**Athlete Tracking Project**

**Final Report**

**Prepared by Harun Yahya Ünal - 210706015**

CONTENT

[1. Project Planning and Research 3](#_Toc187592232)

[2. Design 4](#_Toc187592233)

[3. Database Design and Diagrams 15](#_Toc187592234)

[4. What I did in The Project 18](#_Toc187592235)

1. Project Planning and Research

**About Athlete Tracking:**

Athlete tracking is an important part of sports and fitness management. It helps monitor an athlete’s performance, growth, and overall progress. With the use of technology, such as databases and web applications, sports schools can manage training sessions, track performance data, and keep financial records more easily. Many modern systems include features like progress charts, payment reminders, and detailed reports to help with decision-making. These tools are used in sports schools and gyms to support athletes and make work more efficient. This project aims to create an easy-to-use system that meets the needs of sports schools.

**Technologies Planning to Use:**

* **Frontend**: HTML, CSS, ASPX for user interfaces.
* **Backend**: ASP.NET Core for server-side logic.
* **ORM**: .NET Entity Framework Core for using the database easily
* **Database**: MSSQL Server for storing and managing data.
* **Tools**: Visual Studio (for development), SQL Server Management Studio (for database management), Figma (for designing).

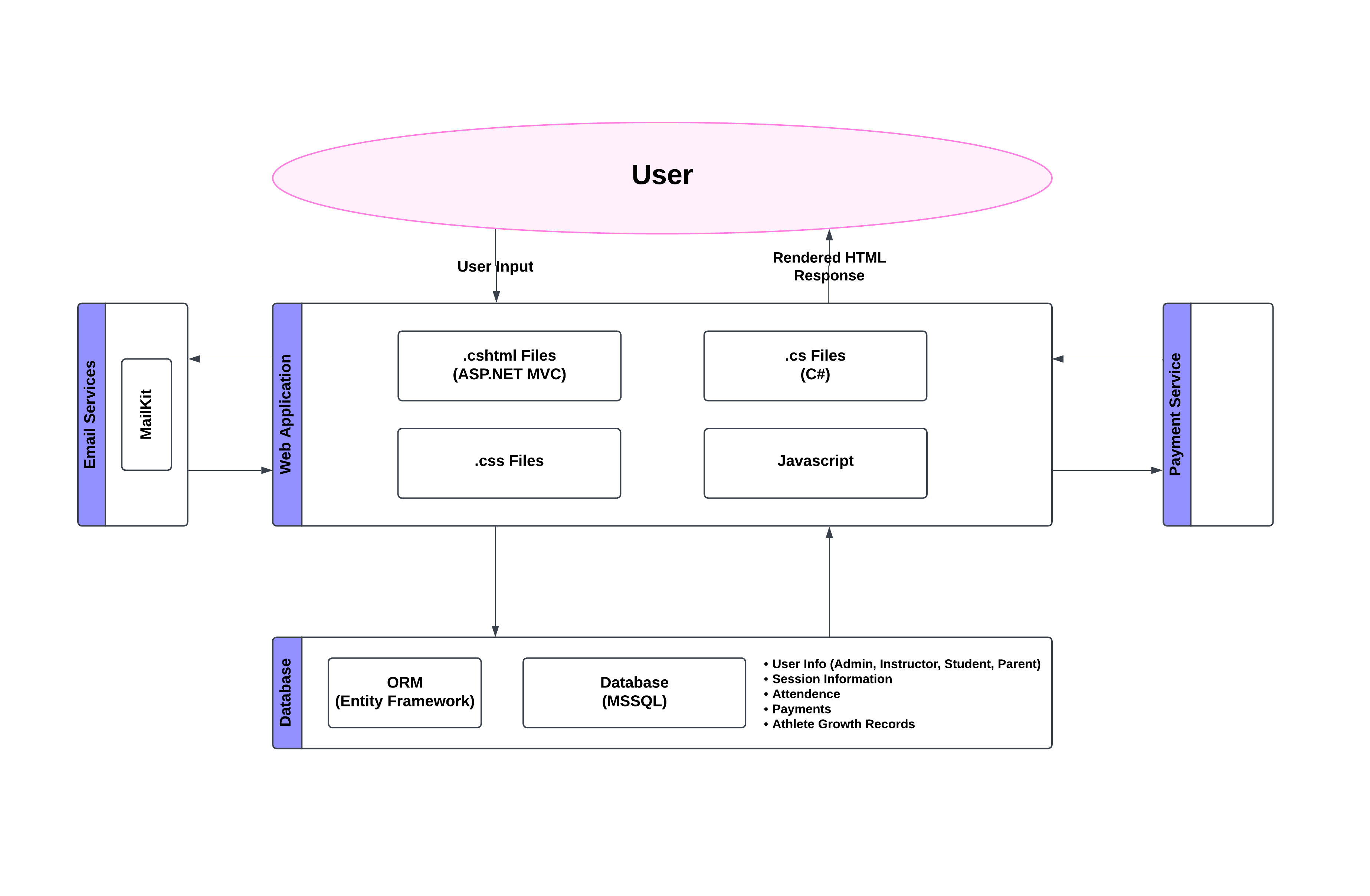
**ASP.NET Core** is ideal for server-side programming due to its cross-platform capabilities and integration with modern tools.

**MSSQL Server** offers strong relational database management features that align well with the structured nature of the data.

In the .NET environment, using **ASP.NET Core,** **MSSQL Server** and **Entity Framework Core** is easier because they work well together, are fast, and help create secure and reliable applications. ASP.NET Core makes it simple to build web applications, and MSSQL Server helps store and manage data effectively.

1. Design

**High-Level Diagram:**



**User Interfaces:**

**LoginPage:**

**A computer screen shot of a blue sign

Description automatically generated**

**RegisterPage:**

**A blue sign with white text

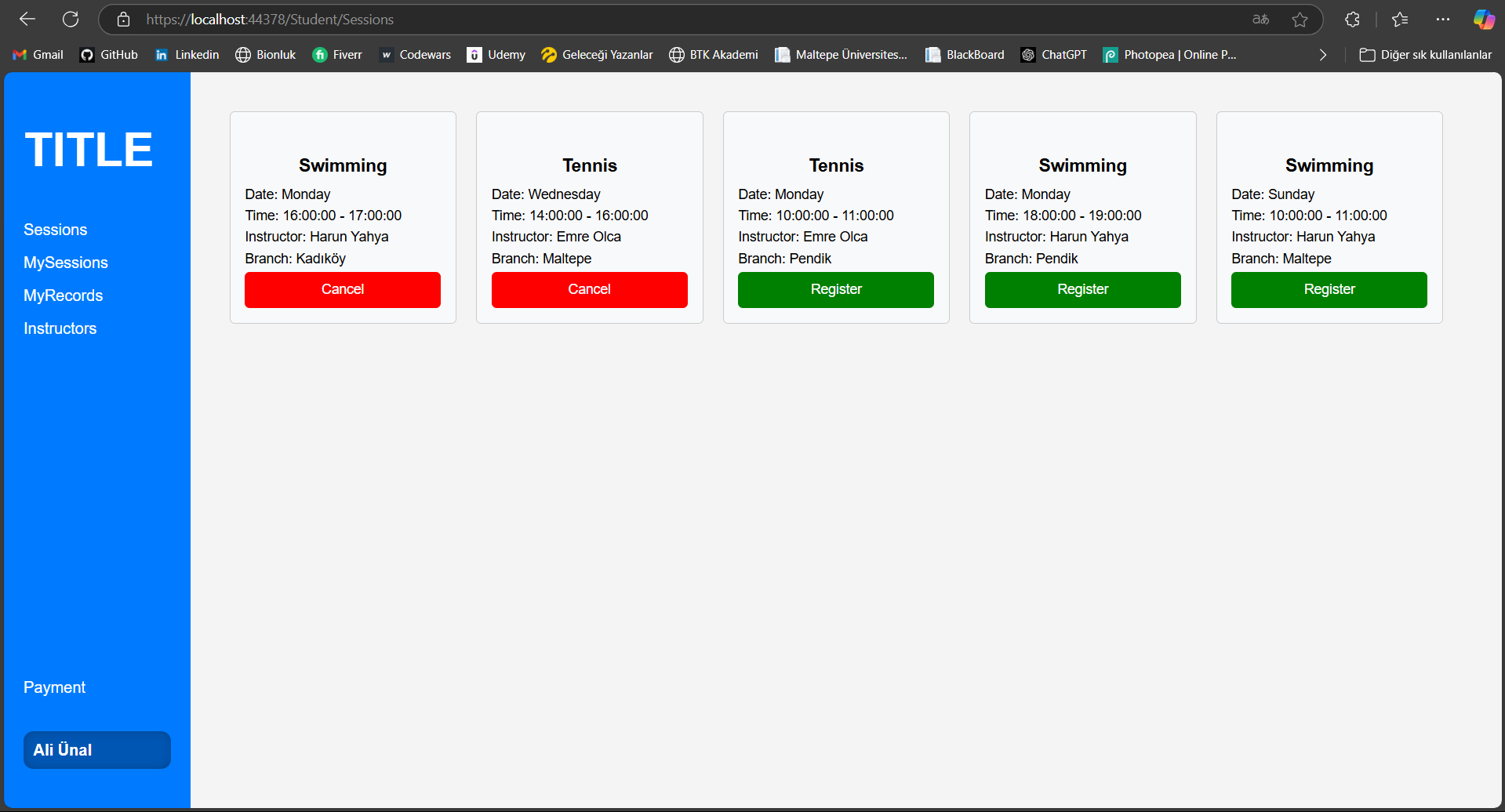
Description automatically generated**

**HomePage(Student):**

**A screenshot of a computer

Description automatically generated**

**SessionsPage(Student):**

****

**MySessionPage(Student):**

**A screenshot of a computer

Description automatically generated**

**MyRecordsPage(Student):**

**A screenshot of a computer

Description automatically generated**

**InstructorsPage(Student):**

**A screenshot of a computer

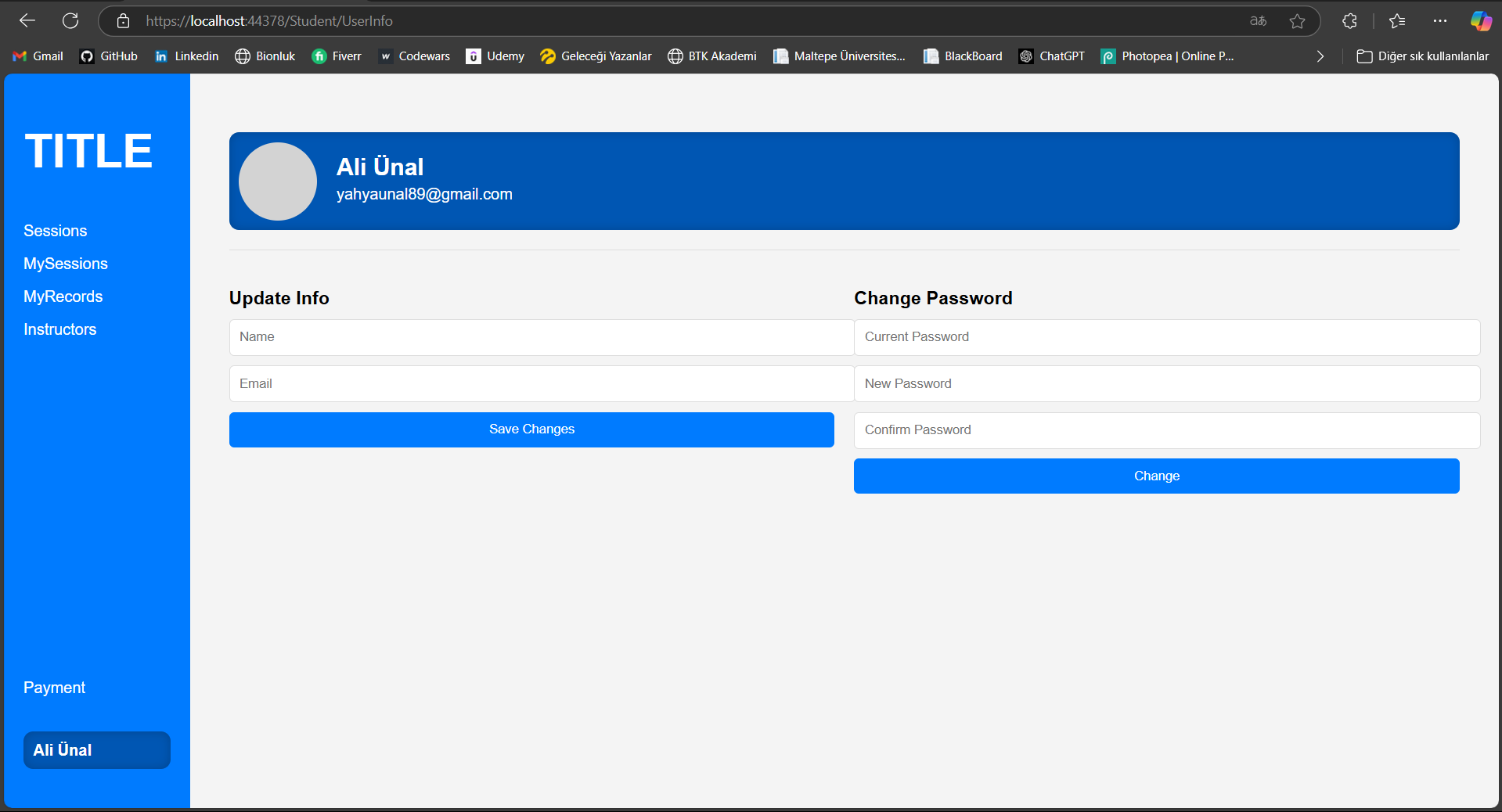
Description automatically generated**

**PaymentPage(Student):**

**A screenshot of a computer

Description automatically generated**

**InfoPage(Student):**

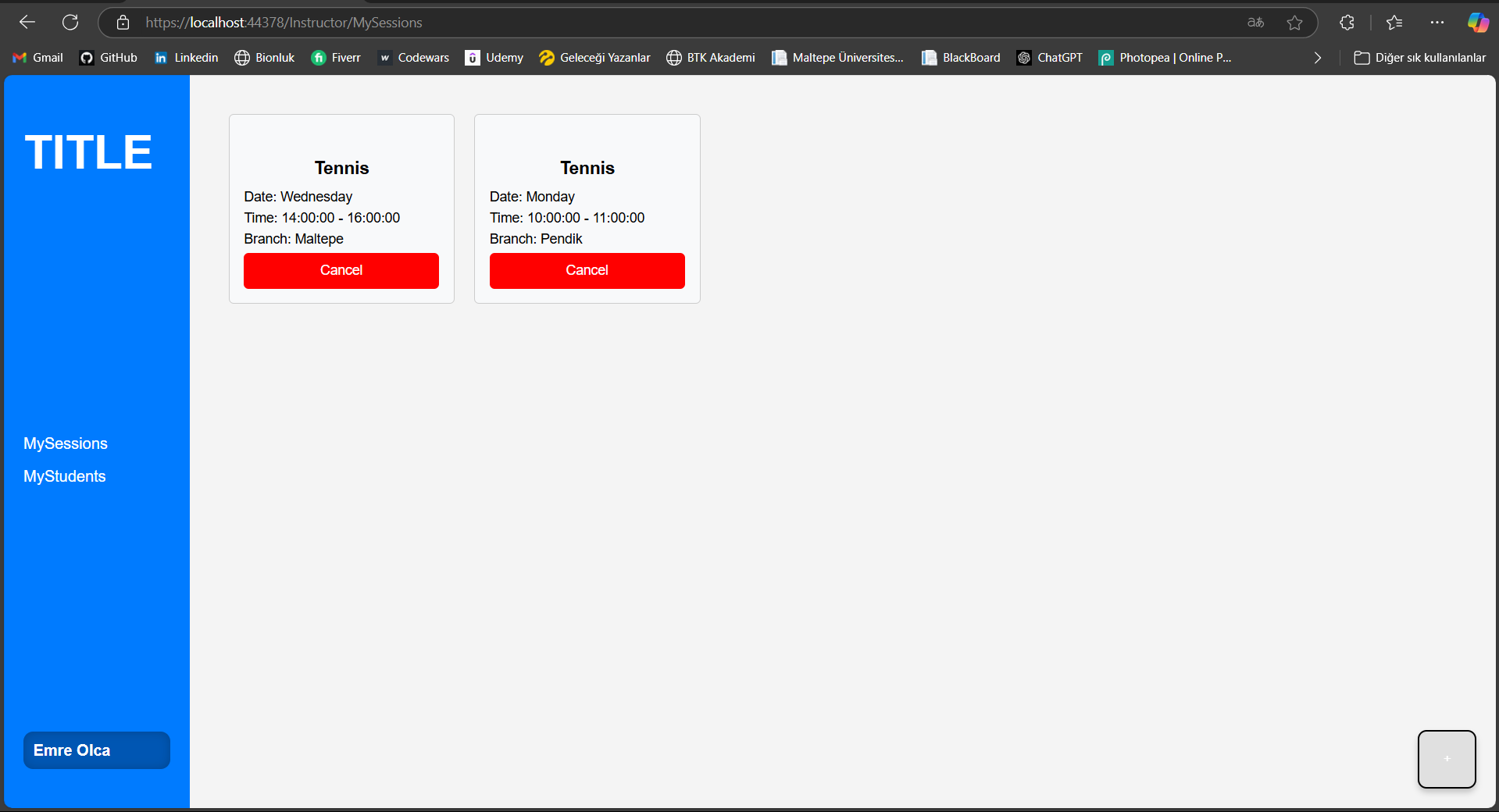
****

**HomePage(Instructor):**

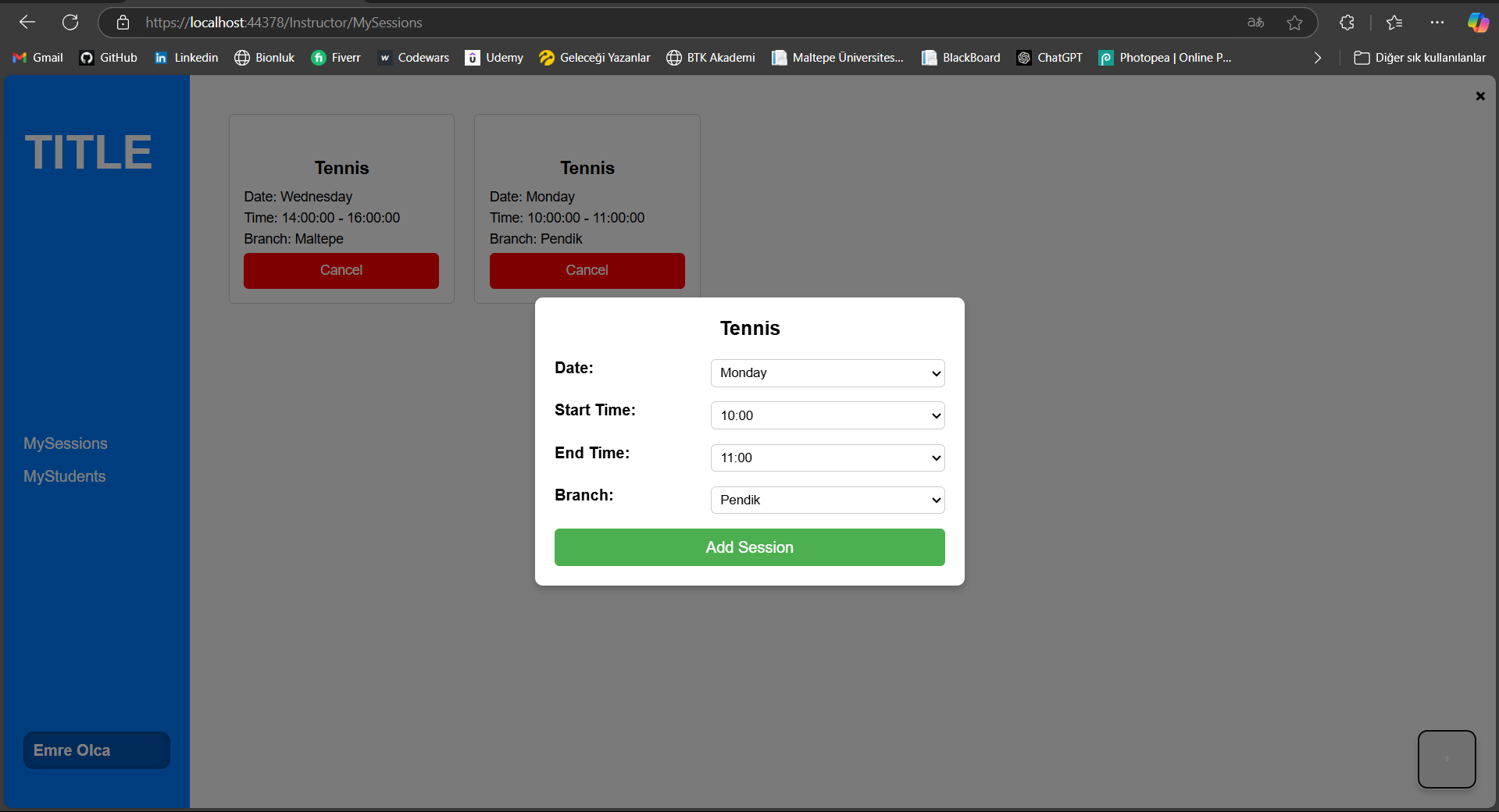
**A screenshot of a computer

Description automatically generated**

**MySessionsPage(Instructor):**

****

**AddSessionPage(Instructor):**

****

**MyStudentsPage(Instructor):**

**A screenshot of a computer

Description automatically generated**

**AddRecordPage(Instructor):**

**A screenshot of a computer

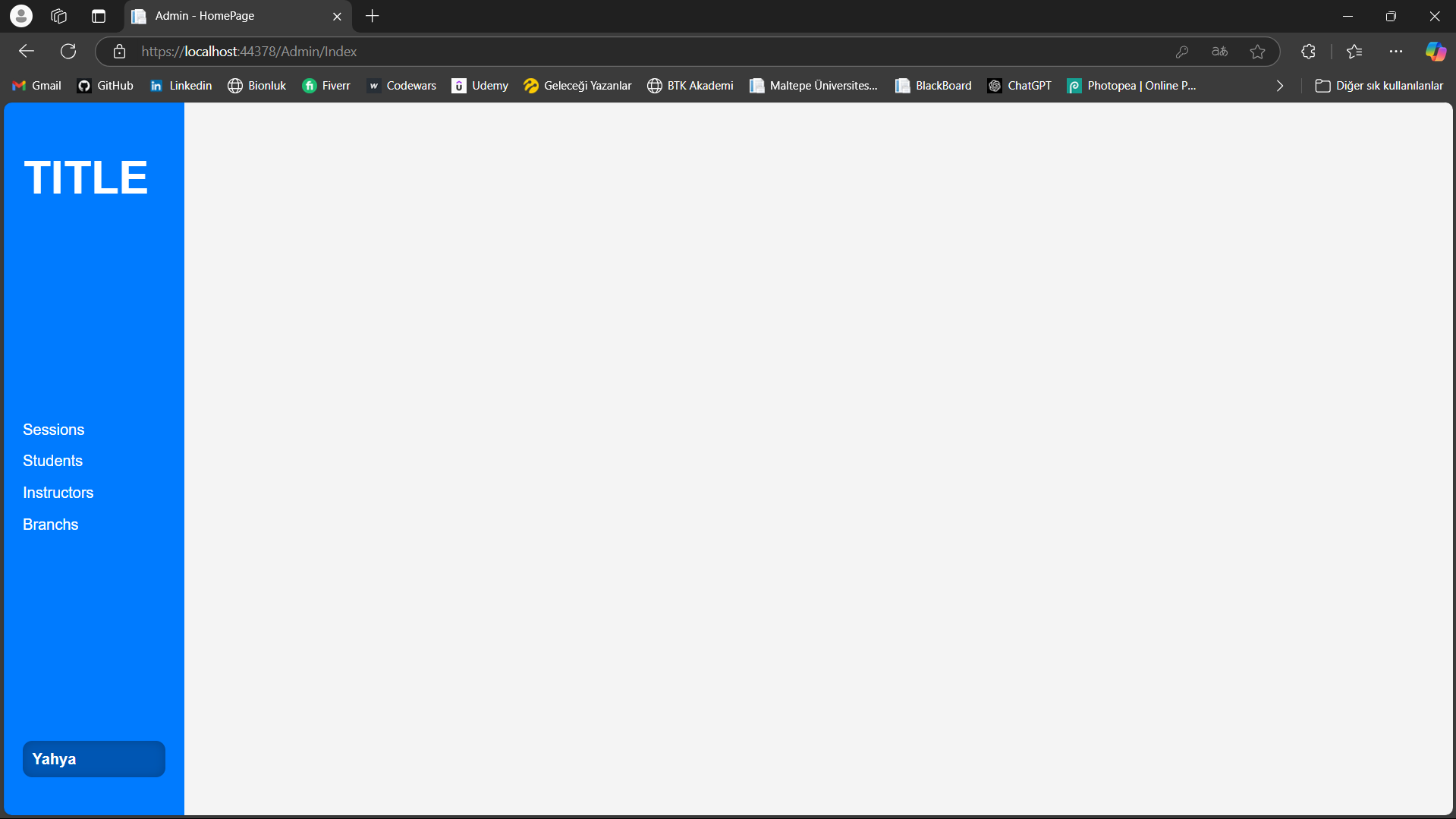
Description automatically generated**

**InfoPage(Instructor):**

**A screenshot of a computer

Description automatically generated**

**HomePage(Admin):**

****

**SessionsPage(Admin):**

**A screenshot of a computer

Description automatically generated**

**StudentsPage(Admin):**

**A screenshot of a computer

Description automatically generated**

**InstructorsPage(Admin):**

**A screenshot of a computer

Description automatically generated**

**BranchsPage(Admin):**

**A screenshot of a computer

Description automatically generated**

**AddBranchPage(Admin):**

**InfoPage(Admin):**

**A screenshot of a computer

Description automatically generated**

1. Database Design and Diagrams

**For Student:**

* Name of Student
* Birthday of Student (Day of Birth)
* Instructor of Student

**For Branch:**

* District of Branch
* Phone Number of Branch

**For Admin:**

* Name of Admin
* User Information
* Branch that responsible for

**For Instructor:**

* Name of Instructor
* Specialization of Instructor
* User Information

**For Session:**

* Instructor of Session
* Session Name
* Branch of Session
* Start Time, End Time and Day of Session

**For Payment:**

* Student who is responsible for payment
* Month paid or unpaid
* Due Date of Payment
* Amount of Payment
* Status of Payment (Paid, Unpaid)

**For Development Record:**

* Student who we want to follow
* Date of Development Information
* Height, Weight and BMI of Student
* Comments of Coach or Instructor for Student

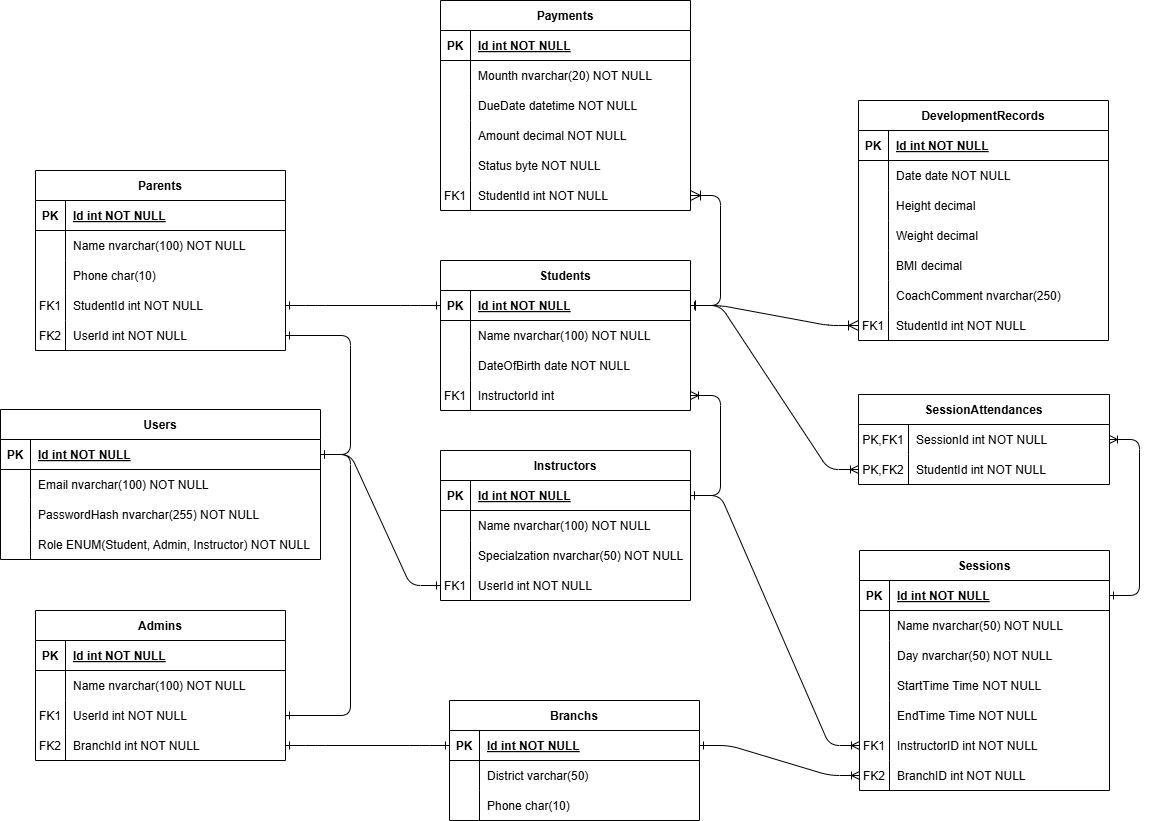
**For User:**

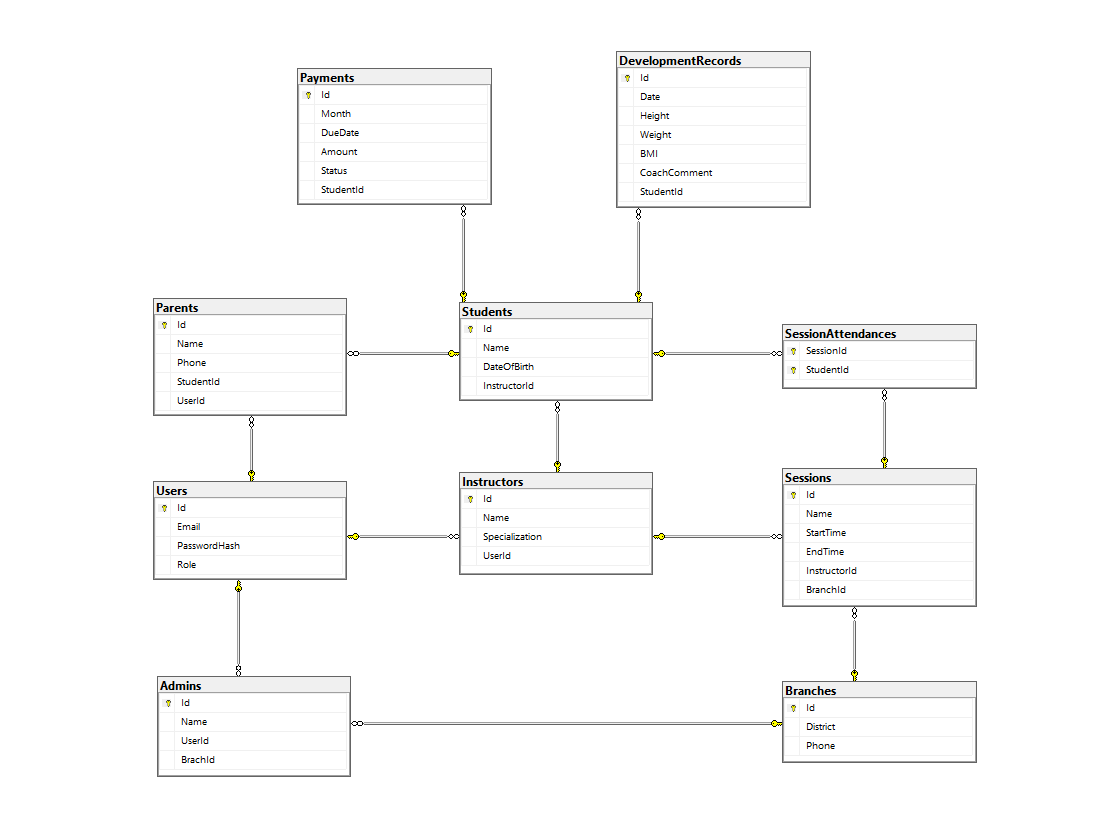
* Email
* Password
* Role (Admin, Student, Instructor)

**For Parent:**

* Name of Parent
* Phone Number of Parent
* Student of Parent
* User Information

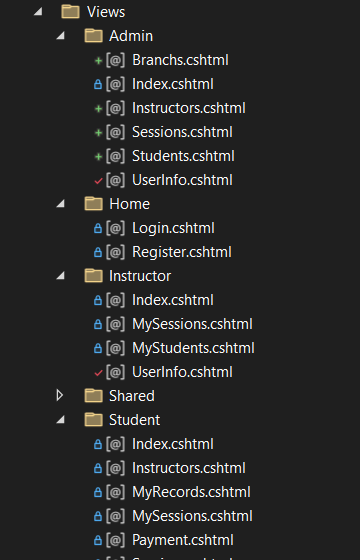
**Diagrams of Database:**

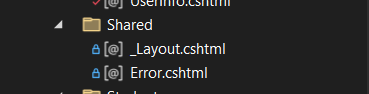
****

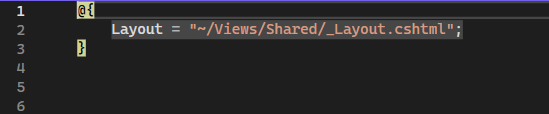
****

1. What I did in The Project

Three main parts was developed (Admin Pages, Student Pages, Instructors Pages). To develop these parts I used ASP.NET MVC.

In a MVC project, the frontend codes are written in the views folder. Each part has its own folder to be organized. And also each subpage has its own .cshtml files.

Most of these parts have common design part. To avoid excessive workload, I wrote this common part in the \_Layout.cshtml file.

After defining the layout. I used it for each pages.

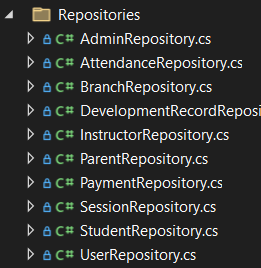
A screen shot of a computer

Description automatically generatedStyle file (.css) are stored with separated folder.

A screenshot of a computer program

Description automatically generatedModels folder is used to store the models that is used to represent each data information.

Also these models are used for creating migration to access the database.



A repository was created for each model. In these repository files, I wrote the database operations such as add, get, update.

A screenshot of a computer program

Description automatically generated

I created controller files to use ActionMethods. These action methos are used for every action such as displaying website http request (GET, POST).

A screenshot of a computer program

Description automatically generatedTo execute these action methods. We need routes for them. These routes are written in the RouteConfig.cs file. Some routes are showed below.

A screen shot of a computer program

Description automatically generated

All I mentioned things are new for me. I have used all these just once before. I learned all of this while doing this project.