🌿 AgroBot - Intelligent Plant Disease Detection and Assistant

**Project Overview**

**AgroBot** is an AI-powered, multilingual **Plant Disease Detection and Farmer Support System**.  
It helps farmers identify plant diseases from images, provides detailed solutions in their **preferred language**, and allows **administrators** to manage users and prediction records efficiently.

Developed using **Python**, **Streamlit**, and **Google Translator**, AgroBot serves as a **smart virtual plant doctor** that makes agriculture more accessible and efficient.

**🧩 Key Features**

**1. Plant Disease Detection**

* Users can upload plant leaf images.
* AgroBot identifies the disease using a pre-trained ML model.
* Displays disease name, symptoms, and recommended treatments.

**2. Farmer Chatbot**

* Users can describe symptoms (e.g., *“yellow leaves”*).
* AgroBot matches input with predefined disease dataset (using difflib for similarity).
* Suggests likely diseases and treatments.

**3. Multilingual Support**

* Integrated **Google Translator API**.
* Supports major Indian and global languages (Telugu, Hindi, Tamil, Kannada, Bengali, English, etc.).
* Translates chatbot messages, disease details, and UI text dynamically.

**4. User System**

* **Registration & Login** system using hashed passwords (bcrypt).
* **Session state** maintained securely in Streamlit.
* Users can view their **previous disease predictions**.

**5. Admin Dashboard**

* Admins can:
  + View all registered users.
  + Delete user accounts.
  + View or delete prediction records of any user.
* Default admin is auto-created at first launch (create\_default\_admin()).

**6. Database Management**

* Uses **JSON-based storage** for simplicity.
* Files:
  + user.json → Stores user credentials & roles.
  + predictions.json → Stores prediction results with timestamps & user IDs.

**7. Predefined Dataset**

* Contains common **plant diseases**, **symptoms**, and **solutions** (e.g., Apple Scab, Tomato Leaf Curl, Rice Blast, etc.).
* Enables chatbot-based identification without an internet connection.

**Project Structure**

AgroBot/

│

├── app.py # Main Streamlit application

├── user.py # User authentication & admin management

├── database.py # Prediction saving & retrieval

├── diseases.json # Predefined plant disease dataset

├── user.json # Registered users (JSON DB)

├── predictions.json # Saved predictions (JSON DB)

├── requirements.txt # Project dependencies

└── README.md # Documentation

**Main Components Description**

**🔹 app.py**

* Launches the Streamlit interface.
* Handles navigation (Login, Register, Disease Recognition, ChatBot, Admin Panel).
* Controls UI language switching.
* Displays predictions and chatbot output.

**🔹 user.py**

Handles:

* User registration and login
* Password encryption using bcrypt
* Admin detection (is\_admin)
* User deletion and creation of default admin

**🔹 database.py**

Manages:

* Saving prediction results to predictions.json
* Fetching user-specific or all predictions
* Deleting records for specific users or entries

**Languages Supported**

* English
* Telugu
* Hindi
* Tamil
* Kannada
* Bengali  
  *(More languages can be added easily using Google Translator API.)*

**Authentication Flow**

1. User registers → credentials stored in user.json.
2. Passwords are encrypted using **bcrypt**.
3. On login, credentials are validated.
4. Logged-in session stored via st.session\_state.
5. Admin users get access to **Admin Dashboard**.

**Admin Dashboard Functionalities**

| **Feature** | **Description** |
| --- | --- |
| 👥 View Users | List all registered users |
| ❌ Delete User | Remove a user account |
| 🗂 View Predictions | See all disease detections |
| 🗑 Delete Prediction | Remove specific prediction entries |

**🤖 Chatbot Functionality**

* Uses **text similarity** (difflib.get\_close\_matches) to find matching disease names.
* Retrieves symptoms and treatments from dataset.
* Supports **language translation** for both input and output.

**Example:**

👨‍🌾 Input: “My plant has yellow leaves” (in Telugu)

🤖 Output: “It seems like a Nitrogen Deficiency. Add urea or organic compost.”

**Tech Stack**

| **Layer** | **Tools Used** |
| --- | --- |
| Frontend | Streamlit |
| Backend | Python |
| Authentication | bcrypt |
| Translation | Googletrans |
| Database | JSON-based |
| Image Processing | PIL (Pillow) |
| Similarity Matching | difflib |
| ML (optional) | TensorFlow / Keras |

**📜 License**

Licensed under the **MIT License** – free for educational and research use.

**Developed By Yagnasree Rampa**  
🌱 AI & Web Developer | Passionate about AgriTech Innovation