Marketplace Development Journey: A Full-Stack Hackathon Experience

Day 1

Marketplace Types

Work: Explored General E-Commerce, Q-Commerce & Rental E-Commerce
Benefit: Understood unique features & purposes of each model

Business Goals

Work: Defined objectives by identifying problems, audience & key offerings
Benefit: Improved problem-solving & strategic thinking

Key Business Jargon

Work: Learned terms like SKU, CAC, LTV, GMV, conversion rate
Benefit: Enhanced knowledge of marketplace metrics for success

Data Schema Basics

Work: Designed schema with key entities (products, orders, customers, shipments, delivery zones)
Benefit: Developed understanding of database structure & entity relationships

Diagram Design

Work: Created a diagram showing entity interactions (orders, customers, products)
Benefit: Visualized workflows for better technical clarity

Documentation Practice

Work: Documented learnings & goals for future reference
Benefit: Improved documentation & presentation skills

Final Outcome: Gained foundational skills in marketplace models, goal-setting & database design, setting a strong base for the next hackathon steps.

Day 2

Technical Requirements

Work: Planned Next.js and Tailwind CSS for the frontend, and Sanity CMS with APIs for the backend.
Benefit: Learned how to integrate frontend and backend and understood how APIs work.

System Architecture

Work: Created a diagram showing the interaction between the frontend, backend, and APIs.
Benefit: Gained a clear understanding of system data flow and component connections.

Workflows

Work: Designed key workflows like product browsing and order placement.
Benefit: Improved understanding of user journeys and technical steps.

API Requirements

Work: Defined endpoints like /products (GET) and /orders (POST).
Benefit: Learned how to design APIs and their role in frontend-backend communication.

Sanity CMS Schema

Work: Designed a product schema (fields: name, price, stock).
Benefit: Understood schema design and data modeling.

Collaboration

Work: Took peer feedback and used GitHub for version control.
Benefit: Gained experience in teamwork and document refinement.

Final Outcome: Improved technical skills and gained a strong understanding of frontend-backend integration, APIs, and Sanity CMS. Boosted confidence for the next steps.

Day 3

API Understanding

Work: Reviewed APIs, endpoints (/products, /categories) & data migration functionality
Benefit: Learned to interpret API docs, test endpoints & align data with requirements

Schema Validation and Customization

Work: Compared Sanity CMS schema with API data & adjusted field names, types & relationships
Benefit: Improved skills in schema validation & modification for smooth data migration

Data Migration Techniques

Work: Explored three methods: automated scripts, manual JSON/CSV upload & external API integration (Shopify, WooCommerce)
Benefit: Gained knowledge of efficient data migration, validation & large dataset management

API Integration in Next.js

Work: Integrated APIs in frontend with utility functions, rendered data in components & tested with Postman
Benefit: Learned to fetch, process & display API data with error handling in Next.js

Debugging and Error Handling

Work: Tested API endpoints using Postman & dev tools, logged errors & implemented handling mechanisms
Benefit: Developed debugging skills & created user-friendly error messages for reliability

Final Outcome: Gained expertise in API integration, schema validation & data migration with hands-on experience in Sanity CMS & Next.js. Improved debugging, error handling, strengthening marketplace development skills.

Day 4 Documentation Overview

Dynamic Component Development

Work: Created dynamic frontend components like product listings, category filters, and search bars using data fetched from Sanity CMS and APIs.
Benefit: Learned how to design scalable, reusable components that handle data dynamically, improving frontend development skills.

Modular and Reusable Design

Work: Developed modular components (e.g., Product Card, Wishlist, Cart) to ensure reusability and maintainability.
Benefit: Improved understanding of component-based architecture, making the codebase more efficient and easier to extend.

State Management

Work: Implemented state management techniques for features like cart, wishlist, and checkout flow using React state and context.
Benefit: Enhanced knowledge of managing application-wide states and syncing UI with backend data in real-time.

Responsive Design and UX/UI Practices

Work: Designed responsive layouts for components like product grids and detail pages, ensuring optimal viewing on different devices.
Benefit: Gained skills in creating user-friendly, visually appealing designs aligned with modern web development practices.

Search and Filtering Functionality

Work: Added a search bar to filter products by name or tags and category filters to refine product listings.
Benefit: Learned how to implement and optimize search and filter logic for better user experience.

Checkout Flow Development

Work: Built a multi-step checkout flow, including billing, shipping, and mock payment forms.
Benefit: Acquired experience in designing structured workflows and integrating form handling in web applications.

Real-World Workflows

Work: Followed professional practices, including dynamic routing in Next.js for product detail pages and debugging errors during API data rendering.
Benefit: Developed skills in replicating real-world scenarios and workflows, preparing for client-based projects.

Final Outcome: Learned to build scalable, responsive frontend components using Sanity CMS and Next.js. Improved expertise in modular design, state management, and UX/UI practices. Gained real-world experience in handling dynamic data and preparing for professional marketplace applications.

Day 5 Documentation Overview

Testing and Debugging

Work: Conducted functional and security testing using tools like Postman, Cypress, and React Testing Library.
Benefit: Learned to test API responses, handle errors, and ensure all features work as expected.

Error Handling and User Experience

Work: Implemented error handling using try-catch and fallback UI elements for API failures.
Benefit: Gained skills in providing clear error messages and enhancing user experience during failures.

Performance Optimization

Work: Optimized performance using tools like Lighthouse, reduced image sizes, and applied lazy loading.
Benefit: Improved website load times and responsiveness, enhancing overall performance.

Cross-Browser and Device Compatibility

Work: Tested website on multiple browsers and devices using BrowserStack.
Benefit: Ensured consistent user experience and functionality across different platforms.

Security Practices

Work: Applied input validation, sanitized forms, and stored sensitive API keys securely.
Benefit: Enhanced security by preventing injection attacks and ensuring secure data transmission.

Professional Documentation

Work: Created a detailed testing report and documented all changes, fixes, and optimization steps.
Benefit: Gained expertise in writing professional documentation that meets industry standards.

Final Outcome: Gained hands-on experience in testing, error handling, performance optimization, and security, preparing the marketplace for real-world deployment.

Day 6

Staging Environment Setup

Work: Deployed the application to Vercel for staging.
Benefit: Learned to test the app in a production-like environment.

Environment Variables Configuration

Work: Configured and secured API keys using .env files.
Benefit: Gained experience in securing sensitive data for deployment.

Staging Environment Testing

Work: Conducted functional, performance, and security testing using Cypress and Lighthouse.
Benefit: Improved testing skills to ensure app functionality and performance.

Performance and Security Testing

Work: Ran tests for performance and security issues.
Benefit: Ensured app performs well and handles data securely.

Professional Documentation

Work: Created test case reports and organized project files on GitHub.
Benefit: Enhanced documentation skills for clear project structure.

Final Outcome:Gained experience in deploying, testing, and securing applications, preparing them for production deployment.

Day 7 Plan: Refining and Preparing for Live Deployment

1. Review and Improve Documentation

Go through all documentation from Day 1 to Day 6.
Refine and enhance explanations for clarity and professionalism.
Ensure the documentation is structured, concise, and aligned with industry standards.

2. Final Testing and Bug Fixes

Conduct end-to-end testing on all marketplace features.
Identify and fix any remaining bugs or inconsistencies.
Perform security audits to check for vulnerabilities.

3. Performance Optimization

Optimize API calls to reduce latency.
Implement caching strategies for frequently accessed data.
Ensure images and assets are fully optimized for web performance.

4. Deployment Readiness Check

Verify all environment variables are properly set.
Check that all third-party integrations (e.g., payment gateways, authentication) work correctly.
Test staging environment thoroughly before final deployment.

5. Live Deployment Preparation

Deploy the marketplace to a production environment.
Perform final testing on the live version.
Set up monitoring tools for uptime and performance tracking.

6. Post-Deployment Documentation & Finalization

Document the deployment process for future reference.
Create a user guide for maintaining and updating the marketplace.
Ensure all project files, reports, and summaries are well-organized.

Final Goal: Make the marketplace production-ready with a professional, real-world approach, ensuring smooth functionality, performance, and scalability.

Testing Report

Test Case ID	Test Case Name	Steps	Expected Result	Actual Result	Status	Remarks	Severity
TC001	Check Responsiveness	Open Edge, Chrome, Mobile	Fully Responsive On All Devices	Fully Responsive On All Devices	Passed	Passed	Minor
TC002	Check Dynamic Routing	Click On Route Links (Home, About, Product, Shop) On Navbar	Clicking On A Route Should Navigate To The Correct Page	Clicking On A Route Navigates To The Correct Page	Passed	Passed	Minor
TC003	Validate Product Listing Page	Open The Product Page And Verify Listed Products	All Products Should Be Displayed Correctly	All Products Are Displayed Correctly	Passed	Passed	Major
TC004	Validate Dynamic Routing (Product Details)	Click On A Product > Verify Full Details	Product Details Should Be Displayed Correctly	Product Details Are Displayed Correctly	Passed	Passed	Major
TC005	Check Cart Functionality	Click On "Add To Cart" > Verify Cart Content	Cart Should Display Correct Size, Quantity, And Product	Cart Displays Correct Size, Quantity, And Product	Passed	Passed	Minor
TC006	Check Cart Functionality (Quantity & Deletion)	Click On Button To Update Quantity Or Delete Item	Quantity Should Update Correctly, Item Should Be Deleted Correctly, Discount And Total Bill Should Be Correct	Quantity Updates Correctly, Item Deletes Correctly, Discount And Total Bill Are Correct	Passed	Passed	Critical
TC007	Check Filtering	Click On Shop Route And Check Filtering Options	Filters (Color, Size, Category, Price) Should Work Correctly	Filters Are Working But Need Improvement	Passed	Needs Improvement	Major

Test Case ID	**Test Case Name**	**Steps**	**Expected Result**	**Actual Result**	**Status**	**Remarks**	**Severity**
TC008	Check Search Bar For Product	Click On Search Bar And Enter Product Name, Price, Or Category	The Searched Product Should Be Filtered Correctly	The Searched Product Is Filtered Correctly	Passed	Successful	Minor
TC009	Check Shipment Process	Click On Product > Add To Cart > Checkout > Enter Details > Verify Shipment	Shipment Status Should Be Correct, Label Should Be Generated, Delivery Details Should Be Correct	Shipment Status And Label Are Generated Correctly But Need Backend Improvements	Passed	Needs Improvement In Backend	Critical

THANK U

Our 7-day hackathon at GAIAC was a valuable journey toward becoming full-stack developers. During this time, we learned essential skills for building a marketplace and deploying it live. From API integration to schema validation, testing, performance optimization, and final deployment—this experience provided us with real-world knowledge of how a marketplace works.

We gained hands-on experience in making the project production-ready, ensuring smooth functionality and performance. Special thanks to Sir Ameen Alam for guiding us throughout this process, as well as to Sir Anas Seth and Sir Ubaid for their invaluable teaching and making us capable of completing this project successfully.