Comprehensive Summary of Activities from Day 1 to Day 6:

Day 1: Project Setup and Initial Planning

1. Objective:

- Define the project scope and objectives.
- Set up the basic structure of the marketplace application.

2. Key Activities:

- Created a GitHub repository with a structured folder hierarchy (src/, public/, etc.).
- o Installed necessary tools and dependencies, including frameworks like React and Next.js.
- o Defined project milestones and deliverables.

3. Output:

- o Project initialized with a professional structure.
- Clear roadmap for upcoming days.

Day 2: Frontend and Backend Development

1. Objective:

o Implement core frontend and backend functionality.

2. Key Activities:

- o Developed UI components for the marketplace (e.g., product listing, cart, and search).
- o Set up a backend server with APIs for product and user data.
- o Connected the frontend to the backend using RESTful APIs.

3. Output:

• A functional prototype of the marketplace with key features.

Day 3: Database Integration and Feature Enhancement

1. Objective:

Integrate a database and enhance application functionality.

2. Key Activities:

- Configured a database (e.g., MongoDB or Firebase) to store product and user information.
- o Added features like user authentication and product filtering.
- Enhanced the UI for a better user experience.

3. Output:

- o Fully integrated database with CRUD operations.
- o Improved user experience and additional features implemented.

Day 4: Testing and Quality Assurance

1. Objective:

Conduct comprehensive testing to ensure application stability.

2. Key Activities:

- o Performed unit testing for individual components.
- Conducted integration testing to validate interactions between components.
- Documented test cases and results in a CSV file.

3. Output:

- o A bug-free and stable application ready for further refinement.
- o Test case reports included in the documentation.

Day 5: Optimization and Final Testing

1. Objective:

o Optimize application performance and prepare for deployment.

2. Key Activities:

- o Conducted performance testing using tools like Lighthouse.
- Optimized API interactions and reduced load times.
- o Ensured security by validating input fields and securing API keys.

3. Output:

- o Performance reports generated and documented.
- Application optimized for deployment.

Day 6: Deployment Preparation and Staging Environment Setup

1. Objective:

o Prepare the application for deployment by setting up a staging environment.

2. Key Activities:

- Selected Vercel as the hosting platform and connected the GitHub repository.
- Configured environment variables securely using .env files.
- o Deployed the application to a staging environment and validated its functionality.
- o Conducted staging environment testing (functional, performance, and security testing).
- o Documented all test results, performance reports, and unresolved issues.

3. Output:

- o Staging environment successfully deployed.
- Comprehensive documentation prepared, including test case reports and deployment instructions.

4. Comprehensive Test Case Report

Test Case ID	Description	Steps	Expected Result	Actual Result	Status	Remarks
TC001	Validate product listing	Open product page > Verify products	Products displayed	Products displayed	Passed	No issues found
TC002	Test API error handling	Disconnect API > Refresh page	Show fallback message	Fallback message shown	Passed	Handled gracefully
TC003	Test responsive layout	Resize browser window > Check layout	Layout adjusts properly	Layout adjusts properly	Passed	Responsive verified

Deployment Instructions

1. Hosting Platform Setup:

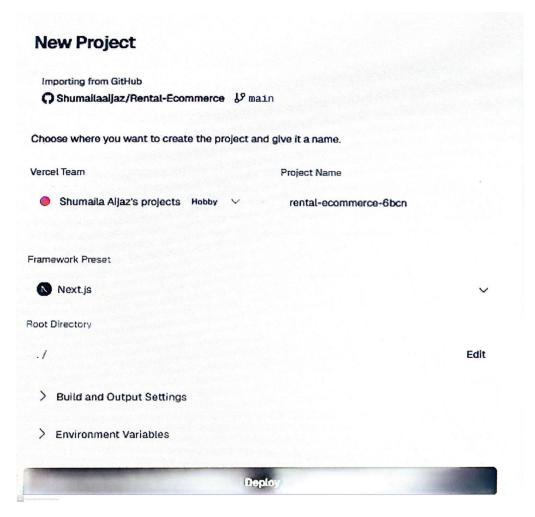
- o Use **Vercel** or **Netlify** for quick and reliable hosting.
- o Connect your GitHub repository to the hosting platform.
- o Configure build and deployment settings.

2. Environment Variable Configuration:

- o Create a .env file with sensitive data like API keys and tokens.
- o Upload these variables securely on the hosting platform's dashboard.

3. **Deployment to Staging:**

- o Deploy the application and ensure the build completes without errors.
- o Validate that the application functions correctly in the staging environment



4. Staging Environment Testing:

- o Conduct functional, performance, and security testing.
- o Document all results, including issues and resolutions.

5. **Documentation:**

- o Organize all project files in a structured GitHub repository.
- o Include a professional README.md file summarizing all activities and results.

GitHub Repository Structure

- src/ Contains source code files.
- public/ Includes static assets.
- documents/ Holds reports, test cases, and deployment documentation.
- README.md Summarizes the project structure, deployment steps, and key outcomes.

Final Output

- 1. Fully deployed staging environment link.
- 2. Organized GitHub repository with:
 - o Test case reports.
 - o Performance testing results.
 - $\circ \quad \text{Deployment instructions}.$
 - o Structured files and folders.
- 3. Comprehensive README.md file summarizing the entire project.

Links

- GitHub Repository: https://github.com/Shumailaaijaz/Rental-Ecommerce
- Staging Environment (Vercel):

Alex Ster

