Poker Game Website

Team JKLS

Today's Agenda

Technical Details and Game Functionality

- Tech Stack
- Authentication and Sessions
- Lobby and Gameroom Chats
- Game State Persistence
- Game Data Flow

Game Demo

Discussions and Insights

- Discussion of Difficulties Encountered
- Discussion of Learnings

Technical Details & Game Functionality

Technologies used for Development

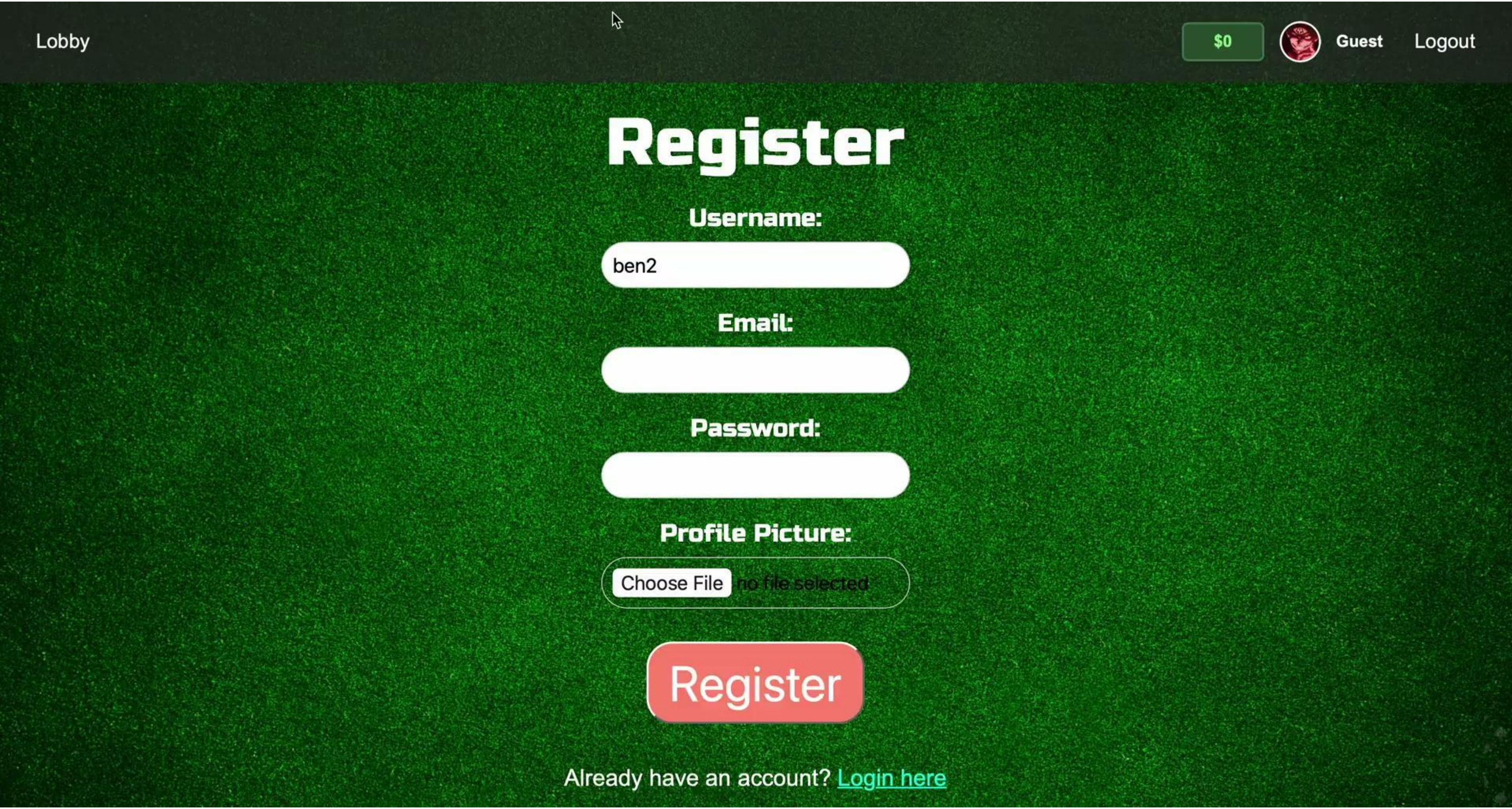
Tech Stack:

- Render
- Node.js
- Express.js
- Postgres
- CSS

Authentication and Sessions

Users can create account

- Register page
- Users need:
    - Username
    - Email
    - Password
    - Profile Picture (optional)



Register

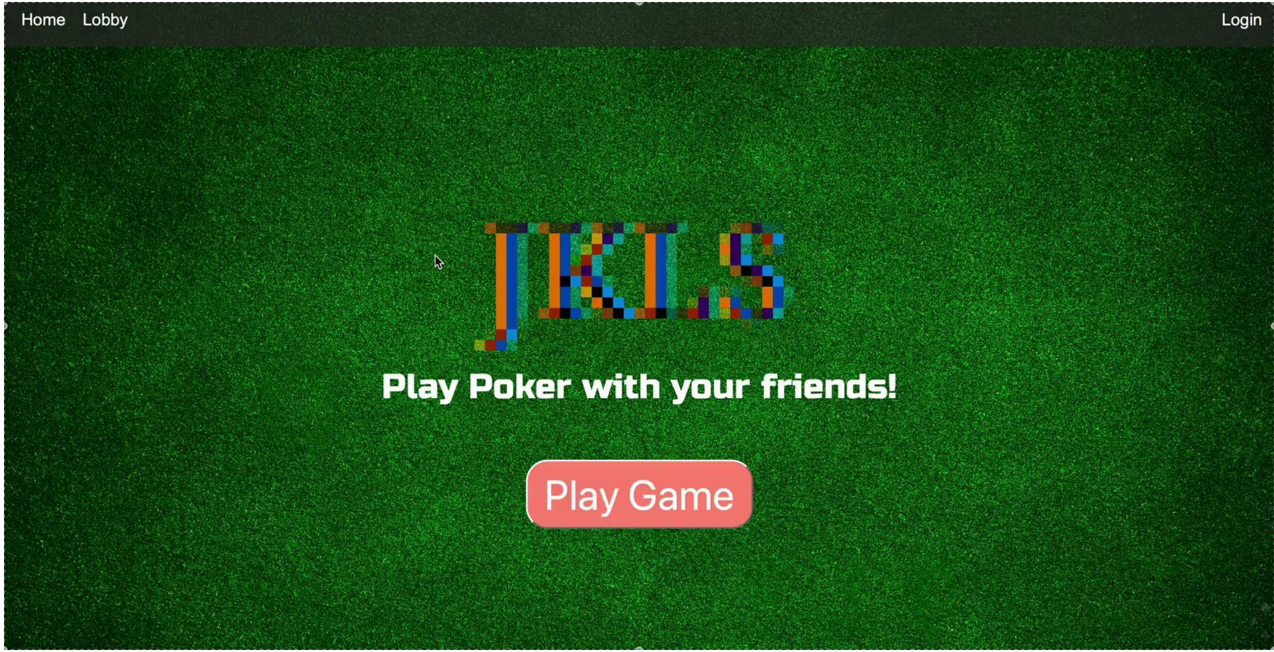
Username: ben2

Already have an account? Login here

Authentication and Sessions

Users can login and logout

- They can login from Home Page and Login Page
- They can logout from Home, Lobby, and Game rooms

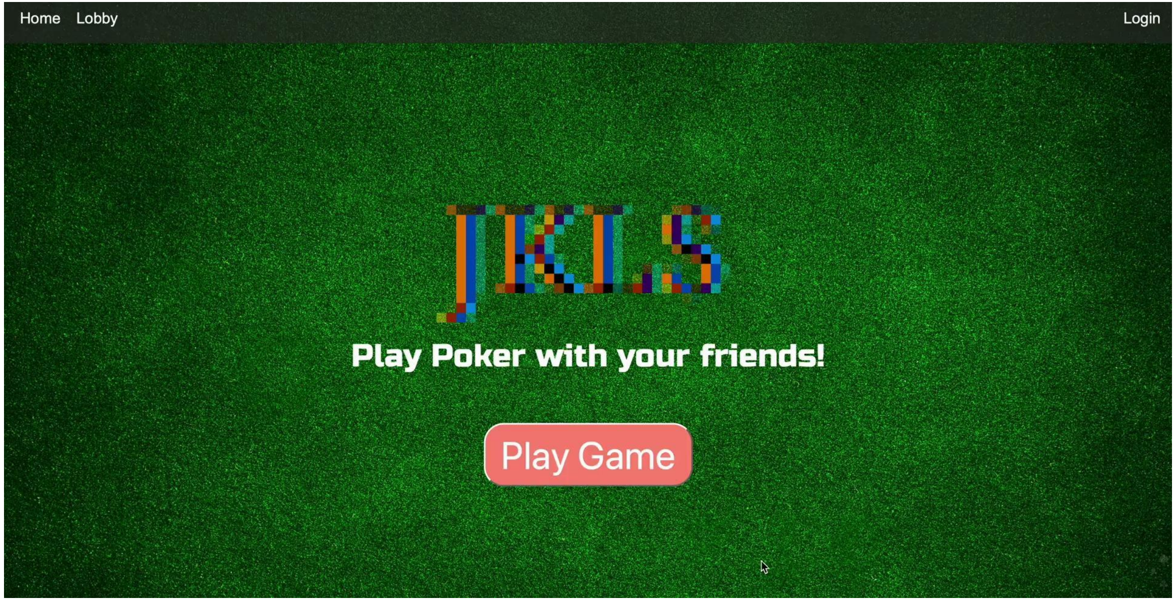


Play Poker with your friends!

Authentication and Sessions

Account is protected from pages it shouldn't have access to. For example,

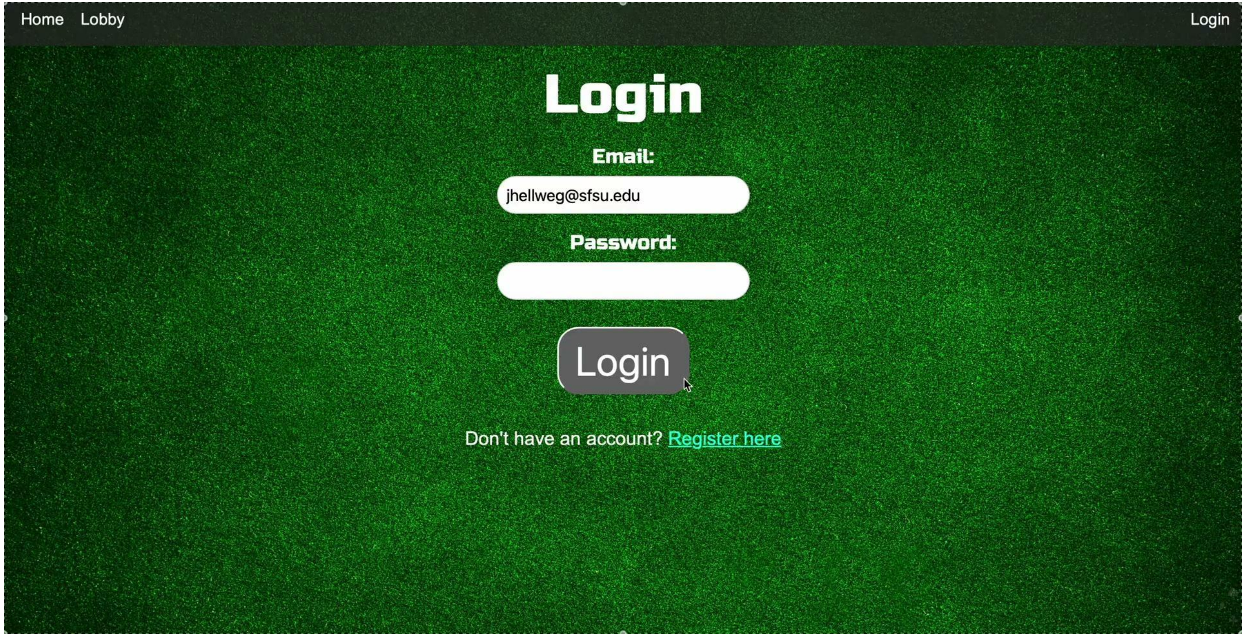
- Lobby Page redirects to the Login page if user is not logged in.



Authentication and Sessions

Account is protected from pages it shouldn't have access to. For example,

- Need to enter correct passwords to login and enter a private game that has a password



Login

Email: jhellweg@sfsu.edu

Password:

Don't have an account? Register here

Lobby Chats

Gameroom Chats

Game State Persistence

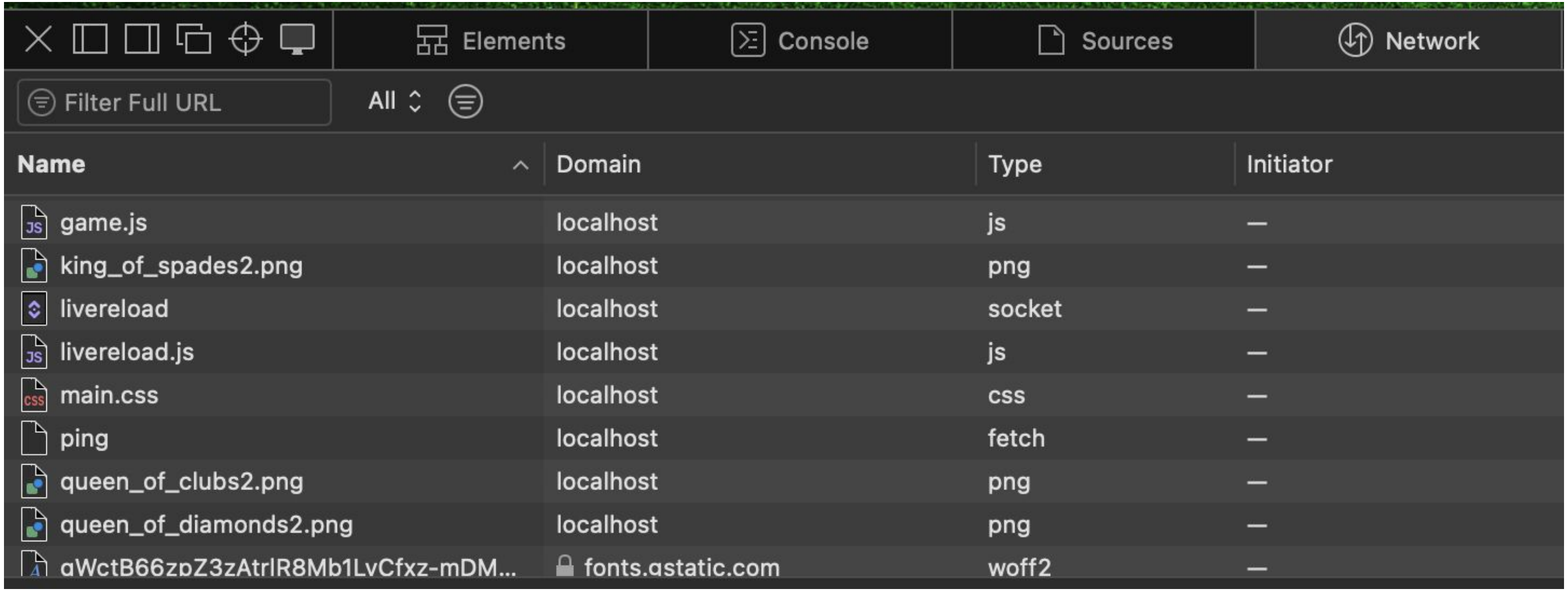
You can leave a game or log out, and then rejoin the room (after logging in) and it will save the state of the game.

- Next page shows an example that it saves whether a game has been started or not.

Game Data Flow

Server sends HTML pages containing CSS, JS, etc.

- Network Screenshot:

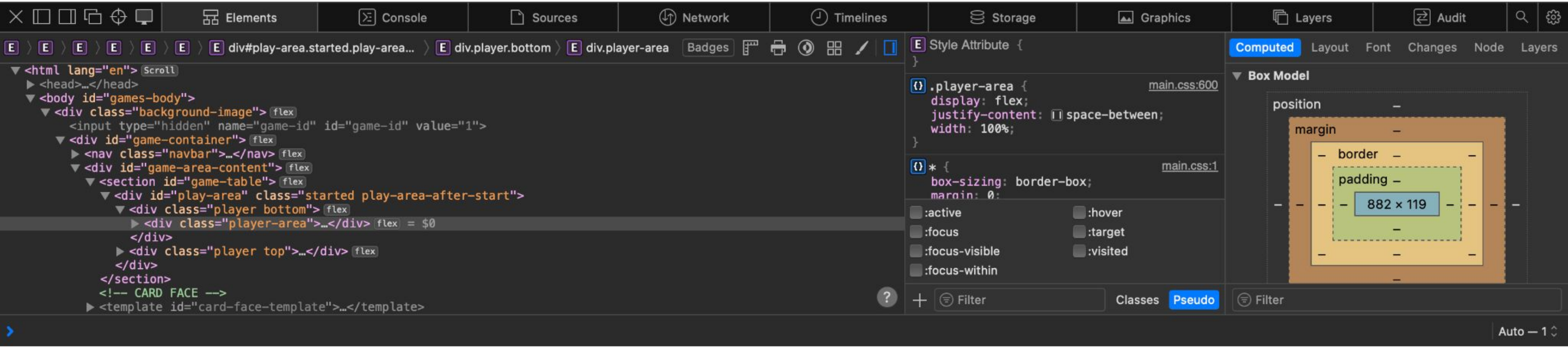


Name	Domain	Type	Initiator
game.js	localhost	js	—
king_of_spades2.png	localhost	png	—
livereload	localhost	socket	—
livereload.js	localhost	js	—
main.css	localhost	css	—
ping	localhost	fetch	—
queen_of_clubs2.png	localhost	png	—
queen_of_diamonds2.png	localhost	png	—
qWctB66zpZ3zAtrlR8Mb1LvCfxz-mDM...	fonts.gstatic.com	woff2	—

Game Data Flow

Server sends HTML pages containing CSS, JS, etc.

- Inspect Element Screenshot:



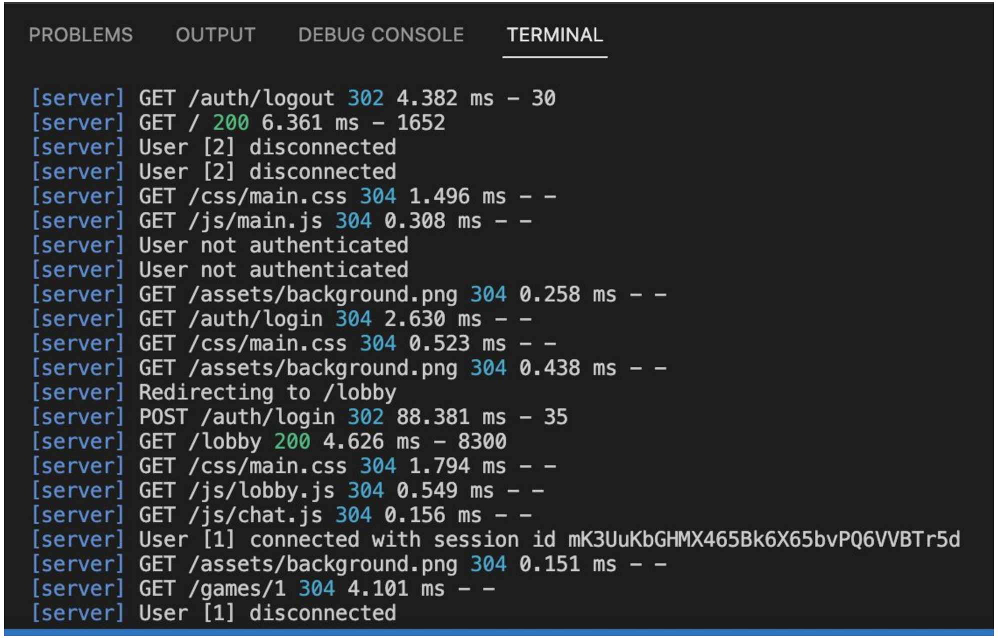
```
▼ <html lang="en"> Scroll
  ▶ <head>…</head>
  ▼ <body id="games-body">
    ▼ <div class="background-image"> flex
        <input type="hidden" name="game-id" id="game-id" value="1">
      ▼ <div id="game-container"> flex
        ▶ <nav class="navbar">…</nav> flex
        ▼ <div id="game-area-content"> flex
          ▼ <section id="game-table"> flex
            ▼ <div id="play-area" class="started play-area-after-start">
              ▼ <div class="player bottom"> flex
                ▶ <div class="player-area">…</div> flex = $0
                </div>
              ▶ <div class="player top">…</div> flex
              </div>
            </section>
            <!-- CARD FACE -->
          ▶ <template id="card-face-template">…</template>
```

```
E Style Attribute {
}
{} .player-area {                              main.css:600
  display: flex;
  justify-content: space-between;
  width: 100%;
}
{} * {                                         main.css:1
  box-sizing: border-box;
  margin: 0;
```

Game Data Flow

State changes are submitted asynchronously to the server using HTTP requests. The server is the source and the client doesn't hold any state information.

- Terminal Log:



```
[server] GET /auth/logout 302 4.382 ms - 30
[server] GET / 200 6.361 ms - 1652
[server] User [2] disconnected
[server] User [2] disconnected
[server] GET /css/main.css 304 1.496 ms - -
[server] GET /js/main.js 304 0.308 ms - -
[server] User not authenticated
[server] User not authenticated
[server] GET /assets/background.png 304 0.258 ms - -
[server] GET /auth/login 304 2.630 ms - -
[server] GET /css/main.css 304 0.523 ms - -
[server] GET /assets/background.png 304 0.438 ms - -
[server] Redirecting to /lobby
[server] POST /auth/login 302 88.381 ms - 35
[server] GET /lobby 200 4.626 ms - 8300
[server] GET /css/main.css 304 1.794 ms - -
[server] GET /js/lobby.js 304 0.549 ms - -
[server] GET /js/chat.js 304 0.156 ms - -
[server] User [1] connected with session id mK3UuKbGHMX465Bk6X65bvPQ6VVBTr5d
[server] GET /assets/background.png 304 0.151 ms - -
[server] GET /games/1 304 4.101 ms - -
[server] User [1] disconnected
```

Poker Game Demo

Discussions and Insights

Discussion of Difficulties Encountered

- **Scheduling Conflicts**

  Different class times and workloads made it hard to coordinate meetings and collaborate consistently

- **Unclear Milestones**

  Instructions on expected progress were vague, leading to uneven pacing and last-minute updates

- **Personal Challenges**

  Team members faced health issues, family responsibilities, and job obligations that affected availability

- **Communication Gaps**

  Inconsistent updates and unclear task delegation caused confusion and doubled effort

Discussion of Learnings

- **WebSockets**
  Learned how to implement real-time communication for live gameplay using WebSocket connections

- **Middleware**
  We gained more experience with using middleware to handle authentication, logging, and request processing in a modular way

- **Database Migrations**
  We understood the importance of migrations for managing schema changes, syncing environments, and preserving data integrity