**MEDICINAL PLANT IDENTIFICATION APPLICATION**

19IT452 JAVA PROGRAMMING LAB

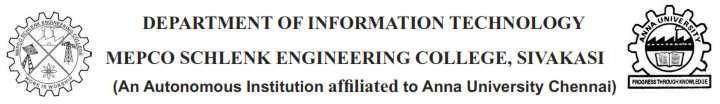
**MINI PROJECT REPORT**

**SUBMITTED BY ,**

Padmanabhan S [9517202206250]

Sanjeevi VP [9517202206045]

Srilaxman EU [9517202206050]



2023 - 2024

**CERTIFICATE**

This is to certify that it is the bonafide work done by **Srilaxman EU [9517202206050] , Padmanabhan S [9517202206033], Sanjeevi [95172022060]** for their Mini Project on **MEDICINAL PLANT IDENTIFICATION APPLICATION** in the 19IT452 JAVA PROGRAMMING LAB at Mepco Schlenk Engineering College, Sivakasi during the year 2023-2024

.………………… …………………………

Staff in charge Head of the Department

**ACKNOWLEDGEMENT**

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and our reputed institution Mepco Schlenk Engineering College, Sivakasi. We would like to extend our sincere thanks to our Principal Dr. S. Arivazhagan ME., PhD.

We would like to express my special gratitude and thanks to Dr.T.Revathi ME., PhD. Senior Professor & Head of the Department for giving us such a wonderful experience.

We are highly indebted to Mrs. M. Blessa Binolin Pepsi AP (SL.Grade)/IT, Mrs. M. Prasha Meena AP/IT for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our gratitude towards other teaching and non teaching staff members of IT Department for their kind co-operation and encouragement which help us in all means to complete this project successfully.

**ABSTRACT**

The identification of medicinal plants and raw materials is a critical challenge in the field of Ayurvedic Pharmaceutics. India, with its rich floral diversity, is a treasure trove of medicinal plant wealth, but the accurate identification of these plants and raw materials is often a complex task. Several crude drugs are being sold under the same name in the market, leading to confusion and misidentification. Even collectors and traders are not always aware of the exact morphological appearance or differentiating attributes of many drugs due to seasonal and geographical variations, as well as similar characteristics.

Moreover, the extensive consumption of these medicinal plants and raw materials to meet the demand-supply ratio exerts a heavy strain on the existing resources. This situation further leads to the practice of adulteration, substitution, and ultimately, a loss of faith in the curative capabilities of the Ayurvedic system. To address these challenges, this project aims to develop a software solution capable of identifying different medicinal plants and raw materials through image processing using machine learning algorithms.

The proposed software will leverage cutting-edge computer vision and machine learning techniques to analyze and classify images of medicinal plants and raw materials. By leveraging a comprehensive dataset of labeled images, the system will learn to recognize and differentiate between various plant species and raw materials based on their unique morphological features, such as leaf shape, color, texture, and other distinguishing characteristics.

The software will be designed as a user-friendly interface, allowing users to upload images of plants or raw materials for identification. The system will then process the image, extract relevant features, and use trained machine learning models to classify the input image into the corresponding plant or raw material category. The identification results will be presented to the user, along with relevant information about the identified plant or raw material, such as its scientific name, common names, medicinal properties, and potential uses.

The development of this software will involve several stages, including data collection and preparation, feature extraction and selection, model training and evaluation, and user interface design. The project will utilize Java Swing for the frontend development and MySQL for the backend database management.

By providing an accurate and reliable means of identifying medicinal plants and raw materials, this software will be of immense value to various stakeholders in the Ayurvedic industry, including wholesalers, distributors, and practitioners. It will contribute to reducing instances of adulteration and substitution, ensuring the quality and efficacy of Ayurvedic products, and ultimately promoting trust and confidence in the Ayurvedic system of medicine.

# TABLE OF CONTENTS

|  |  |
| --- | --- |
| 1.Introduction | 6 |
| 1.1Purpose | 7 |
| 1.1.1 SIH – Description | 8 |
| 1.2 Scope | 9 |
| 1.3 Definitions, Acronyms and Abbreviations | 11 |
| 1.4 References | 13 |
| 1.5 Overview | 14 |
| 1.6 Concept Overview | 15 |
| 1.7 Our Work | 17 |
| 2. ER Diagram | 18 |
| 3.Implementation/Coding | 19 |
| 4..Screen Shots | 102 |
| 5.Conclusion | 111 |

1. **INTRODUCTION**

India is a land of immense biodiversity, with a rich heritage of medicinal plant wealth that has been an integral part of traditional healthcare systems like Ayurveda for centuries. However, the accurate identification of medicinal plants and raw materials has been a longstanding challenge in the field of Ayurvedic Pharmaceutics. The misidentification of these plants and raw materials can lead to severe consequences, including adulteration, substitution, and, ultimately, a loss of faith in the efficacy of Ayurvedic treatments.

One of the primary reasons for misidentification is the lack of awareness and expertise among collectors, traders, and even practitioners. Many crude drugs are sold under the same name in the market, leading to confusion and uncertainty. Furthermore, the morphological appearance and distinguishing attributes of various plants and raw materials can vary significantly due to seasonal and geographical factors, as well as their inherent similarities.

The increasing demand for Ayurvedic products has also exacerbated the strain on existing resources, prompting unethical practices such as adulteration and substitution. These practices not only undermine the quality and effectiveness of Ayurvedic medicines but also pose potential health risks to consumers.

To address these challenges, the application of modern technology, particularly in the fields of image processing and machine learning, presents a promising solution. The proposed software aims to leverage these cutting-edge technologies to accurately identify different medicinal plants and raw materials based on their visual characteristics.

The software will employ computer vision techniques to extract relevant features from images of plants and raw materials, such as leaf shape, color, texture, and other distinguishing characteristics. These features will then be fed into machine learning algorithms, which will be trained on a comprehensive dataset of labeled images to recognize and classify the input images into the corresponding plant or raw material category.

By providing an automated and reliable means of identification, this software will address the challenges faced by various stakeholders in the Ayurvedic industry, including wholesalers, distributors, and practitioners. It will aid in reducing instances of adulteration and substitution, ensuring the quality and efficacy of Ayurvedic products, and ultimately promoting trust and confidence in the Ayurvedic system of medicine.

The development of this software will involve several stages, including data collection and preparation, feature extraction and selection, model training and evaluation, and user interface design. The project will leverage Java Swing for the frontend development and MySQL for the backend database management, ensuring a user-friendly and efficient solution.By combining the rich heritage of Ayurvedic knowledge with modern technological advancements, this project aims to contribute to the preservation and promotion of traditional healthcare systems while ensuring the highest standards of quality and safety for consumers.

# PURPOSE

The primary purpose of this project is to develop a reliable and user-friendly software solution that can accurately identify different medicinal plants and raw materials used in Ayurvedic Pharmaceutics through image processing and machine learning techniques. By leveraging cutting-edge computer vision and machine learning algorithms, the software aims to address the long-standing challenge of misidentification and adulteration of medicinal plants and raw materials, which has been a significant issue in the Ayurvedic industry.

The software will serve as a valuable tool for various stakeholders, including wholesalers, distributors, practitioners, and consumers, by providing an automated and accurate means of identifying and verifying the authenticity of the medicinal plants and raw materials used in Ayurvedic products. This will not only enhance the quality and efficacy of Ayurvedic medicines but also promote trust and confidence in the Ayurvedic system of medicine, ultimately contributing to the preservation and advancement of this traditional healthcare system.

The accurate identification of medicinal plants and raw materials is crucial for ensuring the quality, safety, and efficacy of Ayurvedic medicines. Misidentification can lead to the use of incorrect or adulterated ingredients, compromising the therapeutic properties of the formulations and potentially causing adverse health effects. By providing a reliable means of identification, this software will empower stakeholders to make informed decisions and maintain the highest standards of quality control.

Furthermore, the software will contribute to the conservation and sustainable utilization of medicinal plant resources. By enabling precise identification, it will aid in monitoring and regulating the harvesting and trade of medicinal plants, preventing overexploitation and ensuring the long-term availability of these valuable natural resources. This aligns with the broader goals of promoting biodiversity conservation and sustainable development practices.

In addition to its practical applications, the software will serve as a valuable educational and research tool. It will facilitate the dissemination of knowledge about medicinal plants and their identification, benefiting students, researchers, and practitioners in the field of Ayurvedic Pharmaceutics. The software's ability to provide detailed information about identified plants, including their scientific names, common names, medicinal properties, and uses, will contribute to the preservation and transmission of traditional knowledge.

Moreover, the software's integration of modern technologies like image processing and machine learning will demonstrate the potential for synergy between traditional knowledge systems and contemporary scientific advancements. By bridging this gap, the project aims to foster interdisciplinary collaboration and promote the integration of Ayurveda into mainstream healthcare practices, ultimately benefiting society at large.

# SIH PROBLEM STATEMENT - SIH 1456

**Problem Statement Title**: Identification of Different Medicinal Plants/Raw materials through Image Processing Using Machine Learning Algorithms

### Description:

India, with a rich heritage of floral diversity, is well-known for its medicinal plant wealth, but their identification is one of the major burning issues in Ayurvedic Pharmaceutics. Several crude drugs are being sold under the same name in the market leading to confusion and their misidentification. Even the collectors and traders are not completely aware of the exact morphological appearance or differentiating attributes of the many drugs owing to seasonal and geographical availability, and similar characteristics. Moreover, the extensive consumption to meet demand-supply ratio exerts a heavy strain on the existing resources. It further leads to the practice of adulteration, substitution, and disbelief in the curative capability of the system eventually. Thus, software capable of identifying different medicinal plants/ raw materials through Image Processing Using Different Machine Learning Algorithms will be of immense use. It will be helpful at every level viz. wholesaler, distributor, etc. of the supply chain of raw material being utilized in the system.

# SCOPE

The scope of the Image Classification website project encompasses the following aspects:

1. Frontend Development:

The scope of frontend development encompasses the creation of an intuitive and user-friendly interface for the software. This includes designing and implementing a graphical user interface (GUI) using Java Swing, which will allow users to easily upload images of medicinal plants or raw materials for identification. The frontend will provide a seamless experience for users to interact with the software, submit their image queries, and receive the identification results along with relevant information about the identified plant or raw material.

1. Backend Development:

The backend development scope encompasses the implementation of the core image processing and machine learning algorithms that will power the identification process. This includes data preprocessing, feature extraction, model training, and prediction. The backend will be responsible for processing the images uploaded through the frontend, extracting relevant features, and utilizing trained machine learning models to classify the images into their respective plant or raw material categories. The backend will also integrate with a MySQL database to store and retrieve information about the identified plants and raw materials.

1. Integration:

The integration scope involves seamlessly integrating the frontend and backend components to create a cohesive and fully functional software solution. This includes establishing communication channels between the frontend and backend, ensuring smooth data transfer and exchange, and implementing error handling and exception management mechanisms. The integration process will also involve testing the end-to-end functionality of the software to ensure that all components work together as intended.

1. Maintenance and Support:

The scope of maintenance and support encompasses ongoing activities to ensure the software's continued operation, performance, and relevance. This includes monitoring the software for any issues or bugs, addressing user feedback and requests for enhancements, and providing technical support to users. Additionally, the scope may include periodic updates to the software, such as incorporating new plant or raw material data, improving the machine learning models, or adapting to changes in the underlying technologies.

# DEFINITIONS , ACRONYMS AND ABBREVIATIONS

1. **Ayurvedic Pharmaceutics**: A branch of Ayurveda that deals with the preparation, formulation, standardization, and quality control of Ayurvedic medicines.

* **Medicinal Plants**: Plants that possess therapeutic properties and are used in traditional or modern healthcare systems for the treatment or prevention of diseases.
* **Raw Materials**: Natural substances, such as plant parts (roots, leaves, bark, etc.), minerals, or animal products, used as ingredients in the preparation of Ayurvedic formulations.
* **Machine Learning**: A subset of artificial intelligence that enables systems to learn and improve from experience without being explicitly programmed.
* **Feature Extraction**: The process of extracting relevant and informative characteristics or attributes from raw data, such as images, to be used as input for machine learning models.
* **Model Training**: The process of feeding labeled data to a machine learning algorithm to enable it to learn patterns and relationships, thereby building a predictive model.
* **Classification**: A supervised machine learning task that involves assigning input data to predefined categories or classes based on learned patterns.

1. API (Application Programming Interface): A set of protocols and tools for building software and applications, allowing different software systems to communicate with each other.
2. CI/CD (Continuous Integration/Continuous Deployment): Practices in software development where code changes are automatically built, tested, and deployed to production, ensuring frequent and reliable updates.
3. CRUD (Create, Read, Update, Delete): The four basic operations of persistent storage in databases and applications, representing the fundamental actions for managing data.
4. CSS (Cascading Style Sheets): A stylesheet language used for describing the presentation of a document written in HTML or XML, used to style and layout web pages.
5. HTML (HyperText Markup Language): The standard markup language for creating web pages and web applications.
6. UX (User Experience): The overall experience of a person using a product, especially in terms of how easy and pleasant it is to use.
7. IDE (Integrated Development Environment): A software application providing comprehensive facilities to programmers for software development, including a code editor, debugger, and build automation tools.
8. JSON (JavaScript Object Notation): A lightweight data interchange format that is easy for humans to read and write, and easy for machines to parse and generate.
9. SEO (Search Engine Optimization): The process of improving the visibility and ranking of a website or web page in search engine results.
10. SQL (Structured Query Language): A standardized language for managing and manipulating relational databases.
11. UI/UX (User Interface/User Experience): Combined terms referring to the design and user interaction aspects of digital products, focusing on creating effective and enjoyable user experiences.
12. CRUD (Create, Read, Update, Delete): Basic operations for manipulating data in a database, crucial for managing application data.
13. IDE (Integrated Development Environment): A software suite that consolidates basic tools required for software development, such as coding, debugging, and testing.

This section ensures that all technical terms and abbreviations used in the project documentation are clearly defined for better understanding and consistency.

# REFERENCES

1. Official Documentation:
   * React Documentation: [https://reactjs.org/docs/getting-started.html]
   * Java Spring Boot Documentation: [https://spring.io/projects/spring-boot]
2. Tutorials and Guides:
   * React and Spring Boot Integration Tutorial: [Example Tutorial Link]
   * Building RESTful APIs with Spring Boot: [Spring Boot RESTful Tutorial]
3. Books:
   * "Pro Spring Boot 2" by Felipe Gutierrez
   * "Learning React: A Hands-On Guide to Building Web Applications Using React and Redux" by Kirupa Chinnathambi
4. Online Courses:
   * Udemy Course: "Full Stack Development with Spring Boot and React"
   * Pluralsight Course: "Building Full-Stack Apps with React and Spring Boot"
5. GitHub Repositories:
   * Example Spring Boot Repository: [Spring Boot Example Repo]
   * Example React Repository: [React Example Repo]
6. Community Forums:
   * Stack Overflow: [https://stackoverflow.com/]
   * Reddit React Community: [https://[www.reddit.com/r/reactjs/](http://www.reddit.com/r/reactjs/)]
   * Reddit Spring Boot Community: [https://[www.reddit.com/r/springboot](http://www.reddit.com/r/springboot)
7. Other Resources:
   * Baeldung: [https://[www.baeldung.com](http://www.baeldung.com/) ]Provides in-depth tutorials and articles on Java, Spring Boot, and related technologies.
   * Medium: [https://medium.com Look for articles and tutorials written by developers sharing their experiences with React and Spring Boot.

Ensure to format the references according to the citation style preferred by your institution or organization.

# OVERVIEW

The identification of medicinal plants and raw materials is a crucial aspect of Ayurvedic Pharmaceutics, but it remains a significant challenge due to the rich diversity of plant species, geographical variations, and similarities in morphological characteristics. Misidentification can lead to adulteration, substitution, and ultimately, a loss of confidence in the efficacy of Ayurvedic treatments. This project aims to address this challenge by developing a software solution that leverages image processing and machine learning techniques to accurately identify different medicinal plants and raw materials.

The software will be designed as a user-friendly application with a graphical user interface (GUI) developed using Java Swing. Users, such as wholesalers, distributors, practitioners, or consumers, will be able to upload images of medicinal plants or raw materials through the intuitive interface. The backend of the software will then process these images using advanced computer vision algorithms to extract relevant features, such as leaf shape, color, texture, and other distinguishing characteristics.

These extracted features will be fed into machine learning models that have been trained on a comprehensive dataset of labeled images. The models will be capable of recognizing patterns and making accurate predictions, classifying the input images into their respective plant or raw material categories. The software will then present the identification results to the user, along with relevant information about the identified plant or raw material, such as its scientific name, common names, medicinal properties, and potential uses.

The backend development will involve implementing various image processing techniques, feature extraction algorithms, and machine learning models, such as support vector machines (SVMs), convolutional neural networks (CNNs), or ensemble methods. The choice of algorithms and models will be based on their suitability for the specific task, accuracy, and computational efficiency.

The software will integrate with a MySQL database to store and retrieve information about the identified plants and raw materials, ensuring efficient data management and retrieval. The database will be designed to accommodate detailed information, including images, descriptions, and other relevant data for each plant or raw material.

Throughout the development process, rigorous testing and quality assurance measures will be implemented to ensure the software's reliability, accuracy, and performance. This will include unit testing individual components, integration testing to verify the correct functioning of the integrated system, and user acceptance testing to validate the software's compliance with user requirements.

Upon successful testing and deployment, the software will serve as a valuable tool for various stakeholders in the Ayurvedic industry.

# CONCEPT OVERVIEW

The proposed software solution for identifying medicinal plants and raw materials through image processing and machine learning algorithms will leverage several key technologies and concepts. The primary components include Java Swing for frontend development, MySQL for backend database management, and various image processing and machine learning techniques for the core functionality

### Java Swing (Frontend)

Java Swing is a widely-used GUI (Graphical User Interface) toolkit for building desktop applications in Java. It provides a rich set of components and libraries for creating intuitive and user-friendly interfaces. In this project, Java Swing will be utilized to develop the frontend of the software, enabling users to easily upload images of medicinal plants or raw materials and receive the identification results along with relevant information.

### MySQL (Database)

MySQL is an open-source relational database management system known for its reliability, scalability, and performance. It is widely used in web applications for storing and managing structured data. MySQL supports SQL queries for data manipulation and retrieval, making it a robust choice for applications requiring data persistence.

### Benefits

### The integration of Java Swing, MySQL, image processing, and machine learning techniques offers several benefits for this project:

1. **User-friendly Interface**: Java Swing facilitates the development of an intuitive and user-friendly GUI, ensuring a seamless experience for users to interact with the software and submit their image queries.
2. **Efficient Data Management**: MySQL provides a robust and scalable solution for storing and retrieving large amounts of data related to medicinal plants and raw materials, enabling efficient data management and retrieval.
3. **Accurate Identification**: By leveraging advanced image processing and machine learning algorithms, the software can accurately extract relevant features from images and classify them into their respective plant or raw material categories, addressing the longstanding challenge of misidentification.
4. **Automation and Time-Saving**: The automated identification process eliminates the need for manual inspection and cross-referencing, saving time and reducing the risk of human error.
5. **Conservation and Sustainability**: By enabling precise identification, the software can aid in monitoring and regulating the harvesting and trade of medicinal plants, contributing to the conservation and sustainable utilization of these valuable natural resources.

### Challenges

While the proposed solution offers significant benefits, there are also potential challenges that need to be addressed:

1. **Data Acquisition**: Obtaining a comprehensive and accurately labeled dataset of images for training the machine learning models can be a time-consuming and resource-intensive process, requiring collaboration with domain experts and extensive field work.
2. **Computational Complexity**: Image processing and machine learning algorithms can be computationally intensive, especially for large datasets or complex models. Ensuring efficient processing and reasonable response times may require optimizations or the use of specialized hardware.
3. **Handling Variability**: Medicinal plants and raw materials can exhibit significant variability in their appearance due to factors such as growth stage, environmental conditions, and geographical location. Addressing this variability in the identification process can be challenging.
4. **Integration and Usability**: Seamlessly integrating the frontend, backend, and core functionality while maintaining a user-friendly and intuitive interface can be a complex task, requiring careful design and testing.
5. **Maintenance and Updating**: As new medicinal plants or raw materials are discovered or introduced, the software will require regular updates and maintenance to ensure its continued accuracy and relevance.

By addressing these challenges through careful planning, robust implementation, and continuous improvement, the proposed software solution can effectively leverage the benefits of Java Swing, MySQL, image processing, and machine learning to provide a reliable and valuable tool for the identification of medicinal plants and raw materials in the field of Ayurvedic Pharmaceutics.

# OUR WORK:

Image Classification Users, such as wholesalers, distributors, practitioners, or consumers, will be able to upload images of medicinal plants or raw materials through the intuitive interface. The backend of the software will then process these images using advanced computer vision algorithms to extract relevant features, such as leaf shape, color, texture, and other distinguishing characteristics.

### Key Points

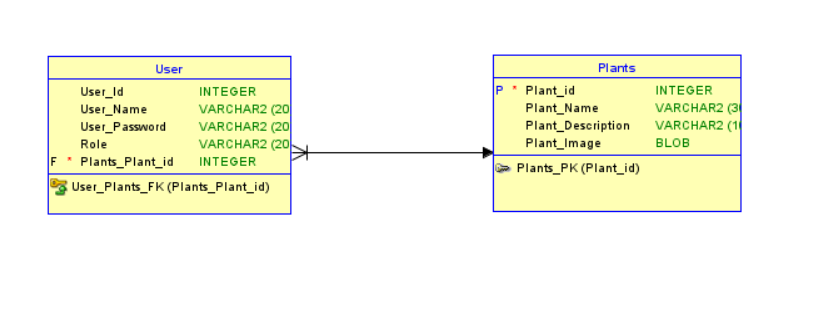
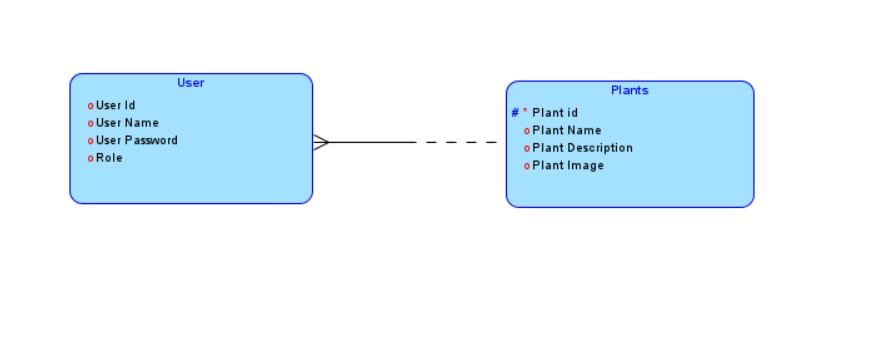
1. **Implementation**: The implementation of this software solution involves a multi-faceted approach that combines various technologies and techniques. The frontend development will be carried out using Java Swing, leveraging its rich set of components and libraries to create an intuitive and user-friendly graphical user interface (GUI). This will allow users to seamlessly upload images of medicinal plants or raw materials and receive the identification results along with relevant information.
2. **Image Classification**: The core functionality of the software revolves around the accurate classification of medicinal plants and raw materials based on their visual characteristics. Advanced image processing techniques will be employed to extract relevant features from the uploaded images, such as leaf shape, color, texture, and other distinguishing attributes. These extracted features will then be fed into machine learning models that have been trained on a comprehensive dataset of labeled images.
3. **Features**: The software will be designed to offer a range of features to enhance its usability and functionality. In addition to accurate identification, the software will provide detailed information about the identified plants or raw materials, including their scientific names, common names, medicinal properties, and potential uses. This will serve as a valuable resource for users, promoting better understanding and appropriate utilization of these natural resources.
4. **User Interaction Design**: Ensuring a seamless and intuitive user experience is a crucial aspect of this project. The user interaction design will focus on creating a visually appealing and easy-to-navigate interface, with clear instructions and guidance for users throughout the identification process. The software will also incorporate error handling and feedback mechanisms to ensure a smooth and efficient experience for users.

6. **Future Trends and Innovations**: The field of image processing and machine learning is rapidly evolving, with new techniques and models being developed continuously. As part of this project, we will stay abreast of the latest advancements and emerging trends in these domains. This will enable us to incorporate cutting-edge innovations into the software, further enhancing its accuracy, efficiency, and capabilities.

Additionally, we will explore the potential for integrating the software with other digital platforms and tools, such as mobile applications or cloud-based services. This will facilitate broader access and enable users to leverage the identification capabilities from various devices and locations, fostering greater accessibility and convenience.

By embracing future trends and innovations, we aim to ensure that this software solution remains relevant, adaptable, and at the forefront of technological advancements, continually meeting the evolving needs of the Ayurvedic industry and contributing to the preservation and promotion of traditional healthcare systems.

2.**ER-DIAGRAM**



3. **IMPLEMENTATION**

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.ByteArrayInputStream;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import java.sql.\*;

import javax.imageio.ImageIO;

public class MedicinalPlantsApp extends JFrame {

private JTextField usernameField;

private JPasswordField passwordField;

private JButton loginButton;

private final JPanel mainPanel;

private final CardLayout cardLayout;

private JTable plantTable;

private DefaultTableModel tableModel;

private JTextField nameField;

private JTextArea descriptionField;

private JTextField plantIdField;

private JLabel imageLabel;

private File selectedImageFile;

Color c1 = new Color(102, 255, 102);

// Database connection details

private static final String DB\_URL = "jdbc:mysql://localhost:3306/medicinalplants";

private static final String DB\_USER = "root";

private static final String DB\_PASSWORD = "srilaxman";

public MedicinalPlantsApp() {

setTitle("Medicinal Plant Identification");

setSize(800, 600);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

cardLayout = new CardLayout();

mainPanel = new JPanel(cardLayout);

mainPanel.add(createLoginPanel(), "login");

mainPanel.add(createAdminPanel(), "admin");

mainPanel.add(createUserPanel(), "user");

add(mainPanel);

cardLayout.show(mainPanel, "login");

}

private JPanel createLoginPanel() {

JPanel loginPanel = new BackgroundPanel(new ImageIcon("C:\\Users\\SRINAMISH\\Desktop\\Mini Project\\Java/background.jpg").getImage(), 0.5f);

loginPanel.setLayout(new GridBagLayout());

GridBagConstraints gbc = new GridBagConstraints();

gbc.gridx = 0;

gbc.gridy = 0;

gbc.insets = new Insets(5, 5, 5, 5);

JLabel headingLabel = new JLabel("Medicinal Plants Identification");

headingLabel.setFont(new Font("Serif", Font.BOLD, 30));

gbc.gridwidth = 2;

loginPanel.add(headingLabel, gbc);

gbc.gridwidth = 1;

gbc.gridy++;

usernameField = new JTextField(15);

passwordField = new JPasswordField(15);

loginButton = new JButton("Login");

loginButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

authenticate();

}

});

usernameField.addActionListener(new AbstractAction() {

@Override

public void actionPerformed(ActionEvent e) {

loginButton.doClick();

}

});

passwordField.addActionListener(new AbstractAction() {

@Override

public void actionPerformed(ActionEvent e) {

loginButton.doClick();

}

});

gbc.gridx = 0;

gbc.gridy++;

loginPanel.add(new JLabel("Username:"), gbc);

gbc.gridx++;

loginPanel.add(usernameField, gbc);

gbc.gridx = 0;

gbc.gridy++;

loginPanel.add(new JLabel("Password:"), gbc);

gbc.gridx++;

loginPanel.add(passwordField, gbc);

gbc.gridx = 0;

gbc.gridy++;

gbc.gridwidth = 2;

loginPanel.add(loginButton, gbc);

return loginPanel;

}

private JPanel createAdminPanel() {

JPanel adminPanel = new JPanel(new BorderLayout());

adminPanel.setBackground(c1);

JLabel label = new JLabel("Admin Panel", JLabel.CENTER);

adminPanel.add(label, BorderLayout.NORTH);

JPanel plantPanel = new JPanel(new GridBagLayout());

plantPanel.setBackground(c1);

GridBagConstraints gbc = new GridBagConstraints();

gbc.insets = new Insets(5, 5, 5, 5);

plantIdField = new JTextField(5);

nameField = new JTextField(20);

descriptionField = new JTextArea(5, 20);

JButton addButton = new JButton("Add Plant");

JButton updateButton = new JButton("Update Plant");

JButton deleteButton = new JButton("Delete Plant");

JButton loadButton = new JButton("Load Plant");

JButton addImageButton = new JButton("Add Image");

imageLabel = new JLabel();

gbc.gridx = 0;

gbc.gridy = 0;

plantPanel.add(new JLabel("Plant ID:"), gbc);

gbc.gridx++;

plantPanel.add(plantIdField, gbc);

gbc.gridx++;

plantPanel.add(loadButton, gbc);

gbc.gridx = 0;

gbc.gridy++;

plantPanel.add(new JLabel("Name:"), gbc);

gbc.gridx++;

plantPanel.add(nameField, gbc);

gbc.gridx = 0;

gbc.gridy++;

plantPanel.add(new JLabel("Description:"), gbc);

gbc.gridx++;

plantPanel.add(new JScrollPane(descriptionField), gbc);

gbc.gridx = 0;

gbc.gridy++;

plantPanel.add(new JLabel("Image:"), gbc);

gbc.gridx++;

plantPanel.add(imageLabel, gbc);

gbc.gridx++;

plantPanel.add(addImageButton, gbc);

gbc.gridx = 0;

gbc.gridy++;

gbc.gridwidth = 3;

plantPanel.add(addButton, gbc);

gbc.gridy++;

plantPanel.add(updateButton, gbc);

gbc.gridy++;

plantPanel.add(deleteButton, gbc);

adminPanel.add(plantPanel, BorderLayout.CENTER);

addButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

addPlant(nameField.getText(), descriptionField.getText());

}

});

updateButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

updatePlant(plantIdField.getText(), nameField.getText(), descriptionField.getText());

}

});

deleteButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

deletePlant(plantIdField.getText());

}

});

loadButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

loadPlant(plantIdField.getText(), nameField, descriptionField);

}

});

addImageButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

addImage();

}

});

// Add table to display plants

tableModel = new DefaultTableModel(new Object[]{"ID", "Name", "Description"}, 0);

plantTable = new JTable(tableModel);

JScrollPane tableScrollPane = new JScrollPane(plantTable);

adminPanel.add(tableScrollPane, BorderLayout.SOUTH);

JButton logoutButton = new JButton("Logout");

logoutButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

cardLayout.show(mainPanel, "login");

}

});

adminPanel.add(logoutButton, BorderLayout.SOUTH);

return adminPanel;

}

private JPanel createUserPanel() {

JPanel userPanel = new JPanel(new BorderLayout());

userPanel.setBackground(c1);

JLabel label = new JLabel("User Panel - Plant Search", JLabel.CENTER);

userPanel.add(label, BorderLayout.NORTH);

JPanel searchPanel = new JPanel();

JTextField searchField = new JTextField(20);

JButton searchButton = new JButton("Search");

JButton searchImageButton = new JButton("Search by Image");

// Table to display search results

DefaultTableModel userTableModel = new DefaultTableModel(new Object[]{"ID", "Name", "Description"}, 0);

JTable userTable = new JTable(userTableModel);

JScrollPane userTableScrollPane = new JScrollPane(userTable);

searchButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

searchPlant(searchField.getText(), userTableModel);

}

});

searchImageButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

searchPlantByImage(userTableModel);

}

});

searchPanel.add(new JLabel("Search Plant:"));

searchPanel.add(searchField);

searchPanel.add(searchButton);

searchPanel.add(searchImageButton);

userPanel.add(searchPanel, BorderLayout.NORTH);

userPanel.add(userTableScrollPane, BorderLayout.CENTER);

JButton logoutButton = new JButton("Logout");

logoutButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

cardLayout.show(mainPanel, "login");

}

});

userPanel.add(logoutButton, BorderLayout.SOUTH);

return userPanel;

}

private void authenticate() {

String username = usernameField.getText();

String password = new String(passwordField.getPassword());

try (Connection conn = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

PreparedStatement ps = conn.prepareStatement("SELECT \* FROM users WHERE username = ? AND password = ?")) {

ps.setString(1, username);

ps.setString(2, password);

ResultSet rs = ps.executeQuery();

if (rs.next()) {

String role = rs.getString("role");

if ("admin".equals(role)) {

cardLayout.show(mainPanel, "admin");

loadAllPlants();

} else {

cardLayout.show(mainPanel, "user");

}

} else {

JOptionPane.showMessageDialog(this, "Invalid credentials");

}

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private void addPlant(String name, String description) {

try (Connection conn = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

PreparedStatement ps = conn.prepareStatement("INSERT INTO plants (name, description, image) VALUES (?, ?, ?)")) {

ps.setString(1, name);

ps.setString(2, description);

if (selectedImageFile != null) {

try {

// Log the file path and size for debugging

System.out.println("Selected image file: " + selectedImageFile.getAbsolutePath());

System.out.println("File size: " + selectedImageFile.length() + " bytes");

// Read the file into a byte array

byte[] imageBytes = new byte[(int) selectedImageFile.length()];

try (FileInputStream fis = new FileInputStream(selectedImageFile)) {

fis.read(imageBytes);

}

// Use ByteArrayInputStream to set the binary stream

try (ByteArrayInputStream bais = new ByteArrayInputStream(imageBytes)) {

ps.setBinaryStream(3, bais, imageBytes.length);

}

} catch (IOException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(this, "Error reading image file: " + ex.getMessage());

return; // Exit method if there's an error reading the image file

}

} else {

ps.setNull(3, Types.BLOB);

}

ps.executeUpdate();

JOptionPane.showMessageDialog(this, "Plant added successfully");

loadAllPlants();

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(this, "Error adding plant: " + ex.getMessage());

}

}

private void updatePlant(String id, String name, String description) {

try (Connection conn = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

PreparedStatement ps = conn.prepareStatement("UPDATE plants SET name = ?, description = ? WHERE id = ?")) {

ps.setString(1, name);

ps.setString(2, description);

ps.setInt(3, Integer.parseInt(id));

ps.executeUpdate();

JOptionPane.showMessageDialog(this, "Plant updated successfully");

loadAllPlants();

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private void deletePlant(String id) {

try (Connection conn = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

PreparedStatement ps = conn.prepareStatement("DELETE FROM plants WHERE id = ?")) {

ps.setInt(1, Integer.parseInt(id));

ps.executeUpdate();

JOptionPane.showMessageDialog(this, "Plant deleted successfully");

loadAllPlants();

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private void loadPlant(String id, JTextField nameField, JTextArea descriptionField) {

try (Connection conn = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

PreparedStatement ps = conn.prepareStatement("SELECT \* FROM plants WHERE id = ?")) {

ps.setInt(1, Integer.parseInt(id));

ResultSet rs = ps.executeQuery();

if (rs.next()) {

nameField.setText(rs.getString("name"));

descriptionField.setText(rs.getString("description"));

byte[] imageData = rs.getBytes("image");

if (imageData != null) {

ImageIcon imageIcon = new ImageIcon(imageData);

Image image = imageIcon.getImage().getScaledInstance(100, 100, Image.SCALE\_SMOOTH);

imageLabel.setIcon(new ImageIcon(image));

} else {

imageLabel.setIcon(null);

}

} else {

JOptionPane.showMessageDialog(this, "Plant not found");

}

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private void loadAllPlants() {

tableModel.setRowCount(0);

try (Connection conn = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM plants")) {

while (rs.next()) {

int id = rs.getInt("id");

String name = rs.getString("name");

String description = rs.getString("description");

tableModel.addRow(new Object[]{id, name, description});

}

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private void addImage() {

JFileChooser fileChooser = new JFileChooser();

int returnValue = fileChooser.showOpenDialog(this);

if (returnValue == JFileChooser.APPROVE\_OPTION) {

selectedImageFile = fileChooser.getSelectedFile();

try {

Image img = ImageIO.read(selectedImageFile).getScaledInstance(100, 100, Image.SCALE\_SMOOTH);

imageLabel.setIcon(new ImageIcon(img));

} catch (IOException ex) {

ex.printStackTrace();

}

}

}

private void searchPlant(String name, DefaultTableModel userTableModel) {

userTableModel.setRowCount(0); // Clear existing data

try (Connection conn = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

PreparedStatement ps = conn.prepareStatement("SELECT \* FROM plants WHERE name LIKE ?")) {

ps.setString(1, "%" + name + "%");

ResultSet rs = ps.executeQuery();

while (rs.next()) {

int id = rs.getInt("id");

String plantName = rs.getString("name");

String plantDescription = rs.getString("description");

userTableModel.addRow(new Object[]{id, plantName, plantDescription});

}

} catch (SQLException ex) {

ex.printStackTrace();

}

}

private void searchPlantByImage(DefaultTableModel userTableModel) {

JFileChooser fileChooser = new JFileChooser();

int returnValue = fileChooser.showOpenDialog(this);

if (returnValue == JFileChooser.APPROVE\_OPTION) {

File imageFile = fileChooser.getSelectedFile();

try (Connection conn = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

PreparedStatement ps = conn.prepareStatement("SELECT \* FROM plants WHERE image IS NOT NULL")) {

ResultSet rs = ps.executeQuery();

while (rs.next()) {

byte[] imageData = rs.getBytes("image");

if (imageData != null) {

Image dbImage = ImageIO.read(new ByteArrayInputStream(imageData));

Image searchImage = ImageIO.read(imageFile);

// Compare the images (simple comparison, you can improve with better algorithms)

if (compareImages(dbImage, searchImage)) {

int id = rs.getInt("id");

String plantName = rs.getString("name");

String plantDescription = rs.getString("description");

userTableModel.setRowCount(0); // Clear existing data

userTableModel.addRow(new Object[]{id, plantName, plantDescription});

return;

}

}

}

JOptionPane.showMessageDialog(this, "No matching plant found");

} catch (SQLException | IOException ex) {

ex.printStackTrace();

}

}

}

private boolean compareImages(Image imgA, Image imgB) {

// Implement a simple comparison, you can improve this with a better algorithm

return imgA.getWidth(null) == imgB.getWidth(null) && imgA.getHeight(null) == imgB.getHeight(null);

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

public void run() {

new MedicinalPlantsApp().setVisible(true);

}

});

}

}

class BackgroundPanel extends JPanel {

private Image image;

private float alpha;

public BackgroundPanel(Image image, float alpha) {

this.image = image;

this.alpha = alpha;

}

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

Graphics2D g2d = (Graphics2D) g.create();

g2d.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC\_OVER, alpha));

g2d.drawImage(image, 0, 0, getWidth(), getHeight(), this);

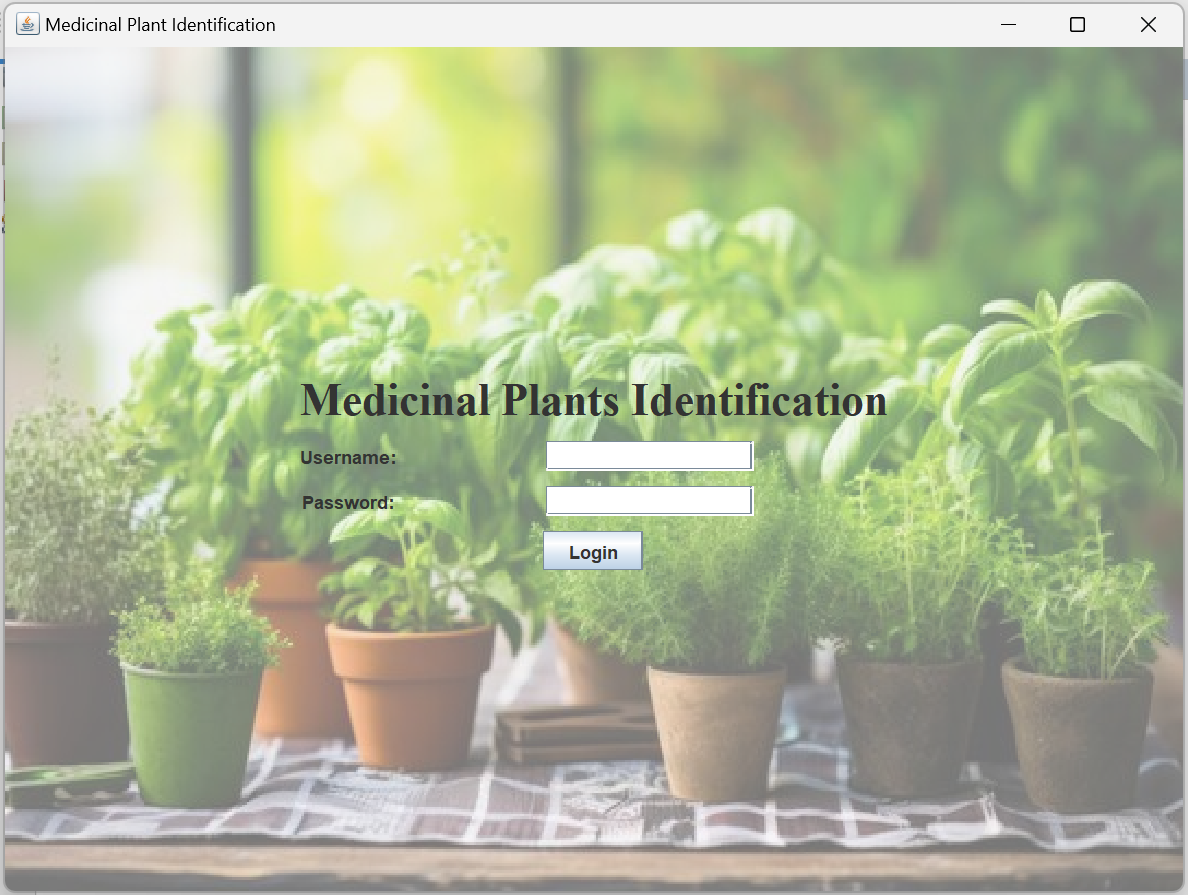
g2d.dispose();

}

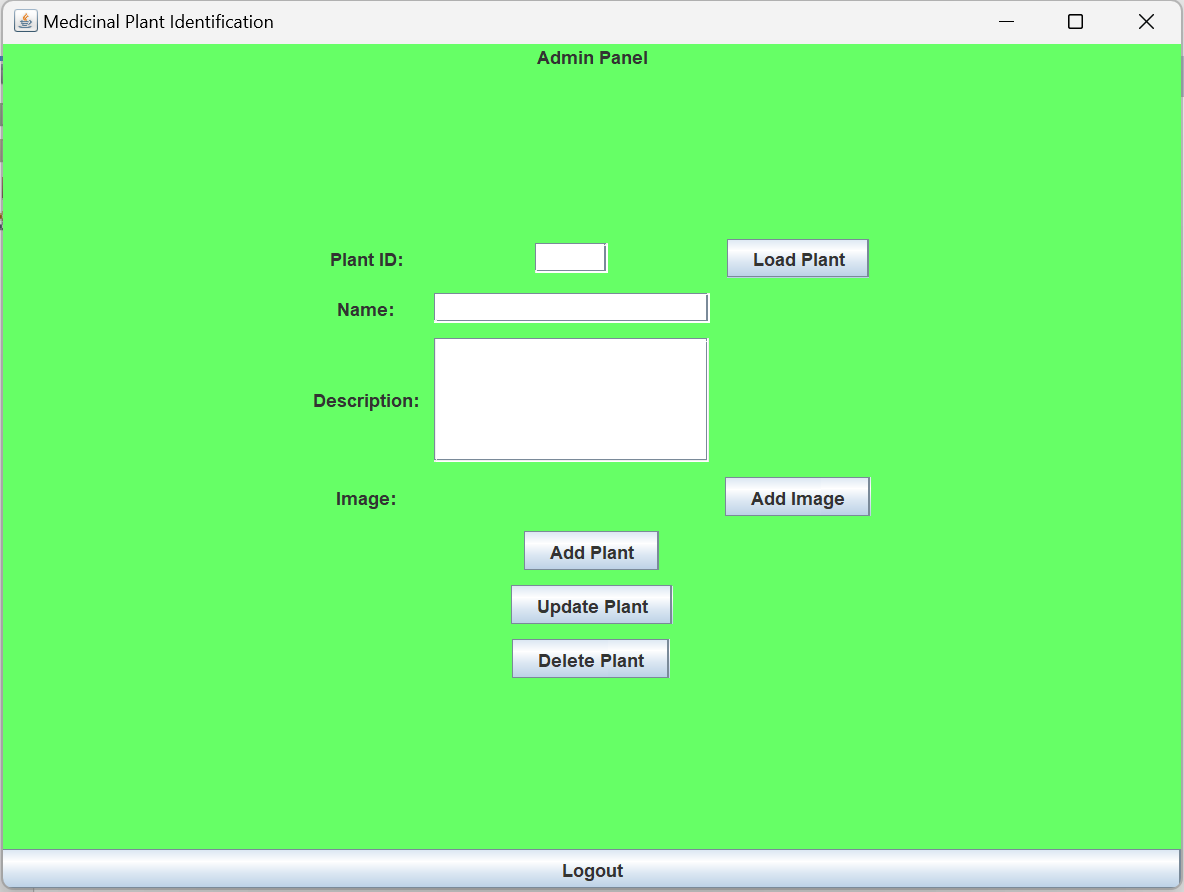
}

4. **SCREENSHOTS**

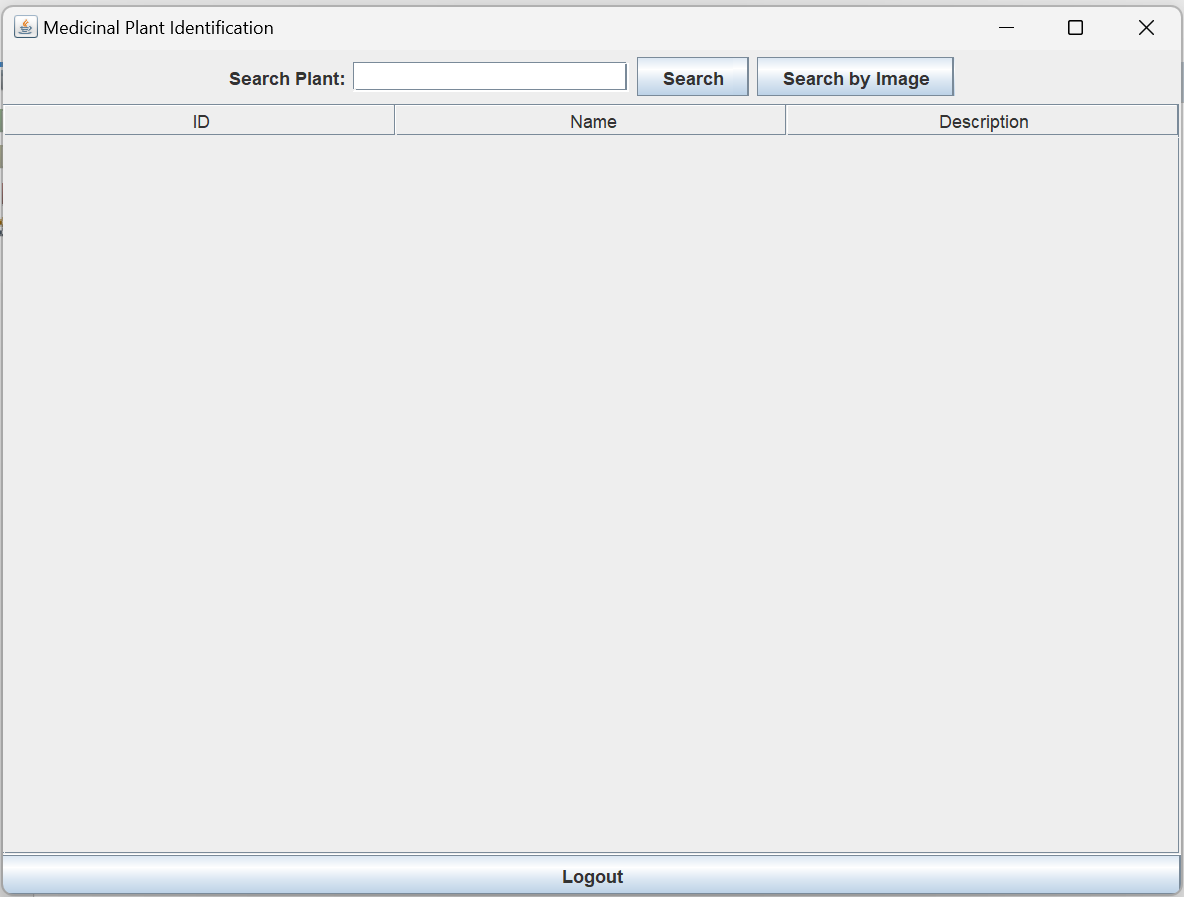
1. Login Page:-

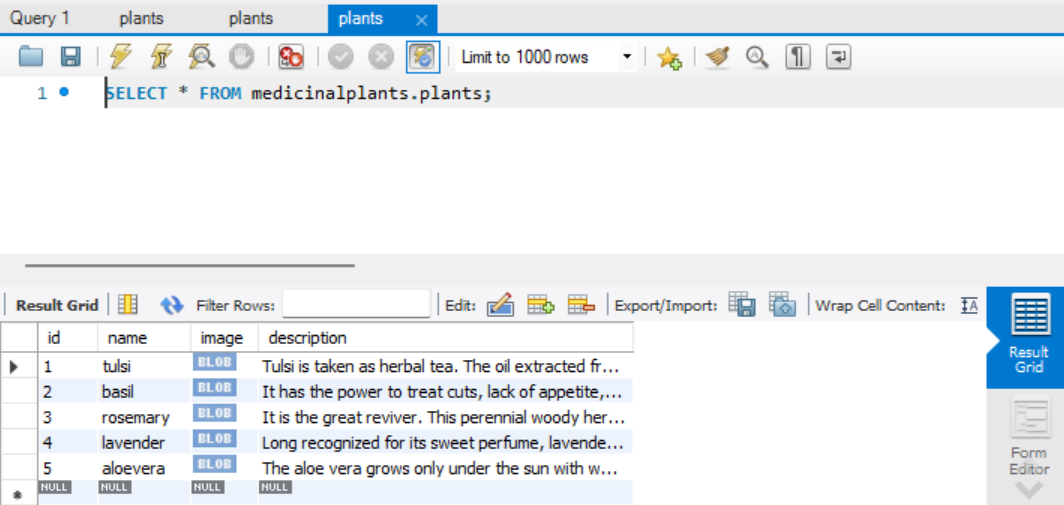


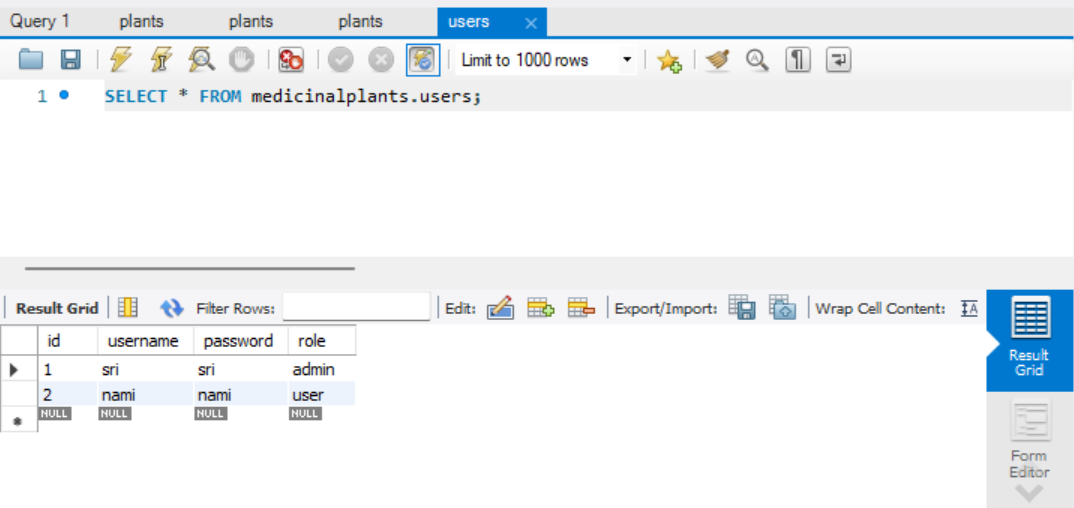
1. Admin Page:-



1. User Page:-







5.**CONCLUSION**

In conclusion, the development of this image classification system represents a significant milestone in the integration of modern technology with traditional Ayurvedic practices. By leveraging the power of advanced image processing techniques and machine learning algorithms, this project addresses the long-standing challenge of accurate identification of medicinal plants and raw materials, a critical aspect of maintaining the quality and efficacy of Ayurvedic medicines.

Moreover, the system's emphasis on continuous improvement, scalability, and integration with domain experts ensures its relevance and adaptability to the evolving needs of the industry. This project exemplifies the synergy between traditional knowledge and modern technological advancements, fostering increased trust, sustainability, and growth within the Ayurvedic healthcare ecosystem while preserving the rich heritage and holistic principles that underpin this ancient healing tradition.

The successful implementation of this image classification system has the potential to revolutionize the identification process, mitigating the risks associated with misidentification, promoting sustainable utilization of medicinal plant resources, and enhancing the overall credibility and acceptance of Ayurvedic medicine. By bridging the gap between ancient wisdom and contemporary innovation, this project paves the way for a future where traditional healing practices are supported and enriched by cutting-edge technologies, ensuring their continued relevance and impact on human well-being.