Frontend Developer Assignment

Objective:
In this assignment, your goal is to develop a frontend application that integrates with the Solana blockchain using a Solana wallet (e.g. Phantom or Solflare). The application should allow users to create tokens and mint tokens by interacting with the solana SPL token program on devnet.

The candidate is expected to build a user interface (UI) that allows seamless interactions with Solana's blockchain, with smooth and responsive wallet integration. The focus is on implementing the frontend, connecting the wallet, and making transactions using pre-deployed smart contracts.

Project Requirements:
1. Solana Wallet Integration :
   - Wallet Authentication: Implement wallet authentication using a Solana wallet (e.g., Phantom, Sollet).
   - Connect and Disconnect Wallet: Users should be able to connect and disconnect their Solana wallet from the application.
   - Display Wallet Information: The app should display the connected wallet address and the user's Solana balance.
   - Error Handling: Handle cases where the wallet fails to connect or the user is unable to interact with the blockchain.
2. Smart Contract Interaction :
   - Token Creation:Implement a feature that allows users to create new tokens using the SPL Token Program. Ensure that the token creation process is intuitive and provides feedback on success or failure.
   - Mint Tokens: Allow users to mint tokens using their wallet. Provide notifications for successful transactions.
   - Send Tokens: Implement functionality for users to send tokens to other accounts. Ensure transactions are handled properly, with clear feedback on success or failure.
   - Transaction Handling: Ensure all transactions are processed correctly and provide real-time feedback to users. Display transaction details and confirmations to users.

3. UI/UX Design - User Interface:
    - Create a modern and responsive interface that is simple to navigate.
    - Connect Wallet Button: Provide a clear call-to-action for users to connect their wallet.
    - Display Wallet Balance: Show the balance of SOL and tokens the user holds.
    - Token Minting UI: Create a simple interface for minting tokens.
    - Transaction Status: Provide feedback such as loading states, error messages, and success notifications after each transaction.
4. Blockchain Data Fetching :
    - Fetching Token Balance: Fetch and display the balance of tokens and SOL for the connected wallet.
    - Transaction History: Display the transaction history for minting and token transfers.
5. Responsiveness - Responsive Design:
    - Ensure the app is mobile-friendly and responsive. The design should adapt across multiple screen sizes.
6. Code Quality & Performance :
    - The code should be modular and organized following best practices.
    - Optimize the app for fast loading and smooth interaction.
    - Ensure proper error handling and edge case management.

Submission Guidelines:

1. GitHub Repository:
    - A public GitHub repository containing the full codebase.
    - Ensure the code is well-structured, well-commented, and follows best practices.
2. Live Demo:
    - Deploy the application on Vercel or Netlify.
    - Provide a working live demo of the application for testing.
3. Video Explanation:

   A 5-10 minute video where you explain:
    - Your approach to solving the problem.
    - How you integrated Solana wallet and interacted with the smart contract.
    - Any challenges faced during development and how you overcame them.

4. UI/UX:

- The application should have a clean, modern, and functional design.
- Handle loading states and errors properly.

Evaluation Criteria:

1. Wallet Integration:

- Proper integration of the Solana wallet for both authentication and blockchain transactions.
- The wallet should be easily connectable and disconnectable.
- Display accurate wallet balances and transaction details.

2. Smart Contract Interaction:

- Efficient interaction with Solana's token program or an existing smart contract for creating ,minting, and sending tokens.
- Transactions should be correctly processed, with user-friendly feedback.

3. UI/UX Design:

- Clear, user-friendly, and intuitive design for interacting with the blockchain.
- A well-structured UI, responsive design, and an engaging experience.

4. Code Quality:

- Clean, maintainable, and modular code with proper comments and documentation.
- Proper error handling and usage of async/await for handling blockchain interactions.

5. Performance:

- Ensure the app performs well and loads quickly.
- Optimize for low-latency blockchain interactions.

6. Transaction Feedback:

- Provide clear notifications and feedback when transactions are processed (creation, minting, transfer).

Additional Resources:

Solana Web3.js Documentation: https://solana-labs.github.io/solana-web3.js/
Phantom Wallet: https://phantom.com/
Solflare Wallet: https://docs.solflare.com/solflare
Solana Token Program: https://spl.solana.com/token
Solana Devnet Explorer: https://explorer.solana.com/?cluster=devnet
Solana Airdrop Tool: https://faucet.solana.com/

Conclusion:

This assignment aims to test the candidate's ability to work with blockchain integration in a frontend application. By interacting with the Solana blockchain, using Solana wallets, and integrating smart contract functions, the candidate will demonstrate their understanding of web3 technologies and frontend best practices.

Optional Features (Bonus Points):

- Real-Time Blockchain Data: Implement real-time updates for token balances and market prices.
- Transaction History: Show a user's transaction history (e.g., minted, bought, and sold tokens).
- Web3 Event Listeners: Implement Solana event listeners to track wallet events, such as token transfers and smart contract interactions.