DETAILED PROJECT REPORT (DPR)

Project name:

AI-Powered Crop Recommendation System

Team:

- **Akash Adak (Team Lead)**
- **Anirban Jana**
- **Ankur Mishra**
- **Ankita Roy**
- **Aditya Raj Singh**
- **Abhinav Manohar**
- **Abhinav Raj**
- **Anushka Maiti**
- **Anushka Goswami**
- **Anisha Patel**

Group – A, Sl. No. – 1 to 10

Cluster – 2 (Full Stack and Java)

Submission date – 31st January' 26

1. Project Title & Abstract

Project Title

AI-Powered Crop Recommendation System

Abstract

The rapid advancement of technology has created a demand for intelligent, scalable, and user-centric software solutions. This project focuses on the design and development of **a Crop Recommendation system**, aimed at solving **agricultural problems** efficiently and reliably.

The system integrates modern frontend technologies, robust backend architecture, an accurate ML model and optimized database management to deliver a seamless user experience. The project emphasizes modularity, performance, security, and scalability, making it suitable for real-world deployment. Through this project, the team has applied industry-level development practices, collaborative workflows, and systematic testing to ensure a high-quality outcome.

---

2. Problem Statement & Objectives

Problem Statement

In today's digital environment, **traditional farming** poses significant challenges such as inefficiency, lack of automation, poor scalability, or limited accessibility. Existing solutions often suffer from **limitations like performance issues, security flaws, manual effort, etc.**, making them unsuitable for modern requirements.

This project aims to overcome these challenges by providing a comprehensive, automated, and scalable solution tailored to current technological standards.

Objectives

The primary objectives of this project are:

- To design and develop a **fully functional and scalable system**
- To automate and optimize **agriculture and crop selection**

- To ensure data integrity, security, and reliability
- To provide a responsive and user-friendly interface
- To follow modular and maintainable coding practices
- To enable future enhancements with minimal restructuring

---

3. System Architecture & Workflow

System Architecture

The system follows a **layered architecture** consisting of:

- **Presentation Layer (Frontend)** – Handles user interaction and UI rendering
- **Application Layer (Backend)** – Processes business logic and API requests
- **Data Layer (Database)** – Stores and manages structured data securely
- **Machine Learning Pipeline** – Random Forest Classifier model training and integration

This separation of concerns improves maintainability, scalability, and debugging efficiency.

Workflow

1. User interacts with the frontend interface
2. Requests are sent to the backend via secure APIs
3. Backend processes the request and communicates with the ML service
4. The packaged information is stored in the Database
5. The response is sent back to the frontend for display

---

4. Technologies Used

Category	Technologies
Frontend	HTML, Tailwind CSS, JavaScript, React.js

Category	Technologies
Backend	Node.js, Express.js
Database	MongoDB
ML	Random Forest Classifier, Python
Version Control	Git, GitHub
API Testing	Postman
Development Tools	VS Code
Deployment	Vercel

---

5. Module-Wise Description

1. Authentication Module

- User registration and login
- Password encryption and validation
- Role-based access control

2. Core Functional Module

- Handles the main features of the application
- Implements business logic and data processing

3. Database Management Module

- CRUD operations
- Data validation and indexing
- Optimized query handling

4. Error Handling & Logging Module

- Centralized error handling
- User-friendly error messages
- Logging for debugging and monitoring

5.The Machine Learning Pipeline

Data Acquisition: Utilized a multi-feature dataset containing over 2,200 records of soil chemistry (N-P-K-pH) and climatic conditions (Temp-Humidity-Rainfall).

Feature Engineering: Performed Exploratory Data Analysis (EDA) to identify correlations between rainfall and crop success.

Implemented StandardScaler to normalize features, ensuring that rainfall (measured in hundreds) didn't statistically overshadow pH levels (measured in units).

Model Selection: Evaluated multiple algorithms including Deep Learning (DNN), XGBoost, and Random Forest.

Final Choice: Random Forest Classifier.

Reasoning: It provides high interpretability and handles non-linear agricultural data better than simple regression models, achieving an accuracy of ~99%.

Model Serialization: Used joblib to export the trained weights into a portable .pkl format, allowing the backend to load the model without retraining.

---

6. Database Design

Database Overview

The database is designed using a **normalized structure** to avoid redundancy and ensure data consistency.

Sample Entities

- Users
- Logs

Key Features

- Primary and foreign keys
- Indexing for faster queries
- Secure data storage
- Backup-friendly structure

7. Individual Roles & Responsibilities (Group – A, Sl.no. 1 to 10)

Member	Role	Responsibilities
Akash Adak	Team Lead, Backend Developer (Connection establishment and ML backend)	Project planning, coordination, final review, CI-CD pipeline for testing
Anirban Jana	AI/ML Engineer	Model training & integration
Ankur Mishra	Documentation Lead and Frontend Developer	DPR & reports, Frontend refinement
Ankita Roy	Backend Developer	Login and Register backend logic
Aditya Raj Singh	Backend Developer	Integrated AI, Database and APIs (Backend Architecture)
Anushka Maiti	Frontend Developer	Login and Register page design
Abhinav Manohar	Frontend Developer	Dashboard design
Anuska Goswami	Frontend Developer	Prediction page
Abhinav Raj	Frontend Developer	Navbar, Footer, Home
Anisha Patel	Frontend Developer	Prediction output page

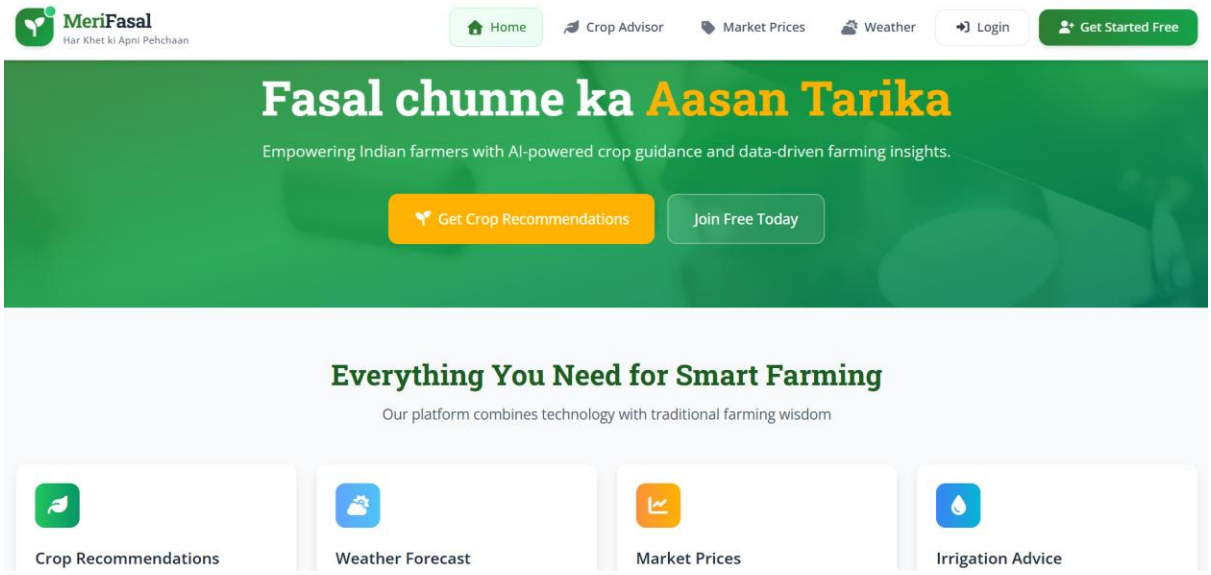
8. Results, Screenshots & Outputs

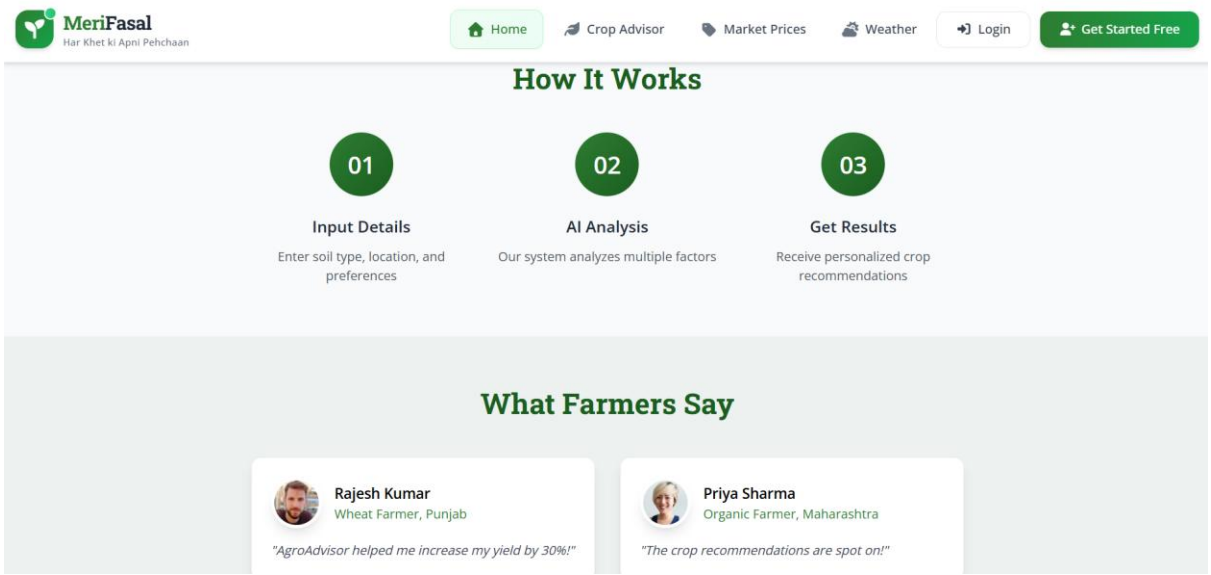
The project successfully meets all defined objectives. Key results include:

- Fully functional system as per requirements
- Smooth frontend-backend communication
- Secure data handling
- Optimized performance

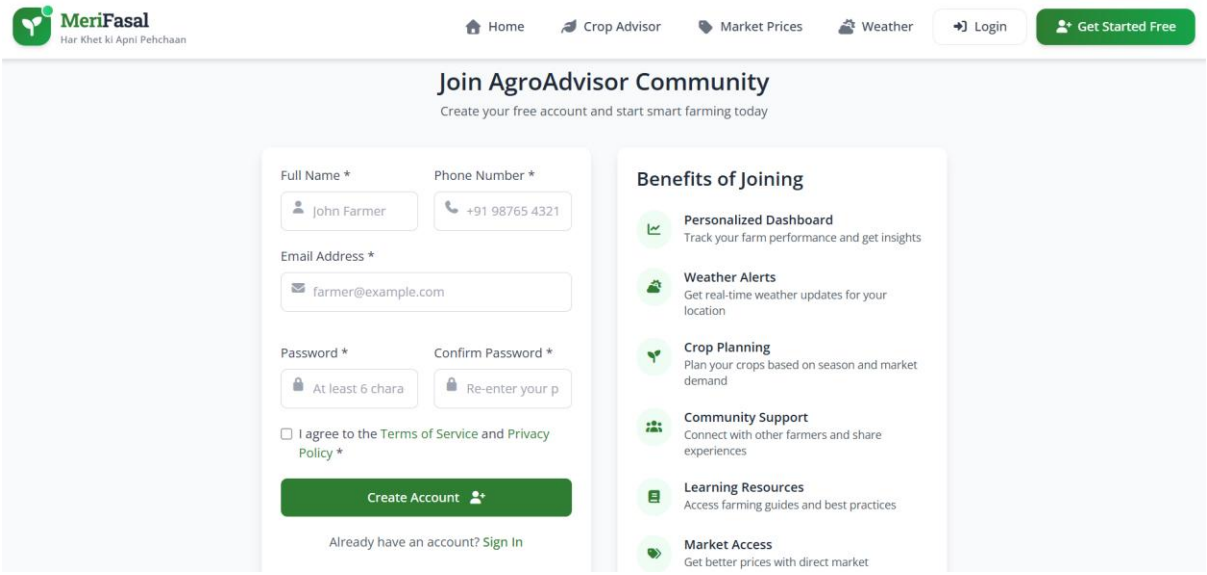
Screenshots Included:

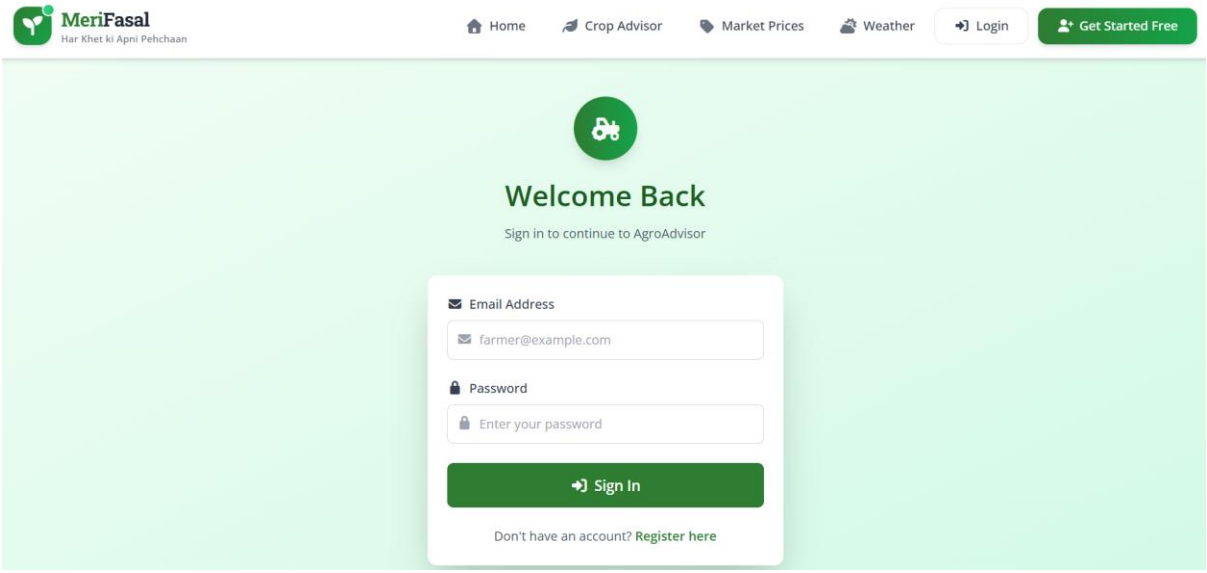
- Home page



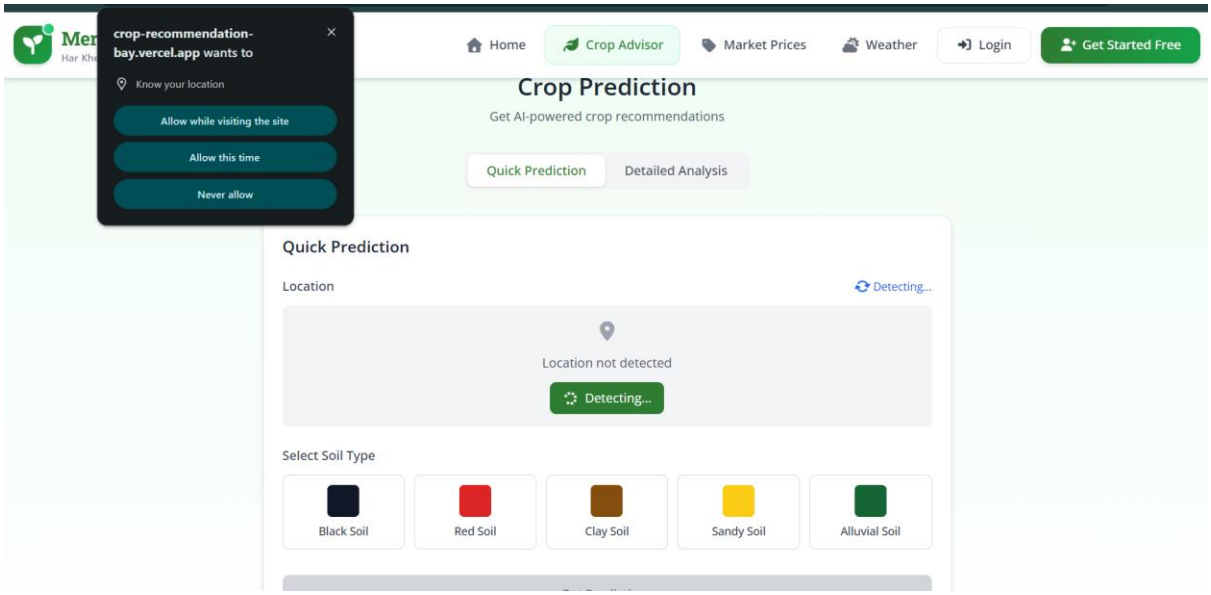


- **Login/Register page**





- Core functionality pages



---

9. Conclusion & Future Scope

Conclusion

The project demonstrates a practical implementation of modern software development principles and technologies. Through effective teamwork and structured development, the system delivers a reliable and scalable solution to the identified problem.

Future Scope

- Integration of AI-based analytics
- Mobile application development
- Cloud-based scaling
- Advanced security enhancements
- Performance optimization using caching techniques