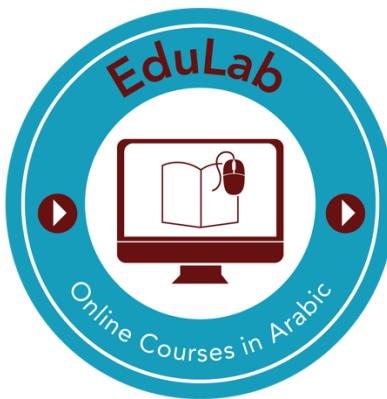




جامعة الأميرة سمية
University for Technology
للتكنولوجيا

Princess Sumaya University for Technology
King Hussein School for Computing Sciences



EduLab
E-learning Platform

Prepared By:
Ahmad Hani Mashal

Supervised By:
Dr. Walid Salameh
Dr. Sara Tadmori

Project Submitted in partial fulfillment for the degree of Bachelor of Science in
Computer Science
Semester-Year 2022/2023

Declaration of Originality

This document has been written entirely by the undersigned team members of the project. The source of every quoted text is clearly cited and there is no ambiguity in where the quoted text begins and ends. The source of any illustration, image or table that is not the work of the team members is also clearly cited. We are aware that using non-original text or material or paraphrasing or modifying it without proper citation is a violation of the university's regulations and is subject to legal actions.

Names and Signatures of team members:

Ahmad Hani Mashal

Acknowledgments

Express your appreciation to whoever helped you during your work or even before!

With overflowing gratitude and appreciation, I want to express my deepest thanks to my dear parents, who have been my rock and my backbone throughout my journey. Their unwavering love and support have been the driving force behind my success and without them, I would not have been able to reach the heights that I have today. My sincerest appreciation also goes out to my closest friends, Hani and Loay especially, who have been by my side through every step of the way, cheering me on and keeping me motivated. They have been an invaluable source of support and inspiration to me. I also want to extend a warm thank you to Dr. Walid Salameh, whose brilliant idea sparked the inception of this project, and for whom I am forever grateful. And last but not least, I cannot forget to thank Dr. Sara Tadmori for her unwavering patience, understanding, and invaluable contributions to this project. I am eternally grateful for all her support and guidance, and I will always remember her kindness and generosity.

Summary

The world nowadays has taught us that learning online is no less important than learning at an educational institution in person! When one thinks of convenience and flexibility regarding education they immediately think of online education. Moreover, online education offers more individual attention and focuses on more real-world skills. Therefore, coming up with an e-learning platform that offers courses spoken in Arabic language is the perfect solution for college students, employees, freelancers, and business owners in the MENA region.

Edu Lab E-learning Platform is a set of paid courses in different fields that serve the purpose of increasing client skills and make them ready for work environment. The platform guides e-learners through the day through seamless lessons, points them to the resources they need to complete their work, and enables them to collaborate with other e-learners in the platform.

The mission is to provide digital courses that are interactive and aligned to standards, improve skills and proficiency, and to increment educational background for better performance. The goal of the EduLab is to be the first platform that offers all of its courses in Arabic to clients from all over the MENA region

List of Abbreviations

List the abbreviations you have used in your project if there are any and what they stand for.

LMS: Learning Management System.

JS: JavaScript.

DLL: Dynamic Link Library.

CMS: Content Management System.

Table of Contents

<i>Chapter 1</i>	
<i>Introduction</i>	11
1.1 Overview	11
1.2 Problem Statement	11
1.3 Related Work	11
1.4 Document Outline	12
<i>Chapter 2</i>	
<i>Project Plan</i>	13
2.1 Project Deliverables	13
2.2 Project Tasks	14
2.2 Survey Results in Arabic	16
2.3 Roles and Responsibilities	20
2.4 Risk Assessment	21
2.4.1 Risk Identification	21
2.4.2 Risk Response	21
2.5 Cost Estimation	22
2.6 Project Management Tools	22
<i>Chapter 3</i>	
<i>Requirements Specification</i>	22
3.1 Stakeholders	22
3.2 Platform Requirements	23
3.3 Functional Requirements	24
3.4 Non-Functional Requirements	26
3.5 Other Requirements	27
<i>Chapter 4</i>	
<i>System Design</i>	28
4.1 Logical Model Design	28
4.1.1 Use Case Diagrams	28
4.1.2 Class Diagram	30
4.1.3 Component Diagram	31

4.1.4 Deployment Diagram	31
4.1.5 Activity Diagram	33
4.1.5.1 User Activity Diagram	33
4.1.5.2 Instructor Activity Diagram	34
4.1.6 Sequence Diagram	35
4.1.6.1 User Sequence Diagram	35
4.1.6.2 Instructor Sequence Diagram	36
4.1.7 State Transition Diagram	37
4.1.7.1 User State Transition Diagram	37
4.1.7.2 Instructor State Transition Diagram	38
4.1.8 ER Diagram	39
4.2 Physical Model Design	40
4.2.1 DB Schema	40
4.2.2 Report Design	41
4.2.3 User Interface	43
Chapter 5 Implementation	47
5.1 Overview	47
5.1.1 Front-End	47
5.1.2 Back-End	47
5.2 Tools	47
5.2.1 Software	47
5.2.2 Libraries	48
5.2.2.1 Front-End	48
5.2.2.1 Back-End	49
5.3 Main Functionalities	51
5.3.1 Sign Up Page	51
5.3.2 Sign In Page	52
5.3.3 Forgot Password	53
5.3.5 Home Page	55
5.3.6 Connecting to the server	55
5.3.7 Schemas	56

5.3.7.1 User Schema	56
5.3.7.2 Course Schema	57
5.3.7.3 Lesson Schema	58
5.4 Features	59
Chapter 6 Testing	60
6.1 Testing Approach	60
6.1.1 Functional Testing	60
6.1.2 Nonfunctional Testing	60
6.2 Testing Tools	62
Chapter 7 Conclusion and Future Work	63
Appendix A	
Document Changes	64
References	65

Table of Figures

FIGURE 1: GANTT CHART	15
FIGURE 2: SURVEY IN ARABIC	20
FIGURE 3: INSTRUCTOR USE CASE DIAGRAM	28
FIGURE 4: USER USE CASE DIAGRAM	29
FIGURE 5: CLASS DIAGRAM	30
FIGURE 6: DEPLOYMENT DIAGRAM	31
FIGURE 7: USER ACTIVITY DIAGRAM	32
FIGURE 8: INSTRUCTOR ACTIVITY DIAGRAM	33
FIGURE 9:USER SEQUENCE DIAGRAM	34
FIGURE 10: INSTRUCTOR SEQUENCE DIAGRAM	35
FIGURE 11: USER STATE TRANSITION DIAGRAM	36
FIGURE 12: INSTRUCTOR STATE TRANSITION DIAGRAM	37
FIGURE 13: ER DIAGRAM	38
FIGURE 14: DB SCHEMA	39
FIGURE 15: HOME SCREEN	41
FIGURE 16: CATEGORIES LIST	41
FIGURE 17: SIGN UP	42
FIGURE 18: LOG IN	42
FIGURE 19: MY COURSES	43
FIGURE 20: CURRENT COURSE	43

Table of Tables

TABLE 1: COMPARISON BETWEEN EDULAB AND OTHER E-LEARNING PLATFORMS	11
TABLE 2: PROJECT OUTLINE	12
TABLE 3: PROJECT DELIVERABLES	12
TABLE 4: PROJECT TASKS	14
TABLE 5: ROLES AND RESPONSIBILITIES	19
TABLE 6: RISK IDENTIFICATION	20
TABLE 7: RISK RESPONSE	20
TABLE 8: STAKEHOLDERS	21
TABLE 9: CLIENT SIDE	22
TABLE 10: SERVER SIDE	22
TABLE 11: STUDENTS AND INSTRUCTORS FUNCTIONAL REQUIREMENTS	23
TABLE 12: NON-FUNCTIONAL REQUIREMENTS	26

Chapter 1

Introduction

1.1 Overview

EduLab E-learning Platform is a set of paid courses in different fields that serve the purpose of increasing client skills and make them more proficient in the work environment. What makes EduLab unique is that all the courses are offered in Arabic language which allow better understanding for clients in the MENA region.

The platform guides e-learners through the day through seamless lessons, points them to the resources they need to complete their work, and enables them to collaborate with other e-learners in the platform.

The courses are connected to a database which collects the information of students' information, instructors and videos uploaded.

1.2 Problem Statement

The notion behind EduLab is to provide online courses provided by Arabic-speaking instructors. There are many E-learning platforms out there with a variety of courses to learn from, yet the Arabic-speaking part is missing which shows limitation in fully understanding the online courses.

EduLab is a web application that is flexible with all browsers that support JS. The clients and instructors' data are all connected to a database along with the videos uploaded by the instructors.

Our target audience will include the young demographic (college students), self-employed (free lancers), employees and job owners. All listed before will benefit from the online courses we offer at EduLab since it will prepare them for work challenges. Client can create an account (as a new user) then the usage of the platform will be divided into two sections, purchased and unpurchased. With clients who already purchased courses they can go to my courses page and surf or start the course they purchased. For unpurchased courses clients can surf the platform through many options, such as (reviews on instructors, courses ratings or ask the community).

All clients from the MENA region will benefit from our online courses since they are given by Arabic-speaking instructors, moreover, they will increase their skills and confidence to challenge all obstacles coming their way in the future.

1.3 Related Work

Udemy:

Udemy is an e-learning platform that offers a variety of online courses from different fields. This online-courses provider allows their clients to follow up with each chapter of the course through modules that includes taking notes, videos, projects, and assignments.

The difference between Udemy and EduLab besides the main obvious feature which is we offer courses spoken in Arabic and Udemy do not, we also offer certified certificates at the end of the course.

Coursera:

Coursera is a web application that offers online courses from higher, well-known institution. The online courses include presentations, lectures, videos, communicating with other students and finishing tasks.

The difference between Coursera and EduLab is the fact that many talented instructors will benefit from our platform when upload their videos and materials, on the other hand Coursera best selling point is to provide online courses from reputable institutions.

Skillshare:

Skillshare is another online learning platforms that offers less variety of courses than Udemy. The platform fixates on subjects such as, design, freelancing, photography and illustration.

The difference between Skillshare and EduLab lies in the fact that Skillshare is subscription-based platform while EduLab you pay for the course you are interested in. More over the variety of courses we offer exceeds the ones included in Skillshare.

	EduLab	Udemy	Coursera	Skillshare
Variety in courses	✓	✓	✓	
Arabic-speaking instructors	✓			
Professional certificates	✓		✓	
Pay per course	✓	✓	✓	

Table 1: Comparison between EduLab and other e-learning platforms

1.4 Document Outline

Chapter Title	Description
Chapter 1: Introduction	Description of project goals, how it works, Problems out there and how it is resolved through our system, and comparison between competitors
Chapter 2: Project Plan	Detailed description of the development of the project. Including project tasks and timeline. Design implementation, risk assessment and roles and responsibilities
Chapter 3: Requirements Specification	A brief description of stakeholders, functional and non-functional needs, and platform requirements
Chapter 4: System Design	Includes low-level and high-level designs using UML diagrams, schema and showing the project user interface.

Table 2: Project Outline

Chapter 2 Project Plan

2.1 Project Deliverables

Name	Description
Source Code	The source code for the web application will be available.
Design Diagrams	Included in this section: diagrams show insights of how the structure and components of the web application

	work.
Databases	The database includes users data (instructors' information and students' information), and all possible data about the courses.
Documentation Files	Web application's source code will be available.
Progress Report	This document will contain a detailed timeline table that shows the work done each week.
User Interface Design	In this project, a list of creative designs for the web application. Some photos are provided in the documentation to show the user interface

Table 3: Project Deliverables

2.2 Project Tasks

Research Phase		
Task ID	Task Name	Description
T1	Define Project Idea	Identifying the problem and coming up with the perfect solution within the platform to serve the target audience in the MENA region.
T2	Kick-off Meeting	A meeting was held with my supervisor to discuss the project, how to proceed and the end results.

T3	The existence of the idea	Research and development of the idea is executed to check competitors projects and what needs to be improved from their existing projects.
T4	Investigating the tools	Mapping and listing the development tools that needed for the project
T5	Research summary	Documentation that states all the details of the project for best outcomes
T6	Scalability	Understanding the system's limitations and ways of how we can improve it in the future
Design Phase		
D1	UML Design	Creating diagrams that shows the functionality of the system
D2	User Interface Design	Designing an eye-catching UI to see how it looks from users' perspectives
D3	Database design	Creating Database designs used in the web application.
D4	Design Revision	A brief discussion with the supervisor to revise the progress and see if any modifications needed

Table 4: Project Tasks

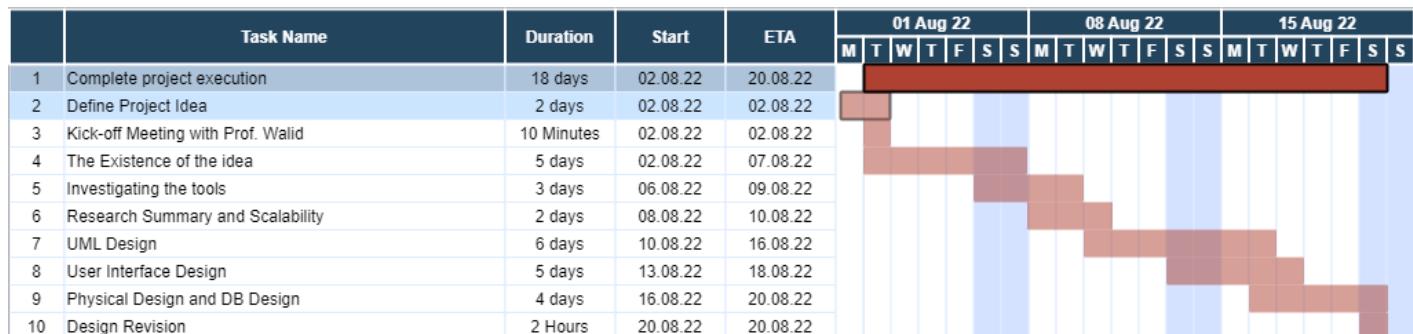
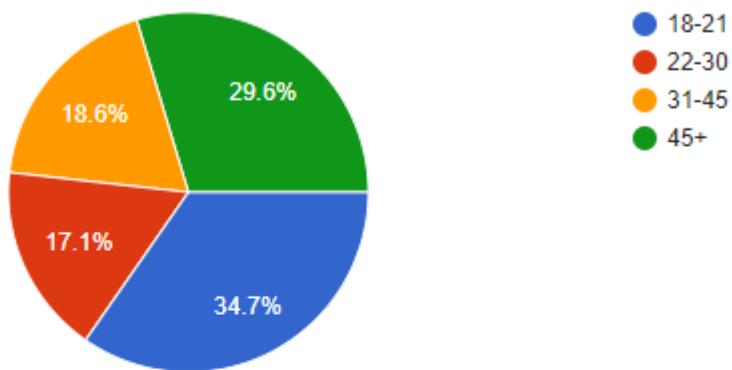


Figure 1: Gantt Chart

2.2 Survey Results in Arabic

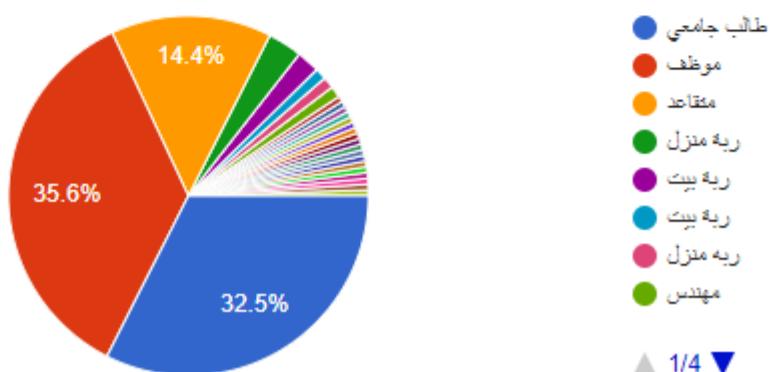
كم عمرك؟

199 responses



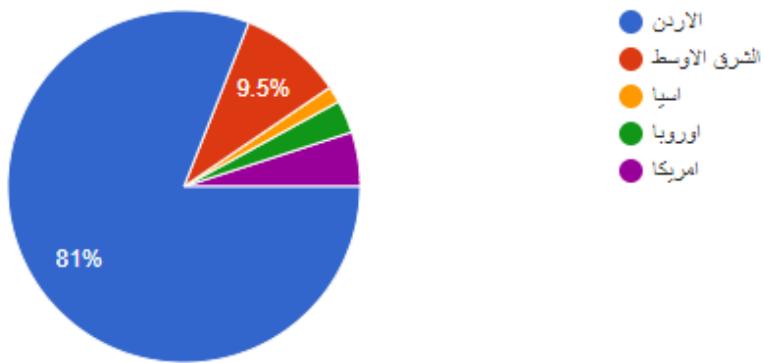
ما هي مهنةك؟

194 responses



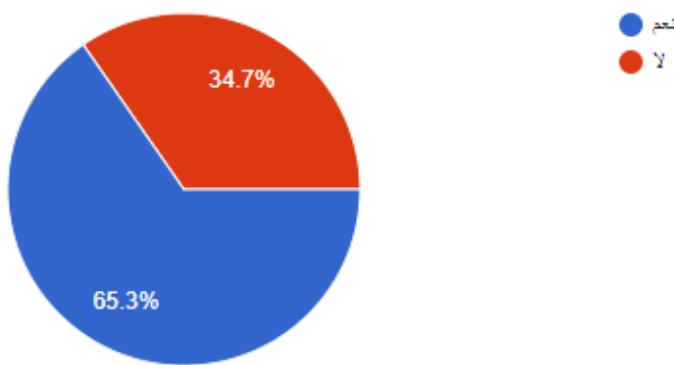
في اي دولة جامعتك، التي تدرس بها حالياً او درست فيها سابقاً؟

200 responses



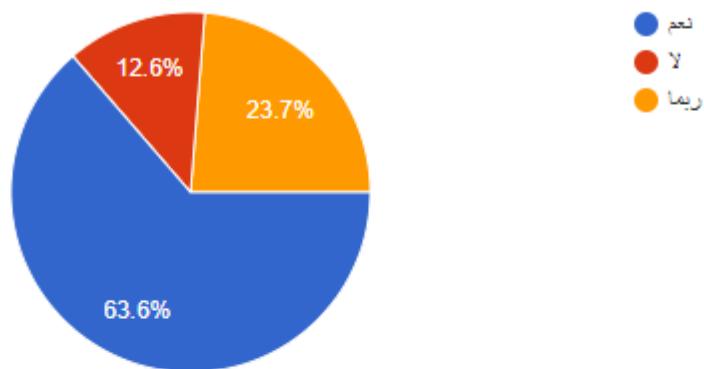
هل أخذت أي نوع من الدورات عبر الإنترنيت من قبل؟

199 responses



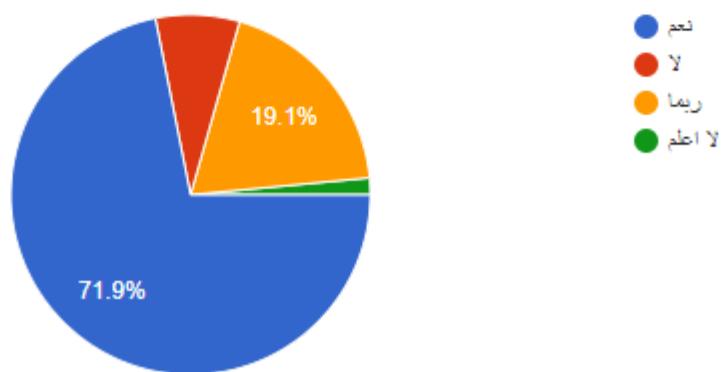
هل تعتقد أن التدريس عبر الإنترن트 هو المستقبل؟

198 responses



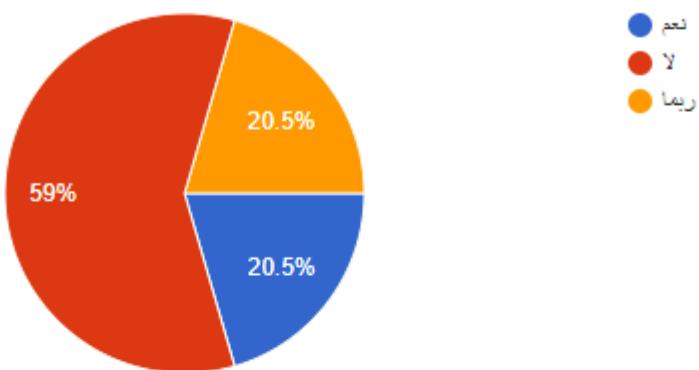
هل تعتقد ان الدورات عبر الإنترنوت تزيد من مهاراتك؟

199 responses



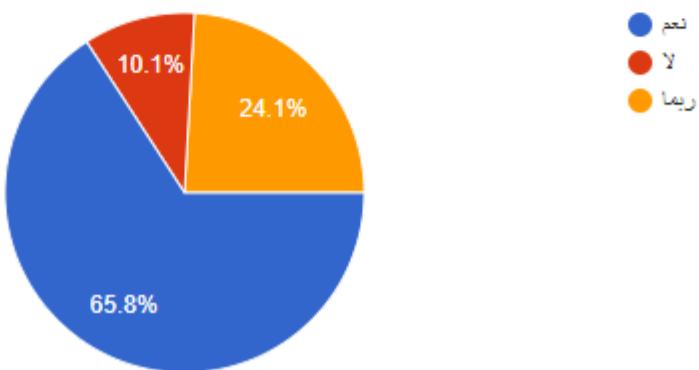
هل تعتقد أن هناك دورات تدريبية كافية على الإنترنط باللغة العربية؟

200 responses



هل تعتقد أن الدورات التدريبية عبر الإنترنط تجعل سيرتك الذاتية أكثر قوه؟

199 responses



هل أنت مهتم بشراء دورات على الإنترنت، يتم التحدث بها باللغة العربية؟

199 responses

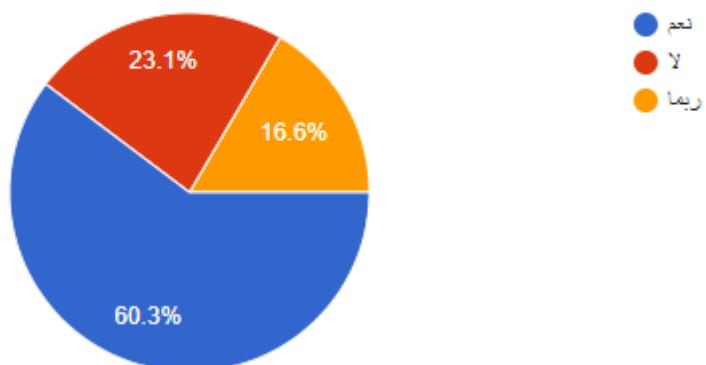


Figure 2: Survey in Arabic

2.3 Roles and Responsibilities

Member's Name	Responsibilities
Ahmad Mashal	Back-End, Front-End, User Interface, Database, Documentation

Table 5: Roles and Responsibilities

2.4 Risk Assessment

2.4.1 Risk Identification

Risk ID	Task ID	Risk description	Possibility
RS1	T1	Fake Instructor Accounts	Medium
RS2	T2	Server Crash	Low
RS3	T3	Payment not going through	Medium

Table 6: Risk Identification

2.4.2 Risk Response

Risk ID	Task ID	Task	Response
RS1	T1	Report	A report button, when pressed a popup will appear letting the reporter to clearly state why he is reporting, where the admin account can investigate and then decide to ban the Instructor or not
RS2	T2	Send notification	Send a notification to the users to assure them that the server is getting fixed.
RS3	T3	Report	Everything will be handled through the Stripe gateway, where there are a policy protecting every side

Table 7: Risk Response

2.5 Cost Estimation

1. Deployment cost: server hosting (API), server capacity= 150 USD/yearly
2. Maintenance fees: 300 USD/monthly
3. Cloud Database: 50 USD/ monthly

2.6 Project Management Tools

Draw.io:	To draw the diagrams.
Adobe Illustrator:	To create the logo and the UI concept
Microsoft office:	To write the documentation.
Visual-paradigm:	To draw the diagrams
Github.com:	For research purposes

Chapter 3 Requirements Specification

3.1 Stakeholders

ID	Stakeholders	Description
S1	Students	Users (college students, employees, freelancers...etc.) who create an account in the system and look up and pay for course of their interest.
S2	Instructors	Arabic-speaking instructors that have their own profile in the system, ability to upload their materials and videos and can be ranked and reviewed by Students
S3	Admin	Managing the web application, verifying the information of the instructors and students and make sure they are valid. Moreover, the ability to monitor the online courses.

Table 8: Stakeholders

3.2 Platform Requirements

Client Side:

Requirement ID	Description	Priority
PRC1	Smart Phone / Personal Computer	Essential
PRC2	Internet Connection Access	Essential
PRC3	Access to browser	Essential

Table 9: Client Side

Server Side:

Requirement ID	Description	Priority
PRS1	Internet Connection Access	Essential
PRS2	Server Hosting	Essential
PRS3	Database Server	Essential

--	--	--

Table 10: Server Side

3.3 Functional Requirements

- Students and Instructor Functional Requirements

ID	Requirement	Priority
FR1-MA	Sign Up	Essential
FR2-MA	Log In	Essential
FR3-MA	Forget Password	Essential
FR4-MA	View Profile	Essential
FR5-MA	Edit Profile	Essential
FR6-MA	Adding a new course	Recommended
FR7-MA	Editing a Course	Recommended
FR8-MA	View Settings	Essential
FR9-MA	Edit Sign-in Information	Essential
FR10-MA	Search Bar	Essential
FR11-MA	View Notification Page	Essential
FR12-MA	View Home Page	Essential

FR13-MA	Write review area	Essential
FR14-MA	Buying a course	Essential
FR15-MA	Sign Out	Essential

Table 11: Students and Instructors Functional Requirements

- Sign Up [FR1-MA]:

This is where the user register in the system and all his data will be stored in the database. The user has to input his full name, mobile number, email gender, and a strong password. The password must be between 8-15 characters and also should contain low and high case letters, numbers and symbols.

- log In [FR2-MA]:

This is how the user can enter the system. all the user has to do is to enter his/her email address or username and password.

- Forget Password [FR3-MA]:

When the user forgets his password, he can reset his password through his/her email.

- View Profile [FR4-MA]:

This button allows the user to view his profile page with all his/her information.

- Edit Profile [FR5-MA]:

It allows the user to edit his/her information. The profile consists of the user's information.

- Adding a New Course [FR6-MA]:

If an Instructor wants to upload his/her own course for a specific category, they will find all tools needed for them to do that.

- Editing a course [FR7-MA]:

If the Instructor wants to edit or update their already uploaded course, they can access all information and tools to do so.

- View Settings [FR8-MA]:

The user can view all the settings by pressing the settings button which will take the user to the settings page.

- Edit Sign-in Information [FR9-MA]:

It's part of the settings that allows the user to edit their signing in information, such as changing the email address or password.

- Search bar [FR10-MA]:

The search bar will allow the user to search the names of the courses that are registered in the system, by name, category , or instructors name.

- View Notification Page [FR11-MA]:

This allows the user to view a page that contains all their notifications, which consist of updates from instructors or new deals on similar courses.

- View Home page [FR12-MA]:

It allows the user to view home page where he can then access al pages in the website.

- View review page [FR13-MA]:

This will allow the user to review the course after completely finishing which will affect the rating of the course.

- Buying a course [FR14-MA]:

It is an area after choosing the course the user is pleased with, he will be redirected to a gateway where he/she can buy the course.

- Sign out [FR15-MA]:

There will be a button for the user to Sign out after being done with everything they want.

3.4 Non-Functional Requirements

ID	Requirement	Description
NFR1	Security	Log in: Asks the user to provide an email, and password which will be encrypted and saved in the database; the password has to

		be 8 characters, including at least one capital letter, numbers and symbols. If the password is forgotten, the system will send an email asking the user to create a new password. Secure data: the data is encrypted to avoid any loss or being leaked.
NFR2	Accessibility	The UI for this web application is simple to use, user who are surfing the web will have an easy access to the courses they are interested in.
NFR3	Data Integrity	The web application makes sure the information inserted by students and instructors is accurate.
NFR4	Capacity	Our web application's database is very large in capacity, as a result the system will be able to contain a large number of users, and uploaded materials
NFR5	Reliability	Any updates in the users' data the system will be updated immediately. Users will not face any crash or loss of information.

Table 12: Non-Functional Requirements

3.5 Other Requirements

Include here any requirements that may not directly fall under any of the sections before. Examples include restrictions on which APIs can be used, data transmission protocols, data storage formats, etc.

Chapter 4

System Design

4.1 Logical Model Design

4.1.1 Use Case Diagrams

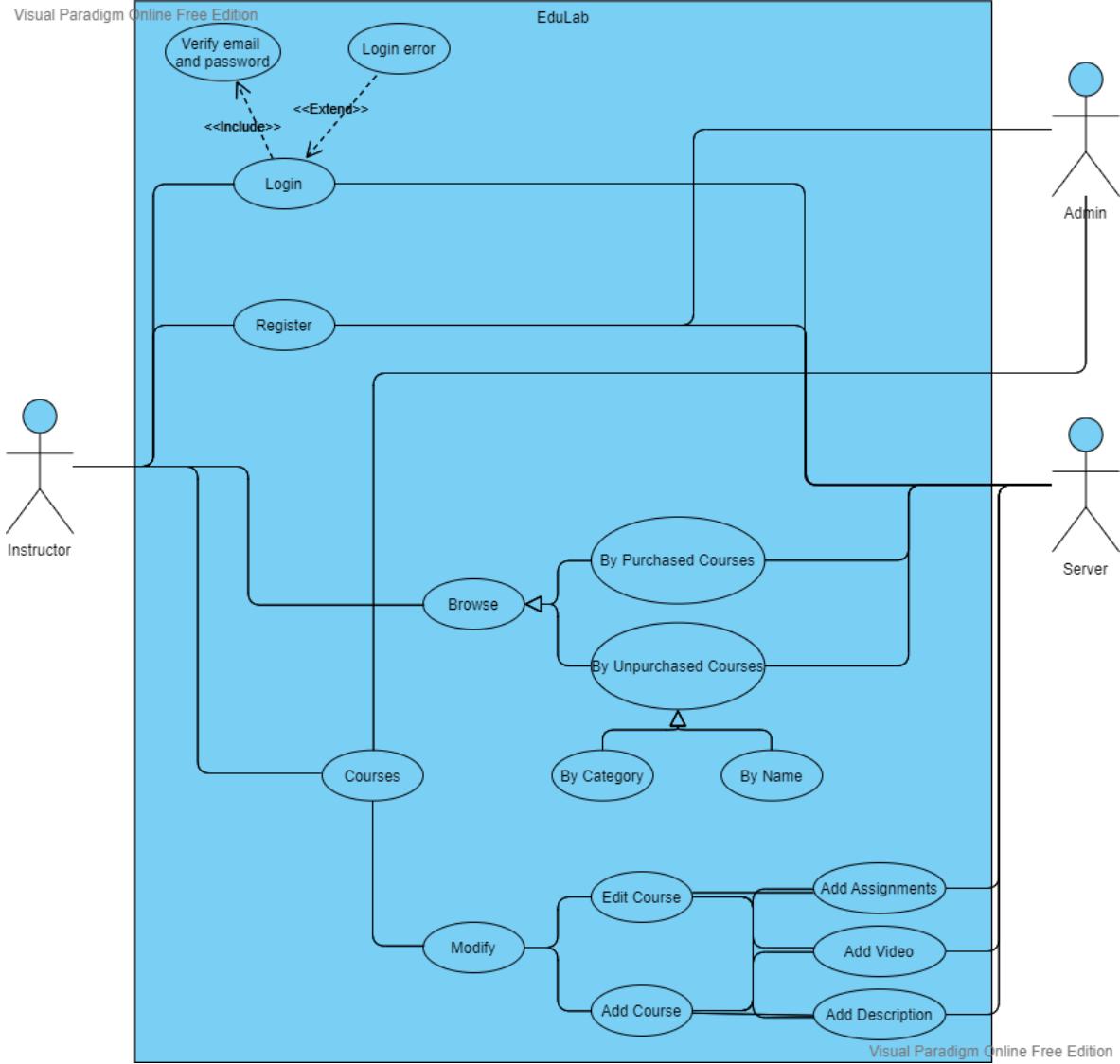


Figure 3: Instructor Use Case Diagram

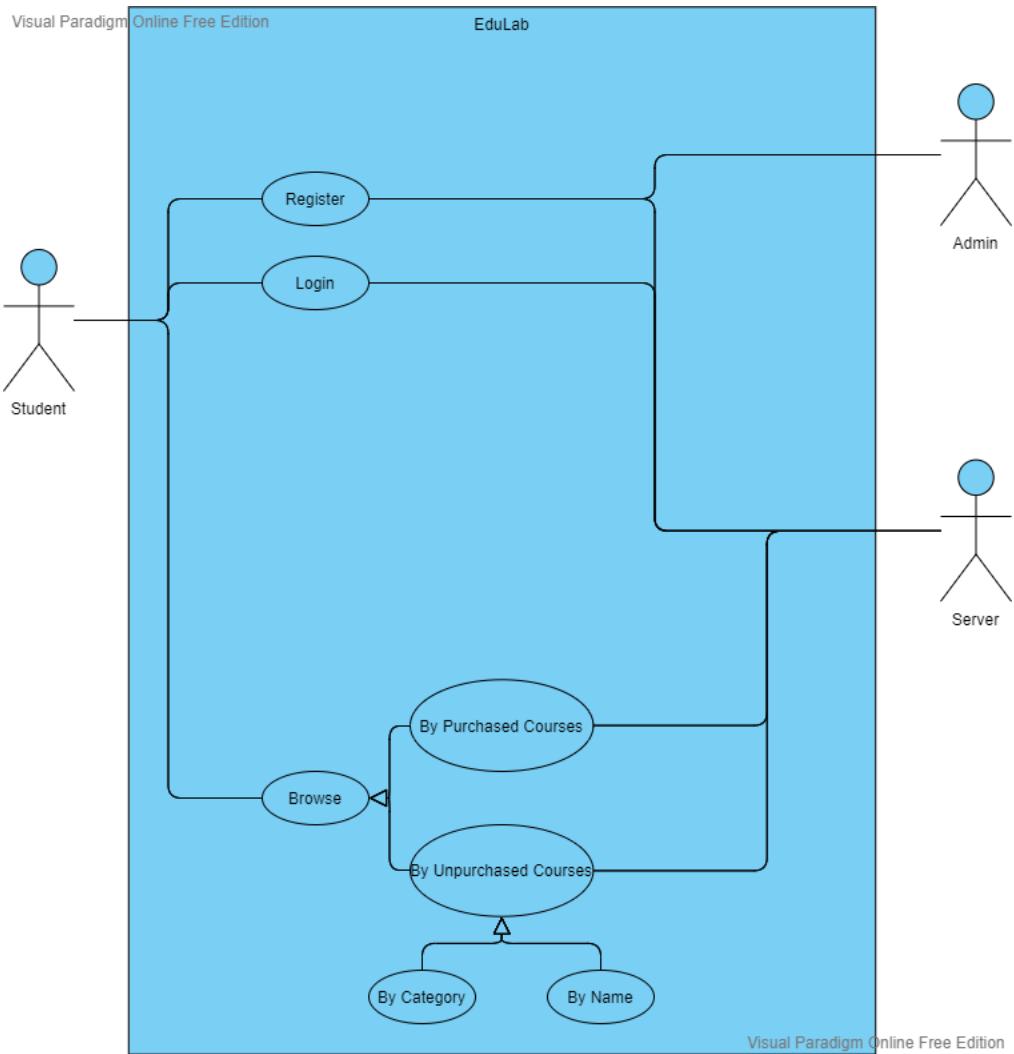


Figure 4: User Use Case Diagram

4.1.2 Class Diagram

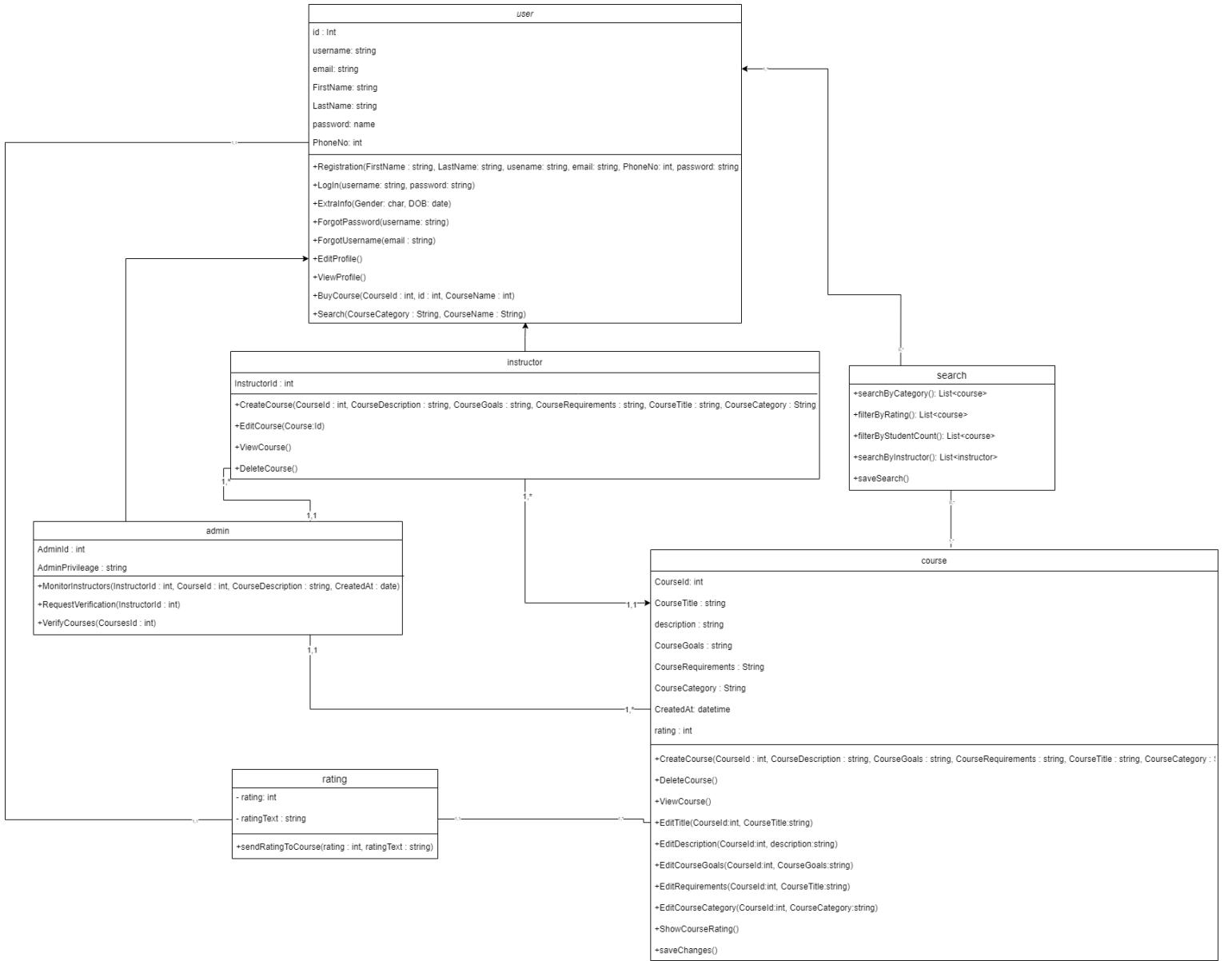


Figure 5: Class Diagram

4.1.3 Component Diagram

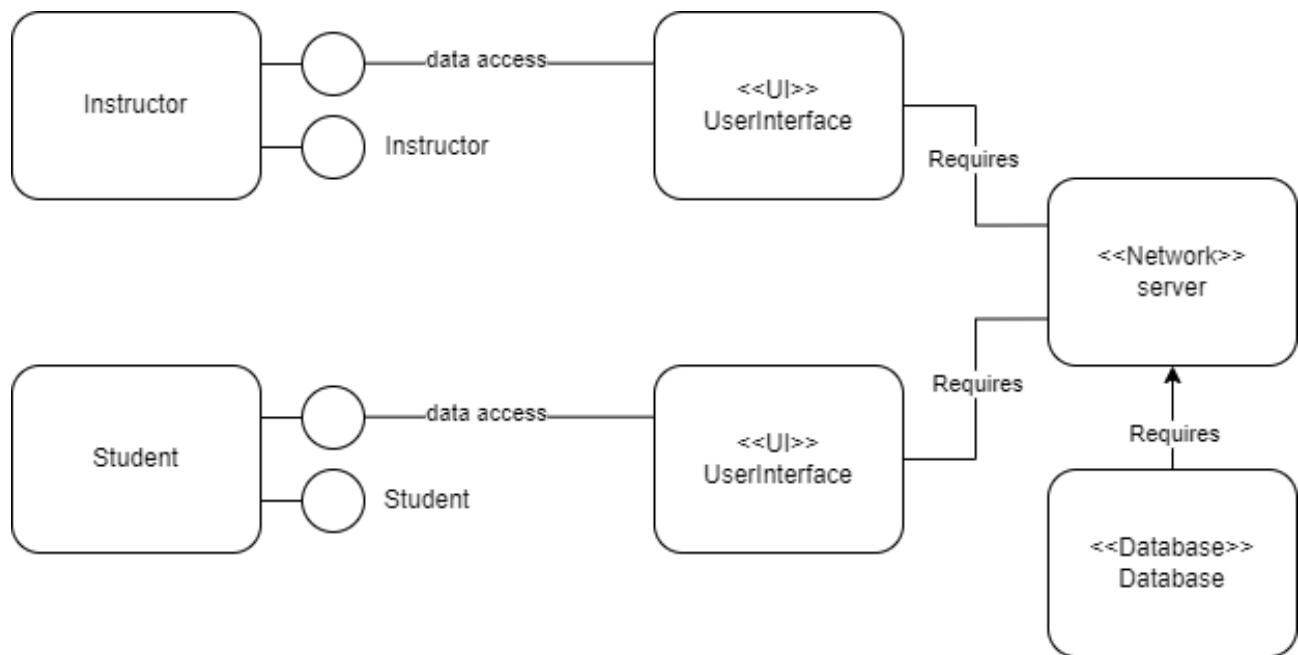


Figure 5: Component Diagram

4.1.4 Deployment Diagram

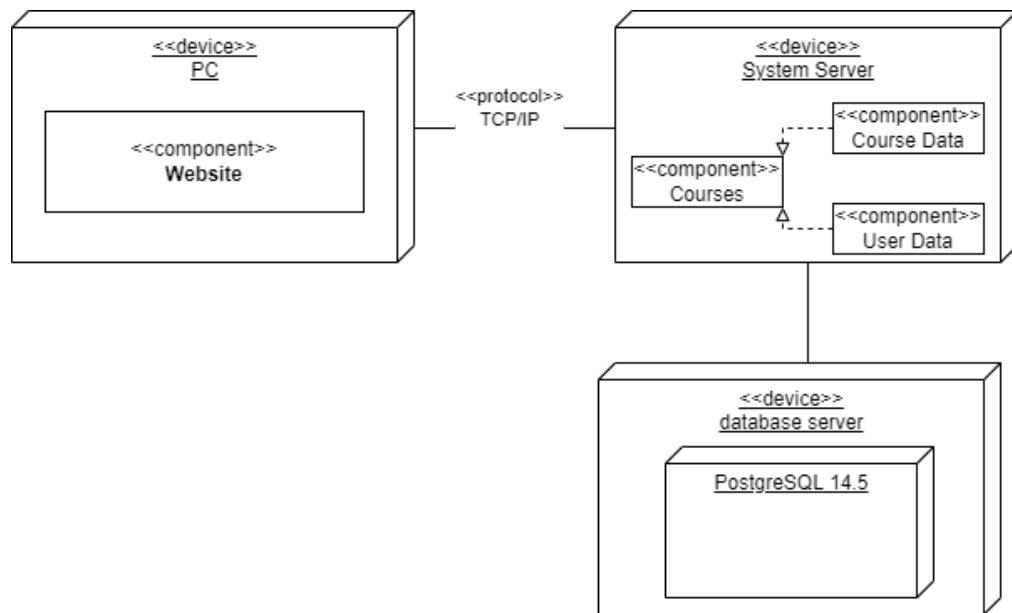


Figure 6: Deployment Diagram

4.1.5 Activity Diagram

4.1.5.1 User Activity Diagram

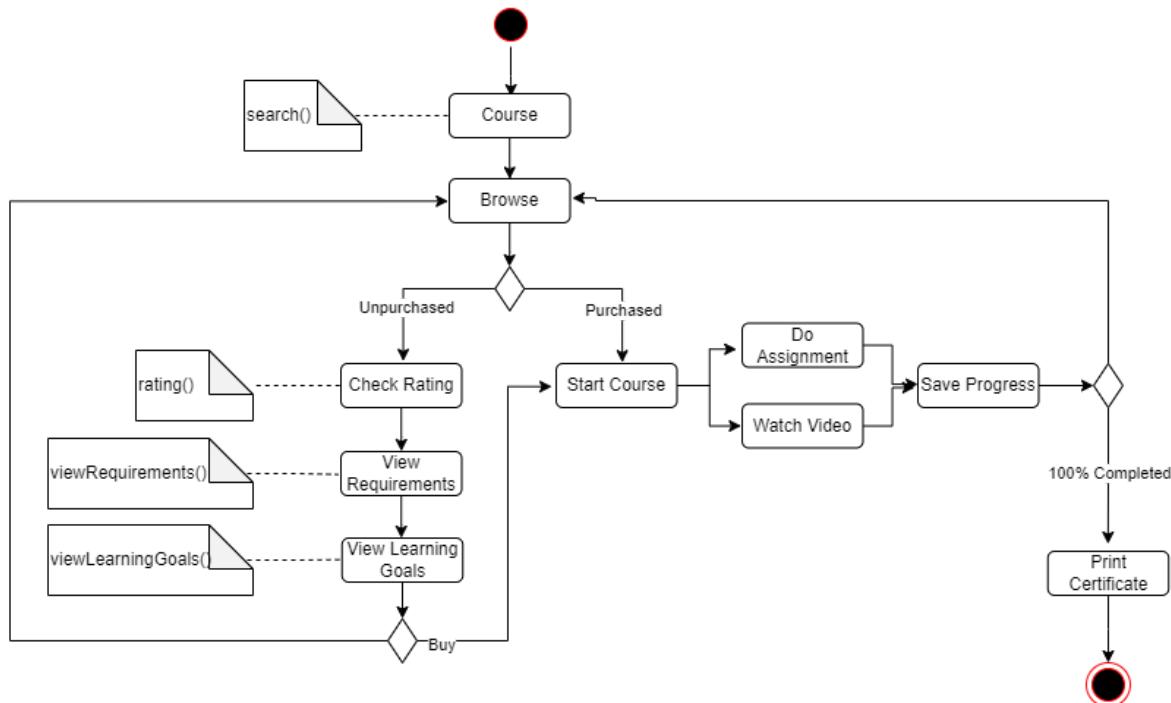


Figure 7: User Activity Diagram

4.1.5.2 Instructor Activity Diagram

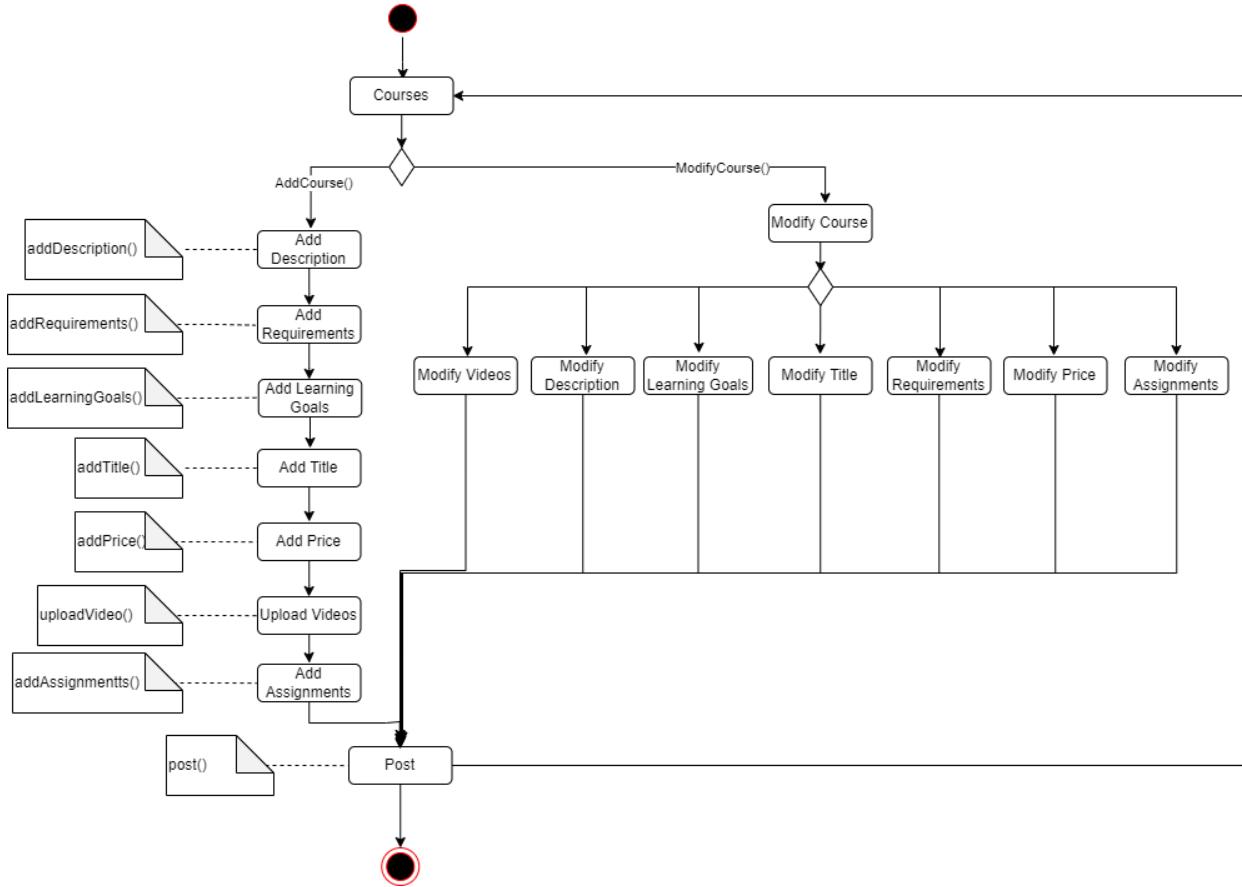


Figure 8: Instructor Activity Diagram

4.1.6 Sequence Diagram

4.1.6.1 User Sequence Diagram

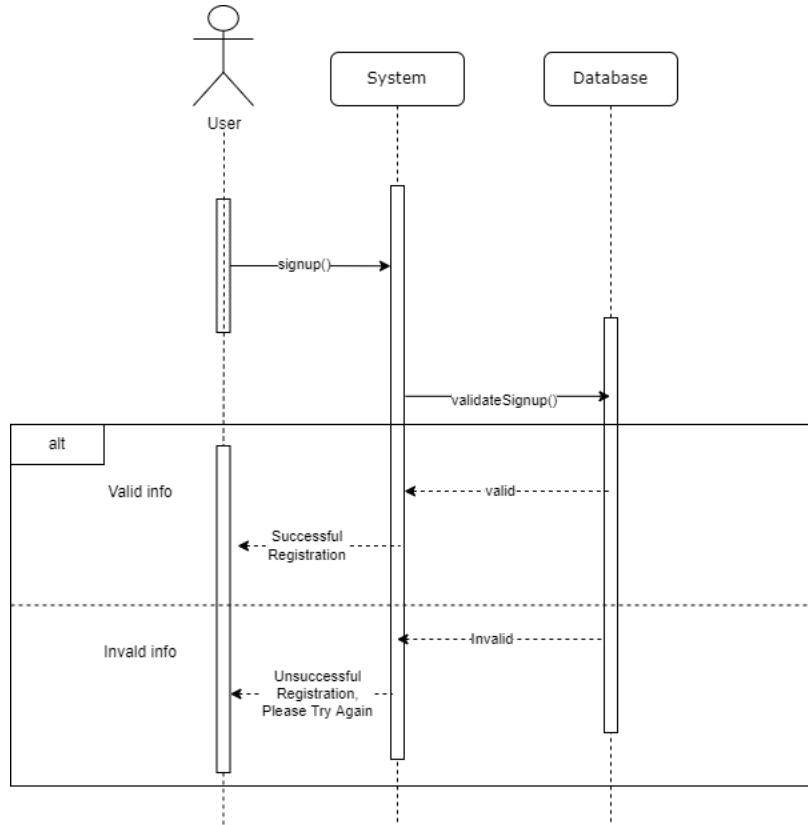


Figure 9:User Sequence Diagram

4.1.6.2 Instructor Sequence Diagram

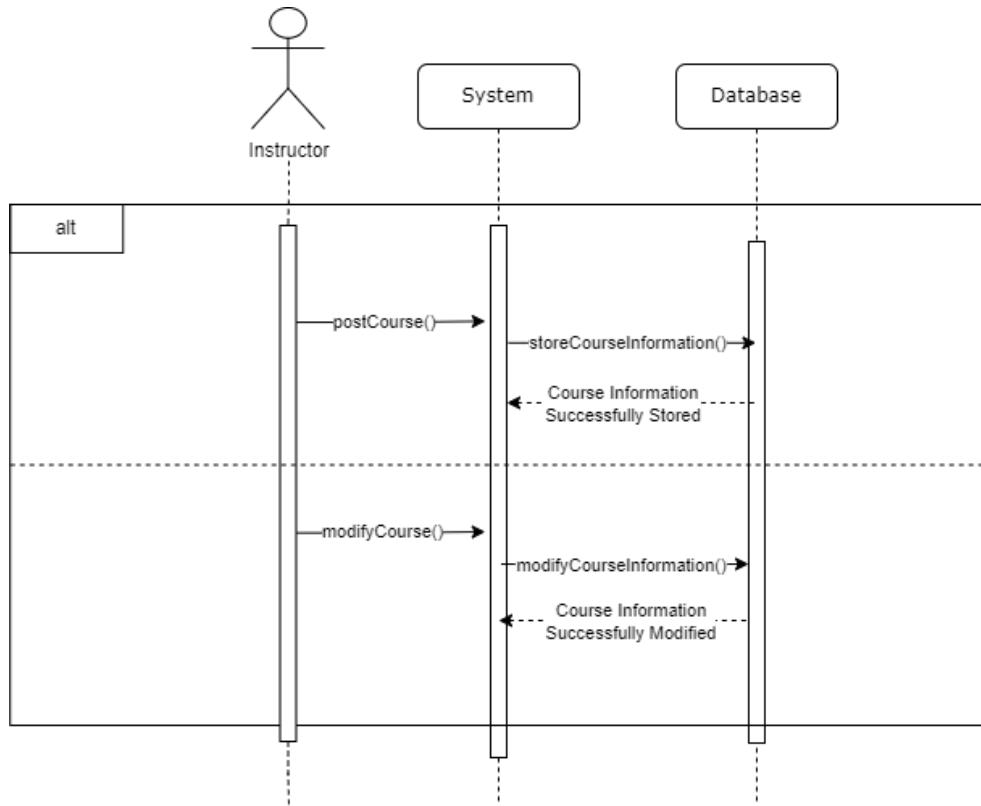


Figure 10: Instructor Sequence Diagram

4.1.7 State Transition Diagram

4.1.7.1 User State Transition Diagram

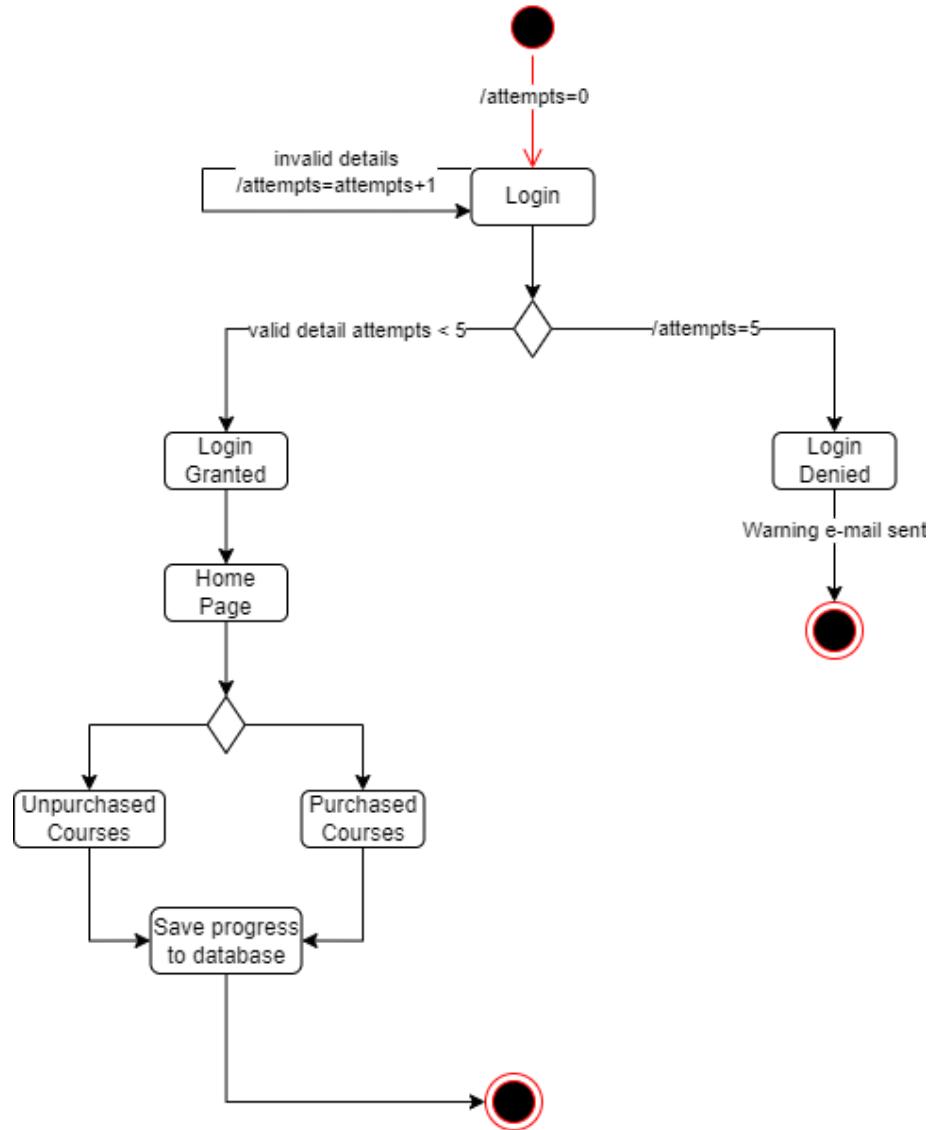


Figure 11: User State Transition Diagram

4.1.7.2 Instructor State Transition Diagram

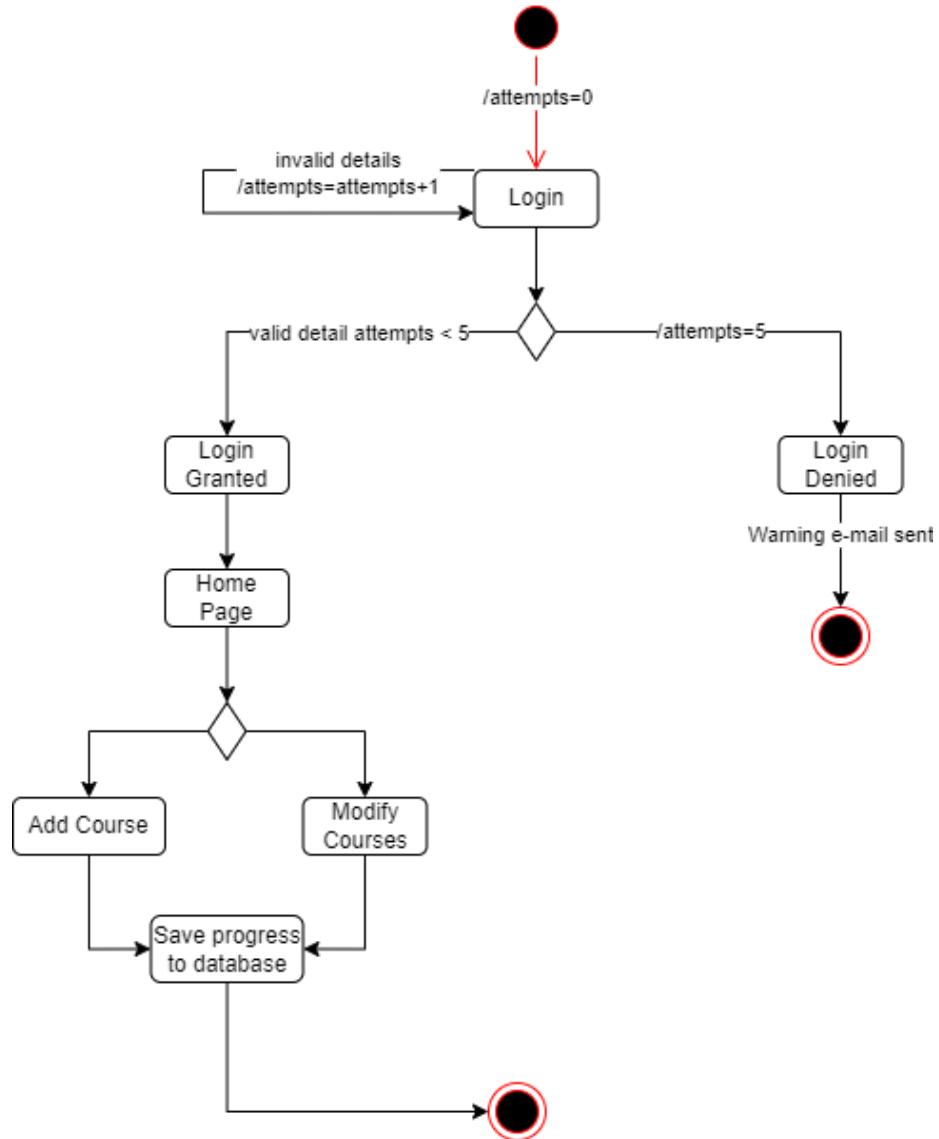


Figure 12: Instructor State Transition Diagram

4.1.8 ER Diagram

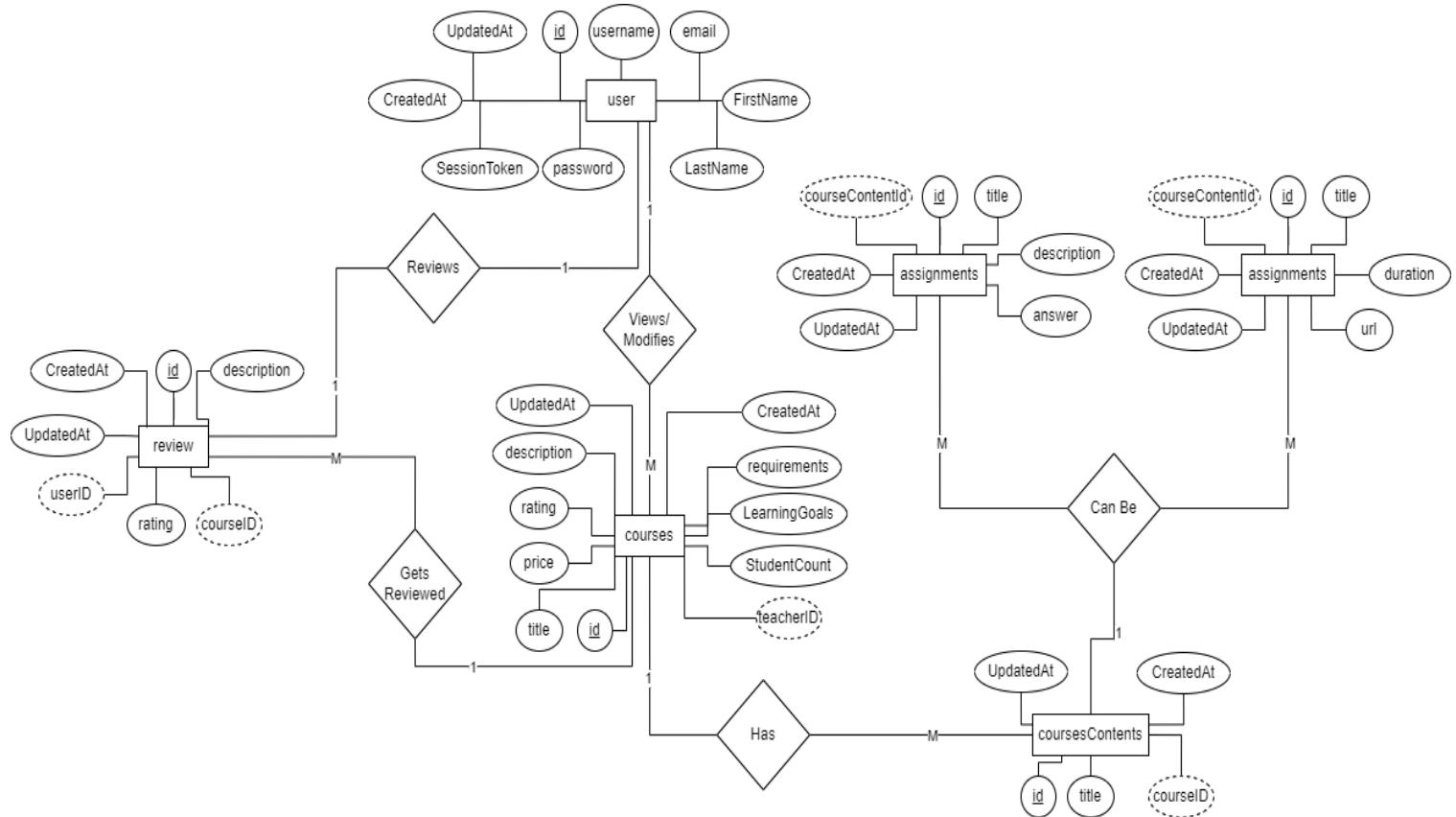


Figure 13: ER Diagram

4.2 Physical Model Design

4.2.1 DB Schema

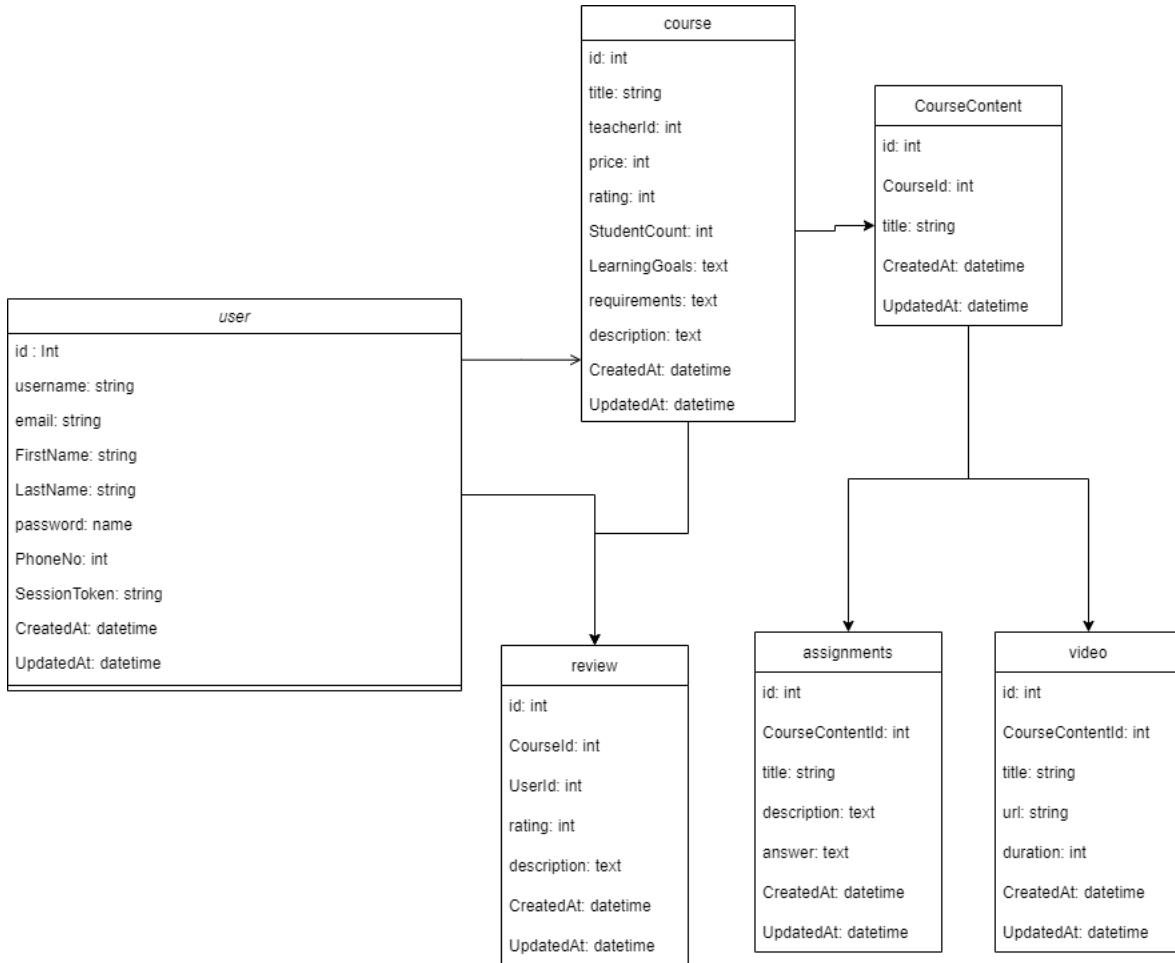


Figure 14: DB Schema

4.2.2 Report Design

User report design:

Id	Username	Email	FirstName	LastName	Password	PhoneNo
SessionToken	CreatedAt	UpdatedAt				

Course report design:

Id	Title	teacherId	Price	Rating	StudentCount	LearningGoals
Requirement	Description	CreatedAt	UpdatedAt			

Course Content report design:

Id	CourseId	Title	CreatedAt	UpdatedAt

Review report design:

Id	CourseId	UserId	Rating	Description	CreatedAt	UpdatedAt

Assignments report design:

Id	CourseContentId	Title	Description	Answer	CreatedAt	UpdatedAt

Video report design:

Id	CourseContentId	Title	url	Duration	CreatedAt	UpdatedAt

4.2.3 User Interface



Figure 15: Home Screen



Figure 16: Categories List

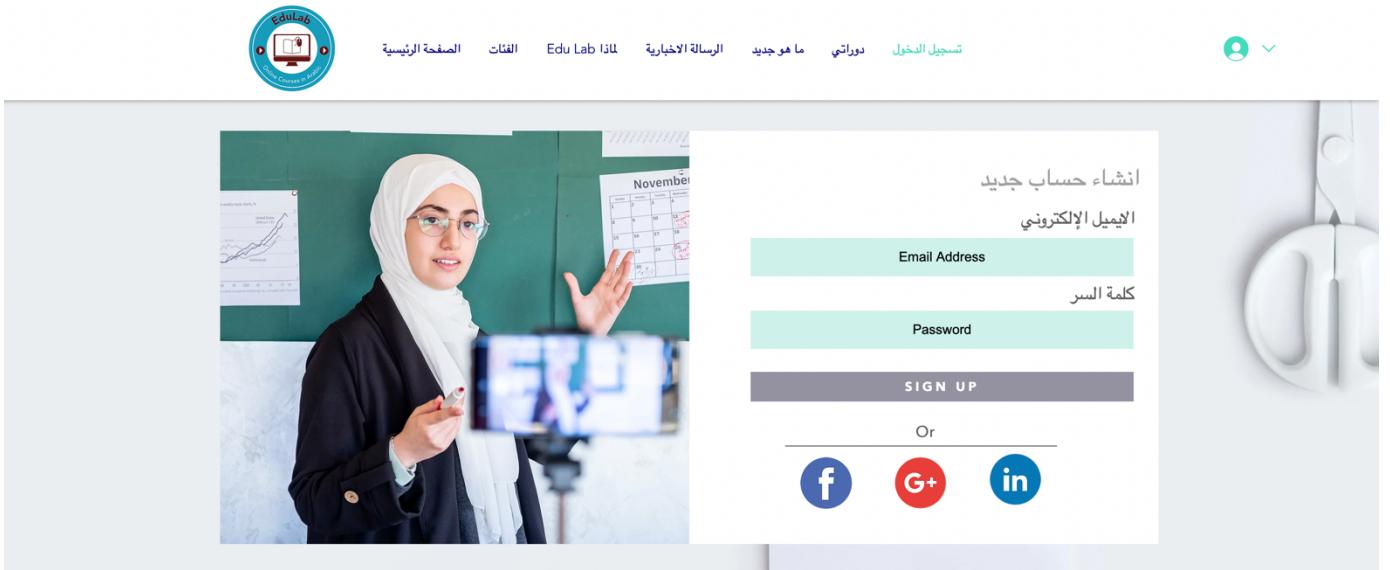


Figure 17: Sign up

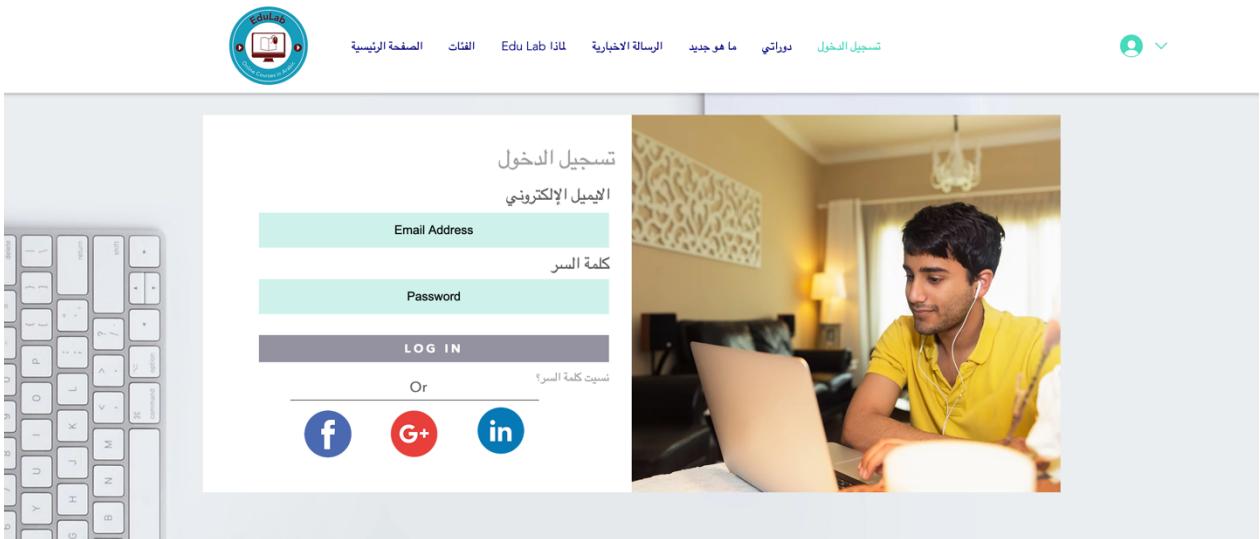


Figure 18: Log in

The screenshot shows the EduLab platform's user interface. At the top, there is a navigation bar with links: الصفحة الرئيسية (Home), الفئات (Categories), Edu Lab, ملانا (Mylana), الرسالة الاخبارية (News), ما هو جديد (What's New), دوراتي (My Courses), تسجيل الدخول (Log In), and a user profile icon.

الدورات المسجلة

Three course cards are listed:

- PRINCIPLES OF INFORMATION TECHNOLOGY 1B**
Working with Computers
Career Ready
- Digital Marketing**
- ACCOUNTING I**

Each card includes a 'تفاصيل الدورة' (Course Details) link and a 'انقر الرابط' (Click the link) button. To the right of the cards, the course details are summarized:

- دورة تكنولوجيا المعلومات June 2021
- دورة التسويق الإلكتروني September 2021
- دورة المحاسبة March 2022

Figure 19: My Courses

The screenshot shows the EduLab platform's user interface. At the top, there is a navigation bar with links: الصفحة الرئيسية (Home), الفئات (Categories), Edu Lab, ملانا (Mylana), الرسالة الاخبارية (News), ما هو جديد (What's New), دوراتي (My Courses), تسجيل الدخول (Log In), and a user profile icon.

دورة تكنولوجيا المعلومات

June 2021

The main content area displays the course details for 'PRINCIPLES OF INFORMATION TECHNOLOGY 1B'.

PRINCIPLES OF INFORMATION TECHNOLOGY 1B

Working with Computers

Career Ready

عن هذه الدورة. اكتب ملاحظاتك. اسئلة وأجوبة. عدد الساعات. عدد الدورات. المحاضر

To the right, a vertical list of course materials is shown:

- المحاضرة رقم ١ الواجبات PDF .اسئل مجتمعك
- المحاضرة رقم ٢ الواجبات PDF .اسئل مجتمعك
- المحاضرة رقم ٣ الواجبات PDF .اسئل مجتمعك
- المحاضرة رقم ٤ الواجبات PDF .اسئل مجتمعك
- المحاضرة رقم ٥ الواجبات PDF .اسئل مجتمعك
- المحاضرة رقم ٦ الواجبات PDF .اسئل مجتمعك
- المحاضرة رقم ٧ الواجبات PDF .اسئل مجتمعك

المزيد

Figure 20: Current Course

Chapter 5 Implementation

5.1 Overview

To ensure the success of the project, various programming languages were utilized in the development of EduLab to include all necessary features in the final product. These languages were chosen for their powerful capabilities.

Subsequently, we have primarily used Visual Studio Code as the main code editor because of its high efficiency in developing and debugging multiple projects. Furthermore, Visual Studio Code offers the capability to obtain the necessary packages and files for the project to function properly.

Additionally, the project necessitated the utilization of MongoDBCompass, where we utilized a non-relational database (NoSQL) rather than the previously mentioned relational database (SQL) for managing and saving data and the personal information of EduLab members. Furthermore, AWS was employed to efficiently upload and retrieve videos and photos.

5.1.1 Front-End

For the front-end, languages that were deemed most effective were used. Specifically, HTML, CSS, and JavaScript were employed to design the website.

5.1.2 Back-End

A significant amount of Node.JS was employed in the development of the project due to its high efficiency and the extensive libraries it offers. The decision to use NoSQL instead of SQL was made, and among the available options, Mongoose was found to be the most convenient and robust solution for our needs. Node.js being non-blocking and event-driven makes it very efficient when working with the network, Mongoose is one of the most popular library for MongoDB, it makes the interactions with MongoDB simple and elegant, it provides a straight-forward, schema-based solution to model your application data and includes built-in type casting, validation, query building, business logic hooks and more, in other words it provide a lot of functionalities and make the development process more easy and efficient.

5.2 Tools

5.2.1 Software

Visual Studio Code was extensively utilized during the development process as it offers the ability to work with multiple programming languages simultaneously, and the

capability to utilize different interpreters without the need to download separate software. The switch to NoSQL prompted the usage of the MongoDB and Mongoose library, with MongoDBCompass being the chosen software for managing the database, and Chrome being utilized for debugging the code.

5.2.2 Libraries

5.2.2.1 Front-End

The Table below represents a detailed description of the technologies and modules used in the project:

Library	Description
ant-design/icons	A library of icons for the Ant Design user interface design system
antd	A library of UI components for building user interfaces in React
bootstrap	A popular front-end development framework for building responsive web designs
express	A minimal and flexible Node.js web application framework
http-proxy-middleware	A middleware for proxying HTTP requests in a Node.js application
next	A framework for building server-rendered React applications
react	A JavaScript library for building user interfaces
react-dom	A package for React that allows it to be rendered on the web
react-image-file-resizer	A package for resizing image files in React applications
react-markdown	A package for rendering Markdown text in React applications

react-player	A package for playing video and audio in React applications
react-toastify	A package for displaying notifications in React applications
toastify	A package for displaying notifications in JavaScript applications

5.2.2.1 Back-End

Library	Description
aws-sdk	The official AWS SDK for JavaScript, which allows you to interact with AWS services such as S3 and DynamoDB.
bcrypt	A library for hashing and comparing passwords, it uses the Blowfish encryption algorithm.
cookie-parser	A middleware for parsing cookie headers and populating req.cookies.
csurf	A middleware for providing CSRF protection.
dotenv	A zero-dependency module that loads environment variables from a .env file.
esm	A ECMAScript module loader that enables the use of ECMAScript modules in Node.js.
express	A minimal and flexible Node.js web application framework
express-formidable	A middleware for handling multipart/form-data, which is used for uploading files.
express-jwt	A middleware for validating JWT tokens in Express.
jsonwebtoken	A library for creating and decoding JSON Web Tokens.

mongoose	A MongoDB object modeling tool designed to work in an asynchronous environment.
morgan	A middleware for logging incoming requests.
nanoid	A small, secure, and URL-friendly unique string generator.
nodemon	A utility that will monitor for any changes in your source and automatically restart your server.
query-string	A library for parsing and stringifying URL query strings.
slugify	A library for converting text to a URL-friendly format.
stripe	A library for integrating with the Stripe API, which allows you to process payments.

5.3 Main Functionalities

5.3.1 Sign Up Page

```
import {useState, useEffect, useContext} from 'react';
import axios from 'axios';
import { toast } from 'react-toastify';
import {SyncOutlined} from '@ant-design/icons';
import Link from 'next/link';
import {Context} from '../context';
import {useRouter} from 'next/router';
import user from '../../server/models/user';

// const User = require('../../server/models/users');

const Register = () => {
  const [name, setName] = useState("");
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [loading, setLoading] = useState(false);

  const {
    state: {user},
  } = useContext(Context);
  //const {user} = state;

  const router = useRouter();

  useEffect(() => {
    if(user !== null) router.push("/");
  }, [user]);

  const handleSubmit = async (e) => {
    e.preventDefault();
    //console.table({name, email, password});
    try{
      setLoading(true);
      const {data} = await axios.post('/api/register', {
        name,
        email,
        password,
      });
      // console.log("REGISTER RESPONSE",data);
      toast.success('Registration Successful. Please Login');
      setName('');
      setEmail('');
      setPassword('');
      setLoading(false);
    } catch (err){
      toast.error(err.response.data);
      setLoading(false);
    }
  }
}

return(
  <>
    <h1 className = "jumbotron text-center bg-primary square">
  Register</h1>

    <div className = "container col-md-4 offset-md-4 pb-5">
      <form onSubmit = {handleSubmit}>
        <input
          type="text"
          className="form-control mb-4 p-4"
          value = {name}
          onChange={(e => setName(e.target.value))}
          placeholder = "Enter Name"
          required
        />

        <input
          type="email"
          className="form-control mb-4 p-4"
          value = {email}
          onChange={(e => setEmail(e.target.value))}
          placeholder = "Enter Email"
          required
        />

        <input
          type="password"
          className="form-control mb-4 p-4"
          value = {password}
          onChange={(e => setPassword(e.target.value))}
          placeholder = "Enter Password"
          required
        />

        <button type = "submit" className = "btn btn-block btn-primary"
          disabled =={!name || !email || !password || loading}>
          {(loading ? <SyncOutlined spin /> : "Submit")}
        </button>
      </form>
      <p className='text-center p-3'>
        Already registered?{' '}
        <Link href = "/login">
          Login
        </Link>
      </p>
    </div>
  );
};

export default Register;
```

5.3.2 Sign In Page

```
import {useState, useContext, useEffect} from 'react';
import axios from 'axios';
import { toast } from 'react-toastify';
import {SyncOutlined} from '@ant-design/icons';
import Link from 'next/link';
import {Context} from '../context';
import {useRouter} from 'next/router';

const Login = () => {
    const [email, setEmail] = useState("ahmad.mashal@outlook.com");
    const [password, setPassword] = useState("Password123");
    const [loading, setLoading] = useState(false);

    //state
    const {state, dispatch} = useContext(Context);
    const {user} = state;
    // console.log("STATE", state);

    const router = useRouter();

    useEffect(() => {
        if(user !== null) router.push("/");
    }, [user]);

    const handleSubmit = async (e) => {
        e.preventDefault();
        //console.table({name, email, password});
        try{
            setLoading(true);
            const {data} = await axios.post('/api/login', {
                email,
                password,
            });
            // console.log("LOGIN RESPONSE", data);
            dispatch({
                type: "LOGIN",
                payload: data
            });
            //save in local storage
            window.localStorage.setItem('user', JSON.stringify(data));

            //redirect to user dashboard
            router.push("/user");
            // toast.success('Login Successful!');
            // setLoading(false);
            // catch (err){
            //     toast.error(err.response.data);
            //     setLoading(false);
            // }
        }
    }

    return(
        <>
            <h1 className = "jumbotron text-center bg-primary square">
        <Login>/h1>

        <div className = "container col-md-4 offset-md-4 pb-5">
            <form onSubmit = {handleSubmit}>

                <input
                    type="email"
                    className="form-control mb-4 p-4"
                    value = {email}
                    onChange={(e => setEmail(e.target.value))}
                    placeholder = "Enter Email"
                    required
                />

                <input
                    type="password"
                    className="form-control mb-4 p-4"
                    value = {password}
                    onChange={(e => setPassword(e.target.value))}
                    placeholder = "Enter Password"
                    required
                />

                <button type = "submit" className =
                    btn btn-block btn-primary"
                    disabled =={!email || !password || loading}
                >
                    {loading ? <SyncOutlined spin /> : "Submit"}
                </button>
            </form>
        </div>
    );
};

export default Login;
```

5.3.3 Forgot Password

```
import { useState, useContext, useEffect } from "react";
import axios from "axios";
import { toast } from "react-toastify";
import { SyncOutlined } from "@ant-design/icons";
import Link from "next/link";
import { Context } from "../context/index";
import { useRouter } from "next/router";

const ForgotPassword = () => {
  // state
  const [email, setEmail] = useState("");
  const [success, setSuccess] = useState(false);
  const [code, setCode] = useState("");
  const [newPassword, setNewPassword] = useState("");
  const [loading, setLoading] = useState(false);

  // context
  const {
    state: { user },
  } = useContext(Context);
  // router
  const router = useRouter();

  // redirect if user is Logged in
  useEffect(() => {
    if (user !== null) router.push("/");
  }, [user]);

  const handleSubmit = async (e) => {
    e.preventDefault();
    setLoading(true);
    try {
      const { data } = await axios.post("/api/forgot-password", {
        email });
      setSuccess(true);
      toast("Check your email for the verification code");
      setLoading(false);
    } catch (err) {
      setLoading(false);
      toast(err.response.data);
    }
  };

  const handleResetPassword = async (e) => {
    e.preventDefault();
    setLoading(true);
    try {
      const { data } = await axios.post("/api/reset-password", {
        email, newPassword, code });
      setEmail('');
      setCode('');
      setNewPassword('');
      setLoading(false);
      toast("Password changed, Please log back in.");
    } catch (err) {
      setLoading(false);
      toast(err.response.data);
    }
  };
}

return (
  <>
    <h1 className="jumbotron text-center bg-primary square">
      Forgot Password
    </h1>

    <div className="container col-md-4 offset-md-4 pb-5">
      <form onSubmit={success ? handleResetPassword : handleSubmit}>
        <input
          type="email"
          className="form-control mb-4 p-4"
          value={email}
          onChange={(e) => setEmail(e.target.value)}
          placeholder="Enter email"
          required
        />
        {success && (
          <>
            <input
              type="text"
              className="form-control mb-4 p-4"
              value={code}
              onChange={(e) => setCode(e.target.value)}
              placeholder="Enter verification code"
              required
            />
            <input
              type="password"
              className="form-control mb-4 p-4"
              value={newPassword}
              onChange={(e) => setNewPassword(e.target.value)}
              placeholder="New Password"
              required
            />
          </>)
        <button
          type="submit"
          className="btn btn-primary btn-block p-2"
          disabled={loading || !email}
        >
          {loading ? <SyncOutlined spin /> : "Submit"}
        </button>
      </form>
    </div>
  </>
);
};

export default ForgotPassword;
```

5.3.4 Become Instructor Page

When a user desires to become an Instructor, they will access the "Become Instructor" page, where they will provide their information and will be directed to Stripe to connect their payment information, here is a snippet of the code..

```
import { useContext, useState } from "react";
import { Context } from "../../context";
import { Button } from "antd";
import axios from "axios";
import {
  SettingOutlined,
  UserSwitchOutlined,
  LoadingOutlined,
} from "@ant-design/icons";
import { toast } from "react-toastify";
import UserRoute from "../../components/routes/UserRoute";

const BecomeInstructor = () => {
  // state
  const [loading, setLoading] = useState(false);
  const {
    state: { user },
  } = useContext(Context);

  const becomeInstructor = () => {
    // console.log("become instructor");
    setLoading(true);
    axios
      .post("/api/make-instructor")
      .then((res) => {
        console.log(res);
        window.location.href = res.data;
      })
      .catch((err) => {
        console.log(err.response.status);
        toast("Stripe onboarding failed. Try again.");
        setLoading(false);
      });
  };

  return (
    <>
      <h1 className="jumbotron text-center square">Become Instructor</h1>

      <div className="container">
        <div className="row">
          <div className="col-md-6 offset-md-3 text-center">
            <div className="pt-4">
              <UserSwitchOutlined className="display-1 pb-3" />
              <br />
              <h2>Setup payout to publish courses on EduLab</h2>
              <p className="lead text-warning">
                EduLab partners with stripe to transfer earnings to your bank
                account
              </p>
              <Button
                className="mb-3"
                type="primary"
                block
                shape="round"
                icon={loading ? <LoadingOutlined /> : <SettingOutlined />}
                size="large"
                onClick={becomeInstructor}
                disabled={
                  (user && user.role && user.role.includes("Instructor")) ||
                  loading
                }
              >
                {loading ? "Processing..." : "Payout Setup"}
              </Button>
              <p className="lead">
                You will be redirected to stripe to complete onboarding process.
              </p>
            </div>
          </div>
        </div>
      </div>
    );
};

export default BecomeInstructor;
```

5.3.5 Home Page

After login or Sign Up the user is directed to the Home page, here is a snippet of its code:

```
import TopNav from '../components/TopNav';
import 'bootstrap/dist/css/bootstrap.min.css';
import 'antd/dist/antd.css';
import '../public/css/style.css';
import {ToastContainer} from 'react-toastify';
import 'react-toastify/dist/ReactToastify.css';
import {Provider} from '../context';

function MyApp({Component, pageProps}){
  return(
    <Provider>
      <ToastContainer position = "top-center" />
      <TopNav />
      <Component {...pageProps} />;
    </Provider>
  )
}

export default MyApp;
```

5.3.6 Connecting to the server

Connecting the Front-End with the Back-End

```
const express = require('express');
const next = require('next');
const {createProxyMiddleware} = require("http-proxy-middleware");

const dev = process.env.NODE_ENV !== 'production';
const app = next({dev});
const handle = app.getRequestHandler();

app
  .prepare()
  .then(() =>{
    const server = express();
    // server.use(express.json());

    //apply proxy in dev mode
    if(dev){
      server.use('/api', createProxyMiddleware({
        target: 'http://localhost:8000',
        changeOrigin: true,
      }));
    }

    server.all('*', (req,res) => {
      return handle(req, res);
    });

    server.listen(3000, err => {
      if(err) throw err;
      console.log(`> Ready on http://localhost:3000`);
    });
  })
  .catch((err) => {
    console.log("Error", err);
});
```

5.3.7 Schemas

5.3.7.1 User Schema

```
import mongoose from "mongoose";
const { Schema } = mongoose;

const userSchema = new Schema(
  {
    name: {
      type: String,
      trim: true,
      required: true,
    },
    email: {
      type: String,
      trim: true,
      required: true,
      unique: true,
    },
    password: {
      type: String,
      required: true,
      min: 6,
      max: 64,
    },
    picture: {
      type: String,
      default: "/avatar.png",
    },
    role: {
      type: [String],
      default: ["Subscriber"],
      enum: ["Subscriber", "Instructor", "Admin"],
    },
    stripe_account_id: "",
    stripe_seller: {},
    stripeSession: {},
    passwordResetCode: {
      data: String,
      default: "",
    },
    { timestamps: true }
  );
  export default mongoose.model("User", userSchema);
```

5.3.7.2 Course Schema

```
const courseSchema = new mongoose.Schema(
{
  name: {
    type: String,
    trim: true,
    minlength: 3,
    maxlength: 320,
    required: true,
  },
  slug: {
    type: String,
    lowercase: true,
  },
  description: {
    type: {},
    minlength: 200,
    required: true,
  },
  price: {
    type: Number,
    default: 9.99,
  },
  image: {},
  category: String,
  published: {
    type: Boolean,
    default: false,
  },
  paid: {
    type: Boolean,
    default: true,
  },
  instructor: {
    type: ObjectId,
    ref: "User",
    required: true,
  },
  lessons: [lessonSchema],
},
{ timestamps: true });
export default mongoose.model("Course", courseSchema);
```

5.3.7.3 Lesson Schema

```
const lessonSchema = new mongoose.Schema(
{
  title: {
    type: String,
    trim: true,
    minlength: 3,
    maxlength: 320,
    required: true,
  },
  slug: {
    type: String,
    lowercase: true,
  },
  content: {
    type: {},
    minlength: 200,
  },
  video: {},
  free_preview: {
    type: Boolean,
    default: false,
  },
},
{ timestamps: true }
);
```

5.4 Features

The table below displays all the intended features to be incorporated and whether they have been implemented or not.

ID	Requirement	Priority
FR1-MA	Sign Up	Implemented
FR2-MA	Log In	Implemented
FR3-MA	Forget Password	Implemented
FR4-MA	View Profile	Implemented
FR5-MA	Edit Profile	Implemented
FR6-MA	Adding a new course	Implemented
FR7-MA	Editing a Course	Implemented
FR8-MA	View Settings	Implemented
FR9-MA	Edit Sign-in Information	Implemented
FR10-MA	Search Bar	Not Implemented
FR11-MA	View Notification Page	Not Implemented
FR12-MA	View Home Page	Implemented
FR13-MA	Write review area	Not Implemented
FR14-MA	Buying a course	Implemented
FR15-MA	Sign Out	Implemented

Chapter 6 Testing

6.1 Testing Approach

6.1.1 Functional Testing

Unit testing: Each function and feature was thoroughly tested on its own before being incorporated into the entire system to ensure proper functionality.

2- Integration testing: Once we confirmed that each function was functioning correctly individually, we began integrating them one by one. This approach was taken because we wanted to identify any issues that may arise when all the functions are working together and pinpoint the source of the problem.

3- Regression Testing: As modifications were made to existing functions for optimization and compatibility improvements during the course of the project, regression testing ensured that all interconnected units continued to function correctly after the changes were implemented.

4- System Testing: After completing the project, all features were thoroughly tested to confirm that there were no problems or incorrect outcomes, such as enrolling in a course and it not appearing or not registering.

6.1.2 Nonfunctional Testing

Security: As the project involves handling sensitive user information, security is a crucial aspect. Any confidential data was stored using highly secure encryption techniques. Two important pieces of user information are the user's password and payment information. The user's password is converted into a hashed password before being sent to the database. Additionally, the user's payment method is stored securely in collaboration with the Stripe API, which is used for processing payments.

Accessibility: We have decided to use one of GoDaddy's supported domains after conducting extensive research. It was determined that GoDaddy is currently one of the most secure and consistently accessible domain providers available in the market. GoDaddy is a well-known company for providing domain registration and web hosting services. They have a wide range of options for different types of customers, from individuals to businesses, and they offer a variety of tools to help with website building, online marketing, and security. Additionally, they have a user-friendly interface and a 24/7 customer support team available. By choosing GoDaddy as our domain provider, we can ensure that our website will have a reliable and secure domain name, which is critical for building trust with our users and ensuring that our website is always accessible to them. Furthermore, by leveraging the power of AWS, we can also ensure high-availability and disaster recovery, which will provide our users with a more reliable and robust system. This means that the platform will be able to handle large scale of data and user traffic without any significant performance degradation or downtime.

Data Integrity: To ensure the accuracy of the information provided by both students and instructors, we employed a variety of measures for verification, such as checking the validity of the email address entered. We understand that the integrity of the data is crucial for the smooth functioning of the platform, as well as for maintaining a high level of trust among our users. By implementing these checks, we can guarantee that the information provided by our users, moving on after opening an official receipt with AWS we can also start asking and verifying phone numbers and many more things.

Capacity: We have taken great care to ensure that our database has ample capacity to handle large amounts of data, and have implemented the use of cloud computing through AWS to make it more efficient and user-friendly. By utilizing the scalability and flexibility of cloud computing, we can ensure that the database can accommodate a large number of users and the materials they upload. This not only ensures that our platform can handle high traffic and user engagement but also provides fast and easy access to the data at all times, even during peak usage.

6.2 Testing Tools

To ensure that our system operates seamlessly and without errors or incorrect outcomes, we utilized a range of testing tools. These tools include:

- 1- Visual Studio Code's debugging functionality
- 2- Google Chrome
- 3- Multiple physical devices
- 4- Other web browsers
- 5- Different operating systems.

By using these tools, we can ensure that the system is thoroughly tested and that any issues are identified and addressed before release. This will allow us to deliver a high-quality product that performs smoothly and provides accurate results.

Chapter 7 Conclusion and Future Work

The driving force behind the development of EduLab was the realization that many individuals struggle to locate online courses that are offered in the Arabic language. Additionally, there are few platforms that provide Arabic-speaking instructors with the opportunity to showcase their skills and to monetize their expertise.

With EduLab, we strive to achieve a multitude of objectives, such as:

- Encouraging more instructors to utilize our platform, thereby expanding the breadth of topics and subject areas covered.
- Developing a user-friendly and intuitive interface that allows instructors to easily create and manage their courses.
- Providing students with a comprehensive and diverse range of course offerings, catering to a wide range of interests and skill levels.
- Building a community of learners and instructors where knowledge and skills can be shared and exchanged.
- Our ultimate goal is to create a platform that empowers Arabic-speaking instructors and makes learning accessible to all.

Appendix A

Document Changes

- We made the decision to adopt a NoSQL approach over the previously mentioned SQL method, this is because we discovered that NoSQL is more efficient. As a result, we did not need to use any database diagrams.
- We also found that the class diagram was unnecessary and so it was removed from the project. This decision was driven by our desire to provide our users with a more efficient and streamlined experience. We believe that this approach will enable us to deliver a more responsive and high-performing system.

References

Information about Udemy:

about.udemy.com

Information about Coursera:

about.coursera.org

Information about SkillShare:

www.skillshare.com/about