Day 6 Assignment: Deployment Preparation and Staging Environment Setup

---

Objective:

The purpose of this assignment is to deploy the e-commerce website to a staging environment, test its functionality, and ensure that all the setup is complete for a seamless deployment to production. This task involves configuring environment variables, preparing for deployment, and generating a report with testing results.

---

1. Hosting Platform: Vercel

For the deployment of my e-commerce website, I used **Vercel** as the hosting platform due to its smooth integration with **Next.js** and its excellent support for static and serverless deployments. Below is a summary of the deployment process:

- **Deployed to Vercel**: The website was successfully deployed to **Vercel** with automatic **GitHub integration**, allowing real-time updates with every push to the repository.
- **GitHub Integration**: My GitHub repository was linked to **Vercel**, ensuring continuous deployment from the staging branch.

---

2. Environment Variables Configuration:

I configured the environment variables necessary for the project, including **API keys**, **database connections**, and **Sanity CMS** credentials, to maintain the security and integrity of sensitive information. Below is the content of the `.env.local` file:

```
# Sanity CMS Configuration
NEXT_PUBLIC_SANITY_PROJECT_ID=your_sanity_project_id
NEXT_PUBLIC_SANITY_DATASET=production
NEXT_PUBLIC_API_URL=https://api.yourdomain.com
SANITY_API_KEY=your_sanity_api_key

# Clerk Authentication
CLERK_FRONTEND_API=your_clerk_frontend_api
CLERK_API_KEY=your_clerk_api_key
CLERK_API_URL=https://api.clerk.dev

# Stripe Payment Gateway
STRIPE_API_KEY=your_stripe_api_key
NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=your_stripe_publishable_key

# Shipment API (Shipping & Tracking)
SHIPMENT_API_KEY=your_shipment_api_key
```

```
NEXT_PUBLIC_SHIPMENT_API_URL=https://api.yourshipmentapi.com
```

- **Sanity CMS Variables**: The environment variables ensure the CMS and other external APIs are securely connected.
- **Vercel Dashboard**: These variables were securely added to the Vercel dashboard for seamless deployment.

---

3. Deployment to Staging:

The deployment to **Vercel** was carried out smoothly, and the website went live on the staging URL within minutes. The following steps were performed:

- **Push Latest Changes**: Pushed the latest changes to the GitHub repository, which triggered the deployment on Vercel.
- **HTTPS Enabled**: Ensured that HTTPS was enabled for the deployed site to ensure secure browsing.
- **No Errors**: Verified that the deployment process had no errors, and the website was fully functional.

---

4. Testing the Staging Environment:

To ensure everything is functioning properly, I performed several tests on the staging environment. Here's a summary of the testing results:

Functional Testing:

- **Product Listing**: Verified that all products are correctly displayed on the homepage and individual product pages.
- **Cart Operations**: Ensured the cart functionality works as expected (add/remove items).
- **Checkout Process**: Tested the checkout flow to confirm it's fully operational.

Performance Testing:

- Used **Google Lighthouse** to analyze the performance of the website:
  - **Performance Score**: 90/100
  - **Accessibility Score**: 95/100
  - **SEO Score**: 92/100
  - **Best Practices Score**: 98/100

Security Testing:

- Ensured **HTTPS** is enabled to secure user data.

- Verified that input fields are sanitized and protected against **XSS** attacks.
- Ensured all sensitive keys are securely managed using **environment variables**.

Test Case Report:

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
TC001	Validate product listing page	Open product page > Verify products	Products displayed correctly	Products displayed correctly	Passed	Low	-	No issues found
TC002	Test API error handling	Disconnect API > Refresh page	Show fallback UI with error message	Error message shown	Passed	Medium	-	Handled gracefully
TC003	Check cart functionality	Add product to cart > Verify cart contents	Cart updates with added product	Cart updates as expected	Passed	High	-	Works as expected
TC004	Ensure responsiveness on mobile	Resize browser window > Check layout	Layout adjusts properly to screen size	Responsive layout working as intended	Passed	Medium	-	Test successful
Authentication	Verify Clerk authentication flow	User should be authenticated successfully	User authenticated successfully	User authenticated successfully	Passed	High	-	Clerk authentication working
Stripe Payment Gateway	Test Stripe payment integration	Complete payment with valid card	Payment processed successfully	Payment successful	Passed	High	-	Stripe gateway works fine
Shipment with ShipEngine	Test shipment process	Create shipment order with valid address	Shipment processed successfully	Shipment processed	Passed	High	-	ShipEngine working correctly

---

5. Documentation Updates:

I updated the **README.md** file to provide clear instructions on how to set up the project, along with the deployment process and environment variable configuration. Here's a summary of the documentation:

- **Project Setup**: Instructions on how to clone and set up the project locally.
- **Deployment Instructions**: Detailed guide for deploying the project to **Vercel** and adding environment variables.
- **Testing Results**: A summary of the test cases and their respective results (as shown above).
- **Technologies Used**: A list of the main technologies used in the project:
    - **Next.js** for the front-end framework.
    - **Tailwind CSS** for styling.
    - **Sanity CMS** for content management.
    - **Stripe** for payments.

- **Clerk** for authentication.
- **Stripe for Shipment & Tracking**.