

Online Tambola

Design Document

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 be 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 registered.
- 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)

