id | INT (8) | FOREIGN KEY | ID of product |

17.tbl_booking

Description : to store the details of booking

Primary key : booking_id

Foreign key: user_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|-----------------|-----------|-------------|----------------------|
| 1. | booking_id | INT(8) | PRIMARY KEY | Unique ID of booking |
| 2. | booking_amount | INT (8) | NOT NULL | Amount of booking |
| 3. | booking_status | INT (8) | NOT NULL | Status of booking |
| 4. | booking_address | TEXT | NOT NULL | Address of booking |
| 5. | user_id | UUID | FOREIGN KEY | ID of user |

18.tbl_complaint

Description : to store the details of complaint

Primary key : complaint_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|---------------------|------------|-------------|-------------------------------|
| 1. | complaint_id | INT(8) | PRIMARY KEY | Unique ID for each complaint. |
| 2. | complaint_title | TEXT | NOT NULL | Title of complaint |
| 3. | complaint_content | TEXT | NOT NULL | Content of complaint |
| 4. | complaint_reply | TEXT | NOT NULL | Reply of complaint |
| 5. | complaint_date | TIMESTAMPZ | NOT NULL | Date of complaint |
| 6. | complaint_status | TEXT | NOT NULL | Status of complaint |
| 7. | complaint_replydate | TIMESTAMPZ | NOT NULL | Reply date of complaint |
| 8. | user_id | UUID | FOREIGN KEY | ID of user |

19.tbl_review

Description : to store the details of review

Primary key : review_id

Foreign key: user_id,post_id

| SL NO | NAME | DATA TYPE | CONSTRAINTS | DESCRIPTION |
|-------|----------------|-----------|-------------|---------------------|
| 1. | review_id | INT(8) | PRIMARY KEY | Unique ID of review |
| 2. | review_rating | FLOAT(8) | NOT NULL | Rating of review |
| 3. | review_content | TEXT | NOT NULL | Content of review |
| 4. | user_id | UUID | FOREIGN KEY | ID of user |
| 5. | product_id | INT (8) | FOREIGN KEY | ID of product |

3.3.3.Data Flow Diagram

3.3.3.1. Introduction to Data Flow Diagrams

Data Flow Diagram is a network that describes the flow of data and processes that change, or transform, data throughout the system. This network is constructed by use a set of symbols that do not imply a physical implementation. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files. Data in files may also flow to processes as inputs.

There are various symbols used in a DFD. Bubbles represent the processes. Named arrows indicate the data flow. External entities are represented by rectangles. Entities supplying data are known as sources and those that consume data are called sinks. Data are stored in a data store by a process in the system. Each component in a DFD is labelled with a descriptive name. Process names are further identified with a number.

The Data Flow Diagram shows the logical flow of a system and defines the boundaries of the system. For a candidate system, it describes the input (source), outputs (destination), database (files) and procedures (data flow), all in a format that meet the user's requirements.


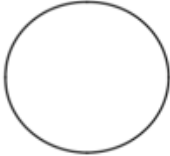


The main merit of DFD is that it can provide an overview of system requirements, what data a system would process, what transformations of data are done, what files are used, and where the results flow.

This network is constructed by use a set of symbols that do not imply a physical implementation. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files. External entities are represented by rectangles. Entities supplying data are known as sources and those that consume data are called sinks. Data are stored in a data store by a process in the system. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files. Data in files may also flow to processes as inputs.

Rules for constructing a Data Flow Diagram

1. Arrows should not cross each other
2. Squares, circles and files must bear names.
3. Decomposed data flow squares and circles can have same time
4. Choose meaningful names for data flow
5. Draw all data flows around the outside of the diagram

Basic Data Flow Diagram Symbols

| | |
|---|---|
|  | <p>A data flow is a route, which enables packets of data to travel from one point to another. Data may flow from a source to a process and from data store or process. An arrow line depicts the flow, with arrow head pointing in the direction of the flow.</p> |
|  | <p>Circles stands for process that converts data into information. A process represents transformation where incoming data flows are changed into outgoing data flows.</p> |
|  | <p>A data store is a repository of data that is to be stored for use by a one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names. If a process merely uses the content of store and does not alter it, the arrowhead goes only from the store to the process. If a process alters the details in the store then a double-headed arrow is used.</p> |
|  | <p>A source or sink is a person or part of an organization, which enters or receives information from the system, but is considered to be outside the contest of data flow model.</p> |

3.3.3.2.Data Flow Diagram

Each component in a DFD is labelled with a descriptive name. Process name are further identified with number. Context level DFD is draw first. Then the process is decomposed into several elementary levels and is represented in the order of importance. A DFD describes what data flow (logical) rather than how they are processed, so it does not depend on hardware, software, and data structure or file organization.

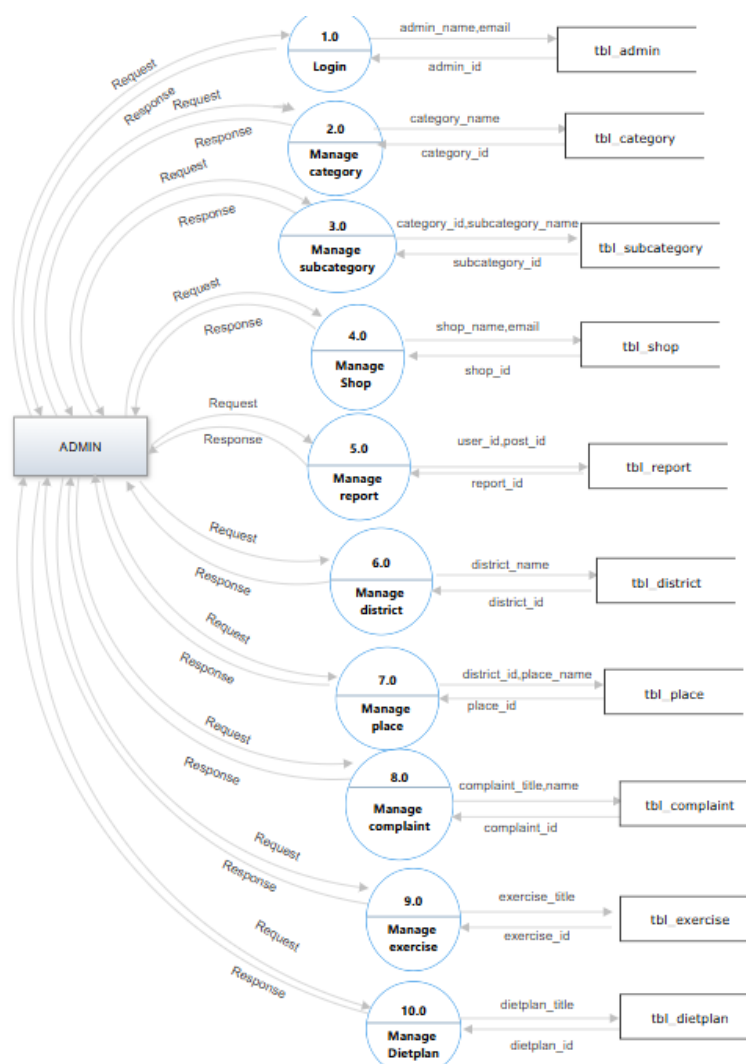
A DFD methodology is quite effective; especially when the required design.

LEVEL 0

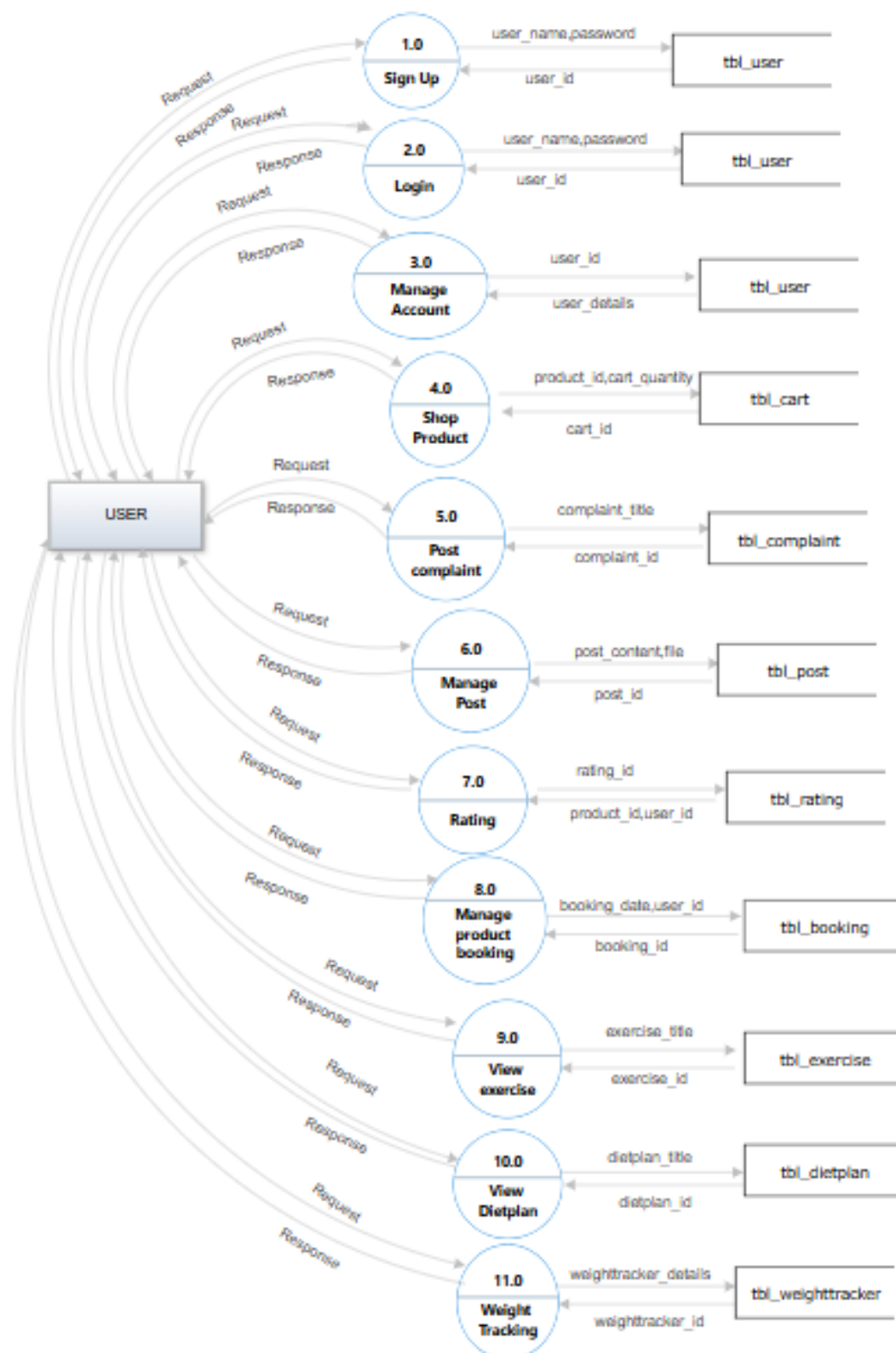


LEVEL 1

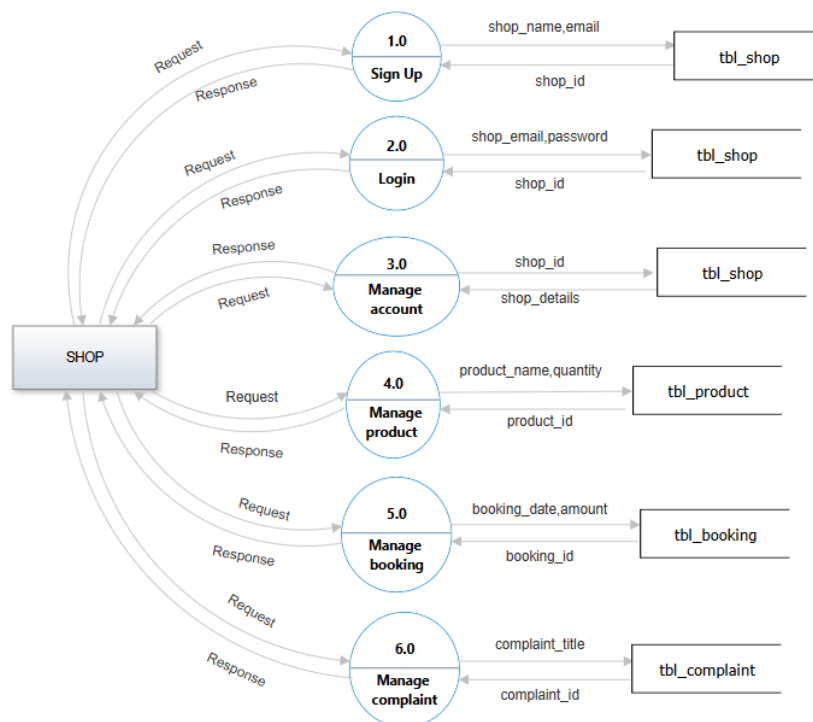
1)Admin



2) User

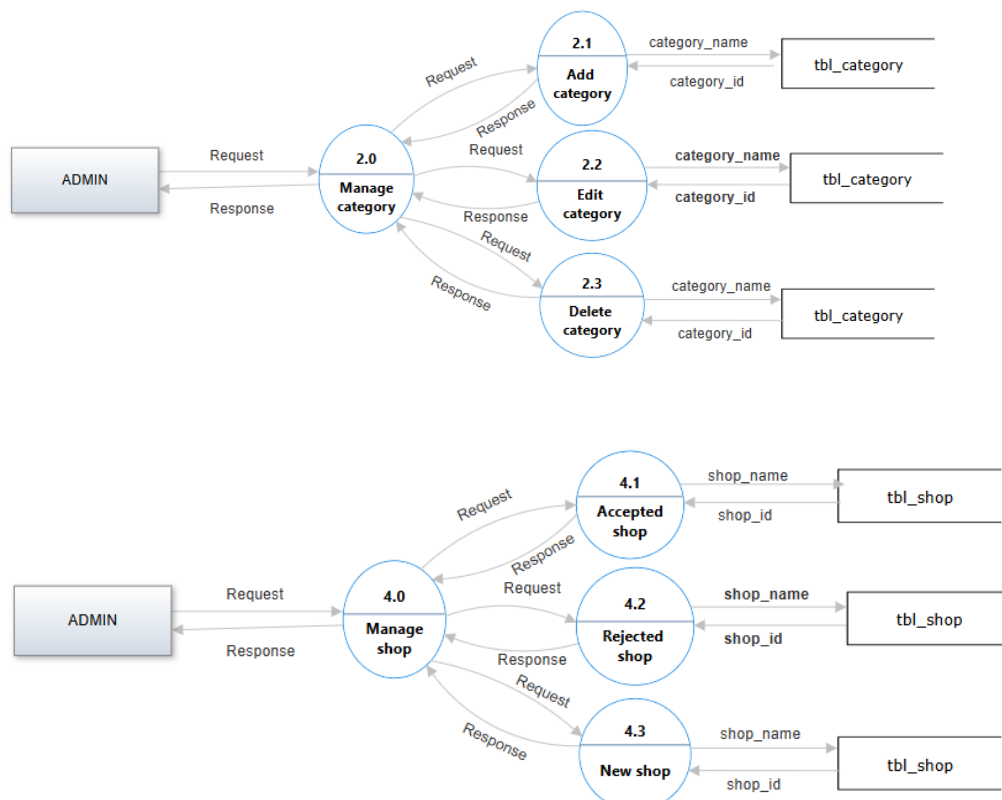


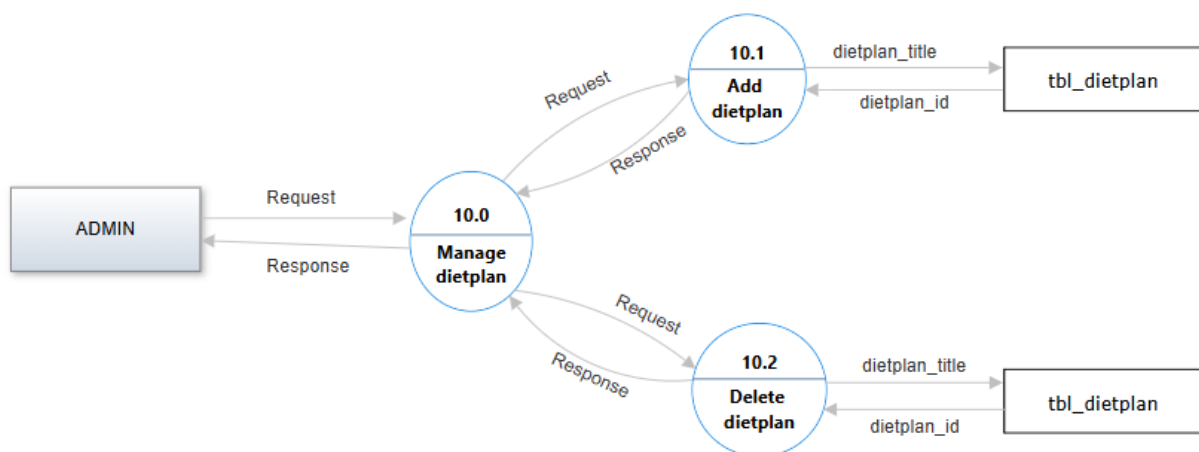
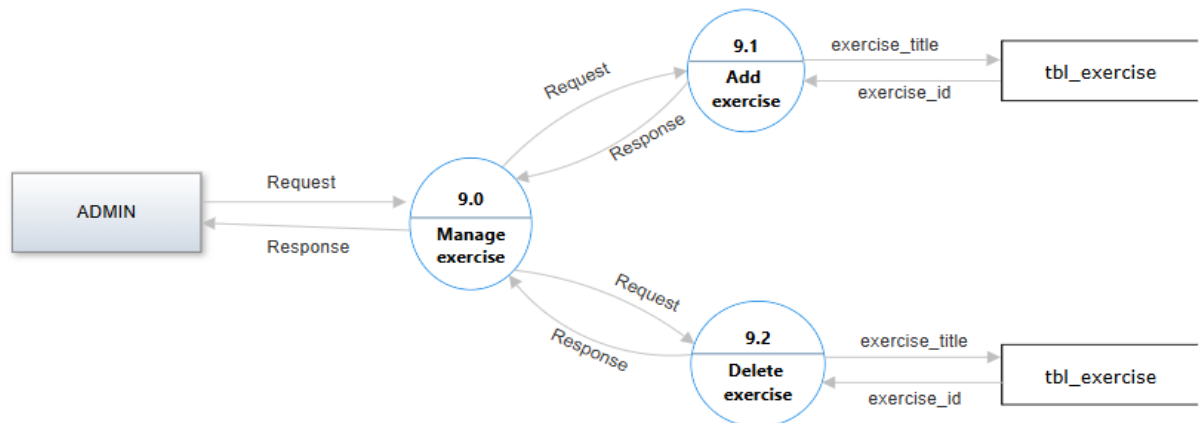
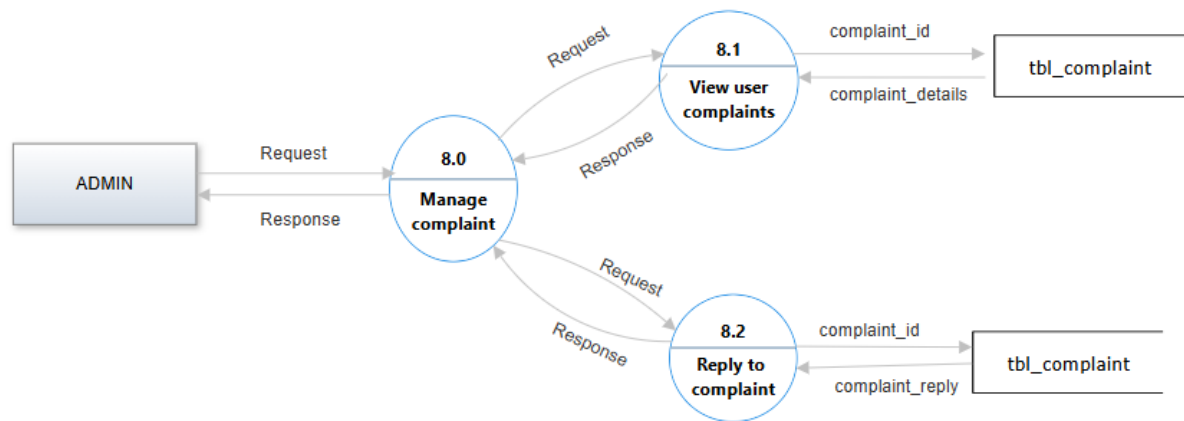
3) Seller



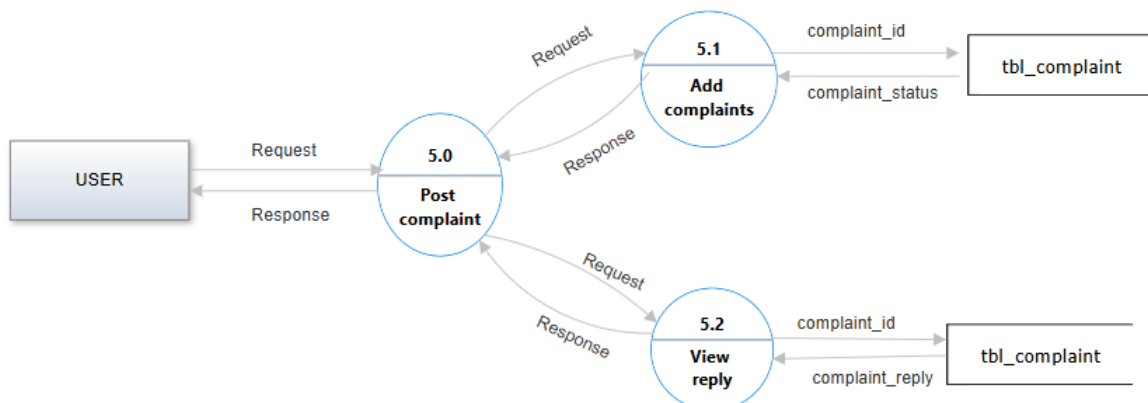
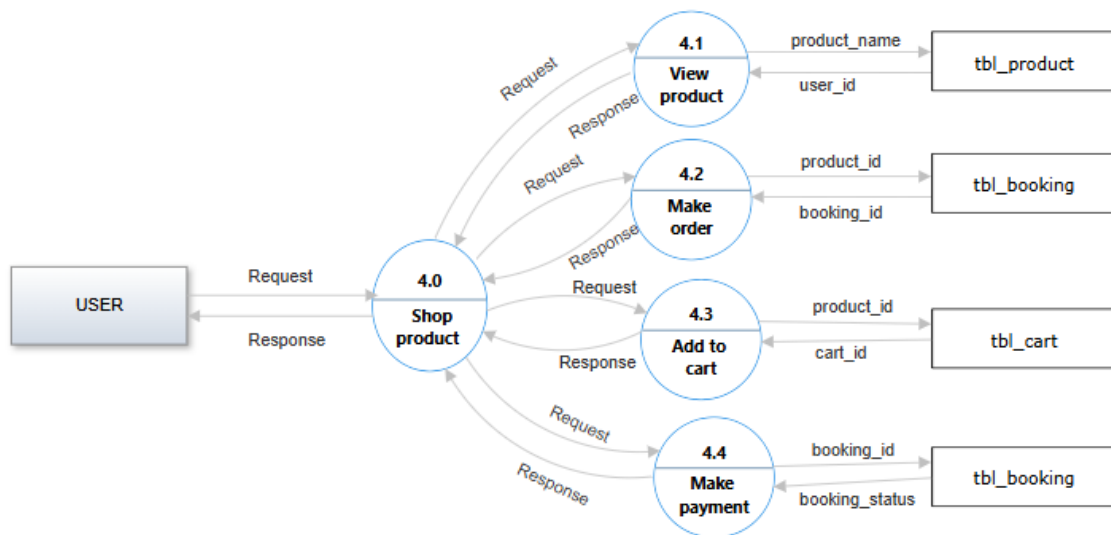
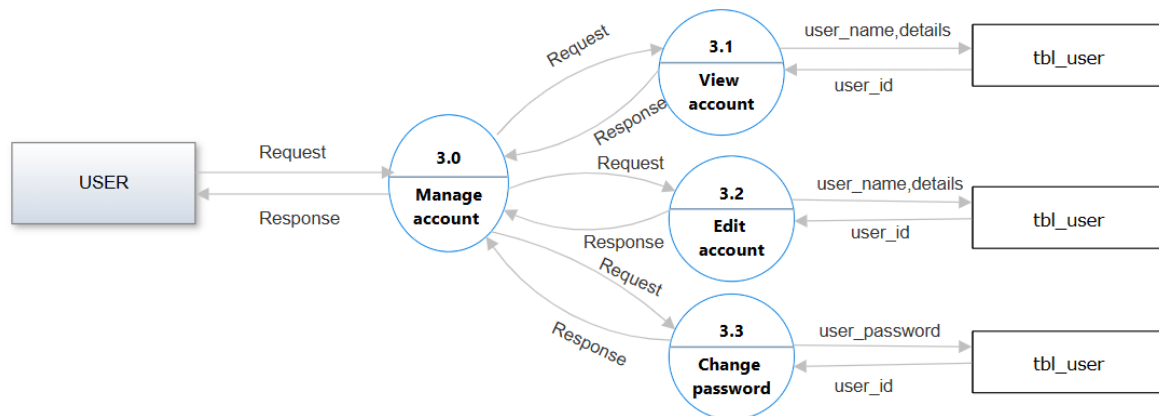
LEVEL 2

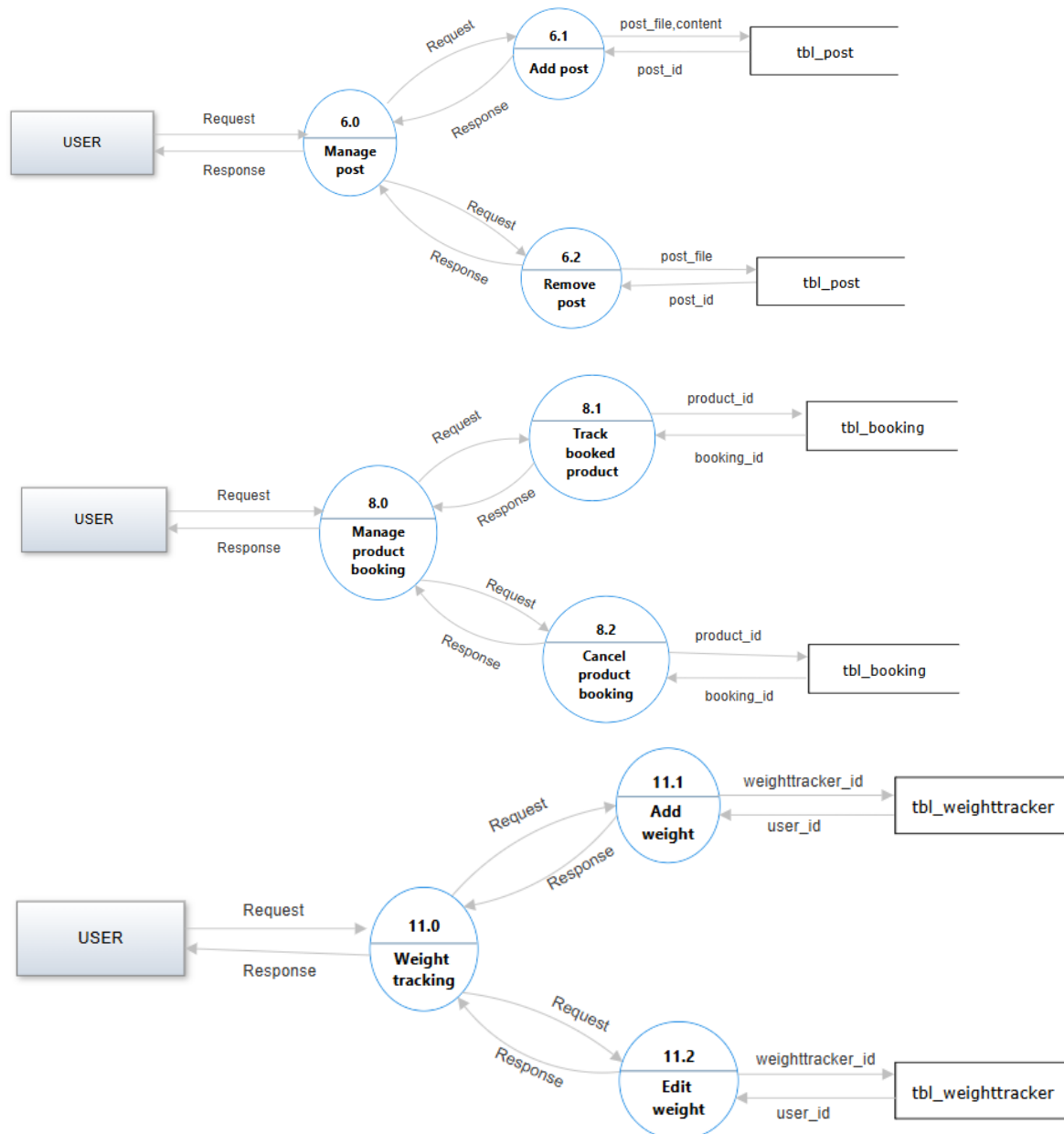
1)Admin



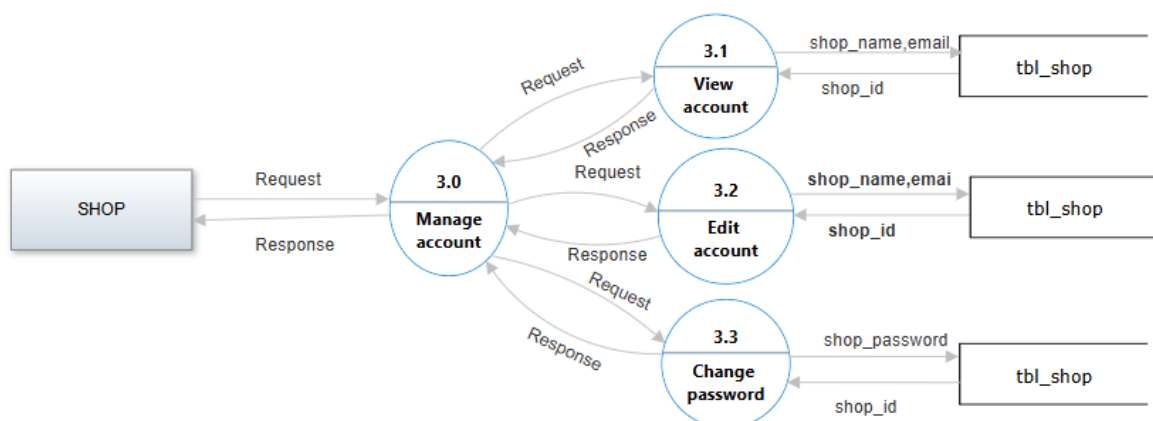


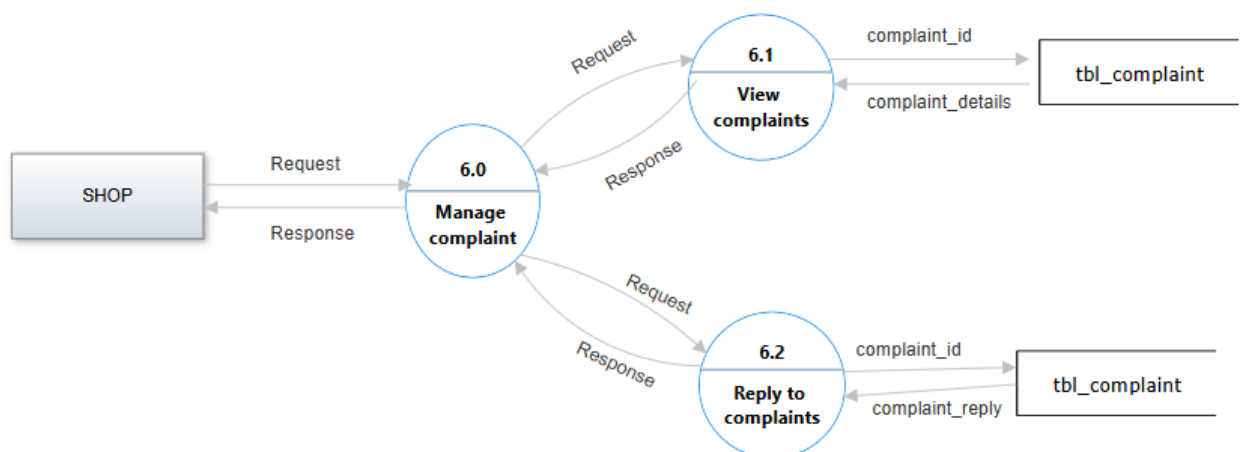
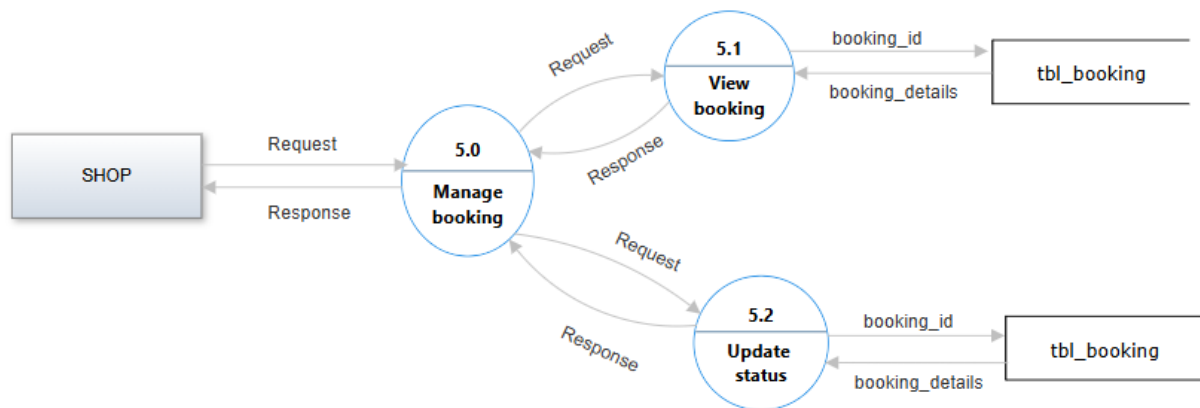
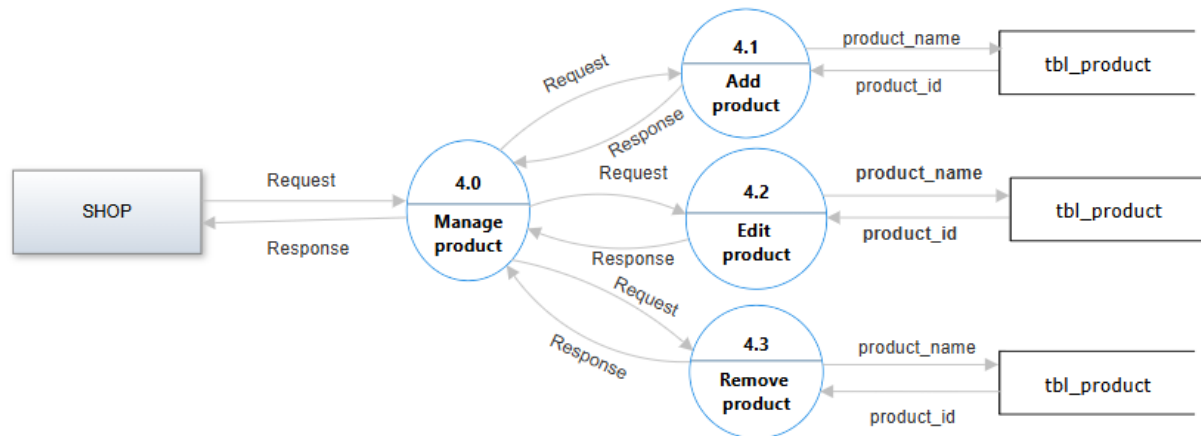
2) User





3)Shop





3.4.INTERFACE DESIGN

These modules can apply to hardware, software or the interface between a user and a machine. An example of a user interface could include a GUI, a control panel for a nuclear power plant, or even the cockpit of an aircraft. In systems engineering, all the inputs and outputs of a system, subsystem, and its components are listed in an interface control document often as part of the requirements of the engineering project. The development of a user interface is a unique field.

3.4.1.User Interface Screen Design

The user interface design is very important for any application. The interface design describes how the software communicates within itself, to system that interpreted with it and with humans who use it. The input design is the process of converting the user-oriented inputs into the computer-based format. The data is fed into the system using simple inactive forms. The forms have been supplied with messages so that the user can enter data without facing any difficulty. The data is validated wherever it requires in the project. This ensures that only the correct data have been incorporated into system. The goal of designing input data is to make the automation as easy and free from errors as possible. For providing a good input design for the application easy data input and selection features are adopted. The input design requirements such as user friendliness, consistent format and interactive dialogue for giving the right messages and help for the user at right are also considered for development for this project.

Input Design is a part of the overall design. The input methods can be broadly classified into batch and online. Internal controls must be established for monitoring the number of inputs and for ensuring that the data are valid. The basic steps involved in input design are:

- Review input requirements.
- Decide how the input data flow will be implemented.
- Decide the source document.
- Prototype on line input screens.
- Design the input screens.

The quality of the system input determines the quality of the system output. Input specifications describe the manner in which data enter the system for processing. Input design features can ensure the reliability of the system and produce results from accurate data. The input design also determines whether the user can interact efficiently with the system.

3.4.2.Output Design

A quality output is one, which meets the requirements of end user and presents the information clearly. In any system result of processing are communicated to the user and to the other system through outputs. In the output design it is determined how the information is to be displayed for immediate need.

It is the most important and direct source information is to the user. Efficient and intelligent output design improves the system's relationships with the user and helps in decision -making. The objective of the output design is to convey the information of all the past activities, current

status and to emphasize important events. The output generally refers to the results and information that is generated from the system. Outputs from computers are required primarily to communicate the results of processing to the users.

Output also provides a means of storage by copying the results for later reference in consultation. There is a chance that some of the end users will not actually operate the input data or information through workstations, but will see the output from the system.

Two phases of the output design are:

1. Output Definition
2. Output Specification

Output Definition takes into account the type of output contents, its frequency and its volume, the appropriate output media is determined for output. Once the media is chosen, the detail specification of output documents are carried out. The nature of output required from the proposed system is determined during logical design stage. It takes the outline of the output from the logical design and produces output as specified during the logical design phase.

In a project, when designing the output, the system analyst must accomplish the following:

- Determine the information to present.
- Decide whether to display, print, speak the information and select the output medium.
- Arrange the information in acceptable format.
- Decide how to distribute the output to the intended receipt.

Thus by following the above specifications, a high quality output can be generated.

SYETEM TESTING
AND
IMPLEMENTATION

4.1 SYSTEM TESTING

Coding conventions are a set of guidelines for a specific programming language that recommend programming style, practices and methods for each aspect of a piece program written in this language. These conventions usually cover file organization, indentation, comments, declarations, statements, white space, naming conventions, programming practices, programming principles, programming rules of thumb, architectural best practices, etc. These are guidelines for software structural quality. Software programmers are highly recommended to follow these guidelines to help improve the readability of their source code and make software maintenance easier.

The objective of system testing is to ensure that all individual programs are working as expected, that the programs link together to meet the requirements specified and to ensure that the computer system and the associated clerical and other procedures work together. The initial phase of system testing is the responsibility of the analyst who determines what conditions are to be tested, generates test data, produced a schedule of expected results, runs the tests and compares the computer produced results with the expected results with the expected results. The analyst may also be involved in procedures testing. When the analyst is satisfied that the system is working properly, he hands it over to the users for testing. The importance of system testing by the user must be stressed. Ultimately it is the user must verify the system and give the go-ahead.

During testing, the system is used experimentally to ensure that the software does not fail, i.e., that it will run according to its specifications and in the way users expect it to. Special test data is input for processing (test plan) and the results are examined to locate unexpected results. A limited number of users may also be allowed to use the system so analysts can see whether they try to use it in unexpected ways. It is preferably to find these surprises before the organization implements the system and depends on it. In many organizations, testing is performed by person other than those who write the original programs. Using persons who do not know how certain parts were designed or programmed ensures more complete and unbiased testing and more reliable software.

Parallel running is often regarded as the final phase of system testing. Since the parallel operation of two systems is very demanding in terms of user resources it should be embarked on only if the user is satisfied with the results of testing -- it should not be started if problems are known to exist. Testing is the major quality control measure during software development. Its basic function is to detect errors in the software.

Thus the goal of testing is to uncover requirement design and coding errors in the program. Testing is the process of correcting a program with intends of finding an error. Different types of testing are,

1. Unit Testing
2. Integrated Testing
3. Black Box Testing
4. White Box Testing
5. Validation Testing
6. User Acceptance Testing

4.1.1. Unit Testing

In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. In this testing we test each module individual and integrated the overall system. Unit testing focuses verification efforts on the smaller unit of software design in the module. This is also known as module testing. The modules of the system are tested separately. The testing is carried out during programming stage itself. In this testing step each module is found to working satisfactory as regard to the expected output from the module. There are some validation checks for verifying the data input given by the user which both the formal and validity of the entered. It is very easy to find error debug the system.

4.1.2. Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed. Data can be lost across an interface; one module can have an adverse effect on the other sub functions when combined by, may not produce the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors within the interface. This testing was done with sample data. The developed system has run success full for this sample data. The need for integrated test is to find the overall system performance.

Integration testing is a logical extension of unit testing. In its simplest form, two units that have already been tested are combined into a component and the interface between them is tested. A component, in this sense, refers to an integrated aggregate of more than one unit. Integration testing identifies problems that occur when units are combined. By using a test plan that requires you to test each unit and ensure the viability of each before combining units, you know that any errors discovered when combining units are likely related to the interface between units. This method reduces the number of possibilities to a far simpler level of analysis. Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

4.1.3. Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well. In black box testing the structure of the program is not considered. Test cases are decided solely on the basis of the requirements or the specification of the program or module, and the internals of the module or program are not considered for selection of the test cases.

In the Black Box testing tester only knows the input that can be given to the system and what output the system should give. In other words, the basis of deciding test cases in functional testing is requirements or specifications of the system or module. This form of testing is also called functional or behavioural testing.

4.1.4. White Box Testing

White-box testing (also known as clear box testing, glass box testing, and transparent box testing and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT).

While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system-level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements. White Box testing is concerned with testing the implementation of the program. The intent of this testing is not to exercise all the different input or output conditions but to exercise the different programming structures and data structures used in the program.

4.1.5. Validation Testing

At the culmination of Black Box testing, software is completely assembled as a package, interface errors have been uncovered and corrected and final series of software tests, Validation tests begins. Validation testing can be defined many ways but a simple definition is that validation succeeds when the software functions in a manner that can be reasonably accepted by the customer. After validation test has been conducted one of the two possible conditions exists.

1. The function or performance characteristics confirm to specification and are accepted.
2. A derivation from specification uncovered and a deficiency list is created.

4.1.6. User Acceptance Testing

Acceptance Testing is a level of the software testing process where a system is tested for acceptability. User Acceptance testing is the software testing process where system tested for acceptability & validates the end to end business flow. Such type of testing executed by client in separate environment & confirms whether system meets the requirements as per requirement specification or not.

User Acceptance testing also known as Customer Acceptance testing (CAT), if the system is being built or developed by an external supplier. The CAT or UAT are the final confirmation from the client before the system is ready for production. The business customers are the primary owners of these UAT tests. These tests are created by business customers and articulated in business domain languages. So ideally it is collaboration between business customers, business analysts, testers and developers. It consists of test suites which involve multiple test cases & each test case contains input data (if required) as well as the expected output. The result of test case is either a pass or fail.

4.2 IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned into a working system. The implementation stage is a systems project in its own right. It includes careful

planning, investigation of current system and its constraints on implementation, design of methods to achieve the changeover, training of the staff in the changeover procedure and evaluation of changeover method.

4.2.1. CODING STANDARDS

Writing an efficient software code requires a thorough knowledge of programming. This knowledge can be implemented by following a coding style which comprises several guidelines that help in writing the software code efficiently and with minimum errors. These guidelines, known as coding guidelines, are used to implement individual programming language constructs, comments, formatting, and so on. These guidelines, if followed, help in preventing errors, controlling the complexity of the program, and increasing the readability and understandability of the program.

A set of comprehensive coding guidelines encompasses all aspects of code development. To ensure that all developers work in a harmonized manner (the source code should reflect a harmonized style as a single developer had written the entire code in one session), the developers should be aware of the coding guidelines before starting a software project. Moreover, coding guidelines should state how to deal with the existing code when the software incorporates it or when maintenance is performed.

Since there are numerous programming languages for writing software codes, each having different features and capabilities, coding style guidelines differ from one language to another. However, there are some basic guidelines which are followed in all programming languages. These include naming conventions, commenting conventions, and formatting conventions.

1. **File header comments** are useful in providing information related to a file as a whole and comprise identification information such as date of creation, Name of the creator, and a brief description of the software code.
2. **Trailing comments** are used to provide explanation of a single line of code. These comments are used to clarify the complex code. These also specify the function of the abbreviated variable names that are not clear. In some languages, trailing comments are used with the help of a double slash (//).
3. **Indentation:** This refers to one or more spaces left at the beginning of statements in the program. Indentation is useful in making the code easily readable. However, the spaces used for indentation should be followed in the entire program.
4. **Implementing coding guidelines:** If coding guidelines are used in a proper manner, errors can be detected at the time of writing the software code. Such detection in early stages helps in increasing the performance of the software as well as reducing the additional and unplanned costs of correcting and removing errors. Moreover, if a well-defined coding guideline is applied, the program yields a software system that is easy to comprehend and maintain.

CONCLUSION

5. CONCLUSION

The project entitled “**MAMASPHERE**” was successfully completed on time, providing a seamless and interactive experience for users. The system was tested thoroughly and proved to be efficient, reliable, and user-friendly. It ensures smooth data management, security, and performance, making it highly effective for expectant mothers. The modular design of the system facilitates faster development, easy implementation, and future scalability.

Developed using Flutter (Dart) for the frontend and Supabase for the backend, MamaSphere is designed to be a versatile and feature-rich platform. The app and website were tested with various real-world scenarios, and all modules, including user authentication, diet tracking, exercise plans, maternity shopping, and community interaction, worked flawlessly. Each module was tested individually and integrated successfully, ensuring a smooth user experience.

The system requires minimal hardware resources and is accessible across different devices, making it convenient for users. MamaSphere fulfills all the objectives set at the beginning of the project, providing a comprehensive maternity care solution. I sincerely thank everyone who contributed to the successful completion of this project, making it a valuable and supportive platform for expectant mothers.

5.1 FUTURE ENHANCEMENTS

In the future, MamaSphere can be enhanced by integrating AI-powered health recommendations, real-time doctor consultations, and pregnancy progress predictions. Adding multilingual support will make the platform more accessible to a wider audience. Push notifications and reminders for diet, exercise, and medical checkups can further improve user engagement. Expanding Maternishop with more seller features, such as discount coupons and personalized product recommendations, will enhance the shopping experience. Additionally, integrating wearable device support for real-time health tracking can make MamaSphere a more advanced and comprehensive maternity care platform.

BIBLIOGRAPHY

6.BIBLIOGRAPHY

- SYSTEM ANALYSIS AND DESIGN
Elias M Award,
Second Edition, 1999,
Galgotica Publications, Delhi
- DATABASE SYSTEM CONCEPT
Silberschatz, Korth,
Sudarshan, Fourth
Edition.

<https://flo.health/>

<https://docs.flutter.dev/platform-integration/web/building>

<https://supabase.com/docs/reference/dart/upgrade-guide>

<https://github.com/shree-10/Pregcare>

https://www.firstcry.com/maternity-clothing-and-care?srsltid=AfmBOooWcX7qXbTNjdyWVf5sSLXKdSWpXdySkjuQkeX6pBXXf_cbQj7h

APPENDIX

7.APPENDIX

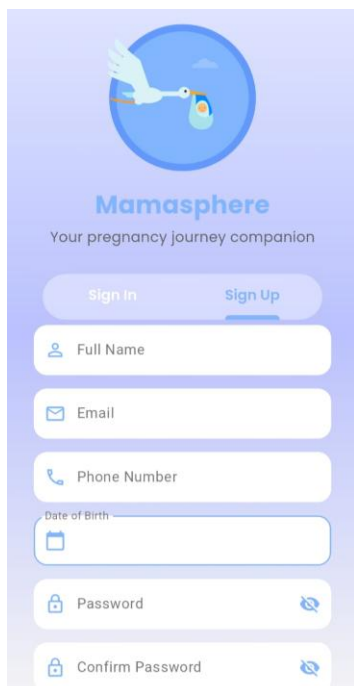
7.1 SCREENSHOTS

FORM 1 – USER LOGIN



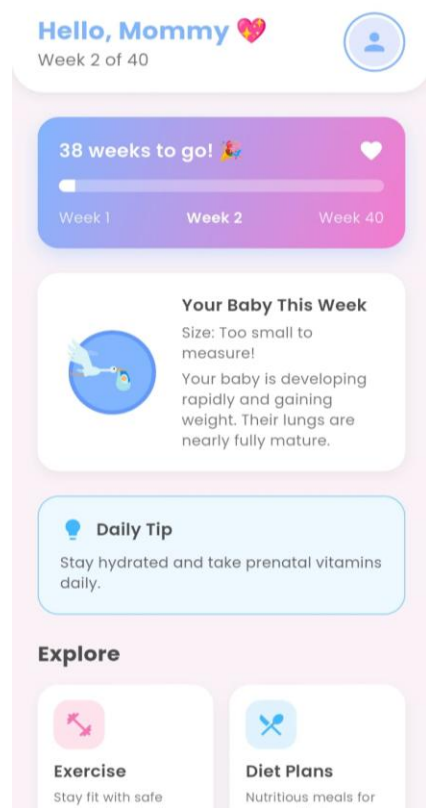
The screenshot shows the Mamasphere login interface. At the top is a circular logo featuring a white stork carrying a bundle. Below the logo is the text "Mamasphere" and "Your pregnancy journey companion". There are two buttons: "Sign In" (active) and "Sign Up". Below these are input fields for "Email" and "Password" (with a toggle icon). A "Login" button is positioned below the password field. At the bottom, it says "Don't have an account? [Sign Up](#)".

FORM 2-USER REGISTRATION



The screenshot shows the Mamasphere registration interface. It features the same logo and header as the login form. The "Sign Up" button is active. Below it are input fields for "Full Name", "Email", "Phone Number", and "Date of Birth" (with a calendar icon). There are also "Password" and "Confirm Password" fields, each with a toggle icon.

FORM 3- USER HOMEPAGE



Hello, Mommy ❤️
 Week 2 of 40

38 weeks to go! 🎉
 Week 1 Week 2 Week 40

Your Baby This Week
 Size: Too small to measure!
 Your baby is developing rapidly and gaining weight. Their lungs are nearly fully mature.

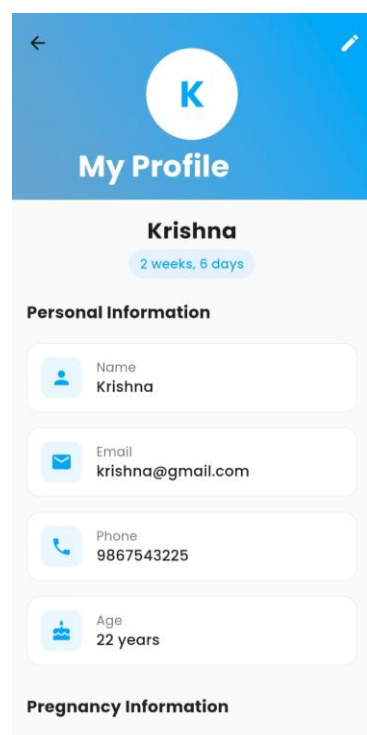
Daily Tip
 Stay hydrated and take prenatal vitamins daily.


Explore

Exercise
 Stay fit with safe

Diet Plans
 Nutritious meals for

FORM 4-USER PROFILE



← **K** 

My Profile

Krishna
 2 weeks, 6 days

Personal Information

Name
 Krishna

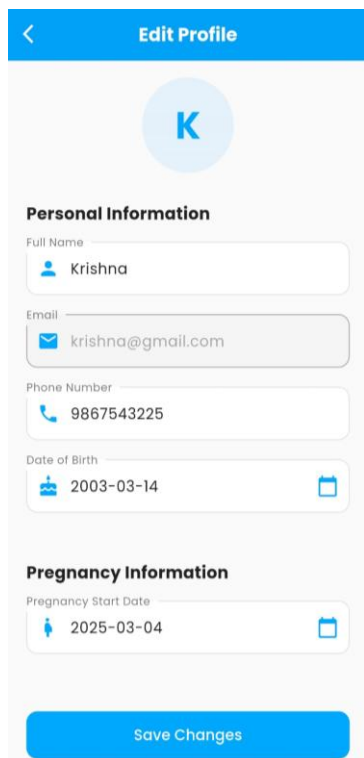
Email
 krishna@gmail.com

Phone
 9867543225

Age
 22 years

Pregnancy Information

FORM 5- EDIT PROFILE



The 'Edit Profile' form features a blue header with a back arrow and the title 'Edit Profile'. Below the header is a circular profile picture placeholder with a blue 'K'. The form is divided into two sections: 'Personal Information' and 'Pregnancy Information'. The 'Personal Information' section includes fields for 'Full Name' (Krishna), 'Email' (krishna@gmail.com), 'Phone Number' (9867543225), and 'Date of Birth' (2003-03-14). The 'Pregnancy Information' section includes a 'Pregnancy Start Date' (2025-03-04). A blue 'Save Changes' button is at the bottom.

Edit Profile

Personal Information

Full Name
Krishna

Email
krishna@gmail.com

Phone Number
9867543225

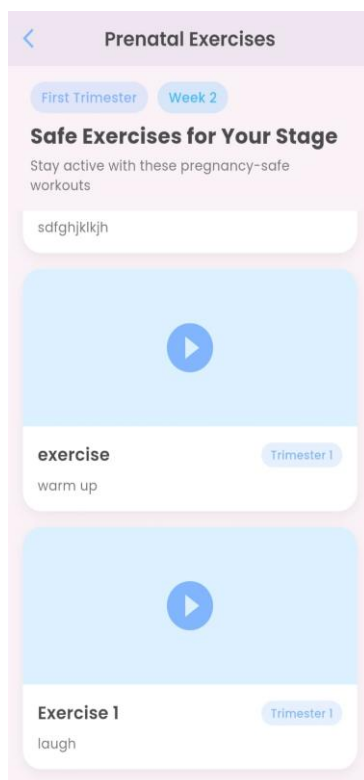
Date of Birth
2003-03-14

Pregnancy Information

Pregnancy Start Date
2025-03-04

Save Changes

FORM 6-VIEW EXERCISES



The 'Prenatal Exercises' form has a purple header with a back arrow and the title 'Prenatal Exercises'. Below the header are two tabs: 'First Trimester' and 'Week 2'. The main heading is 'Safe Exercises for Your Stage', followed by the text 'Stay active with these pregnancy-safe workouts'. There is a search bar containing 'sdfghjklkjh'. Below the search bar are two exercise cards. The first card is titled 'exercise' and 'warm up', with a 'Trimester 1' tag. The second card is titled 'Exercise 1' and 'laugh', also with a 'Trimester 1' tag. Each card has a blue play button icon.

Prenatal Exercises

First Trimester Week 2

Safe Exercises for Your Stage

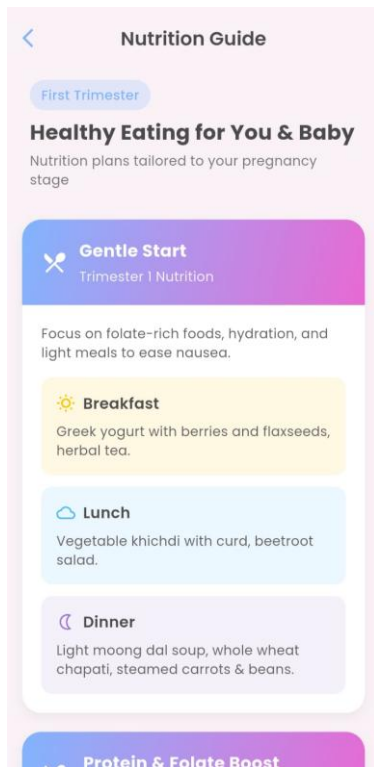
Stay active with these pregnancy-safe workouts

sdfghjklkjh

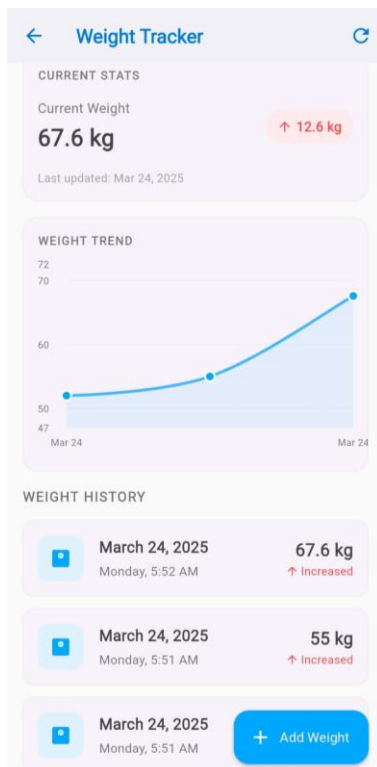
exercise Trimester 1
warm up

Exercise 1 Trimester 1
laugh

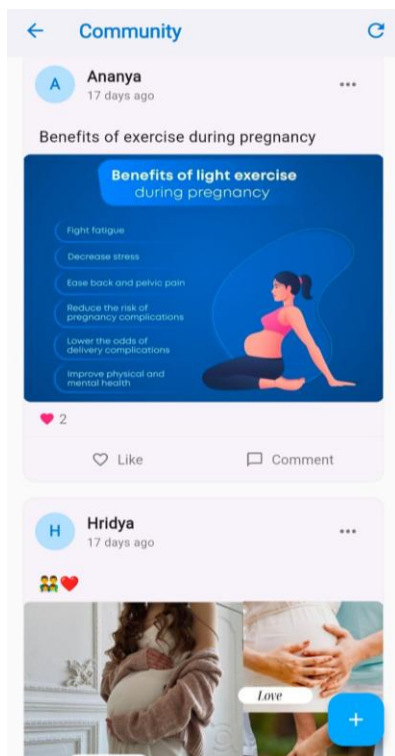
FORM 7- VIEW DIETPLAN



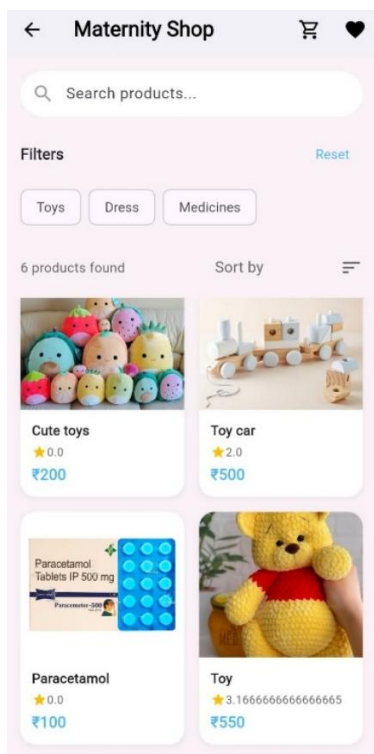
FORM 8- WEIGHT TRACKING



FORM 9- COMMUNITY POST




FORM 10-SHOPPING



FORM 11 – ADD TO CART

[←](#)
Paracetamol
[🛒](#)



Paracetamol
Tablets IP 500 mg

Paracemeter-500
Tablets

7 in stock

Paracetamol
 ★★★★★ 0.0 (0 reviews)
₹100
Description
 medicines



Add to Cart



♥

Customer Reviews

FORM 12-MY CART

[←](#)
Your Cart
[📦 My Orders](#)


Paracetamol
₹100
 - 1 + 



Dextromethorphan
₹150
 - 1 + 
 Only 1 left in stock

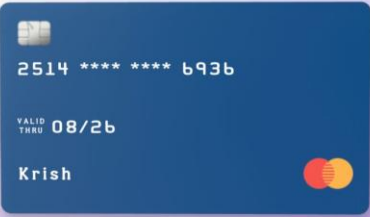
Subtotal **₹250.00**

Total **₹250.00**

Proceed to Checkout

FORM 13-PAYMENT

 Payment Gateway



2514 **** * 6936

VALID THRU 08/26

Krish

Card Number

2514 3686 9758 6936

Expiry Date

08/26

CVV


456

Card Holder

Krish

Pay Now

FORM 14-ORDER DETAILS

 Order #13

Order Status Delivered


Order Placed
Your order has been placed successfully

Processing
Your order is being processed

Shipped
Your order has been shipped

Delivered
Your order has been delivered

Order Information



Paracetamol
Qty: 1
Rs.100

Order Date

March 24, 2025 at 12:23 AM

Payment Status

Paid

Order Summary

Total

Rs.100

FORM 15-COMPLAINTS

← Complaints

Title:

Expired product

Write a Complaint:

Product expired before one month

Submit Report

FORM 16-BILL

← Leave a Review

Rate Your Experience

Your feedback helps us improve.



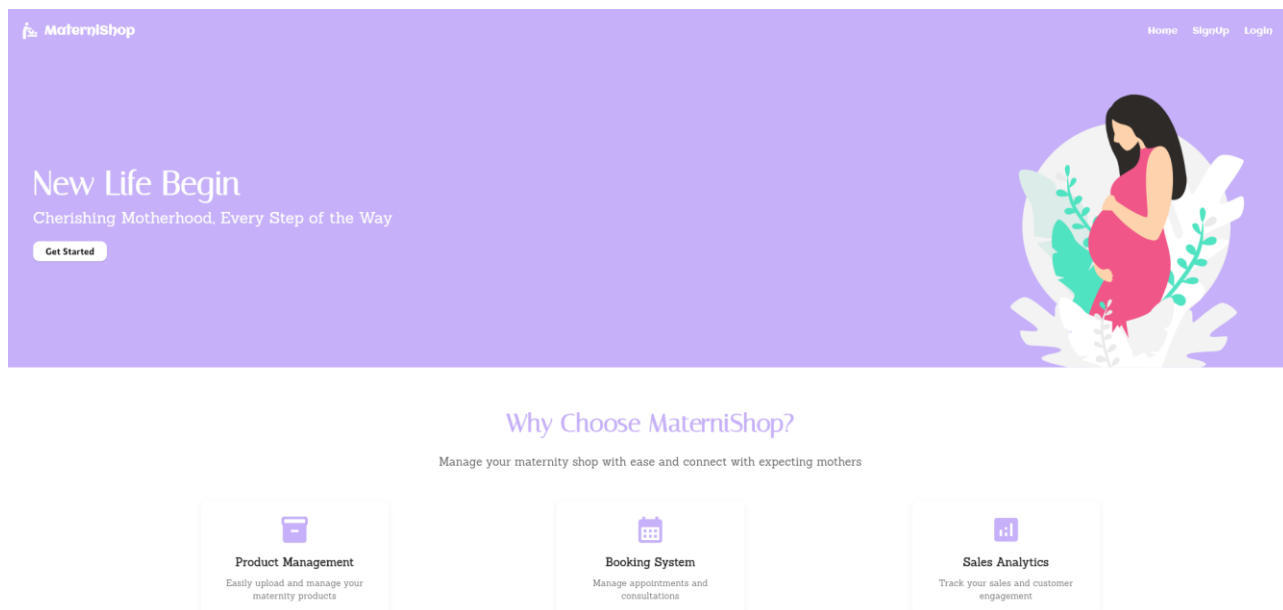
Good

Write a Review

Average

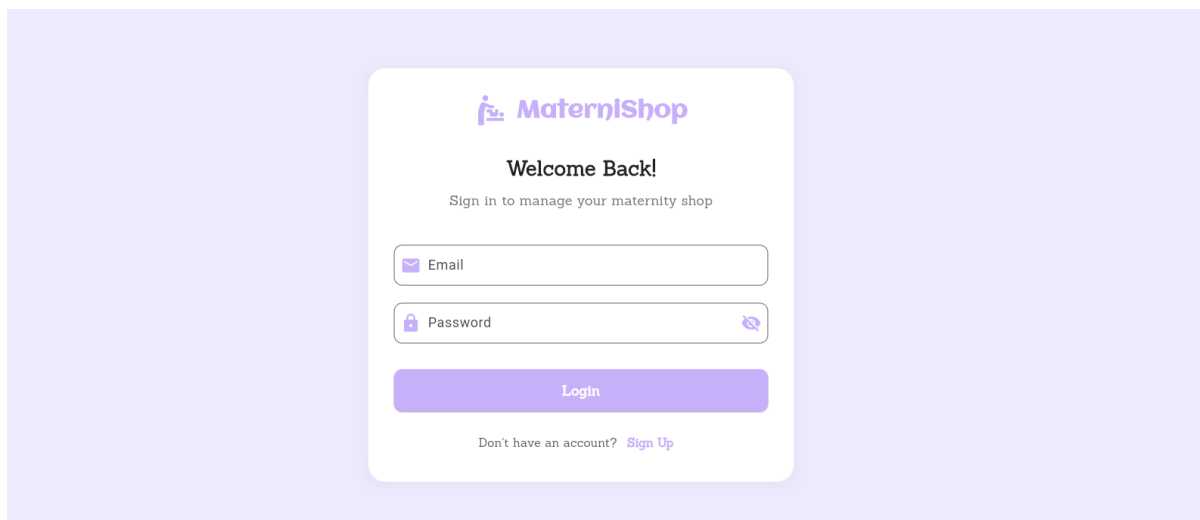
Submit Review

FORM 17-SHOP LANDING PAGE



FORM 18-SHOP REGISTRATION

FORM 19-SHOP LOGIN



MaternlShop

Welcome Back!
Sign in to manage your maternity shop

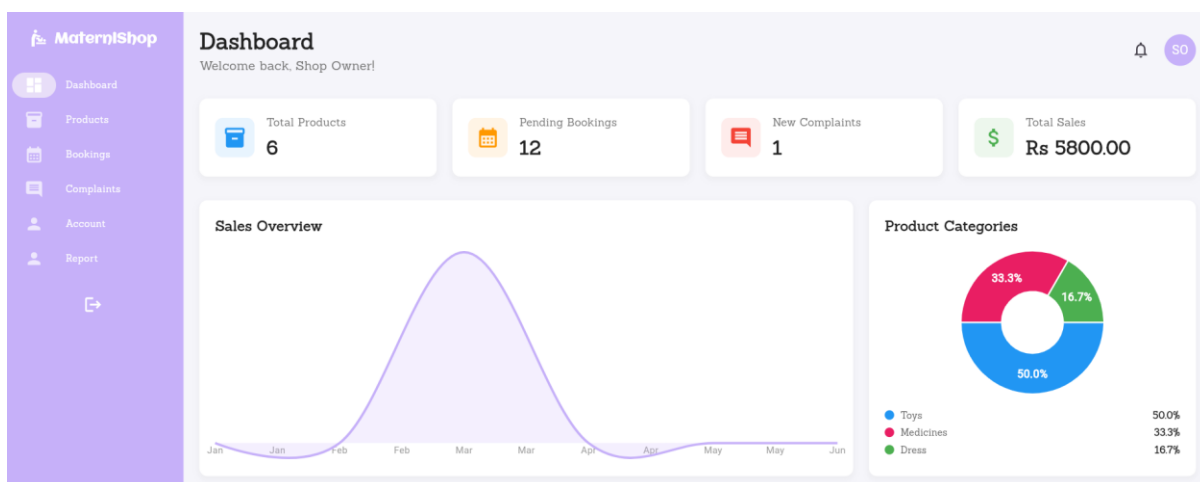
Email

Password

Login

Don't have an account? [Sign Up](#)

FORM 20- SHOP DASHBOARD



FORM 21- ACCOUNT MANAGEMENT



Account Management

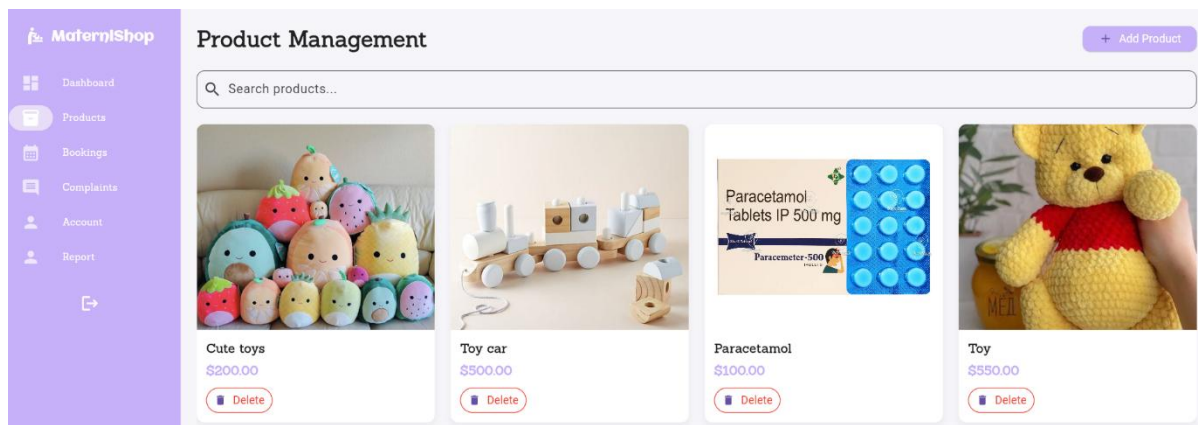
 **kunju**

Email: kunj@gmail.com

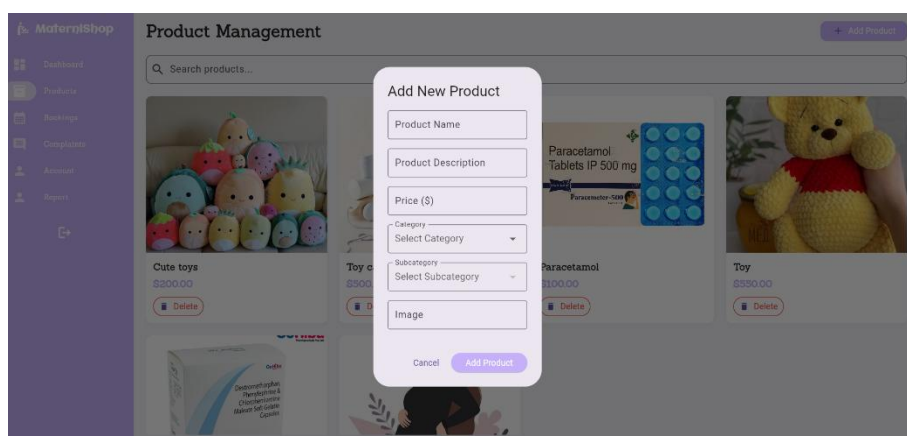
Contact: 9123456789

Address: Chottanikara

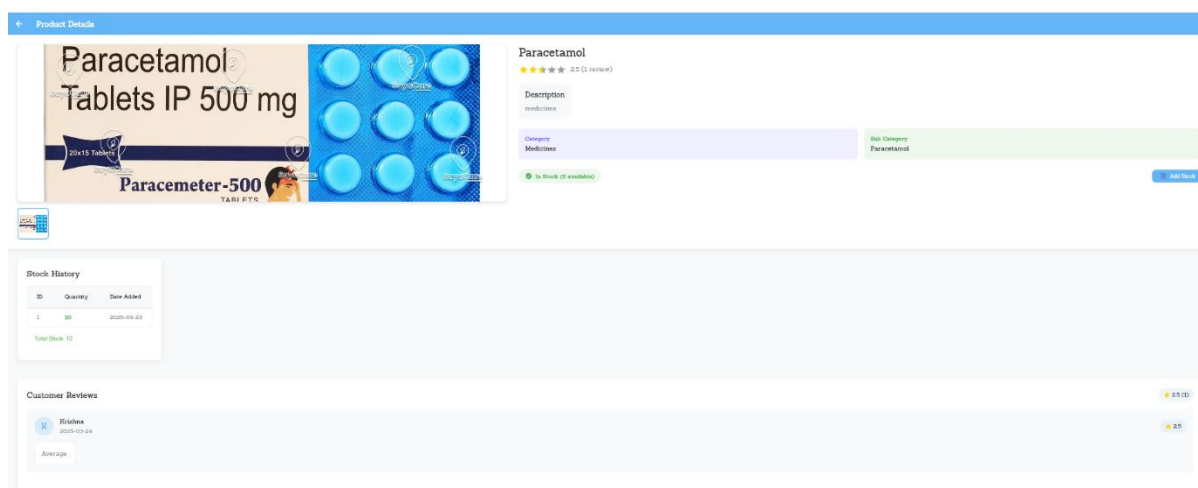
FORM 22- PRODUCT MANAGEMENT



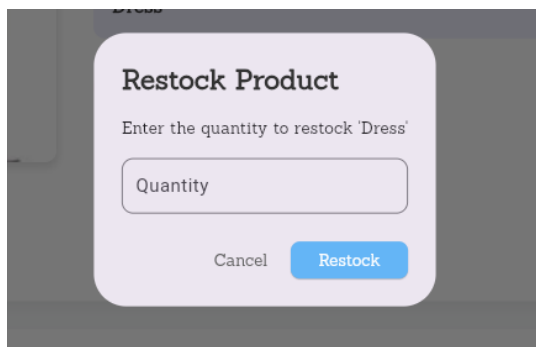
FORM 23-ADD PRODUCT



FORM 24- PRODUCT DETAILS



FORM 25-ADDING STOCK



Restock Product

Enter the quantity to restock 'Dress'

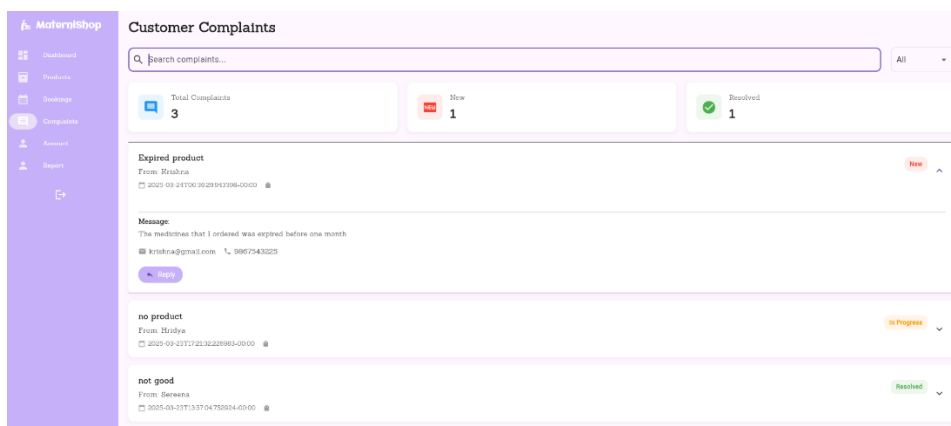
Quantity

Cancel Restock

FORM 26- BOOKING MANAGEMENT

| Booking Management | | | |
|--------------------|---|--|-----------|
| Sereena | 2025-03-11 01:34 8289977854 ramba@gmail.com | | Confirmed |
| Hridya | 2025-03-16 09:34 4579532678 hridya@gmail.com | | Confirmed |
| Hridya | 2025-03-17 00:01 4579532678 hridya@gmail.com | | Confirmed |
| Hridya | 2025-03-17 00:09 4579532678 hridya@gmail.com | | Confirmed |
| Sereena | 2025-03-19 06:16 8289977854 ramba@gmail.com | | Confirmed |
| Sereena | 2025-03-19 11:54 | | Confirmed |

FORM 27- COMPLAINTS AND REPLY



Customer Complaints

Search complaints... All

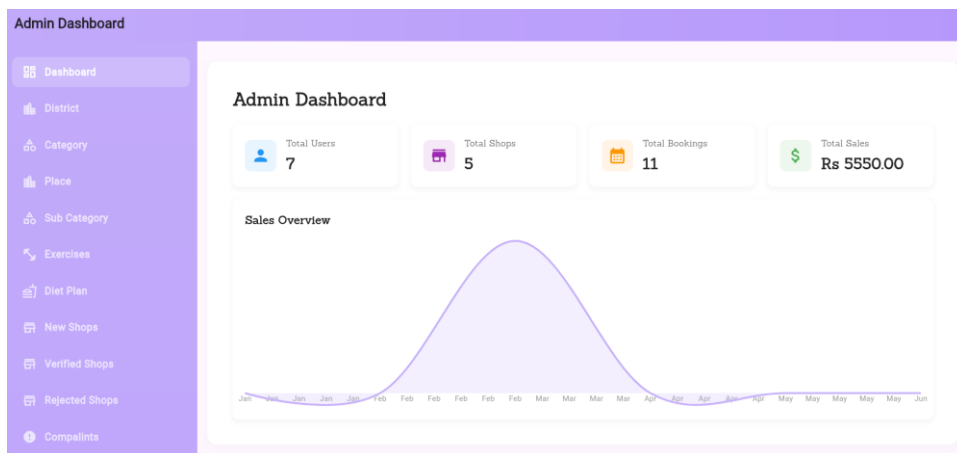
Total Complaints: 3 New: 1 Resolved: 1

Expired product From: Krishna
2025-03-20T03:30:29+05:30
Message: The medicines that I ordered was expired before one month
krishna@gmail.com 9967943225
Reply

no product From: Hridya
2025-03-20T17:32:26+05:30
In Progress

not good From: Sereena
2025-03-20T13:57:04+05:30
Resolved

FORM 28- ADMIN HOMEPAGE



FORM 29- ADDING DISTRICT

This form allows administrators to add new districts. It includes a sidebar with navigation options: Dashboard, District, Category, Place, Sub Category, Exercises, Diet Plan, New Shops, Verified Shops, Rejected Shops, and Complaints. The main content area features a text input field labeled 'Enter District' with an 'Add' button. Below the input field, there is a list of existing districts: 1. Ernakulam, 2. Kollam, and 3. Alleppuzha. Each entry has edit and delete icons.

FORM 30 – ADDING PLACE

This form allows administrators to add new places. It includes a sidebar with navigation options: Dashboard, District, Category, Place, Sub Category, Exercises, Diet Plan, New Shops, Verified Shops, Rejected Shops, and Complaints. The main content area features a dropdown menu labeled 'Select District' and a text input field labeled 'Enter Place' with an 'Add' button. Below the input fields, there is a list of existing places: 1. Thiruvankulam (Ernakulam), 2. Hospital junction (Kollam), 3. Chottanikkara (Ernakulam), 4. Punalur (Kollam), 5. Oachira (Kollam), and 6. Cherthala (Alleppuzha). Each entry has edit and delete icons.

FORM 31- ADDING CATEGORY

Admin Dashboard

- Dashboard
- District
- Category**
- Place
- Sub Category
- Exercises
- Diet Plan
- New Shops
- Verified Shops
- Rejected Shops
- Complaints

| | | | |
|---|-----------|--|--|
| 1 | Toys | | |
| 2 | Dress | | |
| 3 | Medicines | | |

FORM 32-ADDING SUBCATEGORY

Admin Dashboard

- Dashboard
- District
- Category
- Sub Category**
- Place
- Exercises
- Diet Plan
- New Shops
- Verified Shops
- Rejected Shops
- Complaints

| | | | |
|---|--------------------------------------|--|--|
| 1 | Animals Toys | | |
| 2 | Cars Toys | | |
| 3 | Paracetamol Medicines | | |
| 4 | Dextromethorphan Medicines | | |
| 5 | Caramacide Medicines | | |
| 6 | coughsyrupt Medicines | | |

FORM 33- NEW SHOP

Admin Dashboard

- Dashboard
- District
- Category
- Place
- Sub Category
- Exercises
- Diet Plan
- New Shops**
- Verified Shops
- Rejected Shops
- Complaints

Manage New Shops

Midhya
Email: midhya@gmail.com
Contact: 9865327452
Address: 17C Ashok nagar

FORM 34 – VERIFIED SHOP

Manage Verified Shop

Aleena
Email: aleena@gmail.com
Contact: 789456120
Address: Gokulam

Reject

kunju
Email: kunj@gmail.com
Contact: 9123456789
Address: Chottanikara

Reject

FORM 35– REJECTED SHOP

Manage Rejected Shops

Anagha
Email: anagha@gmail.com
Contact: 9856237845
Address: 24A.Royal nagar

Accept

Sradha
Email: sradha@gmail.com
Contact: 8289900956
Address: Home

Accept

FORM 36 – ADD NEW EXERCISE

Add New Exercise

Title

Description

Select Trimester

Upload File

Submit

FORM 37 – ADD DIET PLAN

Add New Dietplan

Title

Description

Breakfast

Lunch

Dinner

Select Trimester

Submit

FORM 38- SALES REPORT

Admin Dashboard

Dashboard

District

Category

Place

Sub Category

Exercises

Diet Plan

New Shops

Verified Shops

Rejected Shops

Complaints

Report

Sales Report

Start DateEnd Date

Jan 01, 2025Jun 30, 2025

Total Sales: Rs 5550.00

| Booking ID | Date | Total Amount | Items | Shop Name |
|------------|--------------|--------------|-----------------------------------|-----------|
| 12 | Mar 23, 2025 | Rs 600.00 | Cute toys (x3) - Rs 600.00 | Kunjju |
| 11 | Mar 23, 2025 | Rs 300.00 | Paracetamol (x3) - Rs 300.00 | Kunjju |
| 10 | Mar 23, 2025 | Rs 1000.00 | Toy car (x2) - Rs 1000.00 | Kunjju |
| 9 | Mar 22, 2025 | Rs 400.00 | Cute toys (x2) - Rs 400.00 | Kunjju |
| 8 | Mar 20, 2025 | Rs 200.00 | Cute toys (x1) - Rs 200.00 | Kunjju |
| 7 | Mar 19, 2025 | Rs 400.00 | Cute toys (x2) - Rs 400.00 | Kunjju |
| 6 | Mar 19, 2025 | Rs 400.00 | Cute toys (x2) - Rs 400.00 | Kunjju |
| 5 | Mar 17, 2025 | Rs 300.00 | Dextromethorphan (x2) - Rs 300.00 | Kunjju |
| 4 | Mar 17, 2025 | Rs 200.00 | Cute toys (x1) - Rs 200.00 | Kunjju |
| | | | Toy car (x2) - Rs 1000.00 | |