

## Team UNO A

Dante Vervelli (921458418)

Jijeong Lee (922960763)

Ruxue Jin (923092817)

Xu Gu (923662217)

# 1. How to run the project:

**Step 1:** install node.js >=16

**Step 2:** add a .env file in the path: [root\_directory]/application/backend/.env

```
PORT=3000

# Database
DB_HOST=dpg-colg04i1hb1s738u9cp0-a.oregon-postgres.render.com
DB_USER=root
DB_PASSWORD=K2QkWxnbfxBYDRrNWJJpcHDmBJbtE7bV
DB_NAME=csc667uno

# JWT
JWT_SECRET_KEY=csc667aunojijiongruxuedantexu

NODE_ENV=development

# Cors Configuration
CORS_ALLOWED_ORIGIN=*
```

**Step 3.** Start Backend

```
cd application/backend
npm install
npm start
```

**Step 4.** Start frontend

```
cd application/frontend
npm install
node index.js
```

## 2. Challenges:

- Changing from local-storage to session-storage so we can have multiple tabs open each playing a unique game
- Reconnecting logic and working with sockets disconnecting and reconnecting upon page loading/redirecting.
- Socket.io became hard to follow, changing the names of the socket events helped us understand the flow

### 3.Presentation Slides:

# Final Presentation

Team A UNO  
Dante Vercelli (921458418)  
Jijeong Lee (922960763)  
Ruxue Jin (923092817)  
Xu Gu (923662217)

## INDEX

1. Game Overview
2. Group Description
3. User Authentication
4. Database
5. SocketIO
6. Demo

## 1. Game Overview - UNO

Uno is a popular card game that is played with a specially printed deck. The game is designed for 2 to 6 players. The objective of Uno is to be the first player to play all the cards in their hand.

We implemented our project based on the basic uno game rules.



## 2. Group Description

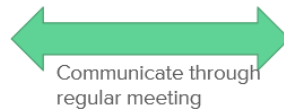
### Front-end team

#### Jijeong

- Sign In
- Game Lobby - Room List
- Game Room - Players & Animation

#### Ruxue

- Sign Up
- Game Lobby - Create room
- Game Room - Game Logic



### Back-end team

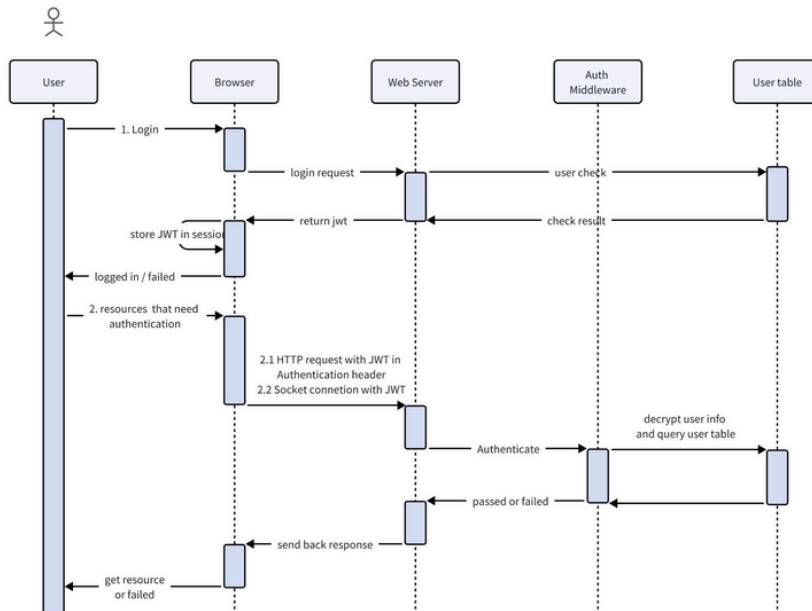
#### Xu

- Sign In / Up API
- Chat system
- Game Logic API

#### Dante

- Database Design
- Create & Join game room API
- Game Logic API

### 3. User Authentication

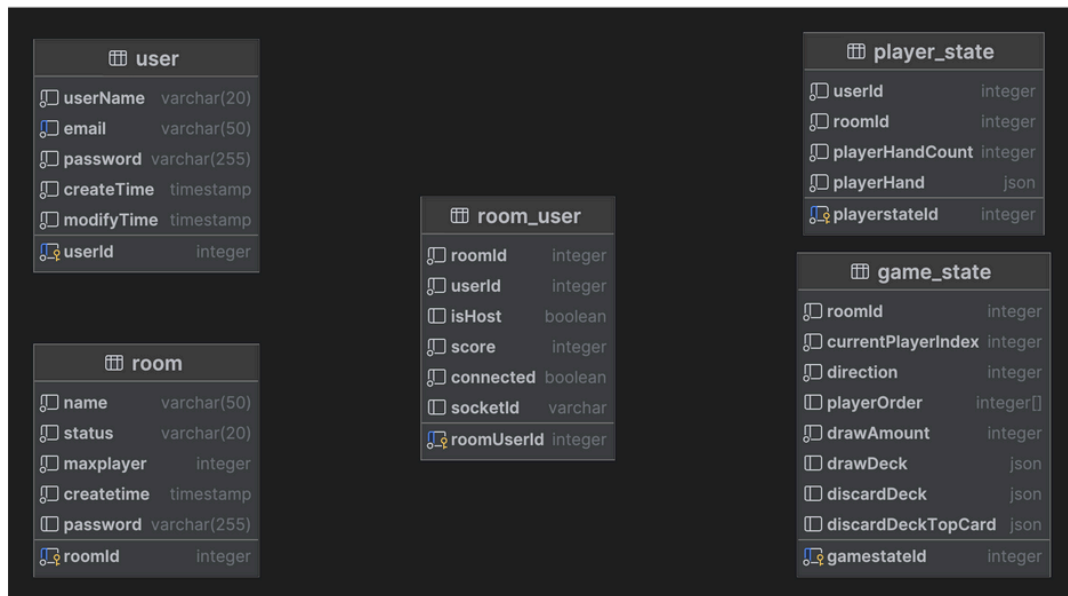


We use **JWT** to implement user authentication.

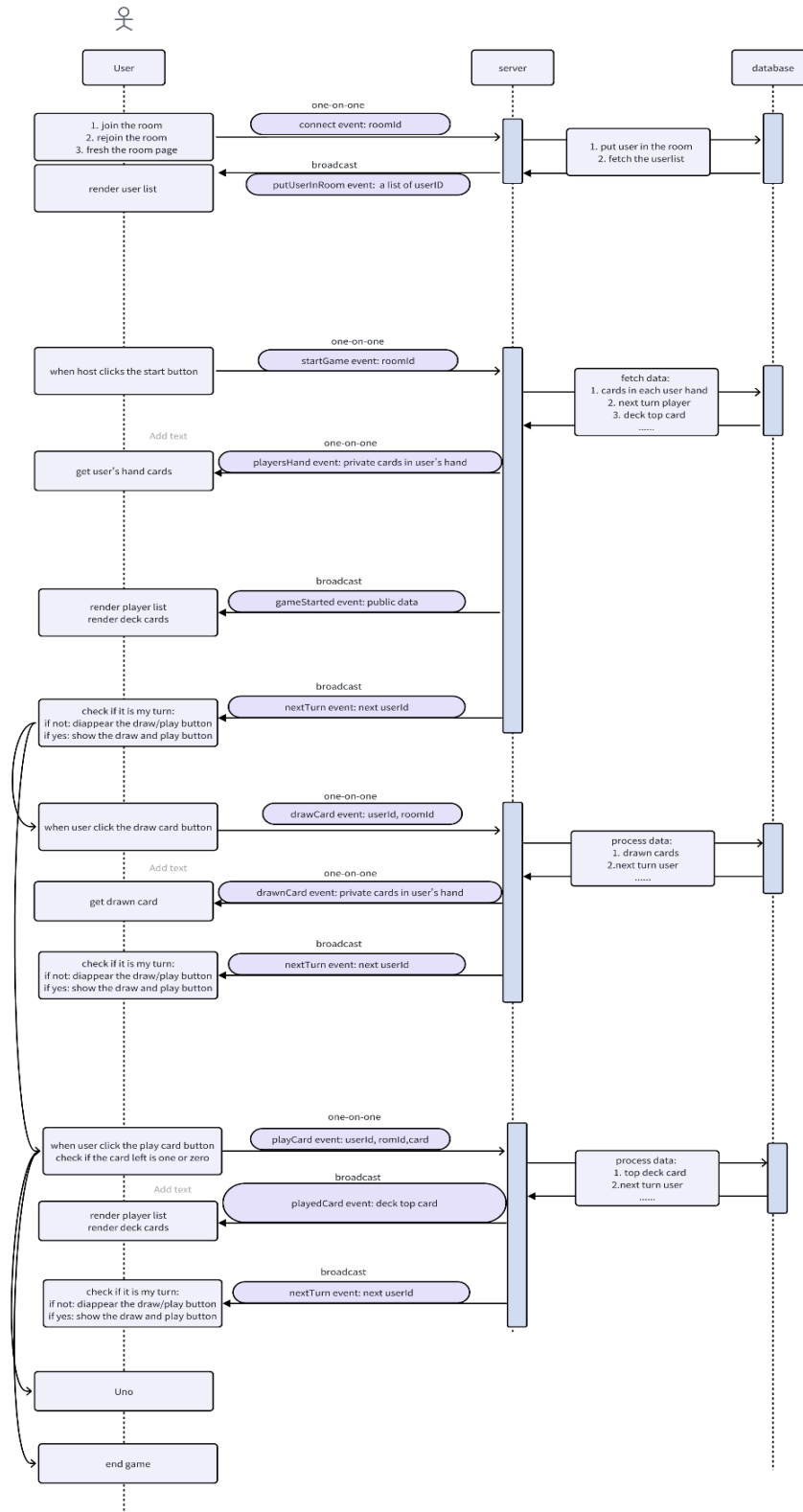
API requests should have JWT in the Authentication header.

Sockets also need to bring the JWT to build connections

### 4. Database



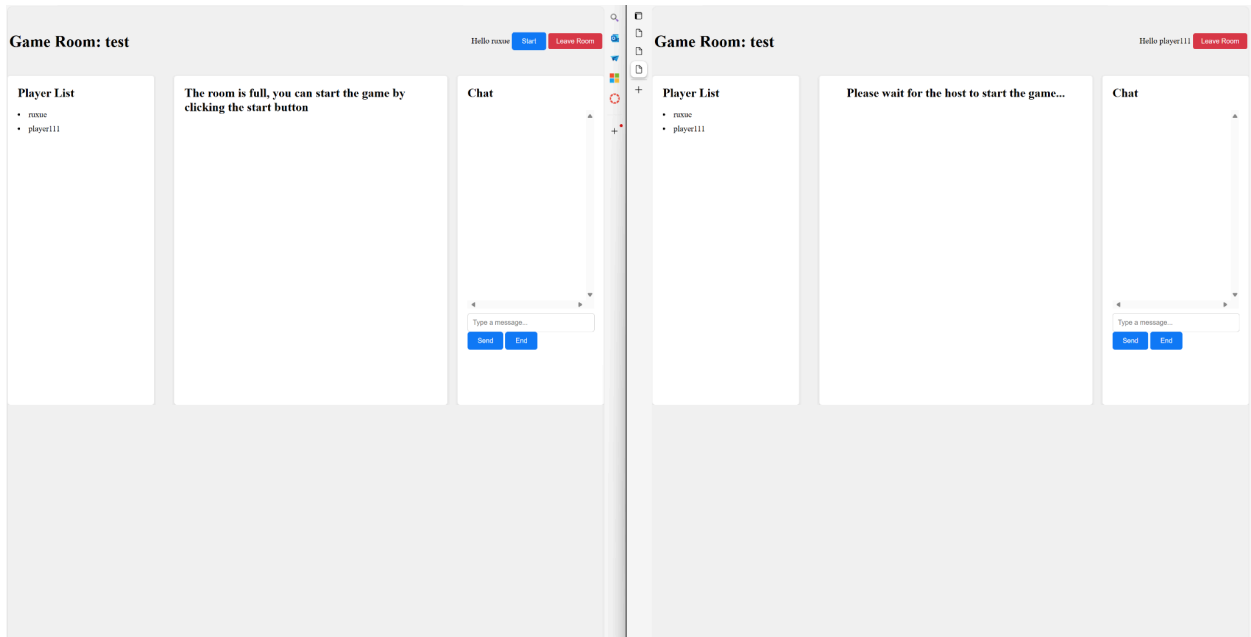
## 4. Socket design:



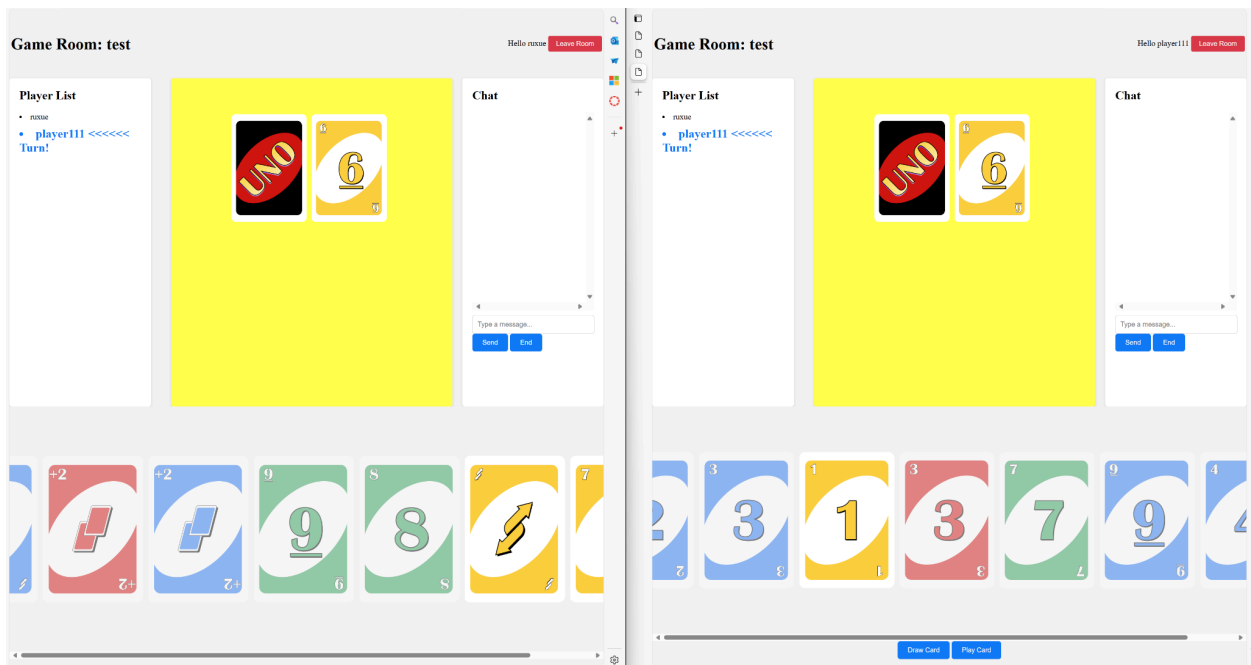
## 5.Demo makeup material

During the presentation, we did not show some game logic, including the wild card choose color feature, Uno feature, and end game feature. We include the screenshot here.

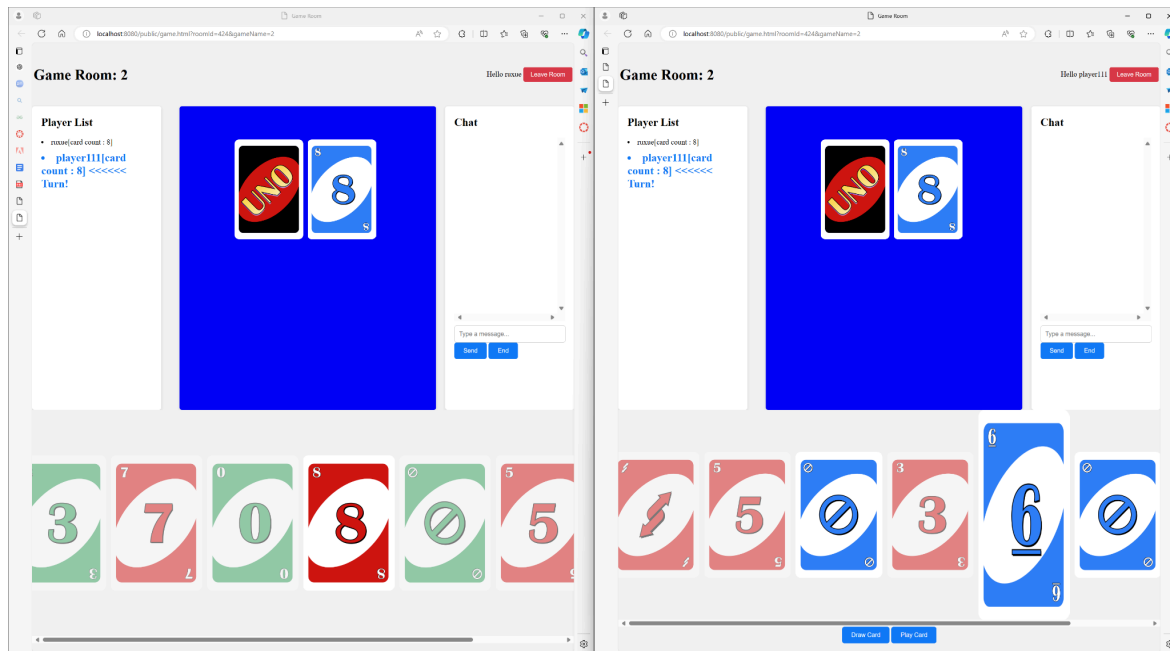
- Host and guest have different prompts. When the room is full, the host will have the start button.



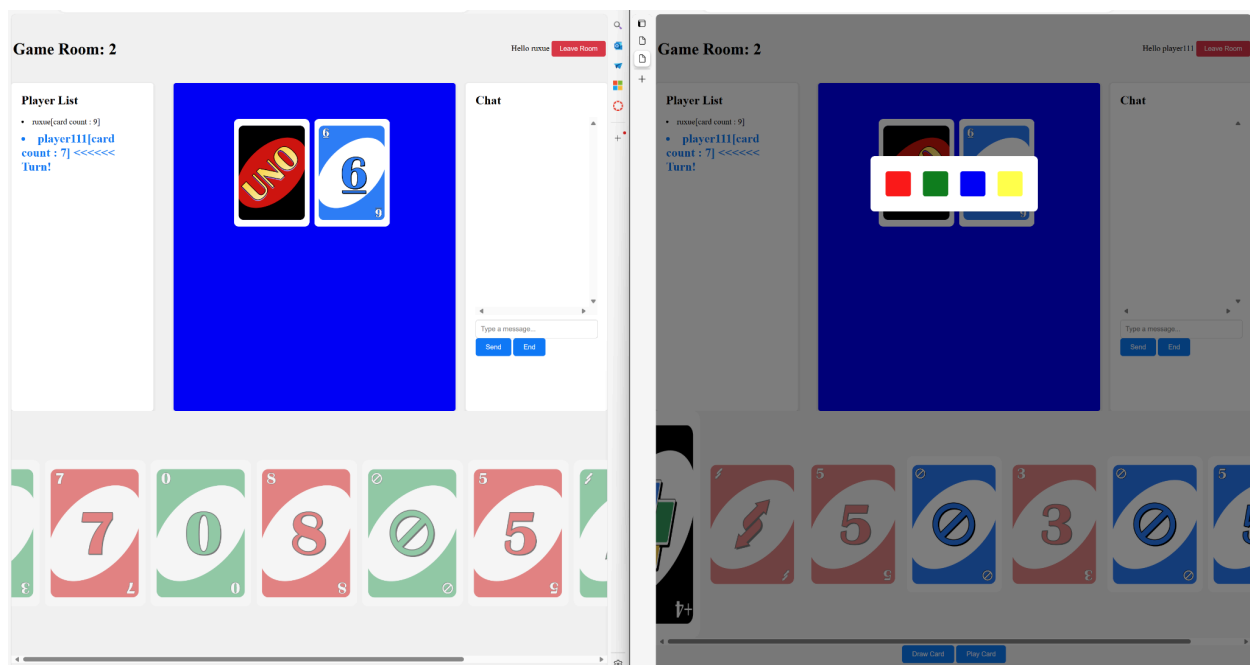
- Game start



- The unplayable cards are grayed out and unclickable.

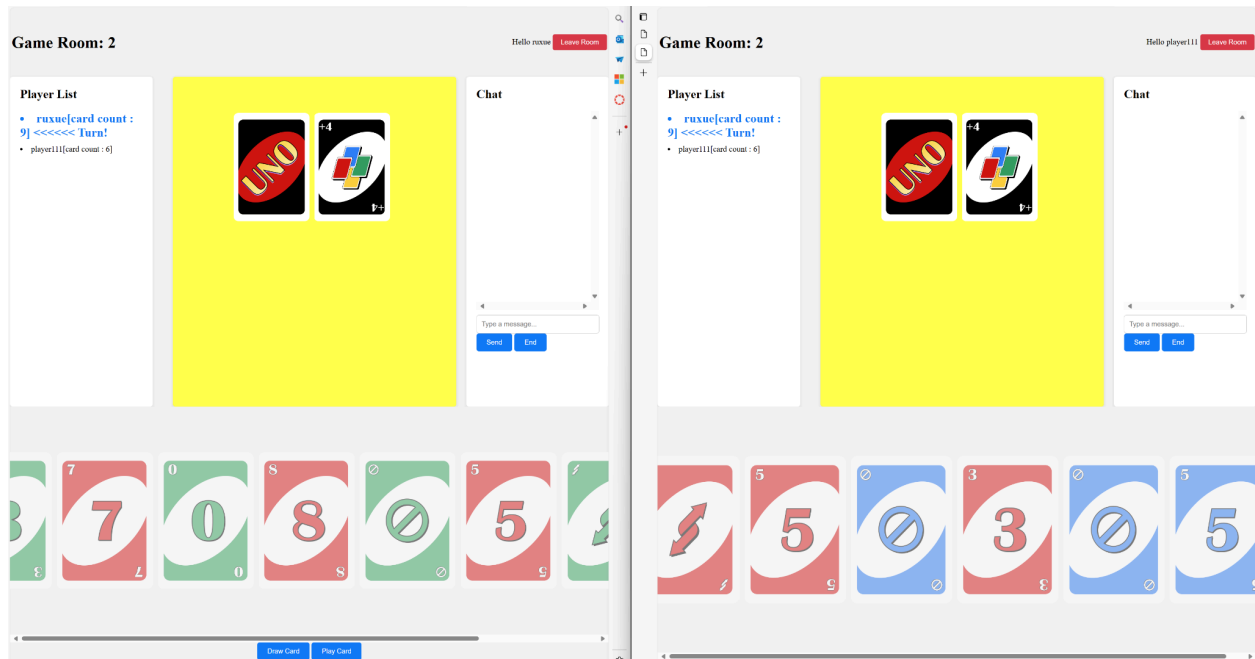


- The wild card feature : allows user to choose one color

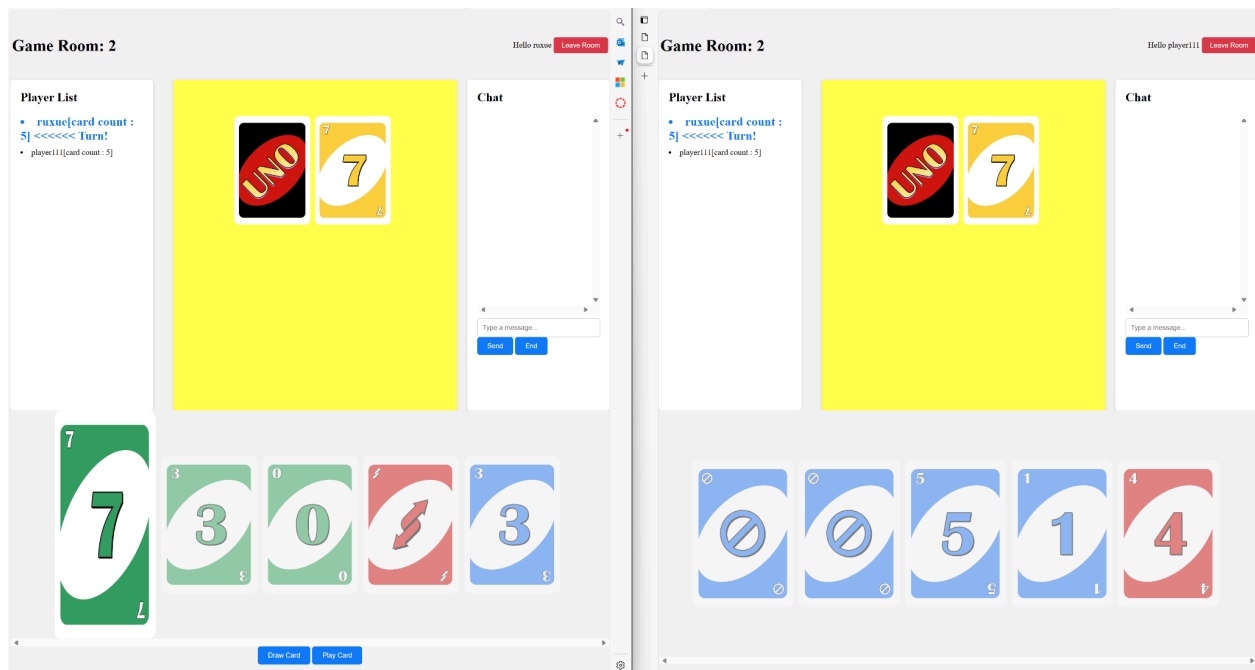


After choosing yellow color

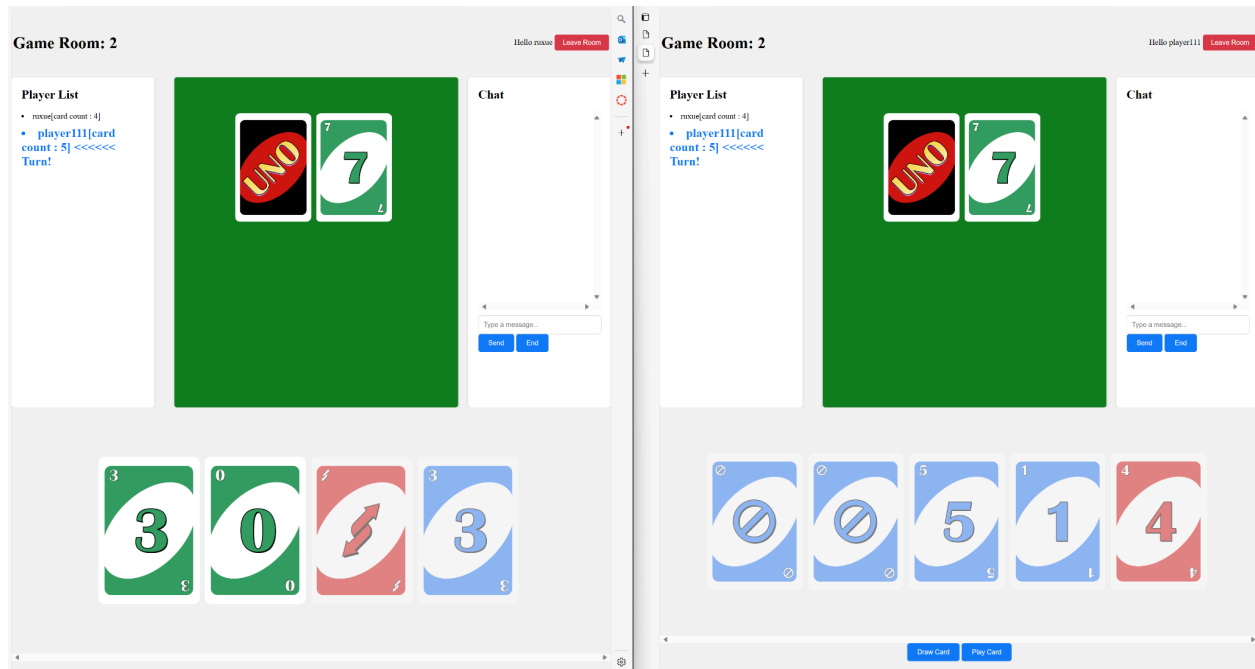




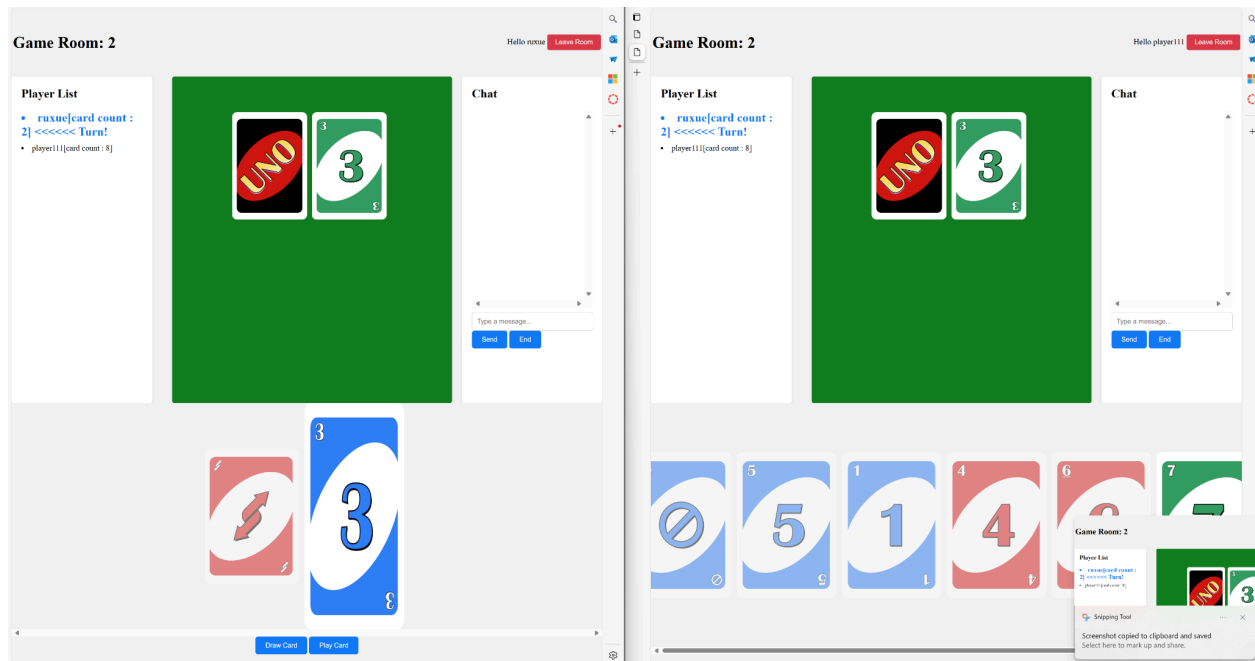
The left user is about to choose green 7



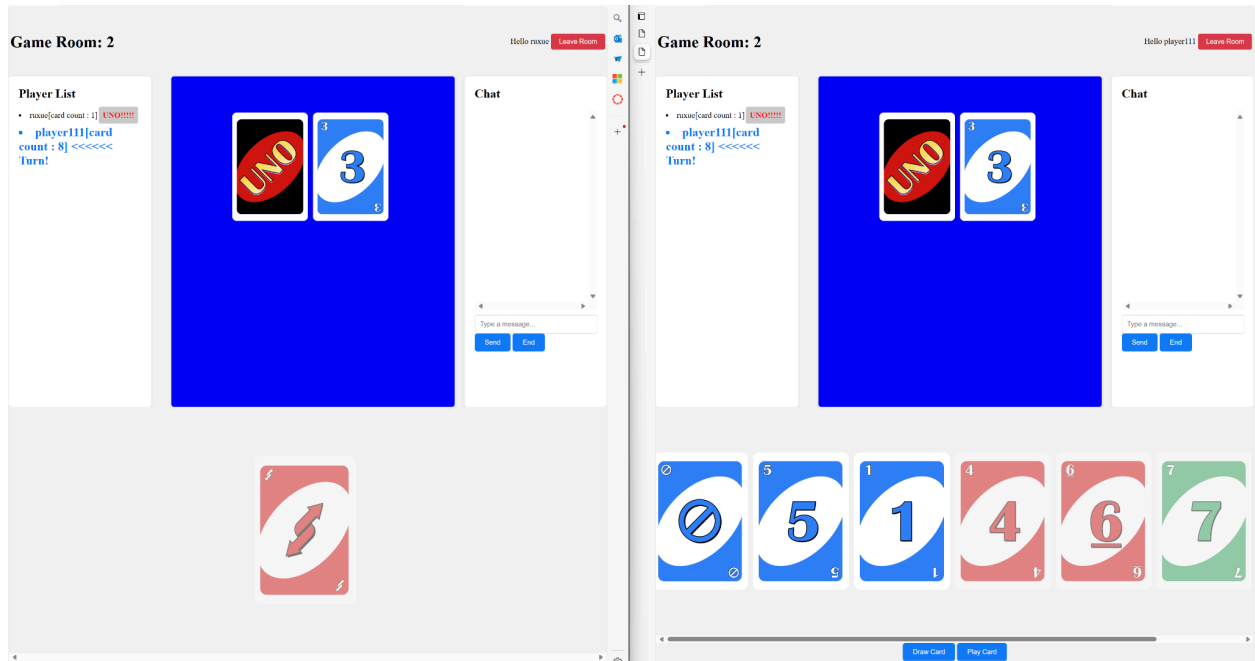
- After playing the green 7 card, background changes, who's turn changes, card count changes.



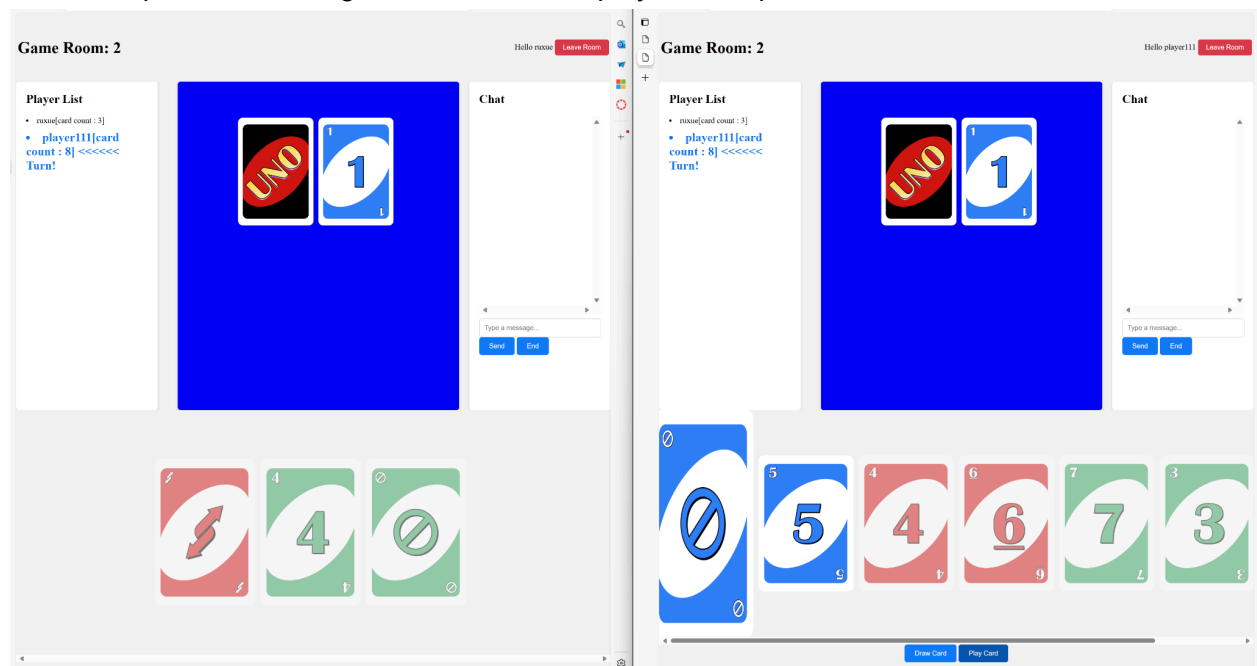
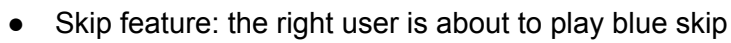
- Uno feature: The left user is about to Uno



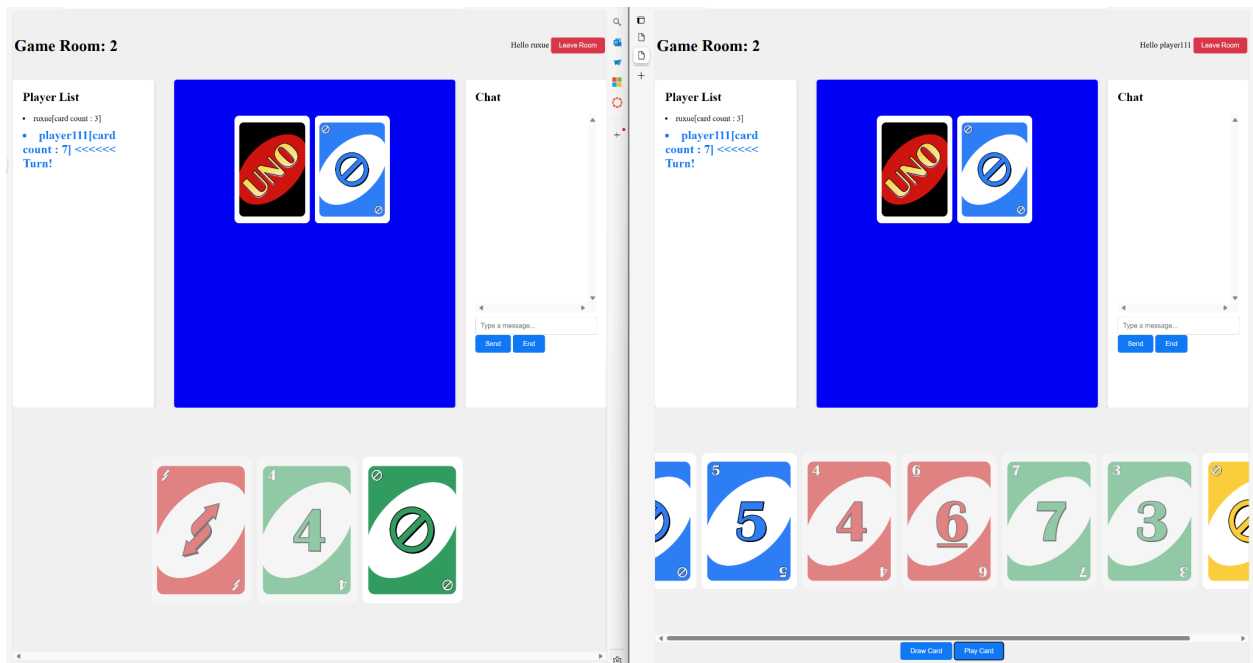
The left user Uno!



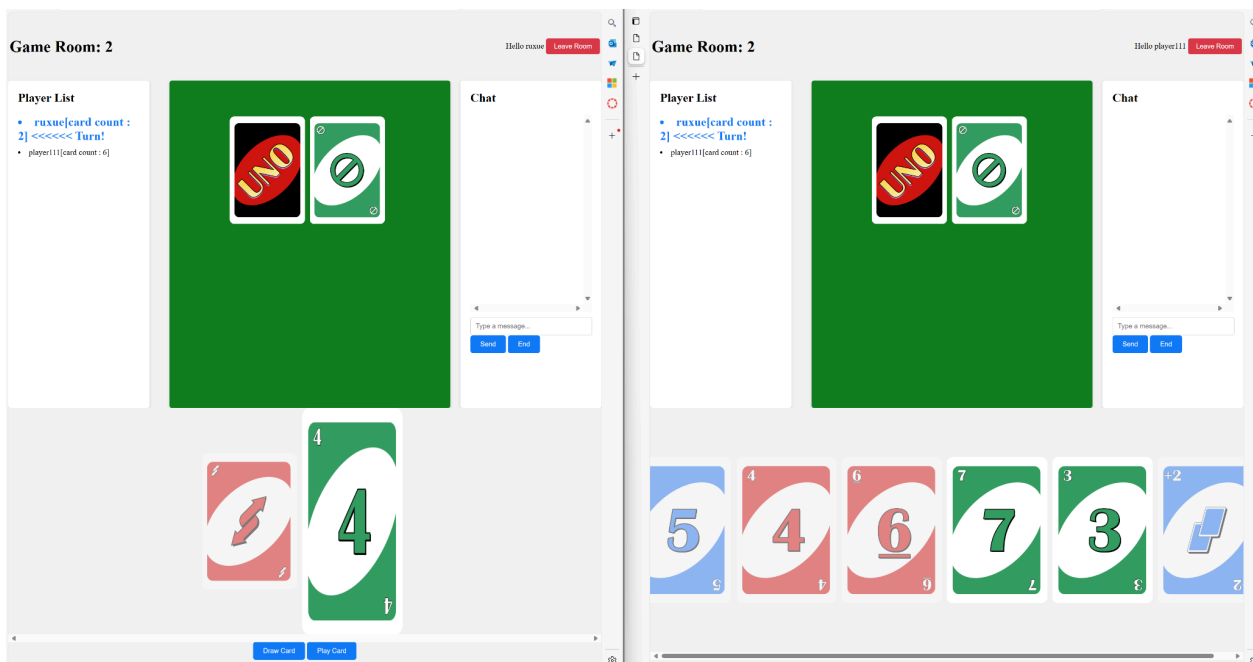
After draw another card, the left user does not uno



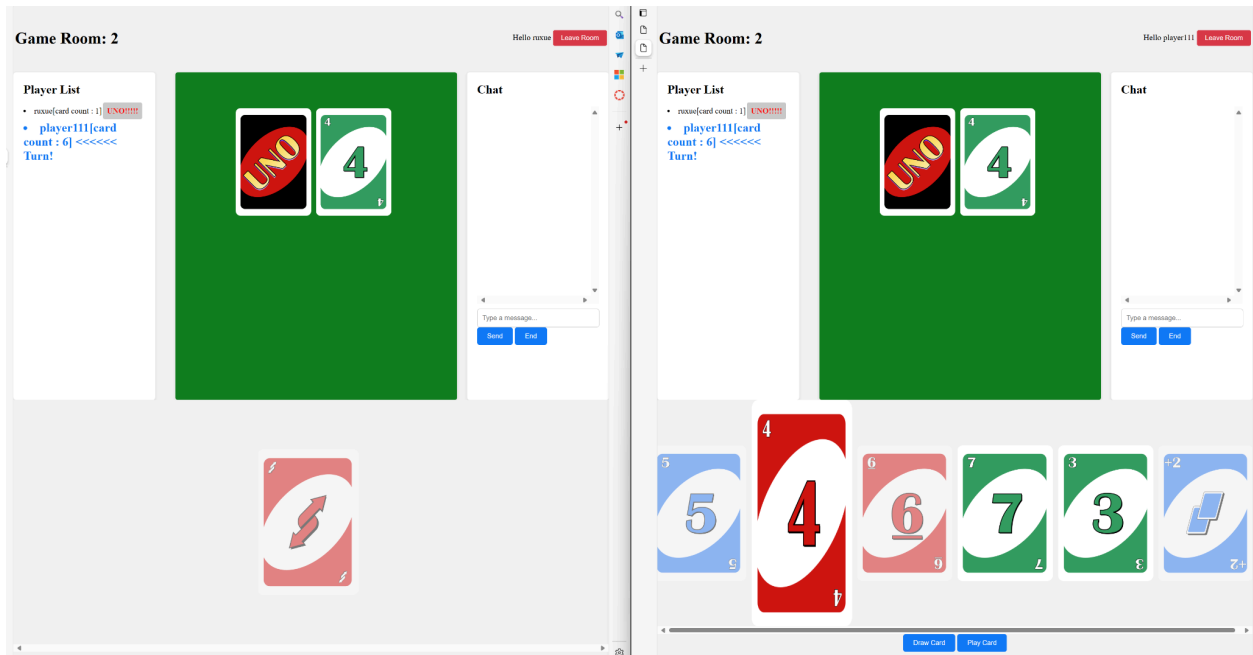
It skips the left user's turn, so it is again for the right user to play/draw.



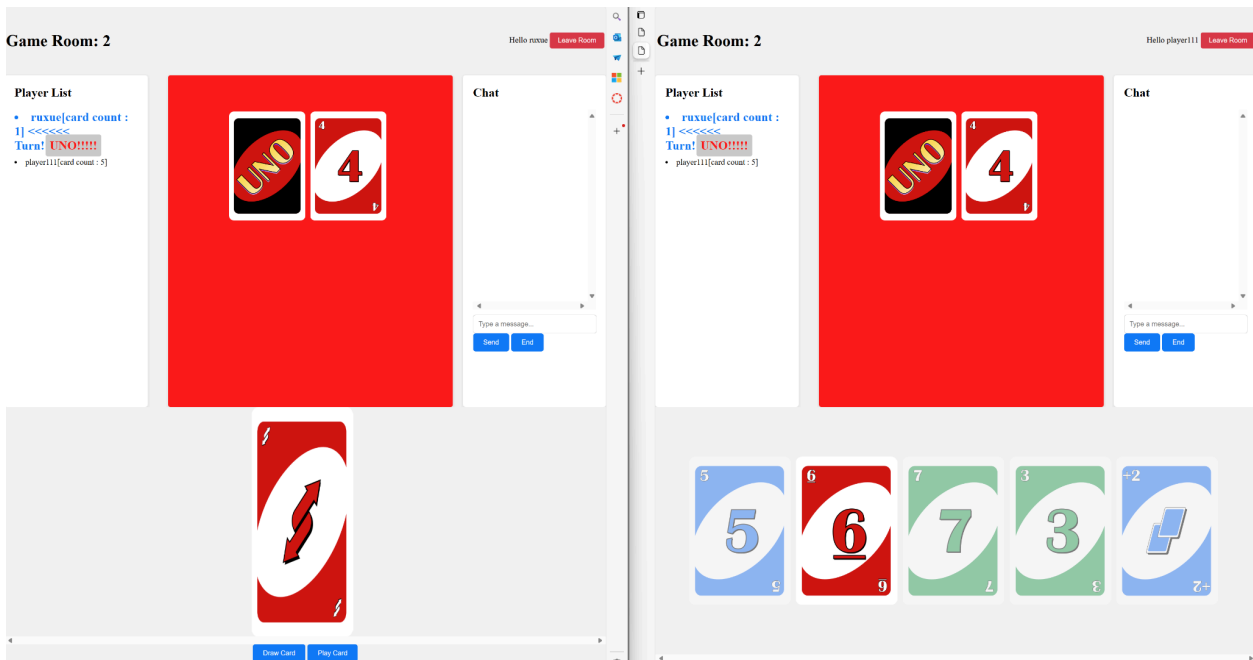
- End game feature  
The left user is about to Uno



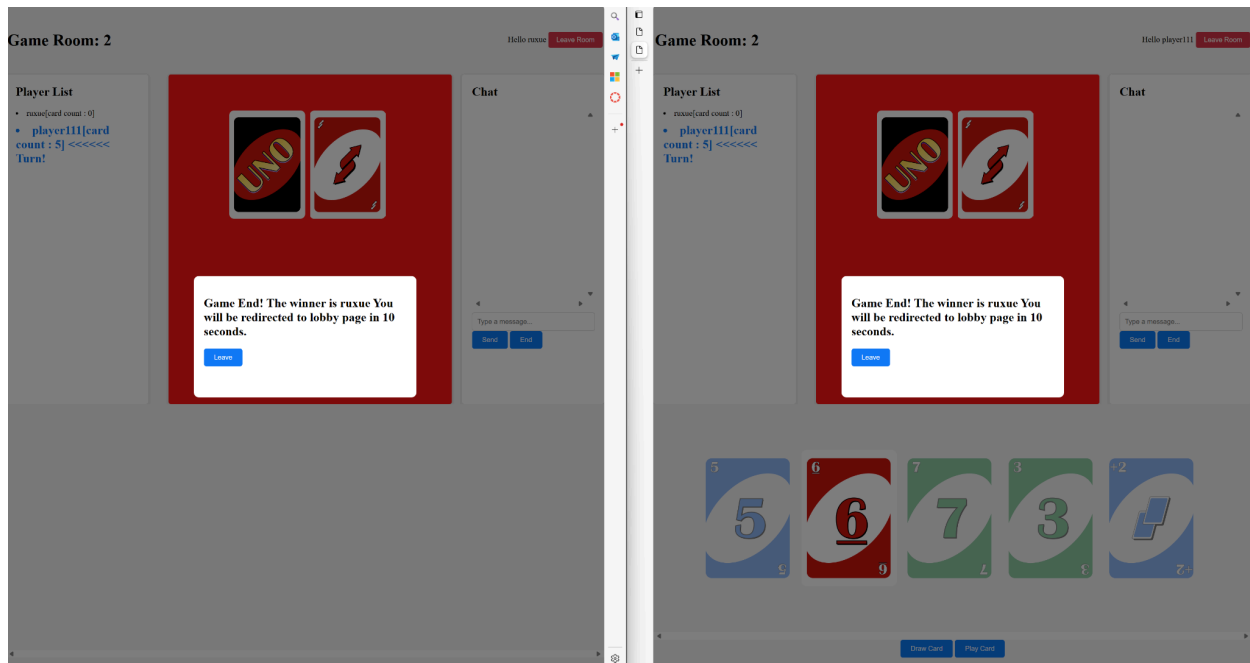
The right user choose red 4 to change to color



Change color



User Ruxue wins the game.



Winner will stay in the room, with no Join button. Loser will have the choice to join the room or not.

