

## **Abstract**

The main aim of this project is to provide a platform through website which determine the “Web-based Learning System”. The project starts with the identification of requirement specification for this website. After the identification of requirement specification, entities, attributes and relationship between the entities, ERD (Entity relationship diagram) is formed. Likewise, required wireframes web pages and navigational structure for this website are creating which leads to develop consistent pages and decentralise system. This project is looking forward to serve for both non-registered as well as registered users who are searching for online learning activities and digital learning resources.

## **Acknowledgement**

We would like to exhibit our genuine gratefulness to everyone who assisted us to complete this web development project. Firstly, we'd love to display extraordinary thankfulness to our subject mentor, Mr. Jupiter Tamrakar. His counsel and encouragement lead us to complete our assignment task within proper time along with appropriate guidelines. Additionally, we'd love to applaud our mates who helped us in collecting required data. Without their cooperation, this paper would not be as good.

<b>Abstract</b>	<b>4</b>
<b>Acknowledgement</b>	<b>5</b>
<b>1. Introduction</b>	<b>8</b>
1.1. Objective	9
1.2. Scope	9
1.3. Project Schedule	10
<b>2. Requirement Specification</b>	<b>11</b>
2.1. Audience Classification	11
2.2. Audience Characterisation	11
<b>3. Design and Modelling</b>	<b>12</b>
3.1. ERD (Entity Relationship Diagram)	12
3.2. Data Dictionary	13
3.3. Wireframe	17
3.4. Website Navigational Structure	24
<b>4. Implementation</b>	<b>27</b>
4.1. Declaring Connection string	27
4.2. User Register Code (Insert)	28
4.3. Login code (Read)	29
4.4. Delete code	30
4.5. Search course	31
4.6. Insert Course	32
4.7. Update code	33
4.8. Show courses in list view	34
<b>5. User Guidance</b>	<b>35</b>
5.1. Home page	35
5.2. Register page	36
5.3. User Profile	37
5.4. Course	38

5.5. View Course	40
5.6. Add Course	41
6. Conclusions (Summary and Future Enhancements)	42
7. Reference	43
8. Appendix	45

## **1. Introduction**

Basically, “Mind Education” web application is an E-learning website which provide knowledge about different courses through online media. Moreover, this website is generally creating as for resources of digital learning. According to research, our team determine the issues faced by various student regarding their studies during the pandemic. Furthermore, we decided to generate a responsive web application which assists user to gain knowledge.

Regarding this website, we supposed that the control of website is divided into two parts: User part and admin part. According to user or student phase, they can create and visit their profile with their detail. In addition, they can see the list of course and in order to select their specific course, they have to register into the enroll form. In other hand, admin should have registered their account with their personal information and have permission to add course, view student list. More importantly, they have approved user details in which user have permission to get specific course that they needed. In addition, admin has an ability to delete and add user. Furthermore, they have also permission as have as user to update their profile.

## **1.1. Objective**

- To create platform where user gets online learning resources.
- Create a platform where user can enroll in interested courses.
- Create a platform where educational resources are shared.
- To improve the learning and teaching experience.
- To encounter student's learning styles and requirements.
- To engage learners in the learning process.

## **1.2. Scope**

Throughout this application, people have opportunities to continue their classes and career. The project of this application will predict to complete at March 2022. This application has facilitated through including add course, searching required course, profile update, view assignment and details about their specific requirement.

### 1.3. Project Schedule

ID	Task Name	Start	Finish	Duration	Dec 2021			Jan 2022				Feb 2022		
					12/12	12/19	12/26	1/2	1/9	1/16	1/23	1/30	2/6	2/13
1	1.0 Project Planning	12/16/2021	12/22/2021	7d										
2	2.0 System Design	12/23/2021	1/11/2022	20d										
3	2.1 System Layout	12/23/2021	12/26/2021	4d										
4	2.2 Detail Design	12/27/2021	12/30/2021	4d										
5	2.3 Design Review	12/31/2021	1/11/2022	12d										
6	3.0 Implementation	1/12/2022	1/31/2022	20d										
7	3.1 Coding	1/12/2022	1/26/2022	15d										
8	3.2 Testing	1/27/2022	1/31/2022	5d										
9	4.0 Documentation	2/1/2022	2/20/2022	20d										
10	4.1 Introduction	2/1/2022	2/4/2022	4d										
11	4.2 Requirement Specification	2/5/2022	2/8/2022	4d										
12	4.3 Design & Modelling	2/9/2022	2/14/2022	6d										
13	4.4 Implementation	2/15/2022	2/20/2022	6d										

## **2. Requirement Specification**

### **2.1. Audience Classification**

Guest - Non-registered audience

Member- Register User

Admin (System Controller)

### **2.2. Audience Characterisation**

#### **Member**

- Permit to Register with their personal information for the user system
- Permit to Sign in and Sign Out for the system.
- Permit to Edit their personal information.
- Permit to Enroll for their required courses.
- Permit to Observe list of courses
- Permit to Observe their personal details.

#### **Administration**

- Permit to Register with their personal information for the admin system
- Permit to Observe the total number of student.
- Permit to Add course and Assignment.
- Permit to Approve the enroll registration of user.

#### **Guest**

- Permit to Visit user system.
- Permit to View available list of courses.
- Permit to register



### 3. Design and Modelling

#### 3.1. ERD (Entity Relationship Diagram)

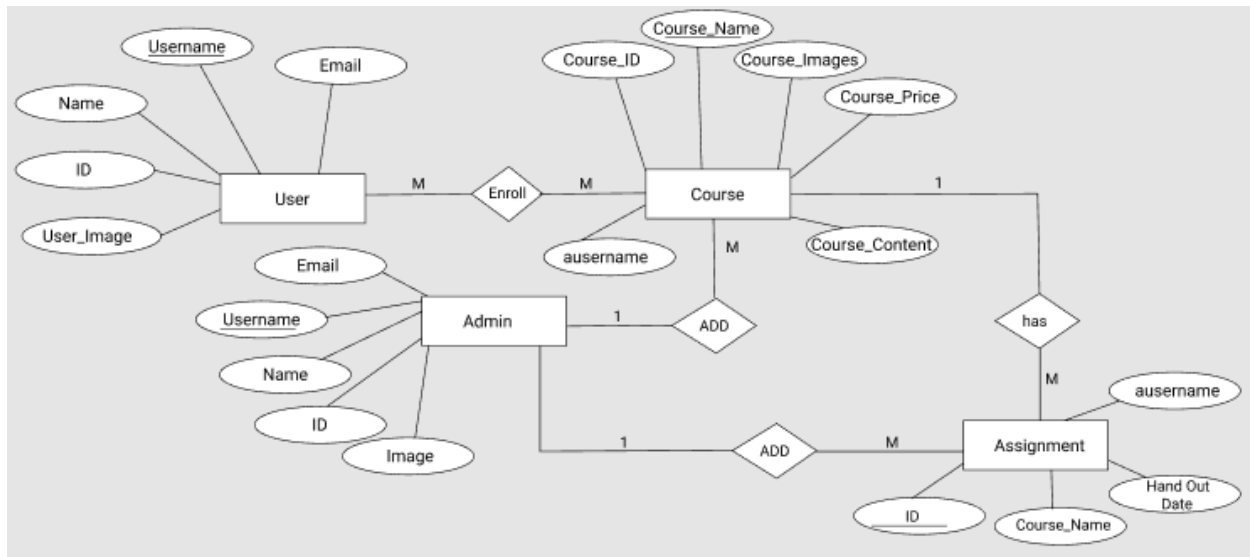


Fig 1: Entity Relationship Diagram

Entity Relationship Diagram(ERD) is a graphical representation that shows relationship between entities. This helps in building relational database. Given diagram is the ERD of 'Mind-education' web application. Box indicated the entity and oval shows the characteristics of those entities. Diamond shape illustrate the relationship between entities. This table helps in building relational database and their connection with each other. Not only this but also it helps to reduce the redundancy of data in database. Although there are four entity available in ERD the table that we need to create in database may vary. If there exist many to many relationship between two entities then third table need to be formed which stores primary key of both the table as foreign key along with composite primary key. In case of our application there is many to many relation between student and course that is one student can enroll in many course as well as many student can get enrolled in single course. So, third table called enroll table will be formed which will stores the primary keys of user table and course table as a foreign key. If their is one to many or many to one relation between two entities then the primary key of the entity of one relation will get stored as foreign key in the entity with many relation. In case of our system there is one to many relationship between admin and course, course and assignment and admin and assignment

so the column with username will be added on course and assignment table along with course name column in assignment table as foreign key. So the Total table that will actually formed are user table, admin table, course table, enroll table and assignment table. If needed table can be resembled or extra tables can be formed according to needs. So, incase of our application database user and admin table will be resembled to form single table called 'Register' table by adding type as column where as rest of the table will remain same.

### 3.2. Data Dictionary

Mentioned tables data dictionary are mapping in ERD and also implementing in Web application.

#### Register Table

Attributes Name	Description	Data Type	PK or FK
ID	Auto increment number of ID	Int	
Name	The name of user or admin	Varchar(50)	
Username	Username for login or admin	Varchar(50)	PK
Password	Password for login or admin	Varchar(50)	
Email	Email of user or admin	Varchar(50)	
User_Image	Picture of user or admin	Varchar(50)	
Type	Type as user or admin	Varchar(50)	

### Course table

Attributes Name	Description	Data Type	PK or FK
Course_ID	Auto Increment number of Course list	Int	
Course_Name	Name of Course	Varchar(50)	PK
Course_Content	Detail of Course	Varchar(MAX)	
Course_Price	Price of Course	Varchar(50)	
Class_Duration	Total time required for course complete	Varchar(50)	
Course_Image	Image of course	Varchar(50)	
Added_by (Admin username)	Username of admin who add course	Varchar(50)	FK

## Enroll

Attributes Name	Description	Data Type	PK or FK
ID	Auto Increment number of Enroll list	Int	
Name	Name of user	Varchar(50)	
Email	Email of user	Varchar(50)	
Username	Username for matching with register username	Varchar(50)	
Course_Name	Name of course	Varchar(50)	
Cardholder_Name	Name of Cardholder	Varchar(50)	
Card_Number	Number of payment card	Varchar(50)	
ExpMonth	Expiry month of Card	Varchar(50)	
ExpYear_Duration	Expiry year of Card	Varchar(50)	
CVV	CVV number of Card	Varchar(50)	

## Enroll Student

Attributes Name	Description	Data Type	PK or FK
ID	Auto increment number of enroll student	Int	PK
Name	Name of enroll student	Varchar(50)	
Email	Email of enroll student	Varchar(50)	
Username	Username of enroll student	Varchar(50)	FK
Course_Name	Name of course which student registered in enroll form	Varchar(50)	FK

## Assignment

Attributes Name	Description	Data Type	PK or FK
ID	Auto increment number of assignment list	Int	PK
Course_Name	Name of course to add assignment	Varchar(50)	FK
Hand_Out	Submit date of assignment	Varchar(50)	
Filename	Name of assignment	Varchar(50)	
Type	Type of assignment	Varchar(200)	
Assignment	Detail of assignment	Varbinary(MAX)	

### 3.3. Wireframe

First of all, wireframe is a software used to create outline layout design of website, landing pages or app. People are using the tools of wireframe to create mockup design of their appropriate structure. Without having a knowledge of coding, a simple GUI design can have created by using wireframes tools. This indicates to assist while creating their prototype execution (Martin, 2021b).

#### Home page

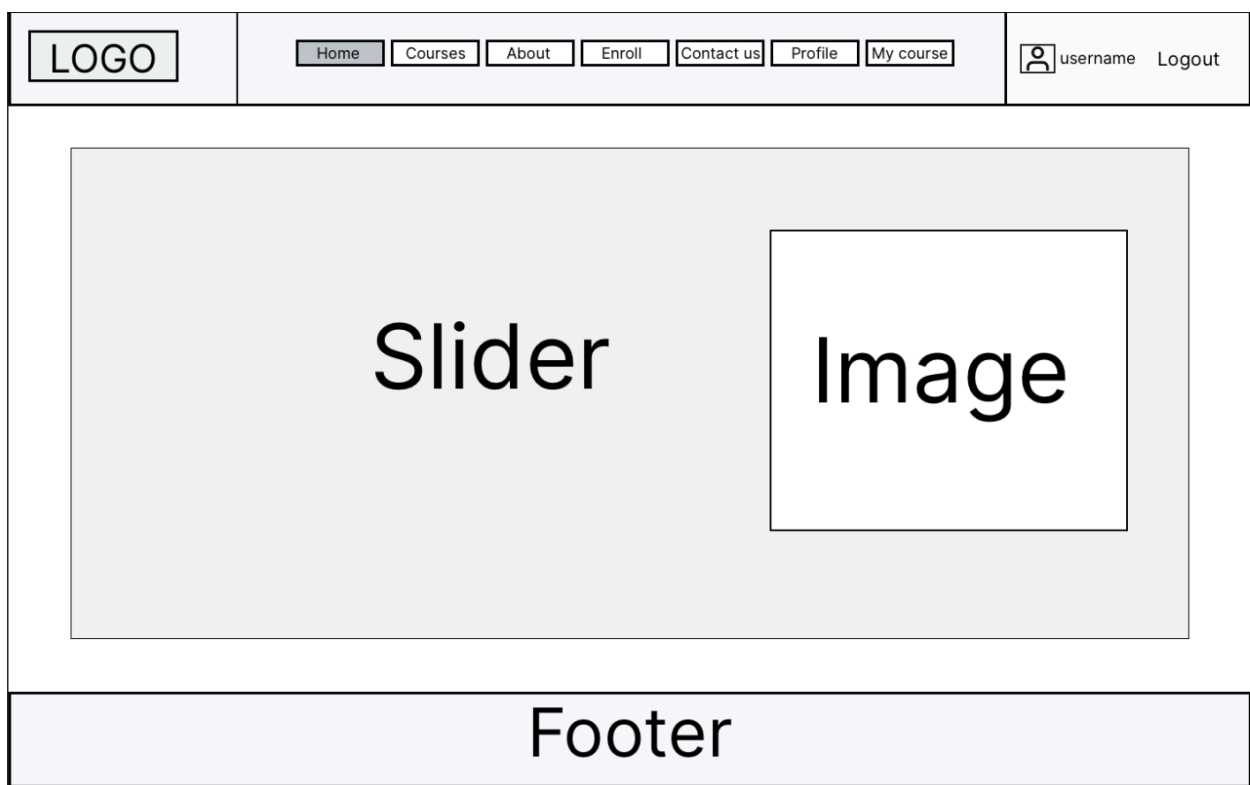


Fig 2 : Home Page Wireframe

This page is the main page of our system. The logo of the website and navigation bar will be situated at top of the page. In home page, there will be a navigation which have different link to related web pages. Courses, About, Enroll and Contact pages will be the linked page from the home page. For new user, login and register function will be provided. Below the header, there is content about the information of e-learning system along with some images in it.

## Courses page

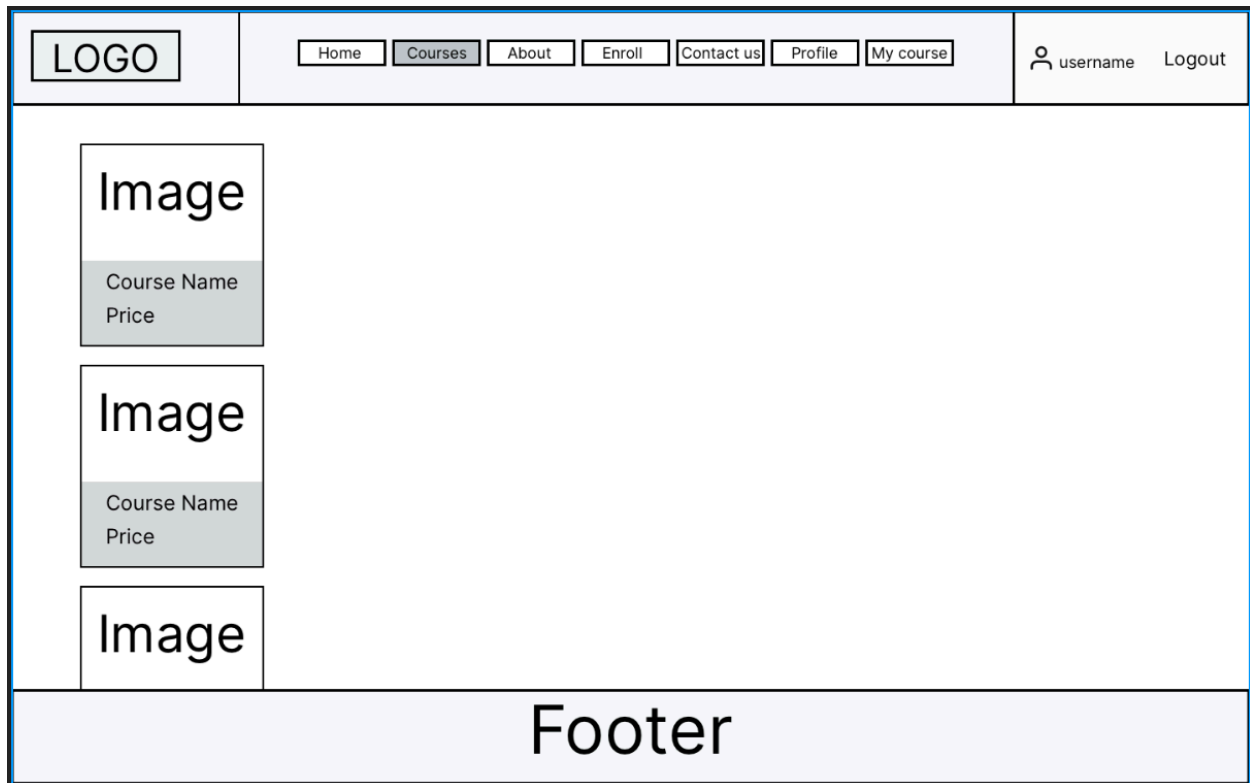


Fig 3 : Course Page Wireframe

As our system is online learning website, course page is the main page. User will enroll the different courses through it. The navigation bar, header, logo and footer is same for almost all pages. The layout for this page is heading lies at top and image of the course along with its name lies at middle left of the page and at bottom there is an overview of course.

## User Profile page

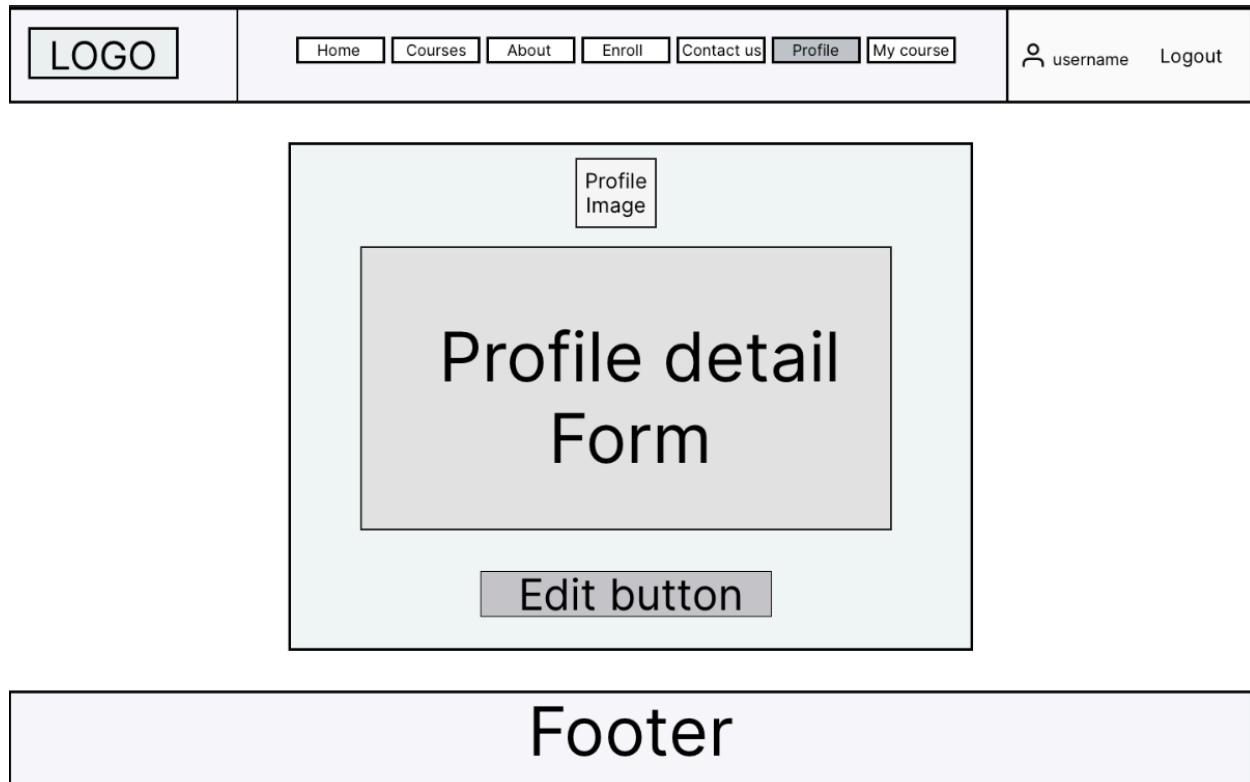


Fig 4 : User Profile Page Wireframe

After logging in into the system by the user, they will see their details in the profile page. The picture of user will be above detail form and they can change their profile picture after uploading it. Edit button will be also situated just below the detail form to edit their information if they want it. Header, footer and navigation bar will be situated at their respective position as same as other related pages.



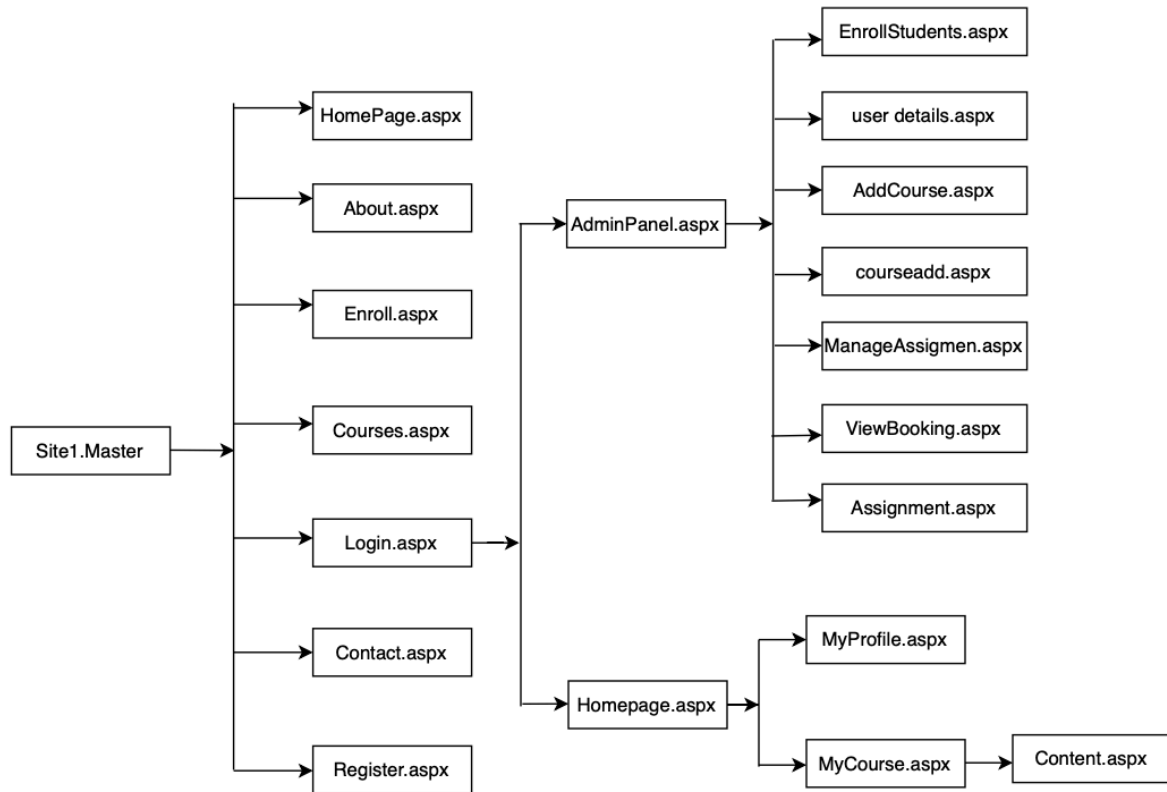


Fig 9 : Website Navigational Structure

**Course content page**

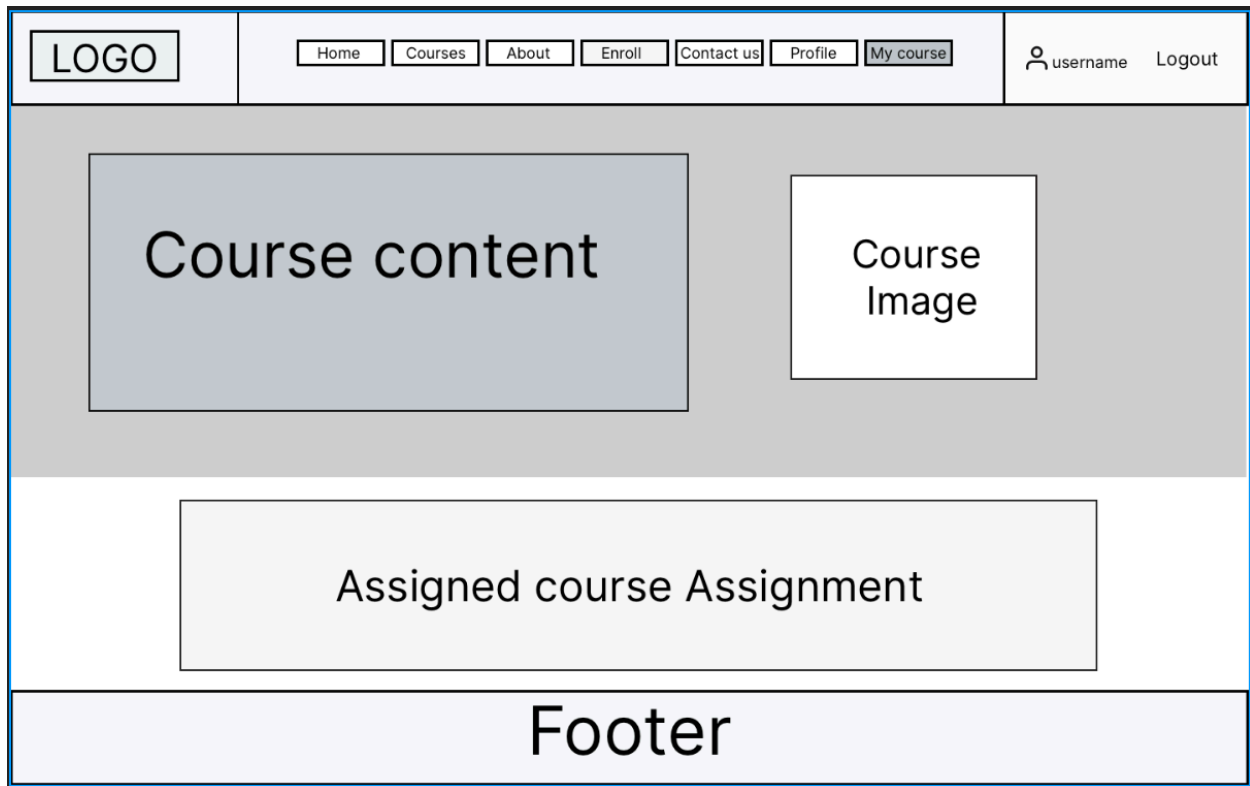
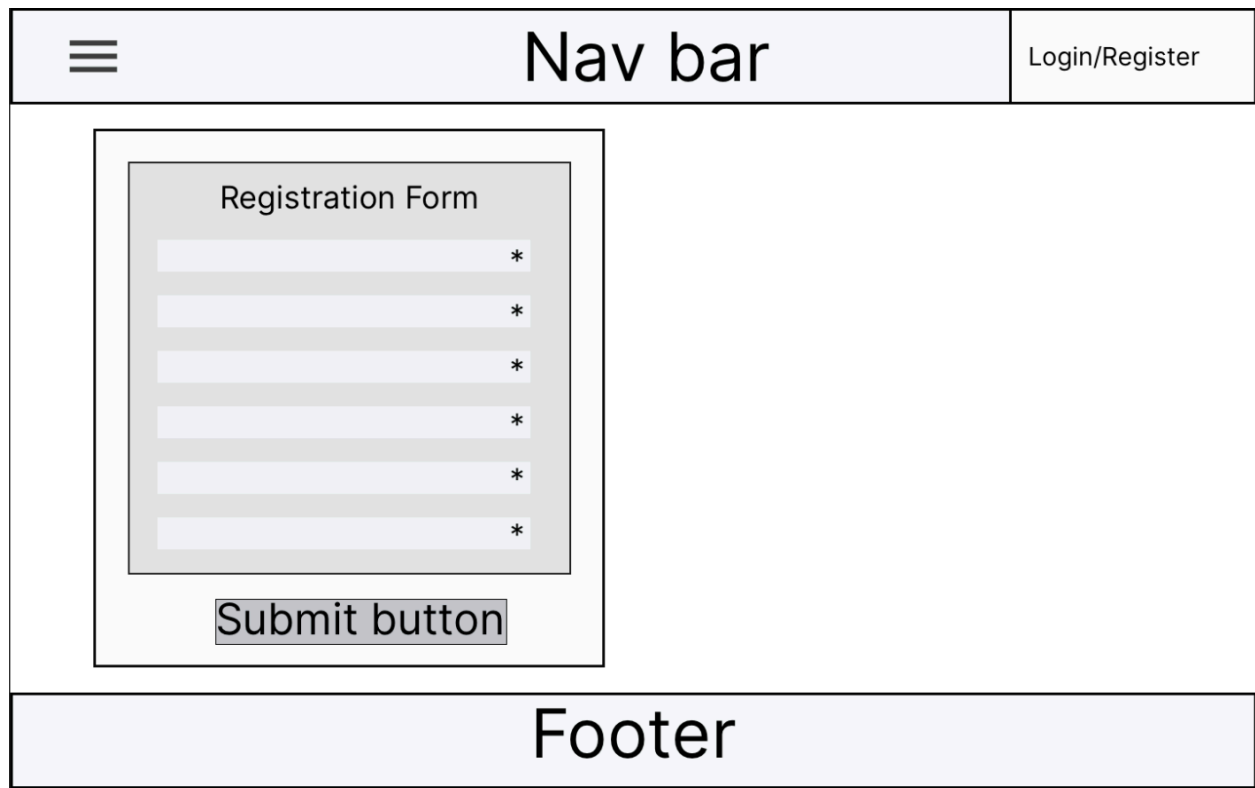


Fig 5 : Course Content Page Wireframe

The user who already signed in will see the button of My Course in navigation bar. They will see their enrolled course name in left position below the header. Along with the name of courses, images and brief description of the courses will be set. The enrolled course name will be clickable and they will view course content page after clicking it.

### **User Registration page**



The wireframe illustrates the layout of a user registration page. At the top, a light blue navigation bar (Nav bar) contains a hamburger menu icon on the left and the text "Login/Register" on the right. The main content area is white and features a registration form on the left side. The form is enclosed in a light gray box and includes the title "Registration Form" at the top. Below the title are six input fields, each followed by an asterisk (\*) indicating a required field. At the bottom of the form is a "Submit button". The footer of the page is a light blue bar with the word "Footer" centered in it.

Fig 6 : User Registration Page Wireframe

This page will be interconnected to login page as the new user must register to the system for logging into the system. User will need to fill up the necessary details and click the submit button. Navigation bar lies at top of this page while the button of menu bar will also be at top of the page.

**Add course form**

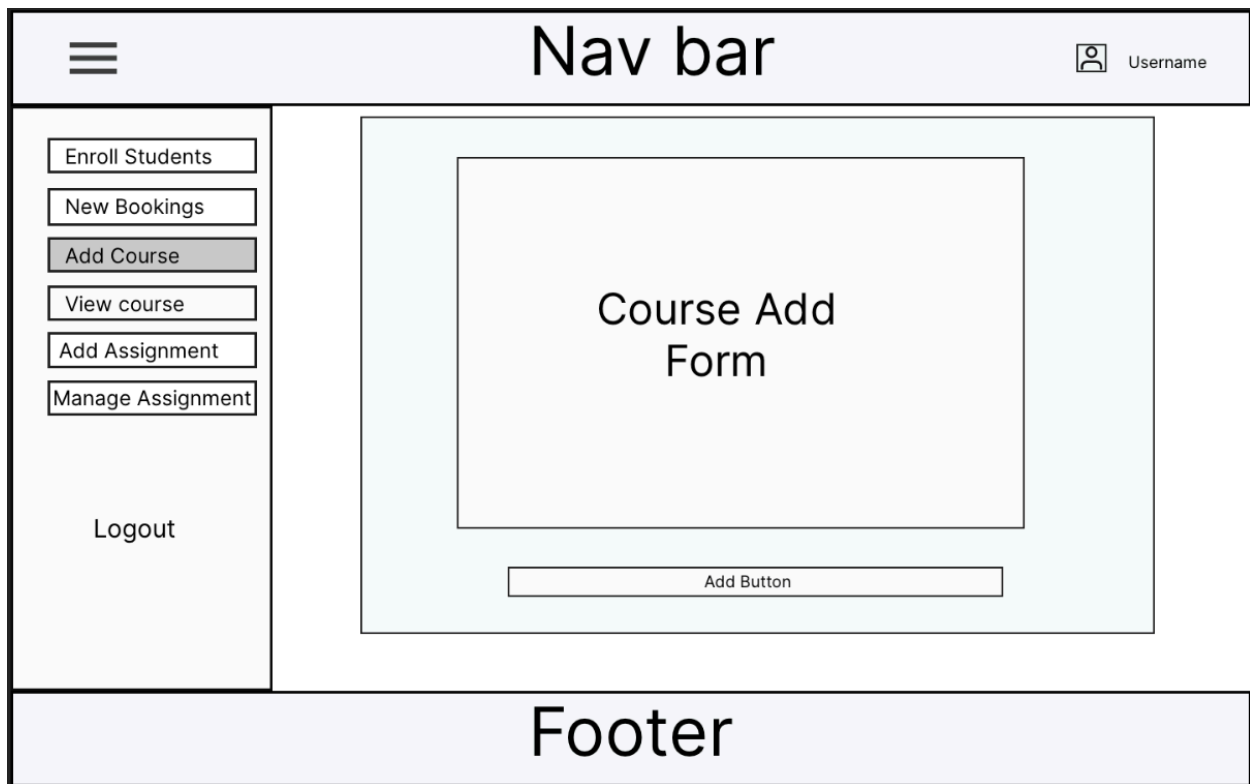


Fig 7 : Add Course Page Wireframe

Add Course page is the page linked from menu bar. Navigation bar will be lies at top position. User will need to fill the course details for which course they wants and click on add button. After adding their willing course, they can go to other pages.

## View Course Page

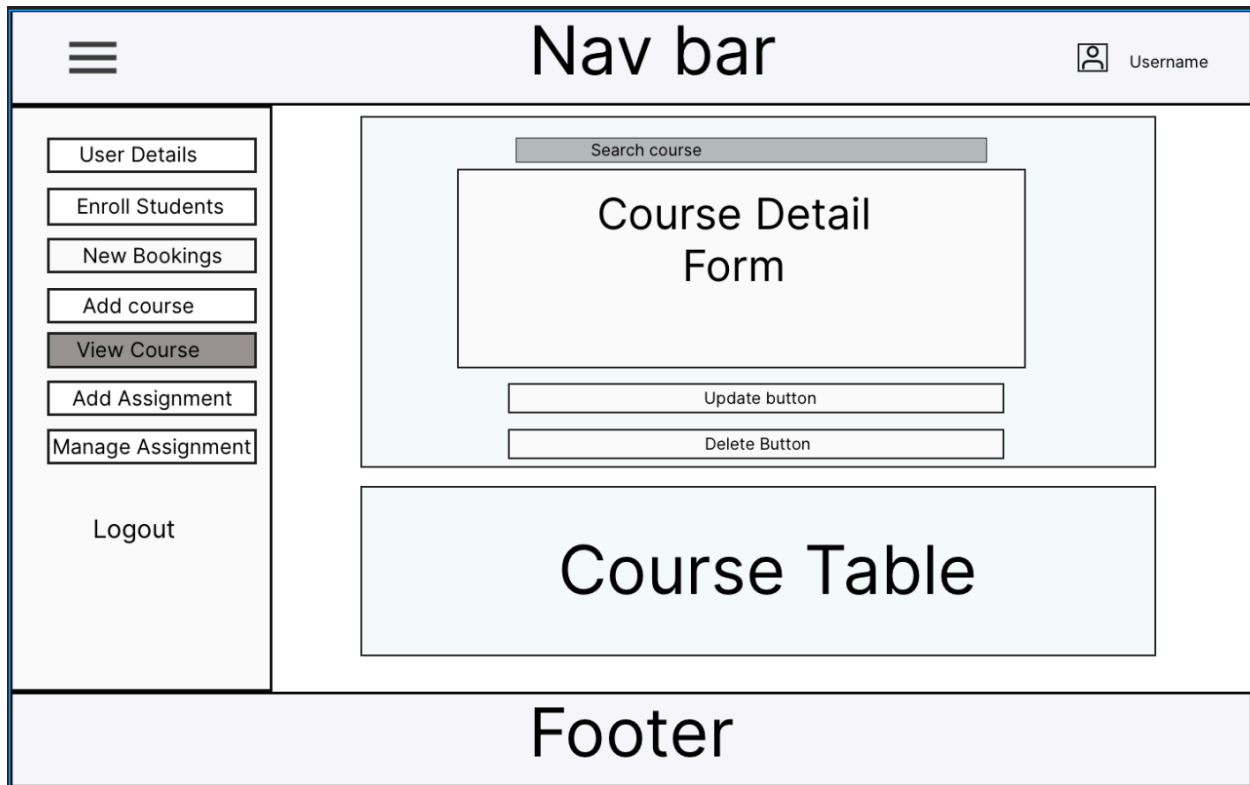


Fig 8 : View Course Page Wireframe

In view course page, navigation bar will be at top position. For the easy usage, they can search their course from search button which is just below the navigation bar if they have lots of courses. The details of courses will be at middle and they can update and delete their course by clicking the respective button. The table of courses will also be at just above the header.

### 3.4. Website Navigational Structure

Above illustrate diagram is the navigation structure of 'Mind-education' application which we are going to develop. Users are categorised into three types that is guest , registered and admin. Guest user can only view the HomePage.aspx, Courses.aspx, About.aspx, Contact.aspx,

Login.aspx and Register.aspx along with enroll button. If the user try to enroll with out logged in into the system then he/she will be automatically directed to the Login.aspx page but they can view the available course which will be displayed in course.aspx content retrieved from database. After user register into the system then they can login into our system with the help of username and password which they have previously registered. After user perform login then additional button such as profile and my course and logout button along with their username and profile are made visible automatically where as login and register button are made invisible. If user clicks profile button then MyProfile.aspx page will be loaded where his/her profile details will be shown in form which they can update by themselves. Now if user clicks enroll button then Enroll.aspx page is opened which enables them to enrolled into the available course. Before they got enrolled they need to pay the charges with the help of their debit card. Without completing the payment they are unable to enroll in the course. Similarly, if they click my course button then they will be directed towards my course.aspx page where they can view the courses on which they have enrolled. When user clicks the specific course button then the content regarding that course are made visible in Content.aspx page which also dynamic page of our website. Here, the data stored on database are made available when user click specific course name.

If admin enters his/her username and password then Adminpanel.aspx page is appeared on the screen. This page navigation bar is linked with other various pages such as Enrollstudent.aspx, ViewBooking.aspx, AddCourse.aspx, courseadd.aspx(view course), Register.aspx and Assignment.aspx. If user clicks view course button on navigation bar then courseadd.aspx button is appeared along with available course in grid view table. Even admin can search the course and update the content stored in database. Also, they can delete the course if needed. Similarly, they can also assigned the assignment for particular course.

## Static and Dynamic pages

Static pages	Dynamic Pages
HomePage.aspx	Register.aspx
About.aspx	Login.aspx
Contact.aspx	Adminlogin.aspx
	Adminpanal.aspx
	Enroll.aspx
	Myprofile.aspx
	Mycourse.aspx
	Content.aspx
	Course.aspx
	Courseadd.aspx
	Viewcourse.aspx
	Addassignment.aspx
	Viewassignment.aspx

## 4. Implementation

### 4.1. Declaring Connection string

```
<connectionStrings>
  <add name="AspNetConn" connectionString="data source=.; database= User; integrated security=SSPI;"
    providerName="System.Data.SqlClient"/>
</connectionStrings>
```

Fig 10: Connection String Code

Above code helps to connect our system to MySQL server. This code is performed inside Web.config file . System is connected to the local server with database name “User”.



## 4.2. User Register Code (Insert)

```
0 references
protected void Rgbtn_Click(object sender, EventArgs e)
{
    try
    {
        if(con.State==System.Data.ConnectionState.Closed)
        {
            con.Open();
        }
        SqlCommand cd = new SqlCommand("select count(*) from Register where Username =" + txtusername.Text.Trim() + ";", con);
        int count = Convert.ToInt32(cd.ExecuteScalar().ToString());
        if (count > 0)
        {
            Response.Write("<script>alert('user already exist')</script>");
        }
        else
        {
            using.SaveAs(Server.MapPath("~/img/" + using.FileName));
            string ins = "insert into [Register](Name,Email,Address,Username>Password,Type,User_image) values('" + txtname.Text + "','" + txtemail.Text + "','" + txtaddress.Text + "','" + txtusername.Text + "','" + txtpw.Text + "','" + "User" + "','" + using.FileName + "')";
            SqlCommand cmd = new SqlCommand(ins, con);
            cmd.ExecuteNonQuery();
            Response.Write("<script>alert('user is registered successfully')</script>");
            Response.Redirect("Login.aspx");
        }
        con.Close();
    } catch(Exception ex)
    {
    }
}
```

Fig 11: User Registration Code

Above code is for the user registration. If user clicks register button after filling all textbox with valid information then above code is executed. SqlCommand object enable us to perform interaction with database according to our needs. So, in case of registration we need to carry out insert command in a database table. Cont(\*) helps to return the total number of rows in table, including duplicate rows. Each row is counted independently. We need to perform this task in order to find out if the username is already exists or not. If count is recorded more than 0 then a popup window shows message with “user already exists” else user gets registered into the table. Insert query is stored in string called ‘ins’ which will be passed to database via SqlCommand. Hence, data will get registered into database table.

### 4.3. Login code (Read)

```
0 references
protected void LogBtn_Click(object sender, EventArgs e)
{
    con.Open();
    SqlCommand cmd = new SqlCommand("select count(*) from Register where Username = '" + txtusername.Text + "' and Password = '" + txtpw.Text + "'", con);
    int count = Convert.ToInt32(cmd.ExecuteScalar().ToString());
    if (count > 0)
    {
        SqlCommand cmdType = new SqlCommand("select Type from Register where Username = '" + txtusername.Text + "'", con);
        string type = cmdType.ExecuteScalar().ToString();
        Session["Type"] = type;
        if (type == "Admin")
        {
            Session["username"] = txtusername.Text;
            Response.Write("successful");
            Response.Redirect("adminpanel.aspx");
        }
        else if (type == "User")
        {
            Session["username"] = txtusername.Text;
            Response.Redirect("HomePage.aspx");
        }
    }
    else
    {
        Response.Redirect("Register.aspx");
    }
    con.Close();
}
```

Fig 12: Login Code

Above shown code is the c sharp code of login button which reads the data from register table of database and compare the username and password which are entered in textbox. So to perform this task at the initial stage with the help of SqlConnection object we have connected our system with MySql server database. 'con' variable is declared to connect the server. Then, SqlCommand object is used to send the query in order to read data from database along with con variable as a parameter. If there are any record which matched the entered password and username on same row of database then user will be redirected to another page according to their types that is if the user is admin then he/she will be directed to the admin panel.aspx page else will be redirected to the HomePage.aspx of user view side. Here, Session variable is used in order to transfer the value from one page to another. Session variable can be used in other page such as in profile.aspx to display the user's data from database into the form. Response.Write() function helps to give feedback if user enter wrong information.

#### 4.4. Delete code

```
0 references
protected void deletbtn_Click(object sender, EventArgs e)
{
    try {
        con.Open();

        SqlCommand cmd = new SqlCommand("delete from Course where Course_Name='" + txtsearchcourse.Text.Trim() + "'", con);
        cmd.ExecuteNonQuery();
        con.Close();
        GridView1.DataBind();
        Response.Write("<script>alert('Course deleted Successfully')</script>");
    }
    catch
    {
    }
}
```

Fig 13: Delete Code

Sample code is an example of delete functionality. When admin clicks the delete button then above code is triggered which helps to delete Course if searched course is available in database. The code is executed in database server with the help of SqlCommand which sends delete query via SqlConnection class. Connection stream gets activated then query helps to delete data from database. Since, the total available courses are stored in table so whenever admin delete the course then course also must be deleted from table. In order to perform that activity after the data is deleted from the Course table then updated table data are bind into the table with the help of DataBind function. Here, con.Close helps to close connection between database and system.

## 4.5. Search course

```
0 references
protected void submitBtn_Click(object sender, EventArgs e)
{
    showcourseDetails();
}

1 reference
void showcourseDetails()
{
    try
    {
        SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["AspNetConn"].ConnectionString);
        if (con.State == System.Data.ConnectionState.Closed)
        {
            con.Open();
        }

        SqlCommand cmd = new SqlCommand("select * from Course where Course_Name='" + txtsearchcourse.Text.Trim() + "';", con);
        SqlDataReader da = cmd.ExecuteReader();
        if (da.HasRows)
        {
            while (da.Read())
            {
                txtcoursename.Text = da.GetValue(1).ToString();
                txtcontent.Text = da.GetValue(2).ToString();
                txtprice.Text = da.GetValue(3).ToString();
                txtlname.Text = da.GetValue(4).ToString();
                txtcduration.Text = da.GetValue(5).ToString();
                txtimage.Text = da.GetValue(6).ToString();
                txtadded.Text = da.GetValue(7).ToString();
            }
        }
    }
    catch
    {
    }
}
```

Fig 14: Search Code

Above code helps us to retrieve specific data from database and bind them to textbox of show course form. A method called showcourseDetails() is being created which fetch data from Course table of database 'AspNetConn'. As soon as user clicks the search button then above mentioned methods gets called. We have connected our system with MySQL server database with the help of SqlConnection String. Then select query is passed as a parameter along with connection string variable 'con' via SqlCommand which helps to retrieve data from database. ExecuteReader function helps to read table's data from start to end according to the query passed through SqlCommand. Hence retrieve data gets populated in DataAdapter. If course name entered in

search textbox resembled with a row of course table then data get stored into DataAdapter object. If DataAdapter object contains rows then If function return 'TRUE' and with the help of Read() function each data from row gets populated into textbox else shows the error message with "Course Does Not Exists".

## 4.6. Insert Course

```
SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["AspNetConn"].ConnectionString);

0 references
protected void Page_Load(object sender, EventArgs e)
{
}

0 references
protected void submitBtn_Click(object sender, EventArgs e)
{
    string ins = "insert into [Course](Course_Name,Course_content,Course_Price,Lecturer_Name,Class_Duration,Course_Image,Added_by) values('" + txtcname.Text + "','" + txtccontent.Text + "','" + txtcprice.Text + "','" + txtlname.Text + "','" + txtcduration.Text + "','" + FileUpload1.FileName + "','" + Session["username"] + "')";
    SqlCommand cmd = new SqlCommand(ins, con);
    con.Open();
    cmd.ExecuteNonQuery();
    FileUpload1.SaveAs(Server.MapPath("~/img/" + FileUpload1.FileName));
    con.Close();
    Response.Write("<script>alert('Course Added Successfully')</script>");
    txtcname.Text = "";
    txtccontent.Text = "";
    txtcprice.Text = "";
    txtlname.Text = "";
    txtcduration.Text = "";
}
```

Fig 15: Insert Code

Above code is implemented in order to store data into database. It is the code to insert course details into course table of database. As mentioned previously SqlConnectionString is being used to connect the system to MySQL server database. Then a string variable called 'ins' is declared with insert query which helps us to insert the data from textbox to their related column of table. SqlCommand helps to pass the query to the database through connection string variable 'con'. Session variable "username" helps to stored the username of logged in admin who have added the course into the column called "Added\_by" of Course table. Then Response.Write() function helps to alert us via popup window after course is added successfully into the table.

## 4.7. Update code

```
protected void updatebtn_Click(object sender, EventArgs e)
{
    try
    {
        con.Open();

        SqlCommand cmd = new SqlCommand("update Course set Course_Name='" + txtcoursename.Text.Trim() + "', Course_content='" + txtcontent.Text.Trim() + 
        "','Course_Price='" + txtprice.Text.Trim() + "','Lecturer_Name='" + txtlname.Text.Trim() + "','Class_Duration='" + txtcduration.Text.Trim() + "','Course_Image='" + 
        txtimage.Text.Trim() + "','Added_By='" + txtadded.Text + "' where Course_Name='" + txtsearchcourse.Text.Trim() + "'", con);
        cmd.ExecuteNonQuery();
        con.Close();
        GridView1.DataBind();
        Response.Write("<script>alert('Course Information Updated Successfully')</script>");
    }
    catch
    {
    }
}
```

Fig 16: Course Update Code

Above illustrate code is implemented to update course data from the table. The database and system are connected with the help of SqlConnection string. The update query code are passed to server via connection string variable with the help of SqlCommand object. Similarly, DataBind function helps to bind the updated data into the table. After the data are updated successfully then Response.Write function helps to inform users showing message “Course Information Updated Successfully ” in popup window.

#### 4.8. Show courses in list view

```
2 references
public partial class mycourse : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["AspNetConn"].ConnectionString);

    0 references
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            if (Session["username"] != null)
            {
                shownenroll();
                Session["mycourse"] = "Course_Name";
            }
        }
    }

    1 reference
    void shownenroll()
    {
        con.Open();
        string s = "select Course_Name from EnrollStudents where Username='"+Session["username"]+"'";
        SqlDataAdapter da = new SqlDataAdapter(s, con);
        DataSet dt = new DataSet();
        da.Fill(dt);
        ListView1.DataSource = dt;
        ListView1.DataBind();
        con.Close();
    }
}
```

Fig 17: Show Course Code

The code helps to display the courses in which particular user has got enrolled. With the help of session variable the list of the courses will be shown on which the logged in user is enrolled. When user clicks the 'Mycourse' button then they will be redirected to the mycourse.aspx page where the courses get loaded so the activity is performed in Page\_Load method. If the session variable has got value then only shownenroll() method get called. In shownenroll function a string variable 's' is declared which helps to retrieve only those Courses on which that particular user has enrolled from EnrollStudent table and data are bind in list view with the help of DataBind function. Similarly, Eval function(server side code) is used in html in order to list down the course name in frontend of the application .

## 5. User Guidance

### 5.1. Home page

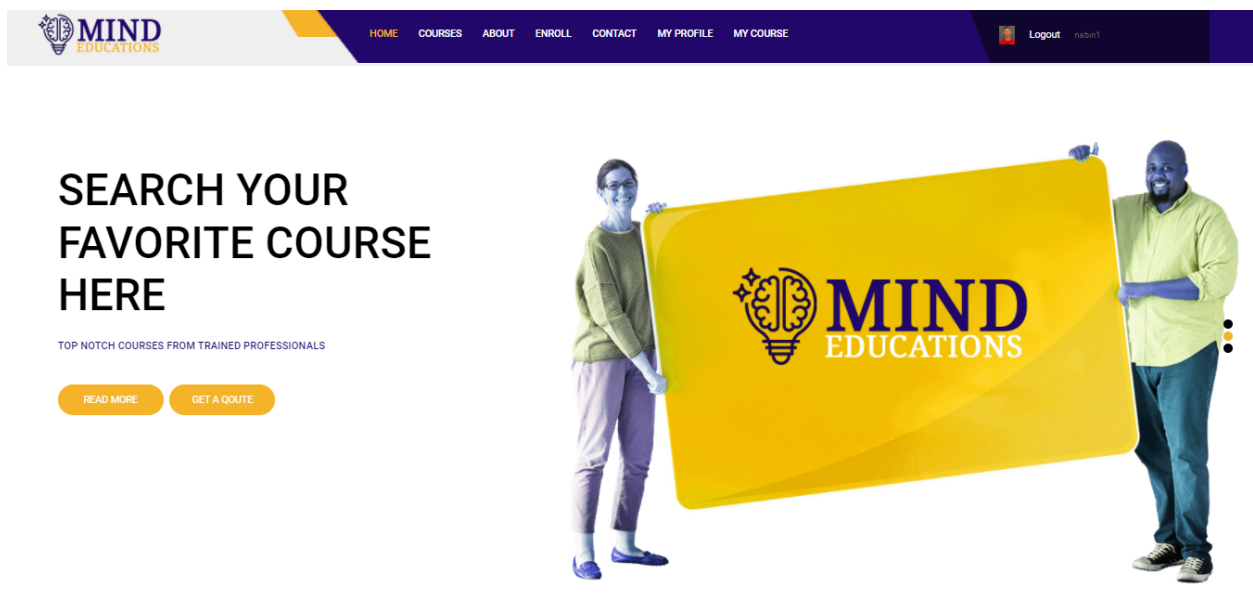


Fig 18: Home Page

The figure above us as 'System Home Page'. This is the first page where user visits while visiting the web application. This page produce features regarding course, enrollment and log in. The navigation bar comprises as 'Home', 'Course', 'About', 'Contact', 'My profile', 'My Course' and Log out.



## 5.2. Register page

**MIND EDUCATIONS**

[HOME](#) [COURSES](#) [ABOUT](#) [ENROLL](#) [CONTACT](#) [Login](#) [Registration](#)

Name  
Sandesh Giri

Email  
sandesh@gmail.com

We'll never share your email with anyone else.

Address  
1234 Main St  
**Address is mandatory!!**

Username  
sandesh123

Password  
\*\*\*\*\*

Upload Image Here  robert.jpg

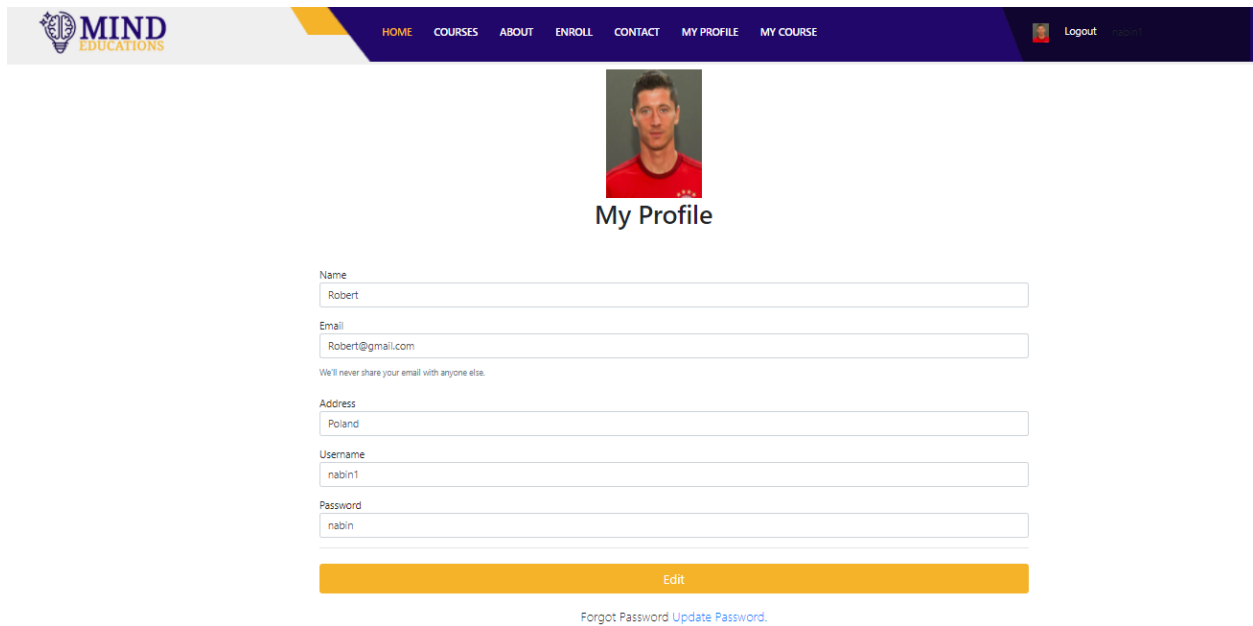
☒ I agree to the terms and conditions

[Already have an account? Sign in.](#)

Fig 19: Registration Page

Register page is used to gather detail about personal information of user. In this page, user can register their particular information in order to login into the “Mind Education” application. With the intension or avoiding user error, several rules and types of validation is implemented. In case of any error, the system will provide error notification for user.

### 5.3. User Profile



The screenshot displays the 'My Profile' page of the MIND EDUCATIONS application. At the top, a dark blue navigation bar contains the MIND EDUCATIONS logo on the left and a series of menu items: HOME, COURSES, ABOUT, ENROLL, CONTACT, MY PROFILE (highlighted), and MY COURSE. On the right side of the navigation bar, there is a 'Logout' button and a user profile icon. Below the navigation bar, the page features a profile picture of a man in a red shirt. Underneath the picture is the heading 'My Profile'. The profile information is presented in a form with the following fields: Name (Robert), Email (Robert@gmail.com), Address (Poland), Username (nabin1), and Password (nabin). A small text note below the email field states, 'We'll never share your email with anyone else.' At the bottom of the form, there is a prominent orange 'Edit' button. Below the button, there are two links: 'Forgot Password' and 'Update Password'.

Fig 20: User Profile Page

In this page, user can see their personal details if only they logged in into the application. Furthermore, there is an update button where user have permission to change their details. The personal details where user register their information are demonstrated in the user profile.

## 5.4. Course



Fig 21: Course List Page



Fig 22: Course Content Display Page

The Course page displays all the available courses. The courses on which particular user is enrolled are displayed on the page. However, user can only access this page, if he/she is logged

into the system. Once user is logged in, he or she can select enroll and access course files and contact. If user wants to view the course content along with assignment of the specific course he/she need to click the course from the list. Hence, above page gets appeared.

## 5.5. View Course

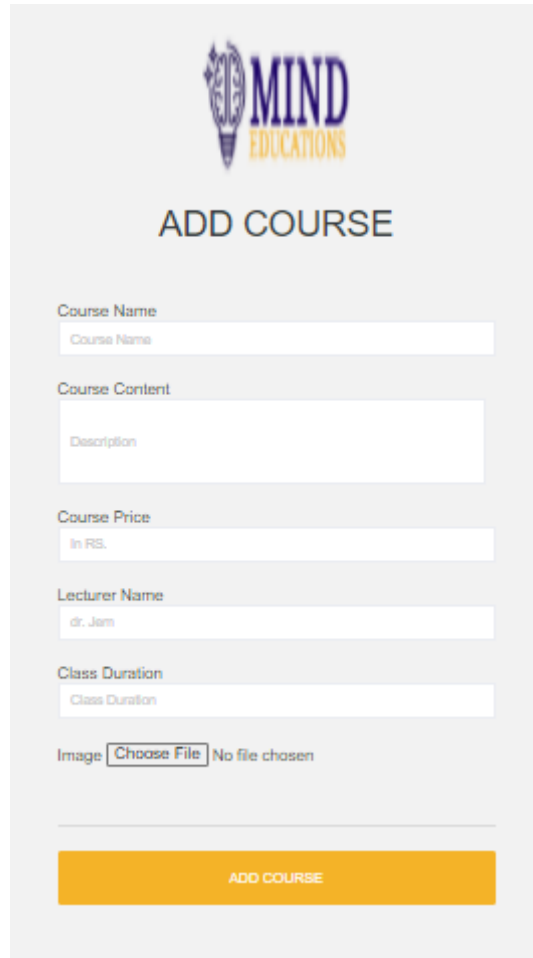
The screenshot displays a web interface for managing courses. At the top, there is a search bar with the text 'Html' and a yellow 'SEARCH' button. Below this is a form with several input fields: 'Course Name' (containing 'Html'), 'Course Content' (containing a paragraph about HTML), 'Course Price' (containing '1200 Rs.'), 'Lecturer Name' (containing 'Mr. Jupiter Tamrakar'), 'Class Duration' (containing '12 hours'), 'Image name:' (containing 'html.png'), and 'Added by:' (containing 'surl23'). At the bottom of the form are two yellow buttons labeled 'UPDATE' and 'DELET'. Below the form is a table with two rows of course data. The first row shows 'C programming language' with a cost of 1200 Rs., lecturer Sushil Adhikari, and a duration of 23 hours, accompanied by a blue 'C' logo. The second row shows 'Html' with a cost of 1200 Rs., lecturer Mr. Jupiter Tamrakar, and a duration of 12 hours, accompanied by an 'HTML' logo.

Course_ID	Course Name	Cost	Lecturer Name	class duration	Image
4	C programming language	1200 Rs.	Sushil Adhikari	23 hours	
3	Html	1200 Rs.	Mr. Jupiter Tamrakar	12 hours	

Fig 23: View Course Page

The 'View Course' page allows system admin to find all the available courses. At the top of this pages, there is a 'Search' option where admin can find their preferred courses.

## 5.6. Add Course



**MIND EDUCATIONS**

### ADD COURSE

Course Name  
Course Name

Course Content  
Description

Course Price  
In RS.

Lecturer Name  
dr. Jam

Class Duration  
Class Duration

Image  No file chosen

**ADD COURSE**

Fig 24: Add Course Page

The 'add course' page is used to upload documents into system. The page can be accessed by admin but not by users. To upload a file, admin must provide name, content information and price of that course.

## **6. Conclusions (Summary and Future Enhancements)**

Overall, this assignment has been completed by our four members team where each individual group member has equal contribution to make it responsive. Dividing task and obstacles faced together while caring out the project are the key points to get succeed in this website. Each individual member shows their eagerness and dedication to make high-quality and good web application. Each individual member's contribution helps us to make easier to complete the websites.

This assignment is proper responsive web application which includes “CSS”, “HTML”, “ASP.NET framework” and “JS”. With the help of “Bootstrap”, the website looks user friendly. After using bootstrap, it needs lots of energy and time to get proper knowledge about “JavaScript”, “HTML”, “CSS”. However, the website has been completed in time. The extra knowledge has been able to gain about functional website while replacing it with hard-coded website.

### **Future Enhancement**

- Online tutors to help students clear doubts.
- Addition of video conferencing and tutorial feature.
- Introduce online community for users interaction.

## 7. Reference

- ("E-Learning Management System ER Diagram | FreeProjectz", 2022)
- *Entity Relationship Diagram (ERD) - What is an ER Diagram?* (1994). Smartdraw. <https://www.smartdraw.com/entity-relationship-diagram/>
- (Friedman, 2022)**Your Bibliography:** Friedman, J. (2022). Lesson 03: The SqlCommand Object - C# Station. Retrieved 20 February 2022, from <https://csharp-station.com/Tutorial/AdoDotNet/Lesson03>
- Martin, M. (2021b, December 31). *25 Best FREE Wireframe Tools & Software in 2022*. Guru99. <https://www.guru99.com/best-wireframe-tools.html#:~:text=Wireframe%20tools%20are%20visual%20mockup%20software%20that%20outline%20s,share%20your%20site%20design%20with%20colleagues%20in%20real-time>.
- ("SQL Server Connection Strings for ASP.NET Web Applications", 2022)**Your Bibliography:** SQL Server Connection Strings for ASP.NET Web Applications. (2022). Retrieved 20 February 2022, from [https://docs.microsoft.com/en-us/previous-versions/aspnet/jj653752\(v=vs.110\)](https://docs.microsoft.com/en-us/previous-versions/aspnet/jj653752(v=vs.110))
- *storyboarding - Zoeken*. (2019). studiobinder. <https://www.bing.com/search?q=storyboarding&cvid=1cd58d1a6a674c60874fd00951abbac8&aqs=edge.0.69i59j69i57j0l5j69i60l2.2462j0j1&pqlt=2083&FORM=ANNTA1&PC=U531>



- Sundar, P. (2018, September 18). *Website Structure and Navigation: Basic Web Design Rules*. CrazyLeaf Design Blog. <https://www.crazyleafdesign.com/blog/website-structure-and-navigation-basic-web-design-rules/>
- *What is Data Dictionary*. (2015). Tutorialpoints. <https://www.tutorialspoint.com/What-is-Data-Dictionary>
- **Your Bibliography:** E-Learning Management System ER Diagram | FreeProjectz. (2022). Retrieved 5 March 2022, from <https://www.freeprojectz.com/entity-relationship/e-learning-management-system-er-diagram>

## **8. Appendix**

### **Website Title**

The title of our web application is “Mind Education”.

### **Objectives**

The objective of our web application are mentioned below:

- Create a platform where educational resources are shared.
- To improve the learning and teaching experience.
- To encounter student’s learning styles and requirements.
- To engage learners in the learning process.

### **Audience Modelling**

Categorisation of audience:

- Guest - Non-registered audience
- Member - Register User
- Admin(System Controller)