STUDENT MENTAL HEALTH

*PREDICTION SYSTEM*

A logo with text and a flower

Description automatically generated

DEPARTMENT OF DATA SCIENCE

ADITYA DEGREE COLLEGE

(Permanently Affiliated to Adikavi Nannaya University, Best Placement Award

By Govt. of AP & Accredited by NAAC with “A++” Grade, Recognised by UGC

Under section 2(f), 12(B), approved by APSCHE),

Lakshmi Narayana Nagar, Kakinada-533003

(2022-2025)

A project report on

STUDENT MENTAL HEALTH PREDICTION SYSTEM

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Science and Computer

**DATA SCIENCE**

B.V.N.SURYA MANOJ - 2344210675

D.D.NAVADEEP - 2344210125

P.MANI KANTA SWAMY - 2344210751

**ADITYA DEGREE COLLEGE :: KAKINADA**

A logo of a national assessment and excellence council

Description automatically generatedA logo with text and a cross

Description automatically generated **Accredited by NAAC with ‘A++’ grade**

**(Affiliated to Adikavi Nannaya University)**

**Approved By APSCHE**

**Lakshminarayana Nagar, KAKINADA, Andhra Pradesh**

Adityan’s Science Talent Show

**SCI-LAND :: 2K25**

An Innovative Learning Platform

**NAME OF THE DEPARTMENT : DATA SCIENCE**

**NAME OF THE CLUB : DATA WRANGLERS**

**NAME OF THE PROJECT : Student mental health**

**prediction system.**

**NAME OF THE PROJECT MEMBERS :** B.V.N.SURYA MANOJ

D.D.NAVADEEP

P.MANI KANTA SWAMY

A logo of a national assessment and excellence council

Description automatically generatedA logo with text and a cross

Description automatically generated**ADITYA DEGREE COLLEGE :: KAKINADA Accredited by NAAC with ‘A++’ grade**

**(Affiliated to Adikavi Nannaya University)**

**Approved By APSCHE**

**Lakshminarayana Nagar, KAKINADA, Andhra Pradesh**

Adityan’s Science Talent Show

**SCI-LAND :: 2K25**

An Innovative Learning Platform

**Certificate of Participation**

This is to certify that Mr. BULUSU VENKATA NAGA SURYA MANOJ of B.Sc. Data science class has successfully presented STUDENT MENTAL HEALTH PREDICTION SYSTEM project in SCILAND-2K25 held on 21-03-2025 and 22 – 03 - 2025.

Coordinator Head of the Department Principal

A logo of a national assessment and excellence council

Description automatically generatedA logo with text and a cross

Description automatically generated**ADITYA DEGREE COLLEGE :: KAKINADA**

**Accredited by NAAC with ‘A++’ grade**

**(Affiliated to Adikavi Nannaya University)**

**Approved By APSCHE**

**Lakshminarayana Nagar, KAKINADA, Andhra Pradesh**

Adityan’s Science Talent Show

**SCI-LAND :: 2K25**

An Innovative Learning Platform

**Certificate of Participation**

This is to certify that Mr. DRAKSHARAPU DURGA NAVADEEP of B.Sc. Data Science has successfully presented STUDENT MENTAL HEALTH PREDICTION SYSTEM project in SCILAND-2K25 held on 21-03-2025 and 22 – 03 - 2025.

Coordinator Head of the Department Principal

A logo of a national assessment and excellence council

Description automatically generatedA logo with text and a cross

Description automatically generated**ADITYA DEGREE COLLEGE :: KAKINADA**

**Accredited by NAAC with ‘A++’ grade**

**(Affiliated to Adikavi Nannaya University)**

**Approved By APSCHE**

**Lakshminarayana Nagar, KAKINADA, Andhra Pradesh**

Adityan’s Science Talent Show

**SCI-LAND :: 2K25**

An Innovative Learning Platform

**Certificate of Participation**

This is to certify that Mr. POLIMERU MANI KANTA SWAMY of B.Sc. Data Science has successfully presented STUDENT MENTAL HEALTH PREDICTION SYSTEM project in SCILAND-2K25 held on 21-03-2025 and 22 – 03 - 2025.

Coordinator Head of the Department Principal

STUDENT MENTAL HEALTH PREDICTION SYSTEM

**Abstract**

The mental health of students is a growing concern globally, with various factors such as academic pressure, social interactions, and personal habits contributing to psychological stress. This project, titled Student Mental Health Prediction System Using Python, focuses on leveraging machine learning and data analysis to predict mental health outcomes for students. By analyzing data collected from surveys, academic performance records, and behavioral patterns, the system aims to provide early detection of stress, anxiety, and other mental health conditions among students. The project uses Python-based libraries such as Pandas, Scikit-learn, and Matplotlib to perform data preprocessing, exploratory data analysis (EDA), and model training. Several predictive models, including logistic regression, random forests, and support vector machines (SVM), are used to predict the likelihood of students experiencing psychological distress. This system not only aims to provide a prediction but also offers recommendations for mental health support based on the identified risk factors. By identifying students at risk early on, the project seeks to empower educational institutions to implement timely and effective mental health interventions, ultimately fostering a more supportive and healthier academic environment.

Apparatus:

* **Hardware**:
  + Personal Computer or Laptop
  + Internet connection (for data gathering)
* **Software**:
  + Python, jupyter notebook, visual studio code - insiders.
  + Libraries: Pandas, NumPy, Scikit-learn, flask, os, pickle, datetime.

Language used:  
Python

Html ,css, javascript.

**Practical Approach**:

The project starts with collecting data from surveys and public datasets that include factors like academic performance, sleep habits, social activities, and self-reported mental health issues. This data is cleaned by handling missing values, encoding categorical variables, and normalizing numerical features to ensure accurate predictions.

Exploratory Data Analysis (EDA) is performed to visualize trends and correlations between features, such as the impact of sleep deprivation and academic stress on mental health. These insights help in selecting the most relevant features for the predictive models.

Various machine learning algorithms, including Logistic Regression, Random Forest, and Support Vector Machines (SVM), are implemented. The model performance is evaluated based on accuracy, precision, recall, and F1-score, with Random Forest emerging as the best-performing model.

After selecting the final model, hyperparameter tuning is done using Grid Search to optimize performance. The system can now predict mental health outcomes in real-time and can be deployed within educational institutions to identify at-risk students and provide early intervention.

**Summary**:  
 This project aimed to develop a predictive system for assessing student mental health using Python-based machine learning techniques. By collecting a wide range of data related to students' academic and personal lives, the system predicts the likelihood of mental health issues such as stress, anxiety, and depression. The system was built using a combination of Logistic Regression, Random Forests, and Support Vector Machines, with Random Forests emerging as the most effective model. Through exploratory data analysis, significant relationships between academic performance, sleep patterns, and stress levels were uncovered, informing the prediction process. This project demonstrates the importance of early identification and intervention in mental health, allowing educational institutions to provide timely support to students. The final model can be deployed as part of a proactive approach to mental health care, potentially improving student well-being and academic success. Future work could involve incorporating additional data sources and refining the system for continuous monitoring and real-time intervention.

Code :

Data PreProcessing and model building :

# %%

import pandas as pd

data = pd.read\_csv(r"C:\Users\SURYA MANOJ\OneDrive\Desktop\dataset\Student Depression Dataset.csv")

data

# ## DATA PREPROCESSING

# %%

data.isnull()

data['Financial Stress'] = data['Financial Stress'].fillna(data['Financial Stress'].mean())

# %%

data.info()

# %%

data = data.drop(['id','Profession','Work Pressure','Job Satisfaction','City'],axis =1)

data

# %%

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

data['Gender'] = le.fit\_transform(data['Gender'])

data['Sleep Duration'] = le.fit\_transform(data['Sleep Duration'])

data['Dietary Habits'] = le.fit\_transform(data['Dietary Habits'])

data['Degree']= le.fit\_transform(data['Degree'])

data['Have you ever had suicidal thoughts ?'] = le.fit\_transform(data['Have you ever had suicidal thoughts ?'])

data['Family History of Mental Illness'] = le.fit\_transform(data['Family History of Mental Illness'])

data

# %%

x = data.drop(['Depression'],axis = 1)

x

# %%

# %%

y = data['Depression']

y

# %%

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size = 0.2)

# %%

from imblearn.over\_sampling import SMOTE

smote = SMOTE()

x\_train, y\_train = smote.fit\_resample(x\_train, y\_train)

# %%

print(x\_train)

# %%

print(x\_test)

# %%

print(y\_train)

# %%

print(y\_test)

# %%

# %%

import xgboost as xgb

xgb\_model = xgb.XGBClassifier(n\_estimators=100, learning\_rate=0.1, max\_depth=3, random\_state=42)

xgb\_model.fit(x\_train, y\_train)

# %%

xgb\_predictions = xgb\_model.predict(x\_test)

print(xgb\_predictions)

# %%

from sklearn.metrics import accuracy\_score

xgbas = accuracy\_score(y\_test,xgb\_predictions)

xgbas

# %%

import pickle

with open('xgb\_model.pkl', 'wb') as file:

    pickle.dump(xgb\_model, file)

# %%

with open('xgb\_model.pkl', 'rb') as file:

    loaded\_model = pickle.load(file)

**Flask application :**

from flask import Flask, render\_template, request

import numpy as np

import pickle

import os

import pandas as pd

from datetime import datetime

app = Flask(\_\_name\_\_, static\_url\_path='/static', static\_folder='static', template\_folder='templates')

app.name = "Student Mental Health Prediction"

model\_file = "xgb\_model.pkl"

if not os.path.exists(model\_file):

raise FileNotFoundError(f"Model file '{model\_file}' not found!")

with open(model\_file, "rb") as file:

clf\_model = pickle.load(file)

@app.route('/')

def home():

return render\_template('index.html')

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

def predict():

try:

time\_stamp = datetime.now()

name = request.form.get('name', 'Student')

Gender = int(request.form.get("Gender", -1))

Age = int(request.form.get("Age", 18))

Academic\_Pressure = float(request.form.get("Academic\_Pressure", 5))

CGPA = float(request.form.get("CGPA", 7))

Study\_Satisfaction = float(request.form.get("Study\_Satisfaction", 5))

Sleep\_Duration = int(request.form.get("Sleep\_Duration", 1))

Dietary\_Habits = int(request.form.get("Dietary\_Habits", 1))

Degree = int(request.form.get("Degree", -1))

Suicidal\_Thoughts = int(request.form.get("Suicidal\_Thoughts", 0))

Work\_Study\_Hours = float(request.form.get("Work\_Study\_Hours", 4))

Financial\_Stress = float(request.form.get("Financial\_Stress", 5))

Family\_History\_Mental\_Illness = int(request.form.get("Family\_History\_Mental\_Illness", 0))

Save\_data = int(request.form.get("Save\_Data",0))

input\_data = np.array([[

Gender, Age, Academic\_Pressure, CGPA, Study\_Satisfaction,

Sleep\_Duration, Dietary\_Habits, Degree, Suicidal\_Thoughts,

Work\_Study\_Hours, Financial\_Stress, Family\_History\_Mental\_Illness

]])

# Make predictions

prediction = clf\_model.predict(input\_data)

prediction\_prob = clf\_model.predict\_proba(input\_data) \* 100

# Interpretation of prediction

result = "The Student's Mental Health is not good" if prediction[0] == 1 else "The Student's Mental Health is good"

prob\_good = f"{prediction\_prob[0][0]:.2f}"

prob\_not\_good = f"{prediction\_prob[0][1]:.2f}"

# Generate suggestions based on input values

suggestions = []

# Sleep Duration

if Sleep\_Duration == 0:

suggestions.append("Try to get at least 7-8 hours of sleep for better mental and physical health.")

elif Sleep\_Duration == 2:

suggestions.append("Less than 5 hours of sleep is harmful. Aim for at least 7 hours.")

elif Sleep\_Duration == 3:

suggestions.append("More than 8 hours of sleep can cause lethargy. Keep it around 7-8 hours.")

else:

suggestions.append("Your sleep duration is good. Keep it up!")

# Academic Pressure

if Academic\_Pressure > 7:

suggestions.append("Consider time management techniques or seeking academic counseling to reduce stress.")

# Dietary Habits

if Dietary\_Habits == 2:

suggestions.append("Improve your diet by including more fruits, vegetables, and whole grains.")

elif Dietary\_Habits == 1:

suggestions.append("Try to consume healthy food regularly and limit junk food.")

# Study Satisfaction

if Study\_Satisfaction < 5:

suggestions.append("Experiment with different study methods to find what works best for you.")

# Financial Stress

if Financial\_Stress > 7:

suggestions.append("Consider financial planning or seeking advice to manage financial stress.")

# Suicidal Thoughts

if Suicidal\_Thoughts > 7:

suggestions.append("Reach out to a mental health professional for support.")

elif 1 <= Suicidal\_Thoughts <= 7:

suggestions.append("Try engaging in positive activities and spend time with loved ones.")

# Family History of Mental Illness

if Family\_History\_Mental\_Illness == 1:

suggestions.append("Stay proactive about your mental health with regular self-check-ins.")

# General Well-being

suggestions.append("Stay physically active to improve mood and well-being.")

suggestions.append("Practice mindfulness or relaxation techniques to reduce stress.")

# Convert categorical values to readable text

gender\_text = "Male" if Gender == 1 else "Female"

sleep\_text = ["5-6 hours", "7-8 hours", "Less than 5 hours", "More than 8 hours"][Sleep\_Duration]

diet\_text = ["Healthy", "Moderate", "Unhealthy"][Dietary\_Habits]

degree\_dict = {

0: "B.Com", 1: "BA", 2: "BCA/BBA", 3: "BSc", 4: "Class 12",

5: "MTech", 6: "MCA/MBA", 7: "BTech", 8: "MSc", 9: "MA", 10: "M.Com"

}

degree\_text = degree\_dict.get(Degree, "Unknown")

suicidal\_text = "Yes" if Suicidal\_Thoughts == 1 else "No"

family\_history\_text = "Yes" if Family\_History\_Mental\_Illness == 1 else "No"

if Save\_data == 1:

df = pd.DataFrame({

'Timestamp': [time\_stamp], # Fixed column name

'Name': [name],

'Age': [Age],

'Gender': [gender\_text],

'Academic Pressure': [Academic\_Pressure],

'CGPA': [CGPA],

'Study Satisfaction': [Study\_Satisfaction], # Fixed column name

'Sleep Duration': [sleep\_text],

'Dietary Habits': [diet\_text],

'Degree': [degree\_text],

'Suicidal Thoughts': [suicidal\_text], # Fixed column name

'Work Study Hours': [Work\_Study\_Hours],

'Financial Stress': [Financial\_Stress],

'Family History': [family\_history\_text], # Fixed extra space issue

'Good Mental Health Probability': [prob\_good], # Fixed name

'Not Good Mental Health Probability': [prob\_not\_good]

})

# Save the data

df.to\_csv("Result.csv", mode='a', header=not os.path.exists("Result.csv"), index=False)

# Save confirmation message

save\_message = "Thanks for allowing us to save your data."

else:

save\_message = "You did not opt to save data."

# Render results page

return render\_template(

'result.html',

result=result,

suggestions=suggestions,

prob\_good=prob\_good,

prob\_not\_good=prob\_not\_good,

name=name,

Age=Age,

Gender=gender\_text,

Academic\_Pressure=Academic\_Pressure,

CGPA=CGPA,

Study\_Satisfaction=Study\_Satisfaction,

Sleep\_Duration=sleep\_text,

Dietary\_Habits=diet\_text,

Degree=degree\_text,

Suicidal\_Thoughts=suicidal\_text,

Work\_Study\_Hours=Work\_Study\_Hours,

Financial\_Stress=Financial\_Stress,

Family\_History\_Mental\_Illness=family\_history\_text,

save\_message = save\_message

)

except Exception as e:

return f"An error occurred: {e}"

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

app.run(debug = True)

**Index WebPage :**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Student Mental Health Prediction</title>

<style>

/\* Global Styles \*/

/\* Global Styles \*/

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

background: url('https://ibcces.org/wp-content/uploads/2020/08/Student-Mental-health-silent-pandemic-student-body-in-schools.jpg') no-repeat center center fixed;

background-size: cover;

height: 100vh; /\* Full viewport height \*/

width: 100vw; /\* Full viewport width \*/

margin: 0;

padding: 0;

color: #060606;

display: flex;

justify-content: center;

background-attachment: fixed;

align-items: first baseline;

}

.container {

width: 90%; /\* Responsive width \*/

max-width: 600px; /\* Limit max size \*/

background: rgba(244, 242, 242, 0.553); /\* Transparent white \*/

padding: 30px 40px;

border-radius: 12px;

box-shadow: 0 8px 16px rgba(0, 0, 0, 0.122);

animation: fadeIn 1s ease-in-out;

backdrop-filter: blur(10px); /\* Glassmorphism effect \*/

}

/\* Heading \*/

h1 {

text-align: center;

color: #2115c9;

font-size: 24px;

margin-bottom: 30px;

position: relative;

forced-color-adjust: "Purple";

background-color: rgb(147, 191, 231);

box-shadow: #333;

border-radius :3%;

background-position: 30%;

border: 50%;

box-decoration-break: slice;

background-position: 0%;

}

h1::after {

content: "";

display: block;

width: 60px;

height: 4px;

background: #be8c56;

margin: 10px auto 0;

border-radius: 2px;

}

/\* Form Labels and Inputs \*/

label {

display: block;

margin-bottom: 8px;

font-weight: bold;

color: #444;

}

input, select {

width: 100%;

padding: 12px;

margin-bottom: 15px;

border: 1px solid #d9b6b6;

border-radius: 6px;

font-size: 14px;

transition: all 0.3s ease;

}

input:focus, select:focus {

border-color: #007bff;

outline: none;

box-shadow: 0 0 5px rgba(0, 123, 255, 0.5);

}

/\* Tooltip \*/

.tooltip {

font-size: 12px;

color: #b7b7b7;

margin-top: -10px;

margin-bottom: 10px;

}

/\* Button \*/

button {

width: 100%;

padding: 12px;

background-color: #1c32d3;

color: white;

border: none;

border-radius: 6px;

font-size: 16px;

font-weight: bold;

text-transform: uppercase;

cursor: pointer;

transition: all 0.3s ease;

}

button:hover {

background-color: #49fb08d2;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

}

/\* Form Group \*/

.form-group {

margin-bottom: 20px;

}

/\* Responsive Design \*/

@media (max-width: 768px) {

.container {

padding: 20px 25px;

}

h1 {

font-size: 20px;

}

}

.checkbox-group {

display: flex;

align-items :center;

margin-top: 15px;

width: 30%; /\* Ensure it takes full width \*/

font-weight: 100;

}

.checkbox-group label {

display :inline-flexbox;

align-items : center;

font-size: 20px;/\*crease text size for visibility \*/

color: #000000; /\* Darker text for readability \*/

white-space:no-repeat

/\* Ensures text spans the full width \*/

}

.checkbox-group input[type="checkbox"] {

margin-right: 10px; /\* Space between checkbox and text \*/

transform: scale(1.2); /\* Slightly increase checkbox size \*/

cursor:auto;

}

/\* Animations \*/

@keyframes fadeIn {

from {

opacity: 0;

transform: translateY(-20px);

}

to {

opacity: 2

transform: translateY(0);

}

}

</style>

</head>

<body>

<div class="container">

<h1>Student Mental Health Prediction</h1>

<form action="/predict" method="post">

<label for="name">Enter Your Name:</label>

<input type="text" name="name" id="name" required>

<div class="form-group">

<label for="Gender">Gender:</label>

<select name="Gender" id="Gender" required>

<option value="" disabled selected>Select your gender</option>

<option value="0">Female</option>

<option value="1">Male</option>

</select>

</div>

<div class="form-group">

<label for="Age">Age:</label>

<input type="number" name="Age" id="Age" min="10" max="100" required>

</div>

<div class="form-group">

<label for="Academic\_Pressure">Academic Pressure (1-10):</label>

<select name="Academic\_Pressure" id="Academic\_Pressure" required>

<option value="" disabled selected>Academic\_Pressure</option>

<option value="0">0</option>

<option value="1">1</option>

<option value = "2">2</option>

<option value = "3">3</option>

<option value ="4">4</option>

<option value ="5">5</option>

<option value ="6">6</option>

<option value ="7">7</option>

<option value ="8">8</option>

<option value ="9">9</option>

<option value ="10">10</option>

</select>

</div>

<div class="form-group">

<label for="CGPA">CGPA:</label>

<input type="number" name="CGPA" id="CGPA" step="0.01" min="0" max="10" required>

</div>

<div class="form-group">

<label for="Study\_Satisfaction">Study Satisfaction (1-10):</label>

<select name="Study\_Satisfaction" id="Study\_Satisfaction" required>

<option value="" disabled selected>Study\_Satisfaction</option>

<option value="0">0</option>

<option value="1">1</option>

<option value = "2">2</option>

<option value = "3">3</option>

<option value ="4">4</option>

<option value ="5">5</option>

<option value ="6">6</option>

<option value ="7">7</option>

<option value ="8">8</option>

<option value ="9">9</option>

<option value ="10">10</option>

</select>

</div>

<div class="form-group">

<label for="Sleep\_Duration">Sleep Duration:</label>

<select name="Sleep\_Duration" id="Sleep\_Duration" required>

<option value="" disabled selected>Select sleep duration</option>

<option value="0">5-6 hours</option>

<option value="1">7-8 hours</option>

<option value="2">Less than 5 hours</option>

<option Value="3">More than 8 hours</option>

</select>

</div>

<div class="form-group">

<label for="Dietary\_Habits">Dietary Habits:</label>

<select name="Dietary\_Habits" id="Dietary\_Habits" required>

<option value="" disabled selected>Select dietary habits</option>

<option value="0">Healthy</option>

<option value="1">Moderate</option>

<option value="2">Unhealthy</option>

</select>

</div>

<div class="form-group">

<label for="Degree">Degree:</label>

<select name="Degree" id="Degree" required>

<option value="" disabled selected>Select your degree</option>

<option value="0">B.com</option>

<option value="1">BA</option>

<option value="2">BCA/BBA</option>

<option value="3">BSc</option>

<option value="4">Class 12</option>

<option value="5">M.Tech</option>

<option value="6">MCA/MBA</option>

<option value = "7">BTech</option>

<option value="8">MSc</option>

<option value ="9">MA</option>

<option value ="10">M.com</option>

</select>

</div>

<div class="form-group">

<label for="Suicidal\_Thoughts">Suicidal Thoughts:</label>

<select name="Suicidal\_Thoughts" id="Suicidal\_Thoughts" required>

<option value="" disabled selected>Have you had suicidal thoughts?</option>

<option value="0">No</option>

<option value="1">Yes</option>

</select>

</div>

<div class="form-group">

<label for="Work\_Study\_Hours">How Much Time you Spend on Study/Work ?</label>

<select name="Work\_Study\_Hours" id="Work\_Study\_Hours" required>

<option value="" disabled selected>Work\_Study\_Hours</option>

<option value="0" style="color: rgb(111, 2, 2);">0</option>

<option value="1">1</option>

<option value = "2">2</option>

<option value = "3">3</option>

<option value ="4">4</option>

<option value ="5">5</option>

<option value ="6">6</option>

<option value ="7">7</option>

<option value ="8">8</option>

<option value ="9">9</option>

<option value ="10">10</option>

<option value = "11">11</option>

<option value ="12">12</option>

</select>

</div>

<div class="form-group">

<label for="Financial\_Stress">On a Scale of 1-10, Select How much Financial Stress You are Facing</label>

<select name="Financial\_Stress" id="Financial\_Stress" required>

<option value="" disabled selected>Financial\_Stress</option>

<option value="0">0</option>

<option value="1">1</option>

<option value = "2">2</option>

<option value = "3">3</option>

<option value ="4">4</option>

<option value ="5">5</option>

<option value ="6">6</option>

<option value ="7">7</option>

<option value ="8">8</option>

<option value ="9">9</option>

<option value ="10">10</option>

</select>

</div>

<div class="form-group">

<label for="Family\_History\_Mental\_Illness">Family History of Mental Illness:</label>

<select name="Family\_History\_Mental\_Illness" id="Family\_History\_Mental\_Illness" required>

<option value="" disabled selected>Select an option</option>

<option value="0">No</option>

<option value="1">Yes</option>

</select>

</div>

<div class="form-group checkbox-group">

<label for="saveData">

<input type="checkbox" id="saveData" name="Save\_Data" value="1" onchange="checkSaveData()">

<span>Save my data for future reference</span>

</label>

</div>

<script type="text/javascript">

function checkSaveData() {

const saveData = document.getElementById('saveData').checked;

if (saveData) {

alert("We use the data only for training purposes.");

}

}

</script>

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

</form>

</div>

</body>

</html>

**Result webpage :**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Prediction Result</title>

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<style>

/\* General Styles \*/

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

background: url(https://slidescorner.com/wp-content/uploads/2022/12/01-Mental-Health-Free-PPT-Backgrounds-by-SlidesCorner.com\_-1536x864.jpg) no-repeat center center fixed;

background-size: cover;

color: #fff;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

text-align: center;

}

.container {

width: 95%;

max-width: 1200px;

background: rgba(15, 15, 15, 0.9);

padding: 40px;

border-radius: 15px;

box-shadow: 0 8px 20px rgba(0, 0, 0, 0.5);

animation: fadeIn 1s ease-in-out;

}

h1 {

color: #ffd700;

font-size: 2rem;

text-transform: uppercase;

letter-spacing: 2px;

text-shadow: 3px 3px 8px rgba(0, 0, 0, 0.7);

margin-bottom: 20px;

}

.result-box {

background: rgba(255, 255, 255, 0.1);

padding: 15px;

border-radius: 8px;

font-size: 1.4rem;

font-weight: bold;

margin-bottom: 15px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.4);

border-left: 5px solid #ffd700;

}

.flex-container {

display: flex;

justify-content: space-between;

align-items: flex-start;

gap: 20px;

flex-wrap: wrap;

margin-top: 20px;

}

.box {

flex: 1;

background: rgba(255, 255, 255, 0.1);

padding: 20px;

border-radius: 12px;

text-align: left;

box-shadow: 0 6px 15px rgba(0, 0, 0, 0.5);

transition: transform 0.3s ease-in-out;

min-width: 45%;

}

.box:hover {

transform: translateY(-5px);

}

table {

width: 100%;

border-collapse: collapse;

color: #fff;

}

th, td {

padding: 12px;

text-align: left;

border-bottom: 1px solid rgba(255, 255, 255, 0.3);

}

th {

background: rgba(255, 255, 255, 0.2);

}

.suggestions ul {

list-style: none;

padding: 0;

}

.suggestions li {

font-size: 1rem;

background: rgba(255, 223, 88, 0.3);

padding: 12px;

margin: 10px 0;

border-radius: 10px;

transition: transform 0.3s ease, background 0.3s ease;

}

.suggestions li:hover {

transform: translateX(12px);

background: rgba(255, 223, 88, 0.7);

}

.back-button {

display: inline-block;

margin-top: 25px;

padding: 12px 30px;

color: #000;

background-color: #ffd700;

text-decoration: none;

border-radius: 30px;

font-weight: bold;

font-size: 1rem;

transition: all 0.4s ease;

box-shadow: 0 4px 10px rgba(0, 0, 0, 0.5);

}

.back-button:hover {

background-color: #cca300;

transform: scale(1.1);

}

@keyframes fadeIn {

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

100% { opacity: 1; transform: translateY(0); }

}

/\* Responsive Design \*/

@media (max-width: 900px) {

.flex-container {

flex-direction: column;

}

.box {

width: 100%;

}

h1 {

font-size: 2rem;

}

.result-box {

font-size: 1.2rem;

}

}

</style>

</head>

<body>

<div class="container">

<h1>Hello, {{ name }}! Here is your Mental Health prediction</h1>

<p class="result-box">📢 {{ result }}</p>

<div class="flex-container">

<div class="box">

<h2>User Details</h2>

<table>

<tr><th>Gender</th><td>{{ Gender }}</td></tr>

<tr><th>Age</th><td>{{ Age }}</td></tr>

<tr><th>Sleep Duration</th><td>{{ Sleep\_Duration }}</td></tr>

<tr><th>Dietary Habits</th><td>{{ Dietary\_Habits }}</td></tr>

<tr><th>Academic Pressure</th><td>{{ Academic\_Pressure }}</td></tr>

<tr><th>Degree</th><td>{{ Degree }}</td></tr>

<tr><th>Study Satisfaction</th><td>{{ Study\_Satisfaction }}</td></tr>

<tr><th>Work/Study Hours</th><td>{{ Work\_Study\_Hours }}</td></tr>

<tr><th>Financial Stress</th><td>{{ Financial\_Stress }}</td></tr>

<tr><th>Suicidal Thoughts</th><td>{{ Suicidal\_Thoughts }}</td></tr>

<tr><th>Family History</th><td>{{ Family\_History\_Mental\_Illness }}</td></tr>

<tr><th>Saving Data</th><td> {{ save\_message }}</td></tr>

</table>

</div>

<div class="box">

<h2>Prediction Chart</h2>

<canvas id="probabilityChart"></canvas>

</div>

</div>

<p class="result-box">

✅ Probability of Good Mental Health: <span style="color: #28a745; font-weight: bold;"> {{ prob\_good }}%</span> <br>

❌ Probability of Not Good Mental Health: <span style="color: #dc3545; font-weight: bold;"> {{ prob\_not\_good }}%</span>

</p>

<div class="suggestions">

<h2>Suggestions</h2>

<ul>

{% for suggestion in suggestions %}

<li>👉 {{ suggestion }}</li>

{% endfor %}

</ul>

</div>

<a href="/" class="back-button">Go Back</a>

</div>

<script>

new Chart(document.getElementById('probabilityChart').getContext('2d'), {

type: 'pie',

data: {

labels: ['Good Mental Health', 'Not Good Mental Health'],

datasets: [{

data: [{{ prob\_good }}, {{ prob\_not\_good }}],

backgroundColor: ['#28a745', '#dc3545'],

hoverBackgroundColor: ['#218838', '#c82333']

}]

},

options: {

responsive: true,

plugins: { legend: { position: 'top' } }

}

});

</script>

</body>

</html>