**Real-Time Team Collaboration Platform For Software Project Management**

E.A.R SAHAN EKANAYAKA

EUSL/TC/IS/2018/COM/63

18/COM/261

Submitted in partial fulfillment of the requirements for the award of the degree

Bachelor of Science in Computer Science [BSc (CS)] to the Department of Computer Science,

Faculty of Applied Science, Trincomalee Campus, Eastern University, Sri Lanka

Date of Submission

31/01/2024

**Declaration**

I hereby declare that the entire work embodied in this research work has been carried out by me. The extent of information derived from the existing literature has been documented and fully acknowledged at the appropriate places, the work is original and has not been submitted in part or full for any Diploma or Degree in this or any other University. I confirm that there is no plagiarism in this document and if detected, I abide by the action that will be taken for such plagiarism by the Faculty of Applied Science, Eastern University, Sri Lanka.

….…………………………….

E.A.R Sahan Ekanayaka

EUSL/TC/IS/2018/COM/63

Department of Computer Science

Faculty of Applied Science

Trincomalee Campus, Eastern University, Sri Lanka

**Certification of the Supervisors**

This is to certify that this research report entitled “Real-Time Team Collaboration Platform For Software Project Management” submitted by E.A.R Sahan Ekanayaka for the degree of Bachelor of Science in Computer Science is a record of research work carried out by him/her under our guidance and direct supervision and that it has not been previously formed the basis for the award of any degree, diploma, associateship, fellowship or any other similar title.

This is also to certify the document represents the original independent work of the candidate.

…..……………………………. …………………………….

Signature of Co-Supervisor Date

Mrs.S.S Ketharan

Lecturer (Probationary)

Department of Computer Science

Trincomalee Campus, Eastern University, Sri Lanka

…..……………………………. …………………………….

Signature of Supervisor Date

Mr.S Thadchanamoorthy

Senior Lecturer Gr-I

Department of Computer Science

Trincomalee Campus, Eastern University, Sri Lanka

**Acknowledgment**

First and foremost, my heartily profound gratitude and appreciation are addressed to my co-supervisor Mrs.S.S Ketharan and the supervisor Mr.S Thadchanamoorthy for their valuable supervision. His/Her advice, discussions and guidance were the real encouragement to complete this work. I admire his/her creativity, simplicity, generosity, work ethic, and ability to balance work and life. It has been an honour to work with him/her. I will always be thankful to him/her for the valuable time that they spent supervising my progress.

I would also like to thank Mr.S Thadchanamoorthy - Head, Department of Computer Science, Faculty of Applied Science, Trincomalee Campus, Eastern University, Sri Lanka and all the lecturers of the faculty for facilitating carrying out my research work.

Finally, I am indebted to my parents who have supported and encouraged me through their kindness and affection so that I could concentrate on my studies. They touched me more deeply than I could have ever expected.

**Abstract**

This study presents a revolutionary "Real-Time Team Collaboration Platform for Software Project Management," created to satisfy the changing requirements of modern software development teams. Built on top of the MERN stack, the platform creates a dynamic, collaborative environment by leveraging cutting-edge technologies like WebSockets. Strengthening security by deploying a strong authentication system based on JSON Web Tokens is a primary goal. Simultaneously, the platform incorporates an advanced machine learning-driven content recommendation engine, namely the K-Nearest Neighbor algorithm with Cosine Similarity. This recommendation system evaluates the user's experience with particular project fields and provides customized recommendations based on personal knowledge.

The project's results show observable advancements in several important areas. First, by using a token-based method, the authentication technique improves security by guaranteeing that the access token is not stored in a persistent manner, which stops unwanted access. Additionally, the platform becomes more efficient when WebSockets are used for real-time communication. The smooth real-time file-sharing function is especially impressive as it lets users transfer big files without compromising the responsiveness or ease of use of the application. A sophisticated approach to user experience is seen in the real-time upload status monitoring available to both senders and recipients.

The content-based recommendation system deviates from traditional approaches in combination with advances in real-time communication. By explicitly assessing the familiarity with domains they have worked with, users create a customized score for project recommendations. These ratings are processed using the instance-based learning-based K-Nearest Neighbor algorithm, which finds projects that match user preferences. To enhance the accuracy of project suggestions, the Cosine Similarity measure offers a more nuanced assessment of domain preferences.

With an eye toward the future, the study suggests possible directions for advancement. Investigating real-time communication protocols other than WebSockets, including Server-Sent Events, Web Real-Time Communication, or Message Queuing Telemetry Transport, is one interesting path to take. Furthermore, including contemporary recommendation techniques like deep learning models and matrix factorization is a viable path forward for improving the content-based recommendation system. These futuristic considerations highlight the project's dedication to being at the forefront of technical developments, guaranteeing the platform's applicability and efficiency in a constantly changing software development environment.

Essentially, this research project integrates security, effective real-time communication, and customized project advice into a comprehensive approach to software project management. The platform is evidence of the inventiveness that can be achieved by utilizing cutting-edge technology to improve teamwork and output in software development teams.

**Table of contents**

[Chapter 01 : Introduction 8](#_Toc157145841)

[Project Overview : 8](#_Toc157145842)

[Background : 9](#_Toc157145843)

[Chapter 02 : Related Work 10](#_Toc157145844)

[1. Evolving Real-Time Communication Protocols: WebSockets as a Paradigm Shift 10](#_Toc157145845)

[2. Innovating Project Recommendations: Content-Based Approach and User Domain Familiarity 11](#_Toc157145846)

[Chapter 03 : Tools and Techniques 12](#_Toc157145847)

[ Programming Languages 12](#_Toc157145848)

[ Front End Technologies 12](#_Toc157145849)

[ Back End Technologies 12](#_Toc157145850)

[ Database 12](#_Toc157145851)

[ Machine Learning Libraries 12](#_Toc157145852)

[ Protocols 12](#_Toc157145853)

[ Application Programming Interface(API) Testing Tools 13](#_Toc157145854)

[ Integrated Development Environment (IDE) 13](#_Toc157145855)

[ Architecture 13](#_Toc157145856)

[Chapter 04 : Methodology 14](#_Toc157145857)

[1). Efficient Websocket management with HTTP protocol utilization. 14](#_Toc157145858)

[2). Establishment of a Secure Authentication Mechanism 15](#_Toc157145859)

[3). Development of Scalable Content-Based Project Recommendation System 17](#_Toc157145860)

[Chapter 05 : Results and Discussion 19](#_Toc157145861)

[1. The enhanced authentication mechanism. 19](#_Toc157145862)

[2. Efficient web socket management. 19](#_Toc157145863)

[3. Content-Based Recommendation of projects 20](#_Toc157145864)

[Chapter 06 : Future Work 23](#_Toc157145865)

[1. Enhancing the real-time socket management further with different architectures. 23](#_Toc157145866)

[2. Enhance the content-based recommendation system further. 23](#_Toc157145867)

[Chapter 07 : Conclusion 24](#_Toc157145868)

[References 25](#_Toc157145869)

# Chapter 01 : Introduction

## Project Overview :

The Software Project Management Platform is conceived as an integrated solution catering to the intricate requirements of modern software development teams. Its primary objectives are to revolutionize collaboration, enhance project management efficiency by providing more convenient WebSockets management, and help users to find out most suitable projects according to their familiarity domain expertise through a machine learning apporach.

The intended audience comprises software development teams across industries, including tech startups, established software firms, and freelance developers. The platform targets individuals and teams seeking a cohesive, real-time collaborative environment for software project execution.

The scope of the project involves the development of a web-based platform leveraging the MERN (MongoDB, Express.js, React, Node.js) stack to facilitate seamless communication and project recommendation. The platform will utilize WebSockets technology to enable real-time interactions, ensuring instant communication and updates for distributed teams.

The approach taken is iterative development following Agile methodologies, allowing for flexibility and adaptability to evolving requirements. Key assumptions underlying this project include a foundational understanding of software development principles, the availability of reliable internet connectivity for users, and an evolving software landscape requiring continuous updates and improvements.

The pivotal outcome of this project is the delivery of a robust, user-friendly platform that combines cutting-edge technologies to empower software development teams. Key features include a secure authentication mechanism, real-time communication tools and a content-based recommendation system powered by machine learning. These components collectively aim to enhance team productivity and foster a dynamic, collaborative environment for software development endeavors.

## Background :

The aim of this research work is to develop a comprehensive Software Project Management Platform that integrates real-time team collaboration and a personalized project recommendation system based on individual domain familiarity ratings.

The goal is to develop an innovative solution that combines cutting-edge technologies, including WebSockets for real-time interactions, secure authentication mechanisms and a content-based recommendation system powered by machine learning algorithms. The project's core challenge lies in seamlessly integrating these components within a unified, scalable platform to optimize software development team productivity and project outcomes.

# Chapter 02 : Related Work

## 1. Evolving Real-Time Communication Protocols: WebSockets as a Paradigm Shift

In exploring existing solutions within real-time communication mechanisms for collaborative platforms, a prevalent method has been the use of polling systems. Polling mechanisms involve repeated client-server requests for updates, often resulting in increased network traffic and server load. While effective in certain contexts, these systems present limitations in providing instant, seamless interactions and can lead to latency issues in dynamic environments such as software development collaboration.

Recent research, exemplified by a study [2], [3] emphasizing the significance of leveraging WebSockets, highlights the drawbacks of traditional polling mechanisms. WebSockets, a protocol enabling bidirectional communication channels between clients and servers, offer a distinct advantage in facilitating real-time interactions without the overhead associated with polling. This protocol enables persistent connections, allowing data to be transmitted instantly in both directions, significantly reducing latency and network overhead compared to polling-based systems.

However, the drawbacks of polling, such as increased latency and network congestion, have been acknowledged in the community. This inadequacy serves as a crucial consideration in the design and implementation of the proposed Software Project Management Platform.

In contrast to prevalent solutions reliant on polling mechanisms, the proposed platform leverages the WebSocket protocol for real-time communication. By adopting WebSockets, the platform aims to mitigate latency issues, reduce network overhead, and offer instantaneous, bidirectional communication channels. This strategic choice addresses the deficiencies identified in traditional polling systems, aligning with the evolving demands of modern software development teams for seamless, real-time collaboration.

## 2. Innovating Project Recommendations: Content-Based Approach and User Domain Familiarity

The proposed Software Project Management Platform aims to innovate project recommendations through a content-based approach. This approach factors in users explicit familiarity feedback on domains they've worked with, enabling a more personalized and precise project recommendation system.

By leveraging users' explicit ratings for their familiarity with specific domains or technologies, the platform aims to provide recommendations aligned with individual expertise. This departure from conventional collaborative filtering methods marks a strategic shift towards personalized project recommendations that enhance team composition and project assignment within software development teams.

The adoption of a content-based recommendation approach based on users' domain familiarity feedback signifies a novel feature. This innovative strategy aims to address the deficiencies observed in generic recommendation systems, fostering a more tailored and effective project-team alignment within software development endeavors.

# Chapter 03 : Tools and Techniques

## Programming Languages

1. Javascript :- Javascript is used for developing the application including both frontend and backend.
2. Python :- Python is used for implementing content-based recommendation system

## Front End Technologies

1. React :- Employed as a Single Page Application (SPA), React facilitates a dynamic, responsive user interface, offering seamless interactions and enhanced user experience.
2. Tailwind CSS :- Chosen as the CSS framework which provides pre built CSS utility classes.

## Back End Technologies

1. NodeJS :- Leveraging its non-blocking event-driven architecture, NodeJS efficiently handles WebSockets, enabling real-time communication and ensuring responsive interactions among users.

## Database

1. MongoDB :- Utilized as a NoSQL database, MongoDB offers flexibility and scalability, accommodating complex data structures inherent in software project management.

## Machine Learning Libraries

1. Surprise :- A Python-based library tailored for building and evaluating recommendation systems. This versatile library offers a wide range of collaborative filtering algorithms, simplifying the process of implementing and testing various recommendation models.

## Protocols

1. Hyper Text Transfer Protocol (HTTP) :- HTTP is used as one of the communication protocols, even though the application leverages WebSocket protocol as well. This is mainly because HTTP is a reliable protocol and it can give more benefits in certain scenarios.
2. WebSocket Protocol :- WebSocket is a communication protocol that provides full-duplex communication channels over a single, long-lived connection between the client and the server. Unlike HTTP, which follows a request-response model, WebSocket enables bi-directional communication, allowing real-time data exchange between the client and server. This persistent connection capability significantly reduces latency and overhead, making it ideal for applications requiring instantaneous and continuous data transfer, such as real-time collaborative environments.

## Application Programming Interface(API) Testing Tools

1. Postman :- Postman is a widely used HTTP client that helps developers to test and document APIs. When testing APIs postman supports a variety of HTTP methods including GET, POST, PUT, PATCH, and DELETE as well. It can be installed as a separate software application on the computer. So in this application, postman has been used to test the Representational State Transfer(REST) Application Programming Interface(API) endpoints as well as to document those endpoints with their necessary details.
2. Thunder Client :- Thunder Client is a Visual Studio Code Extension that is used to test APIs. So using Thunder Client is easier than Postman because there is no need to install it locally as a software application. In this application, Thunder Client has been used to test API endpoints while developing them, because it is much easier to test API endpoints while developing with Thunder Client.

## Integrated Development Environment (IDE)

1. Visual Studio Code (VS Code) :- Visual Studio Code is a code editor that can be used for developing web and cloud-based applications. For this application, VS code has been selected because for the selected tech stack VS code fits well.
2. Anaconda :- Anaconda is a free open-source data science distribution for Python and R. In this application, Anaconda is used for creating Python environments and within those environments managing packages such as Surprise.

## Architecture

1. Hybrid :- In this application, there are two communication protocols are used. They are the Websocket protocol and HTP protocol. Here to achieve real-time communication capabilities with Websockets we have to migrate the server into a websocket server. But still, the HTTP protocol has been used to improve the efficiency. Since both protocols are being used this architecture is called the Hybrid architecture.

# Chapter 04 : Methodology

## 1). Efficient Websocket management with HTTP protocol utilization.

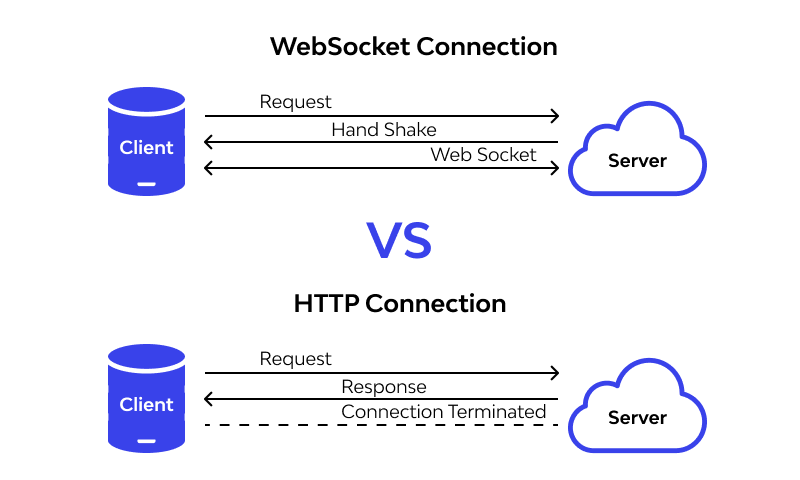
Since the platform needs real-time communication a deep dive into various approaches to achieve this was performed. It was found that there are several approaches. The approaches found are Long Polling, Short Polling, and Websocket. So it was found that the first two approaches are using regular HTTP protocol and those two are built upon top of HTTP protocol. So a decision was made after studying those various options and it was to use websocket protocol to achieve real-time communication along with the HTTP protocol.

Figure 1.0: HTTP vs Web Socket

To implement Websocket protocol a regular HTTP server must be converted into a Websocket server because of the difference in the Websocket protocol compared to the HTTP protocol. Figure 1.0 shows how the two protocols behave. According to it, the Websocket protocol will create a full duplex connection allowing bi-directional communication real-time data flow between the client and the server. Once the initial connection is established, both parties can send and receive data continuously without the need for separate requests. On the other hand, the HTTP protocol operates on a request-response model which will be terminated after the server's response for a given request. Because of this fact, HTTP protocol is a stateless protocol. On the other hand, Websocket protocol is a stateful protocol which makes it difficult to manage on the server. Therefore a study was done on how to manage Websocket protocol on the server using various data structures.

At the same time, the HTTP protocol was utilized to maintain consistency. Especially for tasks like initial resource fetching and create, update, and delete operations the HTTP protocol was utilized due to its cross-browser compatibility and consistency.

## 2). ****Establishment of a Secure Authentication Mechanism****

For application security, an improved authentication mechanism was implemented after studying various authentication strategies. First of all the application architecture was clearly defined. After that, it was found, that the most common authentication strategy which is cookies and sessions is not suitable for this application since the application has a totally decupled architecture. This means the front end acts as a Single Page Application(SPA) and the back end has the role of a Representational State Transfer(REST) Application Programming Interface(API). So the token-based authentication strategy was chosen as the best strategy. Here it was achieved with one of the most widely used token providers called JSON Web Tokens(JWT).

One of the major reasons caused to eliminating the cookie session authentication strategy was how the backend works. Since the backend acts as a REST API, it does not care about holding any state about who was logged in to the application. This makes it useless to store sessions on the server side. But what was done to enhance the security of this strategy further was to use the potential of cookies when signing the token on the server.

So to enhance the security of the token-based authentication two tokens were generated when a user logged in successfully. One is called the access token and the other one is called the refresh token. So the access token is what a user needs to provide when accessing protected API endpoints on the server. However, the access token has a short expiration time of between 5 to 15 minutes. The access token will not be stored on the browser's persistent storage (Neither local nor session storage). So it will be stored in the application state which means the memory so that no one can steal the access token. Since the access token will only be stored in memory after its expiration the user has to log in again to the platform (which means every time the access token expires user has to enter his/her login credentials and log in again that is not a good user experience). This issue was addressed using a refresh token, especially since the refresh token will be stored in an http-only cookie which prevents the cookie from being accessed via any client-side javascript. And then the refresh token can be used to generate a new access token once the existing access token is expired. This process was automated with the help of a library called Axios and its special feature called interceptors.

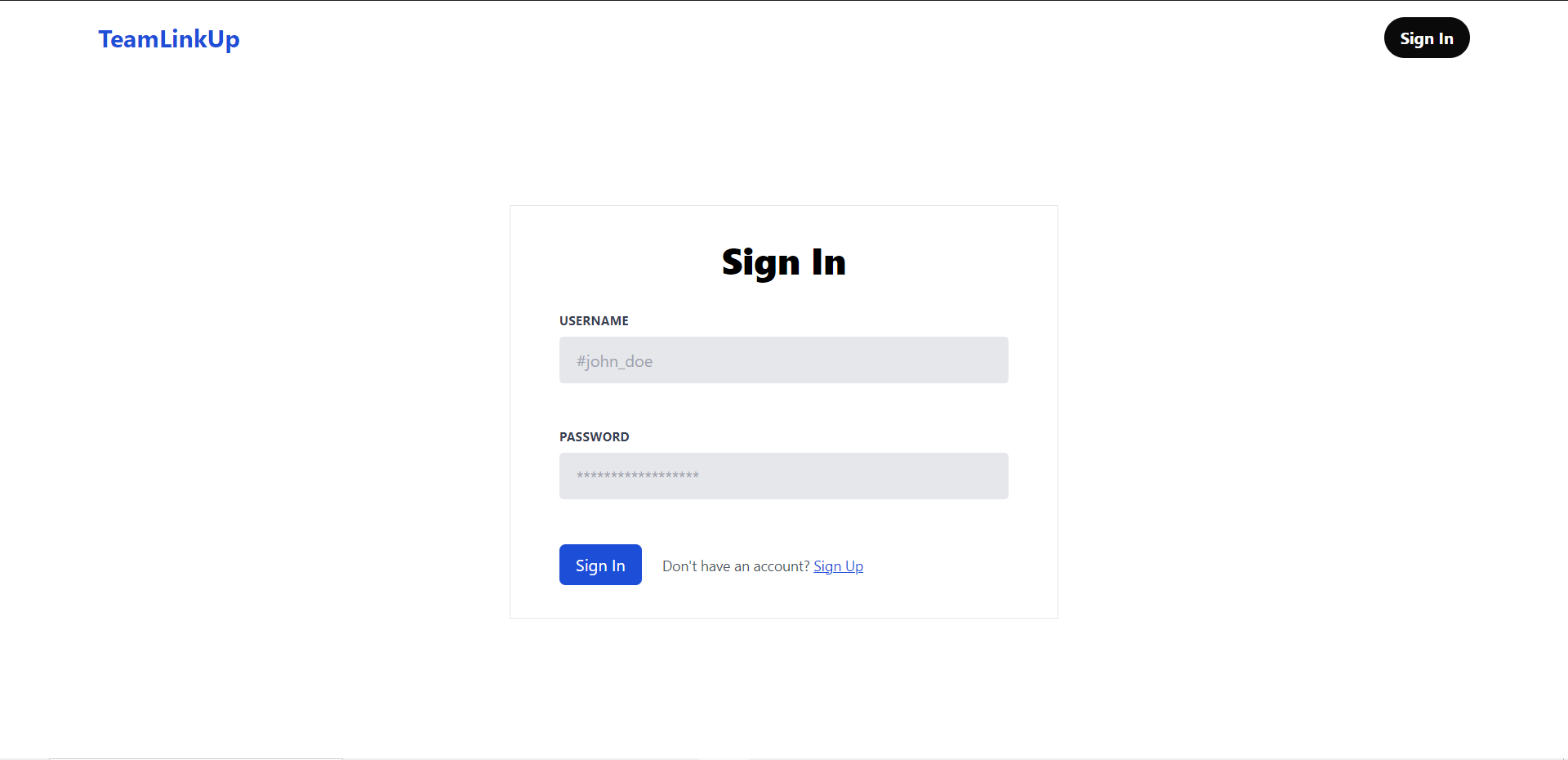


Figure 1.1: Sign In User Interface

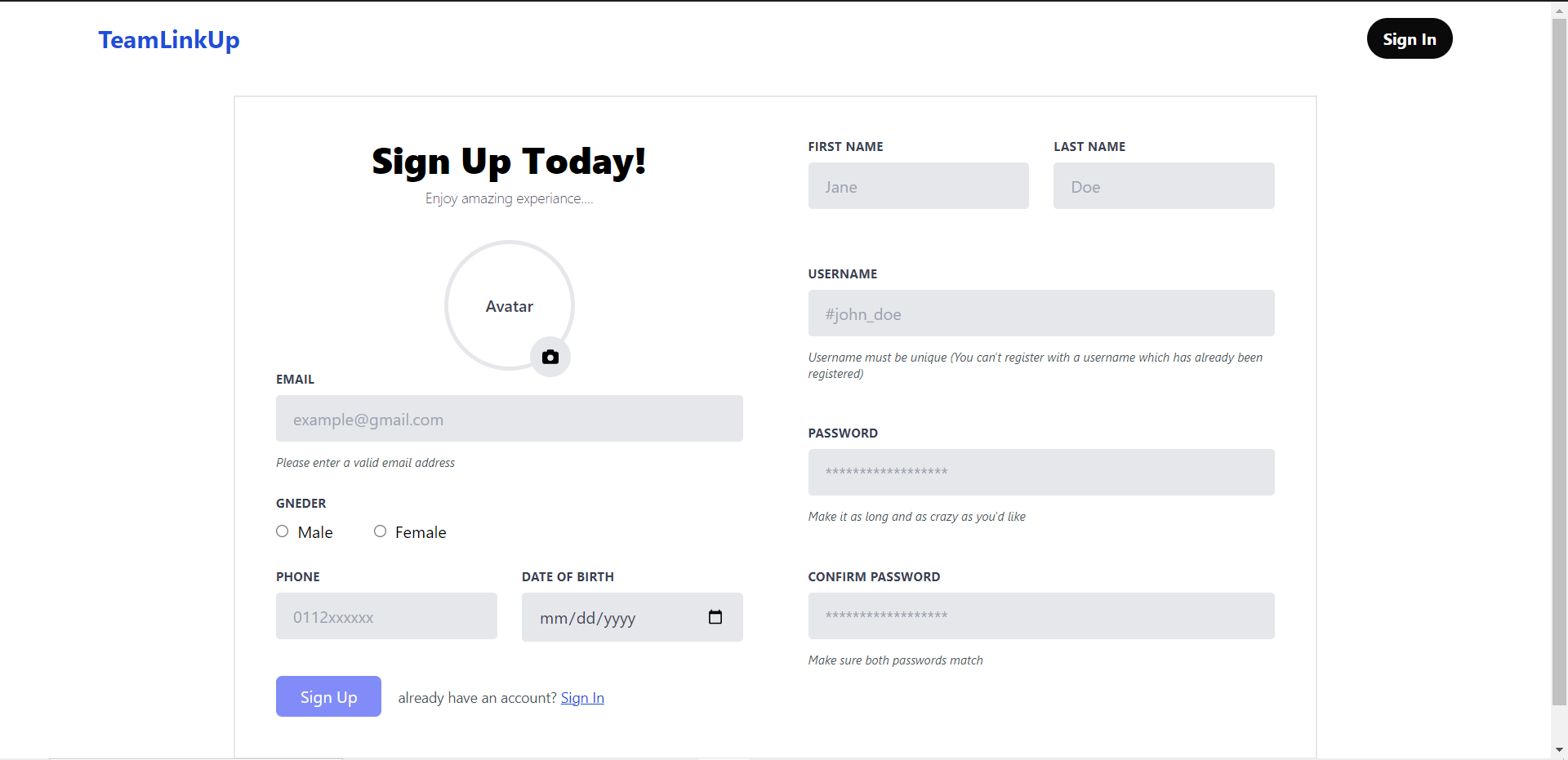


Figure 1.2: Sign Up User Interface

The above two figures (Figure 1.1 and Figure 1.2) show the User Interfaces were created for sign in and sign up processes.

## 3). ****Development of Scalable Content-Based Project Recommendation System****

With the evolvement of Machine Learning in modern applications, a recommendation machine learning system was established using content-based recommendation approach. This method was specifically chosen for the scalability and the lack of initial dataset. So an initial dataset of suggested projects was created and inserted to the database. An ability was also given to admins so that they can update this suggested projects collection. When a new user joins to the application, an ability was given to them to update their profile by providing the details of their previous work experience. There they can specify the domains that they have worked with. So based on those domains a set of suggested projects were displayed and they can provide their familiarity ratings for those suggested projects. Those explicit familiarity ratings were used to recommend further most suitable projects for individual user. This helps individual users to find out which projects are suit them very well. So in this system a machine learning algorithm called K-Nearest Neighbor (KNN) along with a similarity evaluation matric called “Cosine Similarity” were used.

The K-Nearest Neighbor (KNN) algorithm is a versatile and intuitive machine learning technique employed in the development of the recommendation system. KNN is a non-parametric, instance-based learning algorithm, commonly used for classification and regression tasks. In the context of the project recommendation system, KNN is utilized to identify projects that are most similar to a user's preferences based on their specified domains. The fundamental principle behind KNN is to predict the class or value of a data point by examining the majority class or average of its K nearest neighbors in the feature space. In the project recommendation system, each project and user profile is represented as a feature vector in a multi-dimensional space, with dimensions corresponding to the various domains and attributes associated with the projects.

Cosine Similarity is a metric widely employed in content-based recommendation systems to evaluate the similarity between two vectors. In the context of the project recommendation system, the vectors represent the domain preferences of users. Cosine Similarity calculates the cosine of the angle between two vectors, providing a measure of similarity ranging from -1 (completely dissimilar) to 1 (completely similar).

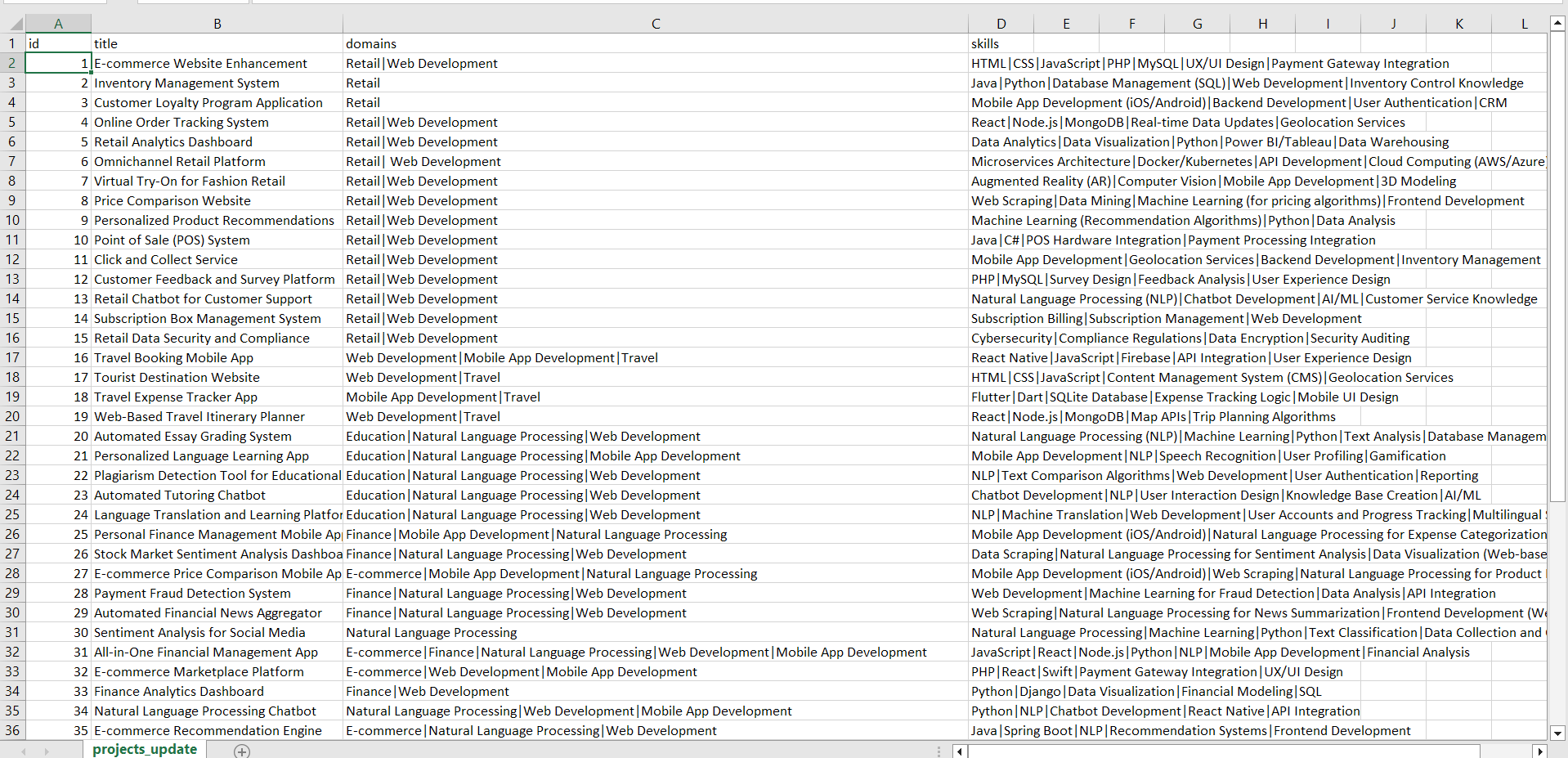


Figure 1.3: Suggested Projects Dataset

So the Figure 1.3 shows the dataset was created for initial suggested projects. About each project there are several details were taken including, project title, domains and skills. So there were eight unique domains included. So that a user can choose from those unique domains. Also the ability was given to admins so that they can update and add new domains and projects manually through the user interface.

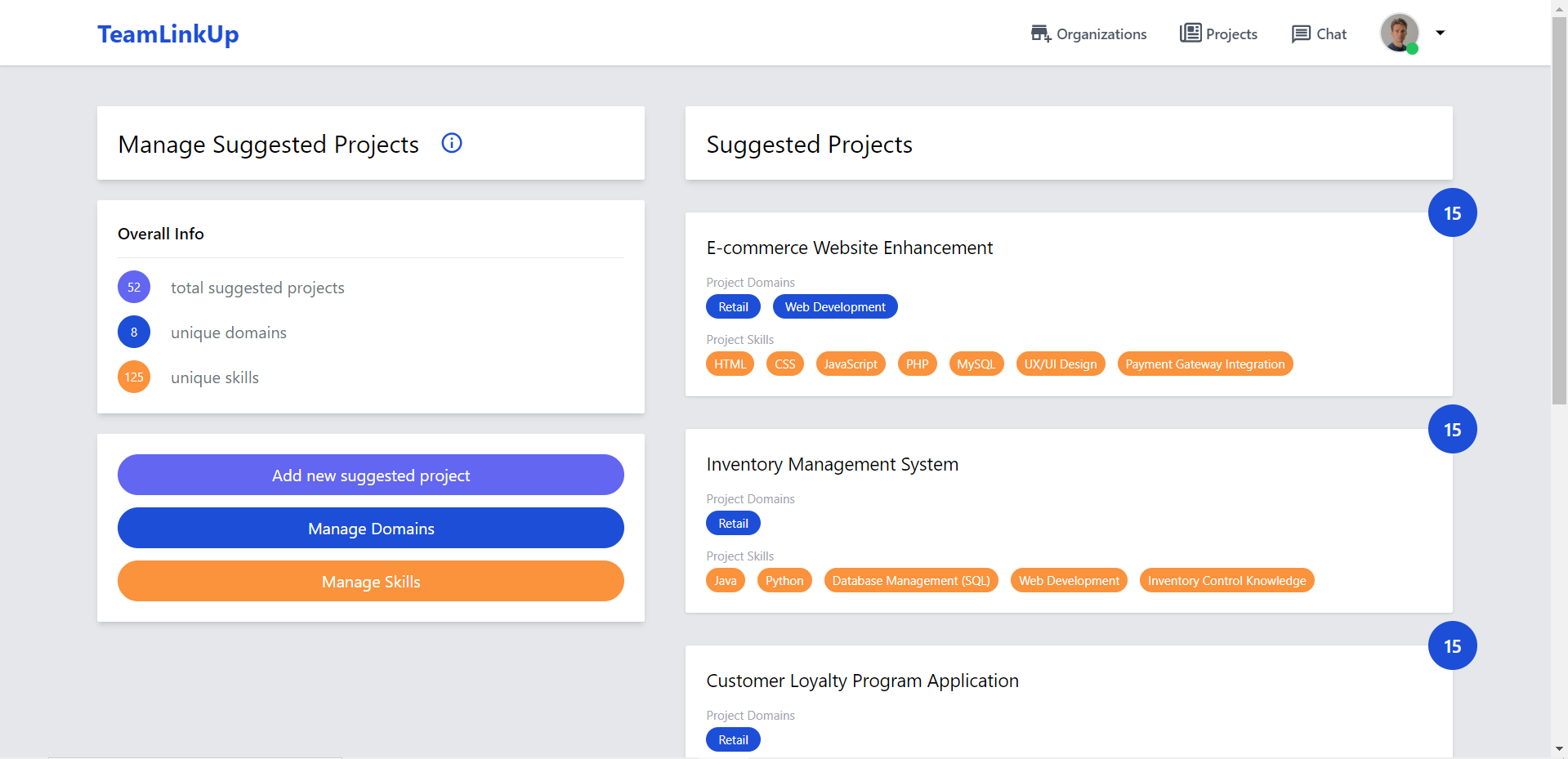


Figure 1.4: Admin User Interface For Managing Suggested Projects

The Figure 1.4 shows the admin User Interface which was Created for managing suggested projects.

# Chapter 05 : Results and Discussion

The results obtained from this research work can be categorized into three parts.

## 1. The enhanced authentication mechanism.

The authentication strategy implemented provides a great and more secure user experience. Because in this approach the access token won’t be stored in any persistence storage neither local storage nor session storage. It eliminates the ability to steal the token via any javascript based scripts. On the other hand by embedding the refresh token in a http only cookie, application ensures it also won’t be accessed via any client side javascript scripts. This helps to improve user experience.

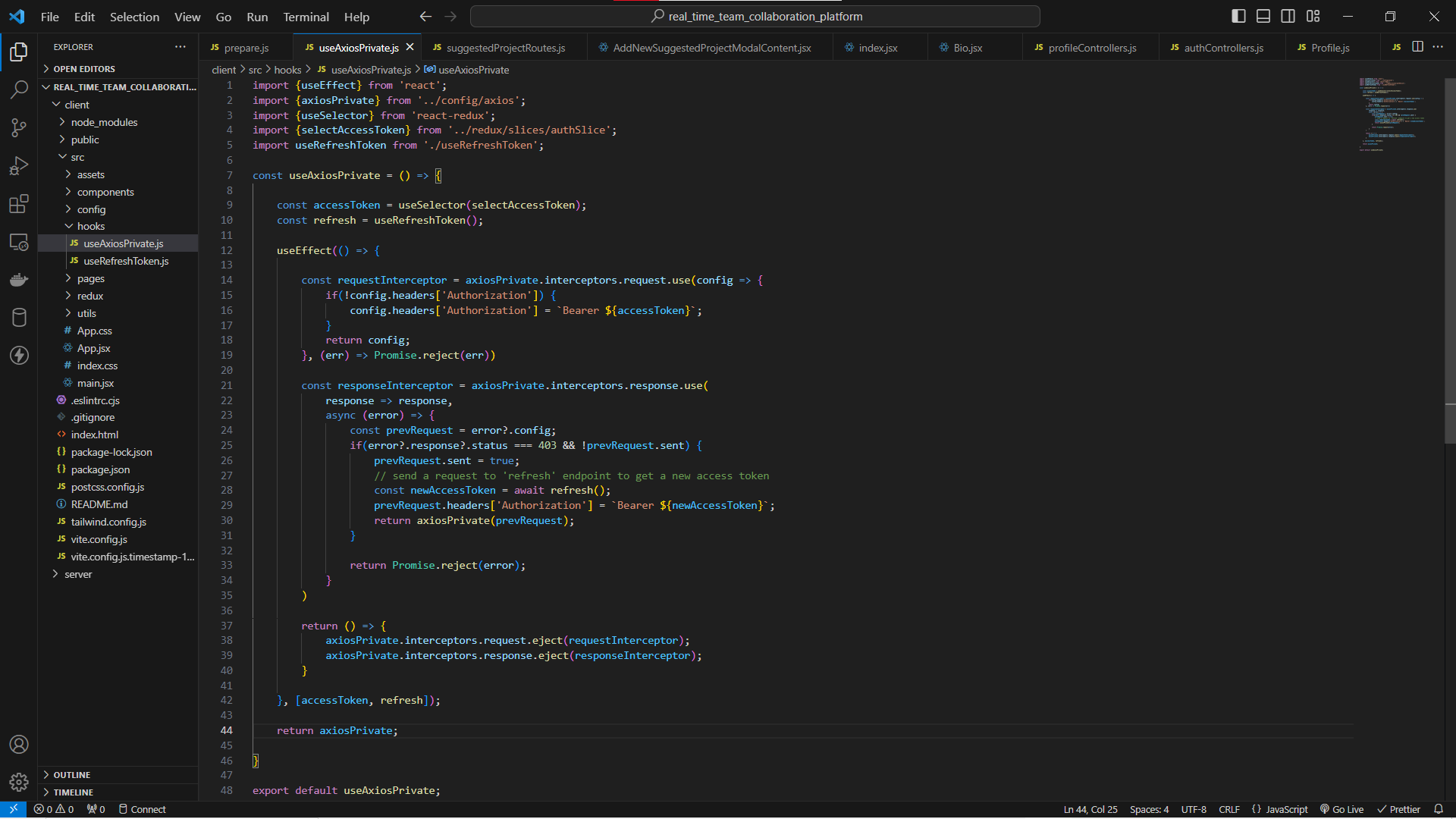


Figure 1.5: Axios Interceptors

Figure 1.5 shows how the axios interceptors were utilized to automate the access token revalidation process.

## 2. Efficient web socket management.

Another result obtained by the application is real-time communication using web socket protocol. So the real-time messaging and file sharing feature provides more efficient and seamless experience to application users. Here one of the most important features is the application provides the ability to share files (which are large in size). Since those large files are going to take more time to upload they all are managed on the server and the user can navigate all around the application as well as refresh the browser at any time without worrying about those updates. And whenever the user comes back to that particular chat window he/she will be able to see the progress in real-time. Not only that but also the receiver can also see the progress in real-time.

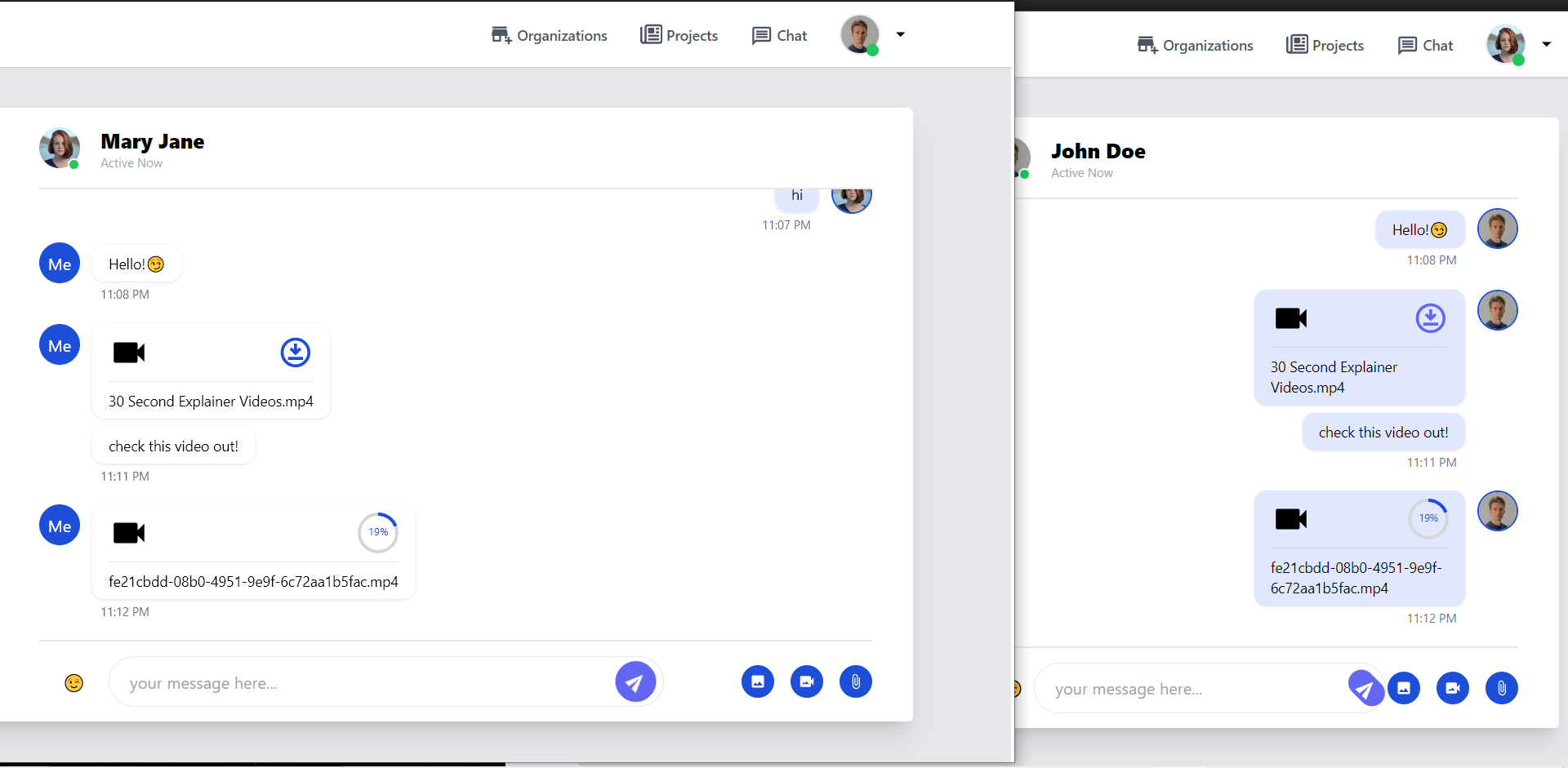
****

Figure 1.6: Real-time streaming upload progress

So the above Figure 1.6 demonstrates that how web sockets have been utilized to stream real-time upload progress to both sender and receiver.

## 3. Content-Based Recommendation of projects

When it comes to implementing the content-based recommendation system, there are several key accuracy metrics. So among those there are two most important metrices. They are Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Since those two metrices are error measuring metrices lower values will be considered as good result. So multiple projects ratings dataset was prepared and tested with Surprise python library with K-Nearest Neighbor algorithm to obtain values for MAE and RMSE. So the Figure 1.7 shows the prepared dataset that contains 200 rating records for the above suggested projects. And Figure 1.8 shows the python code implementation for testing two accuracy metrics.

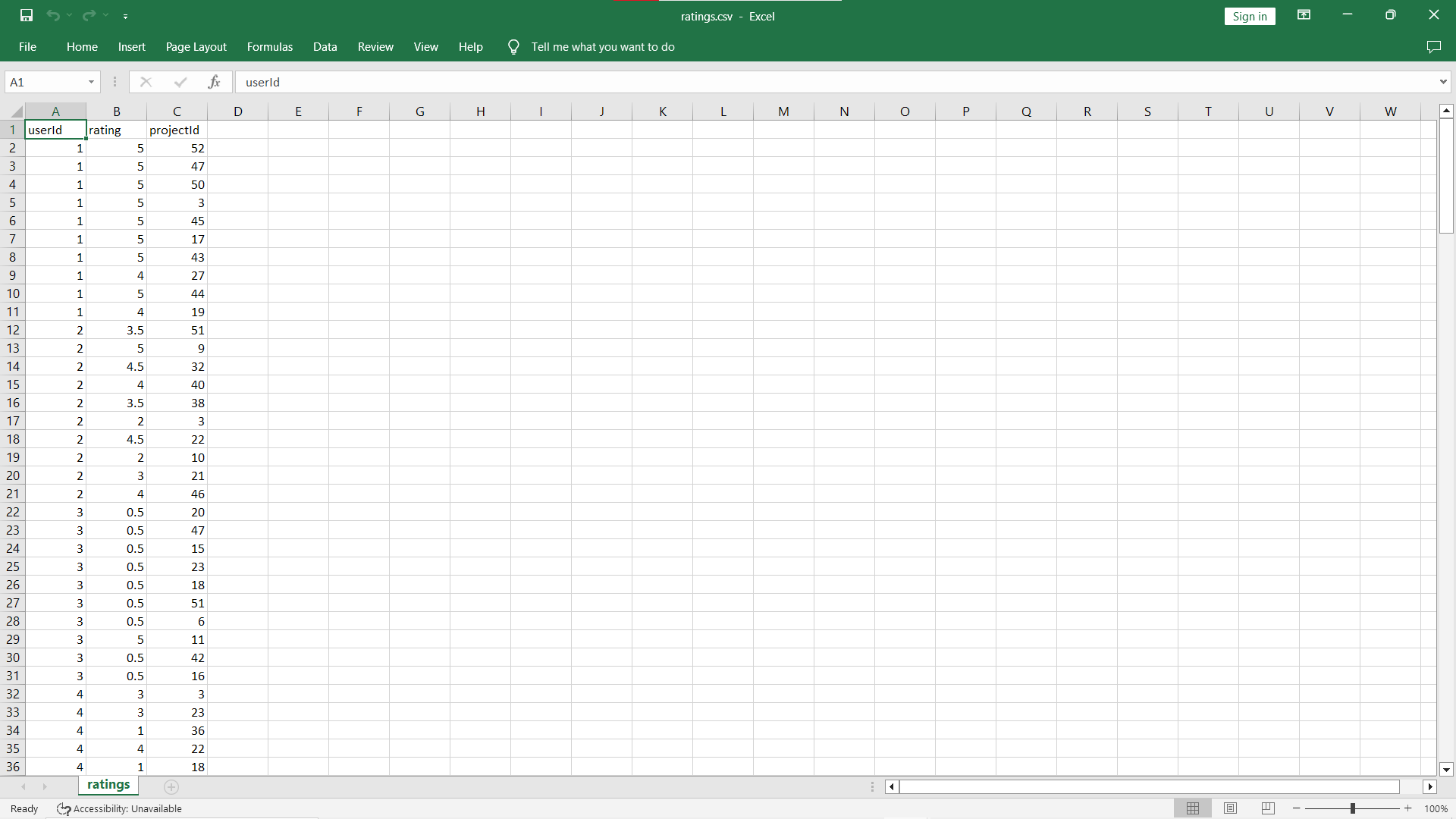


Figure 1.7: Projects Ratings Dataset (at one given time)

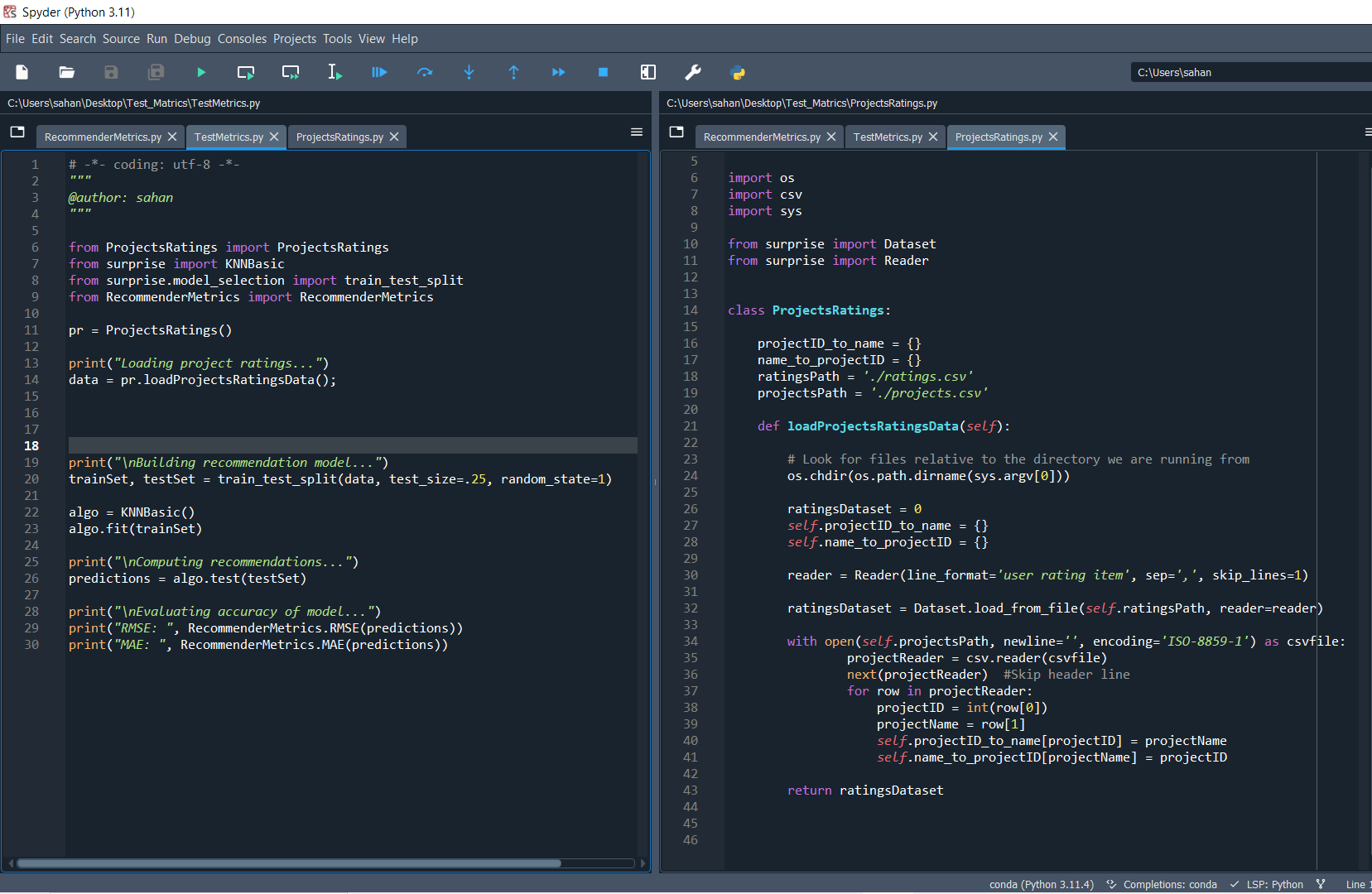
****

Figure 1.8: Python Code Implementation

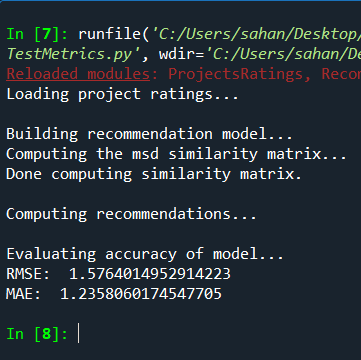
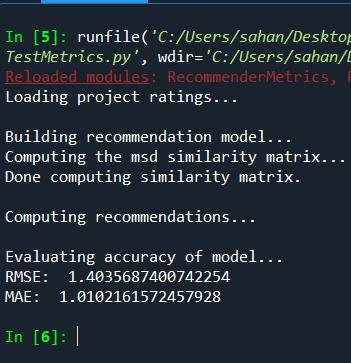
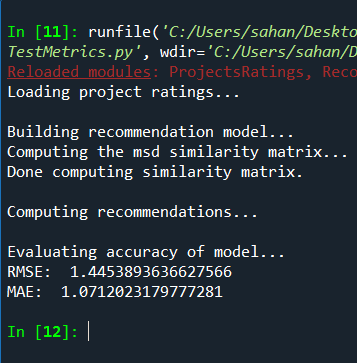
  

Figure 2.1: Dataset 3 Result

Figure 2.0: Dataset 2 Result

Figure 1.9: Dataset 1 Result

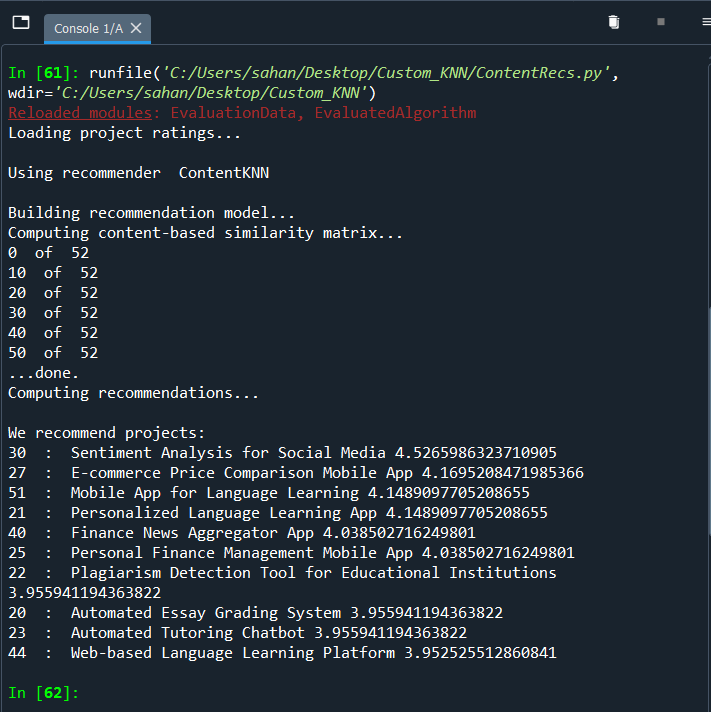
So the above three figures (Figure 1.9, Figure 2.0, Figure 2.1) show the results obtained for Mean Absolute Error and Root Mean Square Error using three different users ratings datasets. So as the results show dataset 2 has given lower MAE, RMSE values. There for this dataset (dataset 2) was taken to build the custom K-Nearest Neighbor algorithm implementation using a similarity calculation method called Cosine Similarity in a multi-dimensional plane and then tested offline. So the Figure 2.2 shows the top 10 recommended projects results along with suggested ratings for the user with the id 2 by the KNN algorithm using Cosine Similarity Calculation method.

Figure 2.2: Recommended Projects for user with id 2

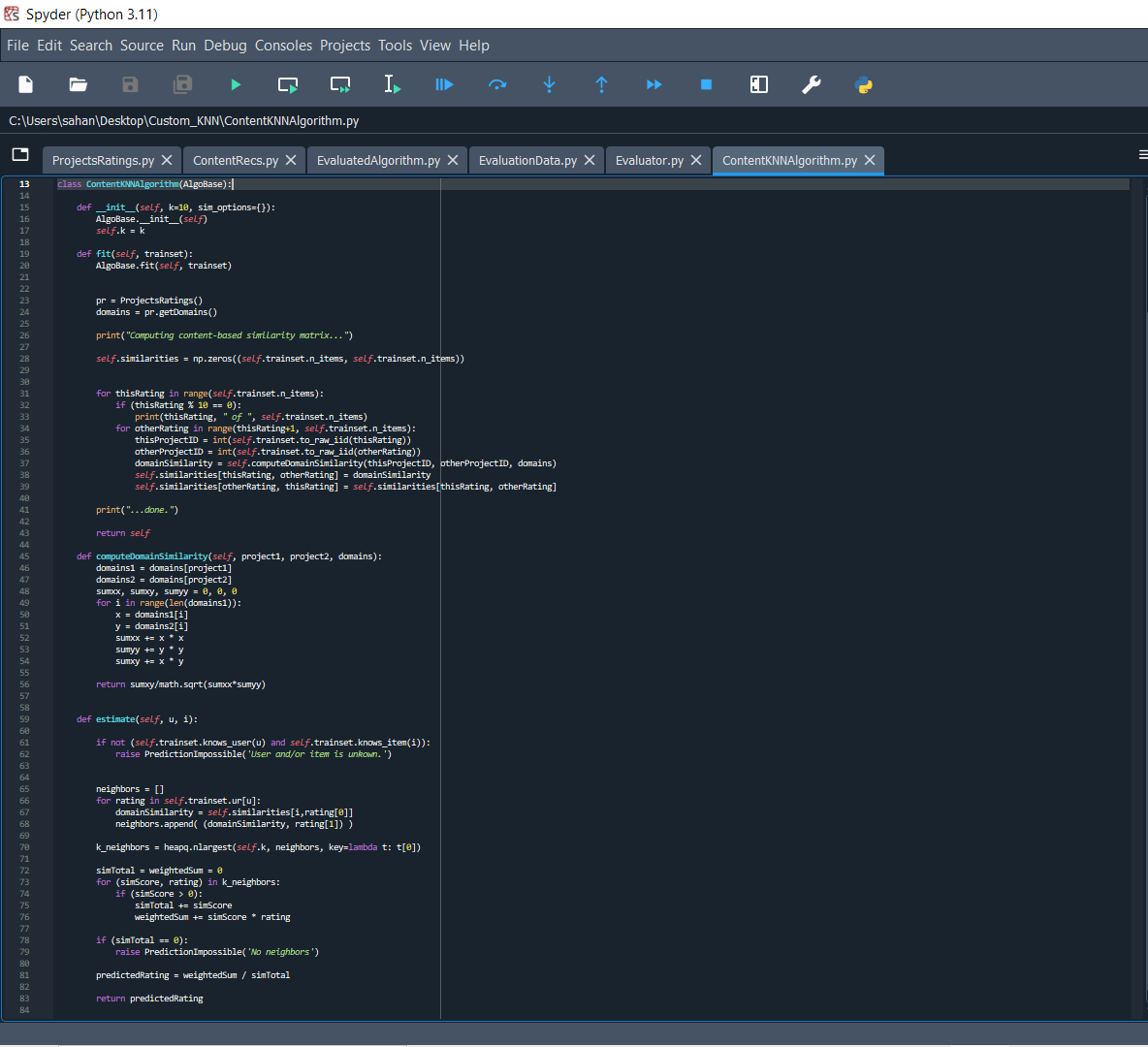
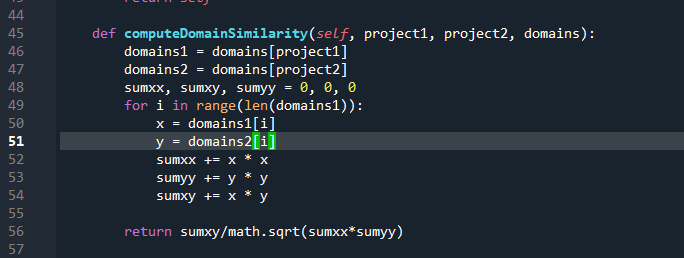
****

Figure 2.3: Custom KNN Algorithm Implementation

Figure 2.4: Cosine Similarity Calculation for Domains

****

So the above two Figures (Figure 2.3 and Figure 2.4) show the implementation of K-Nearest Neighbor algorithm and a method created for calculating Cosine Similarity scores for two projects based on their domain similarities.

# Chapter 06 : Future Work

When developing this application, I have got many experience. So actually I found out some further enhancements which can be implemented to this application. Those improvements can be divided into two sections.

1. Enhancing the real-time socket management further with different architectures.
2. Enhance the content-based recommendation system further.

## 1. Enhancing the real-time socket management further with different architectures.

So when it comes to real-time communication, there are various protocols which we can try out except the web-socket protocol. So some of them are Server-Sent Events (SSE), Web Real-Time Communication (WebRTC), Message Queuing Telemetry Transport (MQTT). So we can try out those different types of protocols and choose what is the most suitable protocol or combination of protocols for this application.

## 2. Enhance the content-based recommendation system further.

So when it comes to building recommendation systems, there are several approaches including some modern technological development based approaches. So we can try out those different approaches in the context of our application to find out which one is most suitable. One of the improvements we can do to this is use something called a collaborative recommendation approach. Which will use a technique called “Matrix Factorization” to generate the results (recommendations). But to implement this we need a large amount of data related to our domain, otherwise this will generate poor results due to lack of data available. Another modern approach which we can use when building such a recommendation system is to use deep learning based models. This is kind of a modern approach with the evolvement of deep neural networks. So another improvement which we can do is to integrate the recommender system with our application, so that we can use and test it online. This will give huge benefits to the application.

# Chapter 07 : Conclusion

The aim of this research project was to develop a platform for users who are working on software projects to achieve their goals. Throughout the journey lots of things have been learnt including implementing a much more secure authentication mechanism, how to use web sockets more efficiently to implement real time communication, as well as how to build a recommendation system using different approaches and techniques. Specially using content-based recommendation approach. Along the journey lots of experience have been gained and achieved lots of things.

# References

* [1] J. Bobadilla, A. Hernando, F. Ortega, and J. Bernal, "A framework for collaborative filtering recommender systems," *Expert Systems with Applications*, vol. 38, no. 12, pp. 14609-14623, November-December 2011. DOI: 10.1016/j.eswa.2011.05.021.
* [2] D. G. Puranik, D. C. Feiock, and J. H. Hill, "Real-Time Monitoring using AJAX and WebSockets," in *2013 20th IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS)*, Scottsdale, AZ, USA, April 22-24, 2013, pp. 141-148. DOI: 10.1109/ECBS.2013.10.
* [3] Q. Liu and X. Sun, "Research of Web Real-Time Communication Based on Web Socket," *Int'l J. of Communications, Network and System Sciences*, vol. 5, no. 12, 2012. DOI: 10.4236/ijcns.2012.512083.
* [4] D. Wang, Y. Liang, D. Xu, X. Feng, and R. Guan, "A content-based recommender system for computer science publications," vol. 152, May 2018. DOI: 10.1016/j.knosys.2018.05.001.