Game Functionality:

1. **Player Login and Authentication**:
    - **Feature**: Players must be able to create an account and log in.
    - **Purpose**: Manage player sessions, track their progress, and store personal data like poker statistics and game history.
    - **Details**: Use Express.js for authentication, and Postgres to store user credentials.
2. **Main Menu**:
    - **Feature**: After login, players press “Play” and are randomly placed in an available game.
    - **Purpose**: Simplify user experience by directly placing players in games without showing a lobby screen.
    - **Details**: Players can also enter a specific code to join a friend’s game if needed.
3. **Poker Game Logic**:
    - **Feature**: Implement poker game mechanics, including dealing cards, betting rounds, winning conditions, and player actions (e.g., check, call, raise, fold).
    - **Purpose**: Handle the main poker gameplay according to standard poker rules.
    - **Details**: Ensure support for poker with standard actions. Each game will manage blinds, community cards, hand rankings, and chip stacks (points?).
4. **Multiplayer Mode** :
    - **Feature**: Real-time gameplay allowing multiple players to participate in a poker game.
    - **Purpose**: Provide the core multiplayer experience, with players joining the same game session and interacting in real time.

- **Details**: This will use Socket.IO or WebSockets to handle real-time communication between players (e.g., actions, chat, game state).

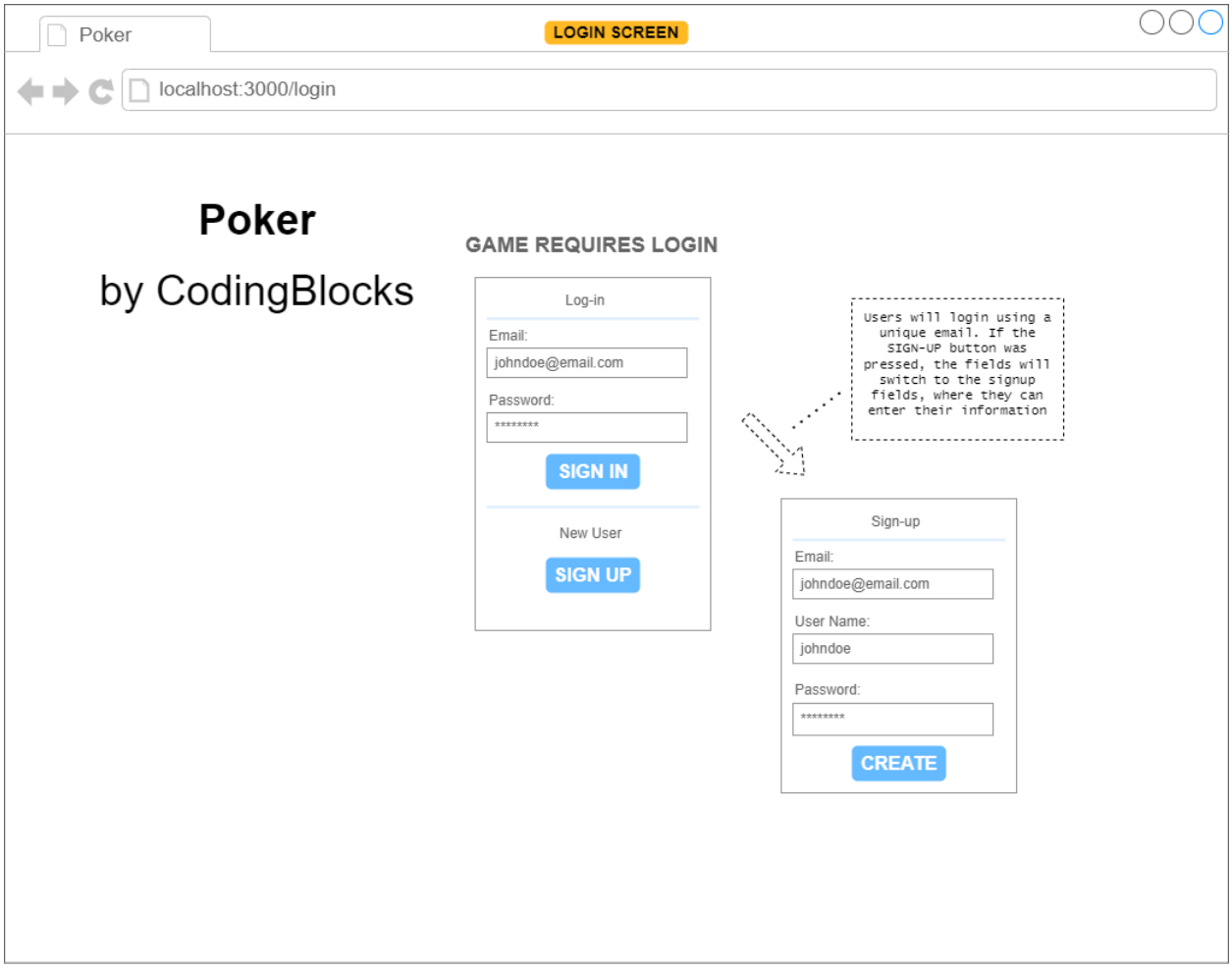
5. **Game Room Management (Optional Lobby Code System)**:
    - **Feature**: Players can join a game via random matchmaking or by entering a code to join a specific friend's game.
    - **Purpose**: Provide flexible game joining options, allowing friends to play together without a traditional lobby system.
    - **Details**: Each game room will have a unique code that players can use to join, which will be managed via the backend.
6. **Poker Betting System**:
    - **Feature**: Implement a full betting system where players can bet, call, check, raise, or fold during each round.
    - **Purpose**: Ensure the core poker experience, with chip management and betting rounds like in a real poker game.
    - **Details**: The system will include pre-flop, flop, turn, and river betting rounds. Players' chip stacks will be updated in real time based on their bets.
7. **Player Actions and Turn-Based System**:
    - **Feature**: Implement a turn-based system where each player has a set amount of time to make their move.
    - **Purpose**: Control the flow of the game, ensuring that each player takes turns and has a time limit to act.
    - **Details**: A timer will prompt players to make decisions in a timely manner. After the timer expires, the game will automatically fold or check for that player.
8. **Card Dealing and Hand Evaluation**:
    - **Feature**: Automatically deal cards to players and evaluate hands at the end of each round.

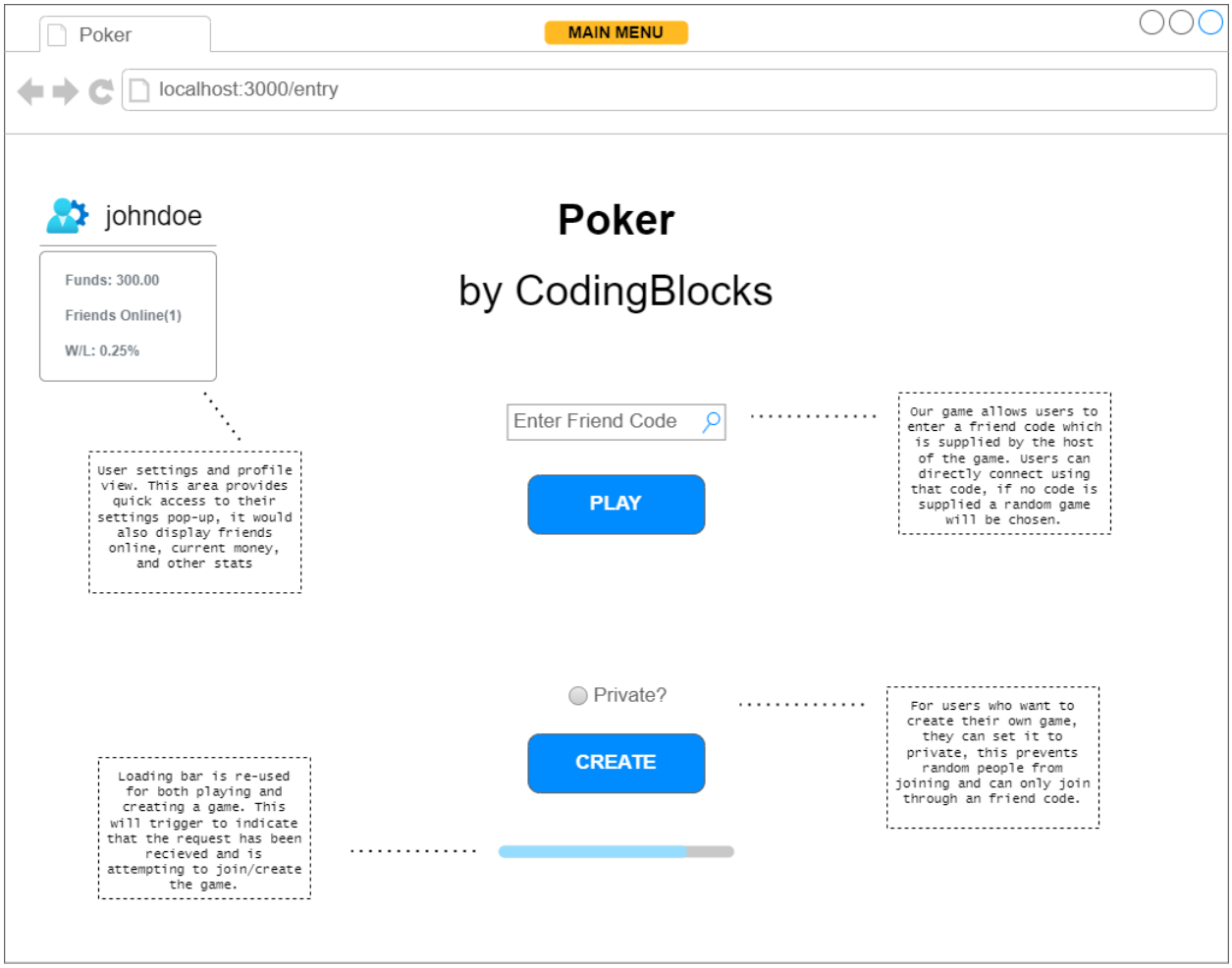
- **Purpose**: Ensure that players are dealt random cards and that hands are evaluated correctly to determine the winner.
- **Details**: Use a standard deck of 52 cards and implement hand evaluation logic (e.g., pair, flush, straight) to determine winners.

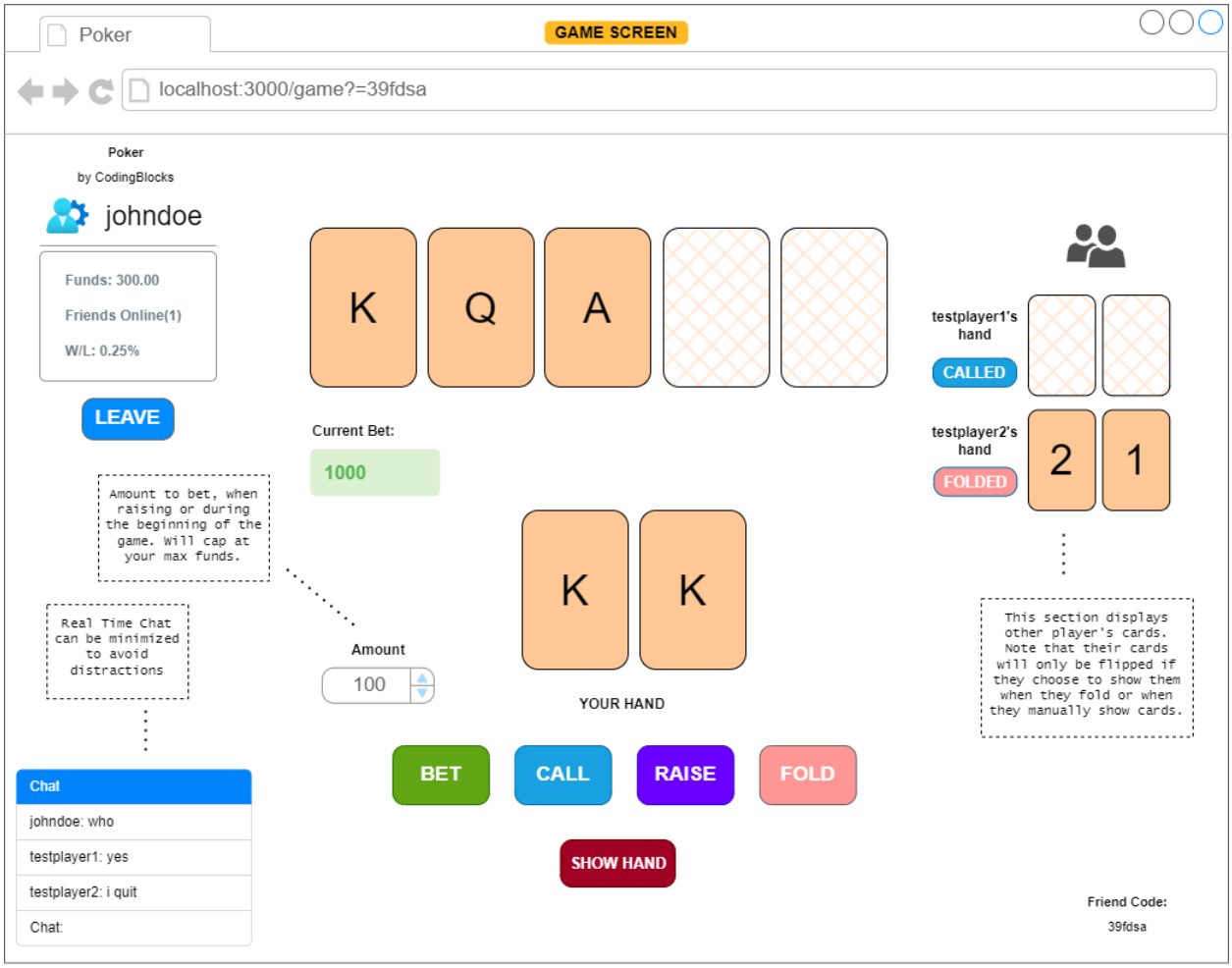
9. **Game Over and Rejoining System**:
    - **Feature**: When a game ends, the system will declare a winner and allow players to start a new game or exit.
    - **Purpose**: End the game properly and allow players to continue playing or leave the game session.
    - **Details**: Once all betting rounds are complete, and a winner is declared, players can either play another round or exit the game.

Required Features:

1. **Backend Server (Node.js with Express.js)**:
    - **Feature**: Serve the static HTML, CSS, and JavaScript files for the game.
    - **Purpose**: Provide the backend to run the game logic and manage players.
2. **Postgres Database**:
    - **Feature**: Store game-related data such as player information, game sessions, and hand history.
    - **Purpose**: Required for maintaining persistent data across game sessions or tracking statistics.
3. **Deployment on Render**:
    - **Feature**: Host the game on Render, making it accessible online.
    - **Purpose**: Required for deployment, making the game available to users publicly.


Poker
LOGIN SCREEN
localhost:3000/login
Poker
by CodingBlocks
GAME REQUIRES LOGIN
Log-in
Email:
johndoe@email.com
Password:
****
SIGN IN
New User
SIGN UP
Users will login using a unique email. If the SIGN-UP button was pressed, the fields will switch to the signup fields, where they can enter their information
Sign-up
Email:
johndoe@email.com
User Name:
johndoe
Password:
****
CREATE


Poker
MAIN MENU
localhost:3000/entry
johndoe
Funds: 300.00
Friends Online(1)
W/L: 0.25%
User settings and profile view. This area provides quick access to their settings pop-up, it would also display friends online, current money, and other stats
Poker
by CodingBlocks
Enter Friend Code
PLAY
Our game allows users to enter a friend code which is supplied by the host of the game. Users can directly connect using that code, if no code is supplied a random game will be chosen.
Private?
CREATE
For users who want to create their own game, they can set it to private, this prevents random people from joining and can only join through an friend code.
Loading bar is re-used for both playing and creating a game. This will trigger to indicate that the request has been recieved and is attempting to join/create the game.


Poker
GAME SCREEN
localhost:3000/game?=39fdsa
Poker
by CodingBlocks
johndoe
Funds: 300.00
Friends Online(1)
W/L: 0.25%
LEAVE
K
Q
A
Current Bet:
1000
Amount to bet, when raising or during the beginning of the game. Will cap at your max funds.
Amount
100
K
K
YOUR HAND
BET
CALL
RAISE
FOLD
SHOW HAND
testplayer1's hand
CALLED
testplayer2's hand
FOLDED
2
1
This section displays other player's cards. Note that their cards will only be flipped if they choose to show them when they fold or when they manually show cards.
Real Time Chat can be minimized to avoid distractions
Chat
johndoe: who
testplayer1: yes
testplayer2: i quit
Chat:
Friend Code:
39fdsa