Optimizing Deployment Speed and Reliability with DevOps

PHASE 4 – Testing

College Name: Dr. Sri Sri Sri Shivakumara Mahaswamy College of Engineering, Bengaluru

Group Members:

- **Name: Binod Khatri Chatri**
  CAN ID Number: CAN_35805323
- **Name: Ashwini J**
  CAN ID Number: CAN_35777960
- **Name: Kavana M**
  CAN ID Number: CAN_35778518
- **Name: Shashi kumar M**
  CAN ID Number: CAN_35805208

Testing :

Ensure application reliability and stability by integrating automated testing into your CI/CD pipeline.

1. Project Structure Overview

```
project-root/
├── client/        ← React App
│   ├── src/
│   │   └── __tests__/      ← Unit tests
├── server/        ← Node.js Express API
│   └── tests/      ← Unit/API tests
├── .github/
│   └── workflows/
│       └── deploy.yml
```

2. Add Tests

2.1 React Frontend Tests

Install testing tools (if not already):

```
cd client
npm install --save-dev @testing-library/react jest
```

Create a sample test: client/src/__tests__/App.test.js

```
import { render, screen } from '@testing-library/react';
import App from '../App';
test('renders welcome message', () => {
  render(<App />);
  expect(screen.getByText(/welcome/i)).toBeInTheDocument();
});
```

Update client/package.json:

```
"scripts": {
  "test": "react-scripts test --watchAll=false"
}
```

2.2 Express Backend Tests

Install testing tools:

```
cd server
npm install --save-dev jest supertest
```

Create test: server/tests/api.test.js

```
const request = require('supertest');
const app = require('../index'); // ensure app is exported

test('GET / responds with server status', async () => {
  const res = await request(app).get('/');
  expect(res.statusCode).toBe(200);
  expect(res.text).toMatch(/server/i);
});
```

Update server/package.json:

```
"scripts": {
  "test": "jest"
}
```

Update index.js to export app:

```
module.exports = app;
```

3. Integrate Tests in GitHub Actions

Update .github/workflows/deploy.yml:

```
jobs:
  build-test-deploy:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v3

    - name: Set up Node.js
      uses: actions/setup-node@v3
      with:
        node-version: '18'

    - name: Install frontend dependencies
      run: |
        cd client
        npm install

    - name: Run frontend tests
      run: |
        cd client
        npm test

    - name: Install backend dependencies
      run: |
        cd server
        npm install

    - name: Run backend tests
      run: |
        cd server
        npm test
```