Online Tambola

1. Introduction

Online Tambola is a web-Application where user can play and interact with friends after auth into application then create or join room of friends the private room or also explore or join into public rooms where they can play game with each other also group chat into their playing room for smooth interaction.

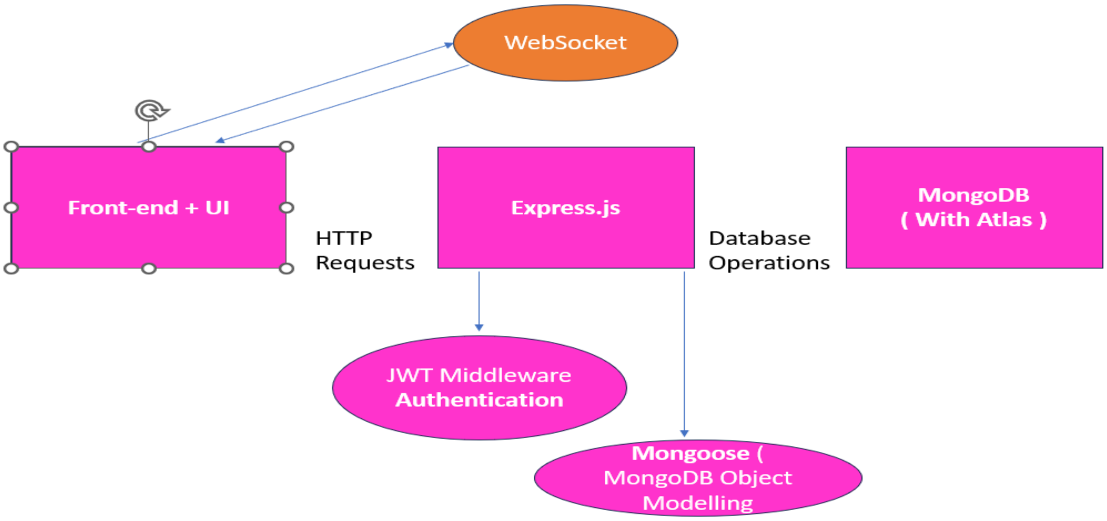
There will me mainly two types of room's:

- **Private Room:** This will be private room which will be made by host and this room can only be join by room code.
- **Public Room:** Host will have to explicitly set room scope to public and then room will be shown to Available room section on website.

2. Tech Stack Overview

This Online Tambola applications made using MERN (MongoDB, Express.js, React.js, Node.js) stack with React-Bootstrap from frontend components.

i. Front-end: Developed using React.js and with the help of bootstrap for responsive components.
ii. Back-end: Developed using Express.js, Node.js and MongoDB for Database.
iii. Database: MongoDB is used here for storing users information, currently ongoing games and other relevant information.



3. Authentication

User can register or sign-in into application by entering user-details after server verify it will generate the JWT (JSON-Web-Token) and will receive by the client which will be saved in client browser then for every request that token will come with request then server verify it, if it is verified then only he can take actions like host or join into room or send message to particular room.

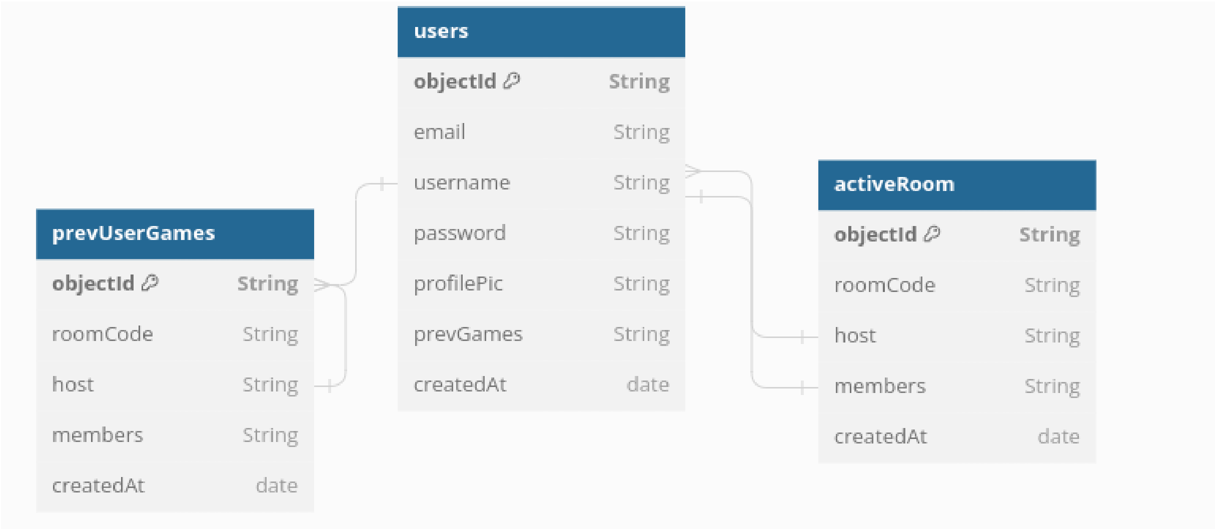
4. Authorization

Only 2 types of users will be there one is host and other who join, So only host have privilege to change the setting of current room and only host is authorize to kick any member out of room.

5. Database Design

In this application mongoDB are consists of following collections:

- users: stores user information including email, username, password, profile-pic, list of previously played games and "createdAt" field which stores date and time when user registed.
- activeRoom: This collection will be dynamic at will update on newly host game and when the game will be ended, it consists of field like roomCode (which is unique roomCode), host which refer to users table collection which describe the person who hosted the game, members field which contains the list of users the are the group and createdAt field when the room was created.
- PrevUserGames: it contains all the same fields that activeRoom as it will move the document from activeRoom schema from prevUserGames schema after every room is terminated (This collection is just for record)

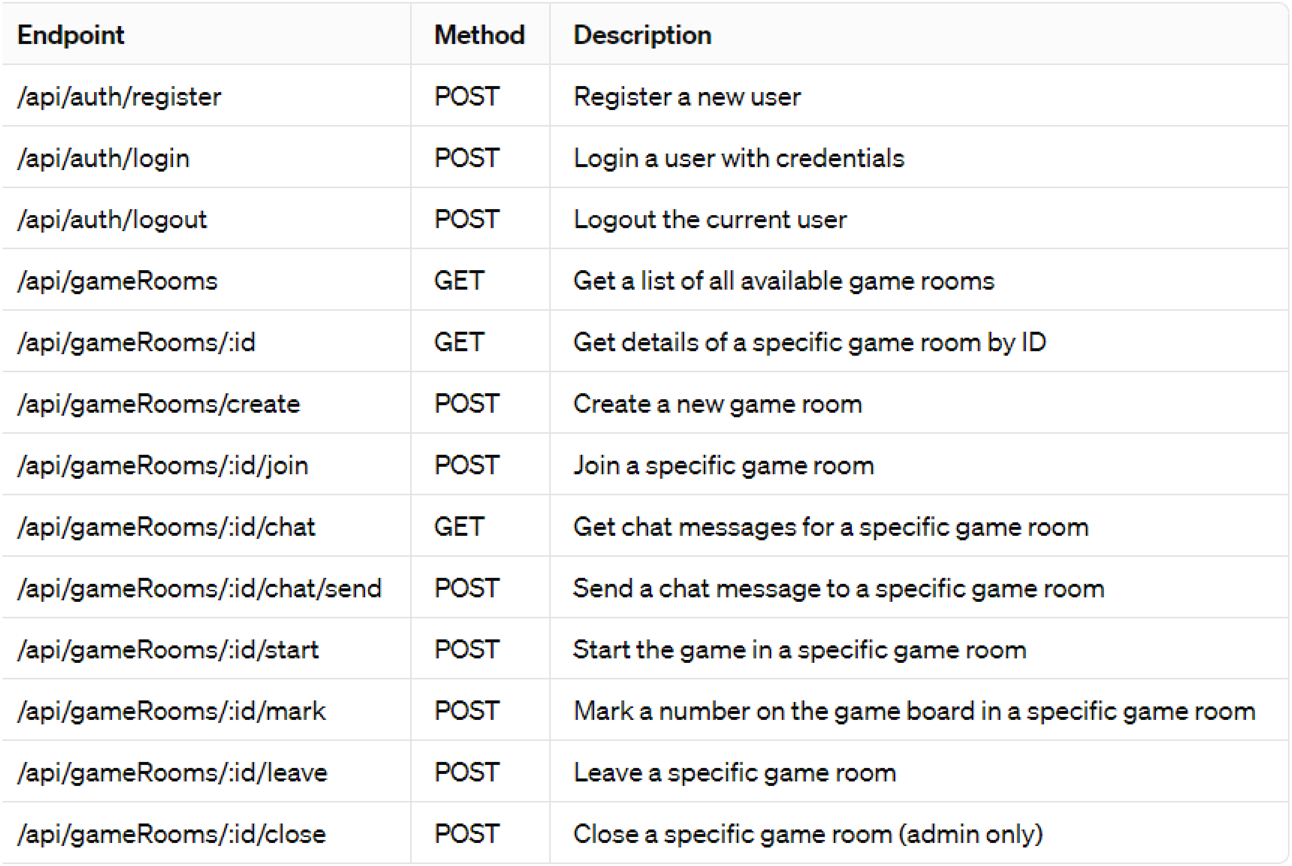


users	
objectId 🔑	String
email	String
username	String
password	String
profilePic	String
prevGames	String
createdAt	date

prevUserGames	
objectId 🔑	String
roomCode	String
host	String
members	String
createdAt	date

activeRoom	
objectId 🔑	String
roomCode	String
host	String
members	String
createdAt	date

6. ENDPOINTS



Endpoint	Method	Description
/api/auth/register	POST	Register a new user
/api/auth/login	POST	Login a user with credentials
/api/auth/logout	POST	Logout the current user
/api/gameRooms	GET	Get a list of all available game rooms
/api/gameRooms/:id	GET	Get details of a specific game room by ID
/api/gameRooms/create	POST	Create a new game room
/api/gameRooms/:id/join	POST	Join a specific game room
/api/gameRooms/:id/chat	GET	Get chat messages for a specific game room
/api/gameRooms/:id/chat/send	POST	Send a chat message to a specific game room
/api/gameRooms/:id/start	POST	Start the game in a specific game room
/api/gameRooms/:id/mark	POST	Mark a number on the game board in a specific game room
/api/gameRooms/:id/leave	POST	Leave a specific game room
/api/gameRooms/:id/close	POST	Close a specific game room (admin only)

6. SECURITY

Our online Tambola game built on the MERN stack, not only offers exhilarating gameplay but also prioritizes robust security measures. With stringent user authentication protocols, encrypted communication channels, and data validation mechanisms, we ensure the utmost protection of user information. As we look ahead, continuous monitoring and updates will fortify our defenses against emerging threats, reaffirming our commitment to providing a safe and enjoyable gaming environment for all players.

7. DEPLOYMENT STRATEGY

Firstly, we'll deploy the frontend on platforms like Netlify or Vercel, hosting the React app's production build. For the backend, cloud providers such as Heroku or AWS are ideal, deploying the Node.js server directly from our Git repository. MongoDB Atlas will serve as our database solution, ensuring secure and scalable storage. We'll implement CI/CD pipelines for automated deployment, focusing on monitoring, security, and backup procedures to maintain reliability and data integrity. Thorough documentation will accompany these processes to facilitate smooth and efficient deployment workflows.

7. CONCLUSION

In summary, the online Tambola game project, built on the MERN stack, offers a comprehensive gaming experience with user authentication, real-time synchronization, and chat functionality. Through a client-server architecture and secure authentication mechanisms, the platform ensures data integrity and user privacy. Moving forward, potential enhancements could include diverse game variants and social features. Overall, this project signifies a successful implementation of modern web technologies, poised to deliver entertainment and engagement to users worldwide.