GIAIC

Farida Bano

21 JA

Marketplace Development Journey (General E-Commerce)

On the first day, I started by exploring various marketplaces that fall under the category of general e-commerce. This helped me understand the core components and features that are typically seen across these platforms.

Next, I defined the objectives and goals for the project. This included understanding what I wanted to achieve with the marketplace and setting up clear targets to track progress. I also created PDF files documenting all the objectives and saved them in a folder for future reference.

I then focused on writing schema code for key models such as product.ts, order.ts, and timeZone.ts. I manually created the links schema in the studio and wrote the necessary queries to fetch and

migrate the data. All the necessary commands were followed carefully, and the data migration process was successfully completed.

After setting up the schemas, I moved on to creating dynamic routing for the marketplace. This allowed users to navigate seamlessly between pages such as product listings, categories, and other essential features. I also worked on adding the Add to Cart functionality and implemented the wishlist feature. There were many challenges along the way, but I cleared each one and gained a deeper understanding of how the system works.

In the final steps of the day, I pushed all the code to GitHub and deployed the marketplace to Vercel. I also used Lighthouse to test the performance of the site. The results were great, and I felt happy with how everything was coming together.

Though the project is still ongoing, I am confident that, insha'Allah (God willing), I will continue to improve and make the marketplace even better.

Day 1: Goals and Breakdown

Define technical goals and UX wireframes.
Conduct market research.
Allocate tasks and resources.

Day 2: Objective Setting

Define project scope and features (product listings, payments, user accounts).
Identify target audience and tech stack.
Set milestones and timeline.

Day 3: API Integration & Data Migration

Integrate payment, shipping, and other third-party APIs.
Migrate existing product/customer data.

Implement security protocols.

Day 4: Dynamic Frontend Development

Build reusable, responsive UI components.
Implement dynamic features (e.g., search, filters).
Optimize for mobile and fast performance.

Day 5: Testing & Backend Refinement

Perform unit testing and error handling.
Refine back-end integration.
Conduct load testing for performance.

Day 6: Deployment & Staging

Set up staging environment for final testing.
Configure version control and CI/CD pipelines.
Perform final tests before deployment.

Key Takeaways:

Automatic routing with Next.js.
Dynamic product pages based on product ID.
Reusable components for product display and UI consistency.
Responsive design for mobile and desktop views.

Header and Footer:

- Automatic routing with Next.js.
- Dynamic product pages based on product ID.
- Reusable components for product display and UI consistency.
- Responsive design for mobile and desktop views.

Product.tsx

Key Takeaways:

- Product Grid: Displays a list of products in a responsive grid.
- Hover Effects: Shows additional options (Add to Cart, Share, Like, Compare) on hover.
- Product Information: Each product card includes an image, title, description, price, and optional badges.

- Responsive: Layout adapts based on screen size using Tailwind’s responsive utilities.
- "Show More" Button: Allows users to view more products by linking to the shop page.