**Day 6 - Deployment Preparation and Staging Environment Setup - TokenRent**

**1. Deployment Strategy and Hosting Setup**

For the deployment of TokenRent, **Vercel** was chosen as the primary hosting platform due to its seamless integration with Next.js, fast build times, and efficient environment management. The **GitHub repository** was connected to Vercel, enabling automatic deployments on code commits. The smart contracts were deployed on the **Ethereum Sepolia testnet**, ensuring a fully functional Web3 experience in a real-world setting before the mainnet launch.

**2. Environment Variable Configuration**

To secure sensitive information, environment variables were properly configured using **.env files**. The following keys were stored securely on Vercel’s dashboard:

NEXT\_PUBLIC\_SANITY\_PROJECT\_ID=id

NEXT\_PUBLIC\_SANITY\_DATASET=production

NEXT\_PUBLIC\_ALCHEMY\_API=your\_alchemy\_api

NEXT\_PUBLIC\_RENTAL\_CONTRACT\_ADDRESS=0xF70CB025c530463370EE1173231d1554fC897901

NEXT\_PULIC\_EQUIPMENT\_NFT=0xB6bE67aB0a647Df76C4e728F7657ae2a49295bDb

This ensures API keys and blockchain addresses remain private while allowing secure contract interactions from the frontend.

**3. Staging Deployment and Testing**

A staging environment was successfully set up on Vercel with the following URL: **[Insert Staging Link]** Deployment logs were reviewed to ensure all smart contract interactions and API calls functioned correctly. The application was tested across different devices and browsers to verify responsiveness and accessibility.

**4. Staging Environment Testing**

Comprehensive testing was conducted to validate the system before the final launch.

**Functional Testing:**

* **Metamask Authentication:** Users can sign in and connect wallets successfully.
* **Equipment Listing & Minting:** NFTs are minted and listed correctly.
* **Rental Agreements:** Users can rent equipment, and transactions are reflected on the blockchain.

**Performance Testing:**

* Lighthouse and GTmetrix were used to optimize page load times and interactive speeds.
* Image optimizations and caching were applied to enhance UX.

**Security Testing:**

* Input validation prevents unauthorized data submissions.
* HTTPS is enforced for all API calls.
* Smart contracts were checked for reentrancy and access control vulnerabilities.

**5. Documentation and Repository Submission**

A detailed **README.md** file was created to document project structure, deployment steps, and testing reports. The repository includes:

* **/contracts/** → Solidity smart contracts.
* **/frontend/** → Next.js frontend.
* **/tests/** → Cypress & Hardhat test cases.
* **/docs/** → Reports, deployment guides, and test case logs.

GitHub Repository: **[Insert Repository Link]**

**6. Checklist & Final Review**

✅ Hosting setup completed on Vercel  
✅ Environment variables securely configured  
✅ Staging environment successfully deployed  
✅ Functional, performance, and security tests passed  
✅ GitHub repository structured and documented  
✅ Final report submitted with test results and staging URL