

Day 6 - Deployment Preparation and Staging Environment Setup

Hosting Platform Setup

I chose **Vercel** as the hosting platform for deploying my marketplace application. Vercel was selected due to its ease of use, seamless integration with GitHub, and quick deployment capabilities. The following steps were taken to set up the hosting platform:









- 1. Connected GitHub Repository:**
 - a. Linked my GitHub repository to Vercel.
 - b. Configured build settings and added necessary scripts for deployment.
- 2. Environment Variables Configuration:**
 - a. Created a .env file to securely store sensitive variables such as API keys and tokens.
 - b. Uploaded the environment variables to Vercel's dashboard for secure deployment.
- 3. Deployed to Staging:**
 - a. Deployed the application to a staging environment using Vercel.
 - b. Validated that the build process completed without errors and the site loaded correctly.

(continue to next page...)

```
src > app > firebase > JS config.js > ...
1
2  import { initializeApp, getApps, getApp } from "firebase/app";
3  import { getAuth } from "firebase/auth";
4  import { getFirestore } from "firebase/firestore";
5
6  const firebaseConfig = {
7    apiKey: process.env.NEXT_PUBLIC_FIREBASE_API_KEY,
8    authDomain: process.env.NEXT_PUBLIC_FIREBASE_AUTH_DOMAIN,
9    projectId: process.env.NEXT_PUBLIC_FIREBASE_PROJECT_ID,
10   storageBucket: process.env.NEXT_PUBLIC_FIREBASE_STORAGE_BUCKET,
11   messagingSenderId: process.env.NEXT_PUBLIC_FIREBASE_MESSAGING_SENDER_ID,
12   appId: process.env.NEXT_PUBLIC_FIREBASE_APP_ID
13 };
14
15 const app = !getApps().length ? initializeApp(firebaseConfig) : getApp()
16
17 const auth = getAuth(app)
18
19 export const db = getFirestore(app)
20 export { app, auth }
```

```
src > sanity > TS env.ts > ...
1  export const apiVersion =
2    process.env.NEXT_PUBLIC_SANITY_API_VERSION || '2025-01-17'
3
4  export const dataset = assertValue(
5    process.env.NEXT_PUBLIC_SANITY_DATASET,
6    'Missing environment variable: NEXT_PUBLIC_SANITY_DATASET'
7  )
8
9  export const projectId = assertValue(
10   process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
11   'Missing environment variable: NEXT_PUBLIC_SANITY_PROJECT_ID'
12 )
13
14 function assertValue<T>(v: T | undefined, errorMessage: string): T {
15   if (v === undefined) {
16     throw new Error(errorMessage)
17   }
18
19   return v
20 }
21
```

(continue to next page...)

<>	NEXT_PUBLIC_SANITY_PROJECT_ID	👁️	Added 8h ago		...
<>	NEXT_PUBLIC_SANITY_DATASET	👁️	Added 8h ago		...
<>	NEXT_PUBLIC_FIREBASE_API_KEY	👁️	Added 8h ago		...
<>	NEXT_PUBLIC_FIREBASE_AUTH_DO...	👁️	Added 8h ago		...
<>	NEXT_PUBLIC_FIREBASE_PROJECT...	👁️	Added 8h ago		...
<>	NEXT_PUBLIC_FIREBASE_STORAGE...	👁️	Added 8h ago		...
<>	NEXT_PUBLIC_FIREBASE_MESSAGI...	👁️	Added 8h ago		...
<>	NEXT_PUBLIC_FIREBASE_APP_ID	👁️	Added 8h ago		...

Create Next App

hackathon-3-ahmed.vercel.app/allproducts


Avion

AB FREESTYLER LOGOUT


TablewareTablesCutleryCutleryChairsCutleryCeramics

All products


CategoryProduct typePriceBrandSorting by: Date added




The Poplar suede sofa
£980



Tropical Vibe
£550



Sleek Living
£300



Serene Seat
£350

Activate Windows
Go to Settings to activate Windows.

Staging Environment Testing

After deploying the application to the staging environment, I conducted thorough testing to ensure the application functions as expected in a production-like setting. The following types of testing were performed:

1. Functional Testing

- **Tools Used:** Cypress for workflow testing, Postman for API validation.
- **Test Cases:** Verified key functionalities such as product listing, cart operations, and API error handling.

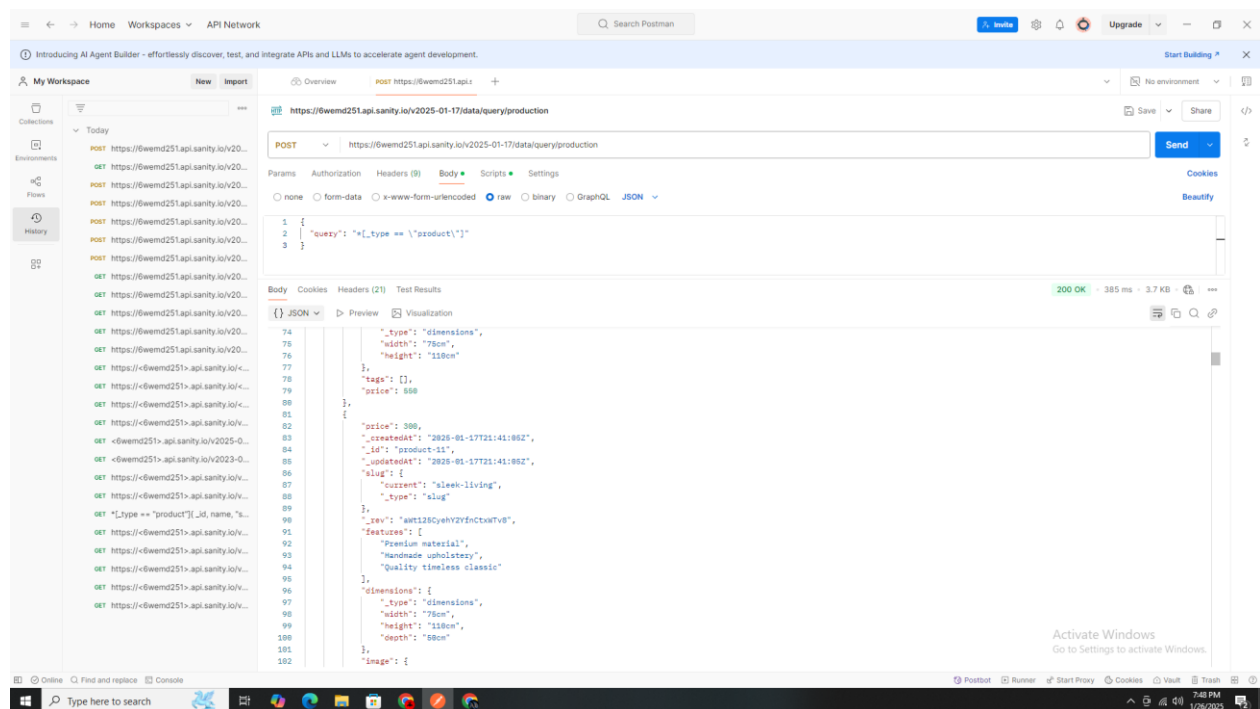
2. Performance Testing

- **Tools Used:** Lighthouse for analysing speed, responsiveness, and load times.
- **Results:** The application performed well, with acceptable load times and responsiveness.

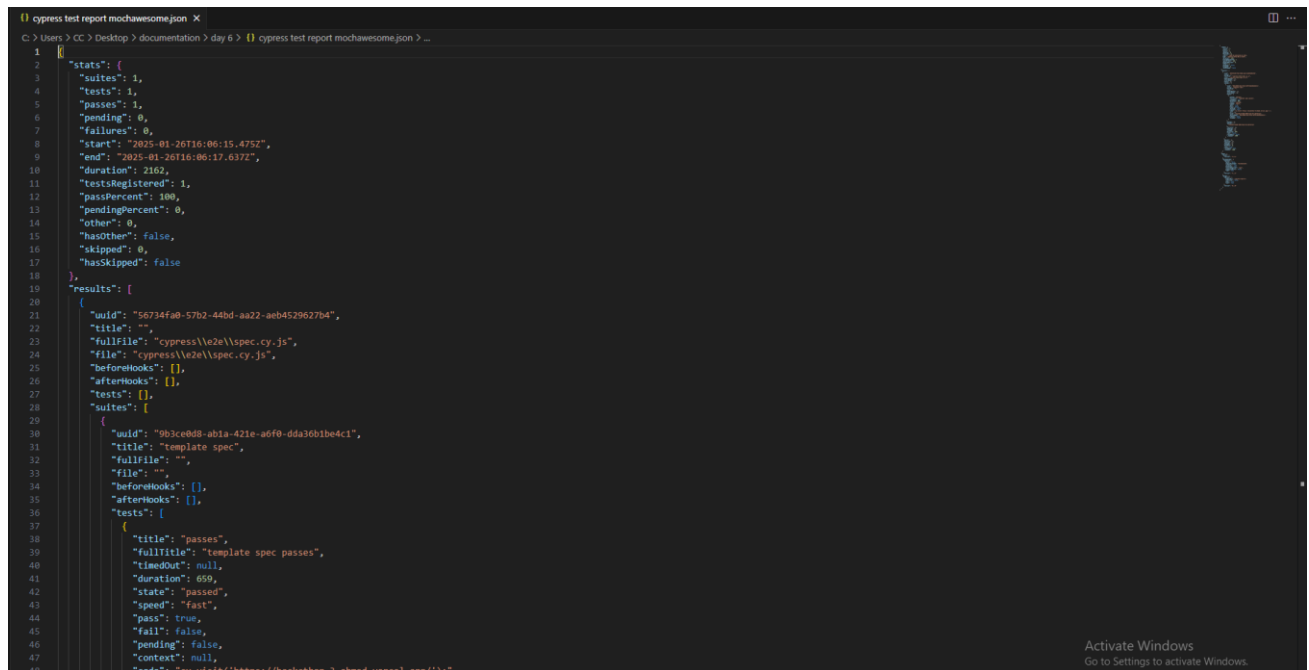
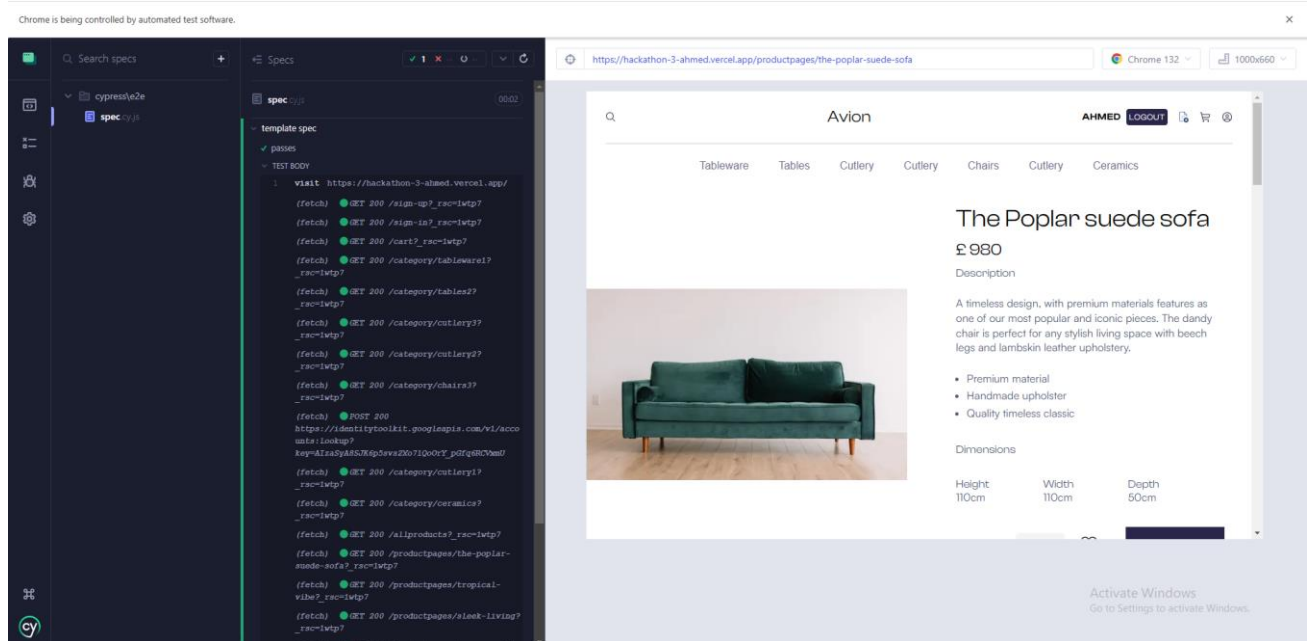
3. Security Testing

- **Validations:** Ensured HTTPS was enabled, input fields were secure, and sensitive data (e.g., API keys) were handled properly.

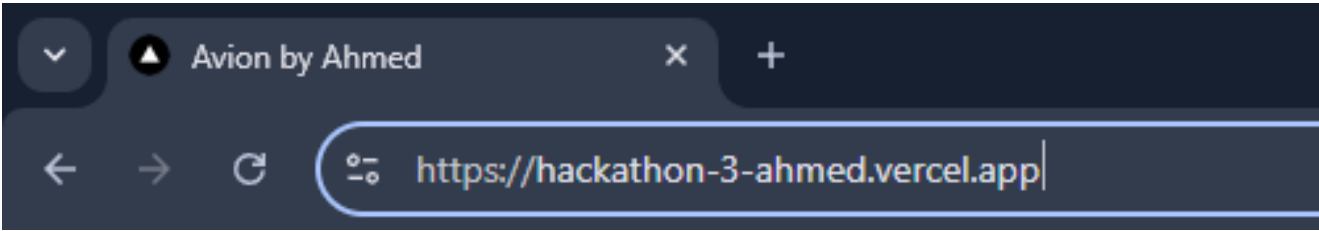
Api test



Cypress test



Https validation



Test Case Report

Below is the detailed test case report documenting the results of the staging environment testing:

	A	B	C	D	E	F	G
1	Test Case ID	Description	Steps	Expected Result	Actual Result	Status	Remarks
2	TC001	Validate product listing	Open product page > Verify products	Products displayed	Products displayed	Passed	No issues found
3	TC002	Test API error handling	Disconnect API > Refresh page	Show fallback message	Fallback message shown	Passed	Handled gracefully
4	TC003	Check cart functionality	Add item to cart > Verify cart	Cart updates correctly	Cart updates correctly	Passed	Works as expected
5	TC004	Test form validation	Submit form with empty fields	Display error message	Error message displayed	Failed	Missing validation check
6	TC005	Verify HTTPS connection	Open site > Check HTTPS status	HTTPS enabled	HTTPS enabled	Passed	Secure connection
7	TC006	Validate login functionality	Enter valid credentials > Submit	Login successful	Login successful	Passed	No issues found
8	TC007	Validate logout functionality	Click logout button	Redirect to homepage > Session terminated	Redirect successful	Passed	Session cleared successfully

Documentation Updates

1. README.md File:

- a. Created a README.md file summarizing all six days of activities, including deployment steps, test results, and project structure.
- b. Provided a clear folder hierarchy in the GitHub repository (e.g., documents/, src/, public/).

2. GitHub Repository:

- a. Organized all project files and documents in a structured manner.
- b. Included the following:
 - i. Test case report in CSV format.
 - ii. Performance testing results generated by Lighthouse.

Expected Output

1. Staging Environment:

- a. The application is fully deployed to the staging environment on Vercel.
- b. Environment variables are securely configured.

2. Test Case and Performance Reports:

- a. All test cases (passed or failed) are documented in a CSV file.
- b. Performance testing results are included in the GitHub repository.

3. GitHub Repository:

- a. All project files and documentation are organized and accessible.
- b. A professional README.md file summarizes the project activities and results.

Checklist for Day 6

- Deployment Preparation
- Staging Environment Testing
- Documentation
- Form Submission
- Final Review