Day 6 - Deployment Preparation and Staging Environment Setup

Objective:

Day 6 focuses on preparing your marketplace for deployment by setting up a staging environment, configuring hosting platforms,
and ensuring readiness for a customer-facing application. Building on the testing and optimization work from Day 5, this stage
emphasizes ensuring the marketplace operates seamlessly in a production-like environment. Students will also learn about industry-
standard practices for managing different environments like non-production (TRN, DEV, SIT) and production (UAT, PROD, DR).

Key Learning Outcomes:

1. Set up and configure a staging environment for your marketplace. This includes:
   - Selecting a hosting platform such as Vercel or Netlify.
   - Connecting your GitHub repository to the platform.
   - Configuring build and deployment settings to ensure successful staging builds.
   - Setting up environment variables securely within the hosting platform.
   - Validating the application functionality in a production-like environment.

2. Understand professional environment management, including TRN, DEV, SIT, UAT, and PROD stages.
3. Conduct staging environment testing and document results.
4. Create professional deployment documentation, including performance and test case reports.
5. Organize all project files and documents in a structured GitHub repository.
   - Ensure a clear folder hierarchy (e.g., documents/, src/, public/).
   - Use consistent naming conventions for files and folders.
   - Provide a README.md file summarizing the project structure.

Professional Environment Types:

- TRN (Training): Used for onboarding and practice.
- DEV (Development): The environment for writing and testing code locally.
- SIT (System Integration Testing): Validates integrations between systems.

- UAT (User Acceptance Testing): Allows stakeholders to test functionality.
- PROD (Production): The live, customer-facing environment.
- DR (Disaster Recovery): A backup environment for critical situations.

Key Areas of Focus:

1. Deployment Strategy Planning
   - Choose a hosting platform like Vercel, Netlify, AWS, or Azure.
   - Finalize interactions with backend services such as Sanity CMS and third-party APIs.

2. Environment Variable Configuration
   - Secure API keys, database credentials, and sensitive data using .env files.
   - Configure environment variables securely in the hosting platform.

3. Staging Environment Setup
   - Deploy the application to a staging environment.
   - Validate that deployment builds successfully and the site loads correctly.

4. Staging Environment Testing
   - Conduct functional, performance, and security testing.
   - Use tools like Cypress, Postman, Lighthouse, and GTmetrix for thorough testing.
   - Verify responsiveness and error handling.
   - Document all test results and unresolved issues.

5. Documentation Updates
   - Create a README.md summarizing all six days of activities.
   - Include reports, test cases, and deployment instructions in the GitHub repository.

Submission Requirements:

What to Submit:

1. Staging environment deployed link.
2. A new GitHub repository with:
   - A documents folder containing all project files.
   - Test case report in CSV format.

- Performance testing results.
- A README.md file summarizing all project activities.