DAY -6

Staging Environment Setup and Deployment for My Website [CLOTHING & BAGS]

MADE BY ZUNAIRA HUSSAIN

Staging Environment

A staging environment is a pre-production setup used to test the website in conditions that closely mimic the live environment. It allows developers to:
Identify bugs and issues.
Test responsiveness and functionality.
Evaluate performance under near-live conditions.

Step-by-Step Process

GitHub Repository Setup

Creating the Repository:
- Create a new repository on GitHub named appropriately
- Organize the project files systematically with clear folder structures for components
. Branch Management:
- Use the `main` branch for production.
- Create a `staging` branch for pre-deployment testing.
Version Control:
- Commit changes with meaningful messages (e.g., 'Added responsive navbar').
- Use pull requests for merging `staging` to `main` after testing.

Vercel Deployment Setup

Connecting GitHub Repository:

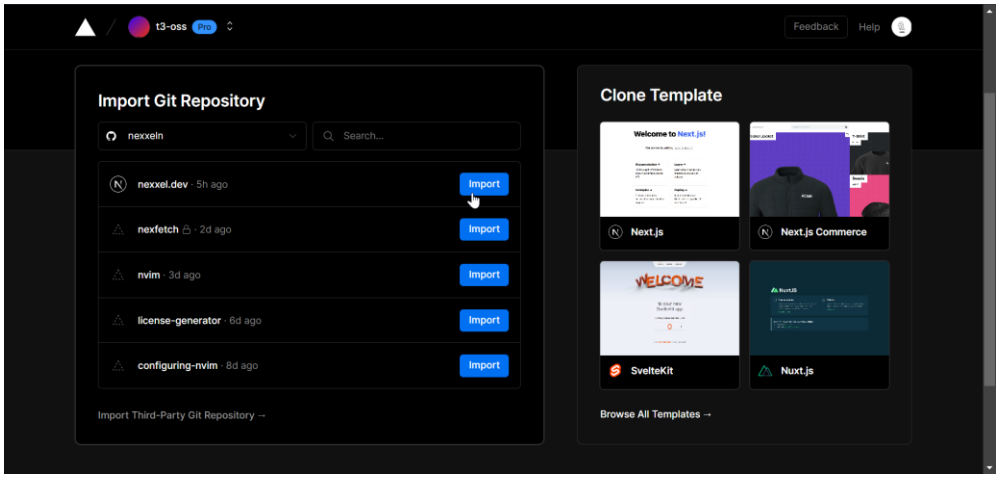
- Log in to Vercel and create a new project.
- Import the GitHub repository and

Environment Variable Configuration:

- Navigate to the Vercel dashboard and go to the 'Environment Variables' section.
- Add necessary variables such as API keys and database URLs.

Branch Settings:

- Configure Vercel to automatically deploy the `staging` branch for previews and the `main` branch for production



Performance Optimization

Google Page Speed Insights:

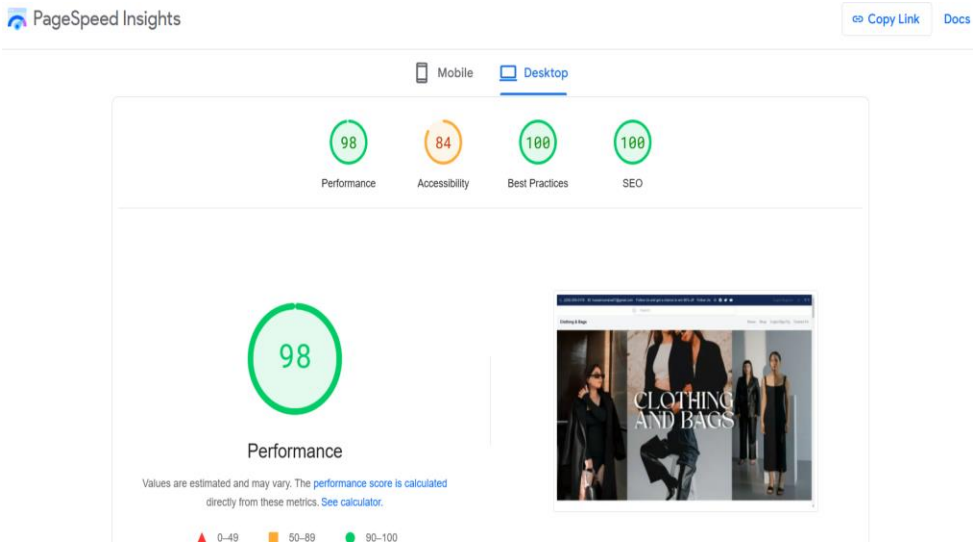
Test the staging site on Google PageSpeed Insights. Achieved:

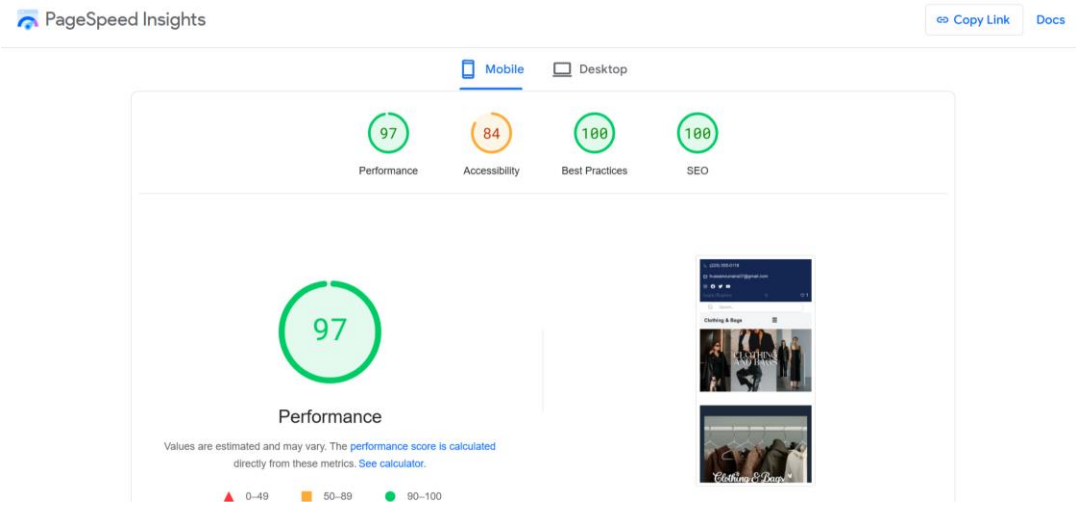
Performance: 97%

Best Practices: 98%

Lighthouse Audits:

- Optimize images, code splitting, and lazy loading for faster load times.





Final Deployment to Production

Merging Branches:
- After successful testing, merge the `staging` branch into the `main` branch on GitHub.

Production Deployment on Vercel:
- Vercel automatically triggers a deployment for the `main` branch.

Post-Deployment Monitoring:
- Integrate tools like Google Analytics and Vercel Analytics to monitor traffic and performance.

Tools and Technologies Used

- React: For building the user interface.
- Next.js: For server-side rendering and dynamic routing.
- GitHub: For version control and collaboration.
- Vercel: For hosting and deployment.
- Chrome DevTools: For debugging and testing responsiveness.
- Google PageSpeed Insights: For performance evaluation.

Next Steps

- Monitor user feedback to further enhance the site.
- Explore new opportunities in e-commerce and frontend development.
- Continue learning and applying modern web development practices.

MADE BY ZUNAIRA HUSSAIN