Hackathon Rule Sheet

Tech Stack Requirements

1. **Frontend**:
   - Must be built using **React.js** or a framework based on React (e.g., Next.js, Remix).
   - UI frameworks like TailwindCSS, Material-UI, Chakra UI are allowed.
2. **Smart Contracts**:
   - Must be written in **Solidity**.
   - Contracts must be **EVM-compatible** and deployable on the **BlockDAG network**.
3. **Blockchain Integration**:
   - Use libraries like **ethers.js** or **web3.js** to interact with smart contracts.
   - Frontend must connect to the BlockDAG network (e.g., using MetaMask or WalletConnect).
4. **Deployment**:
   - Smart contracts must be deployed to a **BlockDAG-compatible EVM chain**.
   - Frontend must be hosted (e.g., Vercel, Netlify, or IPFS).

Basic Rules

1. **Team Size**: 1–5 members per team.
2. **Project Timeline**:
   - Start: [Insert Start Date/Time]
   - End: [Insert End Date/Time]
   - All submissions must be made before the deadline.
3. **Originality**:
   - Projects must be original and built during the hackathon.
   - Reusing small boilerplate code is allowed; cloning existing dApps is not.
4. **Code Submission**:
   - Submit code via a public GitHub repo with a clear README.
   - Include build instructions and a short demo video (max 3 min).
5. **Demo Day**:
   - Live or recorded demo required.
   - Teams should be able to answer questions about their architecture and code.

6. **Judging Criteria**:
    - Innovation & creativity
    - Technical implementation
    - UI/UX design
    - Blockchain integration
    - Alignment with BlockDAG and EVM standards
7. **Disqualification Grounds**:
    - Plagiarism
    - Late submission
    - Breach of code of conduct

Optional Tools & Libraries

- **Hardhat** or **Foundry** for contract development
- **The Graph** (if indexing needed)
- **IPFS/Filecoin** for decentralized storage
- **Web3Modal, RainbowKit**, **Wagmi**, or **ConnectKit** for wallet connections