

DAY 6 - PREPARATION AND STAGING ENVIRONMENT SETUP DEPLOYMENT

Deployment Instructions:

Step 1: Prepare Your GitHub Repository

1. Initialize Your Repository:

- Create a new GitHub repository or use an existing one.
- Ensure your Next.js project is pushed to the repository.

2. Verify Project Setup:

- Ensure your Next.js project has a valid package.json file.
- Check for a .gitignore file with appropriate exclusions (e.g., node_modules, env).

3. Environment Variables:

- Add a .env file for local development, but do not push it to GitHub. Use Vercel's environment variables feature for secure storage.

Step 2: Link GitHub to Vercel

1. Sign in to Vercel:

- Go to [Vercel](#) and log in with your GitHub account.

2. Create a New Project:

- Click on "New Project" in the Vercel dashboard.
- Select your GitHub repository from the list.
- If your repository isn't listed, click on "Import Git Repository" and provide the URL.

3. Configure Build Settings:

- Vercel will auto-detect your project as a Next.js application. If not:
 - Set Framework Preset to Next.js.
- Use the default build and output settings:
 - Build Command: `npm run build`
 - Output Directory: `next`.

Step 3: Add Environment Variables

1. Define Variables:

- In the Vercel dashboard, go to the Settings tab of your project.

- Under Environment Variables, add any necessary variables (e.g., API_KEY, NEXT_PUBLIC_BASE_URL).

2. Set Variable Scopes:

- Add variables for the appropriate environment:
 - Production for the live deployment.
 - Preview for preview builds.
 - Development for local development.

Step 4: Trigger Deployment

1. Automatic Deployment:

- Vercel automatically deploys your application whenever you push changes to the connected GitHub branch (e.g., main or production).

2. Preview Deployments:

- Each pull request will generate a unique preview URL for testing.

3. Manual Deployment (Optional):

- In the Vercel dashboard, click "Deploy" for manual deployments if needed.

Step 5: Test and Verify Deployment

1. Check Live Deployment:

- Vercel provides a domain for your project (e.g., your-project.vercel.app).
- Visit the URL to verify the deployment.

2. Custom Domain Setup:

- Add a custom domain in the Domains section of the Vercel dashboard if required.

3. Monitor Logs:

- Check the Logs section in the Vercel dashboard for any errors during build or runtime.

Performance testing results (GTmetrix):

- Performance: Overall performance score based on Lighthouse benchmarks.

Obtained value: 92%.

- Structure: Measures adherence to web development best practices.

Obtained value: 95%

- Cumulative Layout Shift: Measures the visual stability of page elements.

Obtained value: 0.18

- Largest Contentful Paint: Time taken to render the largest visible content on the page.

Obtained value: 641ms

Speed Visualization:

The page demonstrates excellent loading performance with a **TTFB of 265ms**, **First Contentful Paint at 395ms**, and **Time to Interactive at 590ms**, though the **Fully Loaded Time of 6.2s** suggests potential optimizations for non-critical resource loading.