IBM Assignment - 1

Dharanikumar K
PNT2022PMID19316

Question 1:

Create registration page in html with username, email and phone number and by using POST method display it in next html page.

Code

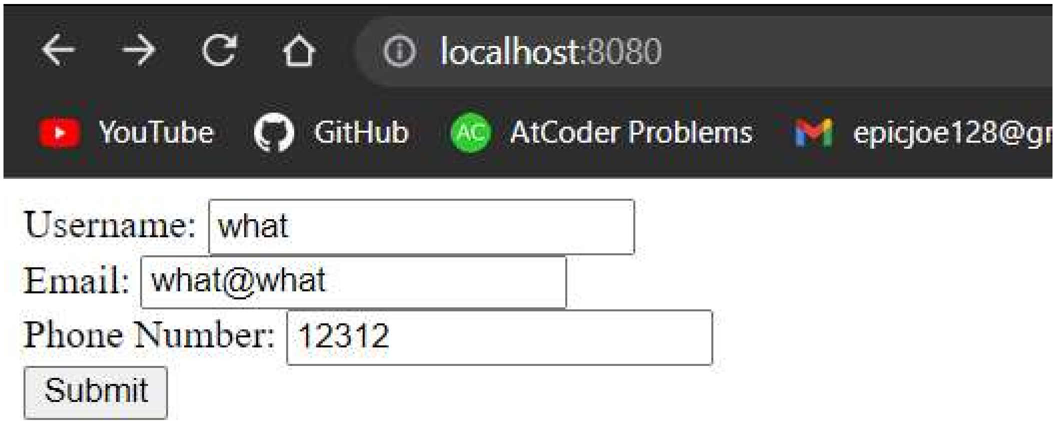
form.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <form action="http://localhost:8080/display" method="POST">
        Username: <input type="name" name="username" /> <br />
        Email: <input type="email" name="email" /> <br />
        Phone Number: <input type="number" name="phone" /> <br />
        <input type="submit" value="Submit" />
    </form>
</body>
</html>
```

server.js

```
const express = require('express');
const app = express();
const bodyParser = require('body-parser');
const path = require('path');

app.use(bodyParser.urlencoded({ extended: true }));


app.get('/', (request, response) => {
    const options = {
        root: path.join(__dirname),
    };
    response.sendFile('form.html', options);
});

app.post('/display', (request, response) => {
    const { username, email, phone } = request.body;
    let html = '';
    html += '<h1>Results</h1>';
    html += `<h2>Username: ${username}</h2>`;
    html += `<h2>Email: ${email}</h2>`;
    html += `<h2>Phone: ${phone}</h2>`;
    response.send(html);
});

app.listen(8080);
console.log('App listening at port 8080');
```

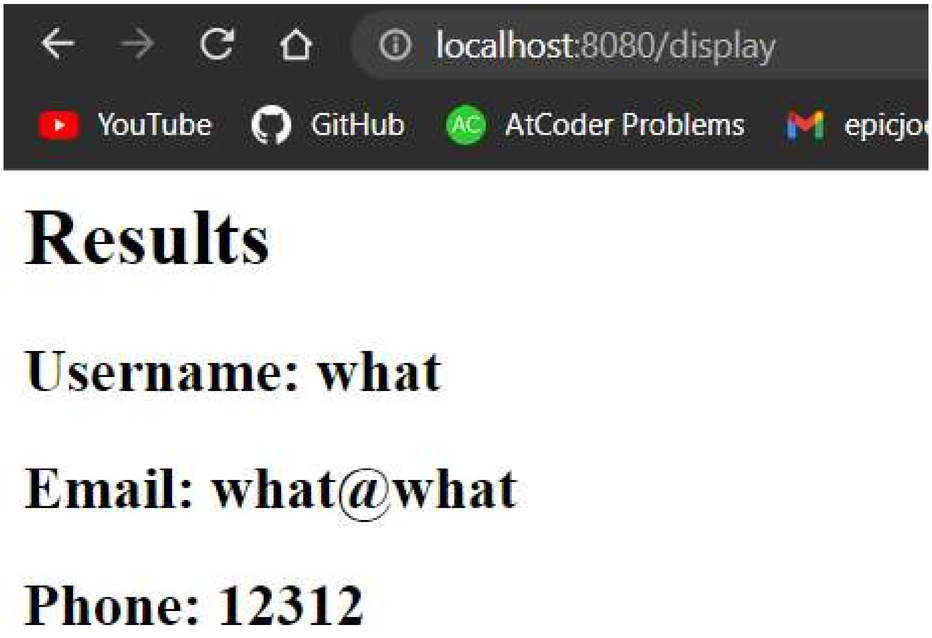
Output



Username: what

Email: what@what

Phone Number: 12312



Results

Username: what

Email: what@what

Phone: 12312

Question 2

Develop a flask program which should contain at least 5 packages used from pypi.org.

Code

select_package.html

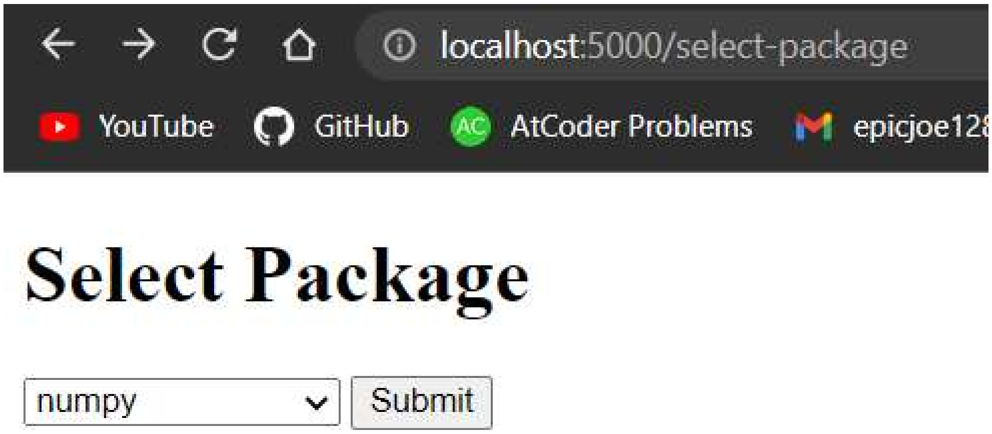
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Select Package</title>
</head>
<body>
    <h1>Select Package</h1>
    <form action="/use-package" method="POST">
        <select name="package" required>
            <option value="">Select a package</option>
            <option value="numpy">numpy</option>
            <option value="matplotlib">matplotlib</option>
            <option value="pandas">pandas</option>
            <option value="pendulum">pendulum</option>
            <option value="sklearn">sklearn</option>
        </select>
        <input type="submit" value="Submit" />
    </form>
</body>
</html>
```

app.py

```
import io

from flask import Response, Flask, request, jsonify

import numpy as np

from matplotlib.backends.backend_agg import FigureCanvasAgg as FigureCanvas
from matplotlib.figure import Figure

import pandas as pd
import pendulum

from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics

app = Flask(__name__)

@app.route('/select-package', methods=['GET'])
def select_package():
    html = open('select_package.html', 'r')
    return html.read()

@app.route('/use-package', methods=['POST'])
def use_package():
    package = request.form.get('package')
    if package == 'numpy':
        return use_numpy()
    if package == 'matplotlib':
        return use_matplotlib()
    if package == 'pandas':
        return use_pandas()
    if package == 'pendulum':
        return use_pendulum()
    if package == 'sklearn':
        return use_sklearn()

def use_numpy():
    a = np.array([1, 2, 3, 4, 5])
    b = np.array([5, 4, 3, 2, 1])
    return f'The dot product of {a} and {b} is {np.dot(a, b)}'

def use_matplotlib():
    fig = Figure()
    axis = fig.add_subplot(1, 1, 1)
    xs = [10, 20, 30]
    ys = [20, 30, 40]
    axis.plot(xs, ys)
    output = io.BytesIO()
    FigureCanvas(fig).print_png(output)
    return Response(output.getvalue(), mimetype='image/png')
```

```
def use_pandas():
    data = {
        'Subject': ['Maths', 'Physics', 'Chemistry'],
        'Marks': [100, 87, 98],
    }
    df = pd.DataFrame(data)
    return jsonify(df.to_dict(orient='records'))

def use_pendulum():
    utc_time = pendulum.now('UTC')
    kolkata_time = utc_time.in_timezone('Asia/Kolkata')
    sydney_time = utc_time.in_timezone('Australia/Sydney')
    return f'Current date time in Kolkata = {kolkata_time}<br/>Current date
time in Sydney = {sydney_time}'

def use_sklearn():
    iris = load_iris()
    X  =  iris.data
    y = iris.target
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.4, random_state = 1)

    classifier_knn = KNeighborsClassifier(n_neighbors = 3)
    classifier_knn.fit(X_train, y_train)

    y_pred = classifier_knn.predict(X_test)

    html = ''
    html += '<h3>Iris Data Set ML Predictions using KNN</h3>'
    html += '<h2>Training Data Input</h2>'
    html += str(X_train)
    html += '<h2>Training Data Output</h2>'
    html += str(y_train)
    html += '<h2>Y Predictions:</h2>'
    html += '<p>' + str(y_pred) + '</p>'
    html += '<h2>Y Test:</h2>'
    html += '<p>' + str(y_test) + '</p>'
    html += '<h2>' + 'Accuracy' + '</h2>' +
str(metrics.accuracy_score(y_test, y_pred))
    return html

if___name___== '__main__':
    app.run(debug=True, port=5000)
```

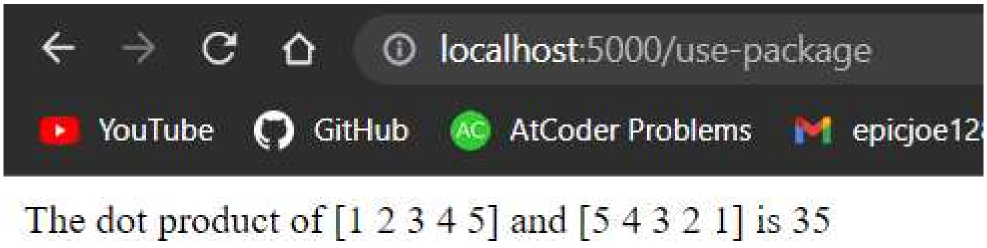
Output



localhost:5000/select-package

Select Package

numpy



localhost:5000/use-package

```
The dot product of [1 2 3 4 5] and [5 4 3 2 1] is 35
```