**/\*Report**

**Members:-**

**Deepak Barwal :- 101531741**

**Dapp :- Gold Digger**

**About the project :-**

The Lottery Draw App is a decentralized application built on the Solana blockchain. It lets users to buy lottery tickets and check results and interact with the lottery system easily. The dapp is designed to be more secure and transparent which makes the lottery process straightforward and fair for everyuser because of its decentralized property.

**Tech Stack:**

**Frontside:**

* **React.js**
* **TypeScript**
* **TailwindCSS**

**Backend:**

* **Solidity**
* **Thirdweb**

**Blockchain:**

* **Solana**

**Architecture:**

**A diagram of a company

Description automatically generated**

**Get Started:**

**Prerequisites:**

* Create a MetaMask account: Install the MetaMask browser extension and create an account to interact with the blockchain.
* Sign up for a Thirdweb account: Register on the Thirdweb platform to manage and deploy your smart contracts.
* Install Node.js: Ensure Node.js is installed on computer for running JavaScript code and managing packages.

**Environmental variables:**

To run this project, you will need to add the following environment variable to your .env file

- NEXT\_PUBLIC\_LOTTERY\_CONTRACT\_ADDRESS

**Installations:**

1. Node.js and npm

Node.js: Make sure Node.js installed, which includes npm (Node Package Manager).

**2. Solana CLI**

**Purpose:** The Solana Command Line Interface (CLI) is needed to interact with the Solana blockchain.

Installation:

* + Open Command Prompt and Run the command:

[ sh -c "$(curl -sSfL <https://release.solana.com/v1.11.14/install>)" ]

**3. Solana Web3.js (for interacting with the Solana blockchain in the frontend)**

* **Purpose:** Solana Web3.js is a library for connecting DApp frontend to the Solana blockchain.
* **Installation:**
  + Add it to your project by running:

“npm install @solana/web3.js”

**4. TypeScript**

* **Purpose:** TypeScript is used for writing error-free code.
* **Installation:**
  + Add it to project by running:

“npm install typescript --save-dev”

* + Initialize a TypeScript configuration file:

“npx tsc –init”

**5. TailwindCSS**

* **Purpose:** TailwindCSS is used for styling the DApp.
* **Installation:**
  + Follow these commands to install and configure TailwindCSS:

“npm install -D tailwindcss postcss autoprefixer”

“npx tailwindcss init -p”

**6. MetaMask**

* **Purpose:** MetaMask is a browser extension wallet used for interacting with the blockchain.
* **Installation:**
  + Install from the MetaMask website .

**Analysis:**

**1. Integration with Solana Blockchain**

* **Easy:**
  + **Using Solana Tools:** Tools and libraries from Solana make it simpler to connect your app to the blockchain.
  + **Testing on Devnet:** You can test everything on Solana’s test network without risking real money.
* **Hard:**
  + **Debugging Issues:** Fixing problems in your smart contracts was tricky.
  + **Handling Transactions:** Managing transaction fees and wallet interactions was complicated.

**2. Smart Contract Development**

* **Easy:**
  + **Using Frameworks:** Frameworks like Anchor help make smart contract development easier by providing useful tools.
  + **Following Patterns:** Using proven methods for lottery contracts can simplify the process.
* **Hard:**
  + **Contract Complexity:** Writing secure and bug-free smart contracts involves detailed work.
  + **Testing and Deployment:** Testing on test networks and deploying to the mainnet was challenging.

**3. React Frontend Development**

* **Easy:**
  + **React Libraries:** React and related libraries help build the user interface and connect to the blockchain.
  + **UI Components:** Creating UI elements with React is straightforward with the help of component libraries.
* **Hard:**
  + **Handling Transactions:** Managing blockchain transactions smoothly on the frontend was complex.
  + **State Management:** Keeping track of blockchain interactions and errors requires careful planning.

**4. Security and Testing**

* **Easy:**
  + **Automated Tools:** Tools helped automatically find common issues early on.
  + **Best Practices:** Following standard security and testing practices simplifies the process.
* **Hard:**
  + **Ensuring Security:** Making sure your smart contracts and frontend are secure requires thorough testing and checks.
  + **End-to-End Testing:** Full testing of the app, including how all parts work together, and was complex and time-consuming.

**What Worked Well:**

* Solana Blockchain: Choosing the Solana blockchain was a good move because it’s fast and has low transaction fees, making your lottery DApp efficient and affordable for users.
* Smart Contracts with Truffle: Using Truffle to develop your smart contracts gave you a strong, reliable framework for building, testing, and deploying your lottery features.
* Fair Lottery System: The decentralized lottery system you created, where winners are chosen by smart contracts, ensured that the process was fair and transparent, which is the most important part for user trust.
* Easy-to-Use Interface: Making the user interface simple and user-friendly helped make your DApp accessible to a wider audience, even those who aren’t familiar with web applications.
* Security Focus: We made sure the smart contracts were safe and had no flaws, which is really important for keeping the DApp and users secure.

User Interface:

A screenshot of a game

Description automatically generated

**Detailed Workflow :**

**1.User Interaction:**

* **Step 1**: **Access DApp**: The user opens the GoldDigger DApp via a web browser.
* **Step 2**: **Connect Wallet**: The user connects their crypto wallet (e.g., Phantom Wallet) to the DApp.
* **Step 3**: **Purchase Tickets**: The user chooses how many lottery tickets they want to buy and confirms the purchase using their wallet.
* **Step 4**: **Participate in Lottery**: The user's purchase is recorded on the blockchain, and they receive a confirmation of their ticket numbers.

**2. Backend Processing:**

* **Step 5**: **Transaction Handling**: The backend (Node.js with Express) processes the transaction by interacting with the user's wallet and the Solana blockchain.
* **Step 6**: **Smart Contract Interaction**: The backend triggers the smart contract to mint the lottery tickets and update the blockchain with the user's entry.
* **Step 7**: **Lottery Pool Update**: The smart contract automatically updates the lottery pool with the new entries and funds.

**3. Lottery Draw:**

* **Step 8**: **Draw Initiation**: At a predefined time or when a specific condition is met, the lottery draw is initiated either automatically by the smart contract or manually by an administrator.
* **Step 9**: **Random Number Generation**: The smart contract uses a secure method to generate a random number corresponding to the winning ticket.
* **Step 10**: **Winner Selection**: The smart contract checks the generated number against the ticket numbers in the pool and identifies the winner.

**4. Prize Distribution:**

* **Step 11**: **Automatic Transfer**: The prize money is automatically transferred to the winner's wallet by the smart contract.
* **Step 12**: **Notification**: Both the winner and other participants are notified of the results via the DApp interface and, optionally, through connected communication channels.

**5. Transparency and Security:**

* **Step 13**: **Blockchain Verification**: All transactions, including ticket purchases, the draw, and prize distribution, are publicly recorded on the Solana blockchain, ensuring transparency.
* **Step 14**: **Audit and Reports**: The DApp can generate reports and allow users to audit the entire lottery process to ensure fairness.

**6. Feedback Loop:**

* **Step 15**: **User Feedback**: Users can provide feedback on their experience through the DApp.
* **Step 16**: **Continuous Improvement**: The development team collects feedback and makes necessary updates or improvements to the DApp.

**Roadmap:**

**Phase 1: Planning**

* + **Research: Study the lottery industry, find out who will use the DApp, and look at competitors.**
  + **Requirements: Decide on key features, how the app will look, and ensure security.**
  + **Tech Choices: Pick the technologies like Solana for blockchain, React.js for the user interface, and Node.js for the backend.**

**Phase 2: Design**

* + **UI Design: Create basic sketches and design the user interface.**
  + **Smart Contract: Plan how the smart contract will handle tickets, prize distribution, and randomness.**
  + **Prototype: Build a simple version of the DApp to show how it works.**

**Phase 3: Development**

* + **Frontend: Develop the user interface using React.js, including wallet integration.**
  + **Backend: Set up the backend using Node.js to manage data and connect to the blockchain.**
  + **Smart Contract: Write and test the smart contract on Solana’s test network.**
  + **Integration: Make sure the frontend, backend, and smart contract work well together.**

**Phase 4: Testing**

* + **Testing: Test each part of the DApp, make sure everything works together, and check for security issues.**
  + **Beta Launch: Release a beta version on the Solana test network for selected users to try.**

**Phase 5: Launch**

* + **Mainnet Launch: Deploy the DApp on the Solana mainnet for everyone to use.**
  + **Marketing: Start marketing to attract users.**
  + **Public Launch: Open the DApp to the public, allowing people to join the lottery.**

**Phase 6: Support and Updates**

* + **User Support: Help users with any issues they face.**
  + **Improvements: Update the DApp based on user feedback.**
  + **New Features: Add new features like supporting different currencies or other lottery games.**

**Market Analysis for GoldDigger DApp :**

**1. Industry Overview**

The online lottery industry is growing because more people have internet access and prefer the convenience of online lotteries. Blockchain technology is being used in lotteries to improve security, transparency, and fairness.

**2. Market Size and Growth**

* Global Lottery Market: The lottery market worldwide was worth about $300 billion in 2023 and is expected to grow by 5-6% each year.
* Blockchain Lottery Market: Lotteries using blockchain are a smaller but fast-growing part of the market, expected to reach $10 billion by 2025. This growth is driven by the demand for secure and transparent lottery systems.

**3. Key Trends**

* Decentralization: People are increasingly choosing decentralized platforms where lottery results are not controlled by any single entity.
* Mobile Accessibility: More users are playing lotteries on their mobile devices, making mobile-friendly platforms important.
* Regulatory Changes: Governments are starting to create rules that support the use of blockchain technology in lotteries.

**4. Target Audience**

* Crypto Enthusiasts: People familiar with blockchain and cryptocurrency are likely to be early users.
* Tech-Savvy Lottery Players: Traditional lottery players who want a more secure and transparent platform.
* Millennials and Gen Z: Younger generations who prefer digital and decentralized options.

**5. Competitive Landscape**

* Traditional Lotteries: Established lottery companies dominate the market but face new competition from blockchain-based platforms.
* Blockchain Lotteries: Competitors like True Flip, LuckyBlock, and PoolTogether are already offering blockchain-based lottery games.

**6. Opportunities**

* Blockchain Benefits: Using blockchain technology for fairness, transparency, and security can set GoldDigger apart from traditional lotteries.
* Growing Crypto Use: As more people use cryptocurrencies, the potential audience for blockchain lotteries like GoldDigger will grow.
* Global Reach: Blockchain lotteries can operate worldwide, overcoming the limitations of local laws.

**7. Challenges**

* Legal Compliance: Navigating different legal requirements in various countries can be difficult.
* Building Trust: Gaining the trust of users who are not familiar with blockchain technology.
* Market Competition: Competing with well-known lottery platforms requires strong marketing and educating users.

**Cost Projections:**

**A screenshot of a black and white screen

Description automatically generated**

**Conclusion :**

GoldDigger is designed to make the lottery experience better by using blockchain technology. It's built on the Solana blockchain, which helps keep the platform secure, transparent, and easy to use. The DApp uses smart contracts to automate the process, making it fair and efficient for everyone involved. With features like low fees and integration with popular wallets, GoldDigger is set to attract many users. By sticking to a clear plan and carefully managing costs, GoldDigger has a strong chance to grow and succeed in the decentralized lottery market.

**REFERENCES:**

<https://phantom.app/learn>

<https://help.phantom.app/hc/en-us>

<https://react.dev/reference/react/apis>

<https://solana.com/developers/guides>

https://www.youtube.com/watch?v=mVxRzkvX\_w0  
<https://www.p2p.org/>

<https://www.bitdeal.net/contact-us>

https://en.wikipedia.org/wiki/Solidity