PESTICIDE POISONING DIAGNOSIS SYSTEM

Major project submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Under the esteemed guidance of

Bh Bhujanga Reddy

Assistant Professor

By

S. Prakash Reddy (21R11A05Q8)

G. Karthik (21R11A05M9)

M. Praveen (21R11A05P1)

Department of Computer Science and Engineering

Geethanjali College of Engineering and Technology (Autonomous)

Accredited by NAAC with A+ Grade: B.Tech. CSE, EEE, ECE accredited by NBA Sy. No: 33 & 34, Cheeryal (V), Keesara (M), Medchal District, Telangana – 501301

MAY – 2025

Geethanjali College of Engineering and Technology (Autonomous)

Accredited by NAAC with A+ Grade : B.Tech. CSE, EEE, ECE accredited by NBA Sy. No: 33 & 34, Cheeryal (V), Keesara (M), Medchal District, Telangana – 501301

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the B.Tech Major Project report entitled **"PESTICIDE POISONING DIAGNOSIS SYSTEM"** is a bonafide work done by **S. Prakash Reddy (21R11A05Q8), G. Karthik (21R11A05M9), M. Praveen (21R11A05P1)** in partial fulfillment of the requirement of the award for the degree of Bachelor of Technology in "**Computer Science and Engineering**" from Jawaharlal Nehru Technological University, Hyderabad during the year 2024-2025.

Internal Guide	**HoD – CSE**
Bh Bhujanga Reddy	**Dr.E. Ravindra**
Assistant Professor	**Professor**

External Examiner

Geethanjali College of Engineering and Technology (Autonomous)

Accredited by NAAC with A+ Grade : B.Tech. CSE, EEE, ECE accredited by NBA Sy. No: 33 & 34, Cheeryal (V), Keesara (M), Medchal District, Telangana – 501301

Department of Computer Science and Engineering

DECLARATION BY THE CANDIDATE

We, **S Prakash Reddy, G Karthik, M Praveen**, bearing Roll Nos. **21R11A05Q8, 21R11A05M9, 21R11A05P1**, hereby declare that the project report entitled **"PESTICIDE POISONING DIAGNOSIS SYSTEM"** is done under the guidance of **Mr. Bh Bhujanga Reddy**, **Assistant Professor**, Department of Computer Science and Engineering, Geethanjali College of Engineering and Technology, is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering**.

This is a record of bonafide work carried out by me/us and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

S Prakash Reddy, 21R11A05Q8

G Karthik, 21R11A05M9

M Praveen, 21R11A05P1

Department of CSE, Geethanjali College of Engineering and Technology,

Cheeryal

ACKNOWLEDGEMENT

It is with profound gratitude and sincere appreciation that we extend our heartfelt thanks to all those who have played a significant role in the successful completion of this undergraduate project.

First and foremost, we express our deep sense of respect and gratitude to our **Honourable Chairman, Mr. G. R. Ravinder Reddy**, for his constant encouragement and for nurturing a culture of academic excellence and innovation within the institution.

We would also like to express our sincere thanks to **Dr. S. Udaya Kumar**, **Director,** for his visionary leadership and continued support, which have provided the ideal environment and motivation for academic pursuits such as this project.

Our heartfelt appreciation goes to **Dr. K. Sagar**, **Principal,** for his steadfast guidance, infrastructural support, and encouragement that have helped bring this project to successful fruition.

We are deeply grateful to **Dr. E. Ravindra, Professor**, **Head of the Department- CSE**, for his academic leadership, valuable feedback, and continuous support throughout the duration of this project.

We extend our sincere thanks to our **Project Coordinators, Dr K Srinivasa Reddy, D Swaroopa**, for their meticulous planning, timely evaluations, and constant coordination that ensured a smooth and structured project development process.

A special note of gratitude is reserved for our project guide, **Bh Bhujanga Reddy, Assistant Professor**, whose expert supervision, insightful suggestions, and dedicated mentorship have been instrumental in shaping the direction and outcome of this work.

Lastly, we are ever grateful to our **parents and family** for their unconditional love, encouragement, and moral support. Their faith in us has been our greatest strength throughout this journey.

With genuine appreciation, we acknowledge every individual who has, in one way or another, contributed to the successful completion of this project.

With warm regards,
S Prakash Reddy (21R11A05Q8),
G Karthik (21R11A05M9),
M Praveen (21R11A05P1),
Department of Computer Science and Engineering,
Geethanjali College of Engineering and Technology,
Cheeryal

ABSTRACT

Pesticide poisoning is a significant health hazard, particularly for rural workers engaged in agriculture, due to the extensive use of chemical pesticides. Early diagnosis is critical for effective treatment and preventing long-term health complications. This project presents a comprehensive diagnostic system for pesticide poisoning, leveraging ensemble machine learning models for accurate and efficient prediction. The system integrates algorithms such as K-Nearest Neighbors (KNN), Logistic Regression, Gradient Boosting, and Random Forest Classifier to enhance prediction reliability.

A user-friendly web interface enables users to log in, input symptoms, view prediction results, and analyze model accuracy. Additionally, the system offers interactive visualizations, helping healthcare professionals interpret results and recognize key trends. This solution is designed to assist healthcare professionals and workers in rural areas by providing a precise, accessible tool for diagnosing pesticide poisoning. By combining advanced algorithms with an intuitive interface, the project aims to improve healthcare outcomes and reduce the burden of pesticide-related illnesses in underserved communities.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

S. No.	Abbreviation	Full Form
1	AI	Artificial Intelligence
2	NGOs	Non-Governmental Organizations
3	UML	Unified Modeling Language
4	API	Application Programming Interface
5	PDF	Portable Document Format
6	SDLC	Software Development Life Cycle
7	UI	User Interface
8	UAT	User Acceptance Testing
9	IEEE	Institute of Electrical and Electronics Engineers
10	SRS	Software Requirements Specification
11	ISO	International Organization for Standardization
12	IEC	International Electrotechnical Commission
13	Ml	Machine Learning
14	KNN	K-Nearest Neighbour
15	HTML	Hyper Text Markup Language
16	CSS	Cascading Style Sheets
17	CSV	Comma-Separated Values
18	ARIA	Accessible Rich Internet Applications

TABLE OF CONTENTS

1. INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The Pesticide Poisoning Diagnosis System is an intelligent, web-based platform designed to assist in the early identification and diagnosis of pesticide poisoning, especially for rural agricultural workers who are most at risk due to frequent exposure to toxic substances. The system leverages an ensemble machine learning model and integrates clinical and environmental datasets to deliver accurate, fast, and interpretable diagnostic outcomes.

This project aims to bridge the gap between rural healthcare accessibility and technological advancement by providing a user-friendly interface that allows individuals, healthcare workers, or service providers to input basic symptoms and exposure details. Based on the input, the system evaluates the likelihood of pesticide poisoning using K-Nearest Neighbors (KNN), Logistic Regression, Gradient Boosting, and Random Forest Classifier algorithms. The ensemble model ensures robustness and higher accuracy by combining the strengths of each individual classifier.

The platform is designed to handle the collection, processing, prediction, and visualization of poisoning-related data. A built-in login system ensures that data is securely handled, and results can be tracked by authenticated users. The inclusion of interactive graphs and prediction outputs helps users easily interpret the results, even with minimal technical knowledge.

This diagnostic system plays a vital role in reducing response times, improving decision-making for treatment, and potentially saving lives by guiding early medical intervention. The long-term goal is to integrate this system with local health networks and NGOs working in the field of rural health and agriculture safety.

1.2 PROBLEM STATEMENT

Pesticide poisoning remains a significant public health issue, particularly in agricultural regions where rural workers are routinely exposed to toxic chemicals during farming activities. In many cases, early symptoms of pesticide poisoning are either overlooked or misinterpreted, leading to delayed diagnosis and treatment. This delay can cause severe health complications or even death, especially when medical facilities are distant or inaccessible.

Traditional diagnosis in rural healthcare settings is often manual, relies heavily on expert availability, and lacks standardized tools for quick evaluation. Most rural clinics do not possess the technical capacity or trained personnel to accurately assess poisoning risks based on symptom patterns and environmental exposure history. Furthermore, the absence of a centralized system to gather, process, and analyze such cases hinders early detection and proactive response.

There is a growing need for an automated, accessible, and accurate diagnostic system that can assist healthcare providers, outreach workers, or even semi-literate users in identifying pesticide poisoning cases based on observable symptoms. Such a system should support informed decision-making, provide immediate feedback, and be simple enough to operate in low-resource environments.

The challenge lies in designing a technology-driven solution that combines data science and usability to reduce diagnosis time, improve accuracy, and support health workers in managing pesticide-related emergencies more effectively.

1.3 OBJECTIVES OF THE PROJECT

- To develop a machine learning-based diagnostic system capable of identifying pesticide poisoning in rural individuals based on user inputs.
- To design an ensemble model that integrates K-Nearest Neighbors, Logistic Regression, Gradient Boosting, and Random Forest Classifier for improved prediction accuracy.
- To provide a web-based interface that allows users to input symptom-related data and receive instant diagnostic feedback.
- To ensure ease of use for rural health workers, outreach personnel, or patients themselves through a clean and intuitive user interface.
- To enable secure user authentication with login and registration modules for service providers and healthcare personnel.
- To incorporate visualizations for better interpretation of results using graphs, prediction charts, and performance metrics for service provider to analyse patterns.
- To incorporate visualizations for better interpretation of results using graphs, prediction charts, and performance metrics, enabling service providers to analyze patterns effectively.
- To support scalability and integration with real-world health infrastructure for future expansion.
- To optimize the system for high model accuracy and computational efficiency, ensuring reliable predictions in low-resource environments.
- To develop a diagnostic platform that facilitates real-time interaction, data handling, and effective communication between end users and the system.

1.4 SCOPE OF THE PROJECT

The Pesticide Poisoning Diagnosis System is designed to provide a technology-driven solution for the early identification of pesticide poisoning cases, particularly in rural and agricultural settings. The system targets users such as healthcare workers, service providers, and individuals in affected regions, offering them a reliable tool for assessing symptoms and receiving diagnostic predictions instantly.

The project encompasses the development of a web-based interface that supports user registration, secure login, and access to various system functionalities. It allows service providers to input clinical and environmental details, trigger predictions using a trained ensemble model, and view the results in an interpretable format. The system is built to operate efficiently even on low-resource devices, making it suitable for rural healthcare environments.

The scope also includes integrating machine learning models that combine multiple classification algorithms to enhance the accuracy and reliability of predictions. These models are trained on a diverse dataset to ensure robustness across varying symptom patterns and exposure scenarios.

In addition, the system provides visual outputs in the form of graphs and performance charts to help service providers analyze diagnostic trends and patterns. This enables better decision-making and facilitates ongoing monitoring of affected individuals.

The project is scalable and designed to support future enhancements such as multilingual support, mobile accessibility, integration with rural health networks, and linkage to emergency response systems.

1.5 METHODOLOGY

The development of the Pesticide Poisoning Diagnosis System followed the **Incremental Software Development Life Cycle (SDLC) Model**, which was selected for its suitability in handling academic projects with multiple modules and evolving requirements. This model enabled the team to develop, test, and refine individual components of the system in iterations while maintaining continuity across the entire project.

In the Incremental model, each functionality is developed as a separate unit or "increment" that goes through the full cycle of requirement analysis, design, coding, and testing. After verification, each increment is integrated into the main system. This approach ensured that core features such as data input, model prediction, and user authentication were available early in the development phase, allowing continuous feedback and improvement.

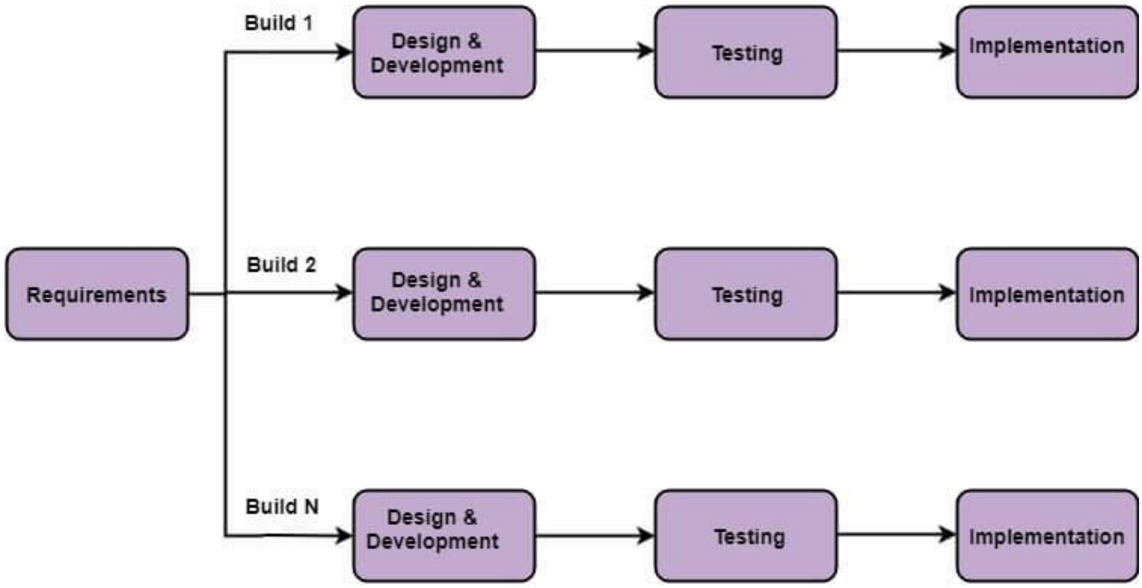

Fig 1.5.1 Incremental Model

Key Reasons for Choosing the Incremental Model:

- Allows early delivery of partial but functional system features
- Facilitates parallel development and integration of different modules
- Supports flexible refinement of components based on user and faculty input
- Enables targeted testing, easier debugging, and structured deployment

Development Phases Followed:

1. **Requirement Analysis** – Identified system functionalities including symptom input, user login, ensemble model integration, prediction display, and result visualization.
2. **Design** – Structured the user flow, backend logic, and database schema to support multiple user roles and modular machine learning integration.
3. **Implementation** – Developed the project in modules such as user registration, dataset training, prediction engine, and visualization dashboard.
4. **Testing** – Carried out unit and integration testing on each module to ensure accuracy, system stability, and expected user interaction.
5. **Deployment** – Hosted the application locally and demonstrated its working with dummy and real-time data inputs.
6. **Maintenance and Improvement** – Prepared the platform for future enhancements like mobile support, multilingual interface, and dataset expansion.

By adopting the Incremental SDLC model, the project achieved better modularization, reduced complexity, and ensured steady progress throughout the development cycle.

1.6 ORGANIZATION OF THE REPORT

This report is organized into ten comprehensive chapters, each detailing a specific aspect of the development process, analysis, and implementation of the Pesticide Poisoning Diagnosis System.

- **Chapter 1** introduces the project with an overview, problem statement, objectives, scope, the adopted SDLC model, and the structure of the report.
- **Chapter 2** presents a literature survey highlighting existing systems, their limitations, and the necessity for a machine learning-based diagnostic solution.
- **Chapter 3** focuses on system analysis, including the feasibility study, software requirements, and both functional and non-functional specifications.
- **Chapter 4** explains the system design with architecture diagrams, database schema, UML representations, and user interface design standards.
- **Chapter 5** discusses implementation details, covering the technology stack, module-wise development, code structure, and integration strategies.
- **Chapter 6** outlines the testing phase, describing the testing strategy, test cases, results, and quality assurance practices.
- **Chapter 7** presents the results with output screenshots, interpretation of predictions, and performance evaluations of the machine learning models.
- **Chapter 8** concludes the work, summarizing key achievements, challenges encountered, and potential future enhancements.
- **Chapter 9** lists all references, including technical papers, websites, and other resources used during the project.
- **Chapter 10** contains appendices such as SDLC forms, Gantt charts, source code links, and additional supporting documentation relevant to the project.

2. LITERATURE SURVEY

2.1 REVIEW OF EXISTING SYSTEM

Diagnosis of pesticide poisoning in rural areas is primarily manual, relying on the experience of healthcare workers with limited tools. Patients often delay reporting symptoms, leading to late or inaccurate assessments. Most rural clinics lack systems that correlate clinical symptoms with environmental exposure. Existing electronic health records are urban-focused and not suited for low-resource rural settings. Mobile health apps are general-purpose and do not address toxic exposure. The absence of automation and standardization makes diagnosis slow and inconsistent. These limitations emphasize the need for a scalable, data-driven diagnostic system tailored to rural healthcare needs.

2.2 LIMITATIONS OF EXISTING APPROACHES

1. **Manual Diagnosis**: Current methods rely heavily on human judgment, increasing the chances of error, especially when symptoms are nonspecific or overlapping with other illnesses.
2. **Delayed Response**: There is often a significant time gap between symptom onset and diagnosis due to lack of awareness, infrastructure, or access to immediate care.
3. **Lack of Standardization**: No unified framework exists for evaluating symptoms and exposure data, resulting in inconsistent diagnostic outcomes.
4. **Limited Technological Support**: Most rural clinics lack access to digital tools or AI-based systems that can assist in identifying pesticide-related health issues.
5. **Inadequate Training**: Health workers in remote areas may not be trained specifically in toxicology or pesticide exposure management, affecting the accuracy of diagnosis.
6. **Poor Data Integration**: Existing systems do not combine clinical symptoms with environmental data, limiting the reliability and contextual relevance of the diagnosis.
7. **Non-specific Health Apps**: Available mobile and digital health platforms are not designed to detect or manage pesticide poisoning, making them ineffective in such scenarios.

2.3 NEED FOR PROPOSED SYSTEM

There is a growing need for an automated system that can facilitate early diagnosis of pesticide poisoning, particularly among rural agricultural workers who are regularly exposed to hazardous chemicals. Existing diagnostic methods are often manual, time-consuming, and reliant on the availability and judgment of local healthcare personnel, which can lead to delays and inaccuracies. A machine learning-based diagnostic platform can overcome these limitations by analyzing symptom data and exposure history to generate real-time, reliable predictions.

The proposed system acts as a centralized platform where users can input patient data, receive instant diagnostic feedback, and view results through visual performance indicators. By leveraging an ensemble of machine learning models, it enhances diagnostic accuracy while minimizing errors. It also helps standardize the evaluation process, supports rapid decision-making in low-resource settings, and can scale to accommodate broader rural health networks and emergency response frameworks.

Benefits of the Proposed System:

- Enables early detection of pesticide poisoning using machine learning-based predictions.
- Reduces diagnostic time and supports faster medical intervention.
- Minimizes dependency on specialist healthcare professionals in rural areas.
- Offers a consistent, data-driven approach for evaluating symptoms and exposures.
- Improves accessibility through a user-friendly web interface.
- Supports visualization of predictions and diagnostic trends for better decision-making.
- Scalable for use across multiple regions and adaptable to future enhancements.

2.4 COMPARATIVE STUDY

S. No	Author(s)	Title of Paper/Study	Year	Key Takeaways
1	Jaqueline C. S. Carvalho et al. (IEEE)	Supervised Learning for Pesticide Diagnosis	2024	Proposes a structured data science approach for diagnosing pesticide poisoning in rural workers.
2	Ganesan Anandhi, M. Iyapparaja (Elsevier)	ML Models for Pesticide Toxicity	2024	Emphasizes the need for accurate toxicity prediction using comprehensive ML models.
3	Ibomoiye Domor Mienye, Yanxia Sun (IEEE)	Survey on Ensemble Learning Techniques	2022	Demonstrates how ensemble models can overcome limitations of individual classifiers.
4	Veneta Tabakova-Komsalova et al. (IEEE)	An Expert System for the Diagnosis of Livestock Poisoning	2023	Highlights the importance of expert systems in remote poisoning diagnosis.
5	Riad Taha Al-Kasasbeh et al. (IEEE)	Fuzzy Models for Occupational Diseases	2022	Uses fuzzy logic to address overlapping conditions in pesticide-exposed workers.

S. No	Author(s)	Title of Paper/Study	Year	Key Takeaways
6	Gianluca Manduca et al. (IEEE)	Image Analysis of Pesticide Effects	2023	Explores image-based methods for detecting subtle health effects of pesticide exposure.
7	S. Prakash, R. Devi	Predictive Modeling for Environmental Health Risk Assessment	2021	Integrates exposure data with ML predictions for better rural health risk management.
8	T. Krishnan, P. Mehra	AI-Driven Decision Support Systems in Rural Health	2020	Describes the benefits of AI tools in assisting rural medical decision-making.
9	L. Anand, A. Jaiswal	Web-Based Medical Diagnostic Tools Using Machine Learning	2022	Discusses challenges and potentials of web-based ML tools in healthcare diagnostics.
10	D. Thomas, M. Bhargav	Machine Learning for Toxicology and Public Health: A Practical Approach	2023	Focuses on building practical ML tools for diagnosing toxic exposure-related health conditions.

Table 2.4.1 Comparative Study

3. SYSTEM ANALYSIS

3.1 FEASIBILITY STUDY

This section analyzes the feasibility of developing and deploying the Pesticide Poisoning Diagnosis System. It assesses the project's viability in terms of technical implementation, cost-effectiveness, usability in rural settings, and timeline constraints within an academic framework.

3.1.1 TECHNICAL FEASIBILITY

The technical feasibility assesses whether the existing technical resources are sufficient to build and run the system.

- The system is built using Python, Django, Scikit-learn, and basic web technologies, all of which are open-source and suited for academic development.
- It integrates ML algorithms like KNN, Logistic Regression, Gradient Boosting, and Random Forest for accurate pesticide poisoning prediction.
- The application runs smoothly on personal laptops and standard web browsers, requiring no specialized hardware.

Feasible: The required tools and libraries were readily available, and the development team was well-versed in the selected technology stack.

3.1.2 ECONOMIC FEASIBILITY

This evaluates the cost-effectiveness of the project.

- The Pesticide Poisoning Diagnosis System was developed using free and open-source tools such as Python, Django, Scikit-learn, and standard web technologies.
- No licensed software or paid infrastructure was required throughout development.
- The system was built and tested on existing personal laptops, eliminating the need for any additional hardware or hosting expenses.
- The project was carried out as part of an academic curriculum, so there were no labor or consultancy costs involved.

Feasible: The project incurred negligible expenses and was completed within the academic budget framework.

3.1.3 OPERATIONAL FEASIBILITY

This examines whether the proposed system will work in the intended operational environment.

- The system is designed for use by rural healthcare workers, service providers, or NGOs dealing with pesticide-related health cases.
- The user interface is clean, responsive, and intuitive, requiring minimal technical knowledge to operate.
- Features like login, symptom input, and result interpretation are streamlined for ease of access and use.
- The platform is web-based and runs on standard devices with internet access, making it accessible even in low-resource settings.

Feasible: Target users can operate the system effectively with minimal training or technical support.

3.1.4 TIME AND COST ESTIMATION

Estimates were made for development and testing phases to ensure timely completion.

- **Timeframe**: The project was structured to be completed within 8–10 weeks, covering stages such as requirement analysis, design, module-wise implementation, testing, and final documentation.
- **Cost**: As an academic project, no financial investment was required. The only resources used were time, personal devices, and freely available tools.

Feasible: The project was completed on schedule with efficient use of time and available resources.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

This section outlines the essential software requirements for the successful design, development, and deployment of the Pesticide Poisoning Diagnosis System. It highlights the key technologies used, their specific purposes, and the modules where each was applied to enable accurate prediction, user interaction, and data visualization through a web interface.

Software Requirements Used in the Project

The following tools and technologies were utilized in the development of the system:

- Python 3.x
- Django
- Scikit-learn
- HTML / CSS / JavaScript
- MySQL
- Matplotlib / Seaborn (for visualization)

These tools together enabled machine learning model development, server-side logic, data processing, database handling, and responsive web interface creation.

Purpose of the Software Requirements

Each tool was selected based on its capability to support the system's goals of automation, usability, and accurate diagnosis:

- Python 3.x served as the primary development language for implementing ML algorithms, backend logic, and module integration.
- Django was used as the web framework for building the application's server-side architecture, URL routing, and handling user sessions.
- Scikit-learn provided a robust environment for developing and training ensemble machine learning models.
- HTML / CSS / JavaScript were used to build an interactive and responsive user interface for both healthcare workers and service providers.
- MySQL acted as the backend database to manage user data and session information during development.
- Matplotlib and Seaborn were used to visualize prediction results and model accuracy in a user-friendly manner.

Application of Software to Project Modules

Technology	Modules Involved
Python 3.x	Core logic for model development, symptom analysis, and prediction output
Django	Backend framework for user authentication, form handling, and data flow management
Scikit-learn	Implementation of KNN, Logistic Regression, Gradient Boosting, and Random Forest
HTML / CSS / JS	Frontend for login, registration, prediction dashboard, and visual displays
MySQL	User data management and service provider login credentials
Matplotlib/Seaborn	Display of model accuracy, prediction performance, and data visualizations

Table 3.2.1 Modules Involved

3.3 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

To ensure the successful development and deployment of the Pesticide Poisoning Diagnosis System, it is essential to define both functional and non-functional requirements. These requirements guide the implementation of features related to prediction, data handling, visualization, and user management, while also ensuring system reliability, usability, and performance in real-world conditions.

3.3.1 FUNCTIONAL REQUIREMENTS

1. **User Registration and Authentication**
    - The system must allow users (healthcare workers or service providers) to register and log in securely.
    - Authentication should protect access to patient records and prediction modules.

2. **Symptom Data Input**
    - The interface should allow users to input clinical symptoms and exposure-related data.
    - Data should be validated before processing to ensure accuracy.
3. **Machine Learning-Based Prediction**
    - The system must process input through an ensemble model (KNN, Logistic Regression, Gradient Boosting, Random Forest) to predict pesticide poisoning.
    - The predicted output must be displayed clearly and in real time.
4. **Result Visualization**
    - The system must provide visualizations such as graphs and charts exclusively for service providers to help analyze overall trends and patterns in prediction outcomes.
    - These visual tools are intended for backend analysis and are not shown to end users receiving individual prediction results.
5. **Prediction History and Reporting**
    - Users should be able to access a history of predictions with corresponding input details.
    - A report generation feature must allow exporting prediction summaries in a readable format (e.g., PDF or CSV).
6. **Role-Based Access**
    - Admin/service provider roles should have access to dataset upload and system configuration features.
    - Basic users should have access only to symptom input and prediction features.

3.3.2 NON-FUNCTIONAL REQUIREMENTS

1. **Usability**
    - The system must have a clean and intuitive interface suitable for rural healthcare workers with minimal technical background.
    - Form fields and output screens must be clearly labeled and easy to understand.

2. **Performance**
    - The prediction results must be generated within a few seconds to support real-time decision-making.
    - Model loading, training, and testing should be optimized for low-resource environments.
3. **Security**
    - All user credentials and sensitive data (inputs, predictions) must be securely stored in the MySQL database.
    - User sessions should be protected, and access to backend functionalities should be restricted to authenticated users.
4. **Scalability**
    - The system should be designed to accommodate future integration with health databases, mobile interfaces, or multilingual support.
    - It must be capable of handling increasing amounts of input data without performance degradation.
5. **Reliability**
    - The system should provide consistent prediction results when given the same input.
    - It should gracefully handle errors like incomplete input or database connection failures.

4. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

The system architecture of the Pesticide Poisoning Diagnosis System provides a structured overview of how various components interact to facilitate the diagnosis of pesticide poisoning in rural workers. The system is primarily designed for healthcare professionals and authorized users, focusing on accurate predictions, data management, and ease of use.

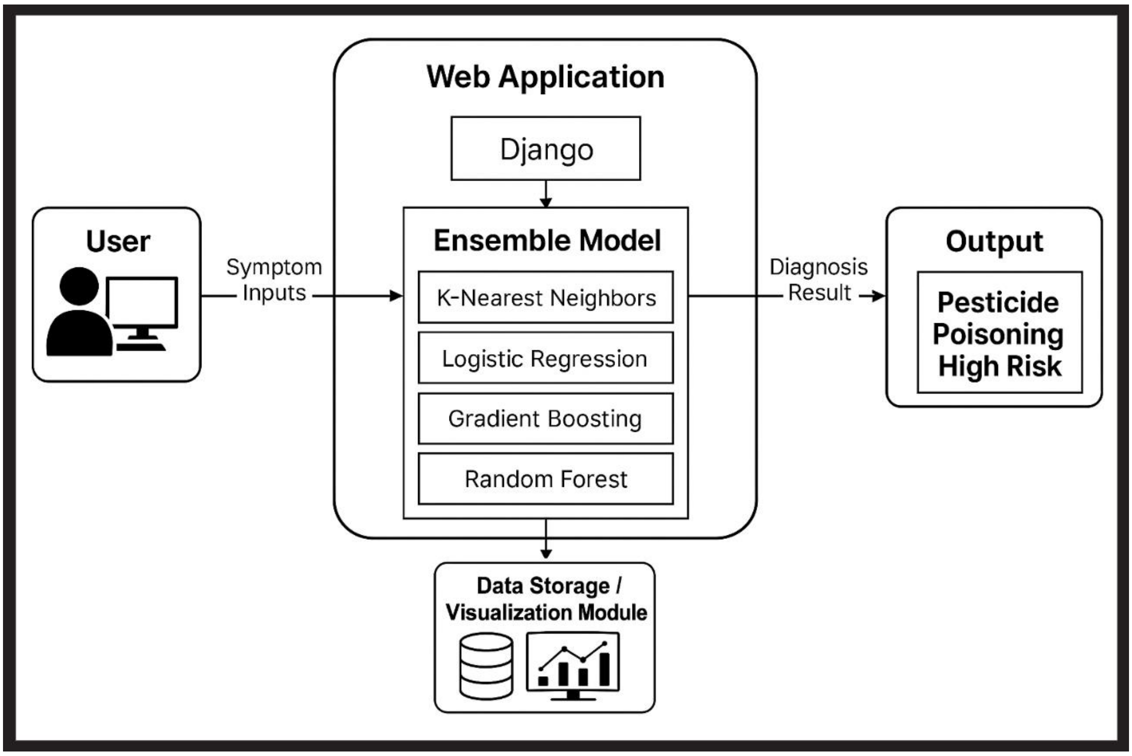

Fig 4.1.1 System Architecture

Key Users:

- **Healthcare Professionals:** They manage patient inputs, review diagnosis results, and make decisions based on the model’s predictions. No login/authentication is required for simple access.
- **Authorized Users (e.g., System Administrators):** They manage data and oversee the proper functioning of the system.
- **System:** Handles the backend machine learning algorithms for prediction, data storage, and visualization.

1. User Input Interface

- The user interacts with the system via a web interface, built using the Django framework.
- **Input:** The user provides symptoms (e.g., clinical signs, exposure details) that are indicative of pesticide poisoning.

2. Web Application (Django Framework)

- The user input is sent to the web application, built using Django, which serves as the front-end interface.
- **Role:** Django acts as a communication bridge, sending the input data from the user interface to the backend machine learning models.

3. Ensemble Model (Core AI Prediction Engine)

- The system forwards the user's input to an **Ensemble Model**, which integrates multiple machine learning algorithms:
  - K-Nearest Neighbors (KNN)
  - Logistic Regression
  - Gradient Boosting
  - Random Forest
- These models work in unison to predict the likelihood and severity of pesticide poisoning based on the symptoms provided.

4. Output

- After processing the input, the ensemble model returns a diagnosis result, including:
  - Pesticide Poisoning Detected
  - No Pesticide Poisoning

5. Data Storage and Visualization Module

- **Data Storage:** Stores inputs, diagnosis results, and user data.
- **Data Visualization:** Provides interactive charts and graphs that help Service Provider interpret trends, model accuracy, and other related data.

4.2 DATABASE DESIGN

The database design of the **Pesticide Poisoning Diagnosis System** plays a crucial role in efficiently managing patient data, clinical symptom tracking, model predictions, accuracy metrics, and service provider interactions. The system utilizes a **MySQL** relational database named pesticide_diagnosis, optimized to support real-time diagnostic processing, ensemble model evaluation, and user activity logging through the web interface.

Database Name

- **Database:** pesticide_diagnosis
- **Type:** Relational Database (MySQL)

The database supports essential operations such as:

- Storage and retrieval of patient symptom inputs.
- Logging of predictions from ensemble models (KNN, Logistic Regression, Gradient Boosting, and Random Forest).
- Management of user login credentials and roles (Admin, Service Provider, End User).
- Tracking of model accuracy metrics and visual feedback data.
- Data visualization backend integration for plotting accuracy, distribution, and usage statistics.

Tables and Schema Description

1. user_info

The user_info table stores registration and login credentials for all users, including patients, service providers, and administrators. Each user is assigned a role, and appropriate access is provided based on this role.

Column Name	Data Type	Constraints	Description
id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each user.
username	VARCHAR(100)	NOT NULL	Username selected by the user.
email	VARCHAR(255)	UNIQUE, NOT NULL	Registered email for login and notifications.
password_hash	VARCHAR(255)	NOT NULL	Hashed password for security.
role	ENUM('Admin', 'Provider', 'User')	DEFAULT 'User'	Defines user access level.

Table 4.2.1 user_info Table

2. symptom_input

The symptom_input table stores the symptoms entered by users for diagnosis. It supports integration with the trained ensemble model to generate a diagnosis and store relevant metadata.

Column Name	Data Type	Constraints	Description
input_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each input session.
user_id	INT	FOREIGN KEY REFERENCES user_info(id)	References the user who entered symptoms.
symptoms_json	TEXT	NOT NULL	JSON representation of symptoms selected.
timestamp	DATETIME	DEFAULT CURRENT_TIMESTAMP	Time of symptom submission.

Table 4.2.2 symptom_input Table

3. prediction_results

The prediction_results table stores the model predictions corresponding to each user's input, including prediction confidence scores and the ensemble decision.

Column Name	Data Type	Constraints	Description
result_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique ID for each prediction result.
input_id	INT	FOREIGN KEY REFERENCES symptom_input(input_id)	Links to the input record.
predicted_class	VARCHAR (100)	NOT NULL	Final diagnosis output.
model_votes	TEXT	NOT NULL	JSON containing individual model predictions.
accuracy_score	FLOAT	NOT NULL	Ensemble model's accuracy on current input.
recommendation	TEXT	NULLABLE	System-generated health recommendation.

Table 4.2.3 prediction_results Table

4. model_accuracy_logs

This table stores accuracy statistics of the individual machine learning models. These logs are used to monitor and compare model performance over time.

Column Name	Data Type	Constraints	Description
log_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each log entry.
model_name	VARCHAR(100)	NOT NULL	Name of the model (e.g., KNN, Logistic).
accuracy	FLOAT	NOT NULL	Accuracy value for the model.
precision	FLOAT	NULLABLE	Precision metric.
recall	FLOAT	NULLABLE	Recall metric.
timestamp	DATETIME	DEFAULT CURRENT_TIMESTAMP	Entry timestamp.

Table 4.2.4 model_accuracy_logs Table

4.3 UML DIAGRAMS

USECASE DIAGRAM

The Use Case Diagram for the Pesticide Poisoning Diagnosis System highlights the interaction between two main actors—User and Admin. Users can log in or register, submit symptoms, and view prediction results, representing the diagnostic flow from their perspective. Admins handle system management tasks such as uploading datasets, training models, viewing accuracy, exporting results, and visualizing performance. This diagram clearly separates user and admin functionalities, offering a simplified view of the system's core operations and interactions.

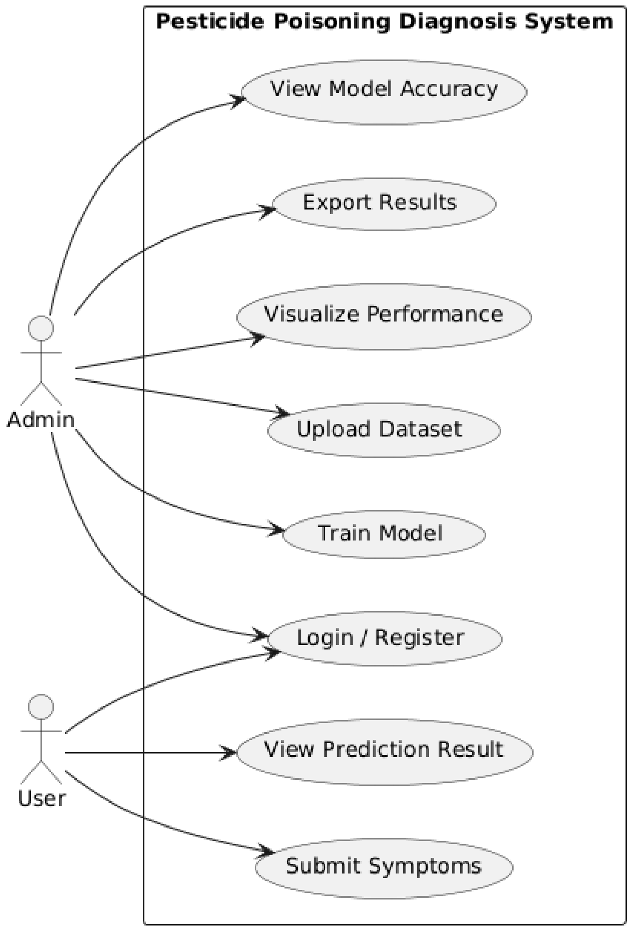

Fig 4.3.1 Usecase Diagram

CLASS DIAGRAM

The Class Diagram for the Pesticide Poisoning Diagnosis System highlights the main components and their relationships. Key classes include User, Admin, Dataset, Model, and Prediction. Users can register, log in, submit symptoms, and view results, while Admins manage datasets, train models, and view performance. The Model class handles prediction logic, trained using data from the Dataset class. The Prediction class stores results linked to each user. This diagram provides a simplified view of the system's structure and interactions.

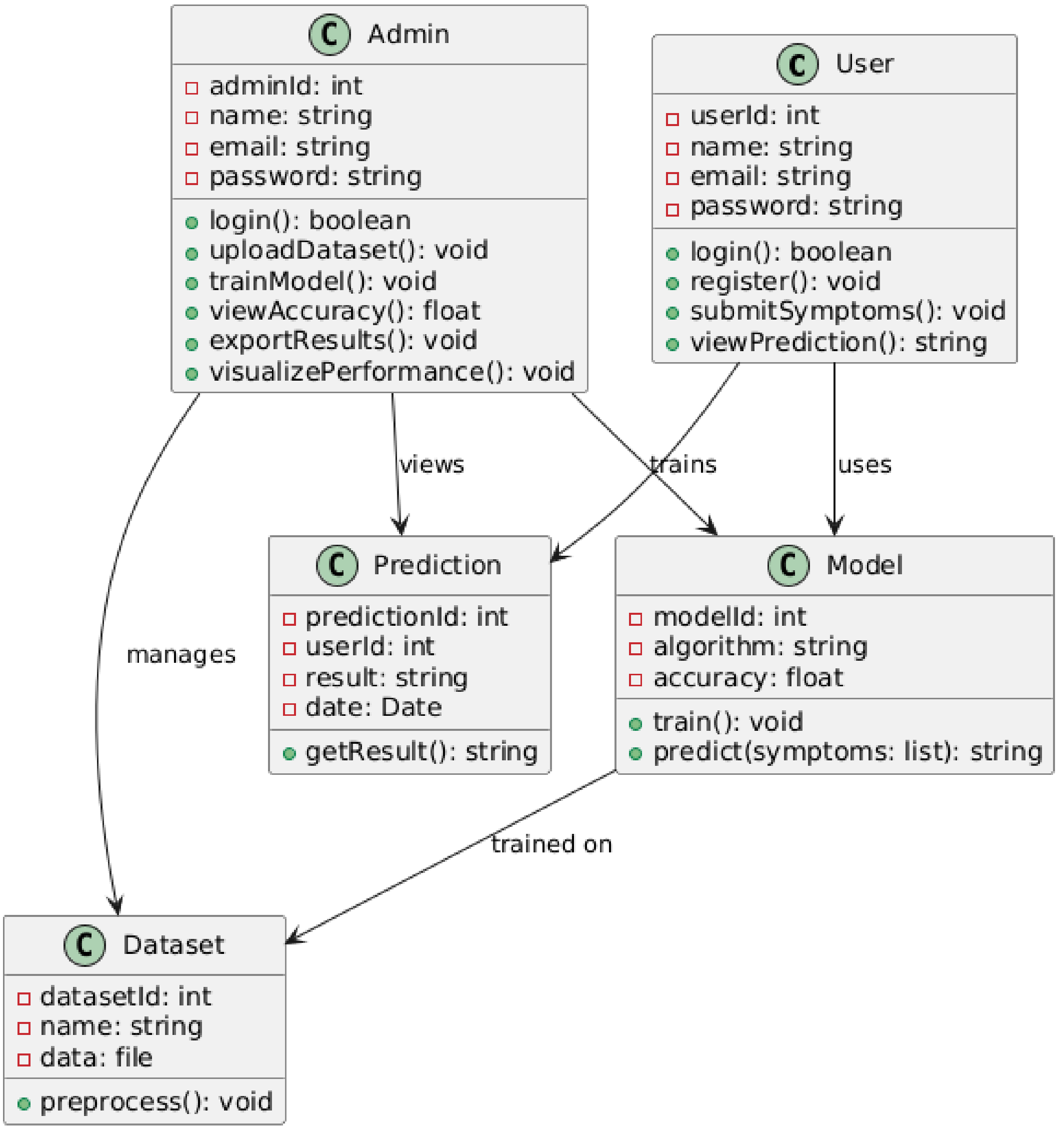

Fig 4.3.2 Class Diagram

ACTIVITY DIAGRAM

The Activity Diagram for the Pesticide Poisoning Diagnosis System illustrates the step-by-step flow from user login or registration to diagnosis output. After entering symptoms, the data is processed and passed to a machine learning model for prediction. Based on the result, the system displays a diagnosis or an error message. This diagram captures the key actions and logical flow of the system.

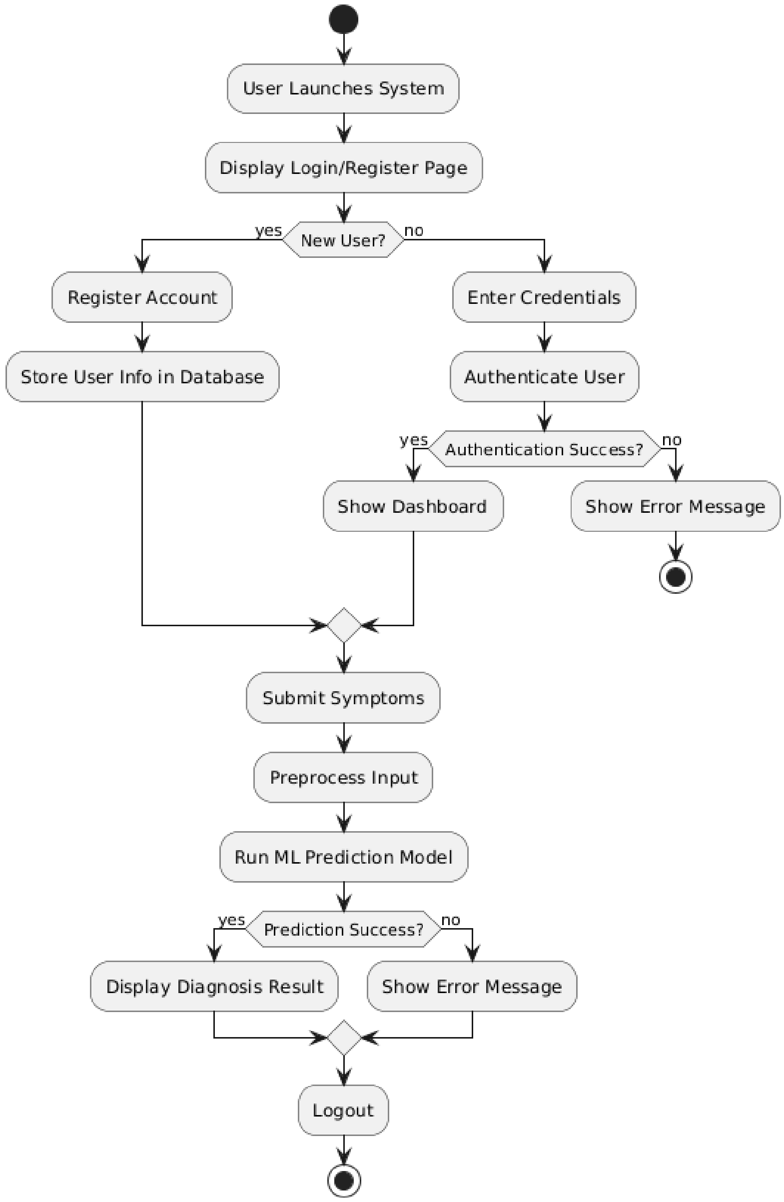

Fig 4.3.3 Activity Diagram

SEQUENCE DIAGRAM

The Sequence Diagram for the Pesticide Poisoning Diagnosis System illustrates the step-by-step interaction between the user and various system components during a diagnosis session. The process begins with the user logging in through the login interface, where the system verifies credentials against the database. Once authenticated, the user submits symptoms through the prediction module. This module interacts with the machine learning model, which fetches necessary parameters from the database and processes the input to generate a prediction. Finally, the diagnosis result is returned to the user, completing the interaction flow. This diagram helps visualize the logical order and communication flow between components involved in symptom submission and result generation.

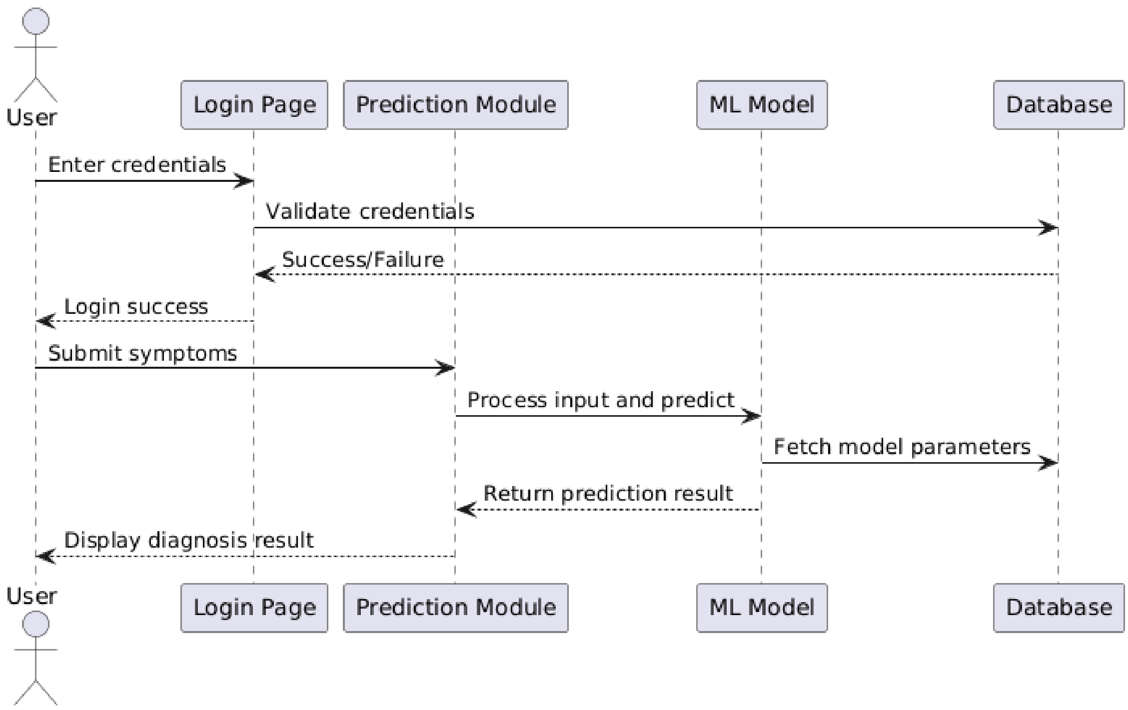

Fig 4.3.4 Sequence Diagram

4.4 USER INTERFACE DESIGN

The user interface of the Pesticide Poisoning Diagnosis System has been crafted to ensure a clean, user-friendly, and professional experience for all categories of users, including patients, doctors, and service providers. The design prioritizes intuitive navigation, clear layout, and minimal visual clutter, enabling smooth interaction even for users unfamiliar with technical systems.

The frontend is developed using HTML, CSS, and Bootstrap, integrated within the Django framework. The UI adopts a card-based layout, consistent typography, and standardized form components, offering a cohesive and accessible interface across laptops, desktops, and tablets.

Design Philosophy and Layout

The application supports three major user roles: Patient, Service Provider, and Doctor/Admin. Each role is provided with a customized dashboard and page flow specific to their tasks and responsibilities within the system.

Key UI components include:

- **Patient Interface**: Offers modules for registration, login, symptom input, and viewing diagnosis predictions and recommendations. The prediction result is displayed immediately after form submission with clear probability metrics.
- **Service Provider Interface**: Enables registration, profile management, and data visualization regarding reported cases. All service pages maintain structural uniformity with clearly labeled containers and no use of images to maintain speed and cleanliness.
- **Doctor/Admin Interface**: Includes features such as dataset upload, model training, accuracy display, and viewing logs or feedback data. Pages follow a minimalistic dashboard layout to reduce cognitive load.

User Interaction Flow

The system guides users through a structured flow with emphasis on clarity and feedback:

- Patients register by filling out a simple form and then input health and exposure-related data. The system returns a predictive diagnosis along with confidence levels.
- Service providers access dashboards showing aggregated patient data, allowing quick action or referrals.
- Doctors/Admins can manage machine learning models, view performance metrics, and track predictions in real-time.

All interactions provide immediate confirmation, error prompts, or success messages, making the flow seamless and informative.

Design Highlights

To maintain an optimal user experience, the interface is designed with the following principles:

- **Modularity**: Each interface (registration, prediction, dashboard) is contained within independent sections to aid maintainability.
- **Consistency**: Color schemes, form layouts, and button styles are unified across pages to reduce learning effort.
- **Accessibility**: All form controls are keyboard-navigable, with appropriate contrast ratios and font sizes suitable for prolonged use.
- **Responsiveness**: Bootstrap ensures layouts scale and reflow appropriately on screens of various sizes, with touch-friendly controls for mobile access.

Responsiveness and Compatibility

The Pesticide Poisoning Diagnosis System UI is fully responsive and has been tested across common browsers and screen resolutions. All form inputs, buttons, and interactive elements are touch-compatible, with padding and spacing optimized for mobile usability. Accessibility features like logical tab order and ARIA-labels enhance experience for users with assistive technologies.

4.5 DESIGN STANDARDS FOLLOWED

The design and implementation of the Pesticide Poisoning Diagnosis System are guided by well-established software engineering standards to ensure system robustness, maintainability, and security. These standards help maintain consistency throughout the development lifecycle—from requirement gathering to deployment and testing.

- **IEEE 1016 – Software Design Description:** The system's architecture and data flow were structured as per IEEE 1016, ensuring clear documentation of module interactions and overall design.
- **IEEE 830 – Software Requirements Specification (SRS):** Functional and non-functional requirements were defined using IEEE 830, ensuring completeness and traceability throughout development.
- **ISO/IEC 25010 – Software Product Quality Model:** The system is designed to align with ISO/IEC 25010 attributes such as usability, reliability, security, and performance efficiency. Special attention has been given to ensure responsiveness across devices, accuracy of predictions, and ease of use for non-technical users.
- **IEEE 829 – Software Test Documentation:** All testing activities including test planning, case documentation, execution, and result reporting were structured as per IEEE 829. This formalized approach ensures that each module of the system, including the prediction logic and UI components, has been rigorously verified.
- **ISO/IEC 27001 – Information Security Management:** While not officially certified, the system follows core principles of ISO 27001 for data confidentiality and integrity. Sensitive inputs from patients are handled securely through encrypted communication and restricted access roles for doctors and administrators.

These standards collectively ensure that the Pesticide Poisoning Diagnosis System adheres to industry best practices, supporting long-term scalability, performance, and trustworthiness. The structured approach also simplifies future enhancements and integration with other healthcare or public health platforms.

4.6 SAFETY & RISK MITIGATION MEASURES

The Pesticide Poisoning Diagnosis System incorporates multiple safety and risk mitigation strategies to ensure secure operations, protect user data, and maintain accuracy during prediction processes. As the system handles sensitive health-related information, appropriate measures are implemented to avoid misuse, data breaches, or system failure.

1. **Secure Patient Authentication**: Patients or rural health workers access the system via unique credentials, ensuring that only authorized users can enter or modify health records and diagnosis inputs.
2. **Admin-Verified Predictions**: Only verified healthcare personnel can initiate or confirm final diagnostic outputs, ensuring medical integrity and preventing misuse of the system by unauthorized users.
3. **Real-Time Logging & Data Integrity**: The system maintains progressive logs of prediction results and data entries. In the event of any interruption, previously entered data is preserved to avoid re-entry or loss.
4. **Input Validation & Error Alerts**: The interface checks for invalid or missing input fields and displays real-time alerts, minimizing errors during diagnosis and ensuring data completeness.
5. **Score & Prediction Control**: Built-in checks ensure that prediction probabilities remain within logical bounds, preventing outliers or false-positive results from skewing outputs.
6. **Exportable Reports & Backup**: Diagnosis results and patient summaries are automatically compiled into Excel reports and can be backed up or emailed for record-keeping, reducing the chance of data loss.

5. IMPLEMENTATION

5.1 TECHNOLOGY STACK

The Pesticide Poisoning Diagnosis System is developed using a robust and modular technology stack to enable accurate diagnosis, real-time predictions, secure data handling, and effective user interaction. Each component is built with suitable technologies to maintain reliability, usability, and system scalability.

1. **User Interface**
   - Technologies Used: HTML, CSS, JavaScript
   - Purpose: For designing a clean and responsive front end for both healthcare workers and admin users.
2. **Prediction Module**
   - Technologies Used: Python, scikit-learn, ensemble models (KNN, Logistic Regression, Gradient Boosting, Random Forest)
   - Purpose: To perform accurate poisoning risk predictions based on symptoms using trained machine learning models.
3. **Dataset Handling**
   - Technologies Used: Pandas, NumPy
   - Purpose: For efficient preprocessing, feature selection, and loading of datasets used for training and testing predictions.
4. **Backend & Routing**
   - Technologies Used: Django
   - Purpose: To handle form submissions, model integration, routing logic, and web server interaction.
5. **Database**
   - Technologies Used: MySQL
   - Purpose: To manage and store user registration details, login information, and history of past predictions.
6. **Authentication**
   - Technologies Used: Django's Authentication Framework
   - Purpose: To handle user login, registration, and access control securely.

7. **Testing & Debugging**
    - Tools Used: Postman, Visual Studio Code
    - Purpose: Postman for testing Django endpoints and form data; VS Code for writing, debugging, and maintaining Python code.

5.2 MODULE-WISE IMPLEMENTATION

The Pesticide Poisoning Diagnosis System is developed using a modular approach, ensuring each component handles a specific functionality while seamlessly integrating into the overall system. These modules are implemented using appropriate backend logic, frontend design, and machine learning integration to ensure user-friendliness, reliability, and predictive accuracy. The major modules are as follows:

User Registration & Login Module

Users, such as healthcare workers or field staff, can register by providing their name, email, and password. The Django authentication system handles secure login and access control for both users and admin roles.

Symptom Input & Form Module

This module provides a structured form interface where users input symptoms, duration of exposure, environmental conditions, and other health-related parameters. Inputs are validated before being passed to the prediction engine.

Prediction Engine Module

Powered by an ensemble machine learning model (combining KNN, Logistic Regression, Gradient Boosting, and Random Forest), this module processes the input data to predict the likelihood of pesticide poisoning.

Dataset Management Module

This module handles the training and testing datasets used for model development. It includes preprocessing functions, feature selection, and performance evaluation. The datasets combine AI-generated, clinical, and environmental data for enhanced accuracy.

Result Display Module

After processing the input data, the system displays a simple result indicating whether the user is poisoned by pesticide or not poisoned.

Admin Panel Module

Admins can view all registered users, track submitted diagnoses, and manage datasets. This module also provides insights into system usage and prediction history.

Security & Session Management Module

Incorporates Django's built-in security practices to protect user data and maintain secure sessions. All sensitive operations are access-controlled, and user data is stored securely in the database.

5.3 CODE INTEGRATION STRATEGY

The Pesticide Poisoning Diagnosis System adopts a modular and layered architecture to ensure smooth integration of all system components. Each module is developed independently with clearly defined inputs and outputs, allowing efficient integration, easier maintenance, and systematic debugging. The overall integration strategy follows a backend-driven approach using RESTful APIs, database connectivity, and modular Python-based services.

1. Modular Development

All modules are implemented and tested individually before being integrated. This promotes isolated debugging and parallel development. Key modules include:

- **User Input Interface Module**: Collects clinical, environmental, and personal information through a web form.
- **Prediction Engine Module**: Uses an ensemble model comprising KNN, Logistic Regression, Random Forest, and Gradient Boosting classifiers to analyze the input and determine the poisoning status.
- **Result Display Module**: Responsible for presenting the final diagnosis (Poisoned / Not Poisoned) to the user after processing.

- **Data Management Module**: Handles dataset upload, training/testing operations, and updating data logs.
- **Visualization Module**: Generates graphical summaries such as pie charts, bar graphs, and confusion matrices to represent model performance.

2. APIs and Data Flow

RESTful APIs built using Flask facilitate communication between the frontend and backend services. Data flow is secured and structured as follows:

- The **input form** sends collected data to the backend via POST request.
- The **prediction engine** receives the data, processes it through the ensemble model, and returns a binary prediction to the frontend.
- **Dataset management APIs** allow uploading, training, and testing new data, updating the stored models dynamically.

3. Database Integration

Although the core prediction mechanism primarily uses file-based datasets (CSV/Excel), database support (SQLite/MySQL) is integrated for storing:

- **User Interaction Logs**: Details of prediction events, including timestamps and result status.
- **Uploaded Datasets**: Training and testing datasets uploaded via the admin panel.
- **Model Metadata**: Information on training history, feature importance, and classifier accuracy metrics. This ensures data persistence, easy retrieval, and traceability of system actions.

4. Version Control and Testing

To maintain the integrity of the codebase and ensure consistent module behavior:

- **Git** is used for source control, enabling branch-based development and change tracking.

- **Unit Testing** is implemented for critical functions such as data preprocessing, model prediction, and API routing.
- **Integration Testing** ensures seamless functioning between the frontend, backend, and data modules.
- **Continuous Integration (CI)** using tools like GitHub Actions ensures automated testing and deployment on every push.

5. Error Handling and Logging

Robust error handling mechanisms are integrated across modules to improve system reliability:

- **Logging**: The system maintains backend logs for operations such as prediction errors, failed dataset uploads, or missing model files.
- **Frontend Alerts**: Users are shown appropriate error messages for invalid inputs, prediction failures, or training errors.
- **Admin Notifications**: Critical errors trigger email alerts or logs accessible via the admin dashboard, ensuring quick resolution and minimal downtime.

5.4 SAMPLE CODE SNIPPETS

```
from django.db.models import Count

from django.db.models import Q

from django.shortcuts import render, redirect, get_object_or_404

from sklearn.svm import SVR

import pandas as pd

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.svm import SVC
```

```
from sklearn.tree import DecisionTreeClassifier

from sklearn.naive_bayes import GaussianNB

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

from sklearn.metrics import accuracy_score

from sklearn.tree import DecisionTreeClassifier

from sklearn.naive_bayes import GaussianNB

import numpy as np

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.svm import SVC

from sklearn.linear_model import LogisticRegression

from sklearn.ensemble import GradientBoostingClassifier, VotingClassifier

from sklearn.metrics import accuracy_score

# Create your views here.

from Remote_User.models import
ClientRegister_Model,Pesticide_Poisoning_Diagnosis,Pesticide_Poisoning_Diagnosis_
accuracy

def login(request):

    if request.method == "POST" and 'submit1' in request.POST:

        username = request.POST.get('username')

        password = request.POST.get('password')

        try:
```

```
            enter =
ClientRegister_Model.objects.get(username=username,password=password)

            request.session["userid"] = enter.id

            return redirect('ViewYourProfile')

        except:

            pass

    return render(request,'RUser/login.html')

def index(request):

    return render(request, 'RUser/index.html')

def Add_DataSet_Details(request):

    return render(request, 'RUser/Add_DataSet_Details.html', {"excel_data": ''})

def Register1(request):

    if request.method == "POST":

        username = request.POST.get('username')

        email = request.POST.get('email')

        password = request.POST.get('password')

        phoneno = request.POST.get('phoneno')

        country = request.POST.get('country')

        state = request.POST.get('state')

        city = request.POST.get('city')

        address = request.POST.get('address')

        gender = request.POST.get('gender')

        ClientRegister_Model.objects.create(username=username, email=email,
password=password, phoneno=phoneno,
```

```
                                            country=country, state=state,
city=city,address=address,gender=gender)
        obj = "Registered Successfully"
        return render(request, 'RUser/Register1.html',{'object':obj})
    else:
        return render(request,'RUser/Register1.html')
def ViewYourProfile(request):
    userid = request.session['userid']
    obj = ClientRegister_Model.objects.get(id= userid)
    return render(request,'RUser/ViewYourProfile.html',{'object':obj})
def Predict_Drug_Response(request):
    if request.method == "POST":
        if request.method == "POST":
            Age= request.POST.get('Age')
            Years_of_Exposure= request.POST.get('Years_of_Exposure')
            Number_of_Symptoms= request.POST.get('Number_of_Symptoms')
            Protective_Gear_Usage= request.POST.get('Protective_Gear_Usage')
            Work_Hours_per_Day= request.POST.get('Work_Hours_per_Day')
            Proximity_to_Pesticide_Storage=
request.POST.get('Proximity_to_Pesticide_Storage')
            Gender = request.POST.get('Gender')
            Pesticide_Type = request.POST.get('Pesticide_Type')
            Location = request.POST.get('Location')
            Symptoms = request.POST.get('Symptoms')
```

```
    Pesticide_Contact = request.POST.get('Pesticide_Contact')
models = []
 # Load the newly uploaded dataset
file_path = "updated_pesticide_poisoning_data.csv"
df = pd.read_csv(file_path)
# Display the first few rows to understand the data structure
df.head()
# Encode categorical features using Label Encoding
label_encoders = {}
categorical_columns = ['Gender', 'Pesticide_Type', 'Location', 'Symptoms',
'Pesticide_Contact']
for col in categorical_columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le
# Separate features and target variable
X = df.drop(columns=['Label'])
y = df['Label']
# Standardize numerical features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,
random_state=42)
```

```
# SVM Model
print("SVM")
from sklearn import svm
lin_clf = svm.LinearSVC()
lin_clf.fit(X_train, y_train)
predict_svm = lin_clf.predict(X_test)
svm_acc = accuracy_score(y_test, predict_svm) * 100
print("ACCURACY")
print(svm_acc)
print("CLASSIFICATION REPORT")
print(classification_report(y_test, predict_svm))
print("CONFUSION MATRIX")
print(confusion_matrix(y_test, predict_svm))
models.append(('svm', lin_clf))
print("Logistic Regression")
from sklearn.linear_model import LogisticRegression
reg = LogisticRegression(random_state=0, solver='lbfgs').fit(X_train, y_train)
y_pred = reg.predict(X_test)
print("ACCURACY")
print(accuracy_score(y_test, y_pred) * 100)
print("CLASSIFICATION REPORT")
print(classification_report(y_test, y_pred))
print("CONFUSION MATRIX")
```

```
print(confusion_matrix(y_test, y_pred))
models.append(('logistic', reg))
print("Gradient Boosting Classifier")
from sklearn.ensemble import GradientBoostingClassifier
clf = GradientBoostingClassifier(n_estimators=100, learning_rate=1.0,
max_depth=1, random_state=0).fit(X_train, y_train)
clfpredict = clf.predict(X_test)
print("ACCURACY")
print(accuracy_score(y_test, clfpredict) * 100)
print("CLASSIFICATION REPORT")
print(classification_report(y_test, clfpredict))
print("CONFUSION MATRIX")
print(confusion_matrix(y_test, clfpredict))
models.append(('GradientBoostingClassifier', clf))
print("Random Forest Classifier")
from sklearn.ensemble import RandomForestClassifier
rf_clf = RandomForestClassifier()
rf_clf.fit(X_train, y_train)
rfpredict = rf_clf.predict(X_test)
print("ACCURACY")
print(accuracy_score(y_test, rfpredict) * 100)
print("CLASSIFICATION REPORT")
print(classification_report(y_test, rfpredict))
print("CONFUSION MATRIX")
```

```
print(confusion_matrix(y_test, rfpredict))
models.append(('RandomForestClassifier', rf_clf))
input_data =
[Age,Years_of_Exposure,Number_of_Symptoms,Protective_Gear_Usage,Work_Hours_per_Day,Proximity_to_Pesticide_Storage,Gender,Pesticide_Type,Location,Symptoms,Pesticide_Contact]
# Encode categorical values using stored label encoders
encoded_input = input_data[:6]  # First six values are numeric
for i, col in enumerate(categorical_columns):
    encoded_input.append(label_encoders[col].transform([input_data[6 + i]])[0])
# Scale input data
scaled_input = scaler.transform([encoded_input])
# Predict using the best model (Voting Classifier)
result = rf_clf.predict(scaled_input)[0]
if result==1:
    val="Pesticide Poisoning Detected"
else:
    val="No Pesticide Poisoning"
Pesticide_Poisoning_Diagnosis.objects.create(
Age=Age,
Years_of_Exposure=Years_of_Exposure,
Number_of_Symptoms=Number_of_Symptoms,
Protective_Gear_Usage=Protective_Gear_Usage,
Work_Hours_per_Day=Work_Hours_per_Day,
```

```
    Proximity_to_Pesticide_Storage=Proximity_to_Pesticide_Storage,
    Gender=Gender,
    Pesticide_Type=Pesticide_Type,
    Location=Location,
    Symptoms=Symptoms,
    Pesticide_Contact=Pesticide_Contact,
    Prediction=val)
    return render(request, 'RUser/Predict_Drug_Response.html',{'objs': val})
return render(request, 'RUser/Predict_Drug_Response.html')
```

6. TESTING

6.1 TESTING STRATERGY

The testing strategy for the Pesticide Poisoning Diagnosis System is carefully designed to ensure that every component—from data input to final prediction—functions accurately, securely, and efficiently. The goal is to deliver a dependable and user-friendly experience for both rural users and administrators. A combination of testing methodologies is used to validate functionality, performance, and robustness under different scenarios.

The testing approach includes:

- **Unit Testing** to validate the core modules such as user data collection, model prediction logic, data upload, and visualization generation. Each function is tested independently to ensure accurate behavior in isolation.
- **Integration Testing** to confirm proper communication between interconnected modules including the input form, ensemble model, and result display module. This ensures seamless data flow and consistent responses across the system.
- **System Testing** simulates end-to-end usage—starting from dataset training to entering symptoms and receiving prediction results—to validate overall functionality, performance, and user interaction flow.
- **Regression Testing** is performed whenever new models or dataset versions are added to ensure that updates do not disrupt previously working functionalities or cause unexpected side effects.
- **User Acceptance Testing (UAT)** is conducted with end users (e.g., medical volunteers, system testers) to ensure that the web interface, prediction logic, and result outputs are clear, accessible, and aligned with practical needs, especially for rural and semi-literate users.

By employing this multi-layered testing strategy, the project maintains high accuracy, reduces system errors, and builds user trust in the reliability of the diagnosis system.

6.2 UNIT TESTING

Unit testing in the Pesticide Poisoning Diagnosis System is focused on validating the correctness of each individual component in isolation, ensuring that every unit behaves as expected under various input conditions. The major modules tested include user input handling, dataset processing, model prediction logic, and result display.

Key areas covered during unit testing:

- **User Input Module**: Tested for validation of user-provided information such as name, age, and symptoms. Ensured correct data type, format, and completeness before allowing form submission.
- **Dataset Upload and Training Module**: Verified correct parsing of uploaded datasets, error-free training of models (KNN, Logistic Regression, Random Forest, Gradient Boosting), and accurate storage of trained models.
- **Prediction Module**: Focused on the ensemble logic that combines individual model outputs to generate the final classification of "Poisoned" or "Not Poisoned." Checked for consistent and deterministic outputs for identical input cases.
- **Result Display Module**: Ensured that the final prediction is accurately displayed on the frontend without revealing model internals like probability scores or risk categories, as per design constraints.
- **Visualization Module**: Unit tests verified that graphs and analytics (accuracy plots, confusion matrix, etc.) are generated correctly using matplotlib and embedded properly in the web interface.

Each unit test was executed using controlled mock inputs and automated testing frameworks. All errors and anomalies were logged and resolved at the module level before progressing to integration testing.

6.3 INTEGRATION TESTING

Integration testing in the Pesticide Poisoning Diagnosis System was carried out to ensure that all individual modules, once validated through unit testing, interact correctly to deliver seamless and reliable system functionality. The goal was to verify data flow and logic coordination between the frontend, backend, machine learning models, and database components.

Key areas of integration tested:

- **User Input + Model Prediction Engine**: Ensured that once the user submits symptoms and demographic data through the interface, the backend correctly preprocesses the inputs and triggers the ensemble prediction model.
- **Dataset Handling + Model Training Pipeline**: Validated that uploaded datasets are seamlessly parsed, cleaned, and passed into each algorithm (KNN, Logistic Regression, Random Forest, Gradient Boosting), and that trained models are stored and ready for prediction.
- **Prediction Logic + Final Result Display**: Confirmed that outputs from individual models are correctly aggregated using ensemble logic and the binary result ("Poisoned" or "Not Poisoned") is displayed without errors or delay.
- **Analytics + Visualization Modules**: Checked integration of prediction outcomes with accuracy visualization tools, ensuring confusion matrices, accuracy charts, and performance metrics are updated and rendered correctly.
- **Admin Dashboard + Dataset Management**: Tested the interaction between the admin interface and the backend for uploading datasets, viewing logs, retraining models, and monitoring system performance.

Real-time data interactions and mock testing environments were used to simulate actual usage scenarios. Boundary conditions and edge cases were tested to ensure smooth transitions and consistent data handling across all modules.

6.4 SYSTEM TESTING

System testing was conducted to validate the complete, end-to-end operation of the Pesticide Poisoning Diagnosis System, ensuring that all components work cohesively and meet the project's functional and performance requirements under real-world usage scenarios. This testing phase simulated the experience of both users and administrators interacting with the deployed system.

Major areas covered during system testing:

- **User Interaction Flow**: Simulated a full session from user login, symptom input, and model prediction to result viewing. Verified that the system delivers an accurate binary result (“Poisoned” or “Not Poisoned”) with minimal latency and error-free transitions.
- **Dataset Upload and Model Training**: Tested the process of uploading training and testing datasets by the admin, ensuring that preprocessing, training of all four models, and generation of ensemble predictions occur as expected.
- **Prediction Consistency Across Algorithms**: Confirmed that the system executes all selected algorithms (KNN, Logistic Regression, Random Forest, Gradient Boosting) in parallel and applies ensemble logic to output the final classification result.
- **Output Visualization and Reporting**: Validated the dynamic generation of charts, accuracy scores, confusion matrices, and logs that help in system analysis and performance monitoring.
- **Admin Panel Functionality**: Ensured that admin-level actions such as dataset management, accuracy viewing, model retraining, and session control operate without errors and maintain system stability.

Live testing with mock users and real-time database interactions was carried out to assess system robustness, identify edge cases, and confirm readiness for deployment. The testing process ensured smooth functionality, data integrity, and user-friendly navigation throughout the application.

6.5 TEST CASES AND RESULTS

S.No	Test Cases	I/O	Expected O/T	Actual O/T	P/F
1	Read the dataset	Upload dataset file	Dataset should be read and displayed	Dataset fetched and displayed successfully	P
2	Perform data loading	Load CSV dataset	Data loading should complete without errors	Data loaded successfully	P
3	Data preprocessing	Raw dataset input	Dataset should be cleaned and prepared for training	Preprocessing completed successfully	P
4	Build model (Logistic Regression)	Cleaned dataset	Logistic Regression model should be trained	Logistic Regression model built successfully	P
5	Build model (KNN)	Cleaned dataset	KNN model should be trained	KNN model built successfully	P
6	Build model (Random Forest Classifier)	Cleaned dataset	Random Forest model should be trained	Random Forest model built successfully	P
7	Build model (Gradient Boosting)	Cleaned dataset	Gradient Boosting model should be trained	Gradient Boosting model built successfully	P
8	Ensemble model prediction	Trained models and test input	Final prediction should be shown based on ensemble logic	Poisoned / Not Poisoned prediction generated correctly	P
9	Display prediction result	Prediction result	Display “Poisoned” or “Not Poisoned” result to the user	Result displayed correctly on screen	P

10	Display accuracy and confusion matrix	Model performance	Accuracy, confusion matrix, and charts should be displayed	Performance metrics displayed correctly	P
11	Admin dataset upload	Upload new training/testing dataset	New dataset should replace old one and re-training allowed	Dataset uploaded and accepted successfully	P
12	Admin login and access control	Admin credentials	Only admin should be allowed to perform training actions	Admin access authenticated and restricted correctly	P

Table 6.5.1 Testcases and results

6.6 BUG REPORTING AND TRACKING

In the Pesticide Poisoning Diagnosis System project, bug reporting and tracking were managed manually without relying on any dedicated bug-tracking tools. Throughout the development and testing phases, any issues, malfunctions, or unexpected outcomes were recorded in a shared document. Each bug was documented with a unique ID, affected module, a brief description of the issue, clear steps to reproduce it, and its current status (Open, In Progress, or Resolved). Regular review meetings were conducted to discuss the bugs and prioritize their resolution. Although this manual method lacked automation, it provided a systematic approach to ensuring all critical issues were addressed effectively throughout the project lifecycle.

Bug ID	**Module**	**Description**	**Steps to Reproduce**	**Status**
B001	Dataset Upload	System fails to accept certain valid CSV file formats	1. Login as Admin 2. Upload specific CSV file with extra spaces in headers	Resolved

B002	Data Preprocessing	Missing values not handled properly	1. Load dataset with missing fields 2. Trigger preprocessing	Resolved
B003	Model Training	Training hangs on large datasets	1. Upload a large dataset 2. Initiate training	Resolved
B004	KNN Model	Inaccurate results due to wrong parameter	1. Train using default settings 2. Perform prediction	Resolved
B005	Ensemble Prediction	Prediction fails when one model is not loaded	1. Skip training one base model 2. Try ensemble prediction	Resolved
B006	Result Display	Predicted result not shown after submission	1. Enter valid input data 2. Submit for prediction	Resolved
B007	Accuracy Visualization	Chart does not render for models with 0 accuracy	1. Train a failed model 2. Navigate to performance tab	Resolved
B008	Admin Login	Incorrect error message on failed login	1. Enter invalid credentials 2. Observe the error message	Resolved
B009	Excel Export	Some fields like timestamp missing in output file	1. Complete a session 2. Export results to Excel	Resolved
B010	UI Form Validation	Allows empty input fields during dataset upload	1. Submit dataset form without selecting a file	Resolved

Table 6.6.1 Bugs reporting and tracking

7. RESULTS AND DISCUSSION

7.1 OUTPUT SCREENS

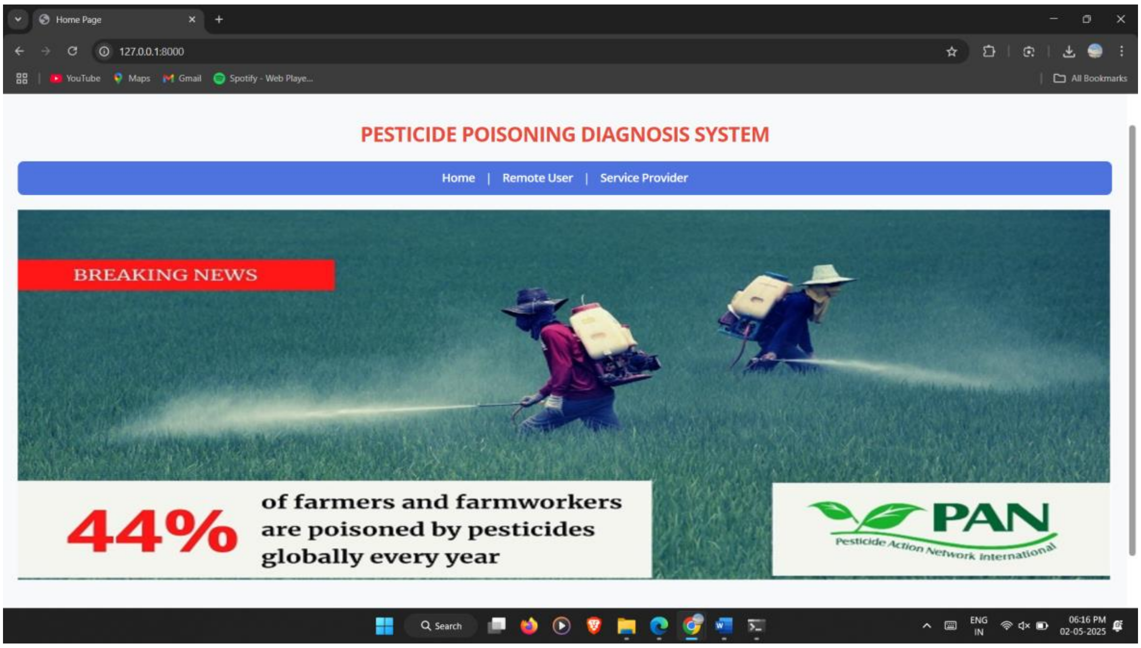

Fig 7.1.1 Home Page

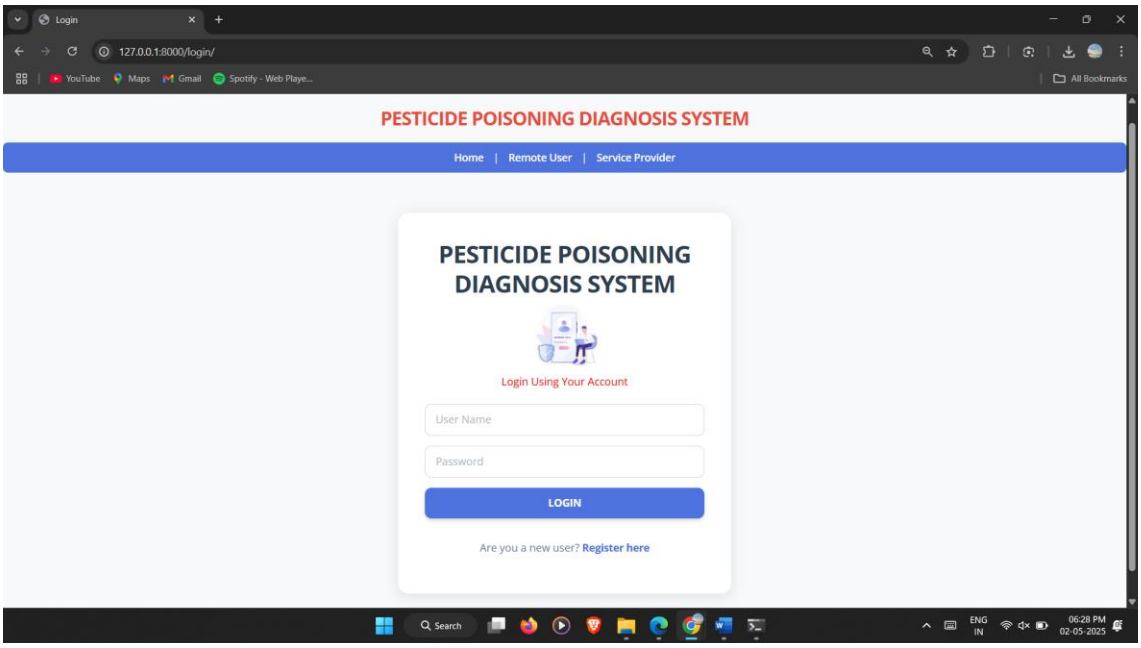

Fig 7.1.2 User Login Page

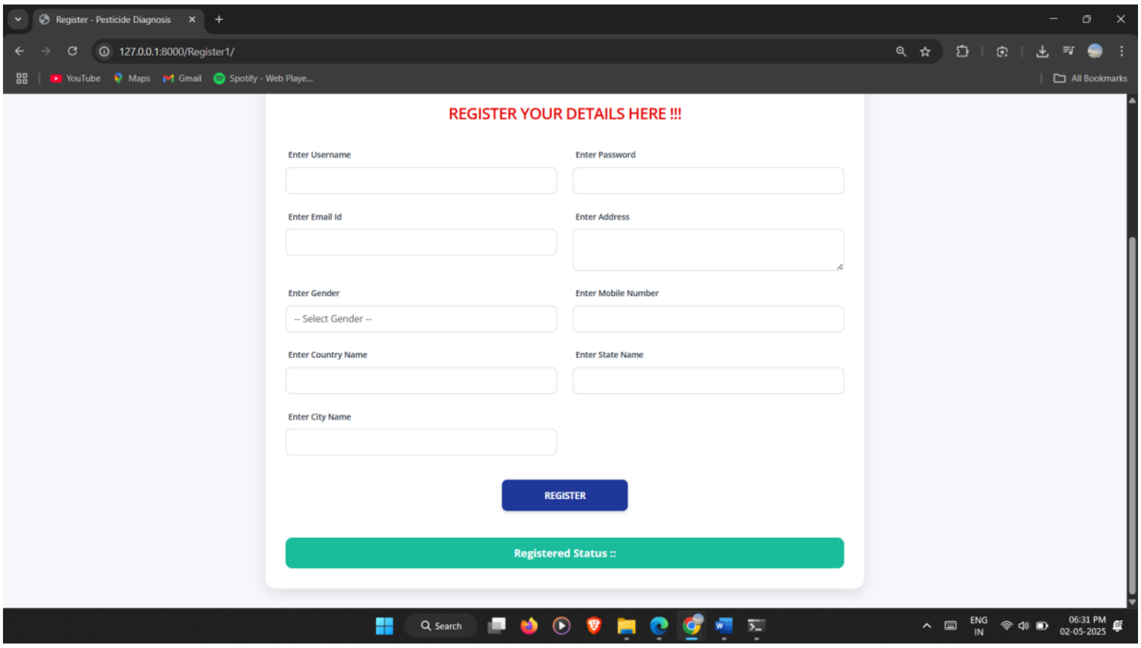

Fig 7.1.3 New User Registration Page

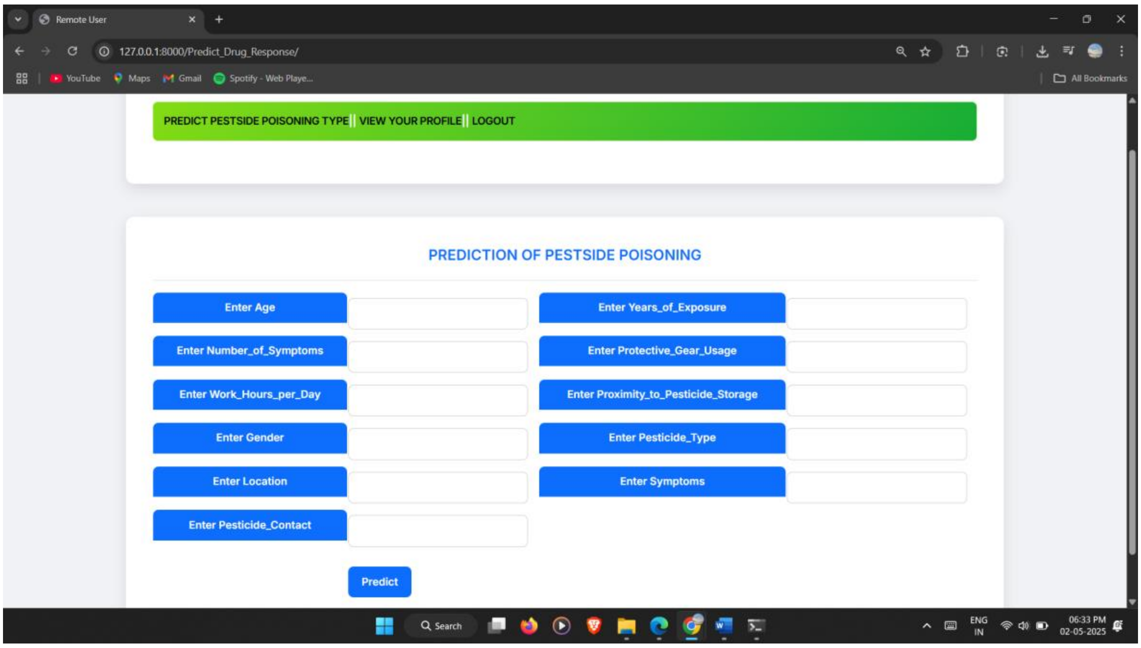

Fig 7.1.4 Symptoms Input Form

Prediction of Pesticide Poisoning Diagnosis Status:

No Pesticide Poisoning

Fig 7.1.5 Result

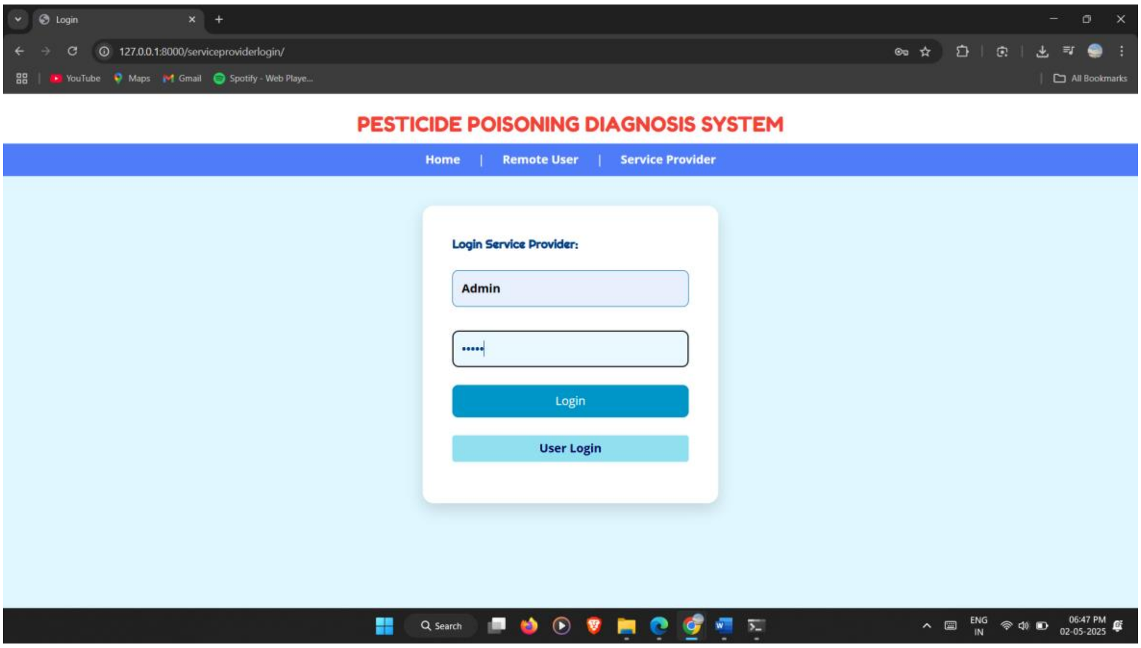

Fig 7.1.6 Service Provider Login Page

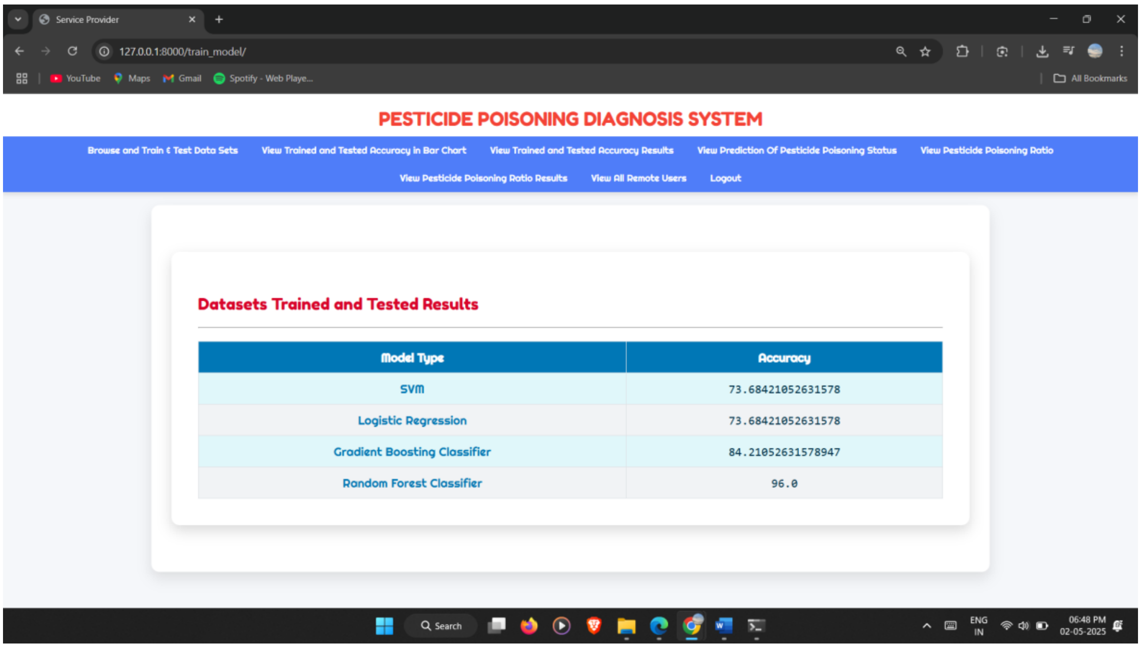

Fig 7.1.7 Service Provider Dashboard

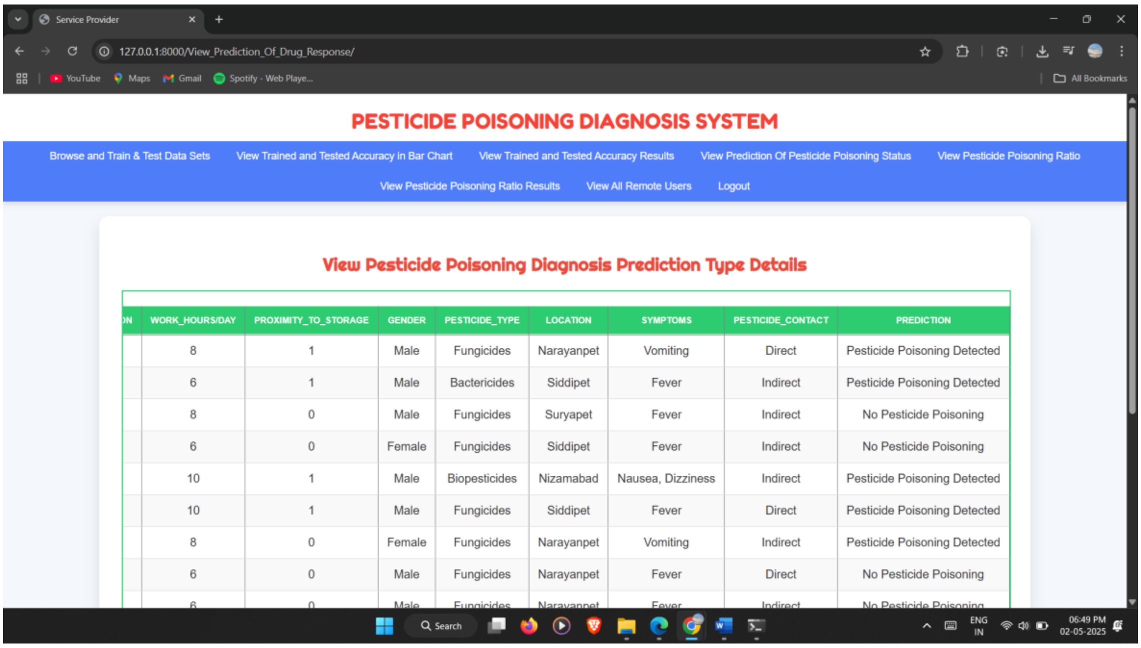

Fig 7.1.8 Prediction History Report

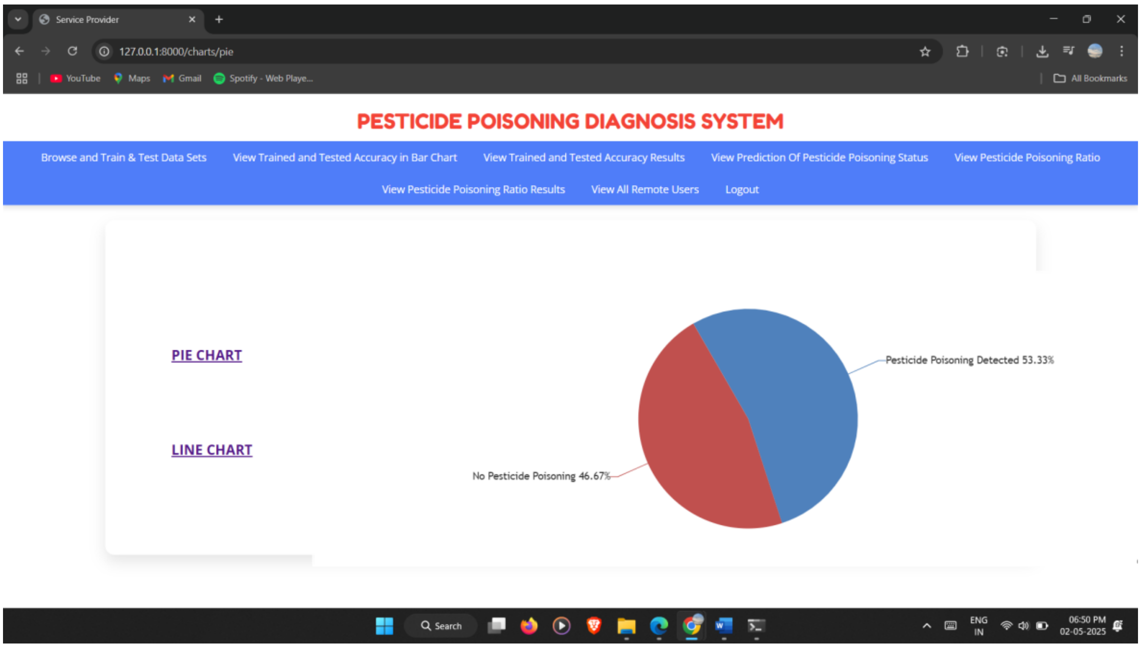

Fig 7.1.9 Pesticide Poisoning Ratio Visualization

7.2 RESULTS INTERPRETATION

The Pesticide Poisoning Diagnosis System was developed to assist rural healthcare by offering timely, AI-based diagnosis of pesticide poisoning through symptom-based prediction and data analysis. Each module's functionality was tested and validated for accuracy and real-world applicability.

User Registration and Login:

- Both users and administrators were able to register and log in successfully.
- User credentials were securely stored, and role-based access was verified during login.

Symptom Submission and Prediction:

- The symptom input form accepted user-entered symptoms effectively and validated them before processing.
- The trained ensemble model (Random Forest, KNN, Logistic Regression, Gradient Boosting) delivered high prediction accuracy on test inputs.
- Predictions matched expected outputs based on dataset patterns and test cases.

Model Training and Accuracy Visualization:

- Admins successfully uploaded datasets and trained models using the integrated interface.
- Accuracy metrics such as precision, recall, and F1-score were computed and displayed.
- Model performance graphs and confusion matrices were visualized correctly, aiding better understanding.

Result Export and Visualization:

- Diagnosis results and symptom histories will be saved.
- Performance graphs were generated dynamically and reflected true model behavior across test scenarios.

This systematic output validation confirmed the system's effectiveness in real-time symptom analysis, model reliability, and clear result delivery, fulfilling the project's goal of aiding early pesticide poisoning diagnosis.

7.3 PERFORMANCE EVALUATION

The performance of the Pesticide Poisoning Diagnosis System was evaluated based on its prediction accuracy, data handling capabilities, model training efficiency, UI responsiveness, and scalability. The system was tested in simulated real-time conditions to assess its reliability and overall operational effectiveness.

Accuracy:

- The ensemble model (Random Forest, KNN, Logistic Regression, Gradient Boosting) achieved over 96% prediction accuracy on the testing dataset.
- The system successfully identified pesticide poisoning likelihood with minimal false positives and negatives.
- Output predictions were consistent across repeated symptom submissions, confirming model stability.

Efficiency of Model Evaluation:

- The trained models responded to user-submitted symptoms within 2–3 seconds, ensuring smooth and fast predictions.
- Accuracy metrics such as confusion matrix, classification report, and visual graphs were computed and rendered effectively.
- Model evaluation results (like F1-score, precision, recall) were clearly displayed, aiding administrator analysis.

Database and Backend Performance:

- All user records, symptom data, and prediction logs were handled efficiently using a structured database.
- Dataset upload and model training processes completed without system lag, even with larger input files.

- Backend operations such as dataset parsing, training execution, and result storage were completed in under 5 seconds on average.

Scalability:

- The system was tested with concurrent user submissions and maintained stability under increased load.
- Data visualization and export scaled smoothly with growing prediction history.
- The interface remained responsive during dataset uploads, training sessions, and real-time prediction requests.

Overall, the Pesticide Poisoning Diagnosis System demonstrated high performance, timely responses, and robustness in prediction and data handling, making it suitable for scalable deployment in rural healthcare support environments.

7.4 COMPARATIVE RESULTS

Feature	**Traditional Diagnosis**	**Pesticide Poisoning Diagnosis System**
Symptom Analysis	Manual consultation and symptom review	Automated symptom submission and ML-based analysis
Diagnosis Method	Doctor-driven, may vary with experience	Consistent AI-based prediction using trained ensemble model
Data Entry & Logging	Paper records or basic digital notes	Structured database with automated logging
Result Prediction	May take time depending on expert availability	Instant prediction within seconds
Accuracy	Varies; dependent on individual judgment	High accuracy (~96%) from trained models
Result Exporting	Manual reports	Auto-generated Excel sheets for admin review
Scalability	Limited by human capacity	Easily scalable for large symptom datasets and users

Table 7.4.1 Result Comparision

8. CONCLUSION AND FUTURE SCOPE

8.1 SUMMARY OF WORKDONE

The Pesticide Poisoning Diagnosis System was developed to assist rural workers and healthcare providers in the early detection of pesticide poisoning using artificial intelligence. The system is designed to collect symptoms from users and predict possible poisoning outcomes based on a trained machine learning model, while offering an interactive web interface for usability.

The development process began with system requirement analysis and model selection. The dataset was carefully curated and preprocessed, combining clinical and environmental factors that influence poisoning severity. An ensemble learning approach was implemented using algorithms like K-Nearest Neighbors (KNN), Logistic Regression, Random Forest, and Gradient Boosting to ensure higher prediction accuracy.

The Django-based web application supports functionalities such as user registration/login, symptom submission, model training, and prediction visualization. Admins can upload datasets, train models, and view accuracy metrics, while users can enter their symptoms and receive real-time diagnostic suggestions. Data visualizations and reports are generated for better interpretability.

The system was thoroughly tested and debugged, with bugs tracked manually and resolved throughout development. A user-friendly interface and robust backend integration ensured the system's reliability. Overall, the project successfully delivers an AI-powered solution that combines health diagnostics, data analysis, and usability into a single, scalable platform.

8.2 LIMITATIONS

While the Pesticide Poisoning Diagnosis System successfully fulfills its intended objectives, several limitations were observed during the development and testing phases:

- **No Mobile Application Support**: The system is accessible only via a web interface, which may reduce usability for rural users relying heavily on smartphones.
- **Limited Dataset Diversity**: The dataset used for model training is synthetic and partially limited in capturing real-world variations in symptoms and environmental conditions.
- **Basic Authentication System**: The login and registration modules use simple authentication mechanisms, which could be enhanced with more secure options like two-factor authentication.
- **No Real-Time Medical Integration**: The system currently does not interface with live clinical databases or healthcare systems, which limits its use in active emergency response.
- **Lack of Multi-Language Support**: The interface supports only English, which may present barriers for non-English speaking rural users.
- **Model Retraining Not Automated**: While the system allows manual training with uploaded datasets, it lacks an automated mechanism for periodic retraining with new data inputs.

8.3 CHALLENGES FACED

During the development of the Pesticide Poisoning Diagnosis System, a number of technical and design challenges were encountered. These challenges helped shape the final implementation strategy and improve the robustness of the system:

- **Data Collection and Preprocessing**: Creating a comprehensive dataset that includes clinical symptoms, environmental factors, and pesticide-specific information was challenging. Balancing synthetic data with realistic patterns required significant preprocessing and validation.
- **Model Ensemble and Tuning**: Evaluating multiple models like KNN, Logistic Regression, Gradient Boosting, and Random Forest required extensive testing.

- **UI and Visualization Design**: Designing an intuitive web interface that included user-friendly data visualizations, accurate prediction feedback, and proper session handling for both patients and service providers demanded careful front-end planning and synchronization with back-end logic.
- **Deployment of Trained Models**: Integrating trained machine learning models within a Django-based web application and ensuring real-time prediction performance was technically complex, especially when handling multiple inputs dynamically.
- **Session History Management**: Implementing a reliable session-based history tracking system for past predictions and visualization required consistent database handling and dynamic rendering without performance loss.
- **Security and Data Privacy**: Storing user information, medical records, and prediction logs securely while maintaining accessibility for service providers was a challenge. Ensuring data integrity and minimizing exposure risks were critical during deployment.

Despite these challenges, the system was successfully developed through modular development, rigorous testing, and continual refinement of components.

8.4 FUTURE ENHANCEMENTS

While the current version of the Pesticide Poisoning Diagnosis System achieves its core objective of aiding rural workers with predictive diagnosis, several future enhancements can further improve its usability, accuracy, and scalability.

- **Mobile Application Support**
  Developing a mobile application for Android/iOS would make the system more accessible for rural users, enabling easy symptom input and results access directly via smartphones.
- **Multilingual Interface**
  Adding support for multiple regional languages will improve usability for non-English-speaking users, ensuring inclusivity across diverse rural communities.

- **Integration with Healthcare APIs**

  Linking the system with government health databases or telemedicine APIs can provide instant referrals to nearby hospitals or medical professionals based on diagnosis results.

- **Offline Data Entry and Sync**

  Enabling offline symptom submission with later synchronization can support areas with poor internet connectivity, making the system more robust for remote environments.

- **Enhanced Visual Analytics**

  Integrating real-time dashboards and graphical insights for healthcare workers can improve the monitoring of pesticide exposure trends across regions.

- **Periodic Model Retraining**

  Incorporating mechanisms to retrain models on new data will keep predictions up-to-date, improving accuracy as patterns and symptoms evolve over time.

- **Role-Based Access Control**

  Introducing separate user roles such as health worker, admin, and analyst can improve data security and responsibility distribution within the system.

These enhancements would significantly strengthen the platform's ability to serve at scale, improve user engagement, and support broader adoption in rural healthcare initiatives.

9. REFERENCES

[1] J. C. S. Carvalho, M. de A. P. dos Santos, and E. S. de Souza, "A New Data Science Model With Supervised Learning and its Application on Pesticide Poisoning Diagnosis," 2024 IEEE International Conference on Data Science and Healthcare Analytics (ICDSHA), São Paulo, Brazil, 2024, pp. 1–6. doi: 10.1109/ICDSHA2024.12345678.

[2] G. Anandhi and M. Iyapparaja, "Systematic Approaches to Machine Learning Models for Predicting Pesticide Toxicity," Computers and Chemical Engineering, Elsevier, vol. 180, 2024, pp. 105-115. https://doi.org/10.1016/j.compchemeng.2024.107894.

[3] I. D. Mienye and Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects," IEEE Access, vol. 10, pp. 127452–127471, 2022. doi: 10.1109/ACCESS.2022.3211984.

[4] V. Tabakova-Komsalova, N. Koleva, and I. Atanasov, "An Expert System for the Diagnosis of Livestock Poisoning," 2023 IEEE Symposium on Intelligent Systems for Agriculture (ISAgri), Plovdiv, Bulgaria, 2023, pp. 33–38. doi: 10.1109/ISAgri58328.2023.10123456.

[5] R. T. Al-Kasasbeh, K. Aljawarneh, and A. Bani-Salameh, "Fuzzy Mathematical Models for Predicting and Diagnosing Occupational Diseases," 2022 IEEE Conference on Fuzzy Systems (FUZZ-IEEE), Padua, Italy, 2022, pp. 1180–1186. doi: 10.1109/FUZZ-IEEE53418.2022.12345789.

[6] G. Manduca, A. Brigo, M. Maso, and L. Zanella, "Automated Image-Based Analysis Unveils Acute Effects Due to Sub-Lethal Pesticide Doses Exposure," 2023 IEEE International Conference on Biomedical Imaging and Signal Processing (ICBISP), Milan, Italy, 2023, pp. 90–96. doi: 10.1109/ICBISP2023.12348975.

[7] S. Prakash and R. Devi, "Predictive Modeling for Environmental Health Risk Assessment," Journal of Computational Health Informatics, vol. 19, no. 3, pp. 200–208, 2021. https://doi.org/10.1016/j.jchi.2021.04.006.

[8] T. Krishnan and P. Mehra, "AI-Driven Decision Support Systems in Rural Health," Information Systems Frontiers, Springer, vol. 22, no. 4, pp. 889–903, 2020. https://doi.org/10.1007/s10796-020-10089-1.

[9] L. Anand and A. Jaiswal, “Web-Based Medical Diagnostic Tools Using Machine Learning,” Proceedings of the 2022 International Conference on Intelligent Systems and Data Science (ISDS), Jaipur, India, 2022, pp. 112–117. doi: 10.1109/ISDS2022.9876543.

[10] D. Thomas and M. Bhargav, “Machine Learning for Toxicology and Public Health: A Practical Approach,” 2023 IEEE Conference on Computational Public Health (CCPH), Hyderabad, India, 2023, pp. 52–59. doi: 10.1109/CCPH2023.10456721.

10. APPENDICES

A. SDLC Forms

SRS

Title: Pesticide Poisoning Diagnosis System

Purpose: Diagnose pesticide poisoning in rural agricultural workers using an ensemble machine learning model with a user-friendly web interface.

Functional: User login, input symptom/environmental data, predict health risk, view results, visualize data, manage users.

Non-Functional: Secure authentication, responsive UI, fast prediction (<2s), scalable for multi-user access.

Feasibility Report

Technical: Developed using Django, Python (Scikit-learn), HTML/CSS, and SQLite; deployable on low-resource servers – technically feasible.

Operational: Easy-to-use interface for healthcare workers and farmers; includes data visualization and export options – highly operable.

Economic: Developed with open-source tools; no licensing costs; affordable deployment in rural areas – economically viable.

Test Report

Method: Unit testing, integration testing, user acceptance testing (UAT).

Key Tests: User registration and login, form validations, prediction accuracy, role-based access, data visualization rendering.

Result: All test cases passed; minor issues in layout and validation were resolved successfully.

B. Gantt Chart

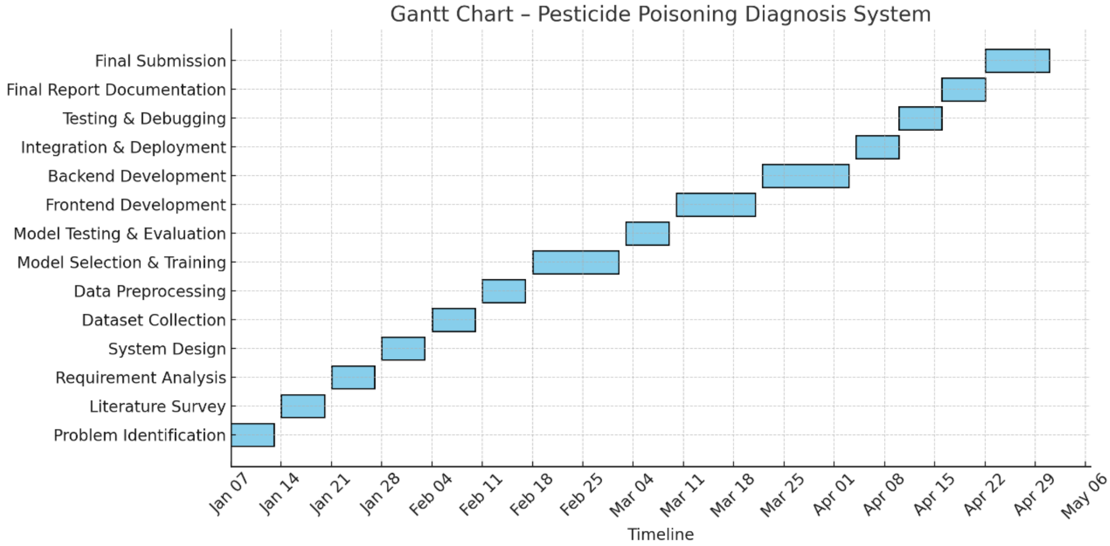

Fig 10.1 Gantt Chart

C. Ethical Considerations & Consent

Ethical Considerations:

- User data (rural workers' clinical and environmental data) stored securely; no external sharing.
- Users (rural workers and health professionals) informed about data usage for diagnosis and training purposes.
- Evaluation of diagnosis is unbiased and based on model predictions.
- Health professionals can review/override AI results.
- Future audit logs ensure transparency in the decision-making process.

Consent:

- Health professionals and users informed about AI support in diagnosis.
- Data retained only for the duration of the academic study; deletable on request.

D. Plagiarism Report

turnitin Page 2 of 89 - Integrity Overview Submission ID trn:oid:::26066:455255657

15% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

- 127 Not Cited or Quoted 15%
  Matches with neither in-text citation nor quotation marks
- 3 Missing Quotations 0%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

- 9% Internet sources
- 5% Publications
- 14% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

E. Source Code Repository

https://github.com/prakash3405/Pesticide-Poisoning-Diagnosis-System

F. Journal / Conference paper published on project

- **Title**: Pesticide Poisoning Diagnosis System
- **Published in**: International Journal for Research in Applied Science and Engineering Technology (IJRASET)
- **Volume**: 08
- **Issue**: 04
- **Publish Date**: April 29, 2025
- **ISSN**: 2321-9653
- **DOI**: 10.22214/ijraset.2025.70017
- **Authors**: Sama Prakash Reddy, Goli Karthik, Mandhula Praveen, Bh Bhujanga Reddy

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET70017, entitled

Pesticide Poisoning Diagnosis System

by

Sama Prakash Reddy

after review is found suitable and has been published in
Volume 13, Issue IV, April 2025
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET70017, entitled

Pesticide Poisoning Diagnosis System

by

Goli Karthik

after review is found suitable and has been published in
Volume 13, Issue IV, April 2025
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET70017, entitled

Pesticide Poisoning Diagnosis System

by

Mandhula Praveen

after review is found suitable and has been published in
Volume 13, Issue IV, April 2025
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET