

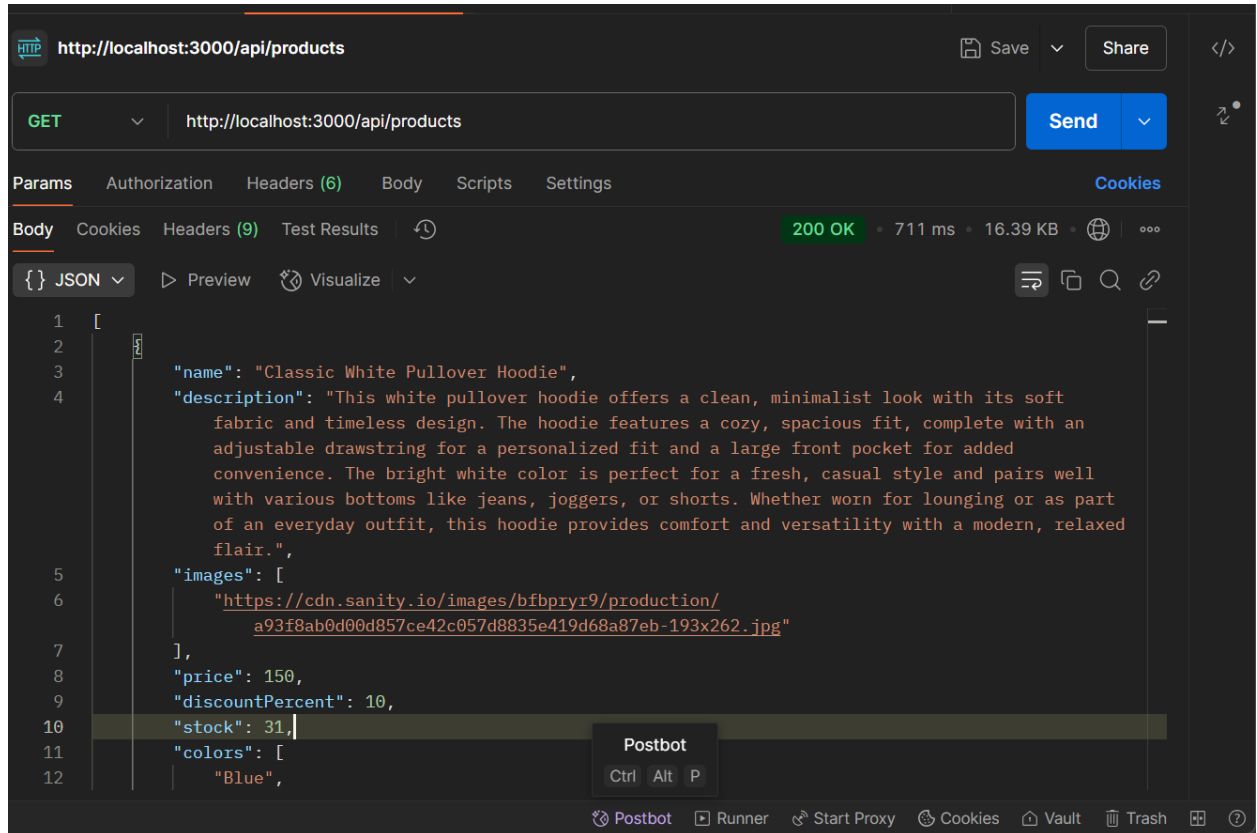
Day 5- Testing and Backend Refinement- Layers

Contents

1. Functional Deliverables	3
• Screenshots:	3
2. Testing Overview	3
• Performance Testing:	3
• Integration Testing:	3
3. Testing Tools	4
• Lighthouse	4
• Postman	4
4. Challenges Faced During Testing	4
• Performance Errors	4
• Data Fetching Errors:	4
• Params Promise Error:	4
5. Solutions to the Challenges	4
• Performance Optimization	4
• Data Fetching Adjustments:	4
• Handling Params as a Promise:	4
6. Backend Refinements	4
• Data Validation and Security Enhancements:	5
7. Future Plans for Testing and Backend Improvements	5
• Automated Testing:	5
• New API Endpoints	5
• Enhanced Security Measures	5
• Database Query Optimization:	5

1. Functional Deliverables

- Screenshots:



2. Testing Overview

- **Performance Testing:** Conducted to evaluate the speed, responsiveness, and stability of the application under different conditions.
- **Integration Testing:** Performed to ensure that different components and APIs work seamlessly together.

3. Testing Tools

- **Lighthouse:** Used for performance testing to analyze page load speed, accessibility, SEO, and more.
 - **Postman:** Utilized for API testing to verify that API endpoints return the correct responses and handle errors appropriately.
-

4. Challenges Faced During Testing

- **Performance Errors:** Encountered issues related to slow page load times and inefficient resource usage.
 - **Data Fetching Errors:** Faced difficulties in fetching data correctly from APIs.
 - **Params Promise Error:** Encountered errors while handling dynamic routing with parameters.
-

5. Solutions to the Challenges

- **Performance Optimization:** Improved code efficiency and resource management to enhance performance.
 - **Data Fetching Adjustments:** Refined data fetching logic to ensure accurate API responses.
 - **Handling Params as a Promise:** Resolved dynamic routing issues by treating params as a promise.
-

6. Backend Refinements

- **API Performance Improvements:** Optimized API endpoints to reduce response times and improve efficiency.

- **Data Validation and Security Enhancements:** Strengthened data validation processes and implemented additional security measures to protect user data.
-

7. Future Plans for Testing and Backend Improvements

- **Automated Testing:** Plan to implement automated testing to streamline the testing process and improve test coverage.
 - **New API Endpoints:** Intend to add new API functionalities to support additional features.
 - **Enhanced Security Measures:** Focus on further enhancing security protocols to safeguard data.
 - **Database Query Optimization:** Aim to optimize database queries for better performance.
-