

Virtual Tokens App

By

**Dhananjay Garg
18BCE052**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Ahmedabad 382481**

Virtual Tokens App

Internship Report

Submitted in partial fulfillment of the requirements

For the degree of

Bachelor of Technology in Computer Science & Engineering

By

Dhananjay Garg
18BCE052

Guided By

Sanjeev Ray

[DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING]



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Ahmedabad 382481

Training Completion Certificate

Private and Confidential

Date: 06/07/2021

Dhananjay Garg

To whom it may concern

This is to certify that Dhananjay Garg has successfully completed the training into Reactjs from 24-05-2021 to 4-07-2021 in our organization under our **Jr. Software Associate Sanjeev Ray.**

He has received training on:

- Learnt React-The complete guide course from udemy.
- Learnt hosting in heroku and firebase.
- Made a real-time application with socket.io ,ReactJS,ExpressJS.

Description:

Real time MultiUser MultiRoom Application for virtual money management in gambling games and other accounting purposes.

Hosted at - <https://virtualtokens.herokuapp.com/>

We found him sincere and result oriented during his training. He was very much confident in representing the Project.

All the best for future endeavors



Vishal Shah

HR | Operations Head

To whom so ever it concern**Dhananjay Garg**

This Internship Confidentiality Agreement, known as the "Agreement", is between Dhananjay Garg, hereinafter known as the "Intern", and Toshali Infotech, hereinafter known as the "Company". Collectively, the Intern and Company shall be known as the "Parties", agree as follows:

KNOWLEDGE AND EXPERIENCE.

The Company shall be recognized as the individual or entity that provides a position to the Intern in order to obtain knowledge and experience in an industry that will merit justification of value in accordance with local and federal laws, hereinafter known as the "Internship Training Program". Intern agrees to serve unpaid for the Internship Training Program and to comply with any and all required policies of the Company and its Internship Training Program. The Intern shall not possess the authorization to represent themselves as an employee of the Company at any time.

The internship is related to an educational purpose and there is no guarantee or expectation that the activity will result in employment with Company.

CONFIDENTIAL INFORMATION.

Confidential Information shall include, but not be limited to, documents, records, information and data (whether verbal, electronic or written), Source Codes, drawings, models, apparatus, sketches, designs, schedules, product plans, technical procedures, compilations, studies, software, prototypes, samples, formulas, methodologies, formulations, patent applications, know-how, experimental results, specifications, and other business information relating to Company's business, assets, operations or contracts, furnished to Intern and/or Intern's affiliates, college, University, consultants or representatives, in the course of their efforts, regardless of whether such Confidential Information has been expressly designated as confidential or proprietary. Confidential Information also includes all Source Code of Program developed during Internship at Company, analyses, compilations, products, studies, and other data or material prepared by or in the possession or control of the Intern which contain, include, refer to or otherwise reflect or are generated from any Confidential Information. Intern acknowledges that no representation or warranty, expressed or implied, has been or is made by or on behalf of Company as to the accuracy or completeness of any of the Confidential information furnished to the Intern.

FORM OF DISCLOSURE.

Confidential Information may be oral, visual, or by demonstration or in some other form not permanently recorded and shall be considered Confidential Information regardless of whether such Confidential Information has been expressly designated as confidential or proprietary.

NOTICE OF UNAUTHORIZED USE OR DISCLOSURE.

Intern shall notify Company immediately upon discovery of any unauthorized use or disclosure of Confidential Information or any other breach of this Agreement by Intern or any third party and will cooperate with Company in every reasonable way to help regain possession of the Confidential Information and prevent its further unauthorized use or disclosure.

OWNERSHIP AND RETURN OF CONFIDENTIAL INFORMATION. All Confidential Information disclosed to Intern shall be and remain the property of Company. Upon Company's written request, Intern shall promptly return all Confidential Information (including all originals, Source Code, copies, reproductions, and summaries of such Confidential Information), or certify its destruction in writing, and keep the same confidential and secret in accordance with this Agreement.

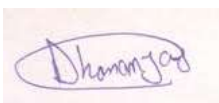
BINDING AGREEMENT.

This Agreement shall be binding upon Intern and its subsidiaries, College, University, assigns, legal representatives, and all corporations controlling the Intern or controlled by the Intern and shall inure to the benefit of Company and its subsidiaries, successors, assigns, legal representatives, and all corporations controlling Company or controlled by Company.

COUNTERPARTS.

This Agreement may be executed in one or more counterparts including signing a facsimile copy. Each counterpart shall be deemed an original and all counterparts together shall constitute one and the same instrument. IN WITNESS WHEREOF, I, the Intern, have read the above Agreement and agree to its terms.

Intern's Signature:



Date: 06-07-2021

Print Name: Dhananjay Garg

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to all those who supported me throughout my internship. I would like to give special thanks to my project manager and mentor Sanjeev Ray, my project partner Dhruv Parekh and finally my boss and CEO Mr. Ashish Agrawal under whose guidance I was able to complete the given project in given time period. Also, I am thankful to other people from the firm as well as my teammates who provided constructive suggestions to enhance the project.

ABSTRACT/ Outline

The Virtual Tokens App is a real time application using sockets which helps in money management for gambling games and other purposes. There are two types of roles in any instance of the game – Host and Player side. The Host creates the game and manages other players with itself also being a player. The other players can join the room created by host and play their part with no management responsibilities.

CONTENTS

Certificate	i
Acknowledgement	iii
Abstract	iv
Chapter 1 Introduction	1
1.1 Brief description about the company	
1.2 Brief description about the internship	
1.3 Objectives	
1.4 Problem Statement	
Chapter 2 Research and Decision Making on the Problem Statement	3
2.1 Research Analysis	
2.2 Solution	
Chapter 3 Methodology	5
3.1 Technologies used	
3.2 Implementation	
3.3 Screenshots with expalination.	
Chapter F Conclusion and Future work	12
Conclusion	
Future work	
Appendices	
A. List of useful websites	
B. Additional Material	

Chapter 1 Introduction

1.1 Brief description about the company

Toshal Infotech is best known for its Creativity, Design, Website, Mobile and Software Development. With an innovative, creative and pragmatic approach, software development at no less than an exciting and engaging process. We have an in-built knack for delivering high-performance websites, mobile and software applications - of all size and complexities. In more simple terms, we make compelling products that scale to your business needs.

To make this possible, Toshal Infotech has a highly qualified in-house team of developers who retain deep domain expertise in their niches and excel at delivering the best, top-of-the-class solutions and products to a wide client base across the globe. We allot a dedicated team to work with you so that we have a better understanding about what you need and why. These insights are useful in driving development and allow us to create a perfect working digital solution for your business. As a value-added advantage, each finalized Toshal Infotech project is backed by expert implementation and deployment using the latest, cutting-edge technologies.

1.2 Brief description about the internship

I was offered a 6-week internship at Toshal Infotech located in Surat, Gujarat by walk in interview. The company uses React for building many projects so I was also trained for same under guidance of my mentor Mr. Sanjeev Ray. My mentor provided me with the Udemy Account of the company which had all the required paid courses. I choose the React – The Complete Guide by Maximilian Schwarzmüller, Academind. This course was the latest available with function-based components so I proceeded with this. As we also had to make a project at last in a small duration of 6 weeks so my mentor listed out the important topics to cover in MERN stack. So, I started with learning JavaScript and NodeJS and created an api for searching movies using OmdbApi and displaying specific results of the fetched JSON object. After that started with React and completed the assignments given in between the course. After learning

all the important topics , we finally moved to create the Virtual Tokens App. Later it was deployed on Heroku which is a free hosting website.

Website link - <https://virtualtokens.herokuapp.com/>

1.3 Objectives

The main objective was to get trained for JavaScript and MERN stack and able to create solution for any problem statement. We also need to learn about real time applications and not just https(request-response) based. So, in order to incorporate all these concepts into a single project we need to find a problem statement which is not being implemented much and fun to do at the same time. So, after a lot of brainstorming and research we finally found our problem statement which is explained in the next section of the chapter.

1.4 Problem Statement

Card Gambling games are quite common for people to spend time with their friends and family which obviously involves money transactions. Many online games are built for this but the playing experience is not the same as it would be if all the people are in a same room playing with the real cards and having fun together. But the minus point in that is every player has to get real cash that too in small denominations which is not feasible for everyone and many a times not possible. Also, it is so tedious to do the transactions and manage your cash. So, what if we only virtualize and create application for the money part of the game and let players play with real cards. It will enable players to play these games anytime and anywhere with having just the deck of cards and easy virtual money/token management.

Usually, such games have each player carry some balance and there is some balance on the center table or play area which will be won by a player at the end of the round or equally split among the winners in case of a tie. Also, this game has to be real time and all the players can see the live changes in their balance and table balance.

Chapter 2 Research and Decision Making on the Problem Statement

Now our problem statement is well defined and we need to figure out how to make it real time . There are multiple ways to achieve it . But the approach which suits us best and also for the user's experience is to be decided. This chapter will describe how we decided to use web sockets and why it was the correct choice for this problem statement.

2.1 Research Analysis

There is a common request-response based approach with https which is used widely in projects but the issue with that is we need to broadcast data with many people simultaneously which is not possible as the server cannot send the data to clients unless requested by them. The workaround is that we can set a timer on all client side which sends request to server asking if there is any new data. But that is not at all optimal. Suppose if there is no new data to transfer there would be a lot of useless request – response between machines and efficiency ratio will be less.

Second way to communicate is creating a local area connection in which players will connect to a same network as the game as per our problem statement is going to be played physically with real cards so players will be in close proximity to each other and hence connect to same network. A common mobile-game known as Mini Militia works on same principle wherein a person creates a hotspot network and gets connected to other players. This approach would have been the best as it does not require an internet connection and there would be no concept of remote server and therefore no delay. The only problem with this approach is of user – experience . This type of communication is no longer used by any developer lately as user will have problems connecting to a person's hotspot as everyone has to change their network connection to the hotspot. Also, the player might not be comfortable to share his hotspot password. This also will make host's internet connection used by all other players and thus draining it fast. To summarize the above problems , user experience will be affected and it needs all this configurations before starting the actual game. So, our product will fail at this point.

Third way to make it real-time will be using push notification technology like mongo pusher . This technology is used worldwide for real time applications like WhatsApp , Facebook and other giant applications. But still this approach is not suited for our project. This approach is not as fast as we want it for our game. As we are trying to virtualize the money transaction which needs to be reflected instantly. Many a times we have observed that the WhatsApp messages are delayed , the live location on application like Zomato, Uber is not smooth and updates with some delay. The reason is that it is connectionless service. Whereas we require is a connection oriented fast reliable service. Also, we don't have database requirements in our product so there is no way we can think of using mongo pusher or firebase real-time service.

Finally, the last and the only option which can be incorporated in our project is the use of web sockets. It is connection-oriented TCP/IP service which will suffice our requirements. Now we will think if this approach is so better then why it doesn't replace pusher technology. The answer is simple that the messaging task and any data of other applications can be exchanged at any time of the day so it would require 24*7 TCP connection and there are billions of users so billions of 24*7 TCP connection is required no matter if there is any message or data exchange between them or not. This is not at all practical and waste enormous amount of bandwidth. Our game duration will not last the whole day and during the gameplay there is continuous data transfer so bandwidth will not go wasted. This is the prime point where one needs to decide when to use sockets or push notification technology.

2.2 Solution

We proceeded with the web sockets approach as it is clear from the above discussion. In particular , we used Sockets.io which is an abstract built over web sockets (ws). Socket.io also has a facility of admin.socket.io which helps in managing the sockets and overall details of all the connected servers and clients.

Chapter 3 Methodology

3.1 Technologies Used

The tech-stack for developing app consists of –

- ReactJS
- NodeJS
- ExpressJS
- Socket.io
- Heroku
- Postman

3.2 Implementation

Exploring socket.io and websockets (ws) documentations. After reading all the resources available we gathered a few functions which help us fulfill our goal and use them in the best practices integrating with backend and frontend JS.

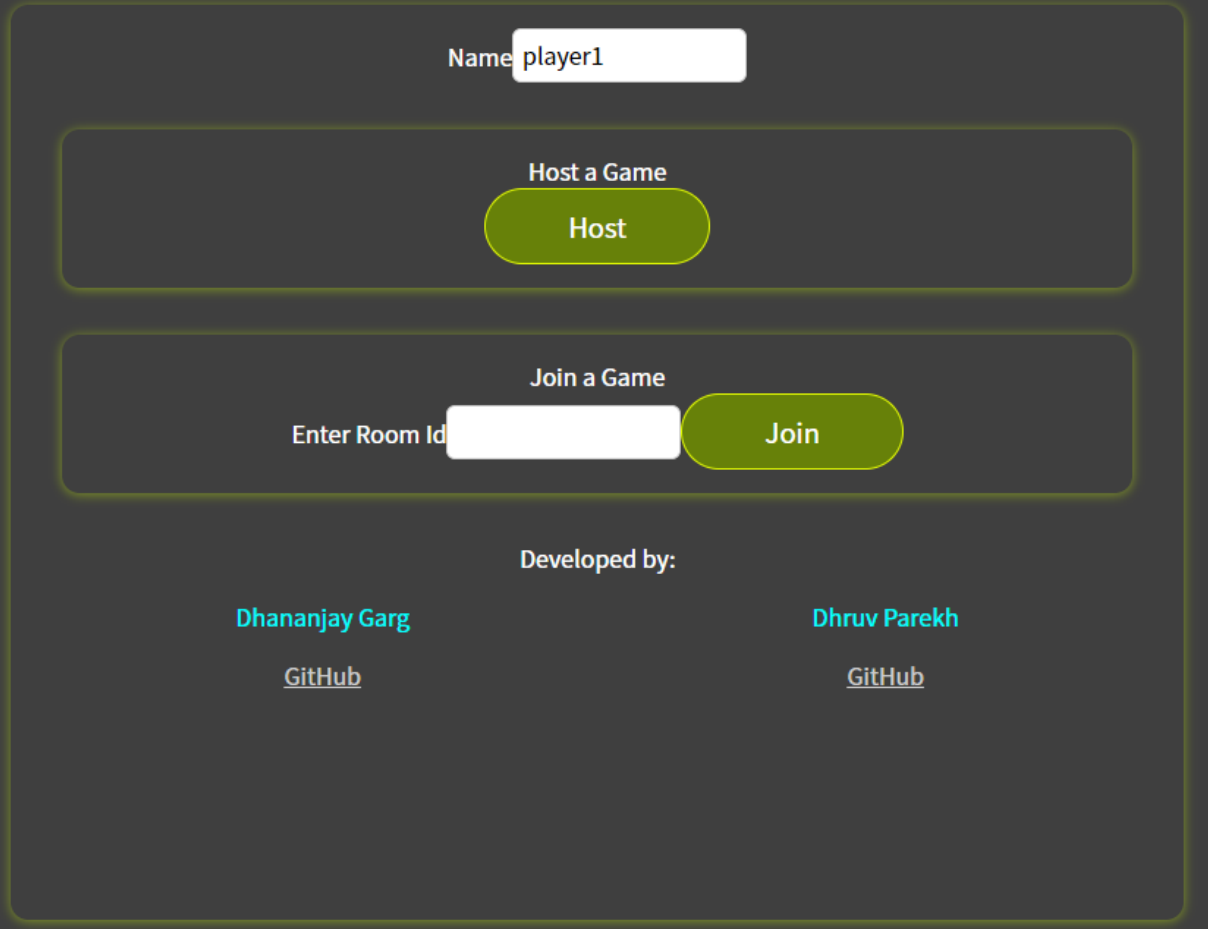
Component division was a crucial task and we added the components as we moved forward in our plan. Later we found that it was getting hard to incorporate the next step or new feature so we refactored the entire front end and backend to make the code scalable and readable. After this we swiftly added all the new features and understood the importance of writing clean and structured code. The data structure in backend has to be very optimal as there is no database and all live games are handled by backend simultaneously. It was also important to maintain and delete the related data once the socket connection is disconnected and it depends on which type of player is disconnected, host or a player. Finally after investing day and night for 3 weeks on the implementation we started deploying on Heroku which is a free hosting website. Hosting a socket based website was not an easy task and it gave a lot of hurdles but after researching various resources we were successful in doing so. Later we discovered few bugs which were solved accordingly as the clean and structured code made it easier to find the issue in such a large codebase. We did unit testing after every small step to ensure everything works perfectly.

Now I will explain the Game Play with help of screenshots of the web application.

All edge cases and constraints are taken care of and there is no way our product will give error to any unexpected input.

3.2 Screenshots with explanation

Landing Page – here user will have two choices either to host a new game or join an existing game created by other user. After hosting it will generate a 4 letter code which is to be shared with other players.

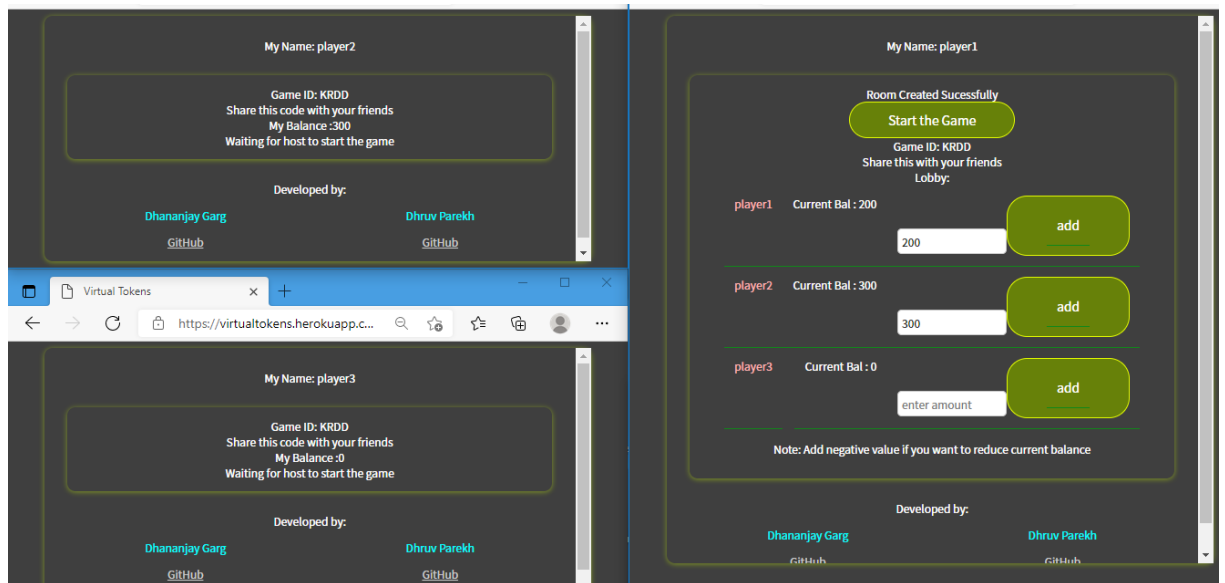


The screenshot shows a dark-themed landing page for a game. At the top, there is a text input field labeled "Name" with the value "player1". Below this, there are two main sections. The first section is titled "Host a Game" and contains a green button labeled "Host". The second section is titled "Join a Game" and contains a text input field labeled "Enter Room Id" and a green button labeled "Join". At the bottom, there is a section titled "Developed by:" which lists two developers: "Dhananjay Garg" and "Dhruv Parekh". Each developer's name is followed by a link to their GitHub profile, indicated by the text "GitHub" below the name.

After hosting and players joining the game everyone will be in waiting lobby and can see their current balance on the screen. It is host's duty to assign each player with initial balance. The logic behind this is the players will deposit real money to the host and the host will in turn assign token money inside the game. After game ends, the player is being returned the real money based on his/her remaining balance in the game summary. Just like it works inside a casino with plastic tokens.

HOST LOBBY – assigning money to players and control of starting the game.

PLAYER LOBBY – ensuring that the correct amount is credited to the balance and waiting for host to start the game.

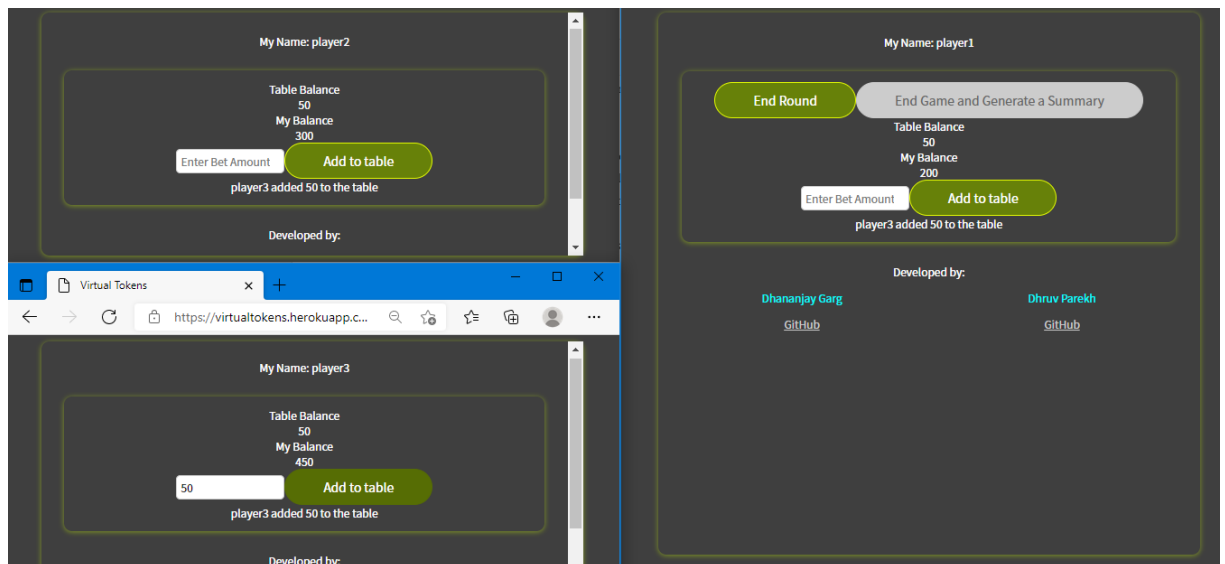


After host starts the game no new player can enter the room and will be displayed with the error message “ Incorrect code or game already started. Try again”.

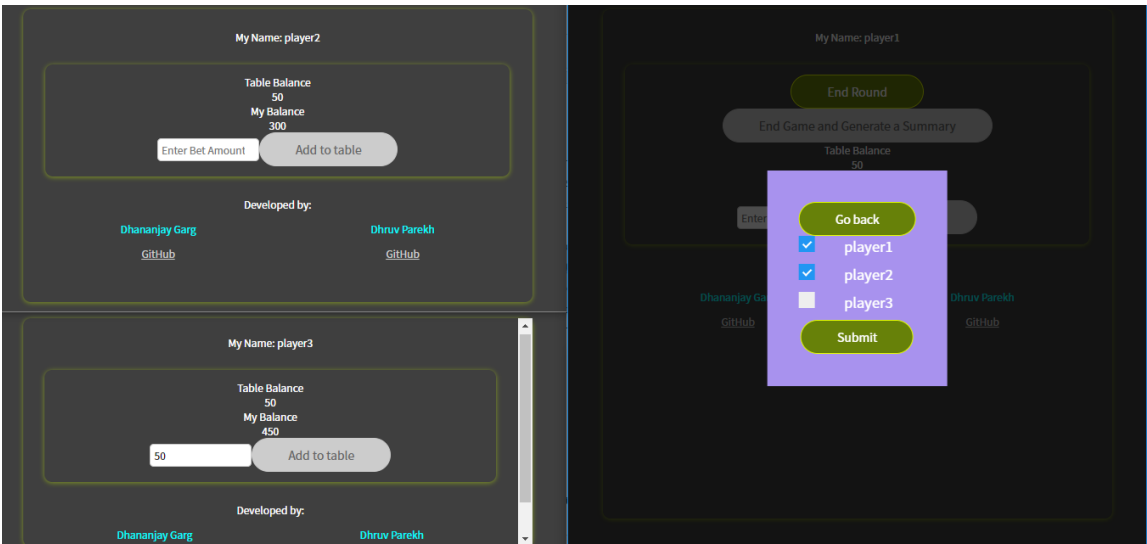
GAMEPLAY

PLAYER – can add money to the table balance. His balance will be deducted and added to the table. And that table balance will be updated for all the players along with the message “player x“ added “y amount” to the table.

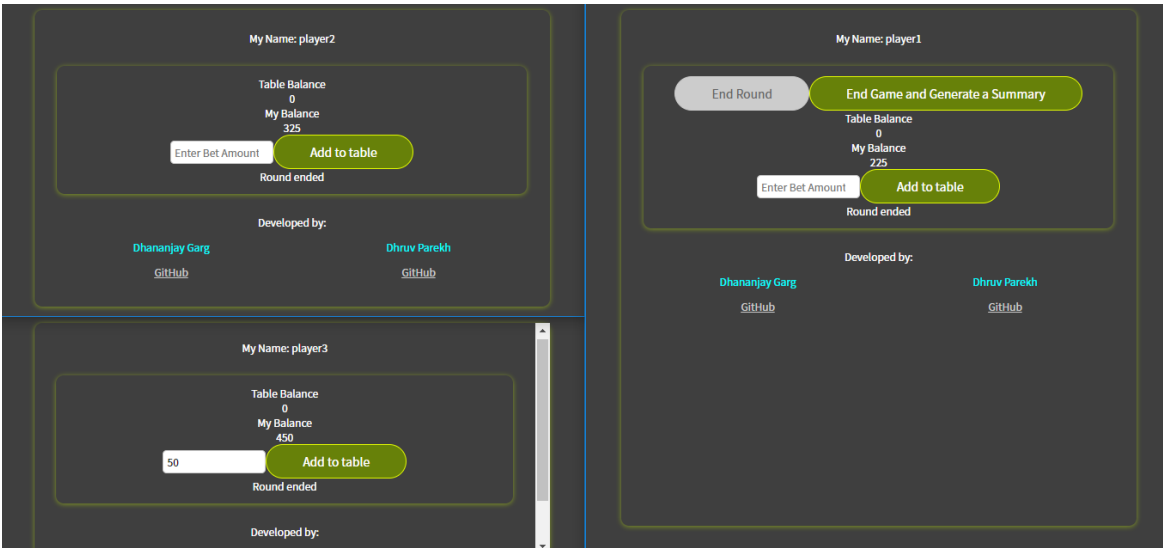
HOST – Can perform all activities of a player along side control of “END ROUND” and “END GAME” .

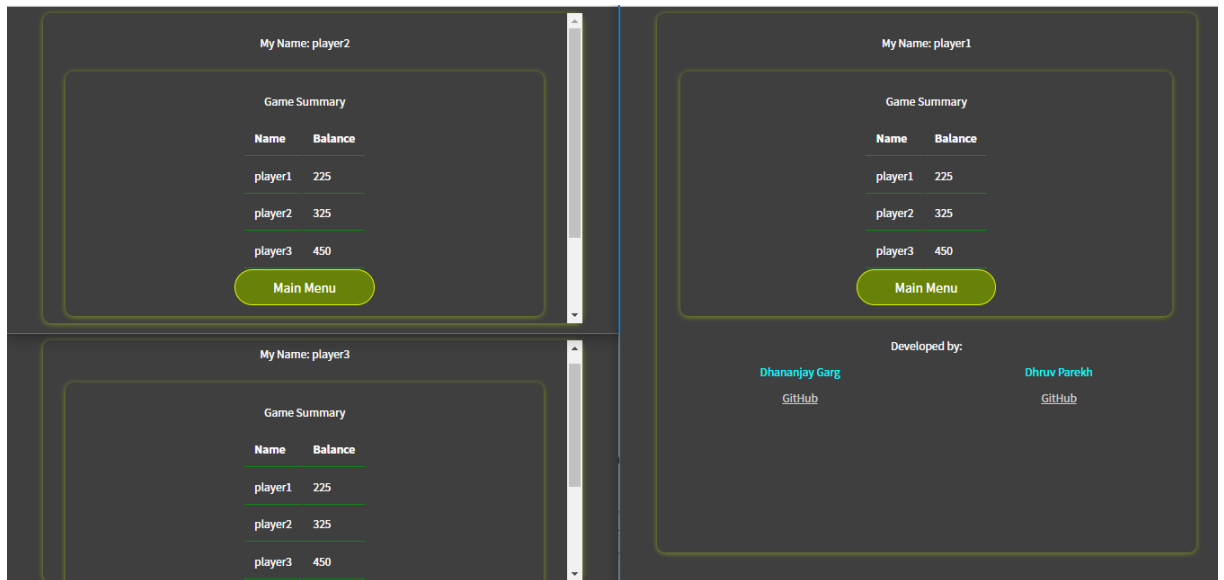


The end game button will be disabled when the table amount is non-zero. The table money has to be distributed among winners which can be multiple and therefore a modal is displayed where host can select the winners. The money will then be equally distributed among winners and a message of round ended will be displayed. At this stage the table money will become zero and we can end the game or play another round depending on user choice. When table balance is zero, the cannot end the round as there is no money to be distributed among anyone. Therefore both buttons will never be activated together. The player can't add money to table during end round decision making is going on by the host as the add button will be disabled during that time. Also player can't add money more than his balance.



After playing multiple rounds as per user choice when host clicks on “END GAME” it will display a balance summary on every player’s screen and then this money can be requested back in next game or withdraw in form of real money.

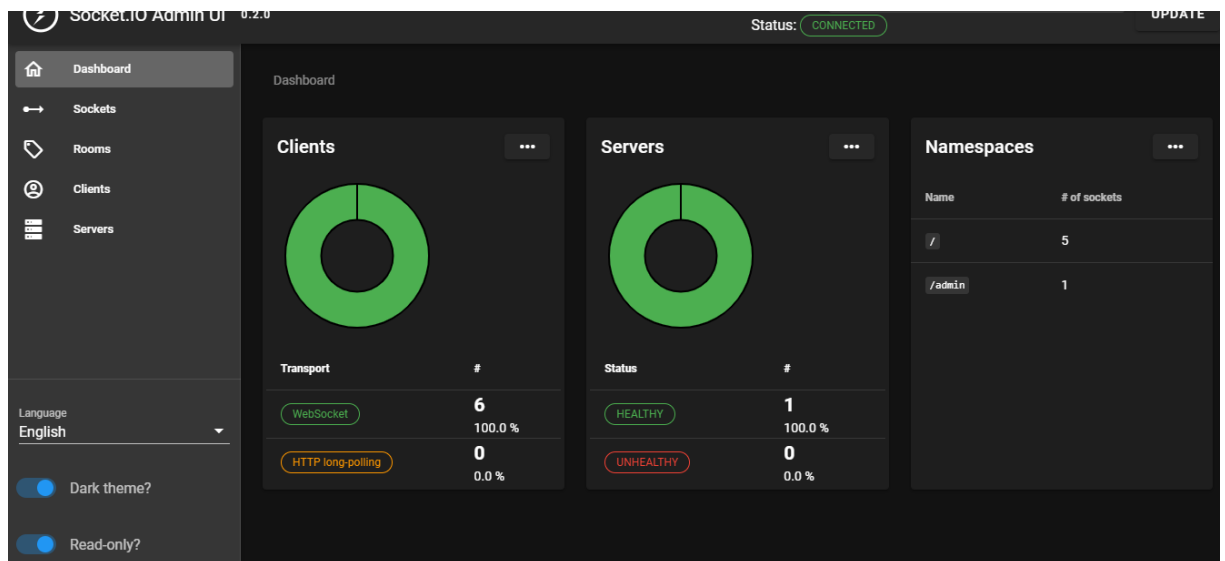




After this page user can either exit the web page or click on main menu to be redirected to Landing Page which is just the refreshing of current page as it is a single page dynamic application.



We can watch the connected sockets and their room along with its health on a monitoring system provided by admin.sockets.io which needs to get permission from backend code to access the sockets informations.



The screenshot shows the 'Clients' section of the Socket.IO Admin UI. It displays a table with the following columns: #, IP address, Transport, and # of sockets. The table lists six clients, all using WebSocket transport. Each client row has a copy icon and a menu icon (three dots). The bottom right of the table shows 'Rows per page: 20' and '1-6 of 6'.

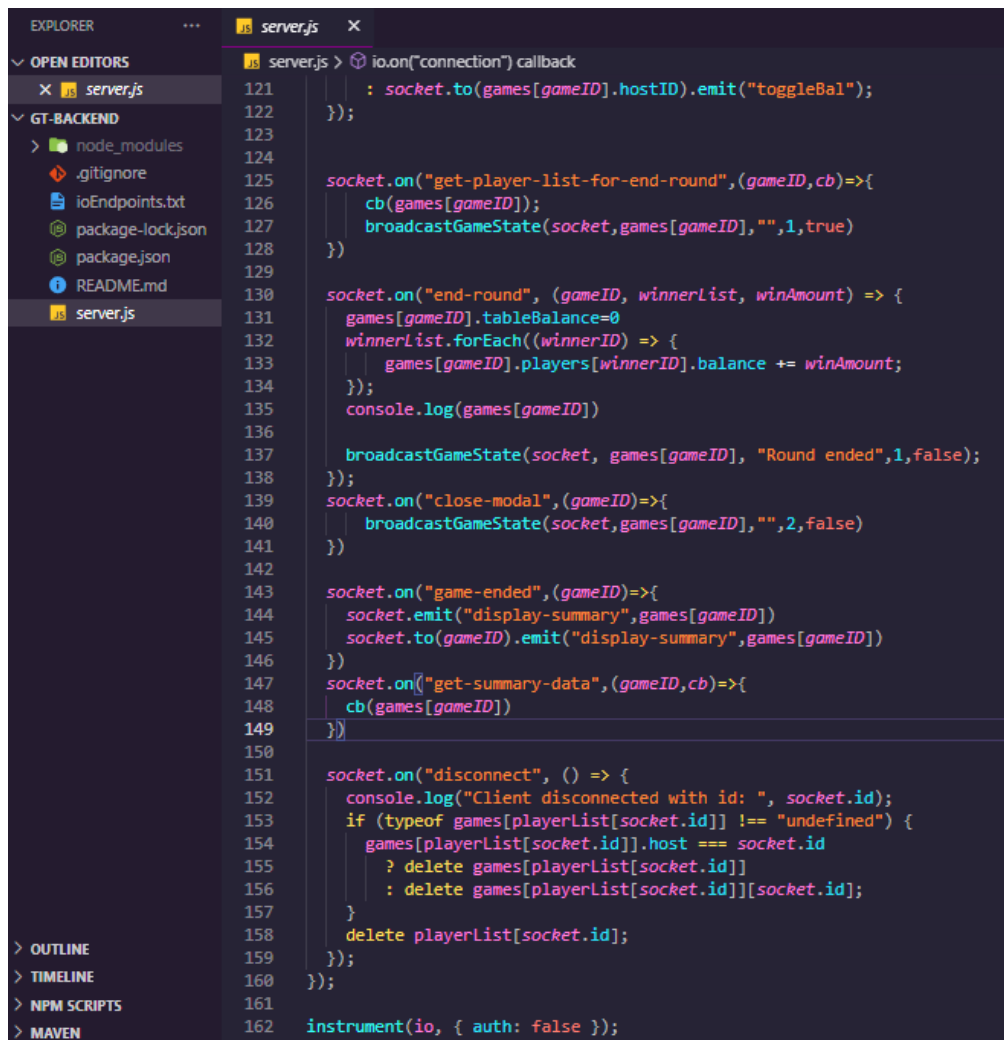
#	IP address	Transport	# of sockets
VMQJlcDpnWyp	::ffff:10.38.230.233	WebSocket	1
96yHzEQ9wTgm	::ffff:10.38.182.183	WebSocket	1
KQhLVmnJB2rW	::ffff:10.38.182.183	WebSocket	1
HsdTRb2aAg_M	::ffff:10.14.2.76	WebSocket	1
E4BmtMAJQwd_	::ffff:10.13.124.73	WebSocket	1
b2gCjX0XHDwc	::ffff:10.8.99.197	WebSocket	1

The code for this is available on Github

Link - <https://github.com/DJGARG7/virtualtokens>

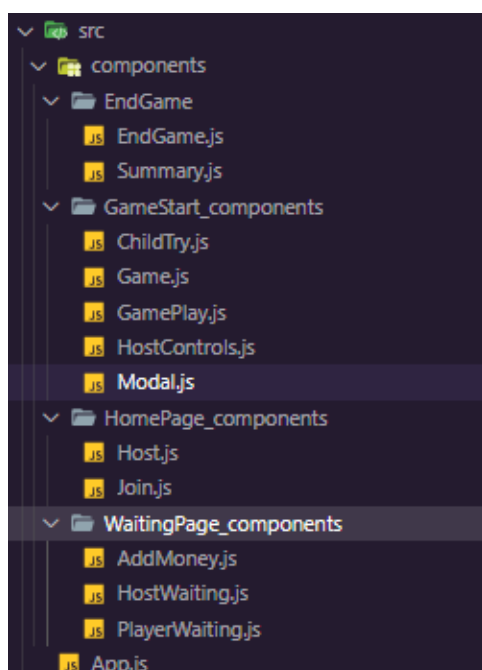
The screenshot shows a GitHub repository page for 'DJGARG7 Delete todos.txt'. The repository has 4530d38 commits on Jun 24 and 21 commits. The file list includes Node_backEnd (Update .gitignore, last month), React_frontEnd (Delete todos.txt, last month), and README.md (added gameplay, last month). The README.md file is expanded, showing the title 'gamble-tokens' and the description 'real time multiuser web app using node and react js and socket.io' and 'host room creation and code generation'.

A code snippet of back-end



```
121 : io.on("connection") callback
122 : socket.to(games[gameID].hostID).emit("toggleBal");
123 });
124
125 socket.on("get-player-list-for-end-round", (gameID, cb) => {
126   cb(games[gameID]);
127   broadcastGameState(socket, games[gameID], "", 1, true)
128 })
129
130 socket.on("end-round", (gameID, winnerList, winAmount) => {
131   games[gameID].tableBalance = 0
132   winnerList.forEach((winnerID) => {
133     games[gameID].players[winnerID].balance += winAmount;
134   });
135   console.log(games[gameID])
136
137   broadcastGameState(socket, games[gameID], "Round ended", 1, false);
138 });
139 socket.on("close-modal", (gameID) => {
140   broadcastGameState(socket, games[gameID], "", 2, false)
141 })
142
143 socket.on("game-ended", (gameID) => {
144   socket.emit("display-summary", games[gameID])
145   socket.to(gameID).emit("display-summary", games[gameID])
146 })
147 socket.on("get-summary-data", (gameID, cb) => {
148   cb(games[gameID])
149 })
150
151 socket.on("disconnect", () => {
152   console.log("Client disconnected with id: ", socket.id);
153   if (typeof games[playerList[socket.id]] !== "undefined") {
154     games[playerList[socket.id]].host === socket.id
155       ? delete games[playerList[socket.id]]
156       : delete games[playerList[socket.id]][socket.id];
157   }
158   delete playerList[socket.id];
159 });
160 });
161
162 instrument(io, { auth: false });
```

Components of front End



Chapter F Conclusion and Future work

Conclusion

In this internship, I dealt with first time experience with a new language JavaScript and learnt it quickly and with a lot of research was able to make such real-time multi-user multi-room product game. I learnt to work under pressure and got some industry exposure proving as a stepping stone in my journey. I learnt the importance of team work and honor the ideas of team mates and gained confidence that I will prove as an important asset to any company.

Future Work

Our project Virtual Token was a great experience and a fun assignment to build. It still requires lot of work to improve it and add animations and good user UI. Also we had to do unit testing after every small change so we could build an automation testing for future work. We can also incorporate payment methods like Paytm or other payment models to make the product more easy to use.