National College of Ireland

Increasing service capacity of peer-to-peer file sharing networks by using a decentralized reputation system

MSc Research Project
MSc in Cloud Computing

Ibrahim Ayodeji
Student ID: x20227329

School of Computing
National College of Ireland

Supervisor: Shivani Jaswal

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ibrahim Ayodeji
Student ID:	x20227329
Programme:	MSc in Cloud Computing
Year:	2022
Module:	MSc Research Project
Supervisor:	Shivani Jaswal
Submission Due Date:	15/08/2022
Project Title:	Increasing service capacity of peer-to-peer file sharing networks by using a decentralized reputation system
Word Count:	2000
Page Count:	5

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	19th September 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Increasing service capacity of peer-to-peer file sharing networks by using a decentralized reputation system

Ibrahim Ayodeji
x20227329

1 Introduction

This project presents a novel approach to improving service availability in peer-to-peer networks. In order to achieve this, a web application is developed with the supporting backend infrastructure as well as a simulator to evaluate the system. The project artifact consists of 4 folders for the 4 applications that make up the project.

- Backend: The application for the signalling server. Currently hosted at https://peer-rep-back.herokuapp.com/.
- Blockchain: This is the application for running blockchain system with ganache. Currently hosted at https://peer-rep-block.herokuapp.com/.
- Frontend: This is the user client web application. Currently hosted at https://peer-rep.web.app.
- Bots: This is the simulation system. Should be run locally for the most accurate results.

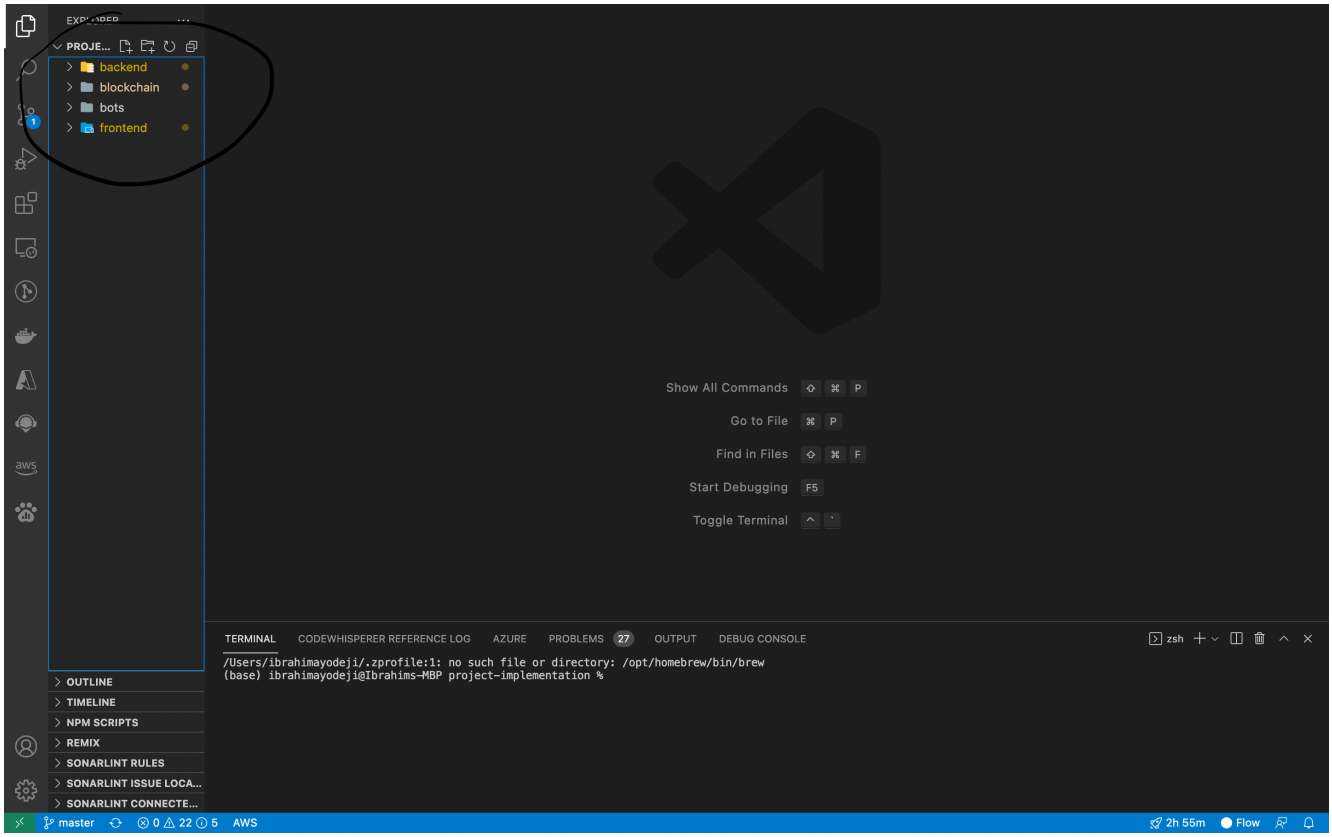

Figure 1: Folder Structure.

2 Technologies Used

This section goes over some technologies used and how to install them.

2.1 NodeJS

Open source, cross-platform Node.js is a JavaScript runtime environment. Node.js runs the V8 JavaScript engine external to the browser. The actions listed below should be followed in order to install nodeJS.

- Step 1: Go to https://nodejs.org/en/download/.
- Step 2: Select your platform and download the installer.
- Step 3: Run the installer and follow the instructions.

In order to run the applications locally, navigate to the project directory in the terminal and run "npm install". Once that is completed, run the commands "npm run build", then "npm run start". In order to test system, both blockchain and backend servers have to be running first.

2.2 Heroku-cli

In order to deploy our nodeJS applications to the internet, we employ the use of the Heroku cloud service. To use this service we have to create a "dyno" compute engine on the platform. The free tier features dynos with 512MB RAM and a single CPU core. One the application has been created use the following steps:

- On a MacOS system, pass the following command into the terminal brew tap heroku/brew && brew install heroku, while on a Windows machine, use https://cli-assets.heroku.com/heroku-x64.exe to download the installer then run it. From the terminal type heroku login and pass credentials.
- Once installed, navigate to the backend folder directory and make sure git is initialised. If not, then initialize it.
- In the terminal in the backend directory, use the following commands:
- – heroku git:clone -a ¡APP_NAME¿
  – git push heroku master

2.3 Firebase

Googles firebase service is used to host the front end application. In order to use firebase, the user must register on the firebase service then install the firebase-cli from https://firebase.google.com/docs/cli. Once installed, run "firebase login" in the terminal. open the terminal at the project directory and run firebase init and follow the prompts setting the public directory "build". Run "npm run build", the "firebase deploy".

3 Hardware Configuration

The system was tested locally using the following hardware specifications.

- Processor Type: 64-bit Apple M1 chip with 8 cores.
- RAM: 8GB.
- Disk capacity: 256GB.
- OS: MacOS Big Sur

3.1 Bot Simulation

To perform analysis using the simulator, the blockchain and backend services must be running. Following the setup instructions as described above for NodeJS applications. To set up parameters in the "bots" directory navigate to src/config/SIMULATION_CONFIG.ts and adjust the parameters. The results of the simulation are stored in the "reports" folder.

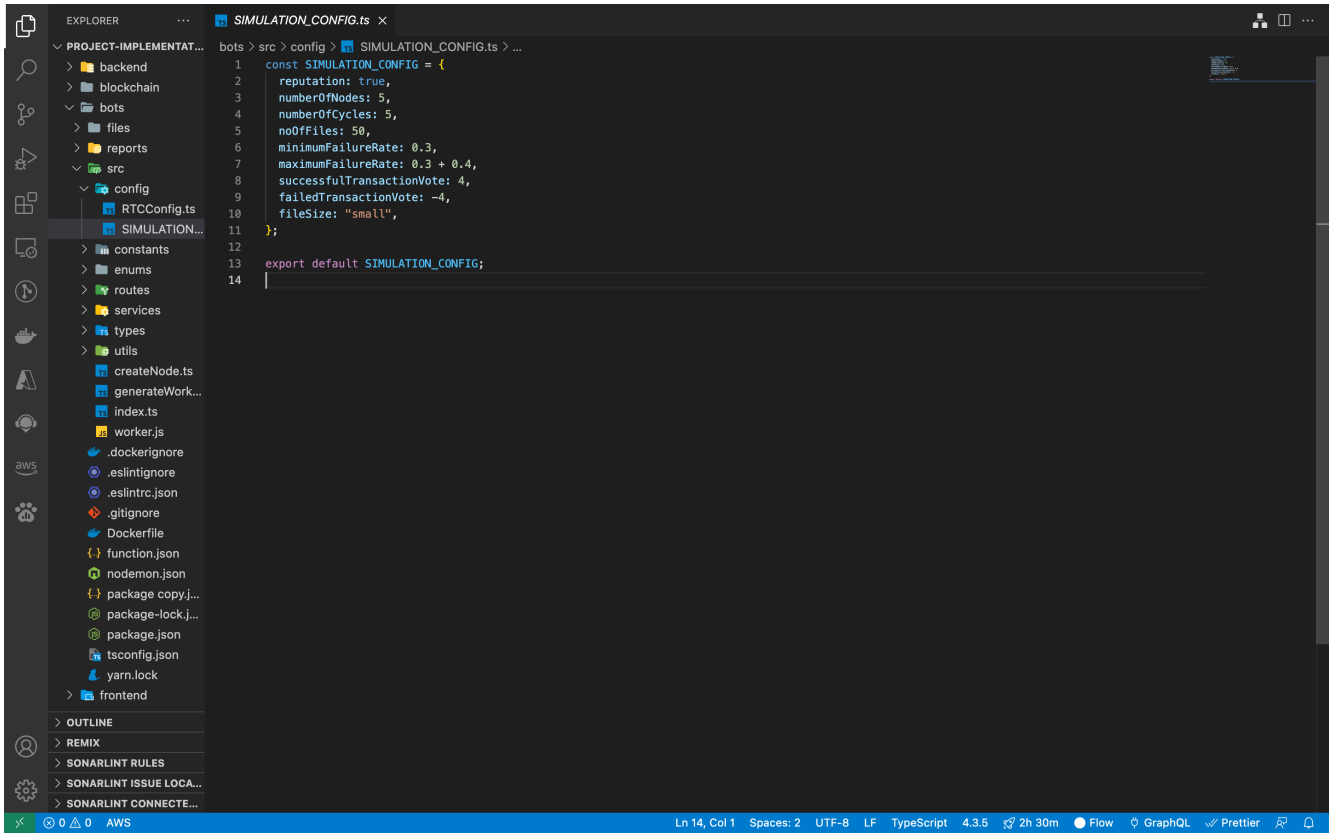

Figure 2: Simulation Configuration.

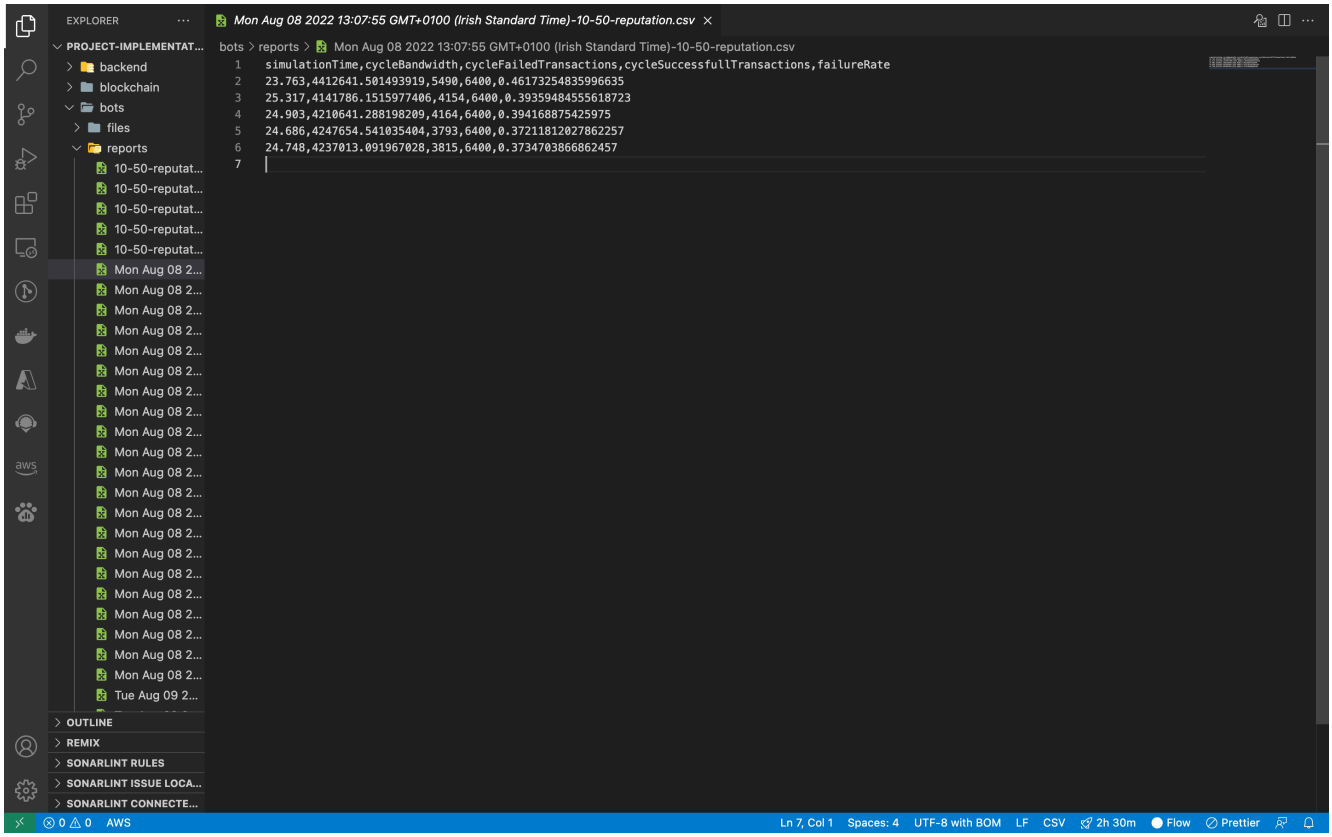

Figure 3: Simulation Report.