1. **Write code for a simple user registration form for an event.**

**Html code File-name: index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Registration Form</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div class="container">

<h1>Register</h1>

<form id="registrationForm">

<input type="text" id="name" placeholder="Full Name" required>

<input type="email" id="email" placeholder="Email" required>

<input type="tel" id="phone" placeholder="Phone" required>

<select id="event" required>

<option value="">Select Event</option>

<option value="workshop">Workshop</option>

<option value="seminar">Seminar</option>

</select>

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

</form>

<p id="successMessage" style="display:none;">Registration successful!</p>

</div>

<script src="script.js"></script>

</body>

</html>

**CSS code File name: styles.css**

body, .container { font-family: Arial, sans-serif; text-align: center; }

.container { margin-top: 100px; }

input, select, button { display: block; margin: 10px auto; padding: 10px; width: 80%; }

button { background-color: #28a745; color: white; cursor: pointer; }

**Javascript code File-name: script.js**

document.getElementById('registrationForm').addEventListener('submit', function(event) {

event.preventDefault();

document.getElementById('successMessage').style.display = 'block';

this.reset();

});

**Output:**

**Execute the code and write your output over here**

**=======================================================**

**Additional Program by using Python flask and Docker**

1. **Write code for a simple user registration form for an event. Using Python flask and Docker**

**Aim**: Write code for a simple user registration form for an event

DESCRIPTION:

Here's an example of a simple user registration form using Flask and Docker

in DevOps:

**Step-1:** Create a Docker file with the following content to create a Docker

image for your Flask application:

FROM python:3.8

WORKDIR /app

COPY . .

RUN pip install --no-cache-dir -r requirements.txt

EXPOSE 5000

CMD ["python", "app.py"]

**Step-2:** Create a requirements.txt file with the following content to list the

dependencies of your Flask application: Flask==1.1.2

**Step-3:** Create a app.py file with the following code for a simple user

registration form in Flask:

from flask import Flask, request, render\_template

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

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

def register():

if request.method == 'POST':

name = request.form['name']

email = request.form['email']

password = request.form['password']

# Store the user data in a database or file

return render\_template('success.html')

return render\_template('register.html')

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

app.run(host='0.0.0.0')

**Step-4:** Create an templates folder and add the following two files:

register.html and success.html.

register.html

<form method="post">

<input type="text" name="name" placeholder="Name">

<input type="email" name="email" placeholder="Email">

<input type="password" name="password" placeholder="Password">

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

</form>

success.html

<h2>Registration Successful</h2>

**Step-5:** Build the Docker image for your Flask application using the

following command:

docker build -t simple-flask-app .

**Step-6:** Run a Docker container from the image using the following command:

docker run -p 5000:5000 simple-flask-app

**Step-7:** Open a web browser and access the registration form at

http://localhost:5000/register.

This example demonstrates how to build a simple user registration form in Flask and run it in a Docker container in DevOps. Note that this code is only meant to demonstrate the basic structure of a user registration form and does not include any security measures or proper error handling. It is highly recommended to add security measures such as password hashing and validation before using it in a production environment

VIVA QUESTIONS

Define Flask in devops

**VIVA QUESTIONS**

Define Flask in Devops

**Output:**

**Execute the code and write your output over here**