A PROJECT REPORT

on

PlantIT

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
Computer Science and Engineering

BY

Siddhartha Mukherjee	21052365
Rishav Das	21052347
Dipankar Khanra	2105963

UNDER THE GUIDANCE OF
Prof. Dr. Prachet Bhuyan

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024

April 2024

A PROJECT REPORT

on

PlantIT

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN

Computer Science and Engineering

BY

Siddhartha Mukherjee	21052365
Rishav Das	21052347
Dipankar Khanra	2105963

UNDER THE GUIDANCE OF

Prof. Dr. Prachet Bhuyan

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAE, ODISHA -751024

April 2024

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is to certify that the project entitled

PlantIT

submitted by

Siddhartha Mukherjee	21052365
Rishav Das	21052347
Dipankar Khanra	2105963

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2024, under our guidance.

Date: 09 / 04 / 2024

(Guide Name)
Prof. Dr. Prachet Buyan

Acknowledgements

We are profoundly grateful to **Prof. Dr. Prachet Bhuyan** of **KIIT Deemed to be University** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

Siddhartha Mukherjee

Rishav Das

Dipankar Khanra

ABSTRACT

The rapid advancement of technology has revolutionized various aspects of our lives, including the way we shop for plants and engage in gardening activities. In response to the growing demand for convenient and reliable online plant shopping platforms, this project presents the development of PlantIT, a comprehensive mobile application aimed at facilitating seamless plant shopping and gardening assistance. Leveraging the power of React Native, the application provides e-commerce facilities, allowing users to browse, purchase, and sell plants with ease. Furthermore, the integration of AI-based plant recommendation and care advice systems enhances the user experience by offering personalized plant suggestions and expert guidance for plant care. Community integration features, such as blogging, foster collaboration and knowledge sharing among plant enthusiasts. By adhering to established design and coding standards, and following rigorous testing protocols, PlantIT aims to deliver a robust and user-friendly solution that meets the needs of both novice and experienced gardeners. The project's future scope includes enhancements in AI capabilities, geolocation-based services, augmented reality integration, and expansion of community features, paving the way for continued innovation in the field of online plant shopping and gardening assistance.

Keywords:
PlantIT, E-Commerce, Gemini-Pro-Vision, React-Native, Firebase, NodeJS, Plant-Recommendation, AI-Based-Plant-Care

Contents

List of Figures

Chapter 1

Introduction

In the ever-evolving landscape of modern society, the importance of greenery and sustainable living has witnessed a remarkable surge, accentuating the pressing need for accessible platforms that facilitate the seamless buying and selling of plants. Recognizing this demand, our dedicated team has meticulously developed "PlantIT," a bespoke online marketplace tailored explicitly for plant enthusiasts. Existing solutions in the market often fall short, failing to provide user-friendly interfaces, diverse plant selections, and reliable service, leaving consumers dissatisfied and yearning for a more comprehensive solution.

PlantIT is a endeavour, aimed at bridging these gaps by offering a seamless digital marketplace where users can effortlessly explore a wide array of plants, connect with a vibrant community of like-minded plant lovers, and benefit from AI-based expert plant care advice, coupled with personalized plant recommendations. The platform places paramount importance on user experience, boasting an intuitive navigation system, detailed plant disease remedies, and tailored plant suggestions catering to individual preferences and needs.

This comprehensive project report will delve into the intricate development journey of PlantIT, highlighting its pivotal features, technological considerations, and the underlying business model that underpins its success. Furthermore, it will unveil the findings of our extensive market research, shedding light on user needs and preferences, and the meticulously crafted strategies employed to effectively address them. PlantIT not only fills a critical market gap but also reflects our unwavering dedication to fostering sustainable lifestyles and environmental consciousness, aligning with the global movement towards a greener and more sustainable future.

Through its innovative approach and user-centric design, PlantIT aims to revolutionize the plant enthusiast community, empowering individuals to embrace their passion for greenery while contributing to an eco-friendlier world. With its comprehensive features and commitment to excellence, PlantIT is poised to become the go-to destination for plant lovers, nurturing a thriving ecosystem of greenery and sustainability.

Chapter 2

Basic Concepts

This section serves as a foundational pillar, encompassing the fundamental concepts and essential knowledge that are imperative for a comprehensive understanding of the project. It delves into the intricate details of the related tools and techniques employed throughout the development process, shedding light on their significance and the pivotal roles they play in shaping the final outcome of the project.

2.1 React Native:

React Native serves as a versatile framework for crafting mobile applications compatible across multiple platforms, utilizing the power of JavaScript and React. Within our project, we harnessed React Native's capabilities to construct a dynamic mobile application tailored for plant shopping, ensuring a seamless and uniform user experience specifically optimized for Android devices.

2.2 Firebase:

Firebase, a robust platform offered by Google, serves as a comprehensive solution for developing both web and mobile applications. Within our project, we heavily relied on Firebase services, leveraging its capabilities for authentication, real-time data storage, and database functionalities.

2.2.1 Firebase Authentication:

Used for user authentication in the mobile application, allowing users to sign in securely using Google OAuth.

2.2.2 Firebase Realtime Database:

Employed for storing real-time data such as user profiles, product information, and blog posts, enabling seamless synchronization between the mobile application and backend services.

2.2.3 Firebase Storage:

Utilized to store and serve user-uploaded images, such as plant images for product, images for blogs and other services, ensuring scalable and secure cloud storage.

2.3 Node.js and Express:

Node.js functions as a JavaScript runtime environment tailored for server-side development, whereas Express.js stands out as a streamlined web framework ideal for crafting web applications and APIs. Within our project scope, we employed Node.js and Express to architect the backend services that underpin both the mobile application and the seller console website.

2.4 Google Gemini Pro:

Google Gemini Pro API is a cloud-based natural language processing service. We used its text classification capabilities to provide personalized plant recommendations based on user input about preferences and conditions.

2.5 Google Gemini Pro Vision API:

Google Gemini Pro Vision API is a machine learning-based image recognition API provided by Google. In our project, Google Gemini Pro Vision API was integrated into the backend services to provide AI-based plant care advice. The API was used to analyse images uploaded by users to identify plant diseases, pests, or other issues, and provide appropriate care advice and remedies within the mobile application.

2.6 React.js:

React.js is a JavaScript library for building user interfaces, commonly used for developing web applications. In our project, React.js was used to create the seller console website, providing sellers with a user-friendly interface for uploading their plant listings and find their current orders.

Chapter-3

Problem Statement

Many individuals interested in purchasing plants encounter difficulties in finding a convenient and reliable online platform. Existing platforms lack user-friendly interfaces, and personalized recommendation system and an online community of like-minded nature enthusiasts. This project aims to address these issues by developing a mobile application with e-commerce functionality, AI-based personalized plant recommendations, plant care advice, and a blogging community to engage like-minded buyers together.

Project Planning

1. **Requirements Gathering:**

The objective of this project is to develop a mobile application that impacts the way individuals shop for plants online. The application aims to address the existing challenges faced by plant enthusiasts by providing a convenient, reliable, and user-friendly platform for purchasing plants.

Key objectives of the project include:

1. **E-commerce Functionality**: Develop an intuitive and secure mobile application with e-commerce capabilities, allowing users to browse, search, and purchase a wide variety of plants from trusted sellers.
2. **AI-based Personalized Recommendations**: Implement an AI-powered recommendation system that analyses user preferences, environmental conditions, and plant characteristics to provide personalized plant recommendations tailored to each user's needs.
3. **Plant Care Advice**: Integrate AI-based plant care advice functionality, enabling users to receive guidance on plant health and maintenance. Users can upload images of diseased leaf, and the system will analyse the images to identify plant diseases and provide appropriate remedies.
4. **Community Engagement**: Create a blog feature within the application to facilitate community engagement and social integration among plant enthusiasts. Users can share their gardening experiences, tips, and advice, fostering a vibrant and supportive community of planters.

2. **Technology Feasibility Assessment:**

    1. **Mobile Application Development:**
    React Native: Utilizing React Native for mobile app development provides cross-platform compatibility with native-like performance making it a suitable choice for our project.

    2. **Backend Development:**
    Node.js: Node.js is a lightweight and efficient runtime environment for server-side applications. It's well-suited for building scalable and high-performance backend services, making it a viable option for the backend of our mobile application.

    Express.js: Express.js is a minimalist web framework for Node.js, providing a robust set of features for building RESTful APIs and handling HTTP requests. It can be used to develop the backend API endpoints required for your application.

    3. **Database:**
    Firebase Realtime Database: Firebase Realtime Database is well-suited for storing real-time data such as user profiles, product information, and blog posts in your application. Its ease of use and integration with other Firebase services make it a convenient choice for our project.

    4. **AI Services:**
    Google Gemini Vision Pro: Google Gemini Vision Pro API is suitable for implementing AI-based plant care advice functionality in your application. By analysing images uploaded by users, the API can identify plant diseases, pests, or other issues, and provide appropriate recommendations for treatment or care. Its integration with Google Cloud Platform ensures reliable performance and scalability for your application's image recognition needs.

3. **Feature Specification and Acceptance Criteria:**

    1. **E-commerce Functionality:**
    Feature Specification:
        - Users can browse a catalogue of plants with detailed descriptions, images, and prices.
        - Users can add plants to their shopping cart and proceed to checkout for payment.

- Sellers can list their plants for sale, including uploading images and specifying prices and description.

Acceptance Criteria:

- Users can successfully view, search, and add plants to their shopping cart.
- Users can complete the checkout process.
- Sellers can successfully list plants for sale, and the listings are displayed correctly to users.

2. **AI-based Personalized Recommendations:**

Feature Specification:

- Users can provide input on their preferences, such as plant type, size, and preferred environmental conditions.
- The AI system analyses user preferences and recommends suitable plants based on the input provided.

Acceptance Criteria:

- Users can input their preferences through an intuitive interface.
- The AI system generates accurate and relevant plant recommendations based on user input.
- Users find the recommended plants to be suitable and aligned with their preferences.

3. **Plant Care Advice:**

Feature Specification:

- Users can upload images of leaves exhibiting symptoms of diseases, pests, or other issues.
- The AI system analyses the images and identifies the underlying problems, providing detailed plant care advice and remedies.

Acceptance Criteria:

- The AI system accurately identifies plant issues from uploaded images and provides appropriate care advice.
- Users can easily access plant care tips and guidelines within the application.

4. **Community engagement:**

Feature Specification:

- Users can browse and read blog posts on various topics related to gardening, plant care, and horticulture.
- Users can contribute their own blog posts, share gardening experiences, tips, and advice with the community.

Acceptance Criteria:

- Users can view a curated collection of informative and engaging blog posts within the application.
- Users can submit their own blog posts and have them published.

3.2 Project Analysis

1. **Feature Analysis**:

- **E-commerce Functionality**:
  - Feature Importance: High
  - Complexity: Moderate to High (Integration with payment gateway, inventory management)
  - Dependencies: Product display, user authentication, payment processing

- **AI-based Personalized Recommendations**:
  - Feature Importance: Medium to High
  - Complexity: High (Integration with AI service, user preference analysis)
  - Dependencies: User preferences input

- **Plant Care Advice**:
  - Feature Importance: Medium
  - Complexity: High (Integration with AI service for image analysis, database of plant care information)
  - Dependencies: User authentication, image upload functionality

- **Community Engagement:**
  - Feature Importance: Medium
  - Complexity: Moderate (Integration with custom backend)
  - Dependencies: User authentication, content upload system.

2. **Technical Analysis**:

- **Mobile Application Development**:
  Utilizing React Native for cross-platform mobile app development ensures code reusability and native-like performance.

- **Backend Development**:
  Firebase Realtime Database and Authentication provide scalable and real-time data storage and user authentication services.

- **AI Integration**:
  Google Gemini Pro Vision API offers advanced image recognition capabilities for AI-based plant care advice.

- **Community Features**:
  Integration with a CMS or custom backend for the blog feature enables easy content management and user engagement.

3. **Risk Analysis**:

- **Technical Risks**:
  - Dependency on third-party services (Firebase, Google Gemini Pro Vision API) may lead to service disruptions or changes in API functionality.
  - Integration complexity between frontend (React Native) and backend (Firebase) components may lead to compatibility issues.
- **Security Risks**:
  - Potential security vulnerabilities in user authentication and data storage mechanisms (Firebase Authentication, Realtime Database) may lead to data breaches or unauthorized access.

3.3 System Design:

3.3.1 Design Constraints:

- **Performance**: Design the application to be fast and responsive, even on devices with limited resources.
- **Scalability**: Design a scalable architecture that can accommodate future growth in user base and features.
- **Security**: Implement robust security measures to protect user data, including encryption, authentication, and authorization mechanisms.

3.3.2 System Architecture:

1. **Client-Side (Mobile Application)**:
   - **Technology**: React Native
   - **Description**: The client-side of the application is developed using React Native framework, which allows for cross-platform development of mobile applications for both iOS and Android platforms. React Native provides a native-like user experience and facilitates code reusability across different platforms.

2. **Authentication**:
   - **Technology**: Firebase OAuth with Google

- **Description**: Firebase Authentication with Google OAuth is used for user authentication and authorization. Users can sign in to the application using their Google accounts, providing a seamless and secure authentication experience.

3. **Realtime Database**:
   - **Technology**: Firebase Realtime Database
   - **Description**: Firebase Realtime Database is utilized for storing real-time data, such as user profiles, product information, and blog posts. It offers real-time synchronization, ensuring that data is always up-to-date and accessible to users.

4. **Backend Services**:
   - **Technology**: Node.js, Express
   - **Description**: Node.js and Express.js are used to develop the backend services of the application. Node.js provides a runtime environment for server-side JavaScript code, while Express.js is a web framework for building RESTful APIs and handling HTTP requests. The backend services handle business logic, data processing, and communication with external APIs.

5. **AI Integration**:
   - **Technology**: Google Gemini Pro Vision API
   - **Description**: Google Gemini Pro Vision API is integrated into the backend services for AI-based plant care advice functionality. It provides advanced image recognition capabilities, allowing the application to analyse images of plants uploaded by users and identify plant diseases, pests, or other issues.

6. **Storage**:
   - **Technology**: Firebase Storage
   - **Description**: Firebase Storage is used for storing and serving user-uploaded images, such as plant images for AI analysis or user profile pictures. It offers scalable cloud storage with powerful security features, ensuring the safe storage and retrieval of user data.

7. **Backend Hosting**:
   - **Technology**: Render.com
   - **Description**: The backend services are hosted on a third-party platform called Render.com, which offers free hosting services for web applications and APIs.

Render.com provides a scalable and reliable hosting environment with built-in support for Node.js applications, ensuring high availability and performance of the backend services.

This system architecture leverages a combination of frontend (React Native), backend (Node.js, Express), and cloud services (Firebase, Google Gemini Pro Vision API, Render.com) to create a robust and scalable plant shopping mobile application. It provides seamless user authentication, real-time data synchronization, AI-based plant care advice, and reliable backend hosting, ensuring a smooth and enjoyable user experience.

3.3.3 UML Use Case Diagrams:

Use Case-1:

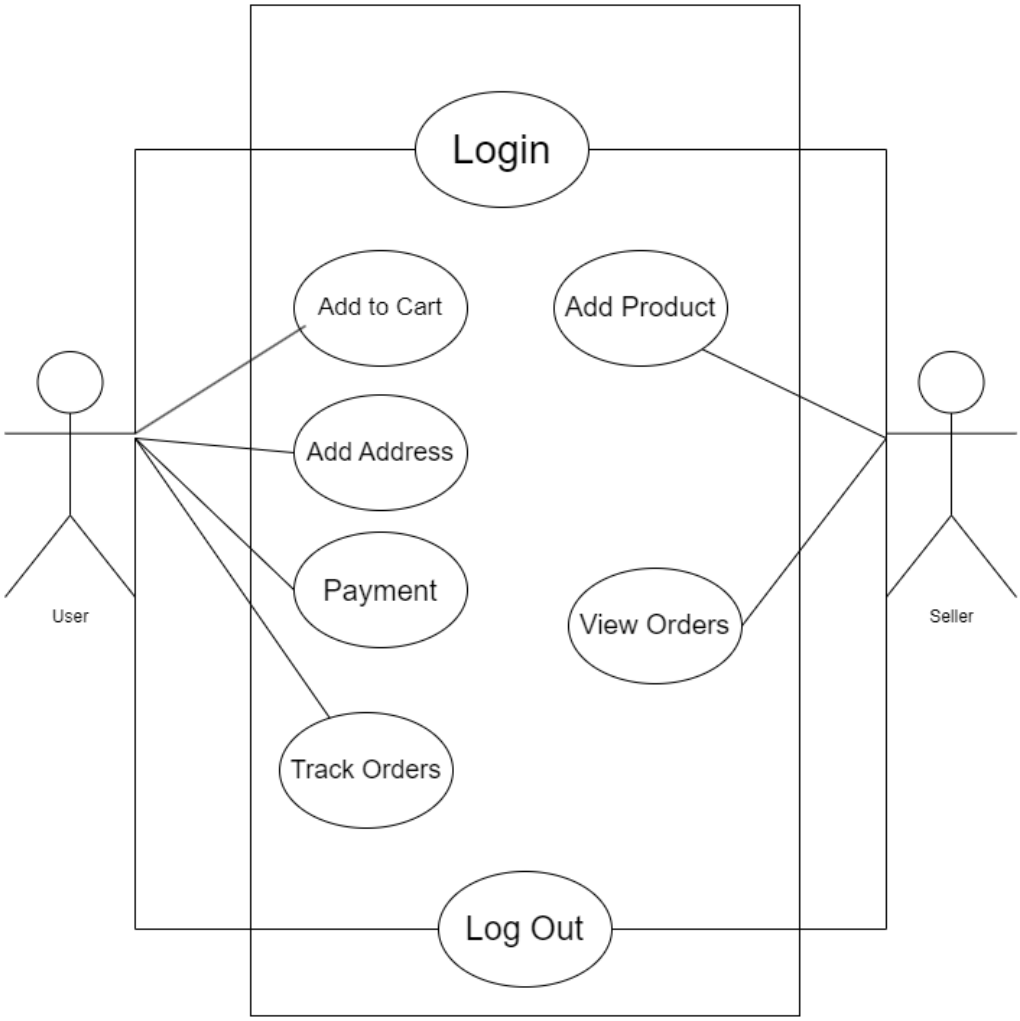

Fig. 3.1: UML Use Case Diagram Case-1

Use Case-2:

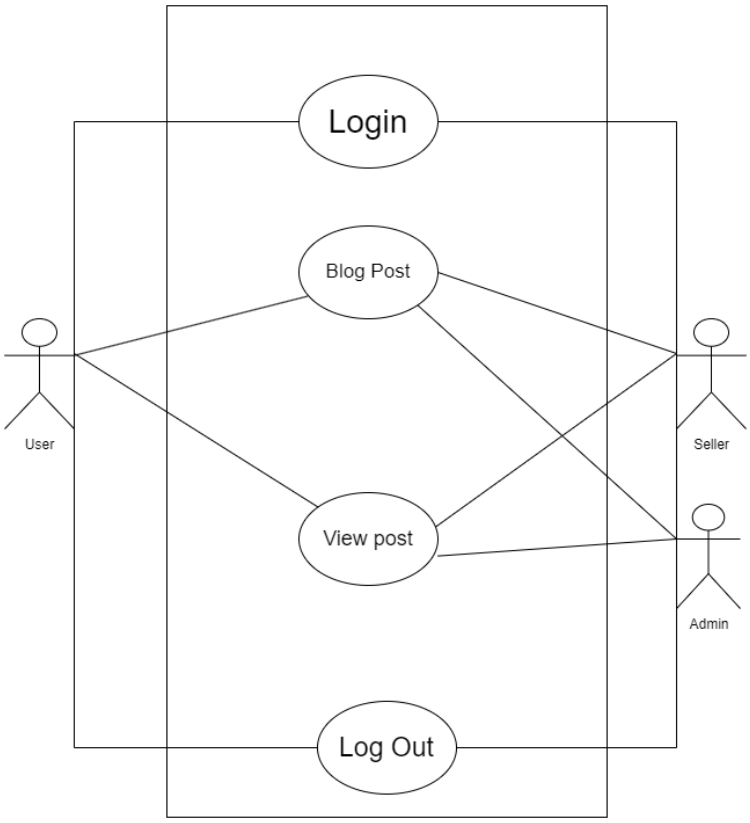

Fig. 3.2: UML Use Case Diagram Case-2

Use Case-3:

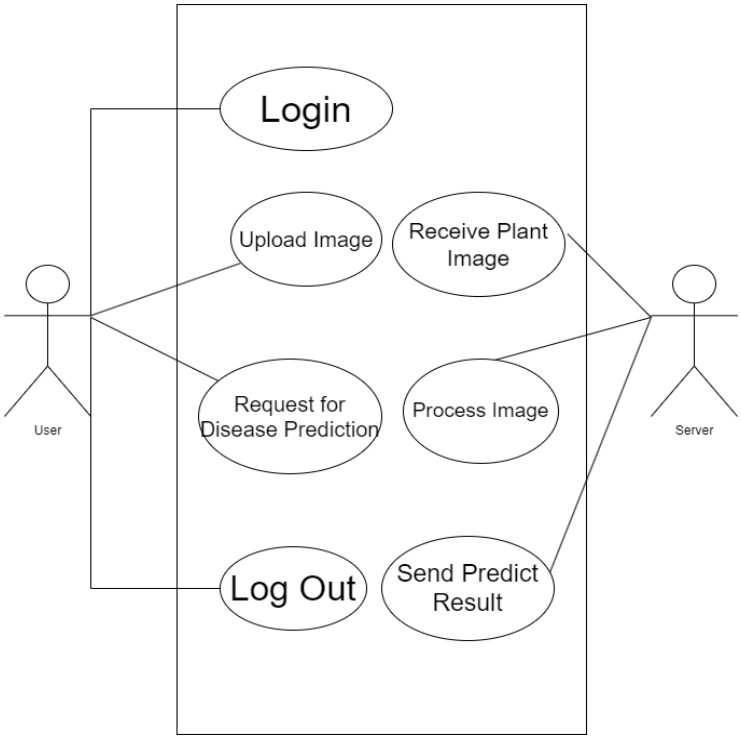

Fig 3.3- UML Use Case Diagram Case-3

Use Case-4:

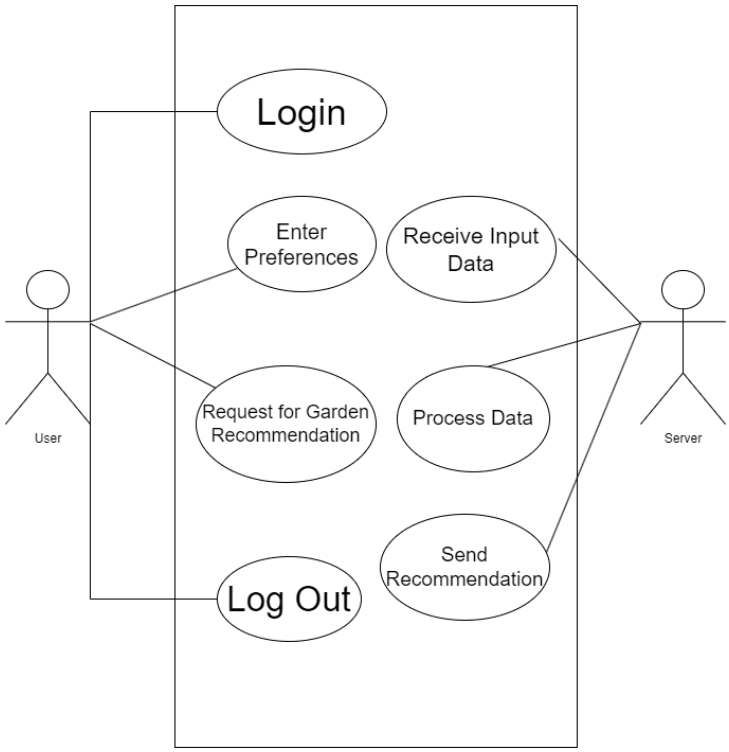

Fig 3.3- UML Use Case Diagram Case- 4

3.3.4 UML Sequence Diagrams:

Sequence Diagram where user orders products using PlantIT Mobile App

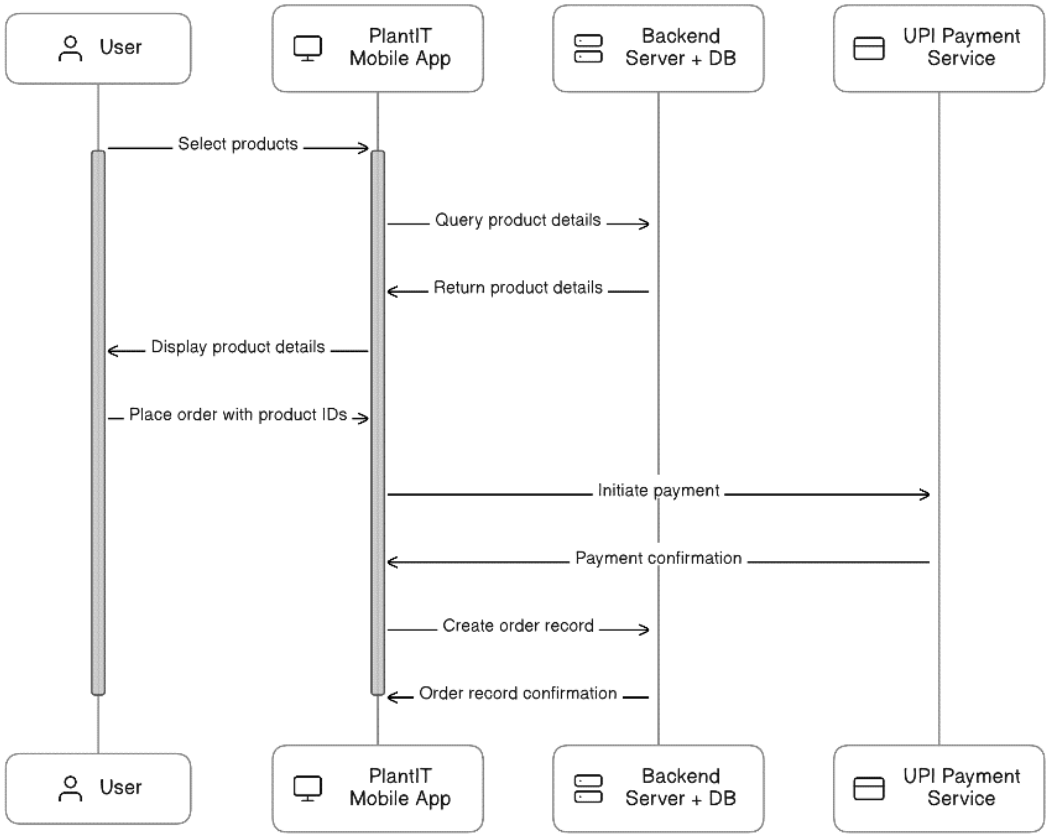

Fig 3.3- UML Sequence Diagram

Chapter 4

Implementation

For the implementation of the project, the following methodology was adopted:

4.1 Methodology:

- **Agile Development**: The project followed an agile development methodology, with iterative development cycles known as sprints. This approach allowed for flexibility in responding to changing requirements and enabled continuous improvement based on feedback received throughout the development process.

- **Feature Prioritization**: Features were prioritized based on user needs, project objectives, and technical feasibility. The team identified core functionalities such as e-commerce capabilities, AI-based recommendations, and plant care advice as the primary focus areas for initial implementation. Additional features, including community engagement features like the blog, were planned for subsequent iterations based on their importance and complexity.

- **Technology Stack Selection**: The chosen technology stack was carefully evaluated to ensure compatibility with project requirements and alignment with development team expertise. React Native was selected for mobile app development due to its ability to deliver cross-platform applications with native-like performance. Firebase services, including authentication, real-time database, and storage, were chosen for their ease of integration and scalability. Node.js and Express were utilized for backend development to provide a lightweight and efficient server-side runtime environment.

- **Collaborative Development**: The development team adopted a collaborative approach, with regular meetings and communication channels established to facilitate collaboration and coordination. GitHub was used as the primary version control system, allowing team members to collaborate on code, track changes, and manage project tasks using features such as issues and pull requests.

- **Iterative Development and Feedback Loop**: The project progressed through iterative development cycles, with each iteration focusing on delivering specific features or user stories. At the end of each iteration, a demo was conducted to showcase the implemented features to stakeholders and gather feedback. This feedback was used to refine

and prioritize the backlog for subsequent iterations, ensuring that the project remained aligned with user expectations and business goals.

4.2 Testing Plan

After the completion of the project implementation, the following testing and verification plan was executed:

- **Unit Testing:** Unit tests were written for individual components and functionalities of the application, ensuring that each part of the system behaved as expected.

- **Integration Testing:** Integration tests were conducted to verify the interactions between different components and services, ensuring seamless communication and data flow.

- **Validation Testing:** Various validation tests, including functional, usability testing was carried out, so that the application's features and functionalities were rigorously assessed to ensure they met the specified requirements and provided a seamless user experience.

- **Performance Testing:** Performance tests were conducted to assess the responsiveness and scalability of the application under different loads and usage scenarios. This ensured that the application could handle concurrent user activity and maintain acceptable performance levels.

Production Build Test Cases and Issue Resolution Log

Test ID	Test Case Title	Test Condition	System Behaviour	Expected Result
T01	Firebase Login SHA Key Issue	Production build after running the application	Developer error due to incorrect SHA key	Correct SHA key added according to Firebase docs
T02	Activity Indicator Visibility Issue	Product page load	Activity indicator not visible	Activity indicator added and visible
T03	Font colour Issue in Dark Mode UI	Dark mode enabled in address input page	Text colour becomes invisible during dark mode	Font colour adjusted for visibility in dark mode
T04	Form Validation for User Input	User submits incomplete form data	Form submitted with missing or invalid fields	Error message prompts user to complete all fields

4.3 Screenshots of Implementation:

1. User Authentication Screenshots:

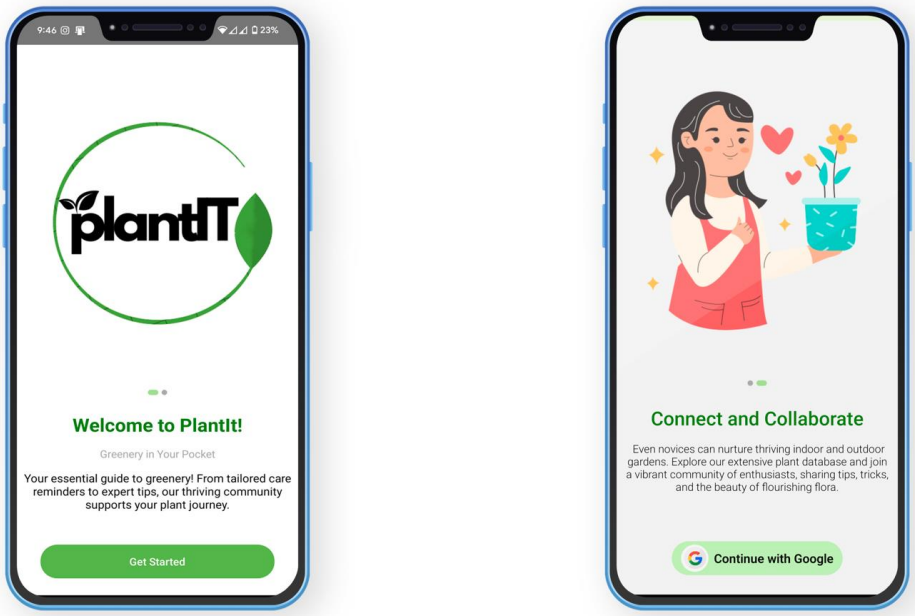

Fig 4.1- User Authentication Screenshot

2. E-Commerce Screenshots:

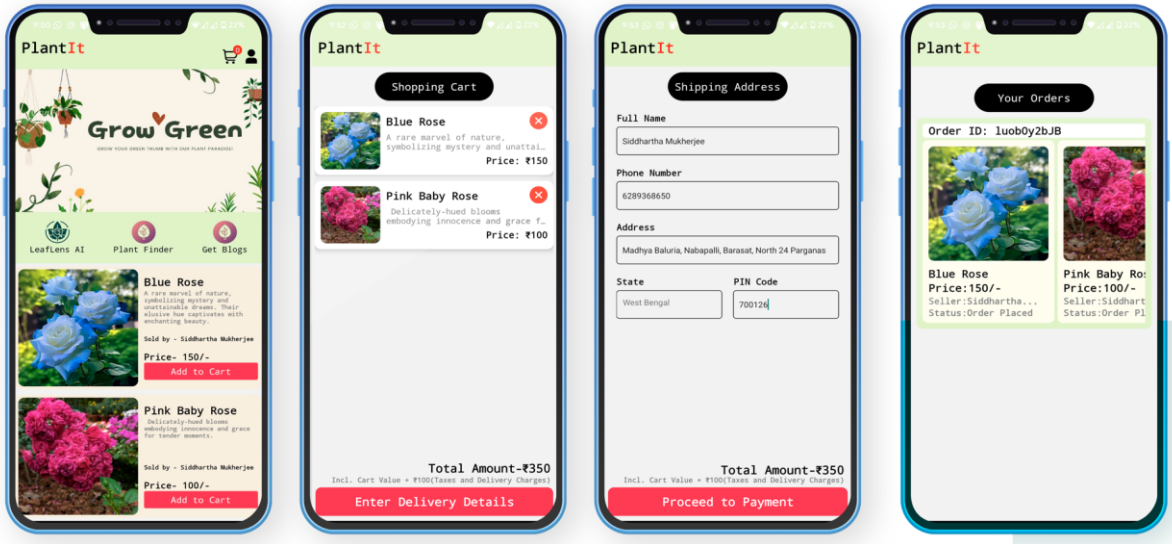

Fig 4.2: E-Commerce Screenshots

3. Disease Detection Screenshot:

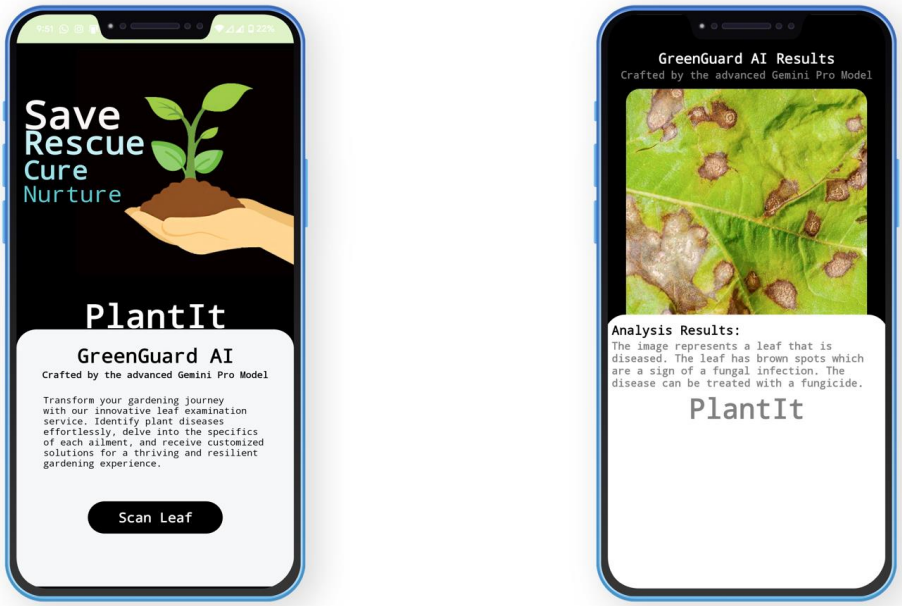

Fig 4.3: Disease Detection Screenshots

4. Plant Recommendation Screenshot:

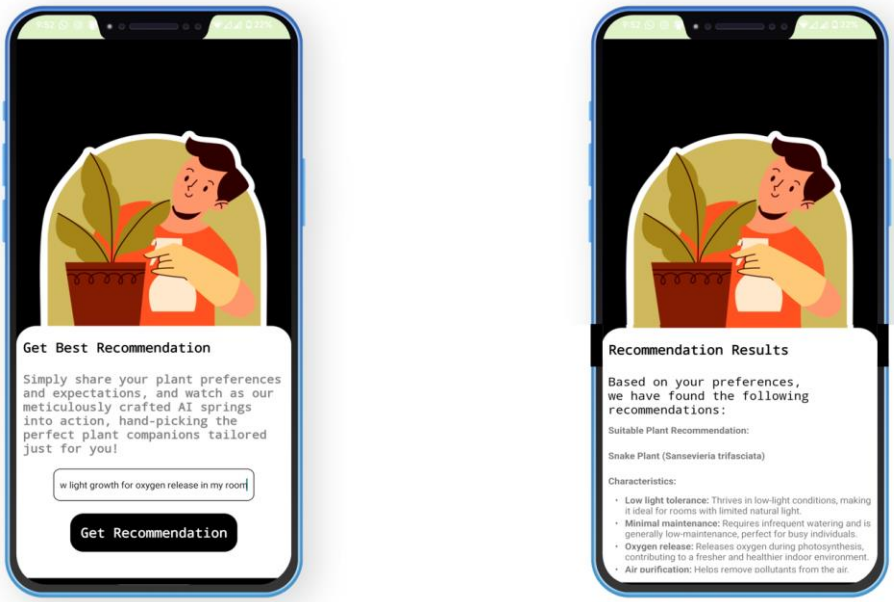

Fig 4.4: Plant Recommendation Screenshots

5. Blog Screenshots:

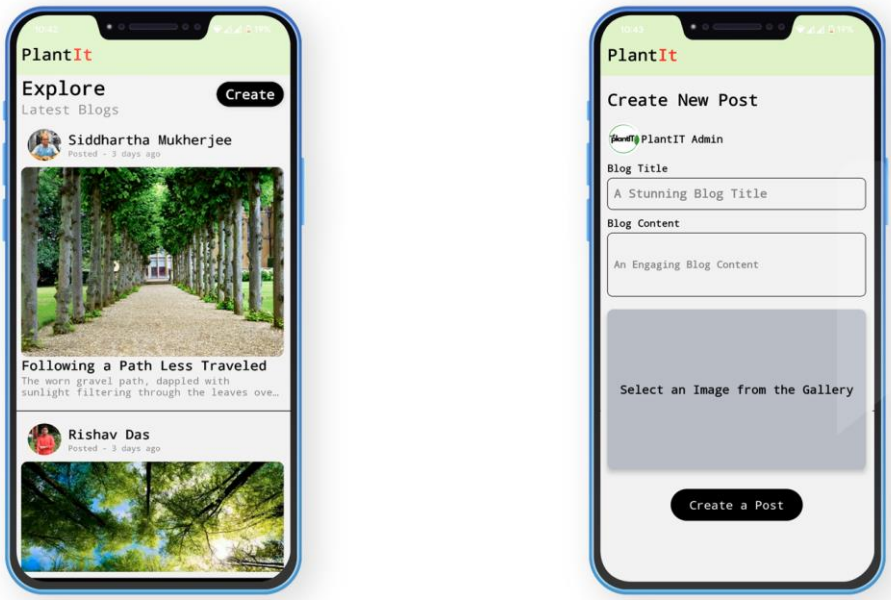

Fig 4.5: Blog Screen Screenshots

Chapter 5

Standards Adopted

5.1 Design Standards:

In adherence to established design standards, our project followed a systematic approach to architecting the software system. We employed the following recommended practices:

- **Utilization of UML Diagrams**: Unified Modelling Language (UML) diagrams were extensively used to visualize and document the system architecture, including use case diagrams, class diagrams. These diagrams provided a clear understanding of the system's structure, behaviour, and interactions.

- **Modular Design Approach**: The system was designed using a modular approach, with complex components broken down into smaller, manageable modules. This modular design facilitated reusability, maintainability, and scalability, allowing for easier management of the codebase.

- **Scalability Considerations**: During the design phase, we meticulously assessed scalability requirements, ensuring that the system was structured to seamlessly accommodate future expansion in user base, data volume, and functionality. We implemented scalable architecture patterns, including utilization of cloud computing services were deemed appropriate, to fortify the system's ability to grow and evolve.

5.2 Coding Standards:

The project adhered to established coding standards to ensure consistency, readability, and maintainability of the codebase. Key coding standards followed include:

- **Meaningful Naming Conventions**: Descriptive and meaningful names were used for variables, functions, classes, and other identifiers, following consistent naming conventions such as camelCase.

- **Structured Code Organization**: Code was organized into logical blocks and modules, with clear separation of concerns were employed to promote code modularity and maintainability.

- **Clear Code Readability**: Code was written with clarity and conciseness in mind, avoiding overly complex or convoluted logic. Comments and documentation were utilized to explain the purpose of code blocks, algorithms, and complex operations.

- **Consistent Indentation and Formatting**: Consistent indentation and formatting styles were applied throughout the codebase to enhance readability. Industry-standard formatting guidelines were followed, and code linters were utilized to enforce consistent coding styles.

- **Robust Error Handling**: Robust error handling mechanisms were implemented to gracefully handle exceptions, errors, and edge cases. Appropriate error codes, logging, and exception handling techniques were employed to aid in troubleshooting and debugging.

5.3 Testing Standards:

The project followed established testing standards to ensure the reliability, functionality, and quality of the software product. Key testing standards observed include:

- **Documentation as per IEEE 829**: Test documentation was prepared in accordance with IEEE 829 standard, outlining the format and content of various test documents, including test plans, test cases, and test reports.

- **Comprehensive Test Coverage**: Comprehensive test coverage was achieved through a combination of unit tests, integration tests, system tests, and acceptance tests.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion:

In conclusion, the project has successfully addressed the challenges faced by individuals in online plant shopping by developing a comprehensive mobile application. The implementation of e-commerce facilities, AI-based plant recommendation and care advice, and community integration through blogging has significantly enhanced the user experience. By providing a user-friendly platform for buying and selling plants, coupled with expert guidance and community support, the project has contributed to making online plant shopping more convenient, reliable, and engaging.

7.2 Future Scope:

While the project has achieved the needed results, there are numerous opportunities for further improvement and expansion in the future:

- **Enhanced AI Capabilities**: Continuously refine and expand the AI-based features, such as plant recommendation and plant care advice, by incorporating advanced machine learning algorithms and data analytics techniques. This will improve the accuracy, coverage, and intelligence of the AI system, providing users with more personalized and effective assistance.

- **Geolocation-based Services**: Integrate geolocation-based services to offer location-specific plant recommendations, gardening tips, and local plant nursery listings. By leveraging geolocation features, the application can provide users with tailored recommendations and advice based on their regional climates and conditions.

- **Expansion of Community Features**: Expand the community integration features by incorporating social networking elements, such as user profiles, forums, and live chat support. This will foster greater user engagement, interaction, and knowledge sharing within the application's community, creating a vibrant ecosystem of plant enthusiasts.

By pursuing these future enhancements and advancements, the project can continue to evolve and innovate, further enhancing its value proposition and impact in the realm of online plant shopping and gardening assistance.

References:

The following references were taken into considerations while preparing the project:

1. React Native Documentation: https://reactnative.dev/docs/getting-started
2. Node JS Documentation: https://nodejs.org/docs/latest/api/
3. Render.com Documentation: https://docs.render.com/
4. React JS Documentation: https://react.dev/learn
5. Google Firebase Documentation https://rnfirebase.io/
6. Google Gemini Documentation https://ai.google.dev/tutorials/get_started_node

These references provided valuable information and guidance during the development of the project.

INDIVIDUAL CONTRIBUTION REPORT

PlantIT

SIDDHARTHA MUKHERJEE
Roll No. - 21052365

Abstract: Our project, PlantIT, aims to establish an online platform for buying and selling plants to meet the growing demand for accessible greenery. Additionally, it offers AI-powered plant disease detection and remedies, personalized plant recommendations, and a blogging community for like-minded plant enthusiasts to share experiences. Through intuitive design and robust functionality, PlantIT seeks to provide a seamless user experience, bridging market gaps and promoting sustainable living within a vibrant plant-loving community. This report outlines PlantIT's development process, emphasizing its contributions to sustainable living and community building.

Individual contribution and findings: Throughout the project, I played a pivotal role in developing both the backend systems using Node.js and the frontend React Native application. This journey allowed me to delve deeply into the intricacies of API utilization and understand the seamless interaction between frontend and backend through API calls. Another part of my integrating the Google Gemini API into our application. This involved not only understanding its functionalities but also its pricing structures and limitations, ensuring optimal usage within our project's context. Additionally, I gained proficiency in handling image uploads using Multer and effectively managing API responses to enhance user experience. In the realm of E-commerce, I honed my skills in designing database schemas, particularly in Firebase real-time database. I became adept at managing orders with unique order IDs and users with unique user IDs, fostering smooth transaction processes. Furthermore, I explored the integration of UPI payment systems into our application, facilitating secure and convenient checkouts for our customers. An invaluable learning experience was implementing Google OAuth login, enhancing user management within our application. This streamlined the login process and bolstered security measures, contributing to an overall seamless user experience. Throughout these endeavours, I not only contributed to the project's development but also expanded my skill set significantly. Each challenge presented an opportunity for growth, and I emerged with a deeper understanding of backend development, frontend design, API integration, and database management.

Individual contribution to project report preparation: In the project report I took the lead in defining the problem statement, and writing the planning of the project (Chapter- 3), I prepared the UML Sequence diagram which provided insights into the project's architecture. Moreover, I played a role in documenting the implementation phase (Chapter- 4), covering methodologies, testing procedures, and providing details about the implementation using appropriate screenshots. Furthermore, I also contributed in framing the standards adopted (Chapter- 5) in design, coding, and testing aspects.

Individual contribution for project presentation and demonstration: In presenting the project, my primary role was in framing and delivering the presentation, where I showcased the functionality of the AI-based systems we developed. One key demonstration involved presenting the AI-based plant disease prediction feature. By capturing a photo of a diseased leaf, the system could diagnose the issue and provide relevant remedies to assist users. Additionally, I showcased the AI-based plant preference prediction feature, where users could input their preferences, and the system would suggest suitable plant options accordingly.

Full Signature of Supervisor: Full signature of the student:

…… ……..

INDIVIDUAL CONTRIBUTION REPORT:

PlantIT

DIPANKAR KHANRA
2105963

Abstract: The objective of our project, PlantIT, is to create an online platform facilitating the buying and selling of plants, catering to the increasing demand for accessible greenery. Through intuitive design and robust functionality, PlantIT aims to bridge the gap in the market by providing a seamless user experience for plant enthusiasts. This report details the development process and key features of PlantIT, highlighting its contribution to promoting sustainable living and fostering a community of plant lovers.

Individual contribution and findings: Within the project group, I focused on two key areas: creating the Software Requirements Specification (SRS) document and designing the user interface (UI) using Figma. For the SRS, I meticulously documented both functional and non-functional requirements, ensuring clarity for the development team. Researching user feedback helped enhance the SRS, expanding the project's scope and usability. In UI design, I prioritised user-centricity, crafting wireframes and prototypes for intuitive navigation and aesthetic appeal. Through these processes, I learned the importance of effective communication, collaboration, and user feedback in shaping the project's direction.

Individual contribution to project report preparation: For the project report preparation, I contributed significantly to the Introduction section, providing context on the project's background and rationale. Additionally, I outlined the basic features of PlantIT in the corresponding section, highlighting its core functionalities and value proposition. My contributions aimed to provide readers with a comprehensive understanding of the project's objectives and scope, setting the stage for the subsequent chapters.

Individual contribution for project presentation and demonstration: In preparation for the project presentation and demonstration, I played a pivotal role in developing the content and structure of the presentation slides. Leveraging insights from the SRS and UI design process, I crafted a compelling narrative that effectively communicates PlantIT's value proposition and key features to the audience. During the demonstration, I showcased the UI prototypes created in Figma, highlighting the intuitive navigation and interactive elements of the platform. Additionally, I addressed audience inquiries and provided insights into the project's development roadmap, ensuring a comprehensive understanding of PlantIT's potential impact and future trajectory.

Full Signature of Supervisor: Full signature of the student:

…… …….

INDIVIDUAL CONTRIBUTION REPORT:

PlantIT

RISHAV DAS
Roll No. - 21052347

Abstract: Our project, PlantIT, aims to establish an online platform for buying and selling plants to meet the growing demand for accessible greenery. Additionally, it offers AI-powered plant disease detection and remedies, personalized plant recommendations, and a blogging community for like-minded plant enthusiasts to share experiences. Through intuitive design and robust functionality, PlantIT seeks to provide a seamless user experience, bridging market gaps and promoting sustainable living within a vibrant plant-loving community. This report outlines PlantIT's development process, emphasizing its contributions to sustainable living and community building.

Individual contribution and findings: Throughout the project, I played a pivotal role in developing the front-end React Native application for our e commerce platform focused on plant sales.I actively participated in designing and implementing the user interface for the application. This involved creating visually appealing layouts, incorporating intuitive navigation, and ensuring a seamless user experience throughout the app. I was responsible for developing various front end components using React Native, such as product listings, cart management, checkout process, and user profile screens. These components were crucial in providing essential functionalities to our users. I collaborated closely with the back-end development team to integrate front-end components with the back-end services. This involved making API calls, handling data responses, and ensuring proper synchronization between the front-end and back-end systems. I conducted performance optimization tasks to enhance the speed and responsiveness of the application. This included optimizing code, reducing unnecessary re-renders, and implementing efficient data fetching techniques to minimize loading times. I actively participated in identifying and fixing bugs within the front-end application. Additionally, I conducted thorough testing to ensure the reliability and stability of the application across different devices and platforms. I played a key role in gathering user feedback and incorporating it into the development process. By understanding user preferences and pain points, I contributed to refining the front-end application to better meet the needs of our target audience. I documented the front-end development process, including code structure, component functionalities, and integration procedures. This documentation facilitated knowledge sharing within the team and served as a valuable resource for future development iterations. Overall, my contributions to the front-end development of the React Native e commerce application for plant sales were essential in delivering a high-quality, user-friendly, and feature-rich platform to our customers.

Individual contribution to project report preparation: In the project report I have written the introduction of the project (Chapter- 1), I prepared the UML Use case diagram which provided insights into the project's architecture. Moreover, I played a role in documenting the implementation phase (Chapter- 4), covering methodologies, testing procedures, and providing details about the implementation using appropriate screen-shots. Furthermore, I also contributed in framing the standards adopted (Chapter- 5) in design, coding, and testing aspects.

Individual contribution for project presentation and demonstration: For the project presentation and demonstration, my individual contributions were crucial in effectively showcasing the features and functionalities of our e-commerce applicationn. I demonstrated the seamless flow of actions, starting with the user logging in, followed by product selection and addition to the cart. Then, I showcased the process of entering address details, proceeding to payment, and completing the transaction. Additionally, I emphasized the user's ability to visit their order page to track past purchases and manage their account information. My role involved effectively presenting and explaining these components to the audience, emphasizing their significance in the overall functionality and user experience of the application.

Full Signature of Supervisor: Full signature of the student:

Rishav Das

…… ……..

TURNITIN PLAGIARISM REPORT

PlantIT

ORIGINALITY REPORT

7%	2%	1%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Central Queensland University Student Paper	1%
2	Submitted to Barnet and Southgate College Student Paper	1%
3	Submitted to Kaplan College Student Paper	1%
4	Submitted to Nottingham Trent University Student Paper	<1%
5	Submitted to Asia Pacific Instutute of Information Technology Student Paper	<1%
6	Submitted to Manchester Metropolitan University Student Paper	<1%
7	"Applications of Computational Methods in Manufacturing and Product Design", Springer Science and Business Media LLC, 2022 Publication	<1%
8	Submitted to Higher Education Commission Pakistan	<1%

	Student Paper	
9	Submitted to IUBH - Internationale Hochschule Bad Honnef-Bonn Student Paper	<1%
10	Submitted to Leyton Sixth Form College Student Paper	<1%
11	Submitted to Trine University Student Paper	<1%
12	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
13	Submitted to University of Witwatersrand Student Paper	<1%
14	Submitted to ea Student Paper	<1%
15	Submitted to Aston University Student Paper	<1%
16	Submitted to Waterford Institute of Technology Student Paper	<1%
17	slideplayer.com Internet Source	<1%
18	www.engineering.com Internet Source	<1%
19	Submitted to University of East London	

	Student Paper	<1%
20	www.groundai.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		