Major Project Final Presentation on

Kheti Sahayak

The future of Farming, Powered by Machine Learning, Web Design and API

SUPERVISOR

DR. B. KIRAN KUMAR
ASSOCIATE PROFESSOR

COORDINATOR

SRI. M. KISHORE
ASSISTANT PROFESSOR

PRESENTED BY,

BLAISE B20IT018
AKSHAY B20IT009
SHRIMAYEE B20IT039
ASHWAK B21IT122L

Abstract

- ➢ Farmers face challenges with limited tech knowledge, affecting weather, labor, and market access, impacting resource management and crop yield.
- ➢ Absence of technology hampers effective farming, hindering weather and market management, impacting resource use, crop yield, and sustainability.
- ➢ Smart farming, driven by machine learning, empowers farmers with precise strategies, weather forecasts, crop monitoring, and disease detection.
- ➢ Automation reduces labor, while market insights and knowledge sharing enhance productivity, giving farmers control over operations and livelihoods.

Proposed Solution

- ➢ The agriculture sector in India faces challenges from erratic climate patterns, impacting crop yields and hindering effective management.
- ➢ Farmers, lacking technological resources, struggle to optimize practices, affecting productivity and food security in agricultural operations.
- ➢ Kheti Sahayak, employing machine learning, predicts crop yields and offers a user-friendly platform for informed decision-making and optimized management.
- ➢ Empowering farmers, the platform provides real-time monitoring, data analysis, recommendations, and community engagement, transforming agriculture for improved productivity and sustainability.

Project Modules

- **Module 1 - Collaborative community for shared learning**

  Kheti Sahayak fosters collaborative learning, connecting farmers for shared experiences and collective problem-solving in the agricultural community.

- **Module 2 - Predict crop yields using machine learning**

  Using machine learning, Kheti Sahayak predicts crop yields, empowering farmers with valuable insights for effective planning and decision-making.

Project Modules Contd.

- **Module 3 - Personalized fertilizer recommendations for optimal growth**

  Personalized fertilizer recommendations optimize growth by tailoring suggestions to individual farm conditions, enhancing resource utilization and overall productivity.

- **Module 4 - Detect and prevent crop diseases**

  The platform also addresses crop health, detecting and preventing diseases to safeguard crops, contributing to improved agricultural sustainability and livelihoods.

Literature Survey

REFERENCE PAPER SUMMARIES

S. No.	AUTHOR	PAPER	METHODOLOGY	ADVANTAGES	DISADVANTAGES
01.	Zeynep Unal	Smart Farming Becomes Even Smarter With Deep Learning A Bibliographical Analysis	Implement deep learning for crop disease detection and plant classification using methods like CNN and transfer learning.	Enhanced disease detection. Automated yield estimation.	Limited specifics on methodologies.
02.	Senthil Kumar Swami Durai Mary Divya Shamili	Smart farming using Machine Learning and Deep Learning techniques	Utilizes IoT, Data Mining, Data Analytics, Machine Learning, and Deep Learning for precision agriculture.	Increased productivity, reduced manual labor, smart farming guidance.	High initial investment, dependence on technology, potential data security risks.
03.	Abhinav Sharma, Arpit Jain, Prateek Gupta	ML Applications for Precision Agriculture: A Comprehensive Review	Implement ML, DL, and IoT for soil prediction, disease detection, and intelligent farming.	Enhanced crop yield, smart irrigation, livestock prediction.	Reliance on sensors, potential cost, and data security concerns.

Literature Survey Contd.

04.	Jakia Sultana, M. N. A. Siddique, Md. Rishad Abdullah	Fertilizer recommendation for Agriculture: practice, practicalities and adaptation in Bangladesh and Netherlands	Literature research combined with personal experience and contact with advisors of BLGG AgroXpertus and SRDI.	Provides insights into the fertilizer recommendation practices in Bangladesh and the Netherlands.	Highlights the need for adjusting and optimizing fertilizer recommendations based on local conditions. Limited information on specific details of fertilizer recommendations. Relies on data and information available up to 2015.
05.	Tiago Domingue, Tomás Brandão, João C. Ferreira	Machine Learning for Detection and Prediction of Crop Diseases and Pests: A Comprehensive Survey	Literature Review	Promotes ML in agriculture for disease and pest detection.	Emphasizes precision farming and smart agriculture. Limited focus on specific ML techniques. No new empirical findings.
06.	Annafi' Franz, Eko Junirianto, Suswanto	Web Design and Application Programming Interface (API) Smart Farming Application	Waterfall Model: Analysis, Design, Coding, Testing, Maintenance.	Efficient data processing, effective planting management, increased productivity.	Requires understanding of REST principles, potential security concerns with data access. Requires accurate data input for accurate feasibility detection.
07.	Jehad Ali, Rehanullah Khan, Nasir Ahmad	Random Forests and Decision Trees	Compared Random Forest and J48 for classifying datasets. Used 10-fold cross-validation.	Improved accuracy for large datasets. Handles missing values.	May not perform as well on small datasets. Random Forest complexity.
08.	Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun	Deep Residual Learning for Image Recognition	Residual learning with shortcut connections, reformulating layers as learning residual functions.	Enables training of very deep networks. Eases optimization with improved accuracy.	Optimization difficulties in deep plain nets. Potential overfitting in aggressively deep models.

Literature Survey Contd.

09.	Andy Neumann, Nuno Laranjeiro, Jorge Bernardino	An Analysis of Public REST Web Service APIs	Examined Alexa.com's top 4000 sites, identified 500 claiming REST APIs. Analyzed 26 features manually, covering compliance, common decisions, and best practices.	Provides insights into REST service compliance, common practices, and best practices. Identifies trends and challenges.	Limited to a subset of web services, potential bias in random selection. Manual analysis may introduce human error. Data may become outdated quickly due to the dynamic nature of the web.
10.	Aashis Rimal	Developing a Web Application on NodeJS and MongoDB using ES6 and Beyond	Full Stack JavaScript	Speedy development with NodeJS. Scalability with MongoDB. Unified language (JavaScript) for both frontend and backend.	NodeJS limitations in handling complex computations. Challenges in MongoDB's document model. Payment processing issues with Stripe and CSRF protection.

Technologies Used

- **Front End:** HTML, CSS, Bootstrap, JavaScript
- **Back End:** Node JS, Express JS, EJS, MongoDB, Mongoose
- **Authentication:** Passport JS
- **Sessions:** Express Sessions, Connect Mongo
- **Media Upload:** Cloudinary
- **Validation:** Joi
- **Security:** Express Mongo Sanitize, Helmet, Sanitize HTML

Technologies Used Contd.

- **Python:** Used for developing the backend logic for recommendation and detection systems.
- **Pandas, NumPy, Matplotlib:** Python libraries used for data cleaning, manipulation, analysis, and visualization.
- **Scikit-Learn:** A Machine Learning library employed for selecting the best algorithm by comparing various predefined ones for the crop recommendation model.
- **ResNet9:** A Convolutional Neural Network architecture specifically employed for plant disease detection.
- **Render:** A platform used to deploy the web application online.

Algorithms Used

- **Crop Recommendation Algorithm (Random Forest):**
  Utilizes Random Forest, an ensemble method, to predict optimal crops based on soil, pH, rainfall, and weather data, offering localized and accurate farming recommendations.

- **Fertilizer Recommendation Algorithm (Custom Logic):**
  Applies custom logic to analyze nutrient disparities and recommends fertilizers tailored to address specific deficiencies or excesses, aiding informed decisions for crop optimization.

- **Plant Disease Classification (Deep Learning with ResNet):**
  Employs ResNet, a deep learning architecture, to classify plant diseases from uploaded images, facilitating early and precise diagnosis for prompt treatment and crop protection.

Stages of Module I Development

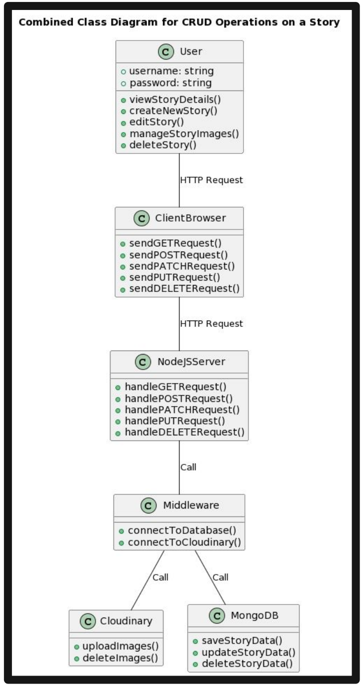
DESCRIPTION: COLLABORATIVE COMMUNITY FOR SHARED LEARNING

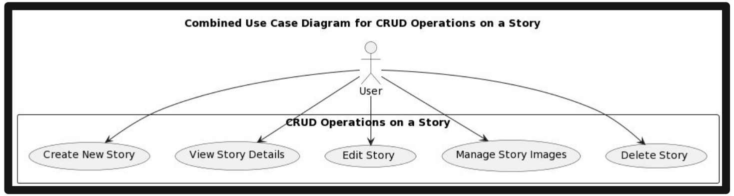
STAGE 1: CREATING A SIMPLE CRUD WEB APPLICATION USING NODE JS, EXPRESS AND MONGODB

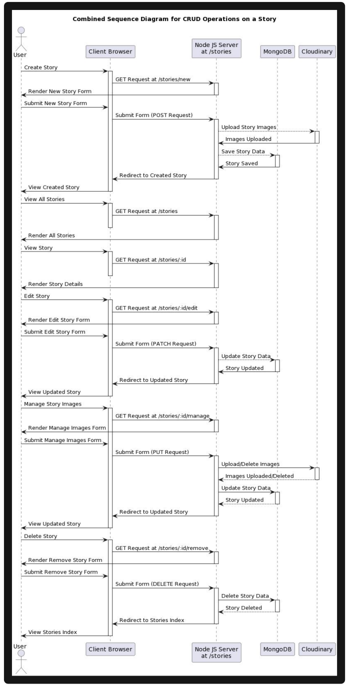
STAGE 2: ADD DATA VALIDATION, USER AUTHENTICATION, USER AUTHORISATION AND UPLOADING MEDIA

STAGE 3: ENHANCING SECURITY AND DEPLOYING WEB APPLICATION

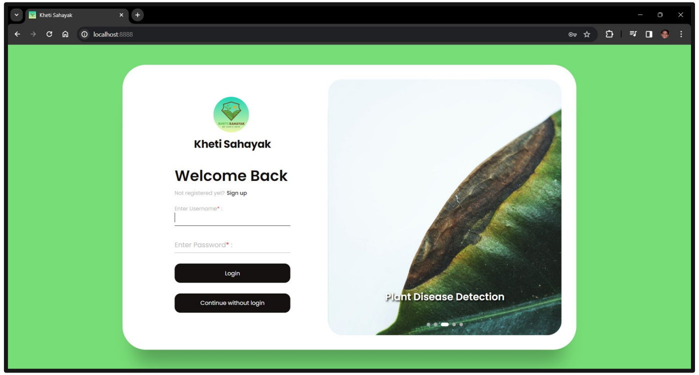
UML Diagrams of Module I



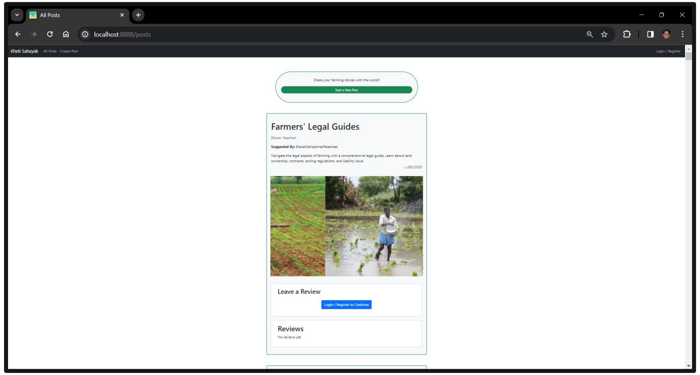




Output of Module I

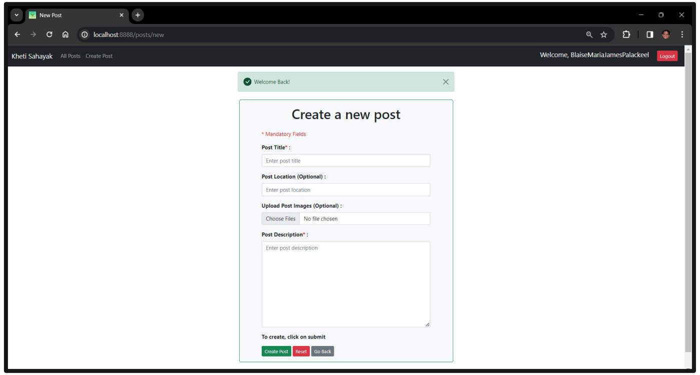


Login Signup Page

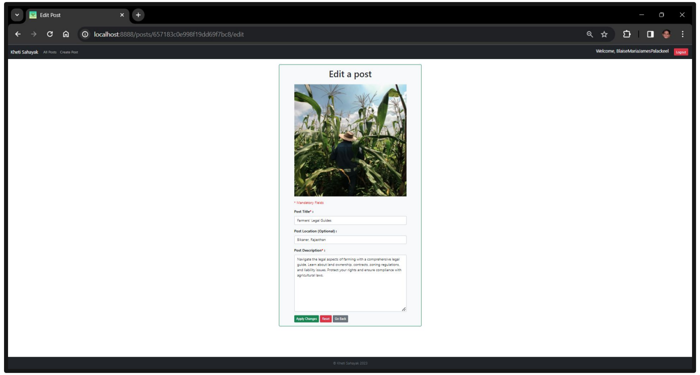


All Posts Page

Output of Module I Contd.

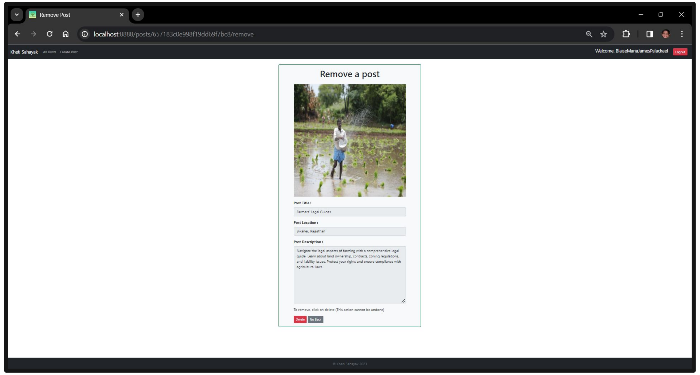


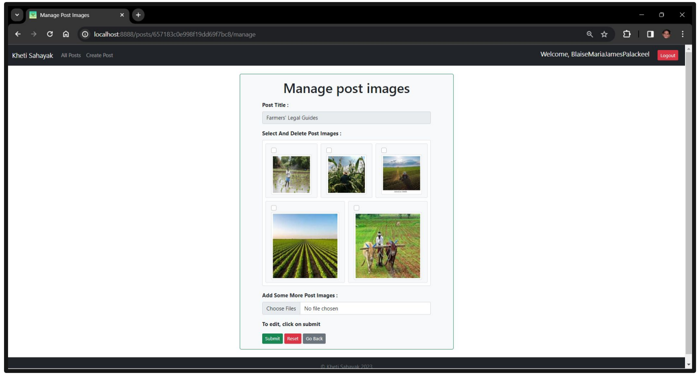
Create Post Page



Edit Post Page

Output of Module I Contd.





Remove Post Page

Manage Post Page

Stages of Module II & III Development

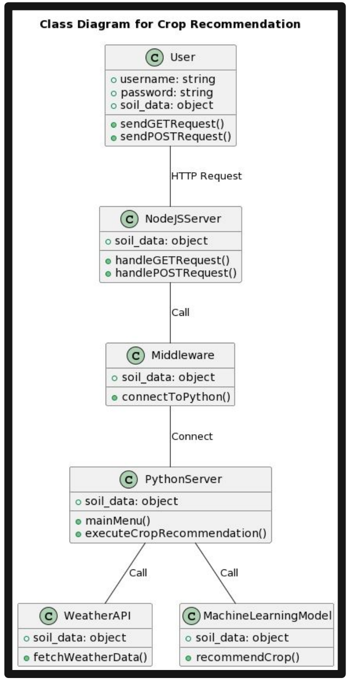
DESCRIPTION: BEST SUITABLE CROP & FERTILIZER RECOMMENDATION

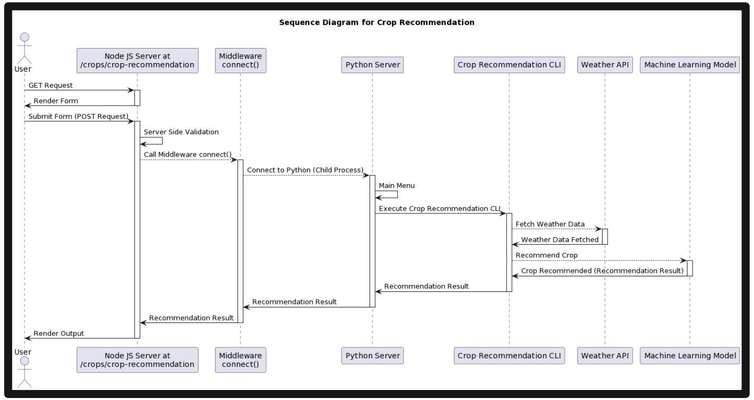
STAGE 1: LOADING, PRE-PROCESSING AND TRAINING DATASETS USING VARIOUS MACHINE LEARNING ALGORITHMS IN PYTHON

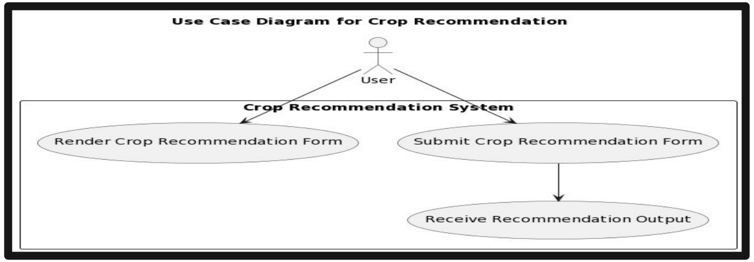
STAGE 2: CREATING A FORM TO ACCEPT DATA USING HTML, CSS AND JAVASCRIPT AND VALIDATE THE DATA USING NODE JS

STAGE 3: CONNECT NODE JS TO PYTHON TO RECOMMEND CROP AND FERTILIZER BASED ON THE VALIDATED DATA AND DISPLAY THE RESULTS THROUGH THE SAME FORM

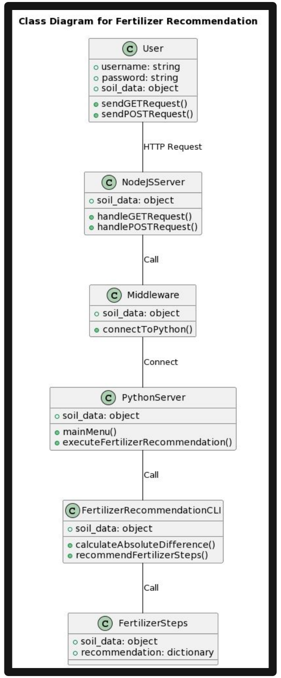
UML Diagrams of Module II

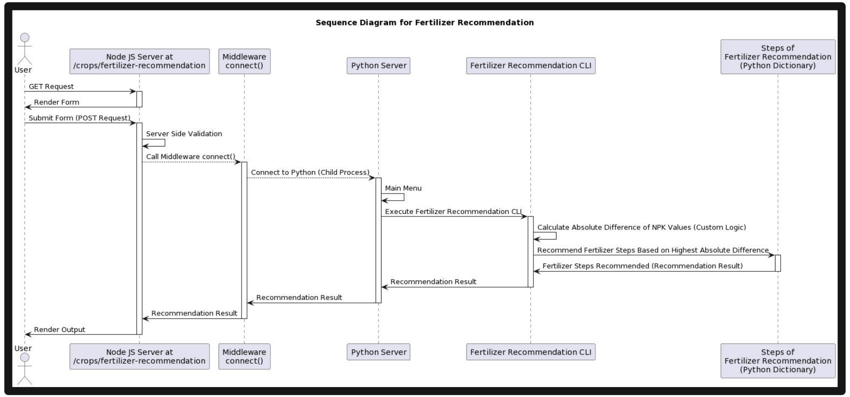


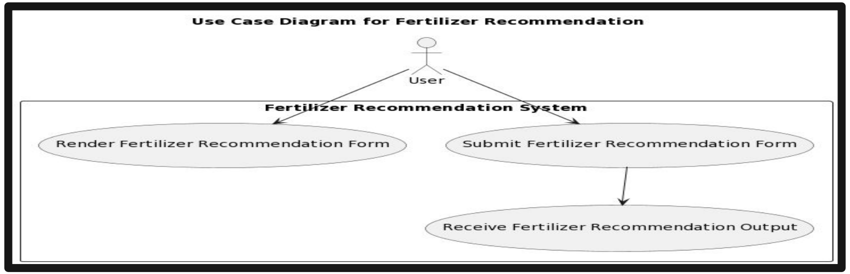




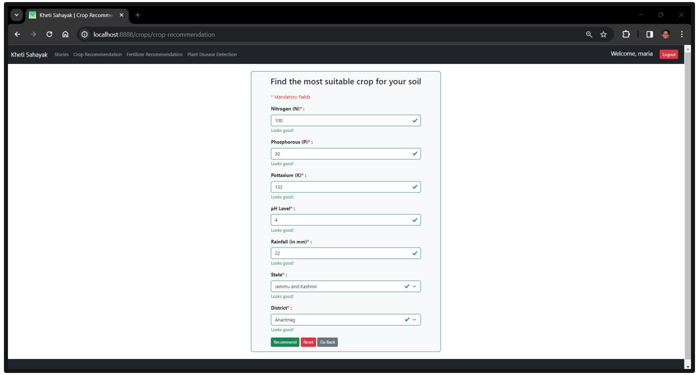
UML Diagrams of Module III



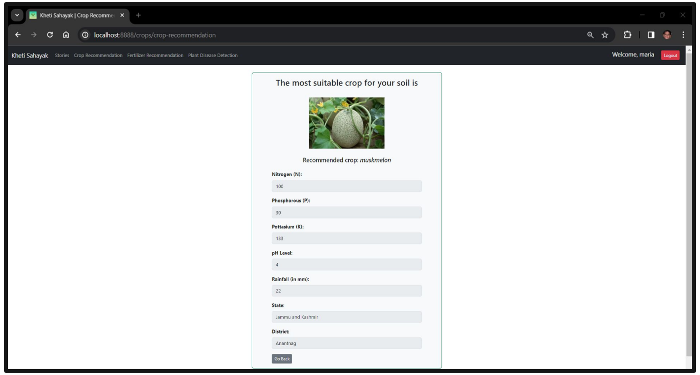




Output of Module II

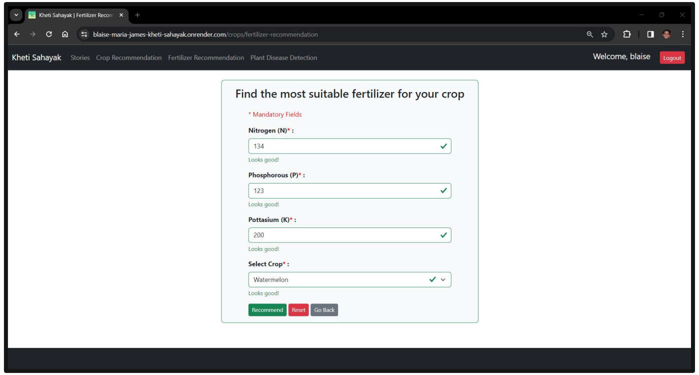


Crop Recommendation
Form

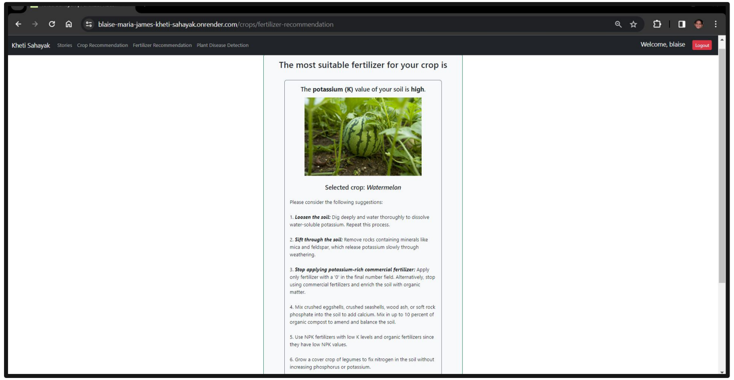


Crop Recommendation
Results

Output of Module III



Fertilizer Recommendation Form



Fertilizer Recommendation Results

Stages of Module IV Development

DESCRIPTION: PLANT DISEASE DETECTION

STAGE 1: COLLECTING AND PRE-PROCESSING DATASETS OF PLANT IMAGES CONTAINING HEALTHY AND DISEASED PLANTS

STAGE 2: DESIGNING AND DEVELOPING A USER INTERFACE TO UPLOAD PLANT IMAGES USING HTML, CSS, AND JAVASCRIPT

STAGE 3: IMPLEMENTING A BACKEND SYSTEM IN NODE JS TO RECEIVE THE UPLOADED IMAGES, CONNECT TO A PRE-TRAINED MACHINE LEARNING MODEL IN PYTHON, AND PROCESS THE IMAGES FOR DISEASE DETECTION

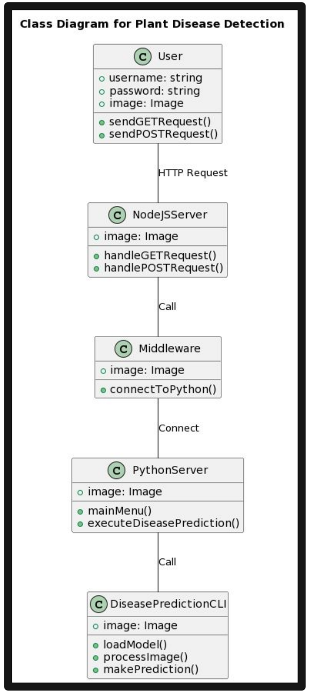
Stages of Module IV Development Contd.

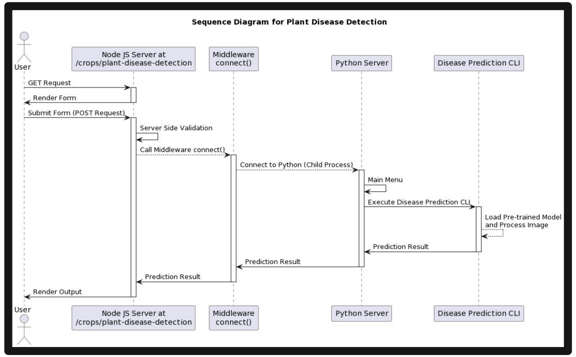
DESCRIPTION: PLANT DISEASE DETECTION

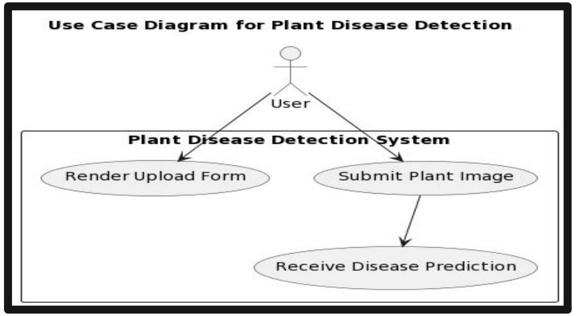
STAGE 4: INTEGRATING THE NODE JS BACKEND WITH THE PYTHON ENVIRONMENT TO PASS THE UPLOADED IMAGES TO THE MACHINE LEARNING MODEL FOR DISEASE DETECTION

STAGE 5: DISPLAYING THE DETECTION RESULTS BACK TO THE USER THROUGH THE WEB INTERFACE, INDICATING WHETHER THE PLANT IS HEALTHY OR INFECTED AND IDENTIFYING THE SPECIFIC DISEASE IF DETECTED

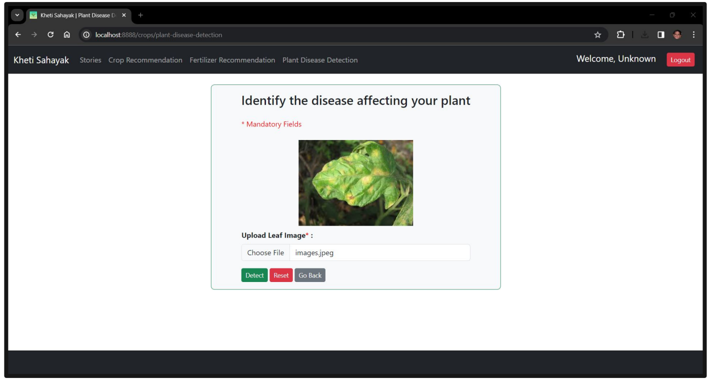
UML Diagrams of Module IV

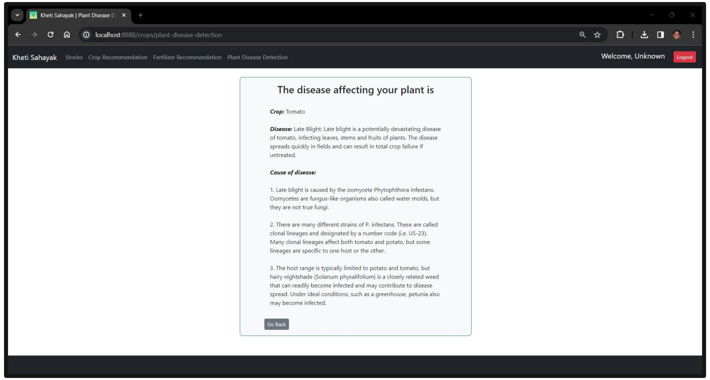






Output of Module IV





Plant Disease Detection Form

Plant Disease Detection Results

Conclusion

Kheti Sahayak is an exciting example of how Python, Machine Learning, and Web Development can come together to create a practical solution for agricultural challenges. By using modern technologies, the project demonstrates how farmers can leverage data-driven recommendations to improve their crop, soil management practices, increase yields and detect plant diseases.

ANY QUERIES?

THANK YOU