

# DAY 6 - DEPLOYMENT PREPARATION AND STAGING ENVIRONMENT SETUP

MediaMart

## Overview

Ensure your Media Mart e-commerce store is deployment-ready by configuring a reliable staging environment. Test all features, integrations, and functionalities in a controlled setup to identify potential issues before going live. This step guarantees a seamless and efficient transition to production.

Day 1: CONCEPTUALIZATION AND MARKETPLACE DESIGN.

### KEY ACHIEVEMENTS:

- Defined the business vision and objectives for Media Mart's e-commerce platform.
- Conducted market research to identify target audience needs and preferences.
- Outlined core features, including product categories, payment methods, and user experience flows.
- Created a wireframe and preliminary design for the marketplace layout.
- Established a roadmap for development aligned with business goals and market trends.

Day2: TECHNICAL PLANNING.

- Finalized the use of **Next.js** for a high-performance, server-side rendered e-commerce platform.
- Integrated **Sanity API** as the content management backend for dynamic and scalable data handling.
- Adopted **Tailwind CSS** for efficient and responsive UI design with a utility-first approach.
- Mapped out the system architecture, ensuring smooth API communication and third-party integrations.
- Established a technical roadmap focused on optimizing the performance, scalability, and design of the e-commerce store.

## API REQUIREMENTS:

- User management
- Product manager
- Orders

## Day3: DATA MIGRATION.

### Key Achievements:

1. Consolidated and cleaned existing product, user, and order data for accurate migration.
2. Structured the data schema in **Sanity API** to align with the e-commerce store's requirements.
3. Successfully imported product catalogs, images, and metadata into **Sanity CMS** for seamless management.
4. Tested data integrity and resolved any inconsistencies post-migration.
5. Implemented automated processes for continuous data synchronization and updates.

## Day 4: Dynamic Frontend Components.

### KEY ACHIEVEMENTS:

- Product Listing Page with Dynamic Data
  - o Description: The product listing page dynamically displays products using a manually exported JSON dataset. Each product includes its image, name, price, and a brief description, presented in a clean and visually appealing layout to enhance user experience.
- 3. Working Category Filters, Search Bar, and Pagination
  - o Screenshots/Recordings:
    - o Demonstration of category filters.
    - o Search bar in action.
    - o Pagination functionality.
  - Description:
    - o Category Filters: Enables filtering of products based on categories like "Electronics,"

"Home Appliances," etc., allowing users to narrow down their preferences.

- o Search Bar: Offers real-time search functionality, dynamically o filtering products as users type keywords.
- o Pagination: Improves performance and usability by dividing the o product list into multiple pages, ensuring faster load times and easier navigation.

- Additional Features
  - o Related Products: Displays similar products on the product detail page, encouraging user engagement.
  - o User ProfileComponents: Includes a basic user profile section showcasing user details.

## Day 5 : TESTING AND BACKEND REFINEMENTS.

### Key Achievements:

1. Conducted comprehensive testing of the e-commerce platform, including functionality, performance, and usability.
2. Refined **Sanity API** configurations to optimize data flow and ensure content accuracy.
3. Tested **Next.js** server-side rendering and dynamic routing for seamless navigation and page load speed.
4. Improved backend API responses and ensured compatibility with frontend components styled using **Tailwind CSS**.
5. Fixed bugs, optimized code, and ensured the backend met scalability and security standards.

CSV content:

### # Testing Report Summary

This document provides an overview of the testing conducted for the application.

### ## Summary of Testing

The following test cases were executed to ensure the application's functionality, performance, and user experience:

#### #### Test Cases:

- \*\*TC001\*\*: Validate product listing displays all products correctly. (Status: Passed)
- \*\*TC002\*\*: Verify search functionality with valid and invalid inputs. (Status: Passed)
- \*\*TC003\*\*: Ensure cart operations work (add, update, remove items). (Status: Passed)
- \*\*TC004\*\*: Test cross-browser compatibility on Chrome, Firefox, and Edge. (Status: Passed)
- \*\*TC005\*\*: Check responsiveness on mobile and tablet devices. (Status: Passed)
- \*\*TC006\*\*: Validate product listing displays all products correctly. (Status: Passed)
- \*\*TC007\*\*: Verify search functionality with valid and invalid inputs. (Status: Passed)
- \*\*TC008\*\*: Ensure cart operations work (add, update, remove items). (Status: Passed)
- \*\*TC009\*\*: Test cross-browser compatibility on Chrome, Firefox, and Edge. (Status: Passed)
- \*\*TC0010\*\*: Check responsiveness on mobile and tablet devices. (Status: Passed)

#### CSV REPORT :

#### # Testing Report Summary

This document provides an overview of the testing conducted for the application.

### ## Summary of Testing

The following test cases were executed to ensure the application's functionality, performance, and user experience:

#### #### Test Cases:

- \*TC001\*: Validate product listing displays all products correctly. (Status: Passed)
- \*TC002\*: Verify search functionality with valid and invalid inputs. (Status: Passed)
- \*TC003\*: Ensure cart operations work (add, update, remove items). (Status: Passed)
- \*TC004\*: Test cross-browser compatibility on Chrome, Firefox, and Edge. (Status: Passed)
- \*TC005\*: Check responsiveness on mobile and tablet devices. (Status: Passed)
- \*TC006\*: Validate product listing displays all products correctly. (Status: Passed)
- \*TC007\*: Verify search functionality with valid and invalid inputs. (Status: Passed)
- \*TC008\*: Ensure cart operations work (add, update, remove items). (Status: Passed)
- \*TC009\*: Test cross-browser compatibility on Chrome, Firefox, and Edge. (Status: Passed)
- \*TC0010\*: Check responsiveness on mobile and tablet devices. (Status: Passed)

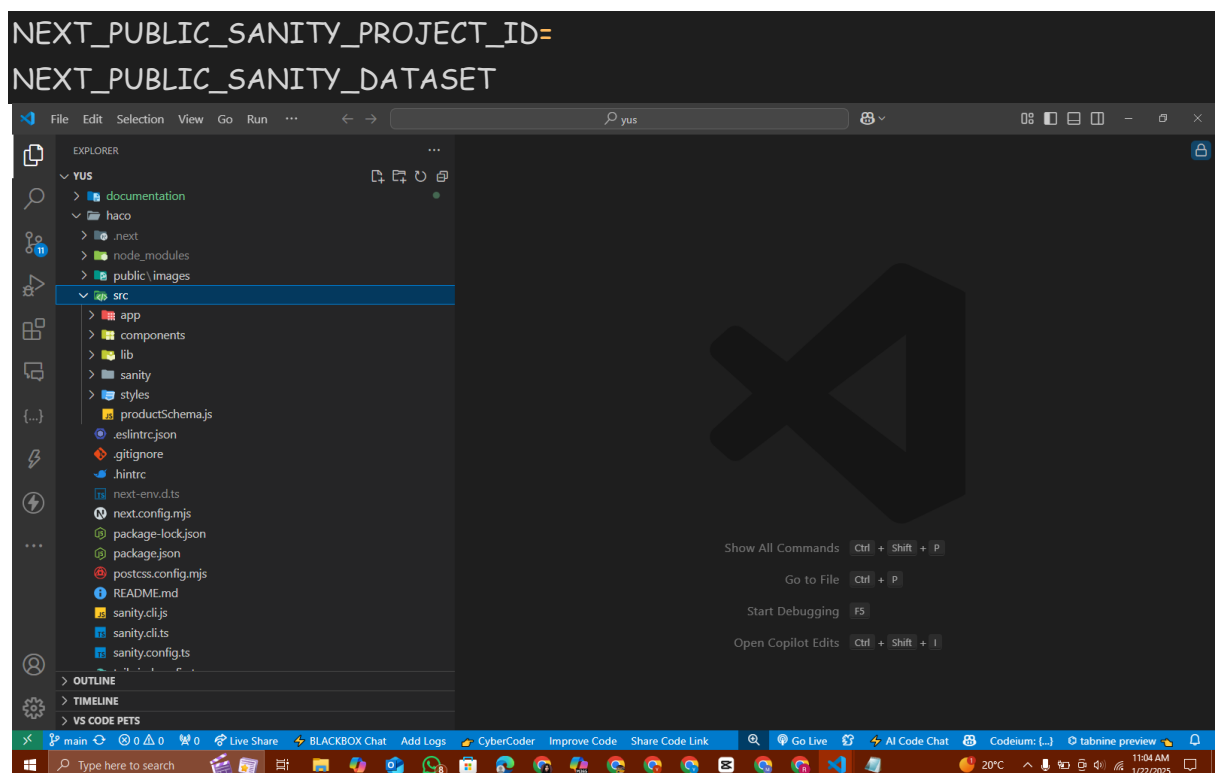
Day 6 - Deployment Preparation and Staging Environment Setup

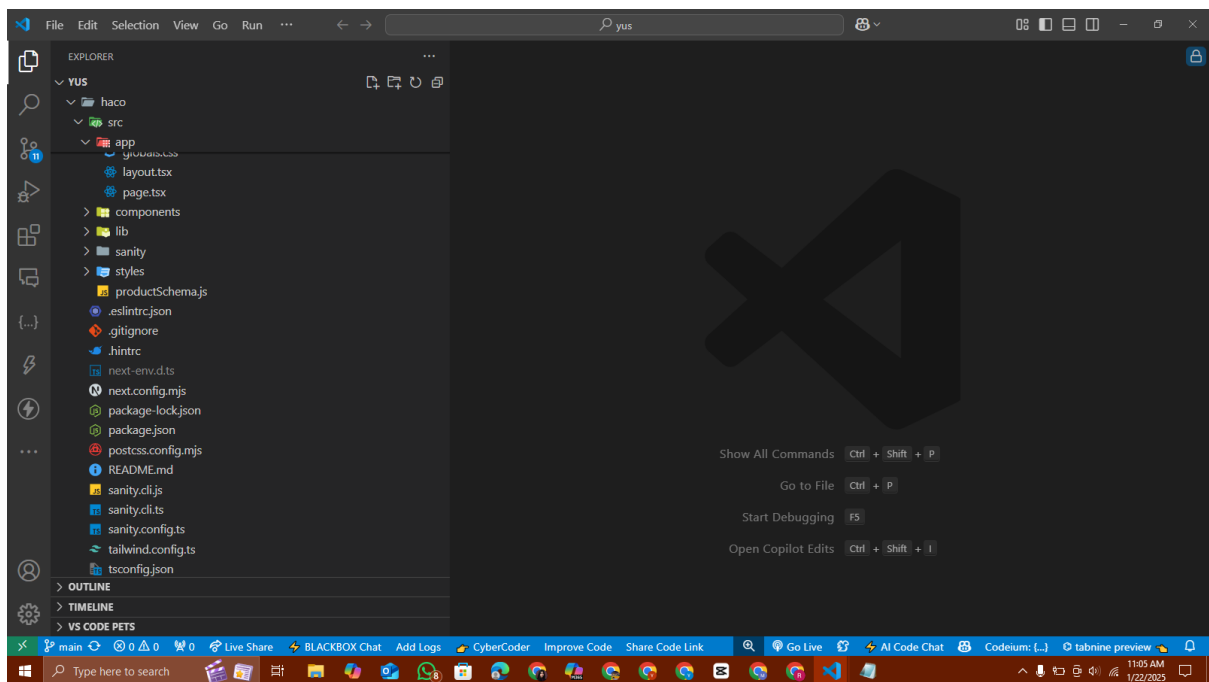
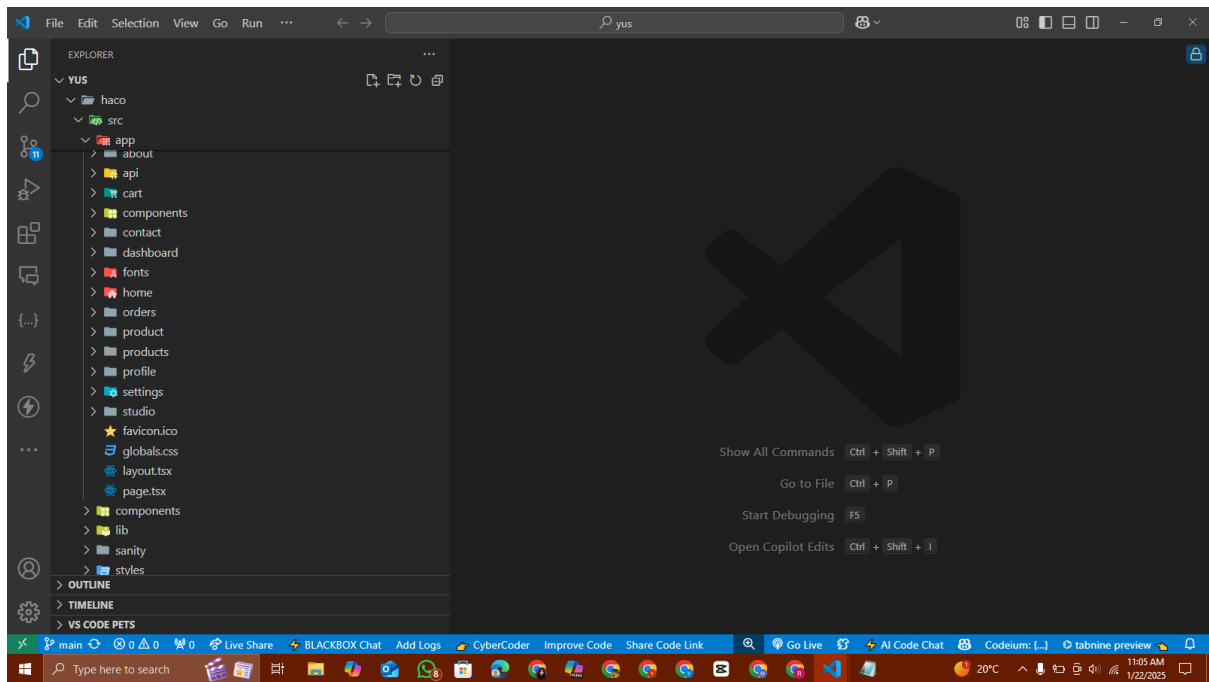
## Key Achievements:

1. Set up a staging environment mirroring the production setup for testing and quality assurance.
2. Deployed the **Next.js** application on a temporary hosting platform to simulate real-world conditions.
3. Integrated **Sanity API** and verified seamless data delivery and content management in the staging environment.
4. Tested **Tailwind CSS**-styled components for responsiveness and cross-device compatibility.
5. Conducted final pre-deployment checks, including performance optimization and security validations.

## Staging environment:

Deployed a staging build to variable functionality.





## CONCLUSION.

The deployment preparation process for Media Mart ensures a robust, secure, and efficient e-commerce platform. With the integration of **Next.js**, **Sanity API**, and **Tailwind CSS**, all technical and functional aspects are aligned for a seamless launch. Comprehensive testing and staging have minimized risks, ensuring the platform is ready for a successful production rollout.