DAY 6 - DEPLOYMENT PREPARATION AND STAGING ENVIRONMENT SETUP -[General Ecommerce]

NAME: Mahnoor Ansari

Deployment Strategy Planning:

- Analyze the requirements and determine the best deployment environment (e.g., cloud, on-premises).
- Set up the necessary infrastructure, such as servers or hosting services.
- Test the application in a staging environment to ensure everything works as expected.
- Plan for backup and rollback procedures in case of deployment issues.
- Deploy the application in the production environment and monitor its performance

.

Environment Variable Configuration:

- Use `.env` files to store API keys, database credentials, and other sensitive information securely.
- Avoid hardcoding sensitive data directly into your code.
- Configure environment variables in your hosting platform to securely access these values during deployment.
- Ensure `.env` files are excluded from version control using `.gitignore`.
- Open the hosting platform's dashboard and navigate to the environment variables section.
- Add your environment variables securely, such as API keys or database credentials, without exposing them in your code.

Example:

```
NEXT_PUBLIC_SANITY_PROJECT_ID=your_project_id
NEXT_PUBLIC_SANITY_DATASET=production
API_KEY=your_api_key
```

Staging Environment Testing:

1.Functional Testing:

- Functional testing ensures that every feature of the application works as expected.
- Testing user actions like clicking buttons, submitting forms, and navigating pages.
- The goal is to verify that the application meets all the requirements and functions properly.

2.Performance Testing:

- Performance testing checks how well the application performs under different conditions.
- It measures speed, responsiveness, and stability during high user traffic or heavy data loads.
- The goal is to ensure the application runs smoothly and efficiently in all scenarios.

3.Security Testing:

- Security testing ensures that the application is protected from unauthorized access and vulnerabilities.
- It checks for weaknesses that could allow attackers to exploit or steal sensitive data.
- The goal is to confirm that the application is secure and meets privacy and security standards.

Test Case Report:

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
2	TC001	Validate product listing page	Open product page > Verify products	Products displayed correctly	Products displayed correctly	Passed	High	Handled gracefully
3	TC002	Test API error handling	Disconnect API > Refresh page	Show fallback UI with error message	Error message shown	Passed	Medium	Handled gracefully
4	TC003	Check cart functionality	Add product to cart > Verify cart contents	Cart updates with added product	Cart updates as expected	Passed	Medium	Works as expected
5	TC004	Ensure responsiveness on mobile	Resize browser window > Check layout	Layout adjusts properly to screen size	Responsive layout working as intended	Passed	Medium	Works as expected