ECO-AGRI GUIDE

The Crop Replacement System

MAJOR PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY
(Information Technology)

SUBMITTED BY:

Armaan Singh -**2004893(URN)**
Navdeep Kaur -**2004960(URN)**

SUBMITTED TO:

Prof.Pankaj Bhambri
Assistant Professor
Major Project Guide

Department of Information Technology

GURU NANAK DEV ENGINEERING COLLEGE

LUDHIANA, INDIA

STUDENT'S DECLARATION

We hereby certify that the work which is being presented in this training report with the project entitled **ECO-AGRI GUIDE** by **Armaan Singh , Navdeep Kaur**, University Roll No. **2004893 , 2004960** in partial fulfillment of requirements for the award of degree of **B.Tech. (Information Technology)** submitted in the Department of Information Technology at **GURU NANAK DEV ENGINEERING COLLEGE, LUDHIANA** under **I.K. GUJRAL PUNJAB TECHNICAL UNIVERSITY** is an authentic record of our own work carried out under the supervision of **Dr. Kulvinder Singh Mann**, HOD ,IT Department **GNDEC ,Ludhiana.**

The matter presented has not been submitted by us in any other University / Institute for the award of B.Tech. Degree.

Student Name: Armaan Singh , Navdeep Kaur

Univ. Roll No. 2004893 , 2004960

(Signature of Students)

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

Signature of Internal Examiner

The External Viva-Voce Examination of the student has been held on .....

Signature of External Examiner

Signature of HOD

ABSTRACT

In today's tech-savvy era, advancements in science and technology have revolutionized key aspects of human life, particularly in the realms of shelter and clothing. However, agriculture, one of the fundamental pillars supporting human survival, often lags behind in technological integration. Many farmers, lacking formal education and scientific insights, resort to traditional trial-and-error methods, leading to inefficiencies in resource utilization. Our system addresses this gap by leveraging technology to develop a predictive model focused on recommending the most suitable crops for cultivation based on various parameters.

This predictive model incorporates cutting-edge technologies such as the Decision Tree algorithm in machine learning. By analyzing factors like soil composition, climate conditions, and crop characteristics, the model identifies the optimal crops that align with a specific farm's capabilities and environmental factors. This approach empowers farmers with data-driven insights, enabling them to make informed decisions and enhance productivity while minimizing resource wastage.Our crop recommendation system integrates sophisticated technologies, prominently featuring the Decision Tree algorithm in machine learning to revolutionize agricultural practices. The Decision Tree algorithm, renowned for its interpret ability and versatility, plays a pivotal role in our predictive model by analyzing a multitude of parameters crucial for crop selection. These parameters encompass a wide array of agricultural variables such as soil composition, pH levels, temperature ranges, humidity levels, rainfall patterns, data. Through a systematic analysis of these factors, the Decision Tree algorithm constructs a hierarchical decision-making framework. This framework, depicted as a tree structure, offers clear and actionable decision rules tailored to specific farms and their unique environmental conditions.

The advantages of leveraging the Decision Tree algorithm in our crop recommendation system are manifold. Firstly, its interpret ability empowers farmers with comprehensible insights into the rationale behind recommended crop choices, enabling informed decision-making even for those without extensive technical knowledge. Additionally, the adaptability of Decision Trees to handle diverse data types, including categorical and numerical data, ensures that all relevant agricultural parameters are considered in the decision-making process. This adaptability extends to accommodating dynamic environmental changes, allowing farmers to adjust crop selections based on evolving conditions.

ACKNOWLEDGEMENT

WE are highly grateful to the **Dr. Sehijpal Singh**, Principal, **Guru Nanak Dev Engineering College (GNDEC)**, Ludhiana, for providing this opportunity to carry out the major project work at **GNDEC , Ludhiana**

The constant guidance and encouragement received from **Dr. Kulvinder Singh Mann**, H.O.D., IT Department, GNDEC Ludhiana has been of great help in carrying out the project work and is acknowledged with reverential thanks.

WE would like to express a deep sense of gratitude and thanks profusely to **Prof. Pankaj Bhambri**, without his wise counsel and able guidance, it would have been impossible to complete the project in this manner.

WE express gratitude to other faculty members of Information Technology Department of GNDEC for their intellectual support throughout the course of this work.

Finally, WE are indebted to all whosoever have contributed in this report work.

Armaan Singh

Navdeep Kaur

List of Figures

Contents

1 Introduction

1.1 Introduction to Project

Introducing EcoAgri Guide, it is decision support revolutionary companion in agriculture. In an era where environmental consciousness is paramount, farmers need a tool that minimizes the ecological footprint of their practices. EcoAgri Guide is an innovative web application designed to provide farmers with crucial insights into crop replacements, helping them make environmentally responsible decisions that safeguard soil fertility.

At the core of EcoAgri Guide is a sophisticated machine learning algorithms that analyzes the dataset about impact of various crops on environmental components of particular region. And provide prediction report on the potential effects on soil health and other soil component. The application goes beyond traditional considerations of yield and profit, offering a holistic view of the ecological consequences associated with different crops.

EcoAgri Guide doesn't just stop at identifying problems; it provides practical solutions. The application suggests environmentally friendly crop replacements that align with the specific needs and conditions of each farm. Whether it's recommending nitrogen-fixing cover crops to enhance soil fertility, EcoAgri Guide guides farmers towards choices that promote sustainable agriculture.

One of the standout features of EcoAgri Guide is its real-time monitoring capabilities. The app keeps farmers informed about changing conditions, allowing for dynamic adjustments to crop plans. This proactive approach enables farmers to adapt to environmental fluctuations, ensuring a resilient and sustainable agricultural system.

EcoAgri Guide is more than just an app; it's a movement towards a greener and more sustainable future for agriculture. By fostering awareness and providing actionable insights, EcoAgri Guide empowers farmers to be stewards of the land, cultivating a harmonious relationship between agriculture and the environment.It is a robust machine learning engine that continuously analyzes and processes data related to soil health, crop performance, and environmental factors. Through this ML backbone, EcoAgri Guide offers predictive analytics and smart recommendations to optimize crop planning. Farmers can benefit from automated insights that improve decision-making. Admins can update and improve the machine learning datasets based on the latest research findings, ensuring that the platform remains at the forefront of agricultural innovation. The administrative interface also facilitates the customization of recommendations to align with specific regional or

environmental considerations.

However,there is a big role of human expertise in agriculture. EcoAgri Guide's semi-manual interface empowers administrators to have direct control over the system. The administrative dashboard provides a user-friendly interface where administrators can input specific parameters, set priorities, and fine-tune recommendations generated by the machine learning algorithms. This dynamic collaboration between ML and human administrators ensures a personalized and adaptive approach to agricultural management.

1.2 Project Category

EcoAgri Guide falls under the category of Application or System Development, specifically targeting sustainable agriculture. By harnessing the power of machine learning algorithms, it offers farmers a comprehensive web application that revolutionizes their approach to crop planning. Through real-time monitoring and predictive analytics, EcoAgri Guide provides actionable insights to minimize ecological footprint while maximizing yield. Admins play a pivotal role in fine-tuning recommendations and ensuring alignment with specific regional or environmental considerations, highlighting the symbiotic relationship between human expertise and technological innovation. This platform bridges the gap between traditional agricultural practices and modern environmental consciousness, facilitating a transition towards greener and more sustainable farming methods.

1.3 Objectives

- To identify the significant features related to the crop fertilization.
- To suggest the optimal crop fertilizer as per the various factors like soil composition , climate conditions and crop characteristics.

1.4 Problem Formulation

The problem formulation for EcoAgri Guide centers on optimizing crop selection and agricultural practices based on multiple environmental factors and inputs, utilizing a machine learning-based prediction model. Specifically, the key aspects include:

Multi-factor Crop Prediction: Developing a predictive model that analyzes multiple environmental variables such as nitrogen, phosphorus, potassium, temperature, humidity, pH levels, and rainfall to forecast the most suitable crops for a given region or farm.

Optimization Objective: Defining the optimization objective, which involves maximizing crop yield while minimizing environmental impact, soil degradation, water consumption, and air pollution. This objective encompasses both economic and ecological considerations.

Model Development: Develop machine learning algorithms capable of accurately predicting crop suitability based on the selected features. Consider various regression or classification techniques to build a robust prediction model.

Model Training and Evaluation: Training the machine learning model using historical data on crop performance under varying environmental conditions, and evaluating its predictive accuracy and performance through cross-validation and testing on independent datasets.

Recommendation and Decision Support: Developing a user-friendly interface that provides farmers with actionable recommendations on crop selection, cultivation practices, and resource management based on the predictions generated by the machine learning model. Empowering farmers with decision support tools to make informed choices for sustainable agriculture.

1.5 Identification of Need

With India being a leading producer of agricultural goods and a significant contributor to the country's GDP, the agriculture industry plays a crucial role in the economy. However, Indian farmers face various challenges, including the necessity to choose crops that align with the terrain and climate of their region to maximize yield. Given the direct impact of climatic conditions and soil characteristics on crop productivity, there's a pressing need to develop crop management practices that consider site appropriateness and soil suitability. Additionally, with weather and agriculture being closely intertwined, adapting to changing climatic trends becomes imperative for productive farming. Precision agriculture, which emphasizes technologies like irrigation systems, fertilization, crop monitoring, and yield prediction, has shown promising advancements. However, to fully leverage these technologies, farmers require tools that enable them to choose the best-suited crops based on their region's unique meteorological and soil conditions. Therefore, the development of a crop recommendation system, such as EcoAgri Guide, utilizing Ma-

chine learning techniques like Decision Tree, becomes essential. Such a system would empower farmers with personalized recommendations tailored to their specific terrain and environmental factors, ultimately enhancing production and sustainability in Indian agriculture.

1.6 Existing System

The existing agricultural system in India typically relies on traditional farming practices passed down through generations. Farmers often make crop decisions based on their own experience, local knowledge, and advice from agricultural extension officers. However, these methods may not always be optimal, especially in the face of climate change and shifting environmental conditions.

Extension services and government agencies provide some support through advisory services, weather forecasts, and occasional training programs. However, these resources may not always reach all farmers, particularly those in remote or marginalized communities. Additionally, the advice provided may not be tailored to the specific needs and conditions of individual farms or regions.

Some technological interventions have been introduced in recent years to address these challenges. For example, there are mobile applications and online platforms that provide weather forecasts, market prices, and basic agronomic information. However, these tools often lack the sophisticated analytics and personalized recommendations needed to truly optimize crop selection and management.

Overall, while there are existing systems and support mechanisms in place for Indian farmers, there is still a significant gap in terms of access to advanced technology and personalized advisory services. This gap presents an opportunity for innovative solutions like EcoAgri Guide to provide more tailored, data-driven recommendations and support for sustainable agriculture.

1.7 Proposed System

In our proposed system, EcoAgri Guide, we introduce a comprehensive solution for crop recommendations based on key environmental factors such as nitrogen, phosphorus, potassium levels in soil, as well as temperature, humidity, pH, and rainfall. Leveraging machine learning techniques, specifically a decision tree algorithm, our system analyzes these parameters to

generate personalized crop recommendations tailored to the specific needs of each farm.

- **Crop Recommendations Based on Key Environmental Factors:** The proposed system utilizes machine learning algorithms, specifically decision trees, to analyze key environmental factors such as nitrogen, phosphorus, potassium levels in soil, temperature, humidity, pH, and rainfall. These factors are crucial determinants of crop suitability and yield potential.

- **Decision Tree Algorithm:** Decision trees are employed as the underlying algorithm for crop recommendation due to their simplicity, interpretability, and ability to handle both categorical and numerical data effectively. Decision trees partition the feature space based on the provided environmental variables to make predictions about the most suitable crops for a given set of conditions.

- **User Interface for Interaction:** The system features a user-friendly interface that allows farmers to interact with the platform easily. Through the interface, farmers can input information about their soil characteristics, climate conditions, and other relevant factors. They can also view the recommended crops based on the provided data.

- **Personalized Recommendations:** The decision tree algorithm generates personalized recommendations tailored to the specific soil and climate conditions of each farm or region. By considering the unique combination of environmental factors, the system provides farmers with actionable insights to optimize crop selection and management practices.

- **Scalability and Adaptability:** The proposed system is designed to be scalable and adaptable to different regions and farming contexts. The decision tree model can be trained and updated using data from various locations, allowing the system to accommodate diverse agricultural landscapes and climatic conditions.

- **Continuous Improvement:** The system is built with provisions for continuous improvement and refinement. As new data becomes available and the model's performance is evaluated, updates can be made to enhance the accuracy and relevance of the crop recommendations provided to farmers.

Overall, the proposed system aims to leverage machine learning techniques, particularly decision trees, to offer personalized crop recommendations based on soil and climate data. With its user-

friendly interface and adaptable nature, the system seeks to empower farmers with the insights needed to make informed decisions and optimize agricultural productivity sustainably.

1.8 Unique Features of the System

The unique features of ecoagriguide are:

Intelligent Decision Making: EcoAgri Guide employs decision tree technology to make intelligent and data-driven crop recommendations based on a comprehensive analysis of environmental factors. This technology enables the system to navigate complex datasets and identify the most suitable crops for specific soil and climate conditions.

Interpretability and Transparency: Decision trees offer interpretability and transparency, allowing users to understand the reasoning behind each recommendation. Farmers can easily interpret the decision-making process of the algorithm, fostering trust and confidence in the system's suggestions.

Adaptability to Diverse Conditions: The decision tree algorithm used in EcoAgri Guide is adaptable to diverse agricultural landscapes and climatic conditions. It can handle both categorical and numerical data, making it versatile in generating recommendations for different regions and farming contexts.

Real-Time Updates: The decision tree model can be updated in real-time with new data, ensuring that the recommendations remain relevant and accurate as environmental conditions change. This feature enhances the system's responsiveness to dynamic agricultural environments.

Personalized Recommendations: Through decision tree technology, EcoAgri Guide delivers personalized crop recommendations tailored to each farm's specific needs and conditions. By considering factors such as soil composition, temperature, humidity, and rainfall, the system provides farmers with actionable insights to optimize crop selection and management practices.

Scalability and Efficiency: Decision tree algorithms are known for their scalability and efficiency in handling large datasets. EcoAgri Guide can process vast amounts of environmental data quickly and effectively, enabling timely and informed decision-making for farmers across different regions.

2 Requirement Analysis and System Specification

2.1 Feasibility Study

A feasibility study based on the content you provided would likely focus on assessing the practicality and viability of implementing the described system with React.js for the frontend, Node.js for the backend, Flask for ML model interaction, and MongoDB as the database. Here's an outline of what such a feasibility study might include:

Technical Feasibility:

Frontend Technology (React.js):

Assess the availability of React.js developers and resources. Evaluate the compatibility of React.js with the project requirements, such as handling dynamic UI components for displaying crop recommendations.

Backend Technology (Node.js):

Determine if Node.js is suitable for handling backend logic, API interactions, and communication with the ML model. Evaluate scalability and performance considerations with Node.js, especially in handling concurrent requests from the frontend.

Database (MongoDB):

Assess MongoDB's compatibility with the data structure required for storing agricultural parameters, crop recommendations, and user data. Evaluate MongoDB's scalability and data handling capabilities for a large number of farms and environmental variables.

ML Model Integration (Flask):

Assess Flask's suitability for integrating the Decision Tree ML model with the Node.js backend. Evaluate the ease of deploying and managing the ML model within Flask for real-time crop recommendation generation. Consider the communication protocols and data exchange formats between Flask and the frontend/backend components.

Resource and Skill Availability:

Evaluate the availability of resources (developers, infrastructure, tools) required for implementing and maintaining the system. Assess the skill level and expertise of the development team in React.js, Node.js, Flask, MongoDB, and ML model integration.

Cost Analysis:

Estimate the costs associated with development, deployment, maintenance, and potential scal-

ability of the system. Consider licensing fees, cloud infrastructure costs (if applicable), and ongoing support expenses.

Risk Assessment:

Identify potential risks such as technical challenges in integrating diverse technologies, scalability issues, data security concerns, and regulatory compliance. Develop risk mitigation strategies to address identified risks during implementation and operation.

Market and User Acceptance:

Conduct market research to understand the demand for such a crop recommendation system among farmers and agricultural stakeholders. Gather feedback from potential users to assess user acceptance, usability, and desired features.

Conclusion and Recommendations:

Summarize the findings of the feasibility study, highlighting technical feasibility, resource availability, cost considerations, risks, and market acceptance. Provide recommendations on the feasibility of proceeding with the development and implementation of the described system, along with potential optimizations or alternative approaches.

2.2 Software Requirement Specification Document

1. Introduction EcoAgri Guide is a web-based application designed to provide farmers with personalized crop recommendations based on environmental factors. The system utilizes machine learning algorithms to analyze data and generate recommendations tailored to each farm's specific needs and conditions.

2. Data Requirement:

- The system requires access to comprehensive environmental data, including soil composition, temperature, humidity, pH levels, rainfall, nitrogen, phosphorus, and potassium levels.
- Data sources may include weather stations, soil testing labs, agricultural research institutions, and satellite imagery.
- Historical data is necessary for training machine learning algorithms and generating accurate crop recommendations.

3. Functional Requirement:

- The system should allow farmers to input their farm's data, including soil characteristics, climate conditions, and crop preferences, through a user-friendly interface.
- It should analyze the input data using decision tree algorithms to generate personalized crop recommendations.
- The system must provide clear and understandable explanations for each recommendation to enhance user trust and confidence.
- Users should be able to view and compare recommended crops, along with relevant agronomic information and best practices.

4. Performance Requirement:

- The system should be able to process large volumes of data efficiently to generate timely recommendations.
- Response time for generating recommendations should be fast, ideally within seconds or minutes.
- The system must be capable of handling concurrent user requests without significant degradation in performance.
- It should be scalable to accommodate increasing data volumes and user traffic as the user base grows.

5. Dependability Requirement:

- The system must be reliable and available whenever farmers need to access it.
- It should have built-in mechanisms to handle errors and exceptions gracefully, providing informative error messages to users.
- Regular backups of data and system configurations should be performed to prevent data loss and ensure system integrity.

6. Maintainability Requirement:

- The system should be designed with modularity and scalability in mind to facilitate future updates and enhancements.

- Code should be well-documented and organized, making it easier for developers to understand and maintain.
- Version control systems should be utilized to manage changes to the codebase and track revisions.

7. Security Requirement:

- The system must ensure the confidentiality and integrity of user data, including farm information and crop preferences.
- Access control mechanisms should be implemented to restrict unauthorized access to sensitive data and system functionalities.
- Data encryption techniques should be employed to secure data transmission over networks.
- Regular security audits and vulnerability assessments should be conducted to identify and address potential security risks.

8. Look and Feel Requirement:

- The user interface should be intuitive, visually appealing, and easy to navigate, catering to users with varying levels of technical proficiency.
- Consistent branding and design elements should be maintained throughout the application to provide a cohesive user experience.
- Interactive features such as tooltips, dropdown menus, and interactive charts can enhance user engagement and satisfaction.

2.3 Validation

1. Input Validation:

- Ensure that all user inputs, such as soil composition data, climate conditions, and crop preferences, are properly validated to prevent invalid or malicious data from being processed.
- Validate input formats, ranges, and constraints to ensure data integrity and accuracy in generating crop recommendations.

2.Authentiation and Authorization Validation:

- Implement authentication mechanisms to verify the identity of users accessing the system, such as farmers or administrators.
- Validate user credentials against stored records in the system's database to grant appropriate access levels and permissions.
- Enforce authorization checks to ensure that users can only access functionalities and data relevant to their roles and responsibilities.

3. Form Validation:

- Validate form submissions on the user interface to ensure that all required fields are filled out and that data formats are correct.
- Display informative error messages to users for any validation errors encountered, guiding them to correct their inputs appropriately.

4. Error Handling and Logging:

- Implement robust error handling mechanisms to gracefully handle unexpected errors and exceptions that may occur during system operation.
- Log relevant error details, including timestamps, error codes, and user actions, to facilitate troubleshooting and debugging.

By incorporating these validation techniques, EcoAgri Guide can ensure the integrity, security, and reliability of its data and functionalities, providing users with a safe and seamless experience.

2.4 Expected hurdles

Data Quality and Availability: Obtaining reliable and comprehensive environmental data, including soil composition, weather patterns, and historical crop performance, may pose a significant challenge. Data quality issues such as inaccuracies, inconsistencies, and gaps can affect the accuracy and reliability of the crop recommendations generated by EcoAgri Guide.

Technical Infrastructure: Setting up and maintaining the technical infrastructure required to

support EcoAgri Guide, including servers, databases, and networking components, may be challenging. Ensuring scalability, reliability, and security of the system infrastructure while managing costs and resources effectively is essential.

User Adoption and Acceptance: Encouraging farmers to adopt and use EcoAgri Guide may be met with resistance or skepticism, particularly among those who are accustomed to traditional farming practices. Overcoming barriers such as lack of digital literacy, language barriers, and cultural preferences requires effective communication, training, and support mechanisms.

Algorithm Accuracy and Interpretability: Ensuring the accuracy and interpretability of the machine learning algorithms used in EcoAgri Guide is crucial. Farmers need to trust the system's recommendations and understand the underlying reasoning behind them. Fine-tuning the algorithms and providing transparent explanations for recommendations can help address these challenges.

Regulatory and Compliance Issues: Compliance with regulatory requirements, data privacy laws, and industry standards poses potential hurdles for EcoAgri Guide. Ensuring that the system adheres to relevant regulations and safeguards user data privacy and security is essential to mitigate legal and compliance risks.

Socio-economic Factors: Socio-economic factors such as access to technology, financial resources, and socio-cultural norms may impact the adoption and success of EcoAgri Guide. Addressing disparities in access and resources among farmers, as well as understanding local socio-cultural contexts, is critical for the system's effectiveness and sustainability.

Environmental Variability: Environmental variability and unpredictability, such as changes in climate patterns, extreme weather events, and natural disasters, can impact agricultural productivity and the relevance of crop recommendations provided by EcoAgri Guide. Developing adaptive strategies and contingency plans to address these challenges is essential for ensuring the system's resilience and usefulness in dynamic environments.

Navigating these hurdles requires a holistic approach that addresses technical, social, economic, and environmental factors, as well as ongoing collaboration and feedback from stakeholders to continuously improve and refine EcoAgri Guide.

2.5 SDLC Model used

For a project like EcoAgri Guide, which involves complex data analysis, machine learning, and user interaction, the Iterative Model of the Software Development Life Cycle (SDLC) is well-suited. Here's why:

- **Iterative Development:** The Iterative Model allows for the development process to be broken down into smaller, manageable iterations. Each iteration focuses on a specific aspect or feature of the system, allowing for incremental development and refinement.
- **Feedback and Adaptation:** Given the complexity of the system and the need for data analysis and user feedback, the Iterative Model enables continuous feedback loops. Farmers and stakeholders can provide feedback on early iterations, allowing for adjustments and improvements to be made throughout the development process.
- **Flexibility and Adaptability:** The Iterative Model offers flexibility to accommodate changes in requirements, technology, and environmental factors. As new data becomes available or user needs evolve, the development team can adapt the system accordingly in subsequent iterations.
- **Risk Management:** By breaking down the development process into smaller iterations, the Iterative Model helps mitigate risks associated with uncertainty and complexity. Risks can be identified and addressed early in the development cycle, reducing the likelihood of major setbacks later on.
- **Quality Assurance:** Each iteration undergoes its own testing and quality assurance processes, ensuring that the system meets performance, reliability, and usability standards. Bugs and issues can be identified and addressed iteratively, leading to a more robust and reliable final product.

Overall, the Iterative Model offers a structured yet flexible approach to software development, making it well-suited for projects like EcoAgri Guide where continuous feedback, adaptation, and refinement are essential for success.

3 System Design

3.1 Design Approach

Function-Oriented Approach:

- In the backend code (presumably Node.js with Express), functions like Register, Login, CreateUser, UpdateUser, DeleteUser, and GetUsers are defined to handle specific HTTP requests (e.g., POST, GET).
- Each function is responsible for a distinct operation, such as user registration, login authentication, user creation, updating user information, deleting users, and retrieving user data.
- These functions follow a modular approach, encapsulating related operations within each function. For example, the Register function checks for existing users, creates a new user if none exists, and sends appropriate responses.
- The functions interact with the UserModel object, presumably representing a user schema in a MongoDB database, demonstrating a function-oriented approach to data access and manipulation.
- Error handling is implemented within each function, with try-catch blocks to capture and handle potential errors that may occur during database operations or request processing.

Object-Oriented Approach:

- The frontend code (presumably React) includes a component named Prediction, which encapsulates UI elements and behavior related to predicting crop outcomes.
- The Prediction component utilizes React's component-based architecture, where each component represents a reusable and self-contained unit of UI.
- State management is implemented using React's stateful hooks (useState), allowing the Prediction component to manage and update its state in response to user interactions.
- Event handlers, such as handleChange and handleSubmit, are defined within the Prediction component to handle user input changes and form submissions.

- The component renders JSX markup to define the layout and structure of the prediction form, including input fields for various environmental factors and buttons for form submission and navigation.

Integration:

- The function-oriented backend and object-oriented frontend are integrated to create a cohesive web application. The backend functions handle user authentication and data manipulation, while the frontend component manages user interactions and displays dynamic content.

- Communication between the frontend and backend is facilitated through HTTP requests, with the frontend making requests to the backend API endpoints to perform operations like user registration, login, and data retrieval.

- This integration allows for a seamless user experience, where users interact with the frontend UI to perform actions, and the backend processes these actions and responds accordingly.

In summary, the code snippet combines function-oriented and object-oriented principles to create a full-stack web application. The backend follows a function-oriented approach to handle server-side logic and data management, while the frontend adopts an object-oriented approach to create modular and reusable UI components.

3.2 Detail Design

To provide a detailed design for EcoAgri Guide, we'll outline the architecture, components, and interactions between different parts of the system:

1. Architecture:

- The system follows a client-server architecture, with a frontend client application and a backend server.

- **Frontend:** Developed using React.js, responsible for presenting the user interface and interacting with users.

- **Backend:** Built with Node.js and Express.js, handles business logic, data processing, and serves API endpoints.
- **Database:** MongoDB is used as the database to store user data, crop information, and environmental factors.

2. Components:

- **Frontend:**
  - **User Interface Components:** Different components for user registration, login, profile management, and prediction forms.
  - **State Management:** Utilizes React hooks (useState, useEffect) for managing component-level state and context API for global state management.
  - **Forms and Input Validation:** Form components to capture user input for environmental factors and validate input data before submission.
  - **API Interaction:** Uses Axios or Fetch API to communicate with backend API endpoints for user authentication, data retrieval, and predictions.
  - **Routing:** React Router for handling client-side routing and navigation between different pages and components.
- **Backend:**
  - **Express Routes:** Define routes to handle HTTP requests for user authentication, user management, and crop prediction.
  - **Controllers:** Business logic implemented in controller functions associated with each route, responsible for handling requests, processing data, and sending responses.
  - **Middleware:** Middleware functions for handling request validation, authentication, error handling, and logging.
  - **Database Integration:** Uses Mongoose ORM to interact with MongoDB database, defining schemas, models, and performing CRUD operations on user and crop data.
  - **Machine Learning Integration:** Integrates machine learning models for crop prediction, with endpoints to receive input data, process predictions, and return results.

3. Interactions:

- **User Registration/Login:** Users interact with the frontend to register or login, submitting their credentials via forms. The frontend sends requests to the backend, which validates and processes user data, authenticates users, and sends back responses.
- **Profile Management:** Users can update their profile information through the frontend interface. The frontend sends update requests to the backend, which updates user data in the database and returns success or error responses.
- **Crop Prediction:** Users input environmental factors through the frontend prediction form. The frontend sends the input data to the backend prediction endpoint. The backend processes the data, feeds it into the machine learning model, generates predictions, and returns the results to the frontend for display.

4. Security:

- **User Authentication:** Implements secure authentication mechanisms using JWT (JSON Web Tokens) or sessions to authenticate users and protect sensitive endpoints.
- **Input Validation:** Validates user input on both the frontend and backend to prevent malicious data input and ensure data integrity.
- **HTTPS:** Uses HTTPS protocol to encrypt data transmitted between the client and server, ensuring data privacy and security.

5. Scalability and Performance:

- **Horizontal Scaling:** Utilizes load balancers and containerization (e.g., Docker, Kubernetes) to horizontally scale backend servers and handle increased traffic.
- **Caching:** Implements caching mechanisms (e.g., Redis) to cache frequently accessed data and reduce database load, improving system performance.
- **Performance Monitoring:** Integrates monitoring tools (e.g., New Relic, Prometheus) to monitor system performance, identify bottlenecks, and optimize resource utilization.

This detailed design provides a comprehensive overview of the architecture, components, interactions, and security considerations for EcoAgri Guide, ensuring a robust, scalable, and secure agricultural application.

3.3 System Design

Data Flow Diagram : A data flow diagram (DFD) is a graphical representation of the flow of data within a system, illustrating how data moves between processes, data stores, and external entities. It provides a high-level overview of the system's data flow, helping to identify inputs, outputs, and transformations of data in a visual and easily understandable format.

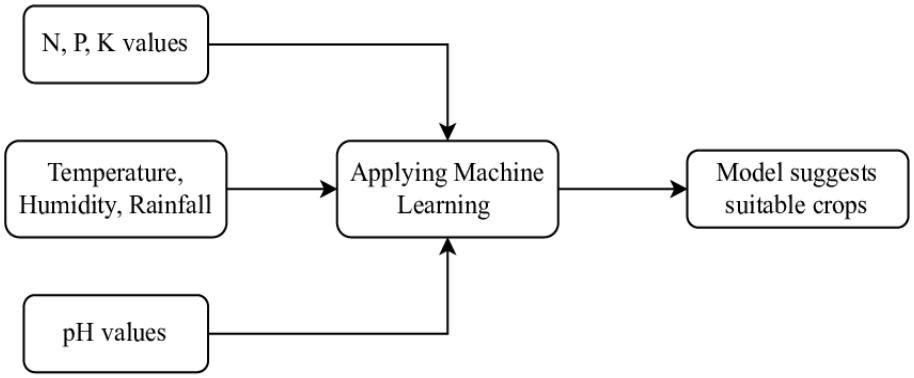

Figure 3.1: Data Flow Diagram

FLow Chart : A flowchart is a graphical representation of a process or algorithm, depicting the sequence of steps and decision points using symbols and arrows. It provides a visual overview of the workflow, enabling easy comprehension and analysis of the process's logic and structure.

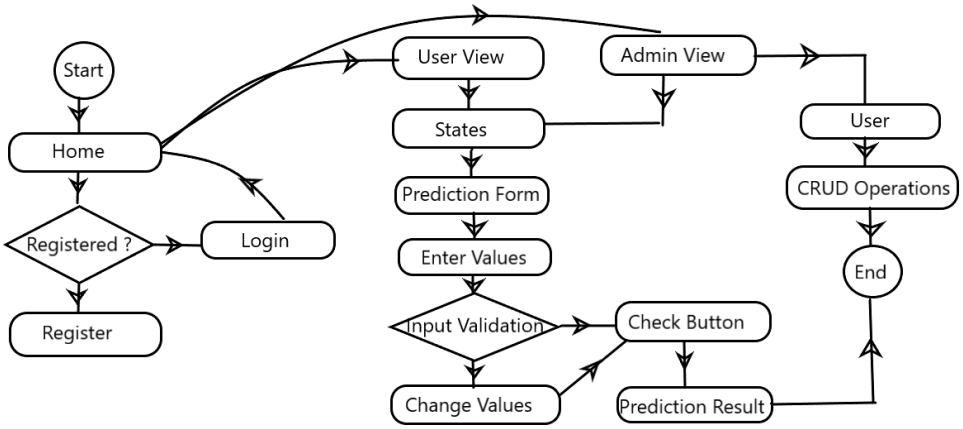

Figure 3.2: Flow Chart

3.4 User Interface Design

User Interface (UI) Design for EcoAgri Guide plays a crucial role in providing users with a seamless and intuitive experience while interacting with the application. Here's a breakdown of the UI design components and considerations:

1. User Authentication:

- **Login Form:** A simple form with input fields for email and password, allowing users to log in to their accounts.
- **Registration Form:** Enables new users to create accounts by providing necessary details such as username, email, password, etc.

2. Dashboard:

- **Home Page:** Upon successful login, users are directed to a dashboard displaying relevant information and options.
- **Navigation Menu:** Clear and intuitive navigation menu or sidebar for accessing different features and sections of the application.
- **Quick Actions:** Buttons or shortcuts for common actions such as predicting crops, managing profile, viewing recommendations, etc.

3. Crop Management:

- **Crop Selection Form:** Allows users to input environmental factors like nitrogen, phosphorus, potassium levels, temperature, humidity, pH, rainfall, etc.
- **Result Display:** Presents the prediction results in a clear and understandable format, indicating recommended crops based on input data.
- **Detailed Information:** Provides additional details about recommended crops, including growth conditions, planting tips, harvest periods, etc.

4. Responsive Design:

- Ensures the application is accessible and usable across various devices and screen sizes, including desktops, tablets, and smartphones.

- Utilizes responsive design principles such as flexible layouts, fluid grids, and media queries to adapt to different viewport sizes.

5. Visual Design:

- **Consistent Branding:** Adheres to the EcoAgri Guide branding guidelines, including logo, color scheme, typography, and imagery.
- **Clean and Minimalistic:** Emphasizes simplicity and clarity in design, avoiding clutter and unnecessary elements.
- **Visual Hierarchy:** Uses appropriate visual hierarchy to highlight important elements, actions, and information.

6. Accessibility and Usability:

- **Accessibility Standards:** Ensures compliance with accessibility standards such as WCAG (Web Content Accessibility Guidelines) to accommodate users with disabilities.
- **User-Friendly Interactions:** Designs intuitive and user-friendly interactions, with clear labels, feedback messages, and error handling.

7. Testing and Iteration:

- Conducts usability testing with real users to gather feedback and identify areas for improvement
- Iteratively improves the UI based on user feedback, usability testing results, and analytics data.

By focusing on these UI design principles and considerations, EcoAgri Guide can deliver an engaging, user-friendly, and accessible interface that enhances the overall user experience and promotes user engagement with the application.

3.5 Database Design

3.5.1 ER Diagram

The Entity Relational Model is a model for identifying entities to be represented in the database and representation of how those entities are related. The ER data model specifies enterprise schema that represents the overall logical structure of a database graphically.

The Entity Relationship Diagram explains the relationship among the entities present in the database. ER models are used to model real-world objects like a person, a car, or a company and the relation between these real-world objects.

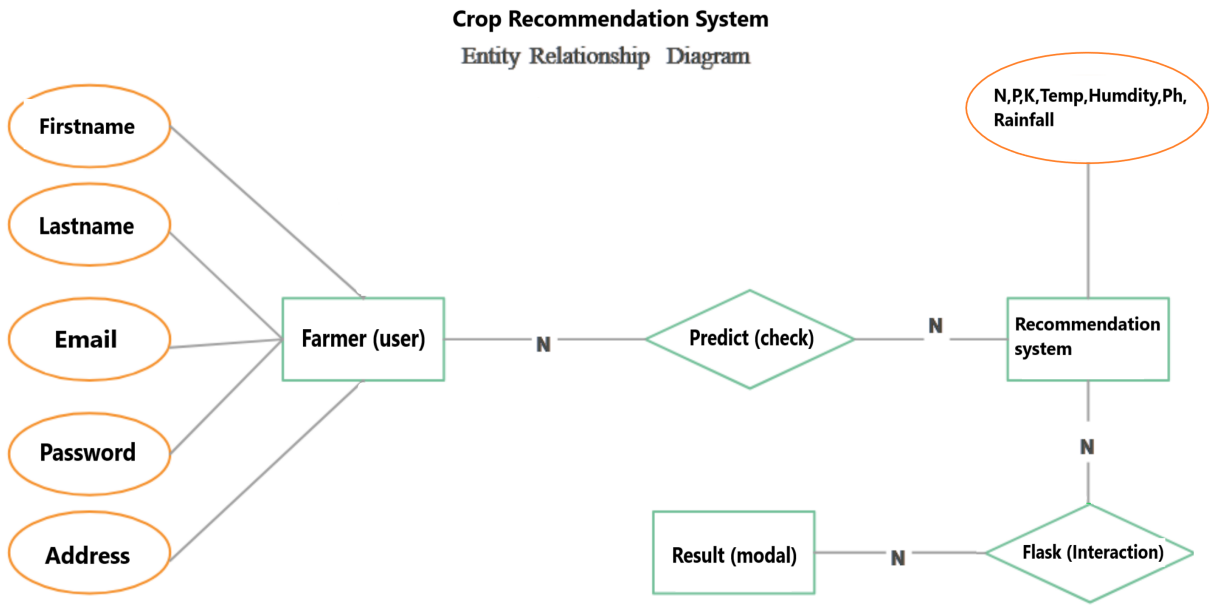

Figure 3.3: ER Diagram

3.5.2 Normalization

Normalization in the context of EcoAgri Guide refers to the process of organizing and structuring data within the application's database to minimize redundancy and dependency, ensuring data integrity and consistency. This is particularly important for handling various agricultural parameters, such as soil nutrients, climate conditions, and crop recommendations, efficiently and effectively. Here's how normalization can be applied:

Identify Entities: Begin by identifying the main entities or objects within the application, such as users, crops, farms, and environmental factors (e.g., temperature, humidity, rainfall).

Define Attributes: Determine the attributes or properties associated with each entity. For example, the "Crop" entity may have attributes such as crop name, recommended soil pH, optimal temperature range, etc.

Eliminate Redundancy: Analyze the attributes to identify any redundant or duplicated information. Redundant data should be removed or consolidated to minimize storage space and avoid inconsistencies.

Create Separate Tables: Organize the data into separate tables based on the identified entities

and their attributes. Each table should represent a single entity and store related attributes.

Establish Relationships: Define relationships between the tables using keys, such as primary keys and foreign keys, to establish connections between related entities. For example, a ”Crop Recommendation” table may have a foreign key referencing the ”Crop” table to indicate the recommended crops for specific environmental conditions.

Apply Normal Forms: Normalize the database tables to ensure they adhere to the principles of normalization, such as First Normal Form (1NF), Second Normal Form (2NF), and Third Normal Form (3NF). This involves removing repeating groups, ensuring each attribute is atomic, and eliminating transitive dependencies.

Optimize Performance: Consider performance optimization techniques, such as indexing frequently queried columns and partitioning large tables, to improve database query performance and efficiency.

By following these steps and principles of normalization, EcoAgri Guide can create a well-structured and efficient database schema that facilitates data management, retrieval, and analysis, ultimately enhancing the application's functionality and user experience.

3.5.3 Database Manipulation

Database manipulation refers to the process of performing various operations on a database to manage and manipulate data stored within it. In the context of EcoAgri Guide, which involves agricultural data management, database manipulation may include the following operations:

Data Insertion: Adding new records or data entries into the database. For example, inserting information about a new crop variety, farm location, or environmental parameter.

Data Retrieval: Retrieving specific data from the database based on predefined criteria. This could involve querying the database to fetch information about recommended crops for a particular region, historical weather data, or soil nutrient levels.

Data Update: Modifying existing data in the database. This may involve updating crop recommendations based on new research findings, editing farm details, or adjusting environmental thresholds.

Data Deletion: Removing unwanted or obsolete data from the database. For instance, deleting outdated crop records, redundant farm entries, or irrelevant environmental data.

Data Transformation: Transforming data into a different format or structure to meet specific

requirements. This could involve converting raw sensor data into standardized formats, aggregating data for analysis, or normalizing data for consistency.

Data Migration: Moving data between different databases or storage systems. This may be necessary when upgrading database software, transitioning to a new database platform, or consolidating data from multiple sources.

Data Validation: Ensuring the accuracy, consistency, and integrity of data within the database. This involves implementing validation rules and constraints to prevent invalid or erroneous data from being entered into the database.

Data Backup and Recovery: Creating regular backups of the database to protect against data loss or corruption. In the event of a database failure or data breach, being able to recover and restore data from backups is crucial for maintaining data integrity and continuity of operations.

Data Security: Implementing security measures to protect sensitive data stored in the database. This includes authentication, authorization, encryption, and auditing mechanisms to safeguard against unauthorized access, data breaches, and data leaks.

3.5.4 Database Connection Controls and Strings

Database connection controls and strings are essential components for establishing and managing connections between applications and databases. They typically include parameters such as the database host, port, username, password, and database name, which are used to authenticate and access the database. These controls and strings are crucial for ensuring secure and reliable communication between the application and the database server, facilitating data retrieval, manipulation, and storage operations.

```
mongoose.connect(process.env.DB_URL)
.then((d)=>{
    console.log("Database Connected");
    app.listen(process.env.PORT,()=>{
        console.log('Server running at port : '+process.env.PORT);
    });
})
.catch((e)=>{
    console.log("Database connection error");
});
```

Figure 3.4: Connections

3.6 Methodology

1. Dataset

In this article, we utilized the Crop Recommendation Dataset available at https://www.kaggle.com/dataset recommendation-dataset (accessed on 21 August 2023). This dataset was compiled by integrating datasets on rainfall, climate, and fertilizer data specifically for India. It serves as the foundation for building a predictive model to recommend the most suitable crops to grow on a particular farm based on various parameters.

2. Data Preparation for Decision Tree Model

Data preparation for training a decision tree model involves several critical steps to ensure the successful learning from the provided variables, including features and targets.

- **Data Collection and Organization:** The first step is to collect and organize the dataset, ensuring it contains relevant features such as temperature, humidity, pH levels, rainfall, and soil nutrients. These features serve as input variables for the decision tree model.
- **Feature Engineering:** Next, feature engineering may be performed to extract or derive additional features from the raw data, enhancing the model's predictive power. This could include calculating averages, aggregating data over time periods, or transforming categorical variables into numerical representations.
- **Data Splitting:** The dataset is divided into two subsets: a training set and a testing set. The training set is used to train the decision tree model, while the testing set is reserved for evaluating its performance.
- **Model Training:** The decision tree model is trained using the training dataset. The model learns to partition the feature space based on the provided features and their relationships with the target variable, which in this case, is the recommended crop.
- **Model Evaluation:** After training, the performance of the decision tree model is evaluated using the testing dataset. Evaluation metrics such as accuracy, precision, recall, and F1-score may be calculated to assess the model's effectiveness in recommending crops.

3. Model Training and Evaluation

After preparing the dataset and training the decision tree model, the next steps involve model training and evaluation.

- **Model Training:** The decision tree model is trained using the training dataset. During training, the model recursively partitions the feature space based on the selected features to minimize impurity or maximize information gain.

- **Model Evaluation:** Once trained, the decision tree model is evaluated using the testing dataset. The model's performance is assessed using various evaluation metrics, such as accuracy, precision, recall, and F1-score, to measure its effectiveness in recommending crops based on the input parameters.

- **Model Deployment:** Finally, the trained decision tree model is deployed for use in the EcoAgri Guide application, where it can provide crop recommendations based on user input data. Regular monitoring and maintenance of the deployed model ensure its continued accuracy and reliability.

4 Implementation, Testing, and Maintenance

4.1 Introduction to Languages, IDE's, Tools and Technologies used for Implementation

1. Languages:

- **Python:** Python is used for implementing machine learning algorithms, data preprocessing, and backend development using frameworks like Flask.
- **JavaScript:** JavaScript is utilized for frontend development with frameworks like React.js and for backend development with Node.js.

2. IDEs (Integrated Development Environments):

- **Visual Studio Code:** Visual Studio Code (VS Code) is a popular code editor used for writing, debugging, and testing code in Python, JavaScript, and other languages. It provides a rich set of features, including syntax highlighting, code completion, and integrated terminal.
- **Jupyter Notebook:** Jupyter Notebook is an interactive computing environment that allows users to create and share documents containing live code, equations, visualizations, and narrative text. It is commonly used for exploratory data analysis, prototyping machine learning models, and presenting research findings.

3. Databases:

- **MongoDB:** MongoDB is a NoSQL database that stores data in JSON-like documents with dynamic schemas. It is well-suited for storing unstructured or semi-structured data and is commonly used in web applications for its scalability and flexibility.

4. Frameworks and Libraries:

- **React.js:** React.js is a JavaScript library for building user interfaces, particularly single-page applications. It allows developers to create reusable UI components and efficiently manage the state of the application.

- **Node.js:** Node.js is a runtime environment that allows JavaScript code to be executed server-side. It is commonly used for building scalable and efficient backend systems, APIs, and web servers.

- **Machine Learning Libraries:** Various Python libraries are used for implementing machine learning algorithms, including scikit-learn, TensorFlow, and PyTorch. These libraries provide tools and APIs for data preprocessing, model training, evaluation, and deployment.

5. ML Tools and Technologies:

- **scikit-learn:** scikit-learn is a popular machine learning library in Python that provides simple and efficient tools for data mining and data analysis. It includes various algorithms for classification, regression, clustering, dimensionality reduction, and model selection.

4.2 Coding Standards of Language Used

Coding standards for the languages used in EcoAgri Guide:

1.Python:

- **Follow PEP 8:** Python Enhancement Proposal 8 (PEP 8) outlines the coding conventions for Python code. Adhering to PEP 8 ensures consistency and readability in Python code.

- **Use meaningful variable and function names:** Variables and functions should have descriptive names that convey their purpose and functionality.

- **Write modular code:** Break down functionality into smaller, reusable functions and modules to promote code reusability and maintainability.

- **Document code:** Use docstrings to provide documentation for classes, functions, and modules. Documenting code helps other developers understand its purpose and usage.

2. JavaScript:

- **Follow JavaScript Standard Style:** JavaScript Standard Style is a coding style guide and linter that defines a set of rules for writing clean and consistent JavaScript code.

- **Use camelCase for variables and function names:** Follow the convention of using camelCase for naming variables and functions in JavaScript.

- **Properly handle asynchronous code:** Use promises, async/await, or callback functions to handle asynchronous operations in JavaScript to avoid callback hell and improve code readability.
- **Avoid global variables:** Minimize the use of global variables to prevent unintended side effects and improve code maintainability.

3. React.js:

- **Follow React's JSX style guide:** Adhere to React's JSX style guide, which recommends using JSX syntax for defining React components and adhering to JavaScript coding standards within JSX.
- **Component organization:** Organize React components into reusable and composable units. Follow the principles of component-based architecture to create modular and maintainable code.
- **Use state and props appropriately:** Follow React's guidelines for managing component state and props. Use state for mutable data that affects the component's behavior and props for passing data from parent to child components.

4. Node.js:

- **Follow Node.js core style guide:** Node.js provides its own style guide for writing JavaScript code. Adhere to this guide for Node.js-specific conventions and best practices.
- **Use npm packages wisely:** Utilize npm packages for common tasks and functionalities, but carefully evaluate dependencies to avoid unnecessary bloat and potential security vulnerabilities.
- **Error handling:** Implement proper error handling in Node.js applications to gracefully handle errors and prevent crashes. Use try-catch blocks, error objects, and error middleware to handle errors effectively.

Adhering to these coding standards ensures consistency, readability, and maintainability across the EcoAgri Guide codebase, making it easier for developers to collaborate, maintain, and extend the application over time.

4.3 Testing Techniques and Test Plans

Testing Techniques and Test Plans for EcoAgri Guide:

1. Prediction Testing:

- **Unit Testing:** Test individual machine learning models and algorithms to ensure they produce accurate predictions based on input data. Use sample datasets with known outcomes to verify the correctness of predictions.
- **Integration Testing:** Test the integration of machine learning models with the application's backend logic. Verify that predictions are correctly passed from the backend to the frontend and displayed accurately to users.
- **Regression Testing:** Periodically retest prediction functionalities after updates or changes to ensure that previous predictions remain accurate and unchanged.

2. Database Testing:

- **Unit Testing:** Test database operations such as CRUD (Create, Read, Update, Delete) operations for each collection or table in the database. Verify that data is stored, retrieved, updated, and deleted correctly.
- **Integration Testing:** Test the integration between the application and the database. Verify that data is properly synchronized between the frontend, backend, and database layers.
- **Performance Testing:** Evaluate the database's performance under different loads and stress levels. Measure response times for data retrieval, insertion, and updates to ensure optimal database performance.

3. Frontend Testing:

- **Unit Testing:** Test individual React components using tools like Jest and Enzyme. Verify that components render correctly, respond to user interactions, and update state as expected.
- **Integration Testing:** Test the interaction between different React components and ensure they work together seamlessly. Verify that user inputs are processed correctly and reflected in the UI.

- **User Acceptance Testing (UAT):** Conduct UAT with real users to validate that the frontend meets their expectations in terms of usability, functionality, and user experience.
- **Cross-browser Testing:** Test the application on different web browsers (e.g., Chrome, Firefox, Safari) to ensure compatibility and consistency across platforms.

Test Plans:

Prediction Test Plan:

Identify sample datasets with known outcomes for testing prediction accuracy. Define test cases to cover various scenarios and edge cases. Execute test cases and compare predicted outcomes with expected results. Record and analyze test results, identifying any discrepancies or inaccuracies in predictions.

Database Test Plan:

Create test cases for CRUD operations on each database collection or table. Execute test cases to verify data storage, retrieval, update, and deletion. Monitor database performance metrics during load testing to ensure scalability and efficiency. Validate data integrity and consistency across different layers of the application.

Frontend Test Plan:

Define test cases for each React component, covering rendering, interaction, and state management. Conduct integration testing to verify the interaction between different components and UI elements. Perform user acceptance testing with real users to gather feedback on usability and user experience. Validate cross-browser compatibility and responsiveness across various devices and screen sizes.

By implementing these testing techniques and test plans, EcoAgri Guide can ensure the reliability, functionality, and performance of its prediction, database, and frontend components, providing users with a seamless and satisfying experience.

5 Results and Discussions

5.1 User Interface Representation

User Interface (UI) Representation refers to the visual and interactive elements of a software application or system that users interact with to perform tasks and access information. It includes graphical user interfaces (GUIs), command-line interfaces (CLIs), and other forms of interaction that facilitate communication between users and the underlying software.

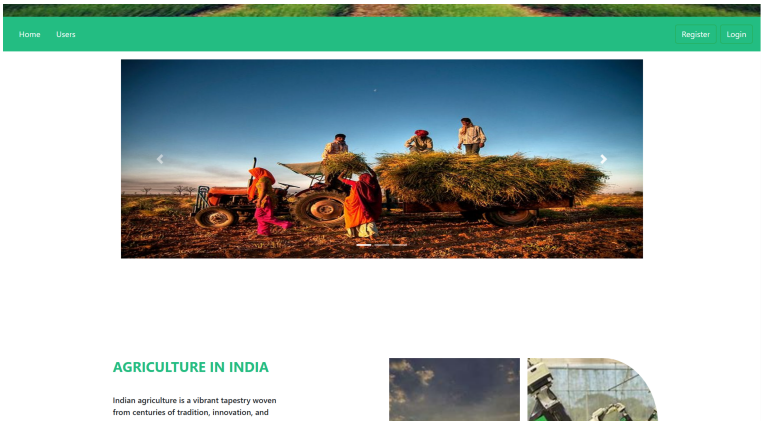

Figure 5.5: User interface

5.2 Snapshots of system

User Interface : It focuses on presenting agricultural data, crop recommendations, and environmental insights in a user-friendly and intuitive manner, enabling farmers to easily navigate the application and make informed decisions.

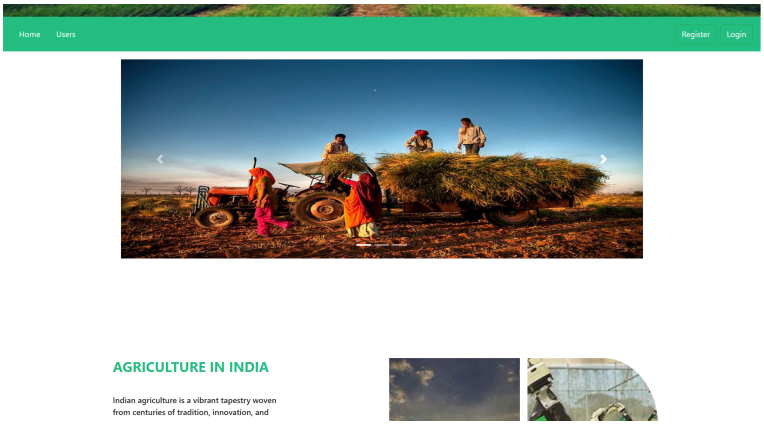

Figure 5.6: User Interface

Navbar : It allow users to navigate to different sections or pages within the application. A navbar can provide users with easy access to various features and functionalities of the application.

Figure 5.7: Navbar

Register : Fill out the registration form with the necessary information. This typically includes details such as: First name , Last name , Email address , Password , Confirm password

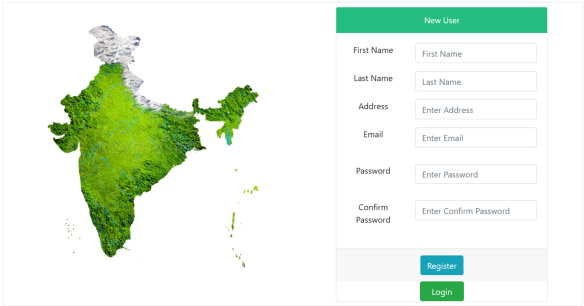

Figure 5.8: Register

Login : After entering your crThe system will verify the provided credentials. If the email address and password combination is correct, you will be authenticated, and access will be granted to your EcoAgri Guide account.

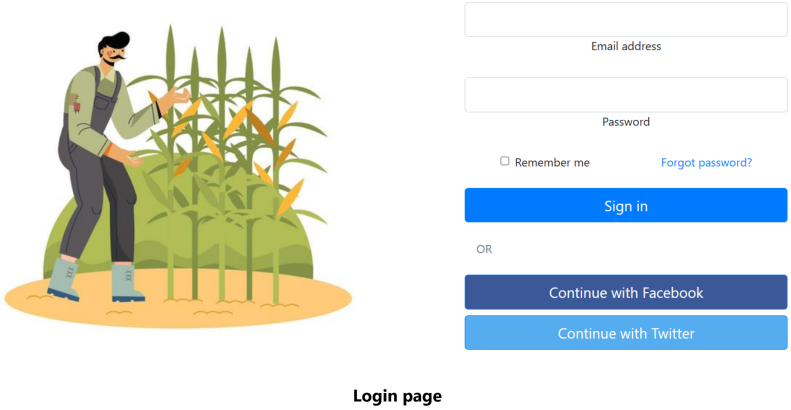

Figure 5.9: Login

Users : If you have administrative privileges, log in to the admin dashboard of EcoAgri Guide and then navigate to the users .

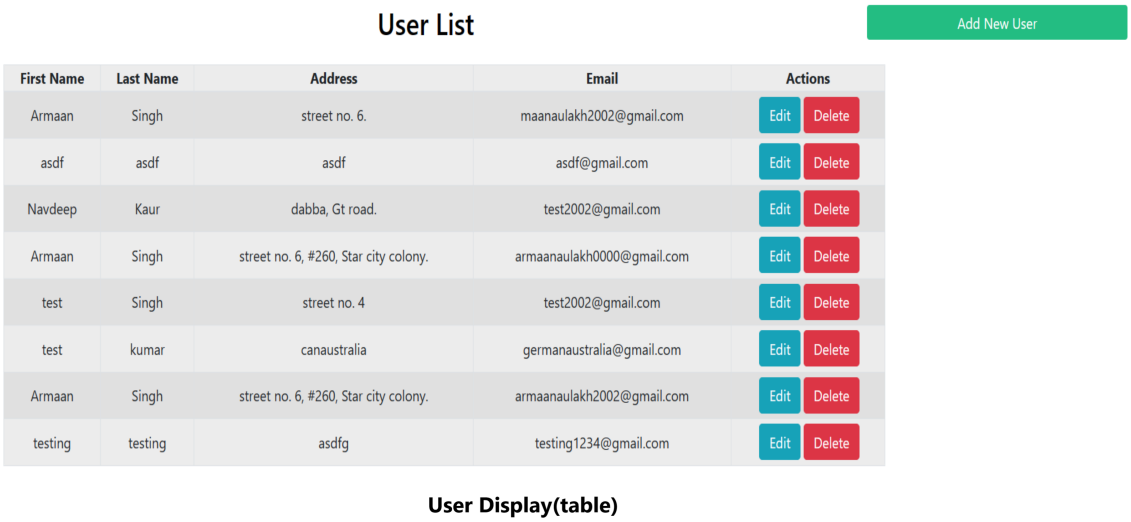

Figure 5.10: Users

States : The states that are included in our project

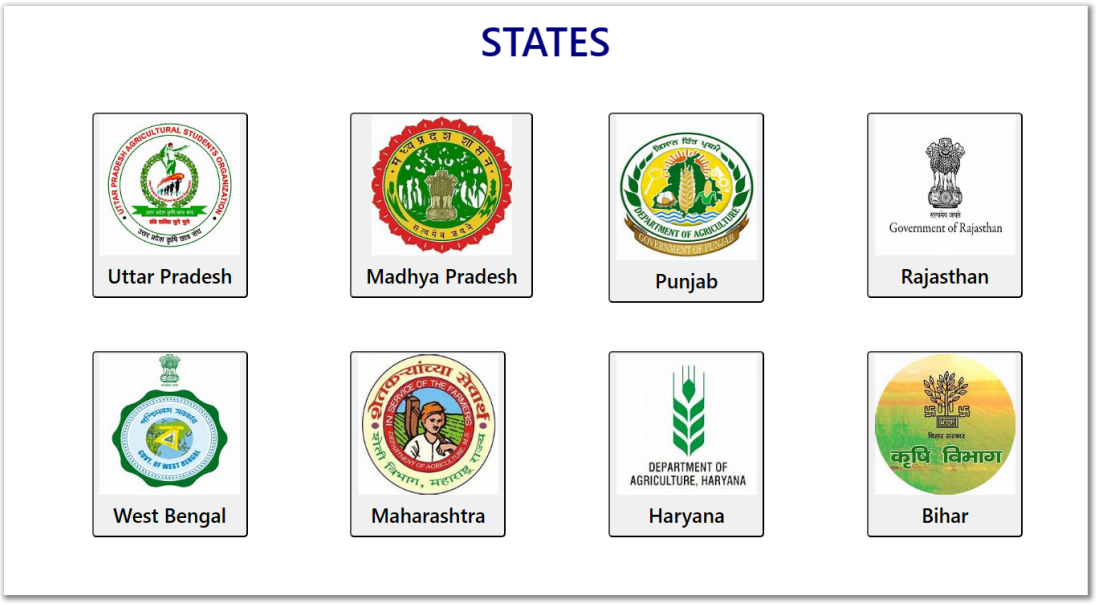

Figure 5.11: States

5.3 Back Ends Representation

EcoAgri Guide utilizes a robust back-end infrastructure, relying on MongoDB as its primary database system. MongoDB is a NoSQL database known for its flexibility, scalability, and high

performance, making it well-suited for handling the diverse and complex data requirements of agricultural systems. The document-oriented nature of MongoDB allows EcoAgri Guide to store and retrieve data in a flexible and efficient manner, accommodating various types of information, including crop recommendations, environmental data, user profiles, and more. Additionally, MongoDB's support for distributed architecture enables EcoAgri Guide to scale horizontally as its user base and data volume grow, ensuring optimal performance and reliability. By leveraging MongoDB as its back-end database, EcoAgri Guide can efficiently manage and manipulate agricultural data, supporting informed decision-making and sustainable farming practices for users across different regions and environments.

5.3.1 Snapshots of Database Tables

Users : In MongoDB, a "users" collection (equivalent to a table in relational databases) typically stores information about registered users of an application. Here's an example of how a "users" collection might be structured in MongoDB:

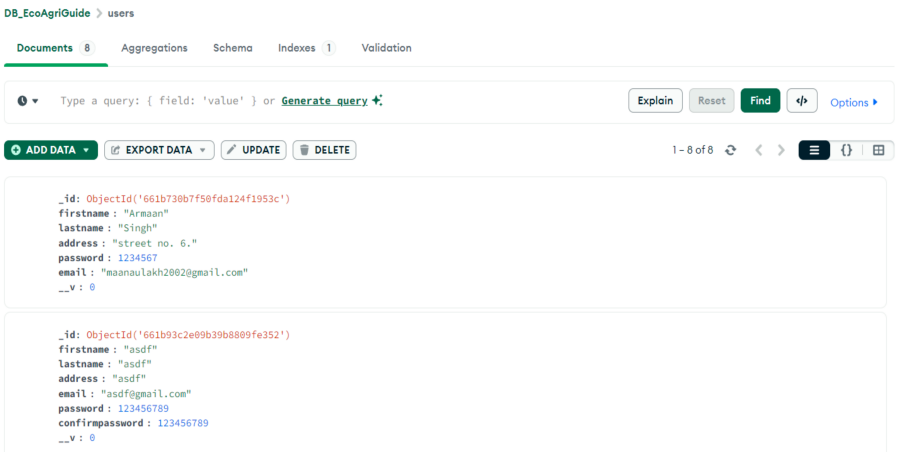

Figure 5.12: User Table

5.3.2 Snapshots of Machine Leanring Model :

1. Dataset Used : The dataset utilized in EcoAgri Guide is sourced from Kaggle, providing comprehensive information on crop recommendations for agricultural regions. Augmented with data on rainfall, climate etc in India, it enables the creation of predictive models for suggesting suitable crops based on various parameters, fostering informed decision-making and sustainable farming practices.

	A	B	C	D	E	F	G	H
1	N	P	K	temperatu	humidity	ph	rainfall	label
2	90	42	43	20.87974	82.00274	6.502985	202.9355	rice
3	85	58	41	21.77046	80.31964	7.038096	226.6555	rice
4	60	55	44	23.00446	82.32076	7.840207	263.9642	rice
5	74	35	40	26.4911	80.15836	6.980401	242.864	rice
6	78	42	42	20.13017	81.60487	7.628473	262.7173	rice
7	69	37	42	23.05805	83.37012	7.073454	251.055	rice
8	69	55	38	22.70884	82.63941	5.700806	271.3249	rice
9	94	53	40	20.27774	82.89409	5.718627	241.9742	rice
10	89	54	38	24.51588	83.53522	6.685346	230.4462	rice

Figure 5.13: Dataset

2. Confusion Matrix : A confusion matrix is a table used in machine learning to evaluate the performance of a classification model. It displays the counts of true positive, false positive, true negative, and false negative predictions made by the model. This matrix helps assess the model's accuracy, precision, recall, and other performance metrics by comparing predicted and actual class labels.

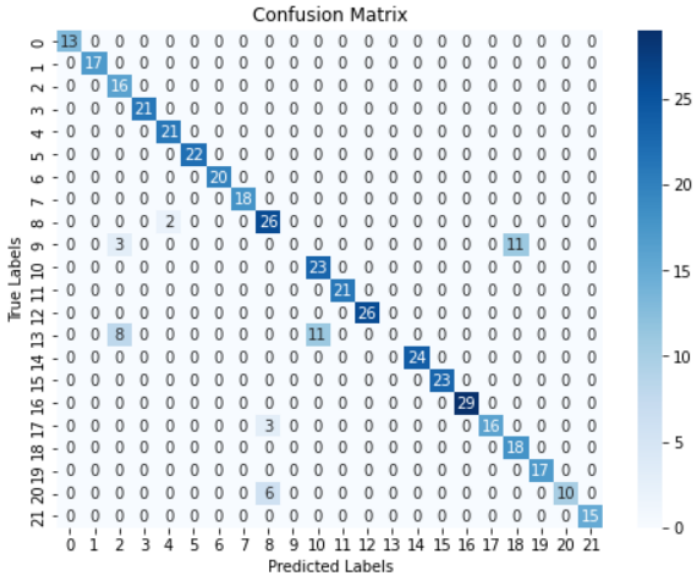

Figure 5.14: Confusion Matrix

3. Classification Report :A classification report provides a summary of key performance metrics for a classification model, including precision, recall, F1-score, and support for each class. It offers insights into the model's ability to correctly classify instances of each class, highlighting areas of strength and areas for improvement. This report aids in understanding the overall effectiveness and reliability of the classification model across different classes or categories.

```
print(classification_report(y_test,predicted_values))
```

```
Decision Tree's Accuracy is :  90.0
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       0.59      1.00      0.74        16
    chickpea       1.00      1.00      1.00        21
     coconut       0.91      1.00      0.95        21
      coffee       1.00      1.00      1.00        22
      cotton       1.00      1.00      1.00        20
      grapes       1.00      1.00      1.00        18
        jute       0.74      0.93      0.83        28
 kidneybeans       0.00      0.00      0.00        14
      lentil       0.68      1.00      0.81        23
       maize       1.00      1.00      1.00        21
       mango       1.00      1.00      1.00        26
   mothbeans       0.00      0.00      0.00        19
    mungbean       1.00      1.00      1.00        24
   muskmelon       1.00      1.00      1.00        23
      orange       1.00      1.00      1.00        29
      papaya       1.00      0.84      0.91        19
  pigeonpeas       0.62      1.00      0.77        18
 pomegranate       1.00      1.00      1.00        17
        rice       1.00      0.62      0.77        16
  watermelon       1.00      1.00      1.00        15

    accuracy                           0.90       440
   macro avg       0.84      0.88      0.85       440
weighted avg       0.86      0.90      0.87       440
```

Figure 5.15: Classification Report

4. Correlation Map :A correlation map visually represents the pairwise correlation between different variables in a dataset using colors or numerical values. It helps identify patterns, relationships, and dependencies between variables, aiding in feature selection, dimensionality reduction, and understanding the data structure.

```
In [17]: sns.heatmap(df_new.corr(),annot=True)
Out[17]: <AxesSubplot:>
```

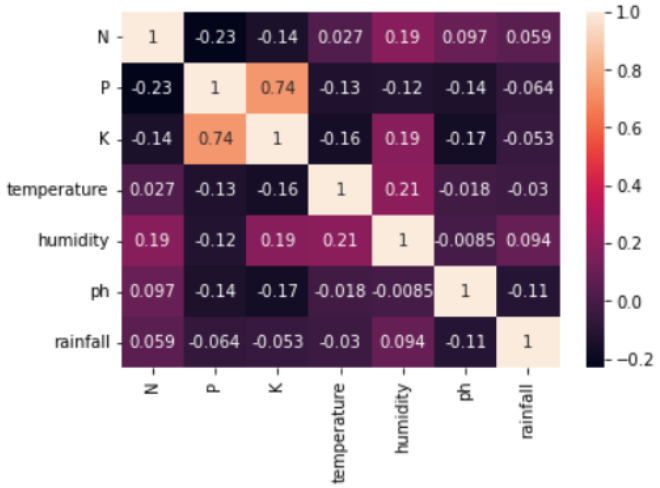

Figure 5.16: Correlation Map

5. Labels : Labels represent the recommended crop types based on environmental factors and soil conditions. These labels provide actionable insights for farmers, guiding them towards sus-

tainable crop selection decisions to optimize agricultural productivity and environmental conservation.

```
In [13]: df['label'].value_counts()
Out[13]: rice           100
         maize          100
         jute           100
         cotton         100
         coconut        100
         papaya         100
         orange         100
         apple          100
         muskmelon      100
         watermelon     100
         grapes         100
         mango          100
         banana         100
         pomegranate    100
         lentil         100
         blackgram      100
         mungbean       100
         mothbeans      100
         pigeonpeas     100
         kidneybeans    100
         chickpea       100
         coffee         100
         Name: label, dtype: int64
```

Figure 5.17: Labels

6 Conclusion and Future Scope

6.1 Conclusion

In conclusion, EcoAgri Guide represents a pivotal advancement in sustainable agriculture, leveraging innovative technologies such as machine learning and data analysis to empower farmers with actionable insights for environmentally responsible crop selection. By integrating diverse environmental factors and soil conditions, EcoAgri Guide facilitates informed decision-making, fostering a harmonious relationship between agriculture and the ecosystem. Its user-friendly interface, real-time monitoring capabilities, and personalized recommendations underscore its commitment to promoting sustainable farming practices. As agriculture faces increasing challenges from climate change and resource depletion, EcoAgri Guide stands as a beacon of hope, paving the way towards a greener and more resilient future for agriculture.

6.2 Future Scope

Looking ahead, the future scope of EcoAgri Guide extends beyond its current functionalities, presenting opportunities for further enhancement and expansion. One avenue for growth lies in integrating e-commerce capabilities directly into the platform, enabling farmers to not only access crop recommendations but also procure seeds, fertilizers, and other agricultural inputs seamlessly. By establishing partnerships with suppliers and retailers, EcoAgri Guide can offer a one-stop solution for farmers, streamlining the procurement process and enhancing convenience. Moreover, incorporating a feature to visualize prediction history directly on the frontend can provide valuable insights for users. By displaying past predictions alongside actual outcomes, farmers can track the accuracy and effectiveness of the recommendations over time. This historical data visualization not only builds trust in the predictive models but also facilitates continuous improvement by identifying patterns and trends in crop performance. Additionally, interactive charts and graphs can enable users to explore historical data dynamically, gaining deeper insights into seasonal variations, climate trends, and other factors influencing agricultural outcomes.
By embracing these future-oriented enhancements, EcoAgri Guide can further solidify its position as a comprehensive platform for sustainable agriculture, empowering farmers with advanced tools and resources to thrive in an ever-changing agricultural landscape.

References

[1] O. Jadhav, ”Crop Recommendation,” Kaggle, [Online]. Available: https://www .kaggle.com/code/omjadhavkaggle/crop-recommendationDecision-Tree.Accessed: February, 2024.

[2] K. S. Kalyani, ”Precision Agriculture Using Machine Learning Techniques,” International Journal of Computer Applications, vol. 184, no. 24, pp. 1-6,Accessed: February, 2024.

[3] M. Otto and J. Thornton, ”Getbootstrap,” [Online]. Available: https://get bootstrap.com/docs/4.0/getting-started/introduction/. Accessed: February, 2024

[4] J. Ramya and M. Sankar, ”Crop Recommendation Using Supervised Learning Techniques,” in Proceedings of the 2023 4th International Conference on Smart Electronics and Communication (ICOSEC), 2023.

[5] G. Ranganathan, Y. El Allioui, and S. Piramuthu, Eds., ”Soft Computing for Security Applications,” Springer Science and Business Media LLC, 2023.

[6] P. A. Barvin and T. Sampradeepraj, ”Crop Recommendation Systems based on soil and environmental factors using graph convolutional neural network: A Systematic Literature Review,” in Proceedings of ECSA 2023, 2023.

[7] D. Ukolov and O. Ukolova, ”OpenWeatherMap,” OpenWeather, [Online]. Available: https://openweathermap.org/.Accessed: March, 2024.