Lab 3: Introduction to Deployment Tools

Objective

To understand the basics of web application deployment and gain hands-on experience using common deployment tools such as **Docker**, **Netlify**, **Vercel**, and **GitHub Pages**.

Theory

What is Deployment?

Deployment is the process of publishing a web application to a live environment, so that users can access it via the internet. Modern deployment tools automate the process, often integrating with Git repositories for CI/CD.

Common Deployment Tools

Docker

- **Purpose**: Packages applications and their dependencies into containers for consistent and portable deployment.
- **Features**:
  - Works on any environment that supports Docker.
  - Great for full-stack or backend services.
  - Commonly used in enterprise and production environments.
- **Use Case**: Running microservices, scalable APIs, or entire environments in containers.

Netlify

- **Best For**: Static websites (e.g., HTML, CSS, JS).
- **Features**:
  - Git integration for CI/CD.
  - Drag-and-drop deployment.
  - Serverless functions.

Vercel

- **Best For**: Frontend frameworks like React, Next.js.
- **Features**:
  - Auto-deploys from GitHub/GitLab/Bitbucket.
  - Optimized for frontend and JAMstack apps.
  - Edge network for fast performance.

GitHub Pages

- **Best For**: Hosting static websites directly from a GitHub repo.
- **Features**:
  - Free and easy to set up.
  - Custom domain support.
  - Limited to static content only.

Procedure

Deploying with Docker

Create a Dockerfile in your project root:
Dockerfile

```
FROM node:22

WORKDIR /app

COPY . .

RUN npm install

RUN npm run build

CMD ["npm", "start"]

EXPOSE 3000
```

Build the Docker image:

```
docker build -t my-app .
```

Run the Docker container:

```
docker run -p 3000:3000 my-app
```

App is now running at `http://localhost:3000`

Conclusion

In this lab, we explored different deployment tools. Platforms like **Vercel** and **Netlify** simplify frontend deployment through Git integration and automated builds. Tools like **Docker** provide full control over backend and full-stack deployments, ensuring consistency across environments. Understanding these tools is crucial for building and launching real-world applications.