Table of Contents

[1. Introduction 2](#_Toc99308029)

[1.1 Objectives 2](#_Toc99308030)

[1.2 Scopes 2](#_Toc99308031)

[1.3 Project Schedule 2](#_Toc99308032)

[2. Requirement Specification 2](#_Toc99308033)

[2.1 Audience Modelling 3](#_Toc99308034)

[2.1.3 Major functions of the Website 4](#_Toc99308035)

[3. Design and Modeling 5](#_Toc99308036)

[3.1 Entity Relationship Diagram (ERD) 5](#_Toc99308037)

[3.2 Data Dictionary 6](#_Toc99308038)

[3.3 Wire Framing 13](#_Toc99308039)

[3.4 Website Navigational structure 16](#_Toc99308040)

[4. Implementation (Code Snippet) 19](#_Toc99308041)

[5. User Guidance 22](#_Toc99308042)

[6. Conclusion 34](#_Toc99308043)

[**7. Future Enhancement 35**](#_Toc99308044)

[8. References 36](#_Toc99308045)

[9. Workload Matrix 37](#_Toc99308046)

# Introduction

Krypton Education is an educational platform for online coding and academic activities. This Krypton Education will be available to any learners who desire to learn online. The goal of creating this website is to help students and the university overcome the challenges that arose during the Covid-19 outbreak. During the pandemic, many students were forced to rely on third-party software to conduct online classes and provide learning materials. But now we've decided to build this website that can handle everything online. This Krypton Education Webspace will be able to manage all academic activities such as downloading courses and course materials, viewing assignments, web results, online assignment submissions, and progress reports. We have been tasked with creating this website as a group.

## 1.1 Objectives

* To allow students to access lecture videos and other courses.
* To enable lecturers/professors to post learning resources such as pdf and slides.
* To enable students to submit assignments and view feedback provided by the lecturer.
* To give potential students and visitors information about available courses, krypton Education's facilities, and its history.

## 1.2 Scopes

Every organization is migrating from a manual to an online management system to manage its operations. Krypton Education has also decided to create an online website to keep track of its digital activities. As the world becomes busier by the day, people's time is becoming increasingly valuable. As a result, our e-learning system will assist people in conserving time and reinvesting it in more profitable endeavors. Many students had to rely on third-party software to keep everything up to date, which was prohibitively expensive. It was difficult to keep track of all of the numerous applications required for each activity. This online student Webspace will be cost-effective and simple to maintain because it will have all necessary features in a single website.

## Project Schedule

# Requirement Specification

A requirement specification is a detailed description of the system's intended function and environment. It outlines in detail what the system will accomplish and how it is intended to function. (n.d., ocw.mit.edu) The definition of requirements not only reduces the time and effort required by developers to achieve desired goals but also reduces development costs. n.d. (Rouse) Before constructing a website, it is critical to define the needs. The requirements acquired before constructing this website are as follows.

# 2.1 Audience Modelling

Audience modeling is an essential element in the Web Site Design Method (WSDM), which is an audience-driven design method for Web applications. The primary goal of audience modeling is to specify and identify the set of target users since they will be the ones utilizing the system. The mission statement's selected targeted users are re-defined into audience classes. This is done in two stages:

* Audience categorization
* Audience characterization.
  + 1. **Audience categorization**

1. **Administration / Super Use**r

The Krypton Education administration is the website's superuser, with access to all of the system's capabilities. This group of users is in charge of adding new courses, revising pricing structures, issuing announcements, and approving newly enrolled students in the system. Superusers have the authority to create, update, delete accounts of all three users i.e., potential users, lecturers, and enrolled students, edit the content of course pages, manage class schedules, and update their respective profiles.

1. **Lecturers/professors**

This class's users are in charge of engaging with registered students. They are in charge of uploading study materials like lecture slides, pdf, tutorials on the website. They will assign assessments and tasks to students and provide feedback. Similarly, they can manage the provided assignments and course materials by updating and deleting and they can manage their respective profile.

1. **Enrolled Students**

These users have access to course materials given by their lecturers/professors. These users are authorized to enroll in new courses and view assignments and class schedules. These students get access to Krypton Education's most recent news and announcements. They can also download the provided courses materials and upload their assignments and examine their feedback.

1. **Potential students.**

Users in this class are visitors to this website. They are not permitted to access the course pages. They can only see information about Krypton Education, available courses, and course cost arrangements. They may also see Krypton Education's historical events, successes, achievements, etc. They are permitted to submit inquiries and are allowed for registering into the system. These users can become enrolled students only if they enroll in a particular course with full payment and approved by the admin.

* + 1. **Audience characterization**

During Audience Characterization, relevant representations of significant traits for each crowd class are given. Because different sorts of customers may have different data and utilitarian requirements, it may be necessary to address something quite similar data or usefulness to different types of clients in different ways, depending on the attributes of the clients. (Olga De Troyer, undated) The following are the many types of clients who will use this site based on attributes:

Technical Users: These are individuals who are knowledgeable about online apps and how to use them. As a result, they are not the ones to provide more clarity of thought.

Beginner Users: These are people who are unfamiliar with the framework and how to use it. As a result, the plan connection point should be easy enough for new customers to use.

## 2.1.3 Major functions of the Website

The following are the primary functions of this website (Krypton Education):

* Students enrolling on the Krypton Education website get access to study materials, notes, and pdfs posted by their professors.
* Students can view and submit assignments as well as view assignment deadlines.
* Professors can post study materials, provide feedback, create assignments, track students' progress, and manage their profiles.
* Administration can register students, lecturers, as well as submit messages, adjust fee structures.

# Design and Modeling

## 3.1 Entity Relationship Diagram (ERD)

Diagram

Description automatically generated

**Figure 1: ERD Diagram of Krypton Education**

## 3.2 Data Dictionary

* + 1. **Users**

This table is used to store the personal information of admin, users, and lecturers.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Name** | **Field Type** | **Field Length** | **Description** | **Example** |
| ID | Integer | -------- | Unique Id of users | 1 |
| First\_Name | Varchar | 50 | First Name of the users | John |
| Last\_Name | Varchar | 50 | Last Name of the users | Doe |
| Address | Varchar | 50 | Address of the users | Pokhara-20 |
| Phone | Varchar | 50 | Contact Number of the users | 9852636267 |
| Email | Varchar | 50 | Email of the users | johndoe12@gmail.com |
| Password | Varchar | 50 | The password of the users | doejohn12 |
| UserType | Varchar | 50 | Type of users | User |

* + 1. **Students**

This table consists of the student information after they get the admin’s approval of a particular course.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Name** | **Field Type** | **Field Length** | **Description** | **Example** |
| ID | Integer | ------------ | Unique Id of a student | 1 |
| Name | varchar | 50 | Name of a Student | John |
| Email | varchar | 50 | Email of a student | johndoe12@gmail.com |
| Address | varchar | 50 | Address of a student | Pokhara-20 |
| Phone | varchar | 50 | Contact Number of a student | 9852636267 |
| Enroll\_Date | varchar | 50 | Date when a user enrolls into a course | 2021-02-04 |
| Course\_Title | varchar | 50 | Name of the enrolled course | Python |
| UserType | varchar | 50 | Type of users | Student |

* + 1. **Student\_Assignment**

This table consists of details of the student who upload their assignment including the file they uploaded. After this lecturer checks their assignment and provides feedback which is then stored in this table and displayed on the feedback section of the course page.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Name** | **Field Type** | **Field Length** | **Description** | **Example** |
| ID | Integer | ------------ | Unique Id | 1 |
| Student\_Name | varchar | 50 | Name of a Student | John |
| Email | varchar | 50 | Email of a student | johndoe12@gmail.com |
| File\_Name | varchar | 50 | Name of a file | Assignment |
| File\_Type | nvarchar | 200 | extension of a file | .pdf |
| Answer | varbinary | 200 | File uploaded by student | Assignment.pdf |
| Feedback | varchar | 50 | Feedback for assignment | Please submit on time. |

* + 1. **Schedule**

This table consists of the schedule of online classes of krypton academy.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Name** | **Field Type** | **Field Length** | **Description** | **Example** |
| ID | Integer | ------------ | Unique Id | 1 |
| Days | varchar | 50 | Days of the week | Sunday |
| Starts | varchar | 50 | Starting time of class | 9:00 am |
| Ends | varchar | 50 | Ending time of class | 11:00am |

* + 1. **Contact**

This table consists of details of when a user sends a message to the administrator through the contact form.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Name** | **Field Type** | **Field Length** | **Description** | **Example** |
| ID | Integer | ------------ | Unique Id | 1 |
| Name | varchar | 40 | Name of the user | John Doe |
| Email | varchar | 40 | Email of a user | johndoe12@gmail.com |
| Subject | varchar | 40 | Subject for contact | Add more Courses |
| Message | varchar | 40 | A query of the user | I want to enroll networking related courses. So please add as soon as possible. |

* + 1. **Assignment**

This table consists of data when a lecturer uploads assignments for the students including the question file.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Name** | **Field Type** | **Field Length** | **Description** | **Example** |
| ID | Integer | **-------------** | Unique id | 1 |
| Subject\_Name | varchar | 50 | Subject Name | Python |
| Hand\_In | varchar | 50 | Hand in date | 2021-02-02 |
| Hand\_Out | varchar | 50 | Hand out date | 2021-02-10 |
| FileName | varchar | 50 | Name of the uploaded file | Assignment |
| FileType | varchar | 50 | Extension of file | .pdf |
| Question | varbinary | max | Assignment file | Assignment.pdf |

* + 1. **Available Course**

This table consists of details of available courses in Krypton academy which are displayed on the website.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Name** | **Field Type** | **Field Length** | **Description** | **Example** |
| ID | Integer | **-------------** | Unique id | 1 |
| Course\_Code | varchar | 50 | Unique code of a course | Py-02 |
| Course\_Name | varchar | 50 | Name of the course | Python |
| Lecture\_Hour | varchar | 50 | Teaching hours for a course per day | 1 Hrs |
| Lecturer\_Name | varchar | 50 | Full Name of the Lecturer | John Doe |
| Course\_Price | varchar | 50 | The total cost of a course | $120 |
| Status | varchar | 50 | Availability status of a course | Available |
| File | varchar | 100 | Course Image | Image.png |

* + 1. **Booking**

This table consists of the details of when a user applies for a course through enroll form.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Name** | **Field Type** | **Field Length** | **Description** | **Example** |
| ID | Integer | **-------------** | Unique id | 1 |
| Name | varchar | 50 | Full name of the user | John Doe |
| Email | varchar | 50 | Email of the user | Johndoe@gmail.com |
| Phone | varchar | 50 | The contact number of a user | 982423421234 |
| Date | varchar | 50 | Date when a user enrolls | 2022-02-0 |
| Course\_Code | varchar | 50 | Code of a course | Py-02 |
| Course\_Title | varchar | 50 | Name of a course | Python |
| CardHolder\_Name | varchar | 50 | Name of Cardholder | John Doe |
| Card\_Number | varchar | 50 | Number of cards | 6223223489202023 |
| Exp\_Month | varchar | 50 | Expiry month of cards | 06 |
| Exp\_Year | varchar | 50 | Expiry year of cards | 19 |
| CVV | varchar | 50 | CVV of cards | 192 |
| UserType | varchar | 50 | Type of a user | Student |

* + 1. **Course\_Materials**

This table consists of course materials with details of files uploaded by the lecturer for the enrolled students.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Name** | **Field Type** | **Field Length** | **Description** | **Example** |
| ID | Integer | **-------------** | Unique id | 1 |
| Subject\_Name | varchar | 50 | Subject Name | Python |
| Chapter | varchar | 50 | Chapter Name | Introduction to variables |
| FileName1 | varchar | 50 | Name of the uploaded file | Chapter\_1 |
| FileType1 | varchar | 50 | Extension of file | .ppt |
| Slides | varbinary | max | Lecture slides | Chapter\_1.ppt |
| FileName2 | varchar | 50 | Name of the uploaded file | Tutorial\_1 |
| FileType2 | varchar | 50 | Extension of file | .pdf |
| Tutorials | varbinary | max | Lecture tutorials | Tutorial\_1.pdf |

* + 1. **Courses**

This table consists of the details of page contents in a certain course page of krypton education.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Name** | **Field Type** | **Field Length** | **Description** | **Example** |
| ID | Integer | **-------------** | Unique id | 1 |
| Course\_Name | varchar | 50 | Name of the course | Python |
| Course\_Heading | varchar | 50 | Name of the course | General Purpose Programming Language |
| Lecturer\_Name | varchar | 50 | Full Name of the Lecturer | John Doe |
| Lecturer\_Experience | varchar | 50 | Teaching experience of a  Lecturer | 10 Years |
| Lectuere\_Email | varchar | 50 | Email of the lecturer | Johndoe@gmail.com |
| Class\_Duration | varchar | 50 | Overall teaching hours for a course | 150 Hrs |
| Lecture\_Hours | varchar | 50 | Lecture hours per day | 1. Hrs |

## Wire Framing

**3.3.1 Homepage**

The home page is the initial landing page of Student Krypton Education. The main page has a navigation bar at the top of the page with information about the college such as courses, join now, and contact us. The body section offers online learning resources as well as a summary of the courses for students.

**Graphical user interface

Description automatically generated**

**3.3.2 Contact us**

**Graphical user interface

Description automatically generated**

**3.3.3 Login**

The login page has an email address and password box where users must enter their email address and password as supplied by them.

**Table

Description automatically generated**

**3.3.3 Course Enroll**

**Graphical user interface

Description automatically generated**

**3.3.4 Admin Panel**

This is an admin dashboard that will appear only when an administrator signs in with his or her account. This panel is only accessible to approved Krypton Education personnel.

**Graphical user interface, application

Description automatically generated**

**3.3.5 Lecture Panel**

**Graphical user interface

Description automatically generated**

## Website Navigational structure

**3.4.1 Admin**

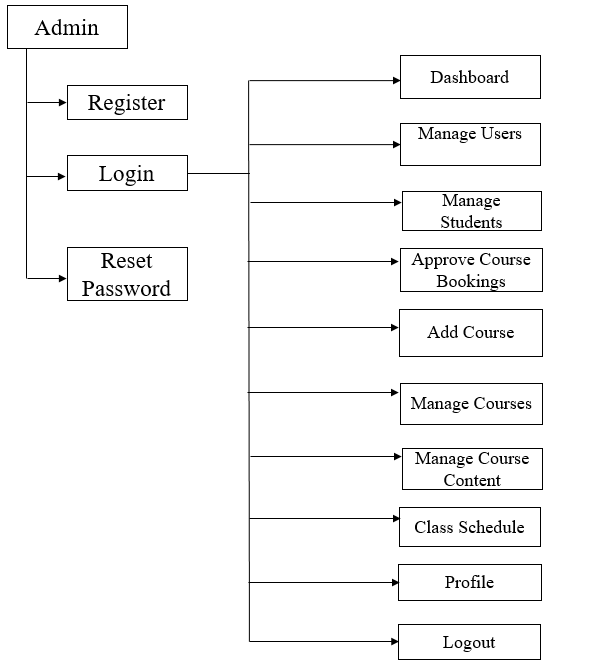
****

Fig: Navigational structure of Admin

**Description**

Admin can register, log in, and reset their password. After successful login, they can perform various operations to manage the entire website of krypton education. The main menus and the operations which they perform inside it are mentioned below:

1. Manage users: Add, update, and delete users and lecturer accounts
2. Manage students: Add, update, and delete enrolled students
3. Approve course bookings: Approve and delete course bookings
4. Add course: Add new Course
5. Manage courses: Update and delete courses
6. Manage course contents: Add, update, and delete the details displayed on the course page
7. Class schedule: Add, update and delete the class schedule
8. Profile: Update individual profile
9. Logout: Logout from the panel
   * 1. **Lecturers**

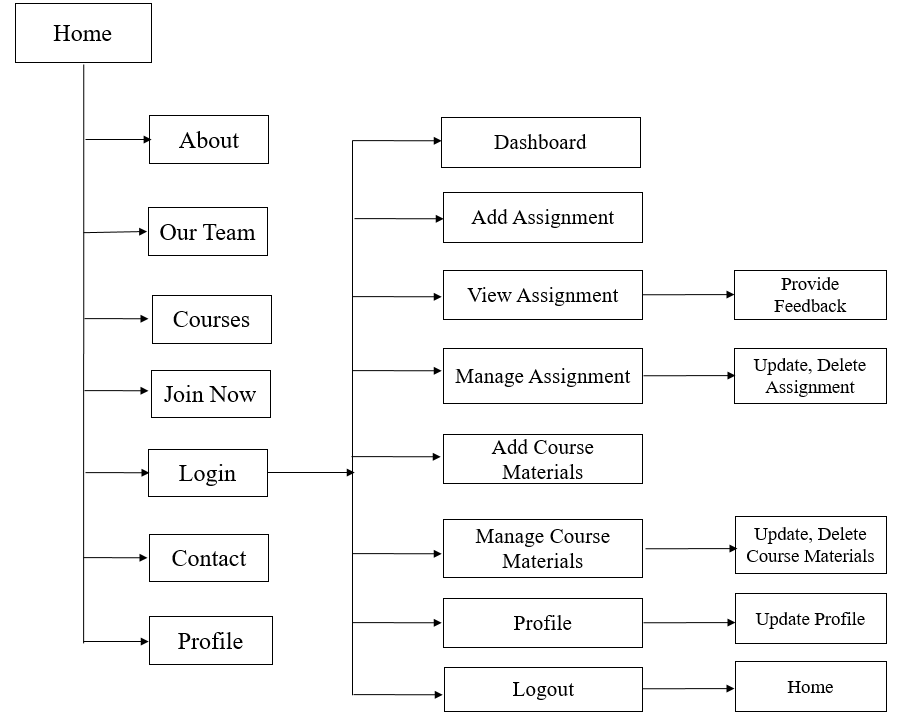
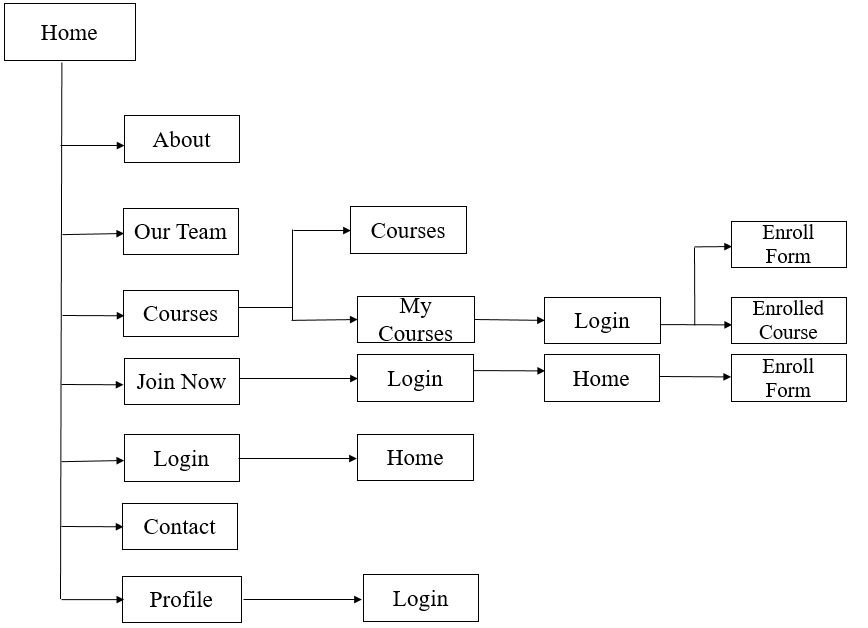
****

Fig: Navigational structure of Lecturers

**Description**

Lecturers in Krypton education can log in with the account created by the admins. They are deprived of creating a lecturer account themselves. But they can reset their password themselves. After logging in successfully the main menus and their operations that a lecturer can perform are mentioned below:

1. Add assignment: Add assignment along with the date of hand in, hand out, and question
2. View assignment: View assignment uploaded by the students and provide feedback
3. Manage assignment: Update and delete the uploaded assignment of students
4. Add course materials: Add course-related materials like slides and tutorials
5. Manage course materials: Update and delete the uploaded course materials
6. Profile: Update individual profile
7. Logout: Logout from the lecturer panel
   * 1. **Users/ Students**

**Fig: Navigational structure of Users/ students**

**Description**

We categorized the types of users for our system. To enroll, a user should have an account using which they log in first and can only access the enroll form. Without an account, a user in our system can only view available courses, our team, and contact us. Similarly, to enroll in a course a user should log in using his/her account. An account must be created to reset the password. Similarly, to view a particular course page a user must be enrolled and approved by the admin. For this, they should log in at first then after clicking they will redirect to their particular course page where they can view and upload assignments, view feedback provided by the lecturer, and access to the course materials which includes the chapter slides and tutorials.

# Implementation (Code Snippet)

* 1. **Insert**

****

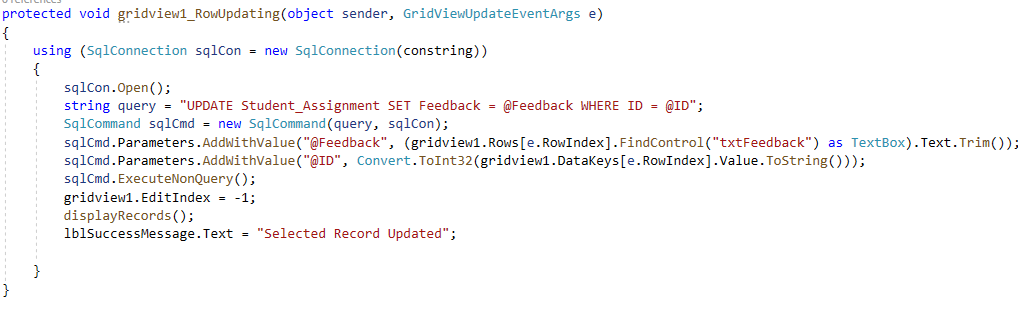
Fig: Code for insert operations

In the above code, an insert query is used to insert data into the database. The insert query is stored inside a string. Further, then the connection and the query string are passed inside the SQL command which is executed using ExecuteNonQuery. Similarly, a script manager is used where the alert message is passed to provide the alert message in the client-side server when a user sent a message.

* 1. **Display**

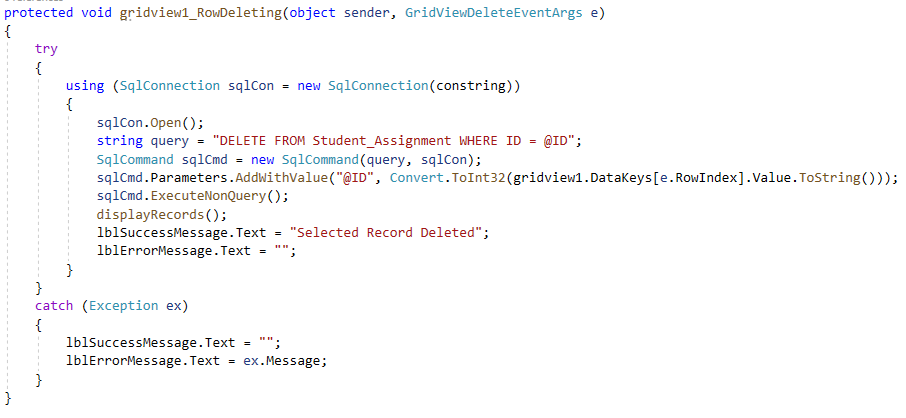
Fig: Code for display operation

In the above code, a method is made where a connection string is used which is passed inside the SQL connection. Similarly, a select query is used to select all the data available in the “AvailableCourse” table. Then the connection is made open. Then using execute reader the SQL command will return the data inside the “AvailableCourse” and using the data-bind method the data will be displayed.

* 1. **Update**

**Fig: Code for an update operation**

In the above code, the connection string is passed inside the SQL connection object using which the update operation is executed. At first, the SQL connection is made open so that the update operation can be executed in the database of that connection. A query string is used which is passed inside the SQL command along with the SQL connection. Similarly, using execute non-query method the string query is executed inside the specified connection. The updated data after updating a row is displayed in the grid view using the display records method.

* 1. **Delete**

**Fig: Code for delete operation**

In the above code, the try and catch are used to handle the exception to avoid user errors. The connection string is passed inside the SQL connection object using which the delete operation is executed. At first, the SQL connection is made open so that the delete operation can be executed in the database of that connection. A query string is used which is passed inside the SQL command along with the SQL connection. Similarly, using execute non-query method the string query is executed inside the specified connection. The updated data after successfully deleting a row is displayed in the grid view using the displayRecords method.

* 1. **Validation**

Fig: Use of validation

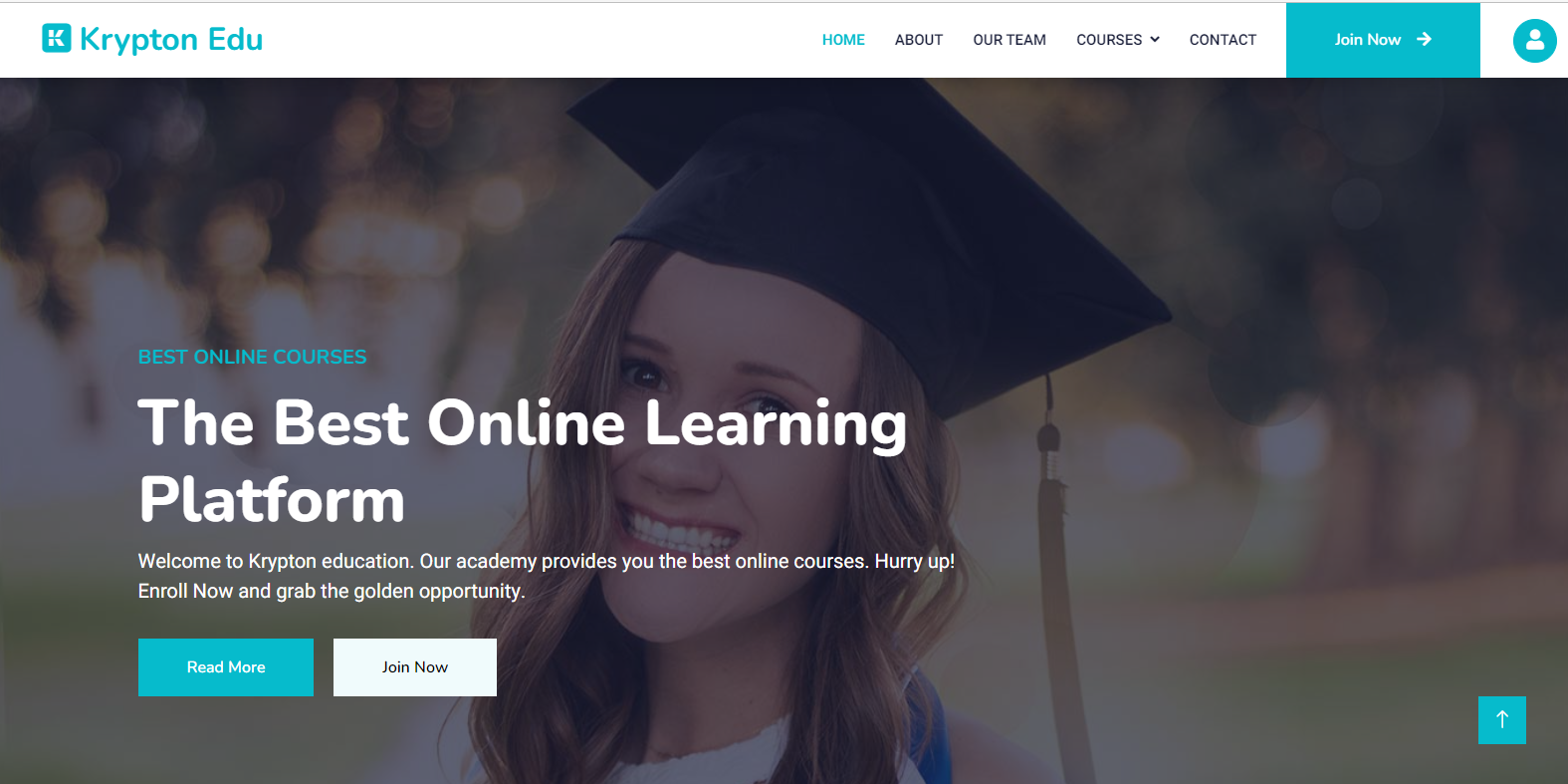
Required field validator is used to check whether a text field is empty or not. This will help to avoid the logical error in the system by providing validation to the text fields.

* 1. **Session variable**

**Fig: Use of session variable**

In the above code, we have used the session variable to store the data of a user who logs in our system. For this, the SQL connection is made open at first so that the command can be executed in the database of that connection. Then Select command is used to select the data which matches the email and password entered in the textboxes. If the data is found inside the database, then the required data will be retrieved from the specified column using the SQL command object where the select query is stored. The data will be stored in string format which is further then stored inside the session variable. As result, the session variable or the method can be called wherever it is necessary.

# User Guidance

* 1. **User**
     1. **Home**

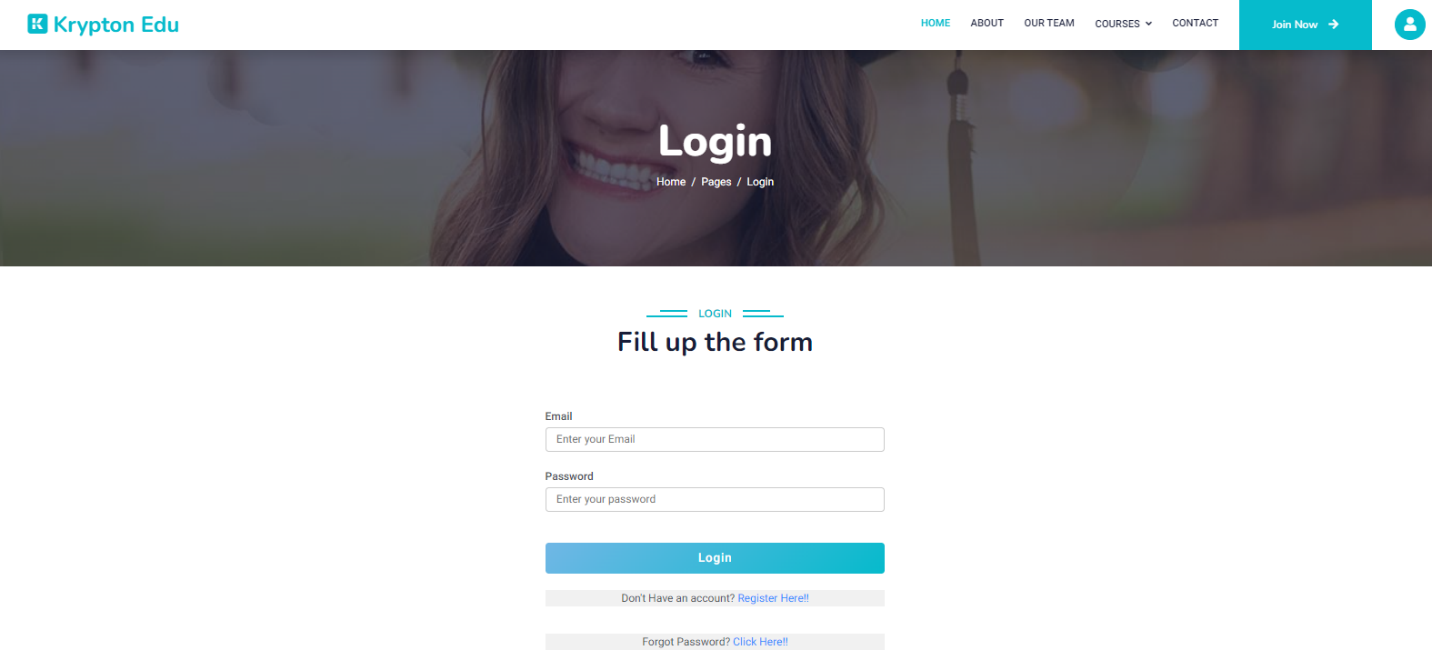
**Fig: Home page**

This is the main page of our system. All the details are provided on this page. Users can also navigate to other pages for more details from the navigation bar.

* + 1. **Course Page**

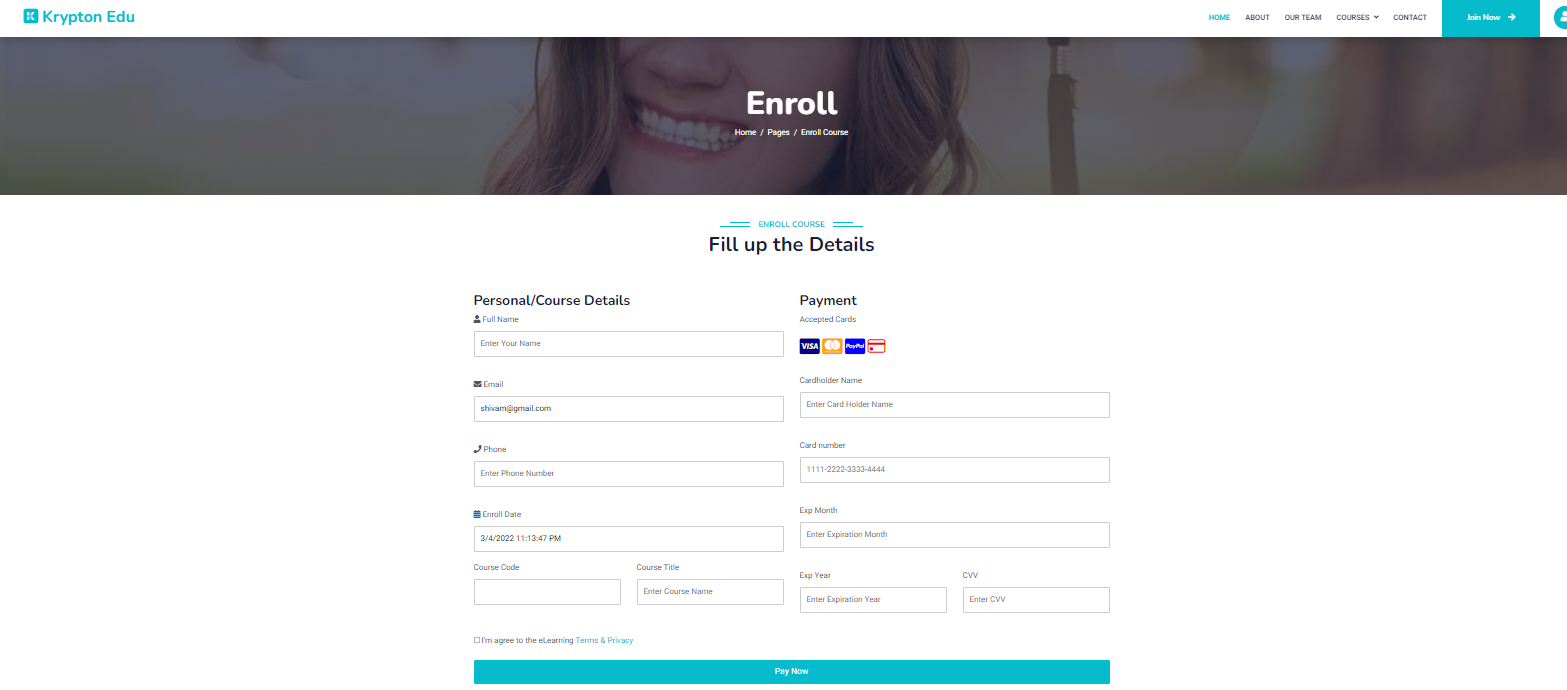


**Fig: Course Page**

* + 1. **Login Page**

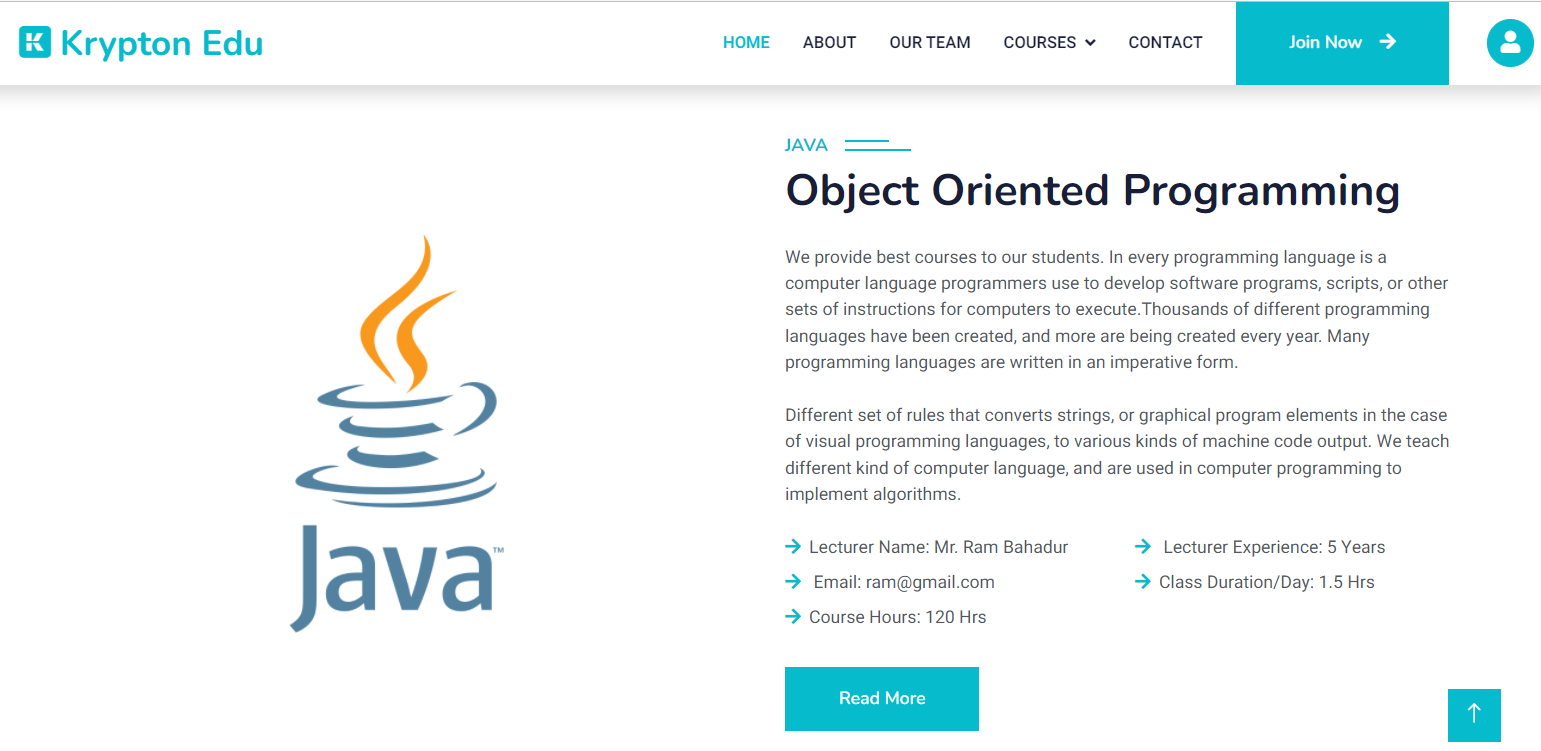
**Fig: Login Page**

Users can log in to our system by fill the login form. Similarly, they can also navigate to the register and reset password page from this page.

* + 1. **Course Enroll Page**

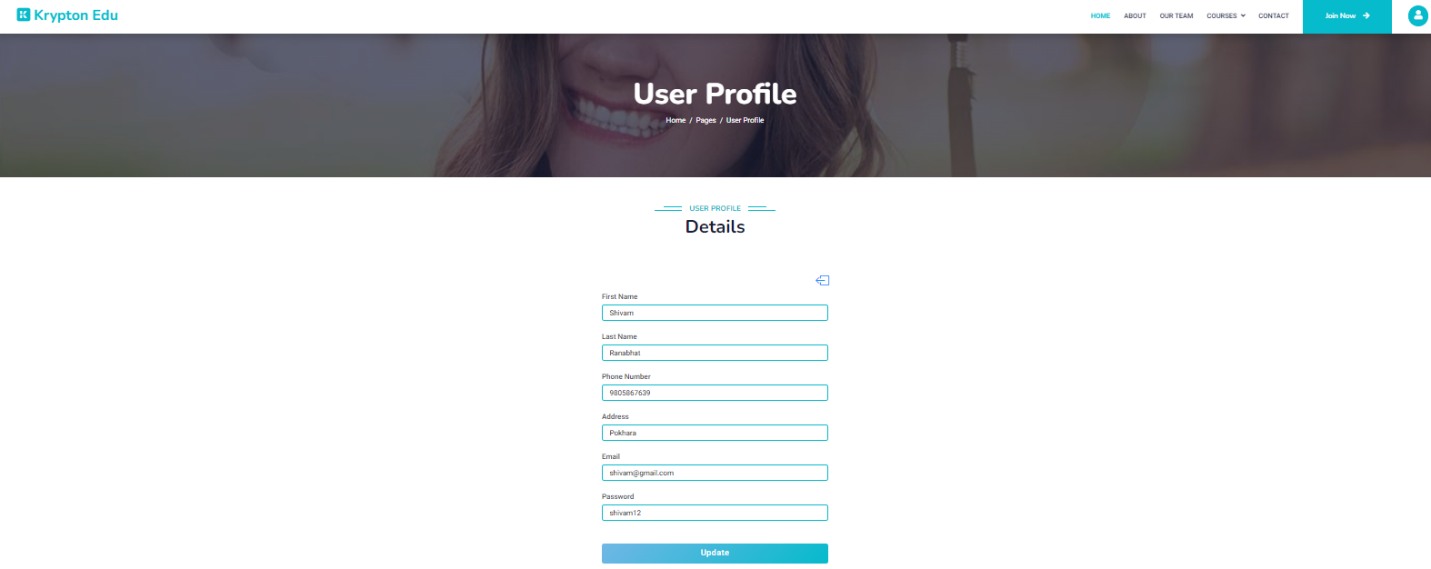
**Fig: Course enroll page**

After logging in, users can enroll in their desired course by filling out the course enrollment form

* + 1. **Enrolled course Page**

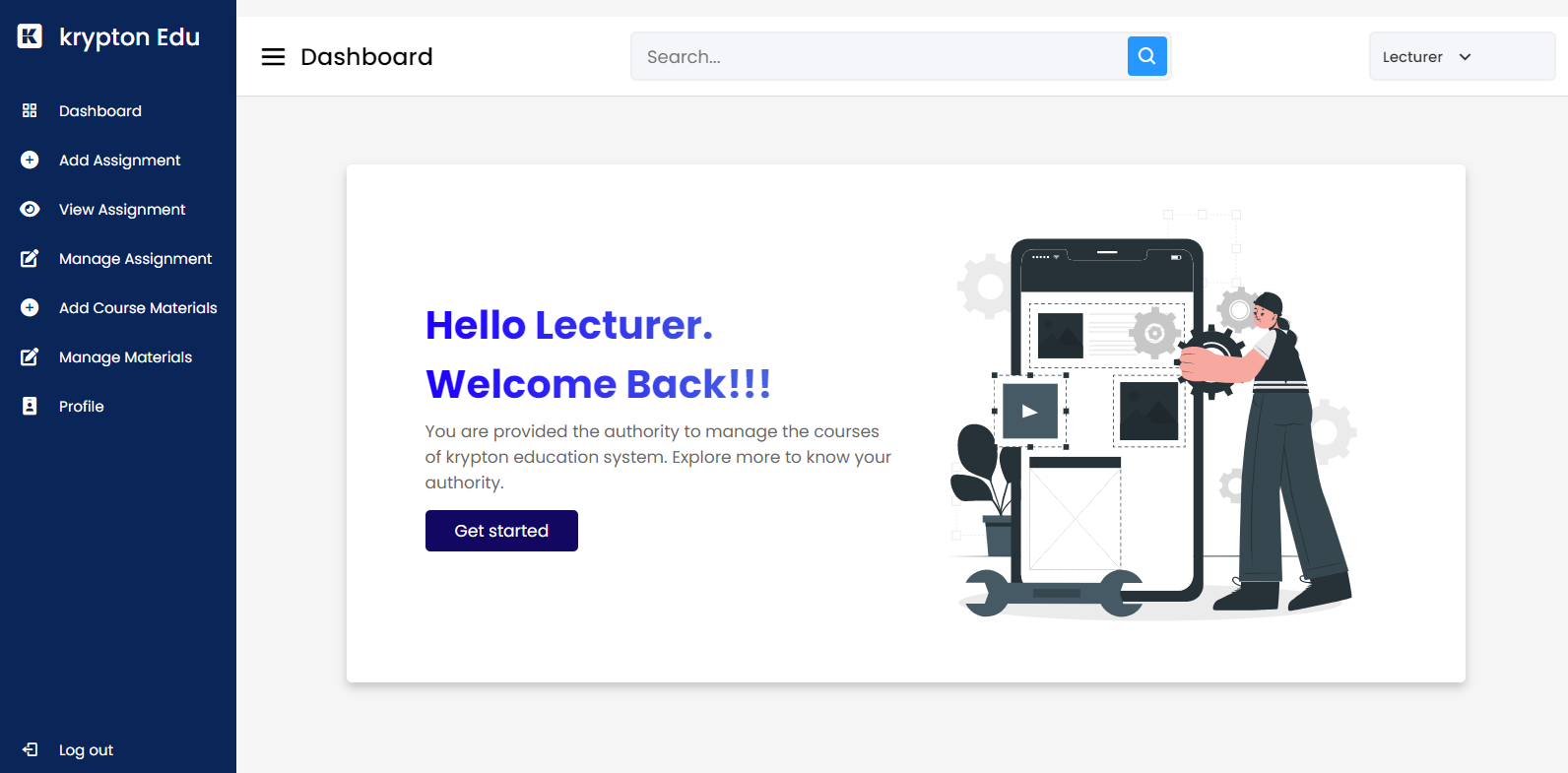
**Fig: Enroll course page**

This page will be displayed if a user is enrolled in a particular course. After logging in, the user will reach this page from the sub-menu i.e., My Course which is arranged in the navigation bar under the main menu i.e., courses.

* + 1. **User profile page**

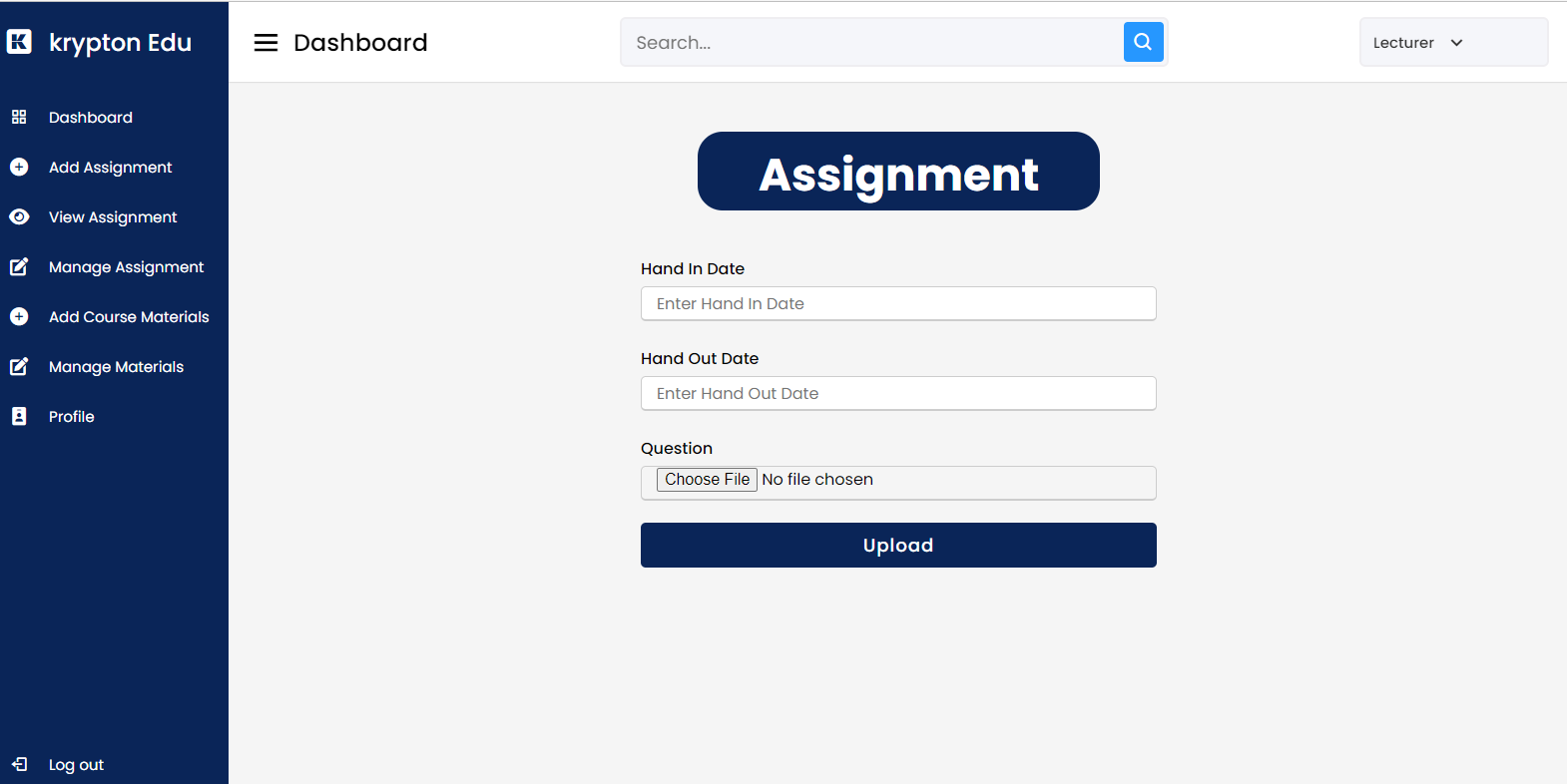
**Fig: User profile page**

On this page, the details of a user will be displayed after logging in. They can update their details from this page.

* 1. **Lecture Panel**
     1. ****Dashboard

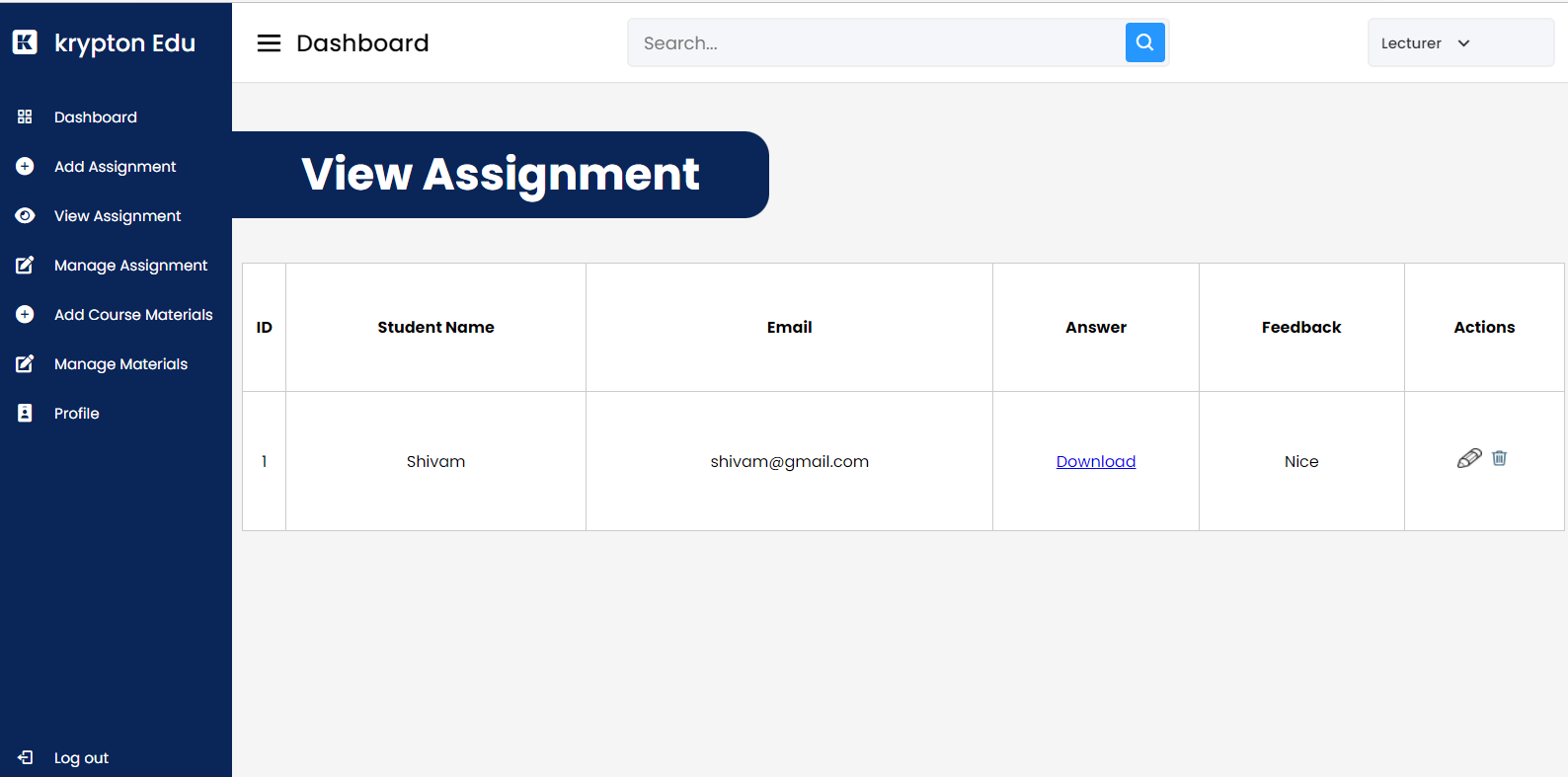
**Fig: Lecturer dashboard**

This is the lecturer dashboard which is displayed only when a lecturer logged in from the krypton education website.

* + 1. ** Add assignment Page**

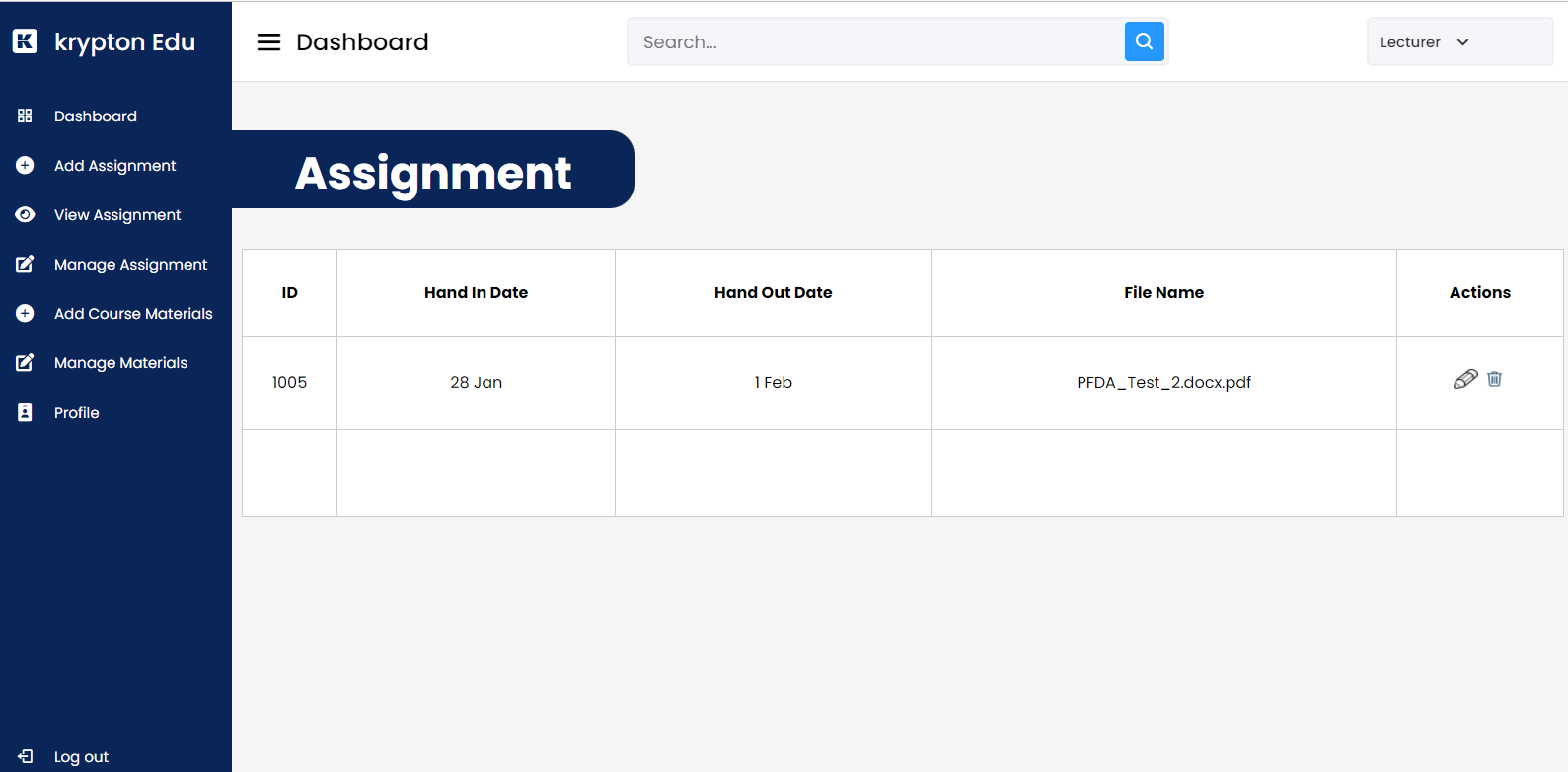
**Fig: Add assignment page**

On this page, lecturers can add assignments for their particular subject. They need to provide the date of hand-in and hand-out with the question file.

**5.2.3 View assignment Page**

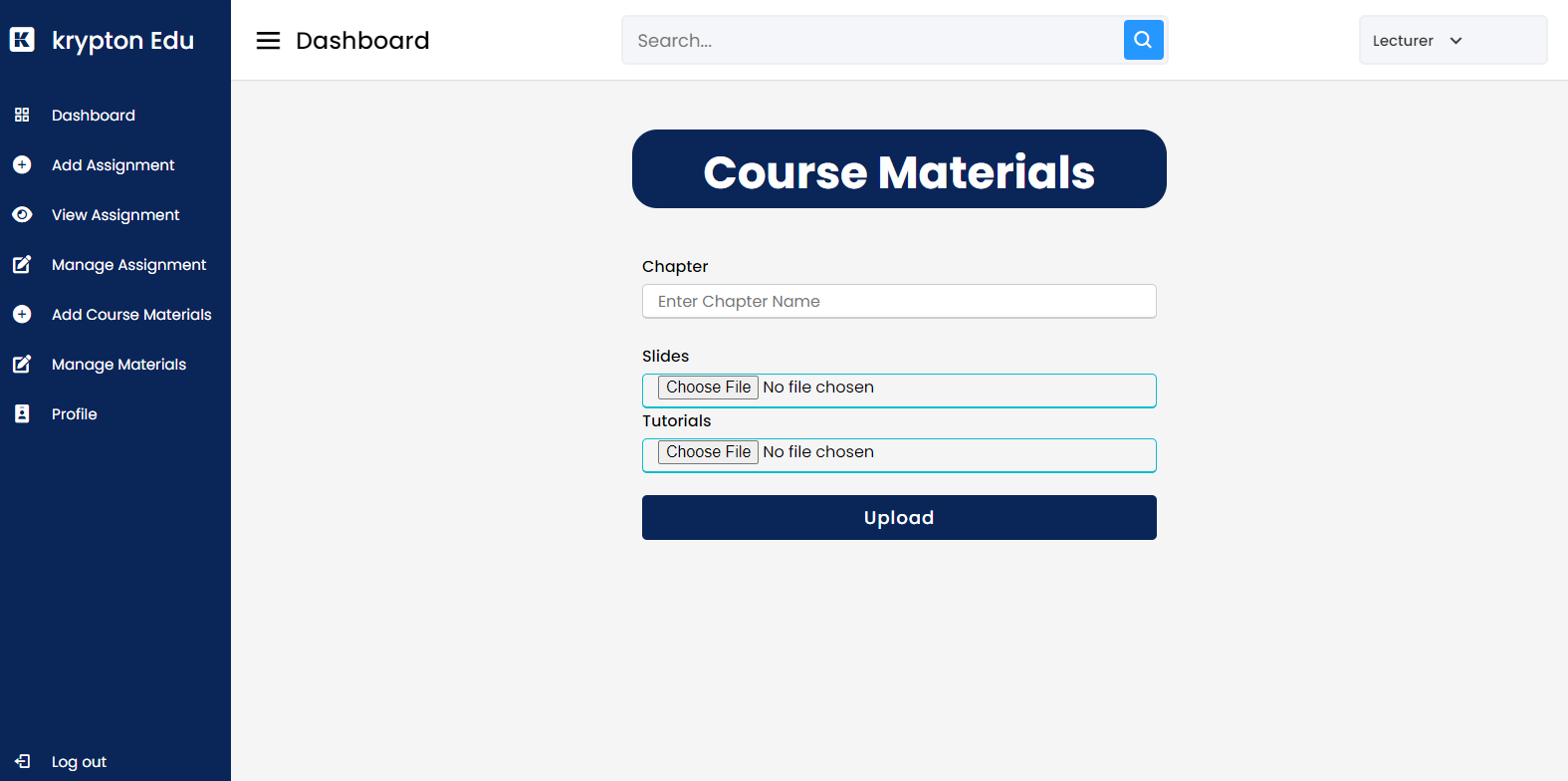
**Fig: View assignment page**

On this page, the lecturer can view assignments uploaded by their students and they can provide feedback individually in response to the assignment uploaded by the student.

**Manage assignment Page**

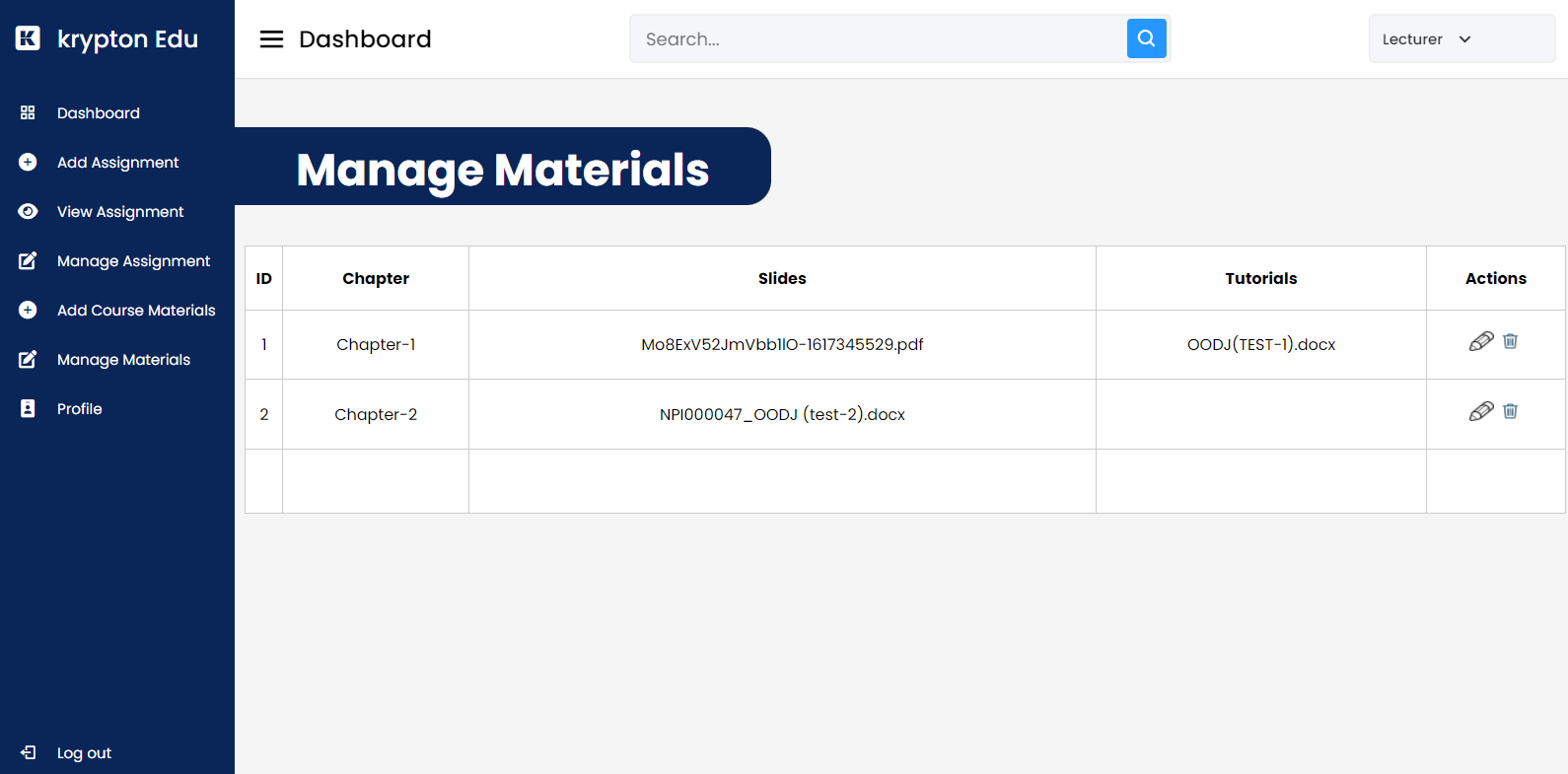
**Fig: Manage assignment page**

On this page, the lecturer can manage the assignment including the date of hand-in and hand-out. They can update and delete the uploaded assignment.

* + 1. **Add course Materials Page**

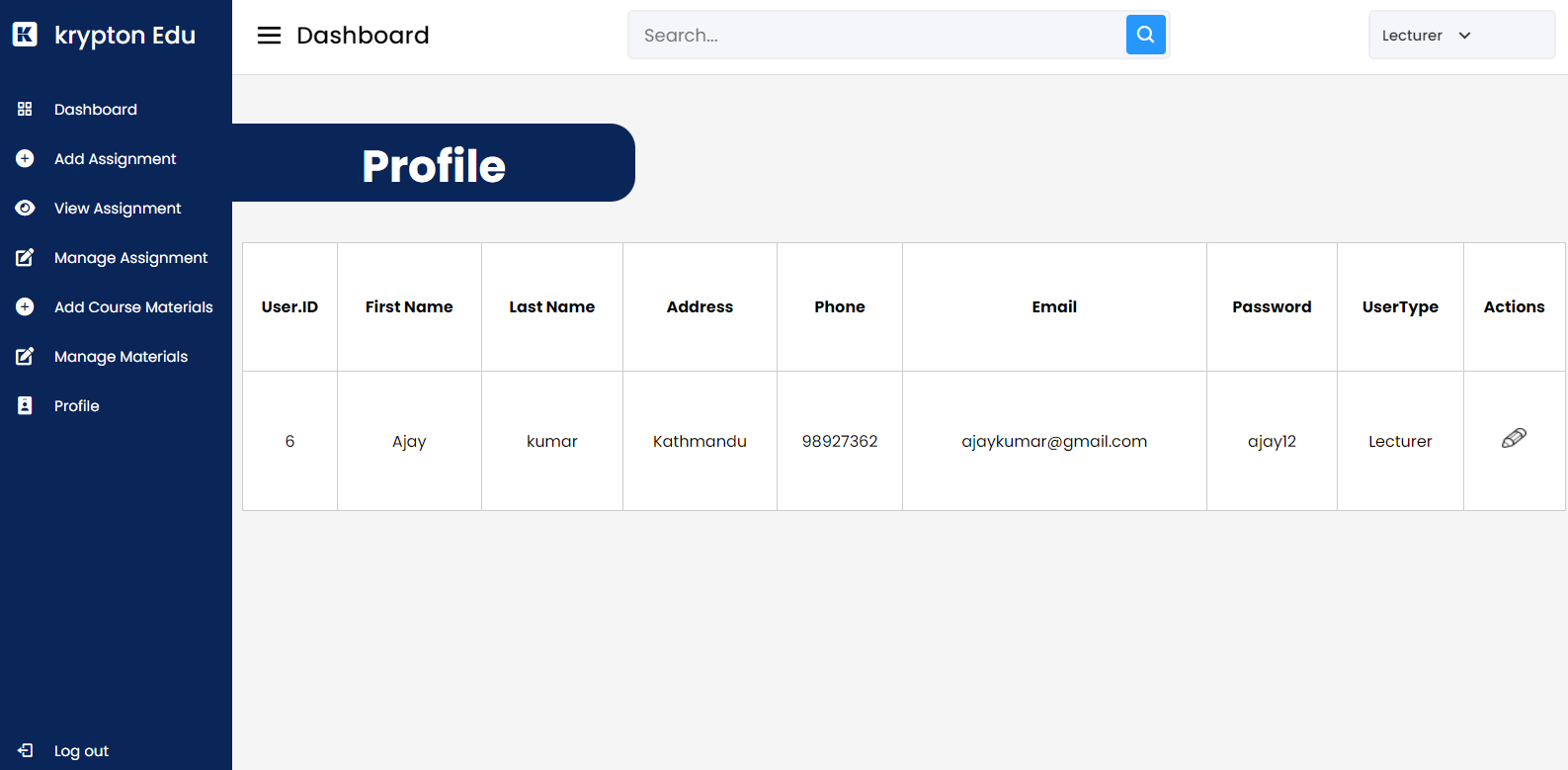
**Fig: Add course materials page**

On this page, the lecturer can add the course-related materials which include a chapter name, slides, and tutorials of that chapter. These will display on the course page of their related subject.

* + 1. **Manage course materials Page**

**Fig: Manage course materials page**

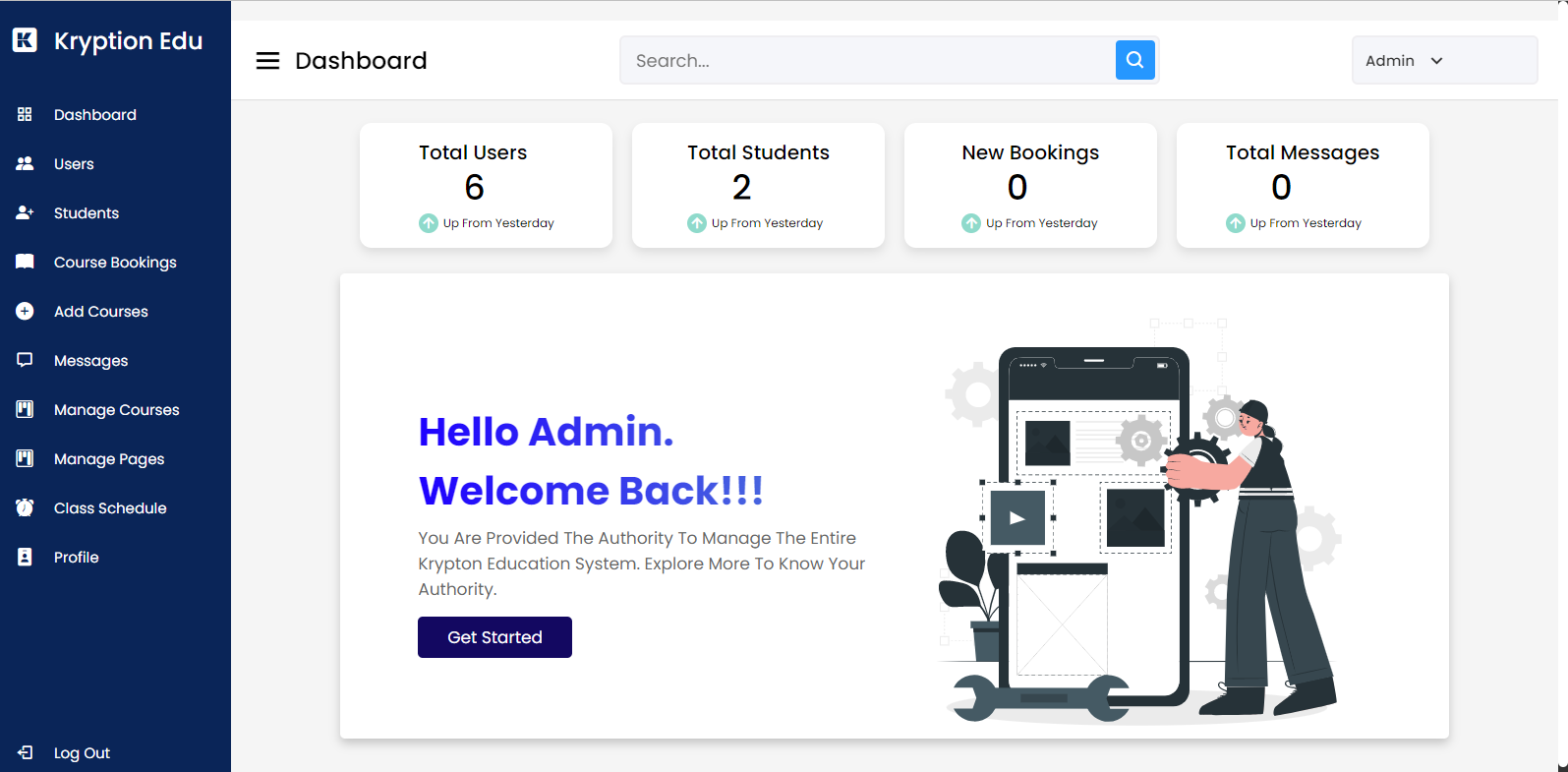
On this page, the lecturer can manage the materials which they uploaded from the add assignment page. They can delete and update the chapter, slides, and tutorial name.

* + 1. **Profile Page**

**Fig: Profile page**

On this page, the profile of the lecturer will be displayed where they can update their profile.

* 1. **Admin Panel**
     1. **Admin Dashboard**



**Fig: Admin dashboard**

This is an admin dashboard that will be displayed only when an admin logs in using his/her account. The authorized person of the krypton education is allowed to access this panel.

* + 1. **Account details page**

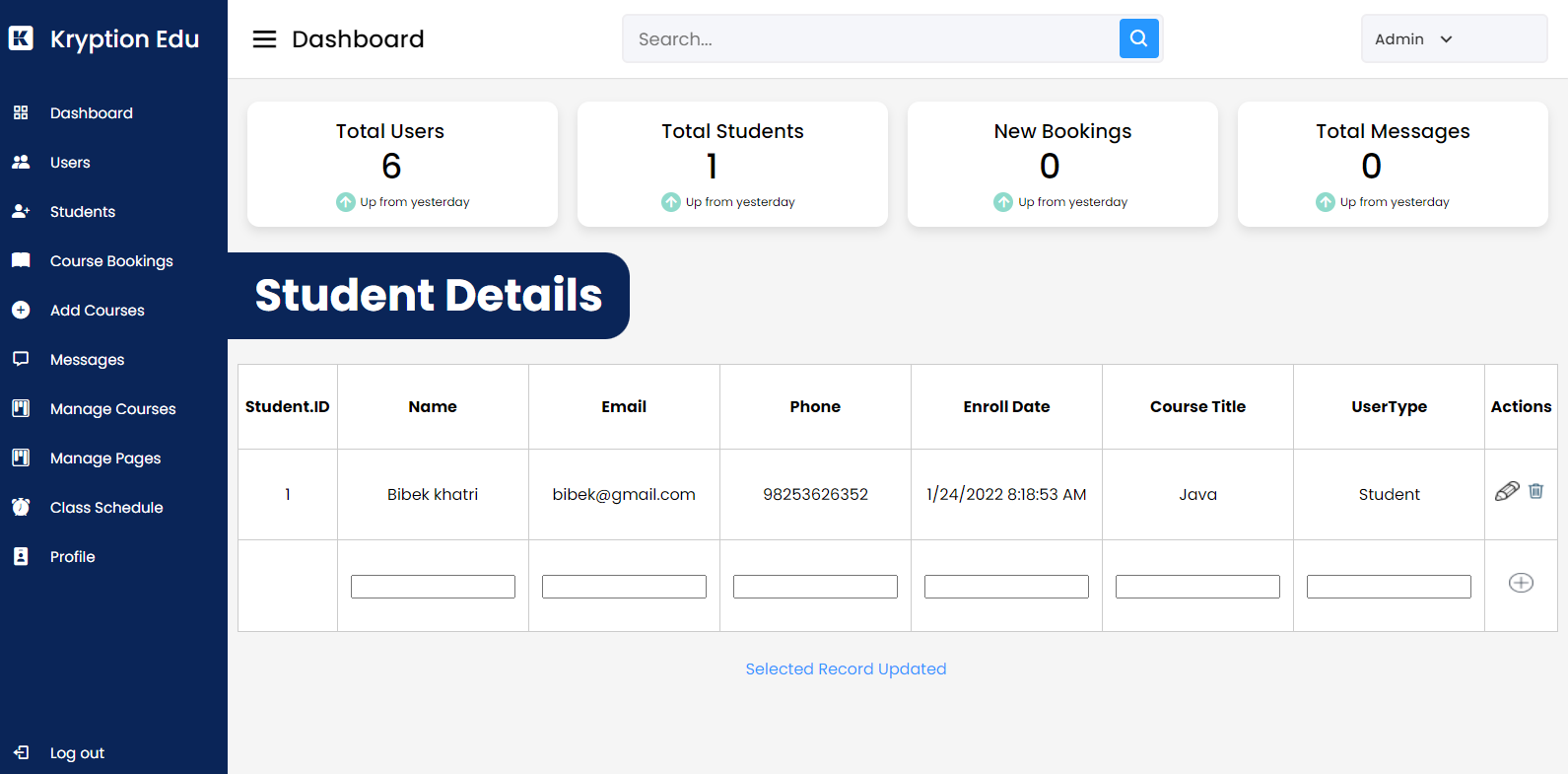
Graphical user interface

Description automatically generated

**Fig: Account details page**

This page contains all the details of the users and lecturer. Admin has the authority to add new, update, and delete a user or lecturer account.

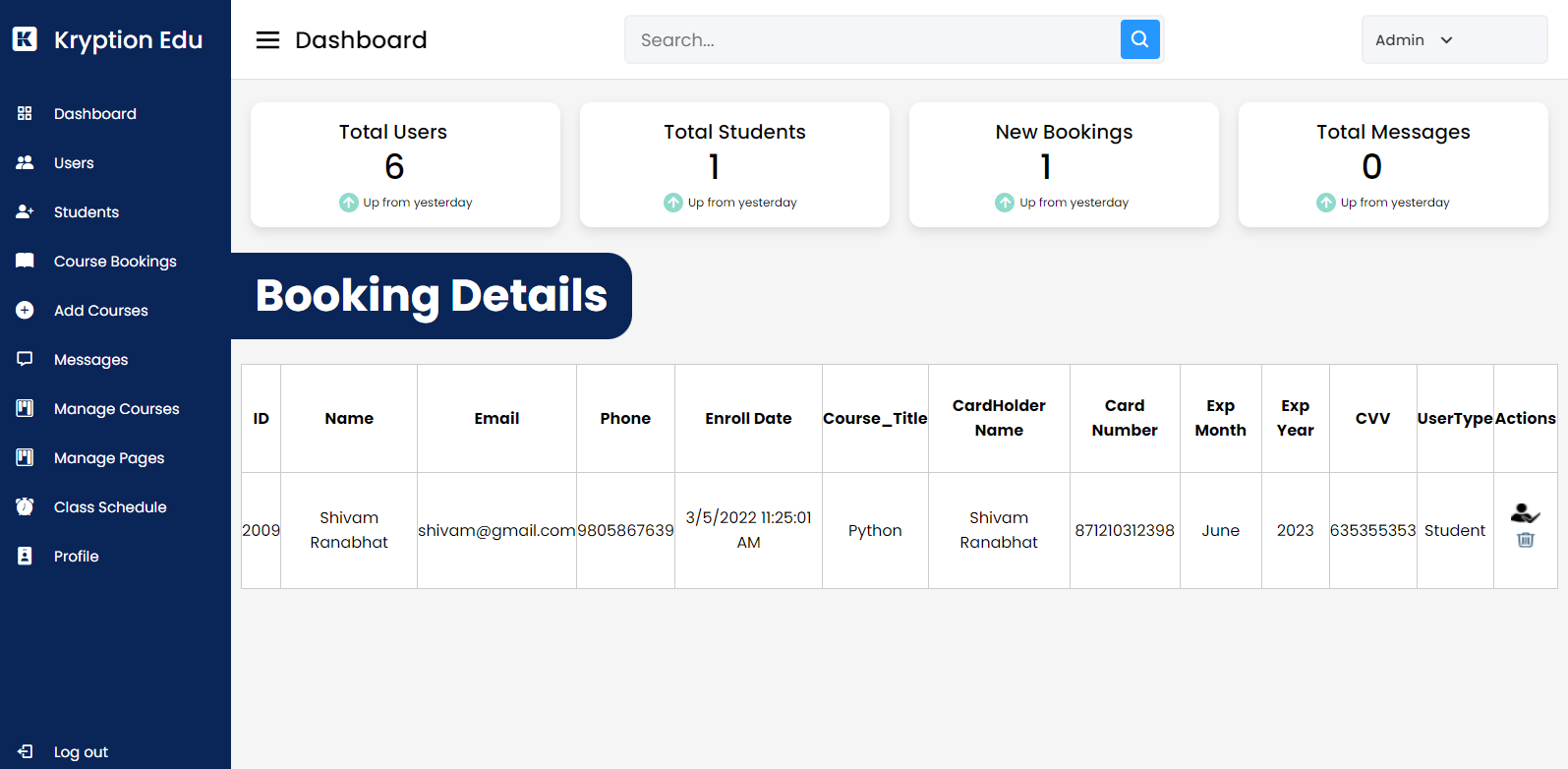
* + 1. **Student details page**



**Fig: Student details page**

This page contains all the details of the enrolled students. Admin has the authority to add new, update, and delete an enrolled student account.

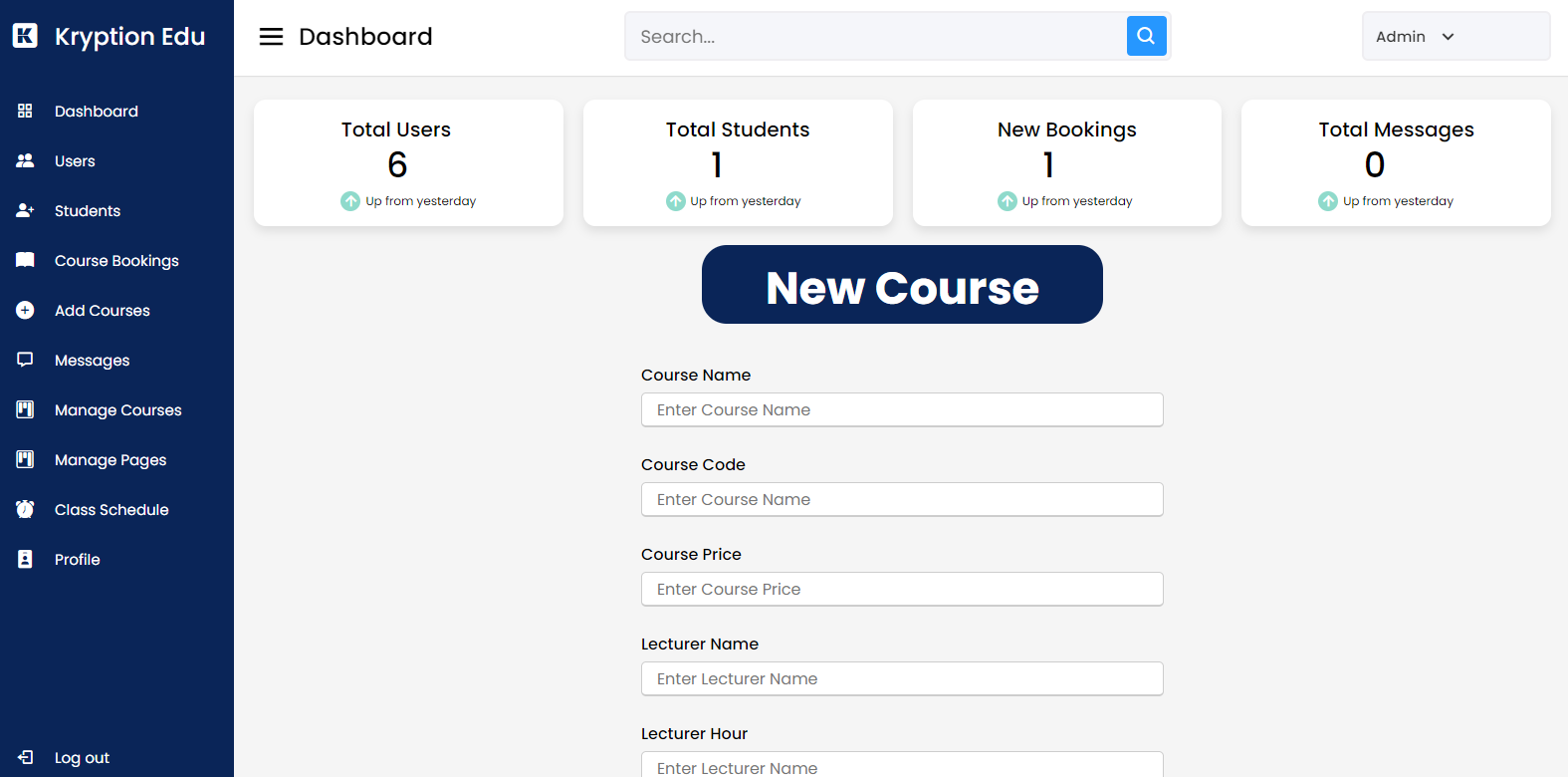
* + 1. **Booking details page**



**Fig: Booking details page**

This page contains all the details of a user when he/she apply for a particular course. Through this page, an admin can approve the user as a student of krypton education if he/she had paid for his/her selected course. If not the admin can delete the booking of course.

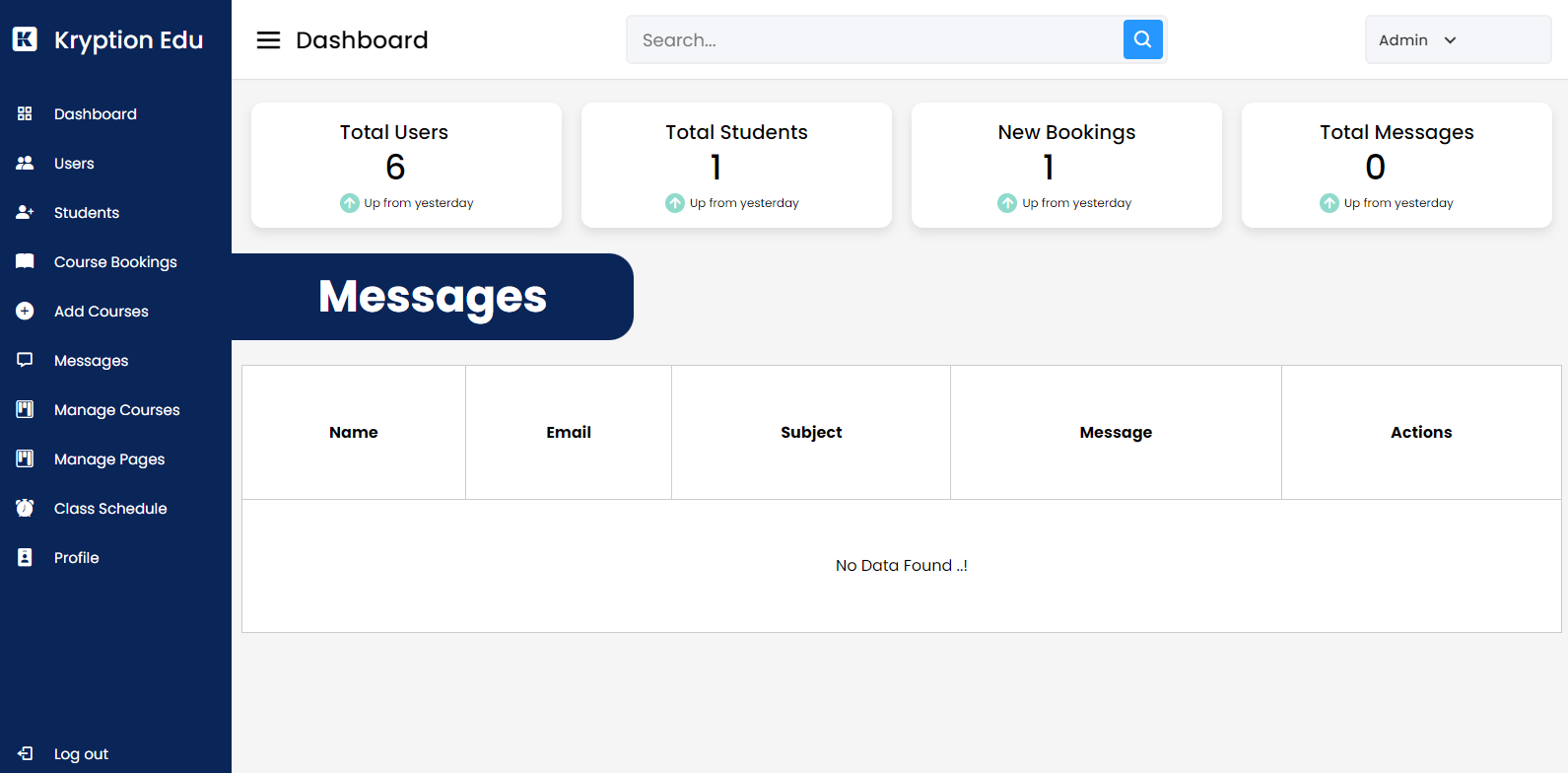
* + 1. **Add course page**



**Fig: Add course page**

From this page, an admin can add a new course along with the course image which will be displayed on the user side server so that they can apply for it.

* + 1. M**anage messages page**



**Fig: Manage message page**

Whenever a user tries to contact the admin of krypton education through the contact us page, his/her message will be displayed here. An admin can view and delete the messages.

* + 1. **Manage courses page**

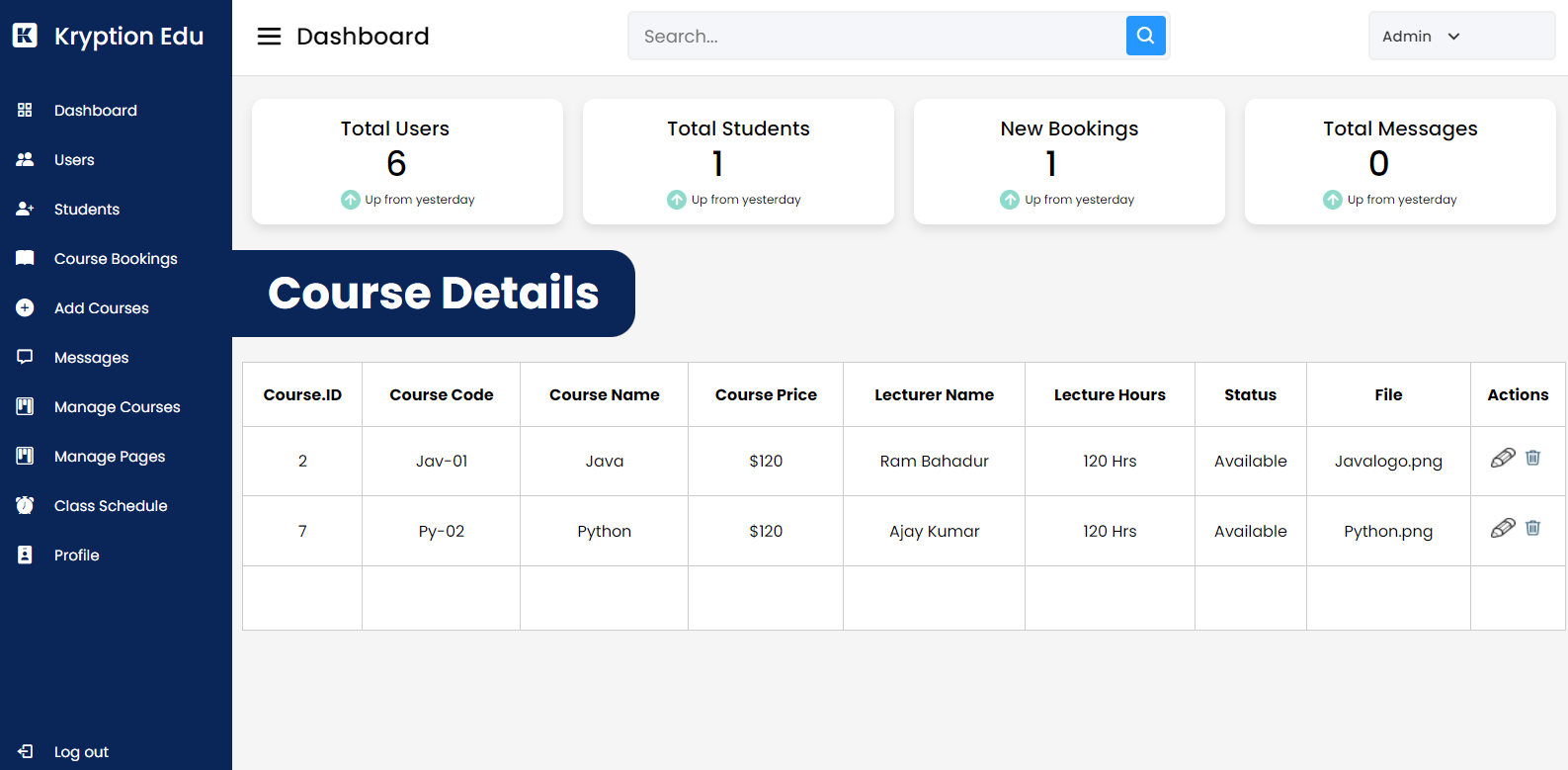


Fig: Manage courses page

On this page, the added course and its information are displayed. An admin can update and delete the courses from this page.

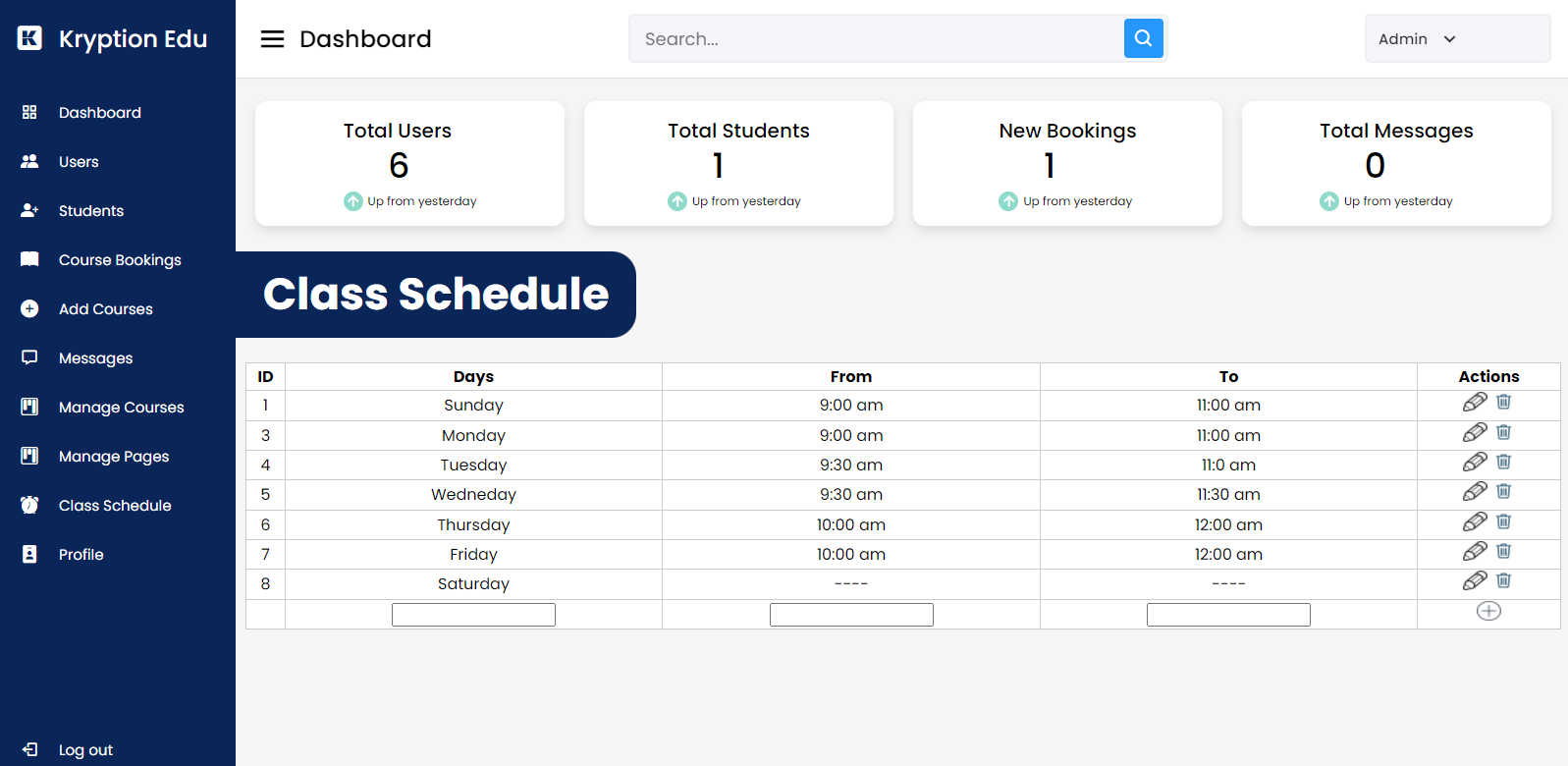
* + 1. **Manage course page details**



**Fig: Manage course page details**

All the details or content displayed on a particular course page are displayed on this page. An admin can add, update and delete the details displayed on the course page.

* + 1. **Class schedule page**



**Fig: Class schedule page**

This page contains the details of the class schedule. An admin can add, update and delete the class schedule.

* + 1. **Profile page**



**Fig: Profile page**

This page contains the details of the admin account. An admin can update his/her account from this page.

# Conclusion

The four-person team completed this project with a lot of hard work and outstanding collaboration from each member. We all share our responsibilities fairly and are always eager to help one another if we encounter difficulties while carrying them out. Everyone on the team is continuously working hard to create a high-quality website.

In this project, we build a web service for the e-learning system. This approach is intended for students who want to expand their knowledge and develop their skills. In the system, we constructed a variety of functions that combine JavaScript, ASP, C#, and CSS, such as a learning material search engine that pulls data from a database, data viewing, and so on. We also learn how to utilize bootstrap and build a robust web page, as well as how to connect the database to the data on the web page and communicate that data via the database. This project taught us the basics of web programming and prepared us to work as web developers.

## Future Enhancement

* + A discussion board for students to communicate with one another.
  + A live class video upload function for teachers to help students learn more readily.
  + The system must support many payment methods, such as PayPal debit cards, credit cards, and so on.
  + The quiz program and the prize for the winner Which method will be most effective?
  + The current user experience is fine, but to increase traffic and make it easier for students, we must upgrade it.

# References

Hallissey, T. (2017). *What is a Target Audience, and why is it so Important*?. Retrieved 1 February 2022, from https://www.fatguymedia.com/inbound-marketing/target-audience/?fbclid=IwAR3T6fYa4Ma8Qn0JuDlR90gHnnGgdT4EQ2lXJNnPvE9WCVp4o0kF4uMTn8I(2022). Retrieved 10 February 2022, from <https://www.skyword.com/contentstandard/?fbclid=IwAR3q4Czm0Du7aIk1CcrsM1j2VyYNNpD7x1pi6B9jLtFPKAAqYjtO2SpxO6>M

Rosencrance, L. (2019). *Software Requirements Specification (SRS*). Retrieved 6 February 2022, from <https://www.techtarget.com/searchsoftwarequality/definition/software-requirements-specification?fbclid=IwAR3QyAz3P06G0b4K8GVK14zBHS1vdIJ9wp3EeAXpwFWAUMimeFhwQQFt2ao>

Casteleyn, S., & Troyer, O. *Structuring Web Sites Using Audience Class Hierarchies*. Retrieved 9 February 2022, from <https://www.academia.edu/369185/Structuring_Web_Sites_Using_Audience_Class_Hierarchies>

Pixel, D. (2018). *Website Structure and Navigation: Basic Web Design Rules.* Retrieved 12 February 2022, from <https://medium.com/@crazyleaf/website-structure-and-navigation-basic-web-design-rules-1314817541be>

Kumar, M. (2015). Session In ASP.NET. Retrieved 12 February 2022, from <https://www.c-sharpcorner.com/UploadFile/8a67c0/session-in-Asp-Net/?fbclid=IwAR2OU4mD2ZMca86nRzWr426kALRkI437NMIzskOxzmgtUIxdR2BF4DnD9lI#:~:text=Sessions%20are%20used%20to%20maintain,It%20is%20secure>

Schneider, D. (2021). *How to Use Website Demographics Effectively in Your Audience Analysis*. Retrieved 23 February 2022, from <https://www.similarweb.com/corp/blog/research/market-research/website-demographics/?fbclid=IwAR1Pk1zUARAkbWvm_NU3vYMftAuHltb9qD1BDHNSqCxh82yFngnvzNQTIQw>

HANNAH, J. (2021). *What Is A Wireframe?* The 2022 Wireframing Guide. Retrieved 24 February 2022, from <https://careerfoundry.com/en/blog/ux-design/what-is-a-wireframe-guide>

# Workload Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Topics | Bikash Baral (NPI000021) | Kapil Pokhrel (NPI000030) | Saroj Kandel (NPI0000) | Shivam Ranabhat (NPI000047) |
| Introduction/ Project Plan |  | 100% |  |  |
| Requirement Specification | 25% | 25% | 25% | 25% |
| Design and Modeling | 25% | 25% | 25% | 25% |
| Implementation |  |  |  | 100% |
| User Guidance | 100% |  |  |  |
| Conclusions |  |  | 100% |  |