

# WAPP

*by Kapil Pokhrel*

---

**Submission date:** 06-Mar-2022 03:14AM (UTC+0800)

**Submission ID:** 1777206047

**File name:** WebAssignment\_Final.docx (4.15M)

**Word count:** 4137

**Character count:** 21586

1

## Contents

|       |   |    |
|-------|---|----|
| 1.    | Introduction .....                          | 2  |
| 1.1   | Objectives .....                            | 2  |
| 1.2   | Scopes .....                                | 2  |
| 1.3   | Project Schedule .....                      | 2  |
| 2.    | Requirement Specification .....             | 2  |
| 2.1   | Audience Modelling.....                     | 3  |
| 2.1.3 | Major functions of the Website.....         | 4  |
| 3.    | Design and Modeling.....                    | 5  |
| 1     | 3.1 Entity Relationship Diagram (ERD) ..... | 5  |
| 3.2   | Data Dictionary .....                       | 6  |
| 3.3   | Wire Framing .....                          | 13 |
| 3.4   | Website Navigational structure .....        | 16 |
| 4.    | Implementation (Code Snippet).....          | 19 |
| 5.    | User Guidance .....                         | 22 |
| 6.    | Conclusion .....                            | 34 |
| 6.2   | Future Enhancement .....                    | 34 |

## **1. 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.

### **1.3 Project Schedule**

## **2. 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 to 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.

### 2.1.1 Audience categorization

#### i. Administration / Super User

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.

#### ii. 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.

#### iii. 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.

#### iv. 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.

### **2.1.2 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.

### 3. Design and Modeling

#### 3.1 Entity Relationship Diagram (ERD)

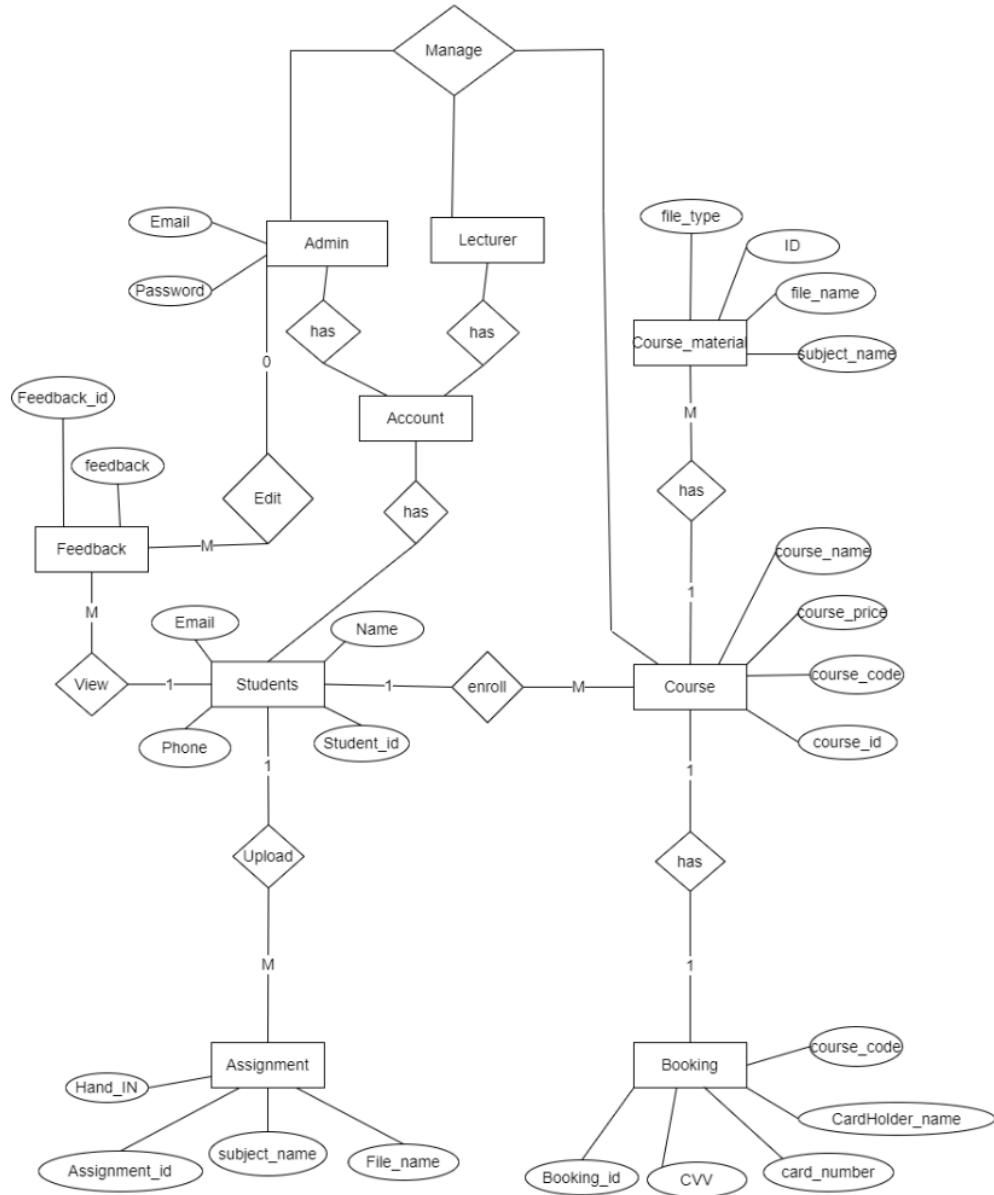


Figure 1: ERD Diagram of Krypton Education

## 3.2 Data Dictionary

### 3.2.1 Users

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

| Variable   | Field Type | Field Length | Description                 | Example             |
|------------|------------|--------------|-----------------------------|---------------------|
| Name       |            |              |                             |                     |
| 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                |

### 3.2.2 Students

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

| Variable | Field Type | Field Length | Description            | Example |
|----------|------------|--------------|------------------------|---------|
| Name     |            |              |                        |         |
| ID       | Integer    | -----        | Unique Id of a student | 1       |
| Name     | varchar    | 50           | Name of a Student      | John    |

|   |              |         |    |  |                     |
|---|--------------|---------|----|--|---------------------|
| 3 | 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             |

### 3.2.3 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     | Field Type | Field Length | Description              | Example             |
|--------------|------------|--------------|--------------------------|---------------------|
| Name         |            |              |                          |                     |
| ID           | Integer    | -----        | Unique Id                | 1                   |
| Student_Name | varchar    | 50           | Name of a Student        | John                |
| 3            | Email      | varchar      | Email of a student       | johndoe12@gmail.com |
|              | File_Name  | varchar      | Name of a file           | Assignment          |
|              | File_Type  | nvarchar     | extension of a file      | .pdf                |
|              | Answer     | varbinary    | File uploaded by student | Assignment.pdf      |

|          |         |    |                         |                        |
|----------|---------|----|-------------------------|------------------------|
| Feedback | varchar | 50 | Feedback for assignment | Please submit on time. |
|----------|---------|----|-------------------------|------------------------|

### 3.2.4 Schedule

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

| Variable | Field Type | Field Length | Description            | Example |
|----------|------------|--------------|------------------------|---------|
| Name     |            |              |                        |         |
| 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 |

### 3.2.5 Contact

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

| Variable | Field Type | Field Length | Description         | Example   |
|----------|------------|--------------|---------------------|---|
| Name     |            |              |                     |   |
| 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. |

### 3.2.6 Assignment

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

| Variable     | Field Type | Field Length | Description               | Example        |
|--------------|------------|--------------|---------------------------|----------------|
| Name         |            |              |                           |                |
| 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 |

### 3.2.7 Available Course

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

| Variable     | Field Type | Field Length | Description                         | Example |
|--------------|------------|--------------|-------------------------------------|---------|
| Name         |            |              |                                     |         |
| 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 |

### 3.2.8 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 |

### 3.2.9 Course\_Materials

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

| Variable<br>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            |

### 3.2.10 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<br>5             | Integer    | -----        | Unique id                           | 1                                    |
| Course_Name         | varchar    | 50           | Name of the course                  | Python                               |
| Course_Heading<br>5 | 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                                |

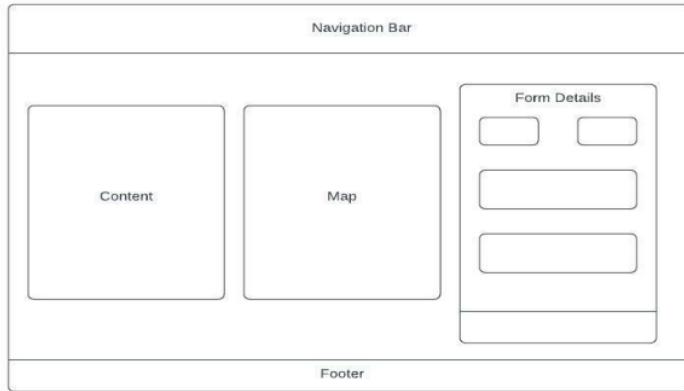
### 3.3 Wire Framing

#### 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.

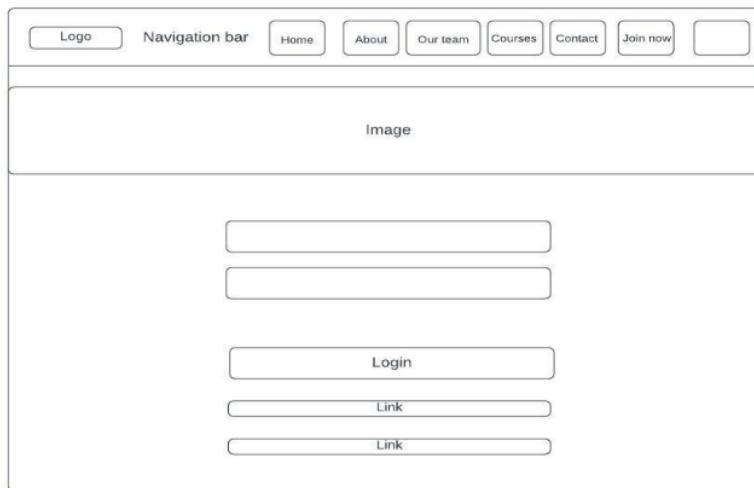


#### 3.3.2 Contact us



### 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.

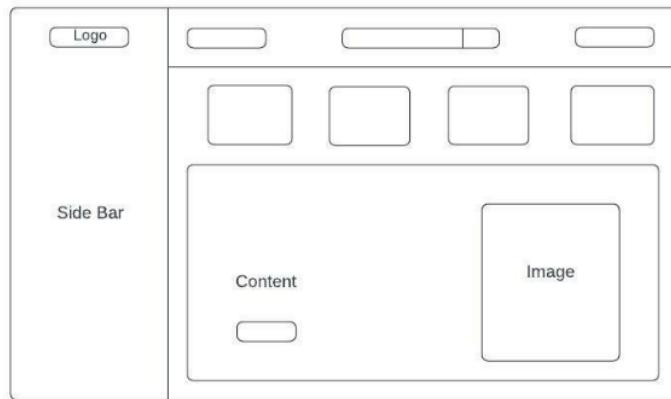


### 3.3.3 Course Enroll

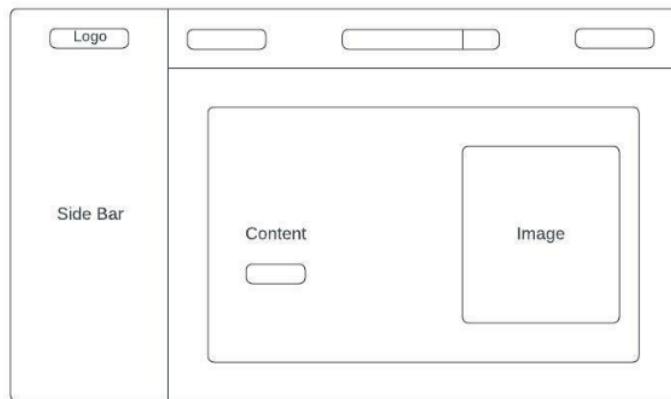


### **3.3.4 Admin Panel**

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



### **3.3.5 Lecture Panel**



### 3.4 Website Navigational structure

#### 3.4.1 Admin

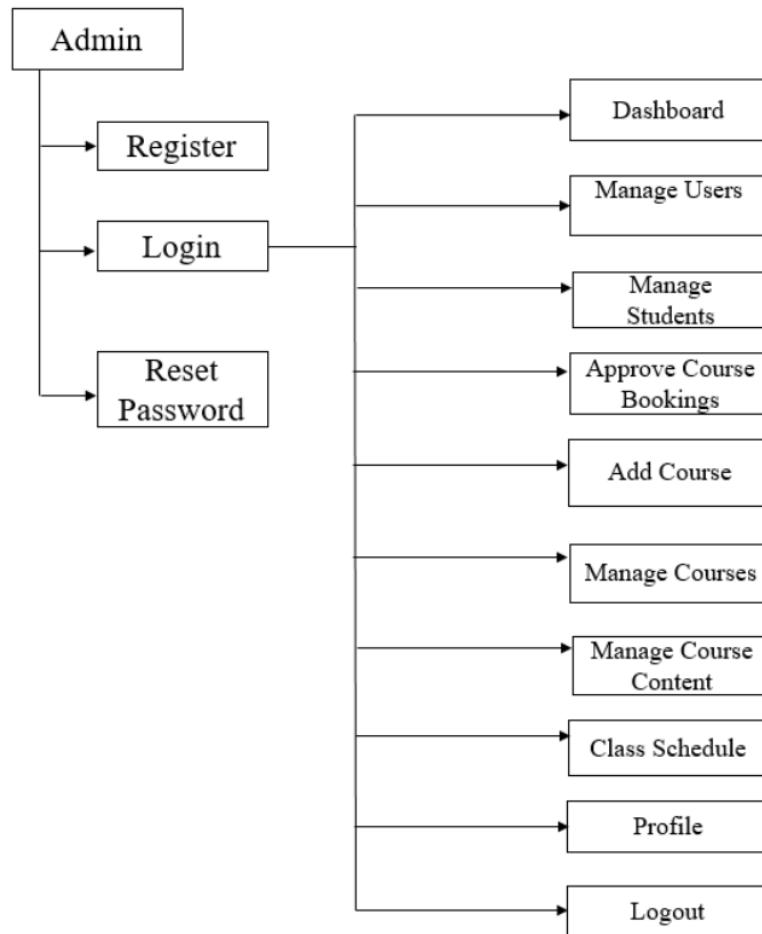


Fig: Navigational structure of Admin

#### 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:

- i. Manage users: Add, update, and delete users and lecturer accounts
- ii. Manage students: Add, update, and delete enrolled students
- iii. Approve course bookings: Approve and delete course bookings
- iv. Add course: Add new Course
- v. Manage courses: Update and delete courses

- vi. Manage course contents: Add, update, and delete the details displayed on the course page
- vii. Class schedule: Add, update and delete the class schedule
- viii. Profile: Update individual profile
- ix. Logout: Logout from the panel

### 3.4.2 Lecturers

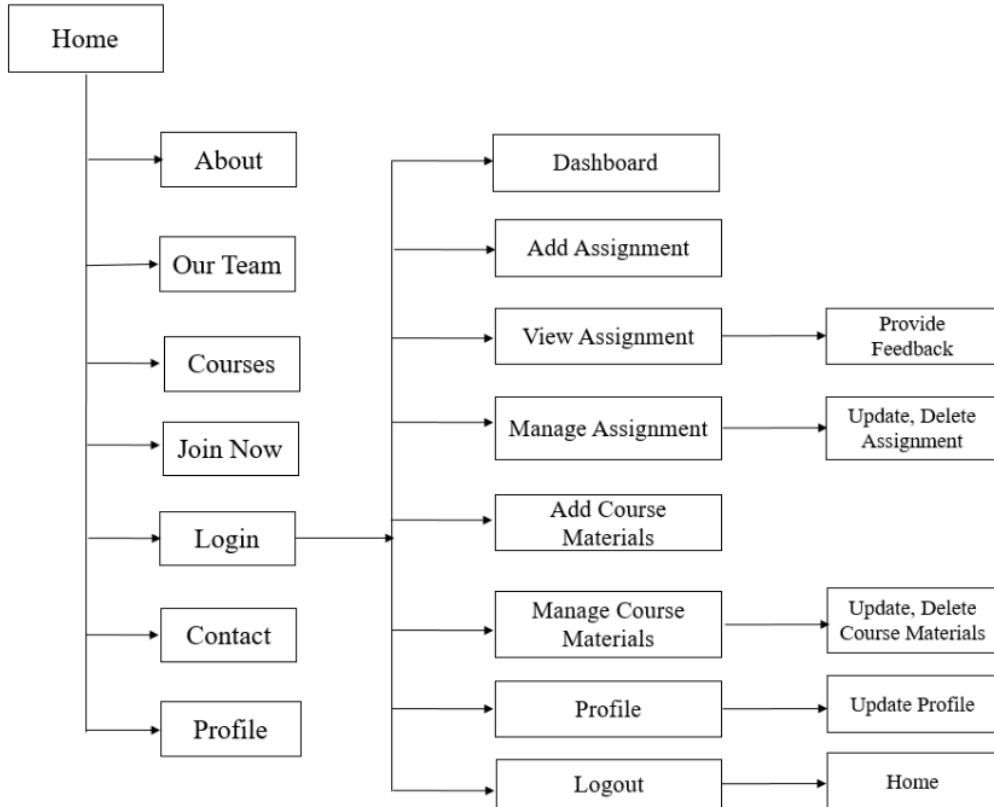


Fig: Navigational structure of Lecturers

#### 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:

- i. Add assignment: Add assignment along with the date of hand in, hand out, and question

- ii. View assignment: View assignment uploaded by the students and provide feedback
- iii. Manage assignment: Update and delete the uploaded assignment of students
- iv. Add course materials: Add course-related materials like slides and tutorials
- v. Manage course materials: Update and delete the uploaded course materials
- vi. Profile: Update individual profile
- vii. Logout: Logout from the lecturer panel

### 3.4.3 Users/ Students

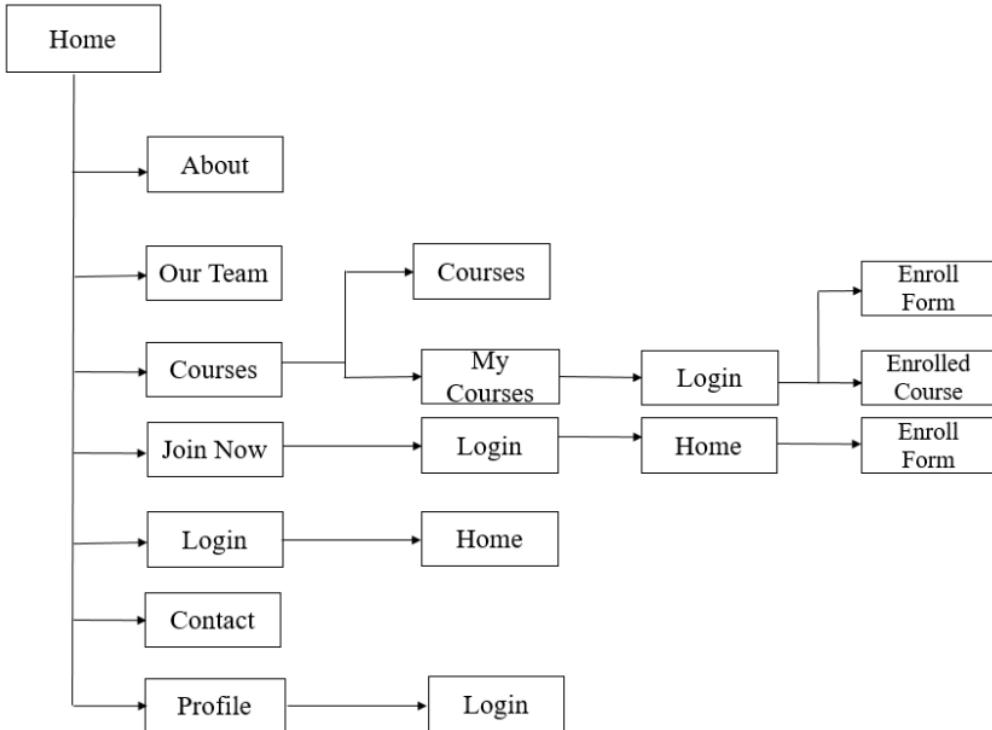


Fig: Navigational structure of Users/ students

#### 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.

## 4. Implementation (Code Snippet)

### 4.1 Insert

```
protected void btn1_Click(object sender, EventArgs e)
{
    string ins = "insert into [Contact](Name,Email,Subject,Message) " +
        "values('" + name.Text + "','" + email.Text + "','" + subject.Text + "','" + message.InnerText + "')";
    SqlCommand cmd = new SqlCommand(ins, con);
    con.Open();
    cmd.ExecuteNonQuery();
    con.Close();
    ScriptManager.RegisterClientScriptBlock(this, this.GetType(), "alertMessage", "alert('Message Sent Successfully')", true);
    name.Text=String.Empty;
    email.Text= String.Empty;
    subject.Text= String.Empty;
    message.InnerText= String.Empty;

}
```

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.

### 4.2 Display

```
public void displayCourses()
{
    string constring = ConfigurationManager.ConnectionStrings["AspNetConn"].ConnectionString;
    using (SqlConnection con = new SqlConnection(constring))
    {
        SqlCommand cmd = new SqlCommand("select * from AvailableCourse", con);
        cmd.CommandType = CommandType.Text;
        con.Open();
        DataList1.DataSource = cmd.ExecuteReader();
        DataList1.DataBind();
    }
}
```

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.

#### 4.3 Update

```
protected void gridview1_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    using (SqlConnection sqlCon = new SqlConnection(constring))
    {
        sqlCon.Open();
        string query = "UPDATE Student_Assignment SET Feedback = @Feedback WHERE ID = @ID";
        SqlCommand sqlCmd = new SqlCommand(query, sqlCon);
        sqlCmd.Parameters.AddWithValue("@Feedback", (gridview1.Rows[e.RowIndex].FindControl("txtFeedback") as TextBox).Text.Trim());
        sqlCmd.Parameters.AddWithValue("@ID", Convert.ToInt32(gridview1.DataKeys[e.RowIndex].Value.ToString()));
        sqlCmd.ExecuteNonQuery();
        gridview1.EditIndex = -1;
        displayRecords();
        lblSuccessMessage.Text = "Selected Record Updated";
    }
}
```

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 displayRecords method.

#### 4.4 Delete

```
protected void gridview1_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    try
    {
        using (SqlConnection sqlCon = new SqlConnection(constring))
        {
            sqlCon.Open();
            string query = "DELETE FROM Student_Assignment WHERE ID = @ID";
            SqlCommand sqlCmd = new SqlCommand(query, sqlCon);
            sqlCmd.Parameters.AddWithValue("@ID", Convert.ToInt32(gridview1.DataKeys[e.RowIndex].Value.ToString()));
            sqlCmd.ExecuteNonQuery();
            displayRecords();
            lblSuccessMessage.Text = "Selected Record Deleted";
            lblErrorMessage.Text = "";
        }
    }
    catch (Exception ex)
    {
        lblSuccessMessage.Text = "";
        lblErrorMessage.Text = ex.Message;
    }
}
```

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.

#### 4.5 Validation

```
<div class="box-card">
    <span class="information">Last Name</span>
    <asp:Textbox runat="server" ID="txtLName" type="text" placeholder="Enter your Last Name" required/>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" ErrorMessage="Last Name is Required"
        ControlToValidate="txtLName" ForeColor="#3399FF"></asp:RequiredFieldValidator>
</div>
```

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.

#### 4.6 Session variable

```
public void showDetails()
{
    con.Open();
    SqlCommand cmd = new SqlCommand("select count(*) from Users where Email = '" + txtEmail.Text + "' and Password = '" + txtpassword.Text + "'", con);
    int count = Convert.ToInt32(cmd.ExecuteScalar().ToString());
    if (count > 0)
    {
        SqlCommand cmdType1 = new SqlCommand("select First_Name from Users where Email = '" + txtEmail.Text + "'", con);
        SqlCommand cmdType2 = new SqlCommand("select Last_Name from Users where Email = '" + txtEmail.Text + "'", con);
        SqlCommand cmdType3 = new SqlCommand("select Phone from Users where Email = '" + txtEmail.Text + "'", con);
        SqlCommand cmdType4 = new SqlCommand("select Address from Users where Email = '" + txtEmail.Text + "'", con);
        SqlCommand cmdType5 = new SqlCommand("select Email from Users where Email = '" + txtEmail.Text + "'", con);
        SqlCommand cmdType6 = new SqlCommand("select Password from Users where Email = '" + txtEmail.Text + "'", con);
        string type1 = cmdType1.ExecuteScalar().ToString().Replace(" ", "");
        string type2 = cmdType2.ExecuteScalar().ToString().Replace(" ", "");
        string type3 = cmdType3.ExecuteScalar().ToString().Replace(" ", "");
        string type4 = cmdType4.ExecuteScalar().ToString().Replace(" ", "");
        string type5 = cmdType5.ExecuteScalar().ToString().Replace(" ", "");
        string type6 = cmdType6.ExecuteScalar().ToString().Replace(" ", "");
        Session["FName"] = type1;
        Session["LName"] = type2;
        Session["Phone"] = type3;
        Session["Address"] = type4;
        Session["Email"] = type5;
        Session["Password"] = type6;
        if (type1 == "txtEmail.Text" && type2 == "txtEmail.Text" && type3 == "txtEmail.Text" && type4 == "txtEmail.Text" && type5 == "txtEmail.Text"
            && type6 == "txtEmail.Text")
        {
            Response.Redirect("Home.aspx");
        }
    }
}
```

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.

## 5. User Guidance

### 5.1 User

#### 5.1.1 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.

### 5.1.2 Course Page

The screenshot displays a user interface for a course platform. At the top, there is a section titled "Courses Categories" with a teal header bar. Below it, there are four course categories represented by images and labels:

- Java** Available: Shows two men working on a laptop.
- Web Development** Available: Shows a group of people working on laptops.
- Video Editing** Available Soon: Shows people working at desks.
- Online Marketing** Available Soon: Shows a person writing in a notebook next to a book titled "Nomatrix Study Guide".

Below this section is another teal header bar labeled "Popular Courses". A large central box features the Java logo (a stylized coffee cup with steam) and the word "Java™". To the right of the logo, course details are listed:  
\$120  
★★★★★ (123)  
Java  
Ram Bahadur | 120 Hrs | Available

Fig: Course Page

On this page, the available courses are displayed. When an admin adds courses it will be displayed on this page. Users can enroll by checking the status of the course i.e., available or unavailable.

### 5.1.3 Login Page

— LOGIN —

Fill up the form

Email  
Enter your Email

Password  
Enter your password

Login

Don't Have an account? Register Here!

Forgot Password? Click Here!

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.

### 5.1.4 Course Enroll Page

— ENROLL COURSE —

Fill up the Details

Personal/Course Details

Full Name  
Enter Your Name

Email  
shawn@gmail.com

Phone  
Enter Phone Number

Email Date  
04/02/2022 11:13:47 PM

Course Code  
Enter Course Name

Course Title  
Enter Course Title

Exp Year  
Enter Exp Year

CVV  
Enter CVV

Payment

Accepted Cards

Visa Mastercard American Express Discover

Cardholder Name  
Enter Card Holder Name

Card number  
1111-2222-3333-4444

Exp Month  
Enter Exp Month

Enter Exp Month

CVV  
Enter CVV

I'm agree to the evaluating Terms & Privacy

Pay Now

Fig: Course enroll page

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

### 5.1.5 Enrolled course Page

The screenshot shows a website for 'Krypton Edu'. At the top, there's a navigation bar with links for HOME, ABOUT, OUR TEAM, COURSES (with a dropdown arrow), and CONTACT. To the right of the navigation is a 'Join Now' button with a right-pointing arrow and a user icon. Below the navigation, the word 'JAVA' is written in blue capital letters, followed by a horizontal line and the title 'Object Oriented Programming' in bold black font. A large Java logo (a steaming coffee cup with the word 'Java' below it) is positioned to the left of the title. To the right of the title, there's a brief description of what programming languages are, mentioning thousands of them and their use in developing software. Below this, there's a section with two columns of information: 'Lecturer Name: Mr. Ram Bahadur' with an arrow icon, 'Email: ram@gmail.com' with an arrow icon, 'Lecturer Experience: 5 Years' with an arrow icon, 'Class Duration/Day: 1.5 Hrs' with an arrow icon, and 'Course Hours: 120 Hrs' with an arrow icon. At the bottom of this section are two buttons: a blue 'Read More' button and a small blue square with an upward arrow icon.

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.

### 5.1.6 User profile page

The screenshot shows the 'User Profile' page. At the top, there's a banner with a woman's face and the text 'User Profile'. Below the banner, the navigation bar includes links for HOME, ABOUT, OUR TEAM, COURSES (with a dropdown arrow), and CONTACT, along with a 'Join Now' button. The main content area has a header 'USER PROFILE' with a sub-header 'Details'. It contains a form with fields for First Name ('Shivam'), Last Name ('Kumar'), Phone Number ('9876543210'), Address ('Pahar Ganj'), Email ('shivam@gmail.com'), and Password ('shivam12'). At the bottom of the form is a blue 'Update' button.

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.

## 5.2 Lecture Panel

### 5.2.1 Dashboard

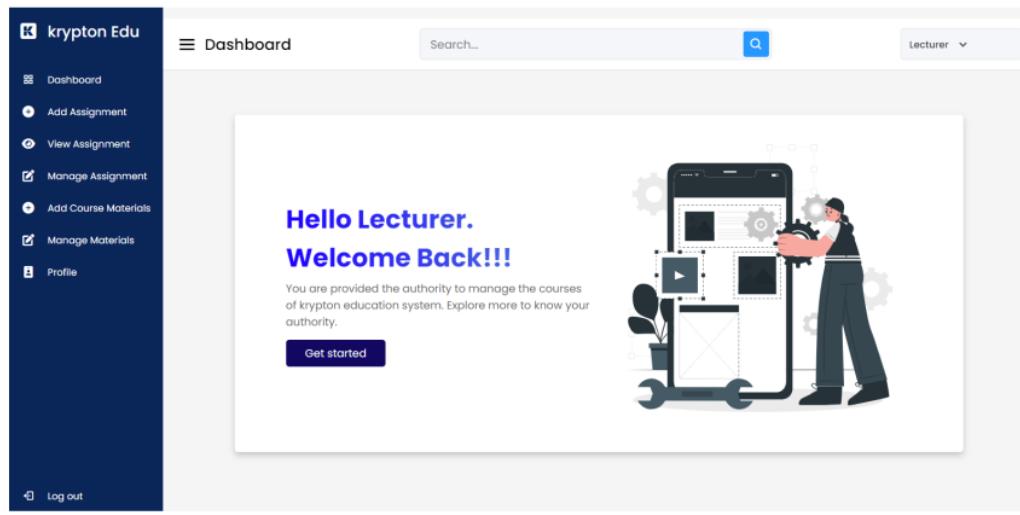


Fig: Lecturer dashboard

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

### 5.2.2 Add assignment Page

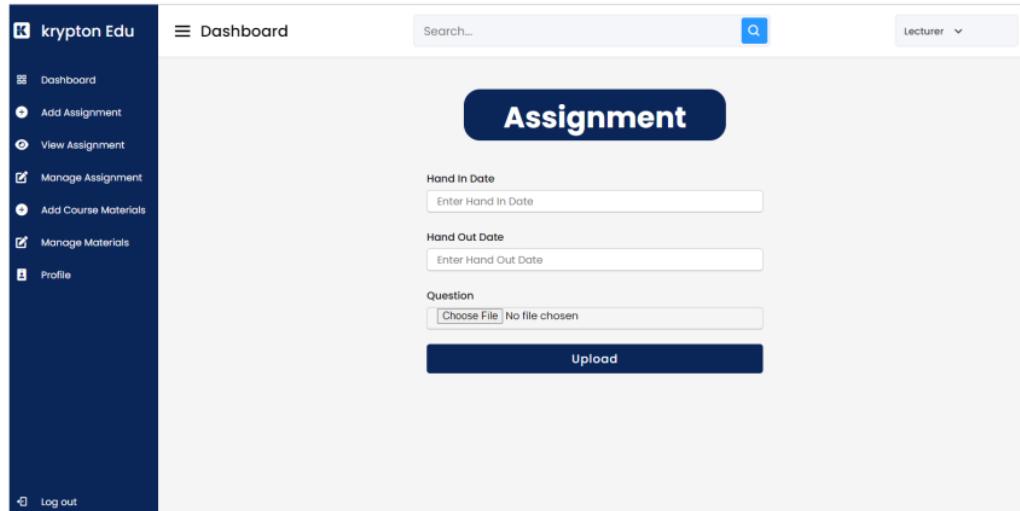


Fig: Add assignment page

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

The screenshot shows the 'View Assignment' page. On the left is a dark sidebar with the 'krypton Edu' logo and a navigation menu containing 'Dashboard', 'Add Assignment', 'View Assignment' (which is highlighted), 'Manage Assignment', 'Add Course Materials', 'Manage Materials', and 'Profile'. At the bottom of the sidebar is a 'Log out' button. The main area has a header 'Dashboard' with a search bar and a dropdown set to 'Lecturer'. Below this is a large blue header 'View Assignment'. A table follows, with columns: ID, Student Name, Email, Answer, Feedback, and Actions. One row is shown with ID 1, Student Name 'Shivam', Email 'shivam@gmail.com', Answer as a link 'Download', Feedback 'Nice', and Actions showing edit and delete icons.

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

The screenshot shows the 'Assignment' page. The sidebar and header are identical to the 'View Assignment' page. The main area has a large blue header 'Assignment'. A table follows, with columns: ID, Hand In Date, Hand Out Date, File Name, and Actions. One row is shown with ID 1005, Hand In Date '28 Jan', Hand Out Date '1 Feb', File Name 'PFDA\_Test\_2.docx.pdf', and Actions showing edit and delete icons.

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.

### 5.2.3 Add course Materials Page

The screenshot shows the 'Course Materials' section of the application. On the left is a sidebar with navigation links: Dashboard, Add Assignment, View Assignment, Manage Assignment, Add Course Materials (which is selected and highlighted in blue), Manage Materials, and Profile. At the bottom of the sidebar is a 'Log out' link. The main content area has a title 'Course Materials'. It includes a 'Chapter' input field with placeholder text 'Enter Chapter Name', two file upload fields for 'Slides' and 'Tutorials', both currently showing 'No file chosen', and a large blue 'Upload' button.

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.

### 5.2.4 Manage course materials Page

The screenshot shows the 'Manage Materials' section of the application. The sidebar on the left is identical to the previous screenshot. The main content area has a title 'Manage Materials'. Below it is a table with the following data:

| ID | Chapter   | Slides                         | Tutorials         | Actions |
|----|-----------|--------------------------------|-------------------|---------|
| 1  | Chapter-1 | Mo8ExV52JmVbbIO-1617345529.pdf | OODJ(TEST-1).docx |         |
| 2  | Chapter-2 | NPI00047_OODJ (test-2).docx    |                   |         |

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.

### 5.2.5 Profile Page

The screenshot shows the 'Profile' section of the Krypton Edu dashboard. On the left, there's a sidebar with navigation links: Dashboard, Add Assignment, View Assignment, Manage Assignment, Add Course Materials, Manage Materials, and Profile. Below these is a 'Log Out' button. The main area has a search bar and a dropdown for 'Lecturer'. A table titled 'Profile' displays one row of data:

| User.ID | First Name | Last Name | Address   | Phone    | Email               | Password | UserType | Actions |
|---------|------------|-----------|-----------|----------|---------------------|----------|----------|---------|
| 6       | Ajay       | kumar     | Kathmandu | 98927362 | ajaykumar@gmail.com | ajayl2   | Lecturer |         |

Fig: Profile page

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

### 5.3 Admin Panel

#### 5.3.1 Admin Dashboard

The screenshot shows the 'Dashboard' section for an admin. The sidebar includes links for Dashboard, Users, Students, Course Bookings, Add Courses, Messages, Manage Courses, Manage Pages, Class Schedule, and Profile, along with a Log Out button. The main dashboard features four summary cards: Total Users (6), Total Students (2), New Bookings (0), and Total Messages (0). Below these is a large central area with a welcome message: 'Hello Admin. Welcome Back!!!' followed by a subtext: 'You Are Provided The Authority To Manage The Entire Krypton Education System. Explore More To Know Your Authority.' At the bottom of this area is a 'Get Started' button. To the right of the text is an illustration of a person working on a smartphone screen surrounded by gears and tools.

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.

### 5.3.2 Account details page

| User.ID | First Name | Last Name | Address    | Phone      | Email               | Password | UserType | Actions |
|---------|------------|-----------|------------|------------|---------------------|----------|----------|---------|
| 1       | Bibek      | KC        | Pokhara    | 9878263126 | bibek@gmail.com     | bibek12  | User     |         |
| 5       | Shivam     | Ranobhat  | Pokhara    | 9805867639 | shivam@gmail.com    | shivam12 | User     |         |
| 3       | Ram        | Bahadur   | Pokhara-27 | 9864545354 | ram@gmail.com       | ram12    | Lecturer |         |
| 6       | Ajay       | kumar     | Kathmandu  | 98927362   | ajaykumar@gmail.com | ajay12   | Lecturer |         |

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.

### 5.3.3 Student details page

| Student.ID | Name         | Email           | Phone       | Enroll Date          | Course Title | UserType | Actions |
|------------|--------------|-----------------|-------------|----------------------|--------------|----------|---------|
| 1          | Bibek khatri | bibek@gmail.com | 98253626352 | 1/24/2022 8:18:53 AM | Java         | Student  |         |

Selected Record Updated

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.

### 5.3.4 Booking details page

The screenshot shows the 'Booking Details' section of the Krypton Edu admin dashboard. On the left is a sidebar with navigation links: Dashboard, Users, Students, Course Bookings, Add Courses, Messages, Manage Courses, Manage Pages, Class Schedule, and Profile. At the bottom of the sidebar is a 'Log out' button. The main area has a header 'Dashboard' with a search bar and an 'Admin' dropdown. Below the header are four summary boxes: 'Total Users' (6, up from yesterday), 'Total Students' (1, up from yesterday), 'New Bookings' (1, up from yesterday), and 'Total Messages' (0, up from yesterday). The 'Booking Details' section has a dark blue header. A table below lists a single booking entry:

| ID   | Name            | Email            | Phone      | Enroll Date          | Course Title | CardHolder Name | Card Number  | Exp Month | Exp Year | CVV       | UserType | Actions |
|------|-----------------|------------------|------------|----------------------|--------------|-----------------|--------------|-----------|----------|-----------|----------|---------|
| 2009 | Shivam Ranabhat | shivam@gmail.com | 9805867639 | 3/5/2022 11:25:01 AM | Python       | Shivam Ranabhat | 871210312398 | June      | 2023     | 635355353 | Student  |         |

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.

### 5.3.5 Add course page

The screenshot shows the 'New Course' section of the Krypton Edu admin dashboard. The sidebar and header are identical to the previous screenshot. The main area has four summary boxes: 'Total Users' (6, up from yesterday), 'Total Students' (1, up from yesterday), 'New Bookings' (1, up from yesterday), and 'Total Messages' (0, up from yesterday). Below these is a large dark blue button labeled 'New Course'. To its right is a form with five input fields: 'Course Name' (placeholder 'Enter Course Name'), 'Course Code' (placeholder 'Enter Course Name'), 'Course Price' (placeholder 'Enter Course Price'), 'Lecturer Name' (placeholder 'Enter Lecturer Name'), and 'Lecturer Hour' (placeholder 'Enter Lecturer Name').

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.

### 5.3.6 Manage messages page

The screenshot shows the Krypton Edu Admin Dashboard. On the left is a sidebar with icons for Dashboard, Users, Students, Course Bookings, Add Courses, Messages, Manage Courses, Manage Pages, Class Schedule, Profile, and Log out. The main area has a header with 'Dashboard', a search bar, and an 'Admin' dropdown. Below the header are four cards: 'Total Users' (6, up from yesterday), 'Total Students' (1, up from yesterday), 'New Bookings' (1, up from yesterday), and 'Total Messages' (0, up from yesterday). A large blue button labeled 'Messages' is centered. Below it is a table with columns: Name, Email, Subject, Message, and Actions. A message 'No Data Found ...' is displayed. At the bottom right of the table is a small 'No Data Found' icon.

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.

### 5.3.7 Manage courses page

The screenshot shows the Krypton Edu Admin Dashboard. The sidebar and header are identical to the previous screenshot. Below the header are four cards: 'Total Users' (6, up from yesterday), 'Total Students' (1, up from yesterday), 'New Bookings' (1, up from yesterday), and 'Total Messages' (0, up from yesterday). A large blue button labeled 'Course Details' is centered. Below it is a table with columns: Course ID, Course Code, Course Name, Course Price, Lecturer Name, Lecture Hours, Status, File, and Actions. Two rows of course data are shown: Row 1 (Course ID 2, Java, \$120, Ram Bahodur, 120 Hrs, Available, JavaLogo.png) and Row 2 (Course ID 7, Py-02, Python, \$120, Ajay Kumar, 120 Hrs, Available, Python.png). Each row has edit and delete icons in the Actions column.

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.

### 5.3.8 Manage course page details

The screenshot shows the 'Page Details' section of the Krypton Edu admin dashboard. On the left is a sidebar with icons for Dashboard, Users, Students, Course Bookings, Add Courses, Messages, Manage Courses, Manage Pages, Class Schedule, and Profile. The main area has a header 'Dashboard' with a search bar and an 'Admin' dropdown. Below are four cards: 'Total Users 6 (Up from yesterday)', 'Total Students 1 (Up from yesterday)', 'New Bookings 1 (Up from yesterday)', and 'Total Messages 0 (Up from yesterday)'. A large table titled 'Page Details' lists course information:

| Course Name | Course Heading              | Lecturer Name   | Lecturer Experience | Lecturer Email      | Class Duration | Lecture Hours | Actions |
|-------------|-----------------------------|-----------------|---------------------|---------------------|----------------|---------------|---------|
| Java        | Object Oriented Programming | Mr. Ram Bahadur | 5 Years             | ram@gmail.com       | 1.5 Hrs        | 120 Hrs       |         |
| Python      | High Level Programming      | Mr. Ajay Kumar  | 10 Years            | ajaykumar@gmail.com | 1.5 Hrs        | 125 Hrs       |         |
|             |                             |                 |                     |                     |                |               |         |

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.

### 5.3.9 Class schedule page

The screenshot shows the 'Class Schedule' section of the Krypton Edu admin dashboard. The sidebar and top navigation are identical to the previous section. The main area has a header 'Dashboard' with a search bar and an 'Admin' dropdown. Below are four cards: 'Total Users 6 (Up from yesterday)', 'Total Students 1 (Up from yesterday)', 'New Bookings 1 (Up from yesterday)', and 'Total Messages 0 (Up from yesterday)'. A large table titled 'Class Schedule' lists class times:

| ID | Days      | From     | To       | Actions |
|----|-----------|----------|----------|---------|
| 1  | Sunday    | 9:00 am  | 11:00 am |         |
| 3  | Monday    | 9:00 am  | 11:00 am |         |
| 4  | Tuesday   | 9:30 am  | 11:00 am |         |
| 5  | Wednesday | 9:30 am  | 11:30 am |         |
| 6  | Thursday  | 10:00 am | 12:00 pm |         |
| 7  | Friday    | 10:00 am | 12:00 pm |         |
| 8  | Saturday  | ----     | ----     |         |
|    |           |          |          |         |

Fig: Class schedule page

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

### 5.3.10 Profile page

The screenshot shows the 'Dashboard' section of the Krypton Edu application. On the left is a sidebar with navigation links: Dashboard, Users, Students, Course Bookings, Add Courses, Messages, Manage Courses, Manage Pages, Class Schedule, and Profile. The 'Profile' link is highlighted. At the top right is a 'Search...' input field and an 'Admin' dropdown. Below the sidebar are four summary cards: 'Total Users 7 Up from yesterday', 'Total Students 1 Up from yesterday', 'New Bookings 1 Up from yesterday', and 'Total Messages 0 Up from yesterday'. The main content area is titled 'Account Details' and contains a table with one row of data. The table columns are: User.ID, First Name, Last Name, Address, Phone, Email, Password, UserType, and Actions. The data row corresponds to the admin user: User.ID 2007, First Name Kapil, Last Name Pokhrel, Address Pokhara-30, Phone 9874343476, Email kapil@gmail.com, Password kapil12, UserType Admin, and Actions (an edit icon).

Fig: Profile page

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

## 6. 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.

### 6.2 Future Enhancement

1. A discussion board for students to communicate with one another.
2. A live class video upload function for teachers to help students learn more readily.

3. The system must support many payment methods, such as PayPal debit cards, credit cards, and so on.
4. The quiz program and the prize for the winner Which method will be most effective?
5. The current user experience is fine, but to increase traffic and make it easier for students, we must upgrade it.

### **Workload Matrix**

| Topics  | Bikash Baral<br>(NPI000021) | Kapil Pokhrel<br>(NPI000030) | Saroj Kandel<br>(NPI0000) | Shivam Ranabhat<br>(NPI000047) |
|---|-----------------------------|------------------------------|---------------------------|--------------------------------|
| Introduction/<br>Project Plan<br><span style="color: red;">1</span> |                             | 100%                         |                           |                                |
| Requirement<br>Specification  | 25%                         | 25%                          | 25%                       | 25%                            |
| Design and<br>Modeling  | 25%                         | 25%                          | 25%                       | 25%                            |
| Implementation  |                             |                              |                           | 100%                           |
| User Guidance   | 100%                        |                              |                           |                                |
| Conclusions   |                             |                              | 100%                      |                                |



PRIMARY SOURCES

- |   |  |      |
|---|--|------|
| 1 | Submitted to Asia Pacific University College of Technology and Innovation (UCTI) | 5%   |
| 2 | Submitted to TMC Education Group   | 1 %  |
| 3 | Submitted to University of Greenwich   | 1 %  |
| 4 | Submitted to Blackpool and The Fylde College, Lancashire                         | 1 %  |
| 5 | Submitted to Kensington College of Business - Brunei                             | 1 %  |
| 6 | Submitted to Middle East College of Information Technology                       | <1 % |
- Student Paper

Exclude quotes

Off

Exclude bibliography

On

Exclude matches

Off

