



UNIVERSITATEA DE VEST DIN TIMIȘOARA  
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ  
PROGRAMUL DE STUDII DE LICENȚĂ: Informatică

# LUCRARE DE LICENȚĂ

**COORDONATOR:**  
Conf. Dr. Monica Sancira

**ABSOLVENT:**  
Săndoiu Maria-Adina

TIMIȘOARA  
2023

**UNIVERSITATEA DE VEST DIN TIMIȘOARA  
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ  
PROGRAMUL DE STUDII DE LICENȚĂ: Informatică**

## **Approval Dashboard**

**COORDONATOR:**  
Conf. Dr. Monica Sancira

**ABSOLVENT:**  
Săndoiu Maria-Adina

**TIMIȘOARA  
2023**

# Abstract

The purpose of this project is to develop an application that facilitates the work of professors in monitoring the progress of each student's thesis, regardless of their specialization. This application will save time for both teachers and students through a simple module for uploading, checking, and providing feedback.

Additionally, the application will include a functionality for professors to reserve rooms within the university for specific periods of time. These reservations will be communicated to the students based on their specialization and the options selected by the professor. This way, each student will be able to see in their interface when and where they will meet with the professor.

This application has two major components: the first part is the application itself, which benefits from a database where all user data is stored, and the second part involves automating the email response from the supervising professor to the written part uploaded by the student.

The first part utilizes the React.js framework for functionality and graphical interface, as well as an SQL database. JSX is an extension of the JavaScript language syntax, which provides a way to represent components using a syntax similar to HTML. This syntax facilitates code writing and makes components more accessible for understanding. Within the React framework, JSX is used to define components, but it can also be combined with pure JavaScript. To transform JSX code into standard JavaScript objects, a pre-processor called Babel is used.

The second part of the application is implemented using the PHP language. With PHP, we establish the connection to the database to manage the necessary data and information for the application's operation. This connection allows us to store and extract information from the database efficiently and securely.

Moreover, using PHP, we will implement the functionality for automating the sending of emails. This can be achieved through the functions and libraries available in PHP for handling and sending email messages. Thus, we can automatically send notifications or information to users based on specific events or actions within the application.

By combining the functionality provided by PHP for interacting with the database and automating emails, I have created a powerful and dynamic web application that can efficiently handle and process data while being easy to maintain. [9]

## Rezumat

Scopul acestei lucrări este de a dezvolta o aplicație care să faciliteze munca profesorilor în monitorizarea stadiului lucrării de licență pentru fiecare student, indiferent de specializare. Această aplicație va economisi timp atât pentru cadrele didactice, cât și pentru studenți, prin intermediul unui modul simplu de încărcare, verificare și transmitere a feedback-ului.

În plus, aplicația va include și o funcționalitate prin care profesorii vor putea rezerva săli din cadrul universității pentru anumite perioade de timp. Aceste rezervări vor fi comunicate și studenților, în funcție de specializarea lor și de opțiunile selectate de profesor. Astfel, fiecare student va putea vedea în interfață să când și unde se va întâlni cu profesorul.

Această aplicație are două componente majore: prima parte o prezintă aplicația propriu zisă care beneficiază de o bază de date, unde sunt stocate toate datele privind fiecare utilizator în parte, iar cea dea două parte conține automatizarea în ceea ce privește emailul cu răspunsul profesorului coordonator la partea scrisă încarcată de student.

În prima parte se folosește limbajul React.js pentru funcționalitate și interfață grafică dar și baza de date SQL. JSX este o extensie a sintaxei limbajului JavaScript, care furnizează o modalitate de a reprezenta componentele utilizând o sintaxă similară cu HTML. Această sintaxă facilitează scrierea codului și face componentele mai accesibile pentru înțelegere. În cadrul framework-ului React, JSX este utilizat pentru a defini componentele, însă acesta poate fi combinat și cu JavaScript pur. Pentru a transforma codul JSX în obiecte JavaScript standard, se utilizează un pre-procesor denumit Babel.

A doua parte a aplicației este implementată utilizând limbajul PHP. Cu ajutorul PHP, vom realiza conexiunea cu baza de date pentru a gestiona datele și informațiile necesare în funcționarea aplicației. Această conexiune ne permite să stocăm și să extragem informații din baza de date în mod eficient și securizat.

De asemenea, folosind PHP, vom implementa funcționalitatea de automatizare a trimiterii de email-uri. Acest lucru poate fi realizat prin intermediul funcțiilor și librăriilor disponibile în PHP pentru manipularea și expedierea mesajelor de email. Astfel, putem trimite notificări sau informații către utilizatori în mod automat, în funcție de anumite evenimente sau acțiuni în aplicație.

Prin combinarea funcționalității oferite de PHP pentru interacțiunea cu baza de date și automatizarea email-urilor, am creat o aplicație web puternică și dinamică, care poate gestiona și procesa datele într-un mod eficient și ușor de întreținut.

# Cuprins

<b>1</b>	<b>Introducere</b>	<b>6</b>
<b>2</b>	<b>Analiza stării de artă</b>	<b>8</b>
2.1	Aplicații similare . . . . .	8
2.2	Moodle.Prezentare generală . . . . .	8
2.2.1	Dezavantaje . . . . .	9
2.3	Google Classroom.Prezentare generală . . . . .	9
2.3.1	Dezavantaje . . . . .	10
<b>3</b>	<b>Metodologia de proiectare și dezvoltare a aplicației</b>	<b>12</b>
3.1	Tehnologii utilizate . . . . .	12
3.1.1	React.js . . . . .	12
3.1.2	React Material UI . . . . .	12
3.1.3	PHP . . . . .	13
3.1.4	MySQL . . . . .	13
3.1.5	Visual Studio Code . . . . .	14
3.1.6	Mailtrap . . . . .	15
3.2	Arhitectura sistemului . . . . .	16
3.3	Diagrama cazurilor de utilizare . . . . .	17
<b>4</b>	<b>Descrierea aplicației</b>	<b>19</b>
4.1	Facilități aplicație . . . . .	19
4.2	Detalii de implementare . . . . .	20
4.3	Interfața . . . . .	21
4.3.1	Ecran login . . . . .	21
4.3.2	Ecran principal(Student) . . . . .	23
4.3.3	Ecran principal(Profesor) . . . . .	26
4.4	Concluzii și direcții viitoare . . . . .	33
4.4.1	Direcții viitoare . . . . .	33
4.4.2	Concluzii . . . . .	33
	<b>Bibliografie</b>	<b>34</b>

# **Capitolul 1**

## **Introducere**

O aplicație web este o aplicație software accesibilă prin intermediul unui browser web. Aceasta rulează pe serverul web și permite utilizatorilor să acceseze și să utilizeze funcționalitățile aplicației prin intermediul unei conexiuni la internet. Principalul avantaj al aplicațiilor web este că nu necesită instalare sau actualizare pe dispozitivul utilizatorului, deoarece sunt accesate prin intermediul unui browser web. Acestea sunt compatibile cu diferite dispozitive și platforme, indiferent de sistemul de operare sau tipul de dispozitiv utilizat. O aplicație web poate afișa o interfață vizuală interactivă în browser, permite utilizatorilor să introducă și să proceseze date, să comunice cu serverul și să primească răspunsuri în timp real. Aceste aplicații pot fi complexe și pot include funcționalități precum înregistrarea și autentificarea utilizatorilor, stocarea și gestionarea datelor, interacțiunea socială, tranzacțiile online și multe altele.

## **Motivația dezvoltării aplicației**

Ideea de a crea această aplicație a apărut ca urmare a experienței personale de a avea lucrarea de licență și de a fi nevoie să participăm la întâlniri periodice cu profesorul coordonator. Situația era adesea provocatoare, deoarece trebuia să așteptăm unii după alții și timpul era limitat. Am realizat că există o nevoie evidentă de a simplifica și a eficientiza procesul de monitorizare și comunicare a progresului lucrării de licență.

Observând procesul tradițional de monitorizare a lucrărilor de licență și comunicare între studenți și profesorii coordonatori, am identificat nevoia evidentă de a simplifica și optimiza acest proces. În cadrul procesului tradițional, studenții și profesorii coordonatori trebuie să se întâlnească personal destul de des, sau prin intermediul platformelor deja existente, în mod regulat pentru a discuta despre progresul lucrărilor de licență. Aceste întâlniri pot fi dificil de programat, consumatoare de timp și pot crea neplăceri pentru ambii participanți. Prin intermediul aplicației, am vrut să simplific comunicarea, permitând studenților să încarce versiunile lucrărilor lor, iar profesorii să ofere feedback și să monitorizeze progresul în mod eficient, evitând astfel necesitatea întâlnirilor frecvente.

## **Obiective**

Obiectivele pentru crearea acestei aplicații au constat în simplificarea și optimizarea procesului de monitorizare a lucrărilor de licență, scopul principal fiind oferirea unei

soluții eficiente și ușor de utilizat pentru studenți și profesori în monitorizarea și comunicarea progresului lucrărilor de licență. Aplicația urmărește eliminarea restricțiilor și dificultăților întâlnite în procesul tradițional de întâlniri fizice dese sau prin intermediul altor platforme existente. Un alt obiectiv îl constituie economisirea timpului și eficientizarea comunicării, aplicația permite studenților să încarce versiunile lucrărilor lor în orice moment convenabil, până la data limită stabilită de profesor, iar profesorii pot oferi feedback și monitoriza progresul în mod flexibil. Aceasta ajută la economisirea timpului și permite utilizatorilor să se concentreze mai bine pe responsabilitățile lor academice sau profesionale, dar și crearea unui mediu prietenos și intuitiv, interfața aplicației fiind proiectată astfel încât să fie ușor de utilizat pentru toți utilizatorii. S-a urmărit eliminarea complexității și crearea unui mediu prietenos, în care utilizatorii să se simtă confortabil în navigarea și utilizarea funcționalităților aplicației.

## Descriere

Aplicația este o soluție completă și eficientă pentru monitorizarea lucrărilor de licență, oferind o platformă simplă și accesibilă pentru profesori și studenți. Aceasta facilitează comunicarea și colaborarea între aceștia, permitându-le să interacționeze și să colaboreze într-un mod mult mai eficient.

Cu ajutorul aplicației, profesorii au posibilitatea de a vizualiza și evalua lucrările încărcate de către studenți. Ei pot accesa informații detaliate despre fiecare student și pot urmări progresul individual al lucrărilor de licență. Prin intermediul acestei platforme, profesorii pot oferi feedback detaliat și relevant, ajutând studenții să își îmbunătățească lucrările.

O alta secțiune importantă a aplicației este faptul că cadrele didactice pot programa întâlniri cu studenții într-un mod rapid și eficient. Ei pot selecta data și ora dorită pentru întâlnire direct în aplicație, fără a fi nevoie de comunicare suplimentară prin intermediul e-mailului sau a altor canale. Aceasta reduce timpul și efortul necesar pentru stabilirea programărilor și facilitează gestionarea întâlnirilor multiple cu studenții.

## Problema și rezolvarea acesteia

Una dintre problemele majore întâmpinate în procesul de monitorizare a lucrărilor de licență este gestionarea eficientă a informațiilor și a comunicării între profesori și studenți. Metodele tradiționale, precum întâlniri fizice dese pentru vizualizarea progresului sau corespondență prin e-mail, pot fi ineficiente și pot duce la pierdere de timp și confuzie.

Soluția este să se evite această interacțiune, iar atunci când o lucrarea de licență este trimisă către aprobare, aceasta să fie transmisă direct către profesorul coordonator, în vederea corectării și soluționării problemelor pe care le poate avea.

Astfel, timpul acordat întâlnirilor este oferit altor activități, totodată feedback-ul transmis de către profesor poate primi durata necesară, pentru a putea fi mai ușor de înțeles de către student.

# **Capitolul 2**

## **Analiza stării de artă**

### **2.1 Aplicații similare**

Există o varietate de aplicații disponibile pe piață care oferă funcționalități similare celei descrise în cadrul aplicației tale. Aceste aplicații au fost create pentru a satisface nevoile utilizatorilor în gestionarea lucrărilor de licență și pentru a îmbunătăți procesele academice asociate precum Moodle sau Turnitin .

### **2.2 Moodle.Prezentare generală**

Moodle este o platformă de învățare online care facilitează procesul de predare și învățare într-un mediu virtual. Este una dintre cele mai populare și larg utilizate soluții de e-learning, fiind folosită în instituții de învățământ de toate nivelurile, de la școli și universități până la organizații de formare profesională și companii.

Platforma Moodle oferă o gamă variată de funcționalități și instrumente care permit profesorilor să creeze și să administreze cursuri online interactiv, iar studenților le oferă posibilitatea să acceseze materialele de curs, să participe la activități și să interacționeze cu colegii și profesorii lor.

Interfața intuitivă și ușor de utilizat a Moodle facilitează accesul la conținutul cursurilor și navigarea în cadrul platformei. Profesorii pot încărca resurse și materiale de studiu, cum ar fi documente, prezenteri, videouri sau linkuri către resurse externe. De asemenea, pot crea activități interactiv, cum ar fi forumuri de discuții, teste, sarcini sau colaborarea la proiecte în grup. Studenții pot accesa materialele de curs, să participe la discuțiile online și să își completeze sarcinile și teste în mod interactiv. Moodle oferă, de asemenea, funcționalități de evaluare și feedback, permitând profesorilor să monitoreze progresul studenților, să evaluateze lucrările și să ofere feedback personalizat. Un aspect important al Moodle este flexibilitatea și personalizarea sa. Platforma permite personalizarea aspectului vizual și configurarea setărilor de curs în funcție de nevoie și preferințele fiecărei instituții sau a fiecărui curs. De asemenea, poate fi extinsă prin adăugarea de module suplimentare și integrarea cu alte sisteme sau instrumente de învățare. Un alt avantaj al Moodle este comunitatea vastă și activă de utilizatori și dezvoltatori. Există o mare varietate de resurse, forumuri și documentație disponibile online, care oferă suport și soluții pentru utilizatori. Moodle este o platformă puternică și versatilă de învățare online, care facilitează interacțiunea și colaborarea între pro-

fesori și studenți într-un mediu virtual. Cu ajutorul său, instituțiile de învățământ și organizațiile pot oferi cursuri și materiale de învățare interactiv, accesibile de oriunde și oricând, contribuind astfel la îmbunătățirea experienței de învățare și predare.

### 2.2.1 Dezavantaje

- Complexitatea initială: Implementarea initială a Moodle poate fi o provocare, mai ales pentru instituțiile sau organizațiile mai mici, care pot avea resurse tehnice limitate. Este necesar să se aloce timp și efort pentru configurarea, personalizarea și menținerea platformei.
- Curba de învățare: Utilizarea Moodle poate necesita o perioadă de familiarizare și învățare pentru utilizatori, mai ales pentru cei care nu sunt obisnuiți cu platformele de învățare online sau nu au experiență tehnică avansată. Acest lucru poate crea dificultăți initiale și poate necesita formare suplimentară pentru cadrele didactice și studenți.
- Nevoia de resurse hardware și software: Pentru a rula Moodle eficient, este nevoie de resurse hardware și software adecvate. Acest lucru poate include un server puternic, spațiu de stocare suficient și o configurație adecvată a serverului web. Aceste resurse pot necesita investiții suplimentare și expertiza tehnică pentru a le gestiona.
- Limitări ale funcționalității: În ciuda oferirii unui set bogat de funcționalități, Moodle poate avea anumite limitări sau poate fi dificil de personalizat pentru a se potrivi exact nevoilor specifice ale unei instituții sau organizații. Poate fi necesara implicarea dezvoltatorilor pentru a adăuga funcționalități personalizate sau a îmbunătăți anumite aspecte ale platformei.

[4]

## 2.3 Google Classroom. Prezentare generală

Google Classroom este o platformă de gestionare a cursurilor online dezvoltată de Google, concepută special pentru mediul educațional. Aceasta oferă profesorilor și elevilor un mediu digital interactiv în care pot comunica, colabora și împărtăși resurse educaționale.

Cu Google Classroom, profesorii pot crea cursuri virtuale și le pot distribui elevilor, permitându-le să acceseze materialele de învățare, să trimită teme, să participe la discuții și să primească feedback. Profesorii pot crea și distribui sarcini, precum teste și cuestionare, iar elevii le pot completa și trimite direct prin platformă. Profesorii pot monitoriza și evalua lucrările studentilor într-un mod eficient. Elevii pot accesa cursurile și materialele de învățare într-un mod organizat și centralizat, având posibilitatea de a primi notificări despre sarcinile noi sau schimbările în cursurile lor. Ei pot colabora cu colegii și pot participa la discuții în cadrul claselor virtuale. Google Classroom oferă, de asemenea, integrare cu alte instrumente Google, precum Google Drive și Google Docs, pentru o colaborare mai eficientă și pentru partajarea resurselor.

Prin intermediul Google Classroom, comunicarea între profesori și elevi este facilitată, iar procesul de învățare devine mai interactiv și mai accesibil. Profesorii pot gestiona cu ușurință cursurile și resursele, iar elevii pot avea acces la materialele de învățare în orice moment și de pe orice dispozitiv conectat la internet. Google Classroom aduce beneficii precum simplificarea procesului de administrare a cursurilor, organizarea și partajarea eficientă a resurselor educaționale, interacțiunea și colaborarea ușoară între profesori și elevi, precum și accesul facil la informații și materiale de învățare. Platforma este utilizată la scară largă în instituții de învățământ din întreaga lume și contribuie la transformarea experienței de învățare în mediul digital.

### 2.3.1 Dezavantaje

- Limitări ale funcționalității: Deși Google Classroom oferă un set solid de caracteristici, poate exista uneori nevoie de funcționalități suplimentare sau personalizate. Anumite cerințe specifice ale instituțiilor de învățământ sau ale profesorilor pot depăși capacitatele platformei și pot necesita utilizarea altor instrumente sau soluții.
- Dependență de infrastructura Google: Utilizarea Google Classroom înseamnă încrederea în infrastructura și serviciile oferite de Google. Dacă există probleme cu serverele sau sistemele Google, sau dacă platforma în sine întâmpină probleme tehnice, poate afecta accesul și funcționarea adecvată a claselor virtuale.
- Dificultăți pentru utilizatorii neexperimentați: Pentru unii profesori și elevi, utilizarea Google Classroom poate necesita o perioadă de învățare și familiarizare. Interfața și fluxul de lucru pot fi inițial complexe pentru cei care nu sunt obișnuiți cu tehnologia sau nu au experiență în utilizarea platformelor de învățare online.

[3] Dacă s-ar face o comparație între ce este deja pe piață(exmplul oferit mai sus) și ce este prezentat în această lucrare ar fi autenticitatea sugerată și în titlu . Am inclus o funcționalitate autentică și importantă pentru procesul de evaluare a lucrărilor de licență - "Approve" și "Decline". Această caracteristică are un impact semnificativ asupra rezultatului final al unei lucrări și asigură o evaluare justă și obiectivă.

Atunci când un student își finalizează lucrarea de licență și o încarcă în aplicație, profesorul coordonator are posibilitatea de a evalua și de a decide dacă acesta primește aprobarea ("Approve") sau respingerea ("Decline"). Această etapă crucială are la bază un criteriu specific și transparent - suma notelor acordate de profesor studentelor.

Pentru a exemplifica, să presupunem că sunt disponibile 10 teme de licență, iar nota maximă pe care un student o poate obține este 10. Înainte ca lucrarea să fie aprobată, profesorul trebuie să evaluateze fiecare aspect al lucrării și să acorde note individuale pentru fiecare student. Suma totală a acestor note acordate trebuie să depășească 6 pentru ca lucrarea să fie considerată pentru aprobare.

Această abordare se asigură că profesorii analizează și evaluatează cu atenție fiecare lucrare și oferă o evaluare echilibrată. În același timp, se promovează standardele academice ridicate și se încurajează studenții să se străduiască să obțină rezultate bune

în lucrările lor de licență.

O funcționalitate unică și extrem de utilă în cadrul aplicației este capacitatea de "Rezervare sală". Această facilitate permite profesorilor să vizualizeze cu ușurință sălile disponibile într-un anumit interval orar preferat și să facă rezervări, iar informația să fie transmisă automat în interfața studentului corespunzător. Această funcționalitate oferă o soluție eficientă pentru gestionarea programărilor întâlnirilor între profesori și studenți, asigurând că acestea se desfășoară în săli adecvate și în intervalul orar dorit. Profesorii pot accesa cu ușurință calendarul disponibil cu toate sălile universității și pot verifica disponibilitatea acestora în timp real.

Prin implementarea acestor funcționalități, aplicația oferă un cadru clar și coerent pentru evaluarea lucrărilor de licență și asigură o transparentă în procesul de aprobată. Profesorii pot monitoriza stadiul lucrărilor și pot oferi feedback în timp real, îmbunătățind astfel comunicarea și eficiența în procesul de coordonare. De asemenea, prin intermediul funcționalității de rezervare a sălilor, profesorii pot planifica întâlniri și discuții cu studenții într-un mod rapid și convenabil. Ei pot vizualiza disponibilitatea sălilor în intervalul orar preferat și pot face rezervări fără dificultate. Această abordare elimină confuzia și ambiguitatea în programarea întâlnirilor și facilitează interacțiunea mult mai usoară între profesori și studenți. În plus, transparenta în procesul de aprobată oferă studenților informații clare despre stadiul lucrărilor lor și le permite să-și monitorizeze progresul într-un mod mai concret. Ei pot accesa feedbackul și recomandările profesorilor în timp util, ceea ce facilitează îmbunătățirea lucrărilor și dezvoltarea lor academică.

# Capitolul 3

## Metodologia de proiectare și dezvoltare a aplicației

### 3.1 Tehnologii utilizate

#### 3.1.1 React.js

React.js se concentrează pe crearea de componente reutilizabile și modularizate. Componentele pot fi definite ca funcții sau clase și pot fi compuse pentru a crea interfețe complexe. Unul dintre principiile fundamentale ale React.js este conceptul de "virtual DOM" (Document Object Model). Virtual DOM este o reprezentare în memorie a structurii interfeței utilizator și se actualizează eficient în comparație cu DOM-ul real, ceea ce duce la o performanță mai bună.

React.js utilizează sintaxa JSX (JavaScript XML) pentru a defini structura interfeței utilizator. JSX combină HTML-ul și JavaScript-ul într-un mod intuitiv, permitând dezvoltatorilor să creeze componente vizuale într-un mod declarativ. React.js promovează un flux unidirectional de date. Datele sunt transmise prin intermediul proprietăților (props) către componente și pot fi actualizate prin intermediul funcțiilor de stare (state). Aceasta duce la o gestionare eficientă și predictibilă a datelor aplicației. React.js suportă reactivitatea și reprezentarea condițională a componentelor. Aceasta înseamnă că componentele pot reacționa la schimbările datelor și pot actualiza automat interfața utilizator.

React.js are o comunitate puternică și un ecosistem bogat de biblioteci și instrumente adiacente, cum ar fi React Router pentru gestionarea rutelor, Redux pentru gestionarea stării aplicației și multe altele.

React.js este folosit într-o varietate de aplicații web, de la aplicații simple și site-uri web până la aplicații web complexe și scalabile. Este frecvent utilizat în dezvoltarea de aplicații de tip single-page (SPA) și în crearea interfețelor utilizator reactiv. [6]

#### 3.1.2 React Material UI

React Material-UI este o bibliotecă populară și puternică pentru dezvoltarea interfețelor de utilizator în aplicații React. Cu un set bogat de componente predefinite și stiluri inspirate din designul Material Design, React Material-UI facilitează crearea de aplicații moderne și atrăgătoare. Caracteristicile cheie ale React Material-UI includ design atraktiv și coerent, componente reutilizabile, flexibilitate și personalizare. Biblioteca

se bazează pe principiile designului Material Design, oferind o estetică plăcută și un aspect vizual coherent în întreaga aplicație. Componentele predefinite au stiluri bine definite și se potrivesc între ele pentru a crea o experiență vizuală armonioasă. React Material-UI pune la dispoziție o colecție extinsă de componente reutilizabile, cum ar fi butoane, cărți, bare de navigare, casete de selectare și câmpuri de text. Aceste componente sunt bine documentate și ușor de utilizat, permitând dezvoltatorilor să creeze rapid interfețe complexe și interactive.

Un alt avantaj al React Material-UI este flexibilitatea și personalizarea. Dezvoltatorii pot modifica și personaliza aspectul și comportamentul componentelor în funcție de nevoile lor specifice. De la schimbarea culorilor și a stilurilor până la adăugarea de animații și efecte speciale, React Material-UI permite adaptarea interfeței la identitatea vizuală a aplicației și la cerințele de design. [8]

### 3.1.3 PHP

PHP, un limbaj de programare open-source, este utilizat în special pentru dezvoltarea aplicațiilor web dinamice. Acest limbaj interpretat rulează pe server și generează cod HTML către clientul web, furnizând funcționalități interactive și dinamice.

PHP are o comunitate mare și activă de dezvoltatori, beneficiind de suport și contribuții constante. Aceasta a dus la dezvoltarea unei colecții bogate de biblioteci și cadre de lucru care facilitează construirea rapidă și eficientă a aplicațiilor web.

Unul dintre avantajele majore ale PHP este ușurința sa de învățare și utilizare. Sintaxa sa este asemănătoare cu cea a altor limbaje de programare, cum ar fi C și JavaScript, ceea ce facilitează tranziția pentru dezvoltatori. PHP oferă, de asemenea, flexibilitate, permitând programatorilor să-și aleagă modul de organizare a codului și să utilizeze diverse paradigmă de programare.

PHP include suport nativ pentru interacțiunea cu baze de date, oferind posibilitatea dezvoltatorilor de a crea și gestiona baze de date MySQL sau alte sisteme de gestionare a bazelor de date. Acest lucru face PHP ideal pentru dezvoltarea aplicațiilor web care necesită stocare și manipulare a datelor.

Compatibilitatea PHP cu o gamă largă de servere web, precum Apache și Nginx, permite implementarea ușoară și rularea aplicațiilor PHP pe diferite platforme.

În rezumat, PHP este un limbaj de programare puternic și versatil pentru dezvoltarea aplicațiilor web. Datorită ușurinței de învățare, flexibilității și suportului extensiv, PHP rămâne o alegere populară pentru dezvoltatorii web în prezent. [7]

### 3.1.4 MySQL

MySQL este un sistem de gestionare a bazelor de date relationale (RDBMS) care a câștigat o popularitate considerabilă în comunitatea dezvoltatorilor și administratorilor de baze de date. Inițial dezvoltat de către compania suedeză MySQL AB, MySQL a fost lansat pentru prima dată în anul 1995 și, de atunci, a evoluat într-un RDBMS robust și fiabil.

Unul dintre motivele principale pentru popularitatea MySQL este faptul că este o soluție open-source. Aceasta înseamnă că codul sursă al sistemului este disponibil public și poate fi modificat și adaptat în funcție de nevoile individuale. Acest aspect a facilitat o comunitate activă și puternică de dezvoltatori care au contribuit la

îmbunătățirea continuă a MySQL de-a lungul anilor.

MySQL oferă o gamă largă de funcționalități care acoperă toate aspectele esențiale ale gestionării bazelor de date. Aceasta include crearea, modificarea și interogarea datelor, precum și gestionarea utilizatorilor și a drepturilor de acces. De asemenea, MySQL suportă multiple tipuri de stocare a datelor, cum ar fi tabelele InnoDB și MyISAM, oferind flexibilitate în funcție de cerințele proiectelor.

Un alt aspect esențial al MySQL este performanța sa ridicată. Datorită arhitecturii sale bine optimizate și tehnologiilor avansate de caching, MySQL poate gestiona un volum mare de date și poate efectua interogări complexe într-un timp scurt. Această caracteristică este extrem de valoroasă în aplicațiile care necesită o prelucrare rapidă a informațiilor, cum ar fi aplicațiile web sau sistemele de gestionare a stocurilor.

MySQL este, de asemenea, cunoscut pentru scalabilitatea sa. Aceasta înseamnă că poate gestiona cu ușurință creșterea volumului de date și a numărului de utilizatori. Acest lucru este esențial într-un mediu în care nevoile de stocare și procesare a datelor pot crește rapid în timp. MySQL oferă opțiuni de scalabilitate orizontală și verticală, permitând administrarea eficientă a creșterii sistemului. [1]

### 3.1.5 Visual Studio Code

Visual Studio este unul dintre cele mai populare și puternice medii de dezvoltare integrate (IDE) disponibile pe piață. Creat de Microsoft, acesta oferă programatorilor o gamă largă de instrumente și funcționalități pentru a crea aplicații de înaltă calitate în diverse limbi de programare.

Unul dintre aspectele remarcabile ale Visual Studio este suportul său extins pentru diferite limbi. De la limbi clasice precum C, C++ și Visual Basic, până la limbi moderne precum Python, JavaScript și TypeScript, Visual Studio permite dezvoltatorilor să lucreze în mediul preferat. IDE-ul oferă completare automată de cod, evidențiere de sintaxă, depanare și multe alte caracteristici utile pentru a îmbunătăți eficiența programatorilor.

Un alt avantaj major al Visual Studio este integrarea puternică cu platformele și serviciile Microsoft. Dezvoltatorii pot crea aplicații pentru Windows, Azure, iOS, Android și multe alte platforme, utilizând instrumentele și bibliotecile specifice fiecărei platforme. De asemenea, Visual Studio permite integrarea cu servicii precum Azure DevOps, GitHub și Docker, facilitând colaborarea și gestionarea proiectelor de echipă. Visual Studio oferă, de asemenea, suport bogat pentru dezvoltarea de aplicații web și cloud. Dezvoltatorii pot crea site-uri web dinamice și aplicații web robuste utilizând ASP.NET și ASP.NET Core, beneficiind de unelte avansate pentru dezvoltarea front-end și back-end. De asemenea, Visual Studio permite crearea și implementarea ușoară a serviciilor și aplicațiilor în cloud utilizând platforma Azure.

Pentru dezvoltarea de aplicații mobile, Visual Studio furnizează şablonane și unelte puternice pentru dezvoltarea aplicațiilor iOS și Android. Dezvoltatorii pot crea aplicații mobile native sau cross-platform utilizând Xamarin, o platformă de dezvoltare mobilă integrată în Visual Studio. Aceasta permite partajarea de cod între platforme și dezvoltarea rapidă a aplicațiilor mobile.

În plus, Visual Studio dispune de o gamă largă de extensii și plugin-uri create de comunitatea dezvoltatorilor. Acestea permit personalizarea IDE-ului și adăugarea de funcționalități suplimentare în funcție de nevoile fiecărui dezvoltator. De la instrumente de testare și depanare la teme și stiluri personalizate, există o varietate de

extensii disponibile pentru a îmbunătăți experiența de dezvoltare.

Visual Studio este o platformă puternică și versatilă pentru dezvoltarea de software, oferind instrumente și funcționalități avansate pentru programatori. Indiferent de limbajul sau platforma de dezvoltare preferate, Visual Studio pune la dispoziție resursele necesare pentru a crea aplicații de înaltă calitate și performanță. [5]

### 3.1.6 Mailtrap

Mailtrap este o platformă de testare și simulare a serviciului de trimitere și recepționare a e-mailurilor. Cu ajutorul Mailtrap, dezvoltatorii pot crea și gestiona cutii de e-mail virtuale pentru a testa și a verifica cum sunt procesate mesajele de e-mail într-un mediu de dezvoltare sau de testare. Mailtrap permite utilizatorilor să creeze cutii de e-mail virtuale pentru a captura și a vizualiza e-mailurile trimise de aplicațiile lor într-un mediu controlat. Aceasta înseamnă că mesajele nu sunt trimise în mod real către destinatarii finali, ci sunt redirectionate către cutiile de e-mail virtuale Mailtrap.

Dezvoltatorii pot utiliza Mailtrap pentru a testa și verifica aspecte precum formatul, conținutul și atașamentele mesajelor de e-mail trimise de aplicații. Acest lucru ajută la identificarea și remedierea problemelor sau erorilor înainte ca mesajele să fie trimise cu adevărat către utilizatori. Utilizatorii pot partaja cutiile de e-mail virtuale Mailtrap cu membrii echipei lor, permitându-le să vizualizeze și să lucreze împreună asupra mesajelor primite. Aceasta facilitează colaborarea și comunicarea între membrii echipei în ceea ce privește testarea și verificarea e-mailurilor.

Mailtrap oferă posibilitatea de a emula diferite medii de livrare a e-mailurilor, cum ar fi serverele SMTP sau API-urile de trimitere a e-mailurilor. Acest lucru permite dezvoltatorilor să simuleze și să testeze modul în care mesajele de e-mail sunt procesate în diferite scenarii de livrare. Aceasta pune accent pe securitatea datelor, oferind opțiuni de protecție a mesajelor și a conținutului acestora. Utilizatorii pot asigura confidențialitatea informațiilor din e-mailuri și pot gestiona accesul la cutiile de e-mail virtuale prin autentificare și permisiuni de utilizator. [2]

## 3.2 Arhitectura sistemului

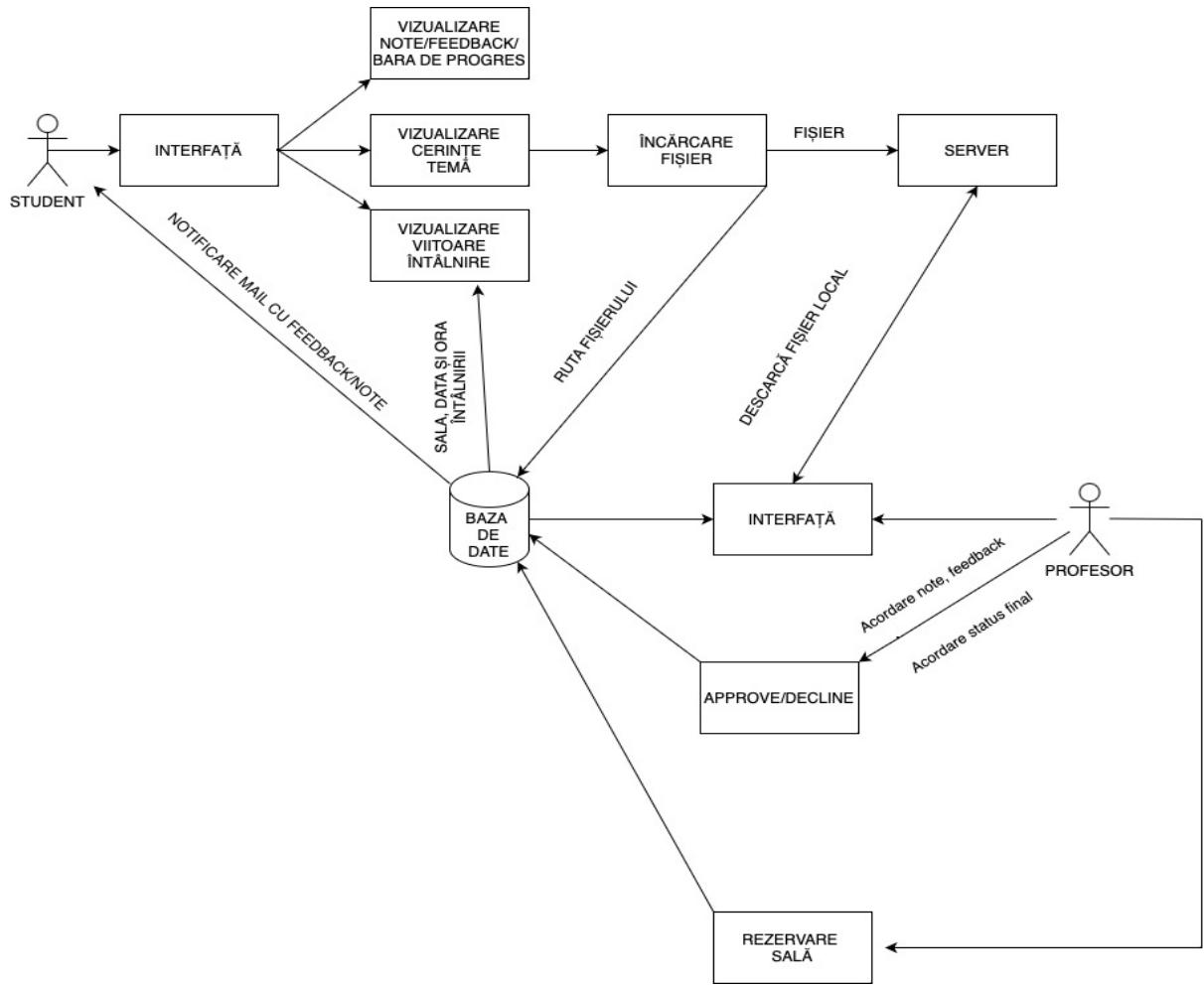


Figura 3.1: Arhitectura sistemului

În figura 3.1 se poate observa procesul de aprobare și refuzare a variantei finale a lucrării de licență, respectiv al rezervării unei săli, în contextul arhitecturii sistemului.

În partea dedicată profesorului coordonator, în interfață acestuia, lucrarea de licență încărcată de către student este afișată. Profesorul coordonator are posibilitatea de a vizualiza conținutul lucrării, dar și de a o descărca pentru o analiză mai detaliată. Această etapă permite profesorului să se familiarizeze cu conținutul lucrării și să evalueze corectitudinea și relevanța informațiilor prezentate.

Următorul pas este acordarea notei dar și a feedback-ului din partea profesorului coordonator către student. Acesta poate oferi observații, sugestii și recomandări cu privire la lucrare prin intermediul unei secțiuni de comentarii. Feedback-ul este important pentru îmbunătățirea lucrării și pentru a asigura că studenții primesc ghidare adecvată în procesul de finalizare a lucrării de licență.

O altă funcționalitate care aparține profesorului, este posibilitatea de a putea rezerva o sală pentru o întâlnire cu studenții de la specializarea selectată. După parcurserea tuturor etapelor necesare pentru finalizarea lucrării de licență, profesorul poate încadra lucrarea la statusul final, acesta putând fi "APPROVE" sau "DECLINE".

Toate aceste etape sunt înregistrate și stocate în baza de date, iar ulterior sunt afișate în interfața studentului. Baza de date este utilizată pentru a gestiona și a păstra informațiile referitoare la stadiul lucrării de licență, evaluări, feedback, rezervări de săli și alte detalii relevante.

În partea dedicată studentului, acesta are posibilitatea de a vizualiza tema asignată și poate trimite fișierul pentru verificare. După încărcare, fișierul este stocat în baza de date a aplicației și, în același timp, salvat pe serverul aplicației pentru accesibilitate și gestionare facilă. Pe lângă vizualizarea temei asignate, în interfață este secțiunea dedicată notelor, unde se evidențiază bara de progres cu notele acumulate până în momentul respectiv, dar și feedback-ul la ultima temă trimisă către evaluare, totodată este vizibilă și viitoare întalnire, programată de profesorul coordonator.

### 3.3 Diagrama cazurilor de utilizare

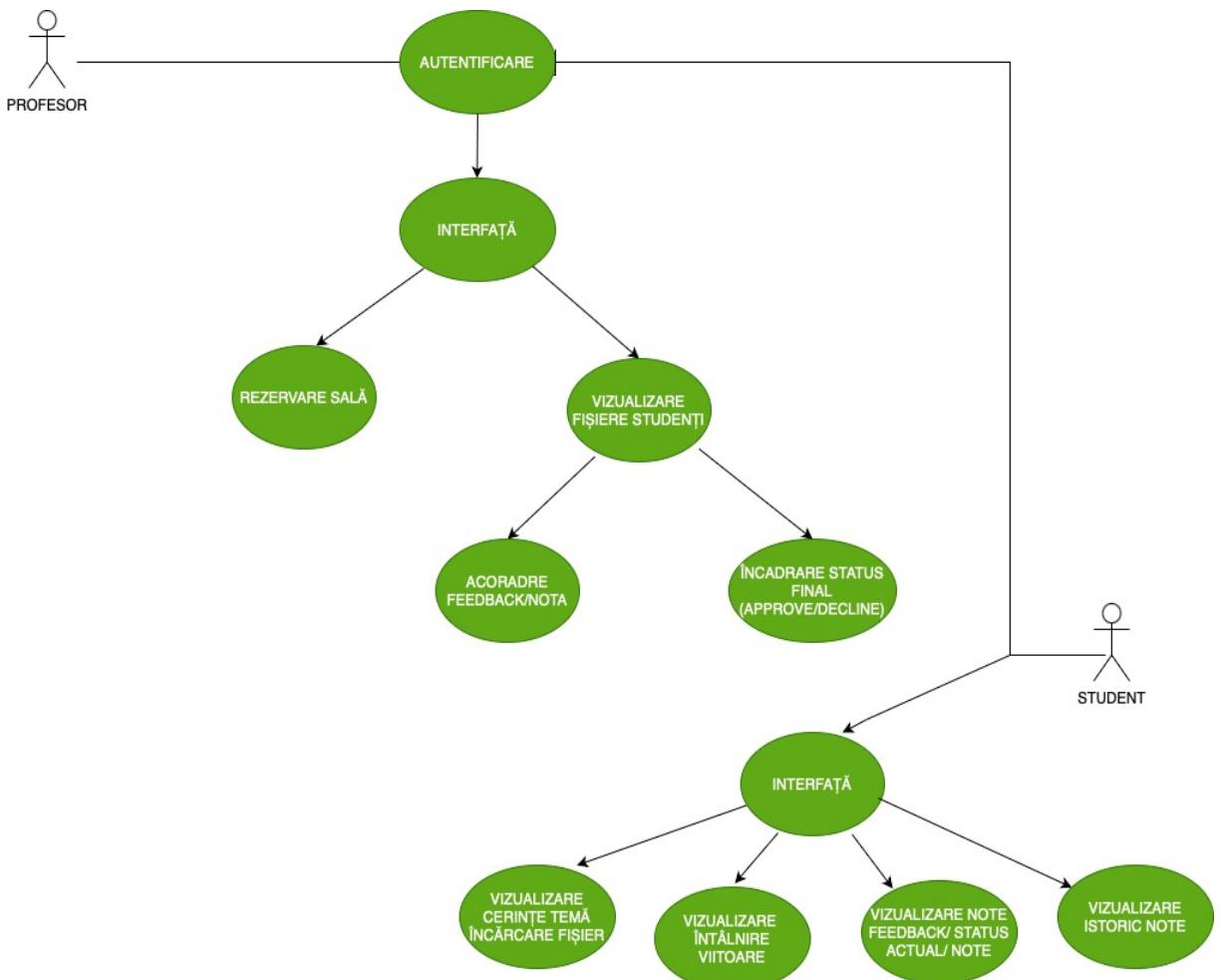


Figura 3.2: Diagrama cazurilor de utilizare

În figura 3.2, sunt prezentate cazurile de utilizare disponibile în sistem pentru profesori și studenți, oferindu-le o imagine generală asupra funcționalităților aplicației.

Studenții pot accesa aplicația prin autentificare utilizând adresa lor de email instituțională UVT și parola corespunzătoare. După autentificare, ei vor vedea în dashboard-ul lor butonul corespunzător temei asignate de profesor. Apăsarea acestui buton permite studenților să obțină detalii despre temă, cum ar fi descrierea și deadline-ul acesteia. De asemenea, ei au posibilitatea de a încărca fișierul cu rezolvarea cerințelor temei, care poate fi în format PDF, video sau imagini. O secțiune dedicată feedback-ului profesorului coordonator este disponibilă, unde studenții pot vedea notele acordate, o bară de progres și notificări referitoare la viitoarele întâlniri fizice.

Profesorii se autentifică în aplicație utilizând adresa lor de email instituțională UVT și parola corespunzătoare. După autentificare, ei pot vizualiza și descărca fișierul încărcat de studenți, pentru a examina rezolvarea cerințelor temei în detaliu și pentru a oferi feedback cât mai precis și adekvat. Profesorii au opțiunea de a oferi feedback și note pentru lucrare, iar apoi pot decide să încadreze lucrarea ca fiind aprobată sau respinsă, în funcție de suma notelor acordate. Este important de menționat că lucrarea nu poate fi considerată pentru sesiunea de licență dacă suma notelor nu a depășit pragul minim de trecere, care poate fi de exemplu 60 de puncte. De asemenea, profesorii pot programa întâlniri viitoare, selectând data și ora întâlnirii din interfața aplicației, iar apoi pot alege din sălile disponibile în acel interval orar pentru a rezerva sala potrivită.

Aceste funcționalități facilitează interacțiunea și colaborarea între profesori și studenți în cadrul aplicației. Studenții pot primi feedback direct de la profesori, iar profesorii pot evalua și monitoriza progresul studenților într-un mod eficient. Astfel, aplicația oferă un mediu interactiv și ușor de utilizat pentru gestionarea temelor și comunicarea în cadrul procesului de învățare. [9]

# Capitolul 4

## Descrierea aplicației

### 4.1 Facilități aplicație

Interfața aplicației are un design minimalist și ușor de accesat și de gestionat, cu un timp de răspuns rapid la comanda acționată de utilizator. Aplicația poate oferi o modalitate de autentificare și autorizare a utilizatorilor, permitându-le să acceseze funcționalitățile aplicației într-un mod securizat și personalizat, totodată poate oferi utilizatorilor posibilitatea de a căuta și filtra conținutul în aplicație în funcție de criteriile lor.

Pentru interfața utilizatorilor, când statusul este 'profesor', după conectare, aceștia pot comunica cu studenții prin intermediul aplicației. Aceasta poate include caracteristici precum trimiterea de mesaje individuale pentru a oferi feedback, notificări privind sarcini sau informații importante. Totodata, aceasta include un sistem de gestionare a notelor, evidențierea punctelor tari și a aspectelor de îmbunătățit, precum și vizualizarea progresului în funcție de notele obținute, printr-o bară de progres, în funcție de care se va putea oferi feedback-ul final 'Approve' sau 'Decline'. O altă funcționalitate importantă o constituie vizibilitatea asupra disponibilității sărilor. Aplicația afișează sălile disponibile în funcție de data și ora selectate de profesor. Acest lucru ajută la evitarea conflictelor de programare și la asigurarea unei săli adecvate pentru întâlnire. Profesorii pot alege cu ușurință o sală care să se potrivească nevoilor lor și ale studenților.

În ceea ce privește interfața utilizatorilor atunci când statusul este 'student', funcționalitățile sunt diferite, studentul poate încărca fisierul reprezentat de tema atribuită, poate vizualiza notele și statusul în care este cu lucrarea prin intermediul barei de progres, dar și dacă o întâlnire a fost programată de către profesor.

Pe lângă partea de interfață, există și partea de automatizare. Aceasta ușurează și mai mult utilizarea aplicației, având ca și element principal mail-ul, acesta trimițându-se automat atunci când profesorul a acordat nota și feedback-ul, dar și atunci când o tema a fost asignată.

## 4.2 Detalii de implementare

Implementarea unei aplicații este un proces complex și captivant, care necesită o abordare atentă și bine planificată. Pentru început, am definit în mod clar cerințele aplicației și am stabilit funcționalitățile pe care am dorit să le includ. Aceste cerințe vor sta la baza întregului proces de dezvoltare.

Odată ce cerințele au fost stabilite, următorul pas a fost proiectarea arhitecturii aplicației. Aici m-am concentrat pe structurarea logică a aplicației și pe definirea modulelor și a interacțiunilor între acestea. Arhitectura trebuie să fie bine gândită și să ofere flexibilitate pentru viitoarele extensii și modificări ale aplicației.

După ce am stabilit arhitectura, m-am concentrat pe dezvoltarea interfeței utilizatorilor. În acest stadiu, m-am asigurat că interfața aplicației este intuitivă, ușor de utilizat și atractivă din punct de vedere vizual.

Una dintre părțile centrale ale implementării este dezvoltarea funcționalităților aplicației. Aici am lucrat la codificarea logicii de bază care permite utilizatorilor să acceseze și să utilizeze diferitele funcționalități ale aplicației. Aceasta poate include conexiunea, vizualizarea statusului, comunicarea între profesori și studenți, evaluarea performanței studenților și multe altele. Am utilizat limbaje de programare precum PHP, JavaScript și tehnologii adecvate pentru a realiza aceste funcționalități.

Un alt aspect important a fost implementarea și gestionarea bazei de date a aplicației. Aceasta permite să stocăm și să gestionăm datele utilizatorilor și alte informații relevante. Am definit o schemă de baze de date care reflectă structura datelor și relațiile dintre ele. Pentru a interacționa cu baza de date, am utilizat limbajul SQL (Structured Query Language) și alte tehnologii asociate acestuia. SQL a oferit un set robust de comenzi și operații pentru a extrage, actualiza și gestiona datele din baza de date. Prin intermediul acestor interogări SQL, am putut obține informații specifice, am realizat actualizări și am efectuat operațiuni complexe de interogare.

Pe măsură ce am avansat în procesul de implementare, a fost important să efectuez teste pentru a mă asigura că aplicația funcționează corect. Am realizat teste unitare, teste de integrare și teste de regresie pentru a identifica și rezolva eventualele erori sau probleme. Testarea și depanarea reprezintă un pas critic pentru a livra o aplicație de calitate utilizatorilor.

## 4.3 Interfață

### 4.3.1 Ecran login

Pentru partea de interfață, mai exact partea de front-end, construită folosind framework-ul de JavaScript, mai exact React, am început cu partea de autentificare folosind adresa instituțională UVT. Această adresă trebuie să fie înregistrată în baza de date, alături de parolă. Dacă aceste date nu sunt corect introduse, va fi afișat un mesaj de avertizare printr-un pop-up.

Pentru a realiza aceasta, am implementat un formular de login care solicită utilizatorului să introducă adresa de e-mail instituțională și parola asociată, prezentată în figura 4.1 . După ce utilizatorul completează aceste câmpuri și apasă pe butonul de autentificare, datele sunt trimise către server pentru a fi verificate în baza de date.

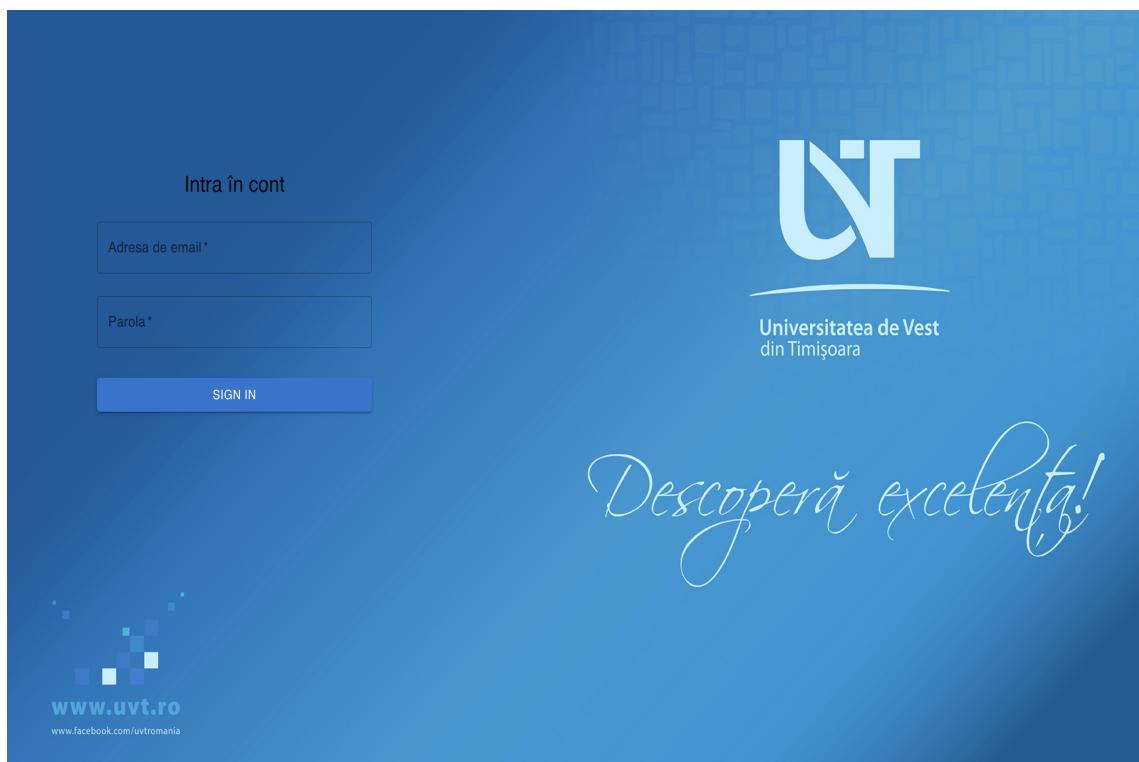


Figura 4.1: Pagina de Login

În cazul în care datele introduse nu corespund cu cele din baza de date, se afișează un mesaj de eroare în mod vizibil, prin intermediul unui pop-up sau a unei notificări. În figura 4.2 este prezentat mesajul de eroare care indică că adresa de e-mail sau parola este incorrectă, iar utilizatorul este încurajat să reîncerce introducerea corectă a datelor de autentificare. Am verificat dacă ceea ce este introdus în textfield corespunde cu ceea ce este în baza de date. Am creat funcția `checkEmailExistsLog()`, este o funcție asincronă (async function) care folosește librăria Axios pentru a efectua o cerere GET către o adresă specificată. Cererea include adresa de email (`emailLog`) și parola (`parolaLog`) ca parametri în URL.



Figura 4.2: Pagina de Login cu eroare

După ce serverul răspunde, rezultatul este stocat în variabila response. Apoi, se verifică dacă valoarea response.data.exists este false. Dacă este false, înseamnă că adresa de email și parola introduse nu există în baza de date, și prin urmare se setează variabila setOpenModalLog ca true pentru a deschide un modul de avertizare. În caz contrar, se presupune că autentificarea a fost reușită și utilizatorul este redirecționat către pagina ”/Portal” utilizând funcția navigateLog(). De asemenea, valoarea response.data.exists este asignată variabilei setTot.

```

async function checkEmailExistsLog() {
    try {
        const response = await axios.get('http://localhost:8080
/approve_app/php/check_email.php?
email=${emailLog}&parola=${parolaLog}');
        if (response.data.exists === false) {
            setOpenModalLog(true);
        } else {
            navigateLog("/Portal");
        }
        setTot(response.data.exists);
    } catch (error) {
        console.error(error);
    }
}
function handleClick() {
    if (emailLog === "" || parolaLog === "") {
        setOpenModalLog2(true);
    } else {
        checkEmailExistsLog();
    }
}

```

#### 4.3.2 Ecran principal(Student)

După ce procesul de conectare este finalizat cu succes, utilizatorul este direcționat către pagina principală, unde are loc atribuirea contului la statutul de "student" sau "profesor". În funcție de statutul atribuit, utilizatorul va avea acces la o interfață specifică, adaptată nevoilor și funcționalităților relevante.

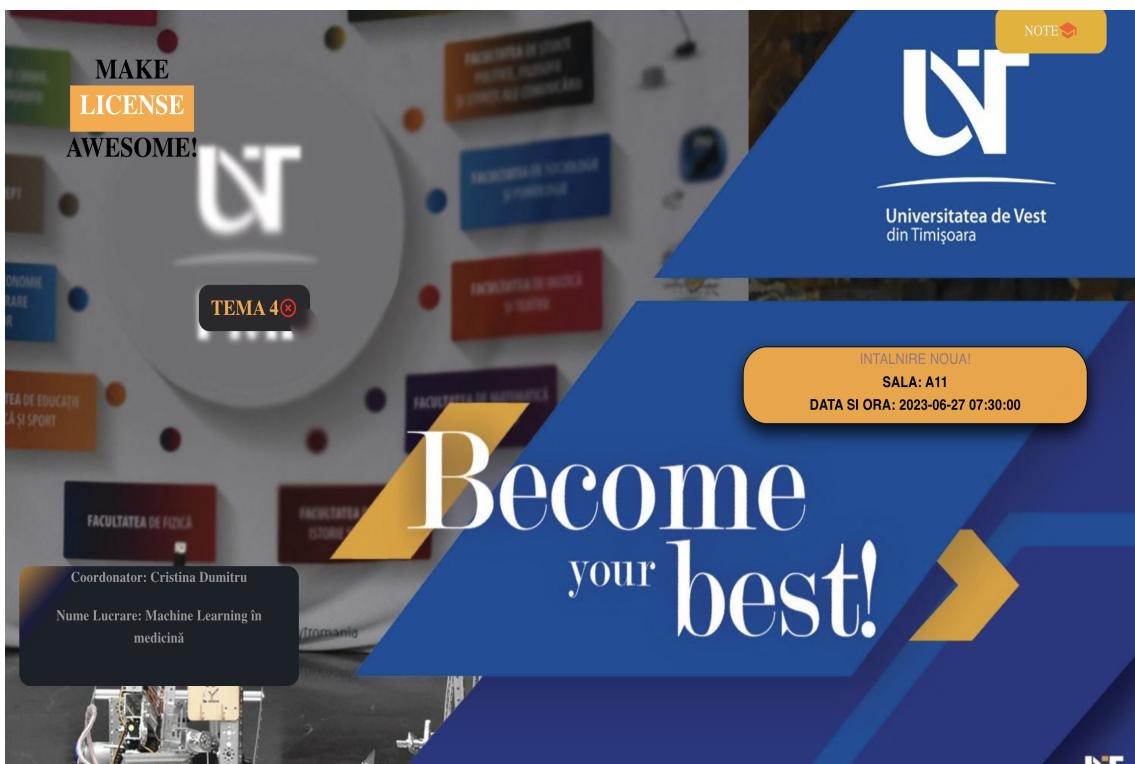


Figura 4.3: Interfață student(Pagina principală)

În ceea ce privește interfața studentului, prezentată în figura 4.3, avem ca și buton principal, opțiunea "Tema" care permite studenților să acceseze tema de licență nou asignată. În această secțiune, ei pot vedea detaliiile temei, inclusiv cerințele specifice care trebuie îndeplinite, dar și data limită până când este disponibilă încărcarea. Odată ce data limită este depășită, fișierul nu mai poate fi încărcat. Această funcționalitate oferă studenților un mod simplu și eficient de a vedea tema lor de licență și de a începe procesul de elaborare a lucrării. Ei pot accesa cerințele și instrucțiunile detaliate pentru a-și orienta munca și pot încărca lucrarea finalizată atunci când sunt gata să o trimită pentru evaluare. O caracteristică utilă pentru studenți este posibilitatea de a fi notificați atunci când profesorul programează o întâlnire fizică pentru discuții referitoare la lucrarea de licență. Această notificare este afișată în aplicație și include detalii despre sala în care va avea loc întâlnirea, precum și data și ora acesteia. Această adăugare oferă studenților un mod convenabil de a fi informați cu privire la orice întâlnire programată, oferindu-le timp suficient pentru a se pregăti și a se asigura că sunt prezenti. Astfel, ei pot să-și ajusteze programul și să se asigure că sunt disponibili pentru discuțiile importante despre lucrarea lor de licență. În cadrul interfeței, se poate observa și afișarea datelor generale, cum ar fi informațiile despre profesorul coordonator și denumirea lucrării de licență.

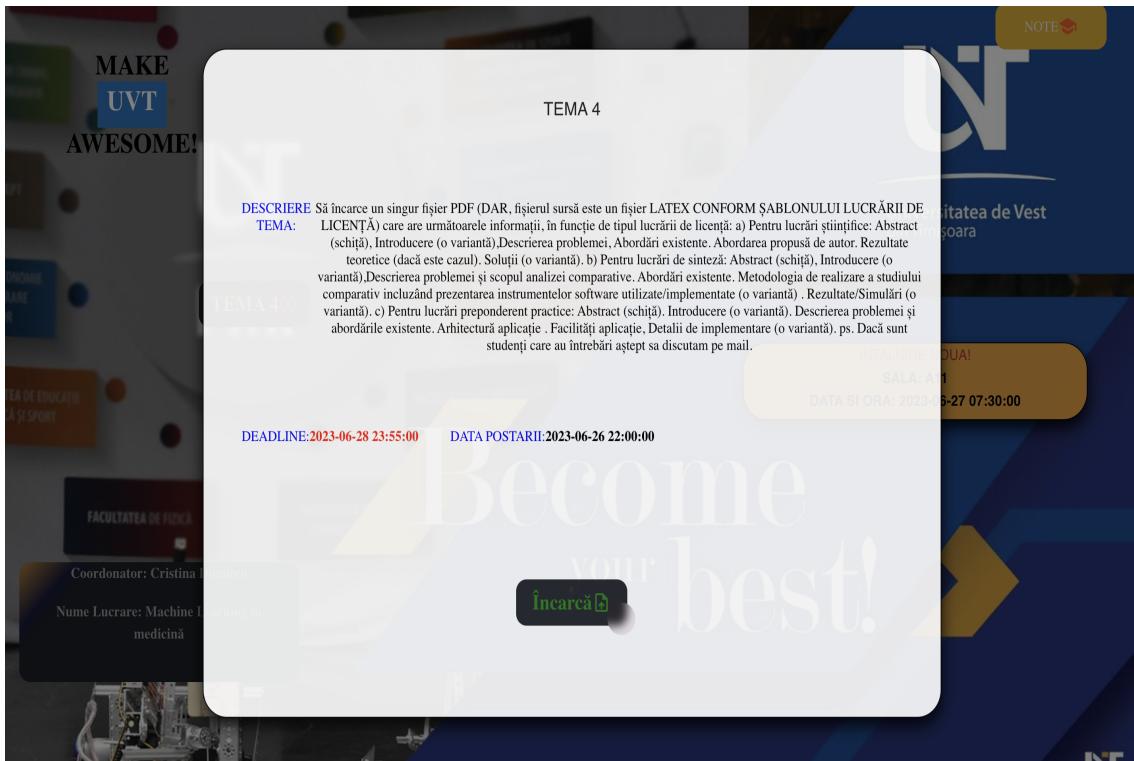


Figura 4.4: Interfață student(Detalii tema/Încărcare)

La apăsarea butonului ”Tema”, în interfața aplicației, expusă în figura 4.4 , se va deschide un câmp de text în care sunt afișate în mod clar și concis detaliile referitoare la temă. Aceste informații includ descrierea temei, obiectivele, cerințele și eventualele resurse sau referințe relevante. De asemenea, sunt vizibile și informații privind deadline-ul, oferind studenților o referință clară în privința termenului limită pentru finalizarea lucrării. În plus, în aceeași interfață, este prezent și un buton ”Încarcă”, care facilitează procesul de transmitere a lucrării finale. Acest buton le permite studenților să selecteze și să încarce fișierul lucrării lor în conformitate cu cerințele temei respective. Prin intermediul acestei funcționalități, studenții pot transmite eficient lucrarea lor către profesor. După încărcarea fișierului, butonul cu ”Tema” va fi automat inaccesibil, iar confirmarea că această temă a fost livrată pentru notare este reprezentată de iconița de lângă, aceasta fiind ori ’Check’, colorat cu verde și indicând că tema a fost încărcată, contrar fiind afișată iconița cu ’X’, având culoarea roșie.

Prin intermediul acestui flux de lucru bine definit, se asigură că studenții au o înțelegere clară a temei lor și a cerințelor asociate, iar procesul de încărcare a lucrării se realizează într-un mod structurat și eficient. Acest lucru facilitează gestionarea și evaluarea lucrărilor de licență de către profesori, oferindu-le o vizionare coerentă și ușor de urmărit asupra progresului și rezultatelor studenților.

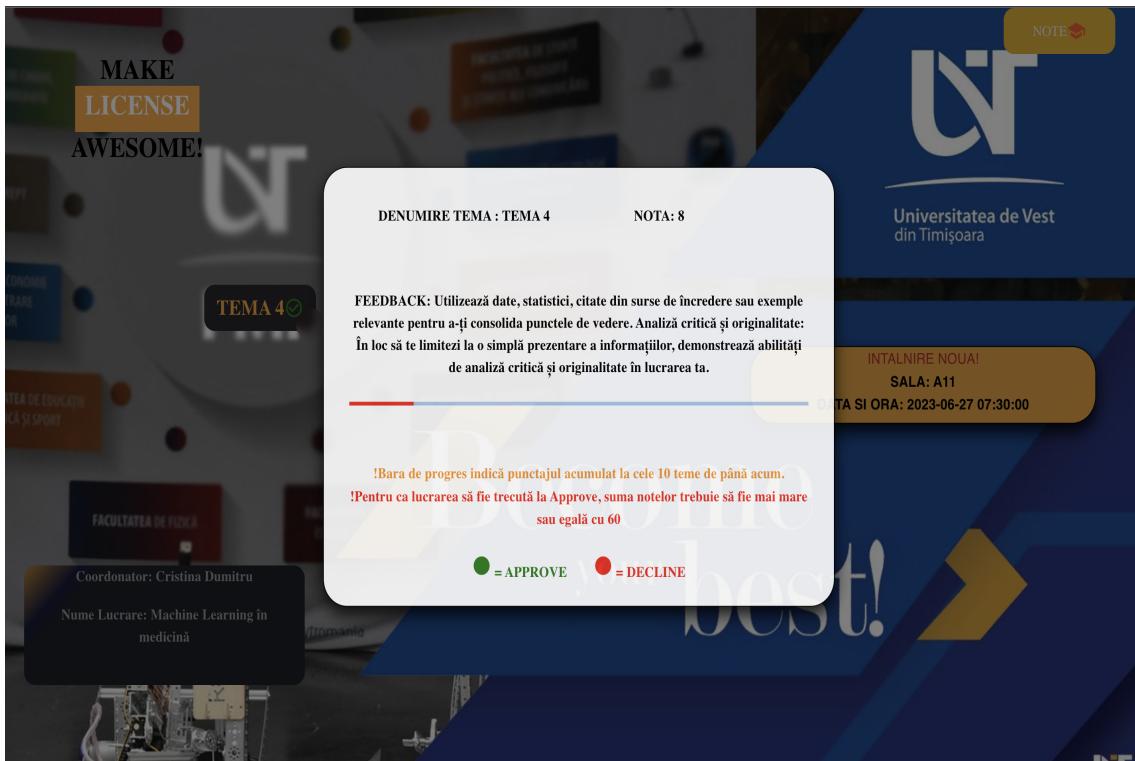


Figura 4.5: Interfață student(Bara progres/Note/Feedback)

De asemenea, există o secțiune dedicată temelor, numită 'NOTE', ilustrată în figura 4.5, care au fost trimise pentru notare, și care au fost deja evaluate, afișându-se și nota obținută de către student. Pentru a facilita urmărirea progresului și evaluarea lucrărilor, studenților le este pusă la dispoziție o bara de progres (progress bar) care indică nivelul de avans al lucrării de licență.

Această bara de progres reflectă completitudinea lucrării și poate fi actualizată pe măsură ce studenții își finalizează diferitele etape sau secțiuni ale lucrării lor. Cu ajutorul acesteia, studenții pot vizualiza cu ușurință cât de aproape se află de finalizarea lucrării de licență și cât mai au de lucru. Această funcționalitate oferă o orientare clară și ajută la gestionarea eficientă a timpului și a eforturilor pentru a finaliza cu succes lucrarea. Pe măsură ce studenții înaintează în lucrarea lor, ei pot observa cum barele de progres se umplu treptat, oferindu-le satisfacția de a vedea cum se apropie de finalizare.

Un aspect important de menționat este că, în cazul în care suma notelor primite pe parcursul anului, acordate de profesorul coordonator, nu depășește 60, lucrarea va fi automat respinsă, iar studentul nu va putea susține examenul de licență. Această regulă asigură că studenții își dedică suficient timp și efort pentru a obține rezultate academice bune pe parcursul anului și îi motivează să se implice activ în procesul de învățare și cercetare.

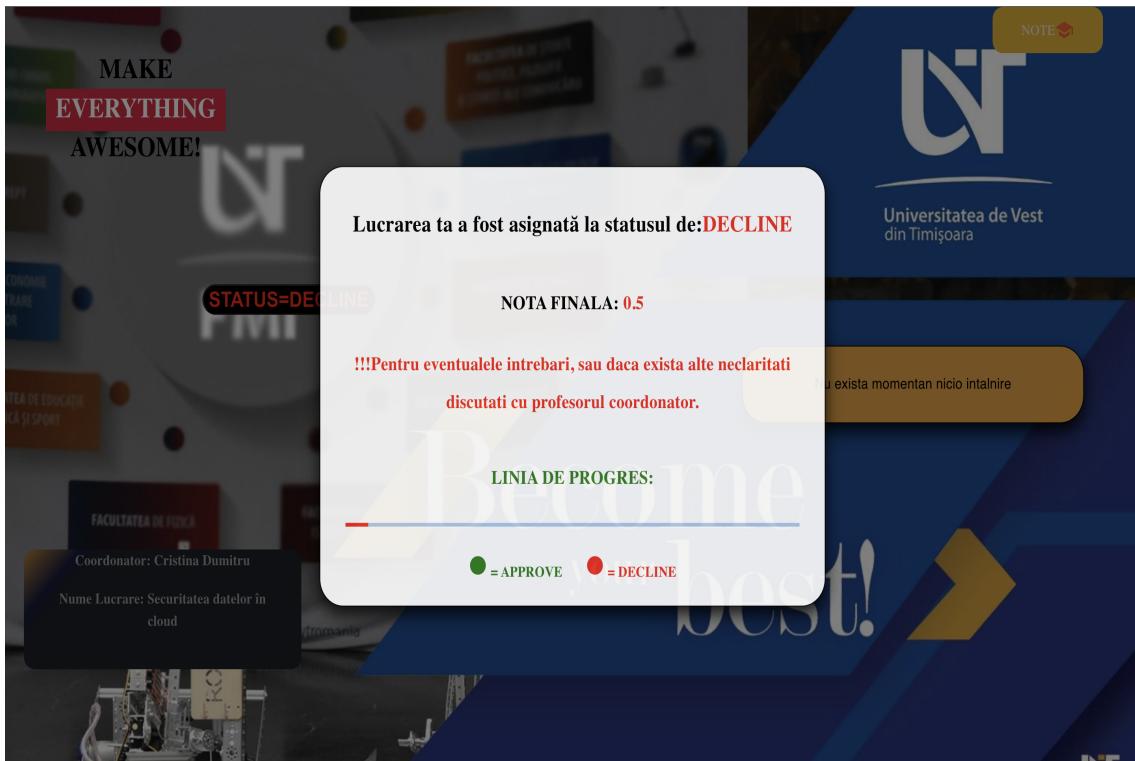


Figura 4.6: Interfață student(Status Final)

În cazul în care suma notelor obținute de student nu a depășit pragul minim pentru a putea prezenta lucrarea comisiei, sau profesorul a considerat că lucrarea prezentată nu este adecvată pentru sesiunea de licență, starea lucrării va fi marcată ca 'DECLINE', iar în interfață afișată studentului vor fi prezentate detaliile relevante care atestă această situație. În figura 4.6, este prezentată o imagine a vizualizării pentru student, unde se poate observa că butonul anterior denumit 'Tema' a fost dezactivat și oferă informații referitoare la statusul lucrării. De asemenea, în secțiunea 'Note' apare un mesaj care sugerează că studentul nu a atins nivelul minim de promovare, fiind afișate și nota finală obținută și o linie de progres care indică gradul de reușită.

### 4.3.3 Ecran principal(Profesor)

În interfața profesorului, ilustrată în figura 4.7, se observă un design intuitiv și ușor de gestionat. Bara de sus conține patru butoane principale, fiecare atribuit unei funcționalități specifice. Pentru pictograma "Home", aceasta are rolul de a redirectiona utilizatorul către pagina principală, oferindu-i un acces rapid la panoul de control central.



Figura 4.7: Interfață profesor(Pagina principală)

Butoanele "Istoric lucrări" permit profesorului să acceseze o pagină dedicată care conține istoricul lucrărilor. Prin apăsarea acestui buton, utilizatorul este redirecționat către acea pagină, unde poate vizualiza informații detaliate despre lucrările anterioare și poate efectua acțiuni specifice, cum ar fi descărcarea sau revizuirea lucrărilor. Butonul de adăugare a temei permite profesorului să introducă o nouă temă în sistem. Prin apăsarea acestui buton, utilizatorul este direcționat către o pagină unde poate completa detaliile temei, cum ar fi descrierea, cerințele și deadline-ul.

Nu în ultimul rând, butonul pentru programarea unei săli de curs pentru o întâlnire permite profesorului să programeze o locație fizică unde poate avea loc o întâlnire cu studenții. Acest buton deschide o pagină în care utilizatorul poate selecta data și ora întâlnirii, precum și sala de curs disponibilă. Interfața include și un "search bar" care facilitează căutarea specifică a unui student după nume. Aceasta oferă profesorului posibilitatea de a găsi rapid și eficient informațiile legate de un student anume. Mesajul de "Remainder" apare pe ecran și este afișat timp de 10 secunde, cu scopul de a reaminti despre întâlnirea programată, astfel încât utilizatorul să nu o uite. Acest mesaj efemer asigură o notificare vizibilă și temporară, care atrage atenția utilizatorului asupra evenimentului planificat și îi oferă timp suficient să ia în considerare această informație importantă.

În centrul atenției se află temele studentilor, afișate în mod prominent în interfața profesorului, fiind afișată doar o previzualizare a acesteia. Pe lângă acestea, există și câteva funcționalități suplimentare esențiale. Un element cheie în interfața profesorului este bara de progres a studentului, care oferă profesorului o perspectivă constantă asupra stadiului în care se află lucrarea studentului. Aceasta îi permite profesorului să urmărească progresul studentului pe parcursul evaluării și să ia decizii importante cu privire la nota finală și încadrarea lucrării în categoria de aprobată sau respingere pentru prezentare. Bara de progres furnizează o imagine vizuală și concisă a evoluției lucrării, oferindu-i profesorului informațiile necesare pentru a gestiona evaluarea și a lua decizii într-un mod eficient.

Butonul "Vizualizare" permite profesorului să deschidă și să vizualizeze lucrarea unui student. Prin apăsarea acestui buton, lucrarea se deschide într-o fereastră dedicată, unde profesorul poate examina conținutul și formatarea lucrării. De asemenea, există și opțiunea de a descărca lucrarea pentru a o accesa offline. În plus, există butoanele "Approve" și "Decline" care permit profesorului să aprobe sau să respingă lucrarea unui student. Aceste butoane sunt asociate cu procesul de evaluare și permit profesorului să ofere un feedback final rapid asupra lucrării și să decidă dacă aceasta îndeplinește cerințele și criteriile pentru a fi prezentată comisei sau necesită ajustări suplimentare.

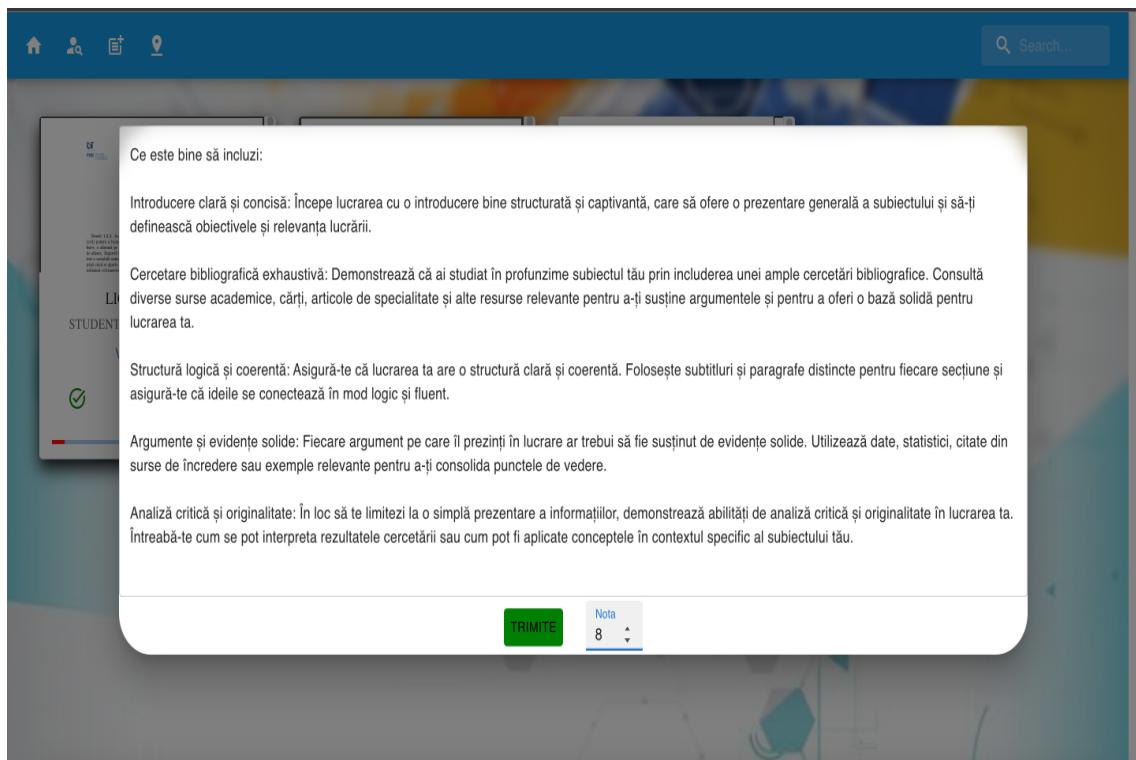


Figura 4.8: Interfață profesor(Buton 'Comentarii')

De asemenea, interfața include o secțiune dedicată feedback-ului, prezentată în figura 4.8 notarea temei, fiind reprezentată de butonul 'Comentarii'. Aici, profesorul poate oferi un feedback detaliat și specific pentru fiecare lucrare evaluată. În cadrul acestei secțiuni, profesorul are posibilitatea de a oferi un feedback detaliat și specific pentru fiecare lucrare evaluată. Acest feedback poate include sugestii de îmbunătățire,

evidențierea punctelor forte și a celor slabe ale lucrării, precum și alte observații relevante. După analiza atentă a lucrării, profesorul acordă și o notă corespunzătoare, astfel încât fiecare aspect al lucrării să fie bine verificat și evaluat în mod corect.

După apăsarea butonului ”Trimite”, nota acordată și detaliile introduse ca și comentarii vor fi trimise către student. În același timp, un e-mail automatizat va fi generat, conținând aceste informații, pentru a informa rapid studentul cu privire la noutățile referitoare la statusul temei trimise pentru evaluare. Astfel, studentul va fi informat prompt și eficient în legătură cu rezultatele și feedback-ul asociate lucrării sale.

Prin integrarea acestor funcționalități, interfața profesorului facilitează procesul de evaluare și gestionare a lucrărilor studenților, oferindu-le profesorilor un control eficient asupra evaluării și comunicării feedback-ului către studenți.



Figura 4.9: Interfață profesor(Istoric)

Atunci când butonul ”Istoric” este apăsat, se va deschide automat o pagină care conține istoricul lucrărilor de licență care au fost deja evaluate și notate, alături de nota acordată, evidențiată în figura 4.9. De asemenea, vor fi afișate și lucrările care încă nu au fost evaluate și așteaptă feedback-ul din partea profesorului.

Pe lângă aceasta, în continuare, butonul ”Vizualizare” este prezent, având aceleași caracteristici. Prin apăsarea acestui buton, profesorul poate vizualiza lucrarea de licență într-o fereastră dedicată, examinând conținutul și formatarea acesteia. De asemenea, profesorul are posibilitatea de a descărca lucrarea pentru a o accesa offline sau pentru a o păstra în arhivă.



Figura 4.10: Interfață profesor(Creare tema)

Este prezentă și interfață unde se poate crea o temă, în figura 4.10, această opțiune devenind disponibilă după apăsarea butonului ”Adaugă temă”. După aceasta, se afișează un formular de creare a temei, adaptat în funcție de specializarea selectată de către profesor. În cadrul acestui formular, sunt necesare următoarele informații:

- Denumirea temei: Profesorul trebuie să introducă un titlu relevant și sugestiv pentru tema propusă.
- Deadline-ul: Aici se specifică data și ora limită până la care studentii trebuie să transmită lucrările finalizate.
- Descrierea detaliată a temei: Această secțiune permite profesorului să ofere informații clare și complete cu privire la cerințele, obiectivele și aspectele relevante ale temei. Aici pot fi incluse detalii despre metodele și resursele necesare, așteptările referitoare la conținut și format, precum și orice alte informații relevante care facilitează înțelegerea și abordarea eficientă a temei.
- Alegerea specializării: Aici trebuie aleasă dintr-un meniu derulant (dropdown list) specializarea căreia tema urmează să îi fie asignată, pentru ca detalii să fie corect transmise studenților.

Prin completarea și trimiterea acestui formular, profesorul poate crea o temă cu toate detaliile necesare, permitându-le studenților să o acceseze și să înceapă procesul de lucru în mod adecvat.

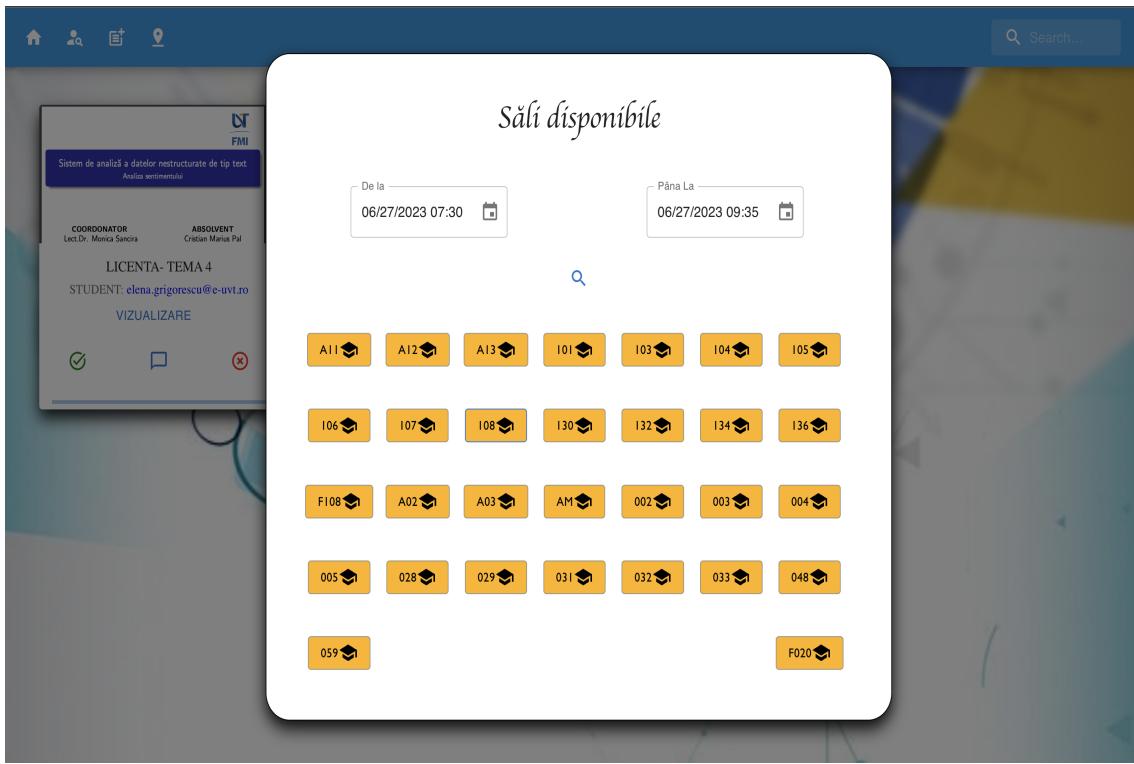


Figura 4.11: Interfață profesor(Rezervare sală)

O ultimă funcționalitate esențială este rezervarea unei săli, ilustrată în figura 4.11 de curs în vederea programării unei întâlniri fizice cu studentii, pentru a discuta diverse subiecte legate de temele propuse sau viitoarele teme. Pentru a facilita acest proces, am introdus în interfață un buton denumit ”Rezervare sală”.

După apăsarea butonului, profesorul are posibilitatea de a selecta data și ora dorite pentru rezervarea sălii. Pentru a simplifica procesul de selecție, am inclus un calendar interactiv, unde data și ora pot fi alese prin simpla selectare a opțiunii corespunzătoare. Astfel, profesorul poate alege cu ușurință momentul convenabil pentru întâlnirea cu studentii. Este important de menționat că durata rezervării sălii nu trebuie să depășească 4 ore, iar în cazul în care se selectează un interval mai lung, sistemul va afișa o eroare și va solicita profesorului să ajusteze durata rezervării. După ce intervalul orar de rezervare a fost selectat, profesorul poate apăsa icona ”Căutare” (Search), iar în urma acestei acțiuni vor fi afișate sălile disponibile în intervalul selectat. Profesorul poate alege sala potrivită din lista rezultată, asigurându-se astfel că rezervarea este realizată corect și că sala va fi disponibilă pentru întâlnirea programată cu studentii.

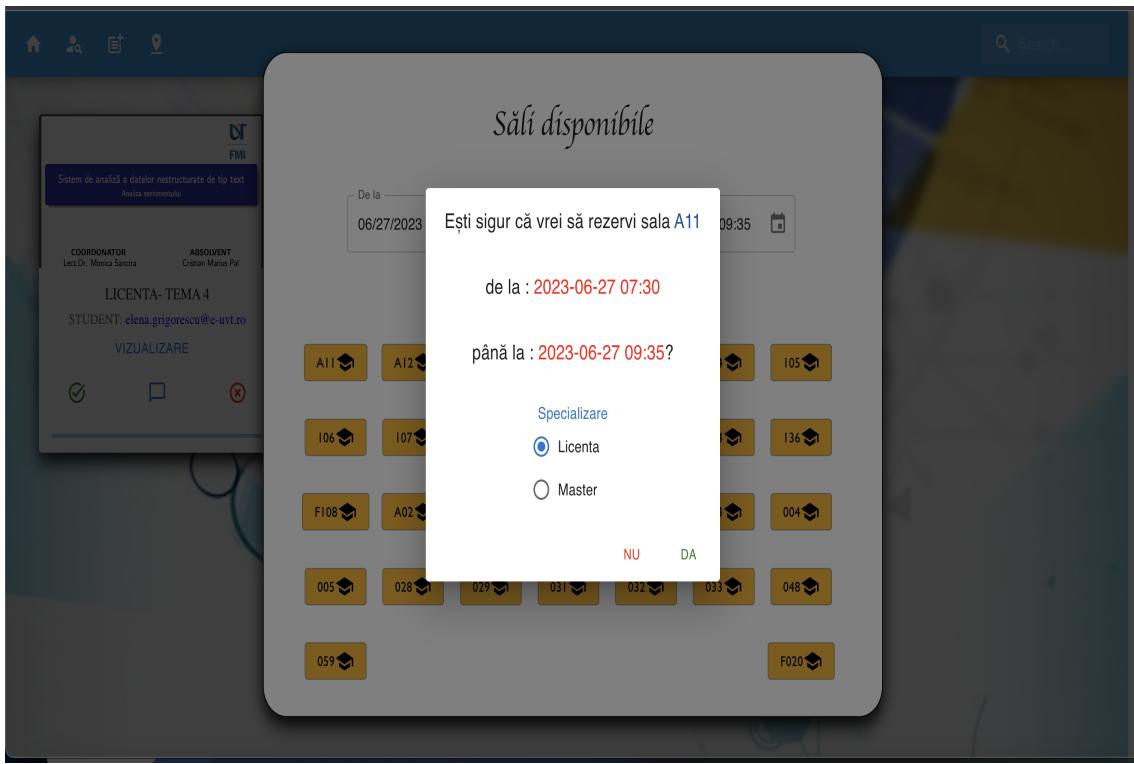


Figura 4.12: Interfață profesor(Rezervare sală-Acord)

După ce profesorul alege sala dorită din lista propusă, acesta primește o întrebare de confirmare pentru a se asigura că datele introduse sunt corecte, funcționalitatea este prezentată în figura 4.12. De asemenea, profesorul are opțiunea de a selecta specializarea pentru care informațiile legate de întâlnire vor fi transmise, prin intermediul unei casete de selectare (checkbox).

Această etapă suplimentară de confirmare și selectare a specializării are rolul de a evita erorile și asigură corespondența corectă a datelor cu privire la întâlnire.

După ce butonul "DA" este apăsat pentru a confirma rezervarea unei săli de curs, informațiile de rezervare trebuie introduse în baza de date pentru a fi stocate și ulterior afișate în interfața studentului. Aceasta implică utilizarea unui script PHP pentru a prelua datele din formularul de rezervare și a le procesa în vederea inserării în baza de date.

## **4.4 Concluzii și direcții viitoare**

### **4.4.1 Direcții viitoare**

Odată ce funcționalitățile aplicației au fost verificate și validate cu succes, atingând obiectivele prestabilite, următorul pas este să se stabilească direcțiile viitoare de dezvoltare. Aceste direcții au ca scop asigurarea unor funcționalități și servicii mai calitative, în vederea îmbunătățirii continue a aplicației.

#### **Introducere livechat**

În prezent, aplicația oferă doar posibilitatea profesorilor de a acorda feedback studenților prin intermediul funcționalității de evaluare. Totuși, adăugarea unui live chat în aplicație ar aduce multiple beneficii și ar îmbunătăți comunicarea bidirectională între profesori și studenți. În primul rând live chat-ul ar permite studenților să răspundă direct la feedback-ul primit de la profesori. Astfel, s-ar crea o comunicare interactivă și bidirectională, unde studenții pot adresa întrebări, solicita clarificări sau oferi informații suplimentare. Acest lucru ar facilita înțelegerea mai bună a feedback-ului și ar permite studenților să răspundă în mod prompt și adecvat. Totodată ar crea un mediu propice pentru un dialog constructiv între profesori și studenți. Această ar putea discuta în detaliu despre aspectele lucrării și ar putea dezvolta soluții și idei împreună. Astfel, ar fi stimulată colaborarea și ar fi favorizată îmbunătățirea lucrării prin intermediul unei comunicări mai interactive și deschise.

#### **Disponibilitate întâlnire fizică**

O altă îmbunătățire ar consta în adăugarea unor butoane de tip "like" și "dislike" pentru studenți, pentru a le oferi posibilitatea de a reacționa la propunerile de întâlniri făcute de profesori. Această funcționalitate ar permite studenților să indice dacă pot să participe sau nu la o anumită întâlnire.

Prin adăugarea acestor butoane de reacție, studenții ar putea exprima rapid și simplu intenția lor referitoare la participarea la o anumită întâlnire programată de către profesor. Aceasta ar aduce mai multă claritate și eficiență în procesul de organizare a întâlnirilor, atât pentru profesori, cât și pentru studenți.

Profesorii ar putea folosi această funcționalitate pentru a primi feedback rapid și agregat cu privire la disponibilitatea studenților pentru anumite întâlniri. Aceasta le-ar oferi o perspectivă mai clară asupra numărului de participanți și ar putea facilita procesul de planificare și gestionare a întâlnirilor. De asemenea, ar putea oferi profesorilor oportunitatea de a face ajustări în programările lor, ținând cont de preferințele și disponibilitățile studenților.

### **4.4.2 Concluzii**

Scopul acestui proiect a fost să ofere o metodă mai usoară și mai eficientă pentru verificarea lucrărilor de licență. Aplicația pentru monitorizarea lucrărilor de licență reprezintă o soluție eficientă și completă pentru profesori și studenți, oferindu-le o platformă simplă și accesibilă pentru comunicare și colaborare. Prin intermediul acestei aplicații, procesul de evaluare și îmbunătățire a lucrărilor de licență devine mult mai ușor și mai eficient.

Profesorii beneficiază de avantajul de a putea vizualiza și evalua lucrările încărcate de către studenți, având acces la informații detaliate despre fiecare student și urmărind progresul individual al lucrărilor. Această funcționalitate le permite să ofere feedback detaliat și relevant, ajutându-i pe studenți să își îmbunătățească proiectele într-un mod mai eficient.

Aplicația facilitează, de asemenea, programarea întâlnirilor între profesori și studenți, eliminând nevoia de comunicare suplimentară prin e-mail sau alte canale. Prin intermediul aplicației, profesorii pot programa rapid și eficient întâlniri, selectând data și ora dorită. Această funcționalitate reduce timpul și efortul necesare pentru stabilirea programărilor și facilitează gestionarea întâlnirilor multiple cu studenții.

Prin urmare, aplicația pentru monitorizarea lucrărilor de licență reprezintă o soluție inovatoare și utilă, care aduce multiple beneficii atât pentru profesori, cât și pentru studenți. Prin intermediul acestei platforme, comunicarea și colaborarea devin mai eficiente, iar procesul de evaluare și îmbunătățire a lucrărilor de licență devine mai ușor și mai transparent.

# Bibliografie

- [1] MySQL Documentation.
- [2] Mailtrap- What is it?
- [3] Google Classroom.
- [4] Moodle- Modular Object-Oriented Dynamic Learning Environment.
- [5] Visual Studio- About.
- [6] What Is React?
- [7] What is PHP? The PHP Programming Language Meaning Explained.
- [8] Material-UI and How to Use It.
- [9] Conf. Dr. Cristina Mîndruță. *Inginerie Software - Curs.* Catedra de Inginerie Software a Facultății de Matematică și Informatică, Universitatea de Vest din Timișoara, 2021.