Document Title: Deliverable 2

Team Name: Smart Sprout

Project Name: Smart Home-Garden System

Group Number: Group 9

Team Member & Student ID:

● Aditi Patel	n01525570
● Birava Prajapati	n01579924
● Darshankumar Prajapati	n01574247
● Zeel Patel	n01526282

<u>**Table Of Content**</u>

Name	Student Id	Github Id	Signature	Efforts
Aditi Patel	n01525570	AditiPatel5570	Apatel	100%
Birava Prajapati	n01579924	BiravaPrajapati9924	Biravap	100%
Darshankumar Prajapati	n01574247	DarshanPrajapati4247	Darshan	100%
Zeel Patel	n01526282	ZeelPatel6282	Zeel	100%

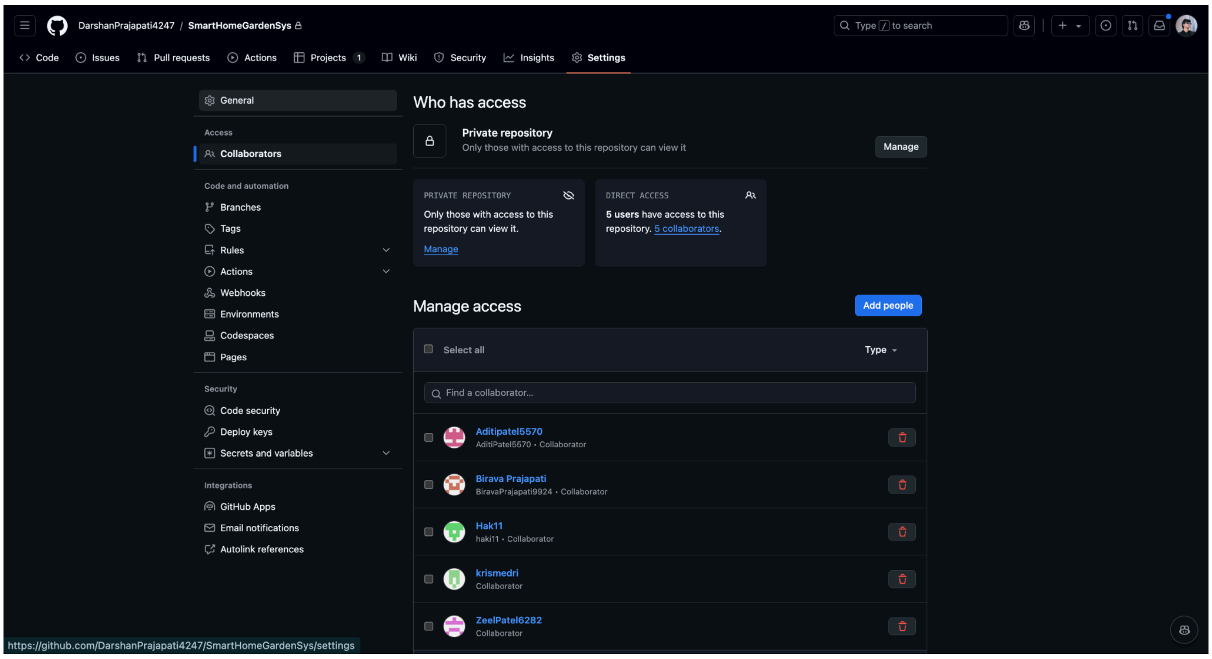
Project Scope and Goals Targeted in This Course

- **Project Scope**: The project involves building a **Smart Sprout application** using Android Studio in Java, utilising a bottom navigation layout and implementing MVVM architecture. The application will feature various screens, including home, profile, search, diagnosis, and sensors.
- **Goals**:
  - To successfully develop and deploy the Smart Sprout application.
  - To implement real-time sensor data handling.
  - To use Firebase for user authentication and database management.
  - To ensure effective collaboration among team members using GitHub.

GitHub Repo Link and Strategy

- **GitHub Repo Link**: https://github.com/DarshanPrajapati4247/SmartHomeGardenSys
- **Strategy**:
  - Each team member will work on separate feature branches (e.g., `feature/login`, `feature/profile`)..
  - Committing frequently to reflect progress.
  - Creating pull requests for code reviews to maintain code quality.
  - Merging approved changes into the main branch for production.

Invitation of Hardware professor:



Who has access

Private repository
Only those with access to this repository can view it

PRIVATE REPOSITORY
Only those with access to this repository can view it.
Manage

DIRECT ACCESS
5 users have access to this repository. 5 collaborators.

Manage access

Aditipatel5570
AditiPatel5570 • Collaborator

Birava Prajapati
BiravaPrajapati9924 • Collaborator

Hak11
haki11 • Collaborator

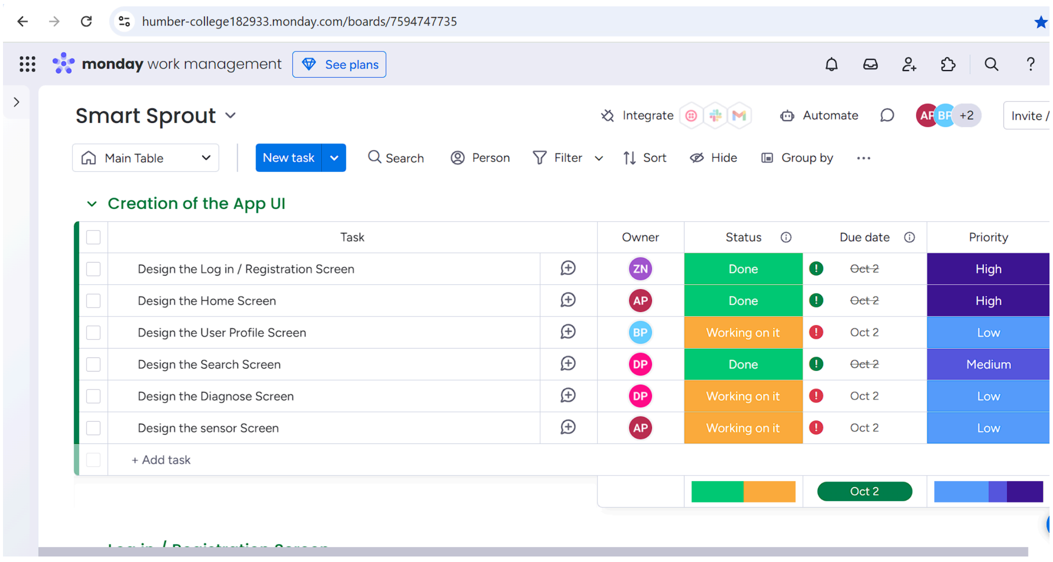
krismedri
Collaborator

ZeelPatel6282
Collaborator

https://github.com/DarshanPrajapati4247/SmartHomeGardenSys/settings

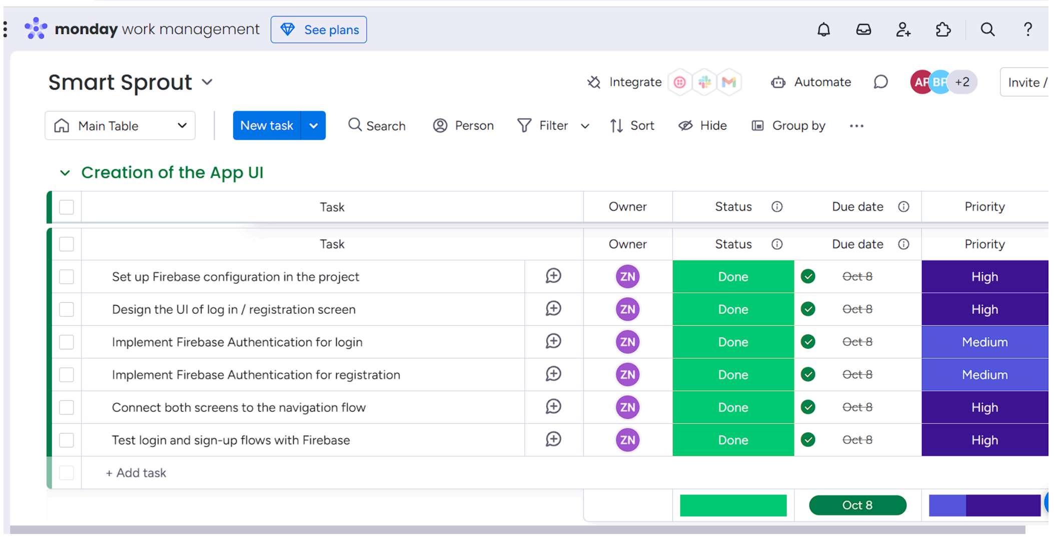
SMART SPROUT

The stories and breakdown of tasks with owner, status, start/end date, size, and priority.



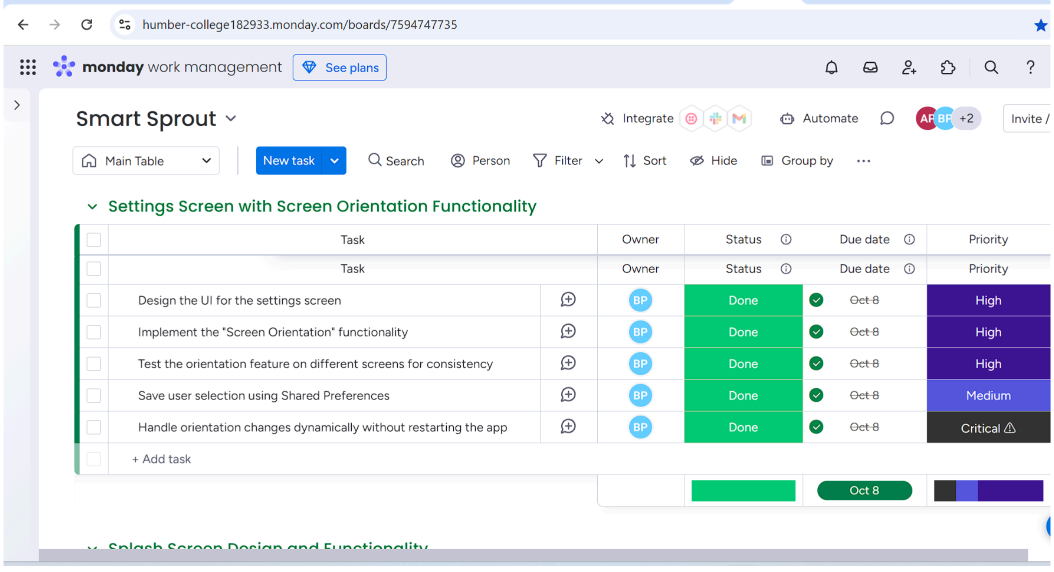
Creation of the App UI

Task	Owner	Status	Due date	Priority
Design the Log in / Registration Screen	ZN	Done	~~Oct 2~~	High
Design the Home Screen	AP	Done	~~Oct 2~~	High
Design the User Profile Screen	BP	Working on it	Oct 2	Low
Design the Search Screen	DP	Done	~~Oct 2~~	Medium
Design the Diagnose Screen	DP	Working on it	Oct 2	Low
Design the sensor Screen	AP	Working on it	Oct 2	Low
+ Add task				
			Oct 2	



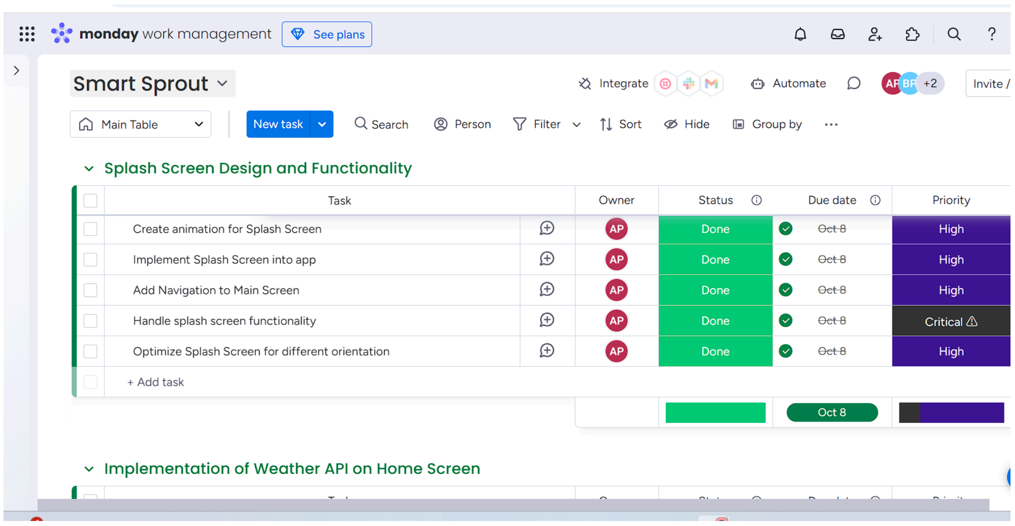
Creation of the App UI

Task	Owner	Status	Due date	Priority
Set up Firebase configuration in the project	ZN	Done	~~Oct 8~~	High
Design the UI of log in / registration screen	ZN	Done	~~Oct 8~~	High
Implement Firebase Authentication for login	ZN	Done	~~Oct 8~~	Medium
Implement Firebase Authentication for registration	ZN	Done	~~Oct 8~~	Medium
Connect both screens to the navigation flow	ZN	Done	~~Oct 8~~	High
Test login and sign-up flows with Firebase	ZN	Done	~~Oct 8~~	High
+ Add task				
			Oct 8	



Settings Screen with Screen Orientation Functionality

Task	Owner	Status	Due date	Priority
Design the UI for the settings screen	BP	Done	~~Oct 8~~	High
Implement the "Screen Orientation" functionality	BP	Done	~~Oct 8~~	High
Test the orientation feature on different screens for consistency	BP	Done	~~Oct 8~~	High
Save user selection using Shared Preferences	BP	Done	~~Oct 8~~	Medium
Handle orientation changes dynamically without restarting the app	BP	Done	~~Oct 8~~	Critical ⚠
+ Add task				
			Oct 8	



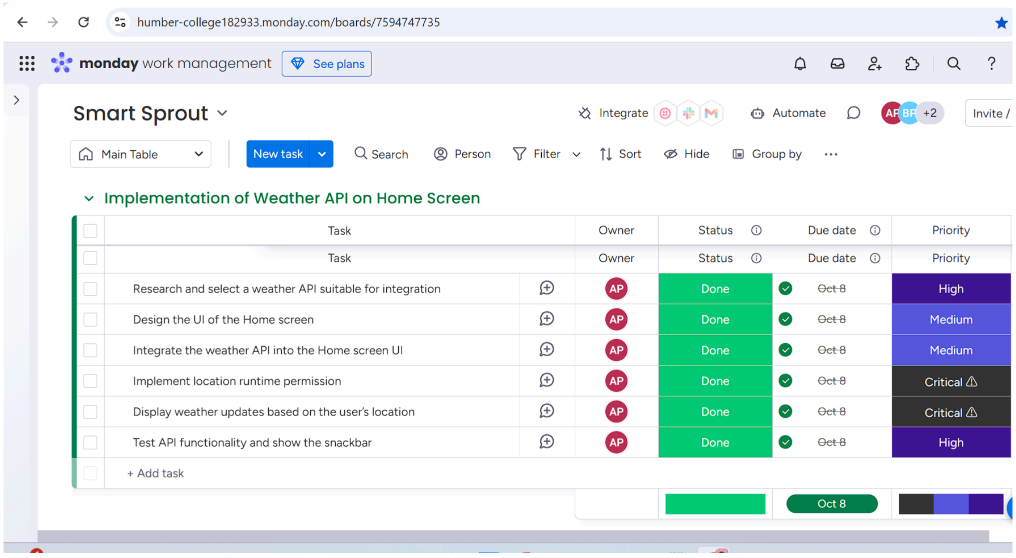
Smart Sprout

Main Table | New task | Search | Person | Filter | Sort | Hide | Group by

Splash Screen Design and Functionality

Task	Owner	Status	Due date	Priority
Create animation for Splash Screen	AP	Done	~~Oct 8~~	High
Implement Splash Screen into app	AP	Done	~~Oct 8~~	High
Add Navigation to Main Screen	AP	Done	~~Oct 8~~	High
Handle splash screen functionality	AP	Done	~~Oct 8~~	Critical ⚠
Optimize Splash Screen for different orientation	AP	Done	~~Oct 8~~	High
+ Add task				
			Oct 8	

Implementation of Weather API on Home Screen



humber-college182933.monday.com/boards/7594747735

Smart Sprout

Main Table | New task | Search | Person | Filter | Sort | Hide | Group by

Implementation of Weather API on Home Screen

Task	Owner	Status	Due date	Priority
Research and select a weather API suitable for integration	AP	Done	~~Oct 8~~	High
Design the UI of the Home screen	AP	Done	~~Oct 8~~	Medium
Integrate the weather API into the Home screen UI	AP	Done	~~Oct 8~~	Medium
Implement location runtime permission	AP	Done	~~Oct 8~~	Critical ⚠
Display weather updates based on the user's location	AP	Done	~~Oct 8~~	Critical ⚠
Test API functionality and show the snackbar	AP	Done	~~Oct 8~~	High
+ Add task				
			Oct 8	

SMART SPROUT

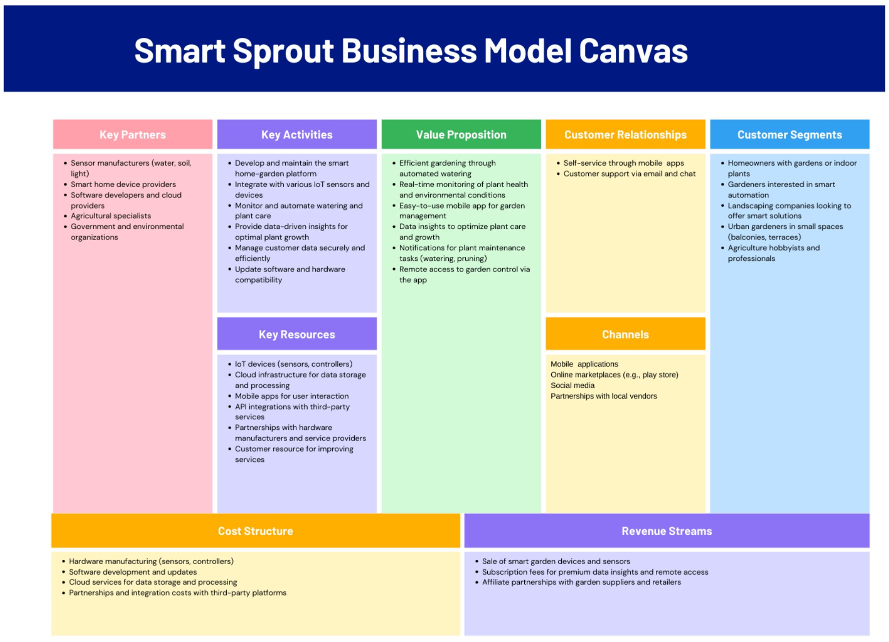
Items in DoD:

- Coding task is completed.
- Application should not be crashing.
- The task should be reviewed by the team members.
- The task should be committed and push to local branch
- Meeting task requirements for the product owner.
- UI should be user friendly.

How We Meet DoD criteria:

- **Coding task is completed:** All coding tasks were finished according to the requirements and specifications provided.
- **Application should not be crashing**: Rigorous testing was conducted to ensure the application is stable and does not crash under various conditions.
- **The task should be reviewed by the team members**: Code reviews were conducted by team members to ensure the quality and correctness of the code. This helps in identifying any potential issues or improvements.
- **The task should be committed and pushed to the local branch:** Once the coding task was completed and reviewed, the changes were committed and pushed to the local branch in the version control system. This ensures that the code is properly versioned and can be tracked.
- **Meeting task requirements for the product owner**: The task was verified to meet the requirements specified by the product owner. This includes ensuring that all acceptance criteria are met and the functionality aligns with the product vision.
- **UI should be user-friendly:** The user interface was designed and tested to be intuitive and easy to use. This includes usability testing to ensure that users can navigate and interact with the application effectively.

<u>Business Model Canvas</u>


Smart Sprout Business Model Canvas

Key Partners
- Sensor manufacturers (water, soil, light)
- Smart home device providers
- Software developers and cloud providers
- Agricultural specialists
- Government and environmental organizations

Key Activities
- Develop and maintain the smart home-garden platform
- Integrate with various IoT sensors and devices
- Monitor and automate watering and plant care
- Provide data-driven insights for optimal plant growth
- Manage customer data securely and efficiently
- Update software and hardware compatibility

Key Resources
- IoT devices (sensors, controllers)
- Cloud infrastructure for data storage and processing
- Mobile apps for user interaction
- API integrations with third-party services
- Partnerships with hardware manufacturers and service providers
- Customer resource for improving services

Value Proposition
- Efficient gardening through automated watering
- Real-time monitoring of plant health and environmental conditions
- Easy-to-use mobile app for garden management
- Data insights to optimize plant care and growth
- Notifications for plant maintenance tasks (watering, pruning)
- Remote access to garden control via the app

Customer Relationships
- Self-service through mobile apps
- Customer support via email and chat

Channels

Mobile applications
Online marketplaces (e.g., play store)
Social media
Partnerships with local vendors

Customer Segments
- Homeowners with gardens or indoor plants
- Gardeners interested in smart automation
- Landscaping companies looking to offer smart solutions
- Urban gardeners in small spaces (balconies, terraces)
- Agriculture hobbyists and professionals

Cost Structure
- Hardware manufacturing (sensors, controllers)
- Software development and updates
- Cloud services for data storage and processing
- Partnerships and integration costs with third-party platforms

Revenue Streams
- Sale of smart garden devices and sensors
- Subscription fees for premium data insights and remote access
- Affiliate partnerships with garden suppliers and retailers

1. Key Partners

- **Sensor manufacturers** (water, soil, light): Essential for providing the sensors that monitor garden conditions.
- **Software developers and cloud providers**: Necessary for building and maintaining the software platform and cloud storage.
- **Agricultural specialists**: Provide expert knowledge for plant care and smart gardening techniques.
- **Government and environmental organisations**: Essential for getting the patent for our application.

2. Key Activities

- **Develop and maintain the smart home-garden platform**: Continuous software and hardware improvements.
- **Integrate with various IoT sensors and devices**: Compatibility with multiple smart home devices.
- **Monitor and automate watering and plant care**: Automatic care of plants through IoT technology.
- **Provide data-driven insights for optimal plant growth**: Use analytics to improve plant care.
- **Manage customer data securely and efficiently**: Ensure privacy and regulatory compliance.
- **Update software and hardware compatibility**: Stay current with technological advancements.

3. Key Resources

- **IoT devices** (sensors, controllers): Hardware that collects data and automates plant care.
- **Cloud infrastructure for data storage and processing**: Essential for storing data and running the system.
- **Mobile apps for user interaction**: User-friendly mobile platforms for monitoring and controlling the garden.
- **API integrations with third-party services**: Expand functionality by adding APIs.
- **Partnerships with hardware manufacturers and service providers**: Collaborations for improving hardware.
- **Customer resource for improving services**: Use customer feedback to enhance the platform.

4. Value Proposition

- **Efficient gardening through automated watering**: Reduce effort through automation.
- **Real-time monitoring of plant health and environmental conditions**: Always be informed of your garden's status.
- **Easy-to-use mobile app for garden management**: User-friendly interface for managing the garden.
- **Data insights to optimize plant care and growth**: Use data to make better decisions.
- **Notifications for plant maintenance tasks**: Reminders for watering, pruning, etc.
- **Remote access to garden control via the app**: Control garden systems from anywhere.

5. Customer Relationships

- **Self-service through mobile apps**: Users manage the garden themselves with the app.
- **Customer support via email and chat**: Assistance for troubleshooting or other inquiries.

6. Customer Segments

- **Homeowners with gardens or indoor plants**: Primary users of the system.
- **Gardeners interested in smart automation**: Individuals who want to automate plant care.
- **Landscaping companies looking to offer smart solutions**: Professional users who can offer the product to clients.
- **Urban gardeners in small spaces**: Those gardening on balconies or terraces.

- **Agriculture hobbyists and professionals**: People passionate about or working in agriculture.

7. Channels

- **Mobile applications**: Main interface for users to control and monitor the system.
- **Online marketplaces** (e.g., Play Store): Where the app and possibly hardware are sold.
- **Social media**: Marketing and engagement.
- **Partnerships with local vendors**: Retail outlets or partners offering the system.

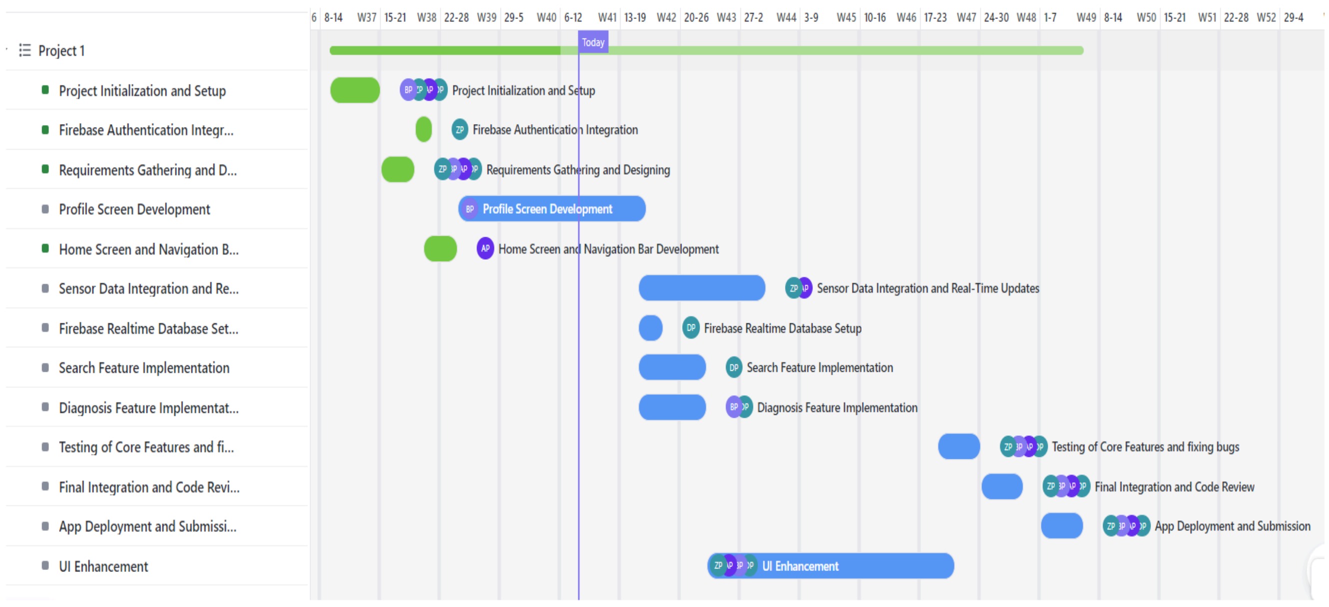
8. Cost Structure

- **Hardware manufacturing** (sensors, controllers): Key costs include producing the physical devices.
- **Software development and updates**: Developing and maintaining the platform.
- **Cloud services for data storage and processing**: Essential for running the back-end infrastructure.
- **Partnerships and integration costs with third-party platforms**: Expenses related to collaborations.

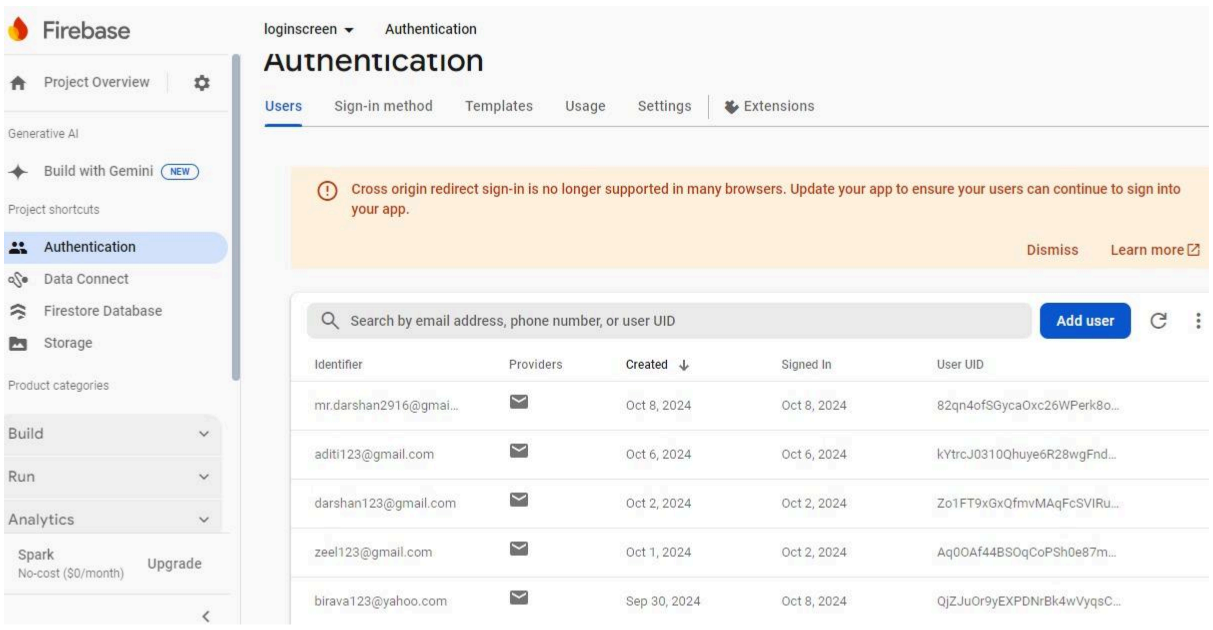
9. Revenue Streams

- **Sale of smart garden devices and sensors**: Revenue from hardware sales.
- **Subscription fees for premium data insights and remote access**: Additional revenue from value-added services.
- **Affiliate partnerships with garden suppliers and retailers**: Partnerships to generate income from garden-related products.

Gantt chart:


6
8-14
W37
15-21
W38
22-28
W39
29-5
W40
6-12
W41
13-19
W42
20-26
W43
27-2
W44
3-9
W45
10-16
W46
17-23
W47
24-30
W48
1-7
W49
8-14
W50
15-21
W51
22-28
W52
29-4
Today
Project 1
Project Initialization and Setup
Firebase Authentication Integr...
Requirements Gathering and D...
Profile Screen Development
Home Screen and Navigation B...
Sensor Data Integration and Re...
Firebase Realtime Database Set...
Search Feature Implementation
Diagnosis Feature Implementat...
Testing of Core Features and fi...
Final Integration and Code Revi...
App Deployment and Submissi...
UI Enhancement
Project Initialization and Setup
Firebase Authentication Integration
Requirements Gathering and Designing
Profile Screen Development
Home Screen and Navigation Bar Development
Sensor Data Integration and Real-Time Updates
Firebase Realtime Database Setup
Search Feature Implementation
Diagnosis Feature Implementation
Testing of Core Features and fixing bugs
Final Integration and Code Review
App Deployment and Submission
UI Enhancement

Database on the cloud



Authentication

Cross origin redirect sign-in is no longer supported in many browsers. Update your app to ensure your users can continue to sign into your app.

Search by email address, phone number, or user UID

Identifier	Providers	Created	Signed In	User UID
mr.darshan2916@gmai...	✉	Oct 8, 2024	Oct 8, 2024	82qn4ofSGycaOxc26WPerk8o...
aditi123@gmail.com	✉	Oct 6, 2024	Oct 6, 2024	kYtrcJ0310Qhuye6R28wgFnd...
darshan123@gmail.com	✉	Oct 2, 2024	Oct 2, 2024	Zo1FT9xGxQfmvMAqFcSVIRu...
zeel123@gmail.com	✉	Oct 1, 2024	Oct 2, 2024	Aq0OAf44BSOqCoPSh0e87m...
birava123@yahoo.com	✉	Sep 30, 2024	Oct 8, 2024	QjZJuOr9yEXPDNrBk4wVyqsC...

Explain Database Usage

Firebase Authentication: We're using this to handle user logins and registrations. It securely manages user credentials like email addresses and passwords, making it easier to create a smooth sign-up and sign-in experience for the app. This will ensure that each user has a personalised and secure account.
Firebase Realtime Database: This is where we'll store and sync real-time data for the app. It'll hold information like user profiles and sensor data from the plants. By using this database, we can ensure that any updates made by users or sensors are immediately reflected across all devices, so users always have the most current information available

Coding work progress since deliverable 1.

What additional features/functionality added since deliverable

- Login/Registration screen
- Implementation of weather API which gives real time data of weather.
- Designed the UI of the application.
- Implemented Setting Screen with functionality.

A Table records the daily stand-ups outcome, use any tool you like, include a screenshot into the document. Table, include date and info. Show minimum of 3 meetings.

Date	Meeting Topic	Discussion Points	Action Items
Sep 26	UI Design & MVVM Structure	- Discussed overall UI layout for application screens. - Reviewed MVVM architecture.	- Finalise UI design by Oct 2.
Sep 30	User Stories, Tasks, and Definition of Done (DoD)	- Identified user stories. - Break down stories into tasks. - Established DoD criteria.	- Complete task breakdown by Oct 2.
Oct 1	Gantt Chart & Project Timeline	- Reviewed the Gantt chart. - Discussed task durations and dependencies.	- Update Gantt chart by Oct 3.
Oct 7	Business Model Canvas	- Created the Business Model Canvas. - Analysed key components.	- Finalise the canvas by Oct 8.

Principles Used:

Single Responsibility Principle (SRP): Each class or component is designed to handle one specific task. For example, `ProfileViewModel` focuses solely on managing user profile data, while `AuthRepository` handles all authentication operations. This clear division makes the code easier to maintain and adapt.

Separation of Concerns: Utilizing the MVVM (Model-View-ViewModel) architecture allows us to separate different responsibilities within the app. The **ViewModel** (like `ProfileViewModel`) manages business logic and data operations, while the **UI layer** (such as `ProfileFragment`) is dedicated to presenting data and responding to user actions. The **Repository layer** (e.g., `AuthRepository`) deals with data management, keeping the architecture organized and clear.

DRY (Don't Repeat Yourself): We follow the DRY principle to eliminate redundancy. Common functionalities, such as data retrieval and authentication processes, are centralized within specific classes. This minimizes code duplication and reduces the chances of errors, making it easier to maintain.

STUPID Principle: This acronym stands for **Single responsibility, Tight coupling, Uncontrolled side effects, Premature optimization, Inconsistent naming, and Duplication**. We ensure our design avoids these pitfalls:

- **Single Responsibility**: Each class has one purpose (aligns with SRP).
- **Tight Coupling**: We strive for loose coupling between components, allowing changes in one part of the app without affecting others.
- **Uncontrolled Side Effects**: Code is written to avoid unexpected interactions, ensuring that changes do not inadvertently impact unrelated features.
- **Premature Optimization**: We focus on clear and maintainable code rather than optimising for performance too early in development.
- **Inconsistent Naming**: We use clear, consistent naming conventions for classes and methods, enhancing readability and understanding.
- **Duplication**: We actively work to eliminate duplicate code, adhering to the DRY principle.