Day 6 - Deployment Preparation and Staging Environment Setup :-

Objective

The focus of Day 6 was to finalize the marketplace project for deployment by ensuring it was fully prepared for a production environment. This phase involved conducting in-depth testing to identify and resolve potential issues, optimizing performance for real-world usage, and crafting professional deployment documentation. The documentation provided clear, concise guidance to streamline the deployment process and deliver an intuitive, seamless user experience. These efforts were critical to ensuring the project's reliability, usability, and readiness for end-users.

Key Learning Outcomes

Deployment Setup

- **Hosting Platform Selection:** Vercel was selected for its robust features and seamless deployment capabilities.
- **GitHub Integration:** The project's GitHub repository was linked to Vercel, automating the deployment process.
- **Configuration:** Build settings, environment variables, and secured API keys were configured on Vercel.
- **Environment Variables:** Critical variables such as projectId, dataset, and API-token were configured to enhance security.
- **Validation:** The application was successfully deployed to a production-ready environment.

Comprehensive Testing

- **Functional Testing:**
  - Cypress was used to validate user workflows.
  - Postman was utilized to test API responses, ensuring accurate cart management and search functionalities.

- **Performance Testing:**
  - Tools like Lighthouse were employed to evaluate page speed, responsiveness, and performance metrics.
- **Security Testing:**
  - Input validation was implemented.
  - HTTPS protocols were enforced.
  - Sensitive data, such as API keys, was secured against unauthorized access.
- **Cross-Device Compatibility:**
  - The application's performance was verified across multiple devices and browsers, ensuring responsiveness.

- **Error Handling:**
  - The system's ability to handle errors gracefully and provide meaningful feedback to users was assessed.

Deployment Strategy

- **Hosting and Backend Integration:**
  - Seamless communication between the frontend and backend services, including Sanity CMS and third-party APIs, was established.
  - Environment variables were secured to maintain data integrity and user privacy.
- **Staging Environment:**
  - The project was deployed to a staging environment for pre-production testing.
  - Successful builds were validated, and all features were confirmed to operate as expected.

Testing Tools Used

- **Postman:** Ensured reliable communication between components by validating API responses.
- **Lighthouse:** Measured and analyzed key performance metrics to optimize the user experience.
- **Cypress:** Automated end-to-end testing for critical workflows.
- **Browser Developer Tools:** Debugged and optimized frontend components for improved performance.

Testing Results:

Here’s an example of documented test cases:

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

Additional Enhancements

- **Monitoring Setup:** Error-tracking tools such as Sentry were integrated to capture and resolve runtime errors p