ABSTRACT

It was a long-held belief that eLearning would eventually overtake traditional learning. With the onset of the global pandemic, however, this timeline has been moved up sooner than expected. Thus, it is interesting to determine the differences in the adoption of eLearning practices throughout the years. Our project **'Mentor Labs'** aims to target students to meet professionals or mentors in virtual environment. The main objective of our project is to provide a platform that acts as a bridge between students and professionals for gaining knowledge, sharing ideas, thoughts, discussions and interactions. We provide a platform that offers students digital solutions which enhance the learning experience. Mentor labs is a web-based streaming platform where students can learn via one-on-one consultations or in a group workshop with payment basis.

Keywords: eLearning, WebRTC, e-Learning, Recommendation Algorithm, Jaccard Index

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRYONMS AND ABBREVIATIONS

AES: Advanced Encryption Standard

API: Application Programming Interface

CD: Continuous Deployment

CI: Continuous Integration

DC: Data Cleaning

DF: Data Frame

DTLS: Datagram Transport Layer Security

E2E Testing: End to End Testing

HTTP: Hypertext Transfer Protocol

ICE: Interactivity Connectivity Establishment

KNN: K-Nearest Neighbor

MDN: Mozilla Developers Network

NAT: Network Address Translation

P2P: Peer to Peer

PCM: Pulse Code Modulation

PFS: Perfect Forward Secrecy

RTCP: Real Time Control Protocol

RTP: Real Time Protocol

RTSP: Real Time Streaming Protocol

SCTP: Stream Control Transmission Protocol

SDLC: Software Development Lifecycle

SDP: Session Description Protocol

SRTP: Secure Real-time Transport Protocol

STUN: Session Traversal Utilities for NAT

TCP: Transmission Control Protocol

TURN: Traversal Using Relays around NAT

UDP: User Defined Protocol

WebRTC: Web Real Time Communication

CHAPTER 1: INTRODUCTION

1.1 Introduction

The Project entitled “Mentor Labs” is an application that allows Web based streaming platform with rich features for communication between the industry expert on programming and computer science topics and the students. The users as well as professionals who wish to use this application, can register for the website and get the User ID and password. It also allows the users to chat with other users at same time. It provides the user up to date information about availability and schedule of any professional industry experts. The project will be initially focused on non-payment basis and later we can integrate payment system too.

By using this project users can ask questions to the professionals so that they can gain knowledge about the course they elect to learn. In Mentor`s Lab everything is done remotely so users can connect with professionals from any part of world virtually. Students as well as mentors/teachers will have their own profile catalog. The profile catalog would include Personal information as well as courses of interest or course of expertism of respective user.

As we know proper guidance from the professionals is very important for the students/users to develop their knowledge in the respective Fields. Through this project we try to mitigate those problems so that students can gain the actual knowledge.

1.2 Statement of Problem

The communication between professionals and students might not be convenient at all the times physically. Also, virtually through mails and other calls, Professionals may have busy schedules and the

y may not be available for all the emails and phone calls to be replied, time and again. Also, same queries can be asked by particular students repeatedly which could frustrate professionals to explain them again and again, this would cause another party wasting time and wait. Also, there may be possibility of students or lecturer has totally forgotten about topic to be discussed in the later appointments, since future appointments schedule are uncertain

On other side, the queries of students may not be fully addressed on any platform or they might have problems of understanding them properly. Also, it is not optimal for students living far away from cities to attend physical meetings.

Sometimes things could get messy for lecturer to schedule the consultation session. The lecturer/professional must be extremely cautious when scheduling the consultant session. It is also possible that sometimes the lecturer may forget or schedule multiple consultation session at the same time. If this happens, multiple students will come at the same time or student may end up waiting while lecturer never attends the session.

Therefore, all these above problems can be accommodated through the emerge and use of the platform we wish to create “Mentor Labs”.

1.3 Objectives

This project we wish to develop is web based. The main objective of the project is to connect students with their idol professionals who can provide guidance to them. Following topics will be the key areas behind the development of this project:

i) To create web based online video streaming and screen sharing feature for students and professionals.

ii) To create separate dashboard for mentors and mentees.

iii) To include an appointment system which will be handling all the appointments between the mentors and the mentees effectively.

iv) To create real time chat system to enable students and lecturers to communicate through chat optionally.

v) To create recommendation system to recommend the mentors based on similarities to mentee.

1.4 Scope and Limitations

1.4.1 Scope

The probable scopes of this project are listed below:

- To be a bridge between the students living in remote as well as Urban areas to interact them with the professionals virtually.
- To help Students gain expertise on any particular course.
- To fill the gap between the academics and professionals in the Computer Science field.

1.4.2 Limitations

The probable limitations of this project are listed below:

- Some old browsers may not support our website as per expected.
- Unable to share screen using browser.
- Absence of chatbot for the prompt response.

1.5 Development Methodology

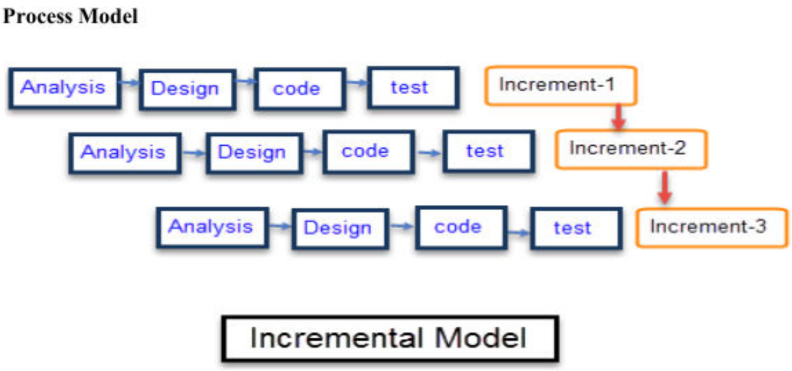


Figure 1. 1 Incremental Model

An incremental model is a process of software development where requirements are broken down into multiple standalone modules of the software development cycle. Each iteration passes through the following phases:

1. Analysis phase
2. Design phase
3. Coding phase
4. Testing phase

1. Analysis Phase:

In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.

2. Design Phase:

In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.

3. Coding Phase:

In the incremental model, Coding of software is done during this stage.

4. Testing Phase:

In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, the various methods are used to test the behavior of each task.

1.6 Report Organization

This report is separated into different chapters which are as follows:

Chapter 1: This chapter includes about short introduction of the project, introduction of the organization, objectives, project scope and limitations in the project.

Chapter 2: The second chapter includes about literature review.

Chapter 3: The third chapter includes requirement analysis, its functional non-functional requirements, software and hardware requirements and tools used for conducting the project. It includes our proposed system, system analysis data collection, its features extraction and training of Machine learning classifiers.

Chapter 4: It includes system design, various pictorial diagrams and software development model used.

Chapter 5: It includes implementation and different testing methodologies done for application.

Chapter 6: This chapter includes the conclusion and future works to be done for the whole project.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

Mentor labs will help to overcome the stated problem which states the difficulties that students have faced in having the desired communication with professionals in the desired domain. For this project we researched on various platforms such as social media and other talks with students. We found that they were not satisfied with the academic professors since their presentations were generalized, not specific in nature. To overcome this, we wished to provide a platform which connects students with the expertise in their preferred domain.

Similarly, we researched about the framework, language and tools which might be suitable to develop our platform. Since, our application is UI heavy we found that React.js library perfect for our use case. React.js is JavaScript library for building user interfaces and it has rich ecosystem of libraries that will be easier to integrate in our project. Similarly, we have planned to use Node.js on backend since we found that one of the most effective tech stacks supporting RTM apps in Node.js. Besides, it's based on JavaScript just like the one used in the web browser, so we can use the same language on both - frontend and backend. [1]

Also, we use some algorithms for recommendation of teachers to the students. The recommendation would be based on the student's interest or other activities.

With WebRTC, we can add real-time communication capabilities to your application that works on top of an open standard. It supports video, voice, and generic data to be sent between peers, allowing developers to build powerful voice- and video-communication solutions. The technology is available on all modern browsers as well as on native clients for all major platforms. The technologies behind WebRTC are implemented as an open web standard and available as regular JavaScript APIs in all major browsers. For native clients, like Android and iOS applications, a library is available that provides the same functionality. The WebRTC project is open-source and supported by Apple, Google, Microsoft and Mozilla, amongst others.

The set of standards that comprise WebRTC makes it possible to share data and perform teleconferencing peer-to-peer, without requiring that the user install plug-ins or any other third-party software. The WebSocket API is an advanced technology that makes it possible to open a two-way interactive communication session between the user's browser and a server. With this API, you can send messages to a server and receive event-driven responses without having to poll the server for a reply.

2.2 Literature Review

We know that there exists a platform like “Hopin” which provides similar kinds of features as ours. Platform like Hopin has many features like:

- It can handle large number of audiences
- It has their own server to store live video.

Metaphor of Hopin is a public hall where the person or institute who needs space can pay rent for a specific time. But we are specific to only the student genre. We take requests form students and make possible arrangements for virtual conferencing so that everyone’s voice is heard. [2]

Recommendation system which is another major feature in our application is done using the Jaccard index calculation. Broadly, there are two approaches to developing a recommendation system. In one approach, the system considers the properties of the content consumed by the individual. For example, if you have watched Matrix trilogy in one day on Netflix, then Netflix understands that you like sci-fi movies and is more likely to recommend other sci-fi movies. In other words, the recommendation is based on the movie genre sci-fi in this case. [6]

In the other approach, the recommendation system considers the preferences of others whose taste is similar to that of yours and recommends movies that they have watched. In contrast to the first approach, the recommendation is made based on the behavior of multiple users, not on the properties of the content that is being consumed. This approach is called collaborative filtering. We have tried to implement this method in our application but with a bit different approach using the Jaccard coefficient calculation.

Jaccard distance is a function of another quantity called Jaccard Similarity. By definition, Jaccard similarity of sets S and T is the ratio of the size of the intersection of S and T to the size of their union. [4]

CHAPTER 3: SYSTEM ANALYSIS

3.1 Feasibility Analysis

Feasibility study is an evaluation and analysis of the potential of the proposed project which is based on extensive investigation and research to give full comfort to the decision makers. A system should have a reasonable cost and should be technically and operationally feasible to be actually implemented. The two criteria to judge feasibility are cost required and value to be attained. The feasibility of “Mentors-Lab” is analyzed under the following. [2]

3.1.1 Economical Feasibility

The application requires UI/UX developer, web engineer, backend engineer, and database designer. Although the skillset requires to build require our project is quite high, we have managed to do all tasks by us. The major benefit of this project is that it will help the mentee and mentor for searching each other. Although we need some of the services like Sentry, Twilio API, Heroku deployment, we have managed to do using the free plan of these services and if the application scales, we can upgrade it on future. There are no such drawbacks of this application based on costs. It is also cheap to operate and maintain. Mentors-Lab do not require manual administration and monitoring.

3.1.2 Operational Feasibility

Mentee or user to use the website and search for the best mentors without any specific training. Any naïve users can operate this website with bare minimum knowledge and no special kind of training is required. The system will operate over internet thus making the user available with the best mentors. Hence this project is operationally feasible.

3.1.3 Technical Feasibility

All the tools and software products required to construct Mentors-lab is easily available in the web. It does not require special environment to execute. It needs a web server and a DBMS to operate. The operation makes use of internet. All these aspects are affordable. The application requires simple user interfaces.

3.2 Requirements Analysis

3.2.1 Functional Requirement

The functional requirements specify the services that the system should provide, how the system should react to particular inputs and how the system should behave in particular situations. The functional requirements of Mentor Labs are as follows:

1. Recommendation Algorithm to recommend mentors to the mentee according their similarities.
2. WebRTC for real-time communication between mentee and mentor.
3. Authentication and authorization of users.
4. Update and delete user information
5. Mentee can apply or request for a mentor.
6. Create a room and join a room.
8. Mentee and mentor dashboards.

3.2.2 Non-Functional Requirement

Non- functional requirements are constraints on the services or functions offered by the system. They include timing constraints, constraints on the developing process and standards. Example of non-functional requirement, “how fast does the website load?” Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non-functional requirements often apply to the system as a whole.

1. The system must be developed or built in a user-friendly and components must be accessible for every people.
2. The system load time must be able as minimum as possible to load website as fast as possible.
3. The system must have fallback page incase of any failure in client and server.

3.2.3 Software and Hardware Requirements

Operating System - Windows, Mac OS, Linux

Browsers - Google, Mozilla, Brave

Internet connectivity

CHAPTER 4: SYSTEM DESIGN

4.1 Design

The object-oriented approach is being used for the system design. We have developed the architecture for the system that shows how different services in the system interact to provide collective functionalities.

4.1.1 Sequence Diagram

A Sequence diagram shows interaction between the external factors and our core system. It represents how objects operate with one another and in what order. It is the object interactions arranged in time sequence. The following sequence diagram depicts the flow of information in Mentors lab.

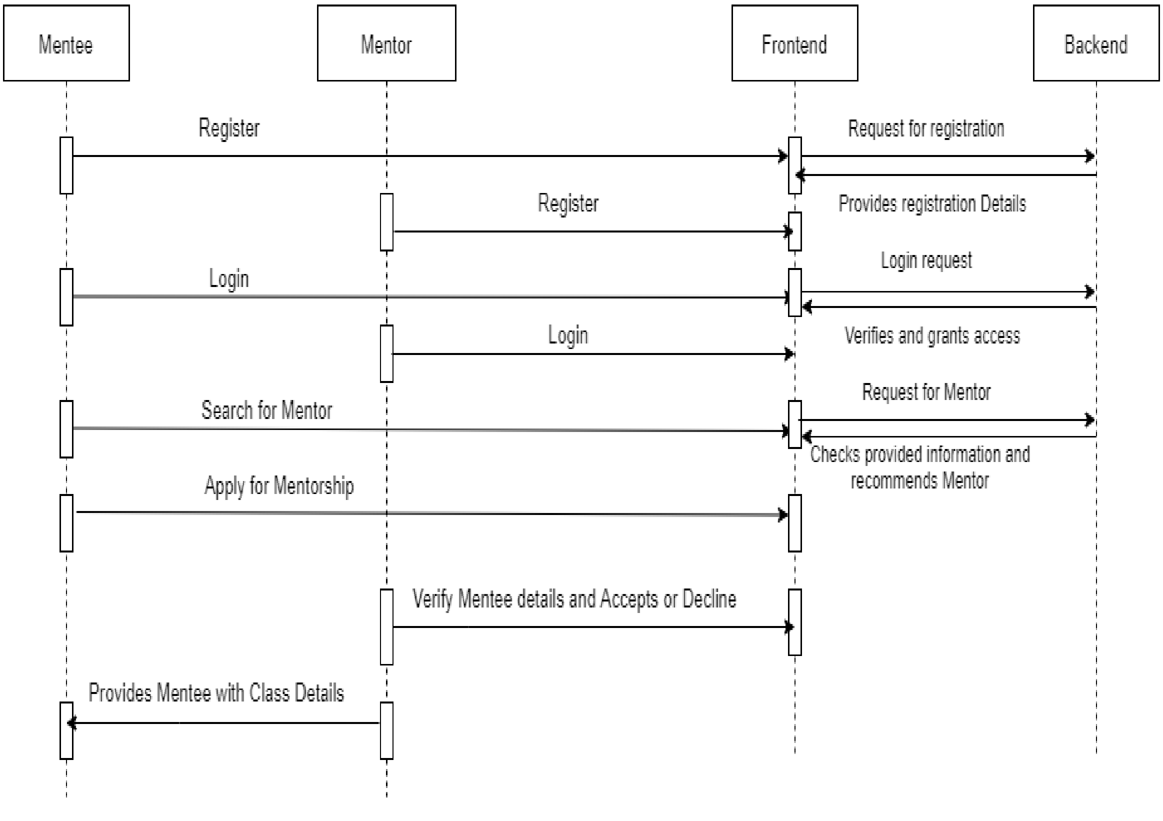

Mentee
Mentor
Frontend
Backend
Register
Request for registration
Register
Provides registration Details
Login request
Login
Login
Verifies and grants access
Request for Mentor
Search for Mentor
Checks provided information and recommends Mentor
Apply for Mentorship
Verify Mentee details and Accepts or Decline
Provides Mentee with Class Details

Figure 4. 1 Sequence Diagram of Mentor Labs

As we can view in the above sequence diagram, here first the Mentee as well as Mentor registers their account in order to login. After the completion of registration, both mentee and mentor can login into their accounts which will be verified in the backend and access will be granted accordingly. Mentee then searches for a mentor and mentors will be recommended to their similarities. Mentee then can apply for mentorship. Mentor then verifies mentee details and accepts or declines the request from mentee. After accepting request mentor provides the details about the session to mentee.

4.1.2 E-R Diagram

The ER-diagram represents the real-world objects called entities and association among these entities. It helps the designer to determine the useful entities of the database, the relationship they hold and the degree of relationship. ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

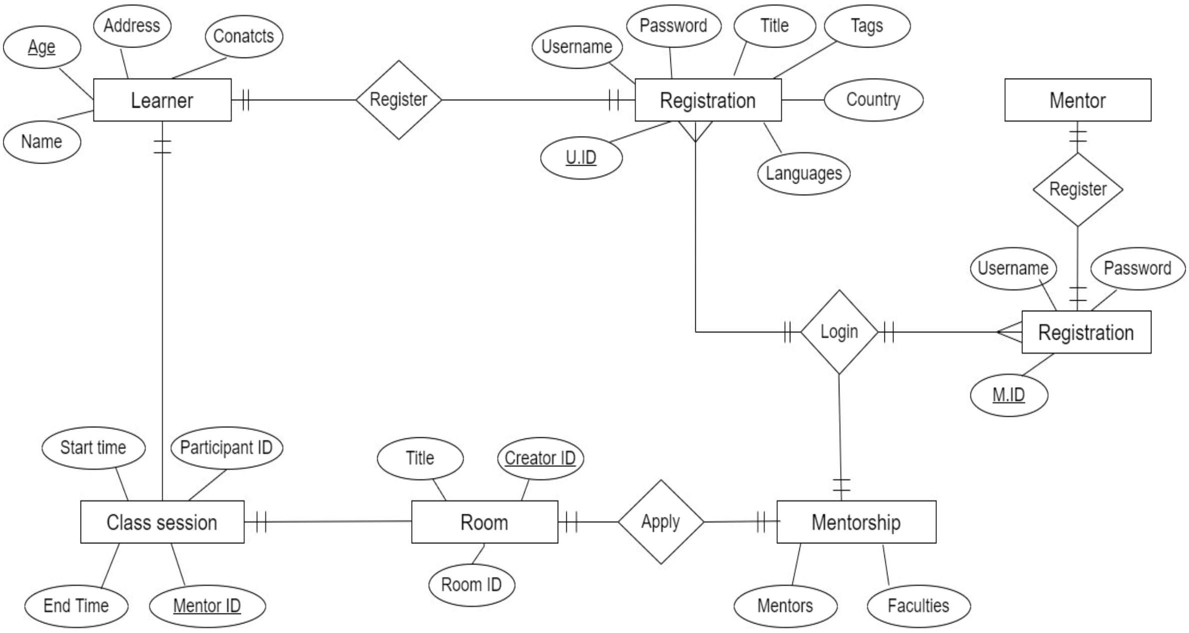


Figure 4. 2 E-R Diagram of Mentor Labs databases

As shown in the above diagram we have entities Learner aka Mentee, Registration, Mentor, Class session, Room, Mentorship. Each of the entities have its own properties. Mentee and Mentor both registers into the application. Similarly, the mentee can book the mentorship session with the mentor as per the recommendation. If the mentor accepts the mentorship request, the he/she can schedule the mentorship along with the room. Then room has properties like room id, title, and the room creator id, which will be joined by mentor and mentee to carry out the mentorship session.

4.1.3 Use Case Diagram:

It shows the interaction between the system and the user in the particular environment. The use case model contains actors and the use cases. The actors are the external entities and the use cases are the functions of the system.

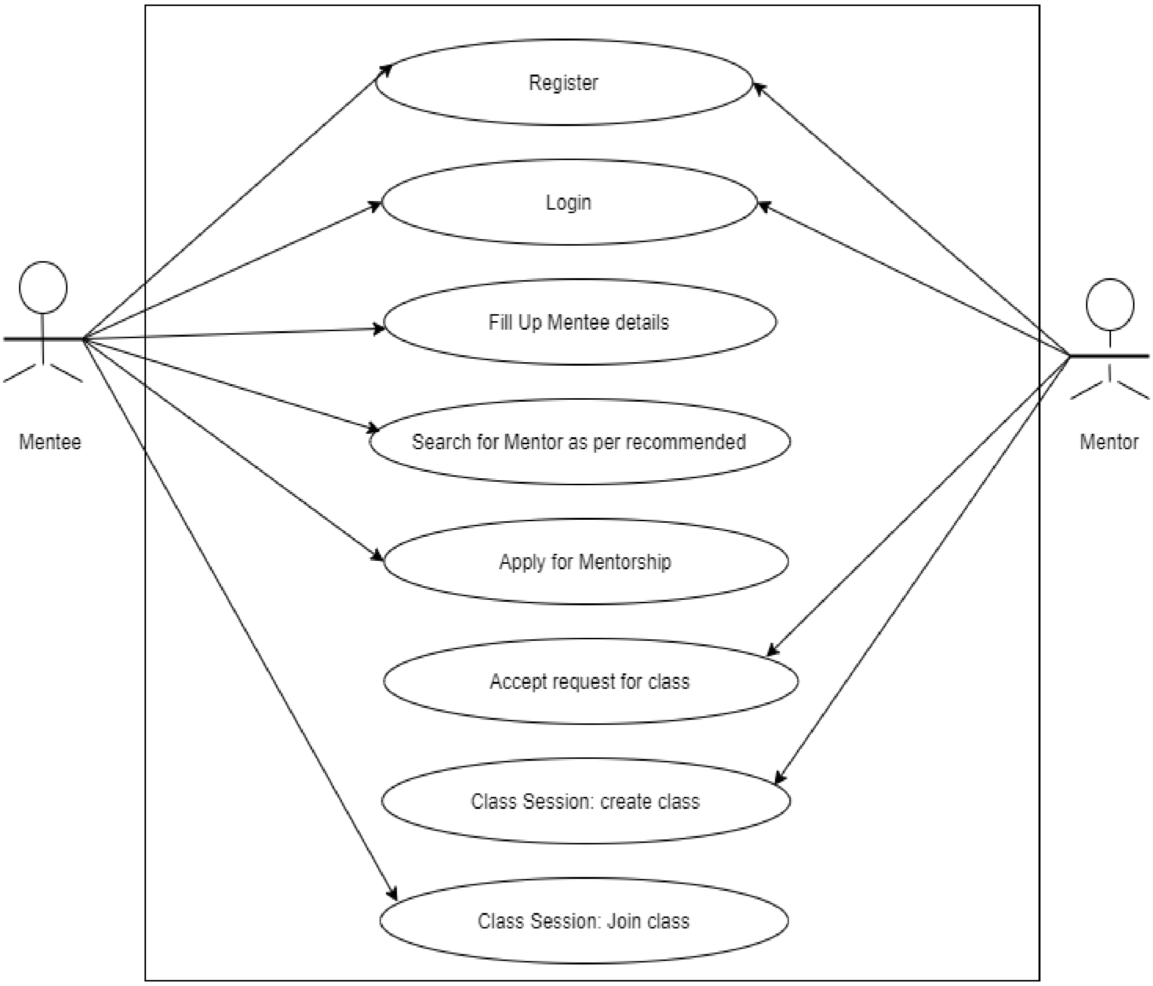


Figure 4. 3 Use Case Diagram between Mentor and Mentees

The above figure depicts the major functions of Mentors lab. There are two major actors- Learner and Mentor. The Learner can perform functions like sign up, sign in, search for Mentor. Similarly, the mentor can perform similar function but in addition it can also accept or decline Learner request. Furthermore, after the acceptance of request by mentor both mentor and learner can join a class session with pre-determined schedule.

We used this model to manage our work and ensure that problem is solved as soon as it arrives and update the project accordingly with increasing knowledge. [3]

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Implementation

System implementation is the process of defining how the information system is built, ensuring that the information system is operational and meets the quality standard. During the research period various methodologies were used for product requirements and data collection. The primary data and information about the project were collected through web scraping and manually doing research on the relevant field. Different algorithms were gathered and analyzed for best performance in our system and the best algorithm was selected for deployment to our website. The website was developed using React as frontend, Express.js as backend and PostgreSQL database. [5]

5.1.1 Tools Used

Frontend: Html, CSS, React, Redux, Ant design, Typescript

Backend: Nodejs, express, Typescript

Data manipulation: TensorFlow.js (Danfo.js)

Unit testing: Jest

Version Control: Git, GitHub

Database: PostgreSQL

5.1.2 Design tools and platform:

i) Figma:

Figma is used to quick prototyping and collaboration while creating design system in the design phase of our application.

ii) Draw.io:

Draw.io is an open-source tool to create flow chart and different diagrams for our application.

iii) Beekeeper Studio:

An open-source GUI tool to easily manipulate the data in PostgreSQL database.

iv) Insomnia:

Insomnia is an open-source tool to test our API without the need of Front-End application.

5.1.2 Implementation Details of Modules

5.1.2.1 Mentors Recommendation Algorithm

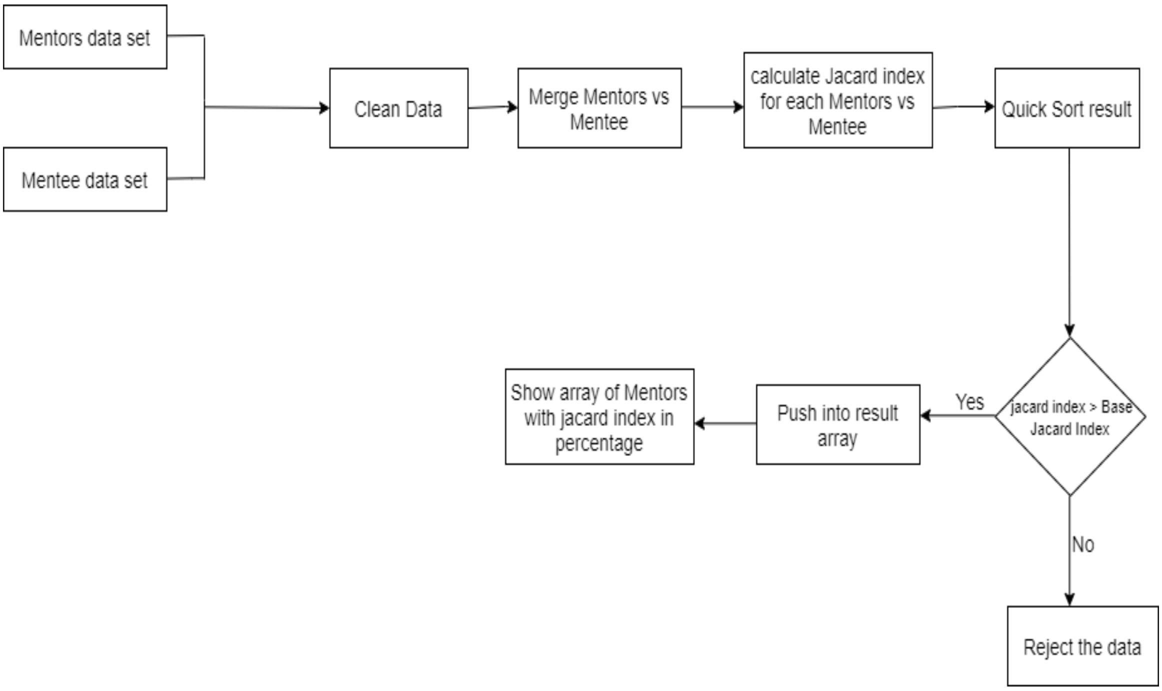

Mentors data set
Mentee data set
Clean Data
Merge Mentors vs Mentee
calculate Jacard index for each Mentors vs Mentee
Quick Sort result
jacard index > Base Jacard Index
Yes
Push into result array
Show array of Mentors with jacard index in percentage
No
Reject the data

Figure 5. 1 Block Diagram of Recommendation System

The above diagram shows the core recommendation algorithm that we have used in this application. We have two data sets mentor data set and mentee data set. These data set might contains extra information that we may not need like social media links, position and so on. So, we need to clean those data, in our case we have used Danfo.js library. After that we need to merge the properties like skillset of mentor, interests of mentee. location and so son. Then after that we need to calculate the Jaccard Index between the mentor and Mentees We have discussed more about the algorithm in the later part of the report. This will result in the Jaccard coefficient between 0 -1. Since, this unusual decimal value may not be good for user experience we have multiply it by 100 to get value in percentage. Also, we have certain base Jaccard index which is 10% in our case. If we have Jaccard index calculated greater than base Jaccard index we will recommend that mentor to mentee else we will ignore that mentor. At the end we have to sort the array of mentors using Quick sort, on descending order of the Jaccard index calculated.

5.1.2.2 Collaborative Filtering Technique

The different aspects of the mentee(me) needs to gets match with the mentors available.

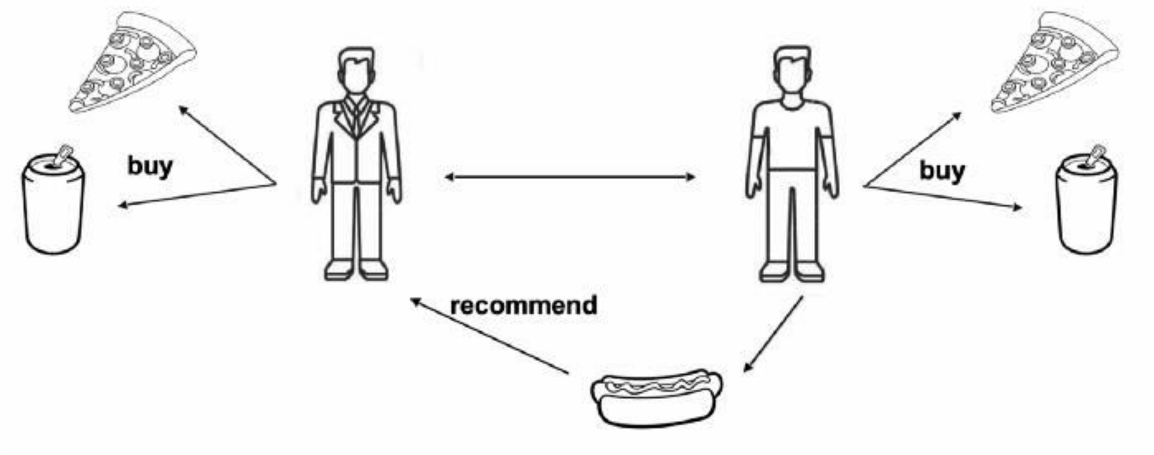


Figure 5. 2 Collaborative Filtering Technique

We need to use collaborative filtering for our purpose. Collaborative filtering models compute their predictions using a dataset of feedback from users to items.

We categorize these models further in how they process this data. Let's see the benefits and drawbacks of each approach.

a) User-User

The most commonly used recommendation algorithm follows the "people like you, like that" logic. It is called a "user-user" algorithm because it recommends an item to a user if similar users liked this item before. The similarity between two users is computed from the number of items they have in common in the dataset.

This algorithm is very efficient when the number of users is way smaller than the number of items. You can think of a medium sized online shop with millions of products. The major drawback is that adding a new user is expensive since it requires updating all similarities between users.

b) Item-Item

The "item-item" algorithm uses the same approach but reverses the view between users and items. It follows the logic "if you like this, you might also like that". It recommends items that are similar to the ones you previously liked. As before the similarity between two items is computed using the number of users they have in common in the dataset.

This algorithm is best when the number of items is way smaller than the number of users, such as large-scale online shops. It is well suited if your items don't change too much, since you can pre-compute the full table of item-item similarities and then serve recommendations in real-time. Updating this table for adding a new item is unfortunately hard. [6]

c) User-Item

There are multiple forms of "user-item" recommendation algorithms which combine both approaches to generate recommendations. In our case we can map its main principle as "If the mentor has similar skills or spoke similar languages or from the same country then you may also prefer that mentor".

Once the users and their preferences are precomputed recommendation can be served at real time. There are many approaches to calculate this like matrix factorization method, Jaccard index calculation.

Matrix factorization methods are too verbose and don't serve well to our purpose. Since, we have to calculate the similarity between the two user preferences but recommend the whole user at the end Jaccard Similarity algorithm serves our purpose well. [4]

5.1.2.3 Quick Sort

We can use Quick sort instead of JavaScript default sort as it will be different on different browsers and Node.js environments. For practical use, ease of implementation might be sacrificed for the sake of efficiency. On a theoretical basis, we can determine the number of element comparisons and swaps to compare performance. Additionally, actual running time will be influenced by other factors, such as caching performance and branch mispredictions.

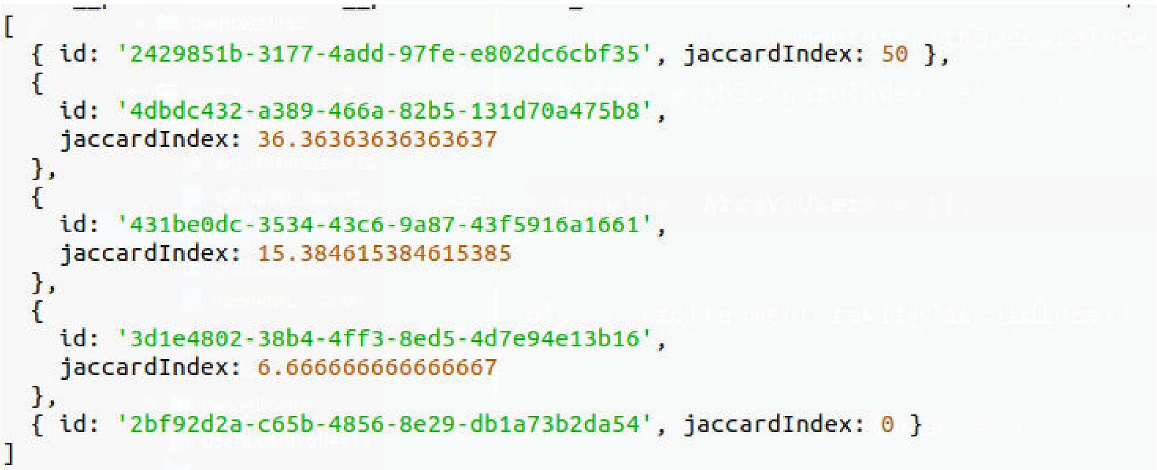
Hoare Partition Scheme Hoare uses two indices that start at the ends of the array being partitioned, then move toward each other until they detect an inversion: a pair of elements, one greater than the pivot, one smaller, in the wrong order relative to each other. The inverted elements are then swapped. When the indices meet, the algorithm stops and returns the final index.

Hoare's scheme is more efficient than Lomuto's partition scheme because it does three times fewer swaps on average, and it creates efficient partitions even when all values are equal. But like Lomuto's partition scheme, Hoare partitioning also causes Quicksort to degrade to O(n2) when the input array is already sorted; it also doesn't produce a stable sort.

Note that in this scheme, the pivot's final location is not necessarily at the index that was returned, and the next two segments that the main algorithm recurs on are [low…pivot] and [pivot+1…high] as opposed to [low…pivot-1] and [pivot+1…high] as in Lomuto's scheme.

Pseudocode:

```
partition(arr[], lo, hi)
  pivot = arr[lo]
  i = lo - 1  // Initialize left index
  j = hi + 1  // Initialize right index
  // Find a value in left side greater
  // than pivot
  do
    i = i + 1
  while arr[i]  pivot
  if i >= j then
    return j
  swap arr[i] with arr[j]
```



```
[
  { id: '2429851b-3177-4add-97fe-e802dc6cbf35', jaccardIndex: 50 },
  {
    id: '4dbdc432-a389-466a-82b5-131d70a475b8',
    jaccardIndex: 36.36363636363637
  },
  {
    id: '431be0dc-3534-43c6-9a87-43f5916a1661',
    jaccardIndex: 15.384615384615385
  },
  {
    id: '3d1e4802-38b4-4ff3-8ed5-4d7e94e13b16',
    jaccardIndex: 6.666666666666667
  },
  { id: '2bf92d2a-c65b-4856-8e29-db1a73b2da54', jaccardIndex: 0 }
]
```

Figure 5. 3 Sorted Mentors according to Jaccard Index

5.1.2.4 Quicksort algorithm using Hoare's partitioning scheme

Hoare partition is an algorithm that is used to partition an array about a pivot. All elements smaller than the pivot are on its left (in any order) and all elements greater than the pivot are on it's right (in any order). Hoare uses two indices that start at the ends of the array being partitioned, then move toward each other, until they detect an inversion: a pair of elements, one greater than or equal to the pivot, one lesser or equal, that are in the wrong order relative to each other. The inverted elements are then swapped. When the indices meet, the algorithm stops and returns the final index.

The indices i and j run towards each other until they cross, which always happens at pivot. This effectively divides the array into two parts: A left part which is scanned by i and a right part scanned by j. Now, a swap is done exactly for every pair of "misplaced" elements, i.e., a large element (larger than pivot, thus belonging in the right partition) which is currently located in the left part and a small element located in the right part.

Algorithm of Hoare Partition

While partitioning arrays using Hoare Partition scheme, we make an assumption:" The pivot element is always the first element of the array."

Pseudocode:

```
fun quicksort (input: T [], low : int, high : int)
   if (low < high)
      p: = partition (input, low, high)
      quicksort (input, low, p) // Note that this is different than when using Lomuto
      quicksort (input, p + 1, high)
fun partition (input: T [], low: int, high: int): int
```

```
pivot Point: = floor ((high + low) / 2)
pivot: = input [pivot Point]
high++
low--
loop while True
    low++
    loop while (input[low] < pivot)
        high--
    loop while (input[high] > pivot)
    if (low >= high)
        return high
    swap(input[low], input[high])
```

5.1.2.5 Jaccard Similarity Algorithm:

Jaccard Similarity is a common proximity measurement used to compute the similarity between two objects, such as two text documents. The Jaccard similarity is calculated by dividing the number of observations in both sets by the number of observations in either set. In other words, the Jaccard similarity can be computed as the size of the intersection divided by the size of the union of two sets. This can be written in set notation using intersection (A∩B) and unions (A∪B) of two sets:

J (A, B) =|A∩B| / |A∪B|

where |A∩B| gives the number of members shared between both sets and |A∪B| gives the total number of members in both sets (shared and unshared). The Jaccard Similarity will be 0 if the two sets don't share any values and 1 if the two sets are identical.

Jaccard Similarity Algorithm works as:

me = "French Hindi Nepali AI business database networking England"

mentor = "Nepali database networking England compiler"

Then we get the set of unique words for each document.

me = {'French', 'Hindi', 'Nepali', 'AI', 'business', 'database', 'networking', 'England'}

mentor = {'Nepali', 'database', 'networking', 'England', 'compiler'}

Now, we will calculate the intersection and union of these two sets of words and measure the Jaccard Similarity between doc_1 and doc_2. [7]

$$J(\text{Mentee ,Mentor}) = \frac{\{\text{'French','Hindi','Nepali','AI','Business',',Database,'Networking','England'}\} \cap \{\text{'Nepali', 'Database', Networking, "Compiler'}\}}{\{\text{'French','Hindi','Nepali','AI','Business',',Database,'Networking','England'}\} \cup \{\text{'Nepali', 'Database, 'Networking, 'Compiler'}\}}$$

$$= \frac{\{\text{'Nepali', 'Databse', 'Networking', 'England'}\}}{\{\text{'French','Hindi','Nepali','AI','Business',',Database,'Networking','England', 'Compiler}\}}$$

$$= \frac{4}{9}$$

$$= 0.444$$

Here, 0.44 is the jaccard coefficient generated after calculating the similarities between mentor and mentee. The similarity between the mentor and mentee can be represented in the Venn diagram as shown below for more clarity. The intersection represents the common skills and interests between those two, which is the main factory for deciding the jaccard coefficient.

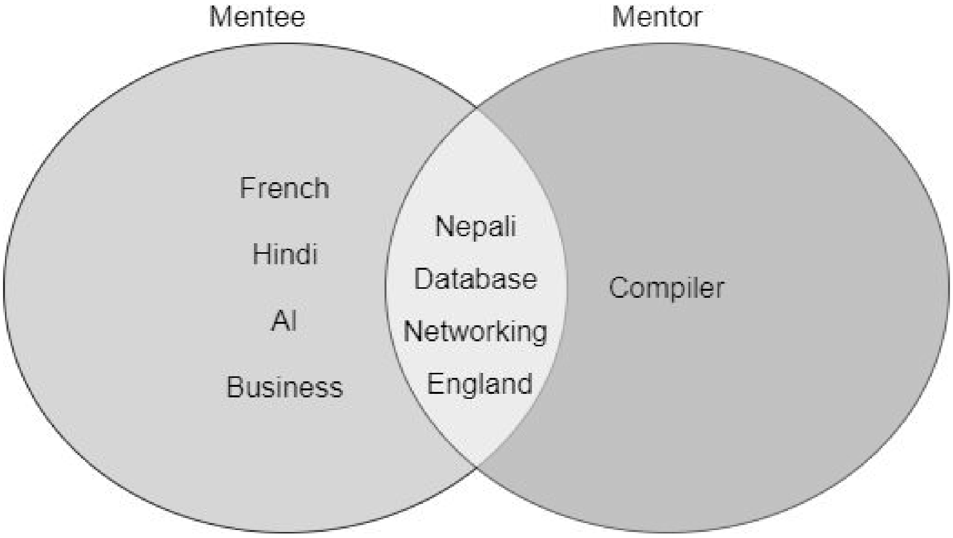


Figure 5. 4 Set representation of common skills

<u>Pseudo Code:</u>

```
Jaccard Similarity(doc1, doc2):
    # List the unique words in a document
    words_doc1 = set(doc1.lower().split())
```

```
words_doc2 = set(doc2.lower().split())
# Find the intersection of words list of doc1 & doc2
intersection = words_doc1.intersection(words_doc2)
# Find the union of words list of doc1 & doc2
union = words_doc1.union(words_doc2)
# Calculate Jaccard similarity score
# Using length of intersection set divided by length of union set
return float (Len(intersection)) / Len(union)
```

Now we have to map the interests of the mentee(me) with the skills of the mentors available one to one. For that purpose, we can take help of K-NN algorithm on how it works and map

Jaccard index-based similarity algorithm is derived from the K-Nearest Neighbor (KNN) algorithm. [4] The K-NN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories.

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

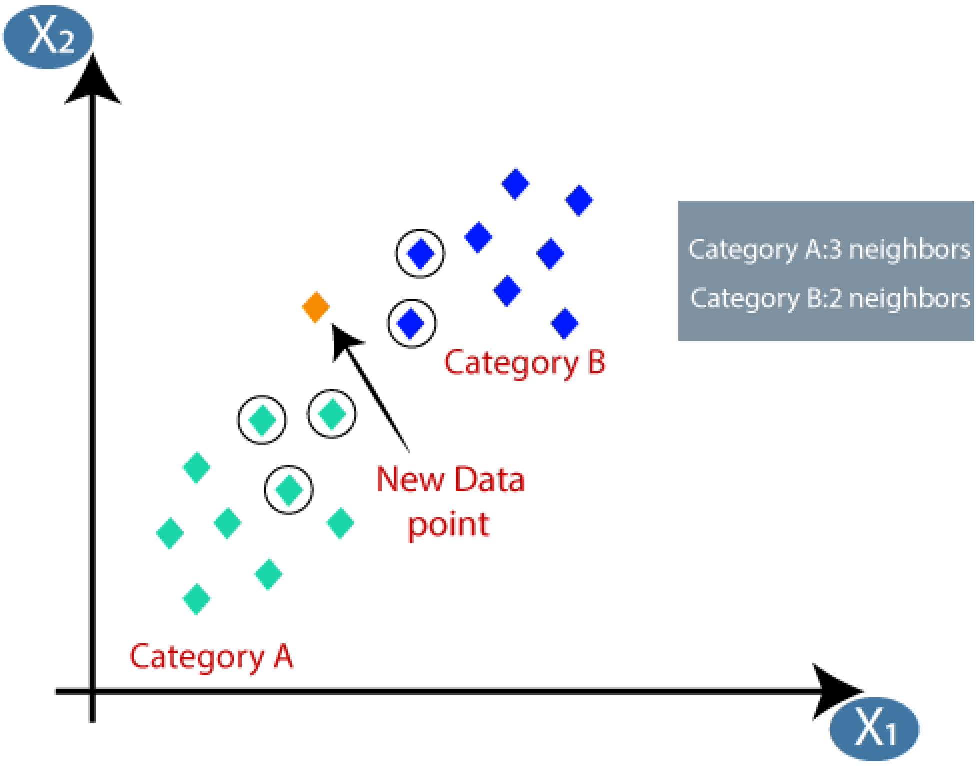


Figure 5. 5 KNN Classification between two data points

By calculating the Euclidean distance, we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B., hence this new data point must belong to category A.

For our case we have only two sets i.e., set of mentor properties and set of mentee properties. For that we have to make the two sets and use the Jaccard index for each mentor. So, it would form a Venn diagram revolving around the mentee (me).

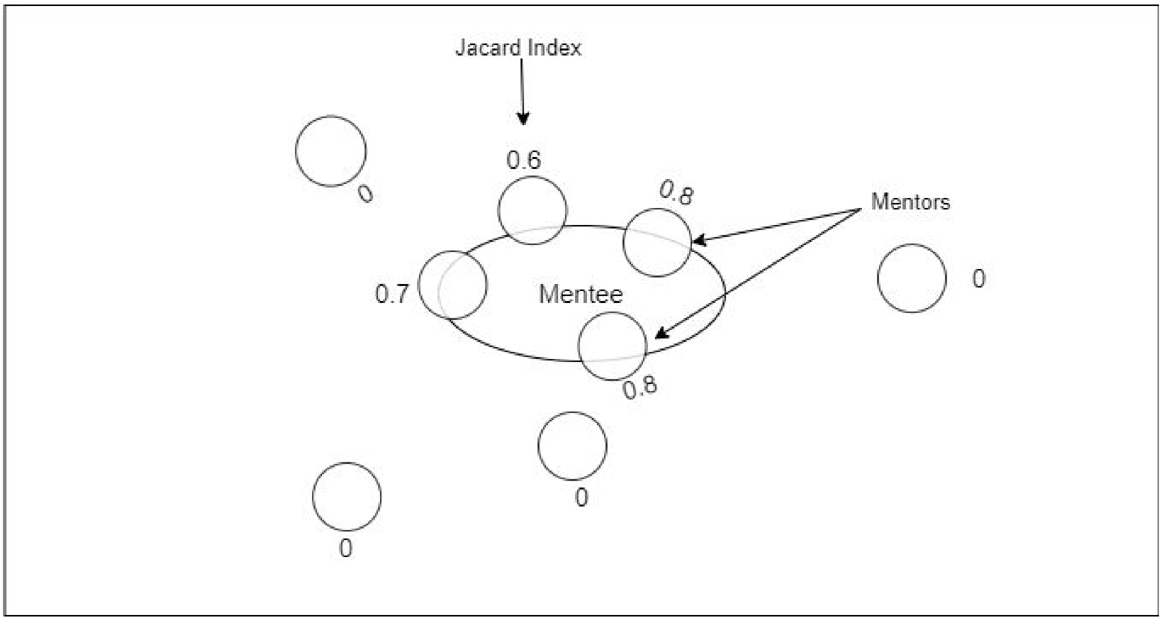
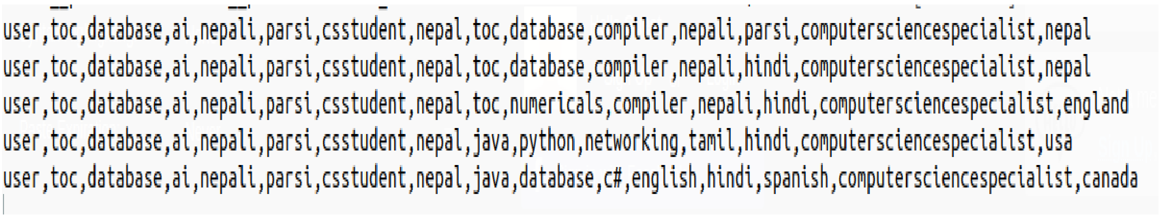


Figure 5. 6 Venn Diagram of Recommendation based on Jaccard Index

As shown in the figure above there are some sets overlapping each other, some have high overlap due to intersection is high, some are far away from mentee with 0 overlapping. It's due to the Jaccard-index. The more the Jaccard-index it indicates that more properties are common between mentor and mentee and if there are less similarities between the mentor and mentee, the less with the Jaccard index. If there are no similarities between the two, then the Jaccard index will be zero. [8]

For calculating the Jaccard Index first we first concatenate all the values and merge it as a string.

Now according to the Jaccard Similarity Coefficient we first prepare the set from these data, so that duplicate values get removed.



```
user,toc,database,ai,nepali,parsi,csstudent,nepal,toc,database,compiler,nepali,parsi,computersciencespecialist,nepal
user,toc,database,ai,nepali,parsi,csstudent,nepal,toc,database,compiler,nepali,hindi,computersciencespecialist,nepal
user,toc,database,ai,nepali,parsi,csstudent,nepal,toc,numericals,compiler,nepali,hindi,computersciencespecialist,england
user,toc,database,ai,nepali,parsi,csstudent,nepal,java,python,networking,tamil,hindi,computersciencespecialist,usa
user,toc,database,ai,nepali,parsi,csstudent,nepal,java,database,c#,english,hindi,spanish,computersciencespecialist,canada
```

Figure 5. 7 Concatenation string based on recommendation

Since we have only one value so we have to calculate the Jaccard Index using the following formula:

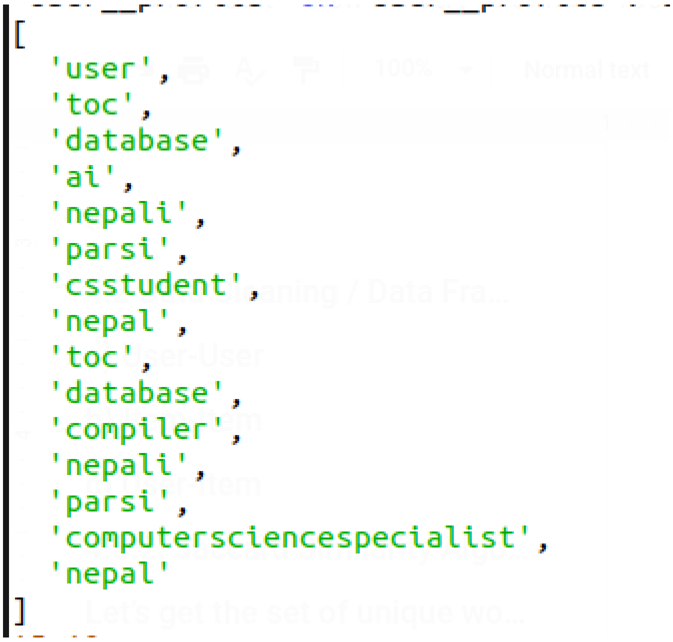
```
function calculateJacardIndex(text) {

  arr = text. Split(',');
  set Data = [...new Set(arr)];
 return ((originalArrayLength - setData.length) / setData.length) * 100;
}
```

Let’s take an example of first concatenated string:

“user,toc,database,ai,nepali,parsi,csstudent,nepal,toc,database,compiler,nepali,parsi, computer science specialist, Nepal”

First, we split it into the array. Then we have an array of tags.



```
[
  'user',
  'toc',
  'database',
  'ai',
  'nepali',
  'parsi',
  'csstudent',
  'nepal',
  'toc',
  'database',
  'compiler',
  'nepali',
  'parsi',
  'computersciencespecialist',
  'nepal'
]
```

Figure 5. 8 Array based on similarities of Mentors and Mentee

Now, we convert the above array into a Set data structure so that duplicate data gets removed. Then, we calculate the original array length and the set data length. If we differentiate the above two with the original array length , it will give the similarity index which is also called the Jaccard Similarity Coefficient Index.

Original_array_length = 15

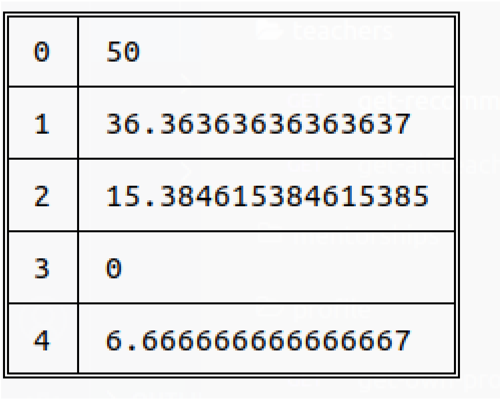
Set length = 10

Jaccard Index = (Original_array_length - Set length) / Set length

= (15-10) / 10

 = 0.5

Map each Jaccard index in the data frame which will result as follows :



0	50
1	36.36363636363637
2	15.384615384615385
3	0
4	6.666666666666667

Figure 5. 9 Data Frame of Mentors id with Jaccard Index in Percentage

Finally, the result of all the mentors will be calculated. But the value may be in random order, we need a sorting algorithm to sort those data in descending order. So, we try to use the quick sort to solve the problem.

5.1.2.6 Filtering Mentors

Since we don't want to display all mentors, we filter by setting up BASE JACCARD INDEX, and display those mentors whose Jaccard index is greater than the base index. For our application, we set it to value 10. In the above example first three mentors will be recommended. [5]

5.1.2.7 WebRTC

Introduction

Web Real-Time Communication (WebRTC) is both an open-source project and specification that enables real-time media communications like voice, video and data transfer natively between browsers and devices. The set of standards that comprise WebRTC makes it possible to share data and perform teleconferencing peer-to-peer, without requiring that the user install plug-ins or any other third-party software. WebRTC serves multiple purposes; together with the Media Capture and Streams API, they provide powerful multimedia capabilities to the Web, including support for audio and video conferencing, file exchange, screen sharing, identity management, and interfacing with legacy telephone systems including support for sending DTMF (touch-tone dialing) signals.

In the absence of Web RTC, devices need to be connected through an intermediary server. It means that both devices need to have the same plugin-in or app installed for communication to work. The one sends the communication to the server, and the server sends it to the other device. [2]

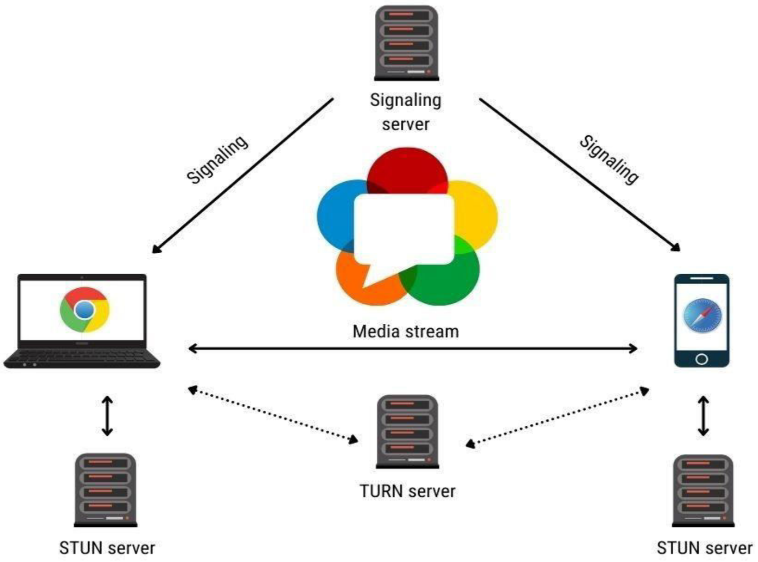


Figure 5. 10 Overall architecture of WebRTC in our application

An example of this can be found in Skype – it is a great video communication tool but only works when you have Skype on every device involved. In an environment where customer service is involved, this can be annoying. The WebRTC is here to save the day! With Web-RTC, devices communicate directly with each other, eliminating the server.

Devices can communicate directly without downloading anything or installing any plugins via the web browser, which automatically embeds the capability. The Web RTC protocol supports audio, video, and data transmission over the internet and can be used to make web calls, video chats, online messaging, and screen sharing within web pages and mobile applications.

5.1.2.8 WebRTC APIs

There are 3 primary components of the WebRTC API and each plays a unique role in WebRTC specification:

Media Stream:

The Media Stream API provides a way to access device cameras and microphones using JavaScript. It controls where multimedia stream data is consumed and provides some control over the devices that produce the media. It also exposes information about devices able to capture and render media. [9]

RTC Peer Connection:

The Peer Connection is the core of the WebRTC standard. It provides a way for participants to create direct connections with their peers without the need for an intermediary server (beyond signaling). Each participant takes the media acquired from the media stream API and plugs it into the peer connection to create an audio or video feed. The Peer Connection API has a lot going on behind the scenes. It handles SDP negotiation, codec implementations, NAT Traversal, packet loss, bandwidth management, and media transfer.

RTC Data Channel:

The RTC Data Channel API was set up to allow bi-directional data transfer of any type of data - media or otherwise - directly between peers. It was designed to mimic the WebSocket API, but rather than relying on a TCP connection which although reliable is high in latency and prone to bottlenecks, data channels use UDP-based streams with the configurability of the Stream Control Transmission Protocol (SCTP) protocol. This design allows the best of both worlds: reliable delivery like in TCP but with reduced congestion on the network like in UDP. [7]

5.2.5.2 Establishing Connections in WebRTC (Working Mechanism)

Before a peer-to-peer video call can begin, a connection between the two clients' needs to be established. This is accomplished through signaling. Signaling falls outside of the realm of the WebRTC specification but is the vital first step in establishing an audio/ video connection.

Signaling:

Signaling allows two endpoints (senders, receivers, or both) to exchange metadata to coordinate communication in order to set up a call. For example, before two endpoints can start a video call, one side has to call the other, and the called side has to respond. This call-and-response message flow (also known as offer-answer message flow) contains critical details about the streaming that will take place - the number and types of streams, how the media will be encoded, etc. - and is often formatted using the Session Description Protocol (SDP), a standard format used by many real-world systems, including VoIP and WebRTC.

This is needed for two reasons:

i) The peers do not know each other's capabilities.

ii) The peers do not know each other’s network addresses.

We will use Client-A and Client-B as examples to explain WebRTC's components below. [8]

SDP (Session Description Protocol):

SDP is a simple string-based protocol and it is to share supported codecs between browsers. When starting the signaling process, an offer is created by the user initiating the call. This offer includes a session description, in SDP format, and needs to be delivered to the receiving user, which we'll call the callee. The callee responds to the offer with an answer message, also containing an SDP description.

It does not deliver the media data but is used for negotiation between peers of various audio and video codecs, network topologies, and other device information. Simply put we need a string-based profile with all the information about the user's device. This is where SDP comes in.

Client-A creates its SDP (called offer) and saves it as local SDP then shares it with Client-B.

Client-B receives the SDP of Client-A and saves it as a remote SDP.

Client-B creates its SDP (called answer) and saves it as a local SDP then shares it with Client-A.

Client-A receives the SDP of Client-B and saves it as a remote SDP.

The signaling server is responsible for these SDP transfers between peers.

Let assume Client-A may support H264, VP8, and VP9 codecs for video, Opus, and PCM codecs for audio. Client-B may support only H264 for video and only Opus codec for audio. For this case, Client-A and Client-B will use H264 and Opus for codecs. If there are no common codecs between peers, peer-to-peer communication cannot be established.

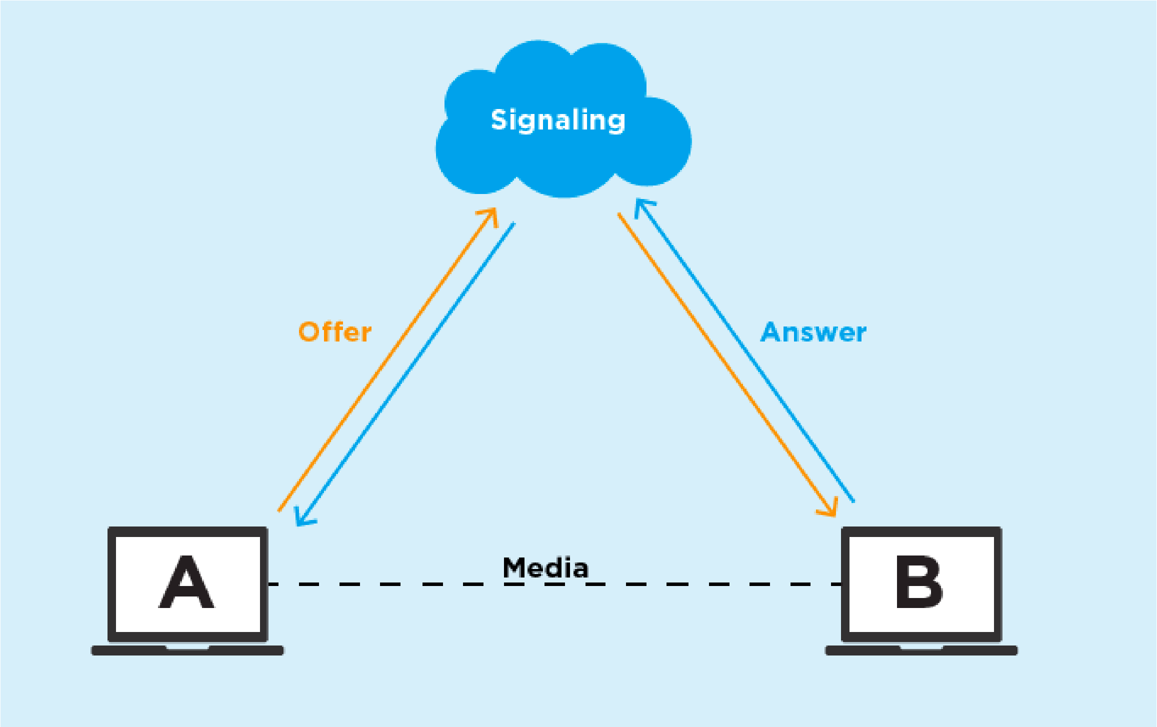


Figure 5. 11 Peer with signaling servers

5.1.2.9 NAT Traversal - ICE, TURN and STUN

Once the initial signaling for a streaming connection has taken place, the two endpoints need to begin the process of NAT (Network Address Translation) traversal. When NAT assigns a public address to a computer inside a private network it can cause difficulties for setting up a real-time video connection. NAT Traversal is a method for getting around the issues associated with IP address translation.

In a WebRTC-enabled video call, unless the two endpoints are on the same local network, there will be one or more intermediary network devices (routers/gateways) between the two. There are three key specifications that are used in WebRTC to overcome these hurdles:

i) Interactive Connectivity Establishment (ICE)

ICE is used to find all the ways for two computers to talk to each other. It has two main roles, gathering candidates and checking connectivity. It guarantees that if there is a path for two clients to communicate, it will find it and ensure it is the most efficient. It makes use of two protocols - STUN and TURN.

ii) Session Traversal Utilities for NAT (STUN)

STUN stands for Session Traversal Utilities for NAT and is a lightweight and simple method for NAT Traversal. STUN Server is responsible to get all the addresses of a machine. For example, our computers generally have one local address in the 192.168.0.0 network and there is a second address we see when we connect to www.whatismyip.com, this IP address is actually the Public IP address of our Internet Gateway (modem, router, etc.) so let’s define STUN server; STUN servers let peers know theirs Public and Local IP addresses. STUN allows WebRTC clients to find out their own public IP address by making a request to a STUN server. Btw, Google provides a free STUN server (stun.l.google.com:19302).

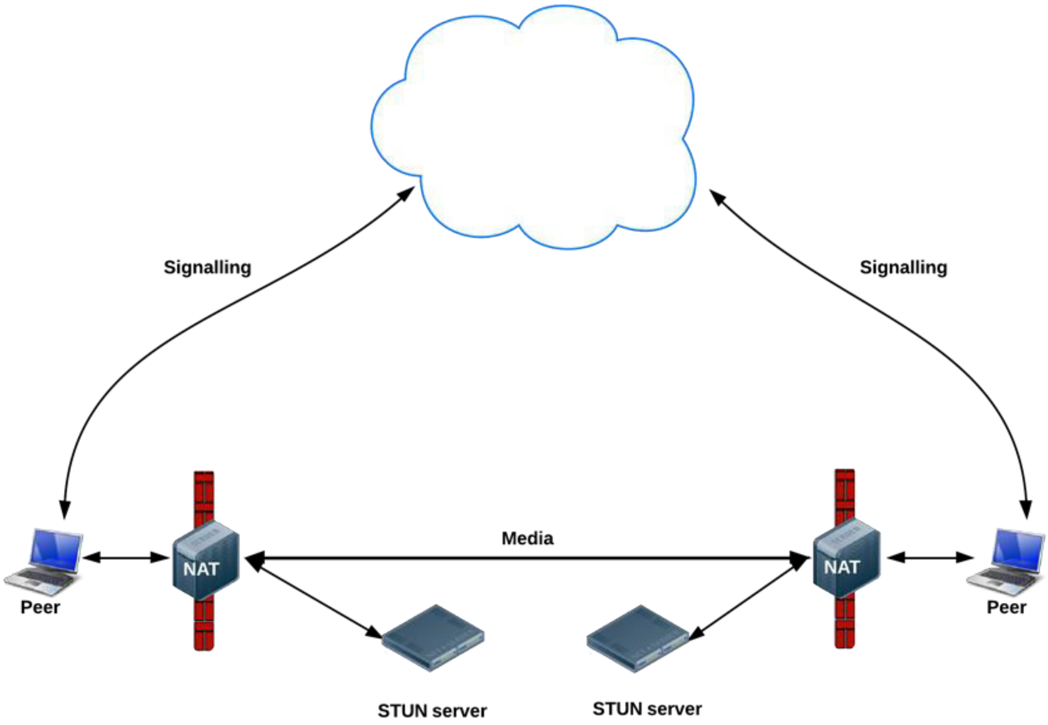


Figure 5. 12 Direct Connection between peers

iii) Traversal Using Relays around NAT (TURN)

TURN (Traversal Using Relays around NAT) is a protocol that assists in the traversal of network address translators (NAT) or firewalls for WebRTC applications. TURN Server allows clients to send and receive data through an intermediary server. The TURN protocol is the extension to STUN. Sometimes, addresses got from the STUN server cannot be used to establish peer to peer connections between peers because of NAT/Firewall. In this case, data relays over TURN Server. The TURN server assists in the NAT traversal by helping the endpoints learn about the routers on their local networks, as well as blindly relaying data for one of the endpoints where a direct connection is not possible due to firewall restrictions.

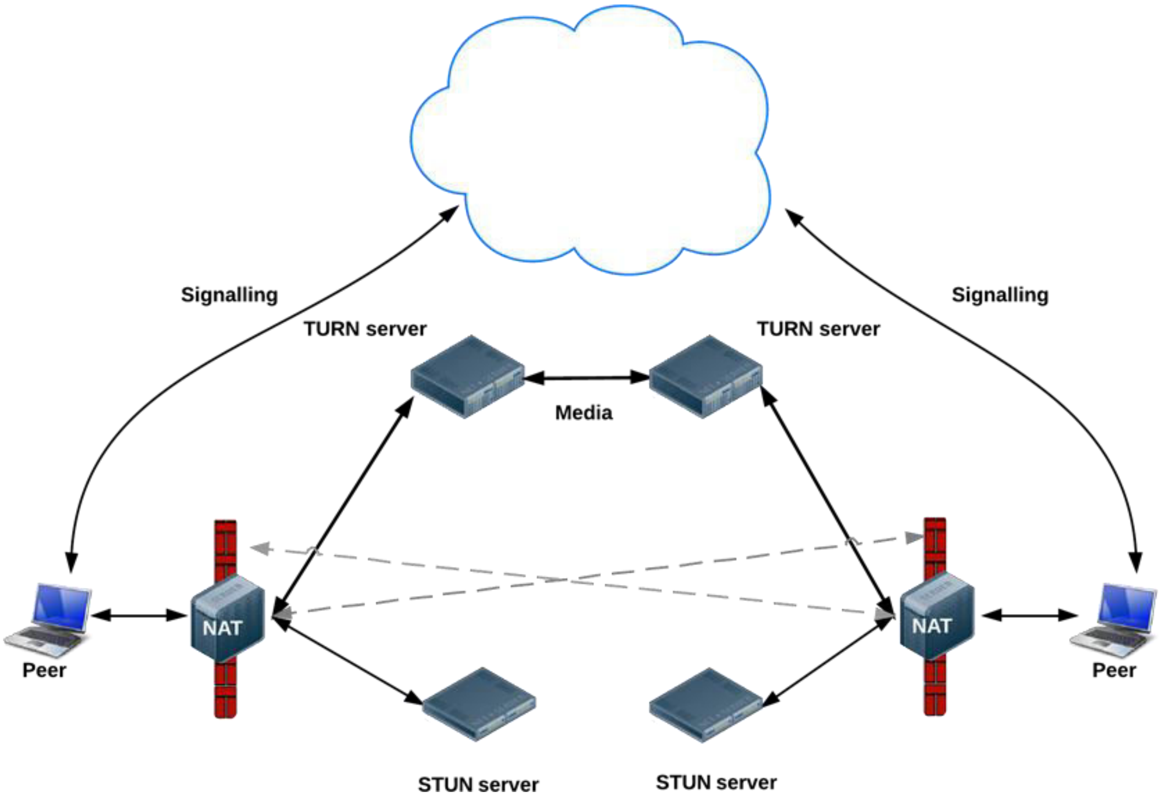


Figure 5. 13 Connection over TURN server between peers

- Client-A finds out their local address and public Internet address by using the STUN server and sends these addresses to Client-B through Signaling Server. Each address received from the STUN server is an ICE candidate.

- Client-B does the same, gets local and public IP addresses from the STUN server, and sends these addresses to Client-A through Signaling Server.

- Client-A receives Client-B's addresses and tries each IP address by sending special pings in order to create the connection with Client-B. If Client-A receives a response from any IP address, it puts that address in a list with its response time and other performance credentials. At last, Client-A chooses the best addresses according to its performance.

- Client-B does the same in order to connect to Client-A.

5.1.2.10 Codecs

Before sending the media over a peer connection, it has to be compressed. Raw audio and video are simply too large to send efficiently in our current Internet infrastructure. Likewise, after receiving media over a peer connection, it has to be decompressed. A media codec (coder-decoder) does exactly this. WebRTC has mandated three audio codecs and two video codecs: [10]

i. Audio - PCMU (G.711µ) running at 8,000Hz with a single channel (mono).

ii. Audio - PCMA (G.711a) running at 8,000Hz with a single channel (mono).

iii. Audio - Opus running at 48,000Hz with two channels (stereo).

iv. Video - VP8.

v. Video - H.264/AVC using Constrained Baseline Profile Level 1.2.

Future media codecs like VP9 and H.265 could be added to the WebRTC standard at some point in the future, but for now, are not mandatory. RTC experts such as Live Switch’s Professional Services team are often able to add additional custom and future codec support to meet any customer’s requirements.

RTP (Real-Time Protocol)

RTP is a mature protocol for transmitting real-time data on top of UDP. Audio and Video are transmitted with RTP in WebRTC. There is a sister protocol of RTP which is RTCP (Real-time Control Protocol) which provides QoS in RTP communication. RTSP (Real-time Streaming Protocol) uses RTP protocol as well in data communication.

WebRTC Signaling Server

The last part is the Signaling Server which is not defined in WebRTC. WebRTC doesn't specify a transport mechanism for the signaling information. You can use anything you like, from WebSocket to XMLHttpRequest to carrier pigeons to exchange the signaling information between the two peers. A signaling server's job is to serve as an intermediary to let two peers find and establish a connection while minimizing exposure of potentially private information as much as possible. It is used to send SDP strings and ICE Candidates between Client-A and Client-B. The Signaling Server also decides which peers get connected to each other. WebSocket technology is the preferred way to Signaling Servers for communication.

It's important to note that the server doesn't need to understand or interpret the signaling data content. Although it's SDP, even this doesn't matter so much: the content of the message going through the signaling server is, in effect, a black box.

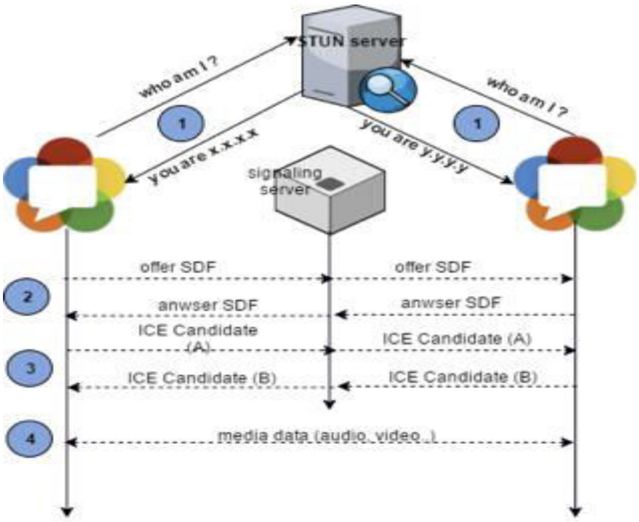
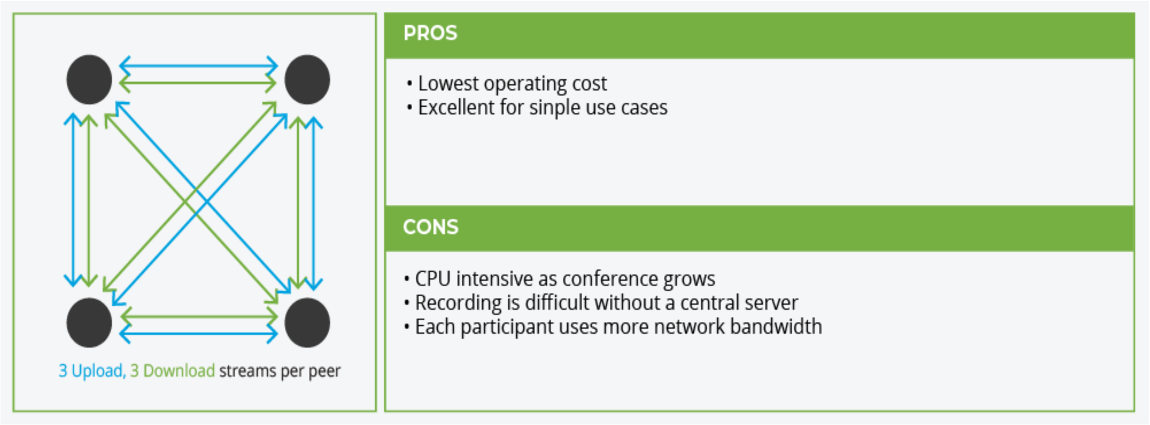


Figure 5. 14 WebRTC with Signaling Server

5.1.2.11 WebRTC Topology

The peer-to-peer (mesh) topology is the only connection type that is covered in the WebRTC specification. However, there are many use cases where a mesh topology is insufficient. Server-based topologies can help address these drawbacks and are often used within the world of WebRTC for transferring media.

The peer-to-peer (mesh) topology is the only connection type that is covered in the WebRTC specification. So, we will be only using that. However, there are many use cases where a mesh topology is insufficient.



PROS

- Lowest operating cost
- Excellent for sinple use cases

CONS

- CPU intensive as conference grows
- Recording is difficult without a central server
- Each participant uses more network bandwidth

Figure 5. 14 Mesh Topology in WebRTC

In a peer-to-peer or mesh topology, each participant in a session directly connects to all other participants without the use of a server. This type of connection is perfect for small video conferences as it is the lowest cost and easiest to set up. However, when conferences grow, maintaining direct connections between all participants becomes unsustainable as it can become too CPU intensive. Since the connections are direct between peers, a mesh topology doesn't lend itself well to recording. For these reasons, a mesh topology is best for simple applications that connect 2 to 3 participants, where low latency is important, and where recording isn't required.

5.2.5.6 WebRTC Security

WebRTC is secure and employs a number of security measures to ensure your data remains secure. These include:

i) Browser Protection:

As we already know, WebRTC is enacted directly between browsers without the need for plugins. This makes WebRTC inherently safer, because it provides an extra level of protection against malware or other undesirable software installations that may be disguised as a plug-in.

ii) Media Access:

The WebRTC specification has addressed potential concerns to allowing access to media resources by requiring explicit permission for the camera or microphone to be used. It is not possible for a WebRTC application to gain access to a device without consent.

iii) Encryption

The preferred method for this is to use Perfect Forward Secrecy (PFS) ciphers in a DTLS (Datagram Transport Layer Security) handshake to securely exchange key data. For audio and video, key data can then be used to generate AES (Advanced Encryption Standard) keys which are in turn used by SRTP (Secure Real-time Transport Protocol) to encrypt and decrypt the media.

5.2 Testing and Deployment

5.2.1 Testing

The testing phase can be carried out manually or by using automated testing tools to ensure each component works fine. After the project is ready its various components were tested in terms of quality, performance to make it error free and remove any sort of technical jargons. Testing also is done to measure the difference between the desired and the developed system. Testing is need on development cycle of system to ensure that the system's every component works fine.

5.2.1.1 Unit Testing

During the coding phase each individual module was tested to check whether it works properly or not. Different errors found during unit testing were debugged. The application is built on JavaScript and we found that Jest library can be used for unit testing as the test runner. Jest is a delightful JavaScript Testing Framework with a focus on simplicity. It works with projects using: Babel, TypeScript, Node, React, Angular, Vue and more! [9]

On the backend side of things, we have built each module as the services so we are going the test each method/ function of the classes. The objects for each service class are created and then the methods of the classes are tested with the test data.

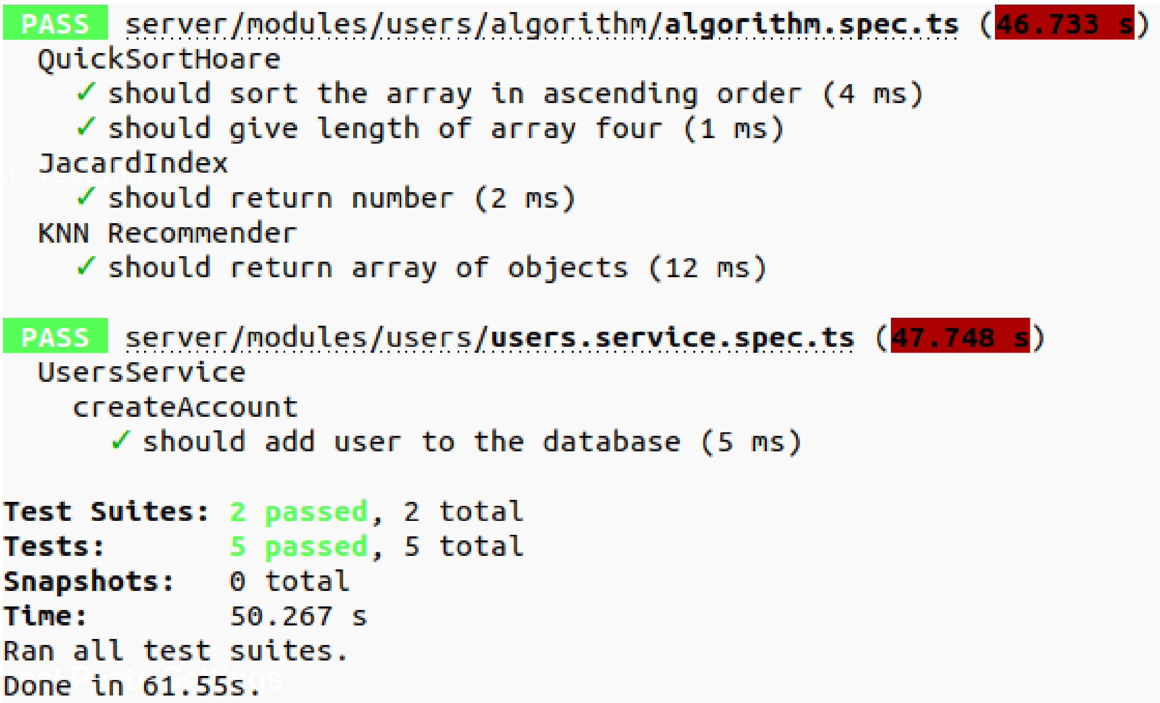
Similarly, on the frontend side each component is tested as the module. Since we are using functional components in React, we should check each function separately using React testing library. [3]

For the algorithm we have mainly used three algorithms on our side. For Quicksort Hoare we need to test if the mentor's data are sorted in descending order on the basis of Jaccard Coefficient Index.

Similarly, for the Jaccard Similarity Coefficient calculation it should give the value greater than 0 and less than 1, which can be later multiplied by 100 to get the percentile value between 0 - 100%.

The recommendation algorithm should return all the mentors list along with their Jaccard index. If the Jaccard index is greater than the base Jaccard index then it should add the mentors to that array otherwise it shouldn't.

Some of the major test cases are listed below:



```
PASS  server/modules/users/algorithm/algorithm.spec.ts (46.733 s)
  QuickSortHoare
    ✓ should sort the array in ascending order (4 ms)
    ✓ should give length of array four (1 ms)
  JacardIndex
    ✓ should return number (2 ms)
  KNN Recommender
    ✓ should return array of objects (12 ms)

PASS  server/modules/users/users.service.spec.ts (47.748 s)
  UsersService
    createAccount
      ✓ should add user to the database (5 ms)

Test Suites: 2 passed, 2 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        50.267 s
Ran all test suites.
Done in 61.55s.
```

Figure 5. 15 Unit tests report for backend module

5.2.1.2 Integration Testing

Integration testing is basically a logical extension of unit testing. This test was done to ensures that different modules work together in the recommendation system like the sorting, filtering and so on. Similarly, it’s done to check if the database of the system has been successfully integrated with the backend and the api has been integrated with the frontend of the system.

Similarly, it`s done to check whether the components in the frontend are combined to get the final page. This ensures that modules are compatible and they can be integrated to form a complete working system. [3]

5.2.2.3 System Testing

System testing has done after integrating testing in order to ensure that the whole systems function properly. After the integration testing the whole system working process was checked. The output was as per the system specifications and hence the system was found to work properly. The api, database and the frontend of the applications are tested at once, as an application.

System testing mainly includes following procedures:

i) Testing the fully integrated applications including external peripherals in order to check how components interact with one another and with the system as a whole. This is also called End to End testing scenario.

ii) Verify thorough testing of every input in the application to check for desired outputs.

iii) Testing of the user's experience with the application.

5.2.2 Test Cases

We have carried out multiple tests to verify that the application performs as per the user expectations or not.

5.2.2.1 User Authentication Testing

Table 5. 1 Test cases for User Authentication

S. N	Test Cases	Input Data	Expected Output
1	Login	Email empty Password: empty	Both username and password field are required
2	Login	Email: random@mail.com Password: random	Invalid email and password please try again.
3	Login	Email: correct@gmail.com Password : correct	User must be redirected to dashboard after successful login.
4	Create Account	Name: Test Email: student@gmail.com Password: 12345 Role: Student	User must be able to register an account and be redirected to the student dashboard.
5	Create Account	Name: Test Email: teacher@gmail.com	User must be able to register an account and be redirected to the teacher dashboard.

		Password: 12345 Role: Teacher	
6	Create Account	Name: Test Email: teacher@gmail.com Password: 12345 Confirm Password: 123456 Role : Teacher	Password you entered didn’t match please try again
7	Create Account	Name: Test Email: teacher@gmail.com Password: 12345 Role: Teacher	User with given email already exists

5.2.2.2 User Profile Creation

Table 5. 2 Test cases for User Profile Creation

S. N	Test Cases	Input Data	Expected Output
1	Create Profile	title: empty description: empty country: empty languages: [] tags: [] channels: []	Please fill up all the fields
2	Check tags length	title: Test Title description: Test description tags: [] country: Nepal	Please select at least 3 tags for profile

		languages: [Urdu, Hindi, Nepali] channels: [{ site : test,link : http://test.com }]	
3	Check languages length	title: Test Title description: Test description tags: [python, toc, ai] country: USA languages: [] channels: [{ site : test,link : http://test.com }]	Please select at least 2 languages for profile
4	Select at least one country	title: Test Title description: Test description tags: [python, toc, ai] country: empty languages: [urdu, hindi] channels: [{ site: test,link : http://test.com }]	Please select at least one country

5	Create teacher/student profile	title: Test Title description: Test description tags: [python, toc, ai] country: USA languages: [urdu, hindi, nepali] channels: [{ site: test,link : http://test.com }]	Profile of user created successfully.

5.2.2.3 Apply for Mentorship

Table 5. 3 Test cases for apply for Mentorship

S. N	Test Cases	Input Data	Expected Output
1	Create mentorship request	Title: empty message: empty expectation: empty background: empty	Please fill up all the fields
2	Create mentorship request	Title: test title Message: test msg expectation: test expectations background: test bg	Mentorship request successfully send

5.2.2.4 Mentorship Reply

Table 5. 4 Test cases for Mentorship reply

S. N	Test Cases	Input Data	Expected Output
1	Accept mentorship request	Status: APPROVED	Mentorship request has been accepted by the mentor.
2	Reject mentorship request	Status: REJECTED	You have rejected a mentorship request.
3	Reply mentorship request	date: 2076/07/20 start Time: 6:30 AM end Time: 7:00 AM room Id: test123 message: test msg	Mentorship details sent to the mentee.
4	Reply mentorship request	date: 2076/07/20 start Time : 6:30 AM end Time: 7:00 AM link: random123 message: test msg	Room with given id hasn’t been created

5.2.2.5 Room Creation and Join Room

Table 5. 5 Test cases for room creation and join

S.N	Test Cases	Input Data	Expected Output
1	Create room	Title: test room	Room created successfully
2	Create room	Title : empty	Title of room should be provided

3	Join room	Room Id : test123	Room joined successfully.
4	Join room	Room Id : random123	Room with given id doesn’t exists

5.2.2.6 Recommendations, Sorting, Jaccard Coefficient Calculation

Table 5. 6 Test cases for recommendations, sorting, jaccard coefficient calculation

S.N	Test Cases	Input Data	Expected Output
1	Recommend users	Array of mentors and mentee	Mentors array with jaccard index > 10
2	Calculate Jaccard coefficient	Mentors : Array<Mentor> Mentee : User	Jaccard index should be between 0-1
3	Quick sort array of mentors	Mentors : Array<Mentor>	Sorted array of mentors

5.2.3 Deployment

The deployment phase is the final phase of the software development life cycle (SDLC) and puts the product into production. After the project team tests the product and the product passes each testing phase, the product is ready to go live. SDLC Deployment Phase provides for production installation and customer acceptance for the software, requiring all test cases to verify successful software execution, completeness, and correctness.

The Deployment Phase in SDLC includes the work necessary to deploy the final solution into the target production environments. Plus, creating guides for installation, system operations, system administration, and end-user functionality.

The methods used by developers to build, test and deploy new code will impact how fast a product can respond to changes in customer preferences or requirements and the quality of each change.

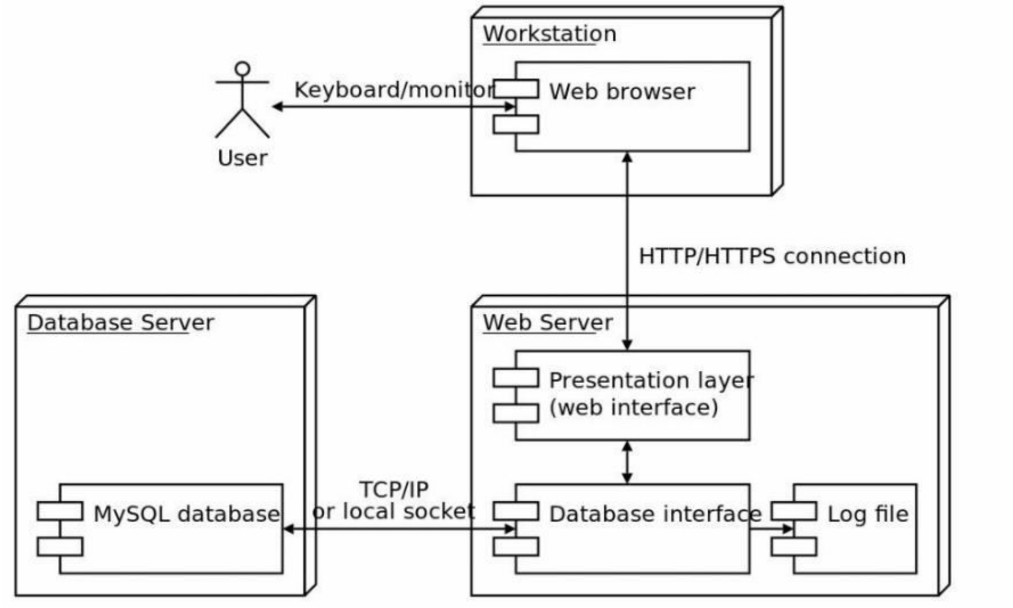


Figure 5. 16 Deployment workflow of Mentor Labs

For the version control we have used Git and GitHub platform. The GitHub runs all the actions for CI/CD and if the tests are passed then it will be pushed to main branch. Then the GitHub bot runs. [7]

on the main branch and the application gets deployed to the Heroku server.

As shown in the above figure we have deployed our backend and frontend of our application on Heroku server. Similarly, we have hosted our PostgreSQL database in the Heroku.

The deployed version of the site can be viewed at: https://mentorlabs.herokuapp.com/

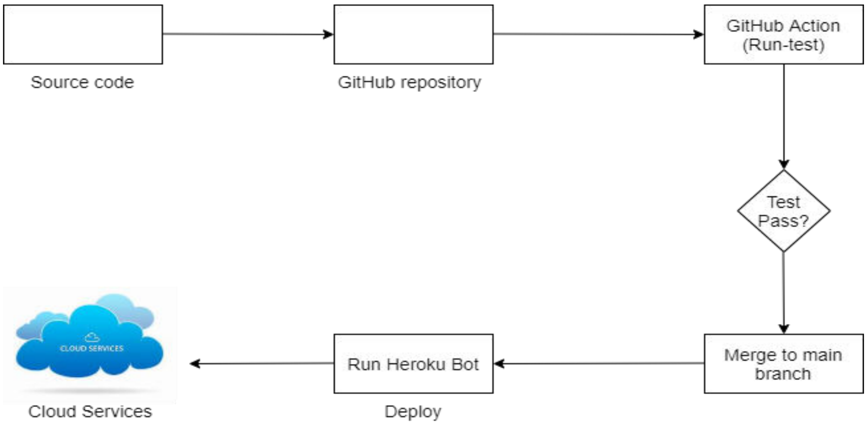


Figure 5. 17 Cloud Architecture of Mentor Labs

5.3 Result Analysis

5.3.1 Data Cleaning / Data Frame Preparation

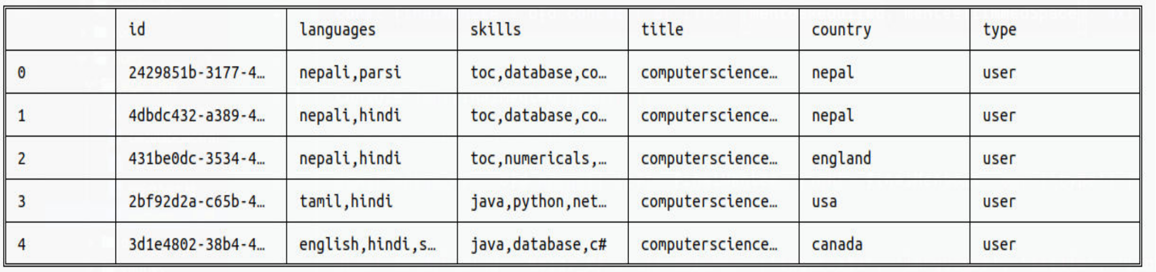
The dataset is obviously taken from Postgres Database. But the problem is that we have to clean the data first and extract only the necessary fields.

First, we have to clean the data of the user and build the final user object in the following form:

```
interface User {
   id : string,
   profile : {
          title : string
         country : string
          languages : Array<string>
          tags : Array<string>
}}
```

First, we convert the data of mentors into the proper data frame so that data can be computed properly. A data frame is a table or a two-dimensional array-like structure in which each column contains values of one variable and each row contains one set of values from each column.

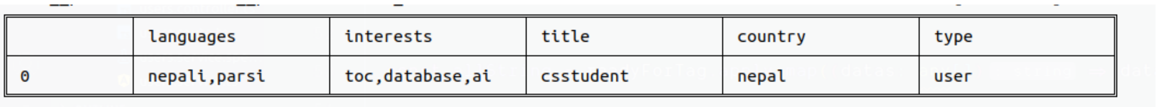
Table 5. 7 Data frame for Mentors



	id	languages	skills	title	country	type
0	2429851b-3177-4…	nepali,parsi	toc,database,co…	computerscience…	nepal	user
1	4dbdc432-a389-4…	nepali,hindi	toc,database,co…	computerscience…	nepal	user
2	431be0dc-3534-4…	nepali,hindi	toc,numericals,…	computerscience…	england	user
3	2bf92d2a-c65b-4…	tamil,hindi	java,python,net…	computerscience…	usa	user
4	3d1e4802-38b4-4…	english,hindi,s…	java,database,c#	computerscience…	canada	user

Similarly, mentee data should be converted into a similar data frame, so that it will be easier to merge the data between mentor and mentee.

Table 5. 8 Data frame for Mentee

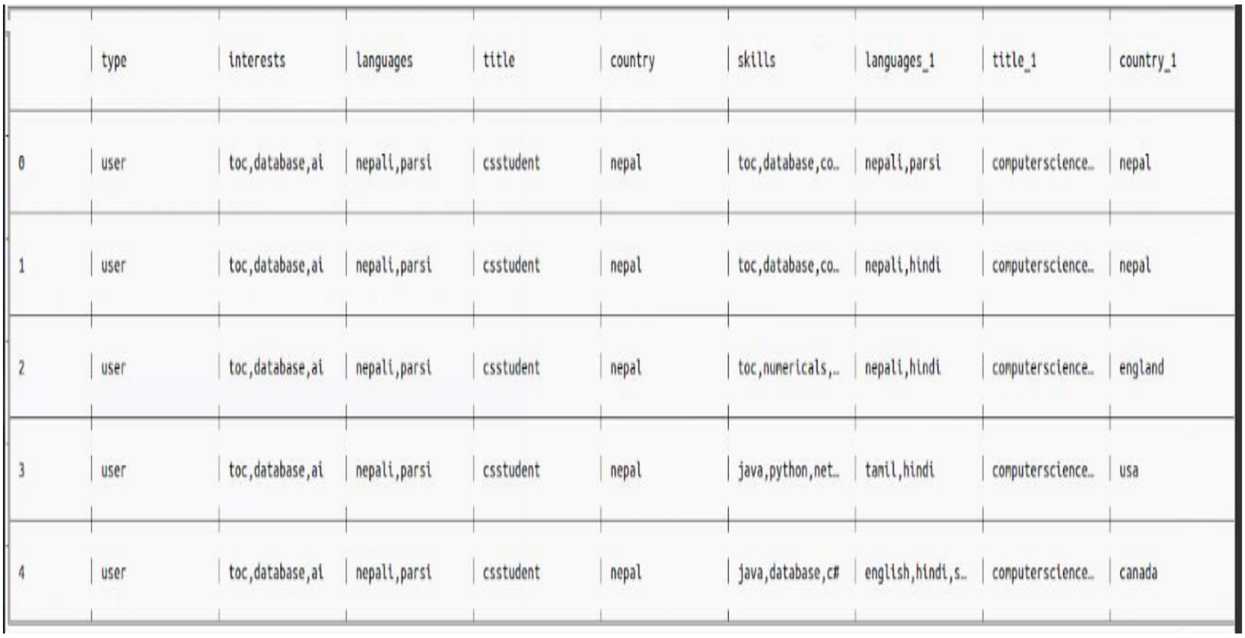


	languages	interests	title	country	type
0	nepali,parsi	toc,database,ai	csstudent	nepal	user

In both of the above data frames we have taken the common properties except the mentee have the interests' properties and the mentors have the skills. Similarly, the new column named "type: user" is added so that they can be merge later. The type column acts like the foreign key in our data frame.

As mentioned, the above data frames are merged into the single table. On the left-hand side, we have mentee (me) and on the right we have mentors. The mentors are mapped one to one with the mentees, with the common column type whose default value is user. Similarly, on the mentor's side the common named columns like languages, title and country are renamed as languages_1, title_1 and country_1.

Table 5. 9 Merged data of Mentors and Mentee

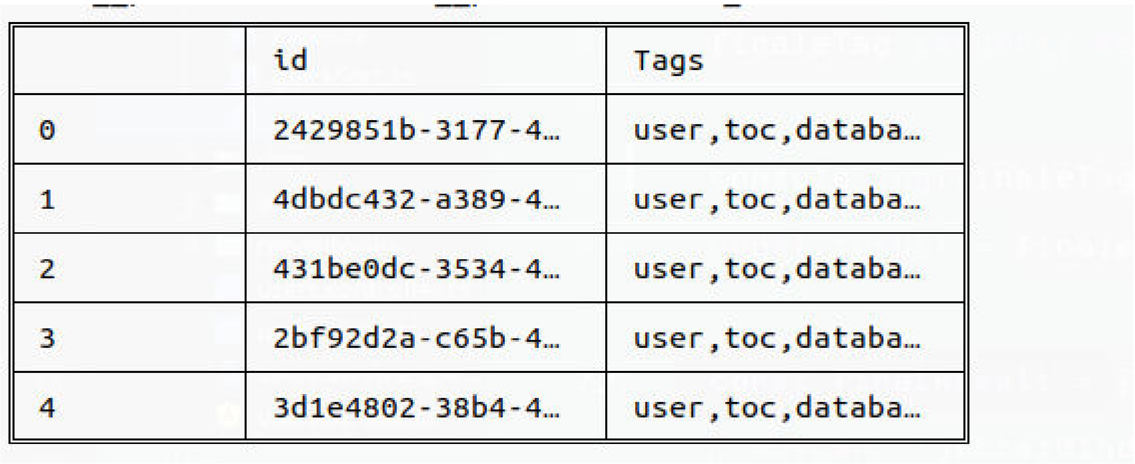


	type	interests	languages	title	country	skills	languages_1	title_1	country_1
0	user	toc,database,ai	nepali,parsi	csstudent	nepal	toc,database,co...	nepali,parsi	computerscience...	nepal
1	user	toc,database,ai	nepali,parsi	csstudent	nepal	toc,database,co...	nepali,hindi	computerscience...	nepal
2	user	toc,database,ai	nepali,parsi	csstudent	nepal	toc,numericals,...	nepali,hindi	computerscience...	england
3	user	toc,database,ai	nepali,parsi	csstudent	nepal	java,python,net...	tamil,hindi	computerscience...	usa
4	user	toc,database,ai	nepali,parsi	csstudent	nepal	java,database,c#	english,hindi,s...	computerscience...	canada

Now we are going to merge all the properties of the mentors and mentee and combine it into one column named "Tags". Point to be noted is that Tags contain all the properties from mentors as well as mentee.

After merging the data between these two it creates the consistency in the data so that it will b easier to calculate the Jaccard coefficient between these two. Now the next step will be to create the data-frame with the id and the tags (combination of interest and skills).

Table 5. 10 Mentors and Mentee merged properties Data Frame



	id	Tags
0	2429851b-3177-4…	user,toc,databa…
1	4dbdc432-a389-4…	user,toc,databa…
2	431be0dc-3534-4…	user,toc,databa…
3	2bf92d2a-c65b-4…	user,toc,databa…
4	3d1e4802-38b4-4…	user,toc,databa…

The above table represents the data frame after merging of skills and interests of mentors and mentee. Now we have to prepare the recommendation algorithm based on the similarities between the mentors and mentee.

CHAPTER 6: CONCLUSION AND FUTURE RECOMMENDATIONS

6.1 Conclusion

Mentor`s Lab online learning platform is the newest and most popular form of distance education today. Within the past decade it has had a major impact on postsecondary education and the trend is only increasing.

We fulfilled the objectives of our project by implementing various important methodologies and algorithms. Initially for the methodology we have used incremental model where increments are done in the system on each ongoing step because of which we got to test smaller units of the system at short period of time. Likewise, we managed to create a web based online video streaming in our project with the use of Web-RTC. For the real time chat system, we used WebSockets which helps to chat even if user doesn't want to share his/her video. We have used browser api for screen sharing feature for students and professionals in our project. We successfully created a separate dashboard for mentors and mentee where the other options like incoming requests, outgoing requests, profile sections etc. can be seen and accessed.

The use of Jaccard algorithm to calculate the coefficient between the mentors and mentee have helped to find out the similarities between them and helps to recommend the mentors to the mentee with similarities in their profiles.

Mentors Lab meets the needs of an ever-growing population of students who cannot or prefer not to participate in traditional classroom settings. These learners include those unable to attend traditional classes, who cannot find a particular class at their chosen institution, who live in remote locations, who work full-time and can only study at or after work, and those who simply prefer to learn independently. Mentors Lab provide an excellent method of course delivery unbound by time or location allowing for accessibility to instruction at anytime from anywhere. The ability to access a course from any computer with Internet access, with scheduled hours of any day is a tremendous incentive for many of today's student

6.2 Future Recommendations

This project lays basic foundation to the enhancement of education opportunities to the students with the professional mentors of their choice with similar backgrounds. Furthermore, the future works will be to make the platform even more mobile, user-friendly, with additional features like:

- Allow students and teacher to make separate workspace like local drives where they can keep their notes, pdfs and other materials which may be required in near future and provide separate dashboard for each.
- Allow recording the streaming videos for future references.
- Allow students to post their queries on their dashboard where mentors as well as fellow students can comment or text them privately to sort their issues.

REFERENCES

[1] "React.js Official Docs," Facebook Open Source, 2015. [Online]. Available: https://reactjs.org/docs/getting-started.html. [Accessed 10 December 2021].

[2] "Real-time communication for the web," Google Developers, 2011. [Online]. Available: https://reactjs.org/docs/getting-started.html. [Accessed 07 October 2021].

[3] V. R. B. C.Larman, "Iterative Modal," *Iterative and Incremental ,* vol. 36, no. 6, pp. 47-56, June, 2003 .

[4] P. Harrington, Machine Learning in Action, Shelter Island, New York 11964: Manning Publications Co. , 2012.

[5] "The TypeScript Handbook," Microsoft, 2012. [Online]. Available: https://www.typescriptlang.org/docs/handbook/intro.html. [Accessed 02 November 2021].

[6] T. Ruether, "RTMP Streaming: The Real-Time Messaging Protocol Explained," March 23, 2021, March 23, 2021. [Online]. Available: https://www.wowza.com/blog/rtmp-streaming-real-time-messaging-protocol. [Accessed 21 November 2021].

[7] B.Eich, "Mozilla Developers Network, MDN," 02 january 2022. [Online]. Available: https://developer.mozilla.org/en-US/.

[8] R. O. a. S. Oni, "Danfo.js Documentation," JSdata, 2020. [Online]. Available: https://danfo.jsdata.org/. [Accessed 25 October 2021].

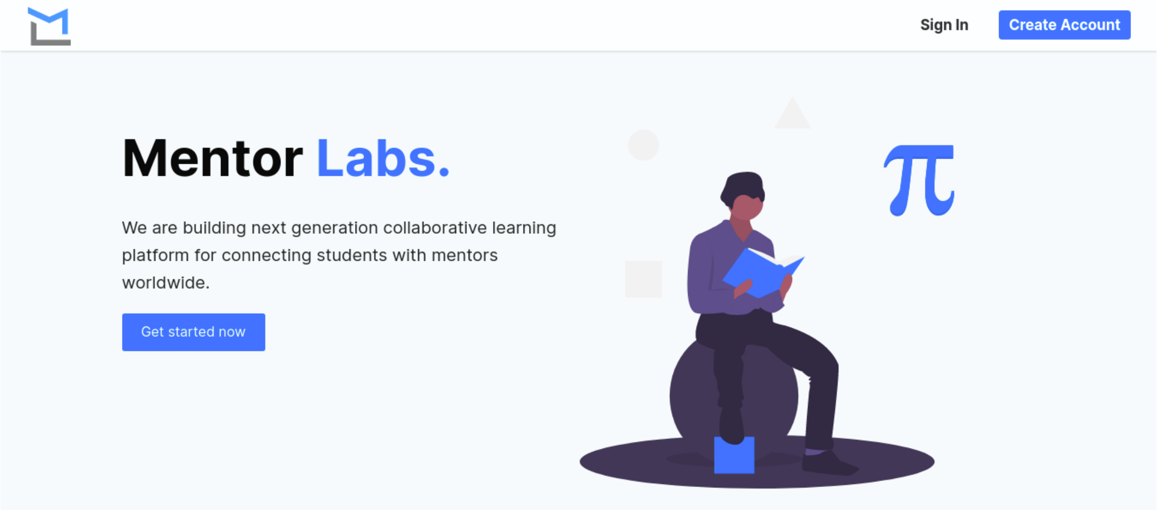
[9] A. D. B. a. C. J. Bergkvist, "WebRTC 1.0," *Real-time Communications Between Browsers,* vol. 08, no. 21, 2012.

[10] G. Ilya., " What every web developer should," in *High Performance Browser Networking:*, O'Reilly Media, Inc, 2013.

APPENDIX

Screenshots:

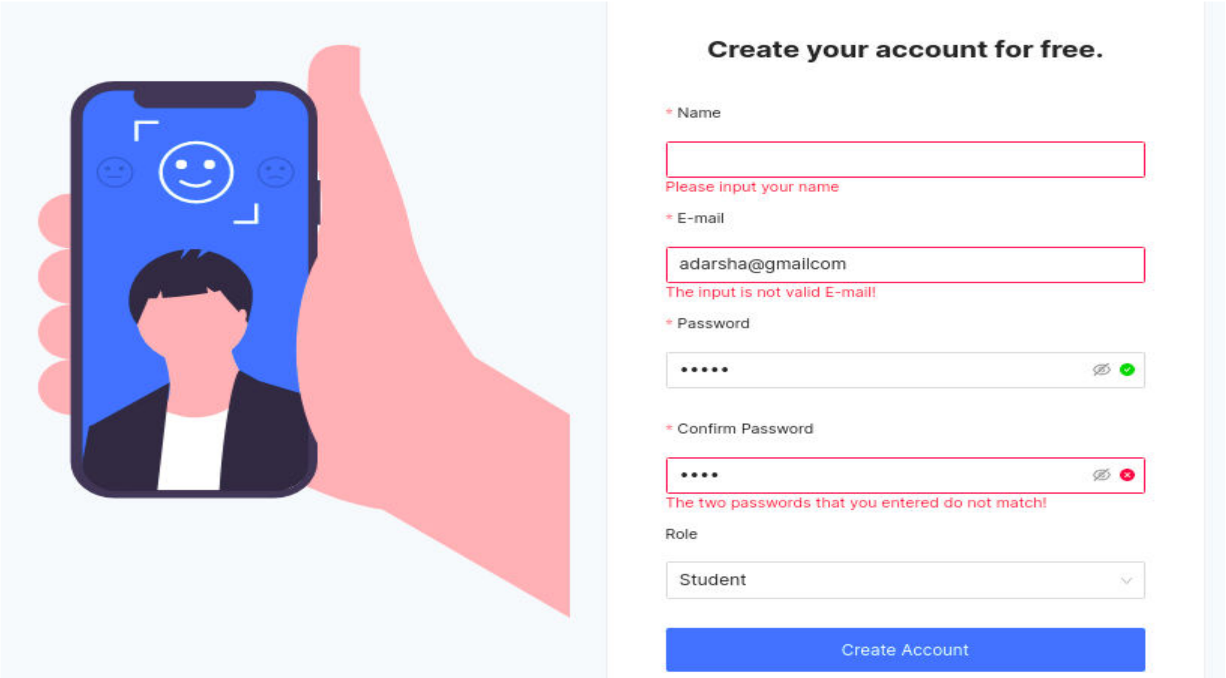
Landing Page



Mentor Labs.

We are building next generation collaborative learning platform for connecting students with mentors worldwide.

Create Account Page



Create your account for free.

* Name

Please input your name

* E-mail

adarsha@gmailcom

The input is not valid E-mail!

* Password

•••••

* Confirm Password

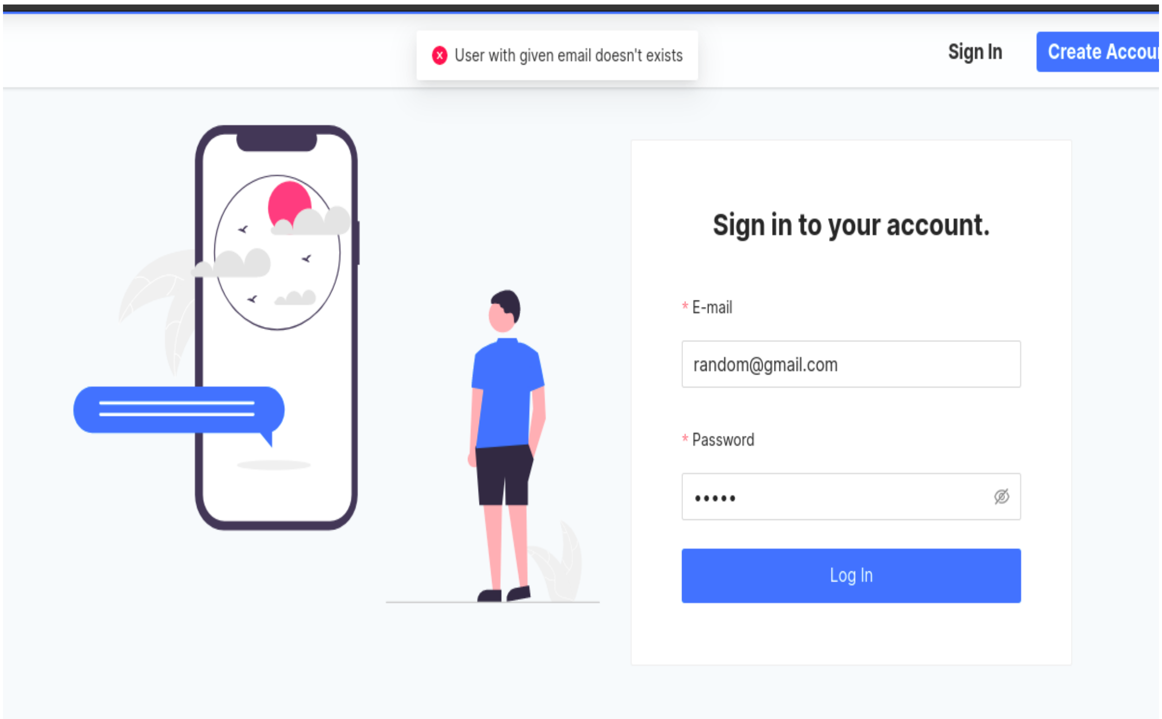
••••

The two passwords that you entered do not match!

Role

Student

Log in Page



User with given email doesn't exists

Sign in to your account.

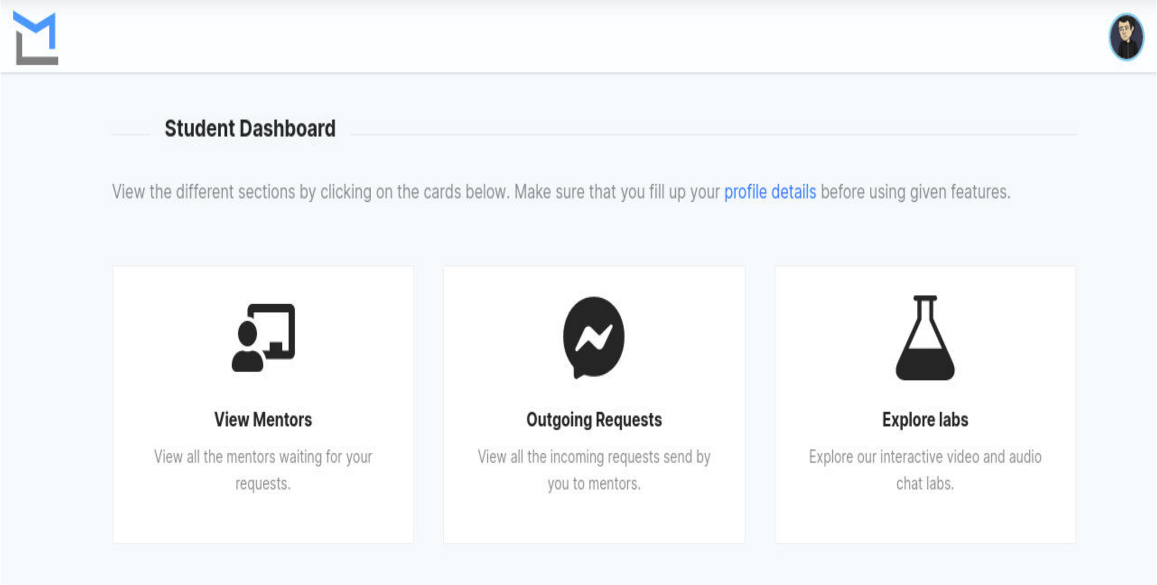
* E-mail

random@gmail.com

* Password

•••••

Student Dashboard Section



Student Dashboard

View the different sections by clicking on the cards below. Make sure that you fill up your profile details before using given features.

View Mentors

View all the mentors waiting for your requests.

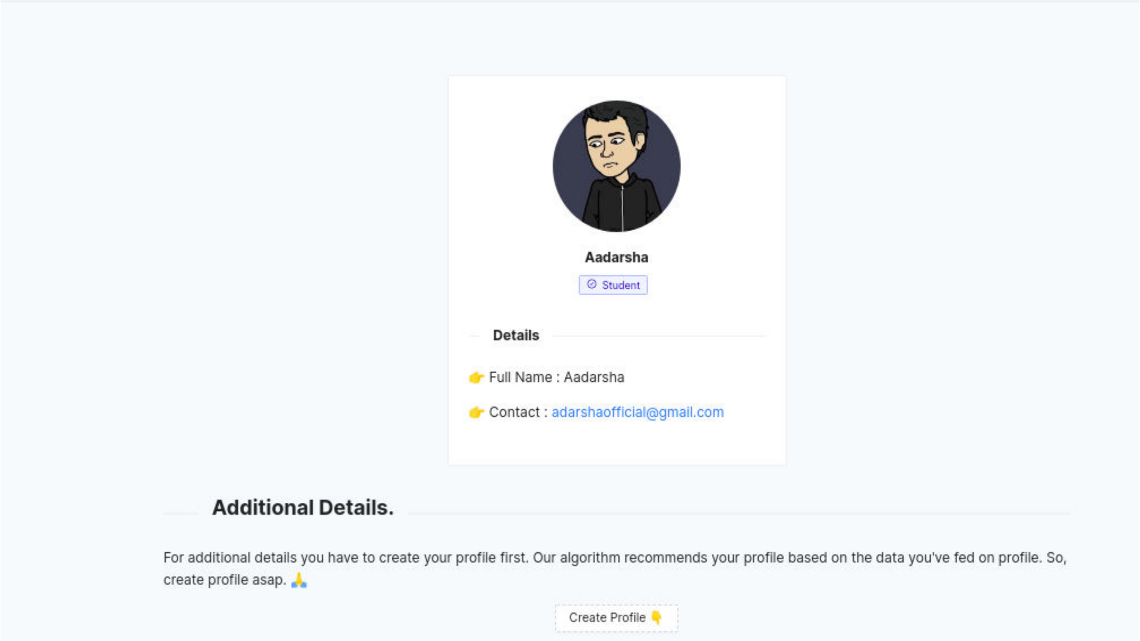
Outgoing Requests

View all the incoming requests send by you to mentors.

Explore labs

Explore our interactive video and audio chat labs.

User Profile


Aadarsha

Details

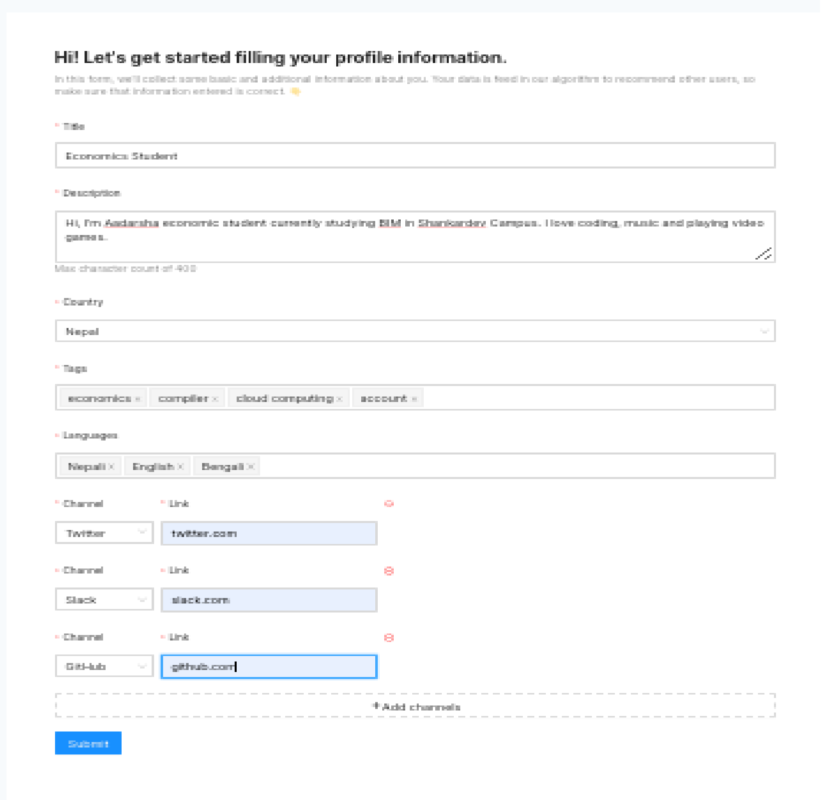
👉 Full Name : Aadarsha

👉 Contact : adarshaofficial@gmail.com

Additional Details.

For additional details you have to create your profile first. Our algorithm recommends your profile based on the data you've fed on profile. So, create profile asap. 🙏

Profile Information Form


Hi! Let's get started filling your profile information.

In this form, we'll collect some basic and additional information about you. Your data is feed in our algorithm to recommend other users, so make sure that information entered is correct. 👇

* Title: Economics Student

* Description: Hi, I'm Aadarsha economic student currently studying BIM in Shankarday Campus. I love coding, music and playing video games.

Max character count of 400

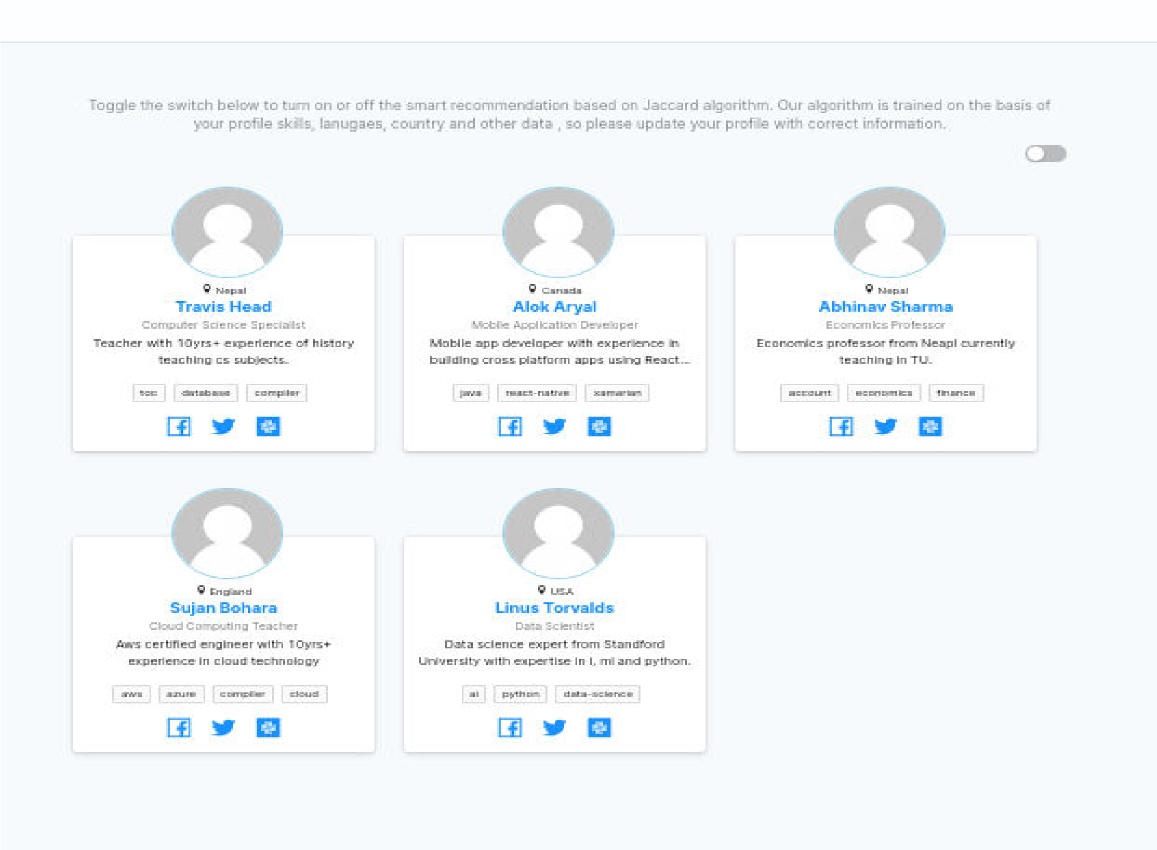
* Country: Nepal

* Tags: economics × compiler × cloud computing × account ×

* Languages: Nepali × English × Bengali ×

* Channel	* Link
Twitter	twitter.com
Slack	slack.com
GitHub	github.com

Available Mentors Profile



Toggle the switch below to turn on or off the smart recommendation based on Jaccard algorithm. Our algorithm is trained on the basis of your profile skills, lanugaes, country and other data , so please update your profile with correct information.

Nepal
Travis Head
Computer Science Specialist
Teacher with 10yrs+ experience of history teaching cs subjects.
toc database compiler

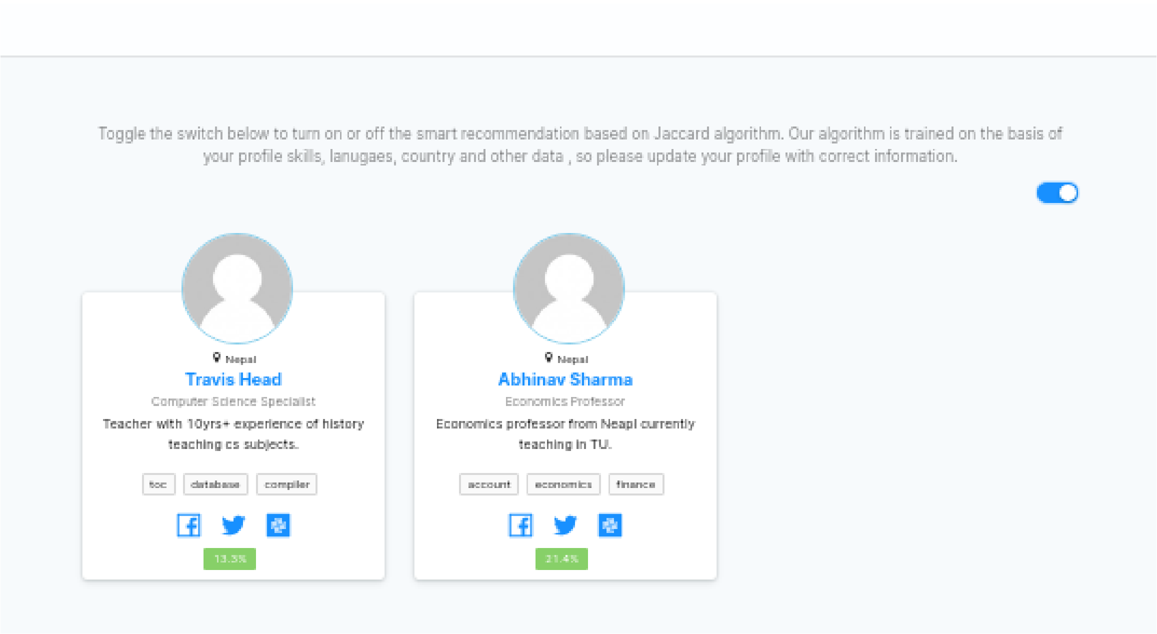
Canada
Alok Aryal
Mobile Application Developer
Mobile app developer with experience in building cross platform apps using React...
java react-native xamarian

Nepal
Abhinav Sharma
Economics Professor
Economics professor from Neapl currently teaching in TU.
account economics finance

England
Sujan Bohara
Cloud Computing Teacher
Aws certified engineer with 10yrs+ experience in cloud technology
aws azure compiler cloud

USA
Linus Torvalds
Data Scientist
Data science expert from Standford University with expertise in i, ml and python.
ai python data-science

Mentors Recommendation Page based on Jaccard algorithm

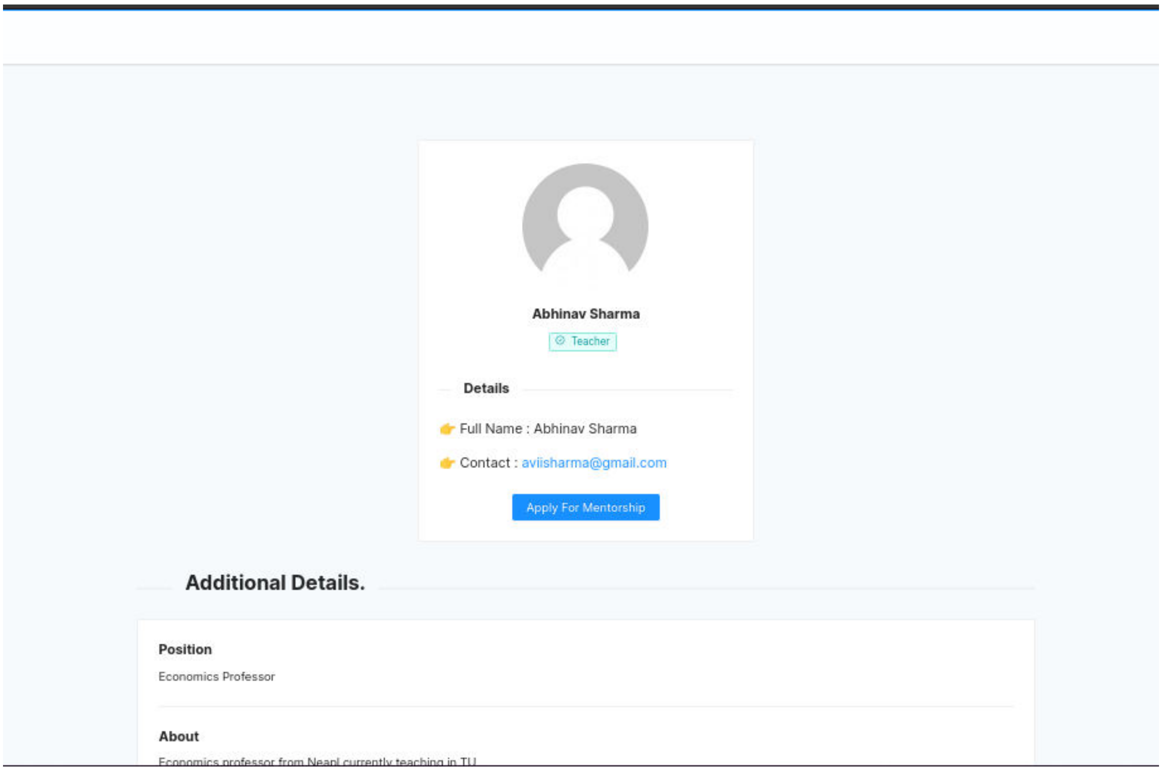


Toggle the switch below to turn on or off the smart recommendation based on Jaccard algorithm. Our algorithm is trained on the basis of your profile skills, lanugaes, country and other data , so please update your profile with correct information.

Nepal
Travis Head
Computer Science Specialist
Teacher with 10yrs+ experience of history teaching cs subjects.
toc database compiler
13.3%

Nepal
Abhinav Sharma
Economics Professor
Economics professor from Neapl currently teaching in TU.
account economics finance
21.4%

Mentor Profile



Abhinav Sharma

Details

Full Name : Abhinav Sharma

Contact : aviisharma@gmail.com

Additional Details.

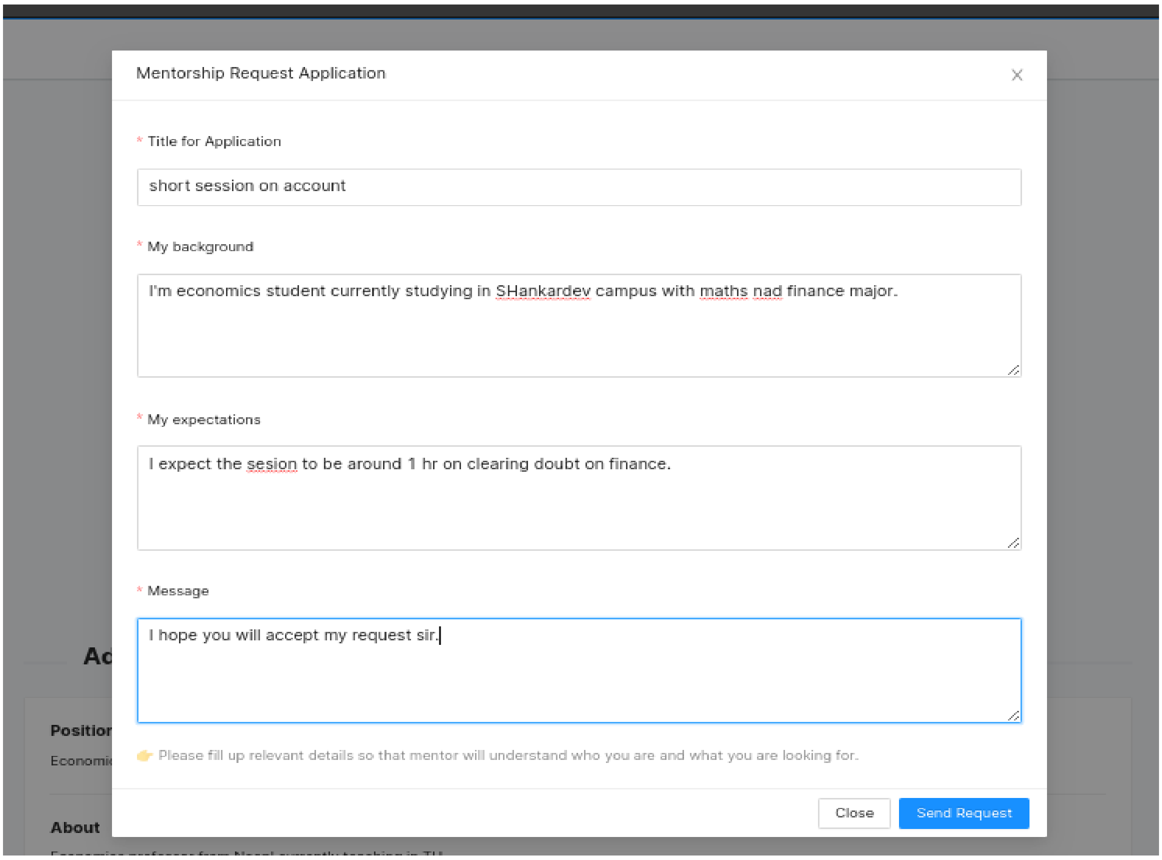
Position

Economics Professor

About

Economics professor from Neapl currently teaching in TU

Mentorship Request Application



Mentorship Request Application

* Title for Application

short session on account

* My background

I'm economics student currently studying in SHankardev campus with maths nad finance major.

* My expectations

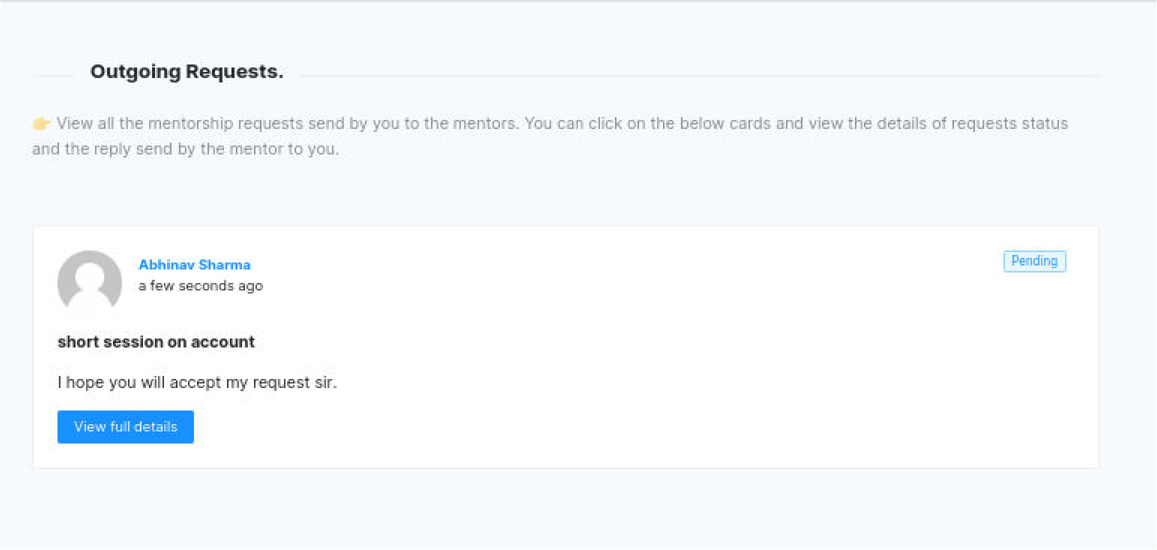
I expect the sesion to be around 1 hr on clearing doubt on finance.

* Message

I hope you will accept my request sir.

Please fill up relevant details so that mentor will understand who you are and what you are looking for.

Outgoing Requests



Outgoing Requests.

👉 View all the mentorship requests send by you to the mentors. You can click on the below cards and view the details of requests status and the reply send by the mentor to you.

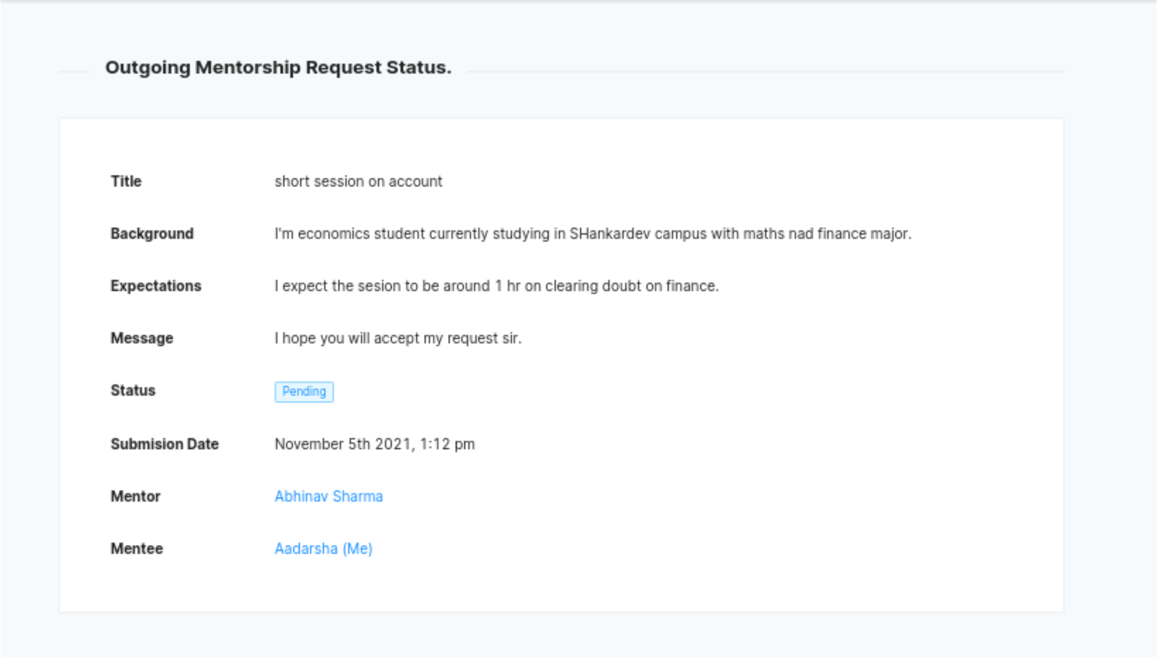
Abhinav Sharma
a few seconds ago

Pending

short session on account

I hope you will accept my request sir.

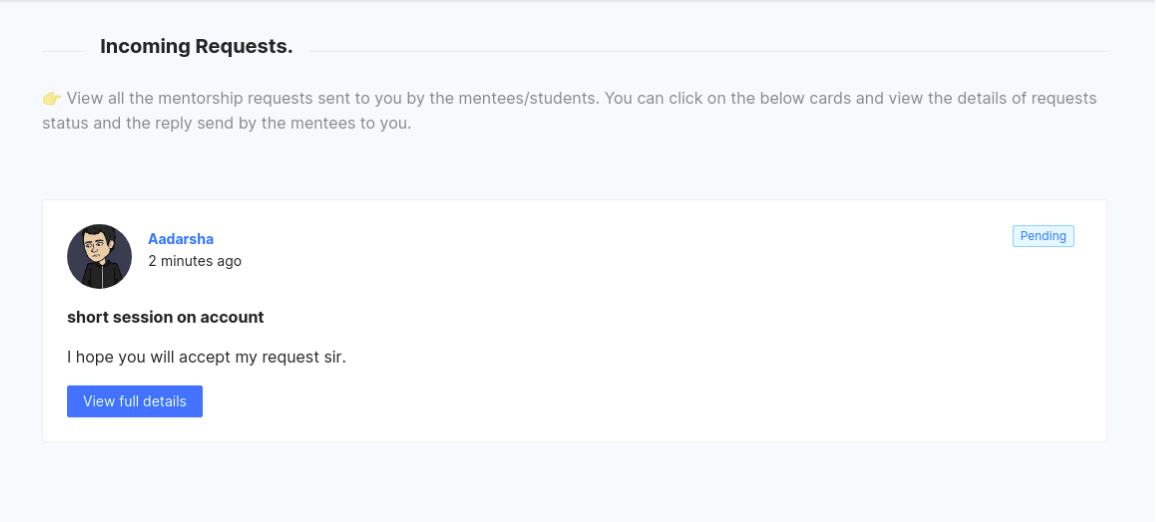
Outgoing Mentorship Request Status



Outgoing Mentorship Request Status.

Title	short session on account
Background	I'm economics student currently studying in SHankardev campus with maths nad finance major.
Expectations	I expect the sesion to be around 1 hr on clearing doubt on finance.
Message	I hope you will accept my request sir.
Status	Pending
Submision Date	November 5th 2021, 1:12 pm
Mentor	Abhinav Sharma
Mentee	Aadarsha (Me)

Incoming Requests



Incoming Requests.

👉 View all the mentorship requests sent to you by the mentees/students. You can click on the below cards and view the details of requests status and the reply send by the mentees to you.

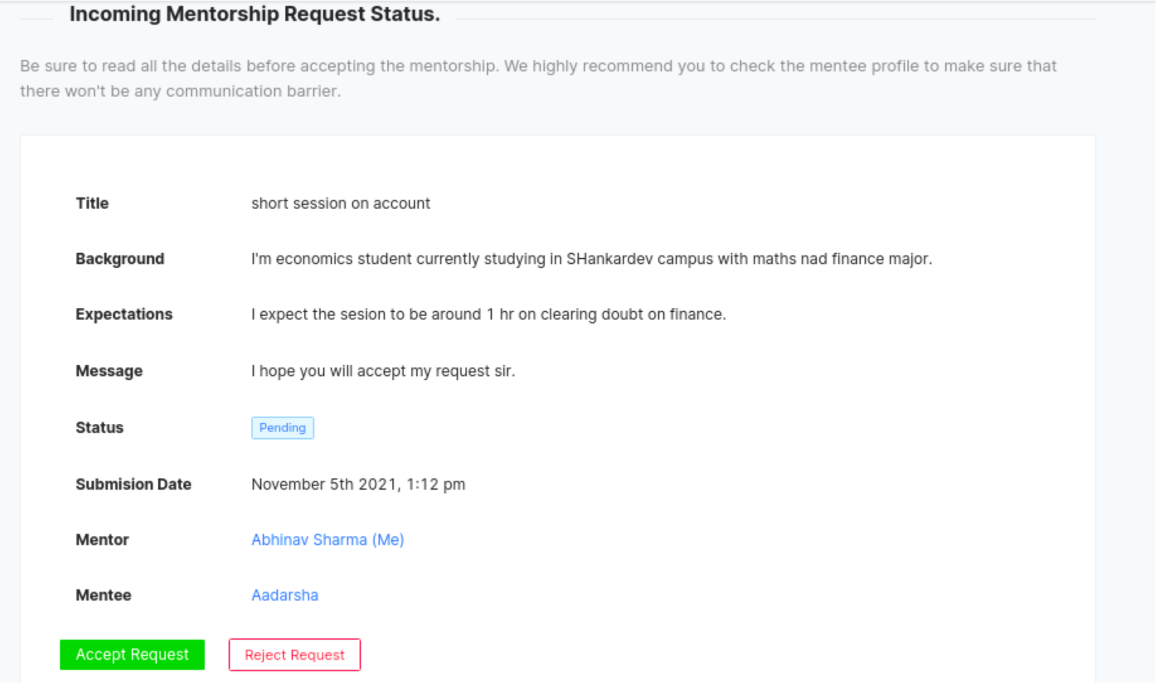
Aadarsha
2 minutes ago

Pending

short session on account

I hope you will accept my request sir.

Incoming Mentorship Request Status

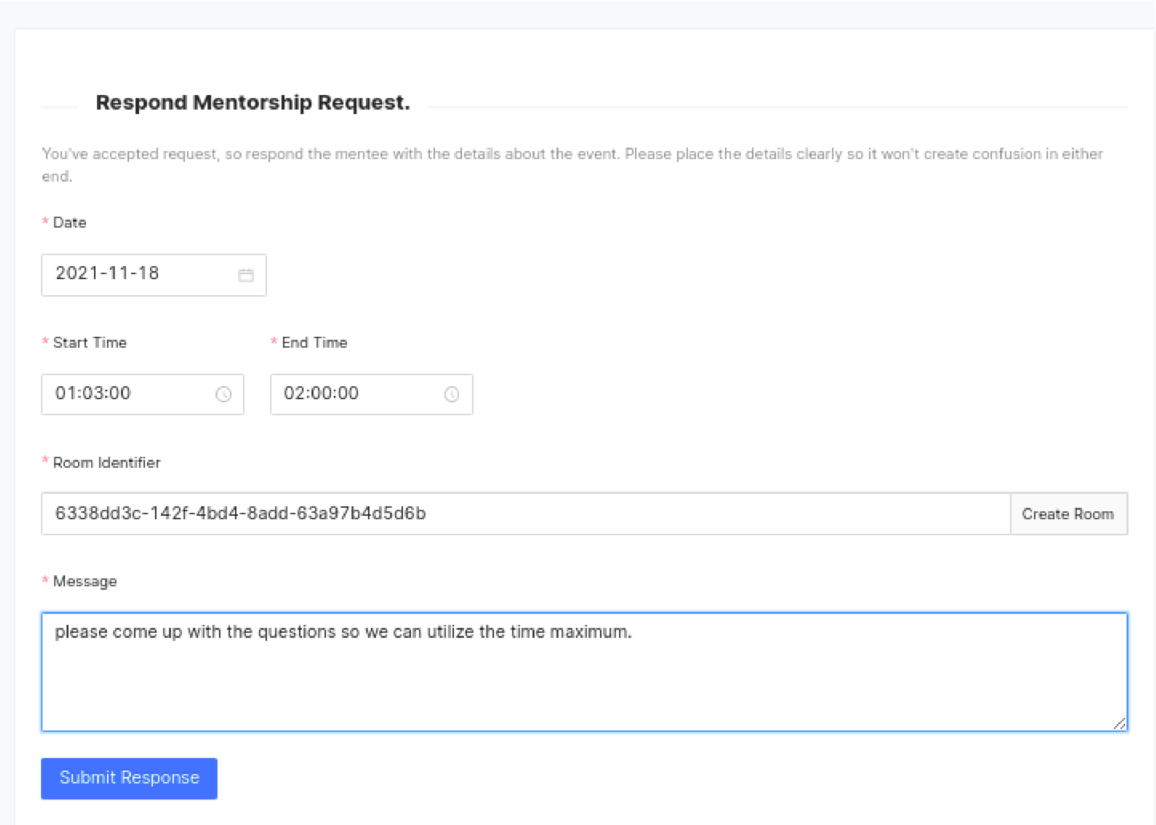


Incoming Mentorship Request Status.

Be sure to read all the details before accepting the mentorship. We highly recommend you to check the mentee profile to make sure that there won't be any communication barrier.

Title	short session on account
Background	I'm economics student currently studying in SHankardev campus with maths nad finance major.
Expectations	I expect the sesion to be around 1 hr on clearing doubt on finance.
Message	I hope you will accept my request sir.
Status	Pending
Submision Date	November 5th 2021, 1:12 pm
Mentor	Abhinav Sharma (Me)
Mentee	Aadarsha

Respond Mentorship Request



Respond Mentorship Request.

You've accepted request, so respond the mentee with the details about the event. Please place the details clearly so it won't create confusion in either end.

* Date

2021-11-18

* Start Time

01:03:00

* End Time

02:00:00

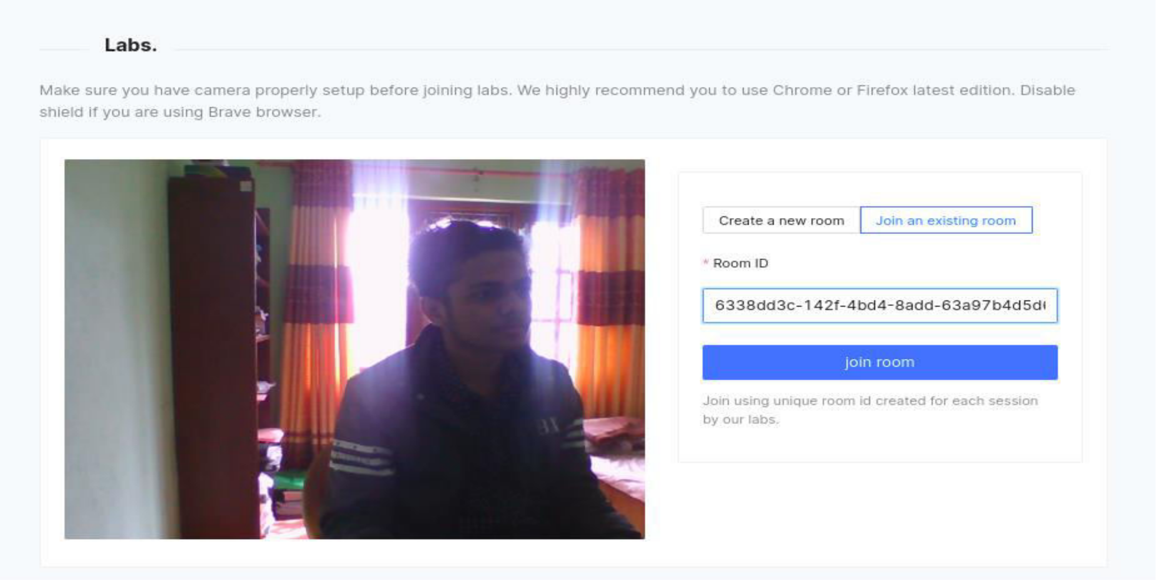
* Room Identifier

6338dd3c-142f-4bd4-8add-63a97b4d5d6b

* Message

please come up with the questions so we can utilize the time maximum.

Room Joining Page



Labs.

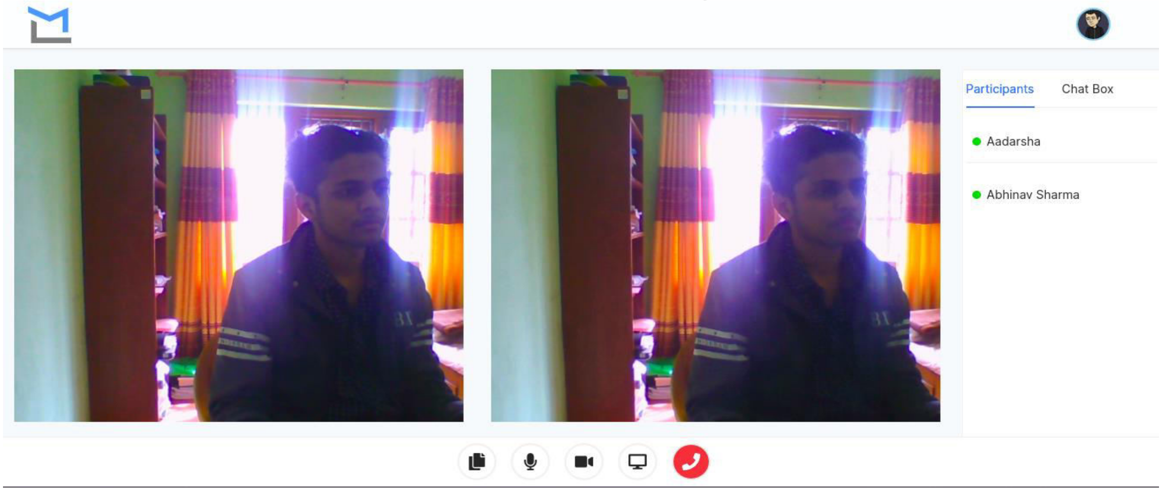
Make sure you have camera properly setup before joining labs. We highly recommend you to use Chrome or Firefox latest edition. Disable shield if you are using Brave browser.

* Room ID

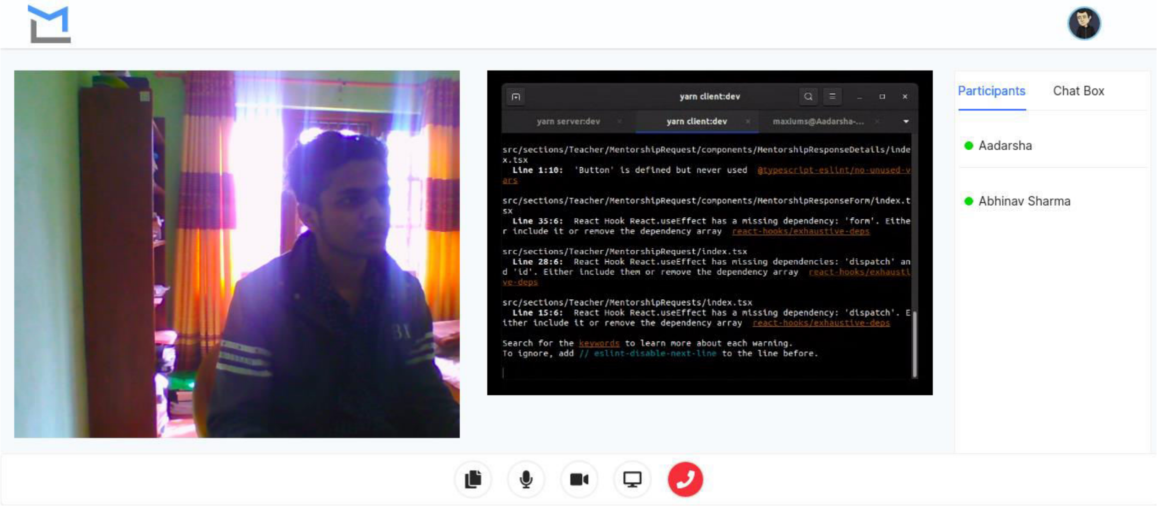
6338dd3c-142f-4bd4-8add-63a97b4d5d(

Join using unique room id created for each session by our labs.

Live Video Sharing



Live Screen Sharing



```
src/sections/Teacher/MentorshipRequest/components/MentorshipResponseDetails/inde
x.tsx
  Line 1:10:  'Button' is defined but never used  @typescript-eslint/no-unused-v
ars

src/sections/Teacher/MentorshipRequest/components/MentorshipResponseForm/index.t
sx
  Line 35:6:  React Hook React.useEffect has a missing dependency: 'form'. Eithe
r include it or remove the dependency array  react-hooks/exhaustive-deps

src/sections/Teacher/MentorshipRequest/index.tsx
  Line 28:6:  React Hook React.useEffect has missing dependencies: 'dispatch' an
d 'id'. Either include them or remove the dependency array  react-hooks/exhausti
ve-deps

src/sections/Teacher/MentorshipRequests/index.tsx
  Line 15:6:  React Hook React.useEffect has a missing dependency: 'dispatch'. E
ither include it or remove the dependency array  react-hooks/exhaustive-deps

Search for the keywords to learn more about each warning.
To ignore, add // eslint-disable-next-line to the line before.
```

The End