**Day 5 - Testing, Error Handling, and Backend Integration Refinement - TokenRent**

**Testing and Refinements**

The TokenRent marketplace underwent comprehensive testing to ensure seamless functionality, security, and performance. Functional testing was performed on core features such as **Metamask authentication, equipment listing, NFT minting, and rental agreement processing**. The system was tested using Cypress for frontend validation, Hardhat for smart contract testing, and Postman for API requests.

Error handling was implemented to enhance user experience and prevent transaction failures. Network failures were managed using try-catch blocks, while incorrect input handling was enforced using form validation and smart contract require conditions. UI notifications were added to guide users when transactions fail due to gas issues or insufficient balance.

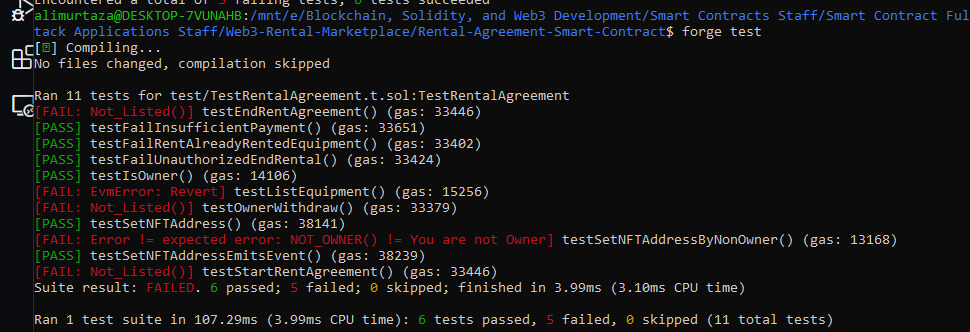
Performance optimizations focused on **lazy loading images, caching API responses, and gas-efficient smart contract functions**. Next.js optimizations such as static site generation (SSG) and server-side rendering (SSR) were applied where necessary to improve speed.

Security testing included **input validation, smart contract access control, and API key protection**. Chainlink Automation was tested for ensuring rental agreements are automatically finalized, preventing manual intervention errors.

User Acceptance Testing (UAT) was conducted, simulating real-world interactions such as listing equipment, renting an item, and checking rental expiry automation. Feedback highlighted the ease of Web3 wallet integration but suggested improvements in UI loading indicators.

**Code and Screenshots**

**RentalAgreemnt smart contract test cases Screenshot**

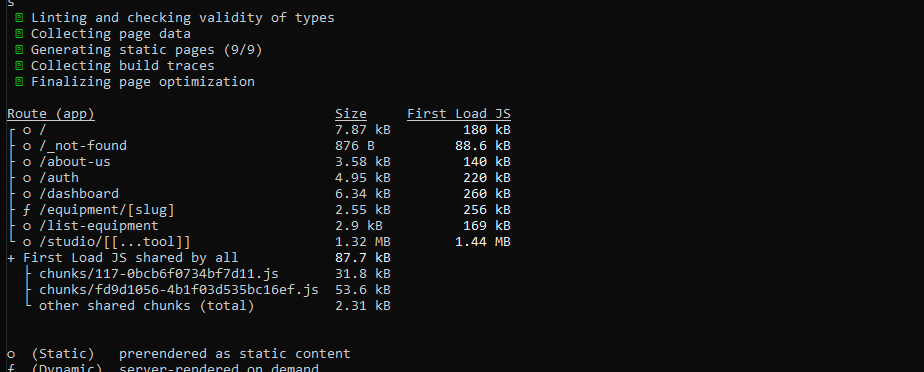
****

**NFT contract test cases**

**A computer screen with text on it

Description automatically generated**

**Include frontend deployments testing**

****