D:/MYPROJECT/app.py

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

import sqlite3

import re

import os

import pandas as pd

import pickle

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

app.secret\_key = 'your\_secret\_key'

@app.route('/')

def home():

return render\_template('home.html')

# Create DB if not exists

def init\_db():

# Users DB

conn = sqlite3.connect('users.db')

c = conn.cursor()

c.execute('''

CREATE TABLE IF NOT EXISTS users (

id INTEGER PRIMARY KEY AUTOINCREMENT,

username TEXT UNIQUE,

email TEXT UNIQUE,

password TEXT

)

''')

conn.commit()

conn.close()

# Questionnaire DB

conn = sqlite3.connect("glowcare.db")

c = conn.cursor()

c.execute('''CREATE TABLE IF NOT EXISTS new\_user\_responses (

id INTEGER PRIMARY KEY AUTOINCREMENT,

Age TEXT,

Gender TEXT,

SkinType TEXT,

Redness TEXT,

WaterIntake TEXT,

DietType TEXT,

SunExposure TEXT,

UsesSPF TEXT,

NaturalRemedyUsed TEXT,

CleanseFreq TEXT,

WashWaterTemp TEXT,

SleepHours TEXT,

StressFreq TEXT,

AwareOfVitaminE TEXT,

TriedDIY TEXT

)''')

conn.commit()

conn.close()

# Call once

init\_db()

@app.route('/about')

def about():

return render\_template('about.html')

@app.route('/login')

def login\_choice():

return render\_template('login.html')

# Placeholder routes for future development

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

def register():

message = ''

if request.method == 'POST':

username = request.form['username']

email = request.form['email']

password = request.form['password']

confirm = request.form['confirm']

# Password validation

if len(password) < 8 or not re.search(r'[A-Z]', password) or \

not re.search(r'[0-9]', password) or not re.search(r'[\W\_]', password):

message = "Password must be 8+ chars with uppercase, digit, and special character."

elif password != confirm:

message = "Passwords do not match."

else:

try:

with sqlite3.connect('users.db') as conn:

c = conn.cursor()

# check if username already exists

c.execute("SELECT \* FROM users WHERE username=?", (username,))

existing\_user = c.fetchone()

if existing\_user:

message = "❌ Username already exists. Please login."

else:

c.execute("INSERT INTO users (username, email, password) VALUES (?, ?, ?)",

(username, email, password))

conn.commit()

return redirect('/loginnew')

except sqlite3.IntegrityError:

message = "❌ User already exists."

return render\_template("register.html", message=message)

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

def loginnew():

if request.method == "POST":

username = request.form["username"].strip()

password = request.form["password"].strip()

conn = sqlite3.connect("users.db")

c = conn.cursor()

c.execute("SELECT password FROM users WHERE username=?", (username,))

row = c.fetchone()

conn.close()

if row:

stored\_password = row[0].strip()

if stored\_password == password:

session["user"] = username

return redirect("/are\_you\_ready")

else:

flash("❌ Incorrect password, please try again")

return redirect("/loginnew")

else:

flash("❌ Username not found")

return redirect("/loginnew")

return render\_template("loginnew.html")

import re

def is\_valid\_password(password):

return (len(password) >= 8 and

re.search(r'[A-Z]', password) and

re.search(r'\d', password) and

re.search(r'[!@#$%^&\*(),.?":{}|<>]', password))

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

def loginuser():

if request.method == "POST":

username = request.form["username"]

password = request.form["password"]

conn = sqlite3.connect("users.db")

c = conn.cursor()

c.execute("SELECT \* FROM users WHERE username=? AND password=?", (username, password))

user = c.fetchone()

conn.close()

if user:

session["user"] = username

return redirect("/returning\_user\_questionnaire")

else:

flash("Invalid username or password")

return redirect("/loginuser")

return render\_template("loginuser.html")

@app.route('/forgot\_password', methods=['GET', 'POST'])

def forgot\_password():

message = ""

if request.method == 'POST':

email = request.form['email']

new\_password = request.form['new\_password']

confirm\_password = request.form['confirm\_password']

if new\_password != confirm\_password:

message = "Passwords do not match."

elif not is\_valid\_password(new\_password):

message = "Password must be 8+ characters with uppercase, digit, and special character."

else:

conn = sqlite3.connect('users.db')

c = conn.cursor()

c.execute("SELECT \* FROM users WHERE email = ?", (email,))

user = c.fetchone()

if user:

c.execute("UPDATE users SET password = ? WHERE email = ?", (new\_password, email))

conn.commit()

conn.close()

return redirect('/loginnew')

else:

message = "Email not found."

return render\_template('forgot\_password.html', message=message)

@app.route("/are\_you\_ready", methods=["GET","POST"])

def are\_you\_ready():

return render\_template("are\_you\_ready.html")

# Ensure DB exists

def init\_db():

conn = sqlite3.connect("glowcare.db")

c = conn.cursor()

c.execute('''CREATE TABLE IF NOT EXISTS new\_user\_responses (

id INTEGER PRIMARY KEY AUTOINCREMENT,

Age TEXT,

Gender TEXT,

SkinType TEXT,

Redness TEXT,

WaterIntake TEXT,

DietType TEXT,

SunExposure TEXT,

UsesSPF TEXT,

NaturalRemedyUsed TEXT,

CleanseFreq TEXT,

WashWaterTemp TEXT,

SleepHours TEXT,

StressFreq TEXT,

AwareOfVitaminE TEXT,

TriedDIY TEXT

)''')

conn.commit()

conn.close()

init\_db()

# Load trained models

new\_user\_morning\_model = pickle.load(open("D:/MYPROJECT/new\_user\_morning\_model.pkl", "rb"))

new\_user\_evening\_model = pickle.load(open("D:/MYPROJECT/new\_user\_evening\_model.pkl", "rb"))

def format\_routine(routine, time\_of\_day):

steps = routine.split(", ")

# Ensure at least 3 items

while len(steps) < 3:

steps.append("suitable product")

if time\_of\_day == "Morning":

return f"Start your day by cleansing with a {steps[0].lower()}, then apply a {steps[1].lower()}, and finish with a {steps[2].lower()} to protect your skin."

else:

return f"Begin your evening by using a {steps[0].lower()}, follow with a {steps[1].lower()}, and complete your routine with a {steps[2].lower()} for overnight nourishment."

@app.route("/new\_user\_questionnaire", methods=["GET", "POST"])

def new\_user\_questionnaire():

if request.method == "POST":

# Get form data

form\_data = {k: request.form[k] for k in request.form}

print("\n=== DEBUG NEW USER ===")

print("Raw form data:", form\_data)

# Save to DB

conn = sqlite3.connect("glowcare.db")

c = conn.cursor()

c.execute('''INSERT INTO new\_user\_responses (

Age, Gender, SkinType, Redness, WaterIntake, DietType,

SunExposure, UsesSPF, NaturalRemedyUsed, CleanseFreq,

WashWaterTemp, SleepHours, StressFreq, AwareOfVitaminE, TriedDIY

) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)''',

tuple(form\_data.values()))

conn.commit()

conn.close()

# Prepare input

input\_df = pd.DataFrame([form\_data])

input\_df\_encoded = pd.get\_dummies(input\_df)

print("Encoded DataFrame (before alignment):\n", input\_df\_encoded.head())

# 🔹 Align with morning model

morning\_cols = new\_user\_morning\_model.feature\_names\_in\_

morning\_input = input\_df\_encoded.copy()

for col in morning\_cols:

if col not in morning\_input:

morning\_input[col] = 0

morning\_input = morning\_input[morning\_cols]

# 🔹 Align with evening model

evening\_cols = new\_user\_evening\_model.feature\_names\_in\_

evening\_input = input\_df\_encoded.copy()

for col in evening\_cols:

if col not in evening\_input:

evening\_input[col] = 0

evening\_input = evening\_input[evening\_cols]

print("Morning model features:", new\_user\_morning\_model.feature\_names\_in\_)

print("Evening model features:", new\_user\_evening\_model.feature\_names\_in\_)

print("Encoded DF for Morning:\n", morning\_input.head())

print("Encoded DF for Evening:\n", evening\_input.head())

print("Evening Input (aligned):\n", evening\_input.head())

print("Evening Model Columns Count:", len(evening\_cols))

print("Evening Prediction:", new\_user\_evening\_model.predict(evening\_input))

# Predict

morning\_prediction = new\_user\_morning\_model.predict(morning\_input)[0]

evening\_prediction = new\_user\_evening\_model.predict(evening\_input)[0]

print("Morning prediction (raw):", morning\_prediction)

print("Evening prediction (raw):", evening\_prediction)

# Format output

prediction = {

"Morning Routine": format\_routine(morning\_prediction, "Morning"),

"Evening Routine": format\_routine(evening\_prediction, "Evening")

}

return render\_template(

"results.html",

morning\_routine=prediction["Morning Routine"],

evening\_routine=prediction["Evening Routine"]

)

# Always return the questionnaire page for GET

return render\_template("new\_user\_questionnaire.html")

# Load returning user models

returning\_morning\_model = pickle.load(open("D:/MYPROJECT/returning\_morning\_model.pkl", "rb"))

returning\_evening\_model = pickle.load(open("D:/MYPROJECT/returning\_evening\_model.pkl", "rb"))

@app.route("/returning\_user\_questionnaire", methods=["GET", "POST"])

def returning\_user\_questionnaire():

if request.method == "POST":

form\_data = {k: request.form[k] for k in request.form}

# 🔹 Map form keys to dataset column names

rename\_map = {

"follows\_routine": "FollowsRoutine",

"irritation": "Irritation",

"skin\_change": "SkinChange",

"improvement": "Improvement",

"new\_diy": "TriedDIY"

}

mapped\_data = {rename\_map[k]: v for k, v in form\_data.items()}

# Save to DB (use original form keys if you want)

conn = sqlite3.connect("glowcare.db")

c = conn.cursor()

c.execute('''INSERT INTO returning\_user\_responses

(username, follows\_routine, irritation, skin\_change, improvement, new\_diy)

VALUES (?, ?, ?, ?, ?, ?)''',

(session.get("user"),

form\_data["follows\_routine"],

form\_data["irritation"],

form\_data["skin\_change"],

form\_data["improvement"],

form\_data["new\_diy"]))

conn.commit()

conn.close()

# 🔹 Prepare input for model with correct column names

input\_df = pd.DataFrame([mapped\_data])

input\_df\_encoded = pd.get\_dummies(input\_df)

# Align with training columns

model\_cols = returning\_morning\_model.feature\_names\_in\_

for col in model\_cols:

if col not in input\_df\_encoded:

input\_df\_encoded[col] = 0

input\_df\_encoded = input\_df\_encoded[model\_cols]

print("DEBUG morning input:\n", input\_df\_encoded.head())

morning\_pred = returning\_morning\_model.predict(input\_df\_encoded)[0]

print("DEBUG morning prediction:", morning\_pred)

evening\_pred = returning\_evening\_model.predict(input\_df\_encoded)[0]

print("DEBUG evening prediction:", evening\_pred)

print("DEBUG Encoded DataFrame:\n", input\_df\_encoded.head())

# Predict

morning\_pred = returning\_morning\_model.predict(input\_df\_encoded)[0]

evening\_pred = returning\_evening\_model.predict(input\_df\_encoded)[0]

# Format output

morning\_routine = format\_routine(morning\_pred, "Morning")

evening\_routine = format\_routine(evening\_pred, "Evening")

return render\_template("returning\_results.html",

morning=morning\_routine,

evening=evening\_routine)

return render\_template("returning\_user\_questionnaire.html")

return render\_template("returning\_user\_questionnaire.html")

@app.route('/questionnaire')

def questionnaire():

return "<h2>Questionnaire Coming Soon!</h2>"

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**D:/MYPROJECT/templtes/home.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>GlowCare</title>

<style>

body, html {

height: 100%;

margin: 0;

font-family: 'Segoe UI', sans-serif;

background: url('https://images.unsplash.com/photo-1617385908365-4ccaa9abf4c0?ixlib=rb-4.0.3&auto=format&fit=crop&w=1470&q=80')

no-repeat center center fixed;

background-size: cover;

}

.overlay {

position: absolute;

width: 100%;

height: 100%;

background: rgba(255, 255, 255, 0.4); /\* soft white overlay \*/

backdrop-filter: blur(6px);

}

.navbar {

position: absolute;

top: 20px;

right: 30px;

}

.navbar a {

text-decoration: none;

color: #2d3436;

background: rgba(255, 255, 255, 0.6);

border: 1px solid #ffffff;

padding: 10px 18px;

margin-left: 10px;

border-radius: 10px;

font-weight: bold;

transition: 0.3s;

}

.navbar a:hover {

background-color: #ffcccc;

color: #2d3436;

border-color: #ff9999;

}

.centered-text {

position: absolute;

top: 45%;

left: 50%;

transform: translate(-50%, -50%);

text-align: center;

color: #2d3436;

}

.centered-text h1 {

font-size: 55px;

font-weight: 800;

margin-bottom: 20px;

background: linear-gradient(90deg, #ff9a9e, #fad0c4);

-webkit-background-clip: text;

-webkit-text-fill-color: transparent;

}

.centered-text p {

font-size: 22px;

color: #444;

}

</style>

</head>

<body>

<div class="overlay"></div>

<div class="navbar">

<a href="/login">Login</a>

<!--<a href="/admin">Admin</a>-->

<a href="/about">About</a>

</div>

<div class="centered-text">

<h1>Welcome to GlowCare</h1>

<p>Your personal skincare assistant</p>

</div>

</body>

</html>

**D:/MYPROJECT/templtes/about.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>About GlowCare</title>

<style>

body {

margin: 0;

padding: 0;

font-family: 'Segoe UI', sans-serif;

background: linear-gradient(135deg, #fceabb, #f8b500);

min-height: 100vh;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

}

.container {

max-width: 800px;

background: #fffbea;

padding: 40px 50px;

border-radius: 20px;

box-shadow: 0px 12px 30px rgba(0, 0, 0, 0.15);

text-align: center;

margin: 20px;

animation: fadeIn 1s ease-in-out;

}

.text-section h1 {

font-size: 38px;

color: #6a4c93;

margin-bottom: 20px;

}

.text-section p {

font-size: 18px;

line-height: 1.8;

color: #444;

margin-bottom: 18px;

text-align: justify;

}

.home-link {

margin-top: 20px;

}

.home-link a {

text-decoration: none;

color: white;

background-color: #ff7b54;

padding: 12px 28px;

border-radius: 10px;

font-weight: bold;

font-size: 16px;

box-shadow: 0px 4px 10px rgba(0,0,0,0.2);

transition: 0.3s ease;

}

.home-link a:hover {

background-color: #ff5722;

transform: translateY(-2px);

}

@keyframes fadeIn {

from { opacity: 0; transform: translateY(30px); }

to { opacity: 1; transform: translateY(0); }

}

@media (max-width: 768px) {

.container {

padding: 25px 20px;

}

.text-section h1 {

font-size: 30px;

}

.text-section p {

font-size: 16px;

}

}

</style>

</head>

<body>

<div class="container">

<div class="text-section">

<h1>About GlowCare</h1>

<p>

GlowCare is an intelligent skincare recommendation system designed to provide users with personalized morning and evening routines.

By collecting information about lifestyle, skin type, and skincare habits, GlowCare uses smart logic and machine learning models

to suggest routines that are best suited for each individual.

</p>

<p>

New users receive their first customized skincare routine after answering a short questionnaire,

while returning users get improved suggestions based on their past routines and progress.

This ensures that every recommendation evolves with the user’s changing skin needs.

</p>

</div>

<div class="home-link">

<a href="/">← Back to Home</a>

</div>

</div>

</body>

</html>

**D:/MYPROJECT/templtes/login.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Choose Login Type - GlowCare</title>

<style>

body {

margin: 0;

padding: 0;

font-family: 'Segoe UI', sans-serif;

background: linear-gradient(to right, #fceff9, #e0f7fa);

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

}

.container {

display: flex;

flex-direction: column;

align-items: center;

gap: 30px;

}

h1 {

color: #444;

font-size: 32px;

margin-bottom: 10px;

}

.card-container {

display: flex;

gap: 30px;

flex-wrap: wrap;

justify-content: center;

}

.card {

background-color: #fff;

padding: 40px 30px;

width: 260px;

border-radius: 20px;

box-shadow: 0 10px 25px rgba(0, 0, 0, 0.1);

text-align: center;

transition: transform 0.3s ease;

}

.card:hover {

transform: translateY(-5px);

}

.card h2 {

color: #333;

font-size: 24px;

margin-bottom: 15px;

}

.card p {

color: #666;

font-size: 16px;

margin-bottom: 20px;

}

.card a {

text-decoration: none;

background-color: #a5d8ff;

color: #003b57;

padding: 10px 20px;

border-radius: 8px;

font-weight: bold;

transition: background-color 0.3s ease;

}

.card a:hover {

background-color: #74c0fc;

}

.back-home {

margin-top: 20px;

}

.back-home a {

text-decoration: none;

color: #555;

font-weight: bold;

font-size: 16px;

}

.back-home a:hover {

text-decoration: underline;

}

</style>

</head>

<body>

<div class="container">

<h1>Welcome to GlowCare</h1>

<div class="card-container">

<div class="card">

<h2>New User</h2>

<p>If you're visiting for the first time, register here.</p>

<a href="/register">Register</a>

</div>

<div class="card">

<h2>Returning User</h2>

<p>If you already have an account, log in here.</p>

<a href="/loginuser">Login</a>

</div>

</div>

<div class="back-home">

<a href="/">← Back to Home</a>

</div>

</div>

</body>

</html>

**D:/MYPROJECT/templtes/register.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>GlowCare | Register</title>

<style>

body {

margin: 0;

font-family: 'Segoe UI', sans-serif;

background: linear-gradient(to right, #f9f4f1, #ffe8ec);

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.register-box {

background-color: white;

padding: 40px;

border-radius: 20px;

box-shadow: 0 10px 25px rgba(0, 0, 0, 0.1);

width: 400px;

}

h2 {

text-align: center;

color: #ff6f91;

margin-bottom: 20px;

}

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

width: 100%;

padding: 12px;

margin: 8px 0 15px 0;

border: 1px solid #ccc;

border-radius: 10px;

box-sizing: border-box;

}

input[type=submit] {

width: 100%;

background-color: #ff6f91;

color: white;

padding: 12px;

border: none;

border-radius: 10px;

cursor: pointer;

font-size: 16px;

}

input[type=submit]:hover {

background-color: #ff4f75;

}

.message {

margin-top: 15px;

text-align: center;

color: red;

}

.link {

margin-top: 10px;

text-align: center;

}

.link a {

color: #333;

text-decoration: none;

}

.link a:hover {

text-decoration: underline;

}

</style>

</head>

<body>

<div class="register-box">

<h2>Create Your Account</h2>

<form method="POST" action="/register">

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

<input type="email" name="email" placeholder="Email" required>

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

<input type="password" name="confirm" placeholder="Confirm Password" required>

<input type="submit" value="Register">

</form>

{% if message %}

<div class="message">{{ message }}</div>

{% endif %}

<div class="link">

Already have an account? <a href="/loginnew">Login here</a>

</div>

</div>

</body>

</html>

**D:/MYPROJECT/templtes/loginnew.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Login - GlowCare</title>

<style>

\* {

box-sizing: border-box;

font-family: 'Segoe UI', sans-serif;

}

body {

margin: 0;

display: flex;

height: 100vh;

}

.left {

flex: 1;

background: linear-gradient(135deg, #c8e6c9, #a5d6a7);

display: flex;

justify-content: center;

align-items: center;

padding: 2rem;

color: #2e7d32;

font-size: 2rem;

font-weight: bold;

}

.right {

flex: 1;

display: flex;

align-items: center;

justify-content: center;

background-color: #f8f8f8;

}

.form-container {

width: 100%;

max-width: 350px;

padding: 30px;

background-color: #fff;

border-radius: 15px;

box-shadow: 0 8px 16px rgba(0,0,0,0.15);

}

h2 {

margin-bottom: 20px;

color: #388e3c;

text-align: center;

}

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

width: 100%;

padding: 10px;

margin: 10px 0 20px 0;

border: 1px solid #ccc;

border-radius: 8px;

}

button {

width: 100%;

padding: 12px;

background-color: #66bb6a;

border: none;

color: white;

border-radius: 8px;

font-size: 16px;

cursor: pointer;

}

button:hover {

background-color: #4caf50;

}

.bottom-links {

margin-top: 15px;

text-align: center;

}

.bottom-links a {

color: #388e3c;

text-decoration: none;

}

.error {

color: red;

font-size: 0.9rem;

text-align: center;

}

</style>

</head>

<body>

<div class="left">

GlowCare ✨

</div>

<div class="right">

<form method="POST" class="form-container" action="/are\_you\_ready">

<h2>User Login</h2>

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

{% if messages %}

<div class="error">{{ messages[0] }}</div>

{% endif %}

{% endwith %}

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

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

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

<div class="bottom-links">

<!--<p><a href="/forgot\_password">Forgot Password?</a></p>-->

<!--<p>New user? <a href="/register">Register here</a></p>-->

</div>

</form>

</div>

</body>

</html>

**D:/MYPROJECT/templtes/forgot\_password.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Forgot Password</title>

<style>

body {

background-color: #fefefc;

font-family: 'Segoe UI', sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.container {

background: #fff;

padding: 40px;

border-radius: 20px;

box-shadow: 0 8px 20px rgba(0, 0, 0, 0.1);

width: 400px;

}

h2 {

color: #5e7755;

margin-bottom: 20px;

}

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

width: 100%;

padding: 10px;

margin: 8px 0;

border: 1px solid #ccc;

border-radius: 10px;

}

button {

width: 100%;

padding: 10px;

background-color: #8fa98e;

color: white;

border: none;

border-radius: 10px;

margin-top: 10px;

cursor: pointer;

}

.message {

margin-top: 15px;

color: red;

text-align: center;

}

</style>

</head>

<body>

<div class="container">

<h2>Reset Your Password</h2>

<form method="POST">

<input type="email" name="email" placeholder="Enter your registered email" required>

<input type="password" name="new\_password" placeholder="Enter new password" required>

<input type="password" name="confirm\_password" placeholder="Confirm new password" required>

<button type="submit">Reset Password</button>

</form>

{% if message %}

<div class="message">{{ message }}</div>

{% endif %}

</div>

</body>

</html>

**D:/MYPROJECT/templtes/are\_you\_ready.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Are You Ready?</title>

<style>

body {

margin: 0;

font-family: 'Segoe UI', sans-serif;

background: linear-gradient(135deg, #c8e6c9, #f1f8e9);

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.box {

text-align: center;

background: white;

padding: 40px;

border-radius: 25px;

box-shadow: 0 8px 20px rgba(0,0,0,0.2);

max-width: 450px;

}

h1 {

font-size: 28px;

color: #4e6e50;

margin-bottom: 20px;

}

p {

font-size: 18px;

color: #666;

margin-bottom: 30px;

}

button {

background-color: #7fae7c;

color: white;

padding: 12px 30px;

border: none;

border-radius: 10px;

font-size: 16px;

cursor: pointer;

}

button:hover {

background-color: #6d9b6a;

}

</style>

</head>

<body>

<div class="box">

<h1>Are you ready to begin your GlowCare journey?</h1>

<p>We’re excited to help you discover the best skincare tailored just for you!</p>

<form action="/new\_user\_questionnaire" method="get">

<button type="submit">Let’s Start</button>

</form>

</div>

</body>

</html>

**D:/MYPROJECT/templtes/loginuser.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Returning User Login - GlowCare</title>

<style>

\* { box-sizing: border-box; font-family: 'Segoe UI', sans-serif; }

body { margin: 0; display: flex; height: 100vh; }

.left {

flex: 1;

background: linear-gradient(135deg, #c8e6c9, #a5d6a7);

display: flex; justify-content: center; align-items: center;

padding: 2rem; color: #2e7d32; font-size: 2rem; font-weight: bold;

}

.right { flex: 1; display: flex; align-items: center; justify-content: center; background-color: #f8f8f8; }

.form-container {

width: 100%; max-width: 350px; padding: 30px;

background-color: #fff; border-radius: 15px;

box-shadow: 0 8px 16px rgba(0,0,0,0.15);

}

h2 { margin-bottom: 20px; color: #388e3c; text-align: center; }

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

width: 100%; padding: 10px; margin: 10px 0 20px 0;

border: 1px solid #ccc; border-radius: 8px;

}

button {

width: 100%; padding: 12px; background-color: #66bb6a;

border: none; color: white; border-radius: 8px; font-size: 16px;

cursor: pointer;

}

button:hover { background-color: #4caf50; }

.bottom-links { margin-top: 15px; text-align: center; }

.bottom-links a { color: #388e3c; text-decoration: none; }

.error { color: red; font-size: 0.9rem; text-align: center; }

</style>

</head>

<body>

<div class="left"><br>GlowCare ✨</div>

<div class="right">

<form method="POST" class="form-container" action="/loginuser">

<h2>Returning User Login</h2>

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

{% if messages %}

<div class="error">{{ messages[0] }}</div>

{% endif %}

{% endwith %}

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

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

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

<div class="bottom-links">

<p><a href="/forgot\_password">Forgot Password?</a></p>

</div>

</form>

</div>

</body>

</html>

**D:/MYPROJECT/templtes/new\_user\_questionnaire.html**

<!DOCTYPE html>

<html>

<head>

<title>New User Questionnaire</title>

<style>

body {

font-family: 'Segoe UI', sans-serif;

background: linear-gradient(135deg, #c3ecb2, #7ee8fa);

padding: 30px;

}

form {

background: white;

padding: 20px;

border-radius: 12px;

max-width: 600px;

margin: auto;

box-shadow: 0px 5px 15px rgba(0,0,0,0.1);

}

h2 {

text-align: center;

color: #2e7d32;

}

label {

display: block;

margin-top: 10px;

font-weight: bold;

}

select {

width: 100%;

padding: 8px;

margin-top: 5px;

border-radius: 8px;

border: 1px solid #ccc;

}

button {

margin-top: 15px;

padding: 10px;

background: #66bb6a;

color: white;

border: none;

width: 100%;

border-radius: 8px;

}

</style>

</head>

<body>

<h2>New User Questionnaire</h2>

<form method="POST">

<label>Age</label>

<select name="Age">

<option>Below 18</option><option>18–25</option><option>26–35</option>

<option>36–45</option><option>46–60</option><option>60+</option>

</select>

<label>Gender</label>

<select name="Gender">

<option>Female</option><option>Male</option>

</select>

<label>Skin Type</label>

<select name="Skin Type">

<option>Oily</option><option>Dry</option><option>Combination</option>

<option>Sensitive</option><option>Normal</option>

</select>

<label>Redness or Inflammation</label>

<select name="Redness or Inflammation">

<option>Yes</option><option>No</option><option>Sometimes</option>

</select>

<label>Water Intake</label>

<select name="Water Intake">

<option>3–4</option><option>5–6</option><option>7–8</option><option>9 or more</option>

</select>

<label>Diet Type</label>

<select name="Diet Type">

<option>High in Sugar</option><option>Balanced (Fruits + Veggies)</option>

<option>Processed or Fast Food</option><option>Vegetarian</option>

<option>High in Protein</option>

</select>

<label>Sun Exposure</label>

<select name="Sun Exposure">

<option>Daily</option><option>Often</option><option>Rarely</option>

</select>

<label>Uses SPF</label>

<select name="Uses SPF">

<option>Yes</option><option>No</option>

</select>

<label>Natural Remedies Used</label>

<select name="Natural Remedies Used">

<option>Turmeric</option><option>Aloe Vera</option><option>Oatmeal</option>

<option>Rose Water</option><option>Besan (Gram Flour)</option><option>None</option>

</select>

<label>Cleansing Frequency</label>

<select name="Cleansing Frequency">

<option>Once</option><option>Twice</option>

</select>

<label>Water Temperature</label>

<select name="Water Temperature">

<option>Cold</option><option>Warm</option><option>Hot</option>

</select>

<label>Sleep Hours</label>

<select name="Sleep Hours">

<option>4 or fewer</option><option>5–6</option><option>7–8</option>

</select>

<label>Stress Frequency</label>

<select name="Stress Frequency">

<option>Rarely</option><option>Moderate</option><option>Often</option><option>High</option>

</select>

<label>Vitamin E Awareness</label>

<select name="Vitamin E Awareness">

<option>Yes</option><option>No</option>

</select>

<label>Tried DIY Skincare Recently</label>

<select name="Tried DIY Skincare Recently">

<option>Yes</option><option>No</option>

</select>

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

</form>

</body>

</html>

**D:/MYPROJECT/templtes/results.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Your Skincare Routine</title>

<style>

body {

font-family: 'Segoe UI', sans-serif;

background: linear-gradient(135deg, #fbc2eb, #a6c1ee);

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

}

.card {

background: #fff;

padding: 30px;

border-radius: 18px;

box-shadow: 0 8px 18px rgba(0,0,0,0.15);

text-align: center;

width: 450px;

animation: fadeIn 1s ease-in-out;

}

h2 {

color: #ff6f61;

margin-bottom: 20px;

}

button {

margin: 10px;

padding: 12px 20px;

font-size: 16px;

border: none;

border-radius: 8px;

cursor: pointer;

background-color: #ff8a80;

color: white;

transition: 0.3s;

}

button:hover {

background-color: #ff5252;

}

.routine {

margin-top: 20px;

font-size: 1.05rem;

display: none;

line-height: 1.6;

color: #444;

}

.home-link a {

text-decoration: none;

color: #ff6f61; /\* pastel coral to match your heading \*/

background-color: #ffe6e1; /\* soft pastel background \*/

padding: 10px 20px;

border-radius: 8px;

font-weight: 500;

transition: 0.3s ease;

display: inline-block;

margin-top: 20px;

}

.home-link a:hover {

background-color: #ff6f61; /\* coral background on hover \*/

color: white; /\* white text on hover \*/

}

@keyframes fadeIn {

from {opacity: 0; transform: translateY(20px);}

to {opacity: 1; transform: translateY(0);}

}

</style>

<script>

function showRoutine(type) {

document.getElementById("morning").style.display = "none";

document.getElementById("evening").style.display = "none";

document.getElementById(type).style.display = "block";

}

</script>

</head>

<body>

<div class="card">

<h2>Your Recommended Skincare Routine ✨</h2>

<button onclick="showRoutine('morning')">Morning Routine</button>

<button onclick="showRoutine('evening')">Evening Routine</button>

<div id="morning" class="routine">

<strong>Morning Routine:</strong> {{ morning\_routine }}

</div>

<div id="evening" class="routine">

<strong>Evening Routine:</strong> {{ evening\_routine }}

</div>

<div class="home-link">

<a href="/">← Back to Home</a>

</div>

</div>

</body>

</html>

**D:/MYPROJECT/ train\_new\_user\_model.py**

# train\_new\_user\_model.py

import pandas as pd

from sklearn.ensemble import RandomForestClassifier

import pickle

# Load dataset

df = pd.read\_csv("D:/MYPROJECT/data/large\_skincare\_users\_dataset\_with\_age.csv")

# Define features and targets

X\_columns = [

"Age", "Gender", "SkinType", "Redness", "WaterIntake",

"DietType", "SunExposure", "UsesSPF", "NaturalRemedyUsed",

"CleanseFreq", "WashWaterTemp", "SleepHours",

"StressFreq", "AwareOfVitaminE", "TriedDIY"

]

y\_morning = df["MorningRoutine"]

y\_evening = df["EveningRoutine"]

# Encode categorical variables

X\_encoded = pd.get\_dummies(df[X\_columns])

# Train models

morning\_model = RandomForestClassifier()

evening\_model = RandomForestClassifier()

morning\_model.fit(X\_encoded, y\_morning)

evening\_model.fit(X\_encoded, y\_evening)

# Save models

pickle.dump(morning\_model, open("new\_user\_morning\_model.pkl", "wb"))

pickle.dump(evening\_model, open("new\_user\_evening\_model.pkl", "wb"))

print("✅ Models trained and saved successfully!")

**D:/MYPROJECT/templtes/returning\_user\_questionnaire.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Returning User Questionnaire - GlowCare</title>

<style>

body {

font-family: 'Segoe UI', sans-serif;

background: linear-gradient(135deg, #d4fc79, #96e6a1);

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

}

.container {

background: white;

border-radius: 12px;

padding: 30px;

box-shadow: 0 8px 20px rgba(0,0,0,0.1);

max-width: 600px;

width: 100%;

}

h1 {

text-align: center;

color: #2e7d32;

margin-bottom: 20px;

}

label {

font-weight: bold;

display: block;

margin: 15px 0 5px;

}

select, input[type="radio"], input[type="checkbox"] {

margin-bottom: 15px;

width: 100%;

padding: 8px;

border-radius: 8px;

border: 1px solid #ccc;

}

button {

background-color: #4caf50;

color: white;

padding: 12px;

border: none;

border-radius: 8px;

cursor: pointer;

width: 100%;

font-size: 16px;

margin-top: 20px;

}

button:hover {

background-color: #388e3c;

}

</style>

</head>

<body>

<div class="container">

<h1>Returning User Questionnaire ✨</h1>

<form method="POST">

<label>Have you been following your recommended skincare routine?</label>

<select name="follows\_routine" required>

<option value="">Select</option>

<option>Yes</option>

<option>No</option>

</select>

<label>Have you experienced any irritation or discomfort?</label>

<select name="irritation" required>

<option value="">Select</option>

<option>Yes</option>

<option>No</option>

</select>

<label>Recent changes in your skin?</label>

<select name="skin\_change" required>

<option value="">Select</option>

<option>Flaky or Peeling Skin</option>

<option>Uneven Skin Tone</option>

<option>Skin Getting Worse</option>

<option>Skin Getting Better</option>

</select>

<label>Has your skin improved in the past 2 weeks?</label>

<select name="improvement" required>

<option value="">Select</option>

<option>Yes</option>

<option>No</option>

</select>

<label>Have you tried any new home remedies recently?</label>

<select name="new\_diy" required>

<option value="">Select</option>

<option>Yes</option>

<option>No</option>

</select>

<button type="submit">Submit Questionnaire</button>

</form>

</div>

</body>

</html>

**D:/MYPROJECT/ training\_returning\_model.py**

# train\_returning\_model.py

import pandas as pd

from sklearn.ensemble import RandomForestClassifier

import pickle

# Load dataset

df = pd.read\_csv("D:/MYPROJECT/data/large\_skincare\_users\_dataset\_with\_age.csv")

# Keep only columns relevant to returning user

X = df[["FollowsRoutine", "Irritation",

"SkinChange", "Improvement", "TriedDIY"]]

y\_morning = df["MorningRoutine"]

y\_evening = df["EveningRoutine"]

# Encode categorical variables

X\_encoded = pd.get\_dummies(X)

# Train separate models for morning & evening

morning\_model = RandomForestClassifier()

evening\_model = RandomForestClassifier()

morning\_model.fit(X\_encoded, y\_morning)

evening\_model.fit(X\_encoded, y\_evening)

# Save models

pickle.dump(morning\_model, open("returning\_morning\_model.pkl", "wb"))

pickle.dump(evening\_model, open("returning\_evening\_model.pkl", "wb"))

print("✅ Models trained and saved successfully!")

**D:/MYPROJECT/templtes/returning\_results.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Your Updated Skincare Routine</title>

<style>

body {

font-family: 'Segoe UI', sans-serif;

background: linear-gradient(135deg, #f6d365, #fda085);

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

}

.card {

background: #fff;

padding: 30px;

border-radius: 18px;

box-shadow: 0 8px 18px rgba(0,0,0,0.15);

text-align: center;

width: 450px;

animation: fadeIn 1s ease-in-out;

}

h2 {

color: #ff7043;

margin-bottom: 20px;

}

button {

margin: 10px;

padding: 12px 20px;

font-size: 16px;

border: none;

border-radius: 8px;

cursor: pointer;

background-color: #ff8a65;

color: white;

transition: 0.3s;

}

button:hover {

background-color: #ff5722;

}

.routine {

margin-top: 20px;

font-size: 1.05rem;

display: none;

line-height: 1.6;

color: #444;

}

.home-link a {

text-decoration: none;

color: #ff7043;

background-color: #ffe6e1;

padding: 10px 20px;

border-radius: 8px;

font-weight: 500;

transition: 0.3s ease;

display: inline-block;

margin-top: 20px;

}

.home-link a:hover {

background-color: #ff7043;

color: white;

}

@keyframes fadeIn {

from {opacity: 0; transform: translateY(20px);}

to {opacity: 1; transform: translateY(0);}

}

</style>

<script>

function showRoutine(type) {

document.getElementById("morning").style.display = "none";

document.getElementById("evening").style.display = "none";

document.getElementById(type).style.display = "block";

}

</script>

</head>

<body>

<div class="card">

<h2>Your Updated Skincare Routine ✨</h2>

<button onclick="showRoutine('morning')">Morning Routine</button>

<button onclick="showRoutine('evening')">Evening Routine</button>

<div id="morning" class="routine">

<strong>Morning Routine:</strong> {{ morning }}

</div>

<div id="evening" class="routine">

<strong>Evening Routine:</strong> {{ evening }}

</div>

<div class="home-link">

<a href="/">← Back to Home</a>

</div>

</div>

</body>

</html>