DAY 6

DEPLOYMENT PREPARATION AND STAGING ENVIRONMENT SETUP

Step 1: Hosting Platform Setup

Choose a Platform:

Vercel for quick deployment and easy integration with GitHub.

Step 2: Configure Environment Variables

Handling Secure API Keys, Database Credentials, and Sensitive Data Using .env Files

When developing web applications, it's essential to protect sensitive data (API keys, database credentials) by using .env files instead of hardcoding them in your source code.

Why Use .env Files?

- **Security:** Prevents accidental exposure.
- **Flexibility:** Allows different configurations for environments.
- **Portability:** Easy to share without exposing secrets.

Configuring Environment Variables in Hosting Platforms When deploying your project, set environment variables directly on the hosting platform.

Steps for Vercel:

- Go to **Settings > Environment Variables**
- Add the necessary keys (e.g., NEXT_PUBLIC_SANITY_PROJECT_ID, API_KEY)
- Redeploy the application

Best Practices for Secure Deployment:

- Never hardcode sensitive information in code.
- Regularly rotate keys.
- Use secrets management tools (AWS Secrets Manager, HashiCorp Vault).
- Restrict access to sensitive data.
- Monitor logs for security threats.

Step 3: Deploy to Staging

Deploying the staging environment allows you to test your application in a production-like setting before the final release. It helps identify bugs, performance issues, and ensures everything works as expected before going live.

Step 4: Staging Environment Testing

This report outlines the testing conducted on the e-commerce website to ensure its functionality, performance, and security.

1. Functional Testing: Functional testing was performed to verify that all website features work correctly. This included checking navigation, forms, and user interactions to ensure they function as expected.

Key Focus Areas:

- Navigation links and buttons
- Product listing and search functionality
- Cart operations (adding/removing items)
- Form validation and submission
- Responsive design across devices

2. Performance Testing: Performance testing was conducted to evaluate the website's speed and responsiveness under different conditions. The goal was to ensure a smooth user experience.

Key Focus Areas:

- Page load speed analysis using Lighthouse
- Handling multiple user requests
- Image and asset optimization

3. Security Testing: Security testing was done to identify vulnerabilities and protect sensitive data. This included checking for secure data handling and protection against common threats.

Key Focus Areas:

- Data encryption and secure storage
- Protection against web attacks (e.g., SQL injection, XSS)
- HTTPS implementation and secure API communications