***Day 6 Preparation and Staging Environment Setup***

Professional Environment Types:

1. TRN (Training) Purpose: Used for onboarding new team members and practice. Key Feature: Helps users get familiar with the system without impacting active environments.

2. DEV (Development) Purpose: Dedicated environment for developers to write and test code locally. Key Feature: Supports iterative coding and debugging without affecting production systems. 3. SIT (System Integration Testing) Purpose: Validates the integration between different systems and components. Key Feature: Ensures seamless communication and compatibility between subsystems. 4. UAT (User Acceptance Testing) Purpose: Allows stakeholders to test application functionality and validate that it meets business requirements. Key Feature: Ensures the system is ready for production deployment by aligning with user expectations. 5. PROD (Production) Purpose: The live, customer-facing environment where the application operates for end-users. Key Feature: Ensures high availability, performance, and security for real-world usage. 6. DR (Disaster Recovery) Purpose: Acts as a backup environment for critical situations such as system failures or disasters. Key Feature: Enables quick recovery and minimizes downtime in emergencies. Key Areas of Focus: Deployment Strategy; Planning Deployed the application on Vercel for staging and production. Integrated with Sanity CMS for dynamic content using tokens and dataset IDs Environment Variable Configuration Stored sensitive data (API keys, tokens) in .env.local file. Configured environment variables securely in Vercel Dashboard for deployment. Staging Environment; Setup Deployed the application to Vercel and verified successful deployment. Checked content fetching from Sanity CMS. Staging Environment Testing. Staging Environment Testing; Conducted Cypress functional tests, Postman API validation, and Lighthouse performance tests. Ensured security with HTTPS, proper data handling, and verified responsiveness across devices. Documentation; Updates Created a README.md file with all deployment instructions, configurations, and test results. Included all reports in the GitHub repository.

Steps for Implementation Step 1: Hosting Platform Setup Platform Chosen: Vercel was selected for quick and easy deployment.

https://hackathon-ecommerce-tp2q.vercel.app/ Connect Repository: Successfully connected the GitHub repository to Vercel for automatic deployments. Configured build settings and added the necessary scripts for deployment in the Vercel dashboard <https://github.com/Al-Aqmarmukarraam/hackathone-finale-q2.git>

Step 2: Configure Environment Variables Create .env.local File: Created the .env.local file to store sensitive data like API keys and tokens. Upload Variables to Vercel: Uploaded the environment variables to Vercel using the platform's dashboard for secure handling. Step 3: Deploy to Staging Deploy Application: Deployed the application to Vercel's staging environment for testing. Validate Deployment: Ensured the deployment build completed without errors. Verified that the application was loading correctly, and all content was fetched properly from Sanity CMS. Step 4: Staging Environment Testing; Testing Types Functional Testing: Verified the following features: Product Listing: Ensured all products were listed correctly. Product Details: Verified product details page displayed the correct information. User Profile: Checked user login, profile update, and profile display. Cart Operations: Ensured products could be added, removed, and quantities updated in the cart. Wishlist: Validated the ability to add and remove products from the wish list. Category: Ensured categories displayed correct product listings and filtered accordingly. Dynamic Routing: Verified that dynamic routing worked properly for product and category pages. Performance Testing: Used Lighthouse and GTmetrix to analyze the performance, speed, and responsiveness of the application. Ensured the application was optimized for various devices, screen sizes, and network conditions to deliver a smooth user experience. Security Testing: Validated input fields to ensure they were protected from vulnerabilities such as SQL injection and other malicious attacks. Ensured HTTPS was enabled for secure communication between the client and server. Verified that sensitive data, including API keys and user credentials, was transmitted securely and stored safely to avoid any data breaches. Conclusion for Deployment Preparation and Staging Step Day 6 focused on setting up a staging environment for deployment, including configuring environment variables, testing functionality, and updating documentation. This ensures a smooth and secure transition to the live platform, minimizing risks and enhancing readiness for production.