

GIAIC: HACKATHON

TASK: 6

Tulos - Deployment Preparation and Staging Environment Setup:

1. Staging Environment Setup:

Staging Server Configuration: Set up a staging environment using platforms like Vercel or Netlify, ensuring it mirrors the production environment. All features, such as product listings, filtering, and cart functionality, should work as they would in production.

Staging APIs & Database: Point the staging environment to a separate database and APIs to ensure that the production database and services remain unaffected by staging changes.

2. Deployment to Staging:

Deploy Tulos to the staging environment, ensuring all code updates are pushed to this environment for final testing.

Set up environment-specific variables such as API keys and backend URLs in the staging environment.

3. Testing in Staging Environment:

Functional Testing: Test all major features—product browsing, filtering by size, color, and category, cart system, user registration, and checkout.

Cross-Browser Testing: Test how Tulos looks and behaves in different browsers (Chrome, Firefox, Safari) to ensure compatibility.

Error Handling: Check error handling for incorrect user actions (e.g., empty cart during checkout, failed payment transactions, and missing product data).

API Testing: Ensure that all API requests and responses are working properly, including product data retrieval and user management (login, registration).

4. Performance Testing:

Load Testing: Simulate simultaneous users browsing through Tulos' clothing catalog and performing checkout to gauge the platform's performance.

Image Optimization: Test if product images are loading efficiently using lazy loading, especially on slower connections. Use Lighthouse to analyze the page's performance.

5. CI/CD Pipeline Setup:

Set up CI/CD tools to automatically test and deploy code to the staging environment whenever new commits are made.

Integrate tools like GitHub Actions for automated deployment and testing processes.

6. Documentation:

Include a detailed README in your GitHub repository, documenting the deployment and staging setup. This should cover everything from environment configurations to deployment steps.

7. Production Environment:

After final tests in staging, move the application to production. Ensure all environment-specific variables, such as API keys and third-party services, are properly configured for production.