Day 6

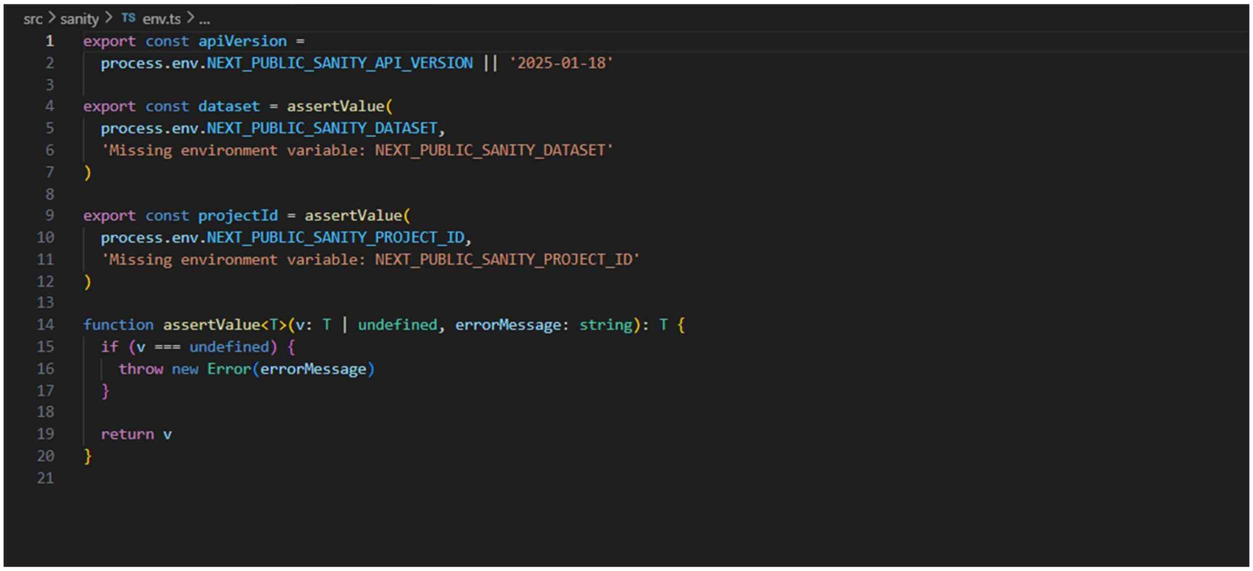
Deployment Preparation and Staging Environment Setup

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

Sanity Configuration (TypeScript)



```
export const apiVersion =
  process.env.NEXT_PUBLIC_SANITY_API_VERSION || '2025-01-18'

export const dataset = assertValue(
  process.env.NEXT_PUBLIC_SANITY_DATASET,
  'Missing environment variable: NEXT_PUBLIC_SANITY_DATASET'
)

export const projectId = assertValue(
  process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  'Missing environment variable: NEXT_PUBLIC_SANITY_PROJECT_ID'
)

function assertValue<T>(v: T | undefined, errorMessage: string): T {
  if (v === undefined) {
    throw new Error(errorMessage)
  }

  return v
}

```

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.

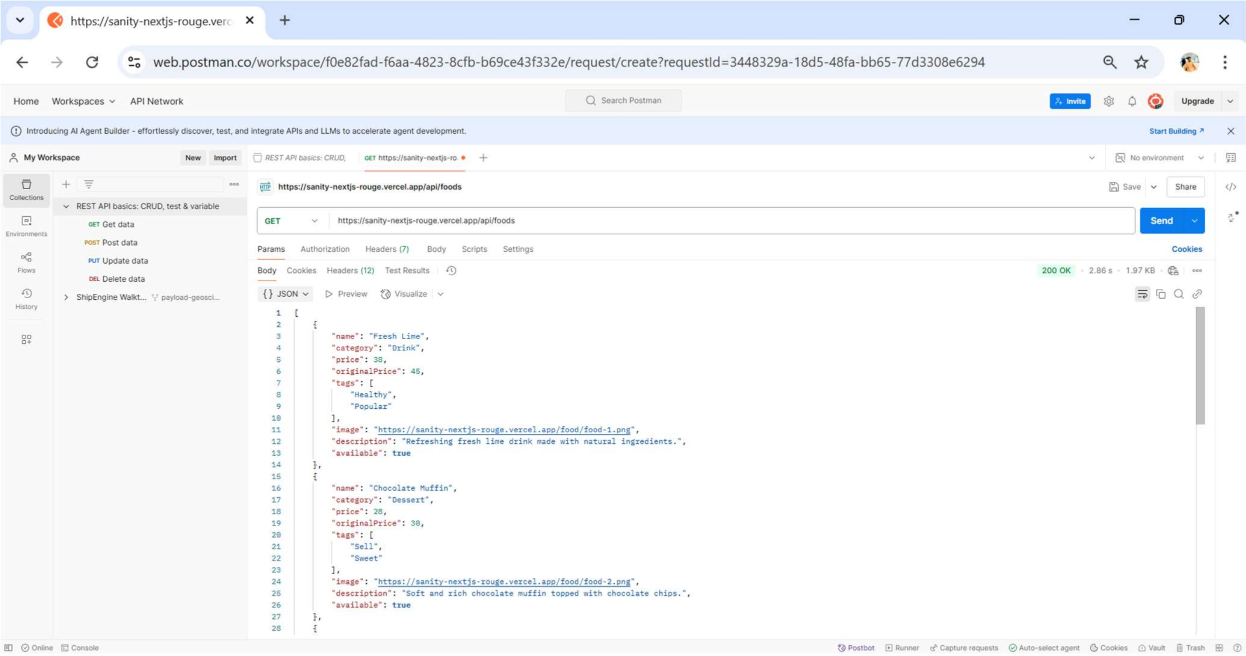
2. Performance Testing

- **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



https://sanity-nextjs-rouge.vercel.app/api/foods

GET https://sanity-nextjs-rouge.vercel.app/api/foods

200 OK · 2.86 s · 1.97 KB

```
[
    {
        "name": "Fresh Lime",
        "category": "Drink",
        "price": 38,
        "originalPrice": 45,
        "tags": [
            "Healthy",
            "Popular"
        ],
        "image": "https://sanity-nextjs-rouge.vercel.app/food/food-1.png",
        "description": "Refreshing fresh lime drink made with natural ingredients.",
        "available": true
    },
    {
        "name": "Chocolate Muffin",
        "category": "Dessert",
        "price": 28,
        "originalPrice": 30,
        "tags": [
            "Sell",
            "Sweet"
        ],
        "image": "https://sanity-nextjs-rouge.vercel.app/food/food-2.png",
        "description": "Soft and rich chocolate muffin topped with chocolate chips.",
        "available": true
    },
    {
```

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.

Checklist for Day 6

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