MUHAMMAD UMER

304953

(Student Leader Friday 9-12)

Day 4 – Technical Report – General Ecommerce Nike Website

Technical Report: Development and Integration Summary

Steps Taken to Build and Integrate Components

1. Add-to-Cart and Wishlist Features:
- Used context API to manage global data easily.
- Created reusable components for smooth user interaction.
- Added inventory management to update product quantities in real time.

2. Shipment and Reviews Section:
- Built and added a shipment tracking feature for user convenience.
- Integrated a reviews section to show customer feedback fetched from the backend.

3. Search, Filter, and Pagination:
- Added a search bar with advanced filters for accurate results.
- Used pagination to improve navigation and performance.

4. Chatbot Integration:
- Added a chatbot for real-time user support using modern libraries and APIs.
- Designed it to be easy to use and responsive on all devices.

5. Using Library Components:
- Used `shadcn`, `Aeternity UI`, and `Magical UI` to create consistent and attractive designs.
- These libraries helped save time and maintain a professional look.

6. Sanity for Data Management:
- Used Sanity for handling backend data efficiently.
- Configured flexible schemas to make data handling easier and scalable.

Challenges Faced and Solutions Implemented

1. Managing Global Data:
- Problem: Keeping track of data across multiple components.
- Solution: Used context API for centralized and easy-to-manage global data.

2. Inventory Management:
- Problem: Keeping inventory updated during multiple user actions.
- Solution: Added real-time updates and validation to avoid errors.

3. Library Compatibility:
- Problem: Ensuring different libraries worked well together.
- Solution: Tested thoroughly and fixed any conflicts during integration.

4. Performance Issues:
- Problem: Handling large amounts of data for search and filters.
- Solution: Used lazy loading and optimized API calls for better speed.

5. Chatbot Features:
- Problem: Making the chatbot work smoothly.
- Solution: Integrated APIs with backups to ensure reliability.

Best Practices Followed During Development

1. Reusable Code:
- Designed modular components for easy reuse.

2. Version Control:
- Used Git for regular updates and organized code management.

3. Accessibility:
- Ensured all components followed accessibility guidelines like keyboard navigation and ARIA roles.

4. Responsive Design:
- Made sure the designs worked well on all screen sizes.

5. Testing and Debugging:
- Performed detailed testing to catch and fix issues early.

6. Clear Documentation:
- Kept well-written documentation for easy reference and future updates.