Computer Networks
COMP421

Auction Application with Socket Programming

Alp Gökçek, 041701014
Dilay Sapmaz, 041701032
Muhammed Rahmetullah Kartal, 041701008

Instructor: İlker Bekmezci

28.01.2021

Abstract: **Socket programming manages the communication between two nodes (client - server) in the network by connecting them to each other. Its main purpose is messaging between users in real time. When a user sends a message to another user, the server changes and this message is not forwarded. In this article, an auction app with socket programming (socket.io), which is one of the cheap and fast solutions that eliminates this problem, is explained.**
Keywords: Socket programming, Computer networks, Real-time communication

1. Introduction

There are five different layers which are in a peer to peer communication in the hybrid network model, these layers are called Application Layer, Transport Layer, Network Layer, Data Link Layer and Physical Layer. The socket can be assumed as a door between application and end-transport layer. Related to the communication method (TCP/UDP) socket types differ, while there is one general socket in UDP, in TCP communication each client interacts with a different socket. In a standard auction application connections should be separate for each user and room, otherwise problems like bidding to the wrong room or bidding by the wrong client may occur.

1.1 Running the Application

1. Create an application from Firebase console and download the secret key file.
2. Instantiate Cloud Firestore database and turn on the email login from the Firebase Authentication.
3. Put the secret key file to backend's directory and put your email and password credentials to the nodemailer.ts file
4. Open three different command prompts.
5. Enter socket-app, frontend, and backend folders respectively in these prompts.
6. Run "npm install" command in each prompt.
7. Run "npm start" command in each prompt.
8. The frontend will directly pop-up in your browser in localhost:3000 address, if it doesn't pop-up you can simply enter the localhost:3000 from a browser.

2. Coding and Implementation

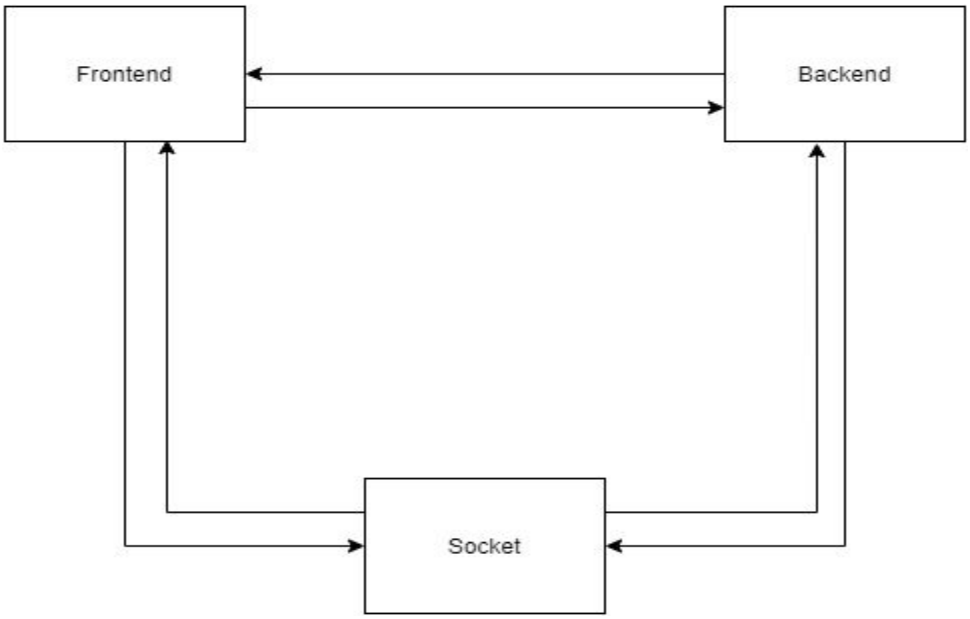


Figure 1: System Architecture

- Frontend

Frontend is written in React.js which is a Javascript library aimed to develop frontend.

The frontend has different aspects like user profiles, login screen, sign up screen, auction shop and specific auction screen.

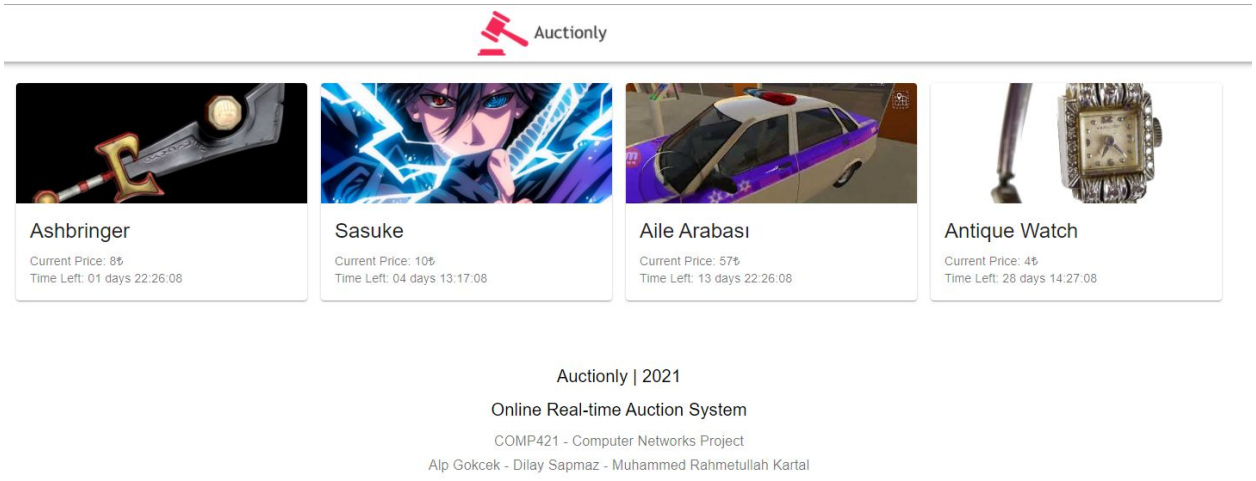


Figure 2: Auction Shop Screen

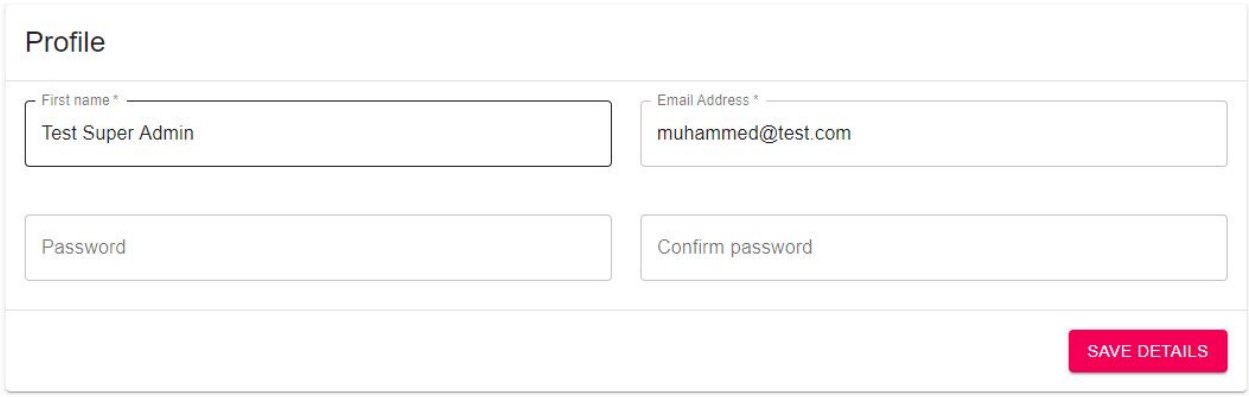
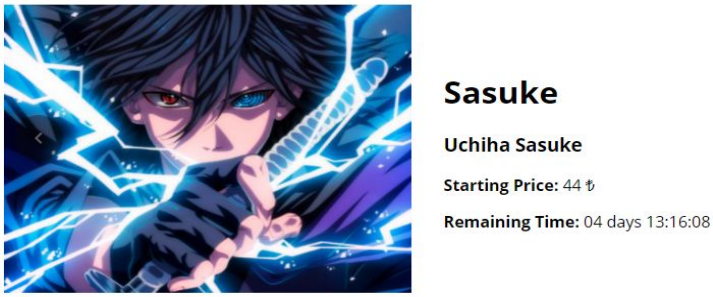


Figure 3: Profile Panel

Figure 4: Users Panel

Figure 5: Products That User Sold and Selling



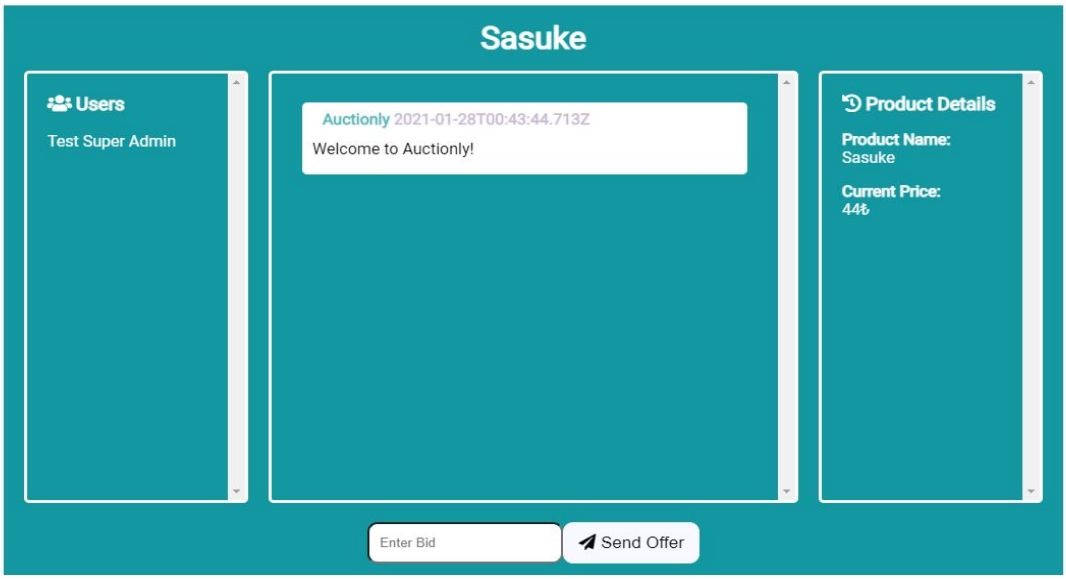


Figure 6: Product Screen

You can see the auction shop, profile panel, users panel that only the admin is able to see and the products panel that the user is currently selling and sold and the product screen in Figures 2, 3, 4, 5, 6.

- Backend

Backend is written in Typescript language which is an extension to Javascript to make the programming more static. We have followed the rules of REST API architecture. For the database and authentication systems, we have used Firebase Cloud Firestore and Firebase Authentication respectively. The backend is aimed to provide the data needed for both Socket.js server and frontend server.

- Socket

We have also implemented a separate application for Socket programming. This application is written in Javascript, HTML, CSS and Node.js's Socket.io library.

All users can log into the site and sell products. There are rooms on the site and users sell their products in these rooms.

Socket.io Functions

socket.emit: The method sends a message to the just one client. Used mostly to give a "Welcome" message to the user who enters the chat as you can see in Figure 7. If other users are already in chat, this message isn't going to be received by them.

Figure 7: Socket.emit example

io.emit: Sends a message to all the users in the chat. This message goes to everyone at the same time.

socket.join: Connects the socket to the chat.

socket.on: When a user writes a bid to the chat, this function takes this bit and displays it in real time so that people in the chat can see it.

io.to(user.room).emit: Writes a message to all users in the user's current room.

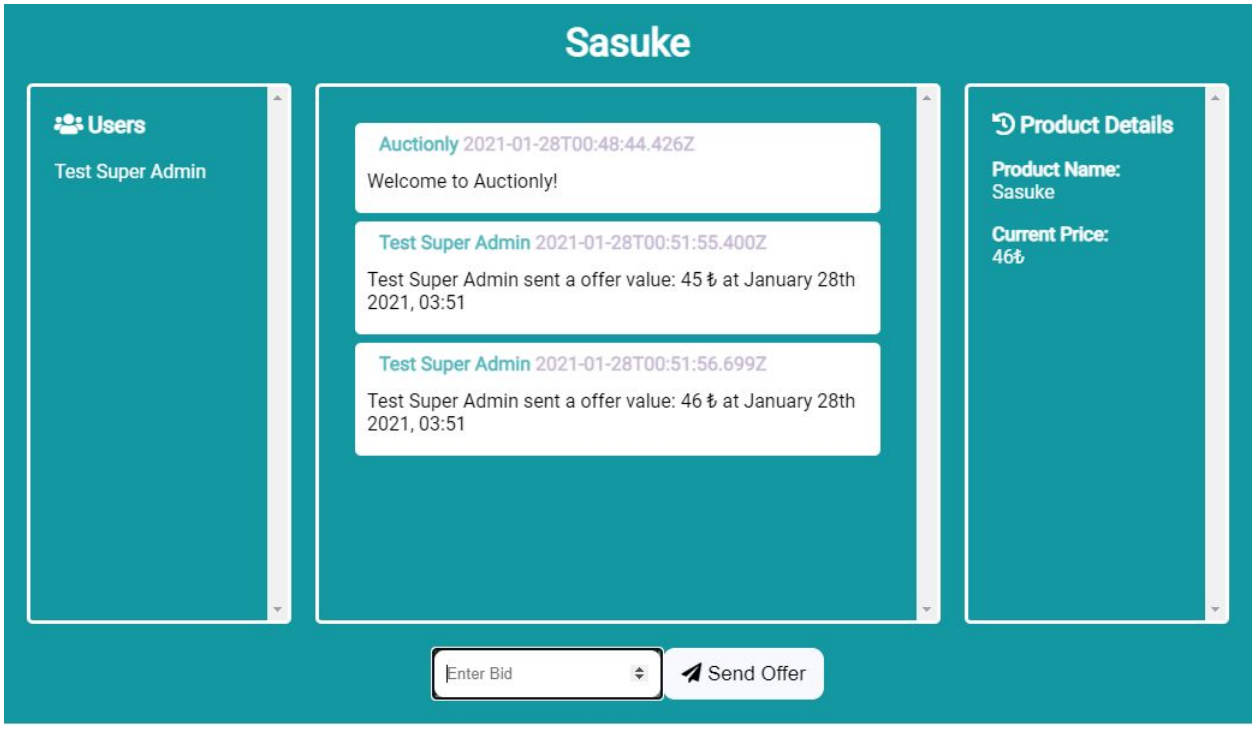


Figure 8: A Room of an Auction

3. Conclusion and Results

As a conclusion a real-time auction website is done by socket programming, frontend, backend and socket uses totally three servers and they are all connected. In the teamwork Alp designed the frontend, Dilay and Muhammed worked on the socket programming and after that Alp tested and debugged it, and all members worked on the backend together.