**chatbot\_model.py ------- File Name**

– Chatbot Logic

**Code**

import random

# Agriculture chatbot responses

responses = {

"greeting": [

"Hello! I'm your agriculture assistant. 🌱 How can I help you today?",

"Hi there! Ask me anything about farming and crops. 🚜"

],

"fertilizer": [

"For better yield, use organic compost and nitrogen-rich fertilizer like urea.",

"Consider using phosphorus and potassium-based fertilizers for root growth."

],

"pest": [

"Neem oil spray is effective for many pests.",

"Introduce natural predators like ladybugs to control pest population."

],

"weather": [

"Please check the local forecast before sowing seeds.",

"Avoid watering plants if heavy rain is predicted."

],

"default": [

"I'm not sure about that. Could you please rephrase?",

"Sorry, I don't understand. Can you ask another question?"

]

}

def get\_response(user\_input):

user\_input = user\_input.lower()

if "hello" in user\_input or "hi" in user\_input:

return random.choice(responses["greeting"])

elif "fertilizer" in user\_input:

return random.choice(responses["fertilizer"])

elif "pest" in user\_input:

return random.choice(responses["pest"])

elif "weather" in user\_input:

return random.choice(responses["weather"])

else:

return random.choice(responses["default"])

app.py ----- File Name

– Flask App with Authentication

code

from flask import Flask, render\_template, request, redirect, url\_for, flash, session, jsonify

from flask\_sqlalchemy import SQLAlchemy

from flask\_bcrypt import Bcrypt

from chatbot\_model import get\_response

import os

# Flask app setup

app = Flask(\_\_name\_\_)

app.secret\_key = "secret123"

# Database setup

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///database.db'

db = SQLAlchemy(app)

bcrypt = Bcrypt(app)

# User Model

class User(db.Model):

id = db.Column(db.Integer, primary\_key=True)

username = db.Column(db.String(150), nullable=False, unique=True)

password = db.Column(db.String(150), nullable=False)

# Create database if not exists

with app.app\_context():

db.create\_all()

# ----------- ROUTES -----------

# Home (Chatbot) - Requires Login

@app.route("/")

def home():

if "user\_id" not in session:

return redirect(url\_for("login"))

return render\_template("index.html")

# Signup

@app.route("/register", methods=["GET", "POST"])

def register():

if request.method == "POST":

username = request.form.get("username")

password = request.form.get("password")

# Check if username exists

existing\_user = User.query.filter\_by(username=username).first()

if existing\_user:

flash("Username already exists. Try another.", "error")

return redirect(url\_for("register"))

# Hash password

hashed\_password = bcrypt.generate\_password\_hash(password).decode('utf-8')

# Save new user

new\_user = User(username=username, password=hashed\_password)

db.session.add(new\_user)

db.session.commit()

flash("Account created successfully! Please login.", "success")

return redirect(url\_for("login"))

return render\_template("register.html")

# Login

@app.route("/login", methods=["GET", "POST"])

def login():

if request.method == "POST":

username = request.form.get("username")

password = request.form.get("password")

user = User.query.filter\_by(username=username).first()

if user and bcrypt.check\_password\_hash(user.password, password):

session["user\_id"] = user.id

session["username"] = user.username

flash("Login successful!", "success")

return redirect(url\_for("home"))

else:

flash("Invalid username or password", "error")

return redirect(url\_for("login"))

return render\_template("login.html")

# Logout

@app.route("/logout")

def logout():

session.clear()

flash("You have been logged out.", "info")

return redirect(url\_for("login"))

# Chatbot API

@app.route("/get", methods=["POST"])

def chatbot\_response():

if "user\_id" not in session:

return jsonify({"response": "Please log in to chat with me!"})

user\_message = request.json["message"]

bot\_reply = get\_response(user\_message)

return jsonify({"response": bot\_reply})

# Run app

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

login.html ------------File Name

– Login Page

<!DOCTYPE html>

<html>

<head>

<title>Login - Agriculture Chatbot</title>

<link rel="stylesheet" href="/static/style.css">

</head>

<body>

<div class="form-container">

<h2>Login</h2>

<form method="POST">

<input type="text" name="username" placeholder="Username" required><br>

<input type="password" name="password" placeholder="Password" required><br>

<button type="submit">Login</button>

</form>

<p>Don't have an account? <a href="/register">Register</a></p>

{% with messages = get\_flashed\_messages(with\_categories=true) %}

{% if messages %}

<div class="flash">

{% for category, message in messages %}

<p>{{ message }}</p>

{% endfor %}

</div>

{% endif %}

{% endwith %}

</div>

</body>

</html>

register.html -----------File Name

– Signup Page

<!DOCTYPE html>

<html>

<head>

<title>Register - Agriculture Chatbot</title>

<link rel="stylesheet" href="/static/style.css">

</head>

<body>

<div class="form-container">

<h2>Create Account</h2>

<form method="POST">

<input type="text" name="username" placeholder="Username" required><br>

<input type="password" name="password" placeholder="Password" required><br>

<button type="submit">Register</button>

</form>

<p>Already have an account? <a href="/login">Login</a></p>

{% with messages = get\_flashed\_messages(with\_categories=true) %}

{% if messages %}

<div class="flash">

{% for category, message in messages %}

<p>{{ message }}</p>

{% endfor %}

</div>

{% endif %}

{% endwith %}

</div>

</body>

</html>

index.html ------------File Name

– Chatbot Page

<!DOCTYPE html>

<html>

<head>

<title>Agriculture AI Chatbot</title>

<link rel="stylesheet" href="/static/style.css">

</head>

<body>

<div class="chat-container">

<h2>🌾 Agriculture AI Chatbot</h2>

<p>Welcome, {{ session['username'] }}! <a href="/logout">(Logout)</a></p>

<div id="chat-box"></div>

<div class="input-box">

<input type="text" id="user-input" placeholder="Type your message..." />

<button onclick="sendMessage()">Send</button>

</div>

</div>

<script>

async function sendMessage() {

const userInput = document.getElementById("user-input").value;

if (!userInput) return;

const chatBox = document.getElementById("chat-box");

chatBox.innerHTML += `<div class='user-msg'>👨‍🌾 You: ${userInput}</div>`;

const response = await fetch("/get", {

method: "POST",

headers: { "Content-Type": "application/json" },

body: JSON.stringify({ message: userInput })

});

const data = await response.json();

chatBox.innerHTML += `<div class='bot-msg'>🤖 Bot: ${data.response}</div>`;

document.getElementById("user-input").value = "";

chatBox.scrollTop = chatBox.scrollHeight;

}

</script>

</body>

</html>

style.css -----------File Name

body {

font-family: Arial, sans-serif;

background-color: #f0f8ff;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.chat-container, .form-container {

width: 400px;

background: #fff;

padding: 20px;

border-radius: 12px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);

}

h2 {

text-align: center;

color: green;

}

#chat-box {

height: 300px;

overflow-y: auto;

border: 1px solid #ccc;

padding: 10px;

margin-bottom: 10px;

border-radius: 8px;

background: #fafafa;

}

.user-msg {

text-align: right;

color: green;

margin: 5px 0;

}

.bot-msg {

text-align: left;

color: brown;

margin: 5px 0;

}

.input-box {

display: flex;

gap: 10px;

}

input[type="text"], input[type="password"] {

flex: 1;

padding: 8px;

border: 1px solid #ccc;

border-radius: 8px;

}

button {

background: green;

color: white;

padding: 8px 15px;

border: none;

border-radius: 8px;

cursor: pointer;

}

.flash p {

color: red;

font-size: 14px;

text-align: center;

}

– Styling