## Sustainable Fertilizer Usage Optimizer for Higher Yield

##### A

##### MINOR PROJECT-I REPORT

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

GROUP NO.03

|  |  |
| --- | --- |
| **Anuj Raghuwanshi** | **0187CS221050** |
| **Basuki Nath pandey** | **0187CS221070** |
|  |  |

Under the guidance of **Mr. Ankit Gupta** (Associate Professor)

Department of Computer Science & Engineering

Sagar Institute of Science &Technology (SISTec), Bhopal (M.P)

**Approved by AICTE, New Delhi & Govt. of M.P.**

**Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)**

May - 2025

**Sagar Institute of Science & Technology (SISTec), Bhopal (M.P) Department of Computer Science & Engineering**

***CERTIFICATE***

We hereby certify that the work which is being presented in the B.Tech. Minor Project-I Report entitled **‘Sustainable Fertilizer Usage Optimizer for Higher Yield’ ,** in partial fulfillment of the requirements for the award of the degree of ***Bachelor of Technology,*** submitted to the Department of **Computer Science & Engineering**, Sagar Institute of Science & Technology (SISTec)**,**Bhopal (M.P.) is an authentic record of our own work carried out during the period from Jul-2024 toDec-2024 under the supervision of **Mr. Ankit Gupta.**

The content presented in this project has not been submitted by me for the award of any other degree elsewhere.**\**

**Anuj Raghuwanshi Basuki Nath pandey**

**0187CS221050 0187CS221070**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

***Date:***

|  |  |  |
| --- | --- | --- |
| **Mr. Ankit Gupta** | **Dr. Amit Kumar Mishra** | **Dr. D.K. Rajoriya** |
| **Project Guide** | **HOD, CSE** | **Principal** |

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Dr. D. K. Rajoriya**, **Principal, SISTec and Dr. Swati Saxena, Vice Principal SISTec** Gandhi Nagar, Bhopal for giving us an opportunity to undertake this project.

We also take this opportunity to express a deep sense of gratitude to **Dr. Amit Kumar Mishra, HOD, Department of Computer Science & Engineering** for his kindhearted support

We extend our sincere and heartfelt thanks to our guide, **Mr. Ankit Gupta,** for providing us with the right guidance and advice at crucial junctures and for showing us the right way.

I am thankful to the **Project Coordinator, Prof. Mayank kurchaniya,** who devoted his precious time in giving us the information about various aspects and gave support and guidance at every point of time.

I would like to thank all those people who helped me directly or indirectly to complete my project whenever I found myself in any issue.

|  |  |
| --- | --- |
| **Table of Content** |  |
| **TITLE** | **PAGE NO.** |
| Abstract | i |
| List of Abbreviation | ii |
| [List of Figures](#_bookmark0) | [iii](#_bookmark0) |
| List of Tables | iv |
| [Chapter 1 Introduction](#_bookmark1) | [1](#_bookmark1) |
| 1.1 [About Project](#_bookmark2) | [2](#_bookmark2) |
| 1.2 [Project Objectives](#_bookmark3) | [3](#_bookmark3) |
| Chapter 2 Software & Hardware Requirements bhb | 4 |
| 2.1 [Software Requirements](#_bookmark4) | [5](#_bookmark4) |
| 2.2 [Hardware Requirements](#_bookmark5) | [6](#_bookmark5) |
| [Chapter 3 Problem Description](#_bookmark6) | [7](#_bookmark6) |
| 3.1 [Introduction](#_bookmark7) | [8](#_bookmark7) |
| 3.2 [Problem Statement](#_bookmark8) | [8](#_bookmark8) |
| 3.3 [Objectives of the Project](#_bookmark9) | [9](#_bookmark9) |
| 3.4 [Scope of the Project](#_bookmark10) | [9](#_bookmark10) |
| 3.5 Challenges & Constraints | 10 |
| 3.6 [Expected Outcomes](#_bookmark11) | [10](#_bookmark11) |
| [Chapter 4 Literature Survey](#_bookmark12) | [11](#_bookmark12) |
| 4.1 [Introduction to the Domain](#_bookmark13) | [12](#_bookmark13) |
| 4.2 [Current Technological Interventions](#_bookmark14) | [12](#_bookmark14) |
| 4.3 [Tools and Systems Already Available](#_bookmark15) | [13](#_bookmark15) |
| 4.4 [Environmental Impact and Policy Considerations](#_bookmark16) | [13](#_bookmark16) |
| 4.5 [Challenges and Future Directions](#_bookmark17) | [14](#_bookmark17) |
| 4.6 [Conclusion](#_bookmark18) | [14](#_bookmark18) |
| [Chapter 5 Software Requirements Specification](#_bookmark19) | [15](#_bookmark19) |
| 5.1 [Functional Requirements](#_bookmark20) | [16](#_bookmark20) |
| 5.2 [Non-Functional Requirements](#_bookmark21) | [17](#_bookmark21) |
| 5.2.1 [Performance](#_bookmark22) | [17](#_bookmark22) |
| 5.2.2 [Reliability](#_bookmark23) | [17](#_bookmark23) |
| 5.2.3 [Usability](#_bookmark24) | [18](#_bookmark24) |

|  |  |  |
| --- | --- | --- |
| Chapter 6 | Software and Hardware Design | 19 |
|  | 6.1 Use Case Diagram | 20 |
|  | 6.2 Architecture | 21 |
| Chapter 7 | Coding | 22 |
|  | 7.1 Code Implementation | 23 |
| Chapter 8 | Result & Output Screens | 45 |
|  | 8.1 Results | 46 |
|  | 8.2 Output Screens | 47 |
| Chapter 9 | Conclusions & Future Work | 50 |
|  | 9.1 Conclusion | 51 |
|  | 9.2 Future Work | 51 |

[References 53](#_bookmark25)

[Project summary 54](#_bookmark26)

[Appendix-1: Glossary of Terms 59](#_bookmark27)

SISTec/BTech/CS/2024/5/MinorProject\_I/3

ABSTRACT

The **Sustainable Fertilizer Usage Optimizer for Higher Yield** is an innovative solution designed to optimize fertilizer application in agriculture. Using machine learning, the system provides tailored fertilizer recommendations based on soil health, crop requirements, and environmental factors. It integrates real-time data on temperature, humidity, and soil nutrients to ensure precise and efficient fertilizer usage. The user-friendly platform helps farmers increase crop yields, reduce environmental impact, and lower costs by promoting sustainable agricultural practices. The system contributes to more productive , cost- effective and efficient farming while preserving soil health and water health .

SISTec/BTech/CS/2024/5/MinorProject\_I/3

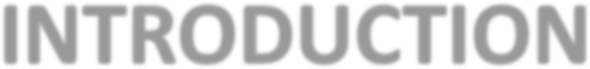
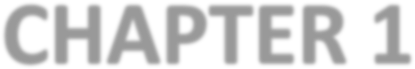
LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| **ACRONYM** | **FULL FORM** |
| PH | Potential of Hydrogen |
| NPK | Nitrogen, Phosphorus, Potassium |
| SHC | Soil Health Card |
| NSS | Nutrient Sufficiency Score |
| FMS | Fertilizer Management System |
| SFI | Soil Fertility Index |
| FAO | Food and Agriculture Organization |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

SISTec/BTech/CS/2024/5/MinorProject\_I/3

LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIG. NO.** | **TITLE** | **PAGE NO.** |
| 5.1 | Use Case Diagram | 23 |
| 8.2.1 | Login and Signup page | 48 |
| 8.2.2 | Signup Details and home page of app | 48 |
| 8.2.3 | Profile Details and Lab details | 49 |
| 8.2.4 | Whether Report and Soil Report Details | 49 |
| 8.2.5 | Lab Booking Details and Soil Test Tracker | 50 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |



SISTec/BTech/CS/2024/5/MajorProject\_I/3

**CHAPTER-1 INTRODUCTION**

###### ABOUT PROJECT

The **“Automated System for Career Advancements of the Faculties of Higher Education”** is a web-based solution developed to modernize and simplify the traditional faculty self-appraisal process used in universities and higher education institutions. The project aims to eliminate the inefficiencies of paper-based systems by providing a digital platform that enables real-time tracking, submission, and evaluation of faculty academic and professional activities.

Faculty members can log in securely and submit their achievements, such as research publications, seminars attended, projects handled, lectures delivered, and other academic contributions through a structured self-appraisal form. The system ensures that all the data is stored safely in a centralized database, easily retrievable for evaluation purposes.

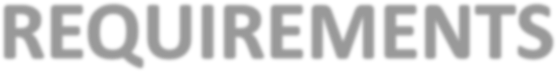
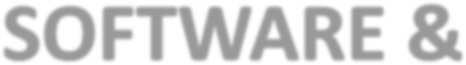
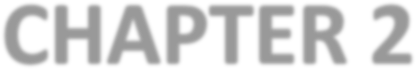
In addition to the faculty interface, the system includes a dedicated **Admin Panel** where authorized university officials can view all submissions, sort them by name, employee code, or date, and download them in PDF format for reporting and record-keeping. The platform also supports automated tracking of research publications by integrating with services like Google Scholar.

By implementing this system, the project not only ensures transparency and efficiency in faculty evaluations but also aligns with the broader objectives of digital transformation in education and sustainable paperless administration. It empowers both faculty and administrators to focus on professional growth, strategic planning, and quality enhancement in academic environments.

###### PROJECT OBJECTIVES

The primary objective of the **“ Career Advancements of the Faculties of Higher Education”** is to develop a secure, efficient, and user-friendly web-based platform that simplifies and digitizes the faculty self-appraisal process. The system aims to achieve the following key goals:

1. **Digitize Faculty Self-Appraisal**  
   Replace traditional paper-based appraisal forms with a digital, real-time submission system for academic activities and achievements.
2. **Enhance Transparency and Accuracy**  
   Provide a transparent system that ensures accurate recording and tracking of faculty contributions like publications, seminars, projects, and lectures.
3. **Automate Data Management**  
   Maintain a centralized and secure database that stores faculty records systematically, reducing manual errors and administrative burden.
4. **Simplify Administrative Evaluation**  
   Enable administrators to access, sort, filter, and download appraisal data efficiently through an interactive admin panel.
5. **Integrate Real-Time Updates**  
   Allow faculty members to update their activities anytime, ensuring that appraisal data is always up-to-date and reflective of current contributions.
6. **Support Paperless Initiatives**  
   Contribute to the “Paperless India” mission by reducing dependency on printed documents and physical filing systems.
7. **Improve Decision-Making**  
   Provide structured and consolidated reports to help university authorities make informed decisions regarding faculty promotions, rewards, and development.
8. **Ensure Data Security**  
   Implement secure login systems and access controls to protect sensitive user data and ensure only authorized access.



**CHAPTER-2 SOFTWARE AND HARDWARE REQUIREMENTS**

###### SOFTWARE REQUIREMENTS

The software components used in the Fertilizer Usage Optimizer are essential for analyzing agricultural data, generating recommendations, and providing an interactive and user- friendly platform for farmers.

The specific requirements are as follows:

* + 1. **Development Tools**
       - **Frontend Development:**
         * Flutter SDK
         * IDE: VS Code/ Android Studio/ IntelliJ IDE
       - **Backend Development:**
         * Node.js and Express.js
         * API Testing Tool: Postman
    2. **APIs:**
       - OpenWeather API: For weather patterns and forecast data.
       - Razorpay API: For payment gateway
    3. **User Requirements**
       - Android 6.0 (Marshmallow) or higher
       - iOS 12.0 or higher
    4. **MONGODB**

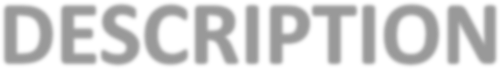
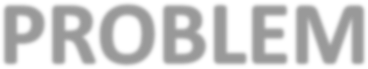
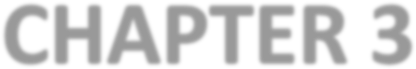
MongoDB is used to store dynamic data like soil reports, weather updates, user profiles, and recommendations. Its scalability and fast querying enable efficient data management and data management and real-time updates for farmers.

###### HARDWARE REQUIREMENTS

The hardware components ensure smooth functioning, development, and usage of the app. The specific requirements are as follows:

* + 1. **Developer Machines**
       - Processor : Quad-core or higher
       - RAM : Minimum 8 GB
       - Storage : 256 GB
       - Graphics Card : Integrated or dedicated GPU
       - Operating System : Windows 10/11, macOS
    2. **Test Devices**
       - Android Device: Smartphone with Android 9+
       - iOS: iPhone with iOS 13+
    3. **User Requirements**
       - Device: Smartphone or Tablet (Android/iOS)
       - Processor: Dual-core or higher
       - RAM: 2GB OR higher

-



**CHAPTER-3 PROBLEM DESCRIPTION**

###### INTRODUCTION

With In many higher education institutions, the faculty self-appraisal process is still conducted using manual and paper-based systems. These traditional methods are often inefficient, time-consuming, and prone to human error, leading to delays, lack of transparency, and poor documentation of faculty contributions. Due to the absence of a centralized and automated system, tracking academic achievements such as research publications, seminars, workshops, and project involvements becomes a cumbersome task, hindering fair and timely career advancements.

Moreover, faculty members find it difficult to maintain organized records of their academic and professional growth, while administrators face challenges in evaluating faculty performance accurately and promptly. These limitations also restrict the institution’s ability to make data-driven decisions regarding promotions, recognitions, and resource allocations.

There is a strong need for a robust, secure, and real-time web-based system that can automate and digitize the entire faculty self-appraisal process. Such a system must enable faculty to easily log their achievements and administrators to access, sort, and evaluate data efficiently. This project addresses these challenges by developing a scalable and user-friendly platform aimed at enhancing the quality, transparency, and efficiency of faculty career advancement mechanisms in higher education institutions.

###### PROBLEM STATEMENT

Farmers currently face the following challenges:

* + - **Inaccessible to soil testing services**: Difficulty in locating nearby soil testing labs, Lack of convenient ways to collect and submit soil samples.
    - **Generalized Fertilizer Recommendations**: Fertilizer advice is based on assumptions, lacking specificity for individual farm conditions, leading to overuse or underuse of nutrients.
    - **Environment and Economic Concerns**: Improper fertilizer usage depletes soil quality, increases costs, and causes environmental harm like chemical runoff and water pollution.
    - **Lack of Automation and Integration**: No system integrates soil testing labs, report analysis, and tailored recommendations in a seamless and automated manner.

###### OBJECTIVES OF THE PROJECT

The primary objectives of this project are as follows:

**To Digitize Faculty Self-Appraisal Processes**

Eliminate manual and paper-based self-appraisal systems by introducing a secure and efficient web-based platform

**To Enable Real-Time Activity Logging**

Allow faculty members to input and update their academic and professional activities — such as publications, seminars, workshops, and lectures — in real-time

**To Provide a Secure Authentication System**

Ensure secure registration and login for both faculty members and administrators to protect confidential academic data

**To Offer a User-Friendly Interface**

Develop a simple and intuitive user interface for easy navigation and data entry, minimizing training requirements.

**To Facilitate Efficient Data Management and Retrieval**

Store appraisal data securely in a centralized database and enable fast, structured access for administrative review

**To Support Informed Decision-Making**

Help administrators evaluate faculty performance based on organized, accurate, and up-to-date data for promotions and resource allocation.

**To Promote Transparency and Accountability**

Foster a transparent system that gives both faculty and administrators a clear view of academic contributions and progress.

**To Align with Sustainable and Paperless Initiatives**

Contribute to the "Paperless India" initiative by reducing the need for physical documentation and supporting digital transformation in academia.

###### 3.4 CHALLENGES AND CONSTRAINTS

Despite the significant advantages of developing an automated faculty appraisal system, the project may face the following challenges and constraints

**Data Accuracy and Completeness**

Ensuring that faculty members enter complete and accurate data for all their activities is crucial. Incomplete or incorrect entries may affect performance evaluations.

**User Adoption Resistance**

Faculty members and administrators accustomed to traditional methods may initially resist shifting to a digital system, requiring orientation and training.

**Integration with Existing Systems**

Integrating the platform with other existing university systems (such as ERP, publication databases like Google Scholar, etc.) could be technically challenging.

**Security and Privacy Concerns**

Handling sensitive academic and personal data requires strong encryption and secure access control to prevent data breaches.

**Scalability for Large Institutions**

The system must be able to handle a large number of users and data entries without performance issues, especially in universities with a large faculty base.

**Real-Time Data Processing**

Processing real-time inputs and updating the database instantly while maintaining consistency and performance may require efficient backend implementation.

**Limited Technical Infrastructure**

Some institutions may lack the technical infrastructure (e.g., high-speed internet, updated browsers) necessary for optimal use of the platform.

**Platform Compatibility**

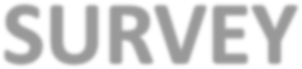
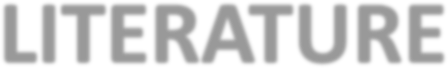
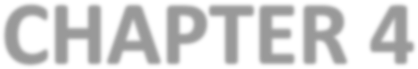
Ensuring compatibility across multiple browsers and devices can be challenging but is necessary for universal accessibility.

**Maintenance and Updates**

Regular updates, bug fixes, and technical support will be required to keep the platform functional and up to date with changing academic processes.

**Budget and Time Constraints**

Developing and deploying a comprehensive and secure platform within a limited time frame and budget can pose a significant challenge.



**CHAPTER-4 LITERATURE SURVEY**

###### INTRODUCTION TOTHE DOMAIN

The domain of this project lies within Higher Education Management Systems, specifically focusing on Academic Human Resource Development. In modern academic institutions, maintaining the quality of education and encouraging faculty growth are essential for institutional success. Faculty appraisal is a key element of this process, as it helps evaluate teaching effectiveness, research contributions, participation in academic events, and overall engagement in institutional development.

Traditionally, faculty self-appraisal has been a manual, paper-based process that is often time-consuming, inconsistent, and prone to errors. With the advancement of digital technologies and the growing emphasis on paperless workflows, data-driven evaluation, and performance transparency, there is an urgent need to automate and modernize these processes.

This domain combines web development, database management, user authentication, and real-time data tracking to create efficient systems that streamline academic performance evaluations. By developing a secure and scalable web-based platform, this project contributes to digital transformation in higher education institutions, aligning with broader goals such as Digital India and academic excellence.

###### 

###### 4.2Current Technological Interventions

Programs In the domain of faculty management and appraisal systems in higher education, several technological interventions have emerged to streamline and digitize the appraisal process. These interventions aim to enhance transparency, accuracy, and efficiency in tracking faculty activities and contributions.

**1. Enterprise Resource Planning (ERP) Systems**

Many universities and colleges have adopted ERP systems such as TCS iON, PeopleSoft, or SAP-based platforms. These systems help manage various academic and administrative functions, including limited support for faculty appraisals. However, most ERP solutions are complex, costly, and not specifically tailored for comprehensive academic performance evaluation.

**2. Google Scholar & Academic Databases**

Platforms like Google Scholar, Scopus, and Web of Science automatically track research publications, citations, and author metrics. These tools are widely used by institutions for assessing research output but do not integrate well with broader appraisal system

**3. LMS Integration**

Learning Management Systems (LMS) like Moodle, Blackboard, or Canvas offer partial support for tracking teaching activities, course content delivery, and student engagement. While useful, these platforms do not cover a full range of faculty responsibilities such as event participation, seminars, and project involvement.

**4. Digital Forms and Cloud Storage**

Some institutions use tools like Google Forms, Microsoft Forms, and Excel Sheets stored in Google Drive or OneDrive for collecting self-appraisal data. While easy to implement, these solutions lack security, standardization, real-time analysis, and integrated admin dashboards.

**5. Custom Web-Based Portals**

A few universities have developed in-house web portals for faculty data management. These systems vary in functionality and often lack automation, real-time data integration, or scalability.

###### Tools and Systems Already Available

* + - **Soil Health Card (Government of India):**
      * A nationwide initiative where farmers receive soil health reports.
      * **Limitation:** Reports are static, generalized, and do not provide real-time updates or dynamic feedback.
    - **Agricultural Mobile Apps:**
      * Tools like **IFFCO Kisan**, **CropIn.**, and **RML** Farmer offer basic features such as weather updates, crop prices, and general agricultural advice.
      * **Limitation:** These apps lack integration with soil testing labs and do not analyze soil reports for personalized recommendations.
    - **Precision Agriculture Platforms:**
      * High-end tools like **Trimble Agriculture** and **John Deere Precision Tools**

use sensors, IoT devices, and satellite data for resource optimization.

* + - * **Limitation:** These platforms are expensive and primarily cater to large-scale farms, leaving small and medium farmers underserved.

###### ENVIRONMENTAL IMPACT AND POLICY CONSIDERATIONS

The app minimizes environmental harm by optimizing fertilizer use, reducing soil degradation, and preventing chemical runoff into water sources. It aligns with government policies on precision farming and soil health management while supporting global sustainability goals, such as reducing the carbon footprint and promoting responsible agricultural practices.

###### CHALLENGES AND FUTURE DIRECTIONS

The development and deployment of the “Automated System for Career Advancements of the Faculties of Higher Education” come with several challenges and opportunities for future growth. One of the primary challenges is ensuring the accuracy and authenticity of the data submitted by faculty members, especially regarding research publications, seminar participation, and project involvement. Manual data entry may lead to errors or inconsistencies, necessitating robust verification mechanisms. Another significant challenge lies in integrating the system with external platforms such as Google Scholar, Scopus, or institutional databases, which often require technical know-how and API-level access permissions. Additionally, user adoption can be a concern, particularly among faculty members who may not be familiar with digital platforms. Hence, designing a user-friendly interface and offering proper training and support becomes critical.

Security is also a major concern, as the system handles confidential academic and personal data. Implementing strong authentication protocols, encrypted storage, and access controls is essential to ensure data privacy. Furthermore, as the system scales to accommodate more users and institutions, maintaining performance and ensuring low latency becomes technically demanding. Administrative workload, despite automation, still exists in the form of data reviews and approvals, requiring efficient backend tools and possibly AI assistance in the future.

Looking forward, several future directions can enhance the effectiveness of the platform. Incorporating AI and machine learning can provide insightful analytics, such as performance trends or promotion-readiness suggestions. The development of a mobile application can significantly increase accessibility and ease of use. Automated tracking of publications by integrating with academic databases would further streamline data entry. Features like gamification can help increase user engagement, while multilingual support and accessibility options will ensure inclusivity for diverse faculty groups. Additionally, scaling the system to serve multiple institutions with customized modules could position it as a comprehensive and standard solution across higher education systems. These enhancements aim to make the appraisal process more dynamic, transparent, and aligned with modern academic and administrative needs.

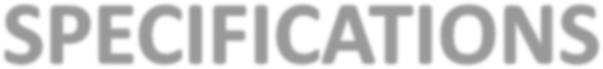
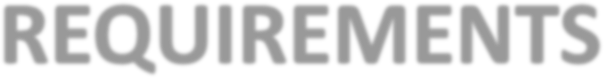
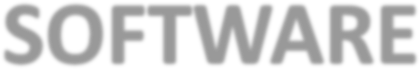
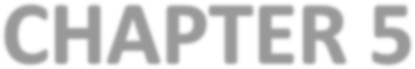
**4.4 CONCLUSION**

Sustainable The project **" Career Advancements of the Faculties of Higher Education"** addresses the pressing need to modernize and streamline the faculty self-appraisal process in academic institutions. Traditional methods of evaluation, often paper-based and inefficient, create bottlenecks in tracking faculty contributions and hinder transparency. This project proposes a web-based solution that integrates secure login systems, dynamic data entry forms, automated activity tracking, and a centralized admin panel—allowing for seamless monitoring, evaluation, and documentation of faculty achievements.

By enabling real-time updates and providing tools for efficient data analysis and report generation, the system empowers both faculty members and university administrators. It not only fosters a culture of continuous improvement and professional development but also contributes to the national vision of a **paperless and digital India**. The platform, with its scalable and secure architecture, lays the foundation for future enhancements, including AI-driven analytics and broader institutional integrations.

In conclusion, this project presents a comprehensive and sustainable solution to an existing academic challenge, reinforcing transparency, efficiency, and accountability in faculty appraisal processes. It serves as a significant step toward transforming the higher education landscape through smart and innovative digital interventions.

IoT integration and AI-driven analytics, the app has the potential to further transform agricultural practices, ensuring higher yields and a sustainable future for farming communities.



**CHAPTER-5** **SOFTWARE REQUIREMENTS SPECIFICATION**

###### FUNCTIONAL REQUIREMENTS

The functional requirements of the Fertilizer Usage Optimizer app outline the core functionalities the system must deliver to meet farmers' needs and project objectives. These requirements define the operations the app is expected to perform, ensuring it provides accurate fertilizer recommendations, integrates soil testing, delivers weather updates, and supports user-friendly interaction to help farmers optimize resources, improve crop yield, and promote sustainable agricultural practices.

* + 1. **User Registration and Authentication**

The app must allow farmers to create accounts, log in, and manage profiles securely.

* + 1. **Weather Data Integration**

The app will integrate with third-party weather APIs (e.g., OpenWeather) to provide real- time and forecasted weather data. This will help farmers make informed decisions about fertilizer application by considering factors such as temperature, humidity, rainfall, and wind speed. The weather information will be used to adjust fertilizer recommendations, ensuring optimal usage and improved crop yield.

* + 1. **Soil Test Scheduling**

The app must enable farmers to schedule soil testing appointments with laboratories and view available testing services.

* + 1. **Soil Report Upload & Analyses**

Farmers will be able to upload their soil test reports directly into the app. The app will process these reports and provide personalized fertilizer recommendations based on the soil composition, helping farmers optimize fertilizer usage for improved soil health and crop yield.

* + 1. **Appointment Management**

The app will allow farmers to easily manage their soil testing appointments by enabling them to schedule, confirm, and track their appointments with local soil testing laboratories. Farmers can view available testing slots, select a preferred date and time, and receive confirmation notifications once their appointments are booked. Additionally, the app will send reminders to ensure the farmer does not miss the appointment. Farmers will also be able to track the status of their appointments, such as whether the test is in progress or the results are ready. This feature streamlines the process of soil testing and ensures that farmers can access timely and accurate data to make informed decisions regarding fertilizer use.

* + 1. **USER EXPERIENCE**

The user interface of the web page must be designed with a focus on **user experience** (UX). It should be easy to navigate and intuitive, allowing users to:

These functional requirements ensure that the **Fertilizer Usage Optimizer** app is comprehensive, user-friendly, and effective in providing essential tools for farmers to optimize fertilizer usage, manage soil testing appointments, and receive personalized recommendations based on weather and soil data, ultimately improving crop yields and promoting sustainable farming practices.

###### NON-FUNCTIONAL REQUIREMENTS

The *Automated System for Career Advancements of the Faculties of Higher Education* is designed to meet several key non-functional requirements. Performance is a priority, ensuring fast response times for data retrieval and form submissions, even under high user load. Scalability is essential, enabling the system to accommodate increasing users and data without performance degradation. Reliability is crucial, with minimal downtime to ensure continuous access and functionality. Security is implemented through strong encryption and access controls to protect sensitive faculty data, ensuring privacy. Additionally, the system prioritizes usability with an intuitive interface that requires minimal training

**PERFORMANCE**

The performance of the *Automated System for Career Advancements of the Faculties of Higher Education* is a critical non-functional aspect that directly impacts user satisfaction and system efficiency. The system is designed to ensure fast response times during form submissions, data retrieval, and admin-side operations, even under high user load. It should be optimized to handle concurrent requests without causing delays or timeouts. Real-time updates of faculty activities such as research publications, event participation, and seminars must be reflected instantly in the system. Efficient database queries, optimized backend services, and lightweight frontend components ensure smooth and uninterrupted interactions. This high-performance capability supports quick decision-making and seamless usage for both faculty members and university administrators, even during peak usage times such as appraisal periods

**RELIABILITY**

Reliability ensures the system operates consistently with minimal downtime. “**Career Advancements”**  will feature robust error handling, automatic recovery mechanisms, and continuous monitoring to prevent disruptions and data loss, ensuring stable performance over time.

* + 1. **USABILITY**

The Usability focuses on creating an intuitive and user-friendly interface, allowing faculty and administrators to navigate the system with ease. The system should require minimal training, ensuring a smooth experience for all users.

* + 1. **SECURITY**

User Security ensures the protection of sensitive faculty data through encryption, secure user authentication, and strict access controls. The system will safeguard user privacy and prevent unauthorized access, maintaining data integrity.

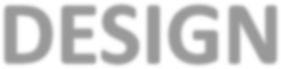
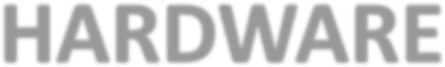
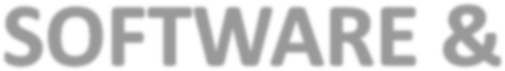
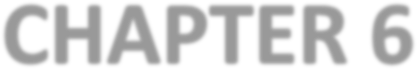
**MAINTAINABILITY**

User Maintainability ensures the system can be easily updated, modified, and fixed. It will feature clear documentation and modular architecture to simplify troubleshooting, updates, and enhancements over time.

* + 1. **COMPLIANCE**

The app must comply with local agricultural regulations and standards regarding fertilizer usage and soil testing practices.

These non-functional requirements ensure that the app is not only effective in optimizing fertilizer use and improving crop yields but also operates with high quality, security, usability and performance standards, meeting the expectations of farmers, soil testing labs.



**CHAPTER-6 SOFTWARE AND HARDWARE DESIGN**

###### USE CASE DIAGRAM

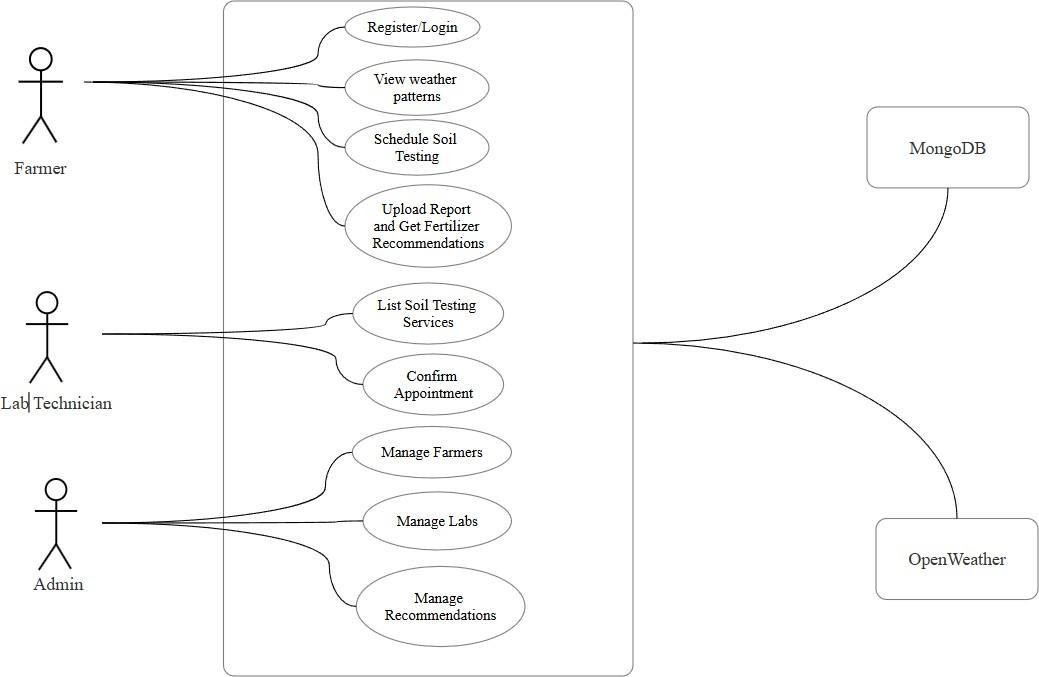
The use case diagram for the **Sustainable Fertilizer Usage Optimizer for Higher Yield** app demonstrates the interaction between various actors and the system, highlighting the key functionalities and how they connect to external entities. Below is an outline of the use case diagram:

**Key Elements:**

* + - **Actors:** Represent users or other systems interacting with the app.
    - **Use Cases:** Denote the actions or services the system performs for the user.

**Description:**

The primary actor in this system is the **Farmer**, who interacts with the mobile application to manage their farming needs. The **Farmer** can view weather patterns, select the appropriate crop type, schedule soil testing, and receive recommendations based on the analysis of weather, soil data, and crop type. The **Soil Testing Labs** provide testing services and confirm appointments with farmers. **Admins** oversee the management of user data, soil lab listings, and fertilizer recommendations to ensure accurate information is available to farmers. Below is the Use-Case diagram:-



**Figure 5.1: Use Case Diagram**

###### ARCHITECTURE

The Farmer's Application is designed with a high-level client-server architecture, where the frontend (mobile application) interacts with the backend services through RESTful APIs. The key components include:

1. **Client (Frontend):**

The mobile application (built with flutter) serves as the user interface for farmers, lab technician, and admins.

1. **Backend:**

The backend system handles API endpoints, and communicate with third party services.

1. **Database:**

MySQL (or Hive for large-scale local data) stores user information, soil reports, weather data, and payment details.

1. **APIs:**

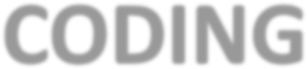
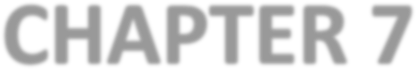
Interaction with OpenWeather API (for weather data) and Razorpay API (for payments)

1. **AI/ML Engine:**

A rule based or decision-tree system processes soil and weather data to generate farming recommendations.

**Data Flow:**

* 1. User interact with the mobile app.
  2. The backend processes the request, queries the database, and/or interacts with APIs.
  3. Processed data is sent back to the client for display.
  4. User notifications are triggered as required.



**CHAPTER-7 CODING**

###### 8.1 CODE IMPLEMENTATION:

The "Krishak Sathi" app is a comprehensive agricultural platform that begins with a user login. Once logged in, the app utilizes the user's location to identify nearby soil testing labs. Farmers can select a preferred lab, proceed with payment, and book a soil test. The app then tracks the progress of the test, providing real-time updates. Upon completion, the app delivers a detailed report directly to the user, highlighting specific nutrient deficiencies or excesses in the soil. This information empowers farmers to make informed decisions about fertilizer application, ensuring optimal crop health and yield.

### Main.dart

import 'package:flutter/material.dart';

import 'package:krishak\_sathi/Home\_Page.dart'; import 'package:krishak\_sathi/Labs/Confirm\_adress.dart'; import 'package:krishak\_sathi/lab.dart';

import 'package:krishak\_sathi/login\_page.dart'; import 'package:krishak\_sathi/otp\_page.dart'; import 'package:krishak\_sathi/profile.dart'; import

'package:krishak\_sathi/report/multiple\_soil\_reports.dart'; import 'package:krishak\_sathi/router.dart';

import 'package:krishak\_sathi/tracker/test\_tracker.dart'; import 'crops/Crops.dart';

import 'report/report\_analize.dart';

void main() { runApp(const MyApp());

}

class MyApp extends StatefulWidget { const MyApp({super.key});

@override

State<MyApp> createState() => MyAppState();

}

class MyAppState extends State<MyApp> { @override

Widget build(BuildContext context) { return MaterialApp( debugShowCheckedModeBanner: false, home: Login(),

onGenerateRoute: generateRoute,

);

}

}

### Home\_page.dart

import 'package:flutter/material.dart';

import 'package:krishak\_sathi/Home\_info.dart'; import 'package:krishak\_sathi/lab.dart'; import 'package:krishak\_sathi/profile.dart';

import 'package:krishak\_sathi/weather\_pages/weather\_pages.dart';

class GobalColors {

static const PrimaryColor = Color(0xFF2DB83D); // Define primary color

}

class HomePage extends StatefulWidget { const HomePage({super.key});

@override

State<HomePage> createState() => \_HomePageState();

}

class \_HomePageState extends State<HomePage> {

// Pages for navigation

final List<Widget> pages = [ LabsPage(), // Replace with LabStatus() HomePageInfo(),

FarmerProfilePage(), // Replace with ProfilePage() WeatherScreen(), // Replace with WeatherPage()

];

int currentIndex = 1; // Default to Home Page @override

Widget build(BuildContext context) { return Scaffold(

body: IndexedStack(

index: currentIndex, // Control which page is displayed children: pages, // Pages to navigate between

),

bottomNavigationBar: Container(

margin: const EdgeInsets.all(15), decoration: BoxDecoration(

color: Colors.white, // White background for floating effect borderRadius: BorderRadius.circular(30),

boxShadow: [ BoxShadow(

color: Colors.black.withOpacity(0.2), blurRadius: 10,

offset: const Offset(0, 5),

),

],

),

child: ClipRRect(

borderRadius: BorderRadius.circular(30), child: BottomNavigationBar(

type:

BottomNavigationBarType.fixed, // Ensure background color works backgroundColor: Colors.white, // Match the container background currentIndex: currentIndex,

selectedItemColor:

GobalColors.PrimaryColor, // Green for selected items unselectedItemColor:

const Color.fromARGB(255, 0, 0, 0), // Grey for unselected items showUnselectedLabels: true,

selectedFontSize: 14,

unselectedFontSize: 12, onTap: (value) { setState(() {

currentIndex = value; // Update the index on tap

});

},

items: const [ BottomNavigationBarItem( icon: Icon(Icons.science), label: 'Lab',

),

BottomNavigationBarItem( icon: Icon(Icons.home), label: 'Home',

),

BottomNavigationBarItem( icon: Icon(Icons.person), label: 'Profile',

),

BottomNavigationBarItem( icon: Icon(Icons.cloud),

label: 'Weather',

),

],

),

),

),

);

}

}

### Signup\_page.dart

import 'package:flutter/material.dart'; class SignUp extends StatelessWidget {

static const String routeName = '/login-screen'; const SignUp({super.key});

@override

Widget build(BuildContext context) {

// Controller to capture the mobile number input

final TextEditingController mobileNumberController = TextEditingController();

return MaterialApp( debugShowCheckedModeBanner: false, home: Scaffold(

body: SingleChildScrollView( child: Column(

children: [

// Top section with a modern gradient background Container(

height: 450,

width: double.infinity, decoration: const BoxDecoration(

borderRadius: BorderRadius.only( bottomLeft: Radius.circular(40), bottomRight: Radius.circular(40),

),

gradient: LinearGradient(

colors: [Color(0xFF2DB83D), Color(0xFF28A745)], begin: Alignment.topLeft,

end: Alignment.bottomRight,

),

),

child: const Column(

mainAxisAlignment: MainAxisAlignment.center, children: [

Icon( Icons.person,

size: 80,

color: Colors.white,

),

SizedBox(height: 10), Text(

"Welcome", style: TextStyle(

color: Colors.white, fontSize: 24,

fontWeight: FontWeight.bold,

),

),

],

),

),

// Login header Container(

margin: const EdgeInsets.symmetric(vertical: 20), child: const Text(

"SignIn",

style: TextStyle(

fontWeight: FontWeight.bold, fontSize: 24,

color: Color(0xFF333333),

),

),

),

// Input fields and button Padding(

padding: const EdgeInsets.symmetric(horizontal: 20), child: Column(

crossAxisAlignment: CrossAxisAlignment.start, children: [

const Text(

"Enter Mobile Number", style: TextStyle( fontSize: 16,

fontWeight: FontWeight.w600,

),

),

const SizedBox(height: 5), const Text(

"Please confirm your mobile number to proceed", style: TextStyle(

fontSize: 12, color: Colors.grey,

),

),

const SizedBox(height: 20),

// Mobile number input field Card(

elevation: 6,

shadowColor: Colors.grey.withOpacity(0.3), shape: RoundedRectangleBorder( borderRadius: BorderRadius.circular(12),

),

child: TextField(

controller: mobileNumberController, keyboardType: TextInputType.phone, decoration: InputDecoration( hintText: "+91 | Mobile number", filled: true,

fillColor: Colors.white,

contentPadding: const EdgeInsets.symmetric( vertical: 16, horizontal: 20),

border: OutlineInputBorder( borderRadius: BorderRadius.circular(12), borderSide: BorderSide.none,

),

suffixIcon: const Icon( Icons.phone,

color: Color(0xFF2DB83D),

),

),

),

),

const SizedBox(height: 20),

// Continue button ElevatedButton( onPressed: () {

// Validate the mobile number

if (mobileNumberController.text == '8423304966') { Navigator.pushReplacementNamed(

context, '/farmer\_details');

} else {

// Show an error message if the number doesn't match ScaffoldMessenger.of(context).showSnackBar(

const SnackBar( content: Text(

"Invalid mobile number. Please try again.", style: TextStyle(color: Colors.white),

),

backgroundColor: Colors.red,

),

);

}

},

style: ElevatedButton.styleFrom( backgroundColor: const Color(0xFF2DB83D), shape: RoundedRectangleBorder( borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50), elevation: 6,

),

child: const Text( "Continue", style: TextStyle( fontSize: 18,

fontWeight: FontWeight.w600, color: Colors.white,

),

),

),

const SizedBox(height: 20),

// Alternative login link Center(

child: TextButton( onPressed: () {

Navigator.pushReplacementNamed(context, '/login');

},

child: const Text( "Login",

style: TextStyle(

color: Color(0xFF2DB83D), fontSize: 16,

fontWeight: FontWeight.w600,

),

),

),

),

],

),

),

],

),

),

),

);

}

}

### Login\_page.dart

import 'package:flutter/material.dart'; class Login extends StatelessWidget {

static const String routeName = '/login-screen'; const Login({super.key});

@override

Widget build(BuildContext context) {

// Controller to capture the mobile number input

final TextEditingController mobileNumberController = TextEditingController();

return MaterialApp( debugShowCheckedModeBanner: false, home: Scaffold(

body: SingleChildScrollView( child: Column(

children: [

// Top section with a modern gradient background Container(

height: 450,

width: double.infinity, decoration: const BoxDecoration(

borderRadius: BorderRadius.only( bottomLeft: Radius.circular(40), bottomRight: Radius.circular(40),

),

gradient: LinearGradient(

colors: [Color(0xFF2DB83D), Color(0xFF28A745)], begin: Alignment.topLeft,

end: Alignment.bottomRight,

),

),

child: const Column(

mainAxisAlignment: MainAxisAlignment.center, children: [

Icon( Icons.person, size: 80,

color: Colors.white,

),

SizedBox(height: 10), Text(

"Welcome Back", style: TextStyle( color: Colors.white, fontSize: 24,

fontWeight: FontWeight.bold,

),

),

],

),

),

// Login header Container(

margin: const EdgeInsets.symmetric(vertical: 20), child: const Text(

"Login",

style: TextStyle(

fontWeight: FontWeight.bold, fontSize: 24,

color: Color(0xFF333333),

),

),

),

// Input fields and button Padding(

padding: const EdgeInsets.symmetric(horizontal: 20), child: Column(

crossAxisAlignment: CrossAxisAlignment.start, children: [

const Text(

"Enter Mobile Number", style: TextStyle( fontSize: 16,

fontWeight: FontWeight.w600,

),

),

const SizedBox(height: 5), const Text(

"Please confirm your mobile number to proceed", style: TextStyle(

fontSize: 12, color: Colors.grey,

),

),

const SizedBox(height: 20),

// Mobile number input field Card(

elevation: 6,

shadowColor: Colors.grey.withOpacity(0.3), shape: RoundedRectangleBorder( borderRadius: BorderRadius.circular(12),

),

child: TextField(

controller: mobileNumberController, keyboardType: TextInputType.phone, decoration: InputDecoration( hintText: "+91 | Mobile number", filled: true,

fillColor: Colors.white,

contentPadding: const EdgeInsets.symmetric( vertical: 16, horizontal: 20),

border: OutlineInputBorder( borderRadius: BorderRadius.circular(12), borderSide: BorderSide.none,

),

suffixIcon: const Icon( Icons.phone,

color: Color(0xFF2DB83D),

),

),

),

),

const SizedBox(height: 20),

// Continue button ElevatedButton( onPressed: () {

// Validate the mobile number

if (mobileNumberController.text == '8423304966') { Navigator.pushReplacementNamed(context, '/homePage');

} else {

// Show an error message if the number doesn't match ScaffoldMessenger.of(context).showSnackBar(

const SnackBar( content: Text(

"Invalid mobile number. Please try again.", style: TextStyle(color: Colors.white),

),

backgroundColor: Colors.red,

),

);

}

},

style: ElevatedButton.styleFrom( backgroundColor: const Color(0xFF2DB83D), shape: RoundedRectangleBorder( borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50), elevation: 6,

),

child: const Text( "Continue", style: TextStyle( fontSize: 18,

fontWeight: FontWeight.w600, color: Colors.white,

),

),

),

const SizedBox(height: 20),

// Alternative login link Center(

child: TextButton( onPressed: () {

Navigator.pushReplacementNamed(context, '/signup');

},

child: const Text( "SignIn",

style: TextStyle(

color: Color(0xFF2DB83D), fontSize: 16,

fontWeight: FontWeight.w600,

),

),

),

),

],

),

),

],

),

),

),

);

}

}

### Home\_info.dart

import 'package:flutter/material.dart';

import 'package:carousel\_slider/carousel\_slider.dart'; import 'package:krishak\_sathi/constant.dart';

class HomePageInfo extends StatefulWidget { const HomePageInfo({super.key});

@override

State<HomePageInfo> createState() => \_HomePageInfoState();

}

class \_HomePageInfoState extends State<HomePageInfo> {

// List of images for the carousel final List<String> imageList = [ 'assets/images/farmer1.jpg', 'assets/images/farmer2.jpg', 'assets/images/farmer3.jpg',

];

// List of news articles related to agriculture and farmers final List<Map<String, String>> newsArticles = [

{

'title': 'Government Scheme for Farmers', 'image': 'assets/images/news1.jpg', 'description':

'This scheme helps farmers with financial aid for better crop production.',

},

{

'title': 'New Agricultural Technology', 'image': 'assets/images/news2.jpg', 'description':

'Learn about new agricultural technology making farming more efficient.',

},

{

'title': 'Farming Tips for Winter',

'image': 'assets/images/news3.jpg', 'description':

'Stay prepared with these essential farming tips for the winter season.',

},

];

@override

Widget build(BuildContext context) { return Scaffold(

appBar: AppBar( title: const Text( 'Krishak Sathi', style: TextStyle(

fontWeight: FontWeight.w600, fontSize: 22, color: Colors.white),

),

backgroundColor: GobalColors.PrimaryColor, elevation: 5,

centerTitle: true,

),

body: ListView(

padding: const EdgeInsets.all(10), children: [

// Carousel with farmer images CarouselSlider(

options: CarouselOptions( height: 250,

autoPlay: true, enlargeCenterPage: true,

autoPlayInterval: const Duration(seconds: 4), aspectRatio: 16 / 9,

enableInfiniteScroll: true, viewportFraction: 0.85,

),

items: imageList

.map((item) => ClipRRect(

borderRadius: BorderRadius.circular(15), child: Stack(

fit: StackFit.expand, children: [

Image.asset(item, fit: BoxFit.cover), Container(

decoration: BoxDecoration( gradient: LinearGradient( colors: [

Colors.black.withOpacity(0.6),

Colors.transparent,

],

begin: Alignment.bottomCenter, end: Alignment.topCenter,

),

),

), // Gradient overlay Positioned(

bottom: 10,

left: 10, child: Text(

'Inspiring Farmer Stories', style: TextStyle( fontSize: 16,

color: Colors.white, fontWeight: FontWeight.bold, shadows: [

Shadow(

color: Colors.black.withOpacity(0.8), blurRadius: 5,

)

],

),

),

),

],

),

))

.toList(),

),

// Section heading

const SizedBox(height: 20), const Padding(

padding: EdgeInsets.symmetric(horizontal: 10), child: Text(

'News Related to Agriculture and Farmers', style: TextStyle(

fontSize: 24,

fontWeight: FontWeight.bold, color: GobalColors.PrimaryColor,

),

),

),

const SizedBox(height: 15),

// News article cards with enhanced design

...newsArticles.map((article) { return Card(

margin: const EdgeInsets.symmetric(vertical: 10, horizontal: 15),

elevation: 8,

shape: RoundedRectangleBorder( borderRadius: BorderRadius.circular(20),

),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start, children: [

// News article image with rounded top corners ClipRRect(

borderRadius:

const BorderRadius.vertical(top: Radius.circular(20)), child: Image.asset(

article['image']!, height: 180,

width: double.infinity, fit: BoxFit.cover,

),

),

// News title Padding(

padding: const EdgeInsets.all(16.0), child: Text(

article['title']!,

style: const TextStyle( fontSize: 20,

fontWeight: FontWeight.w600, color: Colors.black87,

),

),

),

// News description Padding(

padding: const EdgeInsets.symmetric( horizontal: 16.0, vertical: 8.0),

child: Text( article['description']!, style: const TextStyle( fontSize: 14,

color: Colors.black54,

),

),

),

// Read more button Align(

alignment: Alignment.centerRight,

child: TextButton( onPressed: () {

// Placeholder for "Read More" functionality

},

child: const Text( 'Read More',

style: TextStyle(color: Colors.green),

),

),

),

],

),

);

}).toList(),

],

),

);

}

}

### Farmers\_details.dart

import 'package:flutter/material.dart';

class FarmerDetailPage extends StatelessWidget { static const String routeName = '/farmer-detail-page'; const FarmerDetailPage({super.key});

@override

Widget build(BuildContext context) { return MaterialApp( debugShowCheckedModeBanner: false, home: Scaffold(

appBar: AppBar(

backgroundColor: const Color(0xFF2DB83D), title: const Text(

"Farmer Details",

style: TextStyle(color: Colors.white),

),

centerTitle: true,

),

body: SingleChildScrollView( child: Padding(

padding: const EdgeInsets.all(20.0), child: Column(

crossAxisAlignment: CrossAxisAlignment.start, children: [

const SizedBox(height: 20),

// Input for Name const Text( "Name",

style: TextStyle(fontWeight: FontWeight.bold, fontSize: 16),

),

const SizedBox(height: 10), TextField(

decoration: InputDecoration( hintText: "Enter your name", border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

),

filled: true,

fillColor: Colors.white,

),

),

const SizedBox(height: 20),

// Input for House Number const Text(

"House No",

style: TextStyle(fontWeight: FontWeight.bold, fontSize: 16),

),

const SizedBox(height: 10), TextField(

decoration: InputDecoration( hintText: "Enter house number", border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

),

filled: true,

fillColor: Colors.white,

),

),

const SizedBox(height: 20),

// Input for Address const Text( "Address",

style: TextStyle(fontWeight: FontWeight.bold, fontSize: 16),

),

const SizedBox(height: 10), TextField(

maxLines: 3,

decoration: InputDecoration( hintText: "Enter full address", border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

),

filled: true,

fillColor: Colors.white,

),

),

const SizedBox(height: 20),

// Input for City const Text( "City",

style: TextStyle(fontWeight: FontWeight.bold, fontSize: 16),

),

const SizedBox(height: 10), TextField(

decoration: InputDecoration( hintText: "Enter city name", border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

),

filled: true,

fillColor: Colors.white,

),

),

const SizedBox(height: 20),

// Input for State const Text( "State",

style: TextStyle(fontWeight: FontWeight.bold, fontSize: 16),

),

const SizedBox(height: 10), TextField(

decoration: InputDecoration( hintText: "Enter state name", border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

),

filled: true,

fillColor: Colors.white,

),

),

const SizedBox(height: 20),

// Input for Pincode const Text( "Pincode",

style: TextStyle(fontWeight: FontWeight.bold, fontSize: 16),

),

const SizedBox(height: 10), TextField(

keyboardType: TextInputType.number, decoration: InputDecoration(

hintText: "Enter pincode", border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

),

filled: true,

fillColor: Colors.white,

),

),

const SizedBox(height: 30),

// Submit Button SizedBox(

width: double.infinity, child: ElevatedButton( onPressed: () {

Navigator.pushReplacementNamed(context, '/homePage');

},

style: ElevatedButton.styleFrom( backgroundColor: const Color(0xFF2DB83D), shape: RoundedRectangleBorder( borderRadius: BorderRadius.circular(12),

),

minimumSize: const Size(double.infinity, 50), elevation: 6,

),

child: const Text( "Submit",

style: TextStyle( fontSize: 18,

fontWeight: FontWeight.w600, color: Colors.white,

),

),

),

),

],

),

),

),

),

);

}

}

**Lab.dart**

import 'package:flutter/material.dart'; import 'package:krishak\_sathi/constant.dart';

void main() { runApp(const MyApp());

}

class MyApp extends StatelessWidget { const MyApp({super.key});

@override

Widget build(BuildContext context) { return MaterialApp( debugShowCheckedModeBanner: false, theme: ThemeData(

primarySwatch:

Colors.green, // Set the theme color (Green for agriculture) appBarTheme: AppBarTheme(

backgroundColor: Colors.green[700], // Custom AppBar color

),

textTheme: const TextTheme(),

),

home: LabsPage(),

);

}

}

class LabsPage extends StatelessWidget {

// A sample list of labs in Bhopal (this can be fetched from a database or API) final List<Map<String, String>> labs = [

{

'name': 'Bhopal Soil Testing Lab 1', 'address': 'gandhi Nagar, Bhopal', 'contact': '123-456-7890',

},

{

'name': 'Bhopal Soil Testing Lab 2', 'address': 'narela, Bhopal', 'contact': '123-456-7891',

},

{

'name': 'Bhopal Soil Testing Lab 3', 'address': 'indra Puri, Bhopal', 'contact': '123-456-7892',

},

{

'name': 'Bhopal Soil Testing Lab 4', 'address': 'lal ghati, Bhopal', 'contact': '123-456-7892',

},

{

'name': 'Bhopal Soil Testing Lab 5', 'address': 'Minal road, Bhopal', 'contact': '123-456-7892',

},

{

'name': 'Bhopal Soil Testing Lab 6', 'address': 'Ayodhya Nagar, Bhopal', 'contact': '123-456-7892',

},

];

@override

Widget build(BuildContext context) { return Scaffold(

appBar: AppBar(

title: const Text('Labs in Bhopal'), foregroundColor: Colors.white, backgroundColor: GobalColors.PrimaryColor,

),

body: ListView.builder( itemCount: labs.length, itemBuilder: (context, index) { final lab = labs[index];

return Card(

margin: const EdgeInsets.symmetric(vertical: 10, horizontal: 15), shape: RoundedRectangleBorder(

borderRadius:

BorderRadius.circular(20), // Rounded corners for the card

),

elevation: 10, // Add some shadow for a floating effect

color: Colors.green[50], // Light green background for the card child: InkWell(

onTap: () { showDialog( context: context, builder: (context) {

return AlertDialog(

title: Text(lab['name']!), content: Text(

'Address: ${lab['address']}\nContact: ${lab['contact']}'), actions: [

TextButton( onPressed: () => {},

child: const Text('Close'),

),

],

);

},

);

},

child: Padding(

padding: const EdgeInsets.all(16.0), child: Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween, children: [

// Left side with lab details Expanded(

child: Column(

crossAxisAlignment: CrossAxisAlignment.start, children: [

Text( lab['name']!,

style: const TextStyle( fontSize: 18,

fontWeight: FontWeight.bold,

color: Colors.green, // Dark green for title

),

),

const SizedBox(height: 8), Text(

lab['address']!,

style: const TextStyle( fontSize: 16,

color: Colors.grey, // Darker grey for address

),

),

const SizedBox(height: 8), Text(

'Contact: ${lab['contact']}', style: const TextStyle( fontSize: 14,

color: Colors.grey, // Lighter grey for contact

),

),

],

),

),

// Enhanced Arrow button GestureDetector(

onTap: () { showDialog( context: context, builder: (context) {

return AlertDialog(

title: Text(lab['name']!), content: Text(

'Address: ${lab['address']}\nContact: ${lab['contact']}'), actions: [

TextButton(

onPressed: () => Navigator.pop(context), child: const Text('Close'),

),

],

);

},

);

},

child: Container(

decoration: BoxDecoration(

color: Colors.green[700], // Dark green background shape: BoxShape.circle,

boxShadow: [ BoxShadow(

color: Colors.black26, blurRadius: 8,

offset: Offset(0, 4),

),

], // Shadow effect for depth

),

padding: const EdgeInsets.all( 12.0), // Space inside the button

child: GestureDetector( onTap: () {

Navigator.pushNamed(context, '/confirm\_address');

},

child: const Icon( Icons.arrow\_forward\_ios,

color: Colors.white, // White icon for contrast size: 20, // Icon size

),

),

),

),

],

),

),

),

);

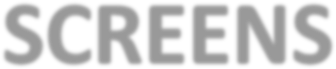
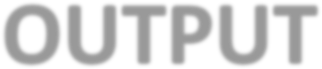
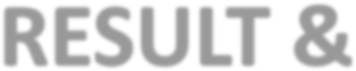
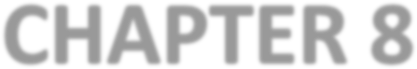
},

),

);

}

}



**CHAPTER-8 RESULT AND OUTPUT SCREENS**

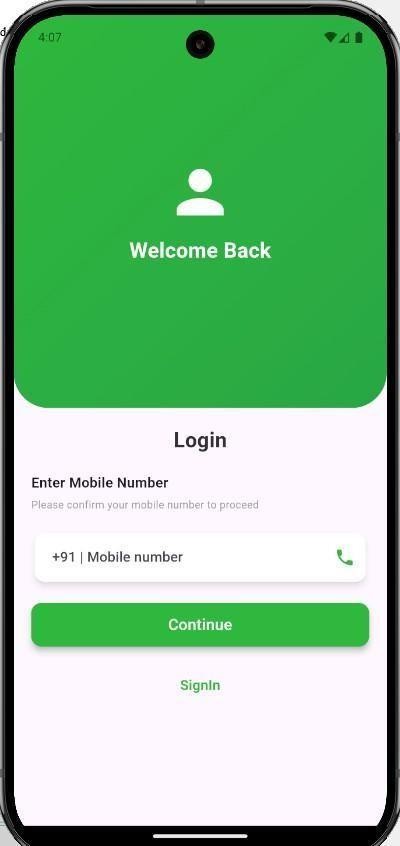
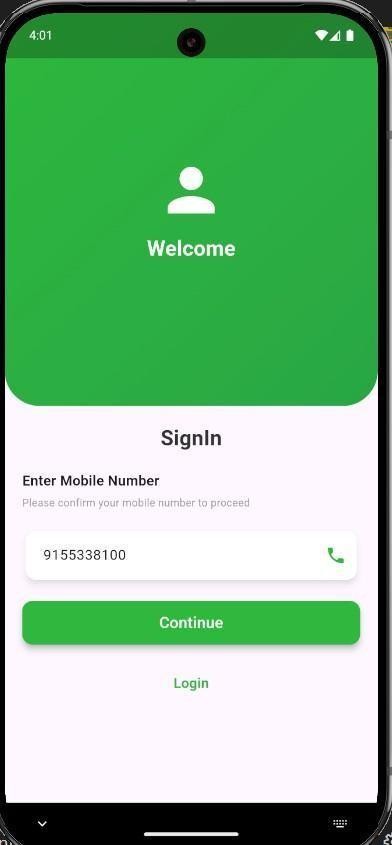
###### RESULTS

**Empowering Farmers with Data-Driven Insights:**

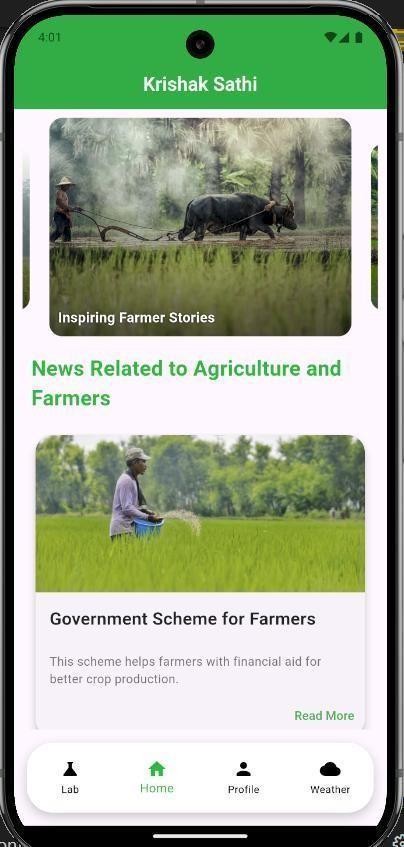
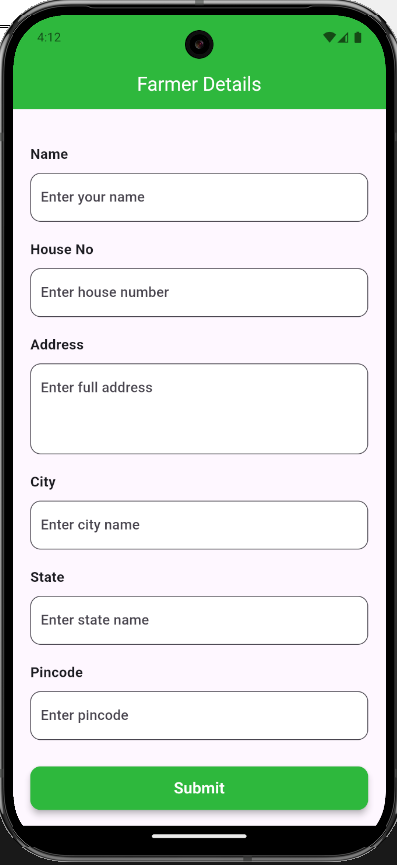
* **Accurate Soil Analysis :** Our app provides precise soil testing reports, identifying deficiencies in essential minerals like nitrogen, phosphorus, and potassium.
* **Personalized Crop Recommendations**: Based on soil analysis, we offer tailored crop recommendations to optimize yield and quality.
* **Real-time Weather Updates :** Farmers can access accurate weather forecasts to plan their farming activities effectively.
* **Market Insights** : Stay informed about market trends and prices to make informed decisions.

###### OUTPUTSCREENS

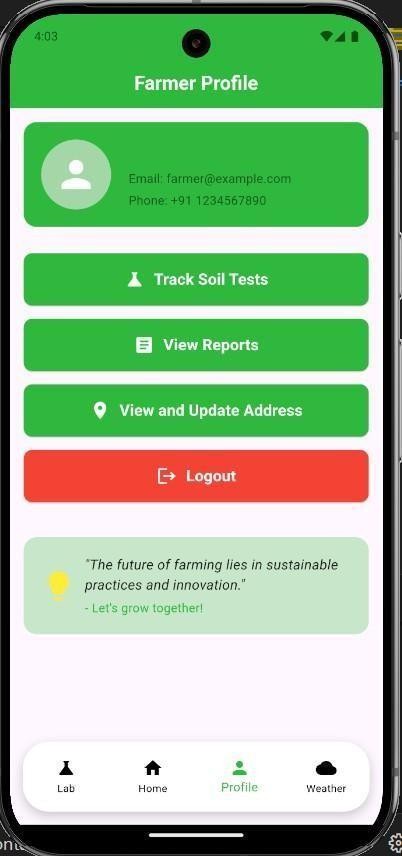
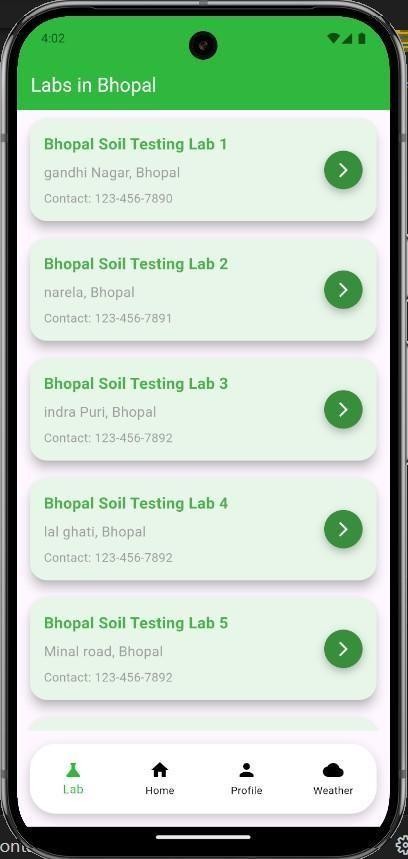
Below are the output screens captured during the testing phase:

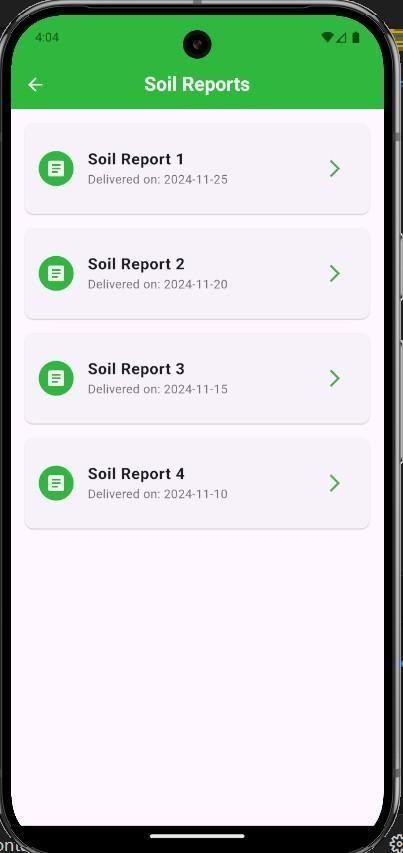
**Figure 8.2.1:Login and Signup Page at app opening**

****

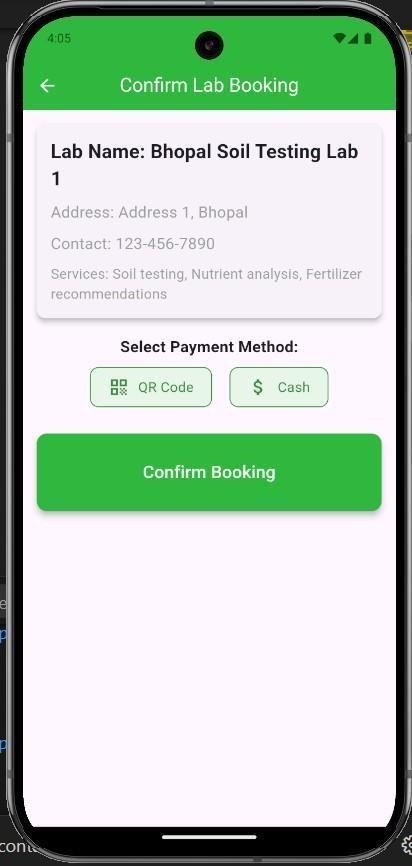
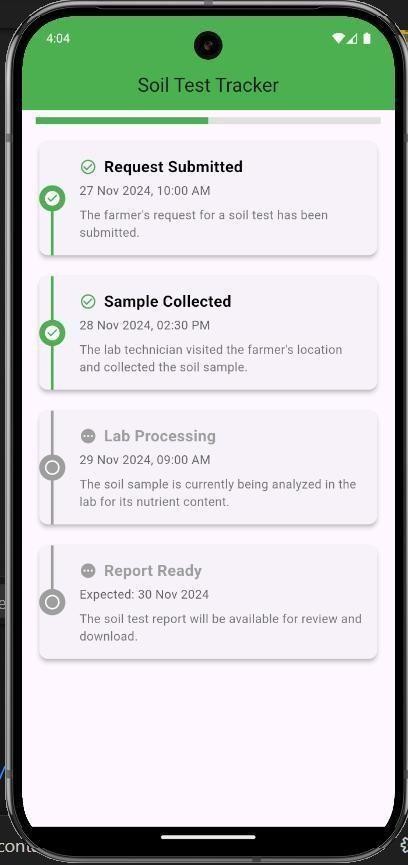
**Figure 8.2.2:Signup Details and home Page of app**

**Figure 8.2.2:Profile Details and Lab’s Details**

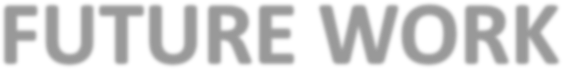
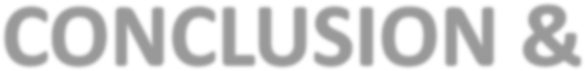
****

**Figure 8.2.4: Weather Report and Soil Report details**

**Figure 8.2.4: Lab Booking Details and Soil Test Tracker**





Sistec/Btech/CS/2024/5/MinorProject\_I/3

**CHAPTER-9 CONCLUSIONS AND FUTURE WORK**

###### CONCLUSION

The proposed solution for the "Annadata" aims to empower farmers by providing them with advanced tools to enhance soil fertility, improve crop productivity, and increase income. By leveraging technology such as the Flutter framework for cross-platform support, APIs for weather, soil, and crop data, and AI integration for personalized recommendations, the application has the potential to significantly improve farming practices.

The solution addresses key challenges faced by farmers, including the efficient use of fertilizers, weather forecasting, and soil health management. With a focus on localized content in multiple languages and an intuitive, farmer-friendly interface, the app ensures accessibility and ease of use, even for those with limited technical knowledge.

Economically, the app provides multiple revenue streams, ensuring financial sustainability, while also offering farmers cost savings through optimized fertilizer use and increased productivity. Additionally, it promotes environmental sustainability by reducing overuse of fertilizers and encouraging responsible farming practices.

Ultimately, this app has the potential to bridge the digital divide, empowering rural communities with the tools and knowledge they need to thrive. By promoting transparency, improving knowledge, and fostering smarter farming practices, the solution contributes to healthier crops, better incomes, and more sustainable agricultural practices.

###### FUTURE WORK

While the current version of the system is functional, there are several potential improvements and extensions that can be made to further enhance its capabilities:

* + 1. **Integration with More Data Sources**: In the future, the app can integrate additional data sources, such as satellite imagery and IoT-based sensors (e.g., soil moisture or crop health sensors), to provide even more accurate and real-time insights for farmers.
    2. **AI and Machine Learning Advancements:** As the app gathers more data from farmers, machine learning algorithms can be enhanced to offer more precise, personalized recommendations based on individual farm conditions and trends. This could also help in predicting long-term soil health or crop performance.

.

* + 1. **Multilingual and Cultural Expansion:** Beyond the initial language support (e.g., Hindi), the app could add more regional languages and dialects to cater to a wider farmer base across various parts of the country or globally. This will help bridge the gap in agricultural knowledge and foster inclusivity.
    2. **Partnerships with Governments and NGOs:** Future work could involve collaborating with governmental bodies and NGOs to integrate the app with existing agricultural schemes, subsidies, and initiatives. This would increase the app’s visibility and ensure it aligns with national and international agricultural development goals
    3. **Expanded Crop and Fertilizer Database:** Over time, the app will expand its database to include a wider variety of crops, fertilizers, and farming practices, enabling farmers to explore even more diverse agricultural options tailored to their specific location and soil type.
    4. **Offline Functionality:** For farmers in remote areas with limited internet connectivity, the app can be enhanced with offline capabilities, enabling them to access basic features like soil reports, fertilizer suggestions, and weather alerts even without an internet connection.

# REFERENCES

###### JOURNALS / RESEARCH PAPERS

1. Sharma, R., & Singh, P. (2017). "Impact of Soil Health Card Scheme on Fertilizer Usage and Productivity." *International Journal of Agricultural Sciences*, 9(3), 23-29.
2. Gupta, V., & Mehta, S. (2019). "Evaluation of Soil Health Card Program in Enhancing Agricultural Productivity." *Journal of Soil and Water Conservation*, 18(2), 12-18.
3. Reddy, A. (2020). "Role of Soil Health Cards in Sustainable Agriculture." *Indian Journal of Agronomy*, 25(1), 5-10.

###### BOOKS

1. Brady, N. C., & Weil, R. R. *The Nature and Properties of Soils*, Pearson Education, 15th Edition, 2016.
2. Lal, R., & Stewart, B. A. *Soil Management for Sustainability*, CRC Press, 1st Edition, 2019.
3. Tisdale, S. L., & Nelson, W. L. *Soil Fertility and Fertilizers*, Macmillan, 8th Edition, 2013.

###### WEBSITES

<https://www.soilhealth.dac.gov.in/home> https://[www.india.gov.in/spotlight/soil-health-card-scheme](http://www.india.gov.in/spotlight/soil-health-card-scheme) https://vikaspedia.in/agriculture/agri-schemes/soil-health-management <https://www.fao.org/soils-portal/en/>

# PROJECT SUMMARY

#### About Project

|  |  |
| --- | --- |
| **Title of the project** | **Career Advancements** |
| **Semester** | 6th |
| **Members** | 02 |
| **Team Leader** | Anuj Raghuwansh |
| **Describe role of every member in the project** | Anuj Raghuwanshi  Basuki Nath pandey |
| **What is the motivation for selecting this project?** | The motivation for this project is to help farmers improve soil fertility, increase crop production, and boost income through a mobile app that offers soil analysis, fertilizer recommendations, weather alerts, and crop diversification options. It aims to empower farmers with transparent, easy-to-understand advice, leveraging technology to optimize farming practices and support sustainable  agriculture. |
| **Project Type**  **(Desktop Application, Web Application, Mobile App, Web)** | Web- Based Application |

#### Tools &Technologies

|  |  |
| --- | --- |
| **Programming language**  **used** | Flutter |
| **APIs used**  **(with version)** | Open Api, APIRazorpay, APIOpenWeather |
| **Front End Technologies (with version, wherever Applicable)** | Flutter |

|  |  |
| --- | --- |
| **Back End Technologies (with version, wherever applicable)** | Node.js, Express.js |
| **Data base used** | MySQL, Local Storage (HIVE) |

#### Software Design & Coding

|  |  |
| --- | --- |
| **Is prototype of the software**  **developed?** | Yes |
| **SDLC model followed (Waterfall, Agile, Spiral etc.)** | Agile |
| **Why above SDLC model is followed?** | The **Agile** SDLC model is best for this project because it allows for iterative development, regular feedback, and flexibility to adapt to farmers' evolving needs. The app will require frequent updates based on user feedback, weather patterns, and data accuracy, which Agile supports through short development  cycles and continuous improvement. |
| **Software Design approach followed**  **(Functional object-oriented)** | Object Oriented Approach |
| **Name the diagrams developed (according to the Design approach followed)** | Use Case Diagram |
| **In case Object Oriented approach is followed, which of the OOPS principles are covered in design?** | Abstraction Polymorphism Encapsulation Inheritance |
| **No. of Tiers**  **(example3-tier)** | 3 |
| **Total no. of front end pages** | 1 |
| **Total no. of tables in database** | 1 |
| **Database is in which Normal**  **Form?** | - |

|  |  |
| --- | --- |
| **Are the entries in database**  **encrypted?** | - |
| **Front end validations applied**  **(Yes/No)** | Yes |
| **Session management done**  **(in case of web applications)** | Yes |
| **Is application browser compatible**  **(in case of web applications)** | Yes |
| **Exception handling done**  **(Yes/No)** | Yes |
| **Commenting done in code**  **(Yes/No)** | Yes |
| **Naming convention followed**  **(Yes/No)** | Yes |
| **What difficulties faced during Deployment of project?** | Integration Challenges,  Constantly evolving project requirements, Version Control Issues,  Dependency Management Complexity |
| **Total no. of Use-cases** | 1 |
| **Give titles of Use-cases** | User Management Project Management Document  Management |

#### Project Requirements

|  |  |
| --- | --- |
| **MVC architecture followed**  **(Yes/No)** | Yes |
| **Design Pattern used**  **(Yes/No)** | No |
| **If yes, write the name of**  **Design Pattern used** | - |
| **Interface type**  **(CLI/GUI)** | GUI |
| **No. of Actors** | 1 |

|  |  |
| --- | --- |
| **Name of Actors** | User |
| **Total no. of Functional**  **Requirements** | 6 |
| **List few important non- Functional Requirements** | Performance, Reliability, Usability |

#### Testing

|  |  |
| --- | --- |
| **Which testing is performed?**  **(Manual or Automation)** | Manual |
| **Is Beta testing done for this project?** | No |

#### Write project narrative covering above mentioned points

The proposed mobile-based application is designed to empower farmers, known as "Annadata," by providing valuable insights into soil fertility, crop optimization, and weather patterns. By leveraging soil analysis, AI-driven recommendations, and weather data, the app aims to help farmers increase crop yields, diversify crops beyond monocropping, and boost their income. With an easy-to-use, localized interface, the app ensures accessibility for farmers with varying levels of technical expertise, offering fertilizer recommendations, crop suggestions, and weather alerts. The project follows an **Agile SDLC model** to ensure iterative development, flexibility, and continuous improvement based on user feedback, as agricultural needs and data evolve. The app will be built using a **3-tier architecture**: a **Presentation Layer** for the user interface, an **Application Layer** for backend processing and AI integration, and a **Data Layer** that stores and processes soil, weather, and crop data. This approach allows for scalability, efficient maintenance, and seamless data integration. With **Flutter** as the development framework, the app will be compatible across Android, iOS, and web platforms, making it accessible to a wide audience. The market feasibility is strong, as farmers are increasingly adopting mobile technology and smart farming solutions. Financially, the project is sustainable with low initial costs and potential revenue streams through a freemium model, lab commissions, and advertisements. Data accuracy and validation are prioritized, ensuring reliable recommendations based on trusted sources.

|  |  |  |
| --- | --- | --- |
| YADAV AMAN  ANUJ RAGHUWANSHI PRANJAL AMULANI | 0187CS221216  0187CS221050  0187CS221152 | Guide Signature (Mr. Ankit Gupta) |
| PRIYAL GUPTA SHIVANSHU KUMAR SHUBHEK KUMAR | 0187CS221154  0187CS221187  0187CS221190 |  |

# APPENDIX-1 GLOSSARY OF TERMS

# F

**Flutter** is an open-source framework developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. It uses the Dart programming language and provides a rich set of pre-designed widgets that make it easier to create beautiful, responsive UIs..

**M**

**MySQL** is an open-source relational database management system

(RDBMS) that uses SQL to store, manage, and retrieve data. It is known for its speed, reliability, and ease of use, and is commonly used in web development, often as part of the LAMP stack..

**E**

**Express.js** is a lightweight, flexible web framework for Node.js that simplifies building web applications and APIs. It provides essential features like routing, middleware, and HTTP request handling, making it popular for backend development.

**N**

**NodeJs** Nodejs is a cross-platform, open-source JavaScript runtime environment that can run on Windows, Linux, Unix, macOS, and more. Node.js runs on the V8 JavaScript engine, and executes JavaScript code outside a web browser.