

# Distributed Computing: Final Project



Program: Computer and Software Engineering

Course Code: CSE-354

Course Name: Distributed Computing

Examination Committee

**Dr. Ayman Bahaa-Eldin**

Ain Shams University

Faculty of Engineering

Spring Session – 2022

Maryam Mohamed Mohamed Mahmoud  
Abdelrahman 18P8171

Mohamed Mamdouh Ibrahim 18P8906

Safia Medhat Abdulaziz Elsayed 18P7367

## Contents

Introduction: .....	3
Detailed project description: .....	3
Beneficiaries of the project: .....	5
Detailed analysis: .....	6
Heroku:.....	6
Aws s3:.....	6
Socket programming: .....	8
Breakdown of the tasks: .....	9
Role of each member: .....	9
System architecture & design: .....	9
Test cases scenarios.....	11
End-user guide: .....	11
Conclusion: .....	11
References: .....	12

## Table of Figures

Figure 1 File Id and password.....	3
Figure 2 Number of people connected .....	4
Figure 3 connection.....	9
Figure 4 MVC Model.....	10

## Introduction:

The project is an online text editor that allows many users to join through entering a password and allowing you to connect to the same editor. There are some features in the text editor including changing the style, bold, italic, underline, ordered list, unordered list and reverse the styling.

## Detailed project description:

To allow the text editor to be working we used socket programming, peer-to-peer, Heroku, and AWS.

Socket programming allows users to use same text editor and share any edit concurrently if they connect to same server

when the client connects to server, he/she write file ID and the password generated when the person who created the file and pressed share. The file id and the password will be visible to that person, and we can share with other people to join this specific file and edit together.

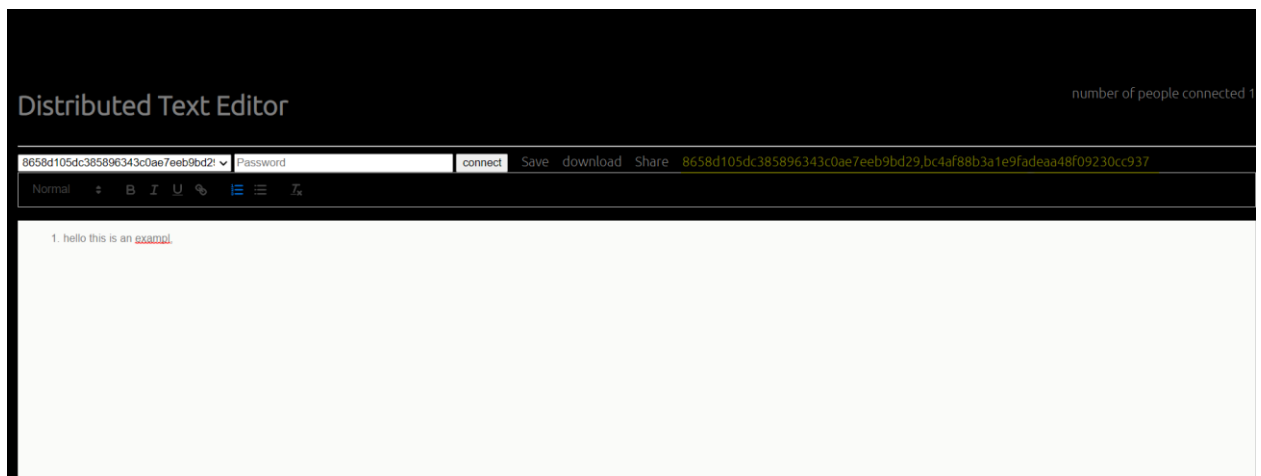


Figure 1 File Id and password

The highlighted part in the picture is the file Id , Password

we used Heroku server for 2 main reasons

1-to save the document on Heroku servers

2- host the code on Heroku

we used Heroku because it has file system and it can act as a host. it also do not change the root file for 12 hours.

we used quill editor as text editor because it is handled by java script.

Peer-to-peer:

We made this project to be decentralized which mean we don't need every time to request data from database. so, when I modify the document i will not send to the server and will send to rest

*Figure 2 Number of people connected*

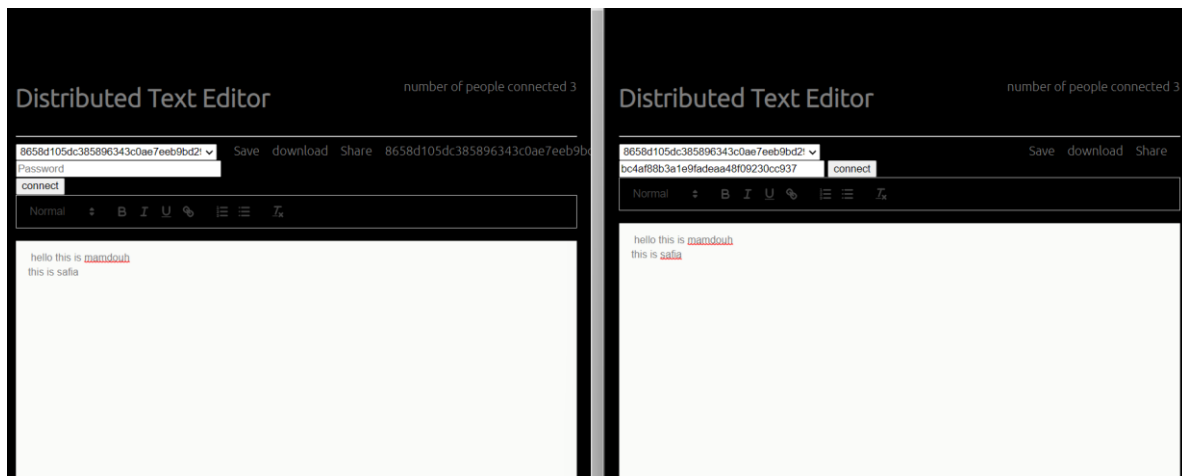
of the users immediately. But if the document is modified by someone else then I will not do anything. We share the document by the document. we make search using buck chain. We used crypto function to generate unique code name. We used p2p so that we don't fall in performance bottleneck when the number of users is big. And simply the user who edit the document just send to others.

What is fault tolerance?

Fault-tolerant distributed figuring alludes to the algorithmic controlling of the distributed framework's parts to offer the ideal support notwithstanding the presence of unavoidable disappointments in the framework by taking advantage of overt repetitiveness in reality.

How did we make our system fault tolerant?

We created two buckets and uploaded all the documents on both buckets so, if one bucket fails we can get the file from the other bucket. If one of the buckets was down and after sometime it was up again, it will reconnect to get the last version of the document.



What is consistency?

Consistency is a property of the distributed system which guarantees that each node or imitator has similar perspective on information at a given time, regardless of which client has refreshed the information. Solid consistency would imply that the distributed system merges on a solitary worth, and the client generally peruses the most recent information.

How did we make our project consistent?

If a user was connected and then their internet went down and he kept editing the document. When his internet connection gets back and he reconnects to the document he will get the latest version without his editing to make it consistent with the other users.

What is robustness?

A distributed system might experience the ill effects of different kinds of equipment disappointment. The disappointment of a connection, the disappointment of a site, and the passing of a message are the most widely recognized types. To guarantee that the system is robust, we should identify any of these disappointments, reconfigure the system so calculation can proceed, and recuperate when a site or a connection is fixed.

How we made our project robust?

When you close the session and reopen it the latest version of the document is what is received

## Beneficiaries of the project:

### 1. No Need To Download

One of the significant downsides of a text editor program for authors in a hurry is their need to be downloaded. Few out of every odd essayist has a reasonable association and downloading these projects can take a ton of time thus. That is when online word processors step in and help essayists out. Due to their virtual nature, these content managers have "on the web" before their name. These web applications assist you with editing content in light of their sites, without the need of downloading at all. In retrospection, these devices assist scholars with saving a tremendous lump of time thus.

### 2. They're Free

A ton of the content editing programs on PCs or cell phones aren't free. Any program that permits you to alter or design text includes some major disadvantages. In any case, these web-based word processors are liberated from those limits too, straightforwardly. As they don't need a penny to be utilized. In all actuality, some of them could have commercials, yet that is how they stay above water. Thus, when you consider the advantages these projects give, those commercials could not hope to compare to the benefits you have by simply utilizing these altering instruments.

### 3. High Convenience

When did you last need to alter a text earnestly however were not near your workstation? If by some stroke of good luck, you had known about a web-based content tool, it would have been such a great deal simpler. This is the sort of issue that a ton of essayists face, especially currently. When the greater part of the work is remote-based, these web-based editors permit essayists to address their substance in a hurry. This accommodation type isn't exceptionally normal in introduced or downloaded programming. In this way, utilizing these improvised devices can give a ton of convenience to scholars today.

#### 4. Compose On the Go

Assume you're accustomed to hauling a PC around. All things considered; all you want is a steady wifi connection for you to access these projects. A large portion of these web-based word processors have cloud-based administrations. Thus, they can assist you with working together with your kindred scholars and guarantee that you compose in a hurry with practically no issue.

### Detailed analysis:

#### Heroku:

Heroku is a generally depended upon stage as a help offering that empowers designers to do bother free application organization, scaling, and the executives. This stage offers support for an extensive variety of programming dialects like Java, Ruby, PHP, Node.js, Python, Scala, and Clojure. Heroku runs applications through virtual compartments known as Dynos.

Heroku charges its clients in view of the quantity of virtual machines that are fundamental for their applications. The Heroku stage and client made applications use Amazon Web Services as the fundamental foundation. Engineers can accomplish quick application advancement involving it as it is very helpful.

#### What is a Dyno?

A Dyno is a compartment on the Heroku stage used for running and scaling Heroku applications. They are generally virtual Linux compartments utilized for running code in light of client orders.

Applications can be increased to explicit quantities of Dynos in view of the necessities of designers. Heroku offers compartment the executives elements to assist clients with performing easy scaling and dyno size, type, and number administration in light of use prerequisites.

Dynos are the essential components fueling up a Heroku application. Engineers can send their applications to Dynos and deal with these units to make and run adaptable applications without any problem.

They get independence from being required to perform framework the board assignments and can rather zero in on the significant parts of making and running applications.

#### Aws s3:

Amazon S3 is a program that is worked to store, secure, and recover information from "pails" whenever from anyplace on any gadget. Associations of any size in any industry can utilize this

help. Use cases incorporate websites, portable applications, documenting, information reinforcements and rebuilding efforts, IoT gadgets, endeavor application capacity, and giving the basic stockpiling layer to your information lake.

### How Does Amazon S3 Work?

Sorting out, putting away and recovering information in Amazon S3 centers around two key parts: buckets and articles that cooperate to make the capacity framework. As AWS depicts it, a S3 climate is a level design — a client makes a bucket; the bucket stores objects in the cloud.

#### Amazon S3 Objects:

As referenced above, in Amazon S3 terms, objects are information records, including reports, photographs, and recordings. Each object is distinguished by an exceptional key inside the S3 climate that separates it from other put away objects. The most extreme object document size is 160 GB for transferring, but there are different AWS devices to assist you with adding records bigger than this.

In a S3 climate, objects need some place to go, which is the reason buckets exist, filling in as key stockpiling compartments for objects.

#### Amazon S3 Buckets:

You can make up to 100 buckets in every one of your AWS cloud accounts, with no restriction on the quantity of objects you can store in a bucket. If necessary, you can demand up to 1,000 additional buckets by presenting a help limit increment.

At the point when you make a bucket, you can pick the AWS district to store it in. To limit expenses and address dormancy concerns, it's best practice to choose a locale that is geologically nearest to you. Objects that live in a bucket inside a particular district stay in that locale except if you move the records somewhere else.

It's additionally essential to realize that Amazon S3 buckets are worldwide extraordinary. No other AWS account in a similar locale can have a similar bucket name as yours except if you initially erase your own buckets.

### For what reason Should I Consider Using Amazon S3?

On the off chance that you're as yet not certain if Amazon S3 is ideal for your association, think about this: Amazon S3 is intended for 99.999999999% (11 9s) of information strength. With that degree of sturdiness, you can anticipate that on the off chance that you store 10,000,000 objects in Amazon S3, you ought to just hope to lose a solitary object like clockwork!

Amazon S3 consequently makes and stores duplicates of all transferred objects across different frameworks, permitting your information to be safeguarded against disappointments, mistakes, and dangers and accessible when required.

## Socket programming:

### What is a Web Socket?

Web Socket is a convention that gives full-duplex(multiway) correspondence i.e permits correspondence in the two headings at the same time. It is a cutting-edge web innovation wherein there is a nonstop association between the client's browser(client) and the server. In this kind of correspondence, between the web server and the web program, the two of them can send messages to one another anytime. Generally, on the web, we had a solicitation/reaction design where a client sends a HTTP demand and that's what the server answers. This is yet relevant generally speaking, particularly those utilizing RESTful API. However, a need was felt for the server to likewise speak with the client, without getting polled (or mentioned) by the client. The server ought to have the option to send data to the client or the program. This is where Web Socket comes into the image. To utilize the Socket in NodeJS, we first need to introduce a reliance that is socket.io. We can just introduce it by running the underneath order in cmd and afterward add this reliance to your server-side JavaScript document likewise introduce an express module which is fundamentally expected for server-side application.

```
const express = require('express'); // using express

const socketIO = require('socket.io');

const http = require('http')

const port = process.env.PORT||3000 // setting the port

let app = express();

let server = http.createServer(app)

let io = socketIO(server)

server.listen(port);
```

Now we need to make a connection from the server-side to the client-side through which the server will be able to send data to the client.



```

on("connection", async (socket) => { //when a connection is established the function below runs
  try {
    const result = await db.query('SELECT name from files')
    socket.emit('files', result.rows)
  } catch (error) {
    console.log(error)
  }

  let currentRoom = '' //define where am i connected currently ,meaning which distributed document or if its a new document
  socket.on('join', async (roomName, password) => {
    socket.to(currentRoom).emit('leaving', socket.id)
    socket.leave(currentRoom)
    try {
      const result = await db.query('SELECT name from files where name=$1 AND password=$2', [roomName, password])
      if (result.rowCount != 1) {
        socket.emit('init', 'error', 'Incorrect Creds')
        return
      }
    } catch (error) {
      socket.emit('init', 'error', 'Incorrect Creds')
      return
    }
    const sockets = await io.in(roomName).fetchSockets();
    socket.join(roomName)
    socket.to(roomName).emit('joining', socket.id)
    if (sockets.length === 0) {
      try {
        const { Body } = await s3Client.send(new GetObjectCommand({
          Bucket: bucket1, // The name of the bucket. For example, 'sample-bucket-101'.
        }));
      } catch (error) {
        // Handle error
      }
    }
  })
})

```

Figure 3 connection

Presently either information can be sent from any side so an association is created among server and client. Then, at that point, if the server radiates a message, the client can pay attention to that message or on the other hand in the event that the client produces a message, the server can pay attention to that message. So, we need to create an attachment for the two messages emitted and message tune in on both the server and the client-side.

## Breakdown of the tasks:

1. We made the layout of the website
2. Then we kept thinking about the functionalities figuring out how to get the updates real time
3. We started using peer-to-peer and socket programming
4. We used sockets to emit the changes for the document
5. The other computer should get the updated document without receiving a message that it got updated
6. Then we thought about the storage of the data and we used s3 buckets
7. We used two buckets incase of failure, if one fails the other steps up and takes its place.
8. We used lambda function to sync between the two buckets and if there is a connection loss in one of the users, when is back he will get the updated version.

## Role of each member:

We were all available to join meetings together and each time we were shifting between coding, researching and brainstorming.

## System architecture & design:

we used MVC model. we divided the application in to 3 parts: model, view and controller.

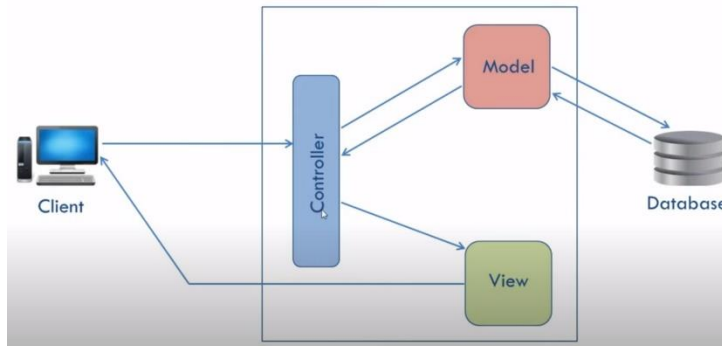


Figure 4 MVC Model

View: html and CSS.

Model: are the document that is sent and received.

Controller: the code on the server side.

What is MVC Framework?

The Model-View-Controller (MVC) system is an engineering design that isolates an application into three fundamental consistent parts Model, View, and Controller. Subsequently the truncation MVC. Every engineering part is worked to deal with explicit improvement part of an application. MVC isolates the business rationale and show layer from one another. It was customarily utilized for work area graphical UIs (GUIs). These days, MVC engineering in web innovation has become famous for planning web applications as well as portable applications.

View:

A View is that piece of the application that addresses the introduction of information.

Sees are made by the information gathered from the model information. A view demands the model to give data, so it presents the result show to the client.

The view additionally addresses the information from outlines, graphs, and tables. For instance, any client view will incorporate all the UI parts like text boxes, drop downs, and so on.

Controller

The Controller is that piece of the application that handles the client communication. The controller deciphers the mouse and console inputs from the client, illuminating model and the view to change as suitable.

A Controller send's orders to the model to refresh its state (E.g., Saving a particular report). The controller likewise sends orders to its related view to change the view's show (For instance looking over a specific report).

Model

The model part stores information and its connected rationale. It addresses information that is being moved between controller parts or some other related business rationale. For instance, a Controller item will recover the client data from the information base. It controls information and sends back to the data set or uses it to deliver similar information.

It answers the solicitation from the perspectives and furthermore answers directions from the controller to refresh itself. It is additionally the most minimal level of the example which is answerable for keeping up with information.

## Test cases scenarios

1. If a user that was connected to the document had internet problems, and kept editing, his updated doc won't be sent to all the other users.
2. If one of the buckets fails, the other bucket will be working because when we created 2 buckets, we made sure that one was for back-up, and we used redundancy for the files.
3. If two users were writing in the same line both of their writing will overlap each other

## End-user guide:

This will be a guide for users to be able to run this project:

1. Please copy and paste the url: <https://distributedtextedit.herokuapp.com/>
2. Either choose an existing file or create a new file, in both cases once you press the share button you will get the password for each so you can join.
3. Enter the file ID and password in their respected places and click the connect button.
4. In the top right corner you can see how many people joined the same file.
5. There are some features you can use like choosing the style, bold, italic, underline, ordered list, unordered list and the last button is to remove any style by highlighting the line/paragraph you want then pressing that button.
6. You can download the file to your computer by pressing the download button.

## Conclusion:

From this project we learnt that distributed systems are the wave of the future. In case you didn't notice, we've stopped using massive mainframes and supercomputers to solve problems. Now that we have whole server farms, the issue is spread across enormous processor arrays. Working with them is difficult. Failures are challenging to rationalise with. There will be behaviour that you never expected. When you can create a distributed system that really performs useful work, it is incredibly satisfying. Just be ready to put in the effort to get there and the effort needed to keep it going.

## References:

1. Martin, M. (2022, April 30). MVC Framework tutorial for beginners: What is, Architecture & Example. Guru99. Retrieved June 24, 2022, from <https://www.guru99.com/mvc-tutorial.html>
2. Web-socket in Node.js. GeeksforGeeks. (2022, February 21). Retrieved June 24, 2022, from <https://www.geeksforgeeks.org/web-socket-in-node-js/>
3. Steen, M. V. (2022, March 20). Distributed systems 3rd edition (2017). DISTRIBUTED. Retrieved June 24, 2022, from <https://www.distributed-systems.net/index.php/books/ds3/>
4. What is Heroku? price, features, benefits, and competitors. (n.d.). Retrieved June 24, 2022, from <https://blog.back4app.com/what-is-heroku/>
5. Chrisrepair. (2022, March 1). What are the benefits of using online text editors " androidtechvilla. AndroidTechVilla. Retrieved June 24, 2022, from <https://androidtechvilla.com/benefits-of-using-online-text-editors/>
6. Guilder, G. V. (n.d.). What is Amazon S3 and why should I use it? Dark2. Retrieved June 24, 2022, from <https://www.onixnet.com/insights/what-is-amazon-s3>