|  |
| --- |
| **SEDCO**  **CVM TEAM**  **C# Windows Form Documentation**  **Survey Question Configurator**  **Mahdi Suleiman**  **Eng. Majd Jaara**  **Date: 25th May 2022** |

Table of Contents

[Students' Property Right Declaration and Anti-Plagiarism Statement i](#_Toc61328405)

[Abstract v](#_Toc61328408)

[Chapter 1: User 1](#_Toc61328409)

[Chapter 2: Admin & Installation 5](#_Toc61328410)

[Chapter 3: Developer 10](#_Toc61328411)

[References 35](#_Toc61328417)

# Abstract

Survey Question Configurator is a C# Windows Form Application for configuring 3 types of questions

1. Smiley Question, where you have to set:
   1. Question Order
   2. Question Text
   3. Number of smiley faces
2. Slider Question, where you have to set:
   1. Question Order
   2. Question Text
   3. Start Value
   4. Start Value Caption
   5. End Value
   6. End Value Caption
3. Stars Question:
   1. Question Order
   2. Question Text
   3. Number of smiley stars.

You can Add, Edit, View and Delete any question.

All changes are reflected to a Microsoft SQL server database.

**Keywords:**

Microsoft, C#, SQL, Windows Form

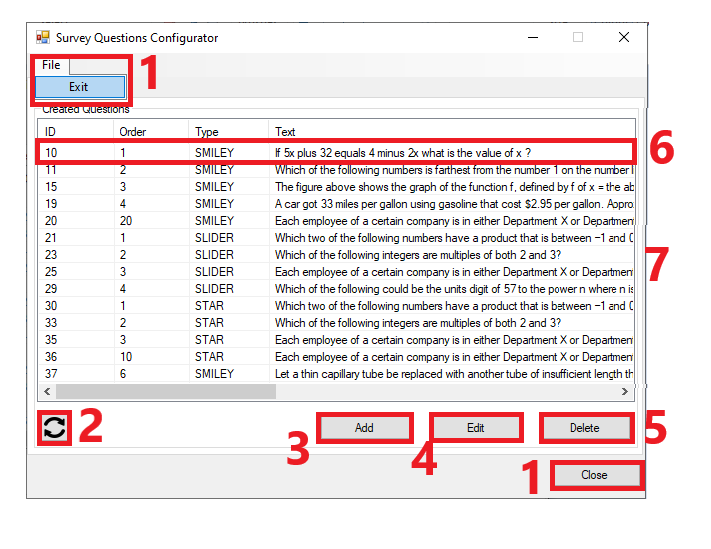
# Chapter 1: User

**1.1 Introduction**

The main idea of this project is to allow its user to view, add, edit and delete 4 types of the previously mentioned questions.

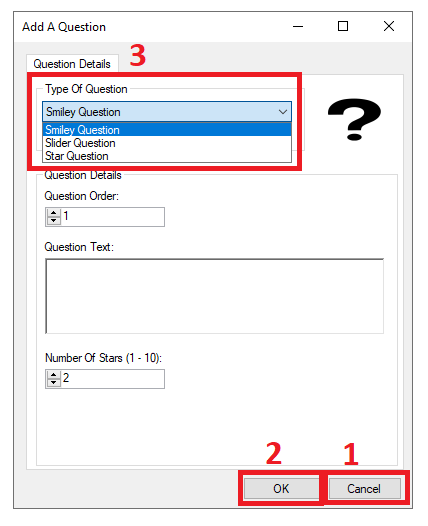
**1.2 How to use**

* First you have the main dialog. It has multiple controls and buttons divided into numbered sections:



1. Close the application
2. Refresh questions list.
3. Add a question (Opens a new dialog).
4. Edit a **selected** question (Opens a new dialog).
5. Delete a **selected** question (Shows a conform dialog).
6. **Double clicking** a question open it’s **editing** dialog.
7. Horizontally resizable dialog.

* Second, if you press “**Add**” you will open “Add A Question” Dialog:

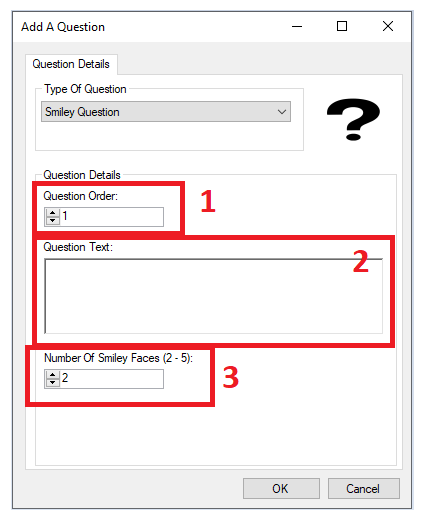


Also divided into 3 section:

1. Close the application.
2. Continue with your selections and inputs.
3. Choose the type of question you want to add.

* There is 3 types of questions to add.

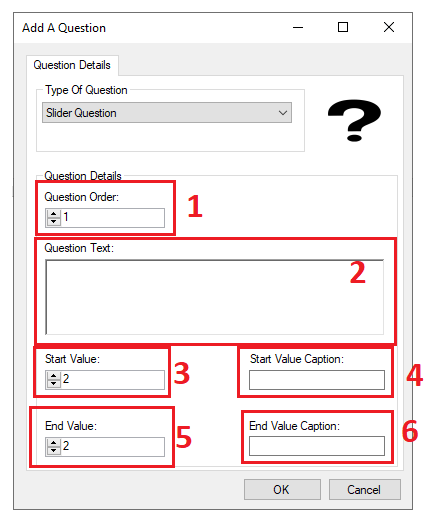
1. Smiley faces question



* 1. Order of the question.
  2. Text of the question.
  3. Number of smiley faces (2-5).

\*Notes:

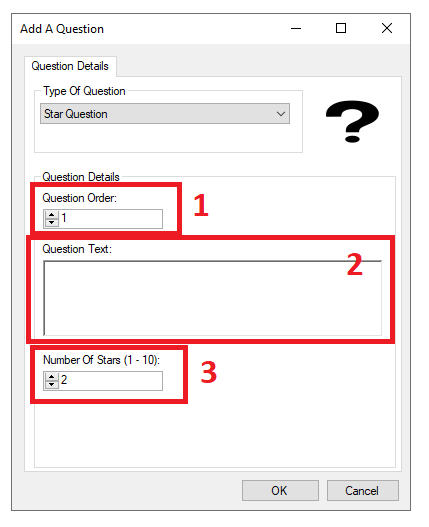
1. Questions orders can **NOT** repeat for every given question type.
2. Question text can **NOT** be empty.
3. Number of smiley faces **MUST** be in the range of (2-5) inclusive.
4. Slider question



* 1. Order of the question.
  2. Text of the question.
  3. Star value of the slider (1-99).
  4. Start value caption.
  5. Star value of the slider (2-100).
  6. Start value caption.

\*Notes:

1. Questions orders can **NOT** repeat for every given question type.
2. Question text can **NOT** be empty.
3. Start value caption and end value caption fields can **NOT** be empty.
4. Start value **MUST** be larger than end value.
5. Stars question



* 1. Order of the question.
  2. Text of the question.
  3. Number of stars (1-10).

\*Notes:

1. Questions orders can **NOT** repeat for every given question type.
2. Question text can **NOT** be empty.
3. Number of stars **MUST** be in the range of (1-10) inclusive.

a bit about third-party applications, A third-party application is created by a developer that is not the manufacturer of the device that the application runs on or the owner of the web page that provides it. A Third-party application can be permitted or prohibited by the device manufacturer or web page owner. For example, Android phones come with their own camera application, but there have been camera applications from third party developers that can offer advanced features such as a “self-timer” and editing tools. Until the phone manufacturer implements the same features in its own application, third-party applications offer advantages to the user, but the Play Store can contain other applications that Google approved for use on Android phones but did not develop. Those apps are third-party applications. Facebook can permit some applications that it did not develop on its own to function on its social media website. These are third-party applications.

**1.3 Aims and objectives**

We aim to build a website built on the need of university and college students which gives them an opportunity to easily connect them with a desired tutor fast and easy and can give tutors another way to communicate and find new students to teach which by itself is a good investment and away to improve their income to the point it is the go-to application where students and tutors can start using our website to be part of their educational purposes. We’re aiming that tutors and we included can have profits in return of using our website.

**1.4 How the problem been solved till now**

Locally we are unaware of a solution to the problem but internationally there are some websites the offers the same services as our website.

**1.5 Main solution idea**

Our main solution idea is a website that can connect students to tutors and vice-versa. A website is able to revolutionize this whole this process and that is what we are aiming for in this project.

**1.6 Key Technical Details**

Since we did not learn how to build web applications or websites at the university, it was necessary for us to learn this skill on our own and find a suitable programming language to build web applications, after we did a comprehensive research, we found out that the most framework we will be comfortable working with is ASP.NET Framework. ASP.NET is an open-source and free framework for building web applications; it has been developed by Microsoft and released in January 2002. Combining it with ASP.NET MVC 5 it allows you to create dynamic and rich websites. This means that you can easily develop, upgrade and change the web application with ease. The programming language that is associated with ASP.NET is named C# it is also was created by Microsoft, and we already learnt it in university so that will make things a bit easier to us. Using Microsoft SQL also helped us and made our work a bit easier because it is also developed by Microsoft so combining all of this with Visual Studio Integrated Development Environment our development experience was good and smooth.

For the Front-End side we used HTML5, CSS and JavaScript to design and develop our “view” for our user and structure it and creating functions to communicate with Back-End side using JavaScript. Adding on top of these technologies we used Bootstrap that help us to create dynamic, rich, resizable and mobile friendly web pages that looks and feels good on every platform.

**1.7 List of contributions claimed by the project**

Our project aims to make the process of finding a good tutor with a decent rating for university students a fast and reliable one. Thus, making their learning experience an easier, less of a hassle one especially for foreign students that they have language barrier and communicating is hard for them so it is hard for them to keep asking their colleagues for good tutors and keep making phone calls with them all day.

**1.8 High Level Diagram of Our Website**

Here is a high-level diagram of our project in Figure. 1 below.

Diagram

Description automatically generated

**Fig. 1 Use Case Diagram**

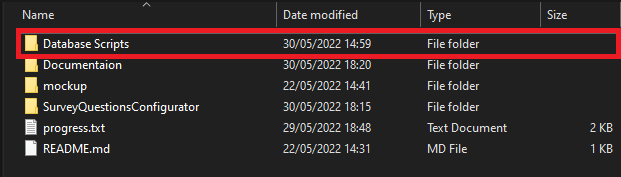
**1.9 Summary of report structure**

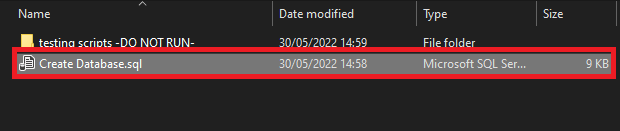
We attempted to describe the project in full in this report, covering all aspects of design, execution, purpose and context, and technologies employed. However, we have provided a user manual with easy designed examples to assist users in utilizing the suggested technology. We prepared a quick summary of the core idea of the problem and the solution in the abstract. We go over the design and application components in further detail in the introduction. We explained the significance of the challenge we addressed in the backdrop. We discussed each feature in detail in the design part, as well as providing screen shots and writing about design options and specifications. We discussed the programming languages that were utilized and the platform that was utilized to implement the project in the implementation part, as well as each function that was utilized in depth. The relationship between all of the developed components is addressed in the outcome and discussion chapter. We also spoke about the application's strengths and weaknesses. We included charts in the project management section that describe the chronological path we went through to complete this assignment. In conclusion and future work, a quick summary of the entire project is presented, as well as a list of prospective extra works as a feasible future effort.

# Chapter 2: Admin & Installation

**2.1 Create & Prepare Database**

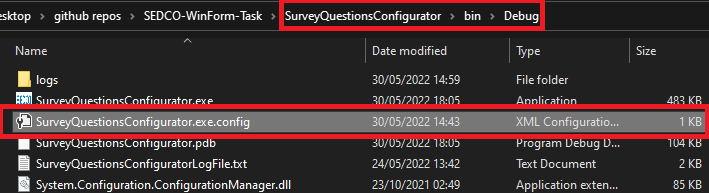
Run Database Script:

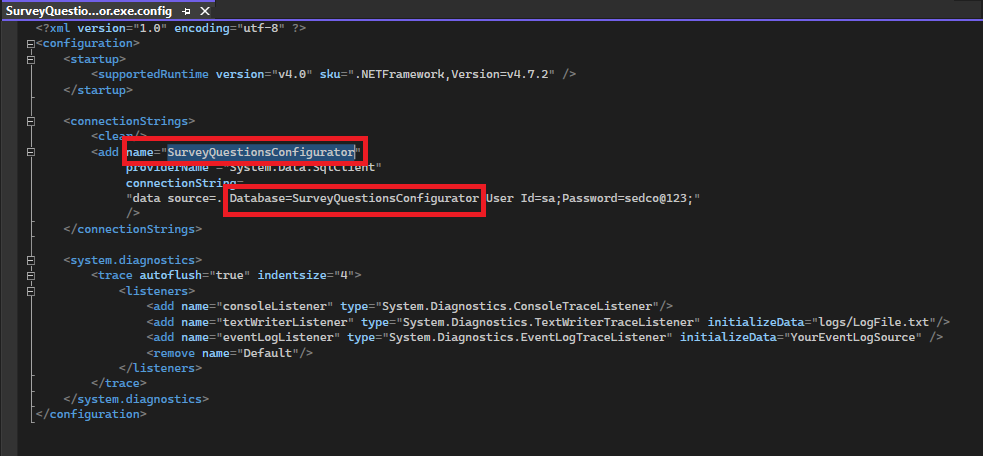




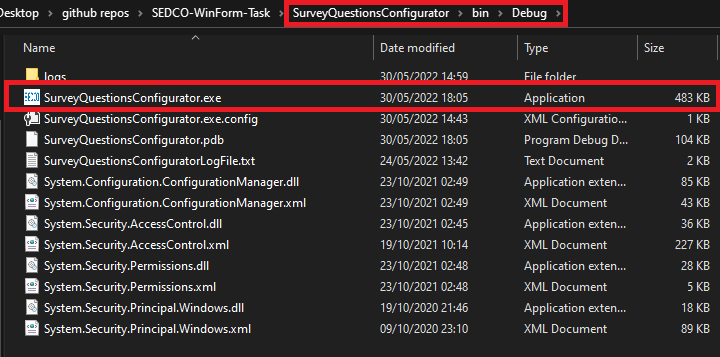
**2.2 Check Application’s Configuration File**

Make sure that “name” and “Database” attributes matches Database actual name in order for the application to start and work as intended.



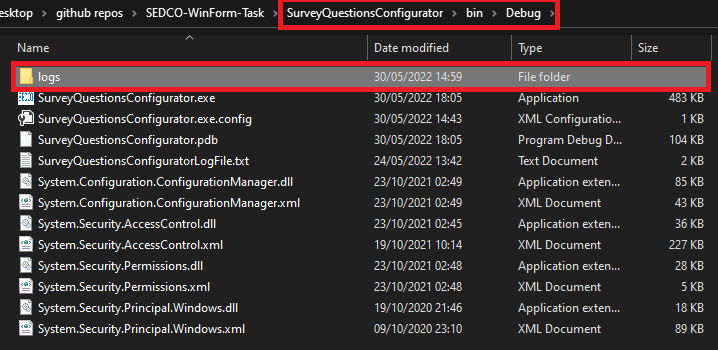


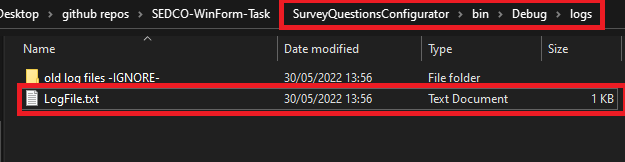
**2.3 Run Application’s Executable File**



**2.4 Error Logs**

Error Logs can be found here.

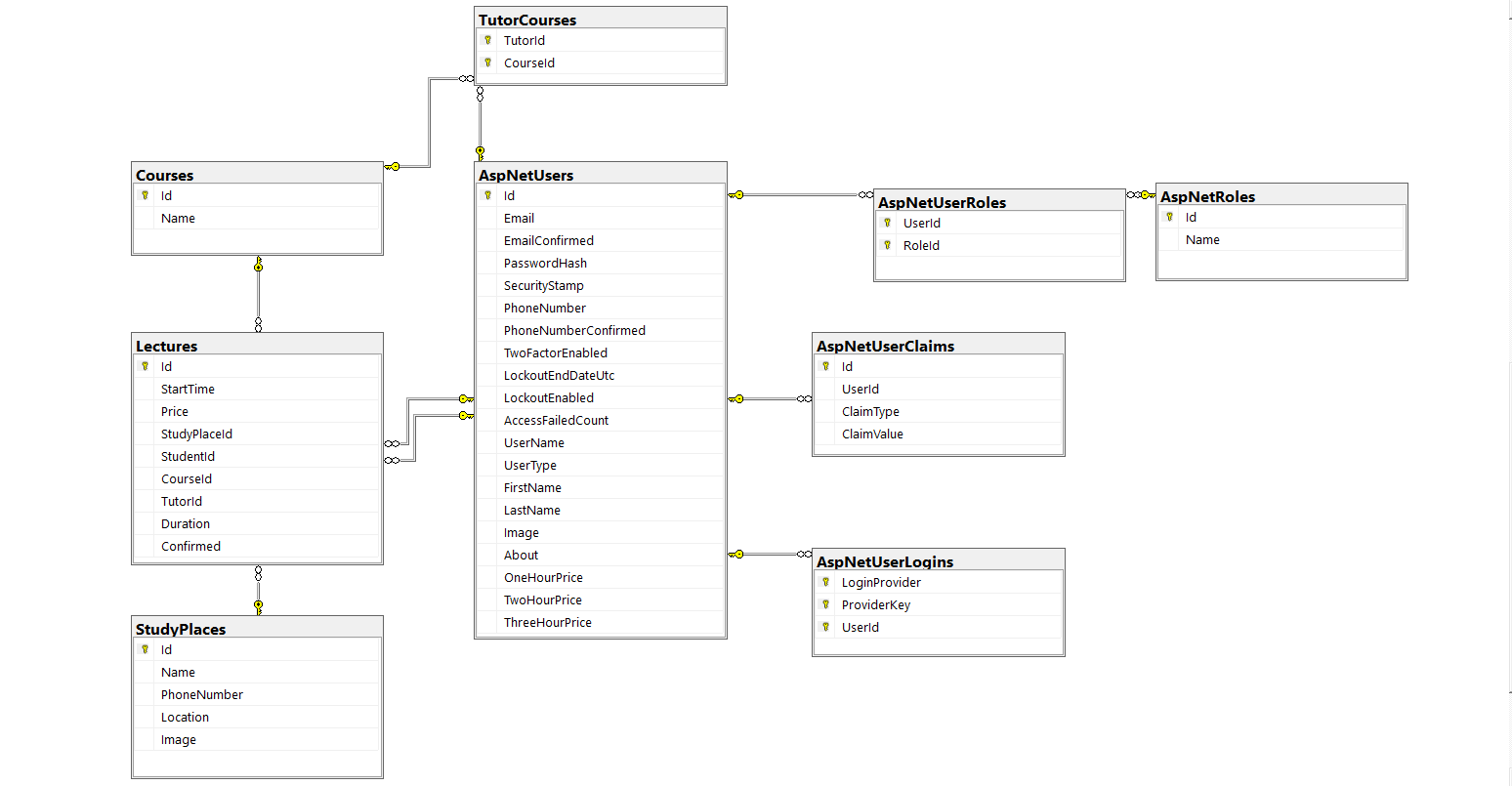




# Chapter 3: Developer

**3.1 Design Overview**

In general, in this project tutor finder we aim to achieve many goals, in order to do that we tried our best to facilitate things to make it easier to achieve our goal of the project. Our website design is tailored around people who are in a universities or colleges and looking for good tutors who will give them some extra lecture in a specific subject. This section is about illustrating the design work using a chart that describes the relation between our database tables, later sections shows languages we used to accomplish it, then we explained the functionality it in details in result and implementation sections. **Figure. 4** below shows our Database diagram.



**Fig. 4** Database diagram

**3.2 Design Details**

**3.2.1 Design Specifications**

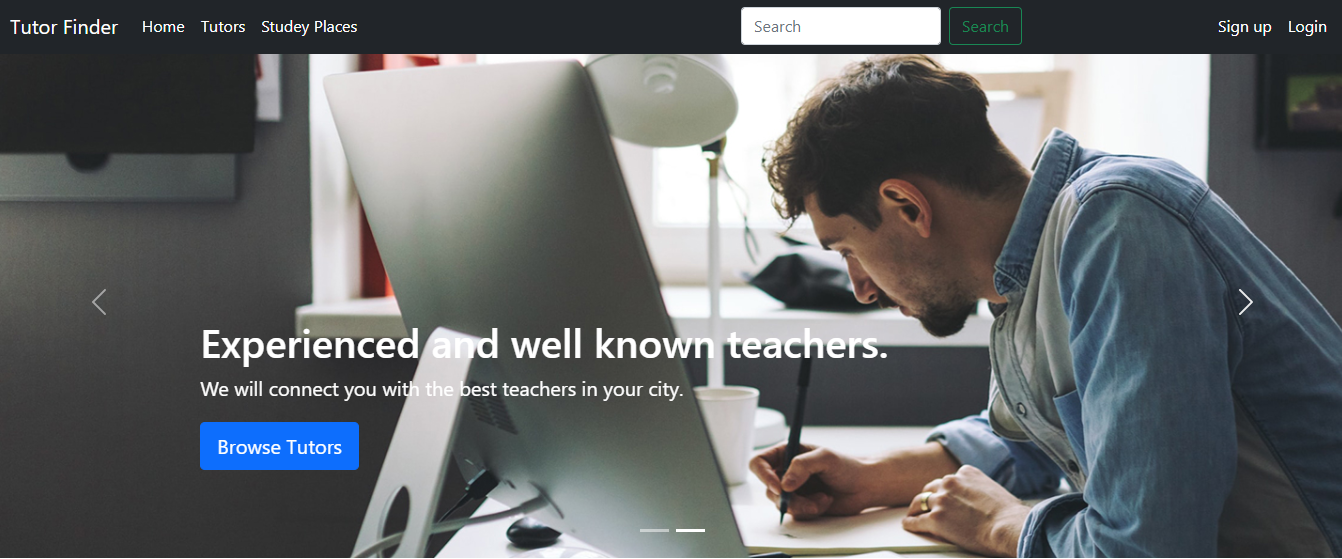
**Table. 1** below shows our design specifications.

**Table. 1** Design Specifications

|  |  |
| --- | --- |
| **Aesthetic Factors** | * Convenient to use. * Basic color and layout. |
| **Cost** | * For us, hosting website services costs. * For students, the cost of lectures. * For tutors, the have around (15%) fees. |
| **Environment Factors** | * N/A |
| **Function** | * This project should allow students in remedial teaching. |
| **Manufacturing** | * It needed approximately 4 months to finish the design. |
| **Material** | * Visual studio. * Good knowledge in HTML, CSS, JavaScript, ASP.NET, C# and Database. * Internet connection. |
| **Safety** | * Available safety tools from Microsoft. |
| **Size** | * It is a website so you do not to install anything except a web browser. |
| **Customers** | * This project is aimed for students who are looking for good tutors. |

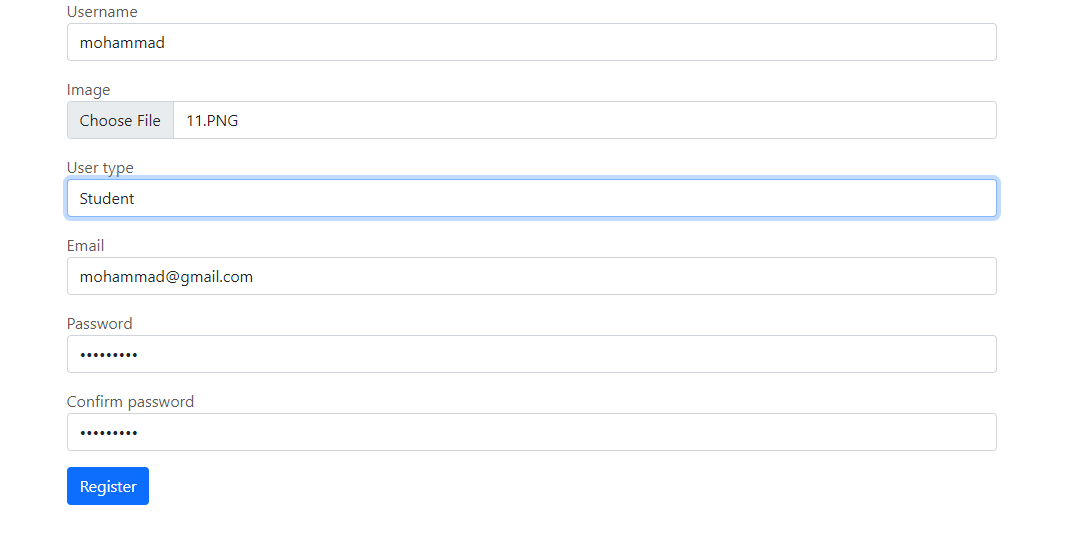
**3.2.2 Design Process**

In this section, we will describe the process of registration for students and tutors, how can students book lectures, and how the tutor accepts the order of student. First In **Figure. 5**, we are looking at Home Page for our project. The user can sign in, search for courses, explore tutors and study place.



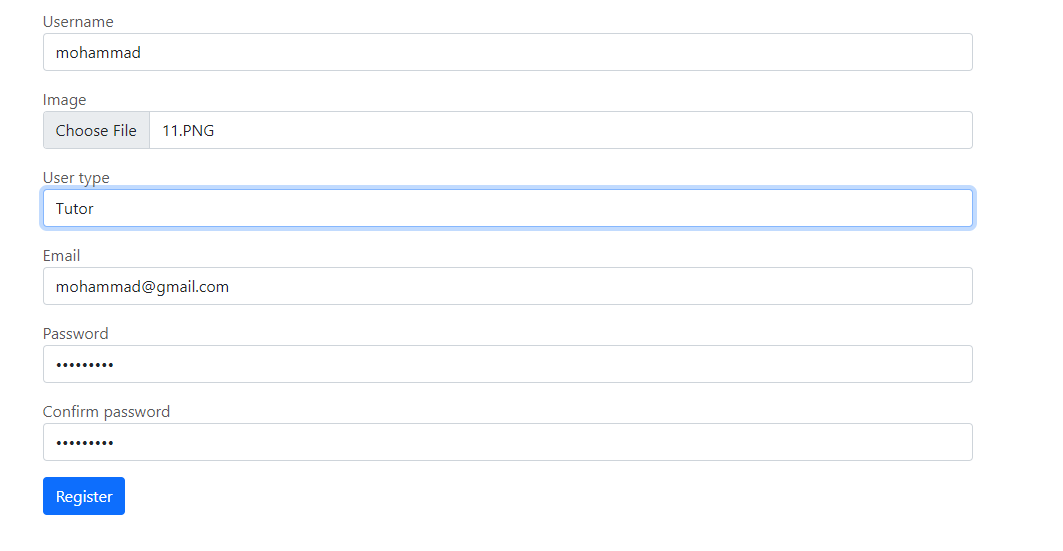
**Fig. 5** Home Page

Next in **Figure. 6**, we are looking at our student sign-up page.



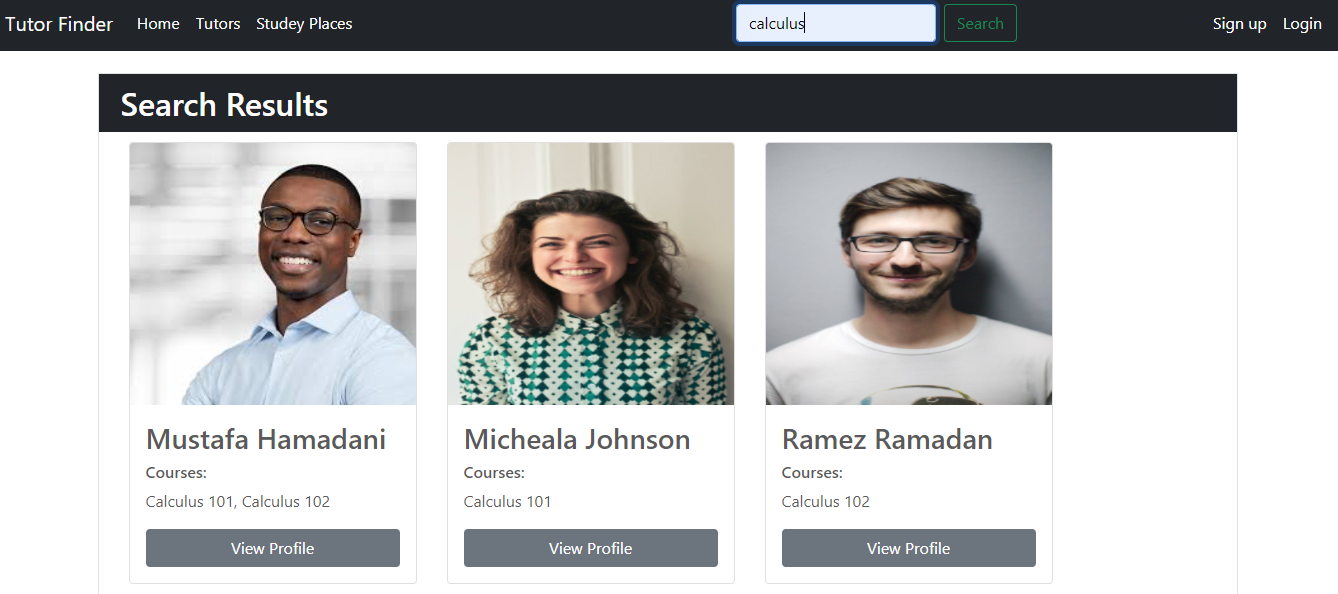
**Fig. 6** Sign up in student role

And **Figure. 7** shows tutor sign-up page.



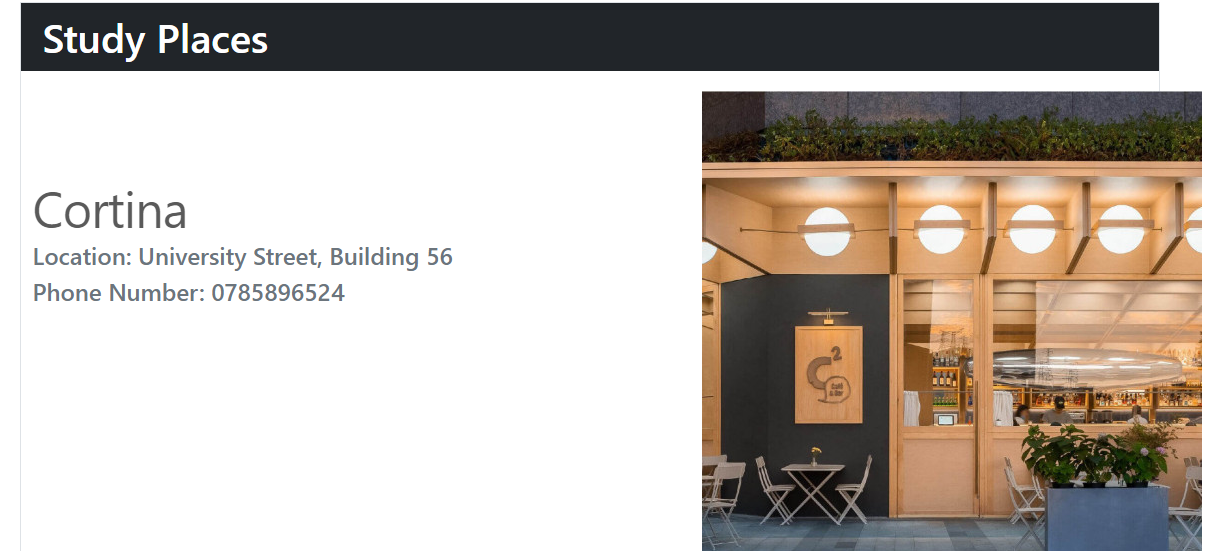
**Fig. 7** Sign up in tutor role

Next in **Figure. 8**, we can see our list of tutor page with a functioning search bar.



**Fig. 8** List ofTutors page and search bar functionality

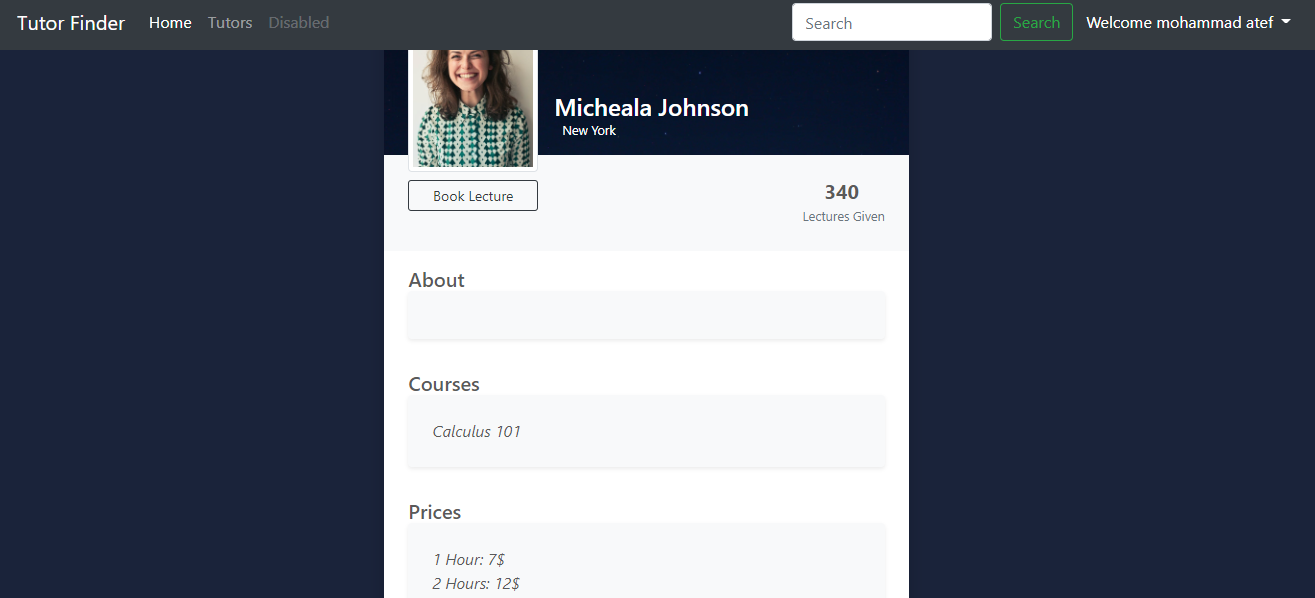
Next in **Figure. 9**, we can see our study places.



**Fig. 9** List of study places page

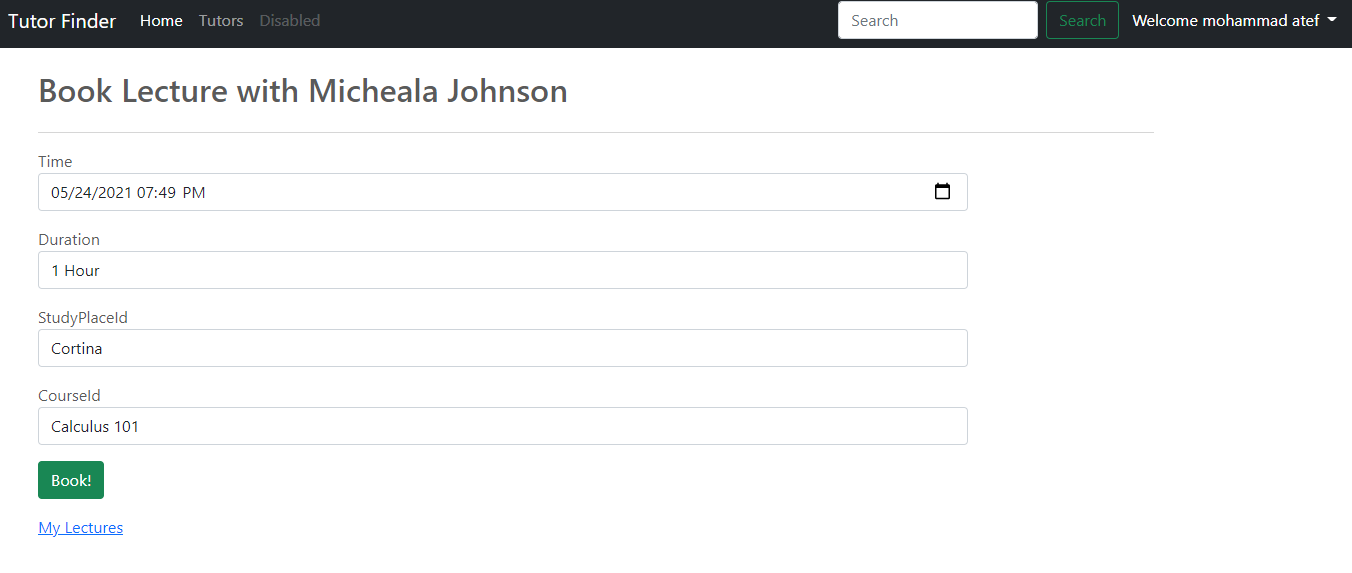
You can sign-in in our website by pressing on the sign-in icon. And you can choose the role that you want to sign-in with. After the user who registered and chosen "user type: student" now can book a lecture, and it is simply searching for the name of the topic and then the student finds a tutor when the student chose a tutor then the student can saw some details about the tutor and the price of an hour.

Next in **Figure. 10**, we are looking at tutor profile page.



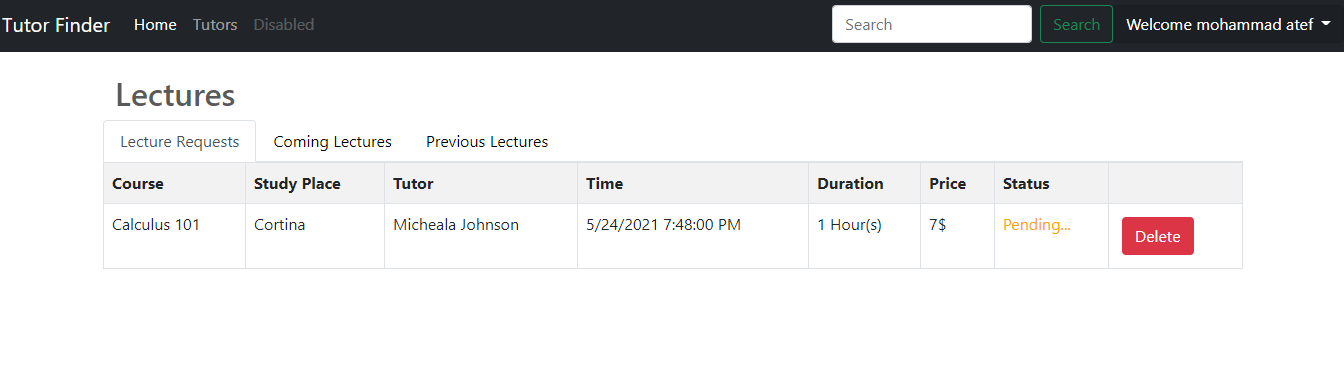
**Fig. 10** Tutor profile page

If the student is okay with the price of the lecture and want to book it, he can click on the **Book Lecture** button and then chose the suitable date, duration and study placed, then the student can click on the **Book** Button. **Figure. 11** below shows this process.



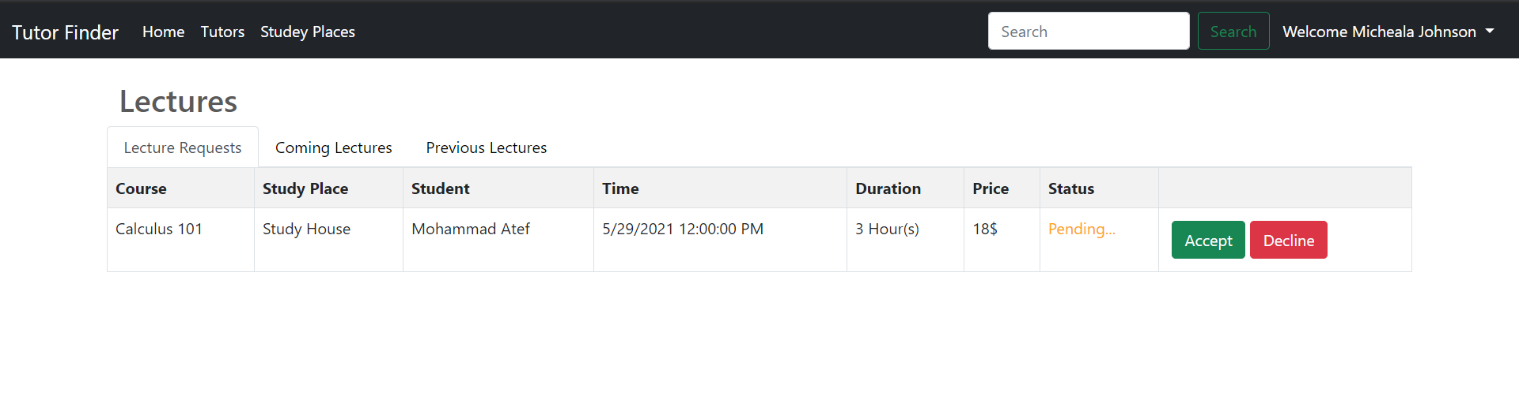
**Fig. 11** Book lecture page

Then, it will display a page containing the required lectures schedule and there is the status tab shows “Pending”. In this case, the student is waiting for the teacher’s response (acceptance or rejection). And the student can delete the lecture before any response from the tutor. As **Figure. 12** shows student’s lectures page.



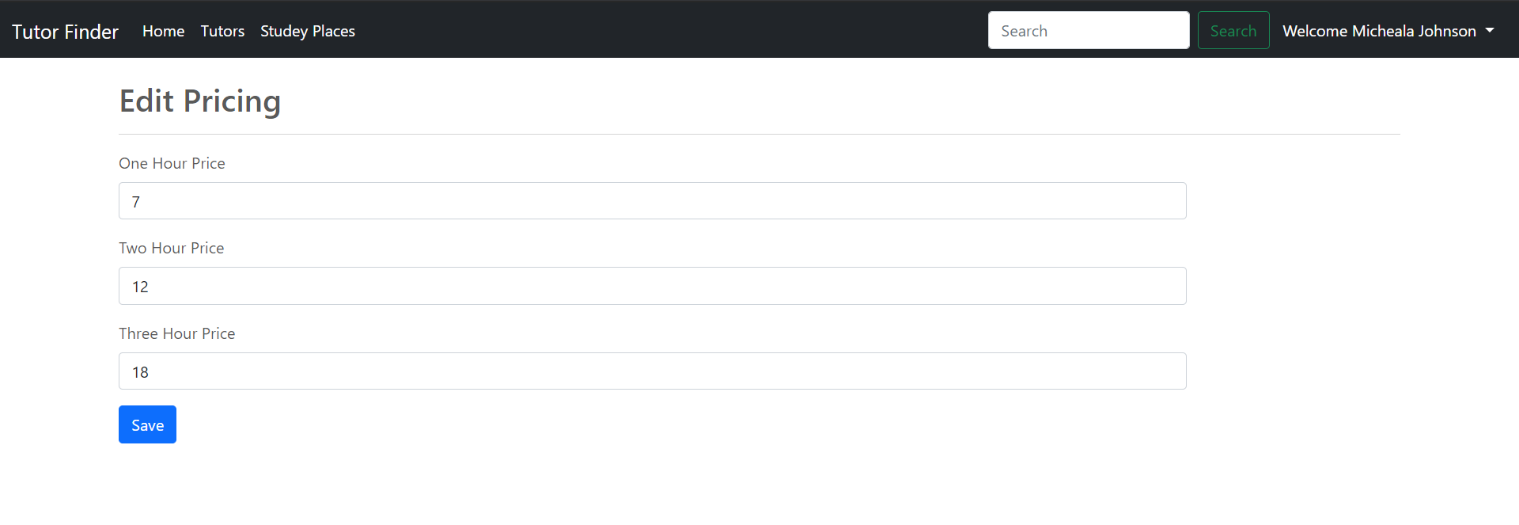
**Fig. 12** Lectures in student role

Now, as for the Tutor role in our website. The tutor can just sign in but without any privilege to insert any courses, because it is done by us (website admins), after we accept the tutor to our website, the tutor can open the lecture page to view, accept or decline student’s lecture request, and any action the tutor take is shown to the student real-time. As seen in **Figure. 13** below.

****

**Fig. 13** Lectures in tutor role

There are some advantages for tutors, and one of these features is to **edit the price** of the lecture according to what the tutor sees as appropriate. As shown in **Figure. 14** below.

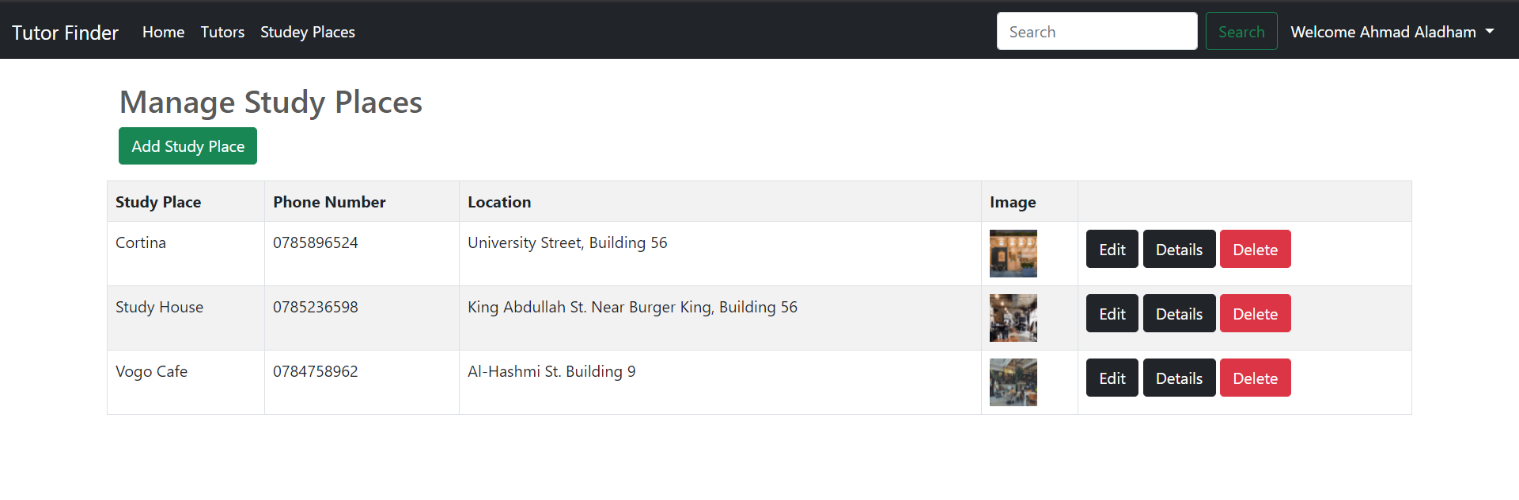


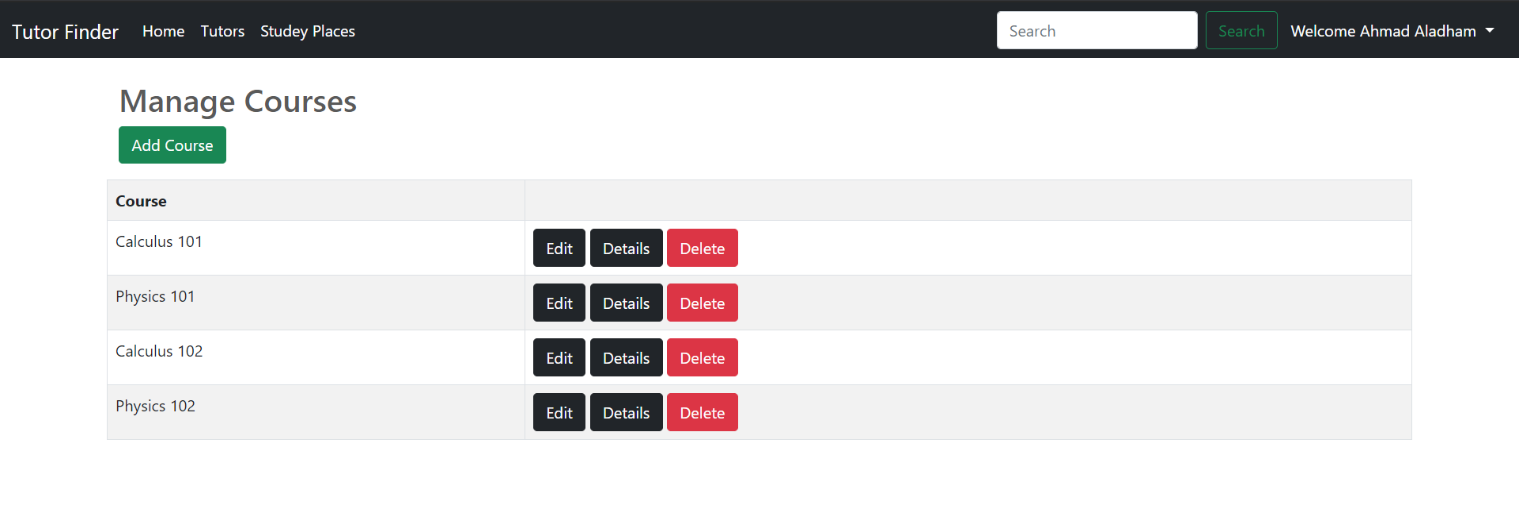
**Fig. 14** Edit pricing page

As for our role as Admins, we have special missions to do like.

1. Accept tutors sign-up request after we interview them personally.
2. Managing study places by inserting, deleting or editing their details.
3. Managing course by inserting, deleting or editing their details from tutors’ profile.

As seen in **Figure. 15** and **Figure. 16** below.

** Fig. 15** Manage study places page

****

**Fig. 16** Manage courses page

**3.2.3 Legal Aspects**

In general, the website does not need access to too personal information about users but, the website will request a profile picture from its users in order to fill it in “image of account” field. If the user did not allow access to it, the user cannot continue the process of registration because the “photo of account” is required.

**3.2.4 Design Constraints**

In terms of security, we used Microsoft Identity for our accounts’ management which gives access to users depending on their role so that no one can make changes on the courses, study places and tutor courses but the admin. We did not ask for any sensitive information during the registration process which requires only first and last name, username, email, password, and a profile picture we only share the tutors’ profile pictures with other website users, and we keep everything else hidden inside our database tables. All the users’ passwords are hashed inside of our database tables and NOT stored as plaintext inside of the users’ table which make it more secure and does not allow anyone even the database managers from reading our users’ passwords. Our project is a website application, and it does not have environmental concerns or real-life limited resources, the only resources we need are a server which can handle the requests of all the current users which is a number that can vary depending on the success of our website.

**3.2.5 Design Standards**

In this project, some international standards were used, here is list of the standers:

1. The IEEE Stander Association (IEEE-SA): Multiple Stander were used.
2. Standards Engineering Society (SES): SES Web Privacy Policy

**3.2.6 Design Alternatives**

An Alternative to our website would be the idea of connecting tutors with students completely online and from all around the world, but the main idea of our website is to connect people that are close to each other and living in the same city which will omit the communication problem caused by tutors and students having different languages, and will make it easier for the student to understand any topic, we are starting in a small area with a small set of courses and study places here in Irbid and then we will expand to include all universities and cities of Jordan. This way we will benefit these talented but unknown tutors who are just starting their careers by suggesting them to locally related students, which will not happen in the case of making our website connect people globally as that will lead to our students taking lectures only with a small amount of fancy, well-known teachers.

There are not many design alternatives when it comes to our project, the idea is straightforward; our website is meant to connect tutors who are struggling to find students to teach and vice versa, we made it easy to find any tutor or course by simply searching their name. We can add a mobile application along with the website which is not an alternative but an addition to make it even easier for our tutors and students to use and connect with each other.

**3.2.7 Safety Consideration**

The main idea of this project providing good tutors for student, we choose the best tutor with us and sure the tutor had a nice experience in teaching. There are many tutors give remedial courses but a big number of them had not to experience or not qualify to teach student, so we want not that the student falls into this problem.

**3.2.8 Design Considerations table.**

**Table. 2** below Shows our design specifications.

**Table. 2** Design Considerations

|  |  |  |
| --- | --- | --- |
| **Design consideration** | **Project application** | **Relevant location in report** |
| Performance | Finding best tutors in local places. | Multiple locations. |
| serviceability | The project mainly serves university and college students. | Section 2.2 |
| Economic | This did not affect our design. | Section 7.2 |
| Environmental | This did not affect our design. | Section 6.4 |
| Environmental  Sustainability | This did not affect our design. | Section 6.4 |
| Manufacturability | This is not design criterion. | N/A |
| Ethical | No ethical constraints and guidelines are knowingly violated. | Section 6.2 |
| Health and safety | This did not affect our design. | N/A |
| Social | This did not affect our design. | N/A |
| Political | This did not affect our design. | N/A |

# References

[1] **Careem**

<https://www.careem.com/>

[2] **Uber**

<https://www.uber.com/>

[3] **Wikipedia MVC**

<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

[4] **Microsoft ASP.NET MVC**

<https://docs.microsoft.com/en-us/aspnet/core/mvc/views/razor?view=aspnetcore-5.0#:~:text=Razor%20is%20a%20markup%20syntax,cshtml%20file%20extension>.

[5] **GoDaddy**

<https://ae.godaddy.com/>

[6] **GanttProject software**

<http://ganttproject.biz>