SET-3

LAB-1

A program that creates a simple web server and serves a static HTML page.

app.py

from flask import Flask,render\_template

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

@app.route ('/')

def abc ():

    return render\_template("index.html")

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

 app.run(debug=True)

index.html

<html lang="en">

<head>

  static website

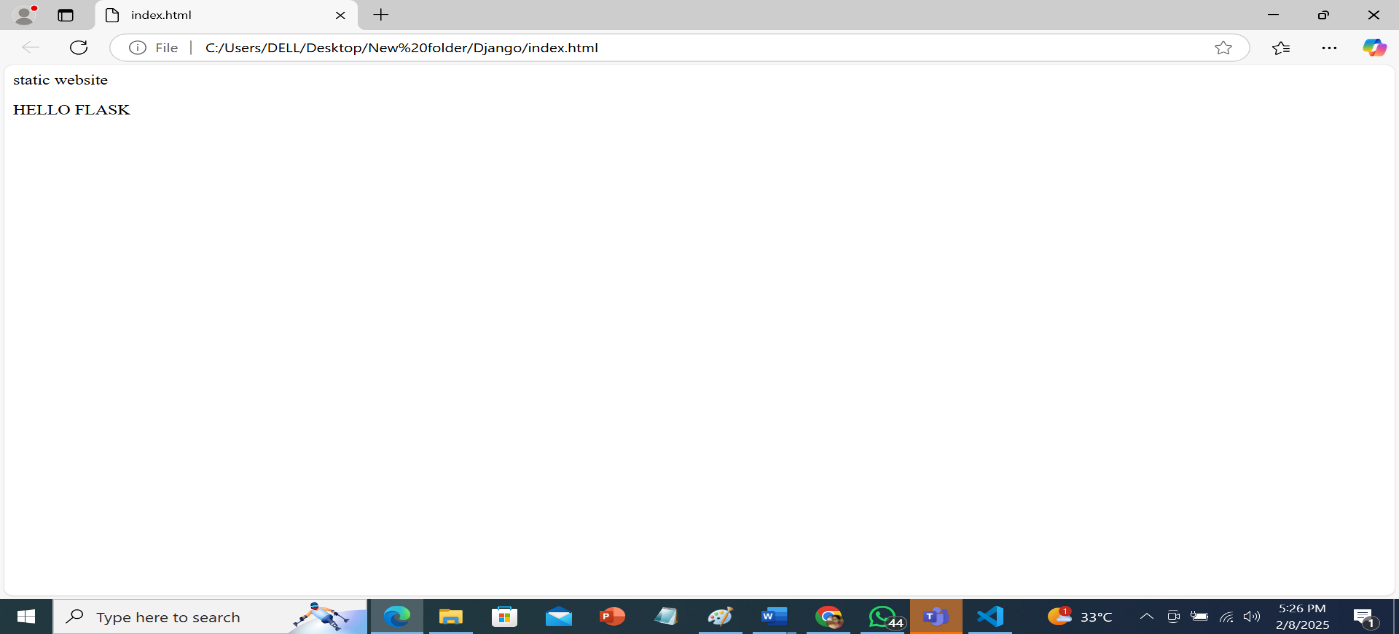
  </head>

<body>

    <p> HELLO FLASK </p>

</body>

</html>



Output:

Lab-2

A program that creates a web application that allows users to register and login.

app.py

from flask import Flask, render\_template

from flask\_sqlalchemy import SQLAlchemy

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

# Configure the SQLite database

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///db.sqlite'

app.config['SQLALCHEMY\_TRACK\_MODIFICATIONS'] = False

# Initialize the database

db = SQLAlchemy(app)

# Define the User model

class User(db.Model):

    id = db.Column(db.Integer, primary\_key=True)  # Primary Key

    user\_name = db.Column(db.String(1000), nullable=False)  # Username

    password = db.Column(db.String(1000), nullable=False)  # Password

# Home route

@app.route('/')

def Home():

    return render\_template("home.html")

# Register route

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

def Register():

    return render\_template("register.html")

# Login route

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

def Login():

    return render\_template("login.html")

if \_\_name\_\_ == "\_\_main\_\_":

    # Create database tables if they don't exist

    with app.app\_context():

        db.create\_all()

    app.run(debug=True)

login.html

<!DOCTYPE html>

<html>

<head>

    <title>Login</title>

</head>

<body>

    <h1>Login</h1>

    <form method="POST">

        <label>Username:</label>

        <input type="text" name="username" required><br>

        <label>Password:</label>

        <input type="password" name="password" required><br>

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

    </form>

    <a href="/">Back to Home</a>

</body>

</html>

Register.html

<!DOCTYPE html>

<html>

<head>

    <title>Register</title>

</head>

<body>

    <h1>Register</h1>

    <form method="POST">

        <label>Username:</label>

        <input type="text" name="username" required><br>

        <label>Password:</label>

        <input type="password" name="password" required><br>

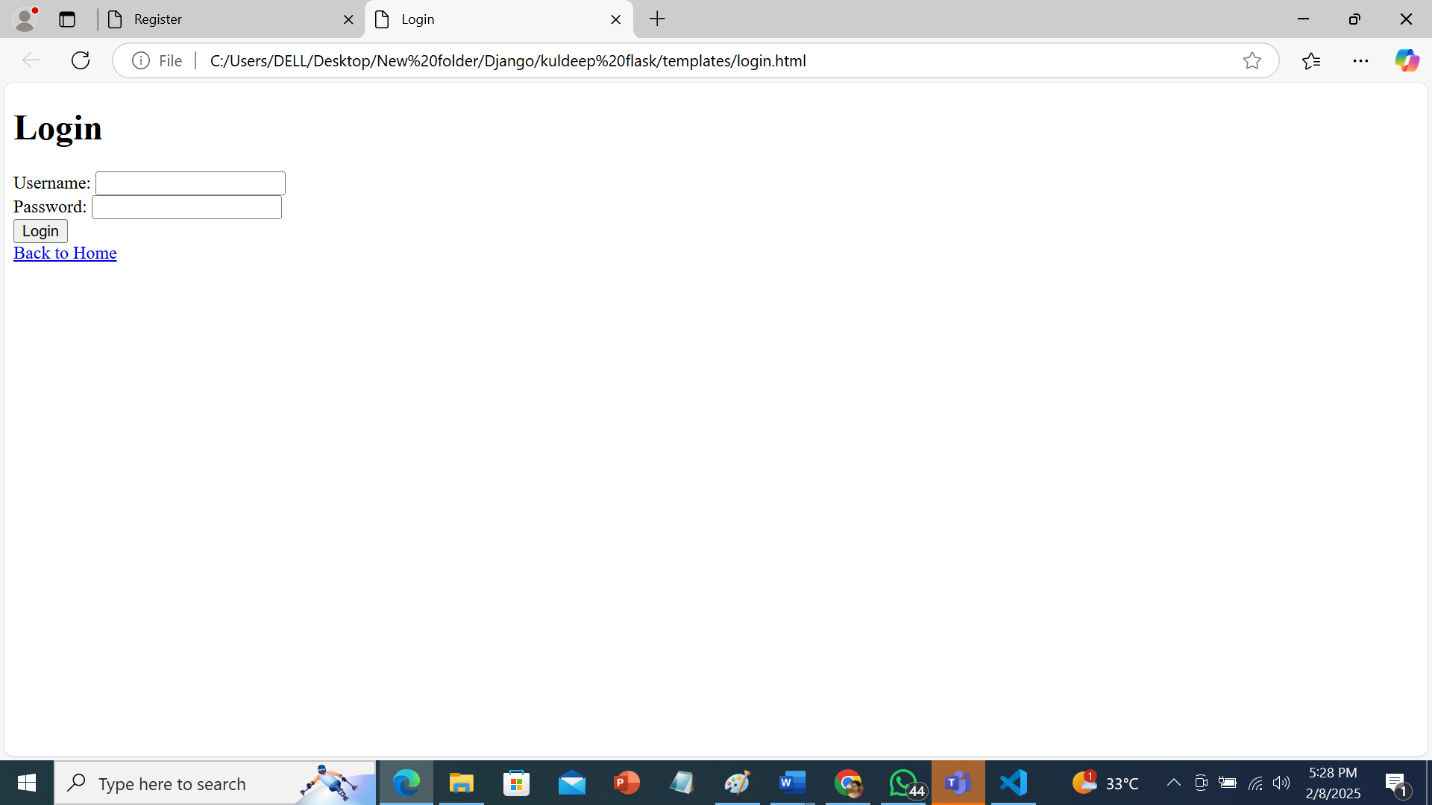
        <button type="submit">Register</button>

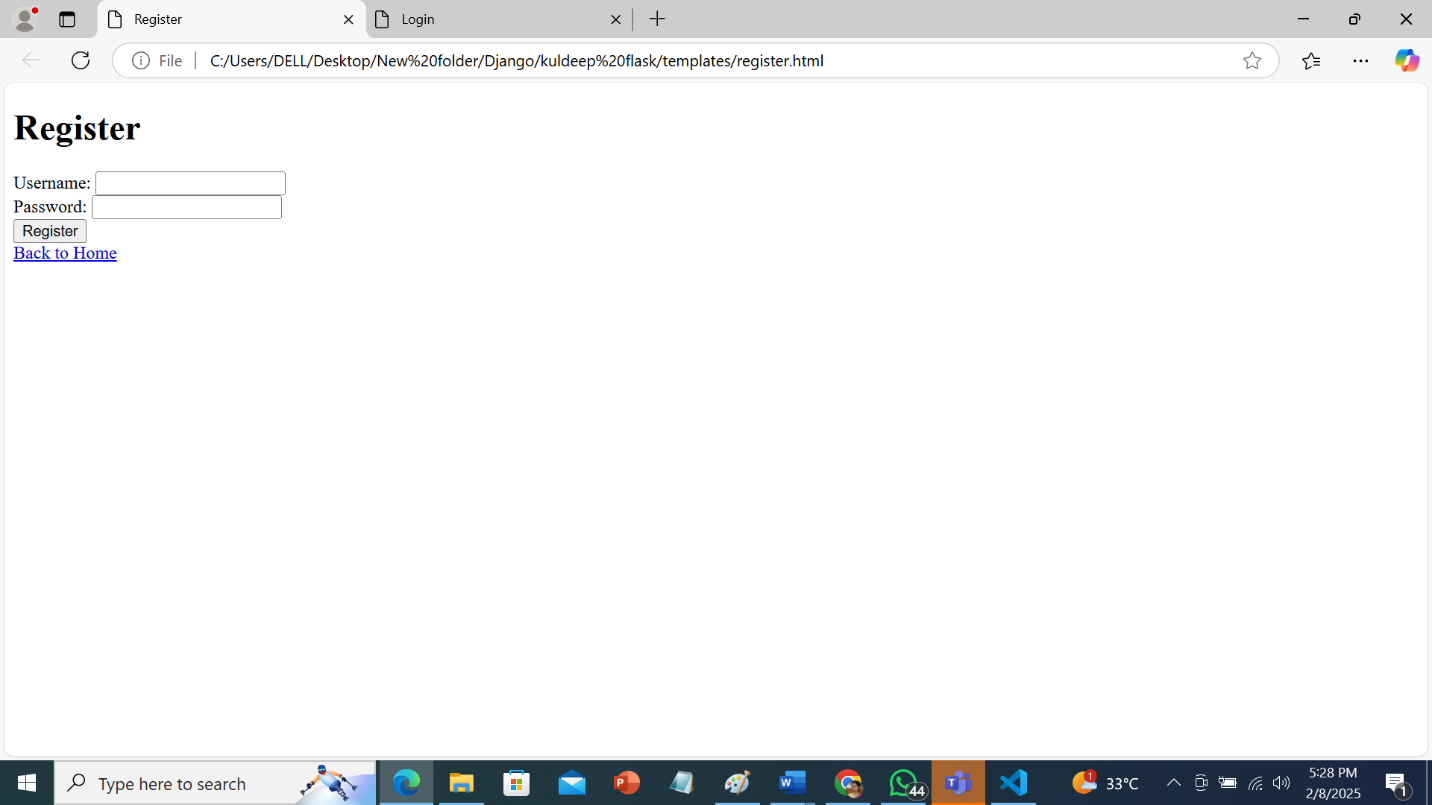
    </form>

    <a href="/">Back to Home</a>

</body>

</html>





Lab-3

A program that creates a web application that allows users to upload and download files.

app.py

# import all libraires

from io import BytesIO

from flask import Flask, render\_template, request, send\_file

from flask\_sqlalchemy import SQLAlchemy

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

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///db.sqlite3'

db = SQLAlchemy(app)

class Upload(db.Model):

    id = db.Column(db.Integer, primary\_key=True)

    filename = db.Column(db.String(50))

    data = db.Column(db.LargeBinary)

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

def index():

    if request.method == 'POST':

        file = request.files['file']

        upload = Upload(filename=file.filename, data=file.read())

        db.session.add(upload)

        db.session.commit()

        return f'Uploaded: {file.filename}'

    return render\_template('file.html')

@app.route('/download/<upload\_id>')

def download(upload\_id):

    upload = Upload.query.filter\_by(id=upload\_id).first()

    return send\_file(BytesIO(upload.data), download\_name=upload.filename, as\_attachment=True)

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

    with app.app\_context():

     db.create\_all()

    app.run(debug=True)

file.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>File Upload Example</title>

    <style>

    </style>

</head>

<body>

    <div class="center">

        <h1> Uploading and Returning Files With a Database in Flask </h1>

        <form method="POST" action="/" enctype="multipart/form-data">

            <input  type="file" name="file">

            <button>Submit</button>

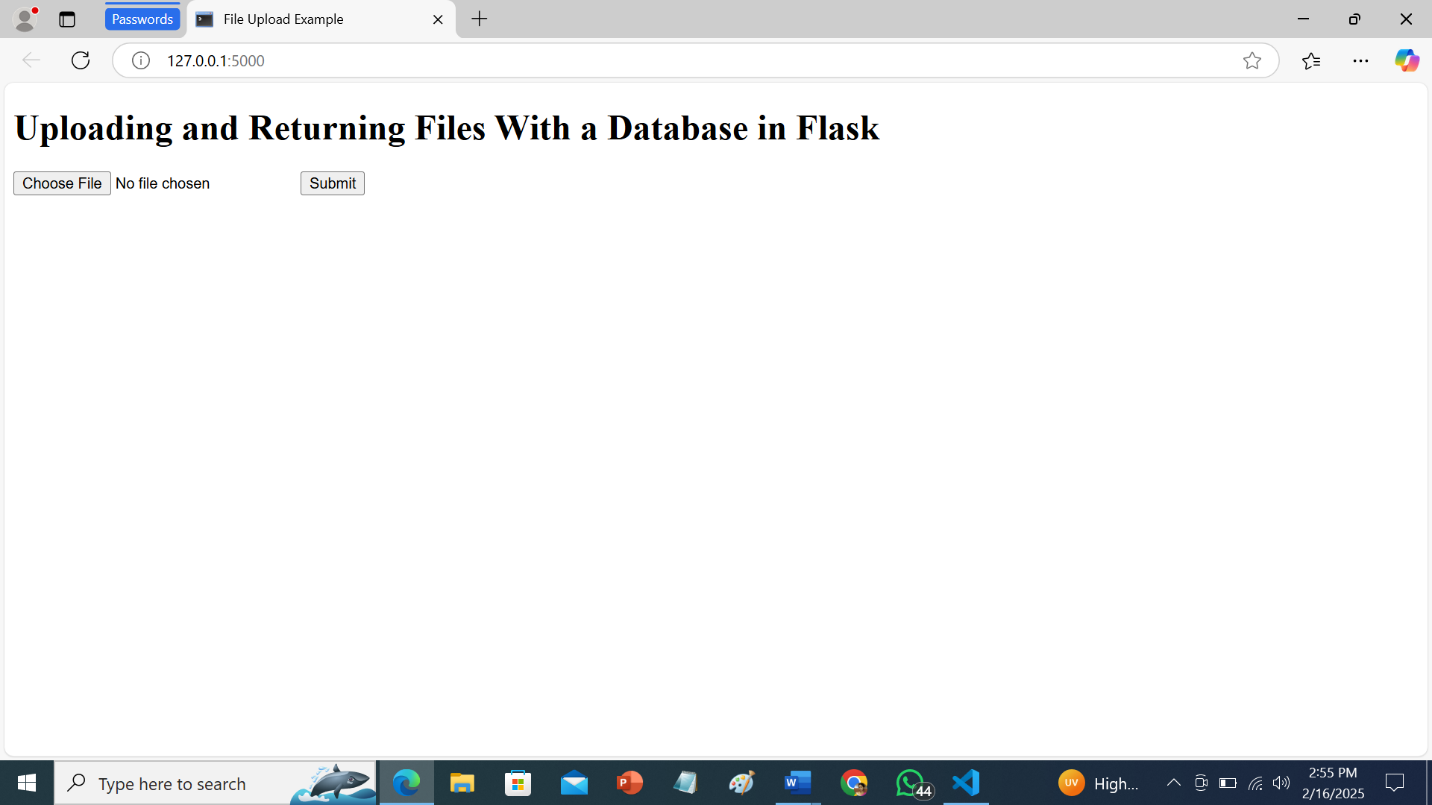
        </form>

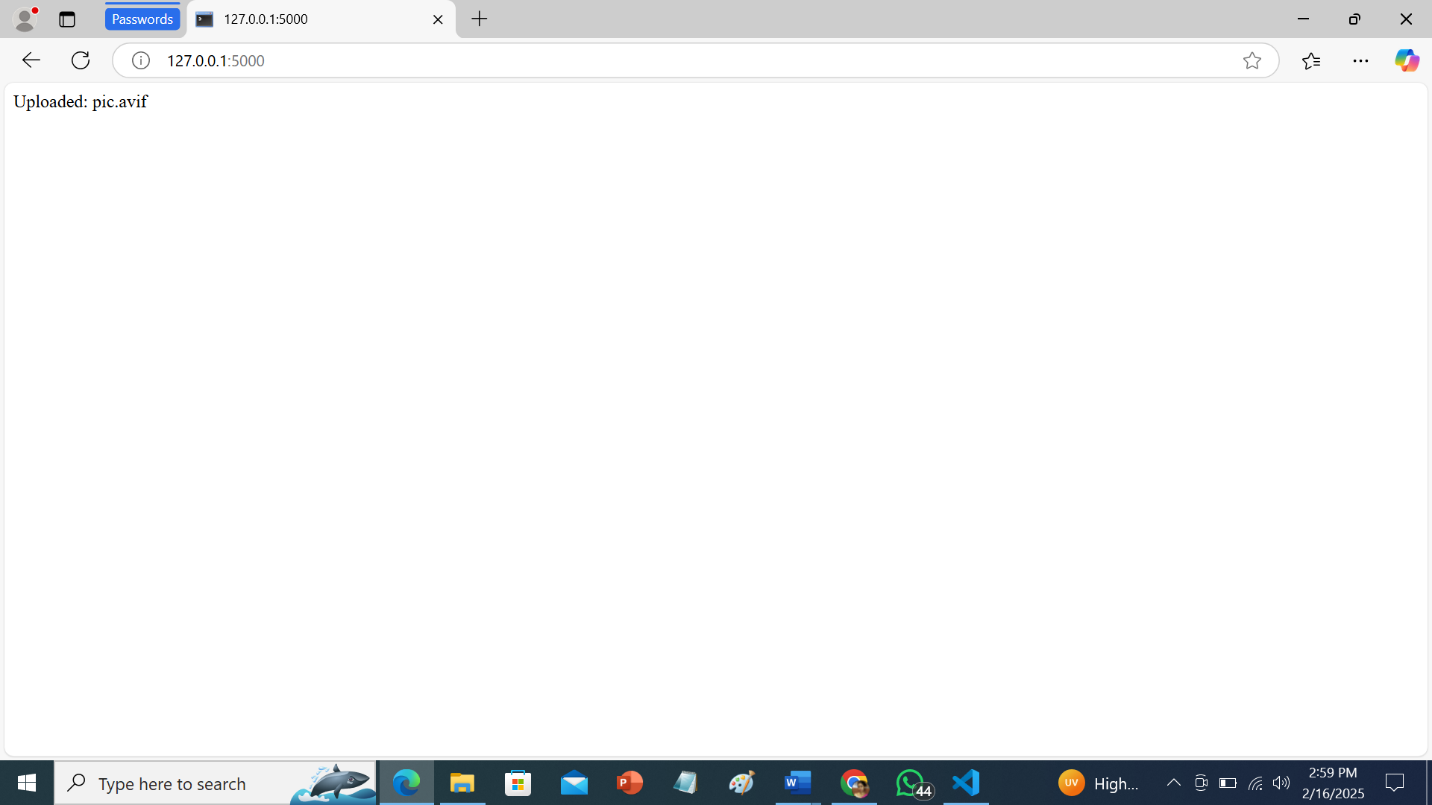
    </div>

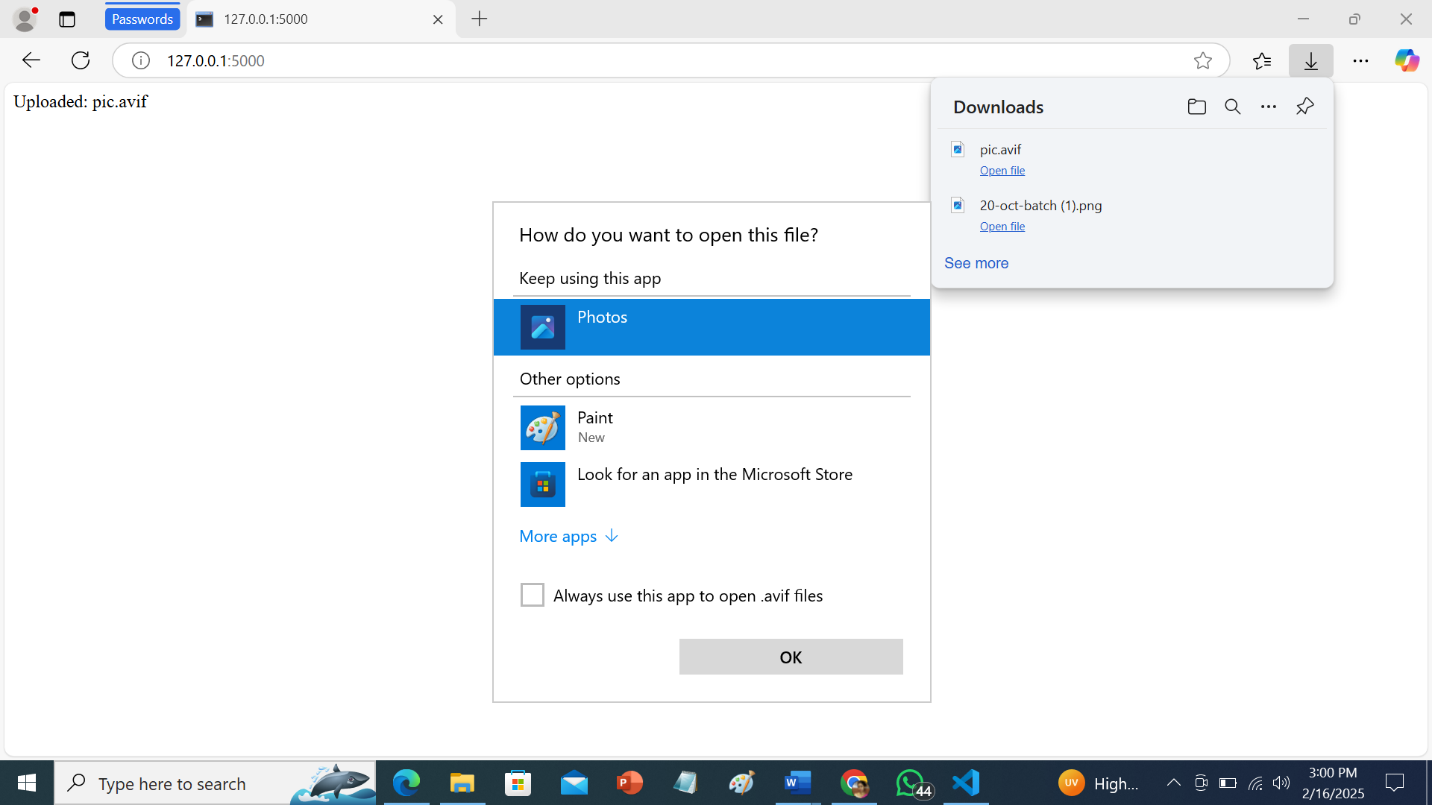
</body>

</html>

Output:







Lab-4

A program that creates a web application that displays data from a database in a tabular format.

app.py

from flask import Flask, render\_template, request

from flask\_sqlalchemy import SQLAlchemy

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

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///abc.db'

db = SQLAlchemy(app)

class User(db.Model):

    id = db.Column(db.Integer, primary\_key=True)

    user\_name = db.Column(db.String(200), nullable=False)

    user\_password = db.Column(db.String(200), nullable=False)

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

def home():

    if request.method == "POST":

        user\_name = request.form["name"]

        user\_password = request.form["password"]

        new\_user = User(user\_name=user\_name, user\_password=user\_password)

        db.session.add(new\_user)

        db.session.commit()

    return render\_template("index2.html")

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

    with app.app\_context():

        db.create\_all()

    app.run(debug=True)

index2.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>User Registration</title>

</head>

<body>

    <form action="/" method="POST">

        <label>Username:</label>

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

        <br>

        <label>Password:</label>

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

        <br>

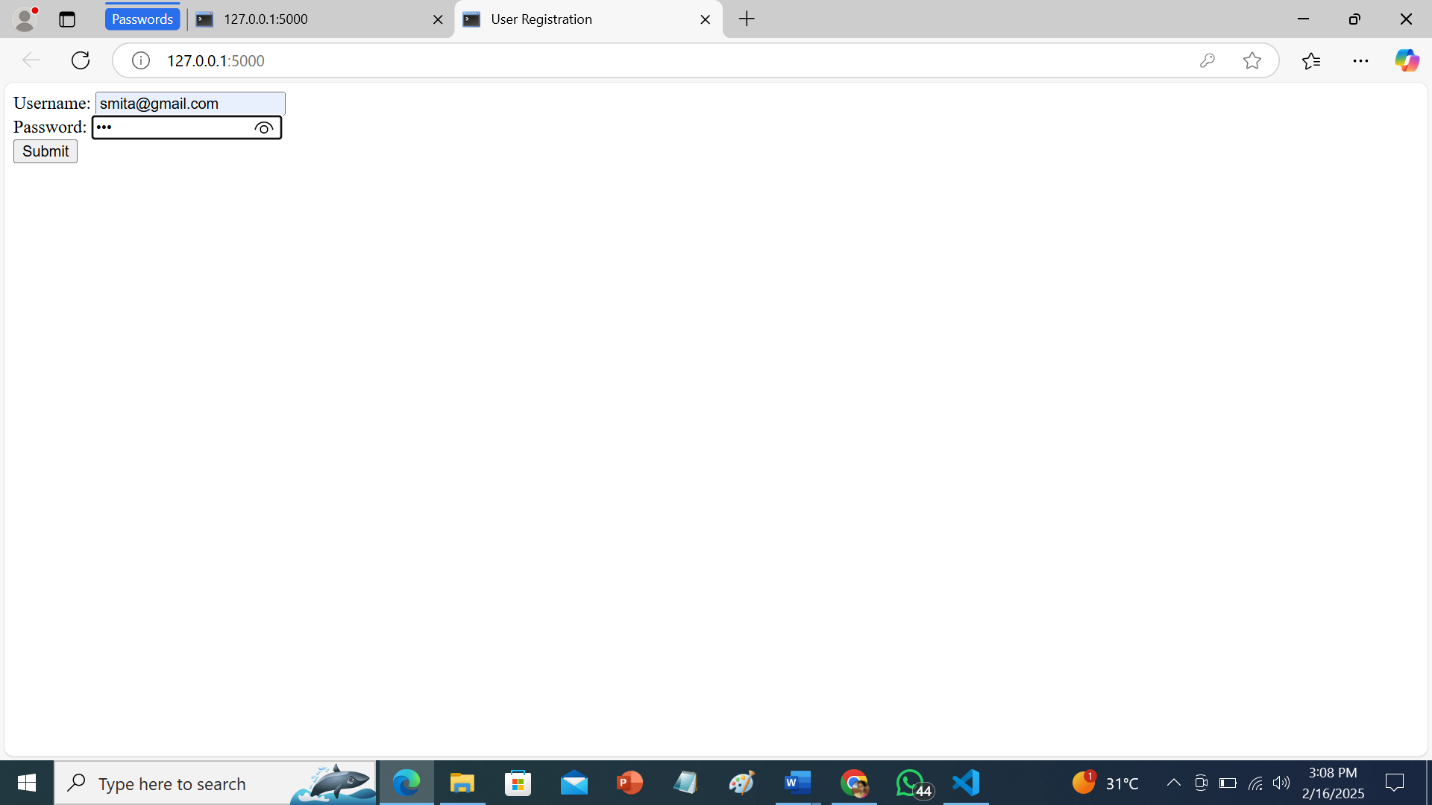
        <button type="submit">Submit</button>

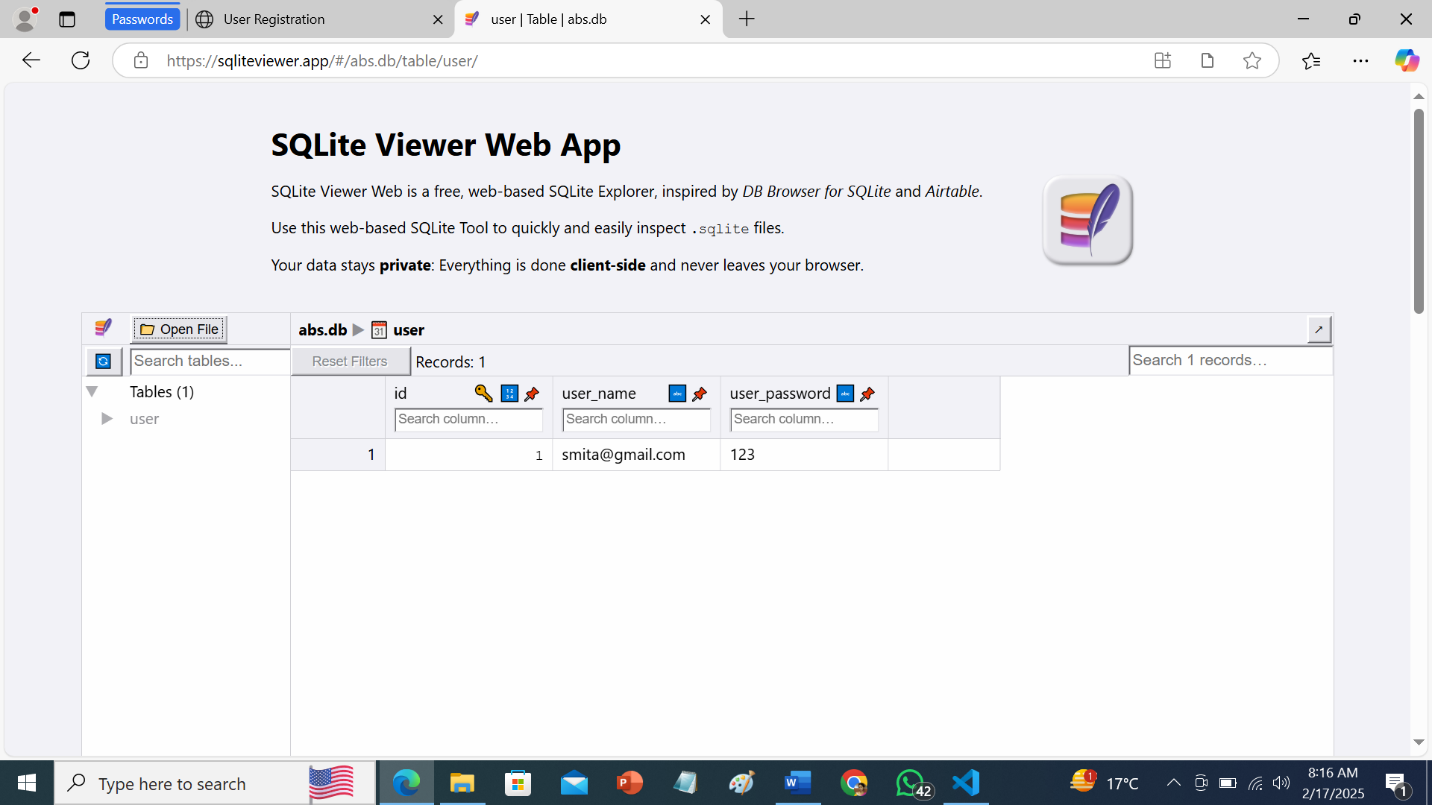
    </form>

</body>

</html>

Output:





Lab-5

A program that creates a web application that accepts user input and sends it to a server-side script for processing.

app.py

from flask import Flask, render\_template, request

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

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

def abc():

    if request.method == "POST":

        result = request.form

        return render\_template("home.html", result=result)

    return render\_template("home.html", result=None)  # Handle GET requests properly

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True)  # Removed the misplaced text

home.html

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <link rel="stylesheet" href="{{ url\_for('static', filename='css/index.css') }}">

  <title>Document</title>

</head>

<body>

  <form action="/home" method="post">

    <p>Name: <input type="text" name="name"></p>

    <p>Number: <input type="number" name="number"></p>

    <p>Email: <input type="email" name="email"></p>

    <p>Password: <input type="password" name="password"></p>

    <p>Comment: <input type="text" name="comment"></p>

    <p><input type="submit" value="Submit"></p>

  </form>

  {% if result %}

  <table border="1">

    <tr>

      <th>Field</th>

      <th>Value</th>

    </tr>

    {% for i, j in result.items() %}

    <tr>

      <td>{{ i }}</td>

      <td>{{ j }}</td>

    </tr>

    {% endfor %}

  </table>

  {% endif %}

</body>

</html>

Output:

