CSE354

Distributed Computing

Project Report

Logo

Description automatically generated

Made by

Seif Ahmed ElSayed Elewa 18P5662

Hana Yasser Amgad 18P5007

Ilaria Refaat Ghobrial 18P3050

Presented to

Dr. Ayman Bahaa

Eng. Mahmoud Ashraf

Contents

[Introduction 2](#_Toc107161276)

[Description 2](#_Toc107161277)

[Project Beneficiaries 2](#_Toc107161278)

[Analysis 2](#_Toc107161279)

[Project tasks and roles 5](#_Toc107161280)

[System Architecture 5](#_Toc107161281)

[Testing scenario 6](#_Toc107161282)

[User guide 8](#_Toc107161283)

[Conclusion 8](#_Toc107161284)

# Introduction

This report is the explanation and the documentation for the project of the distributed computing course. The project goal is to learn how distributed computing is applied and how server and client states are saved throughout any process. It was decided to be a google docs clone that perfectly demonstrates the distributed computing real time and wide spread nature. We decided to use angular framework as a full stack development approach since it would be easier to control and manage both sides.

# Description

This project is a multi-user distributed text editor using full stack angular framework. Different instances of it open on different computers, and when we change in one, we can see this change in the other one.

It also saves any changes made, so when we refresh the page, it is saved in a database on the server so we can always have access to it.

We built it using some tools such as:

* React
* mongo dB
* socket programming and deployment mediums to connect the server and the client together in real time and allow them to communicate correctly and efficiently at any time and from any place.
* Quill is used as a text editor as it is fully featured, it has all the features that we can need in a google doc style text editor and it has an important concept called Delta, where deltas are operations that we can perform to get from one step to the next state( every time we make a change it tells what change is made and where it is made) which is very important in our application because it helps preventing overlapping when two people are modifying the same document at the same time.

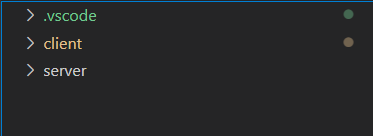
# Project Beneficiaries

As we develop our project, we thought about who the people are who would benefit from it and could use it. Since we use google docs as our reference to clone, we decided to look around us for those who use it. We discovered that it’s used a lot by students whether they are in school or in college, it is also used by instructors to communicate better with their students and their peers. Some companies also use it to help organize the work and the system of their organization and their employees.

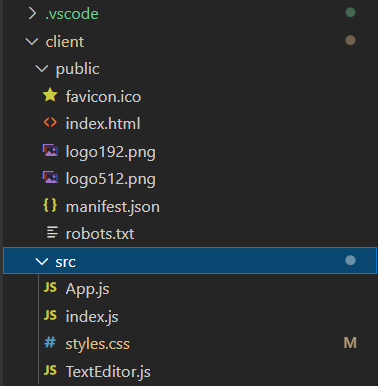
We even use google docs with our colleagues to organize our deadlines and tasks for each week throughout the year in a document called CESS repository. And even in this project we used it to prepare our report together to make sure we are all on the same page.

# Analysis

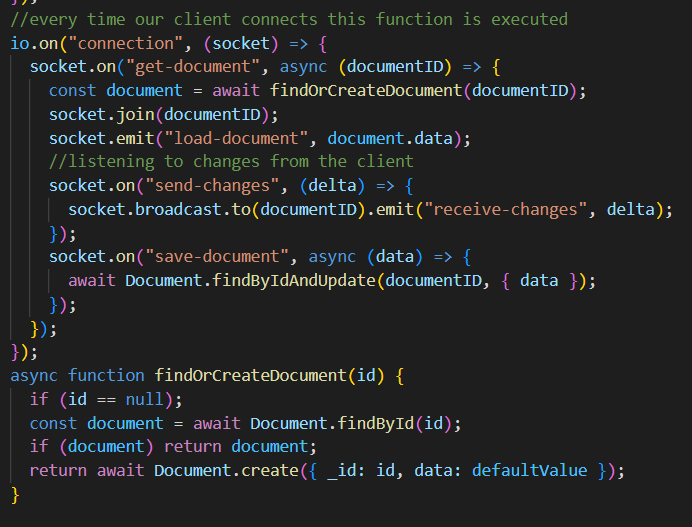
Our project consists of 2 folders, a client and a server.



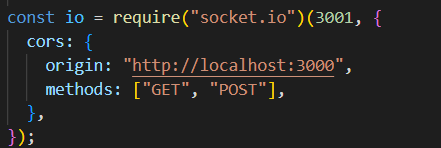
Client folder is where the text editor front end is designed.



Server folder contains socket io code which is used to make real-time connections between client and server.

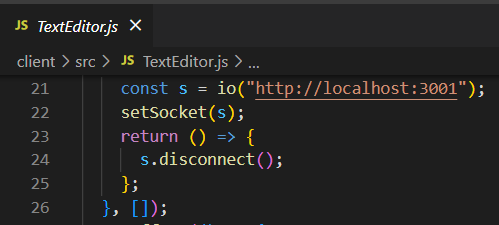


Since client and server each on a different port we used cors which is for cross origin request support, which allows to make request from a different URL to a different URL (client and server are each on a different URL) (client on port 3000, sever on port 3001).

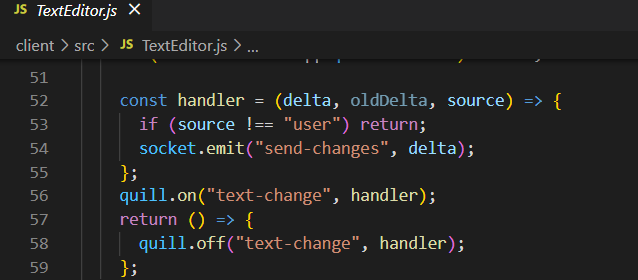


To make a connection we put client URL on the server side and the server URL in the client side after importing io from socket.io-client.

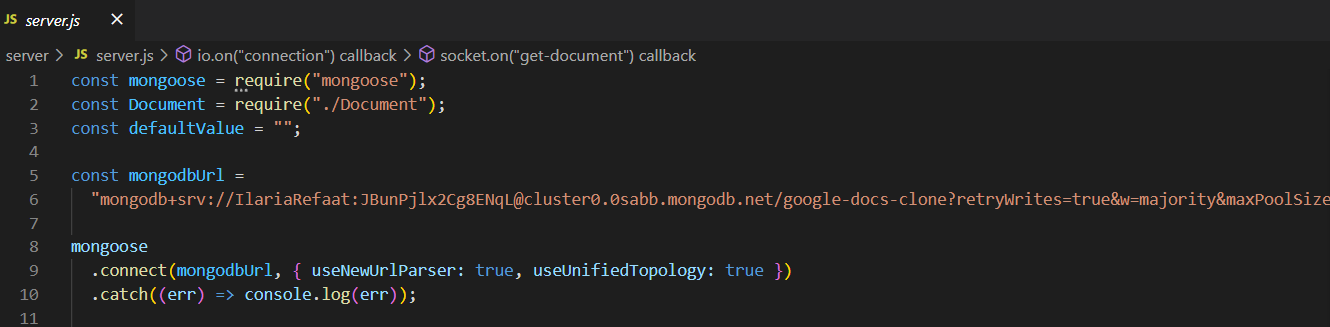




The system checks that every time a change is made this change is made by the user, otherwise the system doesn’t track this change. Every time a change is made, we send to the server that a change has been made using the send changes event. A broadcast is then made to send to every other user that there are some changes that they should receive, and here are those changes.



The system is connected to a database to save all document data in real time.

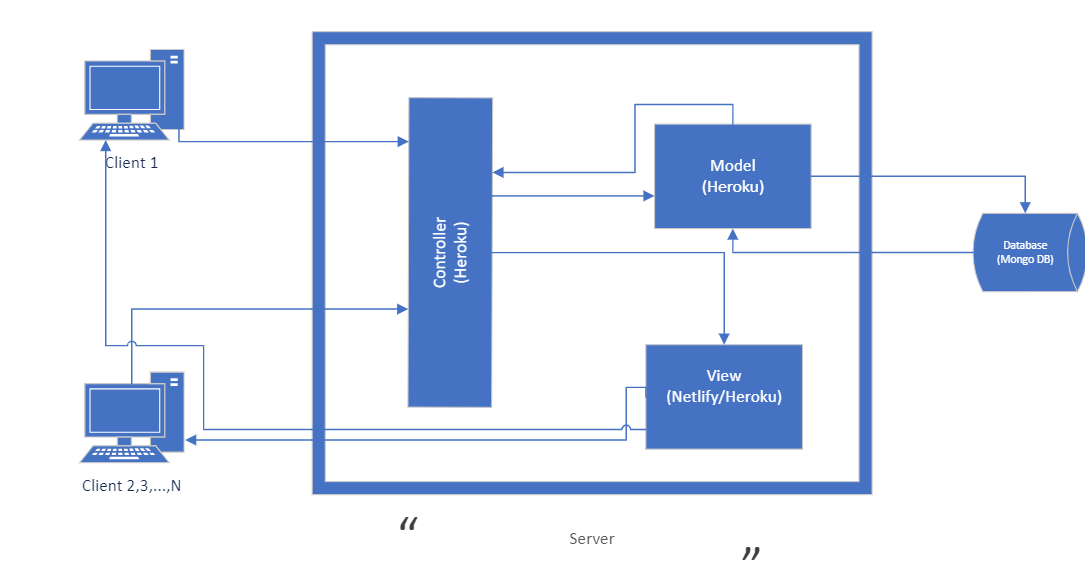


# Project tasks and roles

* Writing the server side : this task consisted in writing the server side and setting up the socket connection along with the database connection. – **Seif and Ilaria**
* Writing the client side : this task consisted of connecting to the server and making sure the users are tracked. – **Hana and Ilaria**
* Preparing the user interface : this task consisted of preparing the Quill extension used for the editor itself and making sure everything ins working correctly and choosing our desired options , layout, and design. – **Hana**
* Connecting the database : this task consisted in creating the database and allowing its access from anywhere while making sure its connection to the server is successful. – **Ilaria**
* Deploying the app : this task consisted of deploying our server and our client to our hosting website ( Heroku , Netlify) and testing their connection together. – **Seif**
* Writing the report : this task consisted of preparing the report , Readme file and the video needed as documentation and proof for the project – **All of the team members.**

# System Architecture

Client server model using the MVC concept to link all the aspects of our project and implement the distributed systems idea correctly.



# Testing scenario

For more details about the testing, you can check the [Demo and Testing video](https://drive.google.com/drive/folders/1GSHoCd0KOyZxfy6F86cHqX9L8dLHlpBI?usp=sharing).

You can also access the [GitHub](https://github.com/IlariaRefaat/Google-Docs-Clone).

Graphical user interface, text, application

Description automatically generatedWe tested our document by typing from several devices at the same time and it worked correctly saving the state of each client along with the document.

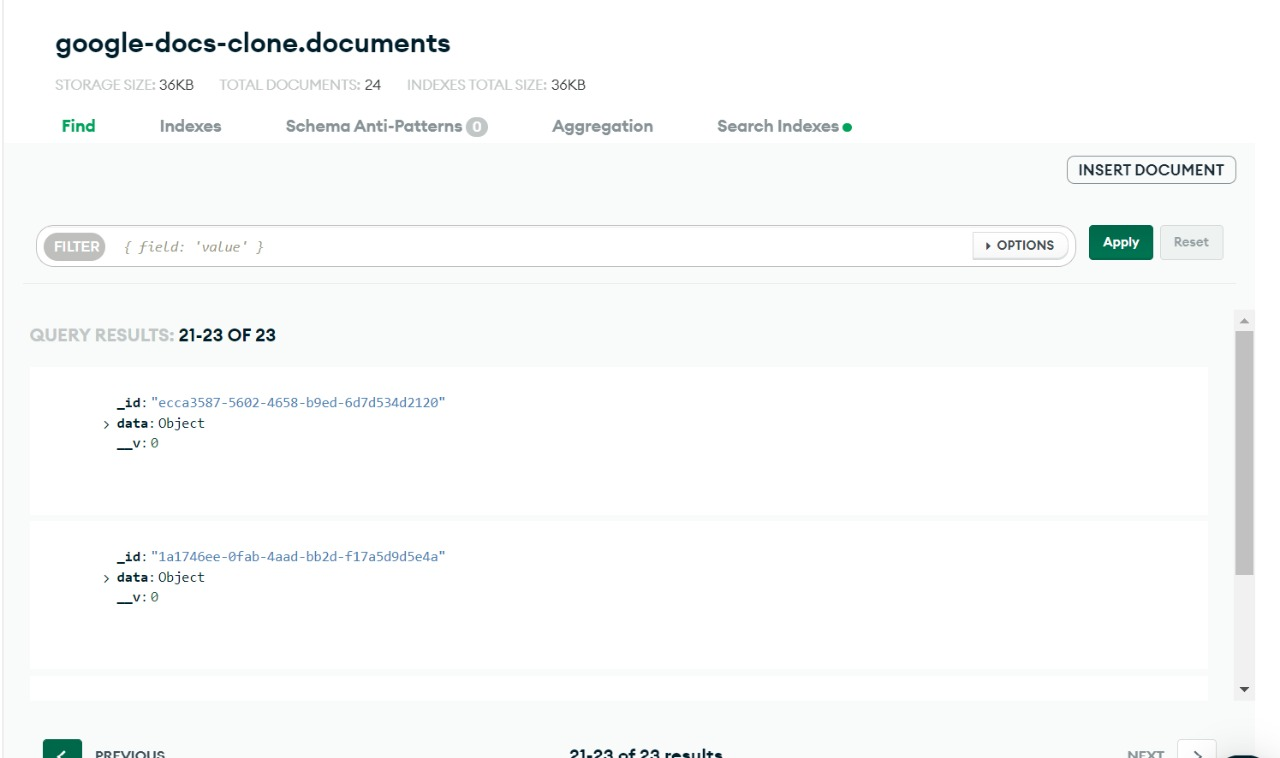
Graphical user interface, text

Description automatically generatedWe also tested changing the document and re-opening the page where it was indicated that the state was saved successfully.

Graphical user interface, text, application

Description automatically generatedAs we can see each document opened has its own unique ID that’s saved in our database.

As we can see in our database , we have the latest created documents that are also used for the demo and testing.



# User guide

To start the application, you just have to go to any of the two following links (each is its own separate document)

[Heroku deployed](https://text-editor-client.herokuapp.com/#/document/1a1746ee-0fab-4aad-bb2d-f17a5d9d5e4a)

[Netlify deployed](https://62b75745ca488e2b05b295ca--precious-mooncake-04214e.netlify.app/#/document/ecca3587-5602-4658-b9ed-6d7d534d2120)

# Conclusion

In conclusion this document shows how we built our own custom google docs which is a distributed system that gives the ability to multiple users to use the system and interact with each other through it. This system is very beneficial to college students in college and school and instructors. We presented a detailed description of the main idea with tools used. Also, there is an analysis of how the project works (main files and how connection is made) with screenshots from the code.