FINAL YEAR GRADUATION PROJECT



Department of Computer Science

Faculty Information Technology and Computer Science Yarmouk University

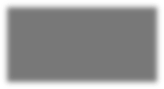
Irbid, Jordan

**Prepared by :**

* Sara AL-Qudah
* Batlla Al-Dalgamoni
* Yaqeen Hazeem

**Supervised by :**

Dr.Raed Alkhatib



Mental Health ChatBot

AI-Based personalized Mental Health ChatBot

Mental Health ChatBot| Graduation Project | Second Semester 2024/2025

### 

### Contact Information

This project report is submitted to the Department of Computer Science at Yarmouk University in partial fulfilment of the requirements for the degree of Bachelor of Information Technology in Computer Science.

###### Author(s):

###### - Sara yahia alqudah (2021901097)

***Address: Ajloun***

***E-mail:*** [***saraqudah911@gmail.com***](mailto:saraqudah911@gmail.com)

***-Battla mamoun aldalgamoni(2021801038)***

***Address: Irbid***

***E-mail:*** [**batllamamon29@gmail.com**](mailto:batllamamon29@gmail.com)

**- Yaqeen ayman hazeem (2021901147)**

***Address: Irbid***

***E-mail:*** [**yaqeenhazeem2003@gmail.com**](mailto:yaqeenhazeem2003@gmail.com)

###### University supervisor(s):

***Dr. Ra’ed AlKhateeb***

***Department of Computer Science***

***Faculty of Information Technology and Computer Sciences***

***Yarmouk University***

**Intellectual Property Right Declaration**

This is to declare that the work under the supervision of **Dr. *Ra’ed AlKhateeb***having title **“*Mental Health Chat Bot*”** carried out in partial fulfilment of the requirements of Bachelor of Information Technology in Computer Science, is the sole property of the Yarmouk University and the respective supervisor and is protected under the intellectual property right laws and conventions. It can only be considered/ used for purposes like extension for further enhancement, product development, adoption for commercial/organizational usage, etc., with the permission of the University and respective supervisor.

***Date:***

|  |  |
| --- | --- |
| ***Author(s):*** |  |
| ***Name:* Sara yahia alqudah** ***Signature:*** |
| ***Name: Battla mamoun aldalgamoni Signature:*** |  |
| ***Name:* Yaqeen ayman hazeem *Signature:*** |

***Supervisor(s):***

***Dr. Ra’ed AlKhateeb***

***Signature:***

## ABSTRACT

AI-Based personalized Mental Health Chat Bot

In our project documentation, we detail and demonstrate the functionalities of the **Mental Health Chatbot System**, developed to support emotional assessment and mental health monitoring for users,  
This system is implemented using **Python programming, machine learning techniques, and emotional state prediction models**.

The primary goal of our project is to provide a supportive tool that detects changes in a user's emotional state over time, offers motivational feedback, and encourages treatment adherence. This chatbot aims to assist users in **tracking their mental well-being**, **receiving timely emotional insights**, and ultimately **enhancing their mental health outcomes** through intelligent, user-friendly interaction.

***Table of contents :***

1. **Introduction ………………………………………………………………………………………………………………….7**

**1.1. problem statement….……………………………………………………………………………………………….7**

**1.2. Objectives of the study ….……………………………………………………………………………………….7**

**1.3. Methodology…………………………………………………………………………………………………………...8**

**1.4. Tools and Technologies used……………………………………………………………………………….…8**

1. **What Is Mental Health ? ……………………………………………………………………………………………….9**
   1. **Components of Mental Health …………………………………………………………………………..9**
   2. **Why is mental health important…………………………………………………………………………9**
2. **Common Mental Health Diagnosis ……………………………………………………………………………….10**
   1. **Important Mental Health Statistic…. …………………………………………………………………10**
3. **Factors Affecting Mental Health …………………………………………………………………………………..11**

**4.1 The Role of Technology In Mental Health………………………………………………………………11**

**5. Analysis of Patient Data …………………………………………………………………………………………………12**

1. **Advice of Treatment ……………………………………………………………………………………………………..14**
2. **Examples of Depression Cases and How They Were Treated………………………………………15**
3. **Analysis of HTML Elements in the Project…………………………………………………………………..16**
   1. **Head Section …………………………………………………………………………………………………………………………..16**
   2. **Body Section……………………………………………………………………………………………………………………………17**
   3. **Form Section(HTML)……………………………………………………………………………………………………………..18**
   4. **Input Fields…………………………………………………………………………………………………………………………….19**
   5. **Submit Button…………………………………………………………………………………………………………………………20**

**9.Flow chart of Model …………………………………………………………………………………………………………..24**

**10.** **Visual Results of the Implemented Code……………………………………………………………………….25**

**11.** **Lifestyle and Mental Health Factors……………………………………………………………………………….29**

**12. Backend Implementation Using Flask…………………………………………………………………………...30**

**13.Refernces ……………………………………………………………………………………………………………………….….43**

***Introduction:***

*During the last few years, mental well-being has become one of the most pressing topics that are gaining increasing global attention. As a result of technological advancements, psychological assistance now has the form of digital services, such as chat bots. The aim of this research is to design a smart chat bot to help diagnose mild mental health conditions and provide initial therapeutic guidance or advice, with the help of artificial intelligence technologies.*

***1.1 Problem Statement:***

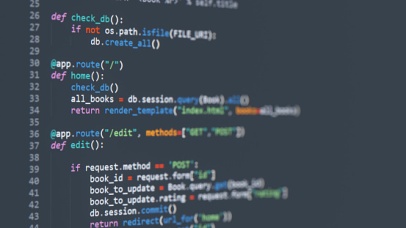
*The majority of individuals undergo mental illnesses such as depression and anxiety without the help of professional psychology due to financial, social, or geographical constraints. There is therefore a need for a smart system with the ability to provide initial and quick psychological help, which has the potential to transform individuals' lives.*

***1.2 Objectives of the Study:***

* *To create an intelligent chat bot that can grasp and analyze users' psychological status from their interactions.*
* *To provide smart responses and preliminary therapy suggestions to users.*
* *In order to support quick and easy access to psychological help without the need for face-to-face human communication*

***1.3 Methodology:***

*It will be developed in Python with Flask as the web framework  for backend architecture and HTML, CSS, and JavaScript for frontend user interface. The chat bot will be trained on a set of psychological data (e.g., questionnaires and previous conversations) to generate intelligent, human-like answers*

**

***1.4 Tools and Technologies Used:***

**

1. ***What is Mental Health?***

*Mental health is a state of well being that allows a person to deal with the usual stresses of life, work successfully, and contribute to the community. It is not just the absence of mental illness but also includes emotional, psychological, and social well-being*

***2.1 Components of Mental Health:***

* *Emotional well-being: Ability to identify and express feelings in a healthy way.*
* *Social well-being: Building and maintaining healthy connections and social support.*
* *Psychological well-being: Good thinking, making decisions, and learning from experience*

***2.2 Why is Mental Health Important?***

* *It shapes the way we think, feel, and act.*
* *It plays a key part in how we cope with stress, connect with others, and make decisions.*
* *Equilibrated mental health adds to overall life quality as well as productivity in the workplace or school*

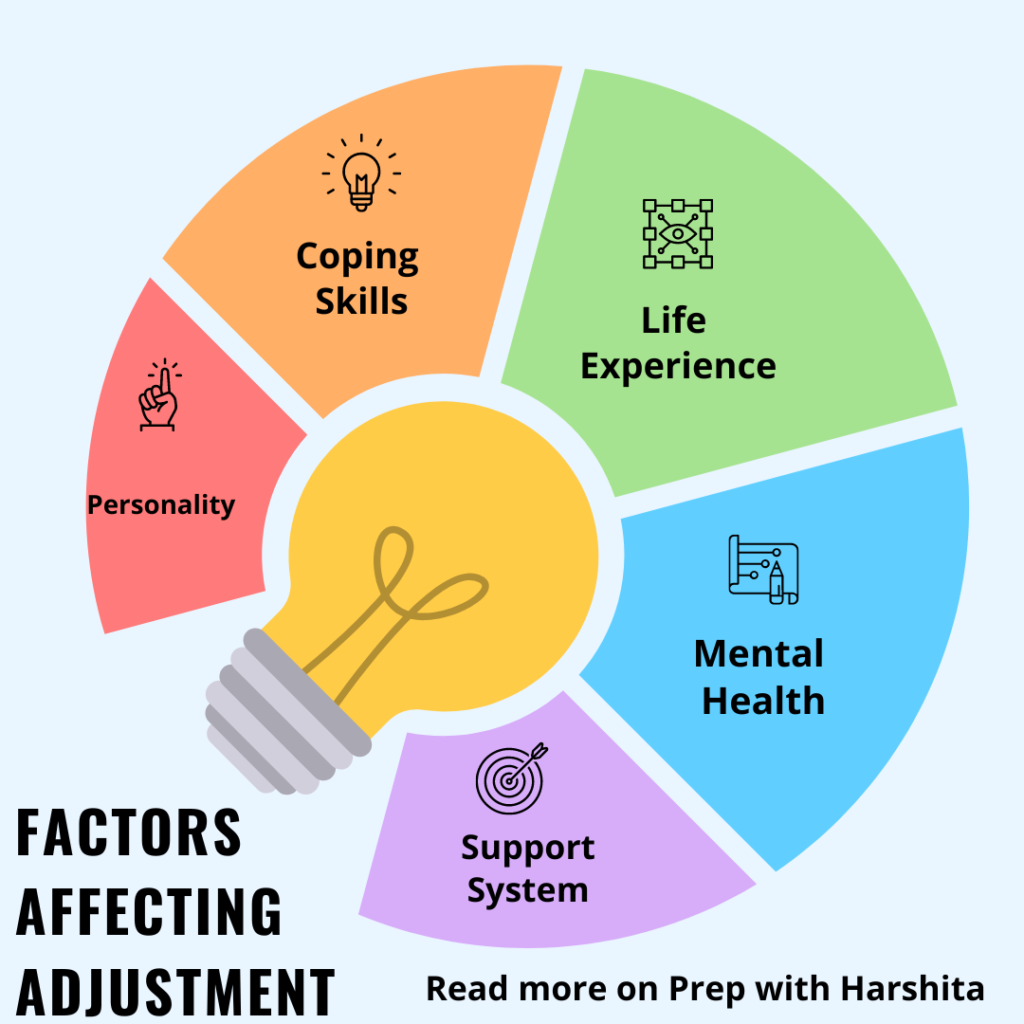
1. *Common Mental Health Diagnosis:*

|  |  |
| --- | --- |
| Diagnosis | Description |
| Bipolar Disorder | Extreme mood swings disorder |
| Major Depressive Disorder | Persistent deep sadness disorder |
| Panic Disorder | Sudden intense fear disorder |
| Generalized Anxiety | Chronic excessive worry disorder |

* 1. ***Important Mental Health Statistics:***

**

***4.Factors Affecting Mental Health:***

**

**4.1 The Role of Technology in Mental Health:**

* Nowadays, technology is being used ever more to improve mental health through:
* Self-help apps Mental health bots   
  (such as this one)Online therapy sessions Peer support forums and groups

1. **AI and Machine Learning-Based Analysis of Chat bot Patient Data for Mental Health Diagnosis and Treatment**

**- Patient ID:** A unique identifier for the patient. It contains no personal information, just used for reference.

**- Age:** represents the patient's age, and age is a very significant factor in examining their physical and mental health condition.

**-** **Gender (female / male):** Gender will influence the experience and response to treatment of mental illness.

**-** **Mood Score (1-10**): 5 Average mood score Reflects neither severely depressed nor euphoric state of mind.

**- Sleep Quality (1-10):** 8 Good sleep tends to be psychologically helpful, but alone may not be sufficient.

**- Physical Activity (hrs/week):** 5 The patient undertakes 5 hours of physical activity per week. This is a moderate amount and generally helps with mood regulation.

**-** **Stress Level (1-10):** 9 Level of stress is extremely low (9). Too much stress could be the reason for minimum progress despite other good indicators

**- Medication:** Medications to balance mood

**-Therapy Type: Interpersonal Therapy:**

The therapy is designed to improve interpersonal relationships and communication to reduce depressive symptoms

**- Treatment Start Date: 1/25/2024**

* Treatment started on January 25, 2024.
* Can be utilized to ascertain improvement or change over time.

**- Treatment Duration (weeks): 11**

* The treatment went for 11 weeks.
* A moderate duration that usually offers quantifiable improvement.

**- Outcome**: Deteriorated

* Last status is "Deteriorated.
* Reports the patient's condition worsened despite treatment.

**-** **Treatment Progress (1-10): 7**

* Progress as rated by the therapist is 7,which represents some positive response.
* Although it didn't appear in the outcome — perhaps due to external factors .

**- AI-Detected Emotional State:** Anxious

* AI labeled the prevailing state of the patient as anxious.
* Consistent with high stress and adverse outcome

**- Adherence to Treatment (%): 66**

* The patient complied with about 66% of the treatment, This is less than optimal and may be responsible for adverse outcomes

1. **Advice of Treatment:**

* **Psychotherapy**

- Cognitive Behavioral Therapy (CBT) is common

* **Medication**

**-** Antidepressants, prescribed by a psychiatrist

* **Self-care**

**-** Restful sleep, healthy diet, exercise

* **Social support**

**Important Note:**

Depression is treatable. A visit with a mental health care provider is the first step in the right direction.  
talking to family and friends, or support groups

1. **Examples of Depression Cases and How They Were Treated :**

**Scenario:**

* Lost interest in studies and hobbies.
* Sleeping excessively and constantly tired.
* Withdrawn from friends and family.
* Recurring thoughts of "life has no purpose.

**Recommended Treatment:**

- Cognitive Behavior Therapy (CBT) to treat negative

1. **Analysis of HTML Elements in the Project:** 
   1. **Head Section:**

|  |
| --- |
| <div class="bg-gray-50 p-4 rounded-xl shadow-md">  <label class="block text-gray-700 font-bold text-lg mb-1" for="username">👤 Username</label>  <input id="username" name="name" type="text" class="w-full border border-gray-300 rounded-lg p-2 focus:outline-blue-500" required />    </div> |

* label: Labels the input field (Username).
* input: An input field for the user to enter their username.
* required: Indicates that the field must be filled out before submitting the form**.**
  1. **Body Section:**

|  |
| --- |
| <body class="bg-gradient-to-br from-blue-50 to-purple-100 min-h-screen flex items-center justify-center"> |

* bg-gradient-to-br from-blue-50 to-purple-100: Sets a gradient background that transitions from light blue to light purple.
* min-h-screen: Ensures the body has a minimum height of the screen size.
* flex items-center justify-center: Uses Flexbox to center the content (the form) both vertically and horizontally.
  1. **Form Section(HTML):**

|  |
| --- |
| <form method="POST" action="/save\_personal\_info" class="w-full max-w-2xl bg-white p-8 rounded-2xl shadow-2xl space-y-6 text-left"> |

* method="POST": Specifies that the form data will be sent using the
* POST method to the server.
* action="/save\_personal\_info": The destination URL where the form data will be submitted to.
* class="w-full max-w-2xl bg-white p-8 rounded-2xl shadow-2xl space-y-6 text-left": Applies Tailwind CSS classes to style the form:
* w-full: The form will take up the full width.
* max-w-2xl: The maximum width of the form is set to 2xl.
* bg-white: The background is white.
* p-8: Padding of 8 units inside the form.
* rounded-2xl: Rounded corners for the form.
* shadow-2xl: Applies a strong shadow to the form.
* space-y-6: Vertical spacing between form elements.
* text-left: Aligns text to the left.

**8.4 Input Fields:**

|  |
| --- |
| <div class="bg-gray-50 p-4 rounded-xl shadow-md">  <label class="block text-gray-700 font-bold text-lg mb-1" for="username">👤 Username</label>  <input id="username" name="name" type="text" class="w-full border border-gray-300 rounded-lg p-2 focus:outline-blue-500" required />  </div> |

* label: Labels the input field (Username).
* input: An input field for the user to enter their username.
* required: Indicates that the field must be filled out before submitting the form.

**8.5 Submit Button:**

|  |
| --- |
| <button type="submit" class="w-full bg-indigo-600 text-white py-3 rounded-xl hover:bg-indigo-700 font-bold text-lg shadow-lg">  Next ➡  </button> |

* type="submit": This button submits the form data.
* class="w-full bg-indigo-600 text-white py-3 rounded-xl hover:bg-indigo-700": Styles the button using Tailwind CSS:
* w-full: The button takes up the full width.
* bg-indigo-600: Background color is indigo.
* text-white: The text is white.
* hover:bg-indigo-700: The background color changes on hover.
* font-bold text-lg: The text is bold and large.

**Dynamic Language Toggle Script – English ↔ Arabic**

**- let currentLang = "en";**

This line sets the default language to English ("en") when the page loads.

**- function toggleLanguage() {**

Defines a function named toggleLanguage that will be called when the user clicks a button to switch languages.

**- const elements = document.querySelectorAll('[data-en]');**

Selects all HTML elements that contain a data-en attribute. These elements are assumed to also contain a data-ar attribute, making them eligible for language toggling.

**- currentLang = currentLang === "en" ? "ar" : "en";**

This line toggles the value of currentLang between "en" and "ar" depending on the current language.

**- document.body.dir = currentLang === "ar" ? "rtl" : "ltr";**

Adjusts the direction of the page content: "rtl" (right-to-left) for Arabic and "ltr" (left- to-right) for English.

**- elements.forEach(el => {**

**el.textContent = el.dataset[currentLang];});**

Iterates over each selected element and updates its text content based on the currently selected language by accessing the appropriate data-\* attribute (data-en or data-ar).

**BODY SYYLING:**

|  |
| --- |
| <body class="bg-gradient-to-br from-blue-50 to-purple-100 min-h-screen flex items-center justify-center"> |

* **Applies a soft gradient background.**
* **Makes the page full-screen height and centers the form both vertically and horizontally using Flexbox.**

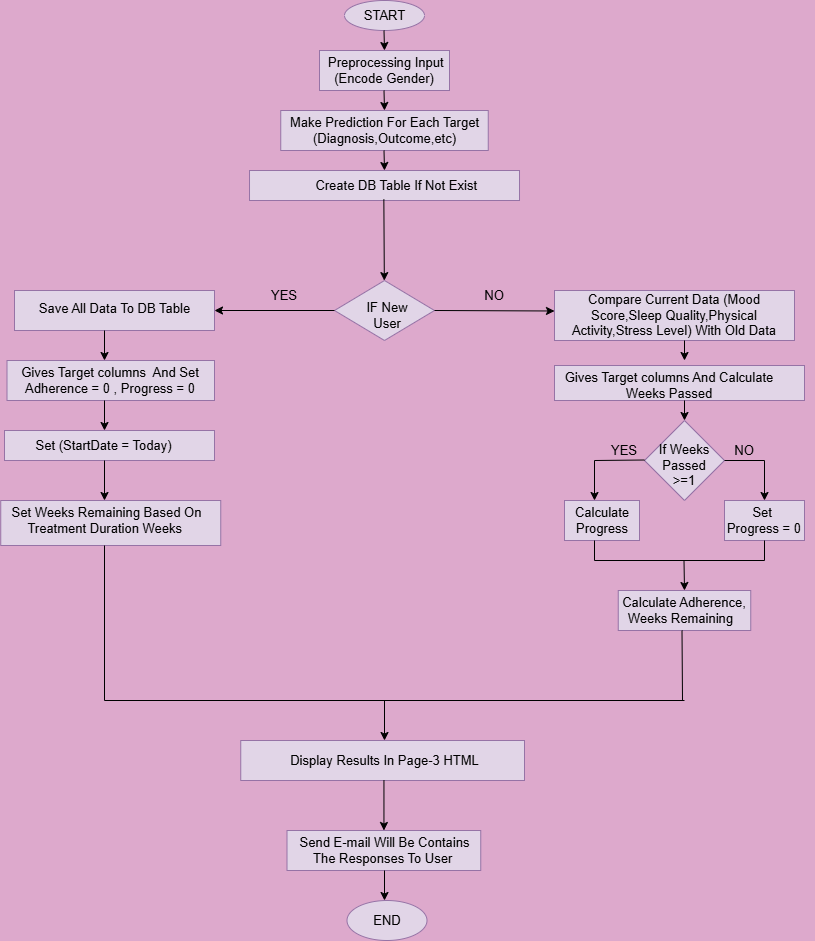
**JavaScript for Dynamic Dropdowns:**

|  |
| --- |
| const fields = ['mood\_score', 'sleep\_quality', 'stress\_level', 'physical\_activity']; fields.forEach(fieldId => { const select = document.getElementById(fieldId); for (let i = 1; i <= 10; i++) { const option = document.createElement('option'); option.value = i; option.textContent = i; select.appendChild(option); }  }); |

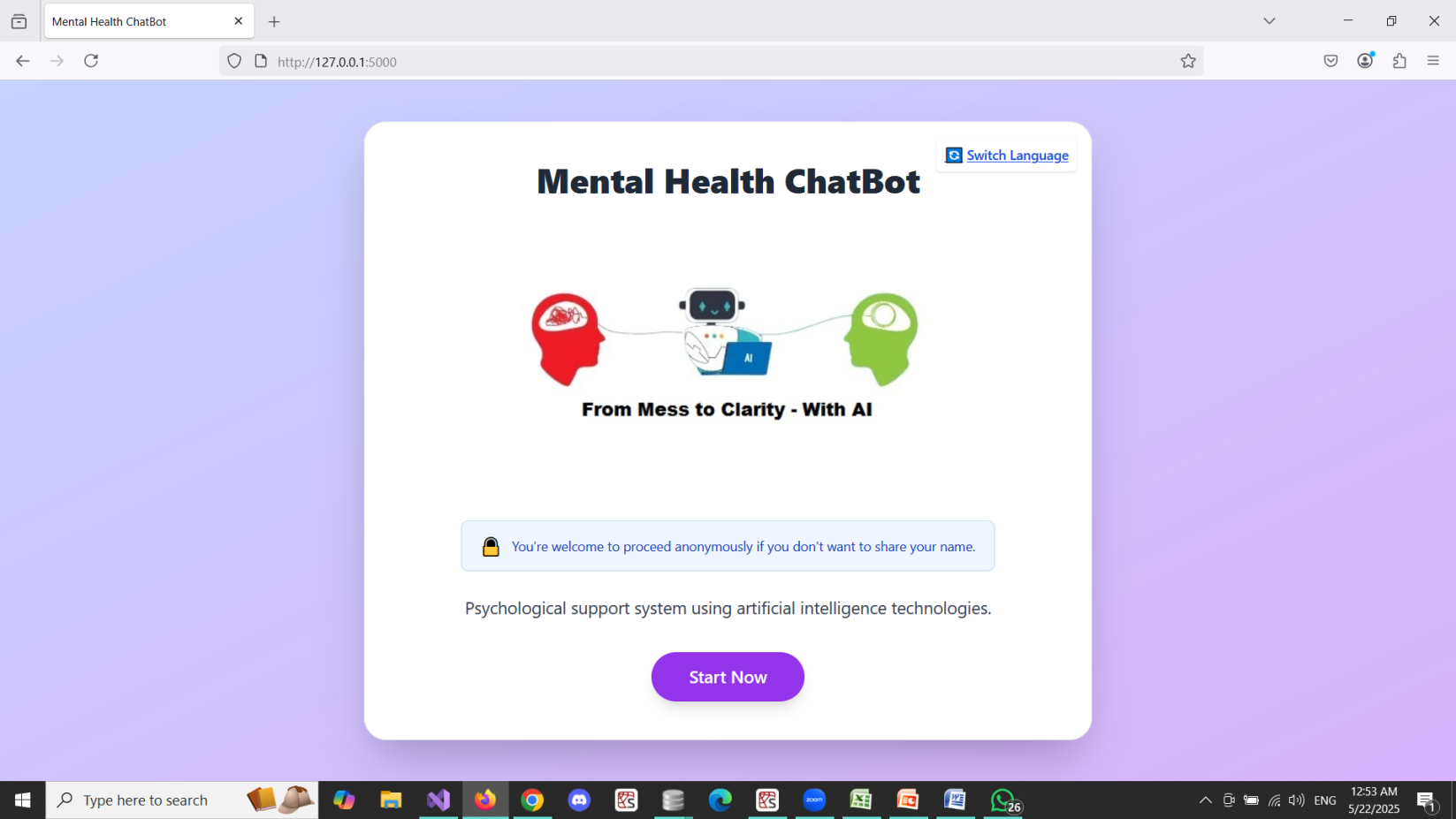
**Automatically generates the options 1-10 for all four dropdowns.**

* **Saves time and avoids repetitive HTML code.**

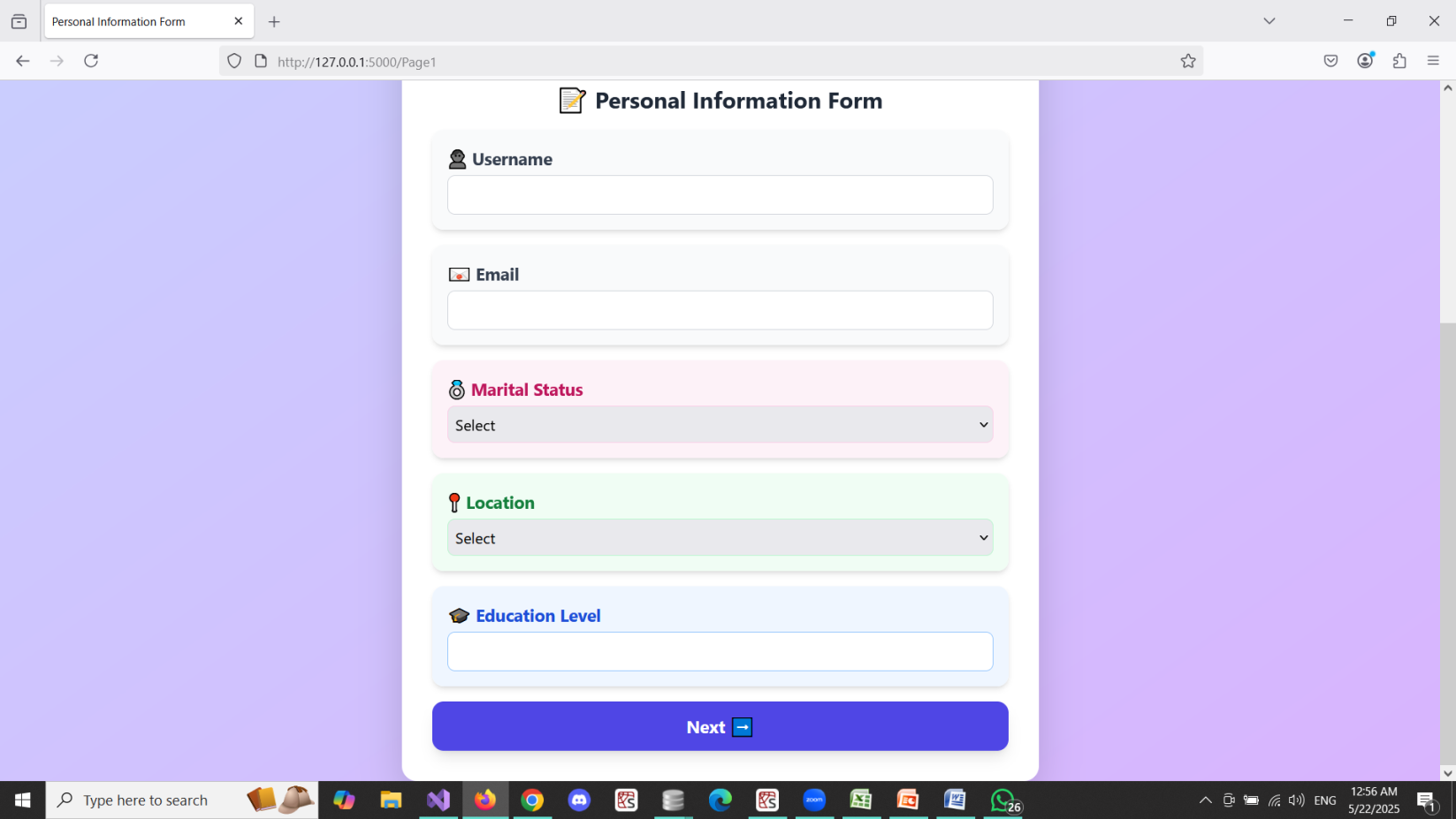
**9. FLOW CHART OF MODEL :**

**11. Visual Results of the Implemented Code:**

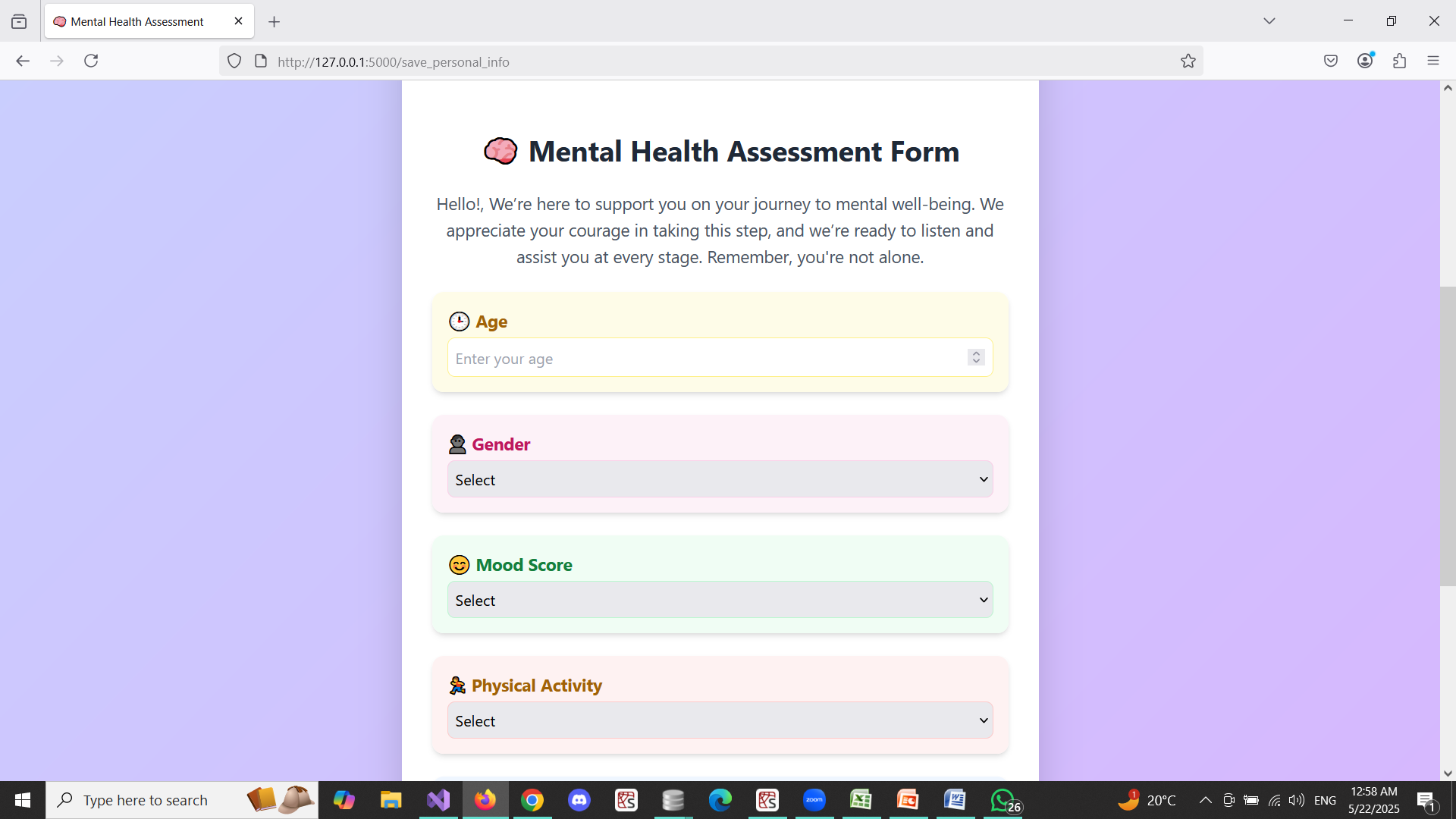
**PAGE 0**

****

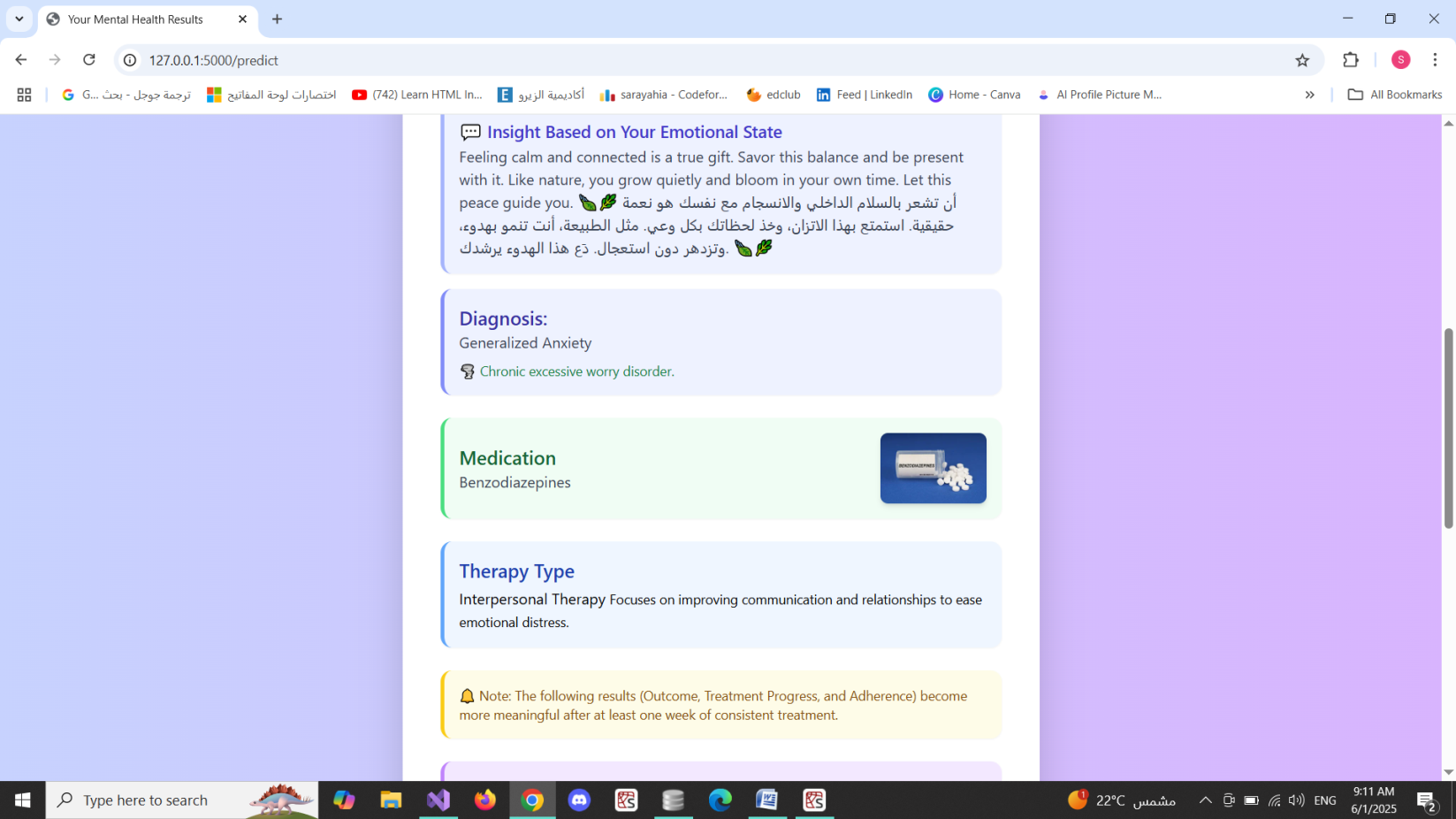
**PAGE 1**

****

**PAGE 2**

****

**PAGE 3**

****

**11. Lifestyle and Mental Health Factors:**

1. **Mood Score :**

A reflection of a person's general mood, indicating how good or bad they have been feeling over a span of time. It can be affected by sleep, stress, and exercise.

1. **Sleep Quality:**

Refers to how restful and restorative one's sleep is. It encompasses such factors as sleep duration, number of nighttime awakenings, and how rested one feels upon waking.

1. **Physical Activity:**

Measures the amount of physical activity or exercise that a person performs. Regular activity is proven to have beneficial effects on physical and mental health.

1. **Stress Level:**

An approximation of the level of psychological pressure or tension an individual is undergoing. Stress, when high, can detrimentally affect mood, sleep, and overall well-being.

**IMPORTANT NOTE ABOUT THEM:**

|  |
| --- |
| They all contribute to the model to determine patterns and make correct predictions regarding the user's mental state. |

**12. Backend Implementation Using Flask :**

## Project Structure :

## PROJECT

## ├── static ( Images that needed in HTML pages )

## ├── templates ( HTML templates: Page0.html, Page1.html, Page2.html, Page3.html)

## ├── data.xlsx (Training data)

## ├── mental\_health.db (SQLite database)

## ├── app.py ( Main application file)

## Requirements :

## Flask

## pandas

## scikit-learn

## openpyxl

## sqlite3 (built-in)

## smtplib (built-in)

## # Install them using:

**pip install flask pandas scikit-learn openpyxl**

## Model Training

**try:**

**df = pd.read\_excel("data.xlsx")**

**print("✅ تم تحميل البيانات بنجاح")**

**feature\_cols = ['Age', 'Gender', 'Mood Score (1-10)', 'Sleep Quality (1-10)',**

**'Physical Activity (hrs/week)', 'Stress Level (1-10)']**

**target\_cols = ['Diagnosis', 'AI-Detected Emotional State', 'Outcome', 'Medication', 'Therapy Type']**

**le\_gender = LabelEncoder()**

**df['Gender'] = le\_gender.fit\_transform(df['Gender'])**

**models = {}**

**accuracies = {}**

**X = df[feature\_cols]**

**for target in target\_cols:**

**y = df[target]**

**le = LabelEncoder()**

**y\_encoded = le.fit\_transform(y)**

**# تقسيم البيانات**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y\_encoded, test\_size=0.2, random\_state=42)**

**# تدريب النموذج**

**model = RandomForestClassifier(n\_estimators=100, random\_state=42)**

**model.fit(X\_train, y\_train)**

**# اختبار النموذج**

**y\_pred = model.predict(X\_test)**

**accuracy = accuracy\_score(y\_test, y\_pred)**

**# حفظ النموذج والمحول والدقة**

**models[target] = {**

**'model': model,**

**'encoder': le,**

**'features': feature\_cols**

**}**

**accuracies[target] = round(accuracy \* 100, 2)**

**# عرض النتائج**

**print("📊 دقة النماذج لكل Target:")**

**for target, acc in accuracies.items():**

**print(f"{target}: {acc:.2f}%")**

* Reads data.xlsx to load input/output features.
* Encodes categorical variables using LabelEncoder.
* Trains a separate RandomForestClassifier for each target variable (e.g., Diagnosis, Emotion).
* Stores trained models and accuracies.

## Database Setup

**Def create\_table\_if\_not\_exists():**

**try:**

**conn = sqlite3.connect('mental\_health.db', timeout=10)**

**cursor = conn.cursor()**

**cursor.execute('''**

**CREATE TABLE IF NOT EXISTS database (**

**id INTEGER PRIMARY KEY AUTOINCREMENT,**

**name TEXT, email TEXT, marital\_status TEXT, location TEXT, education TEXT,**

**age INTEGER, gender TEXT, mood\_score INTEGER, sleep\_quality INTEGER,**

**physical\_activity INTEGER, stress\_level INTEGER,**

**diagnosis TEXT, emotion TEXT, outcome TEXT, medication TEXT, therapy TEXT,**

**treatment\_start\_date TEXT, treatment\_duration\_weeks INTEGER,**

**adherence REAL, progress REAL**

**)**

**''')**

**conn.commit()**

**except Exception as e:**

**print("❌ خطأ أثناء إنشاء أو تحديث الجدول:")**

**traceback.print\_exc()**

**finally:**

**conn.close()**

The function create\_table\_if\_not\_exists():

* Creates the database table if it doesn't exist.
* Columns include personal data, predictions, treatment info, progress, and timestamps.

## Email Sender

Def send\_email(to\_email, subject, body):

try:

msg = EmailMessage()

msg['Subject'] = subject

msg['From'] = EMAIL\_ADDRESS

msg['To'] = to\_email

msg.set\_content(body)

with smtplib.SMTP\_SSL('smtp.gmail.com', 465) as smtp:

smtp.login(EMAIL\_ADDRESS, EMAIL\_PASSWORD)

smtp.send\_message(msg)

print(f"📧 تم إرسال الإيميل إلى {to\_email}")

except Exception:

print("❌ فشل في إرسال البريد الإلكتروني:")

traceback.print\_exc()

Function send\_email(to\_email, subject, body):

* Uses Gmail SMTP to send an email.
* Contains Diagnosis, Emotional State, Medication and Therapy Type.

## Prediction Route (/predict)

@app.route('/')

def home():

return render\_template('Page0.html')

@app.route('/Page1')

def page1():

return render\_template('Page1.html')

@app.route('/save\_personal\_info', methods=['POST'])

def save\_personal\_info():

return render\_template('Page2.html',

name=request.form['name'],

email=request.form['email'],

marital\_status=request.form['marital\_status'],

location=request.form['location'],

education=request.form['education'])

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

def predict():

try:

create\_table\_if\_not\_exists()

age = int(request.form['age'])

gender = request.form['gender']

mood\_score = int(request.form['mood\_score'])

sleep\_quality = int(request.form['sleep\_quality'])

physical\_activity = int(request.form['physical\_activity'])

stress\_level = int(request.form['stress\_level'])

name = request.form['name']

email = request.form['email']

marital\_status = request.form['marital\_status']

location = request.form['location']

education = request.form['education']

gender\_encoded = le\_gender.transform([gender])[0]

full\_input = {

'Age': age,

'Gender': gender\_encoded,

'Mood Score (1-10)': mood\_score,

'Sleep Quality (1-10)': sleep\_quality,

'Physical Activity (hrs/week)': physical\_activity,

'Stress Level (1-10)': stress\_level

}

results = {}

for target in target\_cols:

model = models[target]['model']

le = models[target]['encoder']

features = models[target]['features']

input\_df = pd.DataFrame([{k: full\_input[k] for k in features}])

prediction\_encoded = model.predict(input\_df)[0]

prediction = le.inverse\_transform([prediction\_encoded])[0]

results[target] = prediction

conn = sqlite3.connect('mental\_health.db', timeout=10)

cursor = conn.cursor()

cursor.execute("""

SELECT mood\_score, sleep\_quality, stress\_level, emotion, treatment\_start\_date, treatment\_duration\_weeks

FROM database WHERE email = ? ORDER BY id DESC LIMIT 1

""", (email,))

row = cursor.fetchone()

emotion\_state = results['AI-Detected Emotional State']

emotion\_messages = {

"Stressed": "ou're feeling Stressed —It's okay to feel stressed sometimes — it's part of being human. Take a deep breath and remind yourself that you're doing your best,and that’s enough. Things will calm down, and you are stronger than you think.🌿\nلا بأس أن تشعر بالتوتر أحيانًا، فهذا جزء من كونك إنسانًا. خذ نفسًا عميقًا، وذكّر نفسك أنك تبذل ما بوسعك، وهذا يكفي. الأمور ستهدأ، وأنت أقوى مما تعتقد.🌿",

"Happy": "It's wonderful to feel happy! Embrace that joy with gratitude and let your light inspire those around you. Remember, beautiful moments are meant to be fully lived and cherished. Smile — you're spreading positive energy without even trying 😊☀\nجميل أن تشعر بالسعادة! احتضن هذا الشعور بكل امتنان، ودَع نورك يُلهم من حولك. تذكّر أن اللحظات الجميلة تستحق أن نعيشها بوعي وفرح كامل. ابتسم... فأنت تنشر طاقة إيجابية دون أن تدري 😊☀",

"Anxious": "Anxiety can feel overwhelming, but it doesn’t define you or control you. Remember — thoughts aren’t facts, and this feeling will pass. Take it one moment at a time, and give yourself the peace you deserve. You are safe right now.🕊\nقلق شعور مزعج، لكنه لا يُعرّفك ولا يتحكم بك. تذكّر أن الأفكار ليست حقائق، وأن كل شيء يمر—even هذا القلق. خذ الأمور لحظة بلحظة، وامنح نفسك الطمأنينة التي تستحقها. أنت بأمان الآن. 🕊",

"Depressed": "I know that feeling depressed can make everything feel heavy... but you're not alone. Your presence matters — even on the days it doesn't feel like it. It's okay to rest, and to give yourself time to heal. One small step is enough today. 🌧💙\nأعلم أن الشعور بالاكتئاب يمكن أن يجعل كل شيء يبدو ثقيلاً... لكنك لست وحدك. وجودك مهم، حتى في الأيام التي لا تشعر فيها بذلك. لا بأس أن تطلب الراحة، وأن تمنح نفسك وقتًا للتعافي. خطوة صغيرة واحدة كافية اليوم. 🌧💙",

"Excited": "It's amazing that you're feeling excited! That spark means something truly matters to you. Enjoy the moment, and celebrate every step — no matter how small. You're on a path full of possibilities! ⚡🎉\nرائع أنك تشعر بالحماس! هذه الطاقة الجميلة هي علامة على شيء يهمك حقًا. استمتع بكل لحظة، واسمح لنفسك أن تحتفل بخطواتك مهما كانت صغيرة. أنت على طريق مليء بالإمكانيات! ⚡🎉",

"Neutral": "Feeling calm and connected is a true gift. Savor this balance and be present with it. Like nature, you grow quietly and bloom in your own time. Let this peace guide you. 🍃🌿\nأن تشعر بالسلام الداخلي والانسجام مع نفسك هو نعمة حقيقية. استمتع بهذا الاتزان، وخذ لحظاتك بكل وعي. مثل الطبيعة، أنت تنمو بهدوء، وتزدهر دون استعجال. دَع هذا الهدوء يرشدك. 🍃🌿"

}

emotion\_message = emotion\_messages.get(emotion\_state, "")

is\_new\_user = row is None

emotion\_message = emotion\_messages.get(emotion\_state, "")

is\_new\_user = row is None

if is\_new\_user:

treatment\_start\_date = datetime.today().strftime('%Y-%m-%d')

diagnosis = results['Diagnosis']

duration\_ranges = {

"Major Depressive Disorder": (10, 16),

"Generalized Anxiety": (10, 18),

"Bipolar Disorder": (12, 20),

"Panic Disorder": (8, 14)

}

default\_range = (8, 12)

selected\_range = duration\_ranges.get(diagnosis, default\_range)

treatment\_duration\_weeks = random.randint(\*selected\_range)

adherence = 0

progress = None

weeks\_remaining = treatment\_duration\_weeks

continue\_message = ""

else:

cursor.execute("""

SELECT diagnosis, medication, therapy

FROM database

WHERE email = ?

ORDER BY treatment\_start\_date ASC, id ASC

LIMIT 1

""", (email,))

stable\_values = cursor.fetchone()

if stable\_values:

results['Diagnosis'], results['Medication'], results['Therapy Type'] = stable\_values

old\_mood, old\_sleep, old\_stress, previous\_emotion, start\_date\_str, duration\_weeks = row

treatment\_start\_date = start\_date\_str

treatment\_duration\_weeks = duration\_weeks

improved = 0

if mood\_score > old\_mood: improved += 1

if sleep\_quality > old\_sleep: improved += 1

if stress\_level < old\_stress: improved += 1

start\_date = datetime.strptime(start\_date\_str, "%Y-%m-%d")

today = datetime.today()

weeks\_passed = max(0, (today - start\_date).days // 7)

if weeks\_passed >= 1:

raw\_progress = (mood\_score + sleep\_quality + physical\_activity - stress\_level) / 3

raw\_progress = max(0, min(raw\_progress, 10))

if treatment\_duration\_weeks > 0:

weeks\_ratio = min(weeks\_passed / treatment\_duration\_weeks, 1.0)

progress = round(raw\_progress \* weeks\_ratio)

else:

progress = 0

else:

progress = 0

improvement\_ratio = improved / 3

adherence = improvement\_ratio \* min(weeks\_passed / treatment\_duration\_weeks, 1.0) \* 100 if treatment\_duration\_weeks else 0

weeks\_remaining = max(0, treatment\_duration\_weeks - weeks\_passed)

continue\_message = "استمر بالعلاج 💪"

# ✅ تعديل outcome تلقائياً حسب الحالة العاطفية

positive = ['Happy', 'Excited', 'Neutral']

negative = ['Stressed', 'Anxious', 'Depressed']

if previous\_emotion in negative and emotion\_state in positive:

results['Outcome'] = 'Improved'

elif previous\_emotion in positive and emotion\_state in negative:

results['Outcome'] = 'Deteriorated'

else:

results['Outcome'] = 'No Change'

cursor.execute('''

INSERT INTO database (

name, email, marital\_status, location, education,

age, gender, mood\_score, sleep\_quality, physical\_activity, stress\_level,

diagnosis, emotion, outcome, medication, therapy,

treatment\_start\_date, treatment\_duration\_weeks, adherence, progress

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

(

name, email, marital\_status, location, education,

age, gender, mood\_score, sleep\_quality, physical\_activity, stress\_level,

results['Diagnosis'], emotion\_state, results['Outcome'], results['Medication'], results['Therapy Type'],

treatment\_start\_date, treatment\_duration\_weeks, adherence, progress

))

conn.commit()

if not is\_new\_user and weeks\_passed >= 1:

progress\_score = round((progress / 10) \* 10)

progress\_score = max(1, min(progress\_score, 10))

progress\_level = f"{progress\_score}/10 - {'📈 ممتاز' if progress\_score >= 8 else '🔄 متوسط' if progress\_score >= 5 else '⚠ منخفض'}"

adherence\_level = "✅ عالي" if adherence >= 80 else "🟡 متوسط" if adherence >= 50 else "❌ منخفض"

elif not is\_new\_user:

progress\_level = "🛛 لم يمضِ أسبوع على بدء العلاج"

adherence\_level = "ℹ لا يمكن حساب الالتزام"

else:

progress\_level = "لا يوجد سجل سابق لقياس التقدم 💼"

adherence\_level = "ℹ لم يتم احتساب الالتزام"

show\_duration = not (

results['Diagnosis'] == 'No Disorder' or

results['Medication'] == 'None' or

results['Therapy Type'] == 'None'

)

email\_body = f"""

Dear {name},

Thank you for completing your mental health assessment.

📋 Diagnosis: {results['Diagnosis']}

🌟 Emotional State: {results['AI-Detected Emotional State']}

💊 Medication: {results['Medication']}

🧠 Therapy Type: {results['Therapy Type']}

{"🧾 Outcome: " + results['Outcome'] if not is\_new\_user else ""}

Stay well,

Your Mental Health Team

"""

send\_email(email, "Your Mental Health Results", email\_body.strip())

return render\_template("Page3.html",

diagnosis=results['Diagnosis'],

emotion=emotion\_state,

outcome=results['Outcome'] if not is\_new\_user else None,

medication=results['Medication'],

therapy=results['Therapy Type'],

emotion\_message=emotion\_message,

progress=progress if not is\_new\_user else None,

adherence=round(adherence, 2) if not is\_new\_user else None,

treatment\_start\_date=treatment\_start\_date,

treatment\_duration\_weeks=treatment\_duration\_weeks,

is\_new\_user=is\_new\_user,

progress\_level=progress\_level if not is\_new\_user else None,

adherence\_level=adherence\_level if not is\_new\_user else None,

weeks\_remaining=weeks\_remaining if not is\_new\_user else None,

continue\_message=continue\_message if not is\_new\_user else "")

except Exception as e:

print("❌ خطأ أثناء التنبؤ أو الحفظ:")

traceback.print\_exc()

return "حدث خطأ أثناء المعالجة. تحقق من المدخلات."

finally:

try:

conn.close()

except:

pass

Handles user form submission:

* Processes input data, encodes gender.
* Runs prediction using trained models.
* If previous data exists, compares new vs. old results.
* Calculates progress and adherence.
* Stores new data in the database.
* Renders Page3.html with result and message.

## HTML Templates

* **Page0.html:** Landing page
* **Page1.html:** Personal info form
* **Page2.html:** Mental health input form
* **Page3.html:** Result display page with messages

## Progress and Adherence Logic:

if weeks\_passed >= 1:

raw\_progress = (mood\_score + sleep\_quality + physical\_activity - stress\_level) / 3

raw\_progress = max(0, min(raw\_progress, 10))

if treatment\_duration\_weeks > 0:

weeks\_ratio = min(weeks\_passed / treatment\_duration\_weeks, 1.0)

progress = round(raw\_progress \* weeks\_ratio)

else:

progress = 0

else:

progress = 0

improvement\_ratio = improved / 3

adherence = improvement\_ratio \* min(weeks\_passed / treatment\_duration\_weeks, 1.0) \* 100 if treatment\_duration\_weeks else 0

weeks\_remaining = max(0, treatment\_duration\_weeks - weeks\_passed)

Calculates improvement based on changes in stress level, sleep quality,mood score and physical activity.

## Outcome Logic

To compare emotional change:

## if previous\_emotion in negative and emotion\_state in positive:

## results['Outcome'] = 'Improved'

## elif previous\_emotion in positive and emotion\_state in negative:

## results['Outcome'] = 'Deteriorated'

## else:

## results['Outcome'] = 'No Change'

## Getting Started

## 1. Install dependencies

## pip install flask pandas scikit-learn openpyxl

## 2. Run the app

## python app.py

## 3. Open in browser

## http://localhost:5000/

## 

This is an important point that shows that the email must contain the email symbol @.

## <input id="email" name="email" type="email":

## This command controls the type of data entered, which must be an email, provided that there is @

**13. References :**

* **Dataset:**

[**https://www.kaggle.com/datasets/uom190346a/mental-health-diagnosis-and-treatment-monitoring/data**](https://www.kaggle.com/datasets/uom190346a/mental-health-diagnosis-and-treatment-monitoring/data)

**Note About Dataset : Dataset the preprocessed and cleaned by our team because the size it is small (500 rows)**

* **Codes : colab , Github and by our team**

**Note about Codes : We took parts of the code from more than one site, so the reference contains more than one site.**