Baganbari - An Ecommerce Site For Plants

Submitted by

Hameem Md Sayem

ID: 211-51-045

Department of Information Technology & Management

Daffodil International University

Supervised by

Nafees Imran

Lecturer

Department of Information Technology & Management

Daffodil international University

This project report has been submitted in fulfillment of the requirements for the degree of Bachelor of Science in Information Technology & Management

Approval

This project titled on "**Baganbari– An E-Commerce Site for Plants**", submitted by "**Hameem Md Sayem, 211-51-045**", to the Department of Information Technology & Management, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Information Technology & Management, and approval as to its style and contents.

BOARD OF EXAMINERS

Chairman

Nusrat Jahan
Head, and Assistant Professor
Department of Information Technology & Management
Faculty of Science and Information Technology
Daffodil International University

Internal

Dr. Bimal Chandra Das
Associate Dean, Faculty of Science and Information Technology
Professor, Dept. of Information Technology & Management
Daffodil International University

Internal

Nafees Imran,
Lecturer
Dept. of Information Technology & Management
Faculty of Science and Information Technology
Daffodil International University

External

Dr. M. Shamim Kaiser
Professor
Institute of Information Technology
Jahangirnagar University

Declaration

I hereby declare that I have completed this project, named Baganbari, under the supervision of Mr. Nafees Imran, Lecturer, Department of Information Technology & Management (ITM), Daffodil International University. I also declare that this project or any part of it has not been submitted anywhere for the award of any degree.

Hameem Md Sayem

Hameem Md Sayem
Student ID: 211-51-045
Batch: 3rd
Department of Information Technology & Management (ITM)
Faculty of Science & Information Technology
Daffodil International University

Certified by:

Nafees Imran
Lecturer
Department of Information Technology & Management (ITM)
Faculty of Science & Information Technology
Daffodil International University

Acknowledgment

The successful completion of the ‘Baganbari’ project would not have been possible without the steadfast support, guidance and mentorship of many individuals and organizations. I would like to expand my heartfelt honor to everyone who has contributed to this achievement.

First and foremost, I express my deepest appreciation to my project supervisor, Nafees Imran, for his invaluable insights, constructive feedback, and continuous encouragement throughout the development of this project. His expertise was instrumental in overcoming challenges and achieving the project's objectives.

I am also thankful to Daffodil International University for providing the resources, facilities, and an environment helpful to learning and innovation. The appreciated support of faculty members and administrative staff was significant to the successful execution of this project.

Special recognition is due to the open-source communities and technology providers, including React.js, MongoDB, and Node.js, for offering robust tools and detailed documentation that played a vital role in building a scalable and efficient solution.

Finally, I express my deepest gratitude to my family, friends, and peers for their constant encouragement, patience, and belief in my abilities.

The Baganbari project is the result of collective efforts, and I hope it serves as a purposeful contribution to the digitization of plant retail and gardening practices.

Table Of Content

CHAPTER 1: INTRODUCTION

1.1 Background

The Baganbari project addresses an increasing demand for a digital solution in the home gardening and plant retail industry. With urbanization and a rising interest in home gardening, the need for convenient access to plants and gardening supplies has become more prominent. Traditional nurseries often operate without an online presence, which limits their reach and prevents customers from discovering and purchasing products efficiently.

The concept of Baganbari arises from the need to bridge this gap by providing an online platform that connects nursery owners with customers. By providing an online presence where nursery owners can list and manage their inventory, and customers can browse and purchase their favourite plants, the platform organizes the plant-buying process. Customers can easily find the plants they need, while nursery owners can access a bigger marketplace and tools to manage their operations more effectively.

The platform will also serve some features such as search filters, a secure checkout system, and smooth order management will enhance the overall customer experience and promote a digitalized approach to gardening.

1.2 Motivation

The idea behind Baganbari comes from a desire to bridge the gaps in the gardening and plant retail world. By using modern technology, we aim to make buying and selling plants easier and more accessible for everyone. Traditional ways of selling plants often fall short for both nursery owners and customers, highlighting the need for a simpler, more convenient solution that works for everyone.

Key motivating factors:

1. **Convenience for Customers:** Finding the right plants can be challenging, especially for city people. An online platform simplifies the search and purchase process.
2. **Empowering Nursery Owners:** Many nurseries face problems regarding knowledge and online access to establish an online presence. Baganbari provides a ready-to-use platform, enabling them to reach more customers and grow their businesses.
3. **First-Mover Advantage:** Being the first project of its kind in the country, Baganbari has the unique opportunity to tap into an unique market, establish a strong presence, and build a loyal customer base before any competitors appear.
4. **Streamlining Plant Retail:** Current systems often involve manual processes that are time-consuming and inefficient. The platform digitizes these operations, reducing complexities for both buyers and sellers.

1.3 Objectives

The primary objectives of the Baganbari project are:

1. To develop a web-based platform where nursery owners can create and manage their virtual stores and customers can browse and purchase plants conveniently.
2. To provide a seamless and efficient purchasing experience by enabling real-time search.
3. To implement powerful security measures for protecting sensitive user data and ensuring secure payment transactions.
4. To design a responsive, user-friendly interface that is accessible to users across all devices and technical proficiency levels.

1.4 Expected Outcomes

Upon the successful completion of the Baganbari project, the following outcomes are anticipated:

1. A fully functional web platform that connects nursery owners with customers.
2. Real-time product availability updates to ensure a smooth purchasing experience for customers.
3. A responsive user interface designed to meet the needs of all users , from beginners to gardening experts.
4. A flexible and reliable system architecture built to grow with the platform, supporting future upgrades like advanced analytics, multilingual options, and a larger user base.
5. Significant contribution to the gardening and plant retail industry by promoting accessibility, sustainability, and community engagement through a digital medium.

1.5 Report Layout

This report is structured into the following key sections to provide a comprehensive overview of the Baganbari project:

1. **Introduction:** Establishes the foundation of the project, detailing its background, motivation, objectives, and expected outcomes.
2. **Background:** Explores the context of the project, including market needs, scope, challenges, and comparisons with existing solutions.

3. **Requirement Specification:** Details both functional and non-functional requirements, supported by diagrams such as use case, activity, and sequence diagrams.
4. **Design Specification:** Outlines the architecture of the platform, including front-end and back-end design, interaction flow, and user experience principles.
5. **Implementation and Testing:** Provides an in-depth explanation of the development process, technologies used, testing strategies, and results.
6. **Conclusion and Future Scope:** Summarizes the findings, evaluates project success, and suggests potential areas for improvement and expansion.
7. **Appendix:** Includes supplementary materials, such as workflow diagrams, code snippets, and additional technical resources.
8. **References:** Lists all academic, technical, and practical resources utilized during the development of the project.

1.6 Comparative Studies

To evaluate the significance of the Baganbari project, a comparative study of existing plant retail systems and platforms was conducted. The analysis revealed several limitations in current solutions:

1. **Accessibility:** Many platforms lack mobile-friendly designs and fail to provide a seamless user experience, which discourages customer engagement.
2. **Inventory Management:** Existing systems often do not offer real-time updates for product availability, leading to frustration for customers trying to place orders.
3. **Data Security:** Instances of inadequate data protection and privacy issues have led to distrust among users.
4. **Customer Support:** Current platforms typically lack efficient customer interaction features, such as live support or order tracking.

In contrast, the Baganbari project addresses these shortcomings by offering:

1. **A Modern Interface:** Built with React.js, the platform ensures responsive, mobile-first designs and intuitive navigation for all users.

2. **Secure Data Management:** The system employs robust back-end technologies like MongoDB and Express.js, alongside encrypted connections, to safeguard user data.
3. **Real-Time Inventory Updates:** Nursery owners can manage their stock dynamically, ensuring customers see accurate product availability.
4. **Enhanced User Interaction:** Features such as personalized plant recommendations, detailed care guides, and live support foster a positive customer experience.

1.7 Scope of the Project

The scope of the Baganbari project is clearly defined to ensure a focused and impactful approach:

1. **To address inefficiencies in plant retailing:** The platform aims to simplify and streamline the process of purchasing plants, making it easier for customers to find and buy plants online.
2. **To provide end-to-end support:** From user registration to product delivery, Baganbari will ensure a seamless experience for both nursery owners and customers, including order tracking and payment processing.
3. **To ensure scalability:** The system is designed to handle growth in both user traffic and product inventory. Future expansions may include AI-driven plant recommendations, advanced analytics, and multilingual support for global deployment.

However, the project acknowledges certain limitations:

1. **Geographic Focus:** The initial deployment may be limited to specific regions, with future expansions planned based on market demand.
2. **User Adoption:** The success of the platform relies heavily on user participation. A large and active user base is essential for the platform to achieve its full potential.

1.8 Challenges

During the development of the Baganbari project, several key challenges were identified and addressed to ensure the platform's success and efficiency:

1. **Data Security:** Ensuring the protection of sensitive user information, such as personal details and payment data, was a critical concern. To mitigate this, strong encryption methods and secure authentication protocols, such as HTTPS and JWT, were implemented to safeguard all transactions and user data.

2. **Inventory Management:** Designing a system that effectively handles dynamic plant inventories posed challenges in terms of real-time updates, product availability, and stock management. To address this, a flexible and scalable backend was developed, allowing nursery owners to efficiently update their product listings and ensure accurate stock levels.

3. **User Adoption:** Encouraging both nursery owners and customers to adopt the platform was essential for its success. This required creating a user-friendly interface, educating users about the benefits of using Baganbari, and offering seamless customer support to build trust. Additionally, incentivizing early adoption through promotions and discounts helped in gaining initial traction.

CHAPTER 2 : REQUIREMENT SPECIFICATION

2.1 Business Process Modeling

The Baganbari platform operates through the following key phases to ensure a smooth and efficient experience for both nursery owners and customers:

1. **Registration and Verification:** Users (nursery owners and customers) register on the platform by providing necessary details, such as name, email, address, and phone number. Nursery owners also submit information about their store, including their product offerings. Customer accounts are verified to ensure secure interactions.

2. **Profile Management:** Both nursery owners and customers can update their profiles, including adding new plants to the inventory or modifying their personal information. Nursery owners can update their product listings, set prices, and manage stock availability, while customers can manage their orders, payment information, and preferences.

3. **Product Listing and Search:** Customers can browse or search the product catalog by various filters like plant type, price range, and availability. Nursery owners can list new plants, add detailed descriptions, images, and care instructions, and set availability to ensure customers see accurate information in real-time.

4. **Order Placement and Communication:** Once customers have found the plants they wish to purchase, they can add them to the cart and proceed to checkout. Secure payment gateways are integrated to process payments. After the order is placed, customers and nursery owners can communicate directly through secure channels on the platform to confirm details and coordinate delivery.

2.2 Stakeholder List:

The Baganbari platform involves several key stakeholders, each playing an important role in ensuring the success and smooth operation of the system:

Admin

- Manages the overall platform, including user registration, ensuring data security, and overseeing all platform operations.
- Monitors nursery owner and customer activity, ensures compliance with privacy standards, and manages content such as product listings, promotions, and system updates.
- Responsible for maintaining the platform's integrity, addressing user concerns, and enforcing platform rules.

Nursery Owners

- Users who register on the platform to create and manage their digital stores. They provide detailed information about their plant inventory, including descriptions, prices, and stock levels.
- Can update their plant listings, set prices, and manage product availability.
- Handle order processing, including fulfilling customer orders, confirming delivery, and managing customer communications.

Customers

- Individuals who use the platform to browse plant listings, place orders, and manage their profiles.
- Can search for plants based on specific criteria like plant type, price, and availability.
- Customers can contact nursery owners, track their orders, and receive delivery updates.

2.3 Functional Requirement:

1. **User Registration and Profile Management**
    - **Nursery Owners and Customers:** Users (nursery owners and customers) can register on the platform and create profiles with necessary details such as name, contact information, and preferences.

- **Profile Management:** Both nursery owners and customers can update their profiles, including store information for nursery owners and delivery preferences or order history for customers.

2. **Product Search and Matching**
    - **Search Functionality:** Customers can search for plants based on specific criteria such as plant type, price range, availability, and location.
    - **Matching System:** Search results should display a list of plants that match the customer's requirements, along with options to filter and sort results for easier browsing.
3. **Order and Appointment Scheduling**
    - **Order Placement:** Customers can add plants to their cart and schedule appointments for delivery or pick-up.
    - **Availability Management:** Nursery owners can update their product availability and set dates for delivery or customer pickups, allowing customers to schedule accordingly.
4. **Contact and Communication**
    - **Secure Communication:** Registered users (nursery owners and customers) can contact each other after providing the necessary details, such as order or product-related queries.
    - **Data Privacy:** The platform will securely handle the exchange of contact information, ensuring user privacy and complying with data protection standards.
5. **Educational Content**
    - **Gardening Resources:** The platform will provide valuable educational content on plant care, gardening techniques, and product usage.
    - **Plant Care Instructions:** Detailed instructions and care guides for plants listed on the platform will be available to customers for better plant maintenance.
6. **Community Features**
    - **Incentive Programs:** The platform will encourage customers to make regular purchases or nursery owners to offer loyalty rewards, building a community of gardening enthusiasts.

- **Recognition:** Nursery owners and customers can receive recognition and rewards for their continued participation and contributions to the platform.

7. **Admin Dashboard**
    - **User Management:** Admins can manage user data, including verifying profiles, approving new nursery owners, and monitoring activity to ensure compliance with platform policies.
    - **Content Management:** Admins can update product listings, manage promotions, and ensure the content provided is accurate and up-to-date.
    - **Compliance:** Admins are responsible for ensuring that the platform adheres to data privacy and security regulations, ensuring user data is protected.

2.4 Non-Functional Requirements:

Usability

- The platform should provide an intuitive and user-friendly interface, ensuring easy navigation for all users, regardless of their technical experience.
- Accessibility features must be included to support users with disabilities, including keyboard navigation, screen reader compatibility, and high-contrast themes for better visibility.

Performance

- The system must be capable of handling multiple simultaneous users without delays, ensuring that users can perform actions like searching for plants or completing purchases efficiently.
- Quick load times for search results, product pages, and checkout processes are essential to provide a smooth user experience.

Security

- The platform must implement robust security measures to protect sensitive customer and nursery owner data, including secure login mechanisms, encryption of personal information, and compliance with data protection regulations (e.g., GDPR, CCPA).
- Payment transactions must be securely processed using trusted payment gateways, ensuring customer trust and platform integrity.

Reliability

- The platform should offer high availability with minimal downtime. Scheduled maintenance and updates should be managed to avoid disruptions to user experience.
- Regular backups of user data and system configurations must be performed to prevent data loss in case of a failure.

Scalability

- The system must be designed to scale easily as the platform grows. This includes the ability to handle an increasing number of users, products, orders, and transactions over time without performance degradation.
- Cloud-based infrastructure or scalable server solutions may be used to ensure the platform can grow without requiring a complete redesign.

Compatibility

- The website must be compatible with various devices (desktop, tablet, and mobile) and web browsers (Chrome, Firefox, Safari, etc.), ensuring a consistent and responsive user experience across platforms.
- The platform should be optimized for both Android and iOS devices to ensure accessibility for mobile users.

Privacy

- Strict privacy controls must be implemented to ensure that personal and transactional information is only used for its intended purpose.
- User data should not be shared with third parties without explicit consent, and privacy policies must be transparent and easily accessible to users.

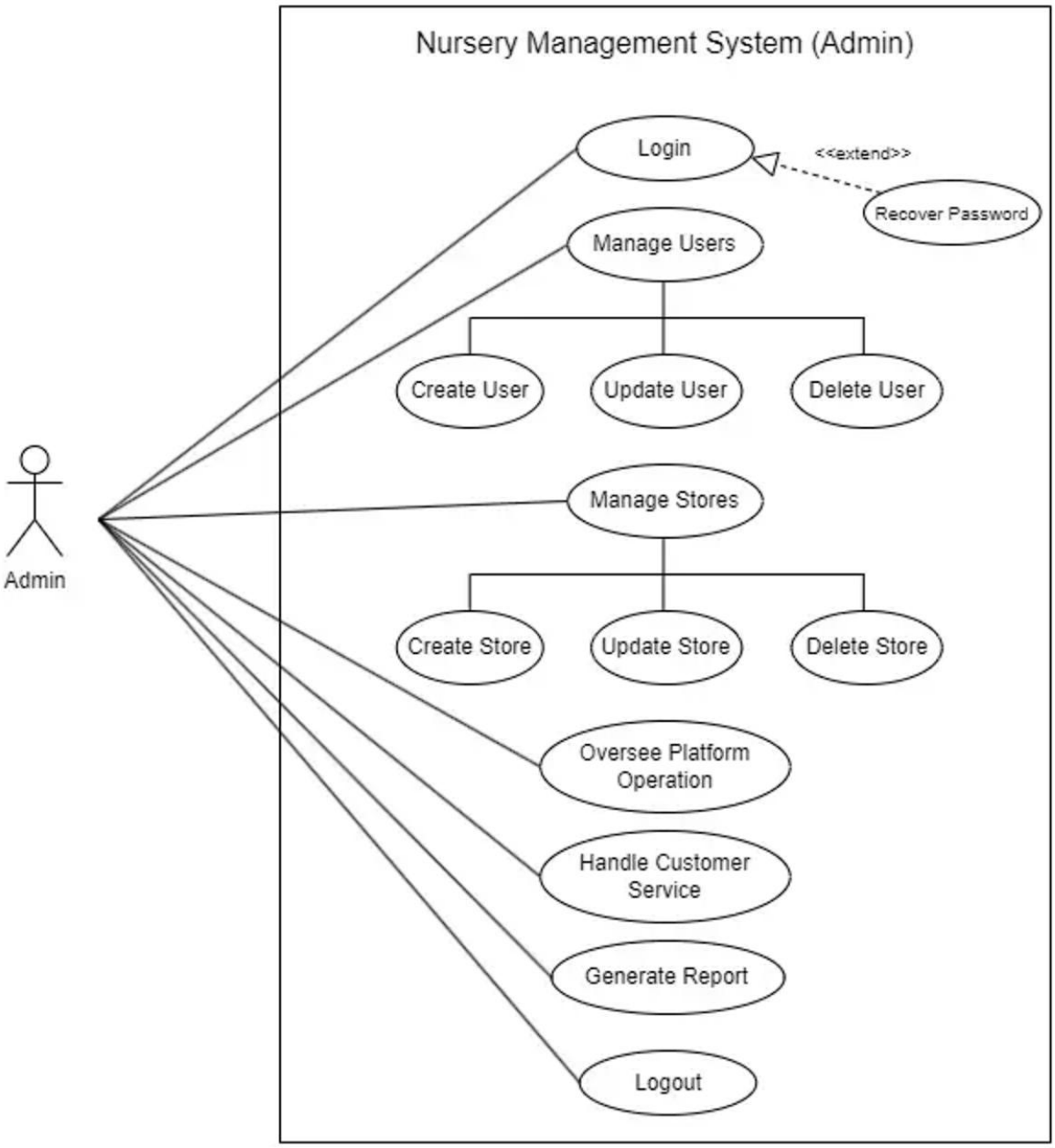
2.5 Diagrams

2.5.1 Use Case

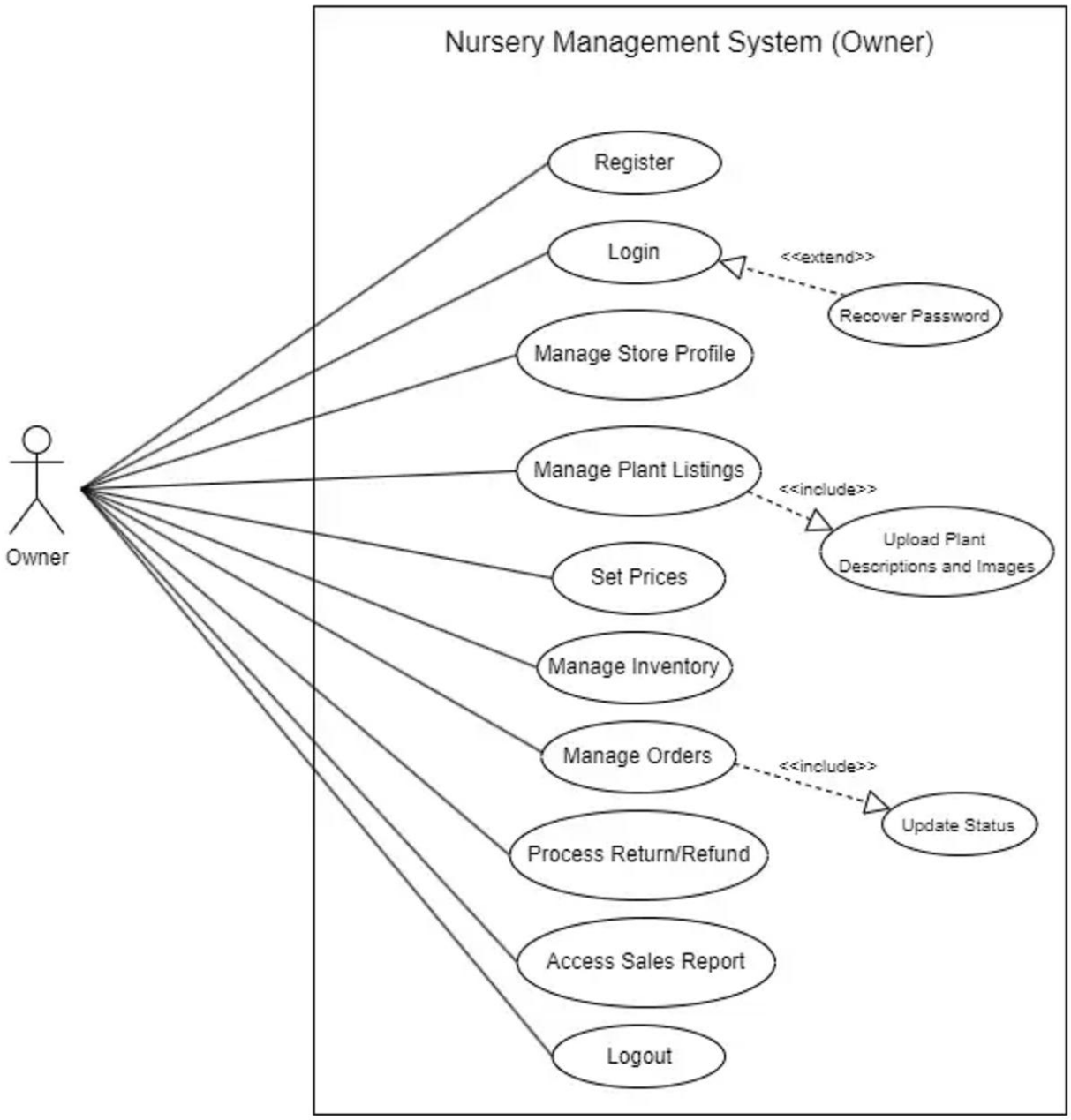
Actors:

1. Admin
2. Owner
3. Customer

Use Case: Admin

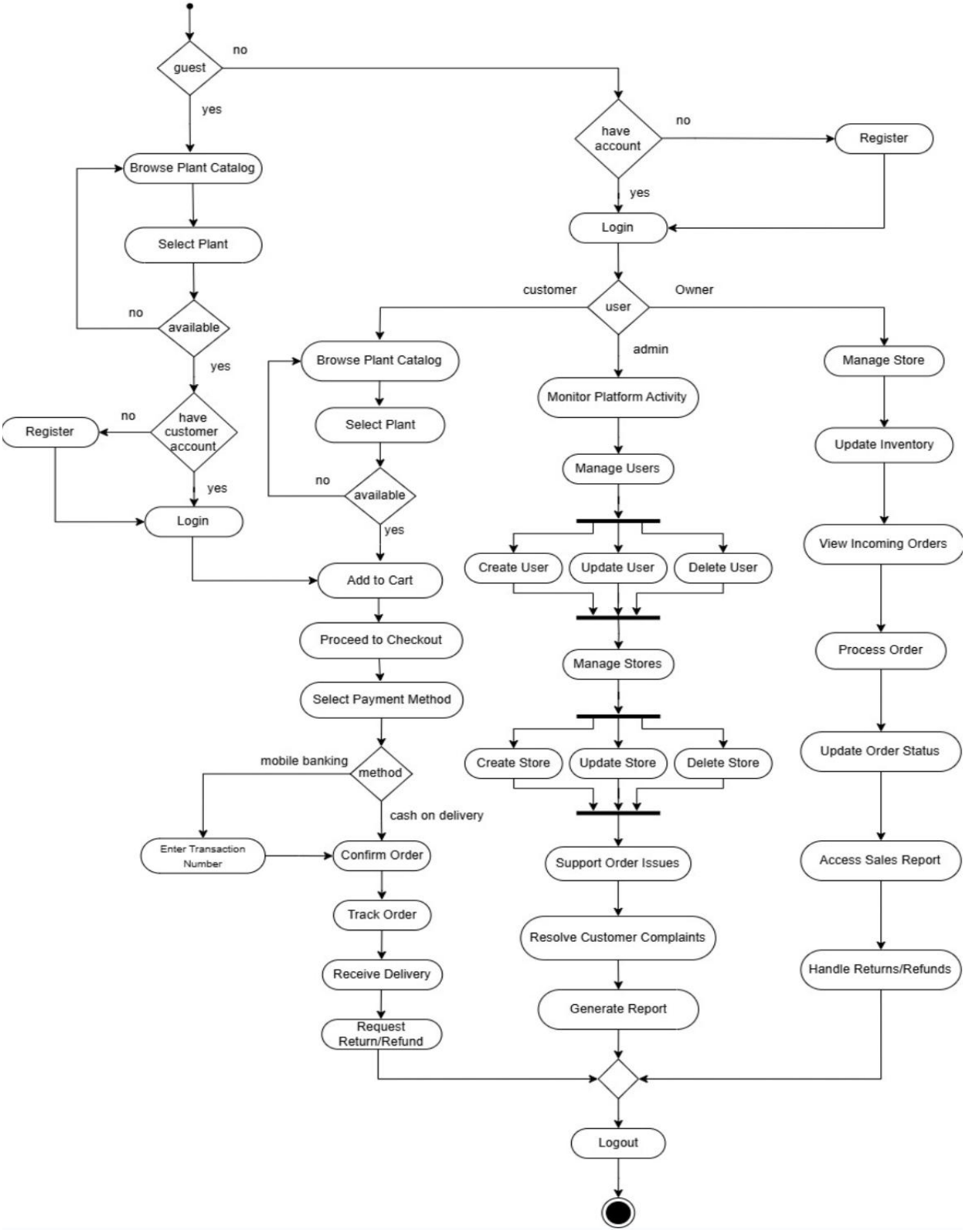


Use Case: Owner

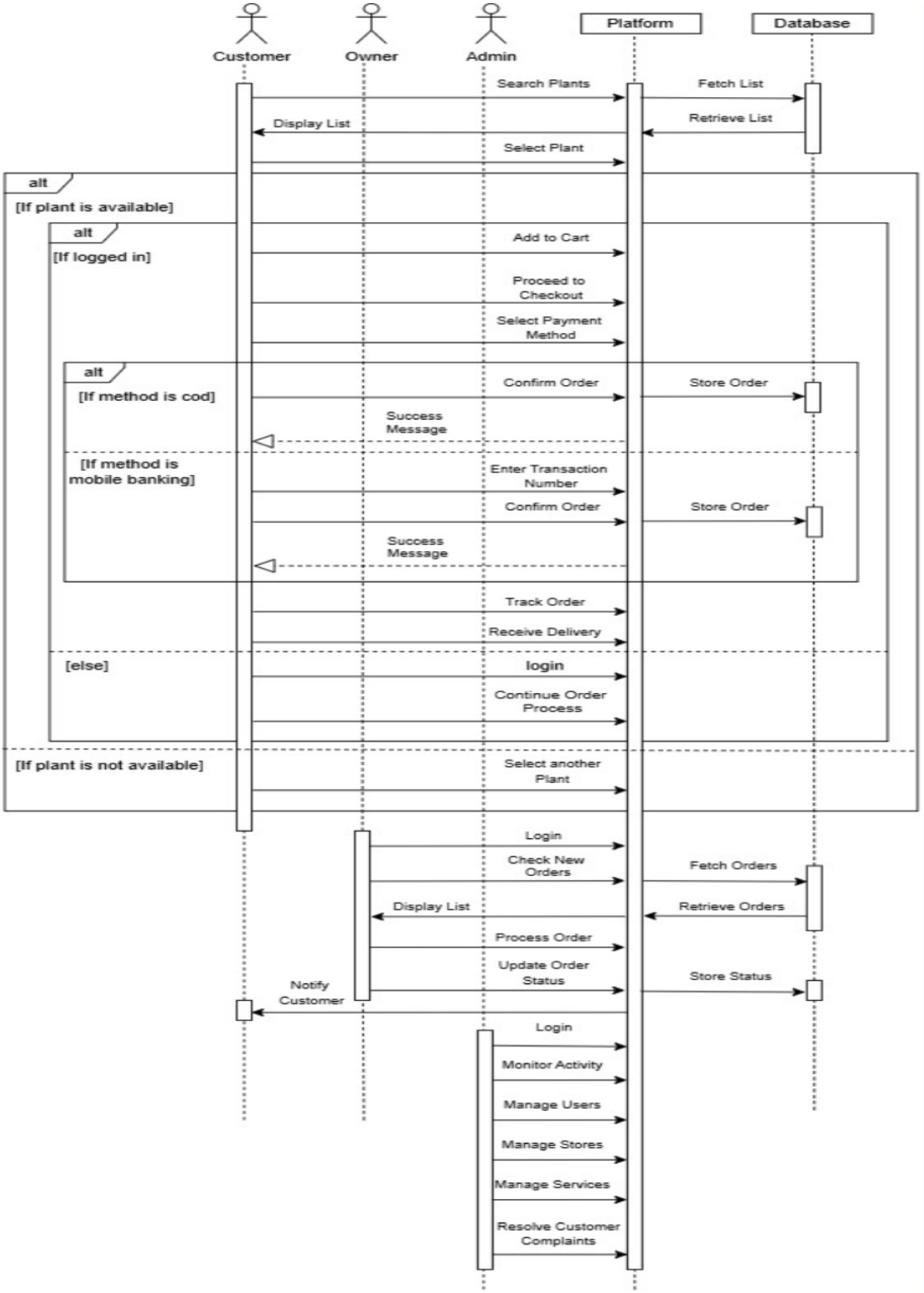


Use Case: Customer

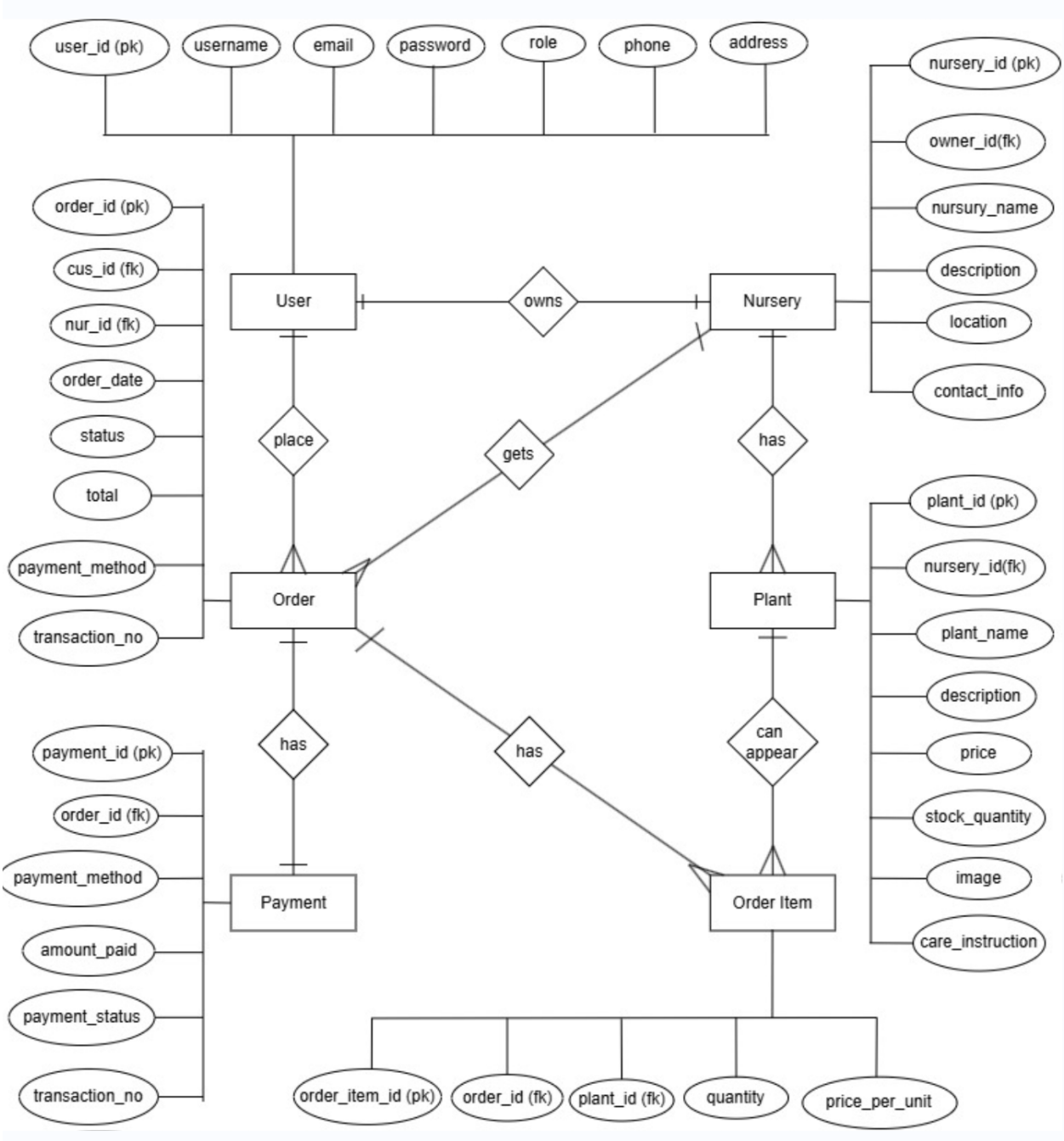
2.5.2 Activity Diagram



2.5.3 Sequence Diagram



2.5.4 Entity Relationship Diagram



2.6 Gantt Chart

Task	Duration	Status
Requirement Analysis	2 weeks	Completed
Frontend Development	6 weeks	Completed
Backend Development	8 weeks	Completed
Integration and Testing	4 weeks	Completed

2.7 SWOT Analysis

- **Strengths:**
    - The sector is not overwhelmed
    - Enhanced Visibility for Nurseries
    - Flexibility in Payment
- **Weaknesses:**
    - Dependence on Internet Access
    - Inventory Management Challenges
    - Initial Trust Issues
    - Limited Payment Integration
- **Opportunities:**
    - Growing interest in home gardening
    - Technological Integration
    - Already established sector
- **Threats:**
    - Seasonal Demand Fluctuations
    - Logistical Challenges

CHAPTER 3: DESIGN SPECIFICATION

This section outlines the design and structure of the Baganbari project, detailing how various components come together to create an effective, user-friendly, and robust system. It includes an in-depth exploration of the frontend design, backend architecture, user interaction design, and implementation requirements.

3.1 Front-End Design

The frontend of Baganbari is designed to provide a seamless, intuitive, and responsive experience for both nursery owners and customers. Key features include:

3.1.1 Framework

The frontend of Baganbari is built using React.js. It is a powerful JavaScript library for creating interactive user interfaces. This enables a component-based architecture, allowing for a modular design where each part of the interface is self-contained and reusable. React's virtual DOM enhances performance by efficiently rendering dynamic content, ensuring a fast and responsive user experience. The framework's ability to update only the changed parts of the user interface reduces unnecessary re-renders, further optimizing performance.

3.1.2 Styling

The Baganbari platform employs Tailwind CSS, a utility-first CSS framework that enables rapid prototyping and consistent styling across the entire application. Key features include:

- **Responsive Design**: Tailwind CSS ensures that the platform is fully accessible across devices of all screen sizes, including desktops, tablets, and mobile phones. This guarantees a seamless experience for users, regardless of the device they are using.

3.1.3 Key Pages

1. **Home Page:** The homepage serves as the entry point to the platform, welcoming users with an overview of Baganbari and highlighting its core features. It offers a clear introduction to the platform's purpose, emphasizing convenience and accessibility for both nursery owners and customers.
2. **Registration/Login:** This page allows users to securely register or log in, enabling profile creation and user authentication. It ensures that both nursery owners and customers can access personalized features and manage their accounts safely.
3. **Dashboard:** The dashboard provides users with personalized information, including order history for customers, inventory management for nursery owners, and notifications for new messages, orders, or updates. It allows users to manage their profiles and view relevant data in one central location.
4. **Search and Match Page:** This page allows customers to search for available plants based on their preferences, such as plant type, price range, and location. It also enables nursery owners to update their availability and showcase new plants, making it easier for customers to find what they need.

3.1.4 User Interactions:

1. **Real-Time Feedback:** The platform provides real-time feedback to users through loading indicators, confirmation messages, and error alerts. This ensures users are informed of their actions, such as when a product is successfully added to the cart, or if there is an issue with the information entered.
2. **Interactive Forms:** Forms throughout the platform include validation checks to ensure users provide accurate and complete information. Whether it's for user registration, product listing, or order placement, these forms guide users to correct any mistakes before submission, improving data integrity and user experience.

3.1.5 Frontend Accessibility:

The frontend of Baganbari is designed to meet international accessibility standards, ensuring that all users, regardless of their abilities, can interact with the platform effectively. Key features include:

- **Keyboard Navigability**: The platform supports full keyboard navigation, allowing users with motor impairments or those who prefer using a keyboard to navigate the website efficiently.
- **Screen-Reader Compatibility**: The design is optimized for screen readers, ensuring visually impaired users can access all content on the platform. This enhances inclusivity and provides a more equitable user experience for people with disabilities.

3.2 Back-End Design:

The backend of Baganbari handles the core logic, data storage, and processing. It is built using Node.js with the Express.js framework, which was chosen for its performance, scalability, and ease of handling asynchronous operations.

3.2.1 Architecture

- **Server-Side Framework**: Express.js provides a robust and flexible structure for building APIs, handling routing, and managing middleware. Its simplicity and extensive middleware support allow for efficient request handling and enhanced performance.
- **Database**: MongoDB is utilized for data storage, offering flexibility in managing unstructured data. This NoSQL database is ideal for handling diverse user profiles, product listings, and other dynamic elements like plant types, descriptions, and inventory. MongoDB’s scalability and schema-less structure allow Baganbari to efficiently store and retrieve data, making it suitable for future growth and feature additions.

3.2.2 Core Functionalities

- **User Management**: The platform supports secure user registration, login, and profile management. User credentials are securely stored using hashed passwords, and all data transfers are encrypted to protect user privacy.
- **Matching Algorithm**: A custom-built algorithm efficiently matches users (customers and nursery owners) based on criteria such as plant type, location, and availability. The algorithm ensures users are connected with the most suitable options based on their preferences and needs.
- **Notification System**: Real-time notifications are powered by WebSockets, ensuring prompt communication between customers and nursery owners regarding order status, updates, and new product listings.

3.2.3 Security Measures

- **Encryption**: All sensitive data is transmitted over secure channels using HTTPS, with additional data encryption protocols like TLS/SSL to ensure the integrity and confidentiality of user information.
- **Role-Based Access Control (RBAC)**: The system implements RBAC to restrict administrative privileges to authorized users only, ensuring that only designated staff can access sensitive features such as user management or financial data.
- **Data Validation**: All user inputs are validated to conform to expected formats, preventing errors or malicious inputs. This ensures that data stored in the database is accurate and secure.

3.2.4 Scalability

- The backend is designed with scalability in mind, capable of handling increasing user traffic as Baganbari grows. The architecture supports future integrations, such as AI-based plant recommendations, personalized search features, or global expansion. With a cloud-based infrastructure and modular design, the backend can efficiently scale to accommodate future demands and ensure continuous platform performance.

3.3 Interaction Design and UX

The interaction design and user experience (UX) strategies of Baganbari focus on enhancing usability and user satisfaction by providing an intuitive, seamless interface and navigation system that minimizes friction at every step.

3.3.1 User Journey Mapping

- From landing on the homepage to completing a purchase, every step of the customer and nursery owner journey is optimized for ease of use.
- For example:
    - **Nursery Owners** can update product availability and manage their inventory with minimal clicks.
    - **Customers** can search for plants by entering just a few key details, such as plant type, price range, and availability, allowing for a smooth browsing experience.

3.3.2 Navigation Design

- **Top Navigation Bar**: This contains key links to the dashboard, user profile, and help center, making essential sections of the platform easily accessible.
- **Side Panels**: Display additional options such as notifications, account settings, and quick access to customer support, ensuring all necessary tools are within reach.
- **Breadcrumb Trails**: Help users retrace their steps and maintain context as they navigate through product categories, search results, or personal settings.

3.3.3 Feedback Mechanisms

- **Success Messages**: Clear confirmation messages such as "Order successfully placed" or "Profile updated" provide users with immediate feedback that their actions have been successfully completed.
- **Error Prompts**: Users receive helpful error messages, such as "No plants found in this category; try refining your search," guiding them to correct actions.

- **Tooltips**: Short, informative tooltips appear to explain more complex actions or form fields, improving user understanding and reducing confusion.

3.3.4 Mobile-First Approach

- The platform is designed with a mobile-first philosophy, ensuring that all features are fully functional and optimized for smartphones. This approach guarantees seamless usability on smaller screens while maintaining a robust, feature-rich experience on larger devices such as tablets and desktops.

3.3.5 Testing for Usability

- User testing has been conducted with a diverse demographic, ensuring that the platform is accessible and intuitive for users with varying preferences, technical skills, and needs. This helps guarantee the system accommodates all user types, providing a satisfactory experience for both nursery owners and customers.

3.4 Implementation Requirements

The implementation of Baganbari requires a well-selected set of tools and technologies to ensure the platform is reliable, scalable, and efficient.

3.4.1 Frontend Tools

- **React.js**: Used for building dynamic and interactive user interfaces. React's component-based architecture allows for efficient development and easy updates, enhancing user experience.
- **Tailwind CSS**: A utility-first CSS framework used for rapid styling and ensuring a consistent, responsive design across the platform. It helps in creating clean and maintainable styles with minimal effort.

3.4.2 Backend Tools

- **Node.js and Express.js**: These technologies handle server-side logic, API creation, and routing. Node.js is well-suited for building fast and scalable network applications, while Express.js streamlines the creation of RESTful APIs.
- **MongoDB**: A NoSQL database used for managing user data, product listings, orders, and inventory. MongoDB's flexible schema allows for the easy handling of diverse data types and scalable growth.

3.4.3 Hosting and Deployment

- **Firebase**: Used for hosting static assets such as images and documents, as well as enabling serverless functions for lightweight backend processes.
- **Vercel**: Simplifies the deployment of frontend components by providing a seamless integration with React.js and ensuring fast, efficient performance.

3.4.4 Development Environment

- **Version Control**: Git and GitHub are used for collaborative development, version tracking, and managing the source code, enabling team members to work together efficiently.
- **Code Editor**: Visual Studio Code, a versatile code editor, is used for writing and debugging code. It includes extensions for linting, debugging, and version control, ensuring a smooth development process.

CHAPTER 4: Implementation and Testing

The database forms the backbone of the Baganbari platform, securely storing user profiles, product listings, order data, and transaction records. MongoDB, a NoSQL database, was chosen for its flexibility and ability to scale, which is ideal for handling the diverse and growing datasets associated with plant retailing.

4.1.1 Database Design

- **Collections**:
    1. **Users Collection**:
        - **Fields**: name, email, role (nursery owner or customer), contact details, profile settings.
        - **Purpose**: Stores user profile information for both nursery owners and customers, enabling personalized experiences and secure login.
    2. **Products Collection**:
        - **Fields**: plantName, type, price, description, stockAvailability, nurseryOwnerID.
        - **Purpose**: Stores details about plants listed by nursery owners, including availability, pricing, and descriptions.
    3. **Orders Collection**:
        - **Fields**: customerID, productID, quantity, totalPrice, orderStatus, orderDate.
        - **Purpose**: Tracks customer orders, including the plants purchased, quantities, total price, and order status (e.g., pending, shipped, delivered).
    4. **Transactions Collection**:
        - **Fields**: orderID, paymentMethod, paymentStatus, transactionDate.
        - **Purpose**: Logs transaction details for completed orders, including payment status and method used.

4.1.2 Data Security

- **Encryption**: All sensitive fields, such as contact details and payment information, are encrypted at rest using strong encryption protocols to protect user data.
- **Secure Connection**: Database connection strings are managed through environment variables, ensuring that sensitive credentials are not exposed in the codebase and preventing unauthorized access.

4.1.3 Optimization

- **Indexes**: Indexed fields, such as product type, price, and stock availability, enable high-speed queries for searching products and filtering by various criteria.
- **Aggregation Pipelines**: MongoDB's aggregation framework is used to perform complex queries, such as generating reports and dynamically calculating order totals or inventory statuses, improving performance during these operations.

4.2 Frontend Implementation:

The frontend of **Baganbari** delivers a responsive and user-friendly interface, ensuring seamless interaction between users and the platform. The design focuses on simplicity, ease of navigation, and responsiveness, providing an intuitive experience for both nursery owners and customers.

4.2.1 Technology Used

- **React.js**: Utilizes a modular, component-based architecture that allows for dynamic content rendering, enabling the platform to update only the necessary parts of the page without full reloads. This enhances the user experience by ensuring a fast and interactive interface.
- **Tailwind CSS**: A utility-first CSS framework that enables rapid development of consistent and clean designs. Tailwind CSS ensures a responsive layout and simplifies styling by providing pre-defined classes, which reduces the need for custom CSS.

4.2.2 Key UI Components

1. **Registration and Login Forms**:
    - **Real-Time Validation**: Ensures that users input valid data, with tooltips appearing when errors are detected (e.g., invalid email format).
    - **Example**: If the user enters "abc.com" as an email address, the system immediately prompts: "Please enter a valid email address."
2. **Dashboard**:
    - **Personalized Metrics**: Displays key information such as active orders, order history, and notifications. The dashboard is customized for different users, with distinct views for nursery owners, customers, and admins, making it easy to navigate and manage activities.
3. **Search Page**:
    - **Filters for Customers**: Customers can search for plants by specific criteria such as type, price range, and availability. Filters like price, location, and plant type allow for refined results.
    - **Dynamic Results**: The search results update instantly without requiring a page reload, providing a smooth browsing experience.

4.2.3 Responsive Design

- **Mobile-First Approach**: The platform is designed with a mobile-first philosophy, ensuring a fully optimized experience across all devices. The layout and UI components adjust seamlessly to different screen sizes, from smartphones to desktops.
- **CSS Media Queries**: Tailwind CSS makes use of media queries to adjust layouts and font sizes based on the device's screen size, ensuring readability and ease of navigation on both small and large screens.

4.3 Backend Implementation

The backend is the core of the Baganbari platform, responsible for managing data processing, business logic, and API interactions. It handles user authentication, product management, and order processing while ensuring scalability and security.

4.3.1 Technology Used

- **Node.js**: Utilized for its ability to handle asynchronous operations efficiently, ensuring that the backend can process multiple requests simultaneously without blocking, providing fast and responsive performance.
- **Express.js**: The framework for building RESTful APIs, enabling the backend to handle HTTP requests, manage routes, and interact with the database effectively. Express.js allows for a modular structure, making the backend development faster and more maintainable.

4.3.2 Core Functionalities

1. **Authentication and Authorization**:
    - **JWT (JSON Web Tokens)**: Used for secure session handling, allowing users to authenticate and authorize their actions securely across multiple requests. JWT ensures the integrity of user sessions and prevents unauthorized access to restricted areas.
    - **Role-Based Access Control (RBAC)**: Restricts access to specific features based on user roles, ensuring that only authorized personnel (e.g., admins) can access sensitive operations such as managing product listings and processing payments.
2. **Matching Algorithm**:
    - **Matching Criteria**: The platform matches nursery owners with customers based on the following factors:
        - Plant type availability and customer needs.
        - Geographic proximity to ensure local delivery.
        - Stock availability and customer preferences.
    - **Algorithm Workflow**:
        - Fetches available products from the Products collection based on the customer's search criteria.
        - Prioritizes matches by factors like stock availability, location proximity, and user preferences.
3. **Real-Time Notifications**:

- Implemented using WebSockets, the system ensures immediate updates for users when new products are listed, an order is confirmed, or when their preferences match a product in the catalog. This feature keeps users informed and engaged without needing to refresh the page.

4.3.3 Middleware and Error Handling

- **Middleware**: Custom middleware validates incoming requests to ensure that only valid and properly formatted data is processed. For instance, it checks if the data submitted through forms or API requests meets the expected format, preventing errors from entering the database.
- **Error Handling**: A centralized error handling system captures issues that occur throughout the platform. Detailed logs are generated for developers, helping them troubleshoot problems quickly. User-friendly error messages are displayed to users, ensuring they understand any issues without being overwhelmed by technical jargon.

4.4 interaction Mechanisms

The Baganbari platform ensures smooth and intuitive interactions between users and the system, enhancing the overall user experience and making interactions straightforward and efficient.

4.4.1 User Interactions

- **Registration:**
  - **Validation Rules:** When users register, mandatory fields such as email and password are validated to ensure they meet the necessary criteria, preventing errors and ensuring data consistency.
  - **Confirmation Message:** After successful registration, users receive a confirmation message, indicating that their account has been created successfully and they can proceed to log in.
- **Search and Match:**

    - **Real-Time Search Results:** As customers search for plants, the platform dynamically updates results based on the recipient's criteria, such as plant type, price range, and location. This provides an interactive and responsive browsing experience.
    - **Notifications:** When a new match is found, donors (nursery owners) receive instant notifications via WebSockets, ensuring prompt communication and increasing engagement.
- **Approval Workflow:**
    - **Match Approval:** Once a match is made, both nursery owners and customers can approve or decline the connection. This ensures that both parties are in control of the transaction process.
    - **Dashboard Updates:** The status of the match is reflected immediately in the user's dashboard, providing real-time feedback and keeping all parties informed about the progress of their interaction.

4.4.2 Admin Interactions

- **User Management:**
    - **Account Control:** Admins can deactivate or update user accounts as necessary, ensuring that only active, verified users have access to the platform. This helps maintain the integrity and security of the system.
- **Match Oversight:**
    - **Reviewing Matches:** Admins can review pending matches and decide whether to approve or provide feedback. They can also intervene to ensure that matches meet the platform's standards and guidelines before proceeding with the next steps.

4.5 Integration and Deployment

4.5.1 Integration

- **Frontend-Backend Communication**:

- APIs are used to facilitate seamless data exchange between the frontend and backend. Axios is utilized for making HTTP requests, ensuring smooth communication between the client-side and server-side components.
- **Example**: The /api/matches endpoint is used to fetch and return donor-recipient matches, allowing customers and nursery owners to see relevant information dynamically without needing to reload the page.

- **Real-Time Features**:
  - **WebSocket** connections are implemented to deliver instant notifications. Whenever there is a new match or an update to existing orders, users (such as nursery owners) receive real-time notifications, ensuring they stay informed and can take immediate action.

4.5.2 Deployment

1. **Frontend**:
   - The frontend is deployed on Firebase Hosting, which ensures low-latency global content delivery. Firebase provides reliable, fast hosting that allows the platform to serve users from any location without delays.
2. **Backend**:
   - The backend is hosted on Vercel, utilizing a serverless architecture for scalable API handling. Vercel automatically adjusts resources based on demand, ensuring the backend can handle variable traffic loads efficiently.
3. **Database**:
   - MongoDB Atlas is used to manage the database, ensuring secure and reliable cloud-based storage for all user data, products, orders, and transactions. MongoDB Atlas offers automatic backups and robust security measures, providing scalability and reliability for the platform's data needs.

4.6 Security Implementation

Data Encryption:

- Sensitive data is encrypted both in transit and at rest to ensure privacy and protect user information. HTTPS is used to encrypt data during transmission, ensuring secure communication between the client and server. Additionally, all sensitive data stored in the database is encrypted, reducing the risk of unauthorized access.

Authentication:

- JSON Web Tokens (JWT) are used to secure login sessions and authenticate users. JWT ensures that only authorized users can access protected resources and that their sessions remain secure throughout interactions with the platform. This method prevents unauthorized access and ensures data integrity across multiple requests.

Role-Based Access Control (RBAC):

- RBAC is implemented to segregate user and admin functionalities, enhancing security by ensuring that only authorized users can access specific platform features. For example, admins have exclusive access to user management and sensitive data, while regular users can only access their own profiles and relevant product information.

4.7 Key Challenges and Solutions

Real-Time Notifications:

- **Challenge**: During peak traffic periods, users experienced delays in receiving real-time notifications about new product matches or order updates.
- **Solution**: To resolve this, WebSocket configurations were optimized, and server resource allocation was improved to handle higher loads more efficiently. Additionally, load balancing techniques were implemented to distribute traffic evenly across multiple servers, ensuring consistent performance and timely delivery of notifications.

Validation Issues:

- **Challenge**: There were instances where edge cases bypassed frontend validation, allowing incorrect or incomplete data to be submitted to the backend.

- **Solution**: The backend middleware was enhanced with additional validation logic to catch any improperly formatted or incomplete data before it reaches the database. This ensures that only valid and sanitized data is processed, maintaining the integrity and accuracy of the platform's content and user information.

4.8 Testing Features

4.8.1 Test Case 01: User Login

Test Case #: 01
Test Case Name: User Login
Subsystem: User Login
Designed By: Hameem Md Sayem
Design Date: 2024-11-27
Executed By: Hameem Md Sayem
Execution Date: 2024-11-27

Test Steps & Expected Results:

1. **Action**: User fills in only the email field and clicks "Login". **Expected Result**: Prompts user to enter the password. **Status**: Pass
2. **Action**: User clicks "Login" without filling in any fields. **Expected Result**: Displays error for missing required fields. **Status**: Pass
3. **Action**: User enters an invalid email format (e.g., abc.com). **Expected Result**: Prompts user to provide a valid email. **Status**: Pass
4. **Action**: User enters valid credentials and clicks "Login". **Expected Result**: Successfully logs in. **Status**: Pass

5. **Action**: User enters incorrect credentials and clicks "Login". **Expected Result**: Displays error for incorrect credentials. **Status**: Pass

4.8.2 Test Case 02: Plants Search

Test Case #: 02
Test Case Name: Plants Search
Subsystem: Search Functionality
Designed By: Hameem Md Sayem
Design Date: 2024-11-27
Executed By: Hameem Md Sayem
Execution Date: 2024-11-27

Test Steps & Expected Results:

1. **Action**: User enters a plant name in the search bar and clicks "Search". **Expected Result**: Displays relevant results matching the plant name. **Status**: Pass
2. **Action**: User filters search results by price range. **Expected Result**: Filters results based on the selected price range. **Status**: Pass
3. **Action**: User filters search results by plant type (e.g., indoor, outdoor). **Expected Result**: Filters results based on plant type. **Status**: Pass
4. **Action**: User searches without entering any keywords. **Expected Result**: Displays message: "Please enter a search term." **Status**: Pass
5. **Action**: User searches using an invalid plant name (e.g., non-existent plant). **Expected Result**: Displays message: "No matching plants found." **Status**: Pass

4.8.3 Test Case 03: Create Shop

Test Case #:	03
Test Case Name:	Create Shop
Subsystem:	User Profile/Shop Management
Designed By:	Hameem Md Sayem
Design Date:	2024-11-27
Executed By:	Hameem Md Sayem

Execution Date: 2024-11-27

Test Steps & Expected Results:

1. **Action**: User clicks "Create Shop" and enters the shop name and location. **Expected Result**: Shop is created successfully and displayed in the dashboard. **Status**: Pass
2. **Action**: User leaves the shop name field blank and clicks "Create Shop". **Expected Result**: Displays error: "Shop name is required." **Status**: Pass
3. **Action**: User enters a shop name with special characters and clicks "Create Shop". **Expected Result**: Displays error: "Invalid characters in shop name." **Status**: Pass
4. **Action**: User fills in all required fields and clicks "Create Shop". **Expected Result**: Shop is successfully created, and a confirmation message appears. **Status**: Pass
5. **Action**: User tries to create a shop without uploading a logo. **Expected Result**: Displays message: "Logo is optional, proceed without it." **Status**: Pass

CHAPTERS 5: CONCLUSION AND FUTURE WORK

5.1 Conclusion

The Baganbari project successfully addresses the challenges faced by traditional plant retail systems by offering a scalable, secure, and user-friendly platform for both nursery owners and customers. Leveraging advanced technologies such as React.js for the frontend, Node.js for the backend, and MongoDB for database management, the platform ensures high performance, responsiveness, and reliability. The intuitive interface enhances accessibility for users with varying technical skills, simplifying tasks like browsing plants, placing orders, and managing inventory.

A key highlight of the project is its efficient matching algorithm, which helps customers find suitable plants based on specific criteria, such as type, price, and availability. This reduces search times and maximizes user satisfaction by providing relevant results quickly. Security is also a cornerstone of the platform, with robust features such as data encryption, secure authentication using JWT, and role-based access control (RBAC), ensuring that user data is protected and only authorized individuals have access to sensitive functionalities.

In summary, Baganbari provides a meaningful solution to the plant retail market, creating a reliable platform that streamlines purchasing and inventory management. It establishes a solid foundation for future growth, including potential feature expansions like AI-driven recommendations and international expansion, ultimately setting a new standard for digital plant retailing.

5.2 Future Scope

The future roadmap for Baganbari includes several exciting opportunities for growth and enhancement:

1. **Mobile Application Development**: Native applications for iOS and Android will be developed to increase accessibility and provide a more personalized experience for mobile users.

2. **AI Integration**: Implement AI-powered features to predict customer preferences, recommend plants based on historical data and trends, and optimize inventory management.
3. **Real-Time Chat**: Introduce a real-time messaging system to facilitate direct communication between nursery owners and customers, improving customer service and order management.
4. **Global Expansion**: The platform will support multiple languages and expand its service network internationally, making it accessible to a wider global audience and enabling cross-border plant retailing.

5.3 Cost Management

Category	Cost Item	Estimated Monthly Cost (BDT)	Description
Development Phase	Developer Salary	30,000 -50,000	For developers who created the platform during the development phase (one-time expense).
	Design and UI Tools	4,500 - 9,999	Tools like Figma, Adobe XD for creating the design.
	Testing Tools	1,500 - 3,000	For tools like BrowserStack, or any paid testing tools.

Domain & Hosting	Domain Name Registration	1,000 - 3,000	Annual domain cost, spread over months (e.g. www.baganbari.com).
	Web Hosting	4,500 - 15,000	Hosting on providers like AWS or Vercel for backend and frontend hosting.
	CDN (Content Delivery Network)	1,000 - 2,000	To speed up loading times (e.g. Cloudflare).
Database	Database Service	2,000 - 9,999	Managed database services like MongoDB Atlas.
Website Maintenance	Administrative Costs	5,000 - 12,500	Managing user queries, resolving bugs, and updating content.
	Website Monitoring	1,000 - 2,500	Tools like Pingdom or UptimeRobot to ensure uptime.
	SSL Certificate	500 - 1,000	SSL encryption for website security.
Third-Party Services	Email Services	1,000 - 5,000	Services like SendGrid or Mailgun for transactional emails.

	Marketing/SEO Tools	5,000 - 16,000	Tools like Google Ads, Facebook Ads, or SEMrush for marketing campaigns and website promotion.
Maintenance Team	Developer/Technical Support	20,000 - 30,000	For bug fixes, updates, and maintaining the site.
	Content Manager/Copywriter	10,000 - 30,000	To manage blog content, service descriptions, and customer support articles.
	Customer Support	10,000 - 40,000	Dedicated customer support personnel to address user queries.
Miscellaneous Costs	Backup Services	1,000 - 3,000	Automated backup solutions like AWS S3 or Backblaze.
	Updates and Feature Enhancements	10,000 - 30,000	Continuous improvement of the platform.

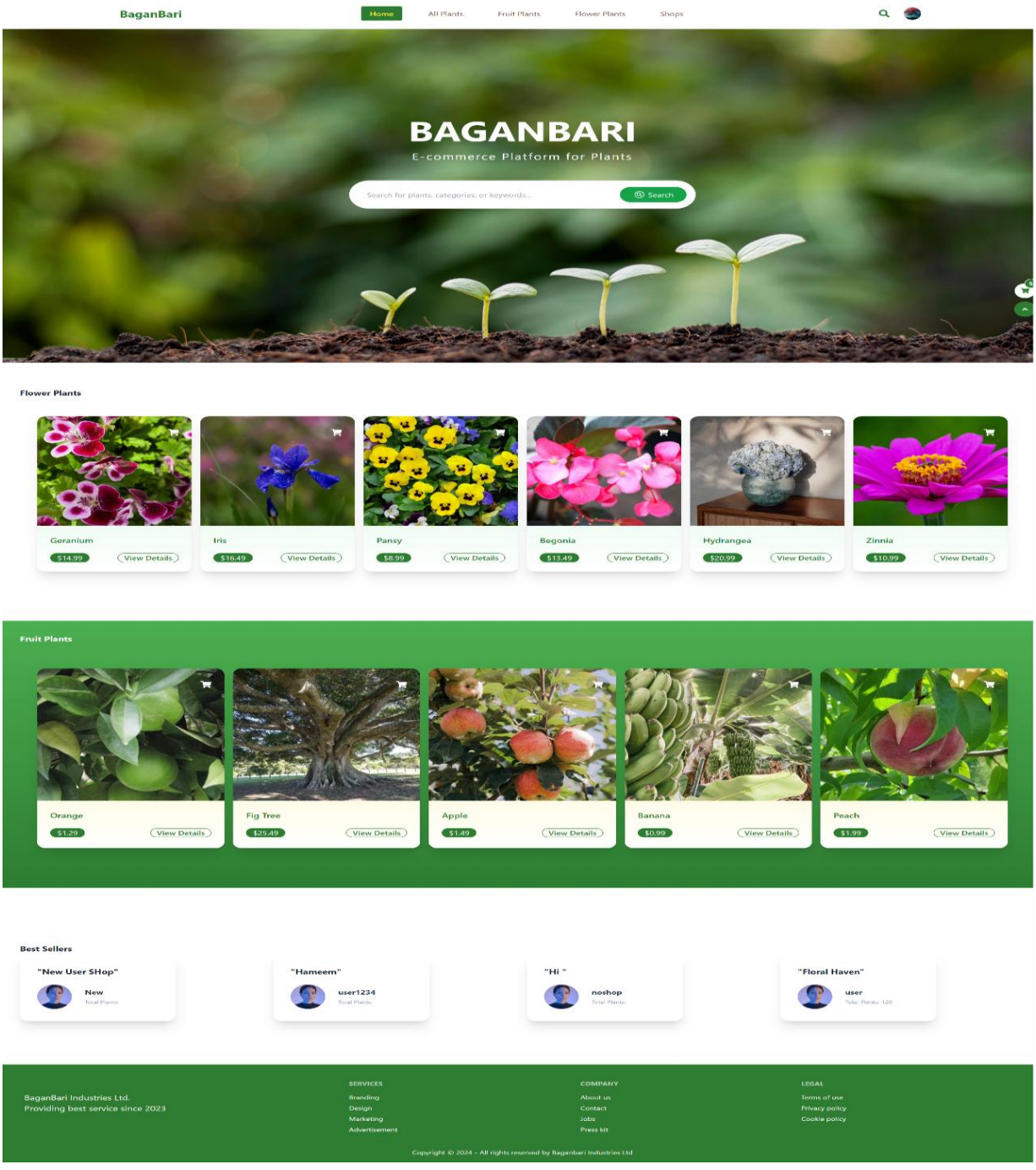
Approximate Monthly Costs (Published Website)

Category	**Low Range (BDT)**	**High Range (BDT)**
Hosting, Domain, CDN	5,000	20,000
Database	2,000	10,000
Maintenance (Admin, Monitoring)	5,000	20,000
Third-Party Services	5,000	15,000
Team (Support, Content, Tech)	15,000	25,000
Total Estimated Costs	32,000	90,000

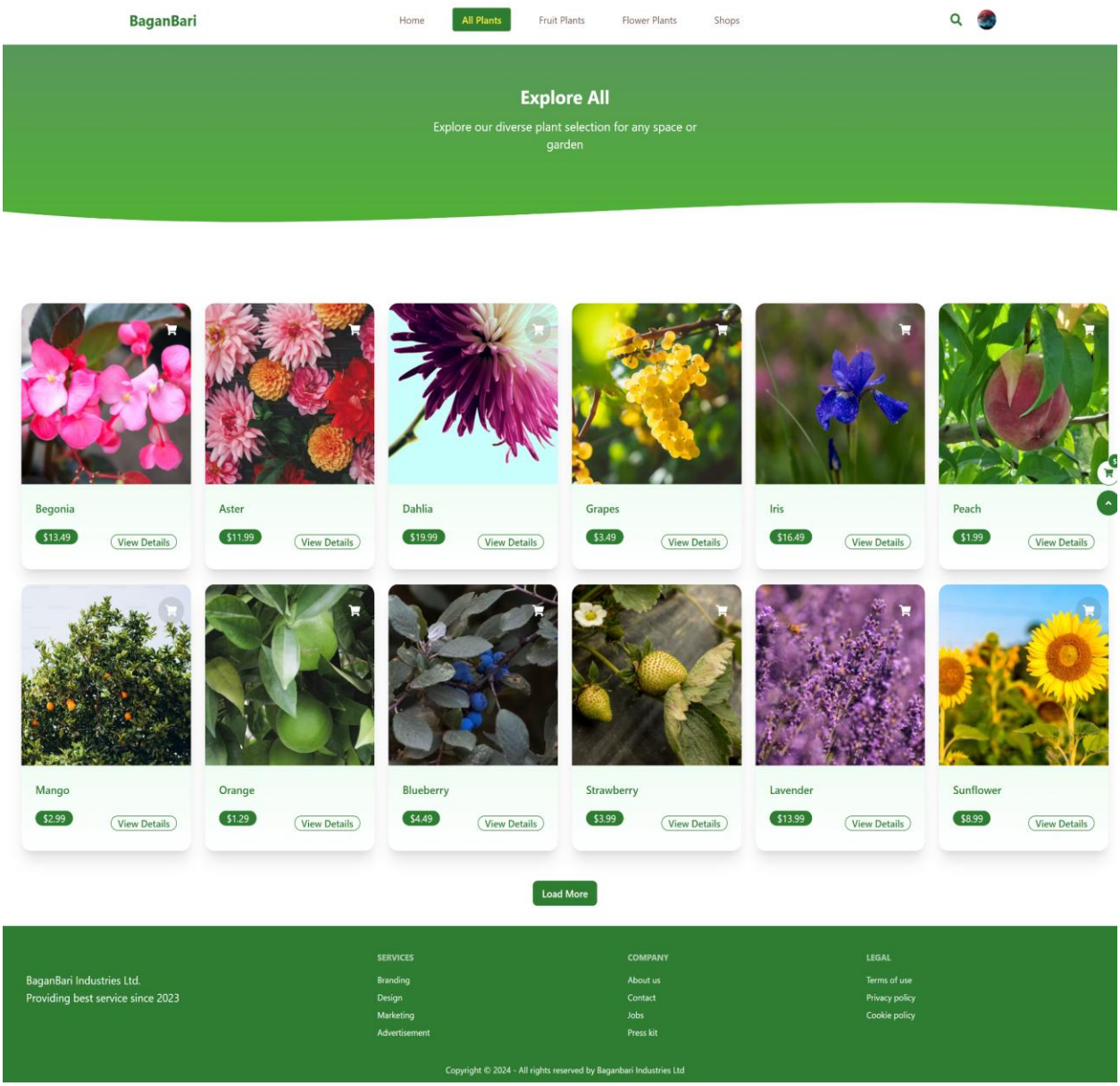
CHAPTER 6: APPENDIX

6.1 System Design

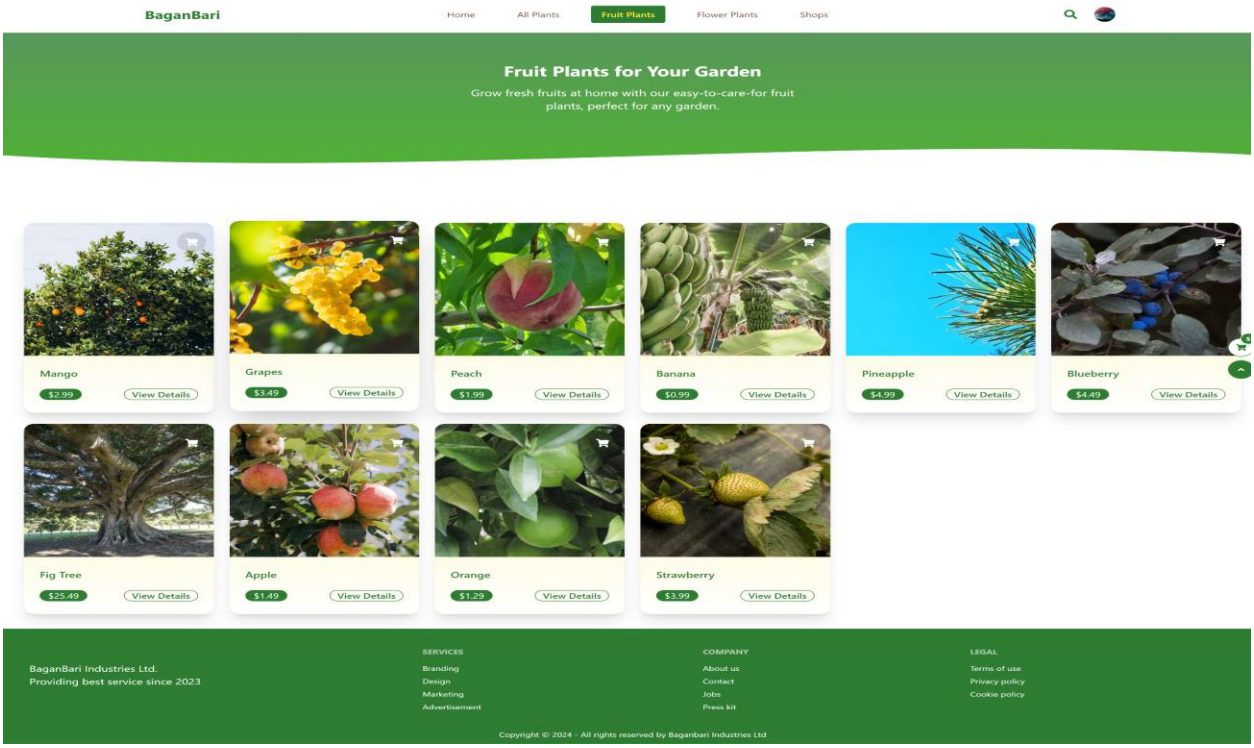
Homepage - User can search and browse plants



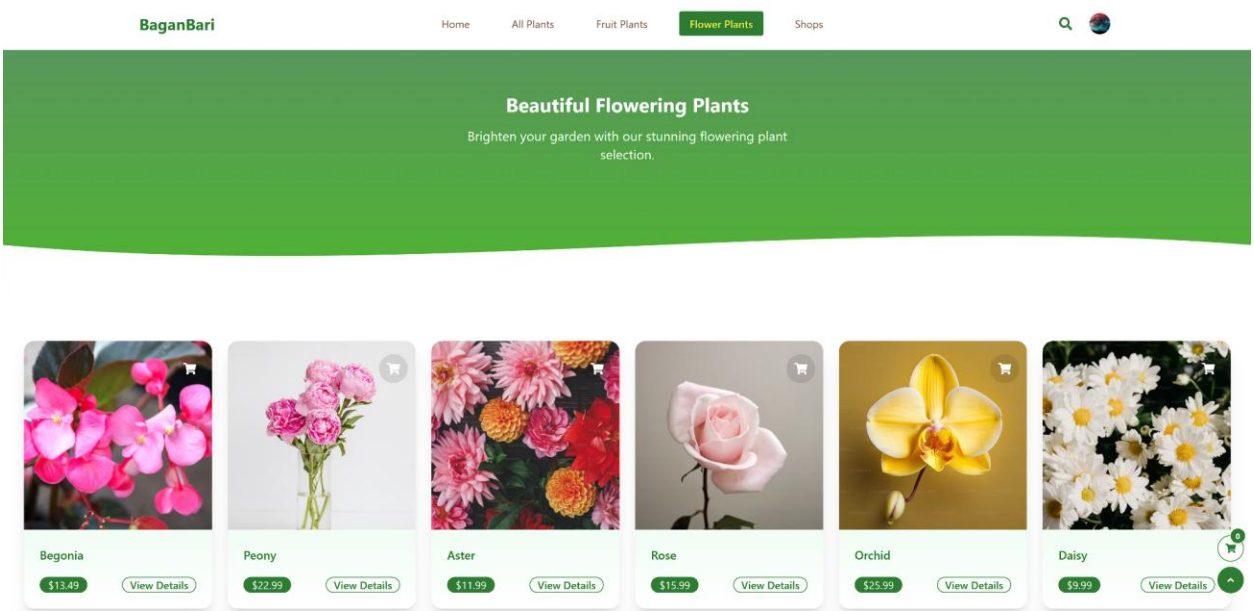
All Plants - User can explore all the plants



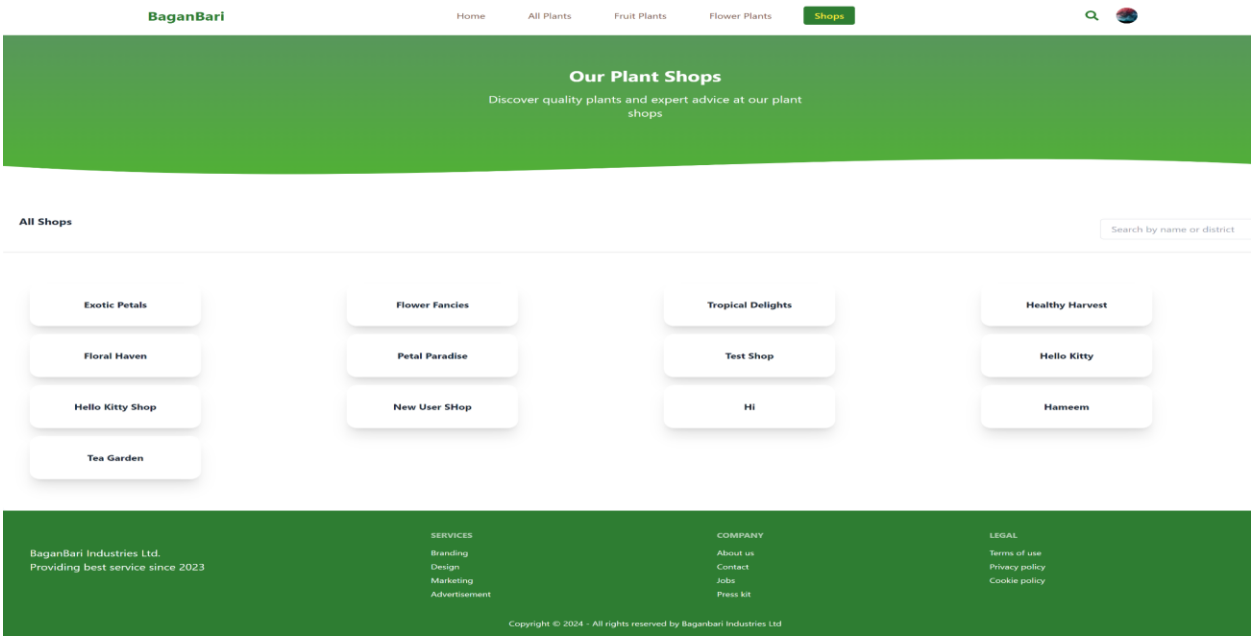
Fruit Plants - User can explore all the fruit plants



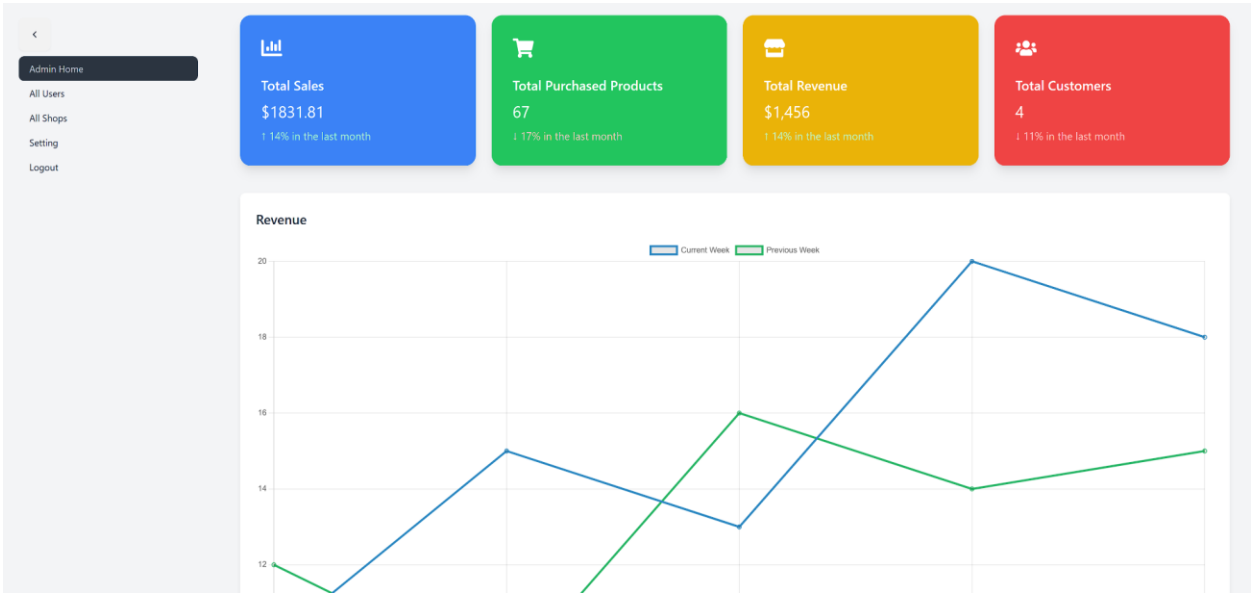
Flowers Plants - User can explore all the flowers plants



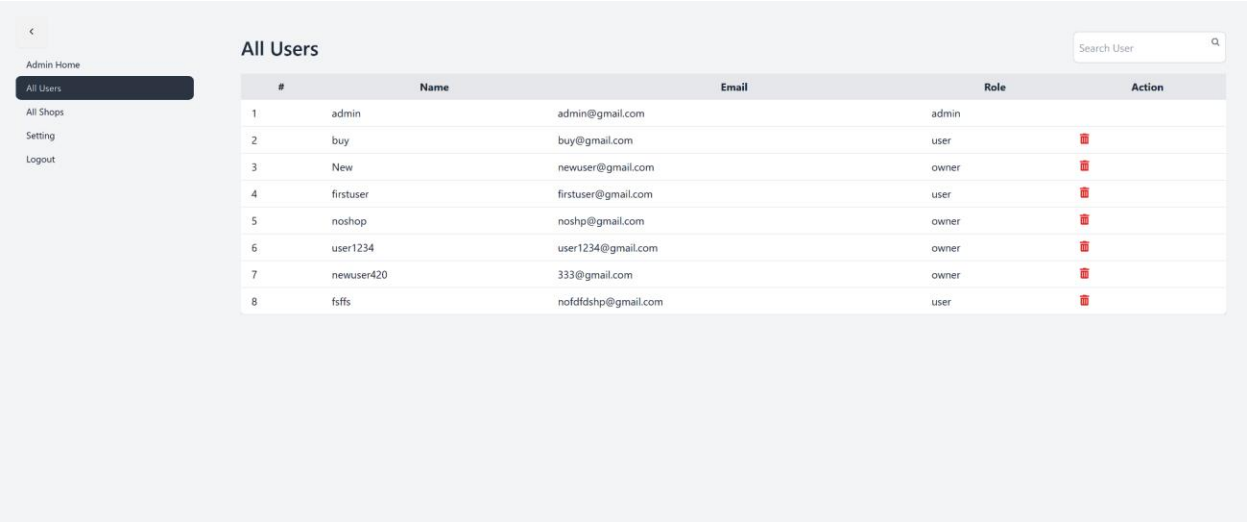
All Shops - User can explore all the shops



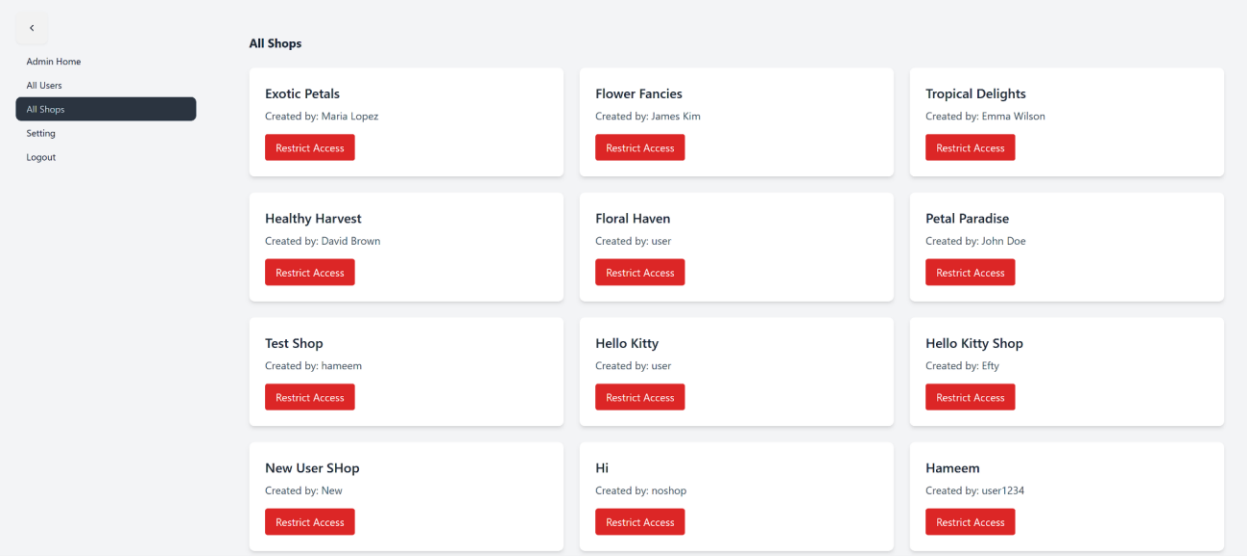
Admin Dashboard (Home) - Admin can view sales report



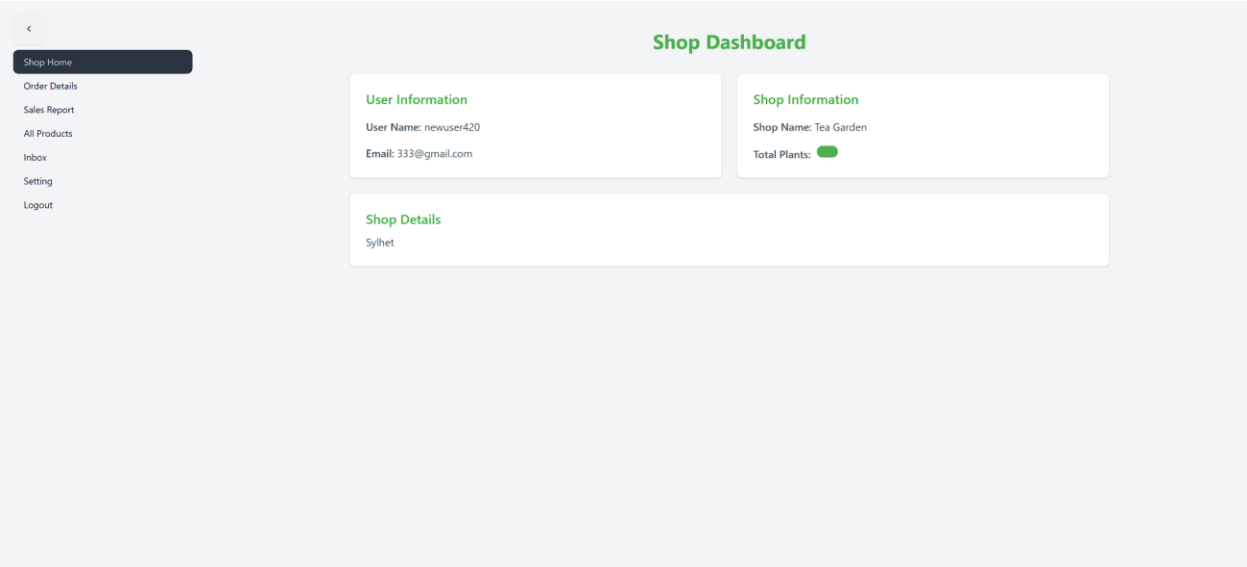
Admin Dashboard (All Users) - Admin can view, search, delete users



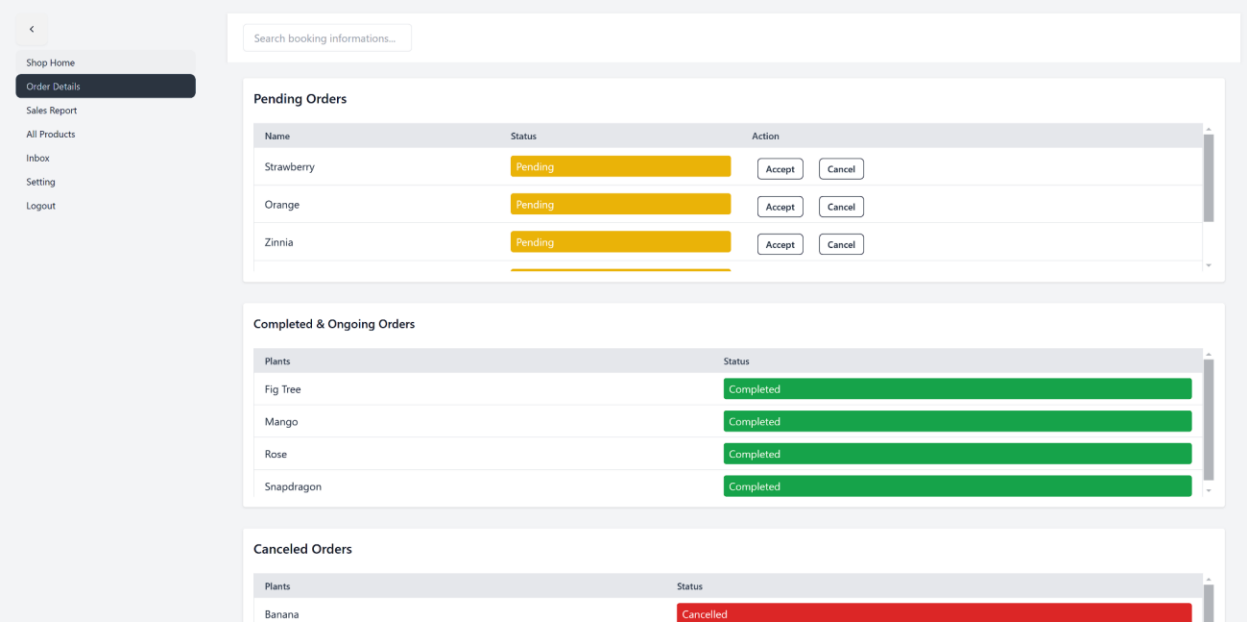
Admin Dashboard (All Shops) - Admin can access or restrict shop



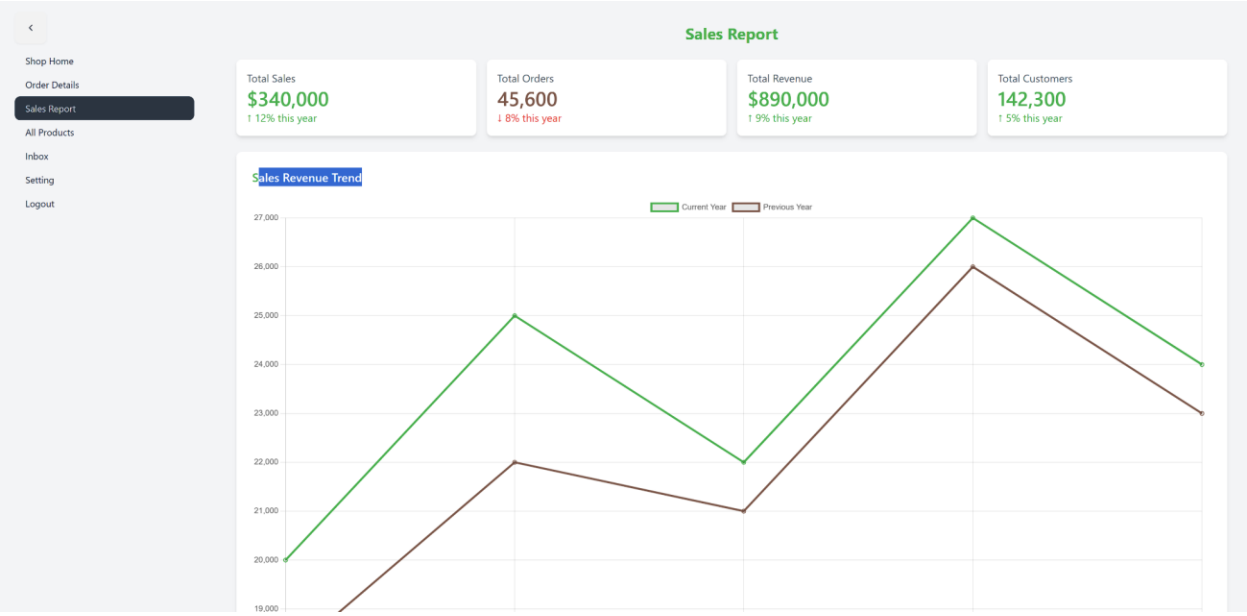
Shop Owners Dashboard (Home) - Owners can view shop details



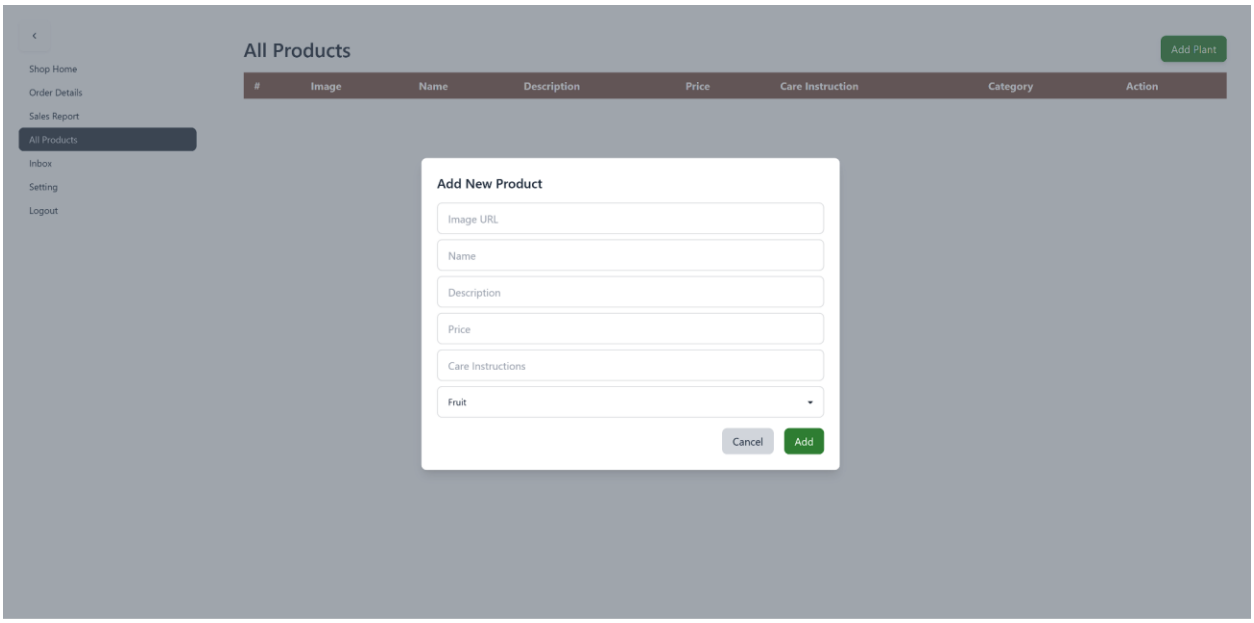
Shop Owners Dashboard (Order Details) - Owners can view, accept, cancel orders



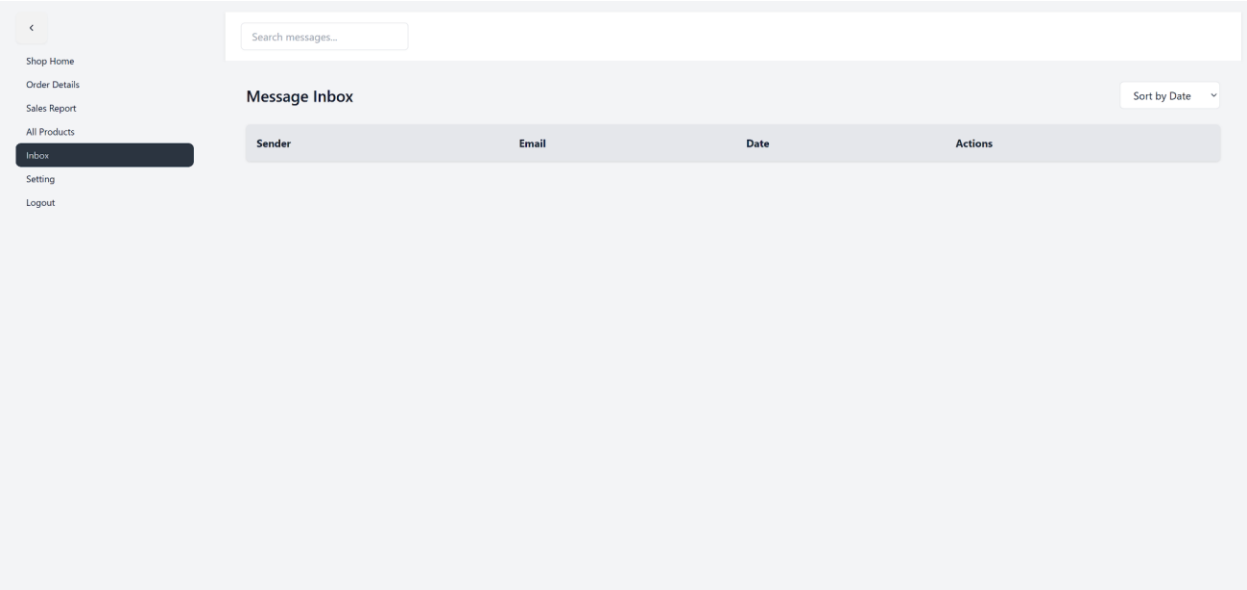
Shop Owners Dashboard (Sales Report) - Owners can view full sales report



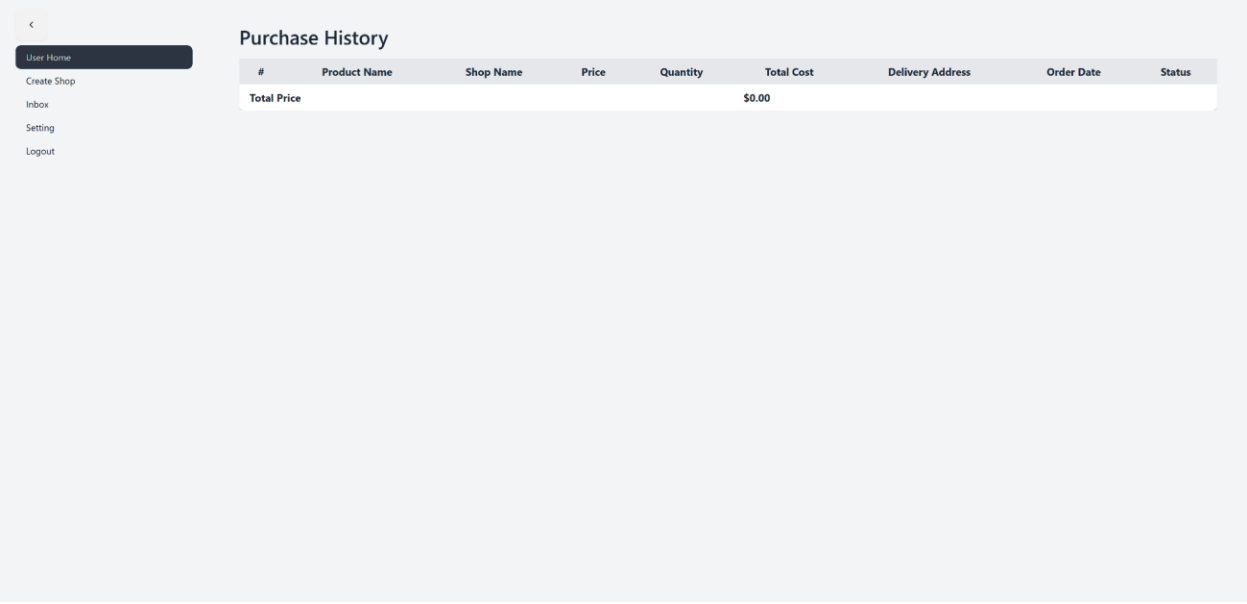
Shop Owners Dashboard (All Products) - Owners can view, add and edit plants details



Shop Owners Dashboard (Inbox) - Owners can view and reply to messages

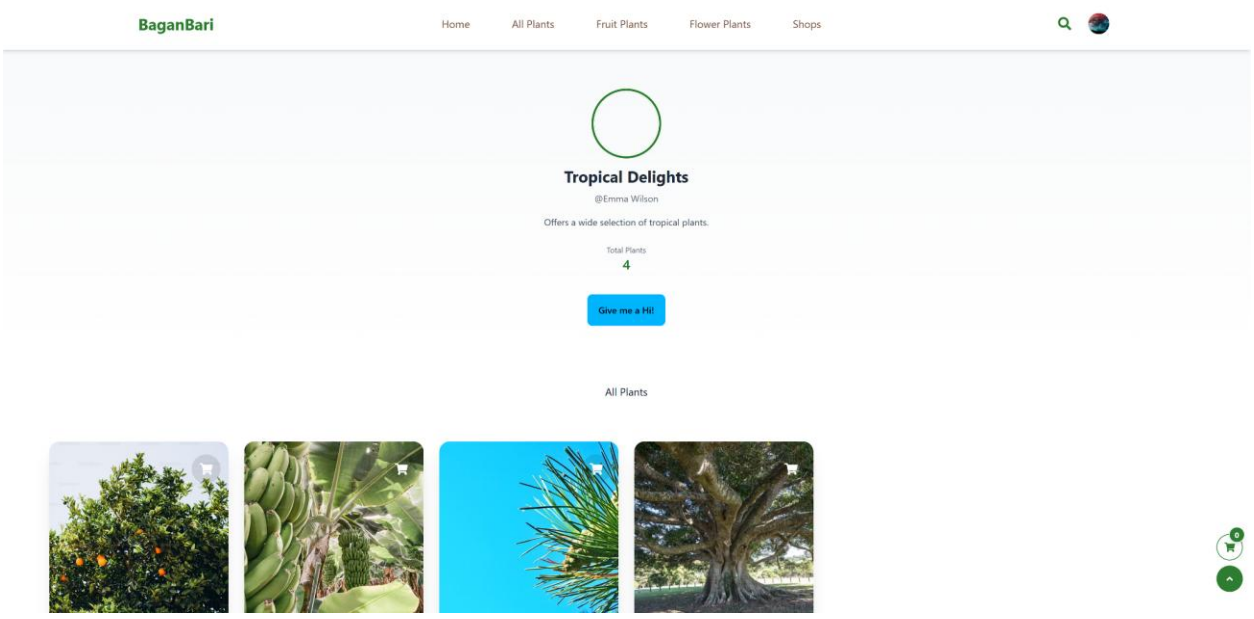


Users Dashboard (Home) - Users can view all purchase history

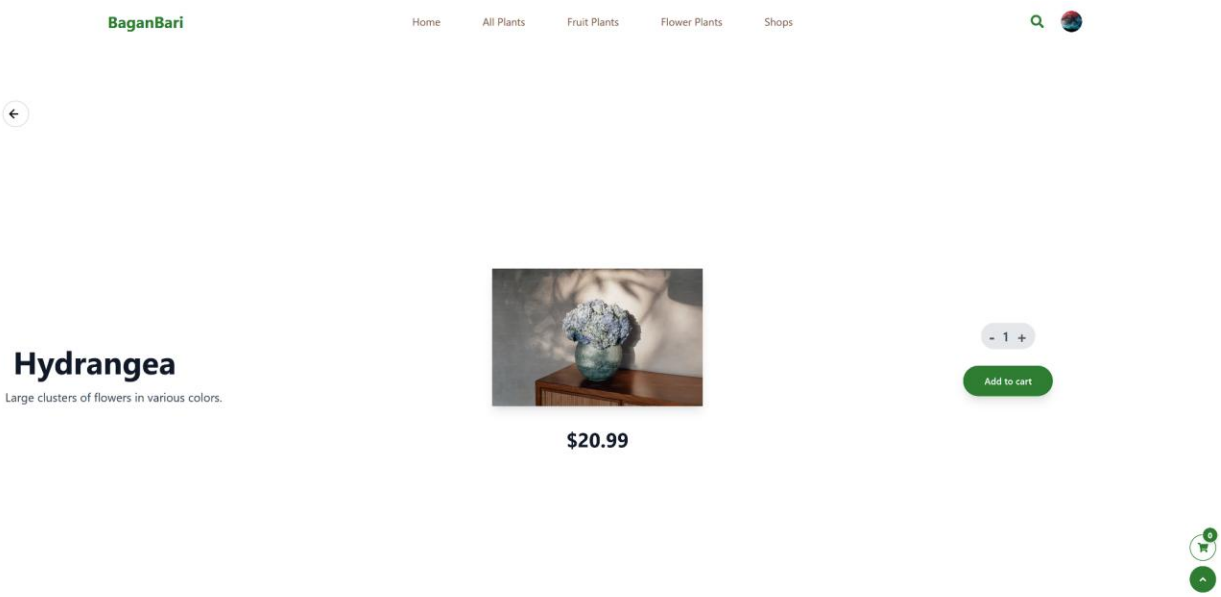


Users Dashboard (Create Shop) - Users can create shop

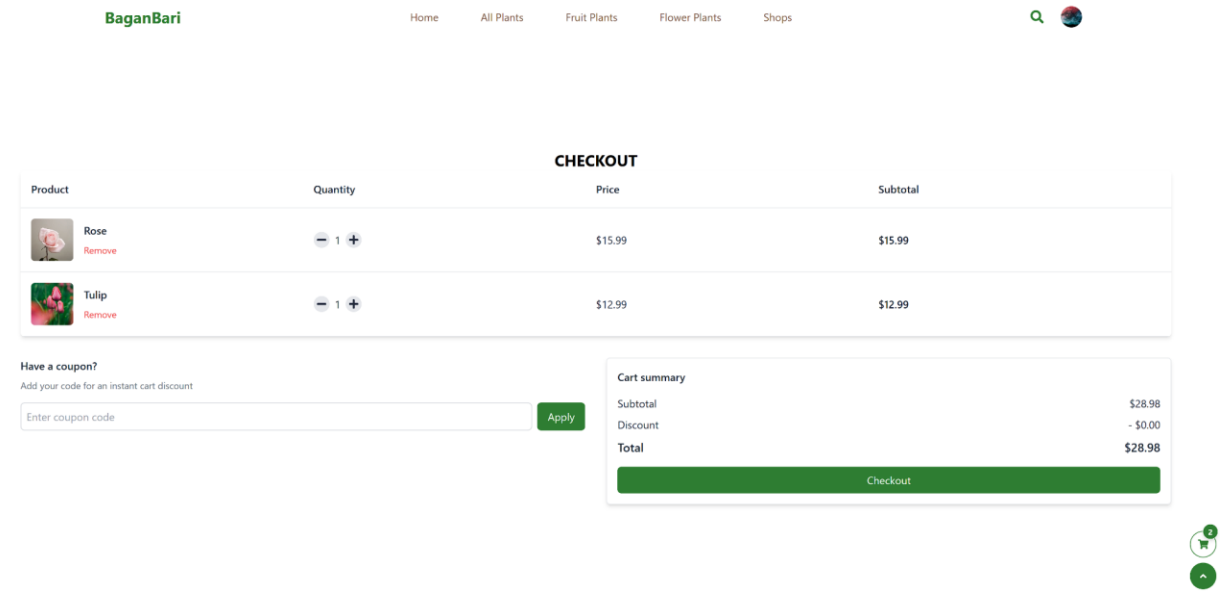
Shop Details Page - Users can message, find shop details



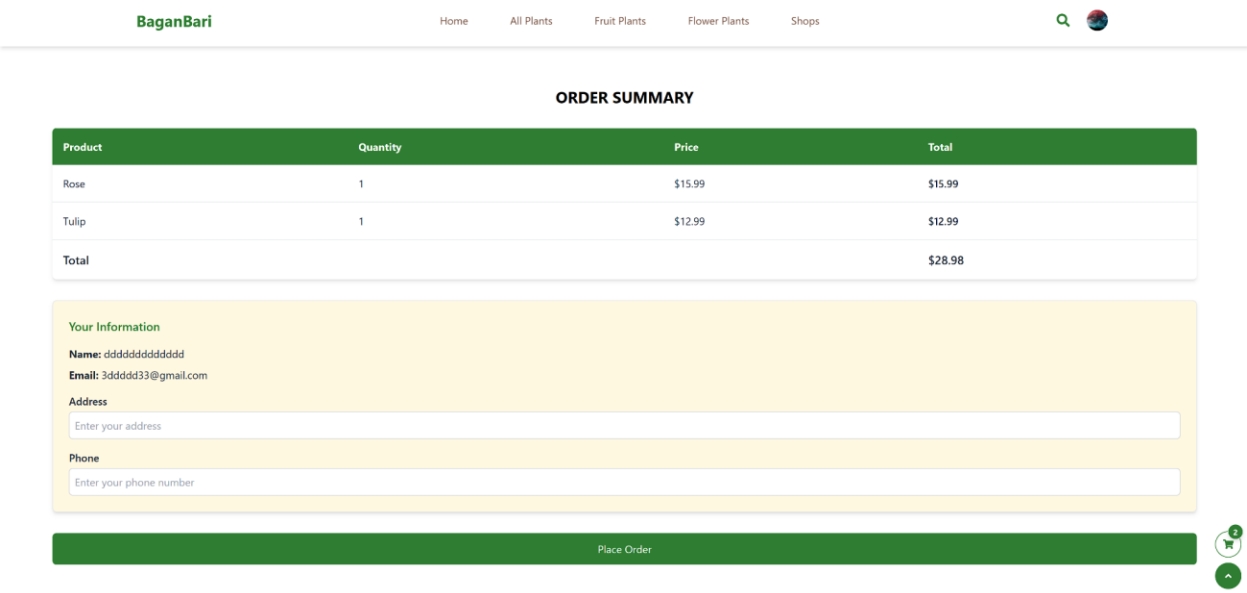
Plants Page - Users can view products details and add to cart



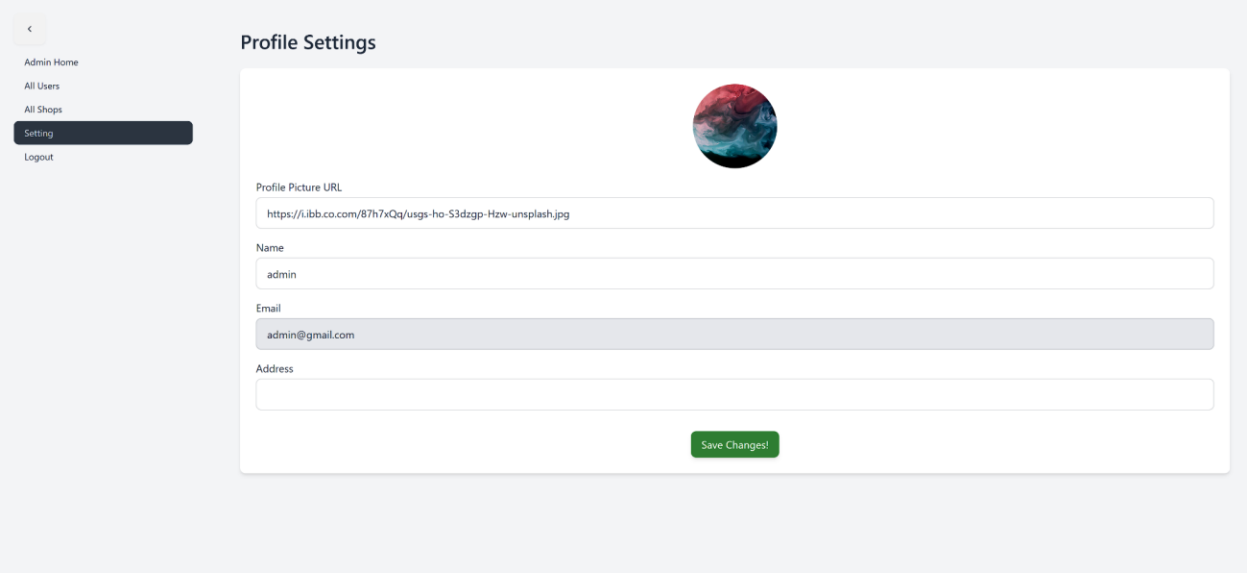
Checkout Page - Users can checkout and proceed



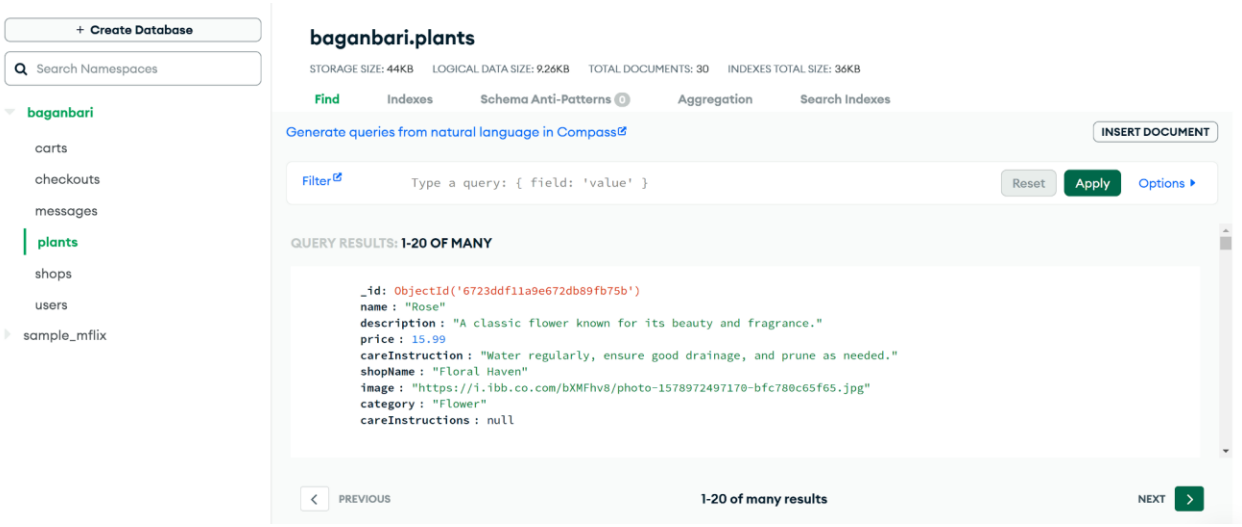
Order Page - Users can input address and phone to confirm the order



Setting Page - Users can change profile pictures, name, addresses



6.2 Database Design



CHAPTER 7: REFERENCES

1. **MongoDB Documentation** – MongoDB, Inc. https://www.mongodb.com/docs/
2. **React Official Documentation** – ReactJS https://reactjs.org/
3. **Express.js Documentation** – ExpressJS https://expressjs.com/
4. **Tailwind CSS Guide** – Tailwind Labs https://tailwindcss.com/
5. **Stack Overflow** – Community-driven Q&A platform for developers https://stackoverflow.com/
6. **YouTube** – Educational and tutorial videos for programming and development https://www.youtube.com/
7. **Google** – Search engine used for gathering general information and tutorials https://www.google.com/
8. **ChatGPT** – AI-powered assistant for research, learning, and problem-solving https://chat.openai.com/

Tree and plants

ORIGINALITY REPORT

9%	5%	2%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
2	Submitted to IIT Delhi Student Paper	1%
3	Submitted to Universitas Muria Kudus Student Paper	<1%
4	Submitted to Liverpool John Moores University Student Paper	<1%
5	123dok.com Internet Source	<1%
6	Submitted to Arab Open University Student Paper	<1%
7	Submitted to Lead College Pty Ltd Student Paper	<1%
8	Submitted to Staffordshire University Student Paper	<1%
9	expertbeacon.com Internet Source	<1%