Milestone 3: Application Functionality

Team E : Bingo

Team members:
Ekarat Buddharuksa - 922254694
Danish Nguyen - 923091933
Karl Xavier Arcilla - 922536937
Mya Phyu - 921759134

Team repo: https://github.com/sfsu-csc-667-spring-2024-roberts/term-project-team-e-bingo
Up-to-date branch: hotfix/ms3

Game description

Bingo is a classic, easy-to-play game that can be enjoyed by people of all ages. In our application, the maximum number of players allowed at one time is four. Each player receives a bingo card, which is a 5x5 grid filled with random numbers. Numbers are drawn randomly from a pool ranging from 1 to 75 and announced to the players. These numbers will also be displayed on the side of the screen for rechecking. Players mark these numbers on their cards if they appear. The goal of the game is to be the first to complete a pattern on the card, which includes a straight line horizontally, vertically, or diagonally. When a player achieves the required pattern, they can press the "Bingo!" button to stop the game. Their card is then checked for accuracy, and If correct, they are declared the winner.

Group Meetings and Collaboration

Our team meets regularly, typically 1-2 times per week, with each meeting lasting about one to two hours. We discuss our progress, plans, tasks, debugging and troubleshooting. If a group member is stuck on their tasks, we break down the problems and come up with solutions.

Team members' contribution:

Mya:

Works on responsive web UI pages and refining the visual appeal across four pages: registration, login, lobby and waiting room. In the registration/login section, she focuses on creating a dynamic and engaging experience. She implements a stylish navbar using animation, bootstrap, CSS for easy navigation and employs flip cards to add an interactive element to the login process. These flip cards not only enhance the visual appeal but also provide a user-friendly way for users to access registration and login options. Additionally, she pays close attention to the background design throughout the pages by carefully selecting colors, patterns, or imagery.

Danish:

Combined the front-end design with the back-end code for the lobby. It was important for him to understand both the design and how the back-end works to ensure they fit together well. While migrating the frontend, he simplified some database and server scripts to make sure everything worked smoothly. He made the code easier to maintain by breaking it into smaller functions and avoiding complicated loops and conditions. He created functions that generate new elements for the frontend in middleware JavaScript. His challenging part was creating a player element with a username, status, buttons, and color to show their status. He updated the check to see if there were enough players in a room before starting the game. For example, the game needed at least 2 to 4 players, and each player had to be ready before starting. He worked with his teammates to decide on the rules for creating user rooms and then implemented these rules in the code. Users couldn't create multiple rooms at once; they could only create one and either wait for more players or return to the lobby to invite others. He designed a simple game page using HTML and CSS for his teammates to start working on. Later, he merged the game page with its back-end functions into the branch he was working on. He fixed some small issues in the front end where elements weren't being found when clicked. He also set up scripts on the server to emit all cards' data and update the cards interface for all users in the same room when any player clicked on a cell. He also improved the entire design of the game page for user experience. Lastly, he created a plan on the server's notes so everyone could see it and give feedback.

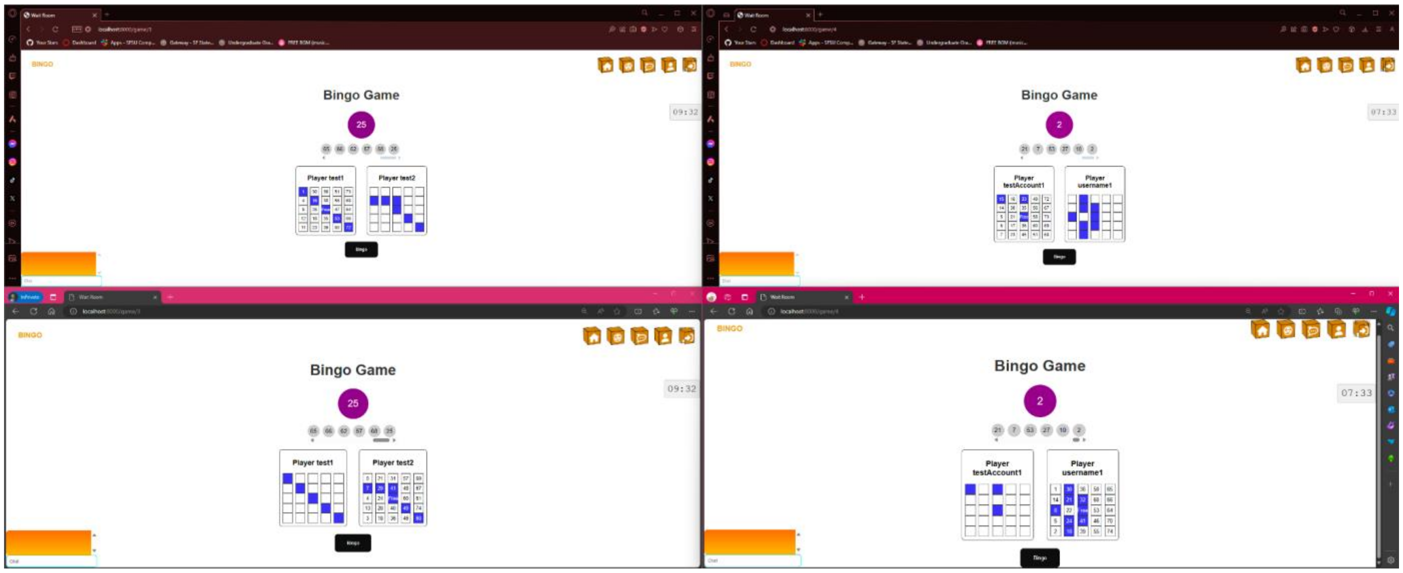
Ekarat

He focused on integrating the backend and connecting it with the frontend across the Lobby, Waiting Room, and Game Page. This task involves managing sockets and pages, as well as implementing rejoining capabilities and more. He laid the groundwork for the socket architecture, provided explanations, and set examples of how sockets operate. He and Karl collaborated to properly configure the database, API, and router, and to establish connections between the backend, frontend and the database. Additionally, he is ensuring that all events on the DOM are synchronized across different users, even if they leave the page and return, including the timer, marked cells, called numbers, and ready statuses. He also assisted in the frontend team in connecting with the socket and have set up separate socket rooms for chat purposes.

Karl

Assigned to manage the database: make the table for specific data, make sure the constraints works fine, make sure the query are inserting or deleting properly, and make sure the foreign keys reference keys properly. Helped Ekarat with making the query functions so the backend communicates with the database properly. Converted the *html* file Danish made and set up the *ejs, js, and css* files for the game page. Properly assigned the randomly generated numbers to the right cells so the card looks like a proper bingo card. Made the page only show the generated numbers for the specific user. Other players' cards will not show their generated number. Added the marking functionality and the users can only mark their own cards. Setup the timer and fixed the drawn number generation. Helped with syncing the drawn numbers and made sure only the host runs the draw number function.

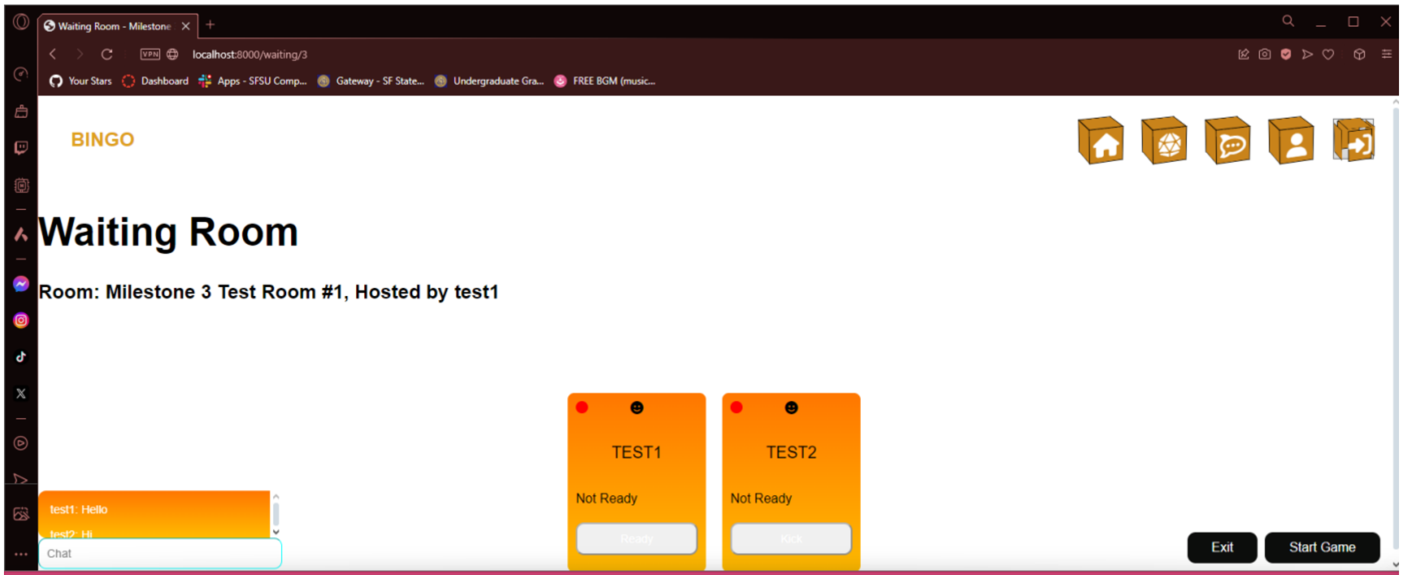
<u>Screenshot of gameplay from two rooms:</u>
Proof of two rooms/games running simultaneously.



Chat Test in Waiting Room:

**Chat is at the bottom left of the page.*

Room 1:



Waiting Room

Room: Milestone 3 Test Room #1, Hosted by test1

TEST1

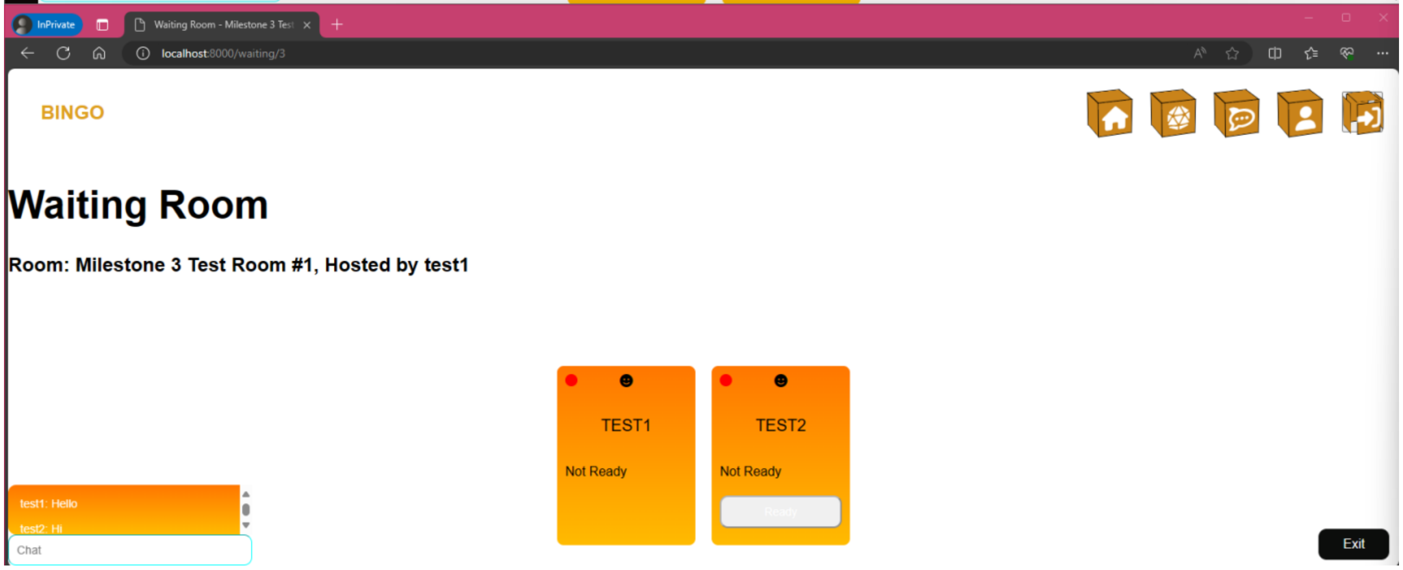
Not Ready

TEST2

Not Ready

test1: Hello

test2: Hi



Waiting Room

Room: Milestone 3 Test Room #1, Hosted by test1

TEST1

Not Ready

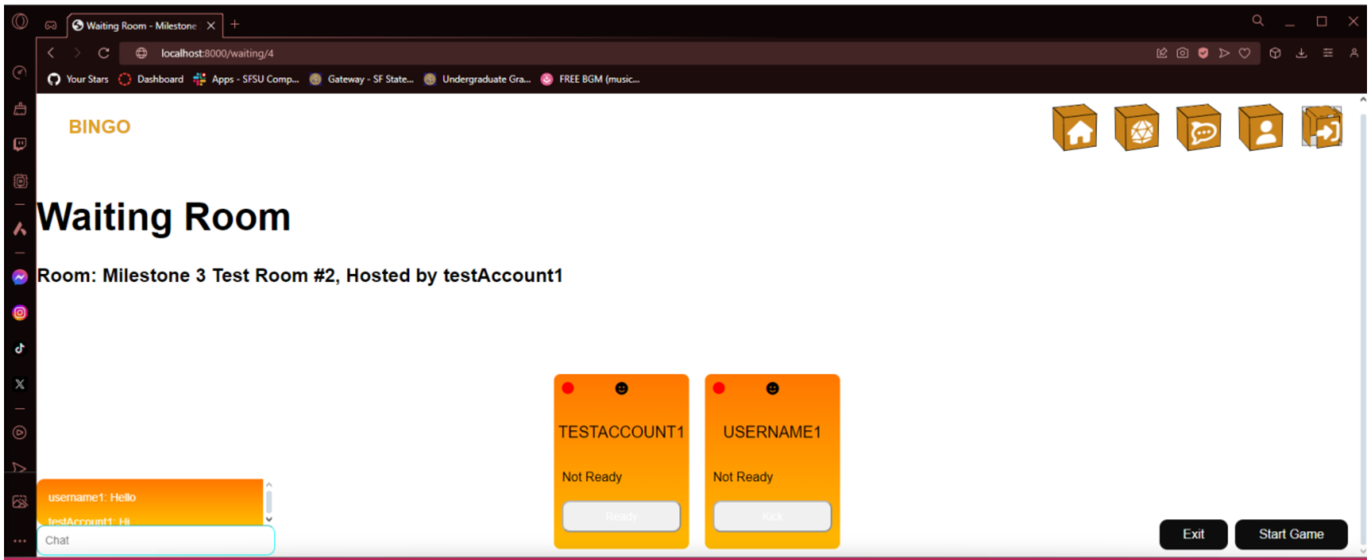
TEST2

Not Ready

test1: Hello

test2: Hi

Room 2:



Waiting Room

Room: Milestone 3 Test Room #2, Hosted by testAccount1

TESTACCOUNT1

Not Ready

Ready

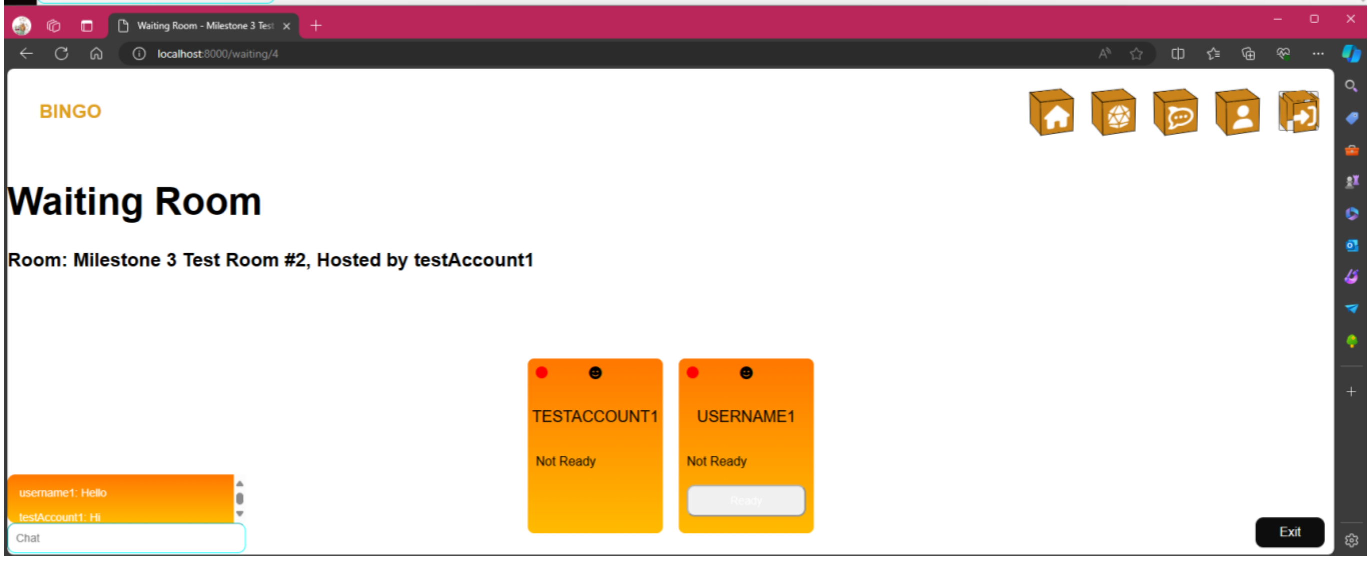
USERNAME1

Not Ready

Kick

username1: Hello

testAccount1: Hi



Waiting Room

Room: Milestone 3 Test Room #2, Hosted by testAccount1

TESTACCOUNT1

Not Ready

USERNAME1

Not Ready

Ready

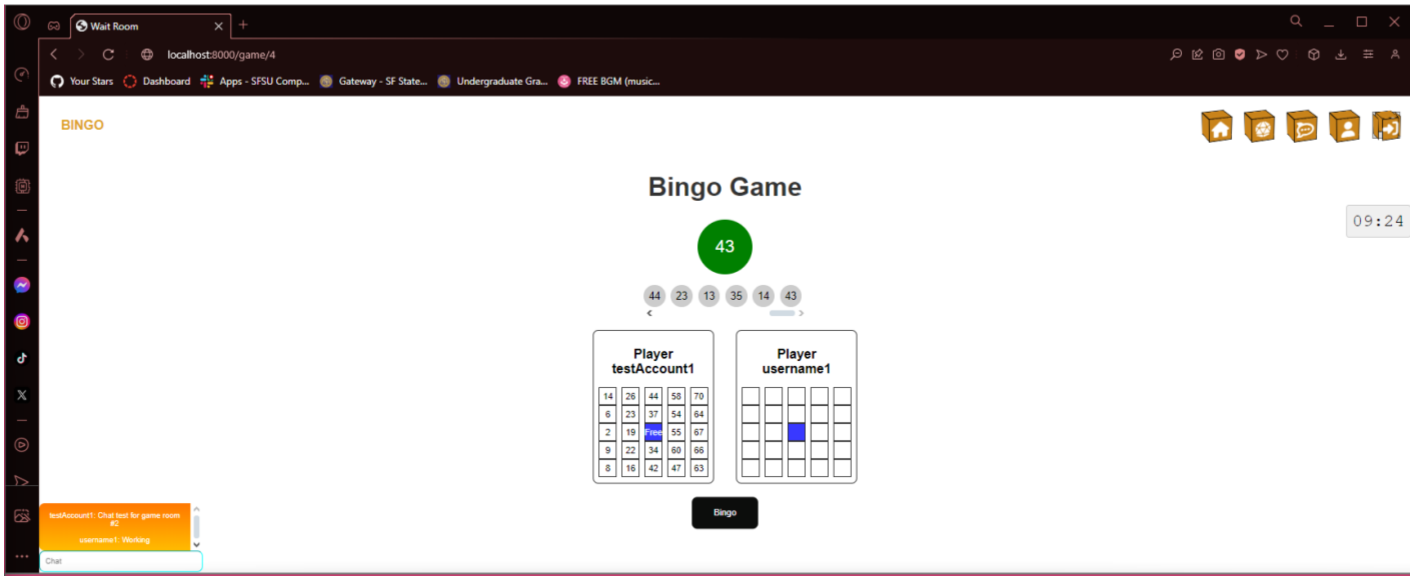
username1: Hello

testAccount1: Hi

<u>Chat Test in Game Room:</u>

**Chat is at the bottom left of the page.*

Room 1:



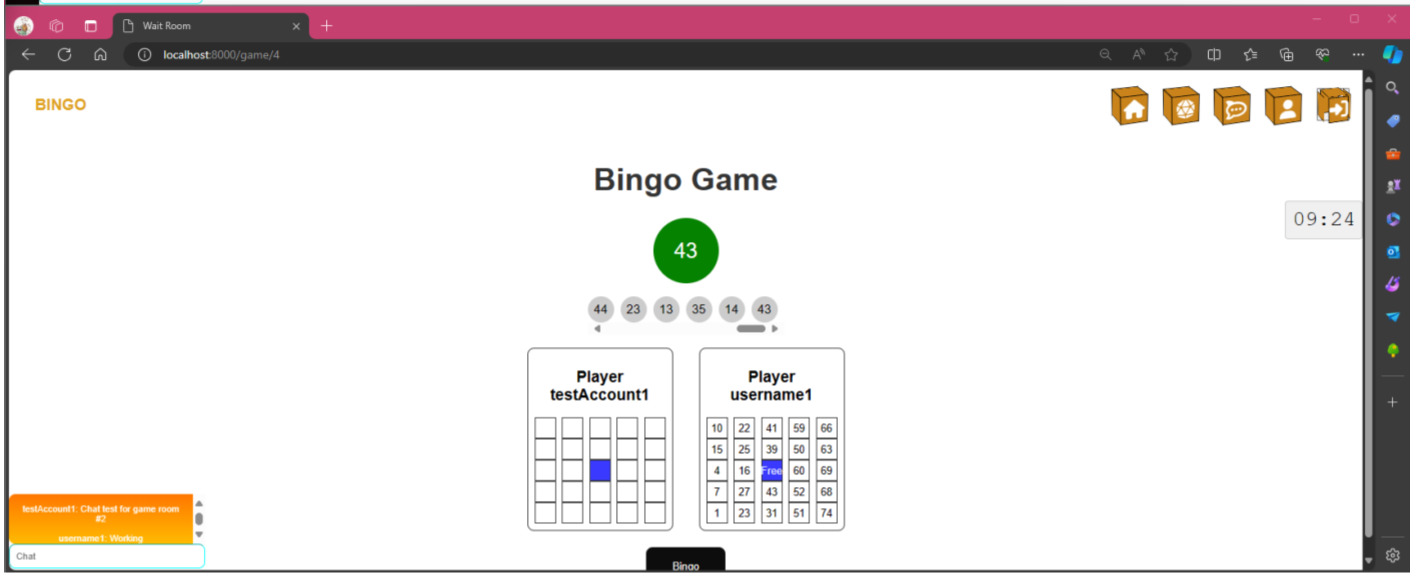
Player testAccount1

14	26	44	58	70
6	23	37	54	64
2	19	Free	55	67
9	22	34	60	66
8	16	42	47	63

Player username1

testAccount1: Chat test for game room #2

username1: Working



Player testAccount1

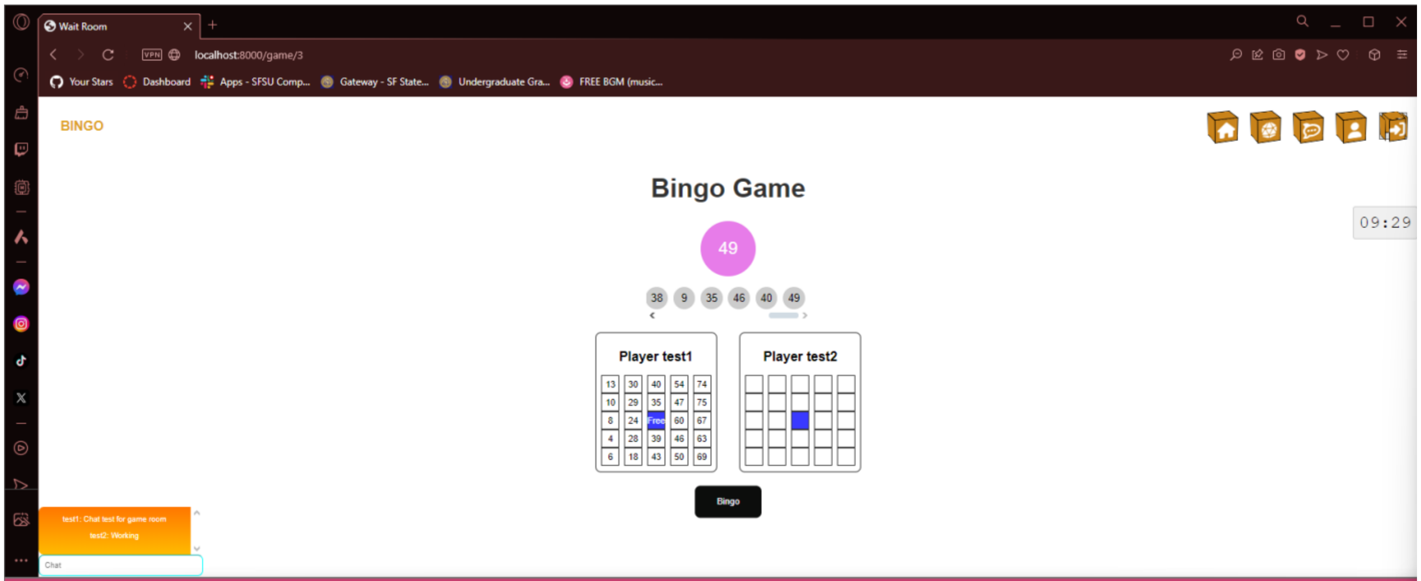
Player username1

10	22	41	59	66
15	25	39	50	63
4	16	Free	60	69
7	27	43	52	68
1	23	31	51	74

testAccount1: Chat test for game room #2

username1: Working

Room 2:



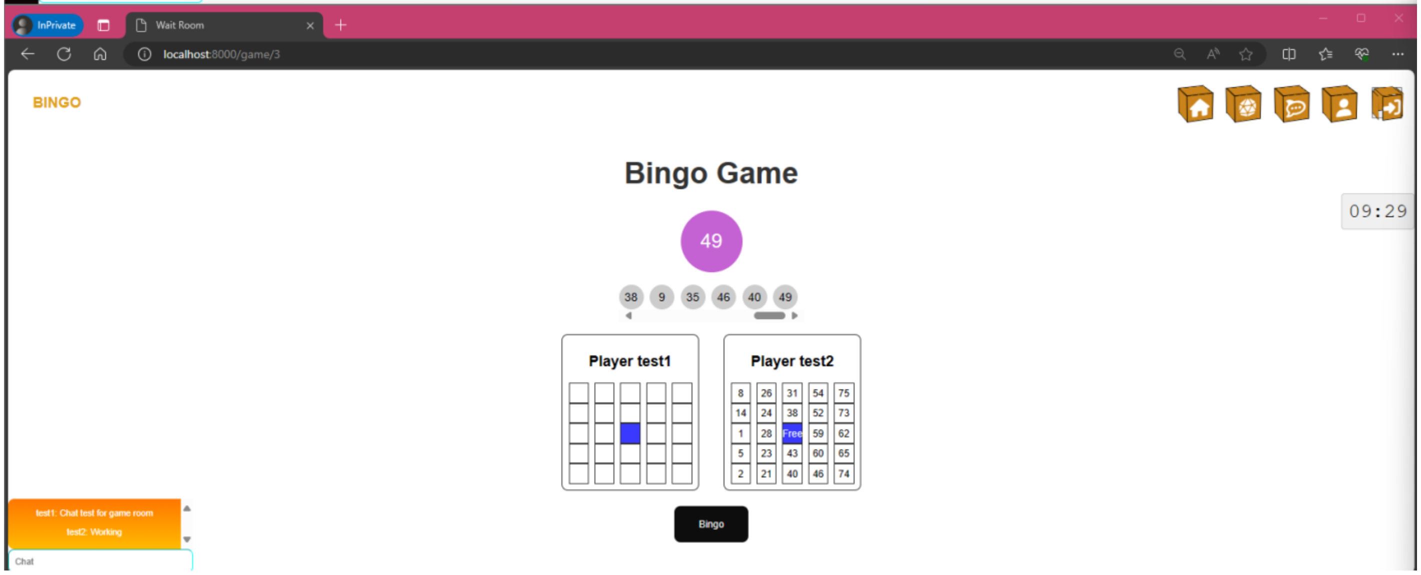
Player test1

13	30	40	54	74
10	29	35	47	75
8	24	Free	60	67
4	28	39	46	63
6	18	43	50	69

Player test2

		🟦		

test1: Chat test for game room
test2: Working



Player test1

		🟦		

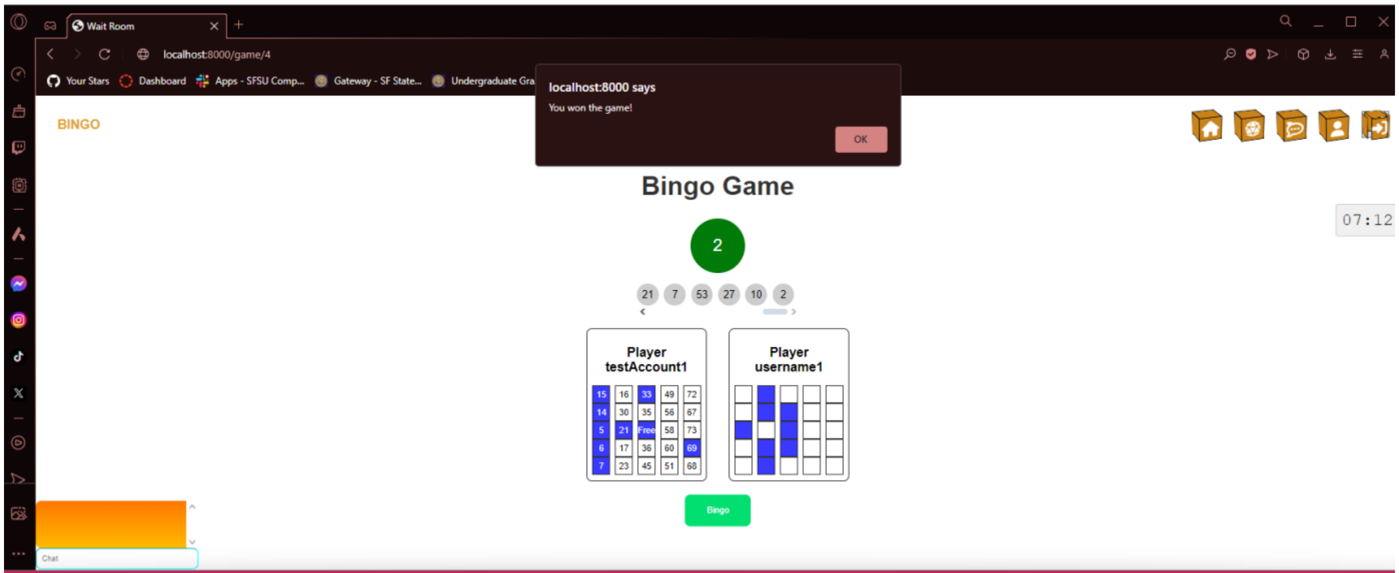
Player test2

8	26	31	54	75
14	24	38	52	73
1	28	Free	59	62
5	23	43	60	65
2	21	40	46	74

test1: Chat test for game room
test2: Working

Screenshot of winning alert element:

Room 1:



localhost:8000/game/4

localhost:8000 says

You won the game!

OK

BINGO

Bingo Game

07:12

2

21 7 53 27 10 2

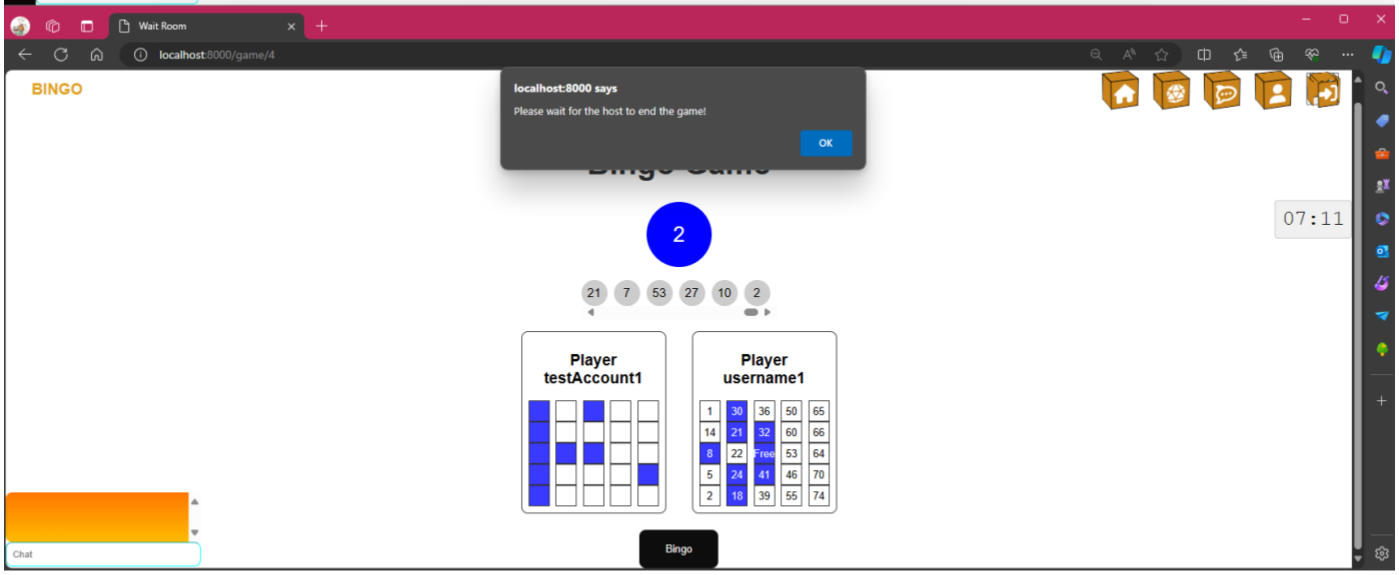
Player testAccount1

15	16	33	49	72
14	30	35	56	67
5	21	Free	58	73
6	17	36	60	69
7	23	45	51	68

Player username1

Bingo

Chat



localhost:8000/game/4

localhost:8000 says

Please wait for the host to end the game!

OK

BINGO

07:11

2

21 7 53 27 10 2

Player testAccount1

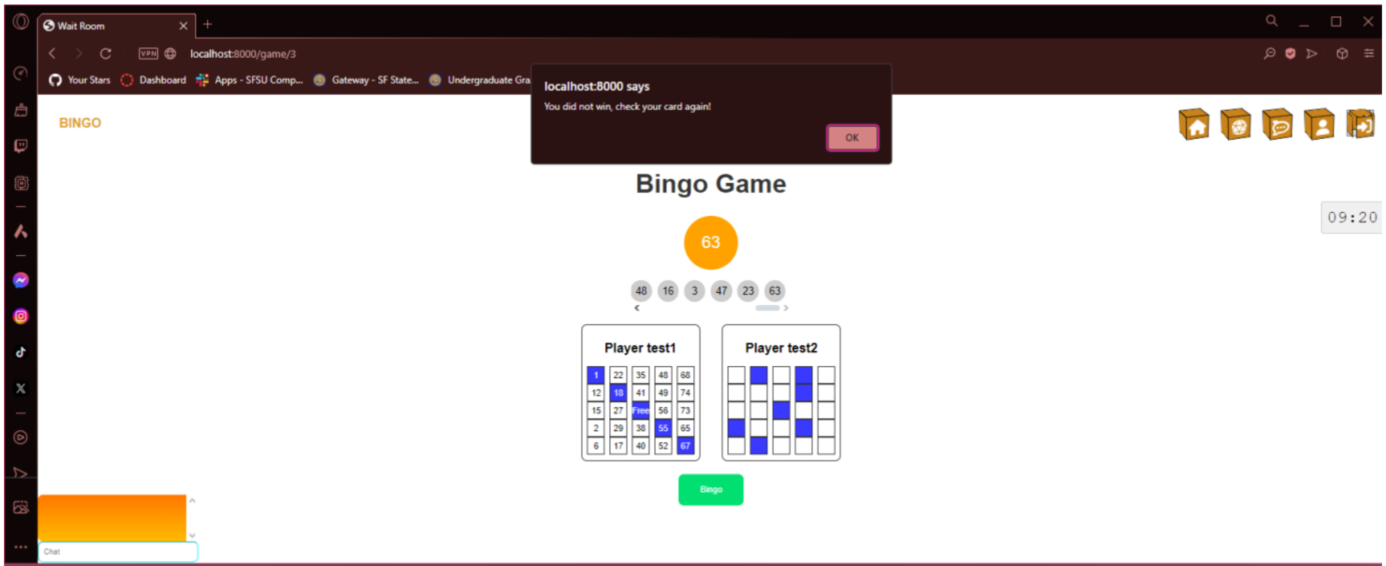
Player username1

1	30	36	50	65
14	21	32	60	66
8	22	Free	53	64
5	24	41	46	70
2	18	39	55	74

Bingo

Chat

Room 2:



localhost:8000 says

You did not win, check your card again!

Bingo Game

63

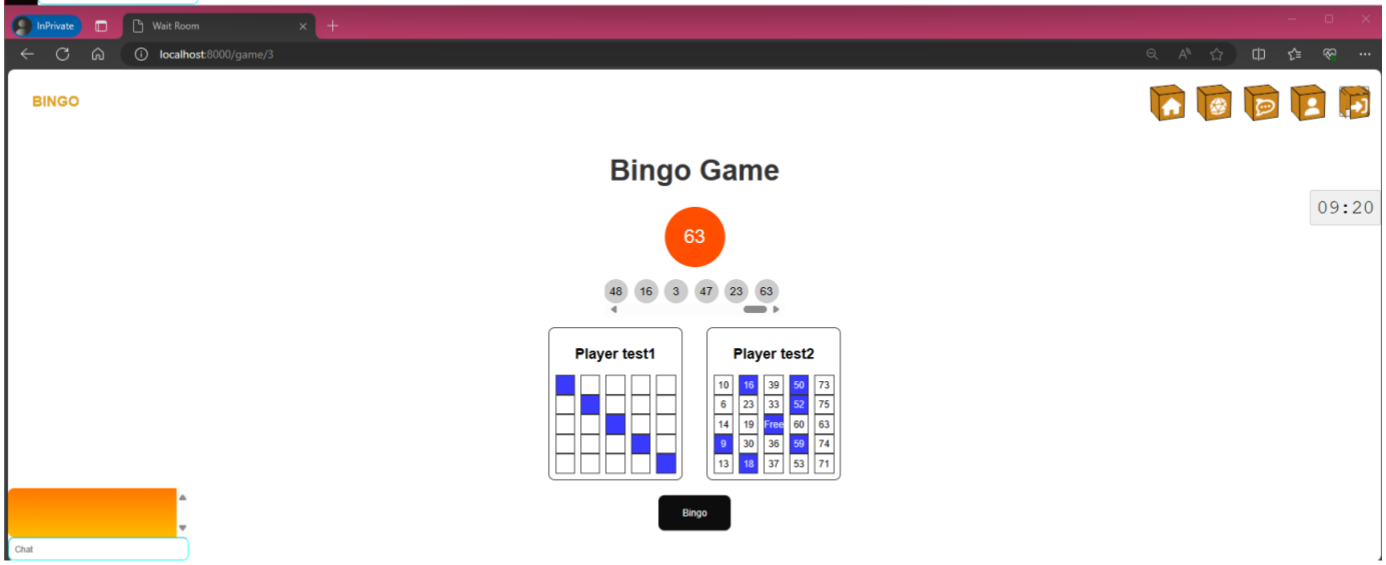
48 16 3 47 23 63

09:20

Player test1

1	22	35	48	68
12	18	41	49	74
15	27	Free	56	73
2	29	38	55	65
6	17	40	52	67

Player test2



Bingo Game

63

48 16 3 47 23 63

09:20

Player test1

Player test2

10	16	39	50	73
6	23	33	52	75
14	19	Free	60	63
9	30	36	59	74
13	18	37	53	71