DevOps Lab

List of Experiments:

- **Week 1:** Write code for a simple user registration form for an event.
- **Week 2:** Explore Git and GitHub commands.
- **Week 3:** Practice Source code management on GitHub. Experiment with the source code written in exercise 1.
- Week 4: Jenkins installation and setup, explore the environment.
- **Week 5:** Demonstrate continuous integration and development using Jenkins.
- **Week 6:** Explore Docker commands for content management.
- **Week 7:** Develop a simple containerized application using Docker.
- **Week 8:** Integrate Kubernetes and Docker.
- **Week 9:** Automate the process of running containerized application developed in exercise 7 using Kubernetes.
- Week 10: Install and Explore Selenium for automated testing.
- **Week 11:** Write a simple program in JavaScript and perform testing using Selenium.
- **Week 12:** Develop test cases for the above containerized application using selenium.

Week	Activity	
Week 1	Application Development	
Week 2, 3	Source control using Git + GitHub	
Week 4, 5	CI/CD pipeline setup using Jenkins to auto-	
	build this Flask app	
Week 6, 7	Containerize the app using Docker	
Week 8, 9	Deploy using Kubernetes	
Week 10, 12	Automate UI testing with Selenium	

Week-1

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

DevOps is the combination of Development + Operations, so in **Week 1**, we focus on the **development** aspect by creating a **simple user registration form**. This form serves as the foundational application and can be developed using various technologies such as:

Method	Description	Tools / Technologies
1. Static Web Page	Simple HTML + CSS	VS Code, Browser
2. With JavaScript	Add validation and interactivity	JS, Bootstrap
3. Using Python Flask	Backend processing	Flask, Python, HTML
4. Using Node.js + Express	JavaScript-based server-side rendering	Node.js, Express
5. Using React or Angular	Modern frontend frameworks	ReactJS, Angular
6. Using Django (Python)	Full-stack web framework	Django
7. PHP + MySQL	Traditional stack	XAMPP/LAMP

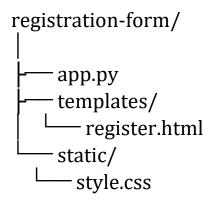
We will go with method 3 - Python + Flask, since DevOps tools like Jenkins, Docker, Kubernetes, etc., are often used to deploy and test Flask apps.

Tools Needed:

- Python 3.7+
- Flask (use pip install flask)
- Command Line Interface
- Browser

Implementation:

• Step 1: Set up project structure



• Step 2: app.py - Main Flask Application

from flask import Flask, render_template, request, redirect, url_for
app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def home():
 if request.method == 'POST':
 print("Form submitted successfully!")
 print("Form Data:", request.form)
 return "Registration successful!"
 return render_template('register.html')

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

• Step 3: register.html - HTML Template in templates folder

```
.form-container {
      margin-top: 50px;
      padding: 30px;
      background: white;
      border-radius: 10px;
      box-shadow: 0 0 10px rgba(0,0,0,0.15);
  </style>
</head>
<body>
<div class="container">
  <div class="row justify-content-md-center">
    <div class="col-md-6 form-container">
      <h2 class="text-center mb-4">User Registration Form</h2>
      <form method="POST" action="/">
        <div class="mb-3">
          <label class="form-label">Full Name</label>
          <input type="text" class="form-control" name="full_name">
        </div>
        <div class="mb-3">
          <label class="form-label">Email Address</label>
          <input type="email" class="form-control" name="email">
        </div>
        <div class="mb-3">
          <label class="form-label">Username</label>
          <input type="text" class="form-control" name="username">
        </div>
        <div class="mb-3">
          <label class="form-label">Password</label>
          <input type="password" class="form-control"</pre>
name="password">
        </div>
        <div class="mb-3">
          <label class="form-label">Confirm Password</label>
```

```
<input type="password" class="form-control"</pre>
name="confirm_password">
       </div>
       <div class="mb-3">
         <label class="form-label">Phone Number</label>
         <input type="tel" class="form-control" name="phone">
       </div>
       <div class="mb-3">
         <label class="form-label">Date of Birth</label>
         <input type="date" class="form-control" name="dob">
       </div>
       <div class="mb-3">
         <label class="form-label">Gender</label>
         <select class="form-select" name="gender">
           <option value="">Select Gender</option>
           <option>Male
           <option>Female
           <option>Other</option>
         </select>
       </div>
       <div class="mb-3">
         <label class="form-label">Address</label>
         <textarea class="form-control" rows="3"
name="address"></textarea>
       </div>
       <div class="text-center">
         <button type="submit" class="btn btn-primary px-
5">Register</button>
       </div>
     </form>
   </div>
 </div>
</div>
</body>
</html>
```

To run the above program:

- ▶ pip install flask
- > python app.py
- ➤ Visit: http://127.0.0.1:5000/ in your browser.

Output:

