

```

from flask import Flask, render_template, request
import pickle
import pandas as pd
import numpy as np
from sklearn.preprocessing import OneHotEncoder

app = Flask(__name__)

# Load the model and OneHotEncoder
model = pickle.load(open("model.pkl", 'rb'))
ohe = pickle.load(open("ohe.pkl", 'rb')) # Load the OneHotEncoder

@app.route('/')
def welcome():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    age = request.form['age']
    gender = request.form['gender']
    occupation = request.form['occupation']
    line_of_work = request.form['line_of_work']
    prefer = request.form['prefer']
    certaindays_hw = request.form['certaindays_hw']
    time_bp = float(request.form['time_bp'])
    time_dp = float(request.form['time_dp'])
    travel_time = float(request.form['travel_time'])
    easeof_online = float(request.form['easeof_online'])
    home_env = float(request.form['home_env'])
    prod_inc = float(request.form['prod_inc'])
    sleep_bal = float(request.form['sleep_bal'])
    new_skill = float(request.form['new_skill'])
    fam_connect = float(request.form['fam_connect'])
    relaxed = float(request.form['relaxed'])
    self_time = float(request.form['self_time'])
    like_hw = float(request.form['like_hw'])
    dislike_hw = float(request.form['dislike_hw'])

    # Validate that the input values are not empty
    if '' in [age, gender, occupation, line_of_work, prefer,
certaindays_hw]:
        return render_template('index.html', predict="Please fill all
fields.")

    # Create a DataFrame from the input data
    input_data = pd.DataFrame({
        'age': [age],
        'gender': [gender],
        'occupation': [occupation],
        'line_of_work': [line_of_work],
        'prefer': [prefer],
        'certaindays_hw': [certaindays_hw],
        'time_bp': [time_bp],
        'time_dp': [time_dp],
        'travel_time': [travel_time],
        'easeof_online': [easeof_online],
        'home_env': [home_env],
        'prod_inc': [prod_inc],

```

```

        'sleep_bal': [sleep_bal],
        'new_skill': [new_skill],
        'fam_connect': [fam_connect],
        'relaxed': [relaxed],
        'self_time': [self_time],
        'like_hw': [like_hw],
        'dislike_hw': [dislike_hw]
    })

    # Transform categorical variables using the loaded OneHotEncoder
    encoded_data = ohe.transform(input_data[['age', 'gender',
'occupation', 'line_of_work', 'prefer', 'certaindays_hw']])
    encoded_data_df = pd.DataFrame(encoded_data,
columns=ohe.get_feature_names_out(['age', 'gender', 'occupation',
'line_of_work', 'prefer', 'certaindays_hw']))

    # Concatenate the encoded data with the rest of the input data
    input_features = pd.concat([encoded_data_df, input_data[['time_bp',
'time_dp', 'travel_time', 'easeof_online', 'home_env', 'prod_inc',
'sleep_bal', 'new_skill', 'fam_connect', 'relaxed', 'self_time',
'like_hw', 'dislike_hw']]], axis=1)

    # Make the prediction using the loaded model
    prediction = model.predict(input_features)

    # Return the prediction result
    return render_template('index.html', predict=f"Predicted value:
{prediction[0]}")

if __name__ == '__main__':
    app.run(debug=True, port=1111)

```