**Group Project Log**

***Note:*** *all information must be filled out. You must hand in the project log along with each group project deliverable for this course (e.g., milestones, proposals, reports). The percentage of work allocated to each group member must add up to 100%.*

|  |  |
| --- | --- |
| **Group Name:** | Group 8 – Tutor Time |

|  |  |
| --- | --- |
| **Group Members:** | Devanshu Srivastava, Sneha Sagar, Bhumi Patel, Ganrong Tan, Tirtha Modi |

|  |  |
| --- | --- |
| **Deliverable:** | Project Proposal |

|  |  |
| --- | --- |
| **Group Member Name** | **Work Done (%)** |
| Devanshu Srivastava | 20 |
| Tirtha Modi | 20 |
| Sneha Sagar | 20 |
| Bhumi Patel | 20 |
| Ganrong Tan | 20 |
|  |  |
| **Total:** | **100%** |

**PROJECT PROPOSAL**

**NAME OF PROJECT GROUP – TUTOR TIME (GROUP 8)**

Members and Contributors

|  |  |  |
| --- | --- | --- |
| Devanshu Srivastava  B00810667  dv960112@dal.ca | **Sneha Sagar**  B00808431  sn891352@dal.ca | **Bhumi Patel**  B00824756  bh842792@dal.ca |
| Ganrong Tan  B00707434  gn485145@dal.ca |  | **Tirtha Modi**  B00826404  tr580079@dal.ca |

Faculty of Computer Science  
Dalhousie University

19th June 2019

**CONTENTS**

[1. PROJECT BACKGROUND 1](#_Toc11874607)

[1.1 Project Objective 1](#_Toc11874608)

[1.2 Live Project URL 1](#_Toc11874609)

[2. APPLICATION DETAILS 2](#_Toc11874610)

[2.1 Target User Insight 2](#_Toc11874611)

[2.2 Brand Attributes 2](#_Toc11874612)

[2.3 Competitive Landscape 2](#_Toc11874613)

[2.4 Project Scope 3](#_Toc11874614)

[2.5 Information Architecture 5](#_Toc11874615)

[2.5.1 Proposed Sitemap 5](#_Toc11874616)

[2.5.2 Website Design 13](#_Toc11874617)

[2.6 User Experience 18](#_Toc11874618)

[3. Asset Inventory 20](#_Toc11874619)

[4. Group Roles 21](#_Toc11874620)

[5. REFERENCES 21](#_Toc11874621)

[APPENDIX A 24](#_Toc11874622)

[A.1 Wireframes 24](#_Toc11874623)

[A.2 Code referencing 28](#_Toc11874624)

# PROJECT BACKGROUND

Tutor Time is a service-based application. The purpose of our project is to develop a platform which will provide an opportunity to users to learn and communicate with others. There concept behind Tutor Time is new and we are not re-designing an existing application. Although there are some applications that are like Tutor Time. We haven’t referred to any such application but have discussed our competitors in the following sections.

With the project idea we are aiming to target two set of population I.e. tutors and students. Tutors can utilize this platform to teach students new skills. On the other hand, students can benefit themselves by getting trained under professionals. For instance, a teacher who is retired can use this application to continue their passion of teaching and can make some money from it *[14].* Also, the students who need coaching in some subjects or a person who has graduated and need some help to enhance their skills can use this application. This service will be helpful for both tutors and students as they can benefit a lot from it.

## Project Objective

The motivation of this project is to provide a platform to tutors and students to teach and learn and thereby benefit themselves. With this project we are aiming to achieve an application which would be service-based and on utilizing this application, two group of population i.e. teachers and students can be benefited. The objective of Tutor Time is to set up a portal for the audience who will be closely related to the curriculum I.e. aiming an application which will concentrate on scheduling meet ups for a tutor and students *[14]*.

We proposed this idea after keeping multiple factors in mind, like, tutor and student meetups, spreading a word for those tutors who wants to earn part time income by doing what they love to do. In student's perspective, we clearly agree that many a times they require quick assistance in some advance subject that they are unable to get from schools or on their own. In such cases, they might seek assistance from our application where they can perform an advanced search for tutors based on location and key skills the students are aiming for. In this manner, we give this problem a feasible solution which is Tutor Time (web portal) and try to eliminate the gap and problems faced by tutors and students *[14].*

## Live Project URL

In this Section, you are expected to provide the URL from which your application can be accessed throughout the term:

**http://web.cs.dal.ca/~yourusername/csci5709/group#/**

This section includes the GITLAB repository link of our project. It does not match the format because we had created the project repo before going through the project proposal.

<https://git.cs.dal.ca/devanshu/tutortime/tree/devbranch>

# APPLICATION DETAILS

## Target User Insight

The purpose of the project is to provide a web-based platform that would help the students to search for appropriate tutors. Our target users are people who want to learn new academic skills, languages and tutors having knowledge and are willing to teach others [14]. We are planning to provide students with an opportunity to connect with the tutors and enroll different courses using this platform. Hence, we would be targeting two types of end-users, namely; students and tutors.

The website can be used by students of different age varying from elementary school students to university level students. This website will provide a platform to tutors who willingly wants to take tuitions from home, teach students as a part-time or passion. Students who need a tutor for enhancing their learning skills or learning new skills can register as students on our website *[14].* A person who wants to work as a tutor can make an account on the website which will help them connect with students and help them to explore their expertise in their subjects.

Users can access this web application from a desktop or mobile devices with an internet connection. Users should have basic computer knowledge about how to access the website.

## Brand Attributes

We a provide platform for students to find highly qualified tutors in their area. This website also provides a platform for tutors to advertise their skills and achievements to get more students. We are going to get feedback from both students and tutors to maintain high quality.

Some popular attributes of our application are the discussion forum, chat with the tutor. These attributes give a competitive edge to other applications.

## Competitive Landscape

We have searched and got few results for the applications that are somewhat like our application idea following are the list of available competitors to our application;

1. First Tutors: Canada *[11]*
2. Findatutor *[12]*
3. Superprof *[13]*

We provide two additional features private chat and discussion board. The personal chat feature allows students to directly communicate with tutors. Students can create discussion thread using the discussion board, students can share this discussion with other students and tutor of that specific course. Students can also add comments on the shared discussion. We will be using Neilson’s heuristics in completing this application. Our application UI will be easy to understand for the user and, we are meticulous[[1]](#footnote-2) when it comes to speed and efficiency.

## Project Scope

We are aiming to complete 10 features of this project. *Table 1* shows the list of features we are intending to complete *[14].*

Table 1 – List of features and their purpose

|  |  |  |
| --- | --- | --- |
| Sr. No. | Features | Purpose |
|  | Generic Search | This is our application’s landing page. This page would have a search box, using which the students can search for tutors. This search box would give the students a basic idea of how our application works. On searching for tutors, the student will be redirected to the next page that would list two tutors only. In order to get access to or view more tutors, the student will have to register with the application. The wireframe related to the feature is shown in the below sections. |
|  | Student Profile Management | For our application, we are planning to have two types of student accounts i.e. Basic student account and Premium student account. Premium student account will have all features as the basic student, but the key difference would be of a Chat Box. These features will enable students to manage their profiles efficiently. Student profile will have a dashboard, feedback section, and a discussion forum. |
|  | Tutor Profile Management | Like Student profile, Tutor profile works in a similar manner. It consists of Dashboard, Feedback and Discussion Forum. However, here the functionality of features differs from the student profile features. |
|  | Advanced Search | Advanced search will display a list of all available tutors based on location and key skills entered by the students. Students will have to register with the application to use this feature. This is different from generic search as the search results here are not just limited to two results. |
|  | Chat Box | Chat Box feature will be available to students who are our premium customers. Using this feature, the student can directly chat with the tutor instead of creating threads and going through the discussion board. This would be a beneficial feature for students who don’t like asking queries through discussion boards and who wish for more detailed explanations. |
|  | Email Notifications | As a team, we introduced this feature so that we can send course enrolment notifications to both students and tutors via email. Email notifications will serve as a platform for confirming class schedules by both, registered students and tutors. |
|  | Confirming registrations via email | Whenever a user registers with the application, a confirmation email would be sent to the email address used while registering. This email would contain a link using which the user will have to confirm their email address with the application. Only after successful email address confirmation, the user will be allowed to log into the application. |
|  | Forgot Password | If a user forgets their password, by using this feature the application will send a link on the user’s registered email address. User can then reset their password by browsing their email and clicking the link. |
|  | Discussion Forum | As discussed in feature sections b and c, Discussion board will allow the students to post query related threads and the tutors can respond to those queries. |
|  | Report Issue | Another feature that we would implement is reporting issues and contacting us. Clients can access this page to send reports of any issues they face while accessing the application. We would plan more about this feature in the next phase of the project. |

Table 1: List of features

Also, we have listed the sections of UI that are complete and the ones those are pending.

Table 2 – User Interface achieved

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No | Feature | User Interface | Validations |
| 1. | Landing Page | Completed | N/A |
| 2. | Tutor login and registration | Completed | Completed |
| 3. | Student login and registration | Completed | Completed |
| 4. | Student dashboard | Completed | Completed |
| 5. | Tutor dashboard | Half done | Pending |
| 6. | Add courses | Completed | Completed |
| 7. | Confirmation via email | Pending | Pending |
| 8. | Advanced Search/Basic Search | Integration pending | Integration Pending |
| 9. | Forgot/reset password | Completed | Completed |
| 10. | Enroll class | Integration Pending | Integration Pending |
| 11. | Discussion Forum | Completed | Completed |

Table 2: Completed and pending sections

As per *Table 2*, we have implemented 80% of our applications user interface along with validations. We have planned to complete the entire user interface by the end of this week.

## Information Architecture

Below sections demonstrate the final sitemaps and the wireframes of our application. We have referred these wireframes and sitemap to design the final UI of our application.

### Proposed Sitemap

For sitemaps *[14]*, we finalized the sitemaps and wireframes of one team member and decided to work on that. Below is the sitemap and a list of wireframes finalized for our project.

The sitemap describes the complete structure of our application. We have illustrated the navigations by three colors name: Blue, yellow and orange.

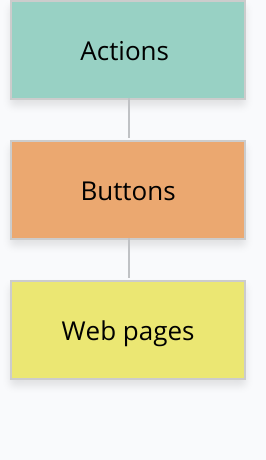


Figure 1 *[14]*: Illustration of colors in sitemap

*Figure 1 [1]* tells about the colors of sitemap i.e. Blue color represents the actions; orange represents buttons and yellow represents the web pages.

Next, *Figure 2 [1]*, represents the actual sitemap of our application Tutor Time. The entire application is split into four parts namely: Search box, Login process, Registration or Sign up and Support. These sections have ten features in all. Search Box lets the student perform basic search and understand the idea behind our application. Login and registration process handle two main features i.e. Student profile and Tutor profile. Support feature will provide an option to contact with us in case of any overheads or difficulties faced while using the application.

A screenshot of a social media post

Description automatically generated

Figure 2 *[14]*: Tutor Time Sitemap

To simplify the sitemap, we have also put the images of each section separately.

*Figure 3* [1] shows the web pages and navigation steps that the basic search would have.

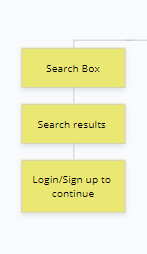


Figure 3 *[14]*: Search Box sitemap

*Figure 4a and 4b [1]* illustrates the all the webpages, actions to be performed and navigations[[2]](#footnote-3) of login function. This feature enables both, student and tutor login features.

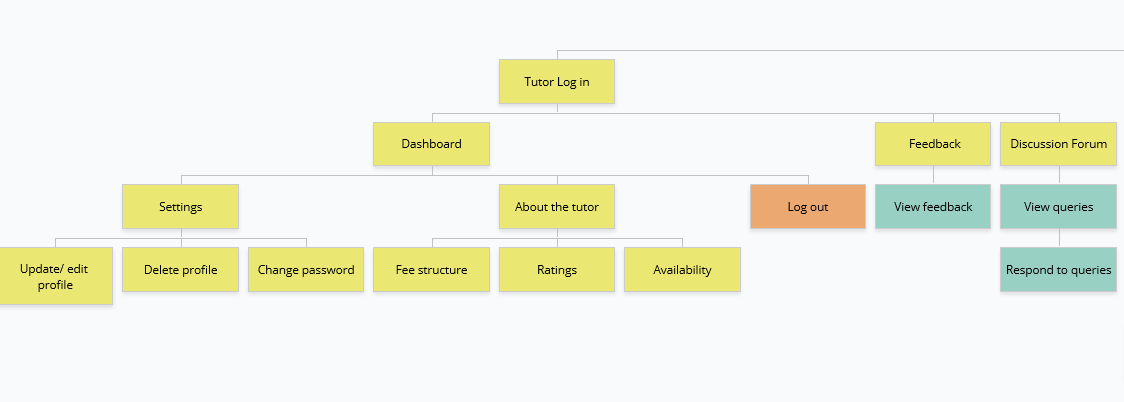


Figure 4a *[14]*: Login function sitemap for tutors

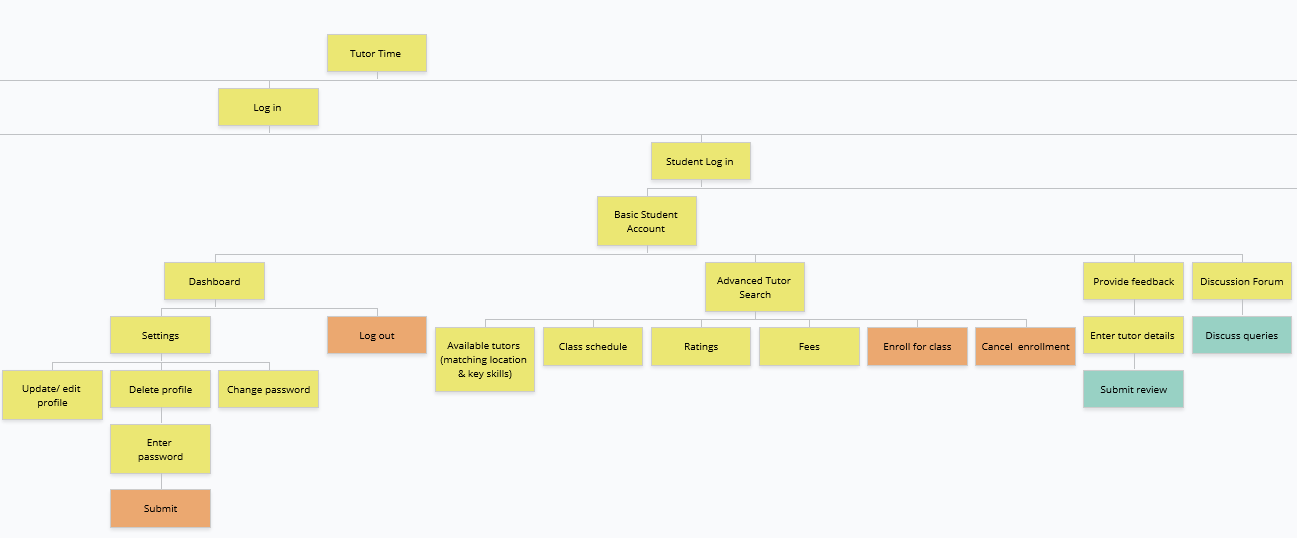


Figure 4b *[14]*: Login for student.

*Figure 5 [1]* represents registration and support features. Registration feature will let the target audience, sign up with the application. In case of troubleshooting, users can contact us with their queries.

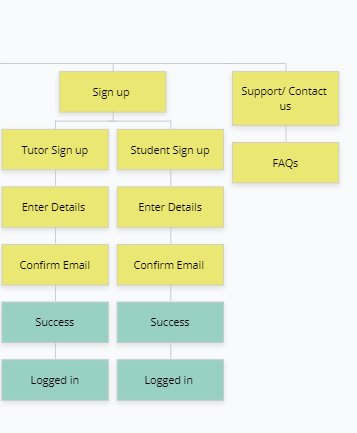


Figure 5 *[14]*: Zoomed in sign up and support sitemaps.

The wireframes that we have used in the proposal are like those used in assignment 1 by the group members. We have merged those wireframes and finalized the wireframes as per our application.

Below is the list of wireframes:

On discussing the wireframes [14] of each member after assignment 1, we came up with a combined idea of how our final application would look like. Below are some wireframes of the application that have built.

*Figure 6* [2] is the wireframe of our applications landing page which demonstrates a search box, options to register and login to the application and report issues if any. The landing page would give the user an idea of how our application will work.

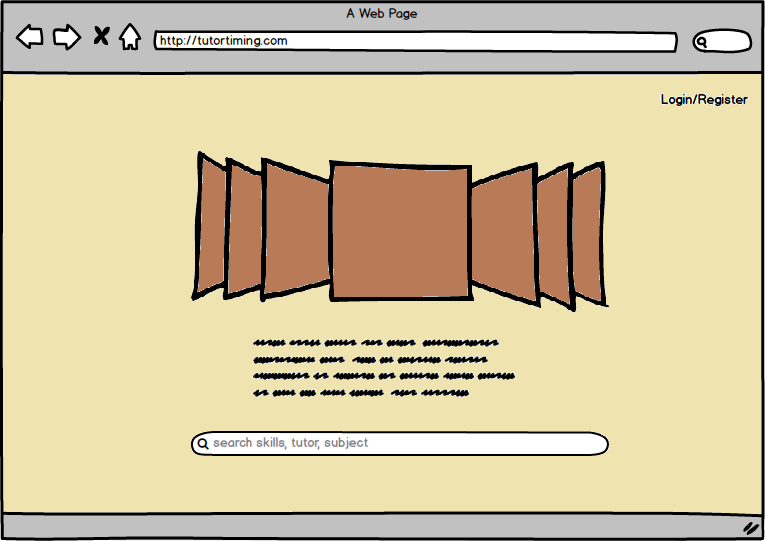


Figure 6: Landing Page

*Figure 7 [2]* shows the wireframe of the search results after performing basic search. This page will show a list of two tutors based on key skills entered by the student. To view more tutors, students are required to register or log in with the application.

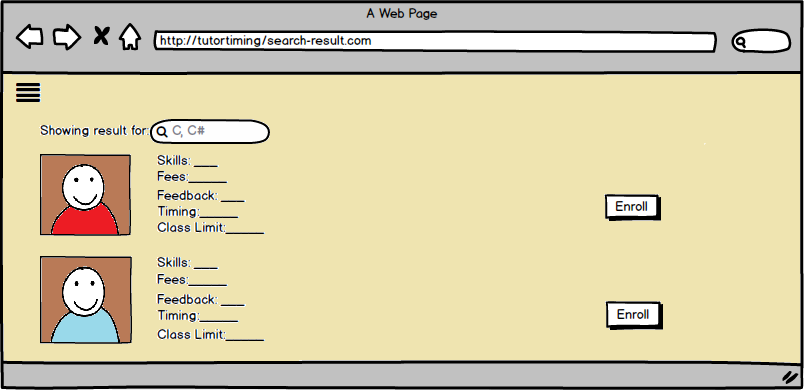


Figure 7: Basic Search Result

Registration page for tutor would be like *Figure 8 [2]*. Tutor will have to fill in all the fields. We have done form validations for the fields present on this page. After filling all the fields tutor will then click Next and proceed to the following page. Though, we have changed some fields in the actual UI of our application.

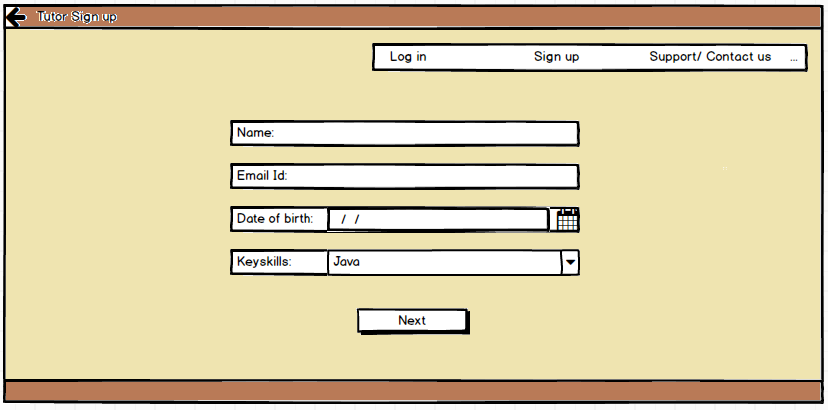


Figure 8: Tutor Register Page

On clicking Next, the wireframe shown in *Figure 9 [2]* will appear. Here, the application will ask the tutors to fill in some more information. Form validations are done, thereby, notifying the users in case some fields are missing. Now, the tutor can log in and access our application.

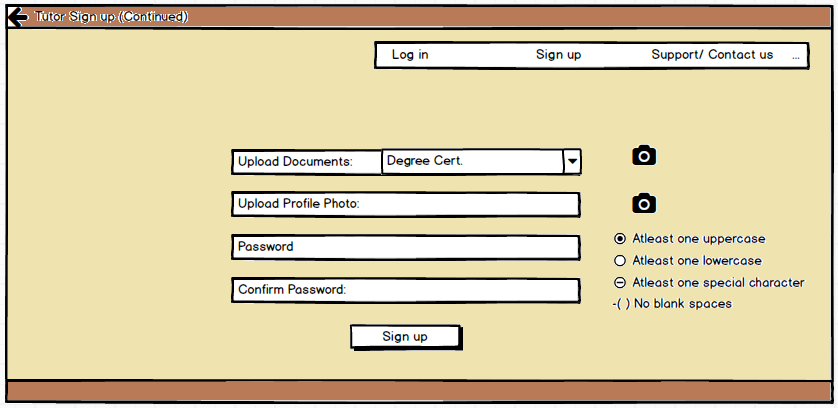


Figure 9: Tutor register page (contd.)

After registering, Tutor can login and browse their account whenever needed. *Figure 10 [2]* shows the wireframe of how our applications login page for tutors look like. Tutors will have to enter the combination of their registered email id and password. If the combination matches, user will be logged in or else the application would show an alert message asking the user to enter correct combination of email id and password (will work after connecting to the backend). In case users forget their passwords, they can fix it by clicking on the forgot password link.

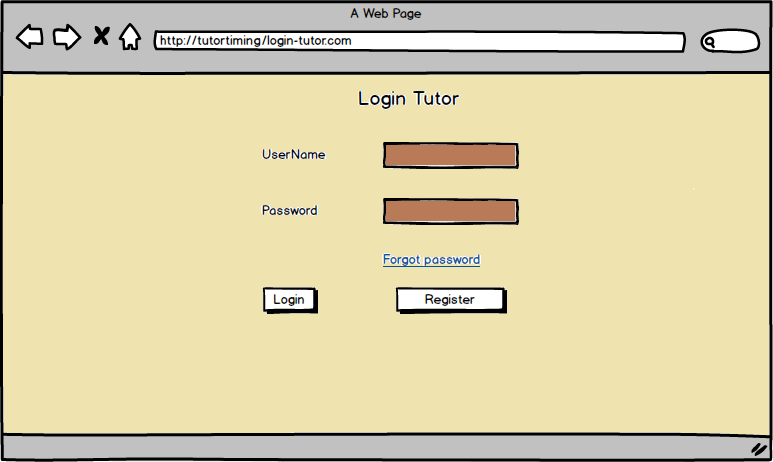


Figure 10: Tutor Login Page

Next wireframe i.e. *Figure 11*. allows the tutor to add courses and make changes if any.

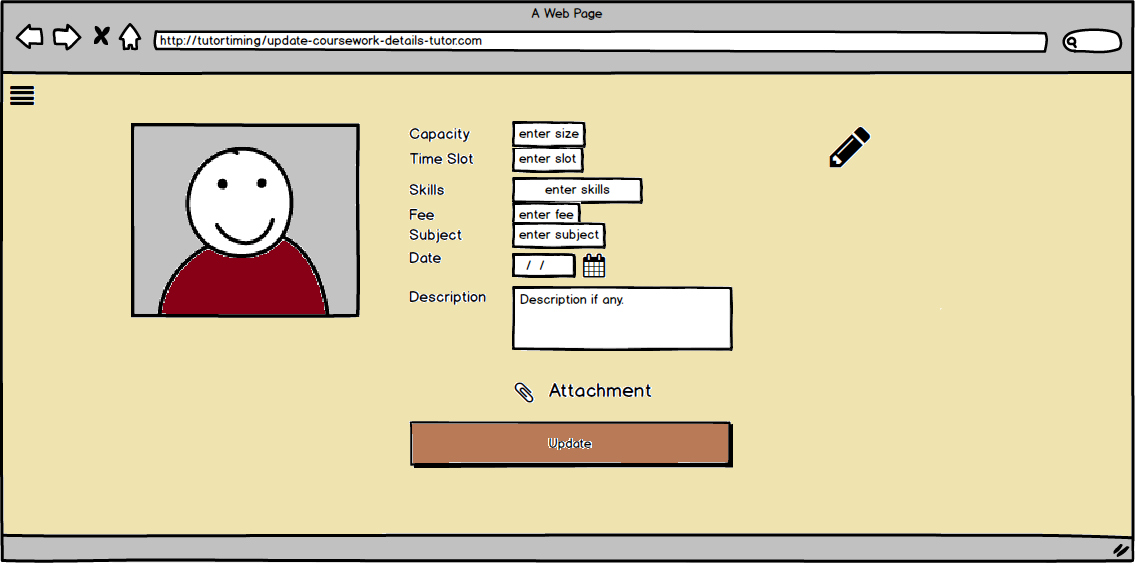


Figure 11: Add course by tutor

*Figure 12*. shows how the wireframe which can be used by the target user to reset their password.

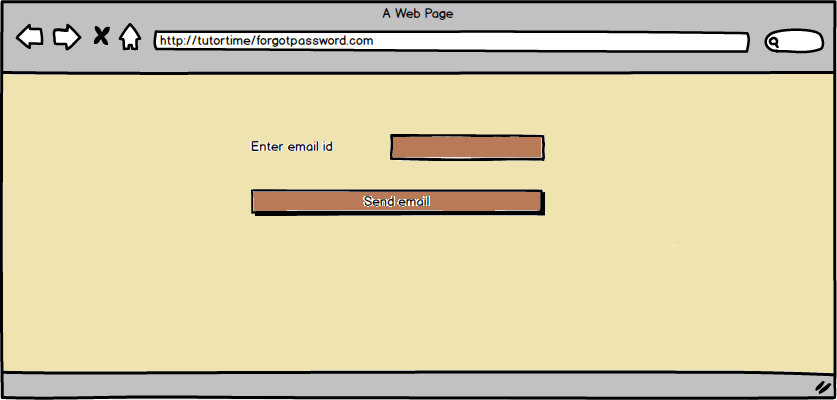


Figure 12: Forgot Password

Another important feature of our application is the advanced tutor search. This search will be performed by the students to search the tutors based on the key skills and location. On searching, a list of all the available tutors will be displayed and the students can enroll to the courses. *Figure 13 [2]* represents the wireframe of the advanced search feature.

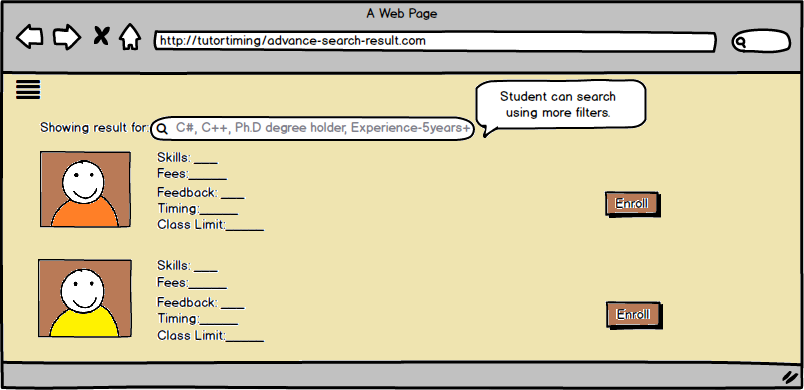


Figure 13: Advanced Search by student

The sign up/ registration page for students is like tutors sign up page except few fields. Also, we have created other pages related to student profile management such as the login page, discussion board where the students can get their queries resolved by posting threads. We have put the low-fidelity prototype for student profile pages in the appendix section.

### Website Design

Below is the list of few webpage designs that we have created. We have used Angular, HTML and CSS to create the webpages.

*Figure 14* shows the design of login page for both students and tutors. Both users can login using this functionality. Validations check are performed on this page.

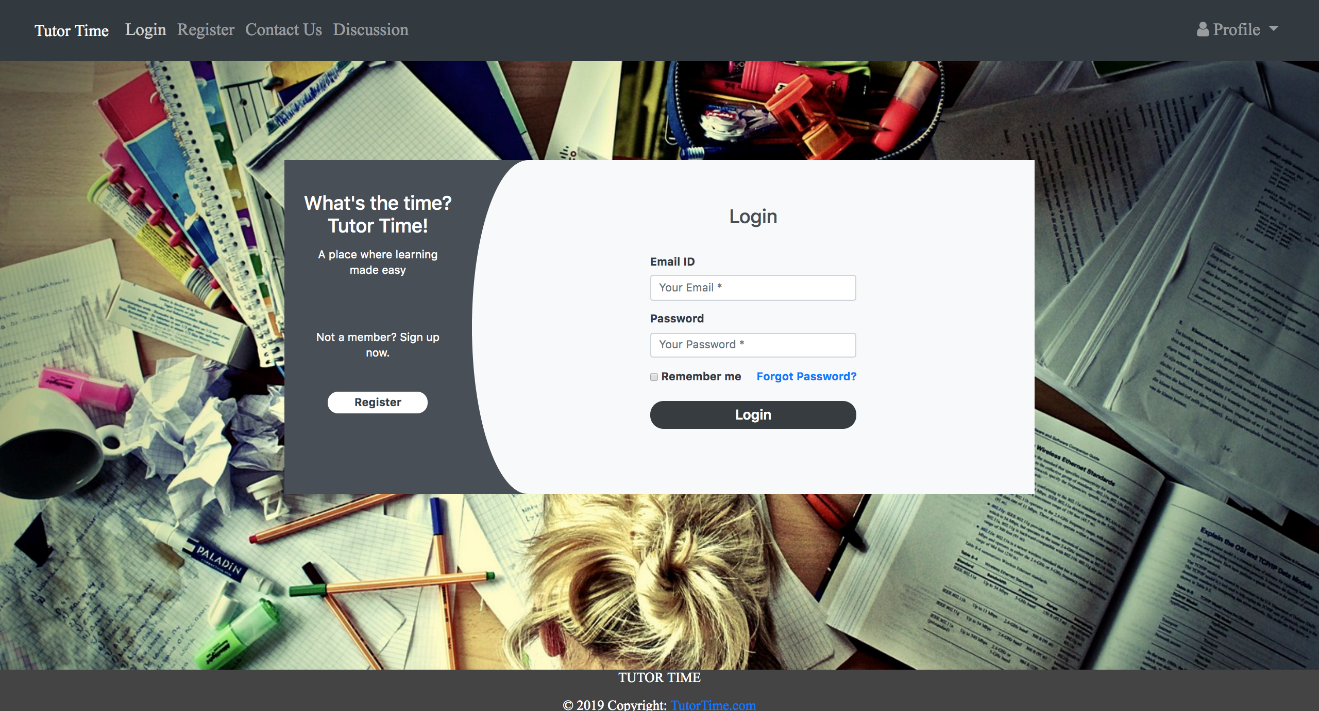


Figure 14: Login Page for both students and tutors

*Figures 15 and 16* shows the registration page for students and tutors. Validations check are done on this page. A toggle is used to switch between registration pages of tutor and student.

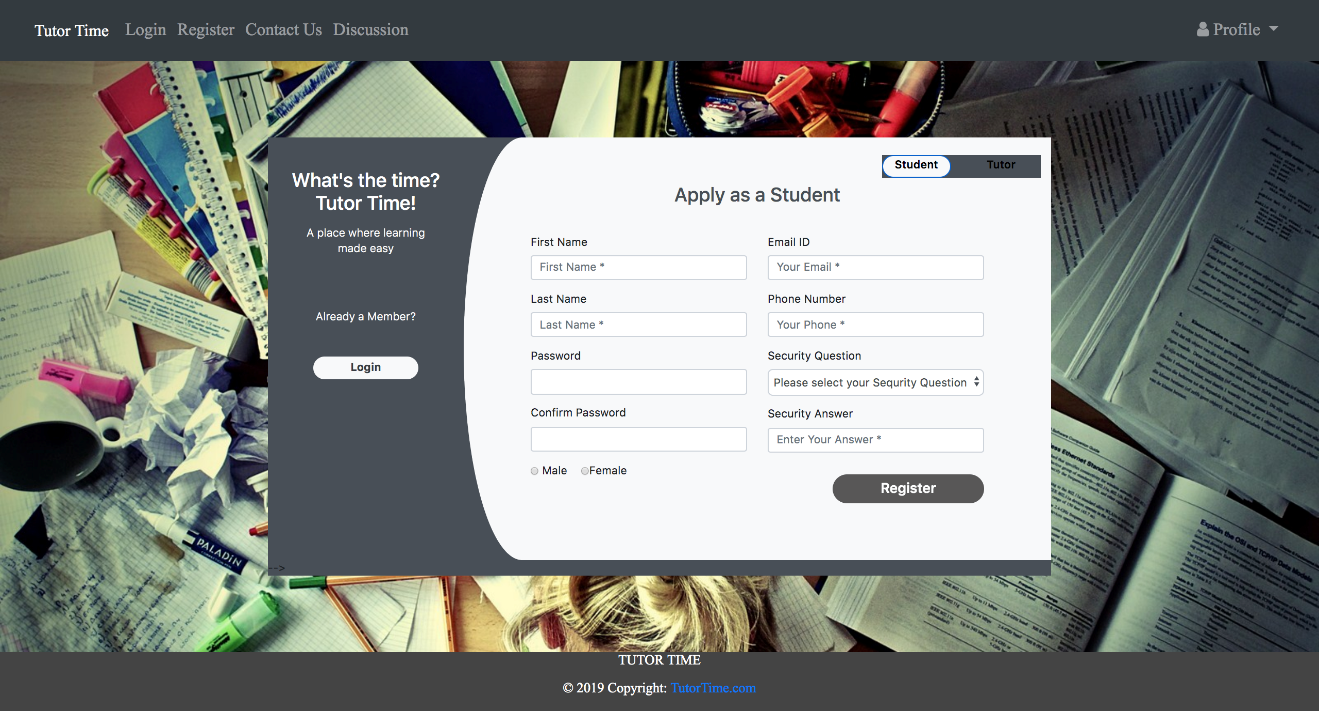


Figure 15: Registration page for students

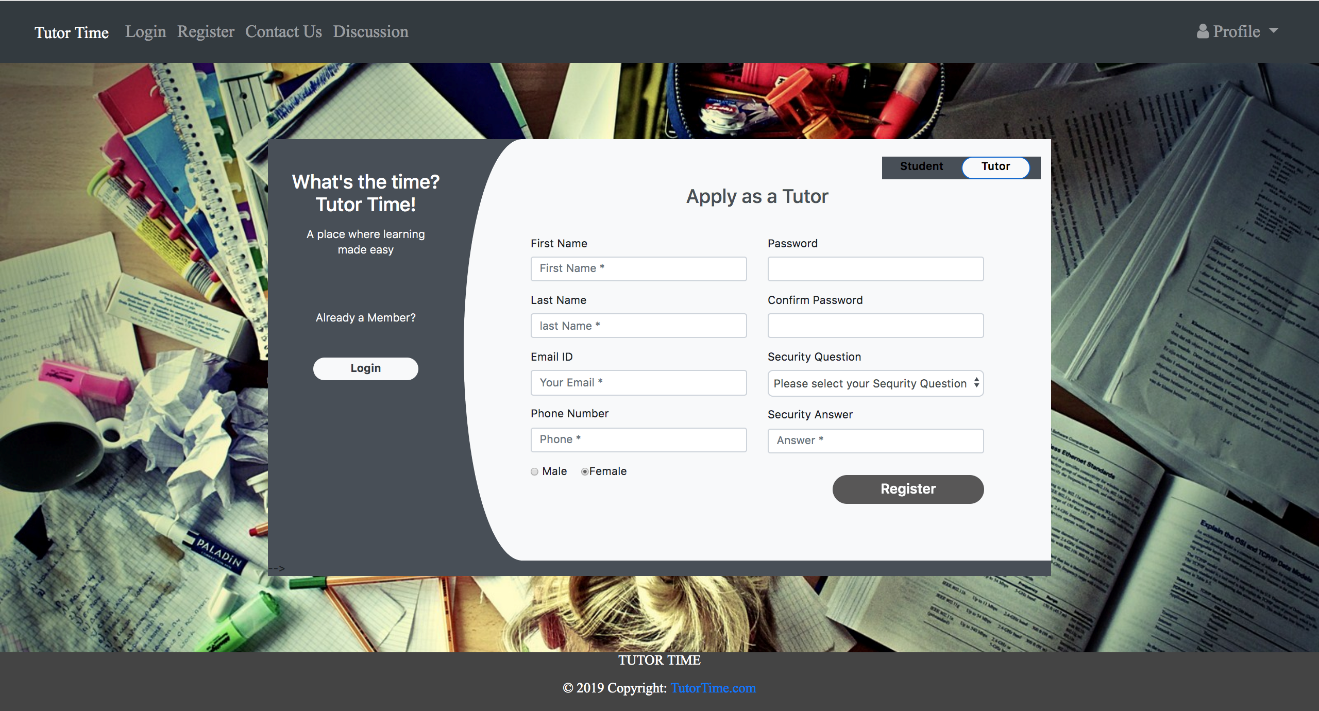


Figure16: Registration page for tutors

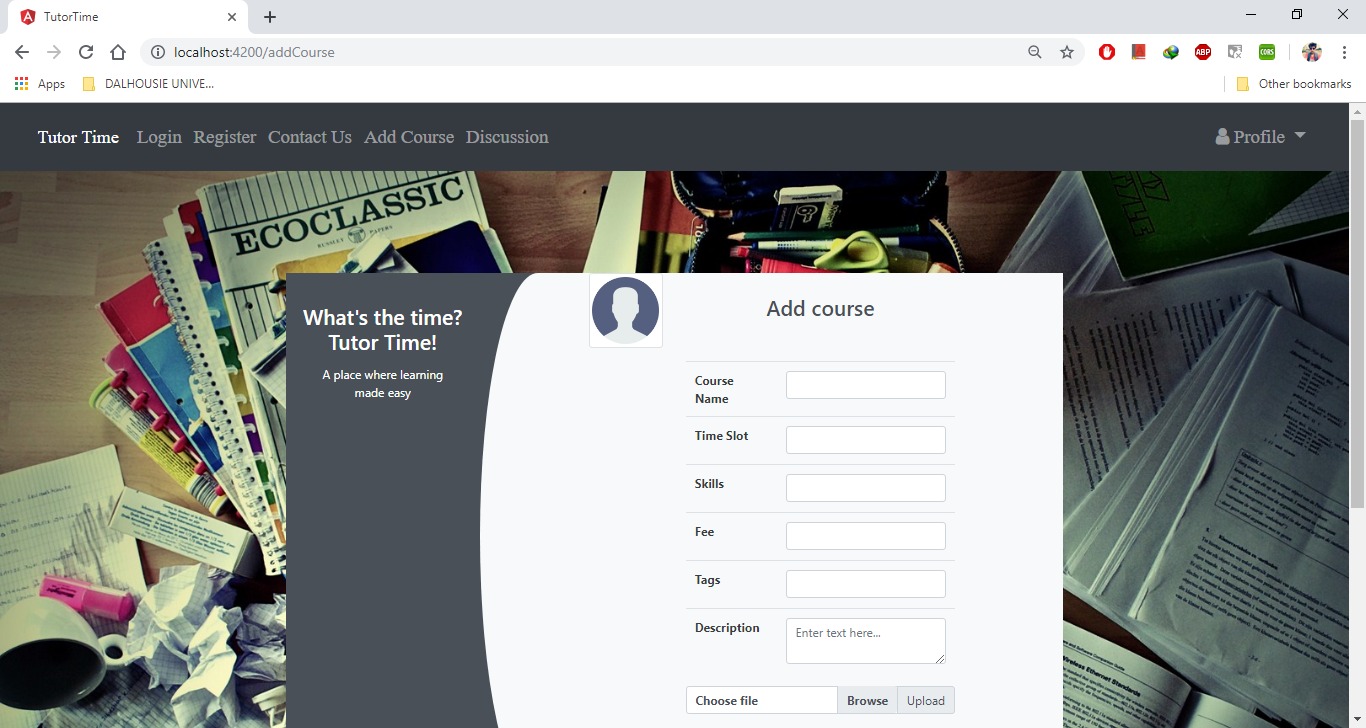


Figure 17: Add course by tutor

*Figure 17* represents the user interface for adding course. Tutors will use this and add a course to their schedule.

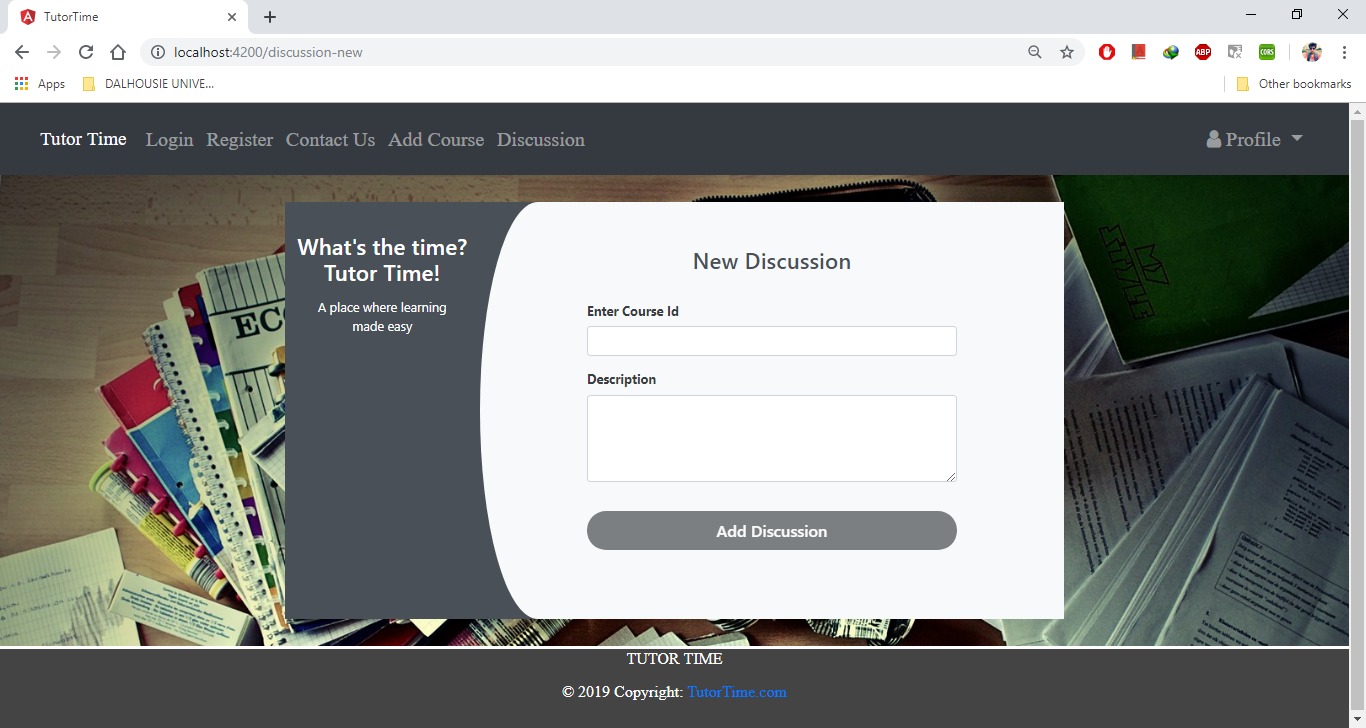


Figure 18: Discussion forum for students and tutors

*Figurs 18 and 19* show the discussion forum which will be same for both tutor and student. Users can start a new discussion and enter an existing one.

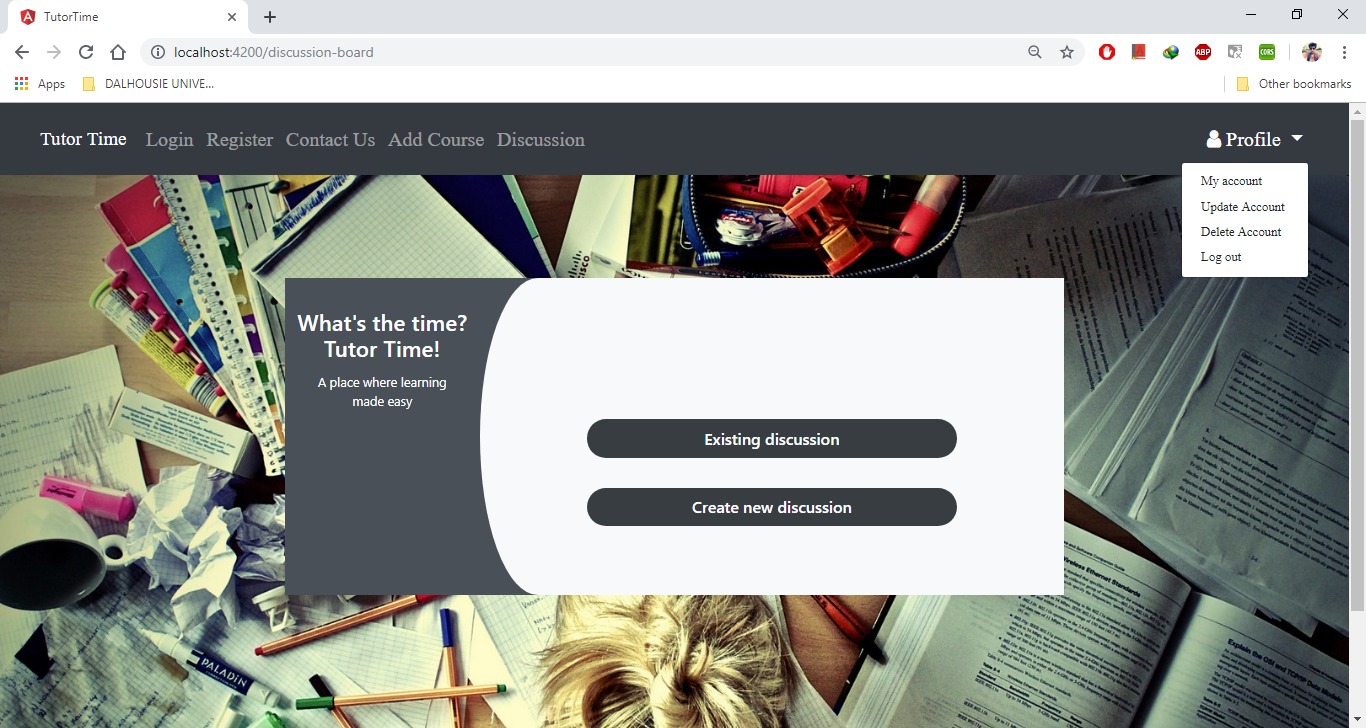


Figure 19: Feature to create or enter an existing discussion

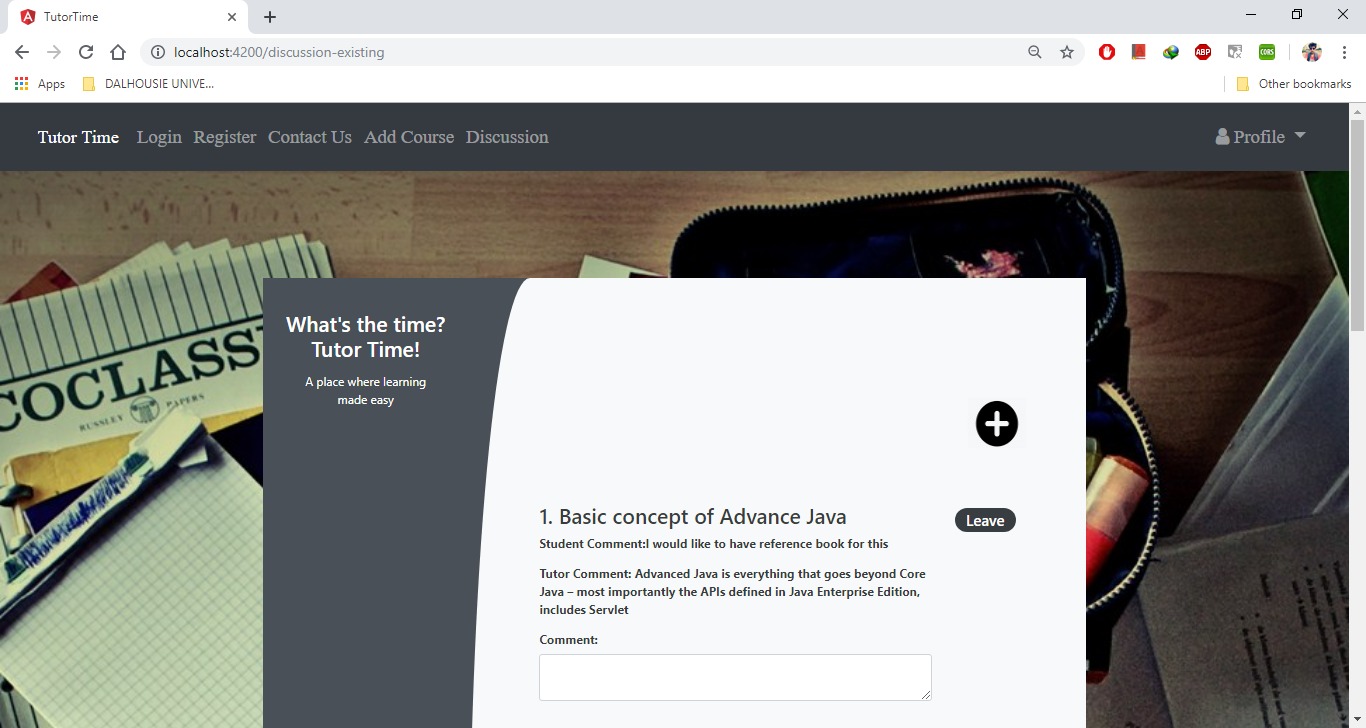


Figure 20: A list of discussion threads

*Figure 21* shows the design of webpage that will allow the users to delete their profile.

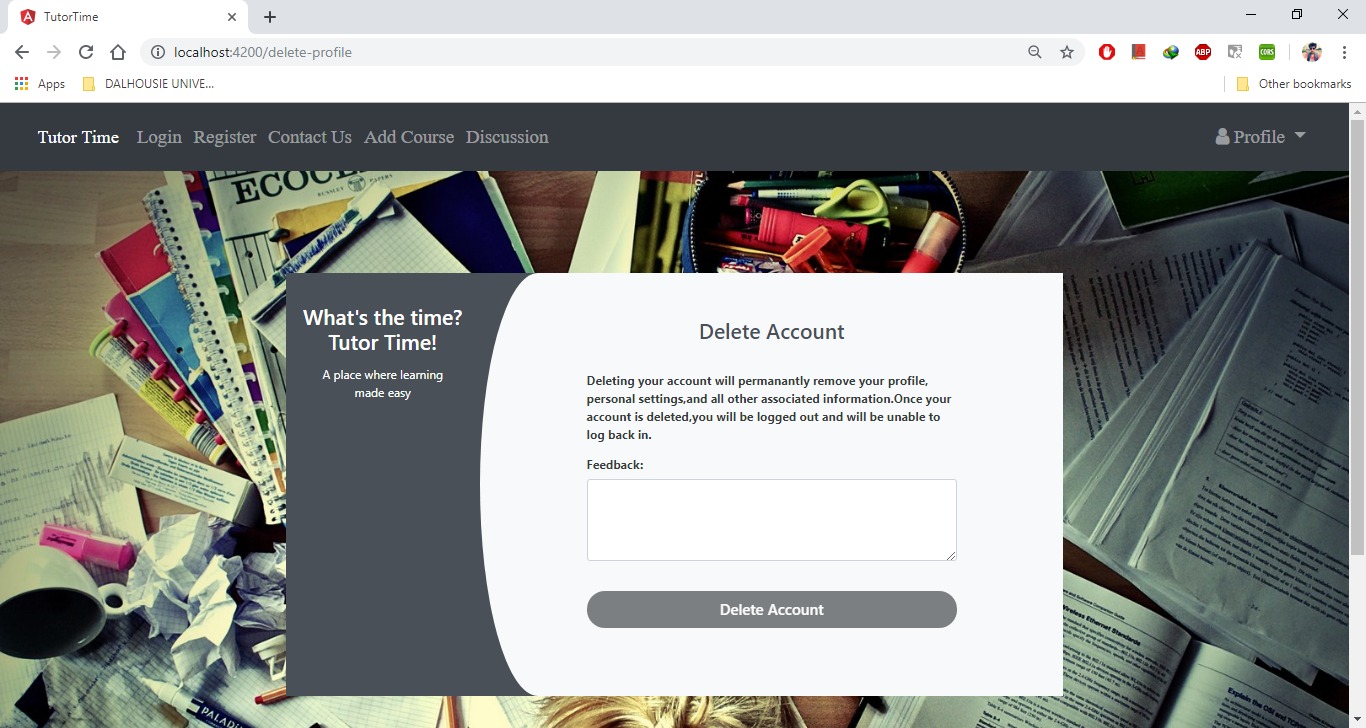


Figure 21: An option to delete account

*Figure 22* represents the webpage that will allow the users to change their passwords in case they forget the current password.

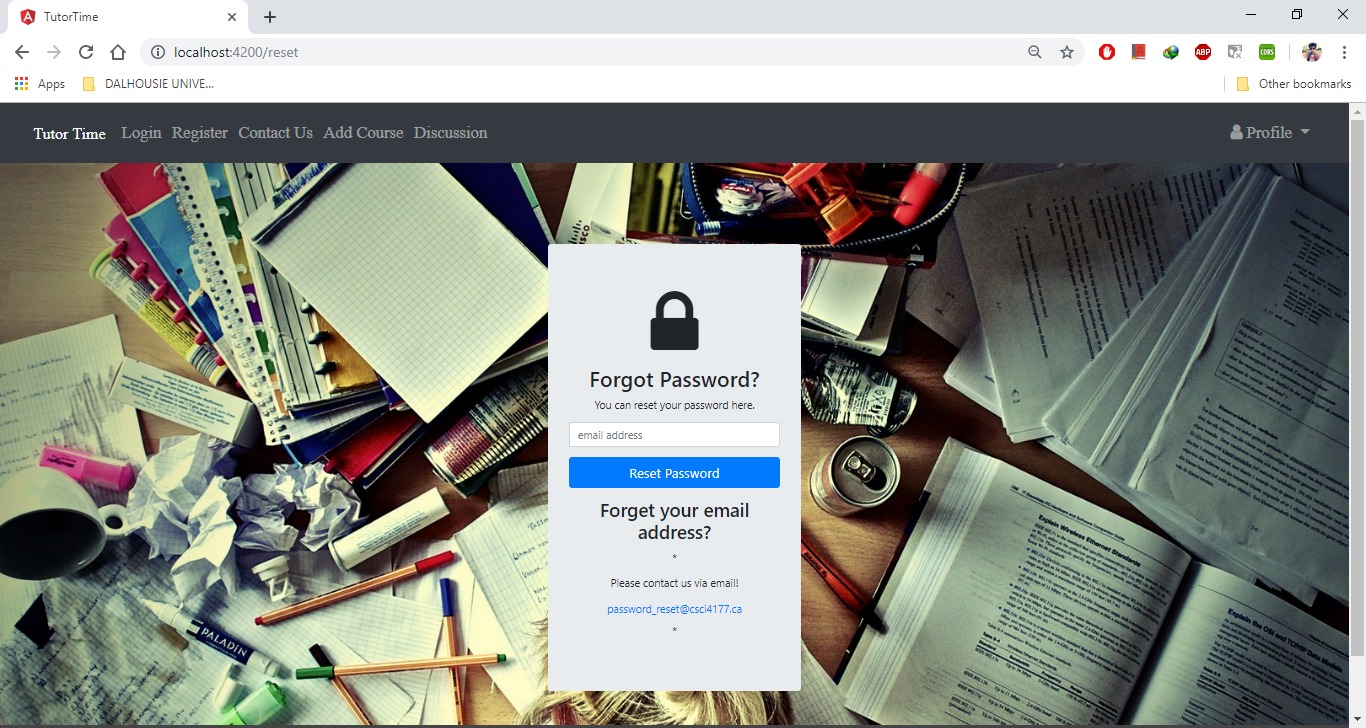


Figure 22: Reset password

While designing the web pages we considered Nielsen’s Heuristics. The design is simple and simple and easy to navigate. We have used Angular, HTML, CSS and W3C principles to design all the webpages.

## User Experience

We had developed the below scenario in the 1st assignment and our application would work similarly if any such instance takes place. Below is the scenario and its cases *[14]*.

**Scenario 1**: Here we shall consider the 1st Persona. Suppose, Bob is a student and he wants to use our application to brush up his knowledge and secure a position in a good IT company.

**Normal Use Case:**

1. Bob will open the application and register himself.
2. The system verifies if all fields are filled.
3. The system sends a confirmation email to customer.
4. Bob will confirm his account.
5. Bob is logged in.
6. Bob can then search for tutors using the key skills.
7. Bob will enroll with a tutor.
8. The system sends an enrolment confirmation email.
9. Bob can browse more courses and tutors.
10. Bob will log out.

**Alternative Use Case:**

1.1. The email ID is already registered with the application.

1.2. Bob can perform limited search prior registering.

2.1. Error, all \* marked fields need to be filled.

2.2. Passwords do not match.

5.1. Bob can update to premium account.

6.1. Bob can update his profile.

6.2. Bob can edit his profile details and can delete his profile.

6.3. The user can provide feedback to the tutor.

7.1. The key skills do not match with any tutor.

7.2. Matching tutors found.

7.3. Seats for the course are full.

7.4. Bob can cancel the enrolment.

8.1. Error while confirming enrolment.

8.2. Student successfully enrolled to the course.

Below is the task flow diagram as per the above scenario.

In the task flow diagram i.e. *Figure 23*, Bob is on the landing page of Tutor Time. Bob flows the following steps to use the application.

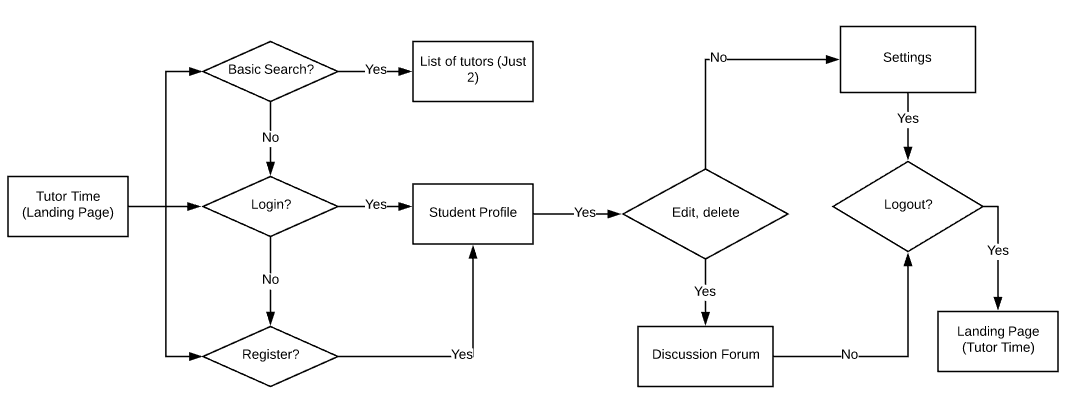


Figure 23 [3]: Task flow diagram of Bob (Student)

**Scenario 2**: Next, we have considered a scenario from tutors’ point. For instance, Robert is tutor who is retired. Teaching is his profession and he wishes for have a platform using which he can continue his passion and make some money.

**Normal use case**

1. Robert will open the application and will either register or login.

2. Accordingly, the system will open the selected page.

3. The user will start registering himself.

4. The system will send an email to verify user account.

5. The user will login to the application.

6. The system will open tutor dashboard.

7. The tutor can make changes to their profile.

8. The will logout.

**Alternate use case**

1.1. User can register.

1.2. User will login.

3.1. System will check for validations.

3.2. User registered successfully.

4.1. The entered email address is wrong.

4.2. Email received and account verified.

6.1. Tutor can update their schedule.

6.2. Tutor can update, edit and delete their profile.

8.1. User will be taken to the landing page.

*Figure 24* shows the task flow diagram from Tutors perspective i.e. Robert will follow the below steps and go through the application.

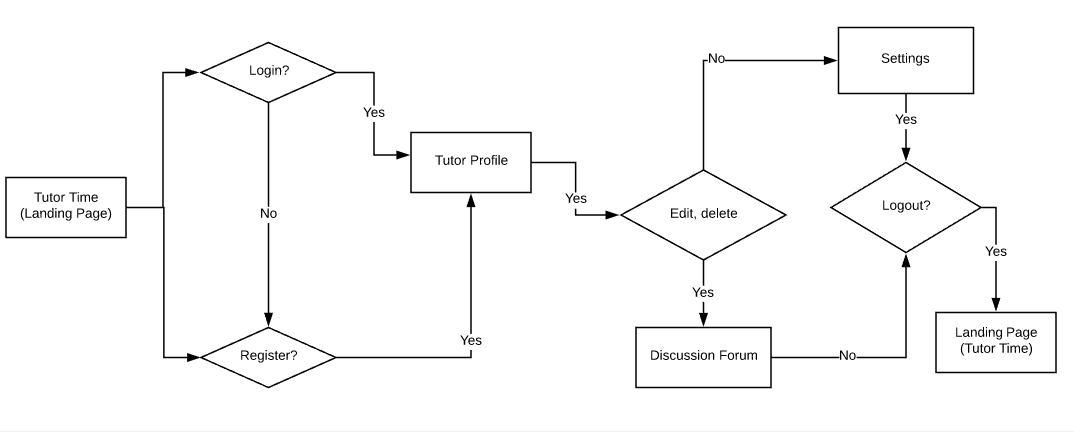


Figure 24 [3]: Tutors task flow diagram

# Asset Inventory

We will be working on MEAN stack i.e. M – Mongo, E – Express, A – Angular, N – Node [4]. All components of the MEAN stack support programs that are written in JavaScript, MEAN applications can be written in one language for both [server-side](https://en.wikipedia.org/wiki/Server-side) and [client-side](https://en.wikipedia.org/wiki/Client-side) execution environments. In addition, we will use JavaScript when needed explicitly by our application. We are using media i.e. we are providing the user a functionality wherein the user can add an avatar for themselves. We are bootstrapping our application and making sure that it is responsive. We have used Plugins like Material Icon theme, Angular essentials, Prettier, W3C validator. Angular essentials [5] this extension grabs on the latest and important plugins that are essential for an angular project. Material icon [6] theme, we are using to make the solution icons look materialistic. Prettier [7], we are using this plugin to make the code aligned properly and neatly.W3C validator [8], we are using this plugin to validate the HTML based on W3C.

|  |  |
| --- | --- |
|  |  |

# Group Roles

*Table 3* shows the list of roles and the name of team members to whom the roles are assigned.

Table 3 – Group Roles

|  |  |  |  |
| --- | --- | --- | --- |
| Sr No. | Name | Primary Role | Secondary Role |
| 1. | Devanshu Srivastava | Database manager | Tester and Git Maintainer |
| 2. | Sneha Sagar | Lead designer | Tester and documents manager |
| 3. | Bhumi Patel | Tester | Designer and backend developer |
| 4. | Tirtha Modi | Lead developer | Tester and documents manager |
| 5. | Ganrong Tan | Backend developer | Supporter |

Table 3: Group role assignment

# REFERENCES

1. [[3]](#footnote-4)Tool used for sitemap:

[1] g. GlooMaps, "GlooMaps - Visual Sitemap Tool", *Gloomaps.com*, 2019. [Online]. Available: <https://www.gloomaps.com/> [Accessed: 11- May- 2019].

1. Tool used for wireframes and low fidelity prototypes:

[2] "Balsamiq. Rapid, effective and fun wireframing software. | Balsamiq", *Balsamiq.com*, 2019. [Online]. Available: <https://balsamiq.com/> [Accessed: 13- May- 2019].

1. Tool used for task flow.

[3] "Log In", *Lucidchart.com*, 2019. [Online]. Available: <https://www.lucidchart.com/documents#docs?folder_id=home&browser=icon&sort=saved-desc> [Accessed: 18- May- 2019].

1. "MEAN (software bundle)", *En.wikipedia.org*, 2019. [Online]. Available: <https://en.wikipedia.org/wiki/MEAN_(software_bundle)> [Accessed: 16- Jun- 2019].
2. "johnpapa/vscode-angular-essentials", *GitHub*, 2019. [Online]. Available: <https://github.com/johnpapa/vscode-angular-essentials> [Accessed: 16- Jun- 2019].
3. "Icons - Material Design", *Material Design*, 2019. [Online]. Available: <https://material.io/tools/icons/?style=baseline> [Accessed: 17- Jun- 2019].
4. "Prettier - Code formatter - Visual Studio Marketplace", *Marketplace.visualstudio.com*, 2019. [Online]. Available: <https://marketplace.visualstudio.com/items?itemName=esbenp.prettier-vscode> [Accessed: 16- Jun- 2019].
5. "W3C Validation - Visual Studio Marketplace", *Marketplace.visualstudio.com*, 2019. [Online]. Available: <https://marketplace.visualstudio.com/items?itemName=Umoxfo.vscode-w3cvalidation> [Accessed: 17- Jun- 2019].
6. "Tryit Editor v3.6", *W3schools.com*, 2019. [Online]. Available: <https://www.w3schools.com/howto/tryit.asp?filename=tryhow_css_user_rating> [Accessed: 16- Jun- 2019].
7. [7]"HTML Snippets for Twitter Boostrap framework : Bootsnipp.com", *Bootsnipp.com*, 2019. [Online]. Available: <https://bootsnipp.com/snippets/dlZAN> [Accessed: 18- Jun- 2019].
8. P. LTD, "First Tutors - Home tutors for private tuition throughout the world", *Firsttutors.com*, 2019. [Online]. Available: https://www.firsttutors.com/. [Accessed: 15- Jun- 2019].
9. "Find A Tutor Canada", *Find A Tutor Canada*, 2019. [Online]. Available: https://findatutor.ca/. [Accessed: 15- Jun- 2019].
10. "Superprof - The Tutoring Community", *Superprof*, 2019. [Online]. Available: https://www.superprof.ca/. [Accessed: 19- Jun- 2019].
11. Referred assignment 1 of two group members i.e. Sneha Sagar and Devanshu Srivastava.

This section consists of remaining wireframes and pages.

* 1. Wireframes



Figure 25: Hamburger showing the dashboard for tutor

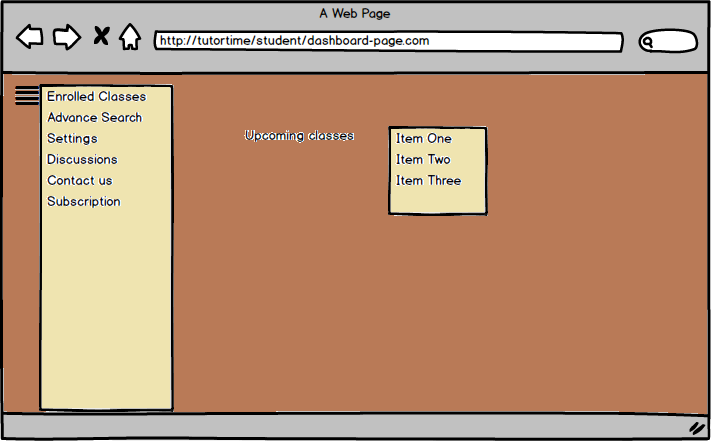


Figure 26: Hamburger showing the dashboard for student

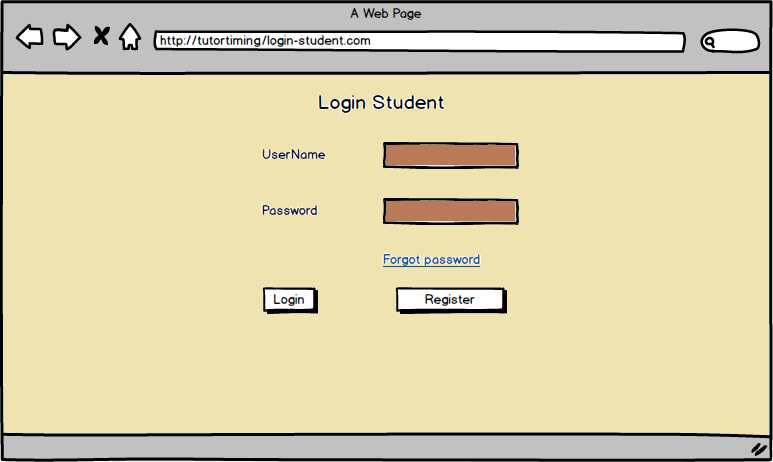


Figure 27: Student login

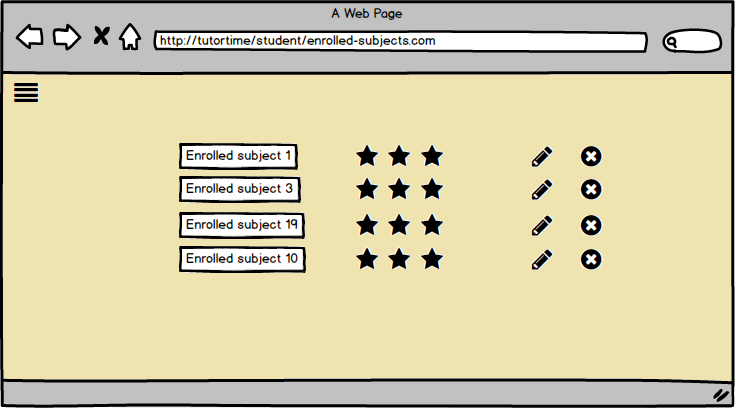


Figure 28: Course enrollment by students

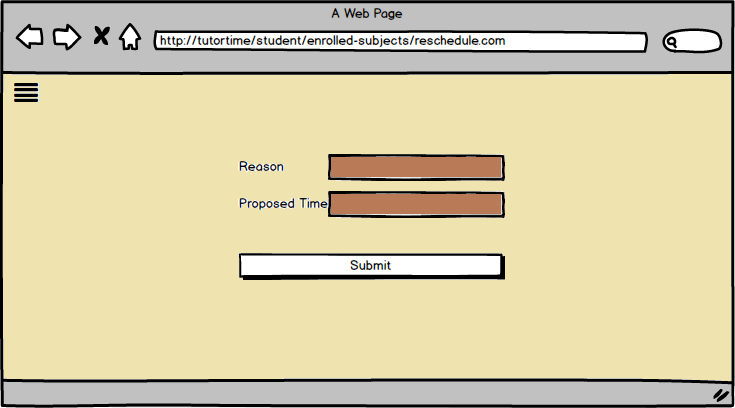


Figure 29: Rescheduling class



Figure 30: Discussion thread for students and tutors



Figure 5: Contact us/ troubleshoot page

* 1. Code referencing

We have referred some code from external sources to create the form of webpages. The form created on login, [9] registration and other pages were referenced form an external source [10]. We have modified the class names, the form fields and styling of the form as per our requirement. We have modified most of the code referenced from external sources.

1. Meticulous- showing great attention to detail; very careful and precise. [↑](#footnote-ref-2)
2. Navigations- the process or activity of accurately ascertaining one's position and planning and following a route. [↑](#footnote-ref-3)
3. Websites[1] [↑](#footnote-ref-4)