Ayush Virtual Herbal Garden: An interactive Guide to medicinal plants and ayurvedic remedies with AI powered plant identification

Submitted for partial fulfilment of the requirements
for the award of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING – ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

by

Narra Purandheswari	**-**	**21BQ1A42D3**
Thella Varshitha	**-**	**21BQ1A42H3**
Nunna Anirudh	**-**	**21BQ1A42D4**
Nandigama Kartikeya	**-**	**21BQ1A42D2**

Under the guidance of

Dr. K. Suresh Babu

Professor & HOD

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING – ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Permanently Affiliated to JNTU Kakinada, Approved by AICTE

Accredited by NAAC with 'A' Grade, ISO 9001:2008 Certified

NAMBUR (V), PEDAKAKANI (M), GUNTUR – 522 508

Telno:0863-2118036, url:www.vvitguntur.com

March-April 2025

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Permanently Affiliated to JNTUK, Kakinada, Approved by AICTE
Accredited by NAAC with 'A' Grade, ISO 9001:20008 Certified
Nambur, Pedakakani (M), Guntur (Gt) -522508

DEPARTMENT OF CSE-ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

CERTIFICATE

This is to certify that this **Project Report** is the bonafide work of **Ms. Narra Purandheswari, Ms. Thella Varshitha, Mr. Nunna Anirudh, Mr. Nandigama Kartikeya**, bearing Reg. No. **21BQ1A42D3, 21BQ1A42H3, 21BQ1A42D4, 21BQ1A42D2** respectively who had carried out the project entitled **"Ayush virtual herbal Garden: An interactive Guide to medicinal plants and ayurvedic remedies with AI powered plant identification"** under our supervision.

Project Guide	**Head of the Department**
(Dr. K. Suresh Babu, Professor)	(Dr. K. Suresh Babu, Professor)

Submitted for Viva Voce Examination held on ______________________________

Internal Examiner **External Examiner**

DECLARATION

We, Ms. Narra Purandheswari, Ms. Thella Varshitha, Mr. Nunna Anirudh, Mr. Nandigama Kartikeya, hereby declare that the Project Report entitled **"Ayush virtual herbal Garden: An interactive Guide to medicinal plants and ayurvedic remedies with AI powered plant identification"** done by us under the guidance of Dr. K. Suresh Babu at Vasireddy Venkatadri Institute of Technology is submitted for partial fulfilment of the requirements for the award of Bachelor of Technology in Computer Science Engineering - Artificial Intelligence & Machine Learning. The results embodied in this report have not been submitted to any other University for the award of any degree.

DATE :

PLACE : Nambur

SIGNATURE OF THE CANDIDATE (S)

Narra Purandheswari,

Thella Varshitha,

Nunna Anirudh,

Nandigama Kartikeya.

ACKNOWLEDGMENT

We take this opportunity to express my deepest gratitude and appreciation to all those people who made this project work easier with words of encouragement, motivation, discipline, and faith by offering different places to look to expand my ideas and helped me towards the successful completion of this project work.

First and foremost, we express my deep gratitude to **Mr. Vasireddy Vidya Sagar**, Chairman, Vasireddy Venkatadri Institute of Technology for providing necessary facilities throughout the B.Tech programme.

We express my sincere thanks to **Dr. Y. Mallikarjuna Reddy**, Principal, Vasireddy Venkatadri Institute of Technology for his constant support and cooperation throughout the B.Tech programme.

We express my sincere gratitude to **Dr. K. Suresh Babu**, Professor & HOD, Computer Science & Engineering (AI&ML), Vasireddy Venkatadri Institute of Technology for his constant encouragement, motivation and faith by offering different places to look to expand my ideas.

We would like to express my sincere gratefulness to our Guide **Dr. K. Suresh Babu**, Professor & HOD, Computer Science & Engineering (AI&ML), for his insightful advice, motivating suggestions, invaluable guidance, help and support in successful completion of this project.

We would like to express our sincere heartfelt thanks to our Project Coordinator **Mrs. K. Deepika**, Assistant Professor, CSE-Artificial Intelligence & Machine Learning for her valuable advices, motivating suggestions, moral support, help and coordination among us in successful completion of this project.

We would like to take this opportunity to express my thanks to the **Teaching and Non-Teaching** Staff in the Department of Computer Science & Engineering, VVIT for their invaluable help and support.

Name of the student(s)

Narra Purandheswari
Thella Varshitha
Nunna Anirudh
Nandigama Kartikeya

Table of Contents

List of Figures

Nomenclature

AYUSH	Ayurveda, Yoga, Unani, Siddha and Homeopathy
AI	Artificial Intelligence
VR	Virtual Reality
ResNet	Residual Neural Network
TTS	Text-to-speech
API	Application Programming Interface
QR Code	Quick Response Code

Abstract

The AYUSH Virtual Herbal Garden offers an innovative and practical solution for identifying Ayurvedic medicinal plants through AI-powered technology, seamlessly integrating traditional knowledge into modern healthcare practices. The system leverages a dataset of approximately 2,000 images spanning 30 species of medicinal leaves, utilizing Convolutional Neural Networks (CNN) to deliver high classification accuracy. Enhanced by Transfer Learning, the plant identification process ensures efficiency and precision, making it a reliable tool for users. The Flask-based web interface serves as the backbone of the system, providing a user-friendly platform that delivers comprehensive plant information—including scientific names, medicinal uses, and geographical details—alongside advanced interactive features.Key functionalities include a Disease and Remedy Module, which allows users to input diseases and receive tailored Ayurvedic remedy suggestions based on a predefined knowledge base of medicinal plants. The Prediction Module employs AI to identify plant species from uploaded images, offering detailed scientific and geographical insights. Extending its scope, the Herbal Teas Module enriches the system by providing users with herbal tea recipes and their associated health benefits, drawing from the medicinal properties of identified plants. Additionally, the Virtual Garden Module introduces an immersive experience, utilizing A-Frame to create a full-screen 3D virtual reality (VR) environment where users can explore plants from Ayurveda, Yoga, Unani, Siddha, and Homeopathy categories. In this VR space, each plant is represented with an image and text displaying its name and description, enhancing user engagement through interactive navigation.

Keywords:

AYUSH, Medicinal Plant Identification, Transfer Learning, Virtual Reality, Ayurvedic Remedies, Traditional Medicine

CHAPTER - 1
INTRODUCTION

1.1 Problem Statement

The AYUSH sector, which encompasses Ayurveda, Yoga, Unani, Siddha, and Homeopathy, relies heavily on medicinal plants for traditional healing practices. However, access to physical herbal gardens is often restricted due to geographical limitations, making it difficult for individuals to explore and learn about these valuable plants. Additionally, identifying medicinal plants correctly requires specialized botanical knowledge, which is not commonly available to students, general practitioners, or enthusiasts. This lack of accessibility and expertise results in misidentification, which can lead to incorrect usage, potential health hazards, and a decline in trust in traditional remedies.

Existing solutions, such as books and online databases, provide theoretical knowledge but fail to offer an interactive and immersive learning experience. Many plant identification tools available today cater to general botanical studies but do not specifically focus on AYUSH medicinal plants, making them less effective for users who require targeted information. There is a pressing need for an innovative digital solution that simplifies plant identification, provides authentic AYUSH-based medicinal knowledge, and engages users in an interactive and immersive learning environment. This platform should bridge the gap between traditional knowledge and modern technology, ensuring the preservation and widespread accessibility of herbal wisdom.

1.2 Objective

The Virtual Herbal Garden aims to bridge the knowledge gap in AYUSH-based traditional medicine by providing an interactive, educational, and immersive digital experience. The platform will offer users a realistic 3D virtual tour, allowing them to explore medicinal plants in a visually engaging way. By integrating AI-powered plant identification, users can upload images of plants and receive accurate identifications along with their medicinal properties and traditional uses.

Additionally, the system will provide personalized herbal medicine recommendations based on users' health concerns, ensuring that natural remedies are easily accessible to those interested in holistic healing. To further promote wellness and preventive healthcare, the platform will suggest yoga postures that align with Ayurvedic treatment principles. By incorporating voice-based search, text-to-speech conversion, and multi-language support, the Virtual Herbal Garden will enhance accessibility, making it inclusive for users of different backgrounds and abilities. Through this fusion of technology and traditional wisdom, the project will encourage a deeper understanding of natural healing practices and sustainable healthcare solutions.

1.3 Scope

The Virtual Herbal Garden will serve as a comprehensive digital platform designed to educate and engage users with medicinal plants and their applications. Users will be able to navigate a 3D virtual garden, interact with plants, and access detailed information about their scientific classification, benefits, and traditional uses. The platform will include an AI-driven plant identification feature, allowing users to upload plant images and obtain reliable insights into their medicinal properties.

Beyond identification, the system will act as a holistic wellness guide, offering personalized recommendations for herbal remedies and yoga postures to address various health concerns. To improve usability, the platform will support search and filter options, voice-based search, text-to-speech features, and multi-language translation, ensuring accessibility for a diverse audience. Additionally, users can generate and download QR codes for plant profiles, facilitating easy sharing and quick reference. The platform will also include a Herbal Teas Module, educating users on the benefits and preparation of traditional herbal drinks for better health.

With its blend of interactive experiences, AI-powered identification, and holistic wellness integration, the Virtual Herbal Garden will be a valuable educational resource for students, practitioners, and health enthusiasts. In the future, it can be expanded with augmented reality (AR) features, expert consultations, and an extended herbal database, further enhancing its impact on natural medicine and healthcare awareness.

1.4 Methodology Overview

The proposed system follows an Agile methodology, structured into six 2-week sprints over a 12-week timeline to ensure systematic and incremental development. Agile development allows flexibility, continuous user feedback, and iterative improvements, making it the ideal choice for this project. Each sprint was designed to deliver a functional component of the system, allowing for adjustments based on feedback received in sprint reviews.

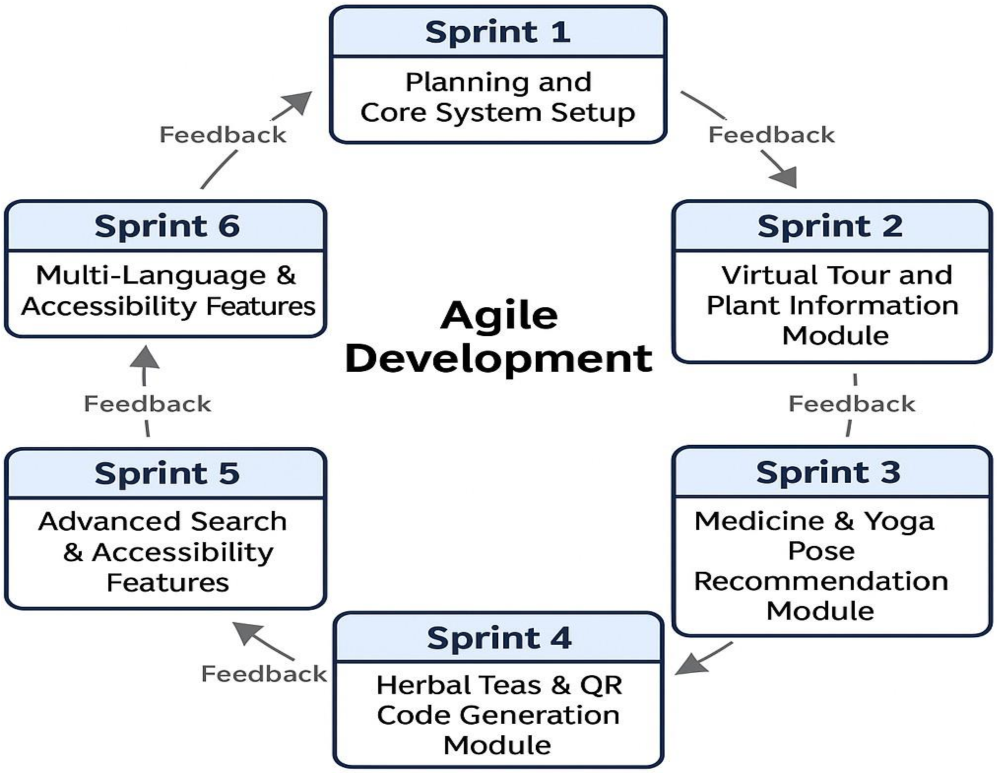


fig 1.1 Agile Methodology

Sprint 1: Planning and Core System Setup

1. The project began with defining the scope, objectives, and features to be implemented. The goal was to establish a clear roadmap for development while ensuring the system effectively promotes AYUSH practices.
2. A key focus was designing an intuitive user interface that encourages engagement and enhances awareness of Ayurveda, Yoga, and traditional medicine.
3. Flask was chosen as the backend framework due to its lightweight nature and flexibility, allowing for easy integration of AI models.
4. AI-driven medicinal plant identification was implemented using EfficientNet and ResNet models, trained on a dataset of 2,000 images representing 30 AYUSH medicinal plant species.
5. Core functionalities were validated to ensure the AI model could accurately classify medicinal plants based on image recognition.
6. User input mechanisms were established to allow feedback collection, forming the basis for iterative improvements in future sprints.

Sprint 2: Virtual Tour and Plant Information Module

1. A 3D virtual tour was developed using A-Frame, providing users with an immersive experience to explore different medicinal plants.
2. High-resolution images of plants were integrated along with detailed information, including their scientific names, medicinal benefits, and traditional applications.
3. The navigation structure was refined to ensure users could easily explore the plants, with features such as zoom.
4. Usability testing was conducted with initial users, who provided feedback on interface design, responsiveness, and ease of use.
5. Performance optimization techniques were implemented to ensure smooth loading times, especially for VR elements, making the platform accessible across different devices.

Sprint 3: Medicine & Yoga Pose Recommendation Module

1. The Ayurvedic medicine recommendation module was developed to suggest herbal treatments based on user-input symptoms and health conditions.
2. Data was sourced from classical Ayurvedic texts and modern research to ensure the recommendations were medically relevant and scientifically backed.
3. The system was designed to retrieve pre-stored images and details from the database, ensuring that users received accurate information based on their health conditions.
4. The Yoga pose recommendation module was developed to provide personalized yoga suggestions based on user health conditions and wellness goals.
5. Yoga pose recommendations were categorized into three difficulty levels: Beginner, Moderate, and Advanced, allowing users to choose suitable exercises.
6. An early prototype was tested by selected users, and their feedback was incorporated to fine-tune the recommendation logic and UI/UX experience.

Sprint 4: Herbal Teas & QR Code Generation Module

1. A specialized herbal teas module was introduced, providing insights into various herbal tea recipes, their health benefits, and preparation techniques.
2. The module was designed to categorize herbal teas based on their effects, such as immunity-boosting, digestion-improving, and stress-relieving properties.

3. QR code generation was added, allowing users to easily download plant-related information and medicine recommendations for offline use.
4. User feedback was incorporated to refine the interface, accessibility, and categorization of herbal tea options.

Sprint 5: Advanced Search & Accessibility Features

1. A voice-based search functionality was developed to enhance accessibility, allowing users to search for information via voice commands instead of typing.
2. Text-to-speech (TTS) was integrated to improve inclusivity, enabling visually impaired users to listen to plant descriptions and recommendations.
3. An advanced search and filter system was implemented, enabling users to sort information based on criteria such as plant category, medicinal properties, symptoms, and traditional usage.
4. Rigorous usability testing was conducted to refine search functionality, ensuring accurate and relevant results.
5. Performance improvements were made to optimize search speed, making information retrieval efficient and seamless for users.

Sprint 6: Multi-Language Support and Final Enhancements

1. A multi-language translation system was implemented, allowing users to access content in multiple languages to cater to a diverse audience.
2. A user feedback system was developed to collect valuable insights from users, enabling continuous improvement and enhancement of system features.
3. Final rounds of bug fixes and performance optimizations were conducted, ensuring the platform was stable and user-friendly.
4. Final usability tests confirmed that the system provided a seamless, informative, and engaging experience aligned with AYUSH principles.

CHAPTER - 2

LITERATURE REVIEW

Kavitha et al., 2023 proposed a real-time medicinal plant identification system using the MobileNet deep learning model. The study focused on identifying six species: betel, curry, tulsi, mint, neem, and Indian beech using 500 images per species from the Kaggle database. After preprocessing through resizing and data augmentation, the model achieved an accuracy of 98.33%. The system was integrated into a mobile application, offering users real-time plant identification and medicinal benefit information. The system's primary limitation was its restriction to six plant species and its dependency on image quality for accurate predictions.

Ambarwari et al., 2020 aimed to identify plant species using leaf venation features. Binary images about leaf venation were extracted using leaf image segmentation, which allowed for the determination of branching and ending points. Calculations were made for skeleton length, number of categories, total skeleton length, number of branching points, ending points, straightness, direction, length ratio, and scale projection. Leaf veneration extraction produced 19 features. A support vector machine (SVM) with an RBF kernel was used to identify plant species; a learning model was constructed using 75% of the training data. According to test results, the accuracy was 82.67%, the recall was 83%, and the average precision was 84%.

Azadnia et al., 2022 developed an automatic CNN to propose an intelligent vision-based system to identify herb plants. A CNN block for extraction of features and the classifier block for feature classification make up the suggested DL model. A Global Average Pooling (GAP) layer, a dense layer, a dropout layer, and a SoftMax layer are all part of the classifier block. Three levels of image definitions (64 × 64, 128 × 128, and 256 × 256 pixels) have been used to test the solution for identifying the leaves of five distinct medicinal plants. Consequently, for every image, therefore, the vision-based system obtained an accuracy of over 99.3%. As a result, the suggested approach can effectively replace conventional methods for real-time medicinal plant identification.

Vanmore et al., 2024 introduced the Virtual Herbal Garden, an interactive platform for exploring medicinal plants in the AYUSH sector. Utilizing 3D models and multimedia elements, the platform presented plant data, including scientific and local names, cultivation techniques, and medicinal applications. High-definition images, videos, and audio tours provided an immersive experience.

Features like search tools and annotations enhanced user engagement. Future improvements involve incorporating augmented reality, virtual reality, and AI-based search functionality for deeper interactivity.

D.S. Prasvita, Y. Herdiyeni introduced MedLeaf, a mobile application for Android that identifies medicinal plants using leaf images, employing a methodology that preprocesses smartphone-captured images, extracts texture features with Local Binary Pattern Variance (LBPV), and classifies them using a Probabilistic Neural Network (PNN); the dataset includes 1,440 leaf images from 30 Indonesian medicinal plant species, achieving 56.33% accuracy; limitations include low accuracy due to variable image quality and similar textures among species, restricting its reliability and scope.

2.1 Existing System:

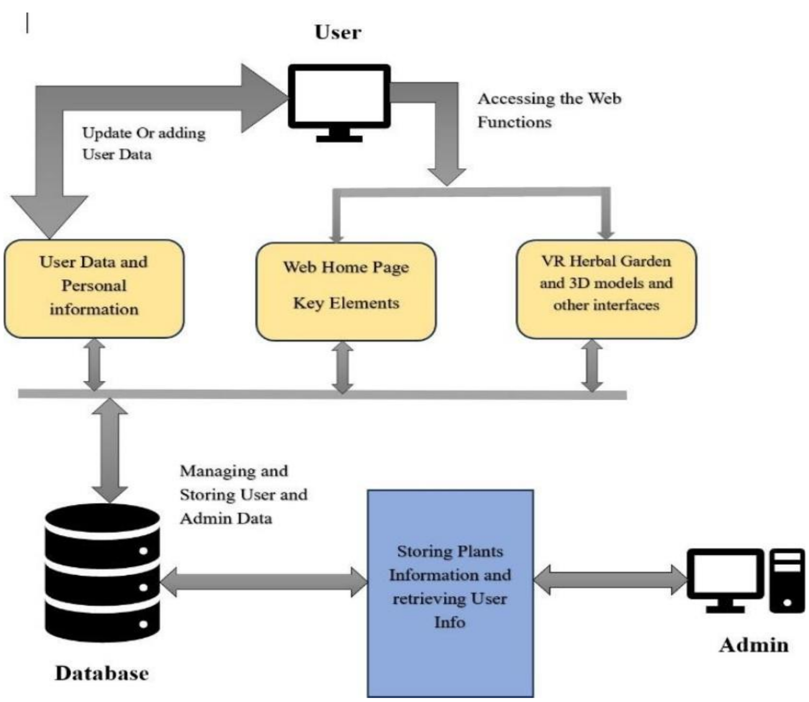


fig 2.1 System Architecture

The Existing System of the AYUSH Virtual Herbal Garden is a comprehensive web-based platform designed to educate users about AYUSH medicinal plants. Built with a frontend utilizing HTML, CSS, and JavaScript, and a backend powered by Node.js and MongoDB, the system offers a seamless and interactive learning experience. The architecture is structured to ensure efficient user engagement and data management. It comprises an admin module that facilitates user validation, ensuring secure access and appropriate content management. The user module enables profile management and virtual exploration, allowing individuals to navigate the digital garden effortlessly.

A key component of the system is the Herbal Plants module, which provides extensive details on various medicinal plants, including their scientific and local names, medicinal uses, and propagation methods. Additionally, the platform features a tab, offering valuable medicinal insights for users interested in alternative healing practices. Registered users can log in to access a rich repository of plant data, accompanied by multimedia content, enhancing their understanding of AYUSH herbal medicine. With an intuitive interface and well-organized content, the platform ensures an engaging and informative experience for learners, researchers and enthusiasts alike.

The Existing system of real-time identification of medicinal plants involves six plants such as Betel, Curry, Tulsi, Mint, Neem, and Indian Beech—using the MobileNet deep learning model. The existing system's architecture is designed in three key stages to ensure accurate identification of medicinal plants. First, the MobileNet model is trained over 15 cycles (epochs) to improve its learning. During this stage, 350 images per plant species are used for training, helping the model recognize patterns effectively. Next, 100 images per species are set aside for validation, which helps fine-tune the model and prevent errors. Finally, the last stage involves testing the model with 50 images per species to check how well it performs on new, unseen data. Through this process, the model is optimized, and after extensive testing, it achieves an impressive 98.33% accuracy, ensuring reliable plant identification for users.

Limitations of Existing Systems

Current methods for exploring and identifying AYUSH medicinal plants rely heavily on physical gardens or human expertise, which are inconsistent and inaccessible to many. While some digital tools use Convolutional Neural Networks (CNNs) to classify plant species based on leaf images, they lack the immersive and educational depth needed for AYUSH-specific learning. These systems face several limitations:

1. Limited Accessibility:

- Physical gardens and expert-led identification remain inaccessible to remote or non-specialist users.

- The CNN-based system (MobileNet) is limited to six species and requires cloud connectivity, challenging for users without reliable internet.
- The Virtual Herbal Garden's prototype lacks VR/AR, failing to fully replace the need for physical garden access.

2. Scalability Issues:

- The CNN tool's dataset of 3,000 images for six species restricts its ability to expand to more plants or adapt to diverse conditions.
- MobileNet's lightweight design limits its capacity for complex tasks, requiring significant resources for scaling.
- The Virtual Herbal Garden's incomplete database and unimplemented features hinder its potential to cover a wide range of plants.

3. Lack of Engagement:

- The CNN system provides basic identification (plant name and benefits) without interactive or multimedia-rich experiences.
- The Virtual Herbal Garden, despite educational goals, lacks 3D/VR immersion in its current prototype, reducing user engagement.

4. Insufficient Context:

- The CNN-based tool focuses solely on identification, omitting comprehensive AYUSH-specific details like medicinal uses or remedies.
- The Virtual Herbal Garden, while aiming to educate, doesn't yet offer detailed Ayurvedic insights or practical applications in its present form.

2.2 Gap Analysis

The existing systems for AYUSH medicinal plant exploration—the web-based Virtual Herbal Garden and the CNN-based real-time identification system using MobileNet—lack the depth and functionality needed for comprehensive AYUSH learning. The Virtual Herbal Garden, built with HTML, CSS, JavaScript, Node JS, and MongoDB, offers basic plant data but misses immersive 3D/VR features, limiting engagement. The MobileNet system, with 98.33% accuracy on images of six species, is restricted by its narrow scope, cloud dependency, and lack of AYUSH-specific medicinal context or interactivity. Both systems fail to scale to more species, offer practical features like remedy or yoga recommendations, or include accessibility tools like multi-language support and voice-based search. The Ayush Virtual Herbal Garden fills these gaps with AI-driven identification (EfficientNet, ResNet) for 30 species, a 3D VR experience, and features like medicine recommendations, multi-language translation, and a herbal teas module, ensuring a more engaging and accessible platform.

2.3 Relevance of the Project

The Ayush Virtual Herbal Garden holds significant relevance in today's context, where there is a growing global demand for natural remedies and holistic healthcare practices rooted in AYUSH traditions (Ayurveda, Yoga & Naturopathy, Unani, Siddha, and Homeopathy). With the increasing interest in sustainable and traditional medicine, particularly amidst concerns over synthetic drugs' side effects, medicinal plants are gaining prominence for their therapeutic potential. However, the inaccessibility of physical herbal gardens, the complexity of plant identification, and the lack of comprehensive AYUSH-specific educational tools create barriers for students, practitioners, and enthusiasts. This project bridges these gaps by leveraging modern technology—AI-driven identification with EfficientNet and ResNet, a Flask-based web interface, and a 3D VR environment using A-Frame—to make AYUSH knowledge widely accessible and engaging. By covering 30 plant species and offering features like medicine and yoga pose recommendations, a herbal teas module, multi-language translation, voice-based search, and QR code generation, the system not only simplifies plant identification but also promotes practical application and cultural preservation of AYUSH practices. Its relevance is further underscored by its potential to educate a global audience, empower rural practitioners with digital tools, and support the integration of traditional medicine into mainstream healthcare, aligning with global health trends and the need for inclusive, technology-driven solutions.

CHAPTER - 3

SYSTEM ANALYSIS

3.1 Functional Requirements:

Virtual Herbal Tour

- The system must provide a virtual reality (VR) environment using A-Frame to showcase medicinal plants in an immersive 3D garden. The module must support full-screen VR mode, triggered automatically upon entering from the tour page, with navigation via WASD keys and mouse controls.
- It should enable users to explore plants from categories like Ayurveda, Yoga, Unani, Siddha, and Homeopathy, with each plant displayed as an image and accompanied by fixed text (name and description) in the VR scene.

Plant Identification Module

- The system must be able to classify Ayurvedic medicinal plants using Convolutional Neural Networks (CNN).
- It should leverage a dataset of around 2,000 images from 30 species of medicinal leaves for accurate classification.
- The module should use Transfer Learning to enhance accuracy and efficiency in plant identification.

User Interface (UI) via Flask

- The system must provide a Flask-based web interface for users to interact with the plant identification module.
- The UI should display detailed plant information, including scientific name, medicinal uses, and geographical details.

Disease and Remedy Module

- The system should include a module that allows users to input symptoms or diseases.

- Based on the input, it should suggest Ayurvedic remedies using a predefined knowledge base of medicinal plants.

Prediction Module

- The module must utilize AI-powered predictions to identify plant species based on uploaded images.
- It should provide scientific and geographical details about the identified plant.

Herbal Teas Module

- The system must include a module dedicated to herbal teas, offering recipes and benefits based on Ayurvedic medicinal plants.
- The module must display preparation ingredients,instructions and health benefits for each herbal tea recipe.

Accessibility and User Engagement

- The system should ensure easy navigation and accessibility for users with varying levels of expertise.
- It must provide interactive elements such as search functionalities, plant comparison, or bookmarking of plants for future reference.

Reliability and Performance

- The AI model should achieve high classification accuracy, minimizing false predictions.
- The system should efficiently handle multiple requests and process plant identifications in real time.

Data Management

- The system must store and manage plant images, descriptions, and remedy information effectively.
- It should allow for future updates and scalability in adding new plant species and Ayurvedic remedies.

3.2 Software and Hardware Requirements:

Hardware:

- · Personal computer or laptop with sufficient RAM (8 GB or higher recommended)
- Adequate storage space for datasets and model files
- Processor capable of handling computational tasks involved in model training

Software:

- Python (programming language), Anaconda
- Jupyter Notebook (for data analysis and model development)
- Scikit-learn (machine learning library)
- Pandas (data manipulation library)
- NumPy (numerical computing library)
- Matplotlib and Seaborn (data visualization libraries)

3.3 Feasibility Study

3.3.1 Technical Feasibility

- The project uses Flask for the web application and EfficientNet and ResNet for AI-based plant identification.
- A-Frame is integrated for creating an interactive virtual herbal garden.
- The AI models are chosen for their accuracy and compatibility with the system.
- Cloud storage ensures efficient data management for images and plant information.
- Features like voice search, text-to-speech, and multilingual support are technically achievable.

3.3.2 Economic Feasibility

- The project uses open-source tools like Flask and A-Frame, reducing development costs.
- Pre-trained AI models minimize the expenses of training from scratch.
- Cloud hosting was selected for its cost-effectiveness and scalability.
- The project provides social benefits by promoting awareness of AYUSH medicinal plants.

3.3.3 Operational Feasibility

- Users like students, researchers, and AYUSH practitioners can easily access the system.
- The plant identification system provides real-time results with high accuracy.
- The virtual tour enhances user engagement through immersive experiences.
- Voice search and multi-language support ensure accessibility for diverse users.
- A feedback system allows users to provide suggestions, improving operational effectiveness.
- Admins can manage the plant database, ensuring up-to-date information.

3.4 Proposed System

Ayush Virtual Herbal Garden is an advanced solution that seamlessly integrates AI-driven plant identification with an interactive virtual platform, specifically designed to overcome the limitations of existing systems for AYUSH (Ayurveda, Yoga & Naturopathy, Unani, Siddha, and Homeopathy) medicinal plants. Leveraging Transfer Learning, the system utilizes pre-trained deep learning models such as EfficientNet and ResNet to achieve superior identification accuracy and computational efficiency, trained on a diverse dataset of 2,000 leaf images representing 30 AYUSH plant species. The Flask-based web application serves as the backbone of the system, providing a user-friendly interface that delivers comprehensive information on medicinal plants, including their botanical names, natural habitats, Ayurvedic medicinal properties, and cultivation techniques. Designed to cater to both experts and enthusiasts, the platform enhances user engagement through a 3D virtual reality (VR) environment developed using A-Frame. This immersive space allows users to navigate a digital herbal garden and interact with high-quality plant images accompanied by detailed descriptions. Additionally, the system offers features such as disease remedy suggestions based on Ayurvedic principles, multimedia content including audio guides, and a search functionality that enables users to filter plants by medicinal use or species type. This holistic approach ensures that the platform serves as a robust, educational, and practical tool for exploring AYUSH herbal knowledge.

The application is structured into five key modules, each designed to enhance accessibility and provide valuable insights. The Tour module offers a virtual garden tour, allowing users to experience an interactive herbal environment, simulating a real-world exploration. The Herbal Info module provides detailed information on 30 medicinal plant species, sourced from the dataset, including their traditional uses and health benefits. The Diseases Info module offers an extensive database on various ailments, along with natural remedies and yoga asanas aimed at alleviating their impact, promoting a holistic approach to wellness. The Herbal Prediction module integrates AI-driven technology using CNN models, enabling users to upload an image of a leaf for accurate plant identification, thereby bridging technology with traditional herbal knowledge. Lastly, the Herbal Teas module provides users with curated insights on the preparation of herbal teas that support immunity, longevity, and overall well-being. Through this structured framework, the application delivers an engaging and informative experience, seamlessly integrating traditional AYUSH wisdom with modern technology to make holistic healing more accessible and practical for a wider audience.

Advantages of Proposed System:

1. **Enhanced Accuracy**: AI ensures precise identification, even for visually similar AYUSH plants.
2. **Immersive Experience**: A 3D VR garden with images offers interactive exploration without the need for physical access.
3. **Comprehensive Knowledge**: Detailed AYUSH-specific insights, including medicinal uses and remedies, are readily available.
4. **Accessibility and Scalability**: The web-based platform and efficient Transfer Learning make it usable by a wide audience with minimal resources.

CHAPTER - 4

SYSTEM DESIGN

4.1 System Architecture:

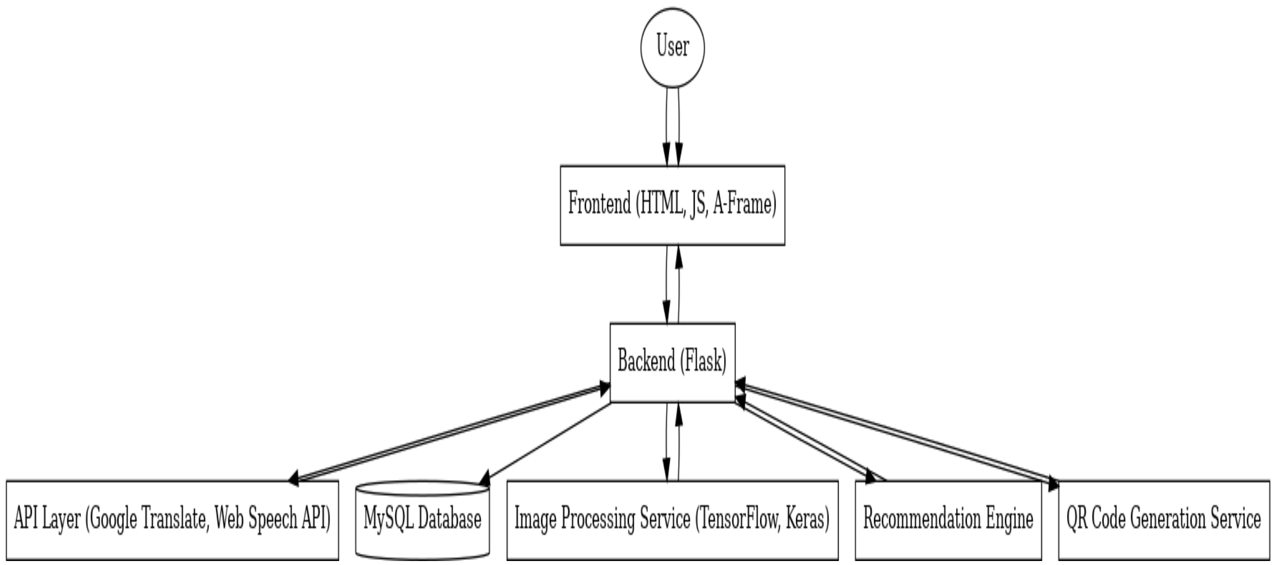


fig 4.1 Proposed System Architecture Diagram

The above fig 4.1 represents proposed system architecture. It is designed to provide an interactive and immersive experience for users exploring medicinal plants and Ayurvedic remedies. The user interacts with the frontend, built using HTML, JavaScript, and A-Frame, which renders a 3D virtual herbal garden. The frontend sends HTTP requests to the backend, developed using Flask, which handles data retrieval, plant identification, medicine recommendations, and yoga posture suggestions. The backend communicates with a MySQL database using SQLAlchemy to fetch and store plant data, user feedback, and herbal tea recipes. Users can upload plant images for identification, where the image is processed by a TensorFlow and Keras model to recognize the plant species. Additionally, the system provides personalized Ayurvedic recommendations using a plant-remedy mapping stored in the database. Features like voice-based search, text-to-speech, and multi-language translation are implemented using the Web Speech API and Google Translate API. Users can also generate and download QR codes for plant information using a QR Code Generation Service. The seamless interaction between the frontend, backend, database, and external APIs ensures an engaging user experience while promoting knowledge about traditional herbal medicine.

4.2 Block Diagram

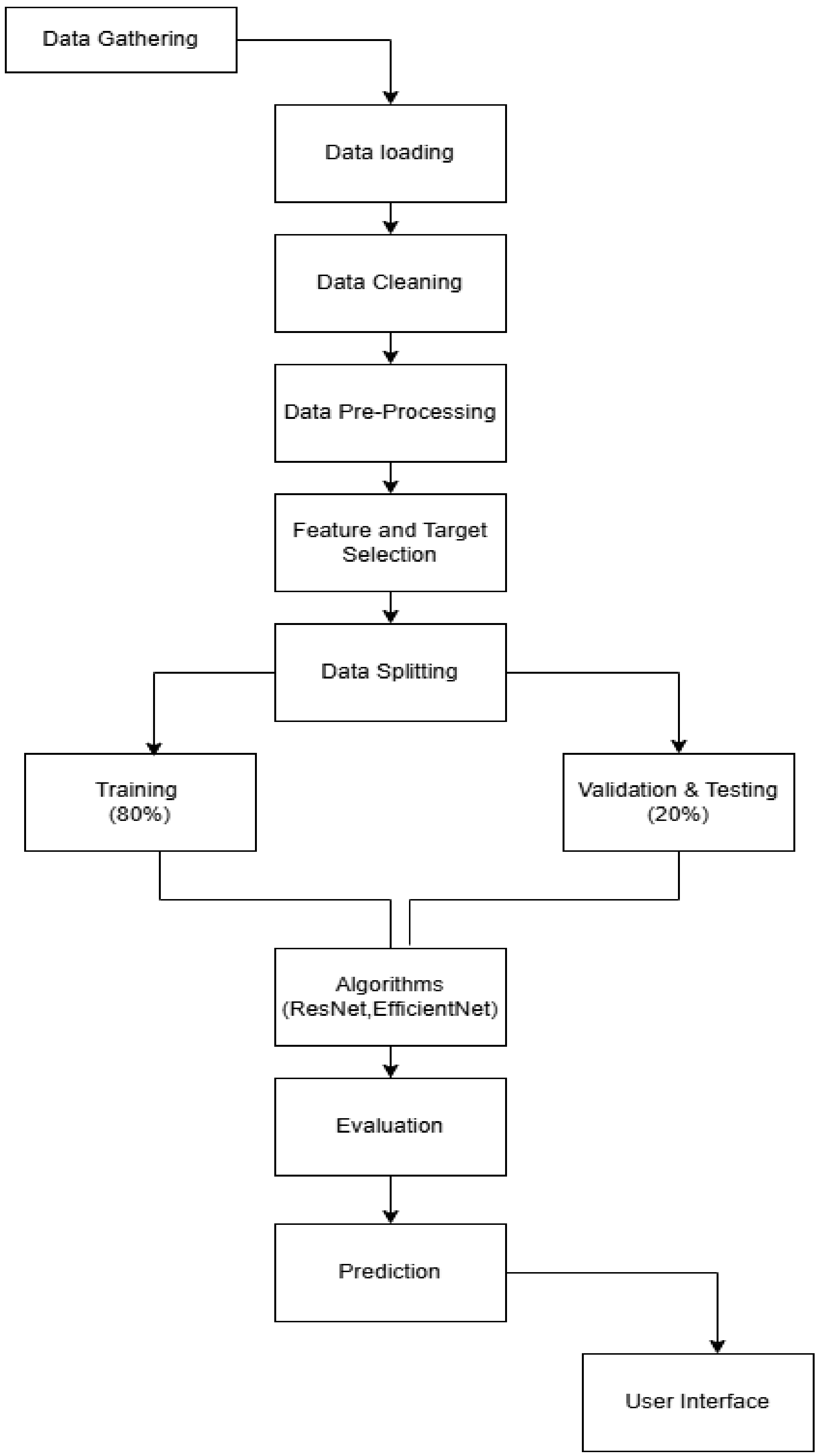


fig 4.2 System Block Diagram

4.3 ER Diagram:

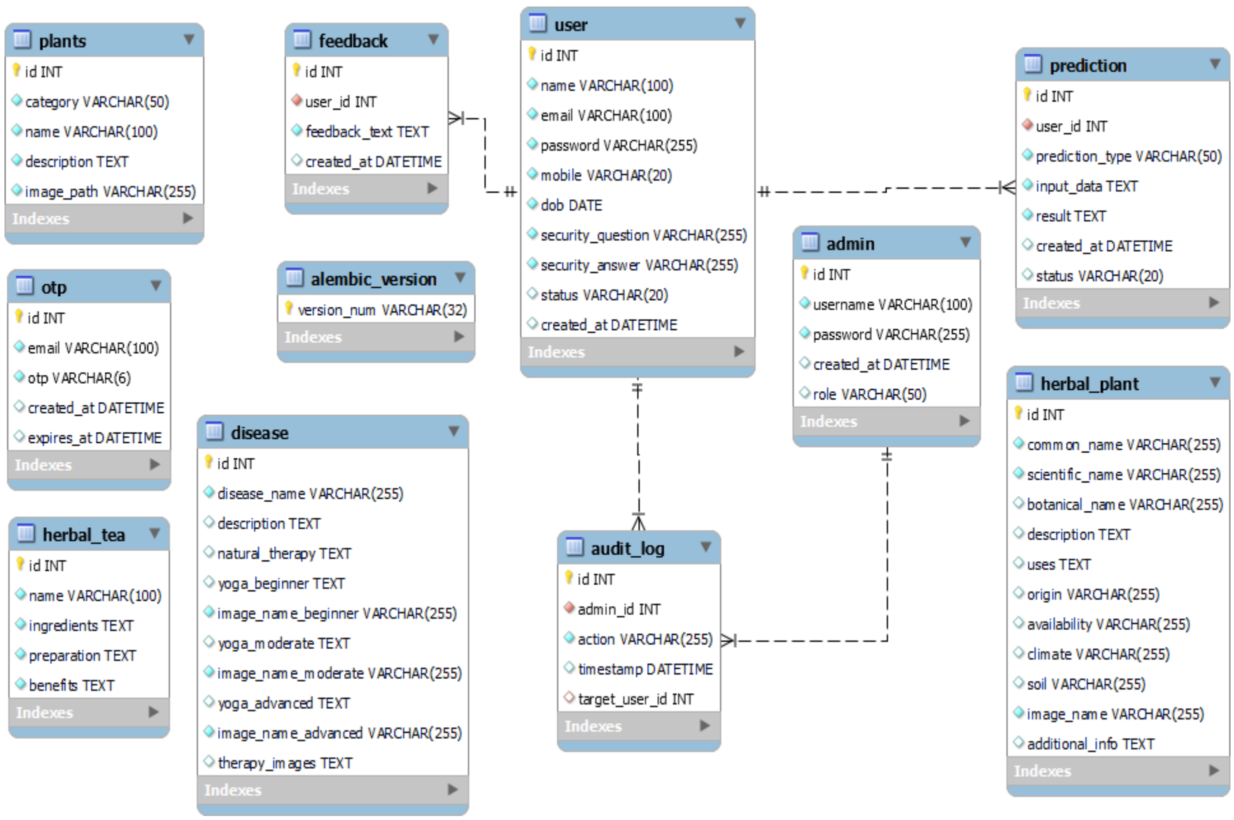


fig 4.3 System ER Diagram

The above fig 4.3 represents ER Diagram.It represents an organized structure of entities in the system and their relationships.

1. The User entity stores user details and has a one-to-many relationship with Prediction and Feedback.
2. The Admin entity manages the system and has a one-to-many relationship with Audit_Log.
3. The Audit_Log entity tracks admin actions and is linked to Admin.
4. The OTP entity is linked to User via email for authentication purposes.
5. The Feedback entity stores user feedback and has a many-to-one relationship with User.
6. The Prediction entity records plant-related predictions and has a many-to-one relationship with Users.

7. The Alembic Version entity tracks database schema versions and is independent.
8. The Plants entity stores plant details and has no direct relationships.
9. The Herbal Plant entity contains information on medicinal plants and is not linked to other tables.
10. The Herbal Tea entity records herbal tea details and has no direct relationships.
11. The Disease entity stores disease-related information and is not directly connected to any other tables.

Relationships Between Tables

User & Prediction → One-to-Many

1. user_id in Prediction references id in User.
2. User (1) → (∞) Prediction

User & Feedback → One-to-Many

1. user_id in Feedback references id in User.
2. User (1) → (∞) Feedback

Admin & Audit_Log → One-to-Many

1. admin_id in Audit_Log references id in Admin.
2. Admin (1) → (∞) Audit_Log

User & OTP → One-to-One (via Email)

1. email in OTP indirectly links to email in User.
2. User (1) → (1) OTP

4.4 Class Diagram

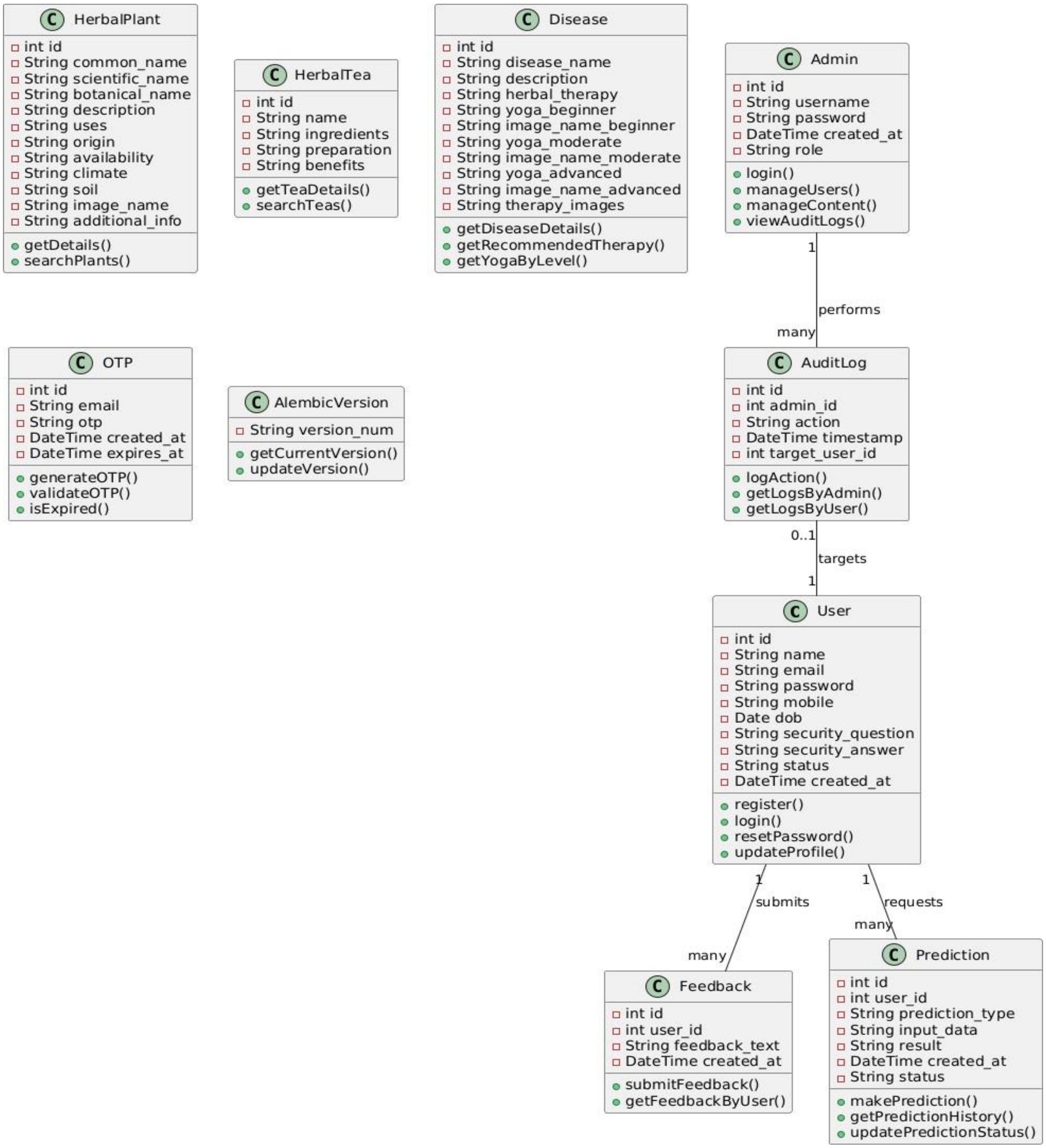


fig 4.4 Class Diagram

1. Classes and Their Attributes & Methods

User Class

1. **Attributes:**
    a. id: Unique identifier.
    b. name: User’s name.
    c. email: User’s email address.
    d. password: Login password.
    e. mobile: User's contact number.
    f. dob: Date of birth.
    g. security_question: Security question for account recovery.
    h. security_answer: Answer to the security question.
    i. status: Status of the user (active/inactive).
    j. created_at: Timestamp of account creation.
2. **Methods:**
    a. register(): Registers a new user.
    b. login(): Handles user authentication.
    c. resetPassword(): Allows users to reset their passwords.
    d. updateProfile(): Updates user profile information.

Admin Class

1. **Attributes:**
    a. id: Unique identifier.
    b. username: Admin username.
    c. password: Admin password.
    d. created_at: Account creation timestamp.
    e. role: Admin role (e.g., super admin, content manager).
2. **Methods:**
    a. login(): Authenticates admin users.
    b. manageUsers(): Admin manages user accounts.

c. manageContent(): Admin handles content updates.
d. viewAuditLogs(): Admin can view audit logs for tracking actions.

AuditLog Class

1. Attributes:
    a. id: Unique identifier.
    b. admin_id: The admin performing an action.
    c. action: Description of the action performed.
    d. timestamp: When the action occurred.
    e. target_user_id: The affected user.
2. Methods:
    a. logAction(): Logs an admin's action.
    b. getLogsByAdmin(): Retrieves logs based on admin actions.
    c. getLogsByUser(): Fetches logs associated with a specific user.

OTP Class

1. Attributes:
    a. id: Unique identifier.
    b. email: Email for OTP validation.
    c. created_at: Timestamp of OTP generation.
    d. expires_at: When the OTP expires.
2. Methods:
    a. generateOTP(): Creates a new OTP.
    b. validateOTP(): Checks if the OTP is correct.
    c. isExpired(): Verifies OTP expiration.

Prediction Class

1. Attributes:
    a. id: Unique identifier.
    b. user_id: User who made the prediction.

c. prediction_type: Type of prediction (e.g., disease diagnosis, herbal suggestion).
d. input_data: Input data for prediction.
e. result: Result of the prediction.
f. created_at: Timestamp of prediction request.
g. status: Status of the prediction (pending, completed, etc.).

2. Methods:
   a. makePrediction(): Performs prediction based on input.
   b. getPredictionHistory(): Retrieves a user’s past predictions.
   c. updatePredictionStatus(): Updates the status of a prediction.

Feedback Class

1. Attributes:
   a. id: Unique identifier.
   b. user_id: User who submitted the feedback.
   c. feedback_text: Content of the feedback.
   d. created_at: Timestamp of submission.
2. Methods:
   a. submitFeedback(): Saves user feedback.
   b. getFeedbackByUser(): Retrieves feedback submitted by a specific user.

Disease Class

1. Attributes:
   a. id: Unique identifier.
   b. disease_name: Name of the disease.
   c. description: Details about the disease.
   d. herbal_therapy: Suggested herbal treatments.
   e. yoga_beginner: Beginner-level yoga recommendations.
   f. yoga_moderate: Moderate-level yoga recommendations.
   g. yoga_advanced: Advanced-level yoga recommendations.
   h. image_name_beginner, image_name_moderate, image_name_advanced: Image references for different yoga levels.

i. therapy_images: Visual representation of the therapy.

2. Methods:
    a. getDiseaseDetails(): Fetches disease-related data.
    b. getRecommendedTherapy(): Retrieves recommended treatments.
    c. getYogaByLevel(): Provides yoga practices based on disease severity.

HerbalPlant Class

1. Attributes:
    a. id: Unique identifier.
    b. common_name: Popular name of the plant.
    c. scientific_name: Scientific classification.
    d. botanical_name: Botanical reference.
    e. description: Explanation of the plant's properties.
    f. uses: Medicinal and therapeutic applications.
    g. origin: Where the plant is found.
    h. climate: Suitable climate for growth.
    i. soil: Soil conditions required.
    j. image_name: Image reference.
    k. additional_info: Extra details.

2. Methods:
    a. getDetails(): Fetches plant information.
    b. searchPlants(): Searches for plants by name or properties.

HerbalTea Class

1. Attributes:
    a. id: Unique identifier.
    b. name: Tea name.
    c. ingredients: Ingredients used.
    d. preparation: Instructions for making tea.
    e. benefits: Health benefits.

2. Methods:
    a. getTeaDetails(): Retrieves details about a specific herbal tea.
    b. searchTeas(): Searches for tea based on ingredients or benefits.

AlembicVersion Class

1. Attributes:
    a. version_num: The version number of the database schema.
2. Methods:
    a. getCurrentVersion(): Fetches the current schema version.
    b. updateVersion(): Updates the database schema version.

2. Relationships Between Classes

1. User & Feedback – One user can submit multiple feedback entries (User 1 → * Feedback).
2. User & Prediction – One user can make multiple predictions (User 1 → * Prediction).
3. Admin & AuditLog – An admin can perform multiple actions recorded in the audit log (Admin 1 → * AuditLog).
4. Disease & HerbalPlant – A disease may be treated with multiple herbal plants (Disease 1 → * HerbalPlant).
5. HerbalPlant & HerbalTea – Herbal plants are ingredients in herbal teas (HerbalPlant 1 → * HerbalTea).

4.5 Use Case Diagram

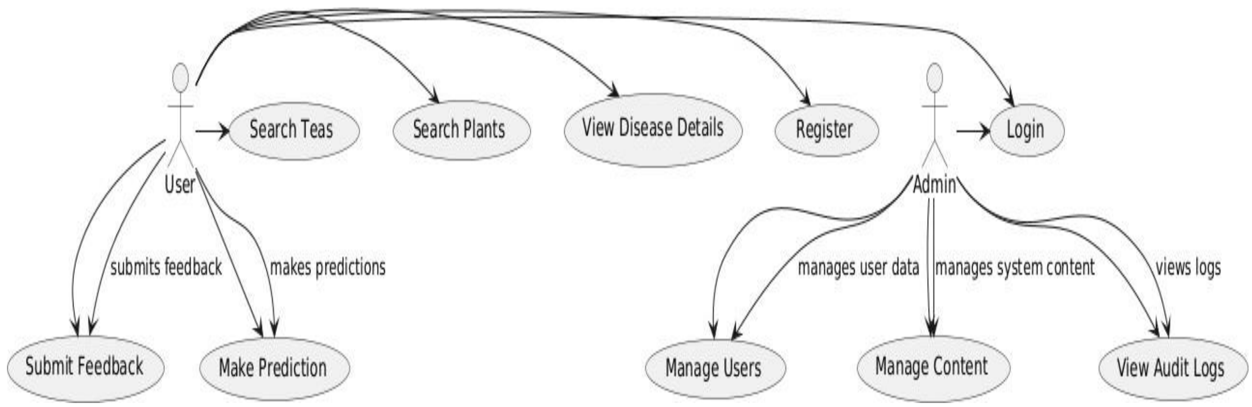


fig 4.5 Use Case Diagram

Actors:

1. User: A general user who can register, login, explore herbal plants, search for diseases, check herbal tea information, and submit feedback
2. Admin: Manages users, handles content, views audit logs, and monitors activities.

Use Cases for Users:

1. Register: Users register by providing their details like name, email, password, and mobile number.
2. Login: Users log in using their email and password.
3. Explore Herbal Plants: Users can search for detailed information about herbal plants.
4. Explore Diseases: Users can view disease-related information and suggested therapies.
5. View Herbal Teas: Users explore different herbal teas, their benefits, and preparation.
6. Submit Feedback: Users can submit feedback or suggestions about the system.

Use Cases for Admins:

1. Login: Admins login using admin credentials.
2. Manage Users: Admins can view, edit, or remove users from the system.
3. Manage Content: Admins can add, update, or delete plant, tea, or disease information.
4. View Audit Logs: Admins can view records of actions performed in the system for monitoring purposes.

4.6 Activity Diagram

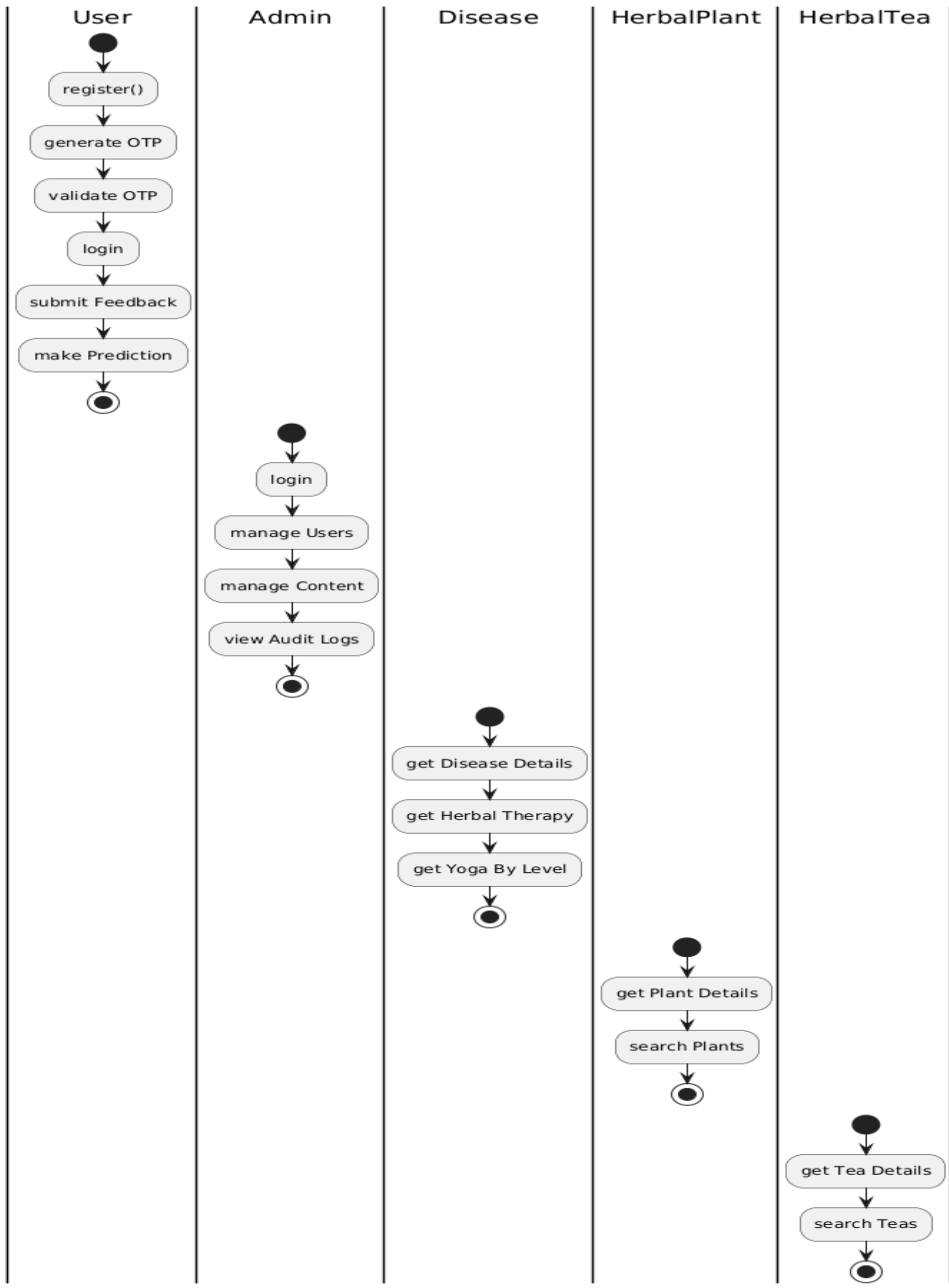


fig 4.6 Activity Diagram

The above fig 4.6 represents Activity Diagram.

Roles & Activities:

1. User:
    a. Registers an account.
    b. Generates and validates OTP for authentication.
    c. Logs into the system.
    d. Submit feedback.
    e. Uses the system to make predictions (likely related to plant identification or herbal recommendations).
2. Admin:
    a. Logs into the system.
    b. Manages users (possibly approving registrations or handling user queries).
    c. Manages content (updating plant/herbal data, adding information, or moderating discussions).
    d. Views audit logs (tracking activities for security and analytics).
3. Disease Module:
    a. Retrieves disease details (fetching information about ailments).
    b. Provides herbal therapy recommendations based on the disease.
    c. Suggests yoga practices categorized by difficulty level.
4. Herbal Plant Module:
    a. Retrieves plant details (scientific name, medicinal uses, etc.).
    b. Allows users to search for specific plants.
5. Herbal Tea Module:
    a. Retrieves details about herbal teas (ingredients, benefits, etc.).
    b. Allows users to search for teas based on preferences or health benefits.

4.7 Sequence Diagram

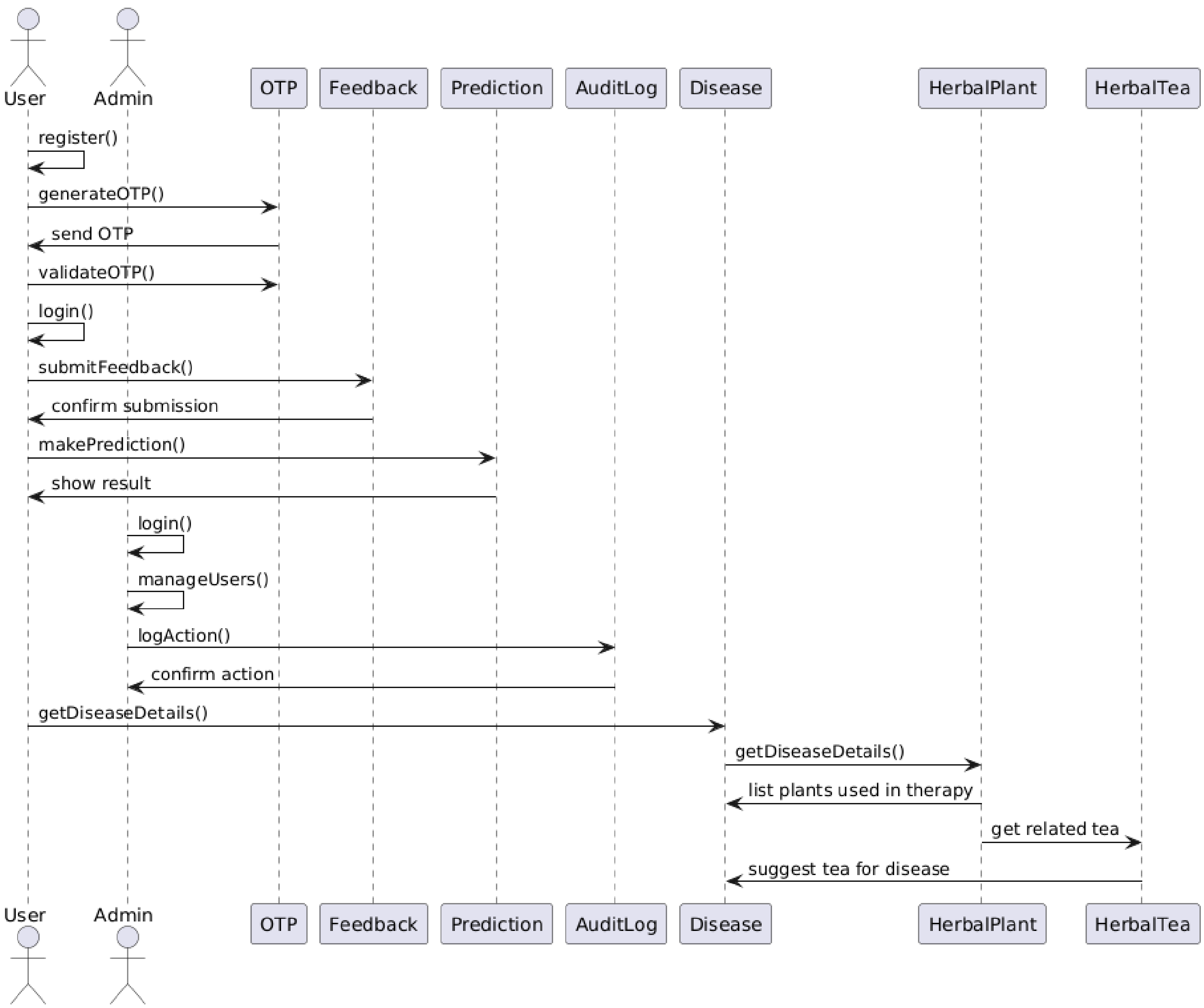


fig 4.7 Sequence Diagram

The above figure 4.7 represents a sequence diagram.

Actors Involved:

1. User – A person who interacts with the system to register, log in, submit feedback, make predictions, and get disease-related information.
2. Admin – A privileged user responsible for managing users and logging actions.

Lifelines (System Components)

1. OTP – Handles the generation, sending, and validation of OTP during registration.
2. Feedback – Manages user feedback submission.
3. Prediction – Allows users to make predictions (likely related to disease diagnosis or herbal treatment suggestions).
4. AuditLog – Records actions performed by the admin for security and tracking.
5. Disease – Provides disease-related information.
6. HerbalPlant – Manages herbal plant data used for treatment.
7. HerbalTea – Suggests related herbal tea based on disease and plant therapy.

Flow of Interactions

1. User Registration & Authentication

- register(): The user initiates registration.
- generateOTP(): The system generates an OTP and sends it.
- send OTP: OTP is sent to the user.
- validateOTP(): The system verifies the OTP before allowing registration.
- login(): The user logs into the system.

2. Feedback Submission

- submitFeedback(): The user submits feedback.
- confirm submission: The system acknowledges feedback submission.

3. Prediction Process

- makePrediction(): The user requests a prediction.
- show result: The system provides the result of the prediction.

4. Admin Actions

- login(): Admin logs into the system.
- manageUsers(): Admin manages user-related activities (e.g., verification, blocking users, etc.).

- logAction(): Admin logs an action for tracking purposes.
- confirm action: The system confirms the action has been logged.

5. Disease & Herbal Treatment Query

- getDiseaseDetails(): The user requests disease-related information.
- getDiseaseDetails(): The system retrieves disease details.
- list plants used in therapy: The system fetches herbal plants used for the therapy.
- get related tea: Herbal tea related to the disease is retrieved.
- suggest tea for disease: The system suggests herbal tea as a remedy.

CHAPTER - 5

IMPLEMENTATION

5.1 FEATURES

1. Virtual Herbal Tour

Users can explore an immersive 3D virtual environment resembling a traditional herbal garden. The garden features realistic representations of medicinal plants, providing an engaging and educational experience. Users can navigate the virtual space, click on plants to learn about their uses. A-Frame creates a VR garden with HTML and JavaScript for navigation.A-Frame entities define the garden layout, compatible with browsers.

2. Plant Information Module

This module offers comprehensive details about medicinal plants, including their names, scientific classifications, properties, benefits, and traditional uses. Users can visually explore the plants and gain insights into their significance in the AYUSH system, enriching their knowledge of natural remedies.Flask retrieves plant data from MySQL using SQLAlchemy.Jinja2 templates display plant details with images on the web interface.

3. Medicine Recommendation for Health Issues

Based on the user's input regarding their health concerns, the system recommends suitable herbal medicines and formulations derived from Ayurvedic classical texts. This feature promotes natural healing by providing personalized solutions aligned with traditional medicine practices.Plant-remedy mappings stored in MySQL.Flask queries remedies based on identified plants using SQLAlchemy.Jinja2 templates display remedies.

4. Yoga Posture Suggestions for Health Issue Cure

To promote holistic well-being, the platform suggests appropriate yoga postures that can help alleviate specific health conditions. Yoga poses are stored in MySQL.Flask suggests poses based on disease of the user.Jinja2 template displays poses with images.

5. Medicinal Plant Identification by Image Upload

Users can upload images of plants to identify them using the system's machine learning-based image recognition feature. The platform accurately identifies the plant species and displays relevant information on its medicinal properties and traditional uses, making plant identification accessible to all users.

2,000 leaf images for 30 species are used.Pillow resizes images to 224x224, normalizes, and augments (rotation, flipping).TensorFlow and Keras fine-tune ResNet50 and EfficientNetB0 for 20 and 5 epochs.Scikit-learn splits data into training and validation data set.Matplotlib/Seaborn visualize metrics (accuracy, precision).Flask backend processes user-uploaded images for identification.

6. User Feedback System

The feedback system enables users to share their experiences, suggest improvements, or report issues. This fosters community engagement and helps the development team enhance the platform continuously. User opinions and suggestions are valued to ensure a user-centric experience.HTML form captures user feedback.Flask stores feedback in MySQL using SQLAlchemy.

7. Search and Filter Functionality

Users can efficiently find medicinal plants by using the search and filter options. They can filter plants based on criteria such as name, health benefits, or Ayurvedic classification. This feature ensures quick access to relevant information, improving the overall user experience.Flask queries MySQL for plants by name.Flask enables filtering and updates results.

8. Voice-Based Search Function

The voice-based search feature allows users to search for plant information through voice commands. This makes the platform accessible to users with disabilities or those who prefer hands-free interaction. It enhances convenience and offers a more inclusive experience.The Web Speech API allows users to search for plants by speaking their queries instead of typing. The API captures the user's voice input through the browser's microphone and converts it into text using speech-to-text technology. This text is then sent to the Flask backend, which queries the MySQL database to find matching plant information.Finally, the relevant plant details are displayed on the web application interface, providing a faster and more interactive search experience.

9. Text-to-Speech(TTS) Feature

Users can listen to plant descriptions, medicinal uses, and other information using the text-to-speech feature. This is particularly helpful for those with visual impairments or users who prefer auditory learning. The feature enhances accessibility and user engagement.Implemented using Web Speech API, it converts plant info to speech.By clicking on the speaker button on the interface, the text will be read aloud.

10. Multi-Language Translation

To cater to a diverse audience, the platform supports multiple languages, including English, Telugu, and Hindi. Users can switch between languages to read plant information in their preferred language, ensuring a more inclusive and accessible experience.It is implemented using the Google Translate API.

11. QR Code Generation with Download Option

Each medicinal plant profile is equipped with a QR code that users can generate and download. This allows users to share plant information easily with others or quickly access it later using a mobile device, promoting the spread of knowledge about traditional medicine.It is implemented by using the qrcode library in Flask to generate QR codes that link directly to specific plant information pages.

12. Herbal Teas Module

This module educates users about various herbal teas, their preparation methods, and their health benefits. Users can explore different herbal tea recipes made from natural ingredients such as ginger, tulsi, and mint. The module promotes the use of herbal beverages for health and wellness, inspired by traditional practices.Flask handles the retrieval of relevant recipes from the database, while the front-end interface displays the recipes in an organized and visually appealing manner.

5.2 Technologies and Packages used in the System

Flask

1. Description: Flask is a lightweight Python web framework that is used for building web applications. It provides essential features such as routing, request handling, and templating without the overhead of larger frameworks.
2. Usage: Manages the core application logic, handles different routes for plant information, herbal teas, plant identification, and user feedback. It integrates with MySQL via SQLAlchemy and uses Jinja2 for rendering dynamic web pages.

SQLAlchemy

1. Description: SQLAlchemy is an ORM (Object-Relational Mapping) tool for Python. It allows interaction with the database through Python objects, eliminating the need for raw SQL queries.
2. Usage: Manages interactions between Flask and MySQL, simplifying database operations like CRUD (Create, Read, Update, Delete) for plant data, herbal teas, and user feedback.

Jinja2

1. Description: Jinja2 is a templating engine for Python, used by Flask to dynamically generate HTML pages.
2. Usage: Renders dynamic plant details, health recommendations, yoga postures, and other content on web pages by combining static HTML with dynamic data.

HTML, CSS, and JavaScript

1. Description: HTML (HyperText Markup Language) provides the structure, CSS (Cascading Style Sheets) defines the design, and JavaScript enables dynamic behavior in web pages.
2. Usage: HTML is used for structuring web pages, CSS for styling the user interface, and JavaScript for interactive features like dynamic plant searches, filtering, and navigating the virtual herbal garden.

A-Frame

1. Description: A-Frame is a web framework based on HTML and JavaScript for building virtual reality (VR) experiences.
2. Usage: Used for creating the Virtual Herbal Tour, where users can navigate a 3D herbal garden and interact with plants for detailed information.

Web Speech API

1. Description: The Web Speech API is a browser-based API for speech recognition and speech synthesis.
2. Usage: Implements Voice-Based Search for plant information using speech recognition. It also integrates Text-to-Speech (TTS) to allow users to hear plant details, making the system more accessible.

TensorFlow & Keras

1. Description: TensorFlow is an open-source machine learning framework developed by Google, and Keras is a high-level neural network API. These tools are used for deep learning tasks.
2. Usage: Used for Medicinal Plant Identification through image uploads. The system uses machine learning models (like ResNet50 and EfficientNetB0) to classify plants based on leaf images.

Scikit-learn

1. Description: Scikit-learn is a Python library for machine learning, used for data mining and data analysis.
2. Usage: Handles tasks like splitting datasets into training and testing sets, as well as evaluating machine learning models using metrics like accuracy, precision, and recall.

Pillow

1. Description: Pillow is an image manipulation library for Python.
2. Usage: Pre-processes plant images for identification, including resizing, normalizing pixel values, and augmenting the dataset (e.g., image rotation, flipping).

Matplotlib/Seaborn

1. Description: Matplotlib and Seaborn are Python libraries for data visualization.
2. Usage: Visualize machine learning model performance metrics, such as accuracy and precision, during the training process.

Google Translate API

1. Description: The Google Translate API enables language translation.
2. Usage: Implements Multi-Language Translation, allowing plant information to be displayed in multiple languages (e.g., English, Telugu, Hindi), increasing accessibility.

QR Code Library (qrcode)

1. Description: The qrcode library allows generation of QR codes in Python.
2. Usage: Enables the generation of QR codes for plant information pages, which users can scan or share for quick access to detailed plant data.

MySQL

1. Description: MySQL is an open-source relational database management system.
2. Usage: Stores data such as plant information, medicinal remedies, yoga poses, herbal tea recipes, user feedback, and more. It is connected to Flask using SQLAlchemy for data retrieval and storage.

Flask-SQLAlchemy

1. Description: Flask-SQLAlchemy is an extension for Flask that simplifies database interaction using SQLAlchemy.
2. Usage: Provides seamless integration between Flask and MySQL, facilitating easy database operations related to plant data, remedies, user feedback, etc.

5.3 Algorithms

ResNet (Residual Network) introduced residual learning with skip connections, solving the vanishing gradient problem that occurs in very deep networks. It is extensively used in medical image diagnosis, face recognition, and self-driving car vision systems. ResNet assumes that deeper networks should ideally perform better, but without residual connections, training deep networks becomes difficult. Its key advantage is its ability to train very deep networks while maintaining performance, though it requires significant computational resources.

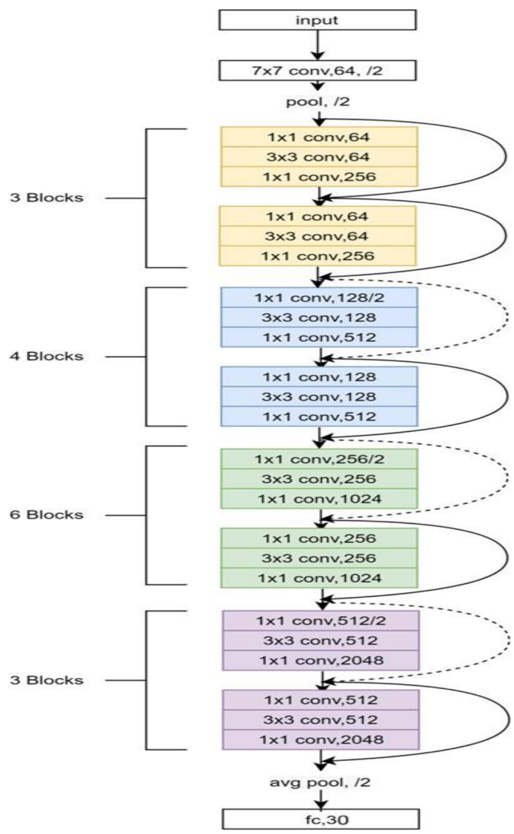


fig 5.1 ResNet Architecture

The ResNet model was implemented for Ayurvedic medicinal plant classification:

Step 1: Data Collection and Preprocessing

Collected a dataset of 2,000 leaf images from 30 plant species. Images resized to 224x224 pixels using Pillow. Images normalized by scaling pixel values to the range [0,1]. Data augmentation applied using random flipping, rotation, zooming, and rescaling for improved generalization.

Step 2: Model Selection and Initialization

Selected ResNet50V2 as the base model for transfer learning. Loaded pre-trained weights from ImageNet. Initial layers were frozen to retain learned features.

Step 3: Model Customization

Applied global average pooling to reduce the spatial dimensions and minimize parameters. Added a dense layer with 30 neurons (equal to the number of plant species) with softmax activation for multi-class classification. The model summary showed a total of 23,564,800 parameters (Trainable: 23,519,360, Non-Trainable: 45,440).

Step 4: Compilation

Used Sparse Categorical Cross entropy as the loss function for multi-class classification. Implemented Adam optimizer with a learning rate of 0.001. Selected accuracy, precision, recall, and F1-score as evaluation metrics.

Step 5: Learning Rate Adjustment

Applied the ReduceLROnPlateau callback to monitor validation loss and reduce the learning rate when stagnation occurred, preventing overfitting.

Step 6: Model Training

Split the dataset into 80% training and 20% validation. Trained the model for 20 epochs using a batch size of 32. Implemented early stopping to terminate training if validation loss stopped improving.

Step 7: Evaluation

Evaluated model performance using the test dataset.

Accuracy	Precision	Recall	F1-score
96%	97%	97%	97%

EfficientNet is a family of convolutional neural networks (CNNs) designed for high-performance image classification tasks with optimized accuracy and computational efficiency. EfficientNet-B0 is the base version, and it uses a compound scaling method to balance depth, width, and resolution for optimal results.

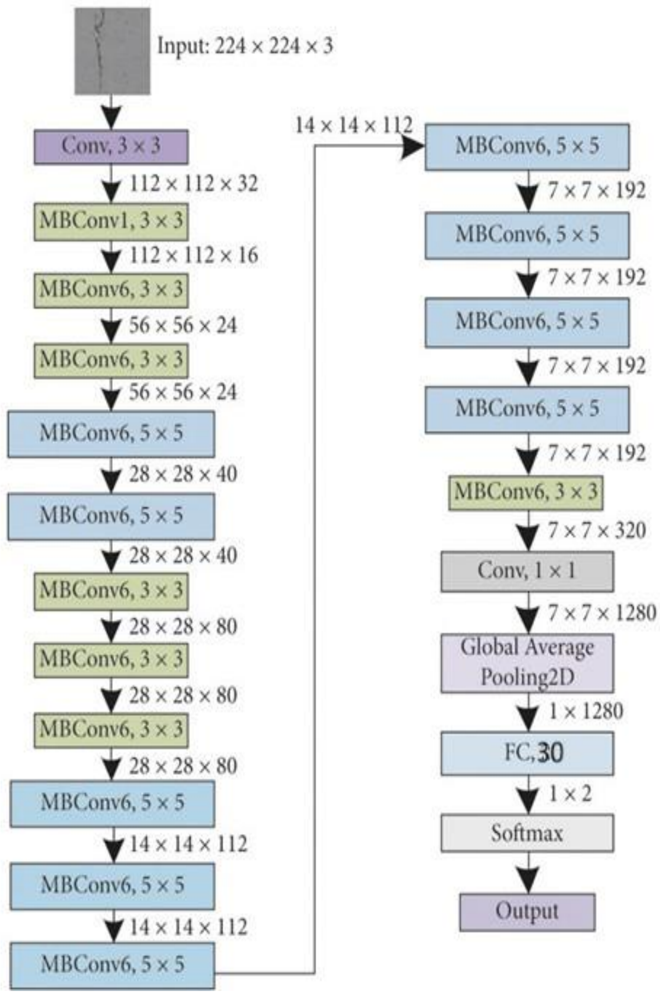


fig 5.2 EfficientNet Architecture

The EfficientNetB0 model was implemented for Ayurvedic medicinal plant classification:

Step 1: Data Collection and Preprocessing Collected a dataset of 2,000 leaf images from 30 plant species. Images resized to 224x224 pixels using Pillow. Pixel values normalized to the range [0,1] for efficient processing. Applied data augmentation techniques such as random flipping, rotation, and zooming to improve generalization.

Step 2: Model Selection and Initialization Choose EfficientNetB0 as the base model for transfer learning. Loaded pre-trained weights from the ImageNet dataset. Set include_top=False to remove the default classification layer.

Step 3: Model Customization Added a GlobalAveragePooling2D layer to reduce feature map dimensions using global averaging, making it computationally efficient. Included a dense layer with 128 neurons and ReLU activation (Dense(128, activation='relu')) for introducing non-linearity and enhancing learning. Added a final dense layer with 30 neurons and softmax activation for multi-class classification. The model consisted of 4,217,409 total parameters, with 167,838 trainable and 4,049,571 non-trainable parameters.

Step 4: Compilation Compiled the model using the Adam optimizer with a learning rate of 0.001. Choose sparse_categorical_crossentropy as the loss function, as it is suitable for integer-encoded multi-class classification. Set accuracy as the primary evaluation metric.

Step 5: Model Training Split the dataset into 80% training and 20% validation. Trained the model using a batch size of 32 for 5 epochs. Applied early stopping to prevent overfitting and save the best-performing model.

Step 6: Evaluation Evaluated the model using the test dataset.

Accuracy	Precision	Recall	F1-score
99.6%	99.6%	99.6%	99.6%

6. TESTING

6.1 Test Cases and Reports

The following test cases were designed and executed to validate the functionality of the project:

1. **Importing Libraries:**
    a. Objective: Ensure necessary libraries such as TensorFlow, Keras, NumPy, and Seaborn are imported without errors.
    b. Expected Result: Libraries should be successfully imported.
    c. Actual Result: Libraries imported without any issues.
    d. Status: Pass
2. **Data Collection and Loading:**
    a. Objective: Verify the dataset is accessible and correctly loaded from the specified directory.
    b. Expected Result: Dataset should load successfully.
    c. Actual Result: Data loaded and displayed for visualization.
    d. Status: Pass
3. **Data Analysis:**
    a. Objective: Perform exploratory data analysis (EDA) to check for class imbalances and visualize data patterns.
    b. Expected Result: Data should be properly analyzed using visualizations.
    c. Actual Result: Visualizations were successfully generated without errors.
    d. Status: Pass
4. **Image Pre-processing:**
    a. Objective: Verify the resizing of images to 224x224, normalization using rescaling, and augmentation using flipping, rotation, and zooming.
    b. Expected Result: Images should be pre-processed without errors.

c. Actual Result: Image pre-processing was successfully completed.

d. Status: Pass

5. Feature Selection and Extraction:

a. Objective: Ensure essential features are extracted using pre-trained models like ResNet50V2 and EfficientNet.

b. Expected Result: Features should be successfully extracted from images.

c. Actual Result: Feature extraction completed without errors.

d. Status: Pass

6. Train-Test Split:

a. Objective: Validate the correct splitting of the dataset into 70% training and 30% validation data.

b. Expected Result: Data should be correctly split with balanced class distribution.

c. Actual Result: Data was successfully split into training and validation sets.

d. Status: Pass

7. Model Application:

a. Objective: Evaluate the performance of ResNet50V2 and EfficientNet models.

b. Expected Result: Models should be applied successfully without errors.

c. Actual Result: Both models were evaluated, and results were obtained as expected.

d. Status: Pass

8. Prediction:

a. Objective: Ensure the model correctly predicts the class of unseen images.

b. Expected Result: Model should predict accurate results.

c. Actual Result: Predictions were successfully made and verified.

d. Status: Pass

6.2 OUTPUTS

fig 6.1 Home Page

The above fig 6.1 Home Page serves as the application's starting point, offering a welcoming interface with an introduction to the platform’s purpose. Users are presented with clear options to either log in or sign up, ensuring easy navigation.

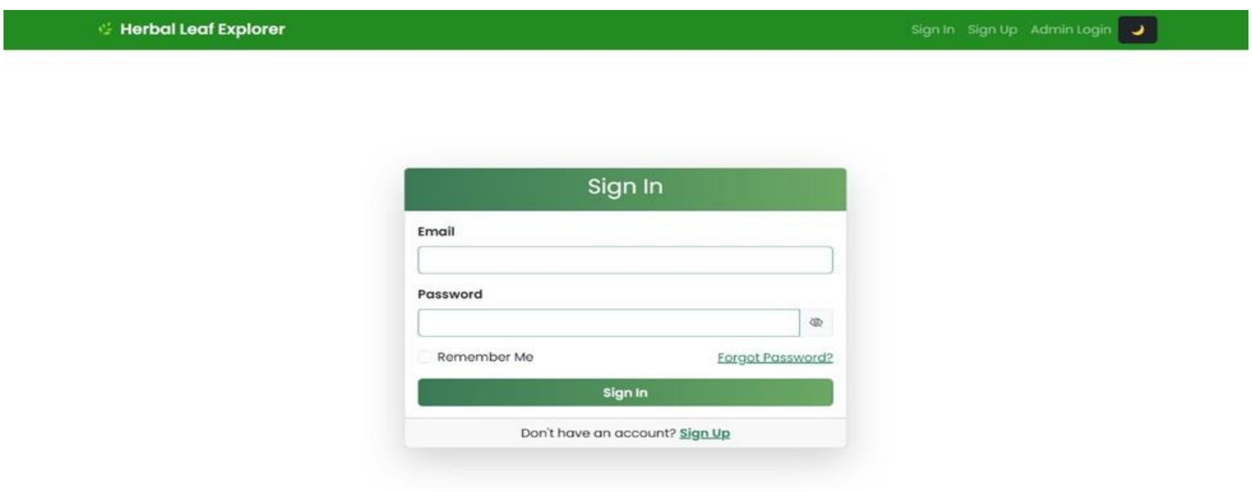


fig 6.2 Sign In Page

The above fig 6.2 represents the Sign In page where the registered users can enter their credentials in this sign in page to access personalized features.

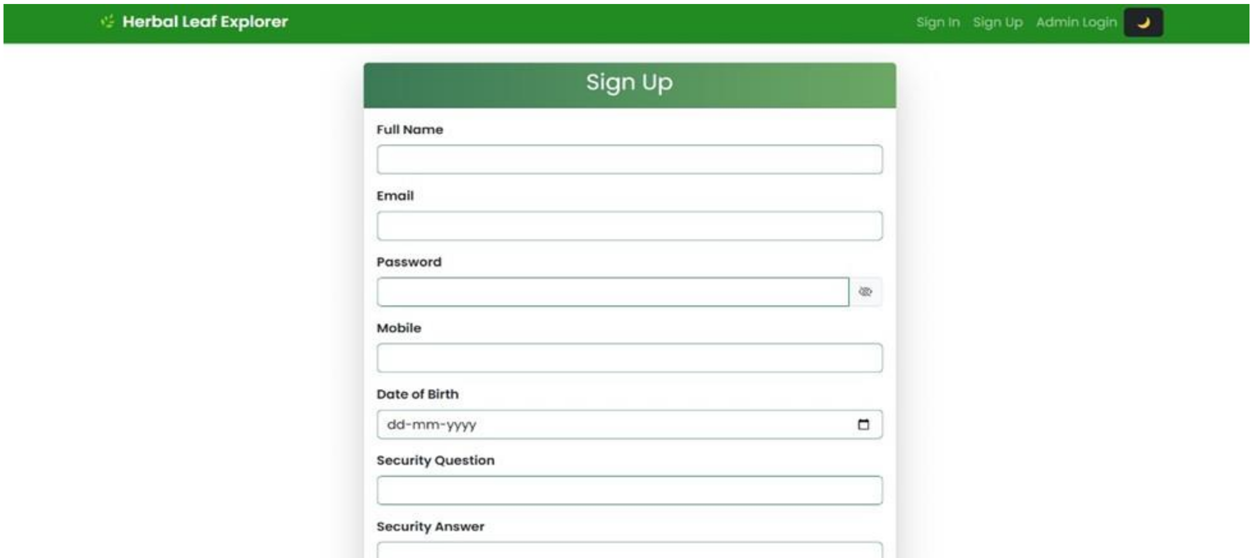


fig 6.3 Sign Up page

The above fig 6.3 represents Sign Up page.New users can navigate to the Sign-Up Page to create an account by providing the required information.

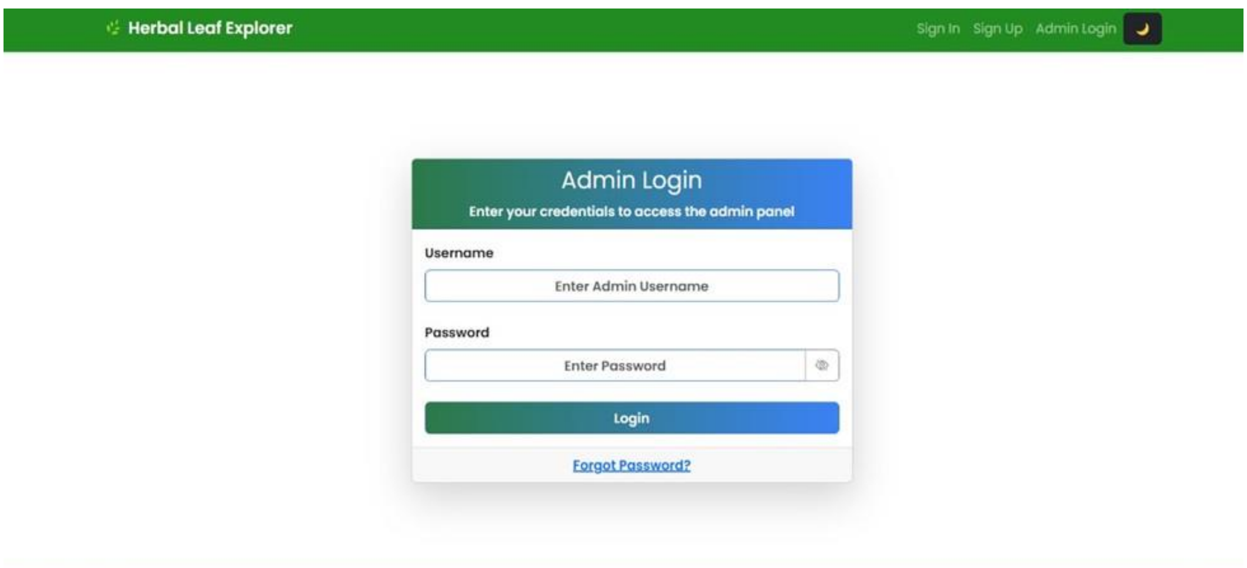


fig 6.4 Admin Login Page

The above fig 6.4 Admin Login Page offers a secure portal for administrators to manage the application.Admin will login into the application with the credentials.

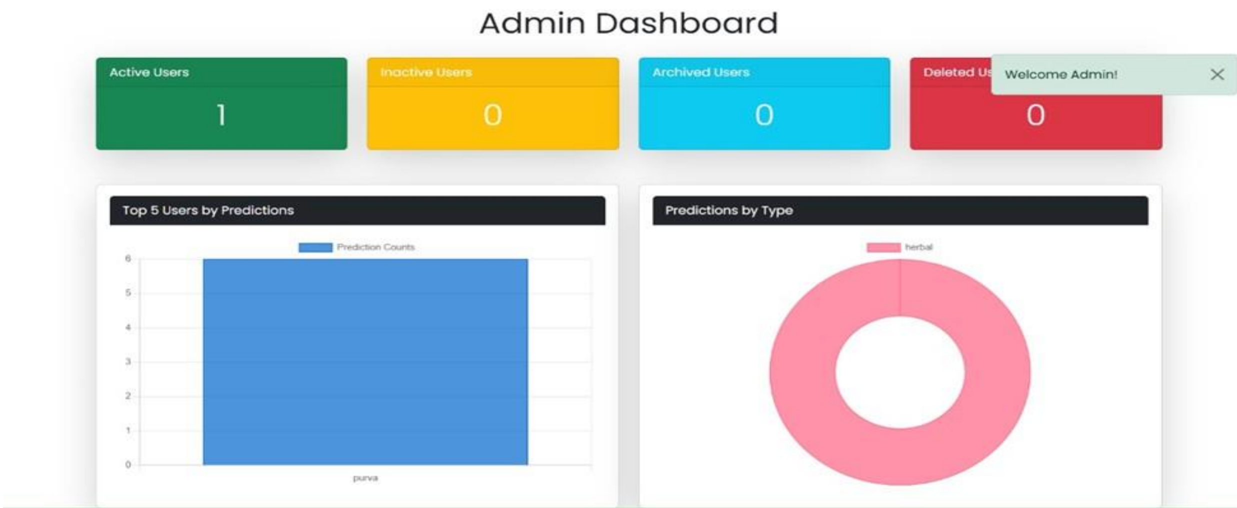


fig 6.5 Admin Dashboard

The above fig 6.5 represents Admin Dashboard where Admins can access the backend dashboard, where they oversee user management, monitor activity, and handle system maintenance.

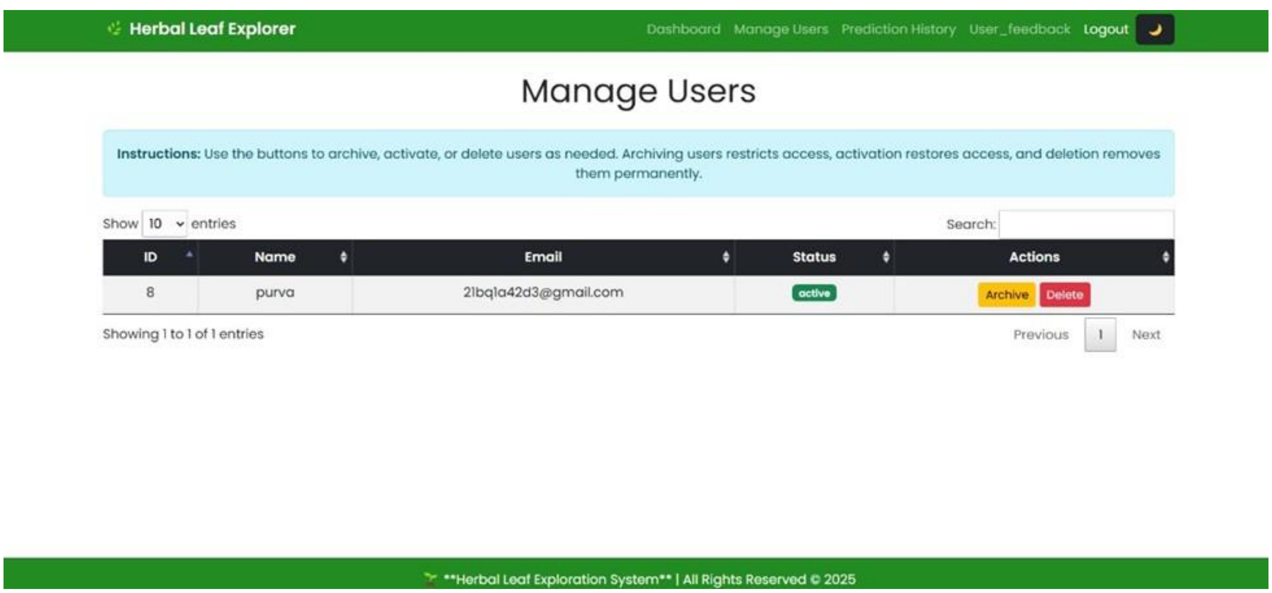


fig 6.6 User Management Dashboard

The above fig 6.6 User Management Dashboard is a comprehensive panel where administrators can manage all user accounts. Admins have the authority to activate, archive, or delete users as needed.

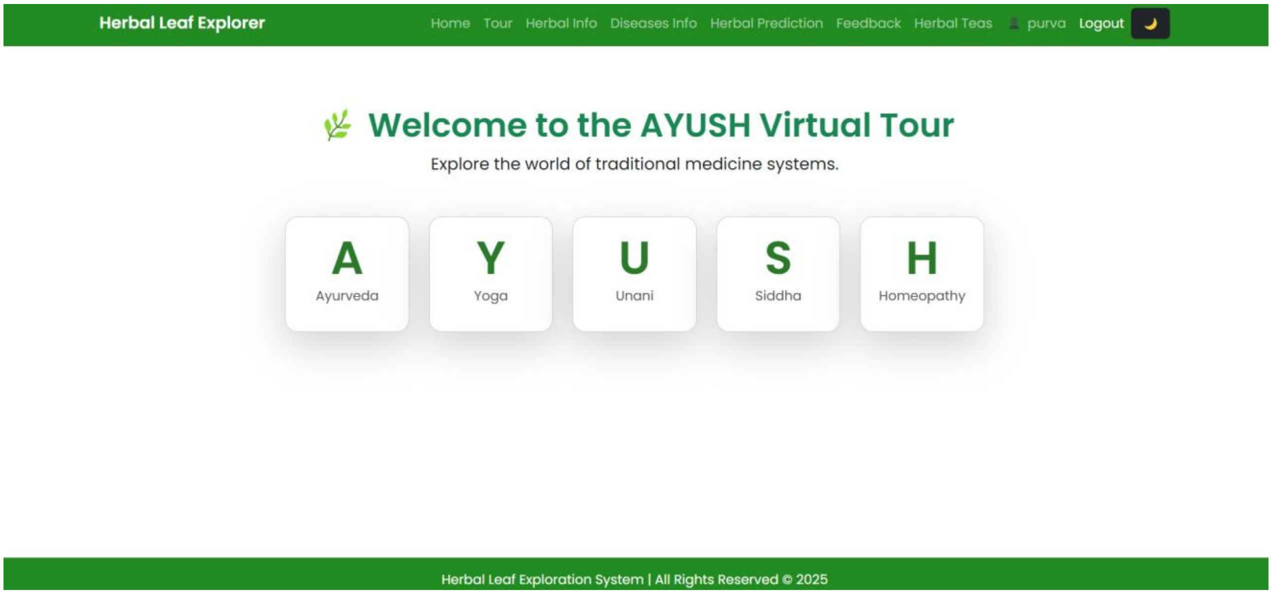


fig 6.7 Ayush Tour Page

The above fig 6.7 AYUSH Tour page offers users an organized overview of the traditional systems of medicine under AYUSH, including Ayurveda, Yoga & Naturopathy, Unani, Siddha, and Homeopathy.

fig 6.8 Ayush Virtual Tour

The above fig 6.8 represents Ayush Virtual Tour where users can navigate through the 3D space, observe plants and learn about their medicinal properties.The realistic visual design and interactive controls make the virtual tour a valuable educational tool.

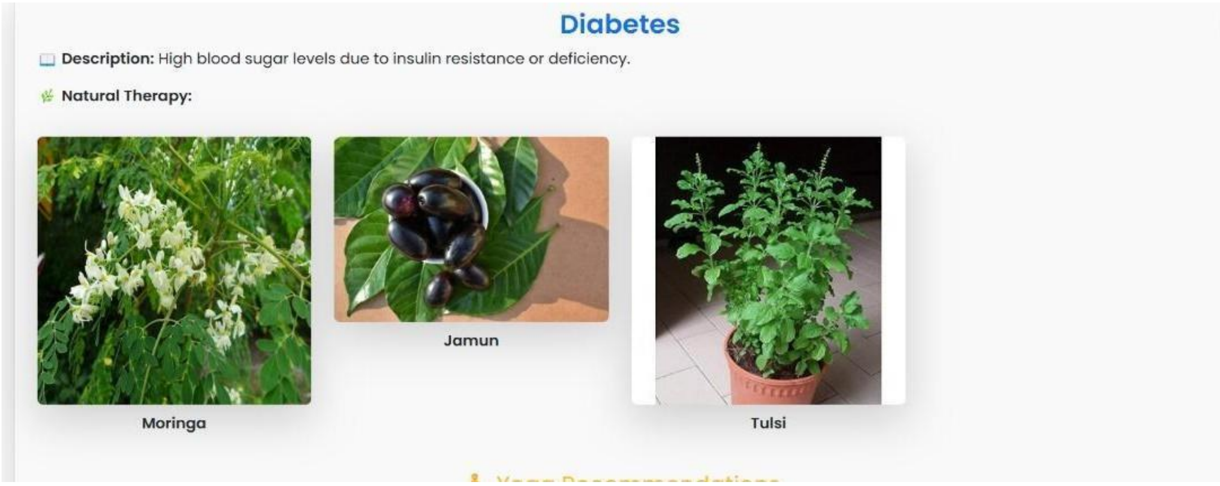


fig 6.9 Disease Module

The above fig 6.9 Disease Module allows users to receive personalized herbal treatment suggestions based on Ayurvedic principles. Users can select from a list of diseases, and the application provides suitable herbal remedies.

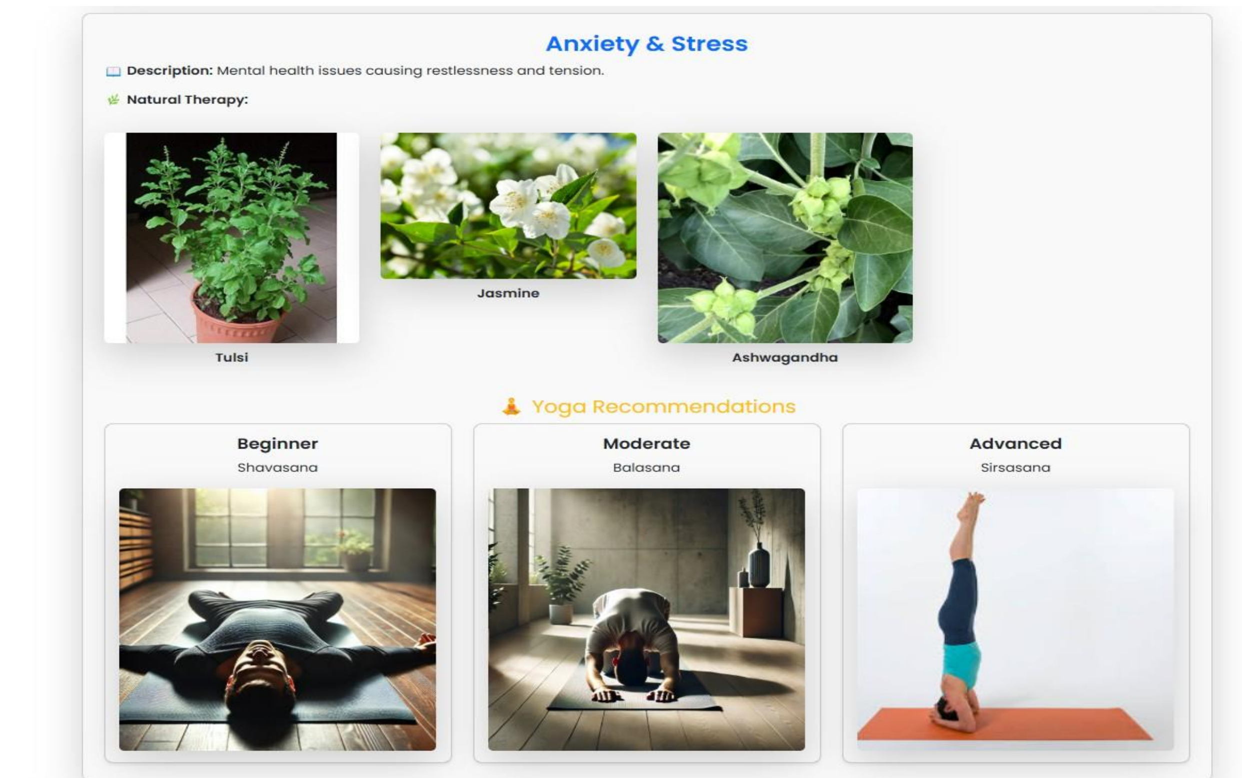


fig 6.10 Natural Therapy and Yoga Recommendation Module

The above fig 6.10 Yoga and Natural Therapies Suggestion module provides personalized yoga asana recommendations and natural therapy suggestions based on disease.

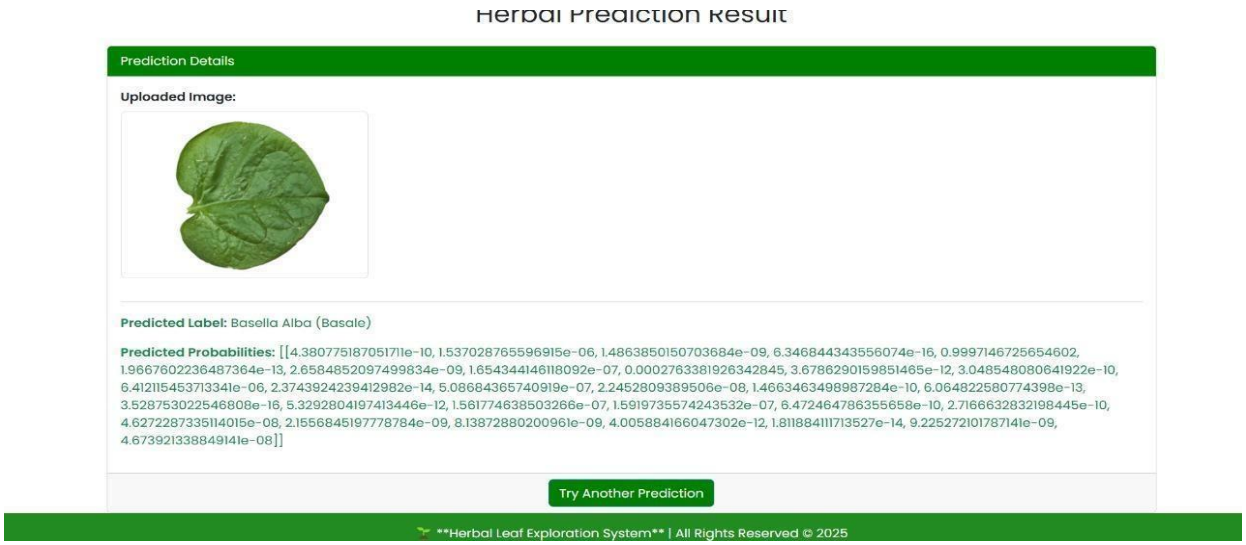


fig 6.11 Plant Prediction Module

The above fig 6.11 Plant Prediction Module allows users to upload images of plants for identification. Using advanced algorithms such as ResNet and EfficientNet,the system predicts the plant species.

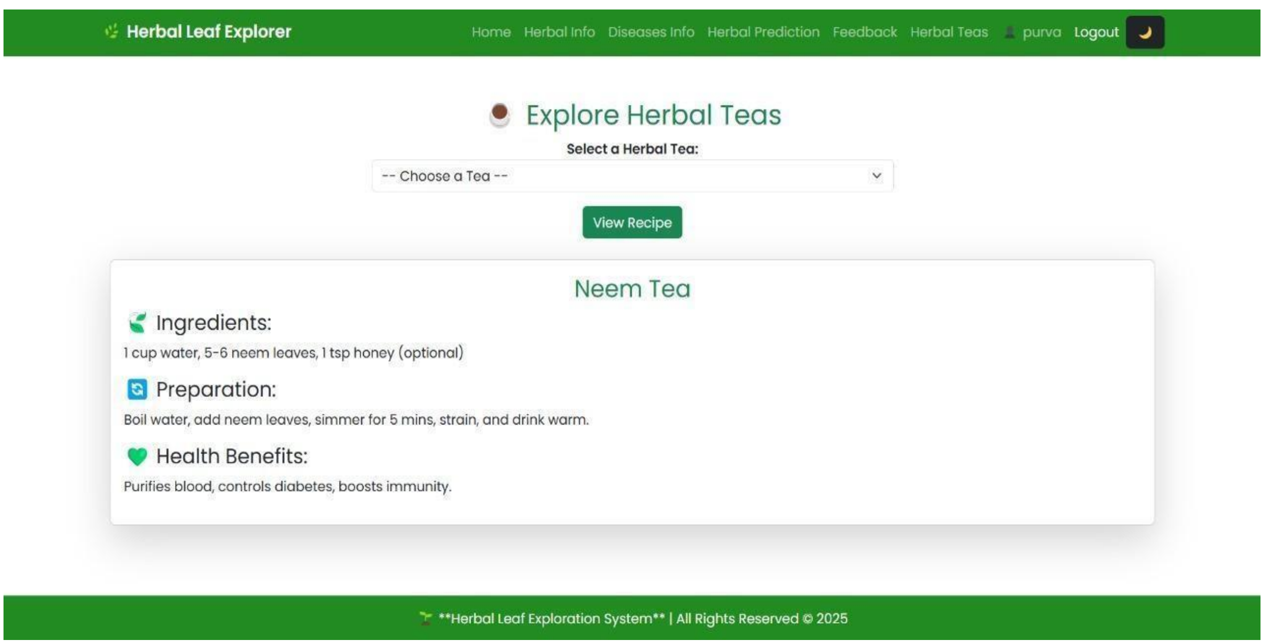


fig 6.12 Herbal Teas Module

The above fig 6.12 Herbal Teas Module offers curated herbal tea recipes based on users' specific health needs. Each recipe includes a list of ingredients, step-by-step preparation instructions, and an explanation of the health benefits.

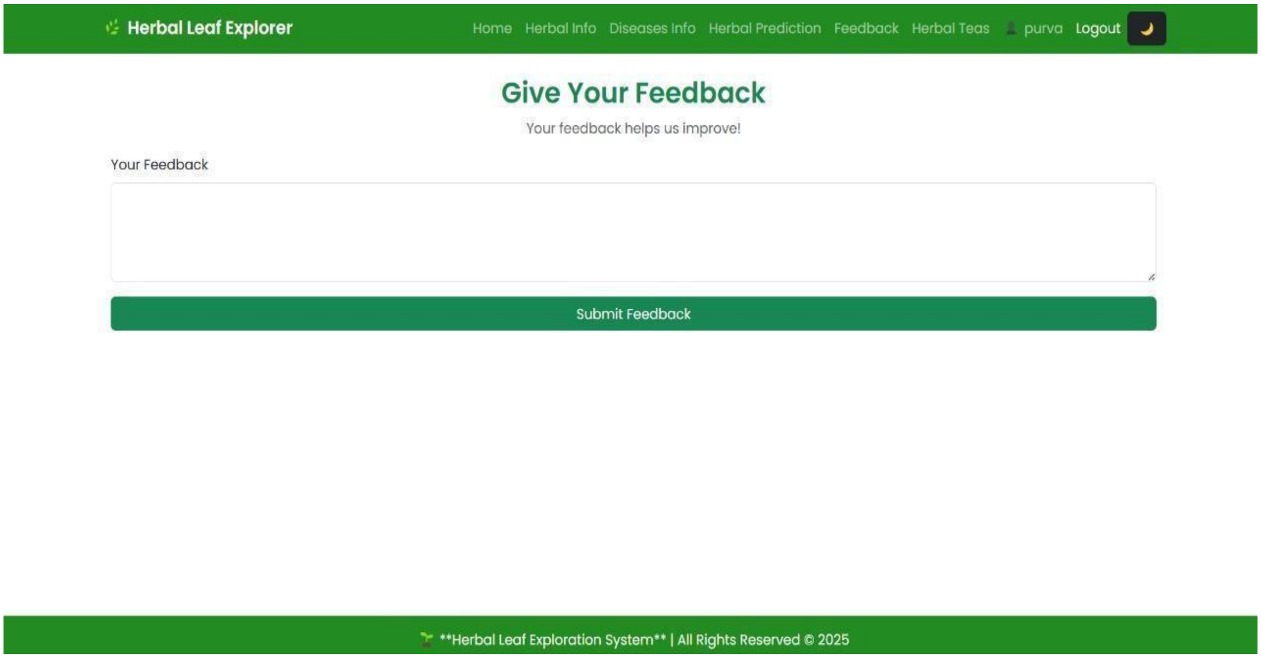


fig 6.13 User Feedback

The above fig 6.13 represents Feedback Form.It serves as a valuable communication channel between users and the system administrators.Users can share their feedback in this form.

fig 6.14 Admin Feedback Page

The above fig 6.14 represents Admin Feedback Page,where Admin can view the feedback submitted by the users which helps in improving the system.It displays user id with feedback.

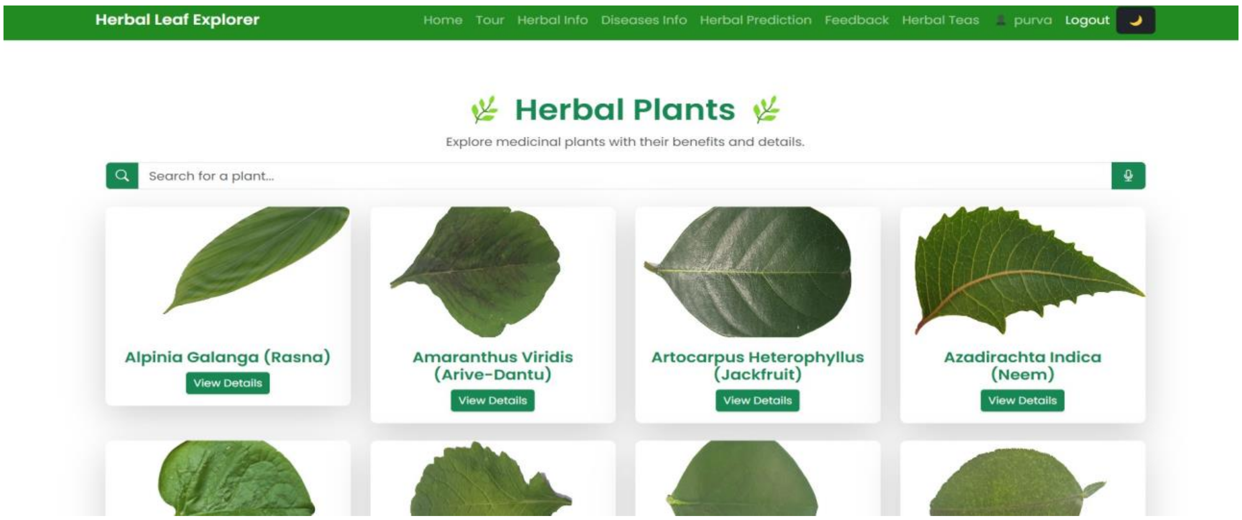
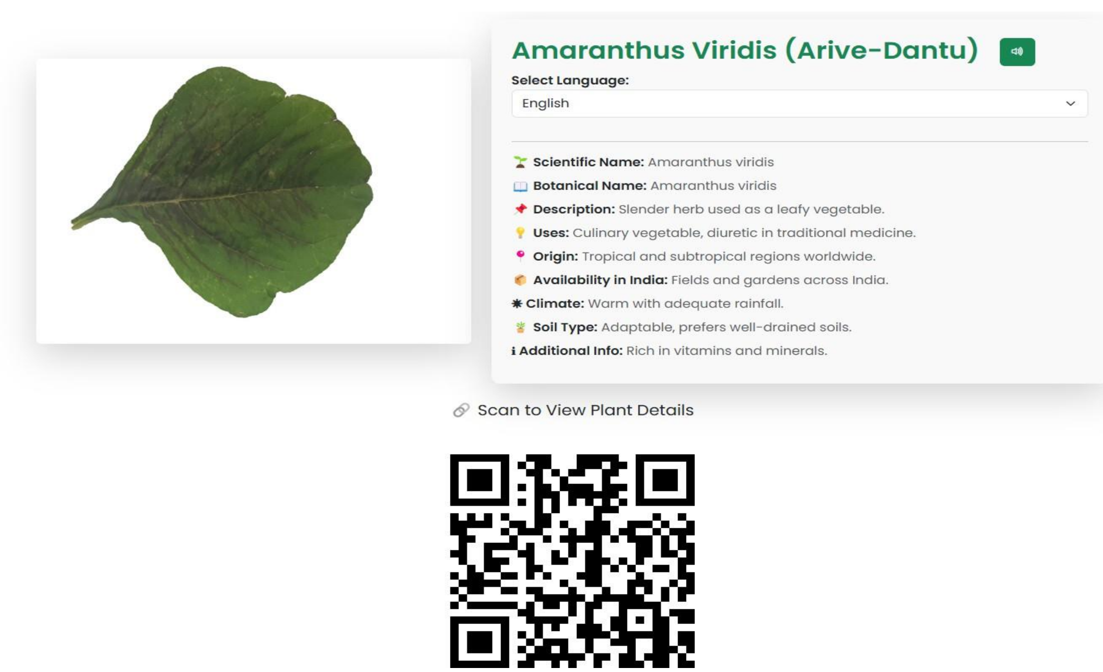


fig 6.15 Herbal Plants Module

The above fig 6.15 Herbal Plants Module showcases a variety of medicinal plants used in the AYUSH system, displayed with their names for easy identification.



fig

6.16 Plant Information Module

The above fig 6.16 serves as a knowledge hub for users to explore detailed information about medicinal plants. Each plant entry includes its botanical name, common uses and additional information.

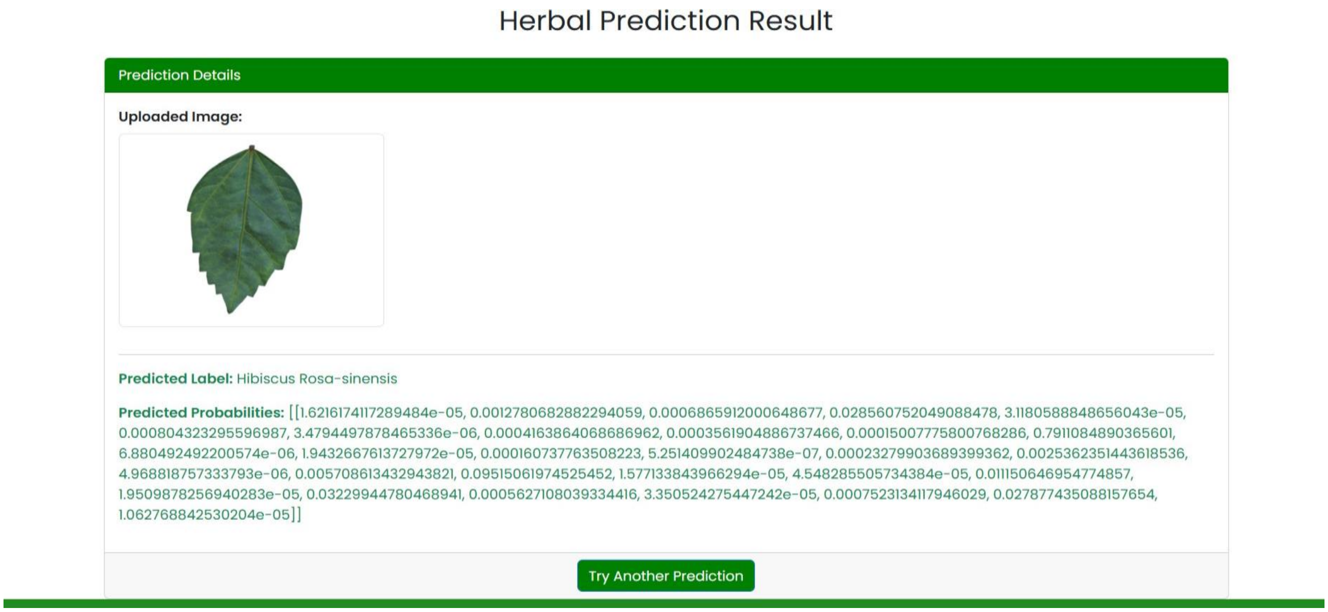


fig 6.17 Herbal Plant prediction1

The Plant Prediction Module allows users to upload images of plants for identification. Using advanced algorithms such as ResNet and EfficientNet,the system predicts the plant species.In the above fig 6.17 the Hibiscus Rosa-sinensis is predicted.

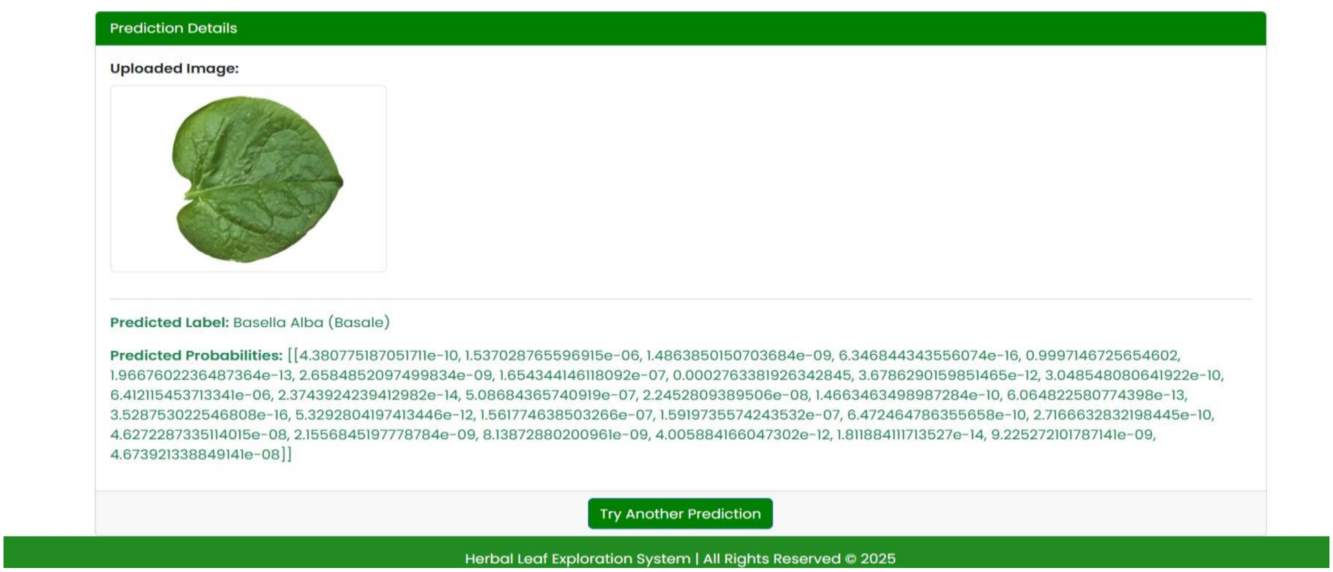


fig 6.18 Herbal plant prediction2

In the above fig 6.18 the Basella Alba plant is predicted when the user uploads the plant image in the Plant Prediction Module, the system processes the image and identifies the plant species.

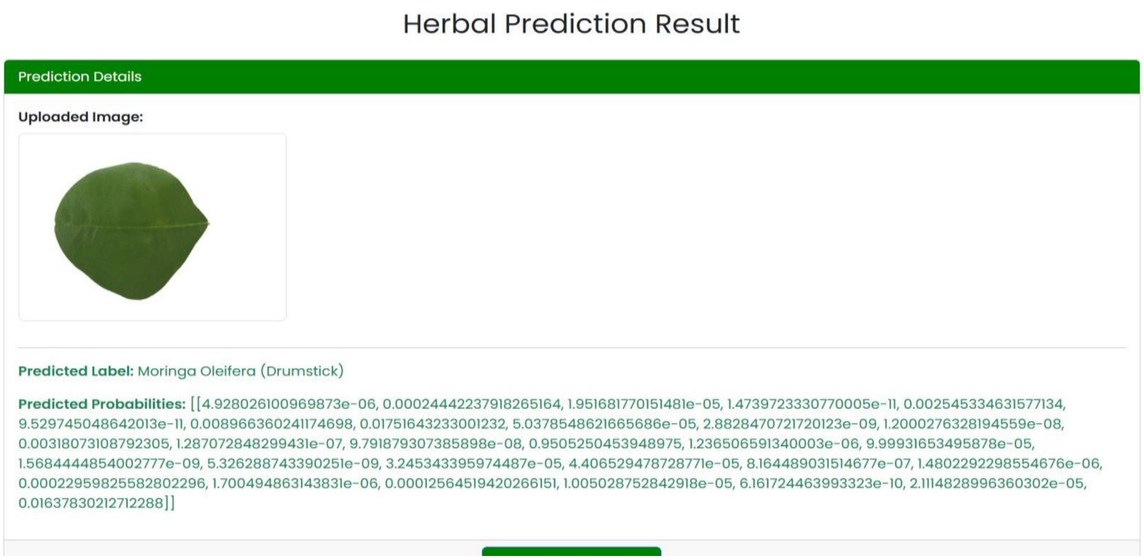


fig 6.19 Herbal Plant prediction3

When a user uploads a plant image in the Plant Prediction Module, the system processes the image and identifies the plant species. In this case, the system predicts the plant as Moringa Oleifera, displaying it to the user.

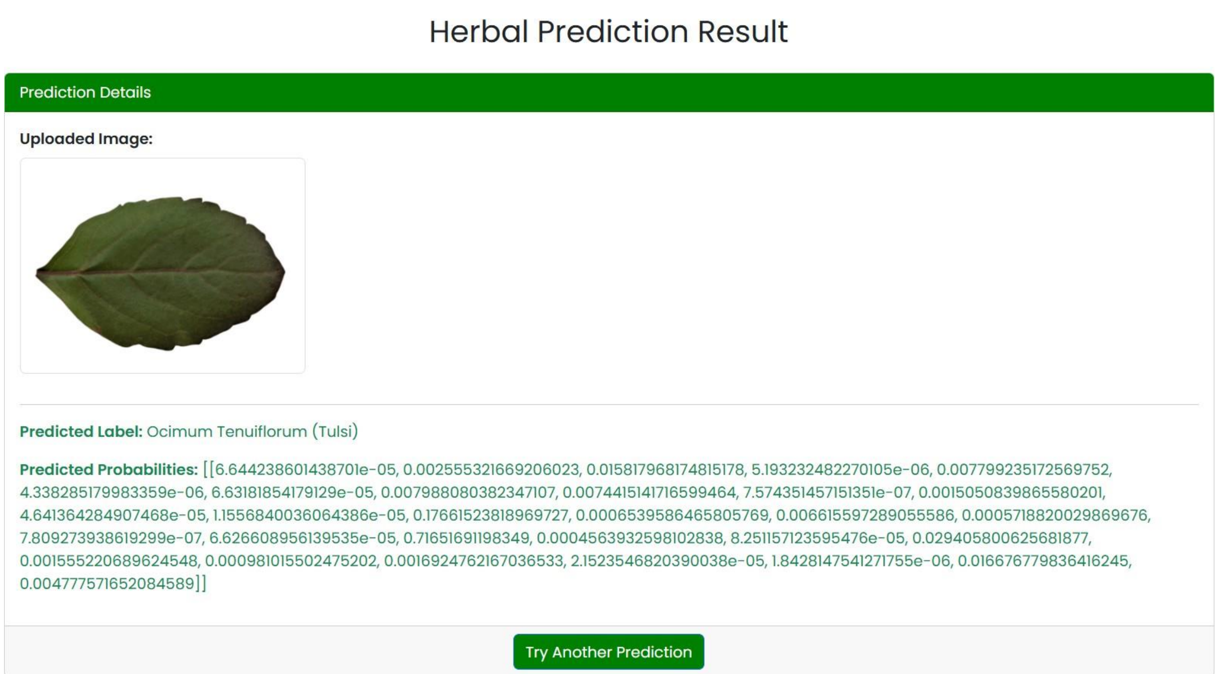


fig 6.20 Herbal Plant Prediction4

The above fig 6.20 represents the herbal prediction module and the Plant Ocimum Tenuiflorum is predicted and displayed to the user.

7. CONCLUSION

The Ayush Virtual Herbal Garden serves as an innovative digital platform that bridges the gap between ancient Ayurvedic knowledge and modern technology. By leveraging AI-driven image recognition, immersive 3D environments, and interactive learning modules, the system offers an engaging and informative experience for users of all backgrounds. With features like virtual herbal tours, plant identification through image uploads, and health-focused medicine recommendations, the platform empowers users to explore and understand the healing potential of medicinal plants. The integration of yoga posture suggestions, herbal tea modules, and voice-based search further enhances the system's usability, ensuring a holistic and inclusive approach to well-being. Additionally, the implementation of multi-language support and text-to-speech functionalities makes the platform accessible to a wider demographic. By providing QR code generation and a comprehensive feedback system, the Ayush Virtual Herbal Garden fosters continuous user engagement and knowledge sharing. With a classification accuracy of 99% using CNN models like ResNet50 and EfficientNetB0, the platform demonstrates the effectiveness of AI in plant identification. The seamless integration of a Flask-based backend, MySQL database, and user-friendly front-end interface ensures an intuitive user experience. Overall, the Ayush Virtual Herbal Garden promotes awareness of traditional herbal practices and encourages the sustainable use of natural remedies, contributing to the preservation of Ayurvedic heritage.

FUTURE SCOPE

The future scope of the Ayush Virtual Herbal Garden includes expanding the dataset to cover a wider variety of medicinal plants, incorporating images of different plant parts such as flowers, bark, and roots for improved identification accuracy. Integrating real-time mobile applications with augmented reality (AR) can offer users an immersive experience, allowing them to identify plants directly in their environment. AI-driven disease prediction models can be further enhanced to recommend personalized Ayurvedic treatments based on user health data. Multi-language support can be extended to cover more regional languages, fostering greater accessibility for a diverse user base. Collaborations with Ayurvedic practitioners and researchers will enable continuous refinement of plant identification and treatment recommendation algorithms. Furthermore, implementing voice-based virtual assistants and chatbot support can provide real-time guidance to users, making traditional healthcare knowledge more accessible. The platform can also be expanded to include educational modules for students and researchers, promoting the preservation and widespread understanding of Ayurvedic medicinal knowledge.

8. REFERENCES

1. S.Kavitha, T. Satish Kumar, E. Naresh, V. H. Kalmani, K. D. Bamane, and P. K. Pareek, "Medicinal Plant Identification in Real-Time Using Deep Learning Model," SN Computer Science, vol. 5, pp. 1-11, 2024, doi: 10.1007/s42979-023-02398-5.
2. Ambarwari, A., Adrian, Q. J., Herdiyeni, Y., & Hermadi, I. (2020). Plant species identification based on leaf venation features using SVM. TELKOMNIKA (Telecommunication Computing Electronics and Control), 18(2), 726-732.
3. Azadnia, R., Al-Amidi, M. M., Mohammadi, H., Cifci, M. A., Daryab, A., & Cavallo, E. (2022). An AI based approach for medicinal plant identification using deep CNN based on global average pooling. Agronomy, 12(11), 2723.
4. Sushant M. Vanmore, Omkar R. Suryawanshi, Milind H. Kamble, Yash V. Latkar, and Ms. Sonal K. Beldar, "The Virtual Herbal Garden," International Journal of Scientific Research in Engineering and Management (IJSREM), Volume: 08 Issue: 10, Oct - 2024, pp. 1-4, ISSN: 2582-3930.
5. Prasvita, D. S., & Herdiyeni, Y. (2013). MedLeaf: mobile application for medicinal plant identification based on leaf image. International Journal on Advanced Science, Engineering and Information Technology, 3(2), 5-8.