EXPEDITING HR MANAGEMENT VIA DYNAMIC E-PORTFOLIO BY EMPLOYING WEB SCRAPPER

A PROJECT REPORT

Submitted by

JEEVAN AKSHAY B	**211419104113**
KEVIN CHRISTOPHER A	**211419104140**
MOHAMMED ABDULLAH	**211419104167**

in partial fulfillment for the award of the

degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2023

PANIMALAR ENGINEERING COLLEGE
(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"EXPEDITING HR MANAGEMENT VIA DYNAMIC E-PORTFOLIO BY EMPLOYING WEB SCRAPPER"** the bonafide work of **JEEVAN AKSHAY B [REGISTER NO. 211419104113], KEVIN CHRISTOPHER A [REGISTER NO. 211419104140]**, and **MOHAMMED ABDULLAH [REGISTER NO. 211419104167]** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.L.JABASHEELA , M.E ., Ph.D., **HEAD OF THE DEPARTMENT**	**Mr.A.KARTHIKEYAN, B.E, M.TECH.,** **ASSISTANT PROFESSOR** **SUPERVISOR**
DEPARTMENT OF CSE, PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, POONAMALLEE, CHENNAI-600 123.	DEPARTMENT OF CSE, PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, POONAMALLEE, CHENNAI-600 123.

Certified that the above mentioned students were examined in End Semester Project Viva-Voce held on...........................

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENTS

We **JEEVAN AKSHAY.B** [REGISTERNO.**211419104113**],**KEVIN CHISTROPHER.A**[REGISTER NO.**211419104140**],**MOHAMMED ABDULLAH** [REGISTER NO. **211419104167**] hereby declare that this project report titled **"EXPEDITING HR MANAGEMENT VIA DYNAMIC E-PORTFOLIO BY EMPLOYING WEB SCRAPPER"**, under the guidance of **KARTHIKEYEN.A** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

JEEVAN AKSHAY

KEVIN CHISTROPHER

MOHAMMED ABDULLAH

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr. P. CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **Tmt. C. VIJAYARAJESWARI**, **Dr. C. SAKTHIKUMAR, M.E., Ph.D** and **Dr. SARANYASREE SAKTHIKUMAR B.E., M.B.A., Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.MANI, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. L. JABASHEELA, M.E., Ph.D.,** for the support extended throughout the project.

We would like to thank **our Project Guide A.KARTHIKEYAN, B.E., M.Tech.,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

JEEVAN AKSHAY B

KEVIN CHRISTOPHER A

MOHAMMED ABDULLAH

ABSTRACT

We have come to a decade where there is a competitive industry. To stand out of the crowd candidates implement interesting projects, attend various symposiums, participate in national and international conferences, publish research papers, and build social media profiles to get in touch with people with similar interests to broaden their scope of getting recruited. A portfolio is one of the many ways to express the candidate's projects and internships. The portfolio compiles materials that exemplify your beliefs, skills, qualifications, education, training, and experiences. It provides insight into your personality and work ethic. In terms of art, a portfolio is a thoughtfully designed visual selection and presentation of art and design. It embodies your ideas, research, innovation, skills, and work process. It includes items such as certificates, transcripts, samples of past work, and letters of recommendation. If this is made online, then the candidate's profile will be more accessible. The authenticity of the work and the candidate's motivation and enthusiasm towards a domain can be easily determined and a candidate's ability to work on a project can be easily established, since a detailed description of the projects and the issues faced during the process and how the hurdles were resolved canbe elaborately detailed out in the portfolio. Thus the candidate with an online portfolio will have more chances of getting recruited by the recruiter.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	Application Programming Interface
IDE	Integrated Development Environment
UI	User Interface
SDK	Software Development Kit
HTTP	Hypertext Transfer Protocol
DB	Database
RAM	Random Access Memory
SSD	Solid-State Drive
GB	Giga Byte
MBPS	Megabytes per second
SSH	Secure Socket Shell
LTS	Long Term Service

CHAPTER 1
INTRODUCTION

CHAPTER- 1

1. INTRODUCTION

1.1 OVERVIEW

Be it promotion or recruitment, HR management is all about giving the right people the right role. Resume printed on paper can serve as an excellent medium to communicate about oneself to the job recruiter. These days jobs with quality income require the candidates appearing for the interview to apply online. However, portfolio hosted on web can aid a student to express projects and achievements. The samples of projects can be displayed through the website.

To make portfolio more robust and dynamic with real-time content updates, this paper introduces the concept of employing a web scrapper to fetch project details from websites like GitHub and certificates from certificate providers such as Coursera.

Keywords: Web Scrapper, HR Recruitment, Dynamic Portfolio, Data Extraction

1.2 PROBLEM DEFINITION

A resume printed on paper is a good means for communicating about oneself to the job recruiter. These days, jobs with quality income require the candidates appearing for the interview to apply online. Portfolio hosted on web is a good means to express about the projects which had been made by the candidate.

The samples of projects can be displayed via the website. Generally, a resume is meant to be a summary of the candidate's profile. On the other hand, portfolio is meant to be more descriptive about the candidate. The achievements of candidates might contain some extra information about the learnings and how the issues were resolved.

1.3 PROPOSED SYSTEM

To overcome the drawbacks of resumes, portfolios are one of the solutions. Any of the candidate's social media profile can be linked to the portfolio and might act as a good description of candidate’s attribute and character.

The approach followed to achieve any tasks can be summarized to give authentically to the achievements. The hurdles which occurred during the projects can be described. The internships which were performed can be outlined with respective progress and timeline. Thus, this world is to enhance the probability of the candidate to be recruited by the HR Manager.

1.3.1 WEB SCRAPPING

The process of scraping data from the Internet can be divided into two sequential steps; acquiring web resources and then extracting desired information from the acquired data. Specifically, a web scraping program starts by composing a HTTP request to acquire resources from a targeted website. This request can be formatted in either a URL containing a GET query or a piece of HTTP message containing a POST query. Once the request is successfully received and processed by the targeted website, the requested resource will be retrieved from the website and then sent back to the give web scraping program. The resource can be in multiple formats, such as web pages that are built from HTML, data feeds in XML or JSON format, or multimedia data such as images, audio, or video files. After the web data is downloaded, the extraction process continues to parse, reformat, and organize the data in a structured way.

1.4 EXISTING SYSTEM

1.4.1 RESUME

The job recruiters generally refer to the resume for getting summarized view of candidate’s profile. This is generally shown in the printout by the candidate to the recruiter. Thus, it has the limitation of just summarizing the information.

1.4.2 LINKEDIN

Other possible alternatives include online social media sites LinkedIn. The LinkedIn app is meant to connect people with similar interests and jobs. Thus, it is not specialized for displaying the projects and experience in internships.

1.4.3 COLLEGE DATABASES

There are colleges with their own databases. Again, the same issue is encountered regarding the information details that only specific and generalized information can be displayed. The database also needs to be constantly maintained by the college management, which is a hectic task.

1.4.4 ONLINE CREDENTIAL SITES

These websites only give a way to display certificates and credentials. They are verified by the company hosting the site using various strategies. On the other hand, the progress and other hurdles faced during projects and internships cannot be displayed here.

1.4.5 JOB SITES

The present jobsites such as indeed.com and naukri.com are purely meant to get the job and thus perform the function of intermediating between the job securities and the candidates applying for jobs.

1.5 ADVANTAGES AND DISADVANTAGES

The advantages of an online portfolio application which we propose is that it is Open Source, so anyone can modify the project for their use and contribute to the project. It is free to use by any student, so a student might not have to pay money for the unnecessary charges. The documentation provided as to how to use the website and generate portfolio.

The disadvantages of the online portfolio application are that the network connectivity issues might stand as a barrier. Also hosting charges might apply based on the number of visits. Exceeded or due to a greater number of read access.

CHAPTER 2
LITERATURE REVIEW

CHAPTER - 2

2. LITERATURE REVIEW

The purpose of literature review is to gain an understanding of the existing research and debates relevant to a particular topic or area of study, and to present that knowledge in the form of a written report. Conducting a literature review helps to build knowledge in a particular field.

TITLE: Extracting the Main Content of Web Pages Using the First Impression Area [1]

DESCRIPTION: Extracting the main content from a web page is essential in various applications such as web crawlers and browser reader modes. Existing extraction methods using text-based algorithms and features for English text can be ineffective for non-English web pages. This study proposes a main content extraction method that obtains visual and structural features from the rendered web p age. Our method uses the first impression area (FIA), a part of a web page that users initially view. In this area, websites have applied many techniques that enable users to find the main content easily. Using the non-textual properties in the FIA, our method selects three points with high content area density and expands the area from each point until it meets several structural and visual-based conditions. We evaluated our method, browsers' (Mozilla Firefox and Google Chrome) reader modes, and existing main content extraction methods on multilingual datasets using two measures: Longest Common Subsequences and matched text blocks. The results showed that our method performed better than other methods in both English (up to 46%, matched text blocks F0.5) and non-English (up to 42%, matched text blocks F0.5) web pages

TITLE: A Review of The Literature on Portfolios and Electronic Portfolios by Philippa ButlerBill Anderson, Bill Anderson, Mark Brown, Mary Simpson

DESCRIPTION: A variety of problems and issues arise with the use of portfolios as an assessment exercise in academic settings, some of which are mitigated by the shift to an electronic environment, and some of which are exacerbated. A lack of well-defined guidelines and a clear structure (Smith & Tillema, 2003) and a lack of examples of past portfolios (Darling, 2001), can lead to student confusion and anxiety about the scope, nature and value of the task (Darling, 2001; Wade & Yarbrough, 1996). Finding a balance between student-driven construction that can lead to superficial reflections and limited evidence, and over-prescribed guidelines that can lead to students lacking ownership and therefore resenting their portfolios, is difficult (Zeichner & Wray, 2001). Students need a lot of guidance and support throughout the portfolio process (Smith & Tillema, 2003), which involves a lot of time on the part of tutors or supervisors (Wade & Yarbrough, 1996). As Darling (2001) points out, students often have little academic experience with writing reflective pieces, so that again needs to be nurtured by their supervisors. The ways in which such feedback is given, and how that (sometimes critical) feedback becomes acceptable to students, are also problematic (Smith & Tillema, 2003). There is an inherent conflict between the goals of students and the goals of their supervisors in constructing portfolios. Students "are understandably most concerned about the uses of their portfolios as aids in gaining employment while…educators are most concerned about using portfolios to promote professional development and to make assessments" (Zeichner & Wray, 2001, p. 618). Zeichner and Wray's solution is that different portfolios be used for each different purpose. Concerns are also expressed over the difficulty of assessing portfolios. Smith and Tillema (2003) see a lack of match between assessment criteria and the goals of the program of study, or what competencies students are expected to develop. They also see a tension between the measurement of standards and capturing development and reflection. The danger is that learning, and reflection will get lost in the drive to measure competency. In addition, students in Darling's (2001) study were concerned with the subjectivity of evaluation.

TITLE: Portfolio Assessment by Margery H Davis and Gominda G Ponnamperuma

DESCRIPTION: The portfolio demonstrates the student's progress toward achievement of curriculum outcomes over time. Portfolio assessment is thus an ongoing process (i.e., continuous assessment), charting student progress toward the expected standard in each exit learning outcome. For each candidate to benefit from such ongoing assessment, his or her portfolio supervisor must have regular review sessions to discuss the ratings with the student and monitor the student's progress toward the curriculum learning outcomes. Supervising a student who is building a portfolio is a process akin to master's degree or PhD supervision.

Portfolio assessment necessarily involves subjective judgements that assess trainees in their natural settings in day to-day practice. The assessment material that an individual candidate selects for his or her portfolio may be somewhat different from what another candidate includes, reflecting differences in individual clinical practice. How, then, can portfolio assessment ensure reliability? Excessive standardization in pursuit of reliability may only add an element of artificiality to an authentic form of assessment. The solution likely lies in sampling across the entire range of potential sources of bias and subjectivity. This involves assessing a candidate with multiple assessment tools, in a variety of settings, on many occasions, by several raters.

CHAPTER 3
SYSTEM ANALYSIS

CHAPTER - 3

3. SYSTEM ANALYSIS

3.1 HARDWARE REQUIREMENTS

The personal desktop can contain the following minimum configuration:

- RAM: 8 GB RSM
- Application: Any Modern Browser installed

For having good website experience the images must be downloaded. Thus, the following are the requirements:

- Download Speed: 25 Mbps
- Upload Speed: 5 Mbps.

The server that hosts the website must contain Firebase Realtime database server, with the following configuration.

- SSD: 1 GB SSD Storage
- Database Access: Any Backend as a Service Host
- Account Access: GitHub Account Access
- Security Services: SSH Service

3.2 SOFTWARE REQUIREMENTS

The following are the packages and libraries required to build the application:

- Node.js – version (16.15.0 LTS)
- React-router-dom@6
- Firebase
- Any IDE

Other npm packages included in the application as dependencies are as follows:

- @mui/material
- @emotion/react
- @emotion/styled

3.2.1 REACT

ReactJS is a declarative, efficient, and flexible JavaScript library for building reusable UI components. It is an open-source, component-based front end library responsible only for the view layer of the application. It was created by Jordan Walke, who was a software engineer at Facebook. Today, most of the websites are built using MVC (model view controller) architecture. In MVC architecture, React is the 'V' which stands for view, whereas the architecture is provided by Redux or Flux. A ReactJS application is made up of multiple components, each component responsible for outputting a small, reusable piece of HTML code. The components are the heart of all React applications. These Components can be nested with other components to allow complex applications to be built of simple building blocks. ReactJS uses virtual DOM based mechanism to fill data in HTML DOM. The virtual DOM works fast as it only changes individual DOM elements instead of reloading complete DOM every time.

To create React app, we write React components that correspond to various elements. We organize these components inside higher level components which define the application structure. For example, we take a form that consists of many elements like input fields, labels, or buttons. We can write each element of the form as React components, and then we combine it into a higher-level component, i.e., the form component itself. The form components would specify the structure of the form along with elements inside of it.

3.2.2 FIREBASE

Firebase is a mobile application development platform from Google with powerful features for developing, handling, and enhancing applications. Firebase is a backend platform for building web and mobile applications.

Firebase is fundamentally a collection of tools developers can rely on, creating applications and expanding them based on demand.

Firebase aims to solve three main problems for developers:

- Build an app, fast
- Release and monitor an app with confidence
- Engage users

Developers relying on this platform get access to services that they would have to develop themselves, and it enables them to lay focus on delivering robust application experiences.

Some of the Google Firebase platform's standout features include databases, authentication, push messages, analytics, file storage, and much more.

Since the services are cloud-hosted, developers can smoothly perform on-demand scaling without any hassle. Firebase is currently among the top app development platforms relied upon by developers across the globe.

It is a pretty comprehensive and flexible platform. It allows users to develop the following categories of applications:

- Android
- iOS
- Web

Advantages of using Firebase are:

1. Free to start
2. Development speed
3. End-to-end app development platform
4. Powered by Google
5. Developers can focus on frontend development

6. It’s serverless
7. It offers machine learning capabilities
8. Generates traffic to your apps

3.2.3 PUPPETEER

Puppeteer is used for automation and streamlining of the front-end development and testing respectively. It was introduced by Google. Puppeteer is based on the Node.js library and is open source.

Puppeteer contains APIs to interact and manage Chrome browser in headless mode or Chromium (following the protocols in DevTools). However, it can also be used for non-headless execution on browsers like Chrome/Chromium/Edge/Firefox.

Puppeteer can be used for the automating majority of UI testing, keyboards, mouse movements, and so on. It can be used to test applications developed in Angular and Angularjs. Puppeteer is not considered as an automation tool like Selenium, Cypress, Protractor, and so on. It is mostly used to manage the internal features of the Chromium browser. We can open DevTools in the Chrome browser, by pressing F12 or Command+Option+C(in MacOS). Puppeteer is like a development tool as it can perform most tasks performed by a developer like handling requests and responses, locating elements, network traffic and performance, and so on.

Puppeteer utilises the Node library that gives a top-class API for managing Chromium or Chrome browsers. This is done by following the protocols of DevTools.

Puppeteer has the below hierarchy :

- **Browser(with/without headless mode)** − The browser performs the actions to be executed on the browser engine.
- **Chromium Development Project or CDP** − The Chromium is the real place where all the operations are executed. The browsers - Microsoft Edge and Chrome utilise Chromium as browser engine.

- **Puppeteer** − This is actually a package based on the node module.
- **Automation test code** − This is also known as the Nodejs level. Here, the actual automation code is developed by the end-user using JavaScript.

3.3 NON-FUNCTIONAL REQUIREMENTS

The application will be made maintainable. The application will be divided into components to make it easier to test and build new features into the web application.

To make it secure the website will be hosted in a server with SSL and HTTPS connection. The access rules can be defined in firebase security rules for real-time databases for both read and write operations and it is a known as non- functional requirements.

Other non-functional requirements include load balancing and auto scaling. The client might connect to multiple connections while implementing socket programming with middleware server.

Auto-scaling basically means that the server's load and handling capacity must be increased with the increase in the number of users.

CHAPTER 4
SYSTEM DESIGN

CHAPTER - 4

4. SYSTEM DESIGN

4.1 ARCHITECTURE DIAGRAMS:

4.1.1 SOFTWARE ARCHITECTURE

The following is the system architecture for the portfolio application expressing the dependencies involved in the working of the portfolio application.

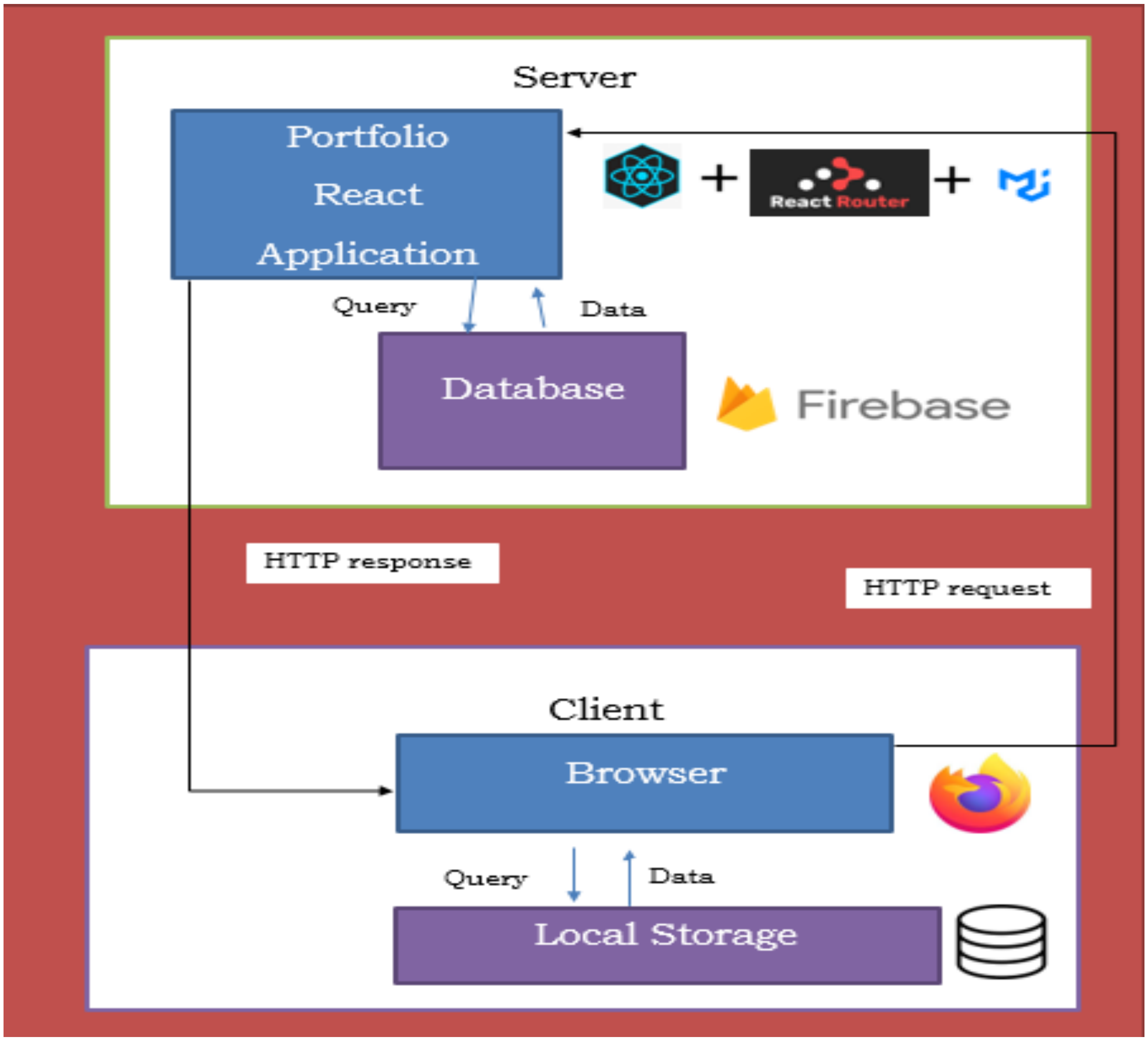


Figure 4.1.1 – Software Architecture

4.1.2 SYSTEM ARCHITECTURE

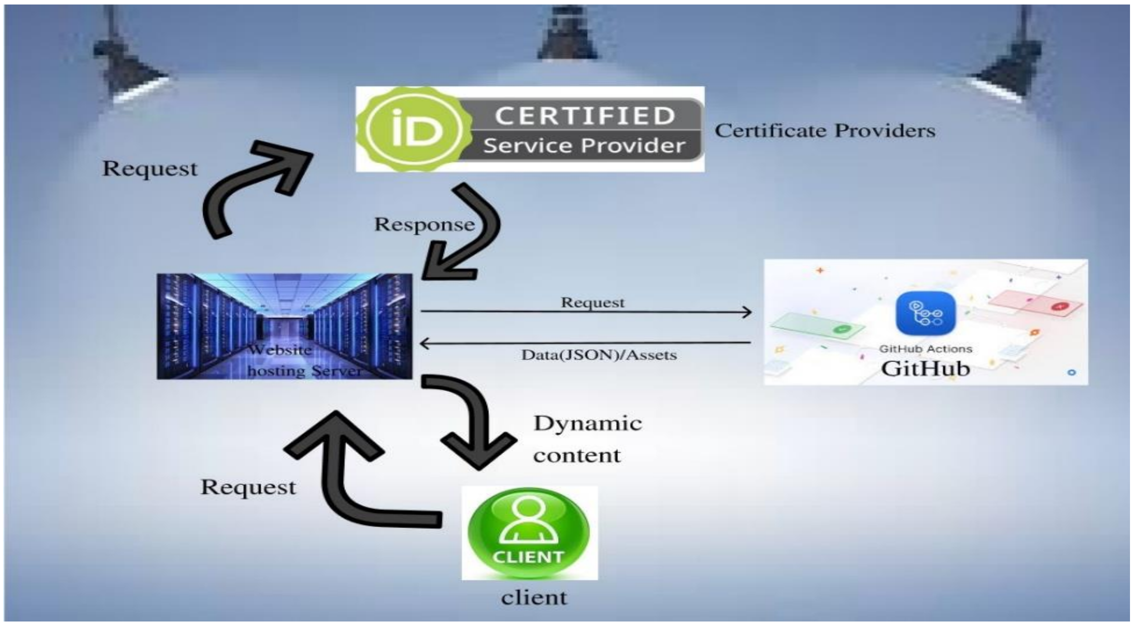


Figure 4.1.2 System Architecture

The external websites which our web scrapper will scan are the certificate providers and websites where one can display their work such as GitHub for developers and dribble for designers.

The web scraper is present on the website, hosted and accessible via internet. When a user views the website, the user's device behaves as a client and the device hosting the website behaves as a server. During the process of web scrapping, where the web scrapper scans the external websites, our website with web scrapper behaves as a client requesting to the external websites, which behave as servers, for structured responsesThe external websites which our web scrapper will scan are the certificate providers and websites where one can display their work such as GitHub for developers and dribble for designers.

4.2 USE CASE DIAGRAM

A use case describes how a user uses a system to accomplish a particular goal. A use case diagram consists of the system, the related use cases and actors and relates these to each other to visualize the system, actors and use case. Use cases help ensure that the correct system is developed by capturing the requirements from the user's point of view.

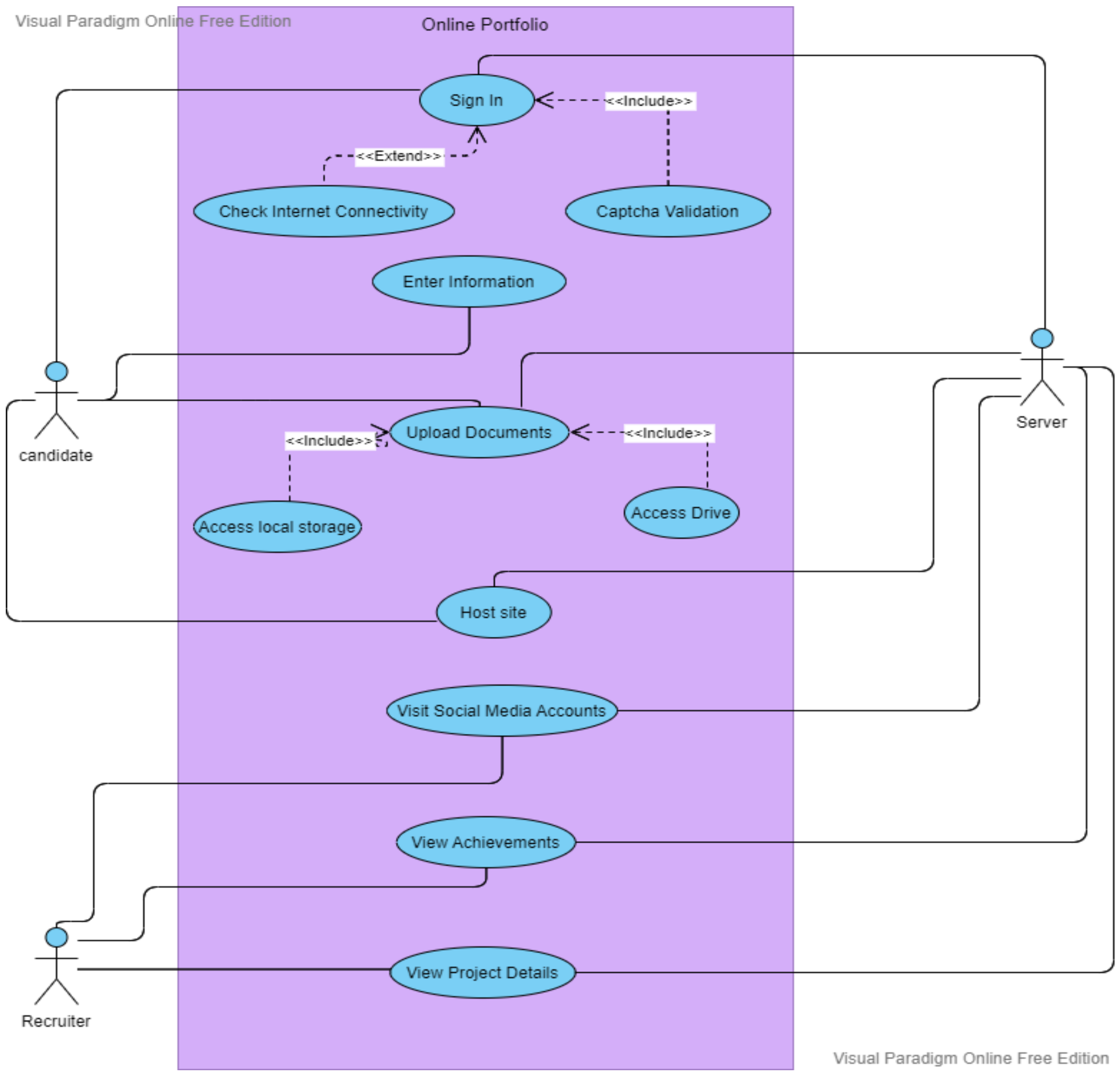


Figure 4.2 – Use Case Diagram

4.3 CLASS DIAGRAM

A class diagram describes the structure of an object-oriented system by showing the classes in that system and the relationships between the classes. A class diagram also shows constraints, and attributes of classes. Class is represented as rectangular box showing class name, attributes, operations. An attribute is a logical data value of an object. Attributes of a classifier are also called structural properties in the UML. The classes involved here are the models used in the Realtime database to store the data. The required functions elucidated in each class are implemented in their own specific JavaScript file to be made as a separate react component. These components are then called and displayed based on the current page the user is visiting.

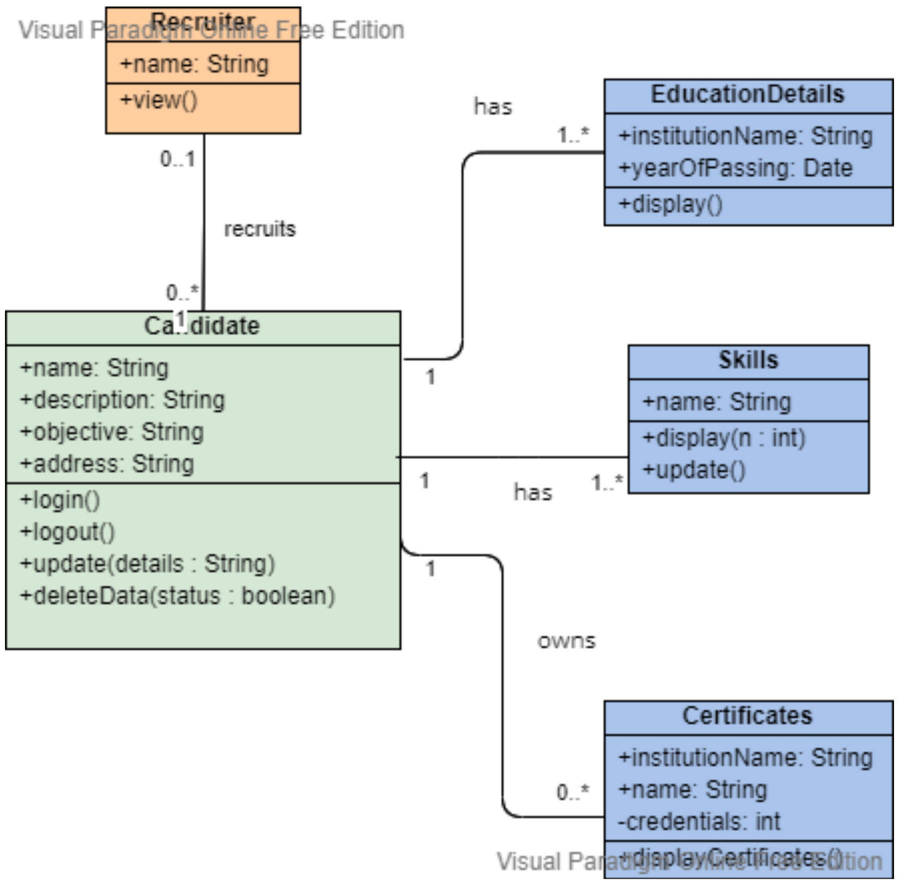


Figure 4.3 – Class Diagram

4.4 SEQUENCE DIAGRAM

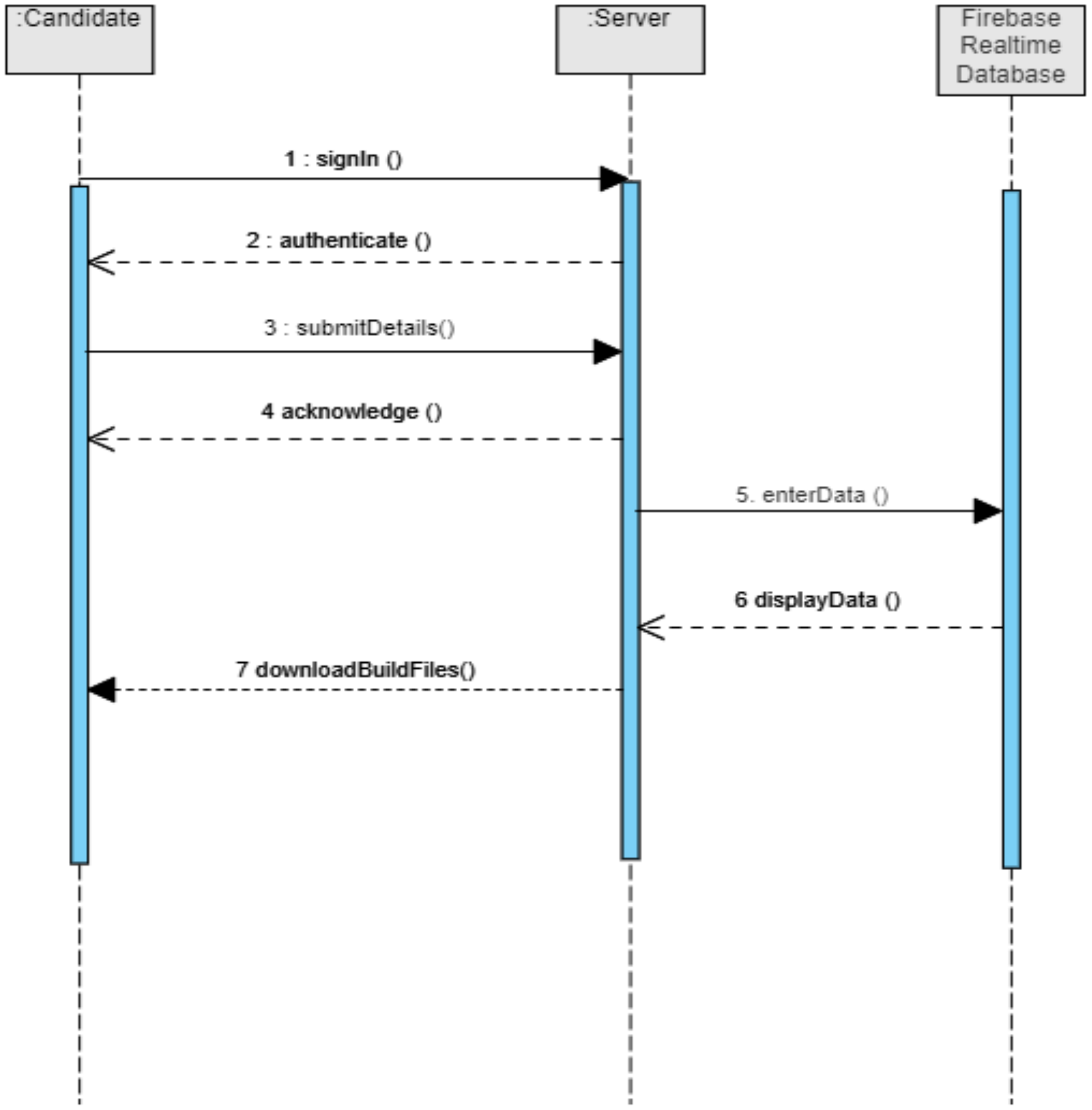


Figure 4.4 – Sequence Diagram

For Authentication, LinkedIn officially provides Authorization Code Flow (3-legged OAuth) using which limited details can be obtained and certain other information with partnership integrations under V2 OAuth.

The following is Authorization Code Flow, i.e., 3-legged OAuth:

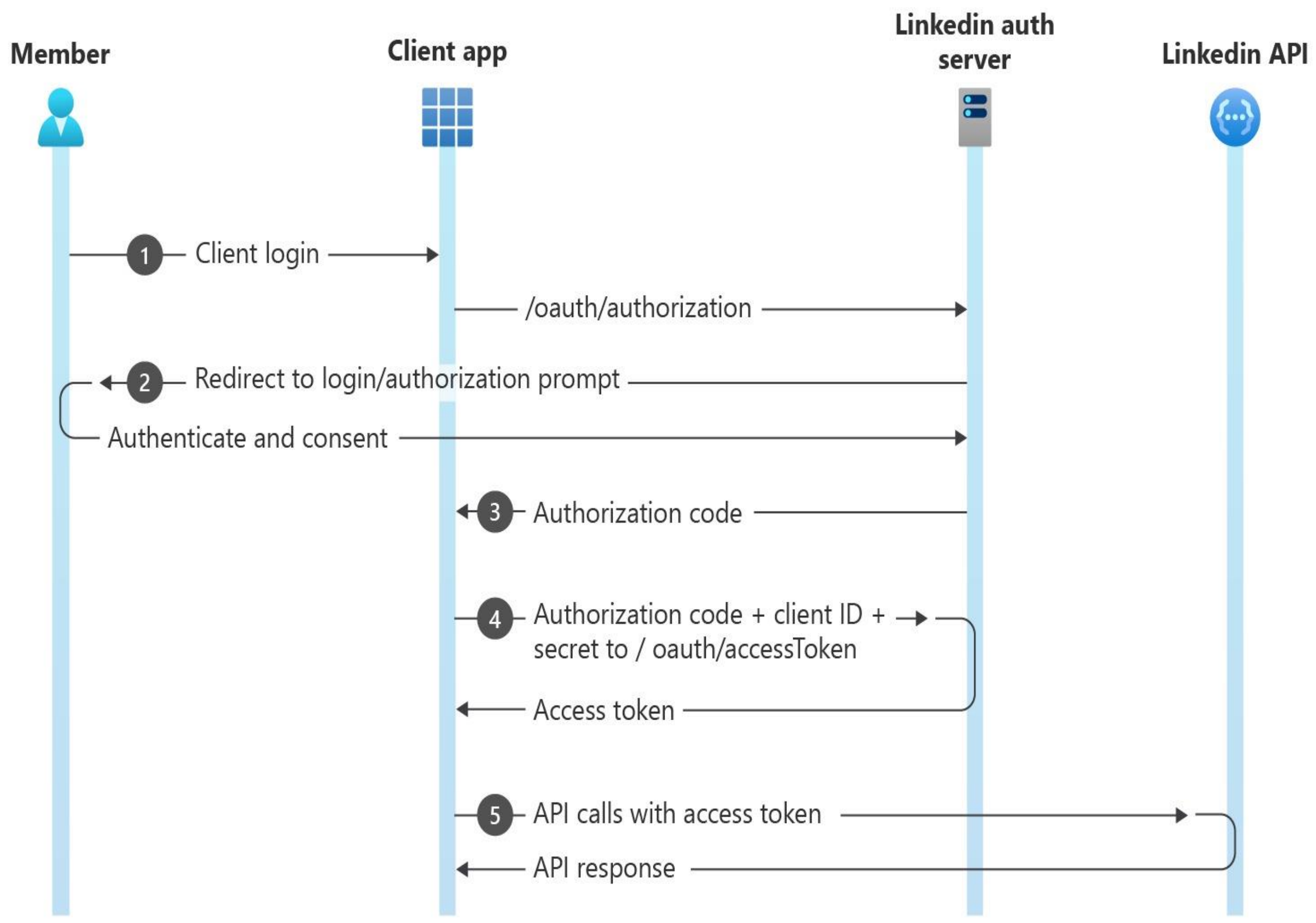


Figure 4.4.1 – Sequence Diagram(i)

To dynamically display GitHub and related data & scrape data from LinkedIn, the following sequence diagram is used, with middleware server handling the requests with the socket programming for transmission of data:

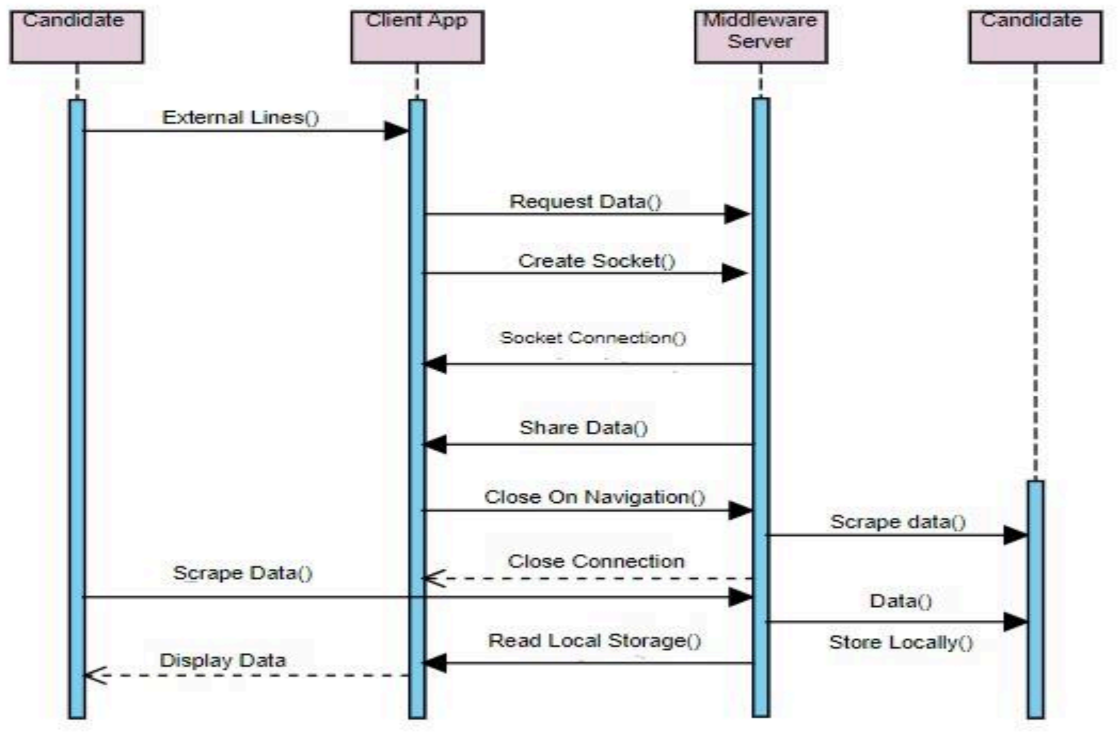


Figure 4.4.2 – Sequence Diagram(ii)

The sequence diagram above elucidates the process in detail with the functioning of middleware server which happens to work with socket programming.

4.5 DEPLOYMENT DIAGRAM

In the UML, deployment diagram is used to visualize the static aspect of these physical nodes and their relationships and to specify their details for construction. Deployment diagrams address the static deployment view of an architecture. The following diagram shows that it has a portfolio application running on a server. This server could be a server with Firebase Realtime database running at its backend. The queries from the browser are fetched from the server which then responds back to the browser. The browser updates the data without refreshing the page.

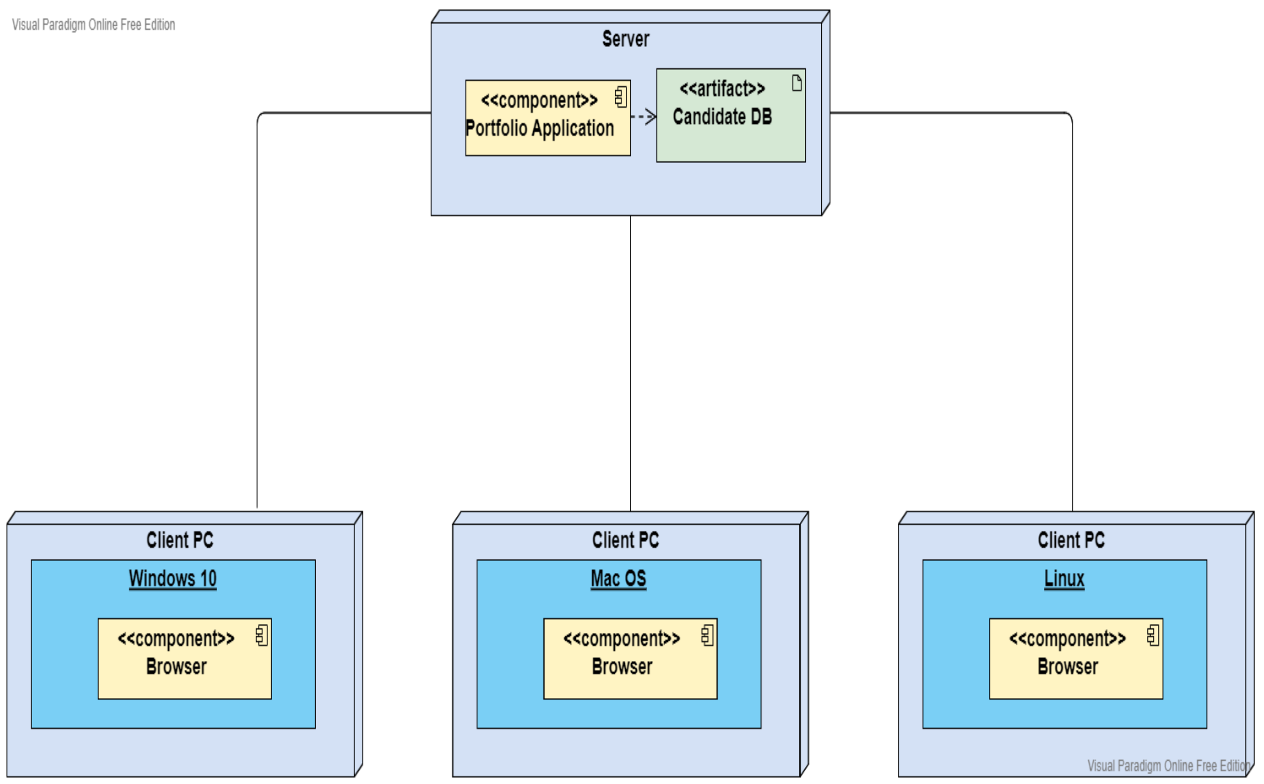


Figure 4.5 – Deployment Diagram

CHAPTER 5
SYSTEM IMPLEMENTATION

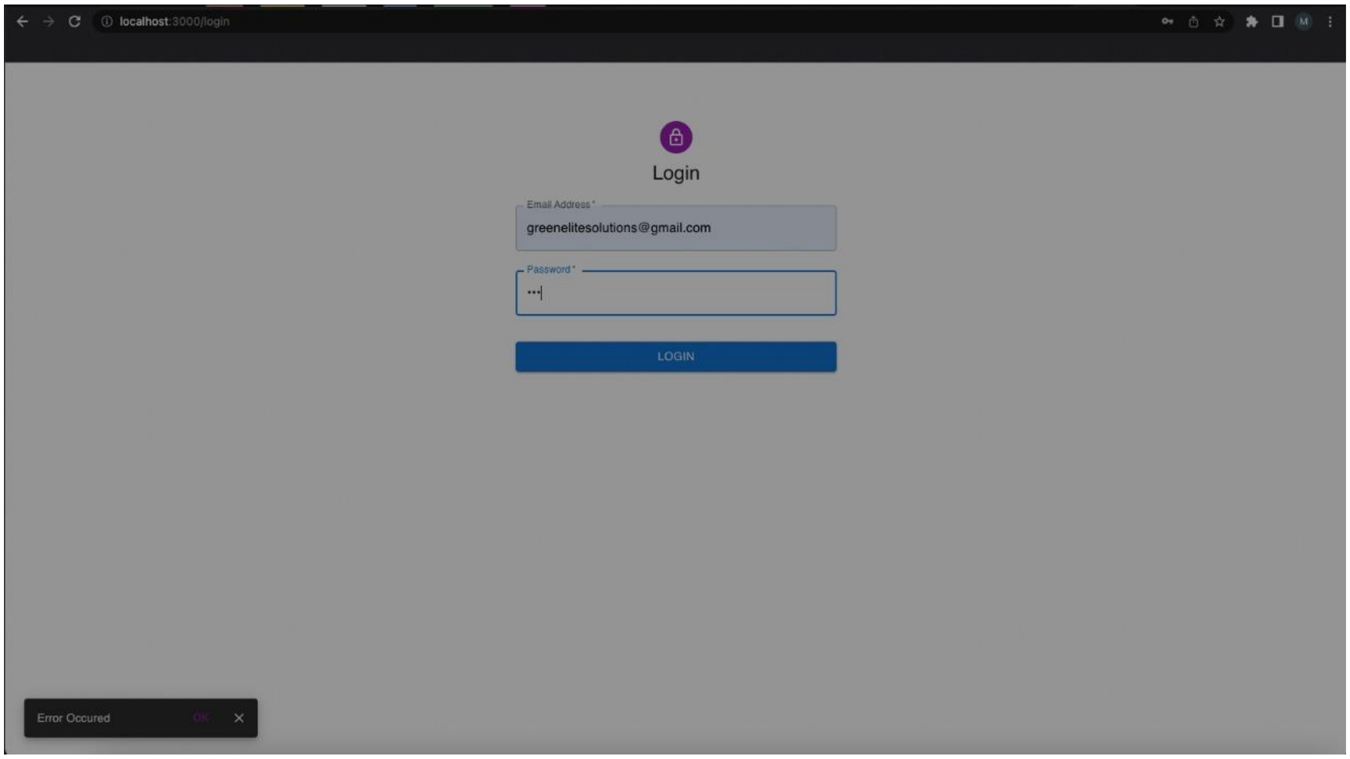
CHAPTER - 5

5. SYSTEM IMPLEMENTATION

This subsection contains the requirements for online portfolio.

5.1 Login Functionality and Authentication

1) The application login is done via firebase API for authentication.
2) The user enters the username and password. If the username or password is invalid, then error is shown via alert message.
3) This is controlled by MVC specifications.
4) Global firebase object is used for further authorization while updating data.



greenelitesolutions@gmail.com

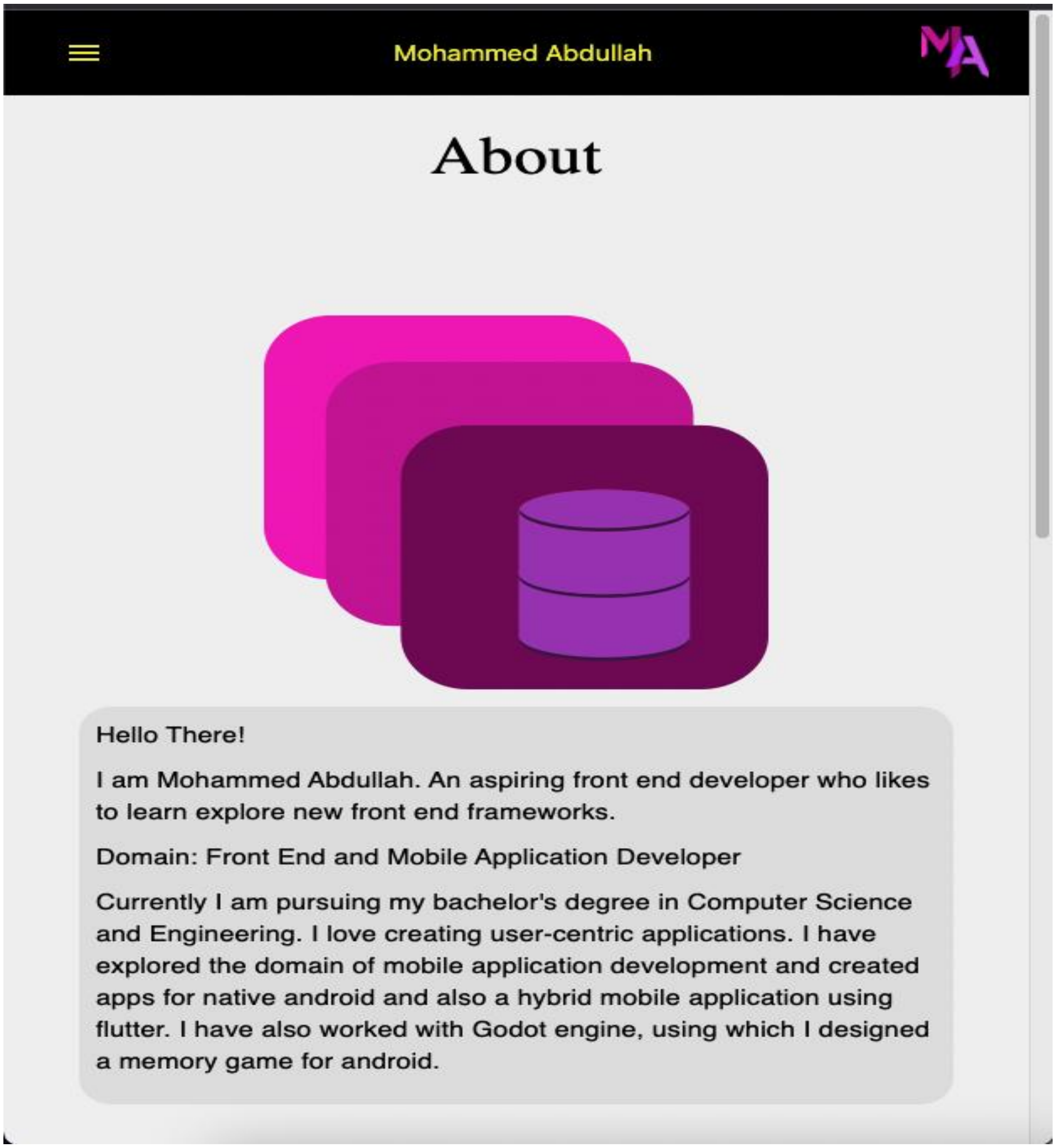
Error Occured

Figure 5.1.1- Login Functionality and Authentication

5.2 PROFILE PAGE OF CANDIDATE

1) The system will display the short term and long-term goals of the candidate.

2) The photo of the candidate will be displayed along with objective of the candidate and the application will be made responsive for mobile use such that the profile photo of the candidate is adjust for mobile view.



About

Hello There!

I am Mohammed Abdullah. An aspiring front end developer who likes to learn explore new front end frameworks.

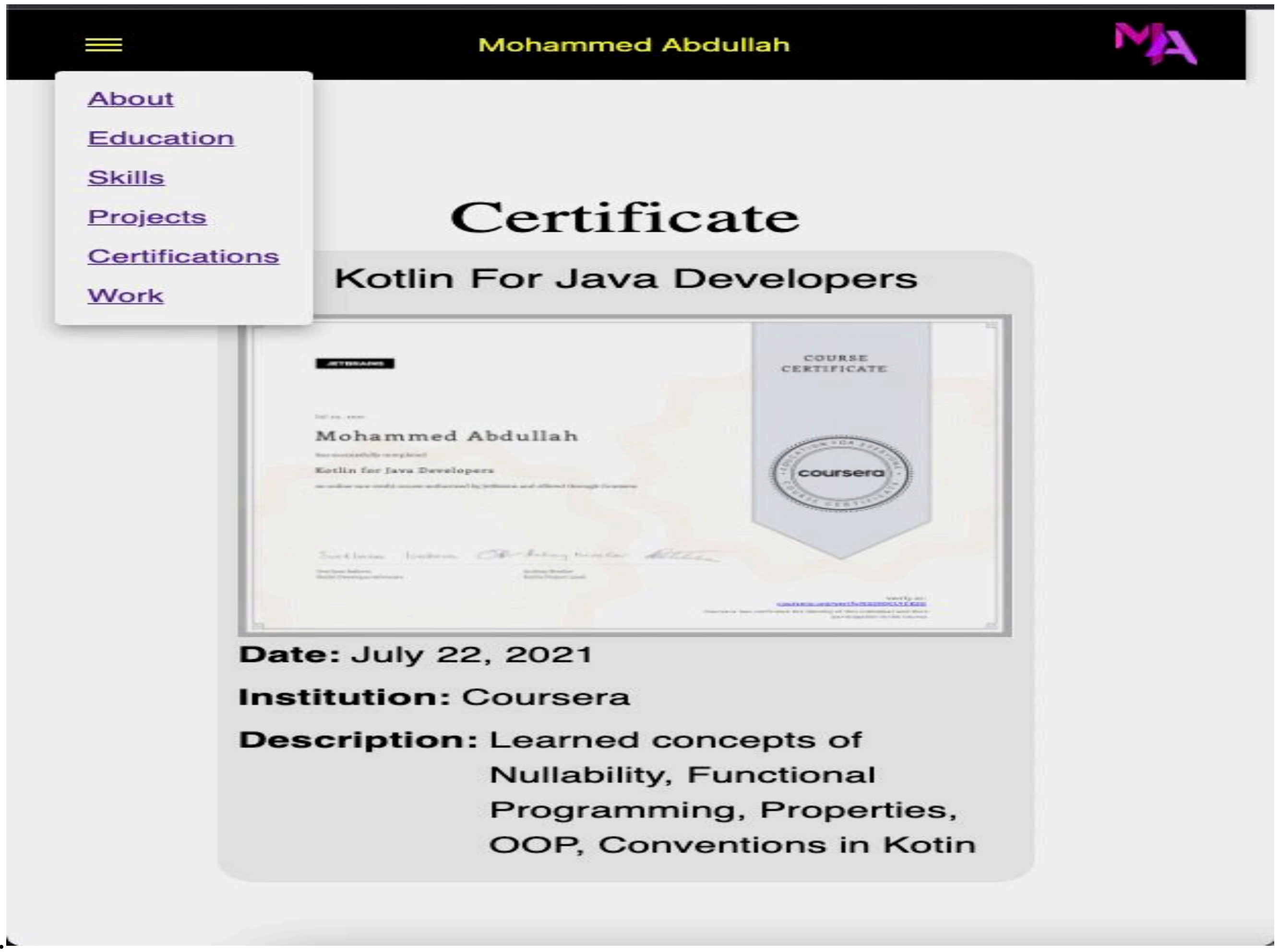
Domain: Front End and Mobile Application Developer

Currently I am pursuing my bachelor's degree in Computer Science and Engineering. I love creating user-centric applications. I have explored the domain of mobile application development and created apps for native android and also a hybrid mobile application using flutter. I have also worked with Godot engine, using which I designed a memory game for android.

Figure 5.2.– Profile Page of Candidate

5.3 RESPONSIVE NAVIGATION AND ROUTING

1) The application will contain links in the navigation bar to display a particular component at a time.
2) The navigation items will be made to another view when the application is viewed in the mobile.
3) The browser history will be saved in the router state to navigate forward, and backward and Nested routing will be performed on certifications, projects and work view each separately with detailed information



Certificate

Kotlin For Java Developers

Mohammed Abdullah

Date: July 22, 2021

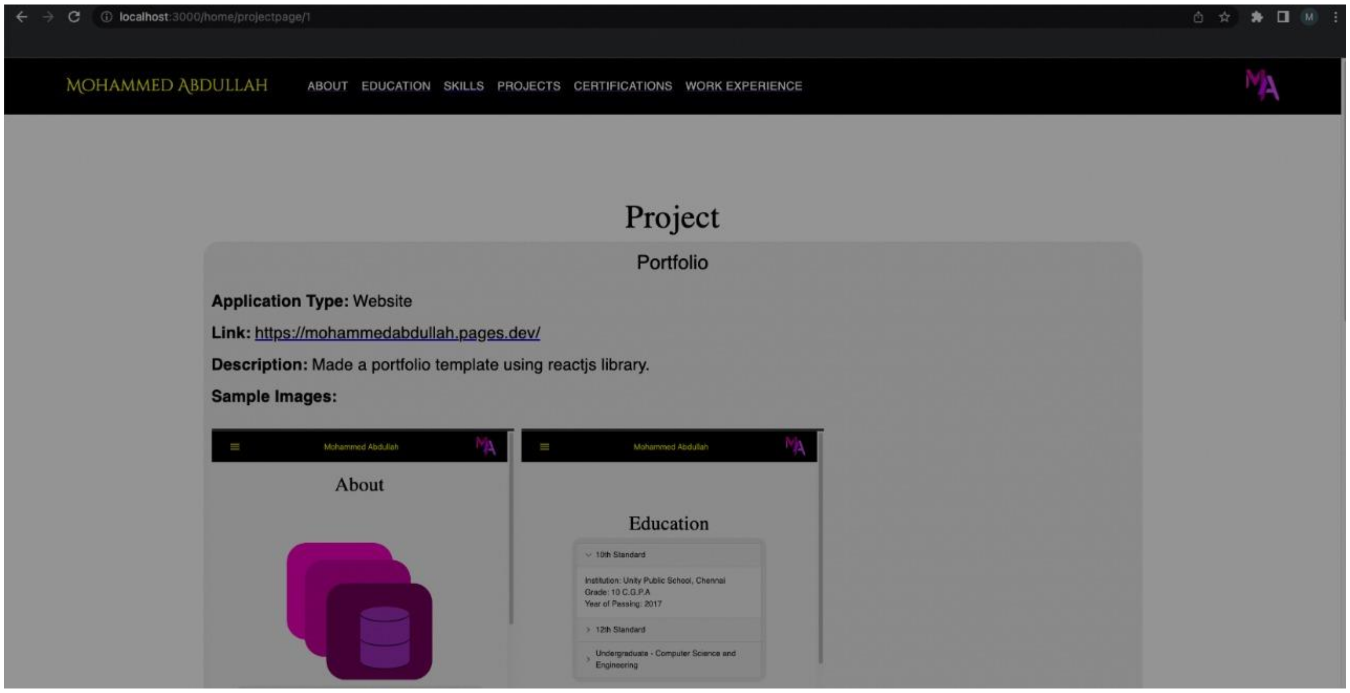
Institution: Coursera

Description: Learned concepts of Nullability, Functional Programming, Properties, OOP, Conventions in Kotin

Figure 5.3 – Responsive Navigation and Routing

5.4 PROJECT INFORMATION PAGE

1) The application will display the candidate's contact information with email and phone number.
2) The candidate’s online media profiles and links to them will by the web scrapper and displayed.
3) This component will be made available on each navigation page to make it accessible.
4) Socket programming is used to send the data from the middleware server to the client.
5) Two connections are made for to visit the links and display the content.



Project

Portfolio

Application Type: Website

Link: https://mohammedabdullah.pages.dev/

Description: Made a portfolio template using reactjs library.

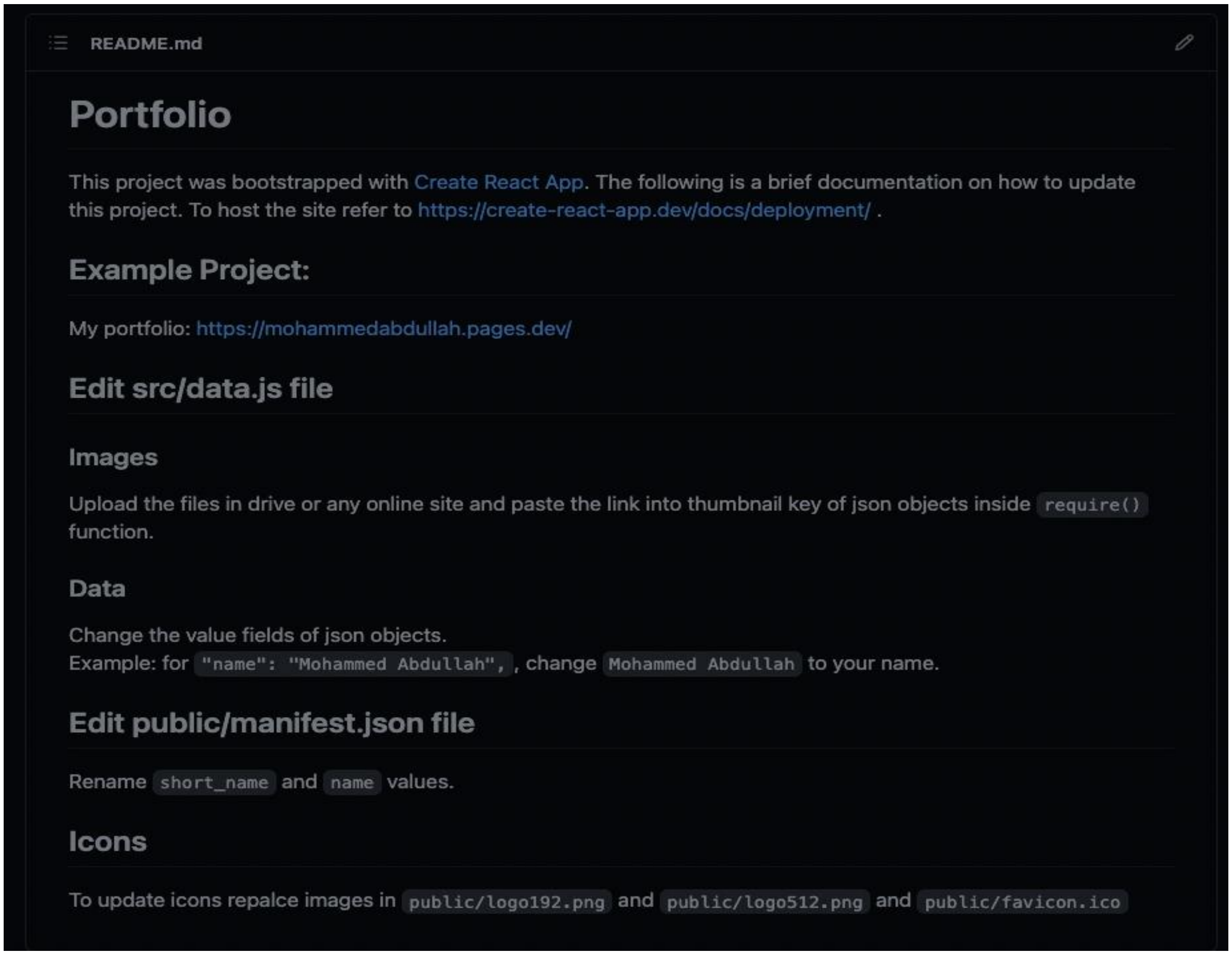
Sample Images:

10th Standard
Institution: Unity Public School, Chennai
Grade: 10 C.G.P.A
Year of Passing: 2017
12th Standard
Undergraduate - Computer Science and Engineering

Figure 5.4. – Project Information Page

5.5 USER DOCUMENTATION AND HOSTING INFORMATION

1) The actions required to use the portfolio template will be mentioned.
2) The steps on how to host the website will be linked to the official website.
3) The process of uploading the images of certificates and icons will be detailed out.
4) The server information in manifest file required to host the website will be documented on how to use it.



README.md

Portfolio

This project was bootstrapped with Create React App. The following is a brief documentation on how to update this project. To host the site refer to https://create-react-app.dev/docs/deployment/ .

Example Project:

My portfolio: https://mohammedabdullah.pages.dev/

Edit src/data.js file

Images

Upload the files in drive or any online site and paste the link into thumbnail key of json objects inside `require()` function.

Data

Change the value fields of json objects.
Example: for `"name": "Mohammed Abdullah",` , change `Mohammed Abdullah` to your name.

Edit public/manifest.json file

Rename `short_name` and `name` values.

Icons

To update icons repalce images in `public/logo192.png` and `public/logo512.png` and `public/favicon.ico`

Figure 5.5.1 – User Documentation and Hosting Information

5.6 USER INPUT IN FIREBASE CONSOLE

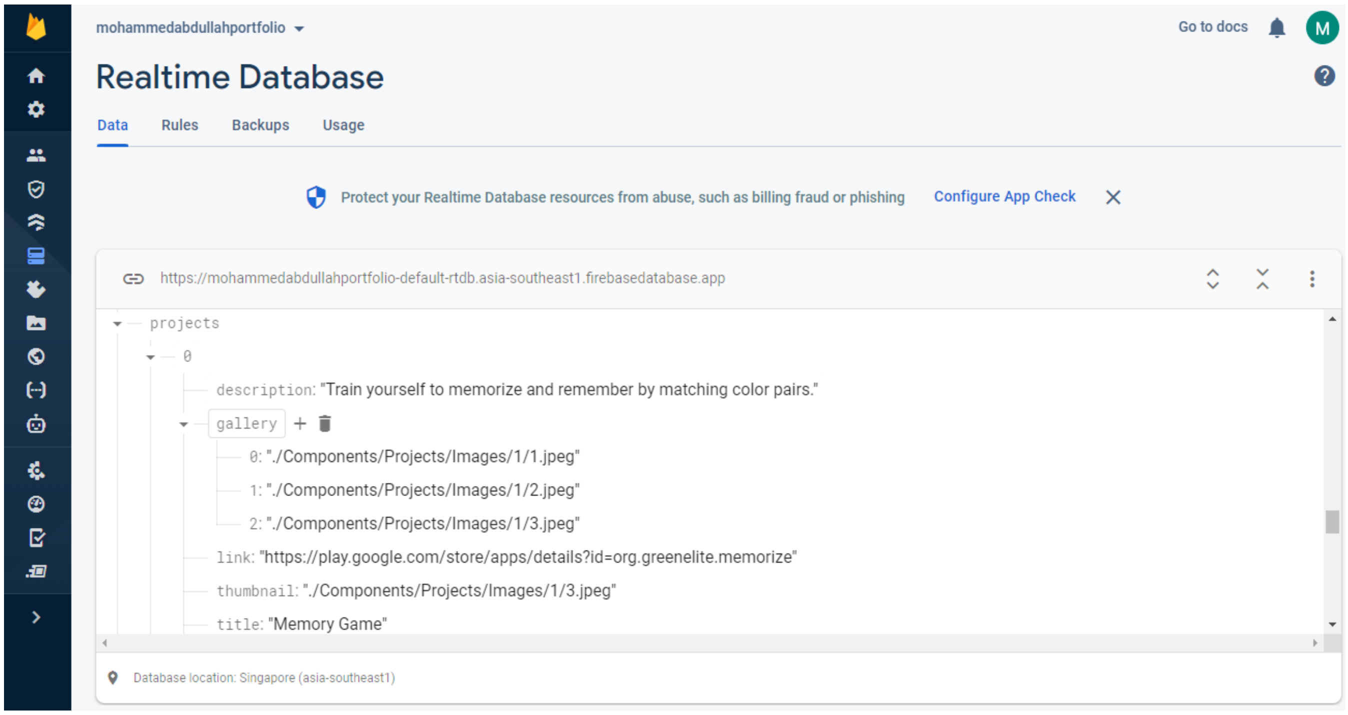
1) The user will be required to either edit the data file or fill the form which updates data in the real-time database.

2) The code of the application shall be cloned from GitHub URL mentioned in the documentation and the data file will be found in the root folder.

3) The files and images shall be uploaded and links to those files must be filled in the data file.

4) There are security rules by which the data gets uploaded only from an authenticated user.

5) This can be controlled in the firebase console.


mohammedabdullahportfolio
Go to docs
M
Realtime Database
Data
Rules
Backups
Usage
Protect your Realtime Database resources from abuse, such as billing fraud or phishing
Configure App Check
Database location: Singapore (asia-southeast1)

https://mohammedabdullahportfolio-default-rtdb.asia-southeast1.firebasedatabase.app

```
projects
  0
    description: "Train yourself to memorize and remember by matching color pairs."
    gallery
      0: "./Components/Projects/Images/1/1.jpeg"
      1: "./Components/Projects/Images/1/2.jpeg"
      2: "./Components/Projects/Images/1/3.jpeg"
    link: "https://play.google.com/store/apps/details?id=org.greenelite.memorize"
    thumbnail: "./Components/Projects/Images/1/3.jpeg"
    title: "Memory Game"
```

Figure 5.6.1 – User Input in Firebase Console

5.7 MIDDLEWARE SERVER

1) The projects page from the client side utilizes socket programming to pass over the entered links to the middleware server.

2) Then, the middleware server visits the website and fetches the data.

3) This is done with puppeteer. Screenshots are taken and are transferred to the client in socket connection.
4) The server is started using the pm2 server which is a production process manager for Node.js applications.
5) This helps in restarting the server when the server is down.

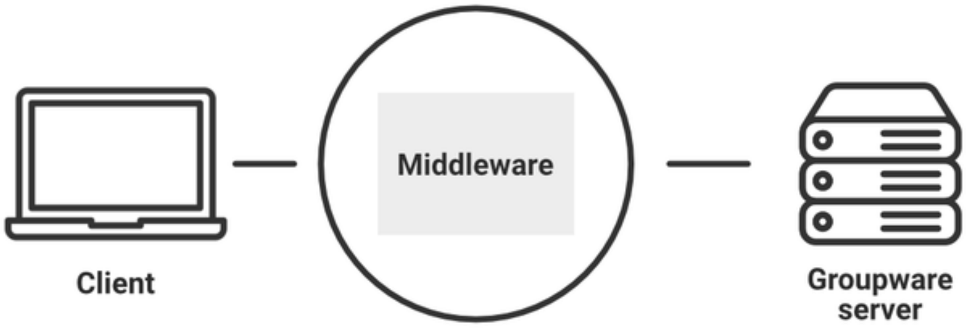


Figure 5.7 Middleware Server

5.8 WEB SCRAPPER SERVER

1) This server utilizes puppeteer to launch the chrome in headless mode.
2) This helps to overcome the CORS policy errors.
3) The client sends the LinkedIn URL to this server using ExpressJS REST API.
4) The server launches the web scrapper to scrape data.
5) The document query selectors are used to parse the HTML content received from the request.
6) This is then written to the files after creating a separate folder for each unique user.
7) When the client requests the data, the above written files are read and transferred back to the client.

CHAPTER 6
SYSTEM TESTING

CHAPTER – 6

6. TESTING

To keep the system error free during the phases of development and during the time when new features are added, the following testing strategies are applied.

Software testing is a very essential and important activity that should be carried out during and after the development of the software. Software that has been developed may have several bugs (errors). The process of testing the software will reveal the errors that are invisible to the developer during the development of the system. Software testing is the activity carried out by the testers. The testers will test the software using various mechanisms with the intention of finding the errors. The main goal of the software testing is to identify the errors that are possible.

Developers may not be aware of the various categories of inputs and circumstances that cause the error to the system. It is the role of the tester to identify the conditions that the system will encounter the error. The software engineering process can be viewed as a spiral model. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and coding. To increase the quality of code, software is tested with variable testing strategies suitable for that software.

Each test investigates each functional and nonfunctional characteristics of the software in detail.

6.1 OBJECTIVES

The major objectives of Software testing are as follows:

- Finding defects which may get created by the programmer while developing the software.

- Gaining confidence in and providing information about the level of quality.
- To prevent defects.
- To make sure that the end result meets the business and user requirements.

6.2 TEST PROCESS

We can divide the activities within the fundamental test process into the following basic steps:

- Planning and Control
- Analysis and Design
- Implementation and Execution
- Evaluating exit criteria and Reporting
- Test Closure activities

6.3 BLACK BOX TESTING

Black box testing is the testing approach in which the software efficiency is checked just by giving the input and checking whether the expected output comes or not. It ensures whether the system is capable of yielding expected output or not. The system under testing process is said to be passed if the expected output is equivalent to the arrived output.

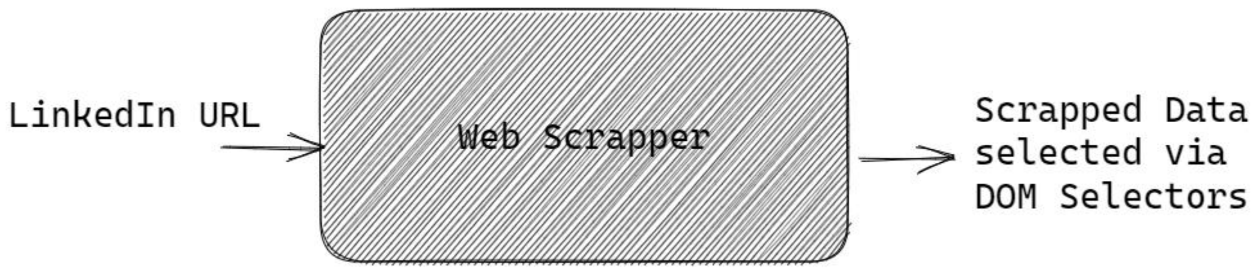


Figure 6.1 Black Box Testing

6.4 EQUIVALENCE CLASS PARTITIONING

Equivalence class partitioning is the test case generation strategy in which the input domains are classified into various classes. The testing procedure involves analyzing the efficiency of the system just by applying one input from each equivalence class input. ECP consists of both valid as well as invalid input classes. ECP reduces the number of test cases.

6.5 SYSTEM TESTING:

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested. System Testing is carried out on the whole system in the context of either **system requirement specifications** or **functional requirement specifications** or in the context of both. System testing tests the design and behavior of the system and the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements specification (SRS). System Testing is basically performed by a testing team that is independent of the development team that helps to test the quality of the system impartial. It has both functional and non-functional testing. System Testing is a black-box testing. System Testing is performed after the integration testing and before the acceptance testing.

System Testing is performed in the following steps:

- **Test Environment Setup:** Create testing environment for the better quality testing.
- **Create Test Case:** Generate test case for the testing process.

- **Create Test Data:** Generate the data that is to be tested.
- **Execute Test Case:** After the generation of the test case and the test data, test cases are executed.
- **Defect Reporting:** Defects in the system are detected.
- **Regression Testing:** It is carried out to test the side effects of the testing process.
- **Log Defects:** Defects are fixed in this step.
- **Retest:** If the test is not successful then again test is performed.

6.6 UNIT TESTING:

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements.

The following is the unit testing for each of the modules:

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITON	TEST STEPS	TEST DATA	EXPECTED RESULT	POST CONDITION	ACTUAL OUTCOME	STATUS (PASS/ FAIL)
1.	Verify username detail	Enter Credentials	Should have a valid portfolio account	1.Enter username 2.Click submit	<Valid login username>	Verify login access with correct credentials	Navigate to form data page	Logged in page is shown	Pass
2	Verify information	Enter invalid credentials	Should have a valid portfolio account	1.Enter Username 2.Enter password 3. Click submit	<Invalid credentials	Error message	Stay in same page	Error message	pass

3.	**Verify password detail**	**Enter Password**	**Should have a valid portfolio account**	**1. Enter Password 2. Wait for 200ms for auto change function to complete**	**<Valid password>**	**Verify whether it is minimum of 6 characters**	**Show data page**	**Error message is hidden**	**Pass**
4	**Click the submit button for the form**	**Click submit Button**	**Should have filled both username and password**	**1.Enter required fields. 2.Enter submit**	**<Valid button field>**	**Check whether button is enabled**	**Show login page**	**Button is enabled**	**Pass**

Table 6.1.1 – Unit Testing

CHAPTER- 7
CONCLUSION AND FUTURE SCOPE

CHAPTER - 7

CONCLUSION

The goal of the project to make portfolio available easily online was implemented by having a good review of the technologies. The implemented project thus helps the candidate to have a portfolio online within minutes. Any person without the knowledge of the domain can work with the project due to the presence of documentation. The application thus makes use of the technology of backend as a service efficiently, which provides a way to scale the project with any new data. Authentication of the website ensures that no unauthorized users access the write access grant of the data. The data filled is transferred based on the application programming interface provided by the firebase software development kit. The application secures both the data and the transmission of data, so it is not updated without the required updating privileges. Thus, the application produces the portfolio of the candidate to be displayed on a website.

FUTURE SCOPE

The project can be expanded to the following features such as when signing in two factor authentication can be performed using additional one-time password being sent via email or mobile number. Similarly, captcha validation can be introduced before accessing the portfolio template to reduce the load on server.

The template can be made to pure html, CSS and JavaScript modules and given to the user. This might be helpful when hosting services require loss server storage. The user data can be stored in the server after the user has logged into an account. Themes can be added for the user. The security of the web application can be made to increase by testing a greater number of modules.

CHAPTER- 8
APPENDICES

CHAPTER - 8

A.1 CODING

INDEX.JS

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import {
    BrowserRouter as Router
} from "react-router-dom";

ReactDOM.render(
    <React.StrictMode>
        <Router>
            <App />
        </Router>
    </React.StrictMode>,
    document.getElementById('root')
);
```

MIDDLEWARESERVER: INDEX.JS

```
const express = require("express");
const app = express();
const cors = require("cors");
const http = require("http").Server(app);
const http2 = require("http").Server(app);
const puppeteer = require("puppeteer");
const PuppeteerMassScreenshots = require("./screen.shooter");
const PORT = 4000;
const PORT2 = 4001;
const socketIO = require("socket.io")(http, {
  cors: {
    origin: "http://localhost:3000",
  },
});
const socketIO2 = require("socket.io")(http2, {
  cors: {
    origin: "http://localhost:3000",
  },
});


app.use(cors());
app.use(express.urlencoded({ extended: true }));
app.use(express.json());
```

```
let channelList = [];
socketIO.on("connection", (socket) => {
    console.log(` ${socket.id} user just connected!`);
    socket.on('browse', async ({ url }) => {
        if (url !== '') {
            console.log("url:::", url);
            if (url == undefined) return;
            const browser = await puppeteer.launch({
               executablePath: "/Applications/Google Chrome.app/Contents/MacOS/Google
Chrome",
                headless: true
            });
            const context = await browser.createIncognitoBrowserContext();
            const page = await context.newPage();
            await page.setViewport({
                width: 1255,
                height: 800,
            });
            console.log("url: :", url)
            await page.goto(url);
            const screenshots = new PuppeteerMassScreenshots();
            await screenshots.init(page, socket);
            await screenshots.start();
            socket.on('mouseMove', async ({ x, y }) => {
                try {
                    await page.mouse.move(x, y);
                    const cur = await page.evaluate((p) => {
                        const elementFromPoint = document.elementFromPoint(p.x, p.y);
                        return window
                            .getComputedStyle(elementFromPoint, null)
                            .getPropertyValue('cursor');
                    }, { x, y });
                    socket.emit('cursor', cur);
                } catch (err) { }
            });
            socket.on('mouseClick', async ({ x, y }) => {
                try {
                    await page.mouse.click(x, y);
                } catch (err) { }
            });
            socket.on('scroll', ({ position }) => {
                page.evaluate((top) => {
                    window.scrollTo({ top });
                }, position);
            });
        }
```

```
    });
    socket.on("disconnect", () => {
        socket.disconnect();
        console.log("A user disconnected");
    });
});
socketIO2.on("connection", (socket) => {
    console.log(` ${socket.id} 2user just connected!`);
    socket.on('browse', async ({ url }) => {
        if (url !== '') {
            console.log("url2:::", url);
            if (url == undefined) return;
            const browser = await puppeteer.launch({
               executablePath: "/Applications/Google Chrome.app/Contents/MacOS/Google
Chrome",
                headless: true
            });
            const context = await browser.createIncognitoBrowserContext();
            const page = await context.newPage();
            await page.setViewport({
                width: 1255,
                height: 800,
            });
```

```
await page.goto(url);
const screenshots = new PuppeteerMassScreenshots();
await screenshots.init(page, socket);
await screenshots.start();
socket.on('mouseMove', async ({ x, y }) => {
  try {
    await page.mouse.move(x, y);
    const cur = await page.evaluate((p) => {
      const elementFromPoint = document.elementFromPoint(p.x, p.y);
      return window
        .getComputedStyle(elementFromPoint, null)
        .getPropertyValue('cursor');
    }, { x, y });
    socket.emit('cursor', cur);
  } catch (err) { }
});
socket.on('mouseClick', async ({ x, y }) => {
  try {
    await page.mouse.click(x, y);
  } catch (err) { }
});
socket.on('scroll', ({ position }) => {
  page.evaluate((top) => {
```

```
                    window.scrollTo({ top });
                }, position);
            });
        }
    });
    socket.on("disconnect", () => {
        socket.disconnect();
        console.log(" 2 user disconnected");
    });
});
http.listen(PORT, () => {
    console.log(`Server listening on ${PORT}`);
});
http2.listen(PORT2, () => {
    console.log(`Server2 listening on ${PORT2}`);
});
```

SERVER: INDEX.JS

```
const express = require('express');
const cors = require('cors');
const fetch = require('node-fetch');
const isHeadlessMode = false;
const PORT = 8080;
const app = express();
app.use(bodyParser.urlencoded({ extended: true }))
app.use(cors());
const corsOptions = {
   origin: "http://localhost:3000"
};
let code = "
AQTLTKeLtaja_Xq3B9UjRkZdE862AOKyWOF7ijzKfwxWi5HyQreF82WC4VoS0Qy-
JPGJz09BVm-qMXcXN-
WmxDxOMnQ0qM_NN9gOXMdk8k5tsjTIN7H828ADW8gz8PMcdSLXeeLy45Ke8xJ
xcVVlON-
ejUpXNBrtWNbAz4mrbmyy2rlGUlSkdN_r5DimUyAaPk21sOd7g6f_nzGsEn8";
app.get('/auth/:code', async (req, res) => {
   code = req.params.code;
   console.log("code :", code);
   const client_id = "863yjvgqsfyyvq";
   const client_secret = "dVfyZFbQ3zPMs4B5";
   const callback_url = "http://localhost:3000/form/callback";
   const url =
`https://www.linkedin.com/oauth/v2/accessToken?code=${code}&grant_type=authorizat
ion_code&client_id=${client_id}&client_secret=${client_secret}&redirect_uri=${callba
ck_url}`;
   const requestEndpoint = url;
   console.log("url :", url);
```

```
    /pp.get('/', cors(corsOptions), async (req, res) => {
    console.log("err ")
    try {
        const fetchOptions = {
            method: 'POST',
            headers: {
                "Content-Type": "application/x-www-form-urlencoded"
            }
        }
        const response = await fetch(requestEndpoint, fetchOptions);
        const jsonResponse = await response.json();
        if (jsonResponse["access_token"] !== undefined) {
            res.send(jsonResponse["access_token"]);
        }
        else
            res.send("ERROR");
        console.log(" jsonResponse[\"access_token\"]", jsonResponse["access_token"]);
        ret = jsonResponse["access_token"]
    } catch (err) {
        console.log(" err ", err)
    }
});
app.listen(PORT, () => {
    console.log(`Example app listening at http://localhost:${PORT}`);
});
/**
 * Puppeteer
 */
const puppeteer = require('puppeteer');
const fs = require('node:fs')
const linkedinURL = 'https://www.linkedin.com/in/curious-mohammed-abdullah/';
const certURL = linkedinURL + "details/certifications/";
const projectsURL = linkedinURL + "details/projects/";
const skillsURL = linkedinURL + "details/skills/";

class SkillClass {
    constructor(skills) {
        this.skillsGlobal = skills;
    }
}
class CertClass {
    constructor(certificates) {
        this.certificatesGlobal = certificates;
    }
}
class LinkDataClass {
```

```
constructor(url) {
    this.linkedinURL = url;
    this.certURL = this.linkedinURL + "details/certifications/";
    this.projectsURL = this.linkedinURL + "details/projects/";
    this.skillsURL = this.linkedinURL + "details/skills/";
    this.start = this.start.bind(this);
    this.linkedinData = this.linkedinData.bind(this);
}
async start() {
    await Promise.all([
         await this.login(),
        await this.skillsData(),
        await this.certData(),
        await this.projData(),
        await this.linkedinData(),
    ])
    console.log("close")
}
async login() {
const browser = await launchBrowser(isHeadlessMode);
  const page = await browser.newPage();
  console.log("linkedinURL", this.linkedinURL)
  await Promise.all([
        await page.goto("https://www.linkedin.com/authwall", { waitUntil: 'load', timeout:
0 }),
         await page.setDefaultNavigationTimeout(60000);
         await page.waitForSelector(".authwall-join-form__title"),
         await page.waitForSelector("body div"),
        await page.waitForNetworkIdle()
    ]
    );
    await browser.close()
   }
  async linkedinData() {
     const browser = await launchBrowser(isHeadlessMode);
     const page = await browser.newPage();
     console.log("linkedinURL", this.linkedinURL)
     await Promise.all([
        await page.goto(this.linkedinURL),
        await            page.waitForSelector(".text-heading-xlarge.inline.t-24.v-align-
middle.break-words"),
        await     page.waitForSelector(".display-flex.flex-wrap.align-items-center.full-
height"),
         await page.waitForSelector("body div"),
        await page.waitForNetworkIdle(),
     ]
     );
```

```
        const work = await page.evaluate(() => {
return JSON.stringify({ "work": [{
"company": document.querySelector(
`#${document.getElementById('experience').parentNode.id}  div ul li div div div div .t-
14.t-normal .visually-hidden`).innerText,
"title": document.querySelector(`
#${document.getElementById('experience').parentNode.id} div ul li div div div div div
span .visually-hidden`).textContent,
"dates":
document.querySelector(`#${document.getElementById('experience').parentNode.id}  .t-
14.t-normal.t-black--light .visually-hidden`).innerText,
                "location":
document.querySelectorAll(`#${document.getElementById('experience').parentNode.id}
.t-14.t-normal.t-black--light .visually-hidden`)[1].textContent,
                "thumbnail":
document.querySelectorAll(`#${document.getElementById('experience').parentNode.id}
.ivm-view-attr__img-wrapper.ivm-view-attr__img-wrapper--use-img-tag.display-flex >
img`)[0].src,
                description:
document.querySelectorAll(`#${document.getElementById('experience').parentNode.id}
.inline-show-more-text.inline-show-more-text--is-collapsed.inline-show-more-text--is-
collapsed-with-line-clamp.full-width .visually-hidden`)[0].textContent
            },
            ],
          })
```

```
    })
    const education = await page.evaluate(() => {
      return JSON.stringify({
        "education": [{
          "Institution":
document.querySelector(`#${document.getElementById("education").parentElement.id}
.pvs-list__outer-container .pvs-list .artdeco-list__item.pvs-list__item--line-separated.pvs-
list__item--one-column .pvs-entity.pvs-entity--padded.pvs-list__item--no-padding-in-
columns .display-flex.flex-column.full-width.align-self-center .display-flex.flex-
row.justify-space-between .optional-action-target-wrapper.display-flex.flex-column.full-
width .display-flex.flex-wrap.align-items-center.full-height .mr1.hoverable-link-text.t-
bold .visually-hidden`).innerText,
          "Program":

document.querySelector(`#${document.getElementById("education").parentElement.id}
.pvs-list__outer-container .pvs-list .artdeco-list__item.pvs-list__item--line-separated.pvs-
list__item--one-column .pvs-entity.pvs-entity--padded.pvs-list__item--no-padding-in-
columns .display-flex.flex-column.full-width.align-self-center .display-flex.flex-
row.justify-space-between .optional-action-target-wrapper.display-flex.flex-column.full-
width .t-14.t-normal .visually-hidden`).innerText,
          "YearOfPassing":

document.querySelector(`#${document.getElementById("education").parentElement.id}
.pvs-list__outer-container .pvs-list .artdeco-list__item.pvs-list__item--line-separated.pvs-
list__item--one-column .pvs-entity.pvs-entity--padded.pvs-list__item--no-padding-in-
columns .display-flex.flex-column.full-width.align-self-center .display-flex.flex-
row.justify-space-between .optional-action-target-wrapper.display-flex.flex-column.full-
width .t-14.t-normal.t-black--light .visually-hidden`).innerText,
          "Grade":

document.querySelector(`#${document.getElementById("education").parentElement.id}
div ul li div div div div div div .visually-hidden`).innerText,
          "website":
            "",
        },
        ],
      })
    })
    const data = await page.evaluate(() => {
      const DATA = {
        "information": {
          "name":   document.querySelectorAll(".text-heading-xlarge.inline.t-24.v-
align-middle.break-words")[0].innerText,
          "domain":          document.querySelectorAll(".text-body-medium.break-
words")[0].innerText,
          "image":           document.querySelectorAll(".ember-view.profile-photo-
edit__preview")[0].getAttribute('src'),
          email: email,
         "description": document.querySelectorAll(".inline-show-more-text.inline-
show-more-text--is-collapsed.inline-show-more-text--is-collapsed-with-line-clamp.full-
width")[0].innerText,
            "profiles": [
                {
                    "media": "Linkedin",
                    "url": "",
                    "icon": "./img/icons/media/linkedin.png"
                },
            ],
        }
    }
    return DATA;
}).then((DATA) => {
    DATA.information.profiles[0].url = this.linkedinURL;
    return JSON.stringify(DATA)
});
Promise.all([
    work,
    education,
])
var DATA = Promise.resolve(data)
DATA.then(() => {
    console.log("data1: ", data);
    fs.writeFile('data/data.json', data, () => console.log("data write... "))
   fs.writeFile('data/education.json', education, () => console.log("data write... "))
```

```
        fs.writeFile('data/work.json', work, () => console.log("data write... "))
      })
      await browser.close()
    }
    async skillsData() {
      const browserSkill = await launchBrowser(isHeadlessMode);
      const pageSkill = await browserSkill.newPage();
      console.log("this.skillsURL", this.skillsURL)
      await Promise.all([
        await pageSkill.goto(this.skillsURL),
        await pageSkill.waitForSelector(".t-20.t-bold.ph3.pt3.pb2"),
        await pageSkill.waitForNetworkIdle(),
      ])
      const skills = await pageSkill.evaluate(() => {
        const data = new Set(Array.from(document.querySelectorAll("li > div > div > div
> div > a > div > span > .visually-hidden")).map((skillsData) => skillsData.textContent))
        return [...data].join(", ");
      });
      this.skillsObj = new SkillClass(skills);
      console.log("skills: ", skills)
      fs.writeFile('data/skills.json', `{
        "skills": [
          {
                "type": "",
                "KnowledgeInAdvanceTopics": "",
                "KnowledgeInMainConcepts": ${JSON.stringify(skills)},
          "Beginner": ""
       }
   ]
}`, () => console.log("skills write.. "))
       await browserSkill.close();
   }
   async certData() {
       const browserCert = await launchBrowser(isHeadlessMode);
       const pageCert = await browserCert.newPage();
       console.log("certURL", this.certURL)
       await Promise.all([
          await pageCert.goto(this.certURL),
          await pageCert.waitForSelector(".t-20.t-bold.ph3.pt3.pb2"),
          await pageCert.waitForNetworkIdle()
       ])
       const certificates = await pageCert.evaluate(() => {
          var titles = Array.from(document.querySelectorAll("div > div div > ul > li > div >
div > div > div > a > div > span > .visually-hidden")).map((e) => e.textContent);
          const cred = Array.from(document.querySelectorAll("section div div div ul li
.pv2 a")).map((e) => e.href)
```

```
        var subData = Array.from(document.querySelectorAll("div > div div > ul > li > div > div > div > div > a > span > .visually-hidden")).map((e) => e.textContent)
        let woCredname = Array.from(document.querySelectorAll("div > div div > ul > li > div > div > div > div > div > div > span > .visually-hidden")).map((e) => e.textContent);
        let woCred = Array.from(document.querySelectorAll("div > div div > ul > li > div > div > div > div > div > span > .visually-hidden")).map((e) => e.textContent);
        titles = titles.concat(woCredname)
        subData = subData.concat(woCred)
        return JSON.stringify(Array.from(titles).map((e, i) => {
            return {
                "title": e,
                "date": subData[3 * i + 1],
                "institution": subData[3 * i],
                "thumbnail": cred[i],
                "description": ""
            }
        }));
    });
    this.certObj = new CertClass(certificates);
    console.log("certificates: ", certificates)
    fs.writeFile('data/certificates.json', `{\"certificates"\: ${certificates}}`, () => console.log("certificate write.. "))
    await browserCert.close()
```

```
}
async projData() {
    const browserProj = await launchBrowser(isHeadlessMode);
    const pageProj = await browserProj.newPage();
    console.log("this.projectsURL", this.projectsURL)
    await Promise.all([
        await pageProj.goto(this.projectsURL),
        await pageProj.waitForSelector(".t-20.t-bold.ph3.pt3.pb2"),
        await pageProj.waitForNetworkIdle()
    ])
    const projects = await pageProj.evaluate(() => {
        var titles = Array.from(document.querySelectorAll(`main section div div div ul li
div div div div div div span .visually-hidden`)).map((e) => e.innerText);
        var dates = Array.from(document.querySelectorAll(`main section div div div ul li
div div div div .display-flex.flex-column.full-width .t-14.t-normal .visually-
hidden`)).map((e) => e.innerText);
        let gitlink = Array.from(document.querySelectorAll(`ul li div div div div ul li
a`)).map((e) => e.href);
        let description= Array.from(document.querySelectorAll(`li div div div div ul li
div ul li div div div .visually-hidden`)).map((e) => e.innerText);
        return JSON.stringify(Array.from(titles).map((e, i) => {
            return {
                "title": e,
                "dates": dates[i],
                "type": "",
                "thumbnail": "",
                "link": "",
                "gitlink": gitlink[i],
                "description": description[i]
            }
        }));
    });
    this.certObj = new CertClass(projects);
    console.log("projects: ", projects)
    fs.writeFile('data/projects.json',      `{\"projects"\:      ${projects}}`,      ()      =>
console.log("certificate write.. "))
    await browserProj.close()
  }
}
app.get("/skills", (req, res) => {
  fs.readFile("data/skills.json", 'utf-8', (err, data) => {
    console.log("read... Skills", data);
    res.send(data);
    console.log("sent skills", data)
  });
})
app.get("/certificates", (req, res) => {
    fs.readFile("data/certificates.json", 'utf-8', (err, data) => {
        console.log("read... Certificates");
        res.send(data);
        console.log("sent certificates", data)
    });
})
app.get("/data", (req, res) => {
    fs.readFile("data/data.json", 'utf-8', (err, data) => {
        res.send(data);
    });
})
app.get("/work", (req, res) => {
    fs.readFile("data/work.json", 'utf-8', (err, data) => {
        res.send(data);
    });
})
app.get("/education", (req, res) => {
    fs.readFile("data/education.json", 'utf-8', (err, data) => { res.send(data);});
 })
```

CALLBACK HANDLER.JS

```
import * as React from 'react'
export default function CallbackHandler() {
```

```
const codeParam = new URL(window.location.href).searchParams;
const auth = codeParam.get('code');
let [error, setError] = React.useState("");
const errorCode = "Error has occured, please retry after sometime.";

localStorage.setItem('linkedinAuthCode', auth);
let code = "";
const url1 = `http://localhost:8080/auth/${auth}`;
if (code !== undefined || code !== "") {
    try {
        fetch(url1, { mode: 'cors' })
            .then(res => {
                res.text().then(res => {
                    console.log("res ", res);
                    if (res === "ERROR") {
                        setError(errorCode);
                        console.log("code: ", code, "---", error)
                    }
                    else {
                        localStorage.setItem('linkedinAccessCode', res);
                        let                    win                    =
window.open(`https://api.linkedin.com/v2/userinfo?oauth2_access_token=${res}`,
"name", 'height=600,width=450');
```

```
                    if (win)
                        win.focus()
                }
            })
        })
        // .then((data) => window.close())
    } catch (err) {
        setError(errorCode);
        console.log(error + err)
    }
} else {
    console.log("skip")
}
const info = <>
    <center>
        <div>
            Redirecting
        </div>
        <br />
        <div>
            This window will close automatically
            {error}
        </div>
        </center>
    </>
    return (
        <>
            {error !== errorCode ? info : errorCode}
        </>
    );
}
```

REFERENCES:

1. G. Jung, S. Han, H. Kim, K. Kim and J. Cha, "Extracting the Main Content of Web Pages Using the First Impression Area," in IEEE Access, vol. 10, pp. 129958-129969, 2022, doi: 10.1109/ACCESS.2022.3229080.
2. Department of Instructional Technology (1999) Portfolio Examination Guidelines for the Master's Degree University of Georgia [online] Accessed 14 March 2007 http://it.coe.uga.edu/pdf/portguide.pdf
3. Ecclestone, K. (1999) Empowering or Ensnaring?: The Implications of Outcome-based Assessment in Higher Education. Higher Education Quarterly, 53, 29-48
4. Phillips, H., Parsons, J., Duranton, H. & Ramos, A. (2004) Using Blackboard to support a portfolio based unit (Interact Magazine, LTSS) University of Bristol [online] http://www.bristol.ac.uk/ceas/members/associates/ phillips.shtml
5. Reeves, T. C. (2000) Alternative assessment approaches for online learning environments in higher education. Higher Education Journal of Educational Computing Research 23, 101-111
6. Smith, K., & Tillema, H. (2003). Clarifying different types of portfolio use. Assessment and Evaluation in Higher Education, 28(6), 625-648.
7. Thorpe, M. (1998) Assessment and 'third generation' distance education. Distance Education, 19, 265-286 University of Dundee (2008) MyDundee University of Dundee [online] https://my.dundee.ac.uk/webapps/portal/ frameset.jsp
8. Wade, R. C., & Yarbrough, D. B. (1996). Portfolios: A tool for reflective thinking in teacher education? Teaching and Teacher Education, 12(1), 63-79.
9. Zeichner, K., & Wray, S. (2001). The teaching portfolio in US teacher education programs: What we know and what we need to know. Teaching and Teacher Education, 17(5), 613-621.

10.Darling, L. F. (2001). Portfolio as practice: The narratives of emerging teachers. Teaching and Teacher Education, 17(1), 107-121.