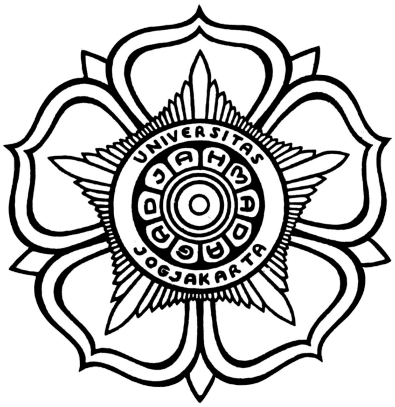
Final Project Planning - Collaborative Text Editor Using Socket Programming



Group 5

Audrey Shafira Fattima	**(21/472678/PA/20320)**
Moehammad Azzriel Ilham	**(21/477994/PA/20724)**
Ramzy Izza Wardhana	**(21/472698/PA/20322)**

COMPUTER SCIENCE STUDY PROGRAM

DEPARTMENT OF COMPUTER SCIENCE AND ELECTRONICS

FACULTY OF MATHEMATICS AND NATURAL SCIENCES

UNIVERSITAS GADJAH MADA

YOGYAKARTA

2022

CHAPTER 1
INTRODUCTION

We are obliged to be entirely digital in our daily lives in this digital world. Even when we're performing our jobs, we are frequently expected to work in groups. Whether as students or as employees in a company or organization, we are often assigned to group tasks. As students, we may be required to work in groups when given group assignments or while doing non-academic organization tasks. Or perhaps, as employees, we are continuously pushed to work in groups to finish the job.

As a result, it is critical that we have a platform that allows us to effectively collaborate remotely and gives us the flexibility to conduct our jobs from anywhere. We may have previously encountered services in our everyday lives such as Visual Studio Live Share, which is used for developer collaboration, Google Docs for document collaboration, FigJam or Figma for designer collaboration, and many others.

Because of this, the goal of this project is to develop a collaborative text editor utilizing socket programming. Since we frequently use collaborative text editors for group assignments and other non-academic organization tasks that call for teamwork, students often feel the need for one. These platforms will allow us more flexibility to collaborate in teams remotely. In the chapters that follow, the specifics of this project will be covered.

CHAPTER 2
SOLUTION

Our solution came from the problem that students and companies mainly encounter, that is group projects. Most of the time, group projects require collaborative workspaces and platforms to edit documents like reports and papers. As a result of this problem, we want to develop a simple platform of collaborative text editor or commonly known as Google Docs clone by using the concept of socket programming that we have been studying in the Computer Network Lab. Our expected result for this project is mainly to enable clients to use the text editor and share the content progress in real-time with other clients connected through

a server socket so that other clients will be able to view and edit text simultaneously. We realized that cloning Google Docs as a whole is a sophisticated challenge for us. Therefore, in the next chapter, we will elaborate on things that we will focus to achieve and develop.

CHAPTER 3
PLANNING

In this chapter, we will address more detail regarding our planning and goals in order to achieve the final result:

Goal:

1. Clients able to connect to each other through local server by using socket programming method.
2. Collaborative text editor can be used by more than one clients.
3. Realtime and seamless text editor accessible by all other clients on the same local server through web browser.
4. A simple yet intuitive GUI for the clients to manipulate text such as bold, italic, underline, bulleted points, and so on.
5. Responsive web to support multiple sizes and dimensions of browsers.

We will implement our project in a web-based format and to accommodate this, tech stack that we are currently planning to use are:

1. React (Javascript library for front-end and back-end)
2. TailwindCSS (A CSS utility-first framework to support responsiveness and efficiency of GUI)
3. Socket.io (Javascript library to fully support the connection between clients and server)
4. MongoDB (A back-end support to save user's last progress to database) *low priority

Although all our plans and goals have been structured accordingly, we will ensure that our top priority in this group project is to be able to achieve our main functionality, which

is a collaborative text editor. As a consequence, additional features such as a beautiful GUI, a back-end database to store the user's last progress, and any other improvements will not be our mandatory features as the development period is quite short. Therefore, we will optimize our time and plan as efficiently as possible to develop our most viable and optimal result.