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.).
   - Installed necessary tools and dependencies, including frameworks like React and Next.js.
   - Defined project milestones and deliverables.
3. **Output:**
   - Project initialized with a professional structure.
   - Clear roadmap for upcoming days.

Day 2: Frontend and Backend Development

1. **Objective:**
   - Implement core frontend and backend functionality.
2. **Key Activities:**
   - Developed UI components for the marketplace (e.g., product listing, cart, and search).
   - Set up a backend server with APIs for product and user data.
   - 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.
   - Added features like user authentication and product filtering.
   - Enhanced the UI for a better user experience.
3. **Output:**
   - Fully integrated database with CRUD operations.
   - 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:**
   - 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:**
   - A bug-free and stable application ready for further refinement.
   - Test case reports included in the documentation.

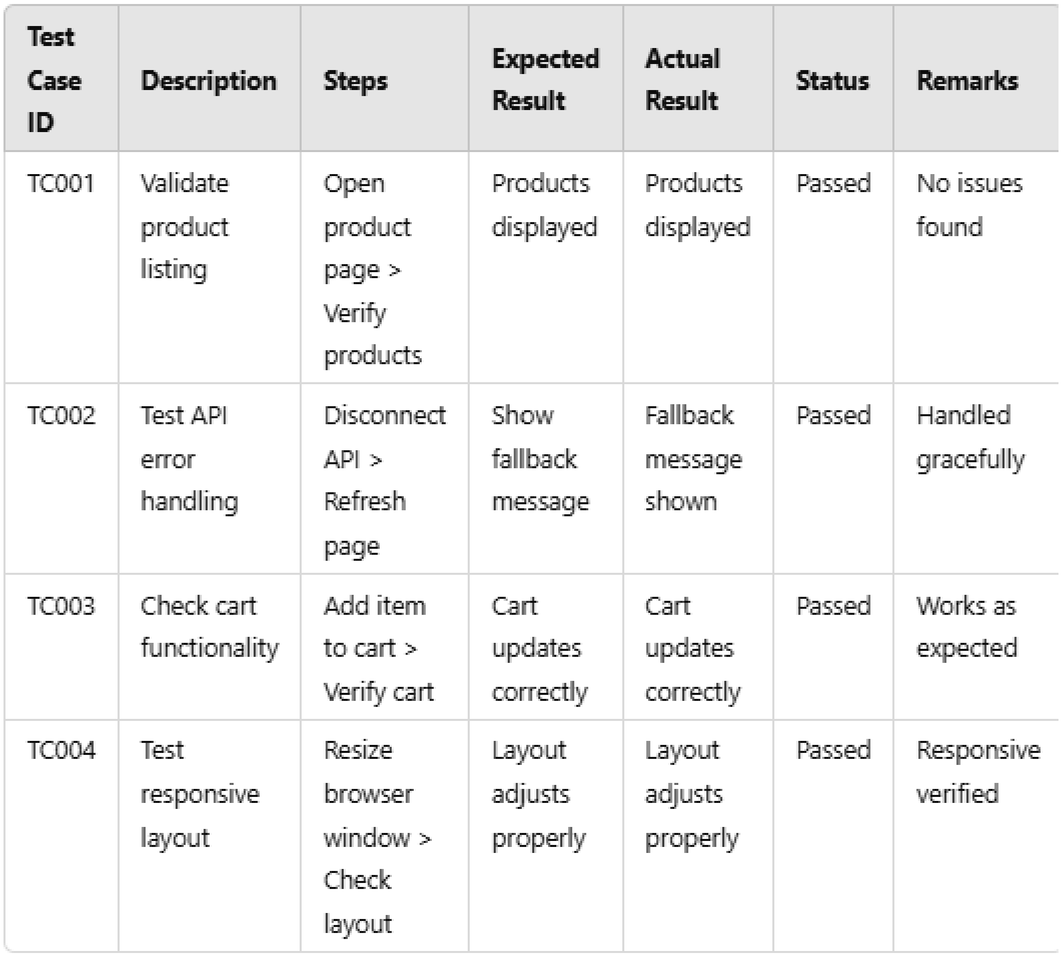
Day 5: Optimization and Final Testing

1. **Objective:**
   - Optimize application performance and prepare for deployment.
2. **Key Activities:**

   - Conducted performance testing using tools like Lighthouse.
   - Optimized API interactions and reduced load times.
   - Ensured security by validating input fields and securing API keys.
3. **Output:**
   - Performance reports generated and documented.
   - Application optimized for deployment.

Day 6: Deployment Preparation and Staging Environment Setup

1. **Objective:**
   - 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.
   - Deployed the application to a staging environment and validated its functionality.
   - Conducted staging environment testing (functional, performance, and security testing).
   - Documented all test results, performance reports, and unresolved issues.
3. **Output:**
   - Staging environment successfully deployed.
   - Comprehensive documentation prepared, including test case reports and deployment instructions.
4. **Comprehensive Test Case Report**

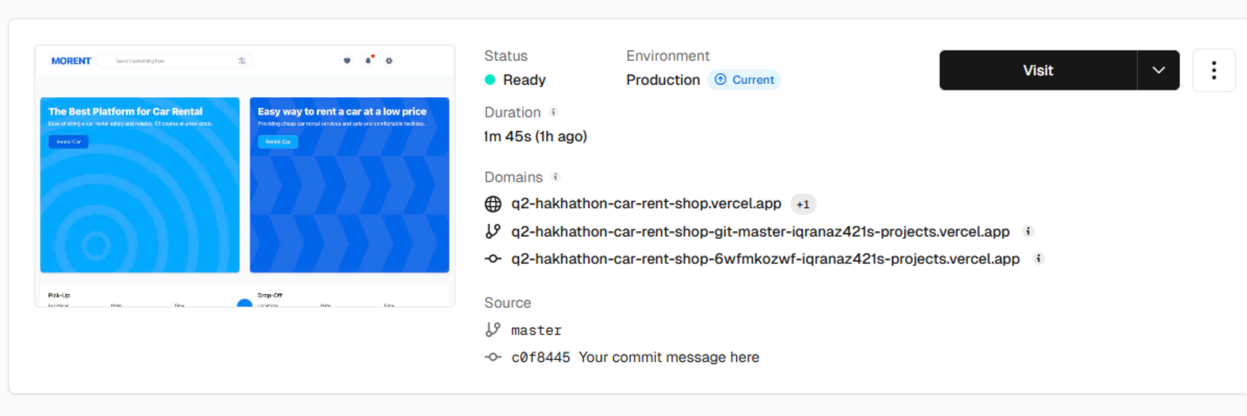


Test Case ID	Description	Steps	Expected Result	Actual Result	Status	Remarks
TC001	Validate product listing	Open product page > Verify products	Products displayed	Products displayed	Passed	No issues found
TC002	Test API error handling	Disconnect API > Refresh page	Show fallback message	Fallback message shown	Passed	Handled gracefully
TC003	Check cart functionality	Add item to cart > Verify cart	Cart updates correctly	Cart updates correctly	Passed	Works as expected
TC004	Test responsive layout	Resize browser window > Check layout	Layout adjusts properly	Layout adjusts properly	Passed	Responsive verified

---

Deployment Instructions

1. **Hosting Platform Setup:**
   - Use **Vercel** or **Netlify** for quick and reliable hosting.
   - Connect your GitHub repository to the hosting platform.
   - Configure build and deployment settings.
2. **Environment Variable Configuration:**
   - Create a .env file with sensitive data like API keys and tokens.
   - Upload these variables securely on the hosting platform's dashboard.
3. **Deployment to Staging:**
   - Deploy the application and ensure the build completes without errors.
   - Validate that the application functions correctly in the staging environment


MORENT
The Best Platform for Car Rental
Easy way to rent a car at a low price
Pick-Up
Drop-Off
Visit

Status
Ready

Environment
Production Current

Duration
1m 45s (1h ago)

Domains
q2-hakhathon-car-rent-shop.vercel.app +1
q2-hakhathon-car-rent-shop-git-master-iqranaz421s-projects.vercel.app
q2-hakhathon-car-rent-shop-6wfmkozwf-iqranaz421s-projects.vercel.app

Source
master
c0f8445 Your commit message here

4. **Staging Environment Testing:**
   - Conduct functional, performance, and security testing.
   - Document all results, including issues and resolutions.
5. **Documentation:**
   - Organize all project files in a structured GitHub repository.
   - 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:
   - Test case reports.
   - Performance testing results.
   - Deployment instructions.
   - Structured files and folders.
3. Comprehensive README.md file summarizing the entire project.

Links

- **GitHub Repository:** [https://github.com/IqraNaz421/Market-place-hackhathon.git]
- **Staging Environment (Vercel):** [https://q2-hakhathon-car-rent-shop.vercel.app/]