IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

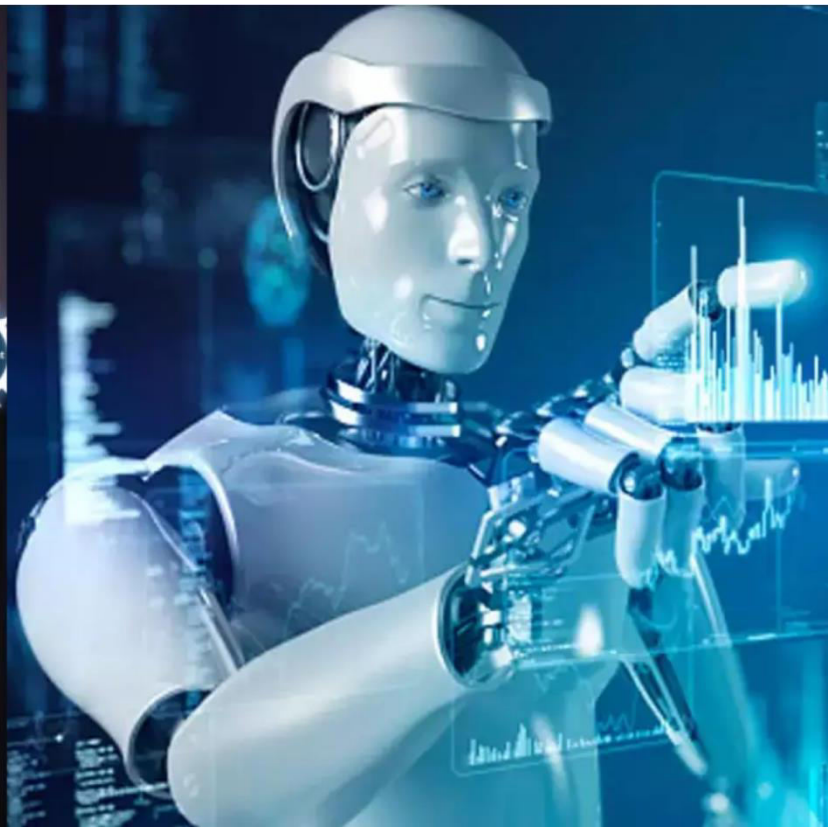
ISSN(O): 2319-8753
ISSN(P): 2347-6710



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 9.935

Volume 15, Issue 10, October 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

AI-Powered Plant Disease Detection and Monitoring System

A Web-Based Framework Combining Rule-Based Reminders, Machine Learning Growth Analysis and Optional IoT Sensing

Nandini S. Surankar[1], Sejal D. Deshmukh[1], Vedant Ravindra Nayse[1], Vaidahi P. Nimbhorkar[1], Prof. C. W. Rawarkar[2]

B. Tech Students, Group C15, Department of Computer Science and Engineering, Prof. Ram Meghe Institute of Technology and Research, Badnera, Amravati, Maharashtra, India[1]

Project Guide, Department of Computer Science and Engineering, Prof. Ram Meghe Institute of Technology and Research, Badnera, Amravati, Maharashtra, India[2]

ABSTRACT: Taking proper care of plants requires regular watering, fertilization, pruning, and observation. However, many plant owners find it difficult to follow care schedules, understand the needs of different plant species, or recognize early signs of unhealthy growth. To address these challenges, this paper presents the design of an AI-Powered Smart Plant Care Reminder and Growth Monitoring System. The proposed web application brings together plant profile management, care reminders, growth tracking, data analysis, and optional IoT-based monitoring. Users can register their plants and maintain records of plant height, leaf count, health condition, and general observations. When sensors are available, the system can collect information such as soil moisture, temperature, humidity, and light intensity. These readings can also be entered manually when sensor devices are not available. The system is developed using Python, Flask, and MySQL, with plant information displayed through a responsive dashboard. A rule-based mechanism provides care reminders and basic suggestions, while regression and tree-based machine learning models may be used to study growth patterns and identify unusual changes when sufficient training data is available. This paper explains the system architecture, development methodology, major modules, database structure, and analytical techniques. It also presents a preliminary evaluation using records maintained in the prototype database. The proposed system is intended to simplify everyday plant maintenance, encourage consistent care practices, and help users make better decisions when caring for indoor plants, home gardens, and small nurseries.

KEYWORDS: Artificial Intelligence, Machine Learning, Plant Care, Growth Monitoring, Internet of Things (IoT), Smart Agriculture, Reminder System, Python, MySQL, Web Application.

I. INTRODUCTION

Plants play an important role in maintaining environmental balance and making our surroundings healthier and more pleasant. Whether they are grown indoors, in home gardens, or on farms, plants need suitable amounts of water, sunlight, and nutrients, along with regular maintenance. However, providing the right care at the right time can be challenging for many plant owners. Problems such as overwatering, insufficient watering, inadequate sunlight, and delayed fertilization can affect plant health. In many cases, these issues are noticed only after changes in the plant become clearly visible.

Traditional plant care generally relies on personal observation, memory, and fixed schedules. Although these methods can work for basic needs, they may not reflect changes in environmental conditions or the different requirements of individual plant species. For example, plants may need different watering intervals depending on their type, growing conditions, and surrounding environment. Maintaining growth records manually can also be inconvenient. When information is recorded on paper or recalled from memory, comparing a plant's condition over time becomes more

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

difficult. As a result, early changes in growth may go unnoticed, and care decisions may not always be based on accurate records.

Artificial Intelligence (AI) and Machine Learning (ML) provide opportunities to analyse plant-related information, recognize patterns, and estimate possible changes in growth. These technologies are increasingly being explored in agriculture to support crop monitoring, plant health assessment, and data-based decision-making [1], [2]. By applying suitable analytical methods to recorded plant data, a smart plant care system can potentially offer more useful insights than a basic reminder application. When combined with regular monitoring and timely notifications, such a system can help users organize plant care more effectively.

Internet of Things (IoT) technology makes it possible to monitor plant-growing conditions by collecting environmental data through low-cost sensors [3]. Although these technologies are increasingly used in agriculture, many existing solutions focus mainly on large-scale farming. People who grow plants at home, maintain terrace gardens, or enjoy gardening as a hobby may not have access to a single system that combines care reminders, growth records, and data analysis.

To address this limitation, this paper proposes an AI-Powered Smart Plant Care Reminder and Growth Monitoring System. The system allows users to maintain individual plant profiles, schedule care reminders, record growth observations, and store environmental readings in a MySQL database. It also analyses the available information to identify growth patterns and highlight unusual changes. Users can begin by entering information manually, without needing to purchase sensors or other hardware. If required, the system can be extended later to collect environmental data through IoT devices.

A. Need for the Project

The project is designed to address six common difficulties faced by plant owners. First, people may forget to water their plants or apply fertilizer at the right time. Second, different plant species have different care requirements, making a single routine unsuitable for every plant. Third, manual observation makes it difficult to maintain and compare growth records over long periods. Fourth, changes in environmental conditions can affect plant health but may not be noticed immediately. Fifth, managing several plants individually can become time-consuming. Finally, a reminder system can help users follow their care schedules more regularly and maintain better records of plant maintenance.

B. Scope of the Project

The proposed system is intended for people who maintain indoor plants, home gardens, terrace gardens, college gardens, small nurseries, and other small-scale growing spaces. The initial version focuses on indoor plants and commonly grown garden plants. It supports plant information management, care reminders, growth tracking, and the recording of environmental observations. Integration with sensors and the use of the system in large-scale agricultural environments may be explored in future development.

C. Contributions

The main contributions of this work are as follows:

1. **Integrated system design:** A layered architecture brings plant management, care reminders, growth tracking, and data analysis together in one application.
2. **Flexible data collection:** The system supports manual data entry and allows optional integration with IoT sensors.
3. **Combined analytical approach:** Rule-based logic provides immediate reminders and basic warnings, while machine learning models can be explored for growth estimation when sufficient data becomes available.
4. **Structured database design:** A relational database is used to organize plant profiles, care schedules, growth observations, and environmental records.
5. **Prototype evaluation:** A preliminary evaluation uses records stored in the implemented prototype to demonstrate the system's main functions and data-handling process.

II. RELATED WORK

Machine learning has been explored in several agricultural applications, including crop management, water management, soil analysis, and livestock monitoring. Liakos et al. reviewed the use of machine learning techniques in agriculture and discussed their potential for improving agricultural practices [1]. Similarly, Kamilaris and Prenafeta-

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

Boldú examined the applications of deep learning in agriculture and emphasized the importance of labelled datasets for developing reliable models [2]. These studies indicate that the usefulness of machine learning depends largely on the availability, quality, and diversity of the data used for training and testing.

The Internet of Things (IoT) is another important area of research in smart agriculture. Elijah et al. reviewed the role of IoT and data analytics in agricultural applications, including data collection, communication, and analysis [4]. Ayaz et al. also discussed the sensors, communication technologies, and applications used in IoT-based smart agriculture systems [5]. In many such systems, low-cost microcontrollers are connected to sensors that measure soil moisture, temperature, humidity, and light intensity. These measurements help monitor growing conditions and support better management decisions. However, much of this research focuses on agricultural fields and greenhouses, rather than the everyday plant care needs of individual users.

Image-based plant disease detection has also been studied using deep learning. Mohanty et al. used deep convolutional neural networks to classify plant diseases from leaf images in a public dataset [6]. Their work demonstrated the potential of image-based classification while also highlighting the challenges of applying trained models to images captured under different conditions. Ferentinos investigated deep learning models for identifying and diagnosing plant diseases using images of plant leaves [7]. Although these approaches show promising results, they generally require substantial collections of labelled images. In addition, disease classification alone does not provide features such as watering reminders, fertilizer scheduling, or long-term growth tracking.

A common observation across these studies is that the reliability of a system depends on the quality of the information it receives. Sensor readings may require calibration and validation, machine learning models need representative training data, and image classification systems depend on suitable labelled examples. Considering these limitations, the proposed plant care system is designed to validate recorded readings and maintain historical data for future analysis. Basic rule-based methods handle reminders and warnings, while machine learning can be introduced when enough reliable data has been collected.

The proposed system differs from many existing approaches in its intended users and combination of features. Rather than focusing mainly on large-scale agricultural monitoring or plant disease classification, it is designed for individuals who want to manage their plants through a single web application. Plant profiles, care reminders, and growth records form the core of the system, while machine learning and IoT-based monitoring remain optional extensions. The system also recognizes that its analytical results have limitations. Therefore, uncertain findings are presented as warnings or suggestions rather than as confirmed plant disease diagnoses.

III. PROBLEM STATEMENT AND OBJECTIVES

A. Problem Statement

Maintaining healthy plants can be challenging for many plant owners due to irregular watering, a lack of knowledge about the specific needs of different plant species, and limited opportunities to monitor growth regularly. Traditional plant care methods usually depend on fixed schedules and personal observation. However, these methods may not always reflect the changing needs or actual condition of a plant. In addition, keeping track of growth records and managing the care routines of multiple plants can become time-consuming when done manually.

To address these issues, there is a need for a simple and user-friendly plant care system that brings essential plant maintenance activities together in one place. Such a system should allow users to maintain plant profiles, receive timely care reminders, record growth-related information, and monitor environmental conditions. By using data analysis and suitable machine learning techniques, the system could also help users understand growth patterns and make more informed plant care decisions.

B. Objectives

The main objectives of the proposed project are listed below:

1. **Plant Management:** To allow users to create, update, and manage profiles for multiple plants.
2. **Smart Reminders:** To provide timely reminders for watering, fertilization, pruning, and other routine care activities.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

3. **Growth Monitoring:** To maintain records of plant height, leaf count, photographs, and observations so that changes can be tracked over time.
4. **Environmental Monitoring:** To collect or manually enter environmental readings, including soil moisture, temperature, humidity, and light intensity.
5. **AI-Based Analysis:** To explore suitable machine learning techniques for analysing recorded data and estimating plant growth patterns.
6. **Personalized Recommendations:** To provide care suggestions based on plant type, recorded observations, and available environmental information.
7. **Data Management:** To store plant profiles, care schedules, reminders, and historical records in a structured database.
8. **User-Friendly Interface:** To develop a dashboard that displays plant information, growth trends, and upcoming care activities in an easy-to-understand format.
9. **Improved Plant Care:** To help users follow regular care routines, monitor plant conditions, and make better-informed decisions about plant maintenance.

IV. SYSTEM REQUIREMENTS

A. Functional Requirements

The system is expected to provide the following functions:

- Allow users to register, log in, and manage their personal profiles.
- Enable users to create, update, and delete plant profiles containing details such as plant species, location, and care schedules.
- Store and display environmental readings and growth observations for each plant.
- Generate and display reminders for watering, fertilization, pruning, and regular growth recording, while allowing users to track their completion.
- Analyse stored records to display growth trends and identify unusual changes in the available data.
- Provide a dashboard that brings together plant details, charts, upcoming reminders, and care recommendations.

B. Non-Functional Requirements

The system should also meet the following quality requirements:

- **Usability:** The interface should be simple to navigate and should work properly on both desktop and mobile browsers.
- **Reliability:** Input values should be checked before they are stored to prevent invalid readings from affecting historical records.
- **Security:** Passwords should be stored in hashed form, user sessions should be authenticated, and users should only be able to access records they are authorized to view.
- **Maintainability and Extensibility:** The application should be divided into separate modules so that individual components can be updated and new sensors or analytical models can be added with minimal changes to the existing system.
- **Scalability:** The database structure should support multiple plants for each user and allow a growing number of readings to be stored for every plant.

C. Typical Usage Scenario

Consider a user who registers on the application and adds a plant by entering its species and watering interval. Based on the selected schedule, the system creates a reminder for the next watering activity. Once the user waters the plant, they can mark the reminder as completed, and the system schedules the next reminder according to the scheduling rule described in Section XI. The user can also record the plant's height and leaf count every week. These observations are stored in the database and used to update the growth chart on the dashboard. If a new measurement differs unexpectedly from previous records, such as a sudden decrease in the recorded plant height, the system displays a warning asking the user to check the plant and verify the observation. This process helps the user maintain regular records and notice possible changes in plant growth.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

V. PROPOSED METHODOLOGY

The proposed system follows a step-by-step process that includes data collection, preprocessing, analysis, database storage, and notification generation. The methodology is divided into nine steps, each describing a major activity involved in the operation of the plant care system.

Step 1: User Registration
The process begins when a user creates an account by providing basic details, including their name, email address, and password. After successful authentication, the user can access a personal dashboard to manage plant information and care activities.

Step 2: Plant Registration
After logging in, the user can add a plant by entering details such as its name, species, category, planting date, location, watering frequency, and fertilizer schedule. The user can also upload a photograph if required. These details are stored in the database and used to manage the plant's care routine.

Step 3: Data Collection
The system supports two ways of collecting plant-related information.

- **Method A: Sensor-Based Data Collection:** Sensors connected to an ESP32 or Arduino-based device can measure soil moisture, temperature, air humidity, and light intensity. The collected readings can then be transmitted to the application for storage and analysis.
- **Method B: Manual Data Entry:** Users can enter environmental measurements and growth observations directly into the application. This option allows the system to operate without additional hardware and is suitable for the initial implementation.

Step 4: Data Preprocessing
Before analysis, the collected information is checked and prepared using suitable data-processing techniques and libraries [8]. This stage may include identifying missing values, removing duplicate records, rejecting invalid sensor readings, and normalizing numerical values when required. Min-max normalization can be applied to numerical features to bring their values within a common range. The normalization formula is:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

Here, x represents the original value, x_{min} is the minimum value, and x_{max} is the maximum value in the selected range. The resulting x′ is the normalized value. If the minimum and maximum values are equal, a separate rule is needed to avoid division by zero.

Step 5: Machine Learning Analysis
The analysis component uses available growth records and environmental information to study changes in plant conditions. Depending on the amount and quality of the collected data, suitable models may be developed to estimate growth trends, explore watering requirements, classify plant-health conditions, or identify deviations from previously recorded patterns. Reliable training and evaluation data are necessary before the performance of these models can be assessed. If the system relies only on predefined conditions and thresholds, its outputs are treated as rule-based recommendations rather than predictions from a trained machine learning model.

Step 6: Reminder Generation
The reminder module manages scheduled activities such as watering, fertilizer application, pruning, and growth recording. Reminders are generated according to the care schedule assigned to each plant. Where reliable environmental measurements are available, these readings may also be considered when adjusting recommendations, provided that the system implements suitable adjustment rules.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

Step 7: Database Storage
MySQL is used to organize and store the system's information, including user profiles, plant details, care schedules, environmental measurements, growth observations, and notification history. Maintaining these records makes it easier to retrieve previous observations and compare plant growth over time.

Step 8: Dashboard and Visualization
The dashboard presents the collected information in an organized format. It can display registered plants, recorded health conditions, environmental measurements, growth charts, upcoming reminders, and previous observations. This allows users to review their plants' progress and identify care activities that need attention.

Step 9: User Feedback
Users can mark completed care activities and add new observations about their plants. These updates are saved in the database and become part of the plant's historical record. As more reliable records are collected, they may support improvements to future recommendations and provide useful data for training and evaluating machine learning models.

VI. SYSTEM ARCHITECTURE

The overall architecture of the proposed system is illustrated in Fig. 1. Users access the application through a web browser on a computer or mobile phone. The presentation layer provides the main interface, including plant information, input forms, growth charts, and care reminders. It allows users to add plant details, update observations, and view information related to their plants in an organized manner.

The application layer is developed using Python and Flask [9]. It manages user accounts, plant information, and care schedules while handling the main operations requested through the web interface.

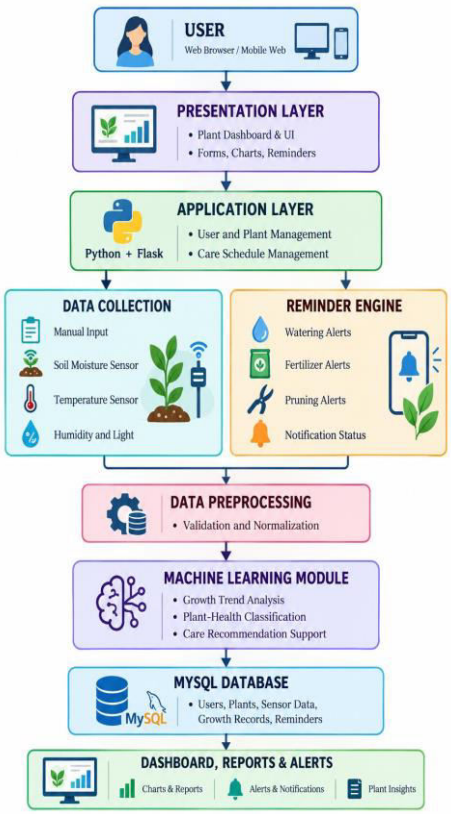


Fig.1. System architecture of the proposed plant care and growth monitoring system

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

The application layer divides the process into two main components: data collection and reminder management. The data collection component accepts information entered manually by users and, when sensors are available, receives readings such as soil moisture, temperature, humidity, and light intensity. At the same time, the reminder engine manages watering, fertilization, and pruning schedules while keeping track of notification status.

The information from both components is sent to the data preprocessing stage, where the values are checked for validity and normalized when necessary. The processed data is then used for further analysis and storage. The machine learning module supports growth trend analysis, plant-health classification, and care recommendations, depending on the available data and implemented models. The MySQL database maintains plant records, environmental readings, growth observations, and reminder information. Finally, the dashboard presents charts, reports, notifications, and plant-related insights in a format that users can easily understand.

One important feature of the architecture is its ability to work without physical sensors during the initial stage. Users can enter the required information manually and connect IoT hardware later if needed. Both manual entries and sensor readings can be handled through a common data structure, allowing the reminder and analysis components to continue working without major changes to their design.

VII. SYSTEM WORKFLOW

The overall workflow of the proposed system is illustrated in Fig. 2. It begins when a user registers or logs in to the application and enters the details of their plants. The system then collects environmental measurements and growth observations, either through manual entry or connected sensors. Before the information is stored, the collected values are checked and processed to ensure they are suitable for use.

Once the records are available, the system uses the implemented rules and analytical methods to generate relevant care suggestions and schedule reminders. The dashboard displays plant details, growth information, and upcoming activities. As users complete watering, fertilization, pruning, or other care tasks, they can update the corresponding records in the application. These updates are saved in the database and become part of the plant's history, helping users review changes and manage future care activities more effectively.

At the end of each cycle, the system checks whether additional plants need to be registered or new data is available. If new information is provided, the workflow returns to the data collection stage, where the records are updated and processed again. If no further updates are required, the current cycle ends. This continuous process helps keep plant records up to date and supports regular monitoring of plant growth and care activities.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

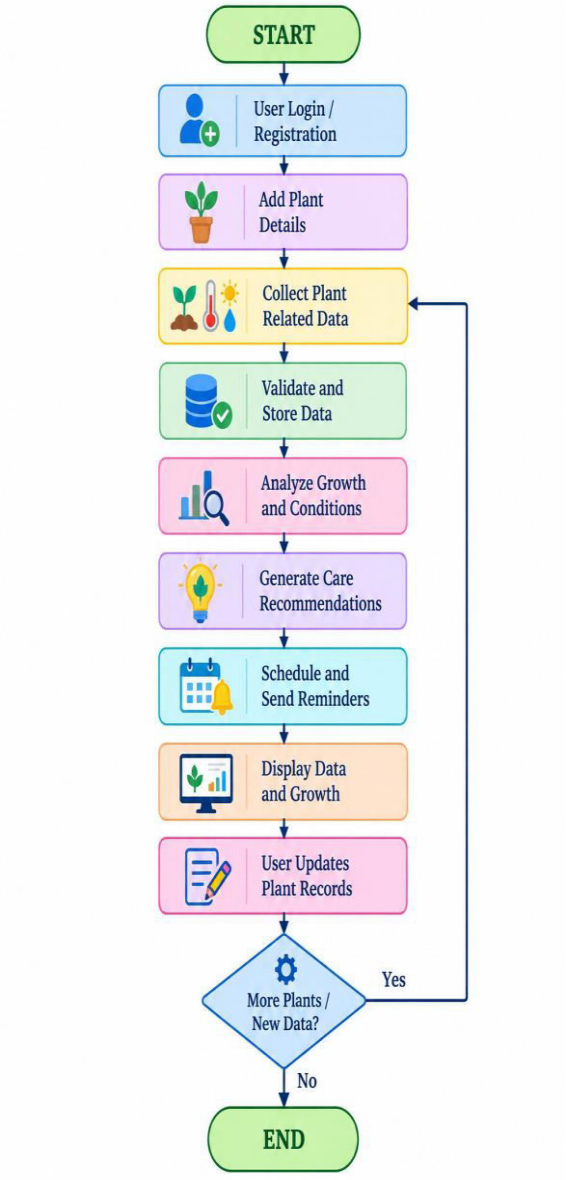


Fig.2. Workflow of the proposed plant care and growth monitoring system.

This continuous cycle ensures that each new observation is added to the plant's historical records. As the records grow over time, they provide updated information for growth charts, trend analysis, and care reminders, helping users monitor plant development more consistently.

VIII. DETAILED SYSTEM MODULES

The proposed system is organized into eight main modules, each responsible for a specific set of functions. Together, these modules handle plant information, data collection, reminders, growth monitoring, and the presentation of results. Their main functions are summarized in Table I.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

TABLE I. SYSTEM MODULES

Module	Main Functions	Technology Used
1. User Management	Handles user registration, login, logout, profile management, and access to personal records.	Python, Flask, MySQL
2. Plant Management	Allows users to add, update, or delete plant details, including species, location, care schedules, and photographs.	Python, Flask, MySQL
3. Plant Monitoring	Maintains environmental readings, growth observations, and historical records. Sensor-based monitoring can be added when hardware is available.	Python, MySQL, optional ESP32/Arduino
4. Care Reminder	Manages watering, fertilization, and pruning schedules. Users can view upcoming tasks, mark them as completed, and check previous activities.	Python, Flask, scheduling functions
5. Growth Prediction	Examines plant height and leaf-count records to study growth patterns and identify unexpected changes.	scikit-learn, Pandas, NumPy
6. Plant Health Analysis	Checks recorded values, highlights unusual growth patterns, and provides warnings about possible plant care problems.	Python, rule-based logic, optional ML
7. Dashboard	Displays plant details, environmental readings, growth charts, upcoming reminders, and care suggestions in one place.	HTML, CSS, JavaScript, Chart.js
8. Notification	Manages in-app and optional email reminders, tracks their status, and avoids unnecessary repeat alerts for completed tasks.	Python, Flask, MySQL

User and Plant Management

The user and plant management module allows each registered user to manage their own plant records. Users can view and update only the information associated with their accounts. Each plant profile contains important details, such as the plant species, planting date, location, and care schedule. These details are used by other modules to manage reminders and analyse plant growth.

Monitoring and Reminders

The monitoring module records plant measurements and observations along with the date and time of each entry. The reminder module uses the stored schedules and timestamps to identify upcoming care activities. It displays reminders within the application and can also send email notifications if this feature is enabled. The system keeps track of notification status to reduce repeated alerts for tasks that have already been completed.

Growth Prediction and Plant Health Analysis

The growth monitoring module examines previous plant height measurements and changes in leaf count to understand how a plant develops over time. The plant health analysis module checks recorded values against predefined ranges and highlights unusual observations that may require attention. However, environmental readings alone cannot confirm every plant disease or health problem. For this reason, the system presents uncertain findings as warnings or care suggestions rather than treating them as definite diagnoses.

Reminder Lifecycle

Each reminder passes through four states: pending, due, notified, and completed. When a scheduled activity becomes due, the system generates a notification. If the task remains incomplete, another notification may be sent after a specified interval. Once the user marks the activity as completed, further notifications for that occurrence are stopped, and the next reminder is scheduled based on the completion time. This approach helps the schedule reflect the user's actual care activities rather than relying only on the original plan.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

A background task checks reminders at regular intervals to identify activities that have become due. This task may run through an application scheduler or an operating-system scheduling service. It compares the scheduled times with the current time and creates the necessary notifications. To maintain consistent scheduling, reminder timestamps are stored using a common reference time zone and converted to the appropriate local time for display. This helps prevent reminders from shifting unexpectedly when users travel between time zones.

Dashboard Design

The dashboard is designed to give users a quick overview of their plants and the care activities that need attention. At the top, summary cards display the total number of registered plants, tasks scheduled for the current day, and plants with warning messages. Below these cards, a plant list shows the latest recorded health status of each plant.

Each plant also has a dedicated page where users can review growth charts, environmental readings, previous records, and diary notes. These features help users compare observations over time and keep track of changes in plant condition. Charts can be displayed using Chart.js in the web browser or generated on the server using Matplotlib [10], depending on the implementation.

IX. TECHNOLOGY STACK

The technology stack is selected according to the system's requirements and whether the application operates independently or includes sensor-based monitoring. The main technologies and their roles are summarized in Table II. Python and Flask [9] are used to develop the application logic and manage communication between the user interface and the backend. MySQL [11] stores user details, plant profiles, care schedules, and historical records. Pandas [8] and NumPy [12] support data processing and numerical calculations, while scikit-learn [13] provides machine learning tools for analysing plant growth data when sufficient records are available. For visualizing the results, the system can use Matplotlib [10] or Chart.js to display growth trends and other relevant information.

TABLE.II. TECHNOLOGY STACK

Layer	Technology	Purpose
Programming Language	Python	Used as the main language for developing the application and its logic.
Backend	Flask	Handles application operations and communication between the frontend and database.
Frontend	HTML, CSS, JavaScript, Bootstrap	Creates the user interface, adds interactive features, and supports a responsive layout.
Database	MySQL	Stores user profiles, plant details, care schedules, and historical records.
Data Processing and Machine Learning	Pandas, NumPy, scikit-learn	Supports data processing, numerical calculations, and machine learning analysis.
Data Visualization	Matplotlib or Chart.js	Displays growth charts, measurement trends, and other plant-related information.
Optional Sensors	ESP32 or Arduino with soil moisture, temperature, humidity, and light sensors	Collects environmental measurements for plant monitoring.
Version Control	Git and GitHub	Tracks code changes and supports project development and collaboration.

X. SENSOR INTERFACE AND DATA VALIDATION

When sensor-based monitoring is enabled, an ESP32 or Arduino-based device collects environmental readings using a soil moisture sensor, a temperature and humidity sensor, and a light sensor. The device sends the collected information to the Flask application through HTTP in a structured message containing the plant identifier and measured values. If

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

the device does not have its own clock, the server records the time at which the data is received. Since houseplants generally show gradual changes in their growing conditions, readings can be collected at intervals ranging from a few minutes to an hour, depending on the monitoring requirements. Less frequent data transmission can also help reduce power consumption in battery-operated devices.

Soil moisture sensors produce raw readings that may vary depending on the sensor model and the type of soil being measured. Therefore, each sensor needs to be calibrated before its readings can be converted into a moisture percentage. Calibration involves recording the dry reference reading (R_{dry}) and the wet reference reading (R_{wet}). A new sensor reading (R) is then converted using the following formula:

$$M = 100 \times \frac{R_{\text{dry}} - R}{R_{\text{dry}} - R_{\text{wet}}} \tag{2}$$

Here, M represents the estimated soil moisture percentage. The calculated value is restricted to the range of 0 to 100. However, the percentage is only an estimate based on the sensor's calibration, and the same reading may indicate different moisture conditions depending on the soil and plant species. For this reason, the system uses plant-specific thresholds when interpreting moisture readings.

All incoming measurements, whether entered manually or collected through sensors, pass through the same validation process before they are stored in the database. The main validation rules are summarized in Table III. Readings outside the accepted physical ranges are rejected and recorded for review. Missing values are identified rather than being filled in automatically without justification. Exact duplicate records are removed to avoid unnecessary repetition. When a new reading shows a sudden change that exceeds a predefined plausible rate, the system flags it for confirmation instead of immediately treating it as a genuine change in plant condition. This approach helps maintain more consistent records for monitoring and future analysis.

A sudden change in a single reading may be caused by a sensor fault rather than an actual change in the plant's condition. Therefore, such readings are flagged for verification before they are used for further analysis.

TABLE.III. VALIDATION RANGES

Quantity	Unit	Accepted Range	Typical Source
Soil Moisture	%	0–100	Capacitive soil moisture sensor
Temperature	°C	−10 to 60	DHT-type sensor
Humidity	% RH	0–100	DHT-type sensor
Light Intensity	Lux	0–100,000	LDR or digital light sensor
Plant Height	cm	Greater than 0 and up to 500	Manual entry
Leaf Count	Count	Integer greater than or equal to 0	Manual entry

The ranges listed in Table III are configurable default limits used to identify invalid or physically implausible readings. They are not intended to determine whether a plant is healthy or unhealthy. Plant-health assessments require separate thresholds that consider the needs of the particular plant species and its growing conditions.

XI. MACHINE LEARNING AND ANALYTICAL METHODS

Machine learning is one of the important components of the proposed system. However, its usefulness depends on the availability of reliable and sufficient data. This section explains the methods used to study plant growth and generate care reminders, while distinguishing between predefined rules and models trained using data.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

A. Growth Rate and Trend Estimation

Plant growth can be monitored by recording its height at different points in time. Let h_k represent the plant height recorded at time t_k. The growth rate between two consecutive measurements can be calculated as follows:

$$GR_k = \frac{h_k - h_{k-1}}{t_k - t_{k-1}} \tag{3}$$

This calculation shows the average change in plant height between two observations. When sufficient historical records are available, linear regression can be used to estimate the plant's growth trend [14]. The relationship between time and estimated height is expressed as:

$$\hat{h}(t) = \beta_0 + \beta_1 t \tag{4}$$

Here, $\hat{h}(t)$ represents the estimated plant height at time t, while β_0 and β_1 are the regression coefficients. They are calculated using the following equations
:

$$\beta_1 = \frac{\sum(t_i - \mu_t)(h_i - \mu_h)}{\sum(t_i - \mu_t)^2} \tag{5}$$

$$\beta_0 = \mu_h - \beta_1 \mu_t \tag{6}$$

In these equations, μ_t and μ_h represent the mean observation time and mean plant height, respectively. Linear regression provides a simple way to understand the overall growth pattern because the slope, β_1, represents the estimated average change in height per unit of time. Additional information, such as plant species and environmental measurements, can be included in the model when enough suitable data becomes available.

B. Rule-Based Reminder Logic

During the initial stage, the system generates reminders using predefined rules based on plant-specific schedules and, when available, soil moisture readings. For tasks that follow a fixed schedule, the next due time is calculated from the last completion time and the selected care interval:

$$t_{\text{due}} = t_{\text{last}} + f_s \tag{7}$$

Here, t_{last} represents the time when the task was last completed, and f_s is the care interval selected for the particular plant.

When a soil moisture sensor is available, the system can also provide a watering suggestion based on the current moisture level and the time elapsed since the last watering. The decision is expressed as:

$$W = \begin{cases} 1, & M < \theta_s \text{ and } \Delta_w \geq \Delta_{\min} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

In this equation, W=1 indicates that the watering conditions are met, while W=0 means that the defined conditions are not met. M represents the measured soil moisture, θ_s is the moisture threshold for the plant species, Δ_w is the time elapsed since the previous watering, and Δ_{min} is the minimum waiting period between watering activities.

A low moisture reading does not necessarily mean that every plant needs immediate watering. The decision also depends on the plant species, soil type, sensor calibration, and growing environment. Therefore, the system uses configurable thresholds and presents the result as a watering suggestion rather than an automatic diagnosis. The different reminder types and their triggering conditions are summarized in Table IV.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

TABLE.IV. REMINDER RULES

Reminder Type	Trigger Condition	Required Input
Watering	The scheduled watering time is reached, or Equation (8) indicates that watering may be needed when a sensor is available.	Watering interval, soil moisture, last watering time
Fertilization	The specified fertilizer interval has elapsed, as calculated using Equation (7).	Fertilizer schedule, last fertilization time
Pruning	The scheduled pruning interval has elapsed.	Pruning schedule, last pruning time
Growth Recording	No new plant height or leaf-count record has been added within the specified period.	Time of the last growth record

C. Unusual Growth and Condition Detection

The system uses two main checks to identify unusual changes in plant growth. The first check looks for a negative growth rate ($GR_k<0$), which indicates that the newly recorded plant height is lower than the previous measurement. Such a change is flagged for review because it may result from pruning, an incorrect measurement, or another issue that needs attention.

The second check uses the z-score to compare the latest growth rate with the average growth rate recorded for that plant. When enough historical observations are available, the z-score is calculated using the following formula:

$$z_k = \frac{GR_k - \mu_{GR}}{\sigma_{GR}} \tag{9}$$

Here, GR_k is the current growth rate, μ_{GR} is the mean growth rate, and σ_{GR} is the standard deviation of the recorded growth rates. If the absolute z-score exceeds 2, the observation is marked for further review. This calculation requires a non-zero standard deviation and enough historical data to establish a meaningful comparison.

The system also checks environmental readings against predefined acceptable ranges and flags values that fall outside them. These warnings help users identify measurements that may need attention. However, an unusual reading does not necessarily indicate a plant health problem. It could result from a measurement error, pruning, or a genuine change in growing conditions. Therefore, the system presents these results as warnings for the user to investigate rather than as confirmed diagnoses.

D. Feature Set and Candidate Models

The features considered for machine learning are summarized in Table V. Before training, categorical information, such as plant species, can be converted into numerical form using one-hot encoding. Numerical features can be normalized using Equation (1) where appropriate.

For growth prediction, the target variable is the plant's next recorded height. For plant-health classification, the target is the health status stored in the database. Existing labels, such as Good and Excellent, can provide an initial basis for analysis. However, a reliable classifier also requires sufficient examples from other health categories, including plants in poor condition. Without representative training data across these categories, the model may not learn to distinguish healthy plants from unhealthy ones accurately.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

TABLE V. FEATURE SET

Feature	Data Source	Purpose
Species and Category	Plant profile	Helps group plants and convert categorical information into a suitable format for analysis.
Days Since Planting	Planting date	Supports the study of plant growth over time.
Soil Moisture, Temperature, Humidity, and Light	Sensors or manual entry	Provides environmental information for growth and plant-health analysis.
Watering and Fertilizer Intervals	Care schedules	Helps monitor care routines and check whether scheduled activities are followed.
Plant Height and Leaf Count	Growth records	Used to estimate growth trends and support regression analysis.
Health Status	Plant growth records	Provides the target label for plant-health classification.

The algorithms considered for the proposed system are compared in Table VI. The final choice depends on the amount and quality of available data, along with the performance observed during model evaluation using scikit-learn [13].

TABLE.VI. CANDIDATE ALGORITHMS

Algorithm	Intended Task	Strengths	Limitations
Linear Regression	Estimating plant growth trends	Simple to implement, easy to interpret, and suitable for exploring basic trends.	Assumes a linear relationship and may not capture complex growth patterns.
Decision Tree [15]	Classifying plant-health categories	Provides understandable decision rules and can model nonlinear relationships.	May overfit when trained on small or unrepresentative datasets.
Random Forest [16]	Analysing complex plant-related data	Can capture nonlinear relationships and combine predictions from multiple decision trees.	Less straightforward to interpret and may require more training data and computational resources.

E. Evaluation Protocol

The performance of the regression models is assessed using records that are kept separate from the training data. A time-based split is used so that earlier observations are used for training and later observations are reserved for testing. This helps prevent future information from influencing the model during training.

Two common metrics are used to measure regression performance: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Their formulas are given below:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{10}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{11}$$

Here, n is the number of test observations, y_i is the actual value, and $\hat{y}_i$ is the predicted value. Lower MAE and RMSE values generally indicate that the predicted measurements are closer to the actual observations.

Classification models are evaluated using accuracy, precision, recall, and the F1-score. When the dataset is small, k-fold cross-validation may also be used to examine how consistently a model performs across different subsets of the data. The evaluation method should be selected according to the size and structure of the available dataset.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

In future development, the system could be extended to analyse plant leaf images and classify them as healthy or potentially unhealthy using a trained convolutional neural network [6], [7]. This extension would require a suitable labelled image dataset, image preprocessing, model training, and performance testing. Image-based disease classification is not included in the current prototype.

XII. DATABASE DESIGN AND IMPLEMENTATION

The prototype uses MySQL [11] to store the information required for plant management, growth monitoring, and care reminders. The database schema is named plant_disease_db. As illustrated in Fig. 3 and summarized in Table VII, the proposed database structure consists of seven tables, each serving a specific purpose.

The growth_tracking table stores plant growth information, including the record ID, plant ID, height, leaf count, health status, and notes. The plant_id field connects each growth record to its corresponding entry in the plants table. Each plant is associated with its owner through the user information stored in the database. This relationship helps organize records by plant and user and makes it easier to retrieve historical growth information when needed.

TABLE.VII. DATABASE TABLES

Table Name	Purpose
users	Stores registered user account details.
plants	Maintains plant profiles, including species, location, and care schedules.
growth_tracking	Records plant height, leaf count, health status, and observations.
care_reminders	Stores scheduled watering, fertilization, pruning, and other care tasks.
notifications	Maintains notification records and their current status.
plant_diary	Stores users' observations and notes about individual plants.
plant_disease_analysis	Stores the results of plant-health analysis.

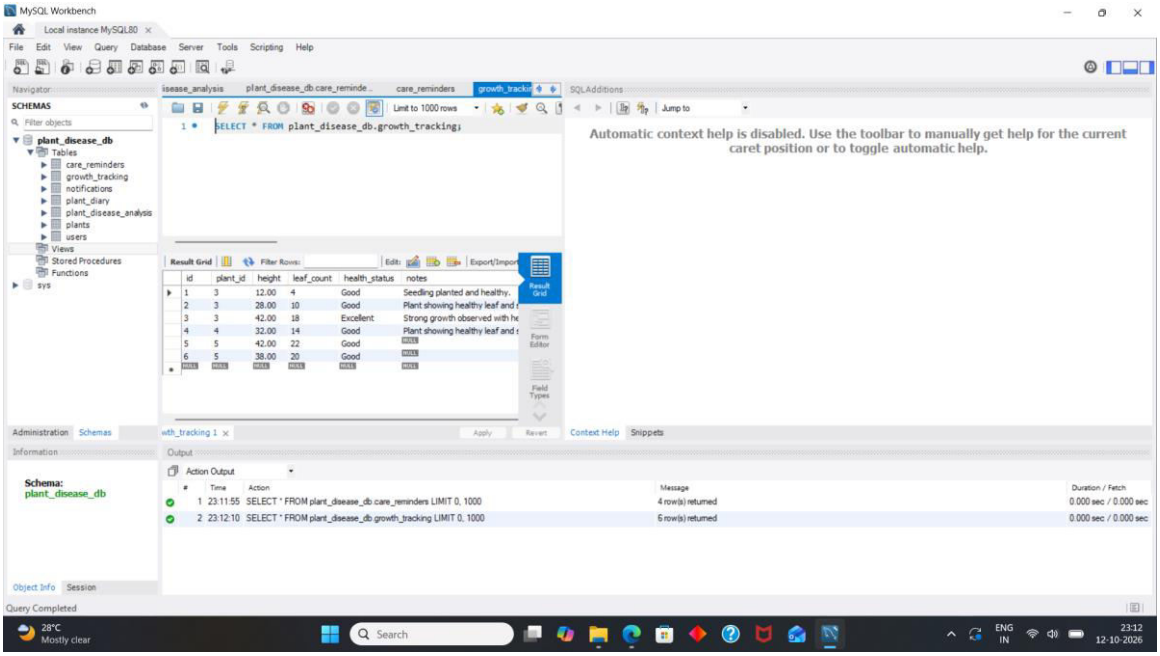


Fig.3. MySQL Workbench view of the schema and the growth_tracking records

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

Keeping reminders and notifications in separate database tables helps the system manage scheduled tasks and their delivery records independently. The care_reminders table stores information about the tasks that need to be completed, while the notifications table maintains their notification history and status. This separation helps prevent unnecessary repeated alerts for tasks that have already been completed. The database structure also allows the dashboard to retrieve a plant's historical records by joining related tables through the plant_id field.

Security and Privacy: User passwords are stored as salted hashes rather than plain text. Access to the application is managed through authenticated sessions, and queries involving plant information are restricted to records belonging to the logged-in user. These measures help protect personal information and prevent unauthorized access to plant records. Access to plant records is restricted using the logged-in user's identifier, ensuring that each user can view and manage only their own information. Parameterized queries are used to reduce the risk of SQL injection attacks. Uploaded plant photographs are checked for file type and size before they are accepted. Email addresses are used to send notifications only when the user has enabled the email notification feature.

XIII. PRELIMINARY RESULTS AND DISCUSSION

The prototype was tested using sample data entered through the application. At the time of evaluation, the growth_tracking table contained six records covering three plants, while the care_reminders table contained four records, as shown in Fig. 3. The growth information is summarized in Table VIII, with plant height expressed in centimetres.

The sample records show how the system stores plant measurements, leaf counts, health-status labels, and observation notes. For example, the recorded height of Plant 3 increases from 12 cm to 42 cm across three entries, while its leaf count rises from 4 to 18. These entries demonstrate how the application can maintain a history of plant growth and make changes easier to review. However, the records are limited and should not be treated as evidence of a general growth pattern.

The available dataset is also too small to train and properly evaluate a reliable machine learning model. Therefore, the results presented in this section focus on demonstrating the data storage and analysis workflow rather than claiming predictive accuracy. A larger dataset collected over a longer period would be required to assess the performance and practical usefulness of the proposed analytical methods.

TABLE.VIII. SAMPLE GROWTH RECORDS

ID	Plant ID	Height (cm)	Leaves	Health Status	Observation Note
1	3	12.00	4	Good	Seedling planted and healthy.
2	3	28.00	10	Good	Plant showing healthy leaf and stem development.
3	3	42.00	18	Excellent	Strong growth observed with healthy development.
4	4	32.00	14	Good	Plant showing healthy leaf and stem development.
5	5	42.00	22	Good	No observation note recorded.
6	5	38.00	20	Good	No observation note recorded.

Growth Tracking

Plant 3 has three growth records, making it possible to compare its measurements over time. Its height increased from 12 cm to 28 cm and then to 42 cm, representing increases of 16 cm and 14 cm between consecutive observations. The leaf count also increased from 4 to 10 and then to 18. In addition, the recorded health status changed from Good to Excellent, as illustrated in Fig. 4. These records demonstrate how the system can display changes in plant growth and maintain a history of observations. Plant 4 has only one recorded measurement, so a growth trend cannot yet be calculated. In this case, the dashboard displays the available information about the plant's current recorded condition.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

Trend Estimation (Illustration)

To demonstrate the working of the regression method, the three height measurements for Plant 3 can be treated as equally spaced observations numbered 1, 2, and 3. Using Equations (5) and (6), the estimated regression coefficients are β_1=15 cm per observation

n and β_0=−2.67 cm. Based on this fitted line, the estimated height for the fourth observation is approximately 57.3 cm. This calculation illustrates how the system could estimate a future growth value using previous measurements. However, three observations are not sufficient to establish the reliability of a prediction model. Plant growth can vary with species, environmental conditions, and other factors, and it does not necessarily follow a linear pattern indefinitely. Therefore, the system should require a sufficient number of reliable records before displaying growth estimates and should present them as estimates rather than guaranteed outcomes.

Unusual-Change Detection

Plant 5 shows a decrease in recorded height from 42 cm to 38 cm, along with a reduction in leaf count from 22 to 20. The negative growth-rate rule identifies this change and flags it for review. However, the available records do not establish the reason for the decrease. It may have resulted from pruning, a measurement error, or an actual change in the plant's condition. For this reason, the system displays a warning asking the user to check the plant and verify the recorded measurements instead of automatically classifying the change as a health problem.

The system responds to this type of change by displaying a warning that asks the user to inspect the plant and add an observation note. It does not provide a definite diagnosis based on the available measurements. The absence of notes for both of Plant 5's records also highlights the importance of maintaining a plant diary. Adding details about watering, pruning, environmental changes, or other observations can help users better understand changes in plant growth.

Visual Comparison

Fig. 5 illustrates how the plant profile's photograph feature can be used to store images showing different leaf conditions. In Fig. 5(a), the leaf has brown, dry margins while its centre remains green. Such an appearance may indicate a possible plant health issue and could be flagged for further inspection by the user or analysed by an image-classification model in future development. In contrast, Fig. 5(b) shows a leaf with a more uniformly green appearance. These images are illustrative examples of different leaf conditions and are not controlled before-and-after photographs of the same leaf. Brown or dry leaf margins can occur for several reasons, including water stress, nutrient-related problems, or infection. Therefore, the system treats such visual signs as a reason to inspect the plant rather than as proof of a particular disease. This approach helps avoid misleading conclusions when the available visual information is limited.

Progress Monitoring

The dashboard can display changes in a plant's recorded health status over a selected period, as illustrated in Fig. 6. For example, consider a ten-day monitoring period with observations recorded on days 1, 5, and 10. In this illustrative case, the plant is marked as Poor on day 1 when browning is first observed. By day 5, its status changes to Fair after the watering and fertilizer schedules are adjusted. On day 10, the recorded status improves to Good.

These values are example entries used to demonstrate how health-status changes can be presented in a chart; they are not results from a controlled experiment. In the implemented system, the chart is generated from the health_status values stored in the growth_tracking table and arranged according to the date of each record. This allows users to review recorded changes over time and compare them with their care observations.

Implications for the System Design

The sample records highlight several important points that need to be considered when designing the database and analysis components. First, Equation (3) requires the time at which each plant measurement was recorded. Since the columns shown in Fig. 3 do not include a recording date, a timestamp field is needed to calculate growth rates and trends using actual elapsed time rather than simply relying on the order of data entry.

Second, some records do not contain observation notes. Therefore, the analysis methods should rely primarily on structured information, such as plant height, leaf count, and environmental readings, rather than depending on text descriptions that may be missing. Finally, the available records contain only the health-status labels Good and

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

Excellent. This limited variety of labels is not sufficient to train a reliable health-classification model. Additional records covering a wider range of plant conditions would be needed before such a model could be developed and properly evaluated.

Planned Testing

The system will be tested in three stages to check whether its main functions work as expected.

The first stage involves unit testing, which will check individual components, including the validation rules listed in Table III, the reminder due-date calculation in Equation (7), and the watering and unusual-change detection rules in Equations (8) and (9). The tests will also include values that fall exactly on the defined thresholds to verify that the system handles boundary conditions correctly.

The second stage involves integration testing. This will follow plant information from the manual entry form or sensor message through validation and database storage to its display on the dashboard. It will also verify that reminders marked as completed do not continue to generate unnecessary notifications.

The third stage involves a usability evaluation with a small group of plant owners. The aim is to understand whether users can interpret the reminders and warnings easily and whether entering plant information regularly is practical for them.

Limitations of the Evaluation

The current evaluation is based on records stored in a prototype rather than a controlled experiment. The leaf photographs and the ten-day progress chart are illustrative examples and should not be interpreted as experimental findings. The examples do not account for factors such as actual elapsed time and plant species in the analytical calculations. No formal model accuracy metrics have been reported, and the system has not been compared experimentally with other plant care applications.

A more complete evaluation will require a larger dataset collected over several weeks of actual plant care. Such a dataset would help assess growth prediction and other analytical methods using the evaluation procedure described in Section XI. Collecting and evaluating these real-world records is an important next step in the development of the system.

XIV. ADVANTAGES, LIMITATIONS AND FUTURE SCOPE

A. Advantages

Table IX compares traditional manual plant care with the proposed system. The application helps users organize care schedules and reduces the need to remember every task individually. It also maintains historical records for each plant, making it easier to review growth measurements and previous observations. Since users can enter information manually, the system can be used from the beginning without purchasing sensors or other hardware.

Another advantage is its modular design. The rule-based approach makes reminders and basic warnings easier to understand, while the architecture allows optional sensor integration and machine learning models to be added as the project develops. This provides flexibility for future improvements without requiring a complete redesign of the application.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

TABLE.IX. MANUAL CARE VERSUS PROPOSED SYSTEM

Criterion	Manual Care	Proposed System
Reminders	Depends on memory or written notes.	Provides scheduled reminders and tracks task completion.
Record Keeping	Records may be scattered or not maintained regularly.	Stores plant information and growth history in one place.
Plant-Specific Needs	Depends mainly on the user's knowledge and experience.	Supports care schedules based on individual plant profiles.
Environmental Conditions	Relies mainly on visual observation.	Stores environmental readings entered manually or collected through optional sensors.
Early Warnings	Problems may be noticed only after visible changes occur.	Flags unusual measurements or changes for further inspection.
Managing Multiple Plants	Tracking different care schedules can be difficult.	Allows users to manage multiple plants through a single dashboard.

B. Limitations

The proposed system has some limitations that need to be considered. The effectiveness of its rules depends on selecting suitable thresholds for different plant species. Sensor readings may also require calibration to ensure that the measurements are meaningful. In addition, environmental data alone cannot identify every plant disease or explain every change in plant health. The reliability of machine learning predictions also depends on the quantity and quality of the available training data.

Until a sufficiently large and representative dataset has been collected and evaluated, the analytical features should be used as decision-support tools rather than as a source of definitive conclusions. Users should review the warnings and suggestions alongside their own observations before deciding what care a plant requires.

C. Practical Considerations

Since unsuitable care recommendations may negatively affect plant health, the system presents its suggestions as optional guidance that users can accept or ignore. Where possible, each suggestion should also identify the measurement or rule that triggered it, helping users understand the reason behind the recommendation.

Plant photographs and observation notes should remain accessible only to the account that owns them. Email notifications are optional, allowing users to enable or disable this feature according to their preferences. These measures are intended to make the application more transparent, practical, and respectful of user privacy.

D. Future Scope

Several improvements can be explored in future versions of the system:

1. **Image-Based Plant Analysis:** Introduce image-classification models to identify visible leaf symptoms using a suitable labelled dataset.
2. **Plant Species Knowledge Base:** Develop a database of plant species with suggested watering schedules, fertilizer intervals, and environmental thresholds.
3. **Weather Integration:** Use local weather information to help adjust watering suggestions according to changing conditions.
4. **Expanded Notifications:** Add push notifications and SMS alongside existing in-app and optional email alerts.
5. **Mobile Application:** Develop a dedicated mobile application to make plant monitoring and care management more convenient.
6. **Automated Irrigation:** Integrate an ESP32-controlled irrigation mechanism to support automatic watering based on suitable sensor readings and configured rules.

These extensions could improve the system's flexibility and usefulness. However, each feature would require appropriate development and testing before being included in a practical deployment.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

The system can also be extended with actuators to support automated watering and with multi-user management features for nurseries and college gardens. These additions would make it easier to manage a larger number of plants and coordinate care activities among multiple users.

XV. CONCLUSION

This paper presented the design of an AI-Powered Smart Plant Care Reminder and Growth Monitoring System, a web-based application intended to make routine plant care easier to manage. The system brings together plant profiles, care reminders, growth records, environmental measurements, and data analysis within a single platform. Its modular architecture supports manual data entry from the beginning, with the option to integrate IoT sensors in future versions. MySQL provides a structured way to store plant information and maintain historical records.

The proposed system uses rule-based logic to manage care reminders and identify unusual changes in recorded measurements. Machine learning techniques, including regression and decision trees, can be introduced to estimate growth trends and classify plant conditions when sufficient reliable data becomes available. The preliminary evaluation demonstrated how the prototype can store growth records, display changes over time, and flag unusual measurements for further inspection. However, the available sample data is not sufficient to establish the accuracy or reliability of machine learning predictions.

Future development will focus on collecting a larger dataset from real plant-care activities, evaluating suitable machine learning models, and integrating additional features such as environmental sensors and image-based plant analysis. With further development and testing, the system could provide a practical tool for organizing plant maintenance and supporting more informed plant care decisions.

XVI. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to their project guide, Prof. C. W. Rawarkar, for the valuable guidance, encouragement, and support provided throughout the project. They also thank the Department of Computer Science and Engineering at Prof. Ram Meghe Institute of Technology and Research, Badnera, Amravati, for providing the support and facilities needed to carry out this work.

REFERENCES

[1] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, “Machine learning in agriculture: A review,” Sensors, vol. 18, no. 8, Art. no. 2674, 2018.
[2] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” Computers and Electronics in Agriculture, vol. 147, pp. 70–90, 2018.
[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, 2013.
[4] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. N. Hindia, “An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges,” IEEE Internet of Things Journal, vol. 5, no. 5, pp. 3758–3773, 2018.
[5] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour, and E. M. Aggoune, “Internet-of-Things (IoT)-based smart agriculture: Toward making the fields talk,” IEEE Access, vol. 7, pp. 129551–129583, 2019.
[6] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” Frontiers in Plant Science, vol. 7, Art. no. 1419, 2016.
[7] K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” Computers and Electronics in Agriculture, vol. 145, pp. 311–318, 2018.
[8] W. McKinney, “Data structures for statistical computing in Python,” in Proc. 9th Python in Science Conf., 2010, pp. 51–56.
[9] M. Grinberg, Flask Web Development: Developing Web Applications with Python, 2nd ed. Sebastopol, CA, USA: O’Reilly Media, 2018.
[10] J. D. Hunter, “Matplotlib: A 2D graphics environment,” Computing in Science & Engineering, vol. 9, no. 3, pp. 90–95, 2007.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510101|

[11] Oracle Corporation, MySQL 8.0 Reference Manual. [Online]. Available: https://dev.mysql.com/doc/refman/8.0/en/
[12] C. R. Harris et al., "Array programming with NumPy," Nature, vol. 585, pp. 357–362, 2020.
[13] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[14] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York, NY, USA: Springer, 2009.
[15] J. R. Quinlan, "Induction of decision trees," Machine Learning, vol. 1, no. 1, pp. 81–106, 1986.
[16] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.



INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details