Lab 12: Flask Introduction

1. **Flask Overview**
   Flask is a lightweight web framework written in Python. It allows you to build web applications quickly with minimal setup. Flask follows the MVC (Model-View-Controller) pattern and is great for both beginners and professionals due to its simplicity and flexibility.

2. **Basic Structure of a Flask Application**
   A typical Flask project is structured as follows:

   - **app.py**: The main Python file where the Flask application is defined and run.
   - **templates/**: Contains HTML files (Jinja2 templating).
     • Example: index.html — the main web page.
   - **static/**: Contains static files like CSS, JavaScript, and images.
     • Example: style.css, script.js

3. **Setting Up Flask**

   To run a Flask application:

   Step 1: Install Flask
   Open your terminal or command prompt and run:

   pip install flask

   Step 3: Run the Flask App
   Navigate to the project folder where app.py is located, and run the following:

   python app.py

   Step 4: Access the App
   Once the app is running, open your browser and go to:

   http://127.0.0.1:5000

Example: Flask Based Tic Tac Toe Game

This project creates a 2-player Tic Tac Toe game using Flask, with a glowing aesthetic grid made from clickable elements. It shows the winner at the end and includes a "Reset Game" button.

Folder Structure:

```
project/
│
├── app.py
├── static/
│   ├── style.css
│   └── script.js
└── templates/
    └── index.html
```

app.py

```
from flask import Flask, render_template

app = Flask(__name__)

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

if __name__ == '__main__':
    app.run(debug=True)
```

templates/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Tic Tac Toe</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
    <h1>Tic Tac Toe</h1>
    <div id="gameBoard">
        {% for i in range(9) %}
        <div class="cell" onclick="makeMove(this)"></div>
        {% endfor %}
    </div>
    <div id="status"></div>
    <button onclick="resetGame()">Reset Game</button>

    <script src="{{ url_for('static', filename='script.js') }}"></script>
</body>
</html>
```

static/style.css

```
body {
    text-align: center;
    font-family: Arial, sans-serif;
    background: #0f0f0f;
    color: #fff;
}

h1 {
    margin-top: 20px;
    color: #00ffcc;
    text-shadow: 0 0 10px #00ffcc;
}

#gameBoard {
    display: grid;
    grid-template-columns: repeat(3, 100px);
    grid-gap: 10px;
    justify-content: center;
    margin: 30px auto;
}

.cell {
    width: 100px;
    height: 100px;
    background: #222;
    border: 2px solid #00ffcc;
    display: flex;
    align-items: center;
    justify-content: center;
```

```
    font-size: 36px;
    cursor: pointer;
    color: #00ffcc;
    box-shadow: 0 0 10px #00ffcc;
    transition: 0.3s;
}

.cell:hover {
    background: #333;
}

#status {
    font-size: 24px;
    margin: 20px;
    color: #ff66cc;
}

button {
    padding: 10px 20px;
    font-size: 16px;
    background: #ff66cc;
    border: none;
    border-radius: 5px;
    color: #fff;
    cursor: pointer;
    box-shadow: 0 0 10px #ff66cc;
}
```

static/script.js

```
let currentPlayer = "X";
let gameActive = true;
let board = ["", "", "", "", "", "", "", "", ""];

function makeMove(cell) {
    const index = Array.from(cell.parentNode.children).indexOf(cell);
    if (!gameActive || board[index]) return;

    board[index] = currentPlayer;
    cell.textContent = currentPlayer;

    if (checkWinner()) {
        document.getElementById("status").textContent = `${currentPlayer} Wins!`;
        gameActive = false;
    } else if (!board.includes("")) {
        document.getElementById("status").textContent = "It's a Draw!";
        gameActive = false;
    } else {
        currentPlayer = currentPlayer === "X" ? "O" : "X";
    }
}

function checkWinner() {
    const winPatterns = [
        [0,1,2], [3,4,5], [6,7,8],
        [0,3,6], [1,4,7], [2,5,8],
        [0,4,8], [2,4,6]
    ];
```

```
    return winPatterns.some(pattern =>
        pattern.every(index => board[index] === currentPlayer)
    );
}

function resetGame() {
    board = ["", "", "", "", "", "", "", "", ""];
    currentPlayer = "X";
    gameActive = true;
    document.querySelectorAll(".cell").forEach(cell => cell.textContent = "");
    document.getElementById("status").textContent = "";
}
```

Lab 12 Task

1. **(Continue from Lab 11):**
    - Develop Model's Application Phase in Flask
    - Use ChatGPT for the frontend code