DAY 6 - DEPLOYMENT PREPARATION AND STAGING ENVIRONMENT SETUP

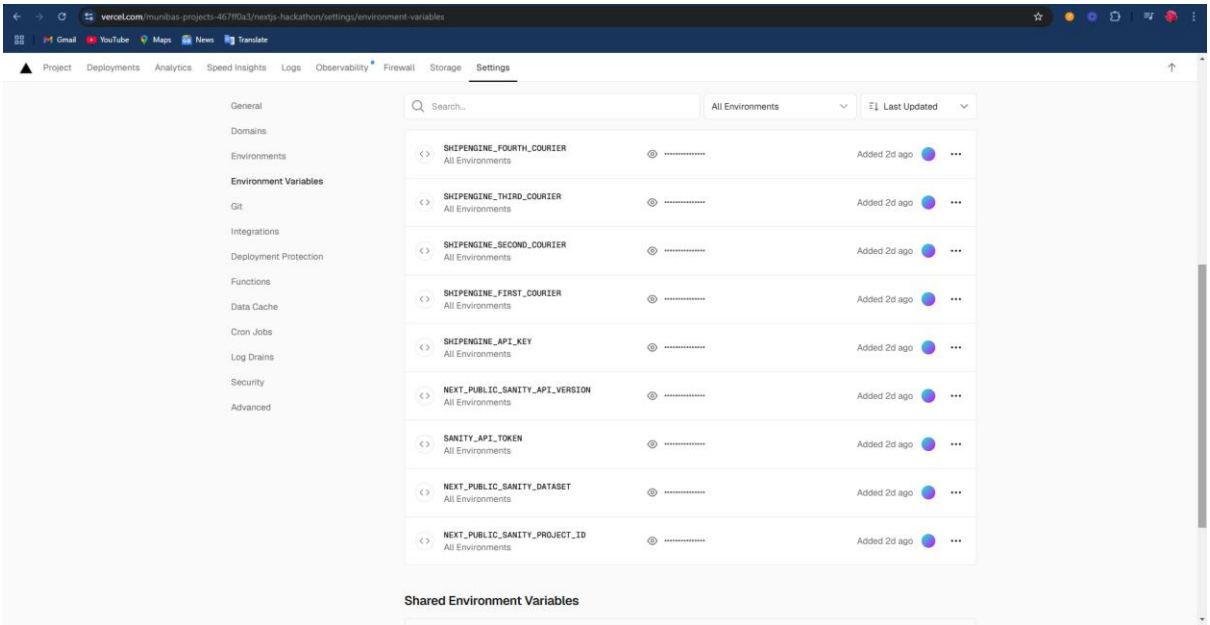
Step 1: Hosting Platform Setup

1. **Choose a Platform:**
   o Use platforms like Vercel for quick deployment.
2- **Connect Repository:**
   o Link your GitHub repository to the hosting platform.
   o Configure build settings and add necessary scripts for deployment.

Step 2: Configure Environment Variables

1. **Create a .env File:**
   Include sensitive variables like API keys and tokens.

```
src > sanity > lib > TS client.ts > [@] default
1   import { createClient } from 'next-sanity'
2
3
4
5   export const client = createClient({
6     projectId:process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
7     dataset:process.env.NEXT_PUBLIC_SANITY_DATASET,
8     apiVersion:process.env.NEXT_PUBLIC_SANITY_API_VERSION,
9     token:process.env.SANITY_API_TOKEN  ,
10    useCdn: true, // Set to false if statically generating pages, using ISR or tag-based revalidation
11  })
12
13  export default client
```



Step 3: Deploy to Staging

1. **Deploy Application:**

- Deploy the application to a staging environment through the hosting platform.
- Verify basic functionality in the staging environment.

Step 4: Staging Environment Testing

1. **Testing Types:**
   o Functional Testing: Verify all features, such as product listing, search, and cart operations.
   o Performance Testing: Use Lighthouse or GTmetrix to analyze speed and responsiveness.
   o Security Testing: Validate input fields, HTTPS usage, and secure API communications.
2. **Test Case Reporting:**
   o Document all test cases in a CSV file with fields like Test Case ID, Description, Steps, Expected Result, Actual Result, Status, and Remarks.
   For example:

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	Low	-	No issues found
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	High	-	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	-	Test successful
6	TC005	Check pagination	Product detail>next/prev button	Navigate to next/prev page	Pagination works and navigate to next/prev page	passed	medium	-	Test successful
7	TC006	Category filter	Product list>dropdown button “All products”	Category filter properly	Product displayed of selected category	passed	medium	-	Test successful

3. **Performance Testing:**
   - Submit a performance report generated by tools like Lighthouse in your GitHub repository.

Step 5: Documentation Updates

1. **Create README.md:**
   o Summarize all project activities, including deployment steps and test case results.
2. **Organize Project Files:**
   o Ensure all files from Days 1 to 6 are in a structured folder hierarchy.