

***Capstone Project - Phase B***

***Teach-Me: A Web Application To Finding Private Tutors For Students***

*Project Number: 23-2-D-21*

**Supervisor:**  
 Prof. Zeev Barzilay

[zbarzilay@braude.ac.il](mailto:zbarzilay@braude.ac.il)

Ilan Vinerski

[ilanv145@gmail.com](mailto:ilanv145@gmail.com)

Orel Damti

[oreldamti1996@gmail.com](mailto:oreldamti1996@gmail.com)

Project Code:  
 **<https://github.com/OrelDamti/Teach-Me>**

**Table of contents**

[1. Introduction 3](#_Toc164782082)

[1.1. Scope of the project: 3](#_Toc164782083)

[1.2. Project’s stakeholders: 3](#_Toc164782084)

[1.3. Project’s Code: 3](#_Toc164782085)

[2. Project review and process description 4](#_Toc164782086)

[2.1. Project accomplishment description 4](#_Toc164782087)

[2.1.1. React.js 6](#_Toc164782088)

[2.1.2. React Router 7](#_Toc164782089)

[2.1.3. React Bootstrap 7](#_Toc164782090)

[2.1.4. Redux 7](#_Toc164782091)

[2.1.5. Redux Toolkit 8](#_Toc164782092)

[2.1.6. Node.js 8](#_Toc164782093)

[2.1.7. Express.js 8](#_Toc164782094)

[2.1.8. MongoDB 8](#_Toc164782095)

[2.1.9. Agile Development 9](#_Toc164782096)

[2.2. Research process description 10](#_Toc164782097)

[2.2.1. Challenges and solutions 12](#_Toc164782098)

[2.3. Testing process description 12](#_Toc164782099)

[2.3.1. Unit Testing 13](#_Toc164782100)

[2.3.2. Functionality Testing 14](#_Toc164782101)

[2.4. Results and conclusions 15](#_Toc164782102)

[3. User Documentation 16](#_Toc164782103)

[3.1. User’s guide: 16](#_Toc164782104)

[3.1.1. General description: 16](#_Toc164782105)

[3.2.1. Use Case Diagram: 26](#_Toc164782106)

[3.2.2. Class Diagram: 27](#_Toc164782107)

[3.2.3. Activity Diagram: 28](#_Toc164782108)

[3.2.4. Program structure and runtime configuration: 29](#_Toc164782109)

[4. References 30](#_Toc164782110)

**Abstract**

Many students are helped by private tutors during their studies. One of the problems is that the private tutors charge high prices, and the students during the degree do not have the money to finance it. So we thought of an idea where students would be helped by students from higher years who have already passed these courses with a high grade and charge a fair price for a private lesson. In this way, students will be able to earn money and earn a living during their degree, and on the other hand, students who need help with their studies will have the opportunity to receive this help at a price they can afford and improve their academic achievements.

# 1. Introduction

## 1.1. Scope of the project:

The "Teach-Me system" is a system that connects students who are looking for a   
private tutor for a certain course and a student who has already passed this course with a high score and wants to work as a private tutor and earn money during his studies. is a web application, Designed to solve the problem of allowing students to teach other students for a fair amount of money. This system is the ultimate tool for the stunts to teach the material they are good at. They will be able to choose which courses they want to teach, they will be able to add more courses later, delete courses they no longer want to teach. Furthermore, the students who will work as private tutors can receive opinions from students they teach. In addition, this solution makes it easier for students to find a suitable private tutor for them by the system showing them all the possible private tutors for a particular course and the ability to contact them directly, thus saving a lot of time.

## 1.2. Project’s stakeholders:

The potential stakeholders of our system are:

**• Registered users** **-** those who want to schedule a private lesson with a student.  
**• Students -** those who provide services as a private teacher.  
**• Guest users -** those who surf the Internet but do not order.

## 1.3. Project’s Code:

**<https://github.com/OrelDamti/Teach-Me>**

# 2. Project review and process description

## 2.1. Project accomplishment description

Our primary objective is to empower students to engage in peer-to-peer tutoring, eliminating intermediaries such as tutoring agencies and platforms. By connecting students seeking tutoring directly with those who have excelled in specific courses, our system ensures fair compensation for tutors and affordable access to quality education for learners.

In the Teach-Me System, there are four different roles: unregistered user, registered user (student), tutor, and admin.

The Teach-Me System's flow is comprised of distinct components, which we've organized into the following sections:

**Student User -Searching Teacher For Specific Course:**

When arriving at the Home Page of the Teach-Me System, the user is presented with one option for accessing the desired tutoring service:

In our system, each tutoring service is a combination of two entities: the course itself and the tutor who teaches it. Hence, we have created the optional path to reach the specific tutoring service.

The system presents a list of all available tutoring services, displaying the corresponding course image and name for each one. Upon selecting a specific course, the system displays a list of all tutors who teach it, including their image, name, rating, number of reviews, and the price at which they offer the tutoring service. Selecting the tutor's name directs the user to the specific tutoring service, while selecting the number of reviews brings the user to the reviews page, where they can add a review for the tutor.

**Student User -Choosing Teacher For Specific Course And Setting The Lesson:**

To complete the scheduling process, the user must be registered with the system. They have the option to register through a provided registration form.

At the beginning of the scheduling process, the user is required to select their preferred tutor and then choose the date and time for the class. After this, the system confirms the selection and sends an email to both the student and the teacher with details about the scheduled class, including the date, time, course, and contact information.

**Student User - Adding Reviews:**

Upon visiting the tutor's profile page, the user has the option to view all of the tutor's reviews and submit their own review through a form. When the form is submitted, it is saved in our Teach-Me database.

To submit a review for a specific tutor, the user must be registered with Teach-Me.

**Student User - View Upcoming Lessons:**

In the Teach-Me system, a registered student user has the option to view all of their upcoming lessons on their profile dashboard. This includes information such as the date and time of the lesson, the course being taught, and the name of the teacher.

**Teacher User Actions – Adding a New Course to Teach:**

The teacher user can view all existing courses on their dashboard.

To add themselves to teach a new course, the teacher selects the existing course they want to teach.

After selecting the course, the teacher fills out a form to specify details about teaching the course. This includes setting the price, specifying availability, and any other relevant information.

Once the form is filled out, the teacher submits the information.

The system then saves the details to our MongoDB database, and the teacher is added as the instructor for the selected course.

**Teacher User Actions – Managing Existing Course:**

If a teacher wants to remove himself from a course he is already teaching, he can enter the dashboard of his courses and manage from there the courses he wants to remove himself from by clicking on change his status

**Teacher User Actions – View Upcoming Lessons:**

In the Teach-Me system, a registered Teacher user has the option to view all of their upcoming lessons on their profile dashboard. This includes information such as the date and time of the lesson, the course being taught, and the name of the teacher.

**Admin User Actions – Managing Courses:**

The admin has the authority to add new course categories to the system. This process involves selecting a name for the category and uploading an image to represent it. Once added, the new category becomes available for teachers to use when adding their courses.

The user interface of our Teach-Me system was constructed using ReactJS, a widely-used JavaScript library for building dynamic user interfaces. Our system comprises multiple pages, and to seamlessly navigate between them, we implemented React Router. To elevate the design and functionality of our UI components, we integrated React Bootstrap.

For efficient state management, we employed the Redux toolkit. This ensures that our application maintains a consistent state across different components and pages.

To interact with our MongoDB database, we developed a RESTful API using Express.js and Node.js. This API serves as the bridge between our front-end and back-end, enabling smooth communication with the database and ensuring seamless data retrieval and manipulation.

### 2.1.1. React.js

React.js, an open-source JavaScript framework developed by Facebook, is widely recognized for its ability to swiftly and efficiently build interactive user interfaces and web applications.

In a React web application, components serve as the fundamental building blocks. These components, akin to independent Lego blocks, collectively form the entire user interface. Each component represents a distinct piece of the interface, enabling easy reuse and facilitating modular design.

React's primary role lies in managing the view layer of the application, much like the V (view) in the model-view-controller (MVC) pattern. It excels in rendering execution, encouraging developers to decompose complex UIs into smaller, reusable components. This approach enhances efficiency by combining JavaScript's speed with a more effective method of manipulating the DOM, resulting in faster rendering and the creation of highly dynamic and responsive web applications.

### 2.1.2. React Router

In web development, routing is the process of mapping a web URL to a specific resource within the application, allowing users to navigate between different pages and views.

React Router, a standard library for routing in React, simplifies this process by facilitating navigation among various components' views in a React application. It enables dynamic changes to the browser URL, ensuring that the user interface remains synchronized with the URL. With React Router, developers can create a seamless and intuitive user experience, enhancing the overall usability of their web applications.

### 2.1.3. React Bootstrap

React Bootstrap is a comprehensive package that provides a wide range of UI components, including buttons, text fields, containers, and more. These components come with modern styling and are designed to enhance the user experience. Additionally, React Bootstrap offers extensive user guides, making it simple for developers to integrate these components into their projects with ease.

### 2.1.4. Redux

Redux is a predictable state container designed for JavaScript applications. As applications scale, managing data flow and organization can become challenging. Redux addresses this issue by centralizing the application's state management with a single global object known as the Store. Its fundamental principles ensure consistency across the application, making debugging and testing more straightforward.

### 2.1.5. Redux Toolkit

Redux Toolkit is a collection of tools aimed at simplifying Redux development. It provides utilities for creating and managing Redux stores, as well as for writing Redux actions and reducers. This toolkit streamlines the process of working with Redux, making development more efficient and less cumbersome.

### 2.1.6. Node.js

Node.js is an efficient, single-threaded, open-source runtime environment used for building fast and scalable server-side and networking applications. It leverages the V8 JavaScript runtime engine and employs an event-driven, non-blocking I/O architecture, making it highly efficient and well-suited for handling concurrent connections and real-time applications. With its asynchronous nature, Node.js enables developers to create high-performance applications that can handle heavy workloads without sacrificing speed or responsiveness.

### 

### 2.1.7. Express.js

Express.js is a widely-used back-end web application framework designed for building RESTful APIs with Node.js. It is released as free and open-source software under the MIT License. Express.js simplifies the process of building web applications and APIs by providing a robust set of features and middleware. Its minimalist design and flexibility make it a popular choice among developers for creating scalable and efficient server-side applications.

### 2.1.8. MongoDB

MongoDB is a powerful, cross-platform document-oriented database program known for its versatility and scalability. It falls under the category of NoSQL databases and is designed to handle unstructured data efficiently. Using MongoDB, you can store data in JSON-like documents with optional schemas, allowing for flexibility in data modeling. The database supports various query operations, including field, range, and regular expression searches. Queries can also return specific fields of documents and include user-defined JavaScript functions. One of MongoDB's standout features is its high availability architecture, achieved through replica sets. A replica set consists of multiple copies of the data, with each member capable of acting as either a primary or secondary replica. By default, all write and read operations are performed on the primary replica, while secondary replicas maintain copies of the primary data through built-in replication. In the event of a primary replica failure, MongoDB's replica set automatically conducts an election process to select a new primary replica. Secondary replicas can also serve read operations, although the data they provide is only eventually consistent by default.

Overall, MongoDB offers robust features for handling diverse data types and provides scalability and high availability for modern applications.

### 2.1.9. Agile Development

During the development stage, we carefully evaluated various software development methodologies to determine the most suitable approach for our project. After thorough research and analysis of methodologies such as Agile and Waterfall, we concluded that Agile is the most appropriate methodology for our project.

Agile methodology is an iterative approach to project management and software development, breaking down projects into small, manageable pieces. These pieces are completed in work sessions known as sprints, which typically last from a few days to a few weeks. Sprints cover various stages, from initial design to testing and quality assurance (QA).

The main advantage of Agile development lies in its flexibility. The development team can respond to changes quickly, even at the last minute, and adapt without significant disruption. This flexible approach allows us to incorporate feedback and make adjustments throughout the development process, ensuring that the final product meets the needs of our users.

By adopting the Agile methodology, we can release an initial version of the system more efficiently and effectively. This iterative process enables us to deliver value to our users sooner and continuously improve the system based on their feedback.

## 2.2. Research process description

In our quest to develop the Teach-Me platform, we researched various platforms facilitating direct tutoring services between students and tutors. These platforms differed in their approaches to connecting learners with educators but all aimed to establish a direct connection between the two parties.

For instance, we looked at platforms like "LearnHub," which offers direct tutoring services online. However, many of these platforms lacked modern marketing and advertising strategies and did not provide an option for online bookings. Instead, students had to book sessions through phone calls or emails.

Other platforms, such as "TutorUp," provided weekly tutoring sessions at designated locations. This format allowed students to directly engage with tutors and receive personalized instruction.

After gathering information about these platforms, we realized that many of them relied on outdated marketing strategies and lacked efficient systems for managing tutoring sessions. This led us to explore ways to modernize marketing efforts and streamline the scheduling process for tutors and students alike.

To address these challenges, we designed a workflow and architecture aimed at improving the efficiency of scheduling sessions and enhancing marketing efforts for tutors. Our solution aims to provide a seamless experience for both students and tutors, allowing for easy scheduling and effective promotion of tutoring services.

As beginners, we started by understanding the fundamentals of web development, learning about HTML, CSS, and JavaScript. We then progressed to more advanced topics, such as server-side programming with Node.js, database management with MongoDB, and state management with Redux.

Learning Node.js allowed us to build the backend of our application, enabling us to handle HTTP requests, manage user authentication, and interact with the database. MongoDB introduced us to the world of NoSQL databases, teaching us how to store and retrieve data efficiently using JSON-like documents.

Understanding JavaScript was fundamental to our development journey, as it is the language of the web. We learned about JavaScript's asynchronous nature, its event-driven architecture, and its role in building interactive user interfaces.

Additionally, we explored Redux for state management, which proved invaluable in managing the application's complex data flow and ensuring consistency throughout the user experience. We learned how to structure our Redux store, define actions and reducers, and connect them to our React components.

Throughout our learning process, we faced numerous challenges and obstacles. To overcome these challenges, we turned to online resources, tutorials, and courses, such as those on Udemy, Stackoverflow, Youtube, which provided us with comprehensive guidance and hands-on practice.

As we gained proficiency in each technology, we applied our newfound knowledge to the development of Teach-Me, continually refining and improving our skills along the way. From version control with GitHub to code editing with VSCode, we utilized a range of tools to streamline our development process and collaborate effectively as a team.

Overall, our journey of learning from scratch was not just about knowing individual technologies but about understanding how they work together to build a cohesive and functional web application. With dedication, perseverance, and a passion for learning, we successfully developed Teach-Me into a platform that empowers students and tutors alike to connect, learn, and grow.

### 2.2.1. Challenges and solutions

During the development of our project, we faced several challenges that tested our problem-solving skills and pushed us to find creative solutions.

Firstly, we encountered difficulties with the database design. Initially, we planned to manage all user transactions and interactions within a single "Courses" table. However, as the complexity of our system grew, we realized the need for a more robust structure to handle various aspects such as user enrollments, lesson schedules, and tutor feedback. This led us to redesign our database schema, introducing additional tables such as "Enrollments" and "Lessons" to better organize and manage the data.

Secondly, we struggled with implementing real-time notifications for users. In a system like Teach Me, where students need to be informed about upcoming lessons and tutors need to be notified of new enrollments or lesson cancellations, implementing a reliable notification system was crucial. We initially attempted to use a third-party service, but we encountered limitations with customization and integration. To address this, we decided to build our own notification system using technologies like WebSockets and event-driven programming. This allowed us to provide real-time updates to users without relying on external services.

Thirdly, we faced challenges with user authentication and authorization. In our initial design, we had a basic authentication system that allowed users to sign up and log in, but we quickly realized the need for more granular access control and user roles. This required us to refactor our authentication logic and implement role-based access control (RBAC) to ensure that only authorized users could perform certain actions within the system.

Despite these challenges, we remained committed to delivering a high-quality product and continuously iterated on our solutions to ensure the success of the Teach Me platform. Through perseverance and collaboration, we overcame each obstacle and built a system that empowers students and tutors to connect, learn, and grow together.

## 

## 2.3. Testing process description

To thoroughly assess the system's functionality and reliability, we conducted two types of evaluation tests: unit testing and functional testing. For unit testing, we expanded upon the tests we had previously written in the "Project Phase A" book. We ensured that our unit tests covered a wide range of scenarios and functionalities, including edge cases and error handling. By doing so, we aimed to verify the correctness of individual components and functions within the system. In addition to unit testing, we performed functional testing to assess the system's overall behavior and user experience. These tests focused on evaluating the system's functionality as a whole, ensuring that all features worked as intended and met the requirements outlined in the project specifications. By conducting both unit and functional testing, we were able to identify and address any issues or inconsistencies in the system's behavior. This comprehensive evaluation process helped us ensure the reliability and effectiveness of the Teach-Me platform before deployment.

### 2.3.1. Unit Testing

|  |  |  |  |
| --- | --- | --- | --- |
| **Registration** | | |  |
| **No.** | **Test Subject** | **Expected result** | **Result** |
| 1 | Enter an email that already exist in system. | "This email address is already being  used". | Pass |
| 2 | Enter invalid Email address | "Invalid email". | Pass |
| 3 | Enter an ID number that already exist in system. | "This ID number is already being  used". | Pass |
| 4 | Entre wrong phone number | “This phone number is invalid”. | Pass |
| 5 | Enter empty one of require fields and click Sign Up | "Please fill all require fields". | Pass |
| 6 | Enter course don’t exist | “This course not exist in academic”. | Pass |
| 7 | Enter wrong price | “Price cant contain (!@# etc.) or The price is very expensive”. | Pass |
| 8 | Enter wrong Grade | “Grade is not enough to teach the course”. | Pass |

# 

|  |  |  |  |
| --- | --- | --- | --- |
| **Login** | | |  |
| **No.** | **Test Subject** | **Expected result** | **Result** |
| 9 | Enter empty email or password | "Please fill all the fields". | Pass |
| 10 | Enter invalid Email address | "Invalid email". | Pass |
| 11 | Enter correct email but incorrect password. | "Invalid email or password, try again". | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | | |  |
| **No.** | **Test Subject** | **Expected result** | **Result** |
| 12 | pressed confirm and don’t choose Date or Time | “Invalid Date or Time” | Pass |
| 13 | Choose a Date that passed | “Invalid Date” | Pass |
| 14 | Choose date equal today and time that passed | “Invalid Time” | Pass |

### 2.3.2. Functionality Testing

|  |  |  |  |
| --- | --- | --- | --- |
| **Login** | | |  |
| **No.** | **Test Subject** | **Expected result** | **Result** |
| 1 | Enter correct Login details, and press on  Login button. | The screen switches to the home page | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| **Select Course** | | |  |
| **No.** | **Test Subject** | **Expected result** | **Result** |
| 2 | Enter course name, and press on Search button. | The screen switches to another page with teacher list. | Pass |
| 3 | Select teacher from teacher list | The screen switches to the choose day and hour. | Pass |
| 4 | After select day and hour | The screen switches to Payment | Pass |
| 5 | If the teacher booked on specific hour on specific course, other lesson cant be booked at the same time with same teacher | The System display a message that this hour unavailable because the teacher already booked on other lesson | Pass |

## 2.4. Results and conclusions

Our Teach-Me project successfully achieved its goal of creating a platform that connects students with tutors, facilitating private tutoring sessions for various courses. Through the application, students can easily find suitable tutors for their needs, while tutors can efficiently manage their teaching services.

One of our primary objectives was to develop a user-friendly, cross-platform application that operates seamlessly on various devices, including mobile phones, computers, and tablets. We accomplished this by implementing responsive design principles, ensuring a consistent and optimal user experience across different platforms.

Throughout the project, we encountered several challenges that required us to acquire new knowledge and skills. In particular, we needed to familiarize ourselves with technologies such as React, Node.js, and MongoDB to effectively implement our project. Despite the initial learning curve, mastering these technologies enabled us to successfully complete our project.

During the implementation phase, we adopted an agile development approach and divided the main functionalities into sprints. For example, the initial sprint focused on implementing the "Lesson Scheduling" feature, allowing us to efficiently build and iterate on different components of the system.

To address coding challenges that arose during the project, we leveraged online resources such as Stack Overflow, youtube, w3school, and passed over a lot of tutorials to seek guidance and assistance from the developer community. This collaborative approach helped us overcome obstacles and move forward with the implementation of our project.

Overall, the development of the Teach-Me system provided us with valuable experience. Utilizing Git for version control allowed us to efficiently track changes and collaborate effectively as a team. Learning JavaScript for React.js and Node.js was particularly rewarding, as we discovered its strengths and potential for building dynamic and responsive web applications.

# 3. User Documentation

## 3.1. User’s guide:

### 3.1.1. General description:

Our Teach-Me system aims to connect students directly with tutors, eliminating the

need for intermediaries such as tutoring agencies or educational institutions. By

providing a platform for private tutoring, students can find suitable tutors for their

needs, while tutors can offer their services and earn income during their studies.

The primary goal of our application is to provide a simple and user-friendly interface

That allows students to easily locate tutors for specific courses. Through the system, students can browse available tutors, view their qualifications and ratings, and schedule lessons according to their preferences.

We have designed the system to accommodate four different user roles: unregistered

users, registered users, tutors, and administrators. Each role has its own set of

permissions and capabilities within the system.

To provide clarity and ease of use, the user guide for our project is divided into several

sections, each focusing on different aspects of the system. This division helps users

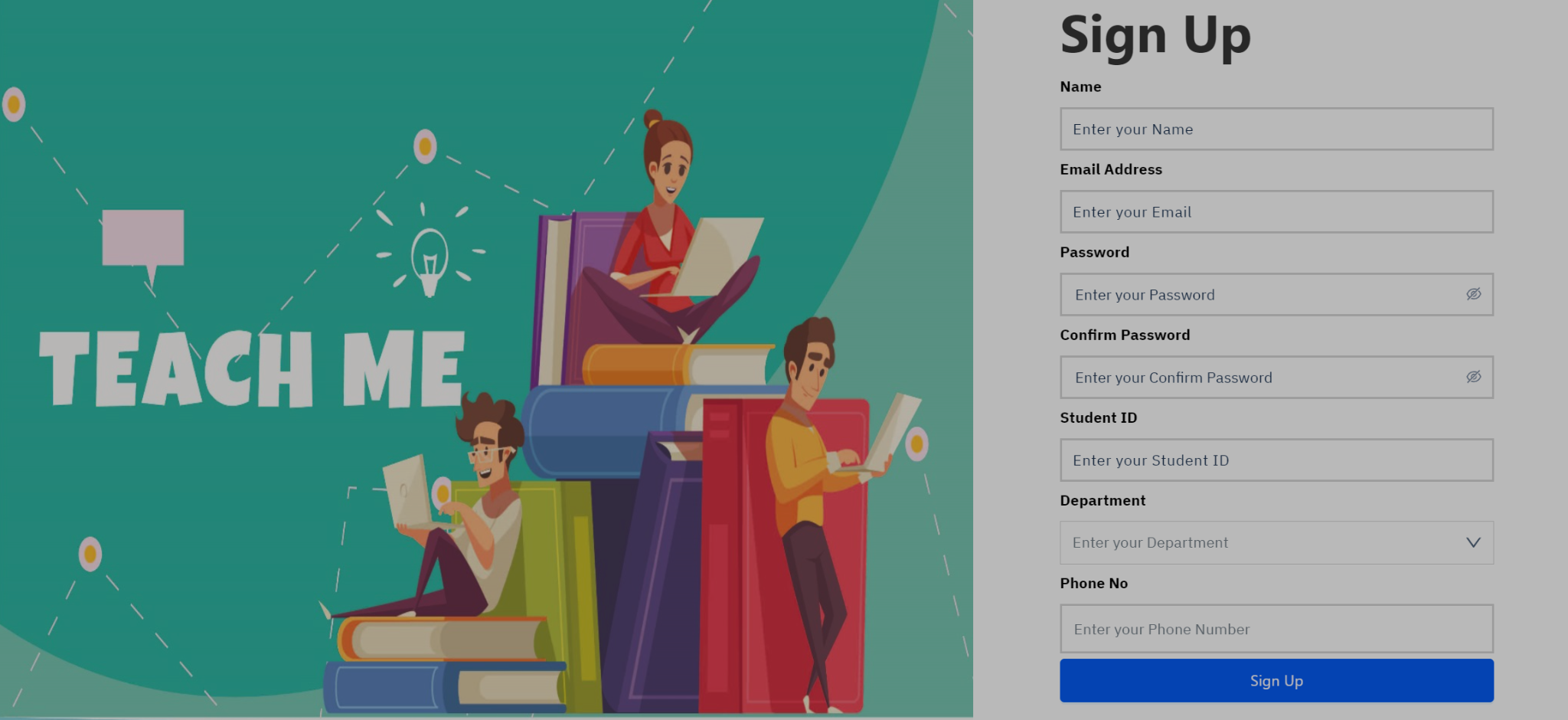
navigate the system more effectively and understand their roles and responsibilities

within it.

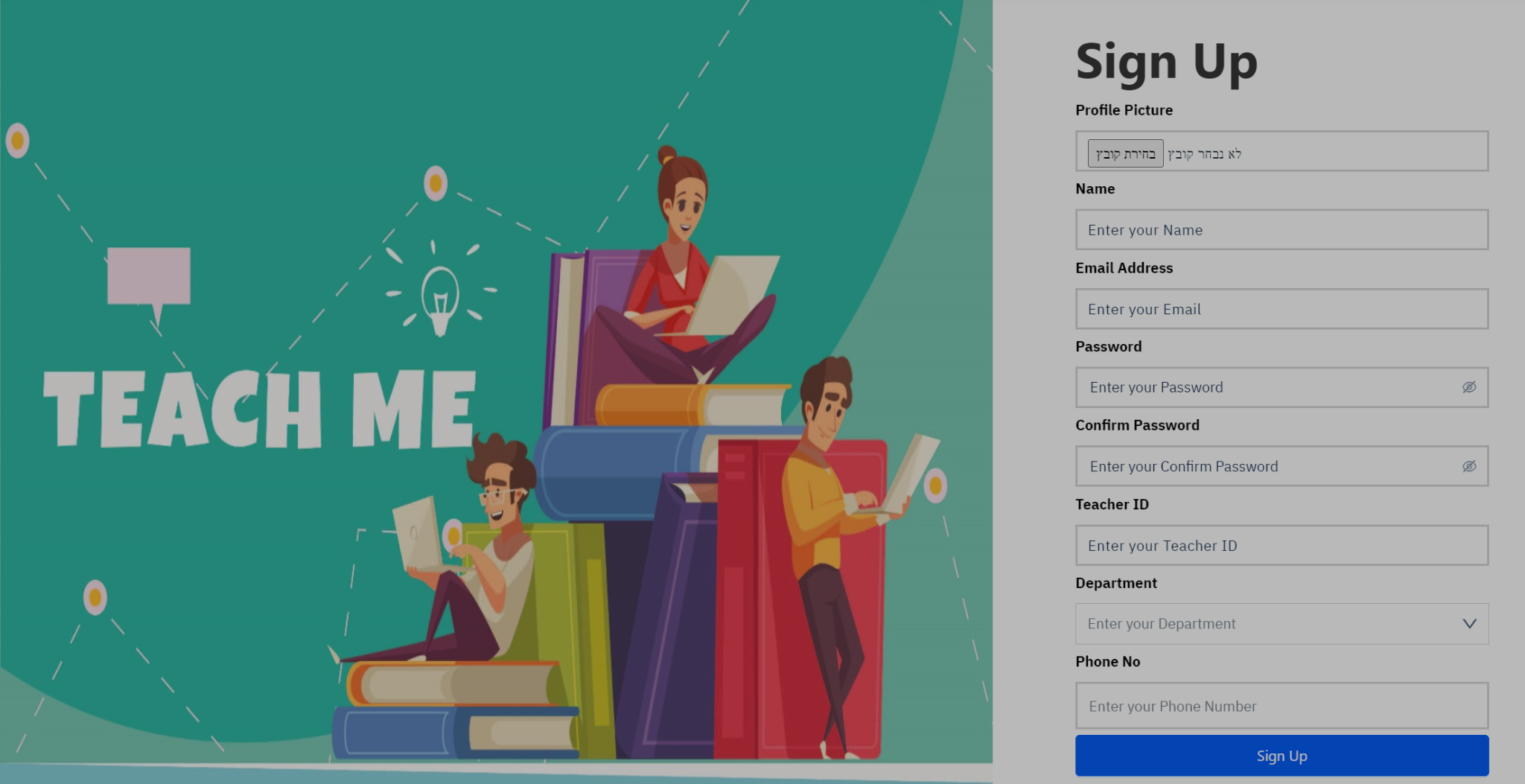
**Registration and Login:**

To sign up for the system, users should follow these steps:

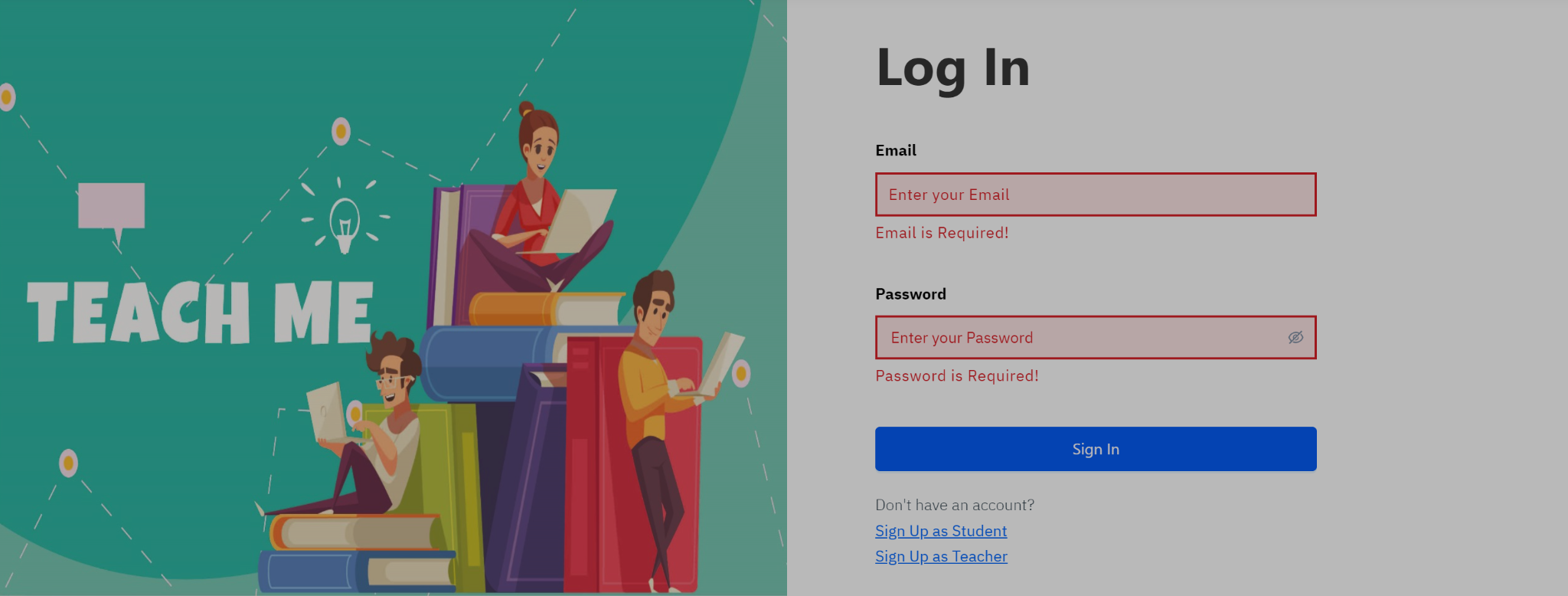
1. To create an account on the Teach-Me platform, users can choose between two types of registration: Student or Teacher. Student registration requires providing a name, email,student ID, department, phone and password.



Teacher registration requires providing a photo, name, email, teacher ID, department.phone and password.



The login page:



**Student user - Choosing Course To Learn:**

In the Teach Me system, every user, including those who haven't

registered, can browse and watch the available courses. To choose a

specific course to learn, users need to choose the course they want to learn

and then choose the teacher they want to learn with him.

In order to book a lesson with teacher, the student user need to make the following

steps:

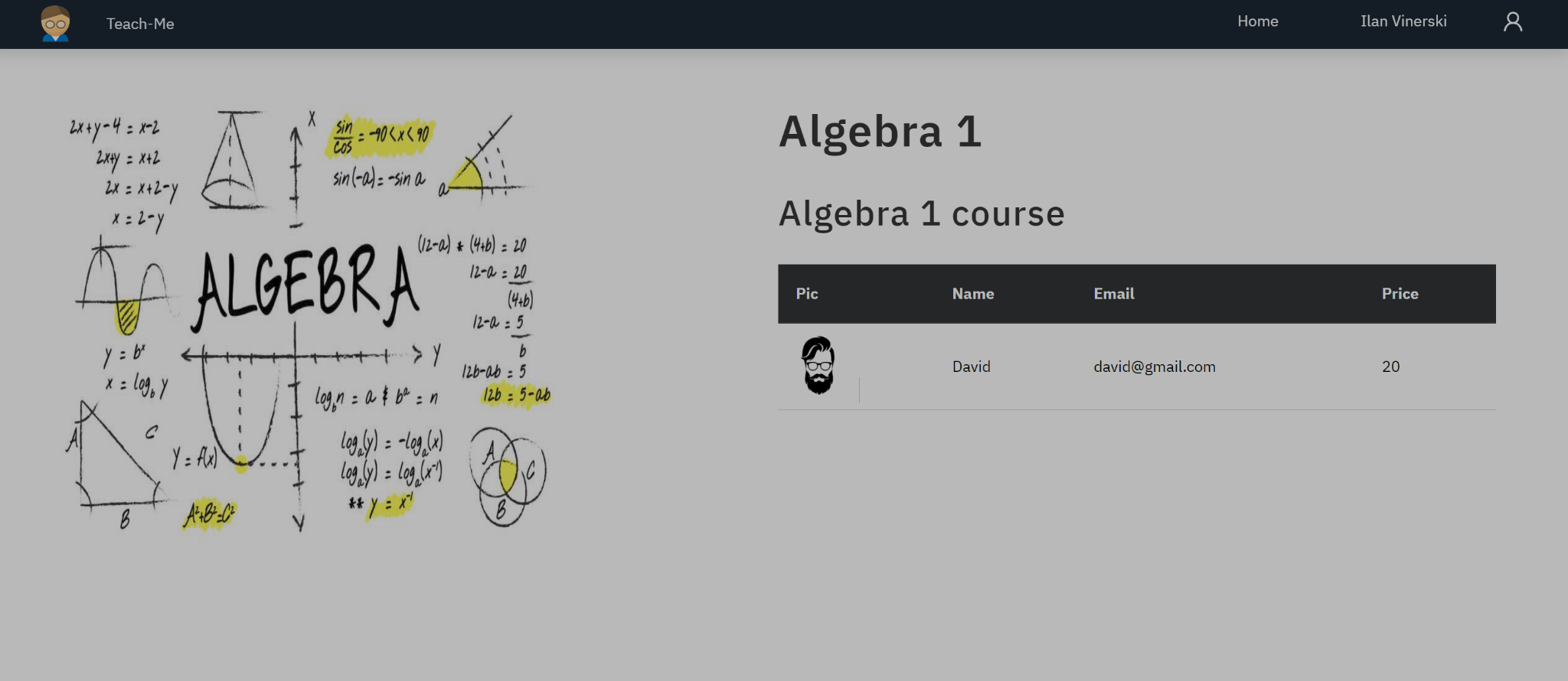
a. At the home page, the student need to choose what course to learn:



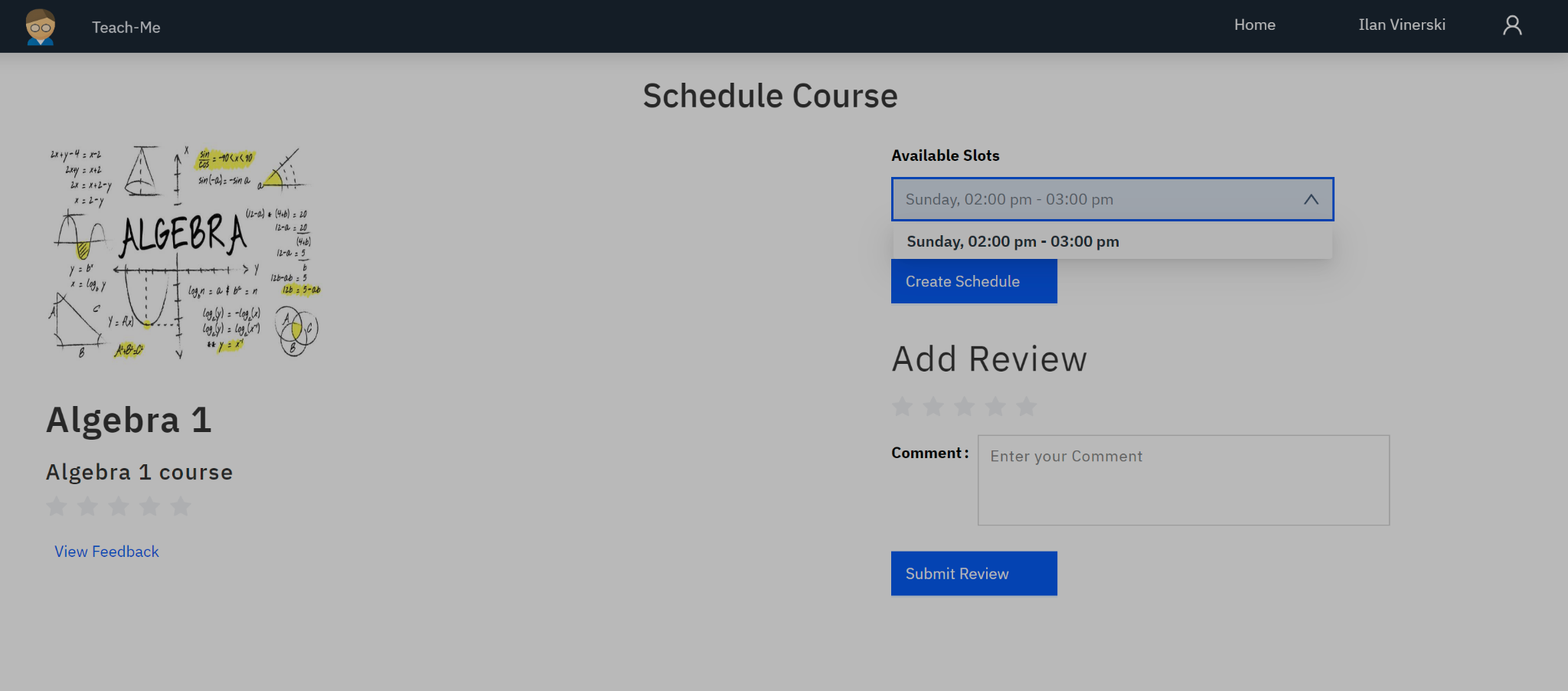
b. after the student choose a course, he need to choose the teacher he wants to learn

with him, he can check the price the teacher wants per lesson and decide who to

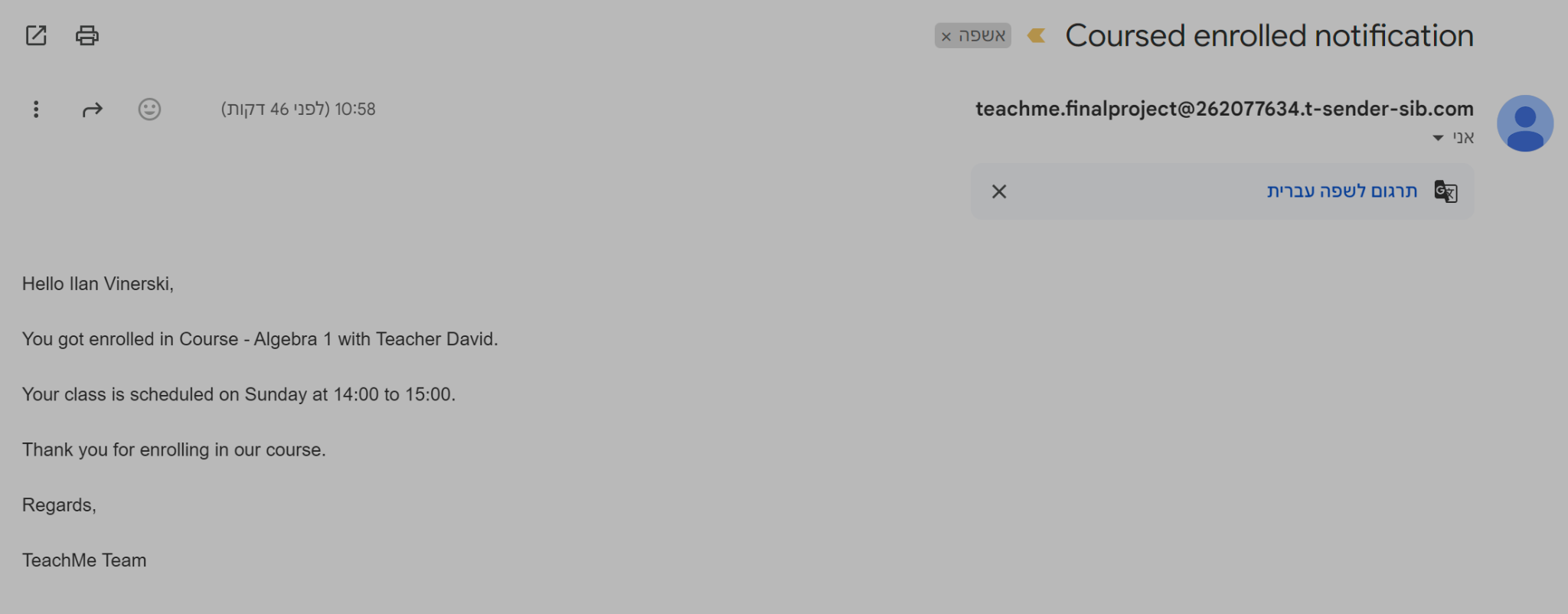
choose:



c. After the student choose the teacher, he need to pick the time he wants to schedule the lesson with the teacher:

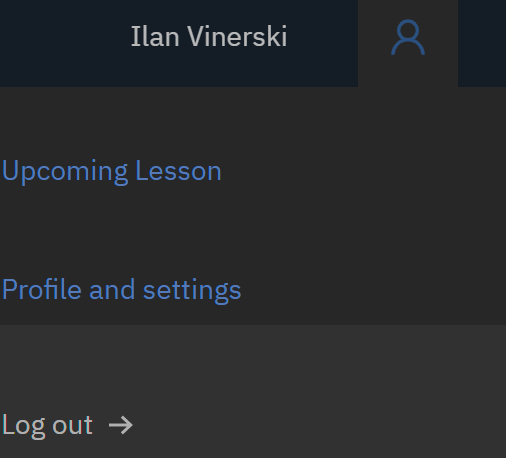


and then he clicks on Create Schedule, and the process ends. And the student redirects t to the home page. At the end, both the teacher and the student get email with details

about the lesson:  


d. After the student scheduled the lesson, he can go to his dashboard panel and watch

his upcoming lessons:



and he see which lessons will be in future:



**Student user - Adding Reviews About Teacher’s Course:**

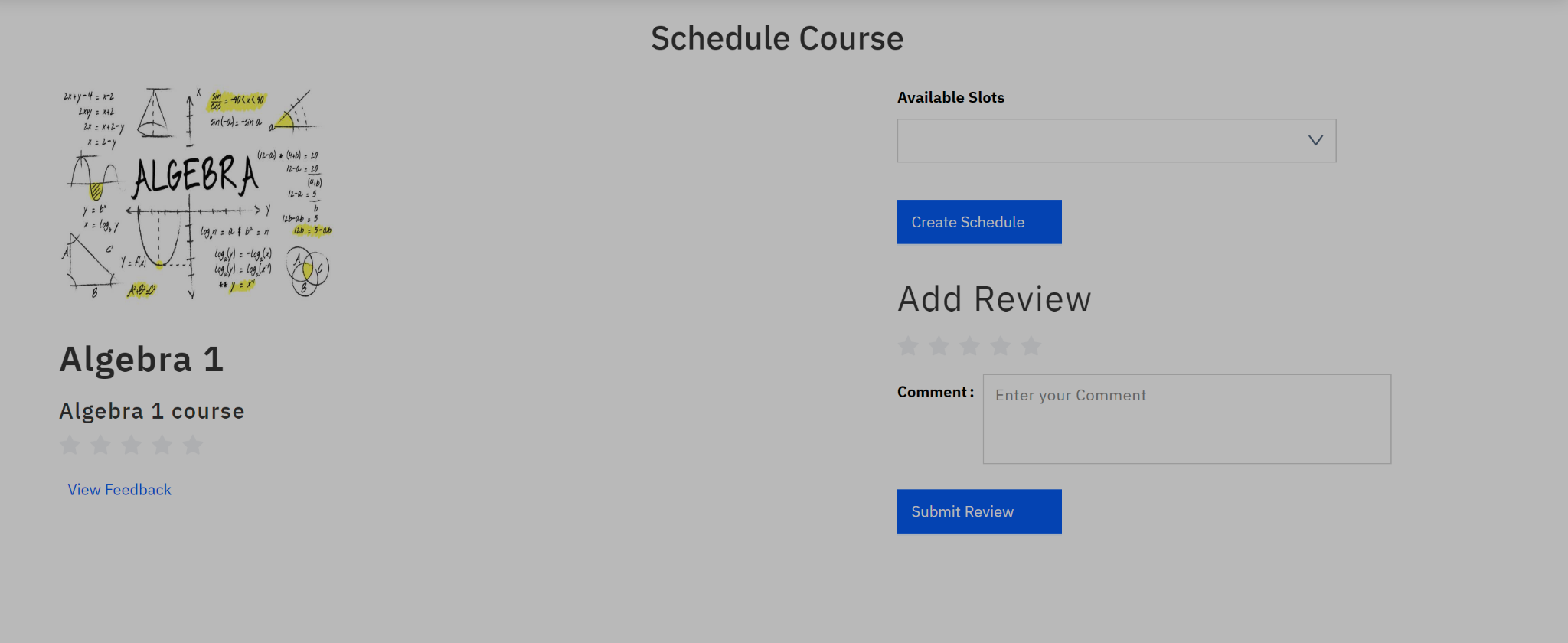
In the Teach Me system, when a student visits the teacher's profile, they have the

option to view all of the teacher's reviews and submit their own review through a

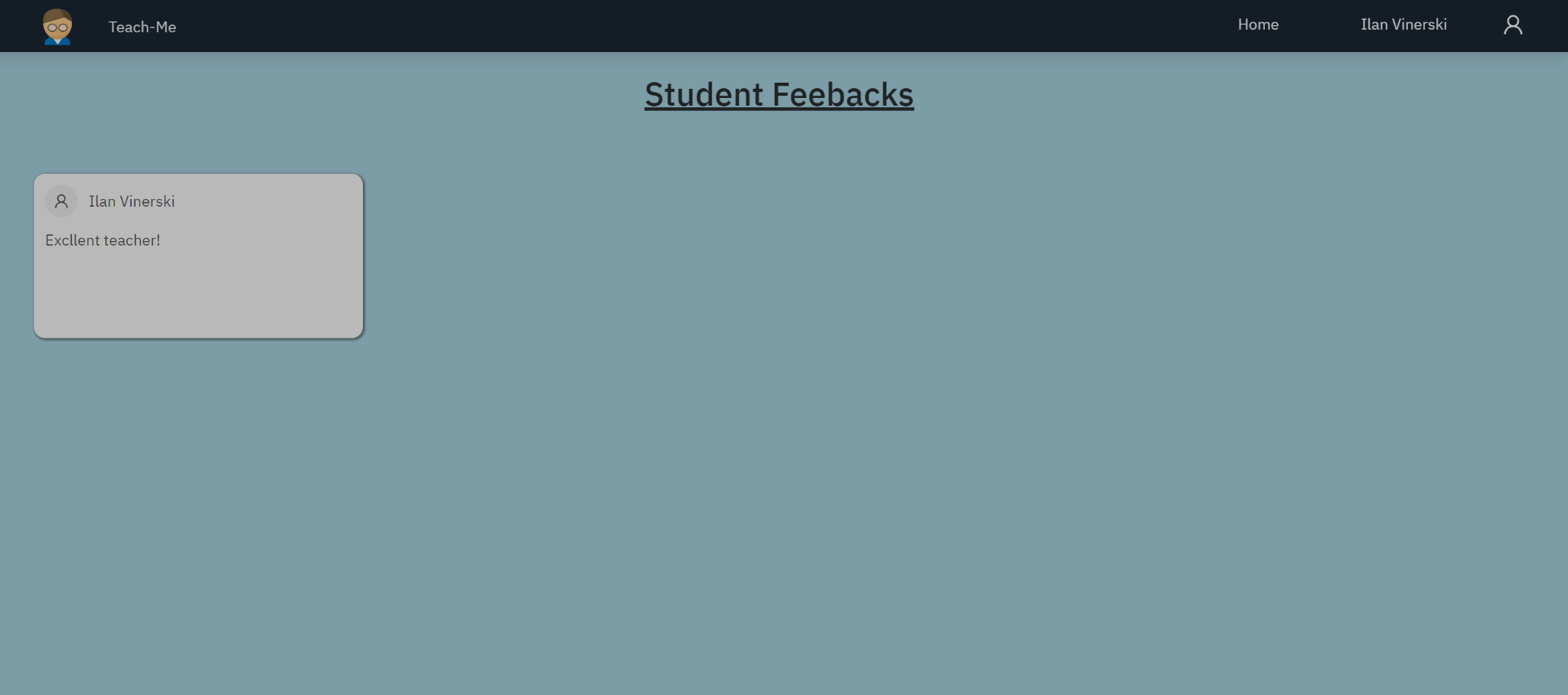
form. To submit a review for a specific teacher, the student needs to

follow these steps:

In the Teach Me system, users can click on the course they want to check teacher feedback, then they need to choose the teacher and it takes them to the teachers page:



To view reviews for a specific teacher, the user needs to click on the "View Feedback" text, and he can watch all the feedbacks that exists about that teacher on that course:



**Teacher user - Managing Courses:**

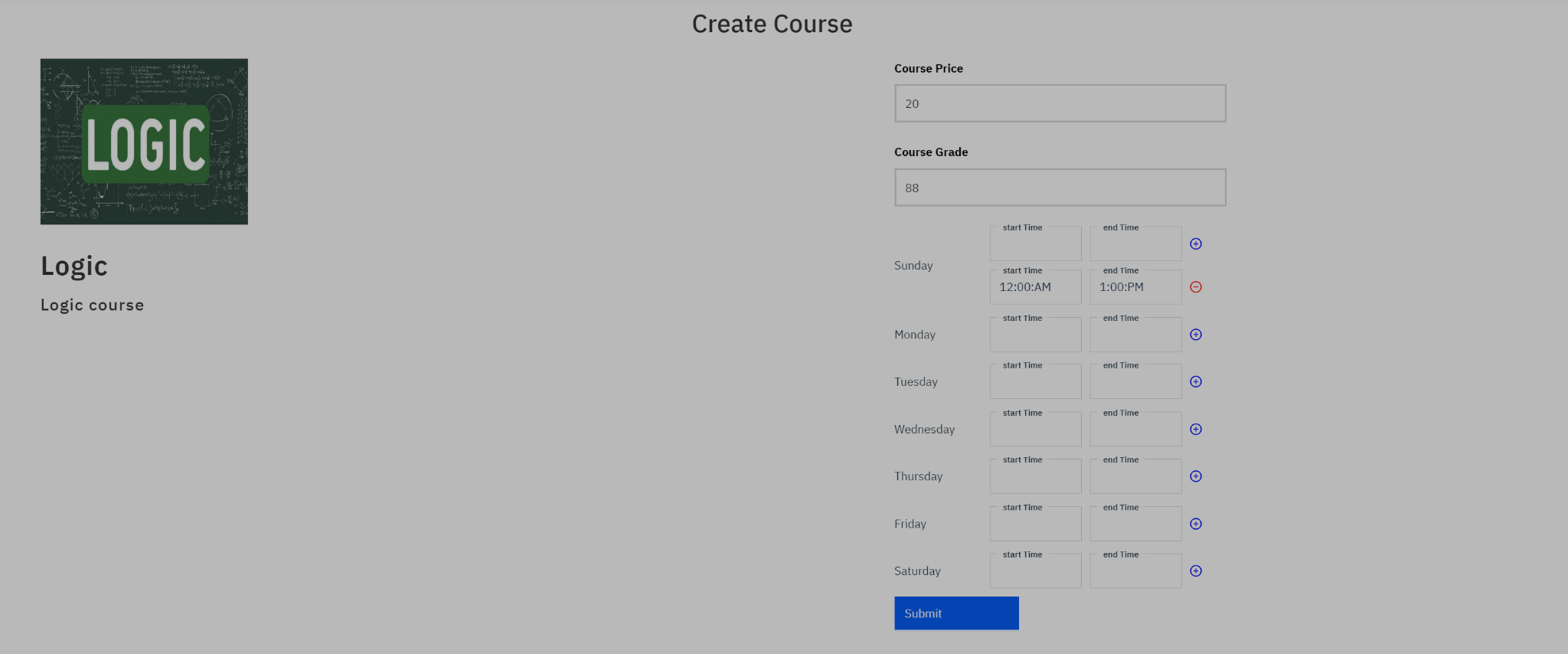
In the Teach Me system, teachers have the option to view a list

of the courses they are teaching. Additionally, teachers have the ability to easily

add new courses or modify existing ones, enabling them to effectively manage

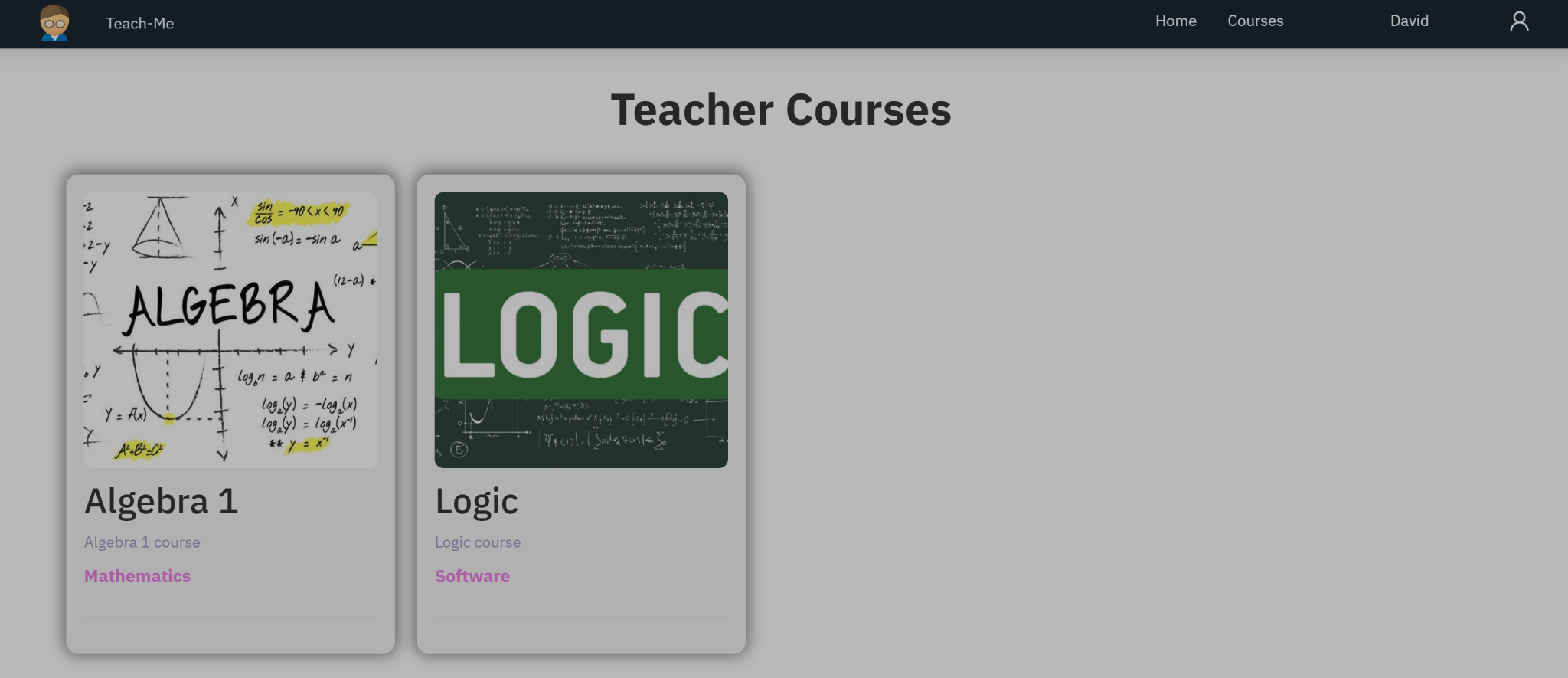
the courses they offer to teach.

1. When the teacher is logged in, he can click on course he wants to teach

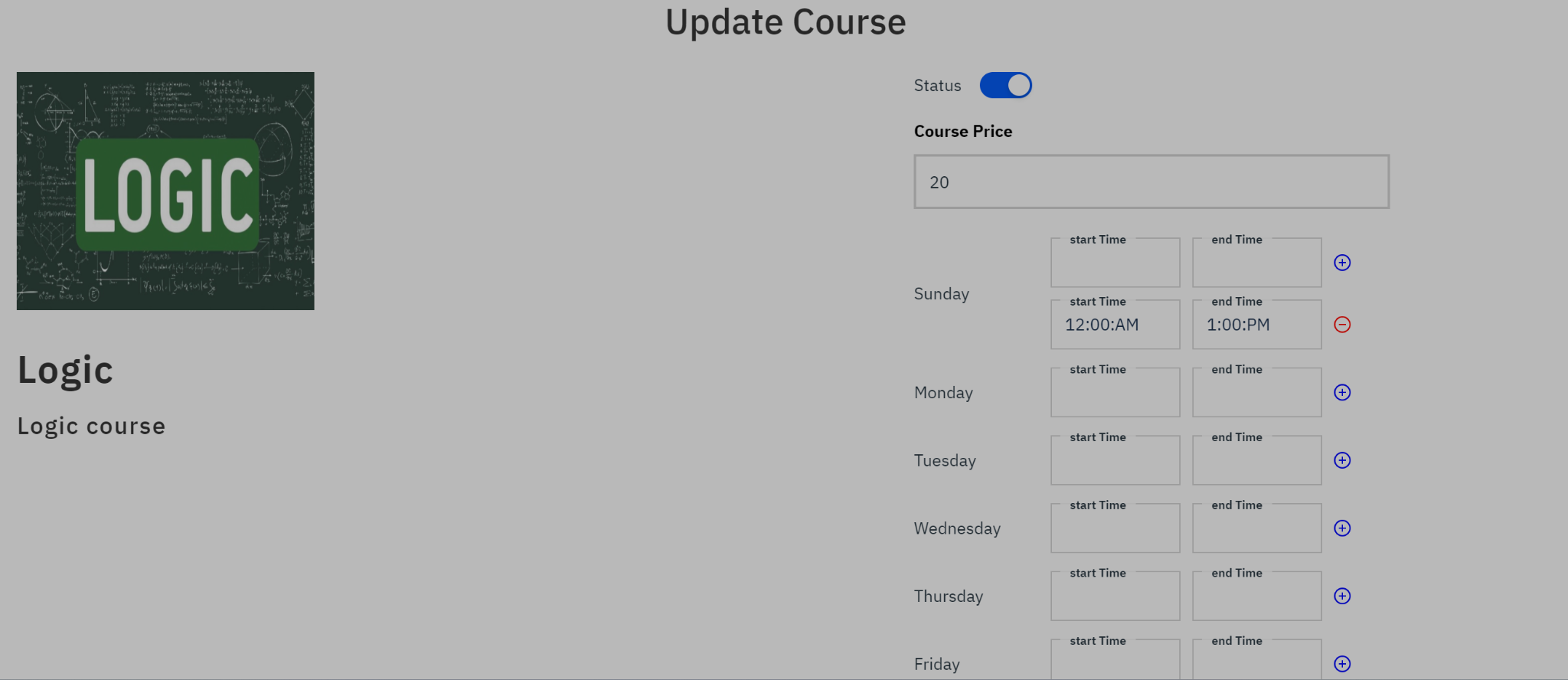
and fill a form of the price he wants to charge per lesson, enter his grade on the course, and enter the schedule that fits him. Teacher need to have grade thats stands on the requirements (above 85) and charge price under the maximum (max 25$) and then click on “Submit” button:  
  


If the teacher stands in all requirements he will successfully add the course to his courses panel.

1. Teacher can enter to his courses panel and he will see all the current courses he register to teach:



1. If the teacher wants to remove course that he doesn't want to teach, he can click on the course and click the “Status” button and he will remove himself from that course:



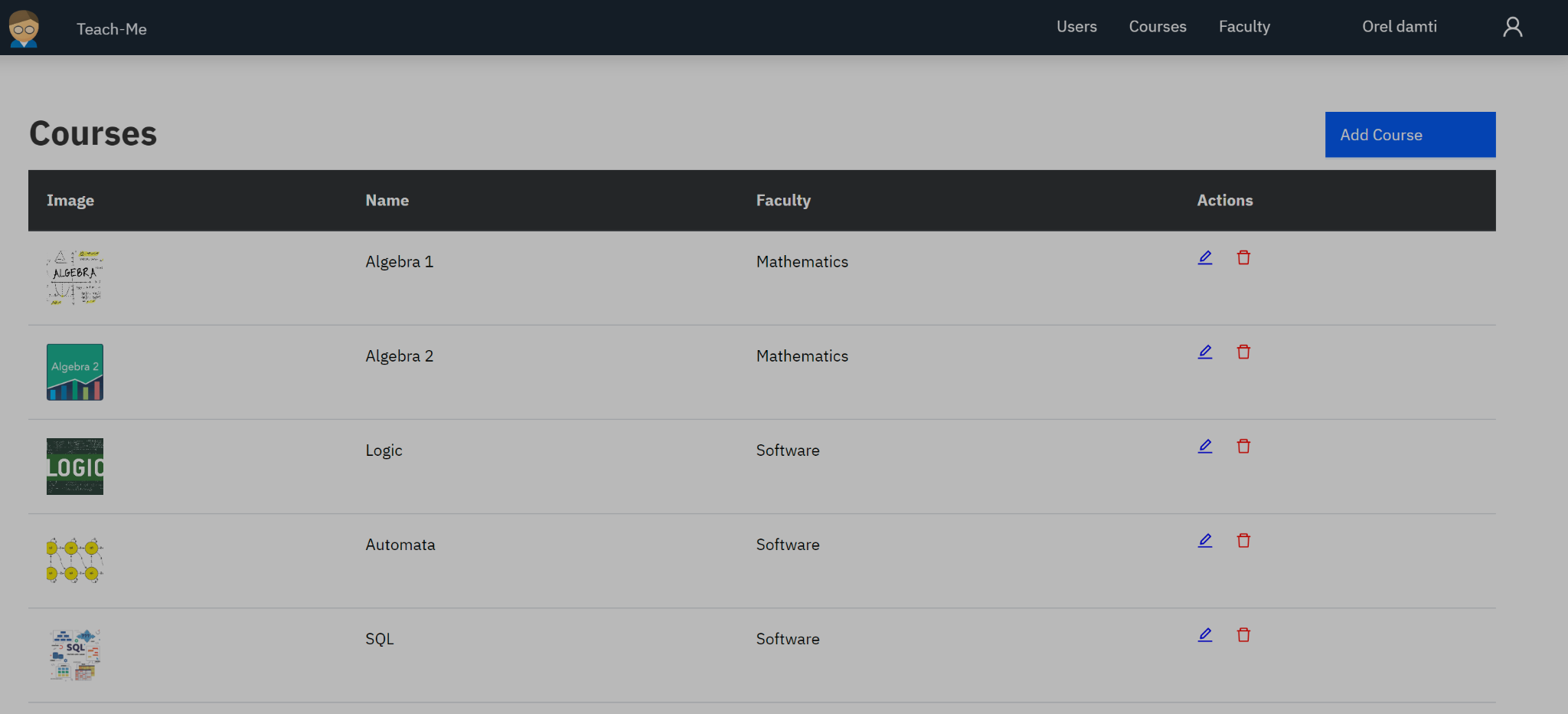
After the change of status, the teacher won’t be shown anymore for students as an available teacher to this course.

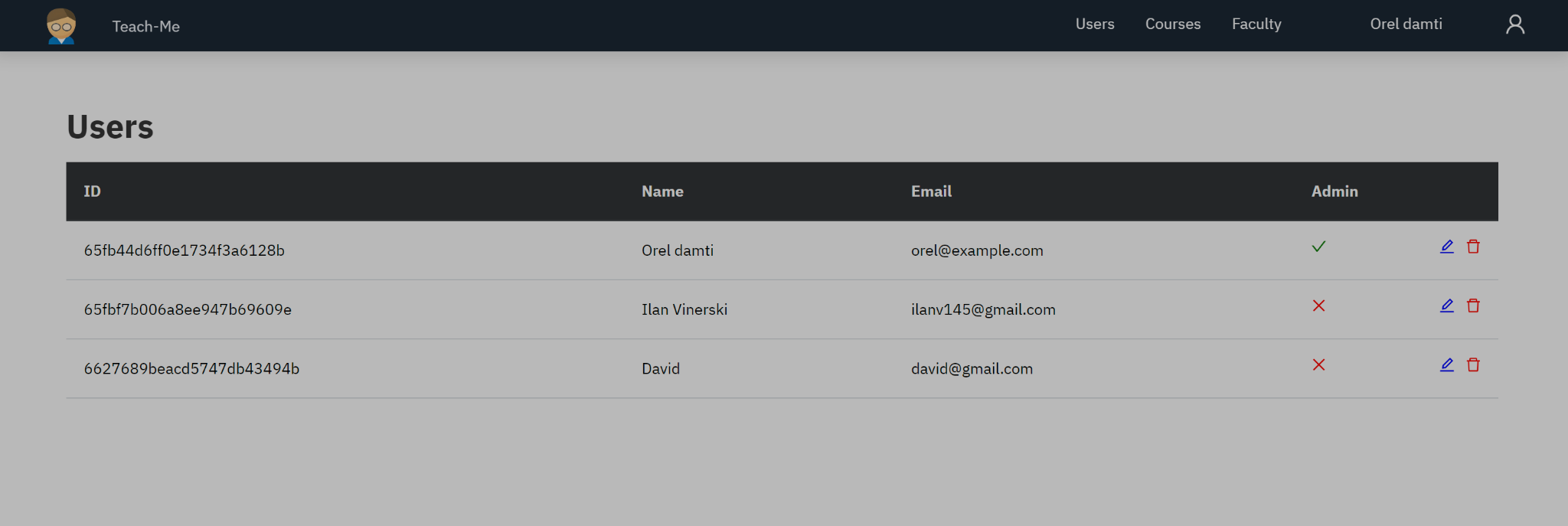
**Admin user - Managing system:**

The admin manages the system. He can add and delete courses, delete users and

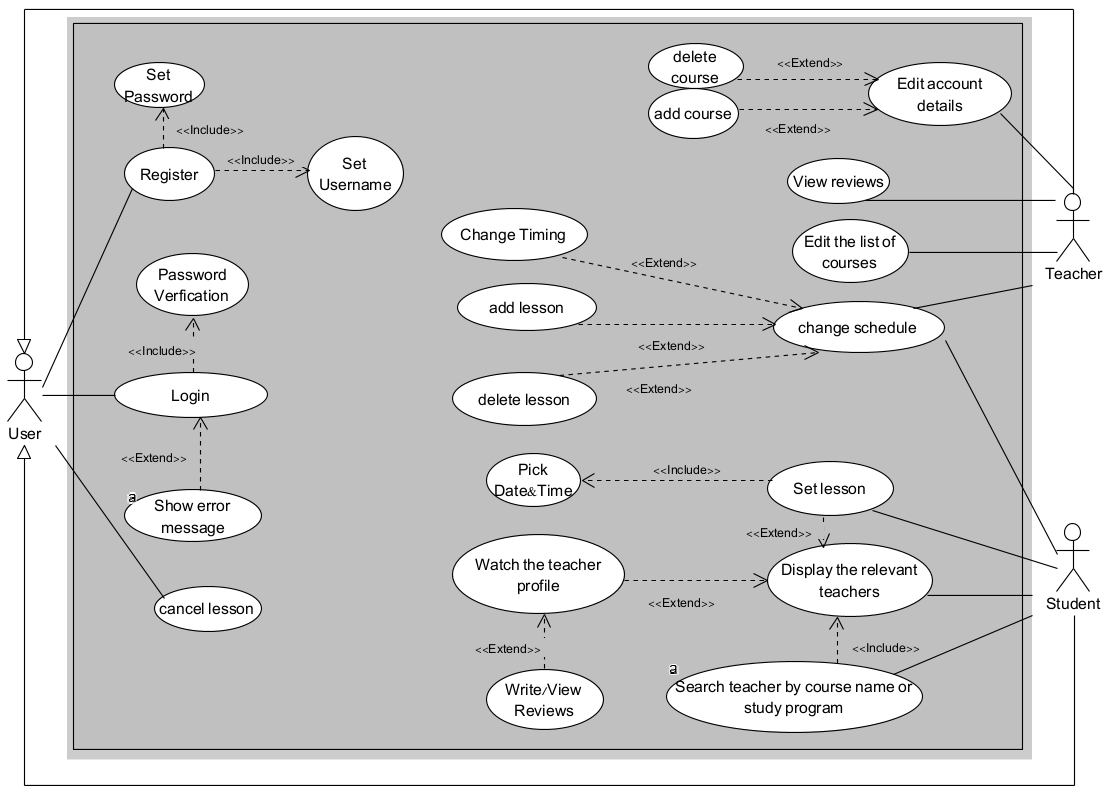
edit users.

1. After the admin logged in to the system, he can choose to edit the courses in the system. By choosing from the above panel the option courses he can change the photo of the course, edit the name of the course or change the faculty that the course belongs to:

  
  
b. The admin also can remove users from the system by choosing the users in the top panel:

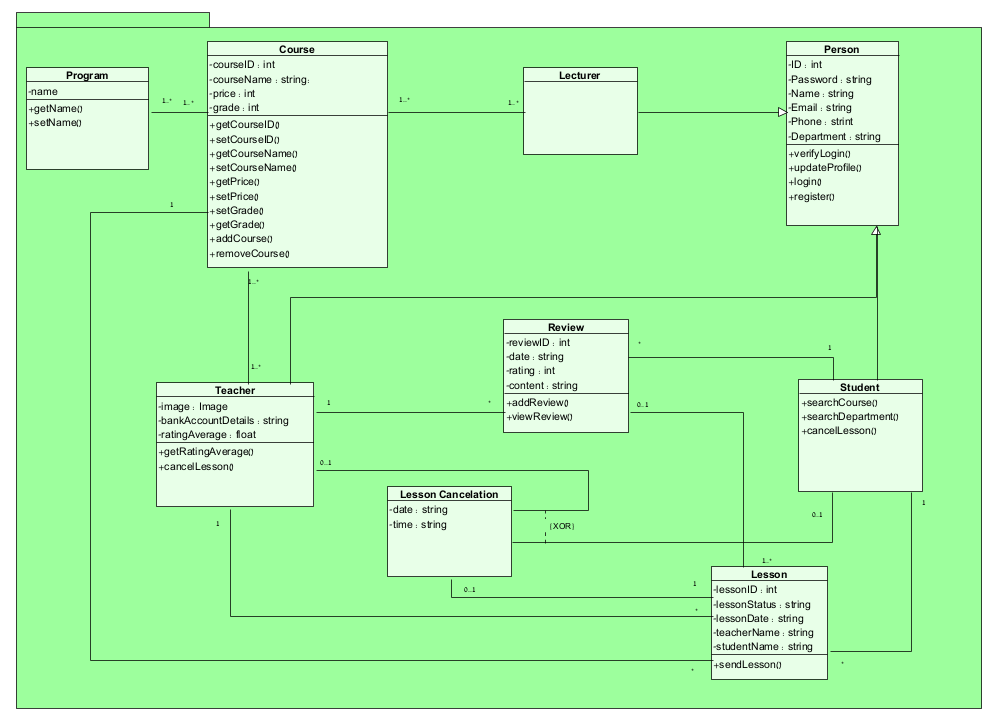


### 3.2.1. Use Case Diagram:

**

**Fig 1** : Use Case Diagram

### 3.2.2. Class Diagram:

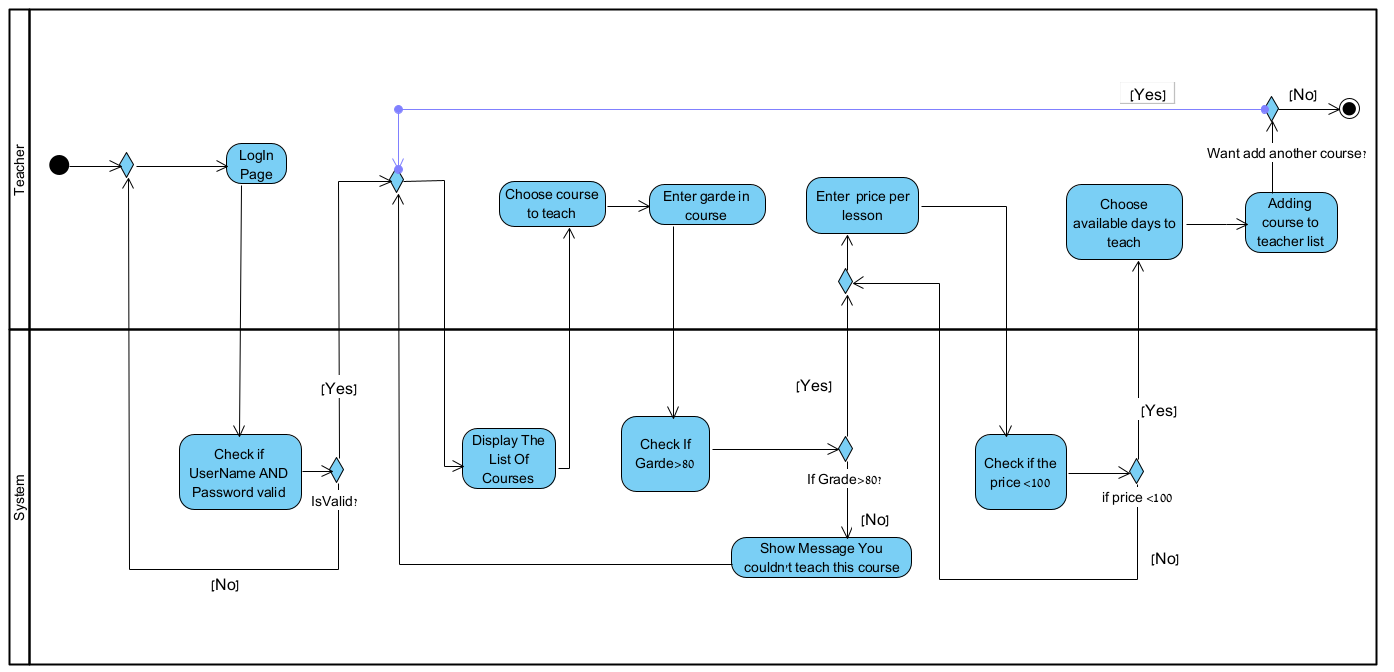
**

**Fig 2** : Class Diagram

### 3.2.3. Activity Diagram:

#### 3.2.3.1. Process of Adding Course of Teacher

* + - 1. The Teacher Login Into the system.
      2. The system checks if Username AND Password is valid.
      3. If the System returns no go to step 1.
      4. The System displays a list of courses.
      5. The teacher selects a course to teach.
      6. The teacher enters the grade in the course he wants to teach.
      7. The system checks if grade above 85.
      8. If the system returns NO the system displays a message that the teacher couldn’t teach this course and go to step 4.
      9. The teacher enters the price per lesson.
      10. The system checks if the price is under 25.
      11. If The system returns NO go to step 9.
      12. The Teacher chooses available days to teach.
      13. The teacher added the course to the teacher list.
      14. The system asks if you want to add another course.
      15. If The Teacher returns YES go to step 4.
      16. The Teacher returns NO the process is END.

******

**Fig 3** : Activity Diagram : Adding Course of Teacher

### 

### 3.2.4. Program structure and runtime configuration:

React version 18.2.0

React-Bootstrap version 2.10.2

React-Router version 6.22.3

React-Redux version 9.1.1

Redux-Toolkit version 2.2.2

NodeJS version 21.6.1

MongoDB version 7.0.0

ExpressJS version 4.19.1

# 

# 4. References

<https://www.methoda.cloud/?section=254>

<https://www.indeed.com/career-advice/career-development/software-development-methodologies>

<https://www.omnisci.com/technical-glossary/client-server> <https://azure.microsoft.com/en-us/overview/what-is-cloud-> [computing](https://azure.microsoft.com/en-us/overview/what-is-cloud-computing)

<https://www.techtarget.com/searchcio/definition/Agile-> [project-management](https://www.techtarget.com/searchcio/definition/Agile-project-management)

<https://react.dev/>

<https://nodejs.org/en>

<https://www.opentext.com/what-is/agile-development>

<https://redux.js.org/>

<https://expressjs.com/>

<https://getbootstrap.com/>

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