



Turkey
Karabük University
Faculty of Engineering
Computer Engineering
Department

SENIOR PROJECT

University Automation System

Supervisor: Prof. Dr. İlhami Muharrem ORAK

2023-2024 fall semester

**2010205590 - Mohamad ZUBI
1910205563 - Muhannad SALKINI
2010213598 - Nisreen BOUTA
2010213538 - Roula KOURANI**

Contents

System Sections	4
Lecturer Profile (Student Point of View):	4
Lecturer Profile (Private for the Lecturer Only):	4
Student Profile:.....	4
Section Details	5
Lecturer Profile (Private for the Lecturer Only):	5
Student Profile:.....	6
Benefits of the System	7
For Students:	7
For Lecturers:.....	7
System Details	8
Common Pages:.....	8
Login:	8
Lecturer Side (Lecturer Pages):	8
1. Lecturer Sidebar:	8
2. Lecturer Navbar:.....	9
3. Lecturer Home Page:.....	9
4. Lecturer Profile:.....	9
Student Side (Student Pages):.....	10
1. Student Sidebar:	10
2. Student Navbar:	10
3. Student Home Page:.....	11
4. Student Profile:.....	11
Admin Side:.....	11
1. Login:.....	11
2. Admin Panel:	11
Use Case Diagrams	12
Login Page (For Users):	12
Navbar:	12
Lecturer Sidebar:	13
Lecturer Home:.....	13
Lecturer Courses:.....	14

Student Sidebar:	14
Student Home:	15
Student Courses:	15
Admin Login:	16
Admin Panel:	16
System Functionality	17
Lecturer Schedule:	17
Academic Information:	18
Active Courses:	19
Past Courses:	20
Chat to Communicate with Students:	20
Receiving Student Requests and Replies:	22
Senior Project Receiving Reports	23
Student Academic Information:	24
Department Profile:	26
AI Chatbot	26
Database Design	28
User Interface Design	29
Low-Fidelity Wireframes	29
Experience Design (Mockups & High-Fidelity Prototypes)	35
Ai Chat Bot	41
Integration:	42
Collected PDFs (Rules):	42
Web Code Implementation	43
Overview:	43
File Structure:	44
Code Snippets:	45

System Sections

Lecturer Profile (Student Point of View):

- Lecturer schedule.
- Academic information.
- Courses
 - Active courses.
 - Past courses.
- Chat to communicate with students (two ways).

Lecturer Profile (Private for the Lecturer Only):

- Courses
 - Uploading notes
 - Send announcements.
 - Uploading external links.
 - Student information per course (automated from OBS).
 - Each course analytics (passed students, failed students, number of students entered the exam etc.)
- Receiving students requests and replying to them.
- Receiving messages from students.
- Senior project receiving reports weekly.

Student Profile:

- Student academic information (automated from OBS)
(student schedule, exam schedule, GPA).
- Send message to lecturer.
- Send requests to lecturer (recommendation letter, exam grade objection, internship request).
- Department profile (department's lecturers, courses, top 10 students, years of department inserting under each year the related courses per each semester, president person).
- Ai Chatbot.
- Courses list.

For each course:

- Course information (automated).
- files (notes, slides, previous exams, books).
- course related announcements.

Section Details

Lecturer Profile (Private for the Lecturer Only):

- **Lecturer Schedule:** *POV: As a lecturer* You log into the system and access your private dashboard. You see your schedule for the upcoming week, which includes details of your classes, and office hours. (automated)
- **Academic Information:** *POV: As a lecturer* You navigate to your academic information section, where you can view and edit your academic background, qualifications, and research interests. This information is accessible to students who may want to learn more about your expertise.
- **Courses:** *POV: As a lecturer* you can see course analytics for each one which will show the number of students who passed, failed, entered the exam, entered the makeup exam, the average of the class. In addition to Uploaded Notes and External Links where You share lecture notes, presentation slides, or other educational resources, external web links, articles, and resources relevant to the course. Students can access and download these materials. Also, you have access to automated student information, including attendance records, grades, and contact details, which help you keep track of your students' progress. Plus, you send course-related announcements to all students in the class. These announcements can include important updates, changes in schedules, or general reminders.
 - **Active Courses:** *POV: As a lecturer* In the Courses section, you view the list of courses you are currently teaching for the semester. You can access course materials, student information, including attendance records, grades, and contact details, which help you keep track of your students' progress and other relevant details for each class. (automated)
 - **Past Courses:** *POV: As a lecturer* You review your previous semesters' courses and access important data such as course outlines, links, files, and performance statistics. (not automated)
- **Chat to Communicate with Students (Two Ways):** *POV: As a lecturer* You use the chat feature to communicate with students individually. Students can reach out to you with questions, and you can reply to them. (not automated)
- **Receiving Student Requests and Replies:** *POV: As a lecturer* You receive requests from students for recommendation letters, exam grade objections, and internship opportunities. You can review and respond to these requests through the platform. (not automated)
- **Senior project receiving reports weekly:** the senior project student will be allowed to submit their reports weekly, and the lecturer will be notified, and the assignments are kept in the system. (not automated)

Student Profile:

- **Student Academic Information (Automated from OBS):** *POV: As a student* You access your academic profile, which includes your schedule, exam schedule, and GPA. This information helps you stay organized and monitor your academic progress. (automated)
- **Send Message to Lecturer:** *POV: As a student* You use the messaging feature to send messages to your lecturer. You can ask questions, seek clarification, or request additional support as needed. (not automated)
- **Send Requests to Lecturer (Recommendation Letter, Exam Grade Objection, Internship Request):** *POV: As a student* You send requests to your lecturer, such as requests for recommendation letters, objections to exam grades, or applications for internships. These requests are managed through the platform. (not automated)
- **Department Profile (Lecturers, Courses, Years of Department Inserting):** *POV: As a student* You explore the department's profile to learn about the lecturers, courses offered, and the top-performing students in your department. You can also view the department's history and courses offered in each semester. This information helps you make informed decisions about your academic journey. (automated)
- **Ai Chatbot:** *POV: As a student* You can see a chatbot at the bottom of the page. And can ask the bot questions that are related to the university, for example university rules and it will answer you.

These scenarios demonstrate how the platform serves both lecturers and students, facilitating effective communication and enhancing the educational experience.

Benefits of the System

For Students:

The platform offers a comprehensive set of benefits aimed at enhancing their academic journey. Through the automated Student Academic Information section, students gain easy access to their schedules, exam schedules, and GPA, providing a centralized hub for organizational purposes and continuous monitoring of their academic progress. The messaging feature allows students to communicate directly with lecturers, fostering a dynamic and interactive learning environment. Additionally, the platform facilitates the submission of various requests, including recommendation letters, exam grade objections, and internship applications, streamlining the process and ensuring efficient communication between students and lecturers. Exploring the department profile further empowers students by offering insights into lecturers, courses, and department history, aiding informed decision-making regarding their academic pursuits. The inclusion of an AI chatbot adds another layer of support, allowing students to quickly access information about university rules and guidelines.

For Lecturers:

The platform significantly simplifies administrative tasks and enhances their teaching capabilities. The automated Lecturer Schedule ensures efficient time management, providing a clear overview of upcoming classes and office hours. The Academic Information section enables lecturers to showcase their expertise, fostering transparency for students interested in their background, qualifications, and research interests. Course analytics offer valuable insights into student performance, allowing lecturers to identify areas for improvement and tailor their teaching approach. The ability to share lecture notes, presentation slides, and external resources enhances the learning experience, making educational materials easily accessible to students. The platform's automated features, such as access to student information, attendance records, and grades, facilitate streamlined monitoring of student progress. The chat feature allows personalized communication with students, fostering a supportive learning environment. Furthermore, the platform assists in managing requests, including recommendation letters and internship opportunities, streamlining the process for both lecturers and students. Overall, the platform serves as a comprehensive tool for lecturers, promoting effective communication, efficient administration, and an improved teaching and learning experience.

System Details

Common Pages:

Login:

- Authentication
 - Username
 - Password
- Reset Password

Lecturer Side (Lecturer Pages):

1. Lecturer Sidebar:

- Profile:
 - See the profile
- Courses:
 - View the courses list the lecturer gives (With filter option to view the past semesters).
In each course page:
 - Uploading notes.
 - Add external links.
 - See course analytics (Passed students, failed students, number of students entered the exam etc.)
 - See the students list.
 - Lecture settings (Change course questions & messages preferences)
- Messages (One to one chat):
 - View the messages (Sorted by date, include search and filter options).
 - See the archived chats.
In each chat:
 - Read & reply to the message.
 - See the user profile
- Requests (Student requests):
 - View the received requests (Sorted by date, include search and filter options).
For each request:
 - View & answer the request (Approve or reject with a message).

- Senior project groups:
 - See the senior project groups.

For each group:

 - Check the reports.
 - See the group members.
 - Send a note (Request).
- Notifications:
 - View the received notifications. (The lecturer will get notifications for new messages, requests, questions, reports, system updates etc...
The notifications will be sorted by date, include search and filter options)
- Settings:
 - Change password
 - Change notification preferences
 - Change the language

2. Lecturer Navbar:

- Profile (Drop list):
 - See the profile
 - Open the settings
 - Logout
- Language:
 - Change the system language (EN - TR).
- Return home button

3. Lecturer Home Page:

- See the current courses
- See today's schedule
- Incoming notifications (messages, requests, reports, activates, etc...)

4. Lecturer Profile:

- Lecturer information (Name, picture, department).

Student Side (Student Pages):

1. Student Sidebar:

- Profile:
 - See the profile.
- Courses:
 - View the courses (With filter option to view the past semesters). For each course:
 - View lecturer profile
 - Course Information
 - See the notes & external links
- Department:
 - See the department details and lectures
- Schedules:
 - Exam schedule
 - Lecture schedule
 - Academic schedule
- Messages:
 - View the messages. (Sorted by date, include search and filter options)
- Requests:
 - View the sent requests. (Sorted by date, include search and filter options)
- Notifications:
 - View and read the notifications. (The student will get notifications for new messages, responses, system updates etc. The notifications will be sorted by date, include search and filter options)
- Settings:
 - Change password
 - Change notification preferences
 - Change the language
 - Set a profile picture

2. Student Navbar:

- Profile (Drop list):
 - See the profile
 - Open the settings
 - Logout
- Language:
 - Change the system language (EN - TR).
- Return home button

3. Student Home Page:

- See the current courses
- See today's schedule
- Announcements
- Ai Chatbot

4. Student Profile:

- Student information (Name, picture, department).
 - All past year's info

Admin Side:**1. Login:**

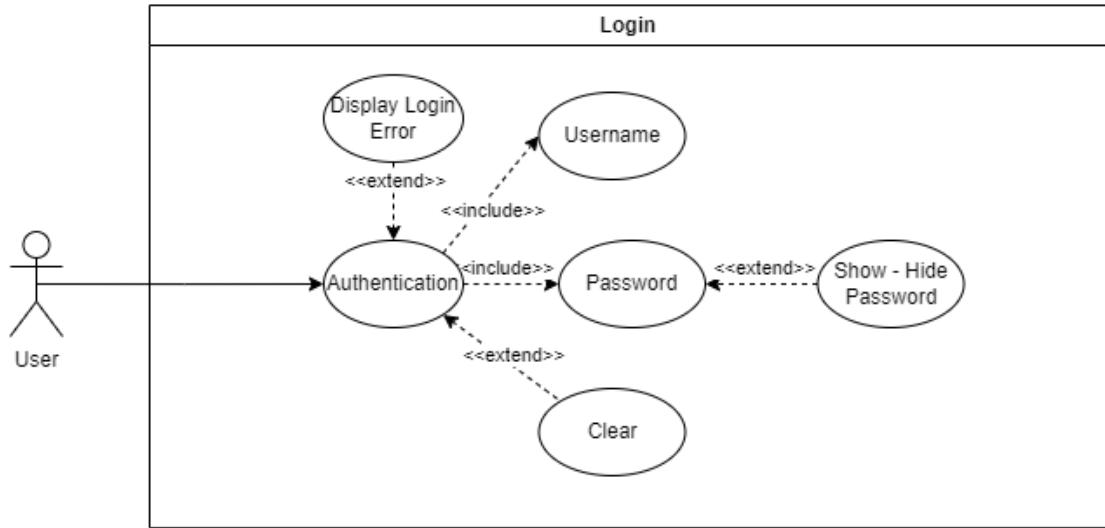
- Authentication
 - Username
 - Password
- Reset Password

2. Admin Panel:

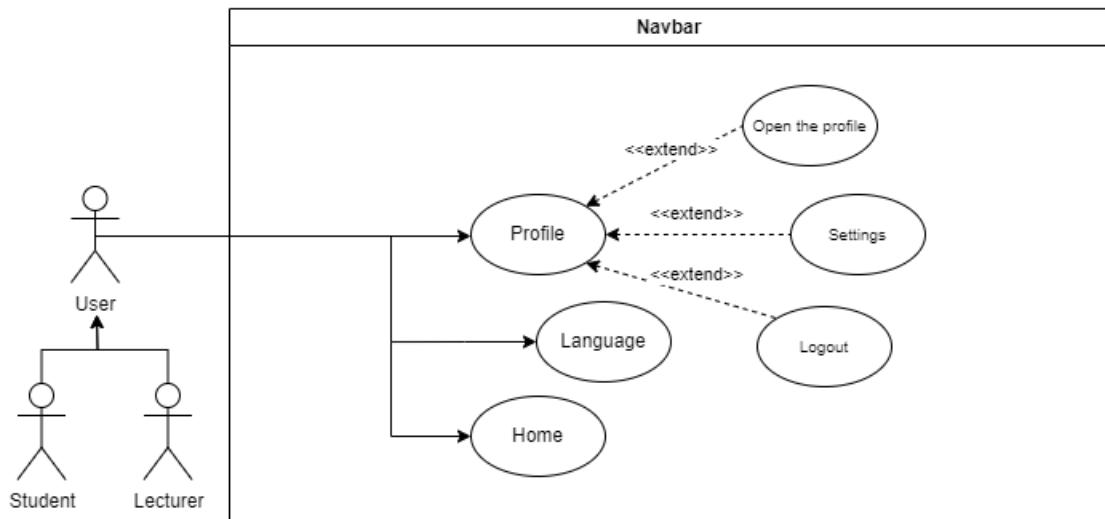
- See the Students (Students list with each student information include filter options.)
- See the Lecturers
- See the Faculties and departments
- Technical support (Complaints and Requests)
- Announcements (Send activity, system etc. announcements)

Use Case Diagrams

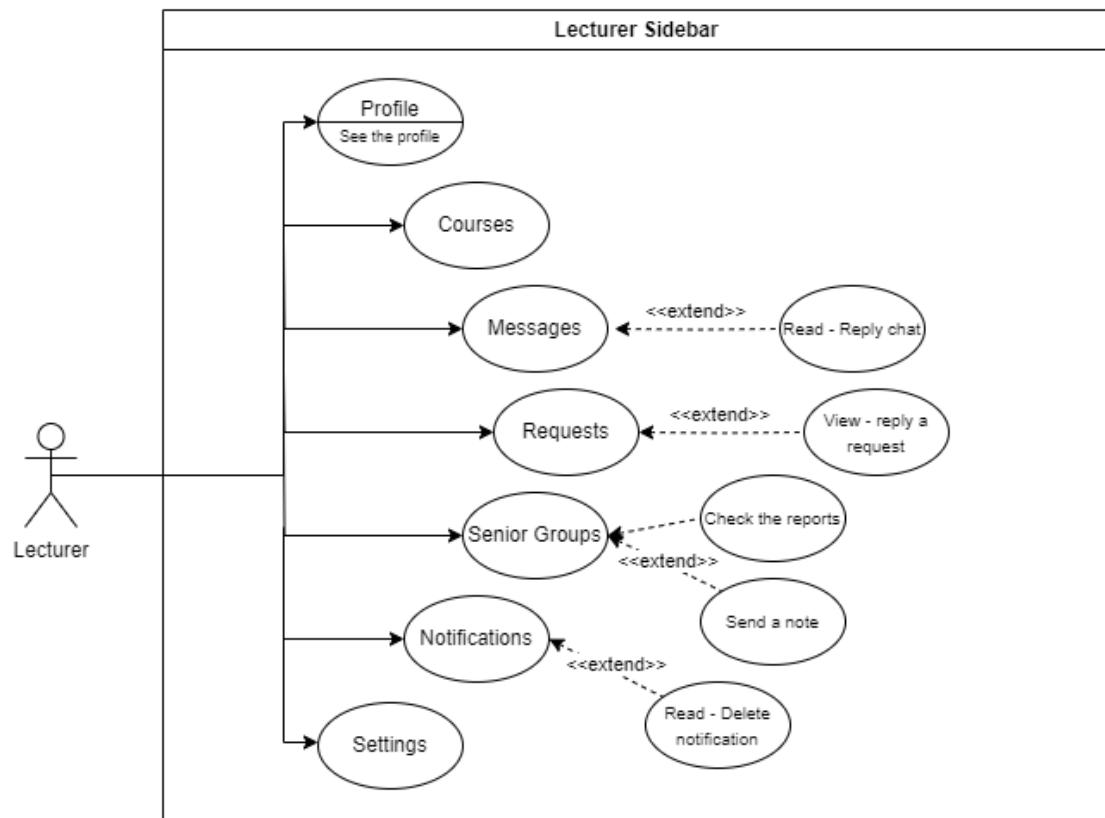
Login Page (For Users):



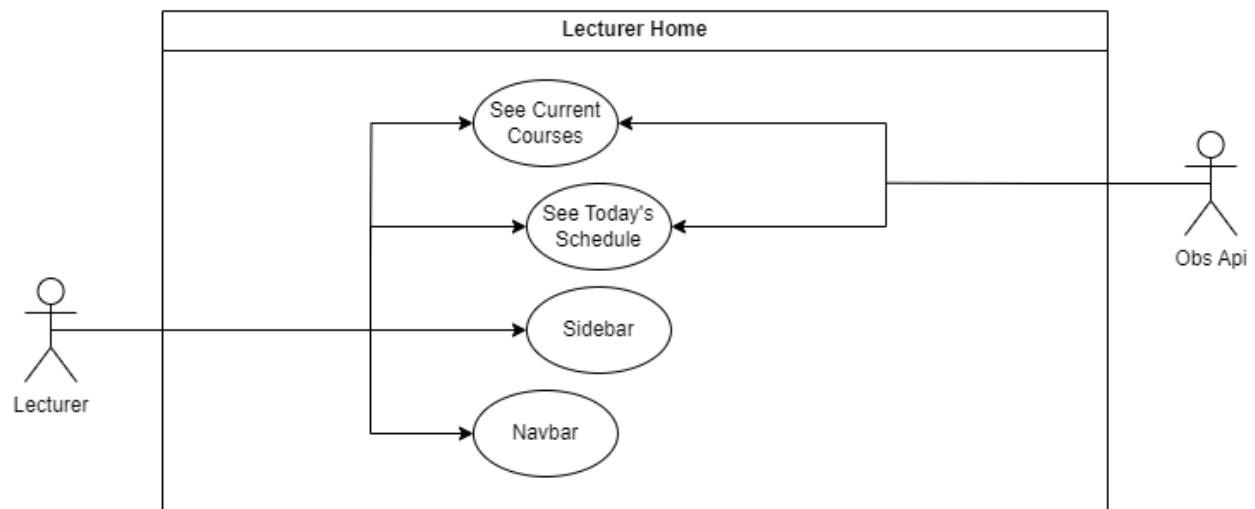
Navbar:



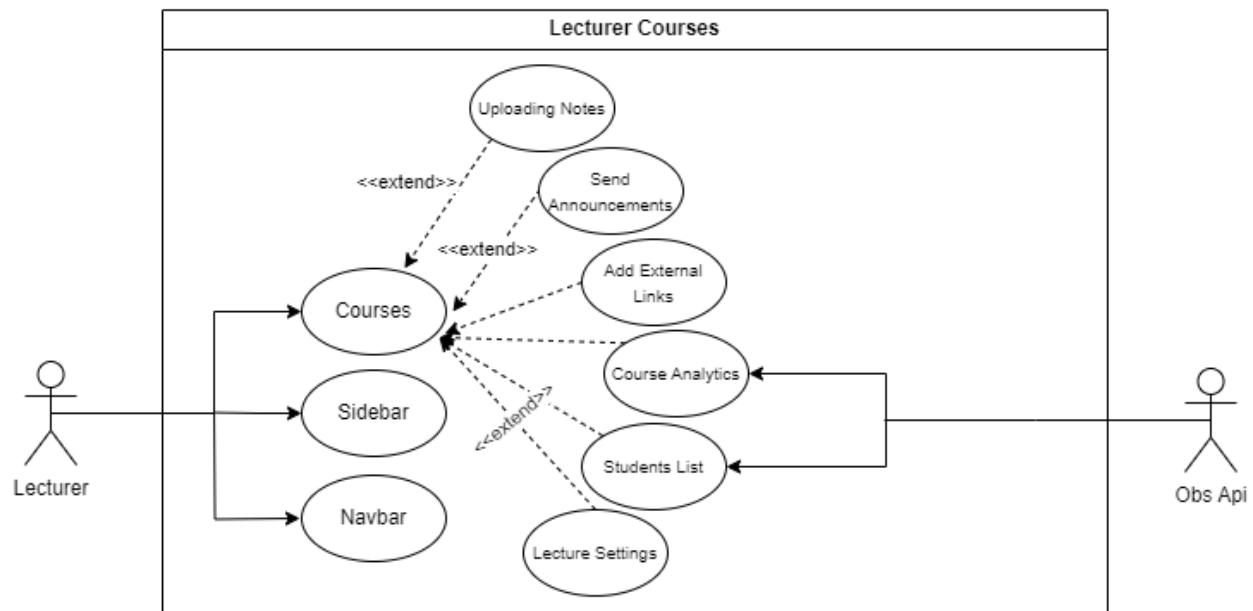
Lecturer Sidebar:



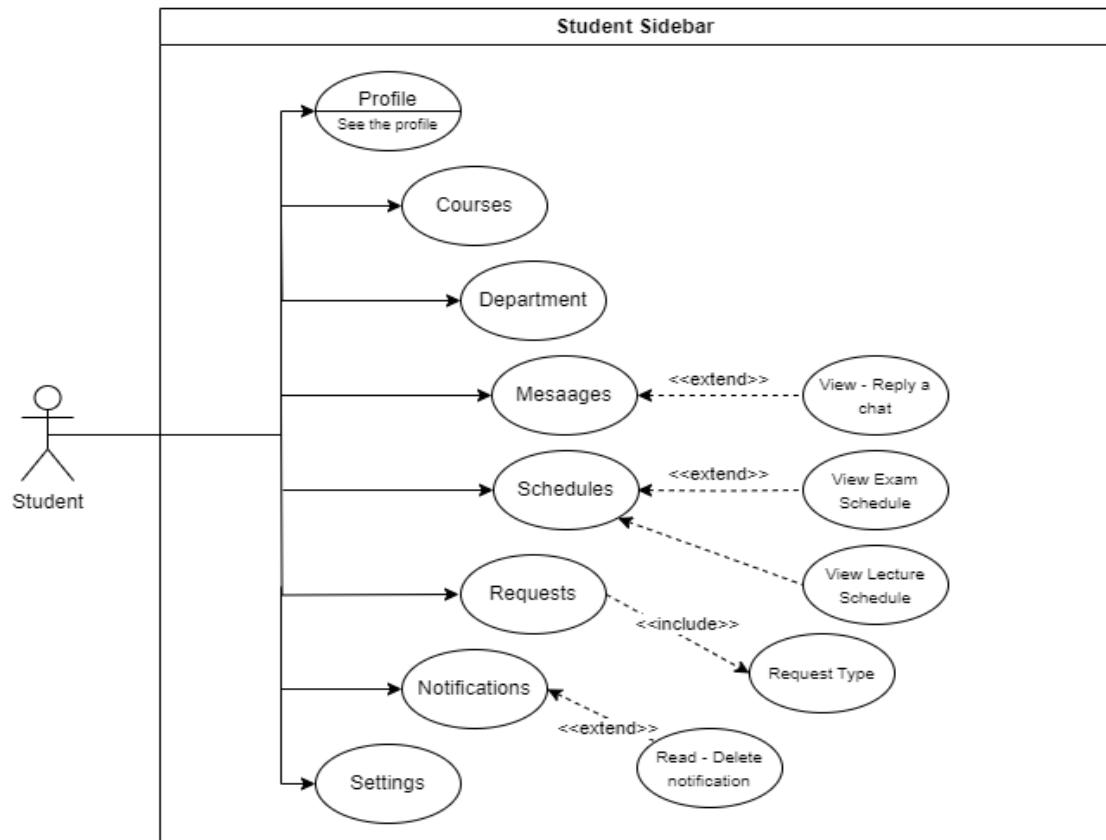
Lecturer Home:



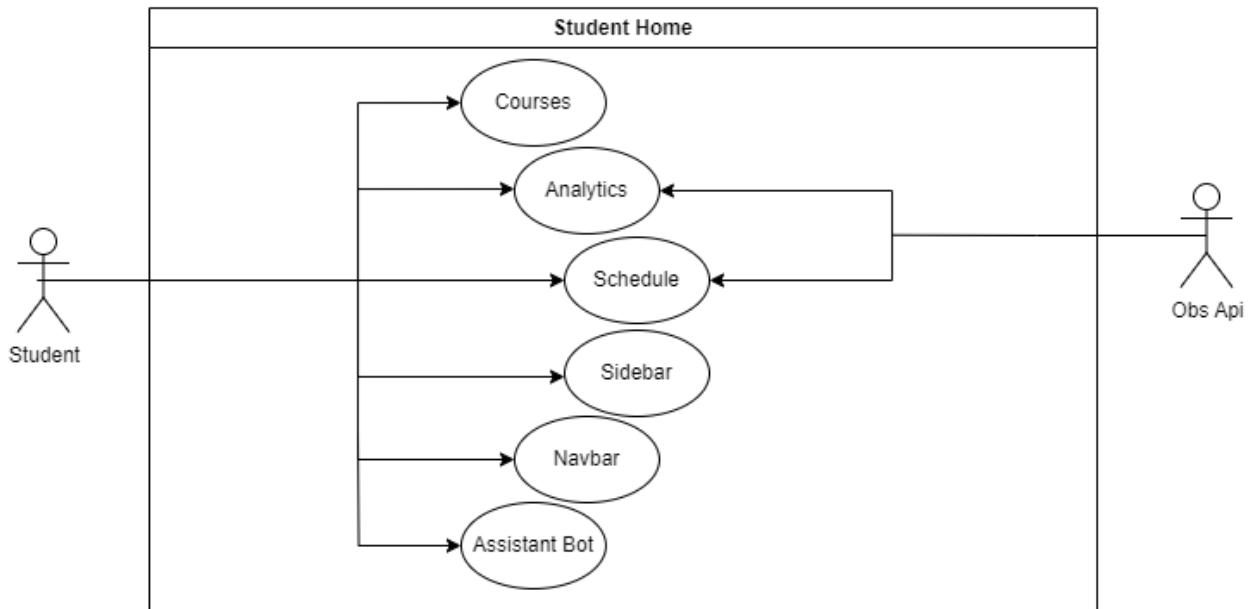
Lecturer Courses:



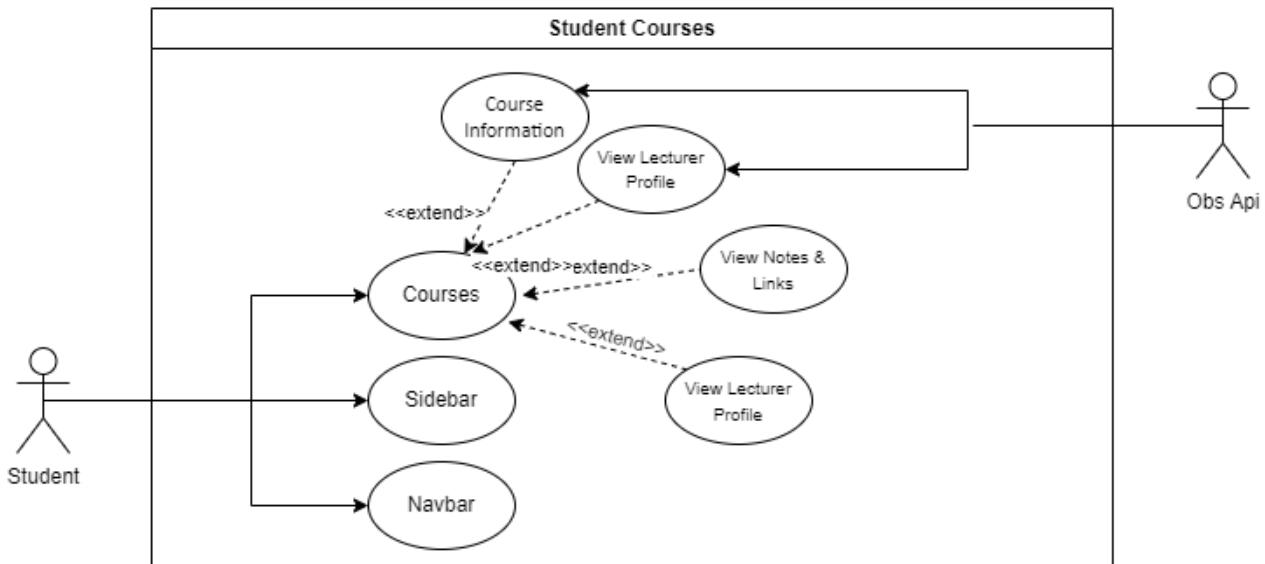
Student Sidebar:



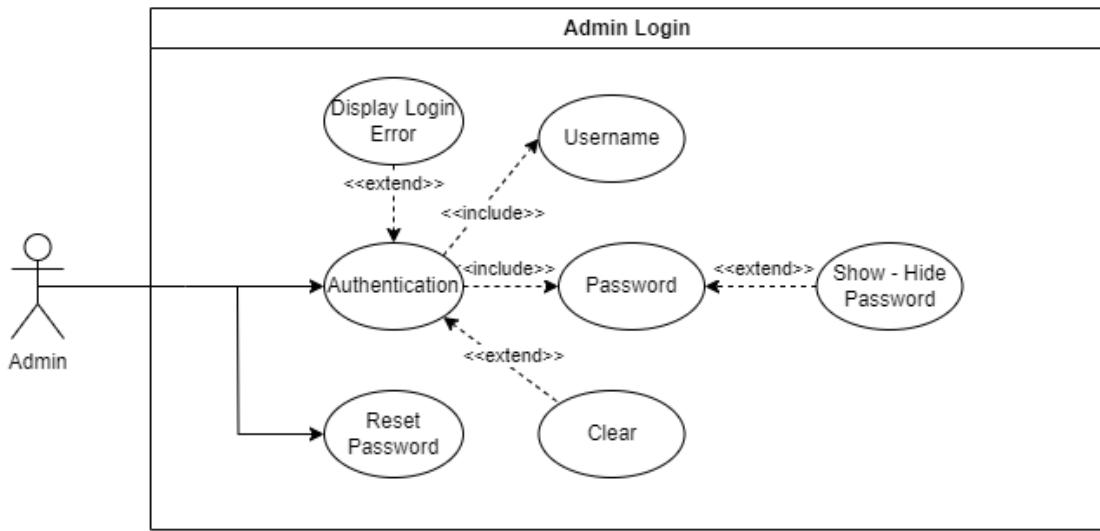
Student Home:



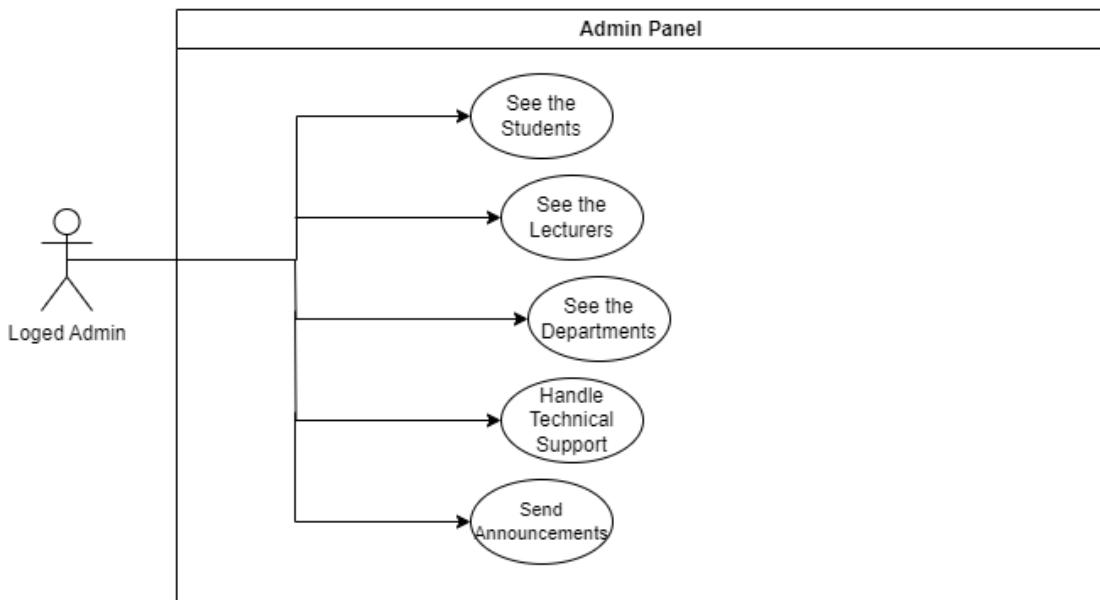
Student Courses:



Admin Login:



Admin Panel:



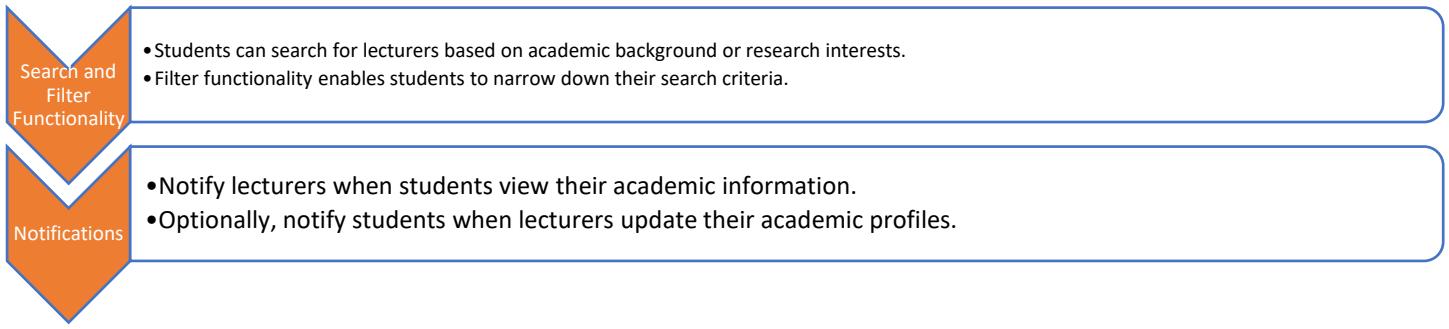
System Functionality

Lecturer Schedule:

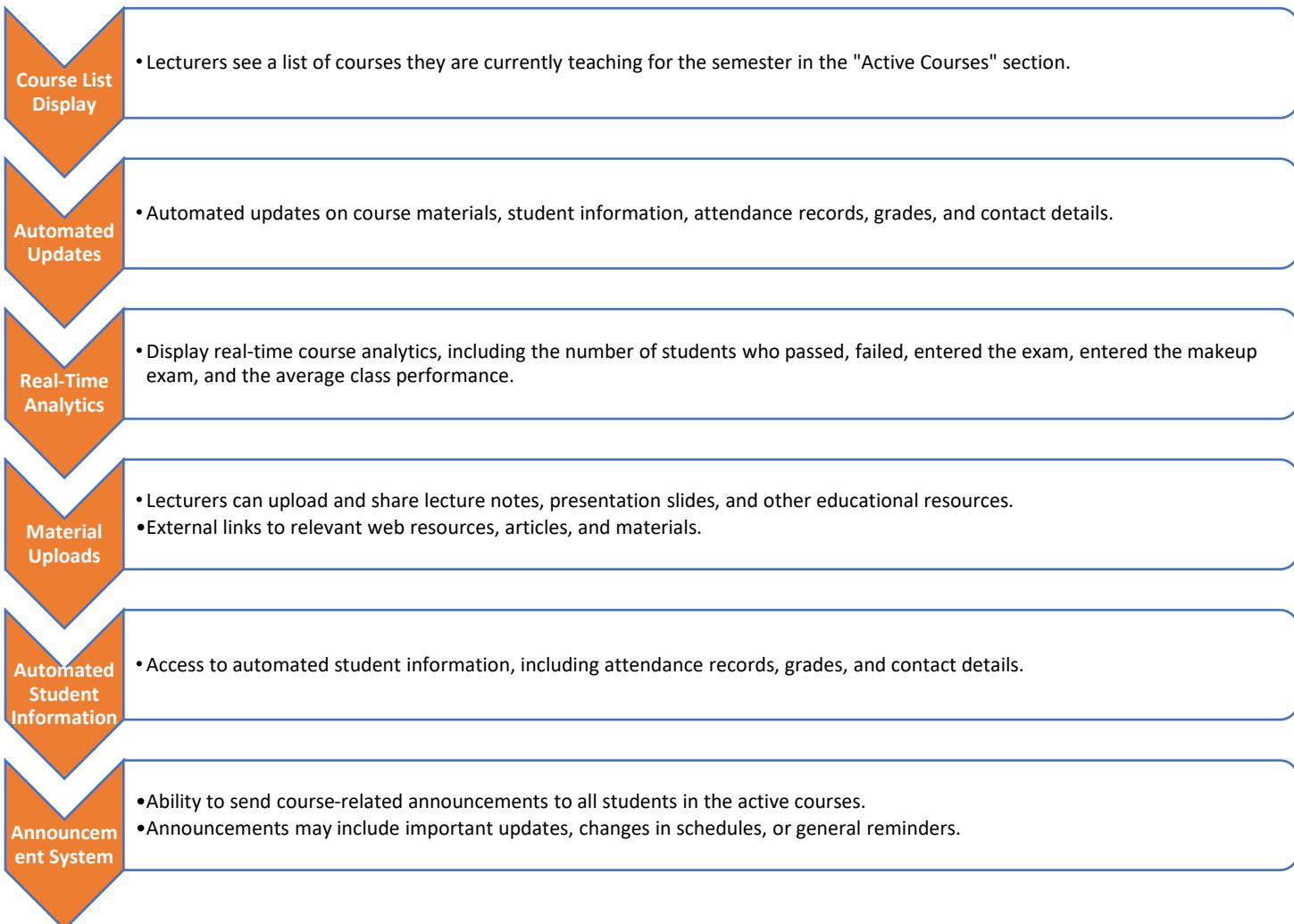


Academic Information:

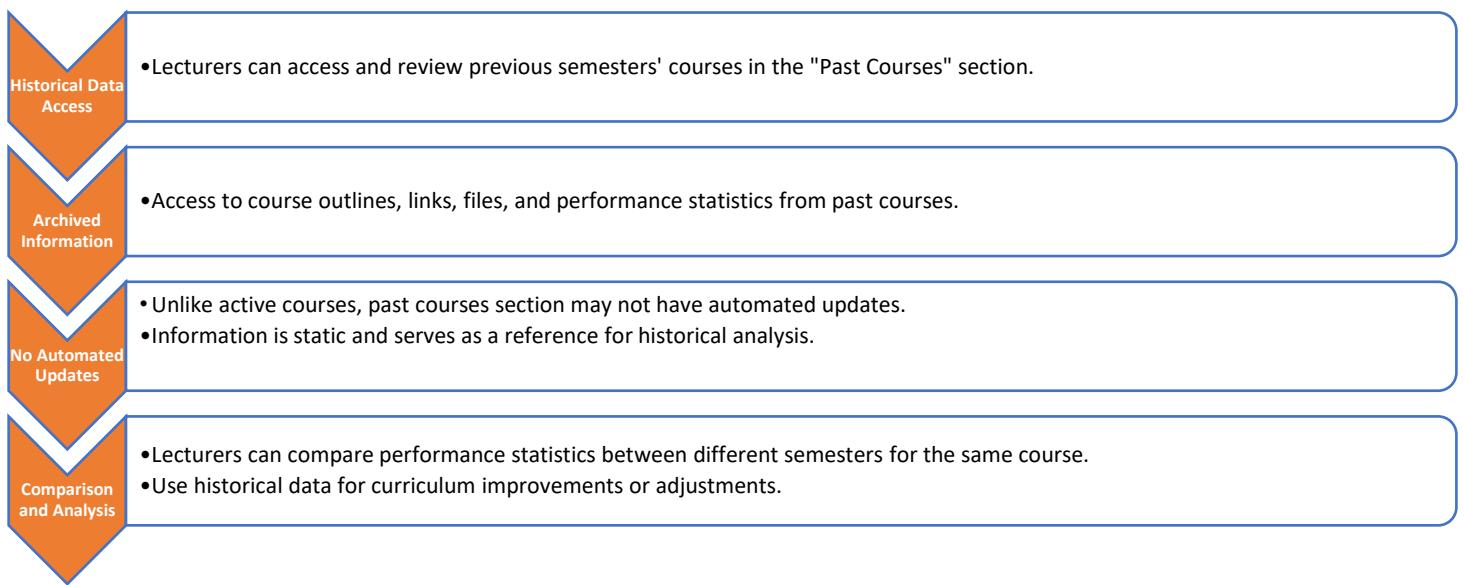




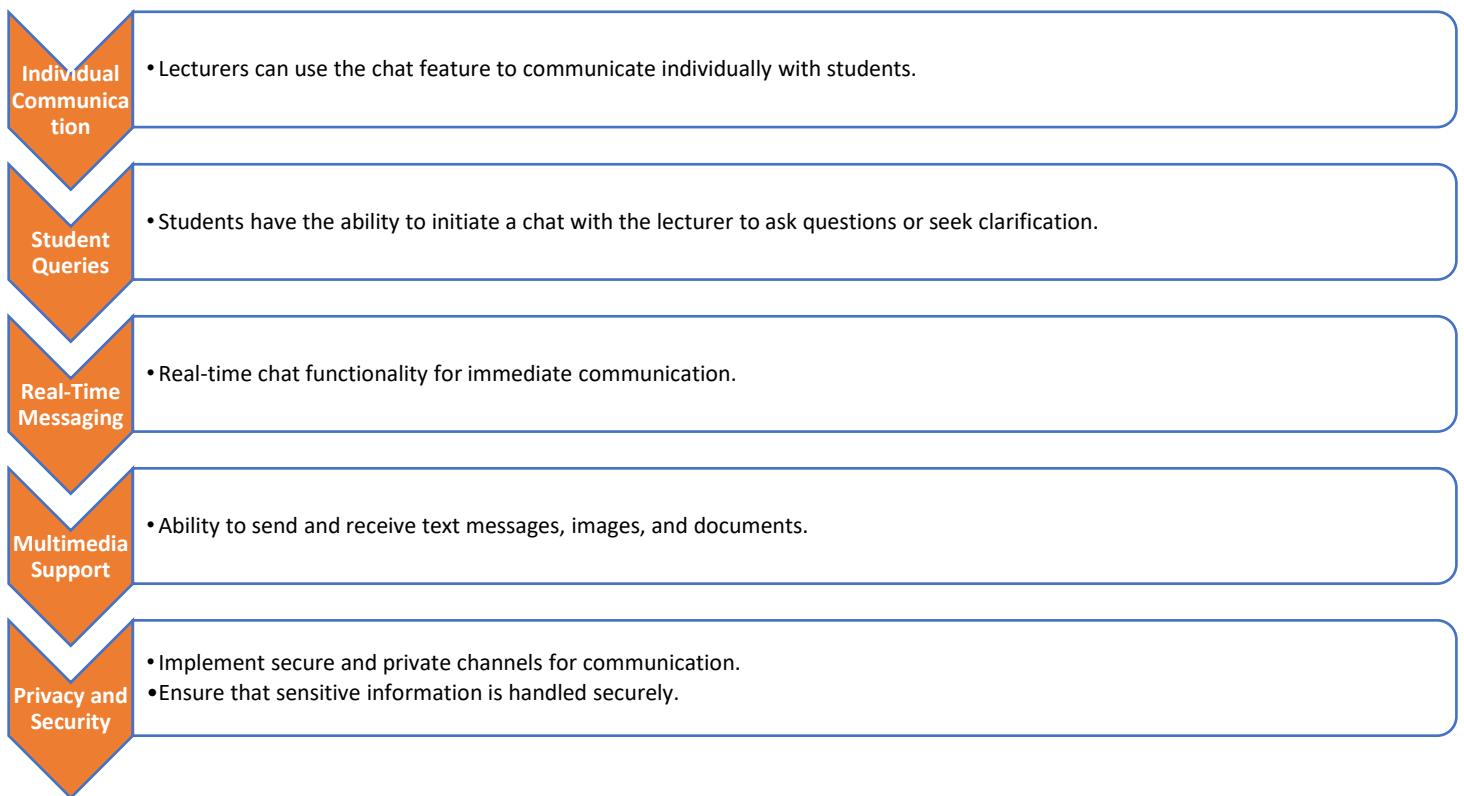
Active Courses:



Past Courses:



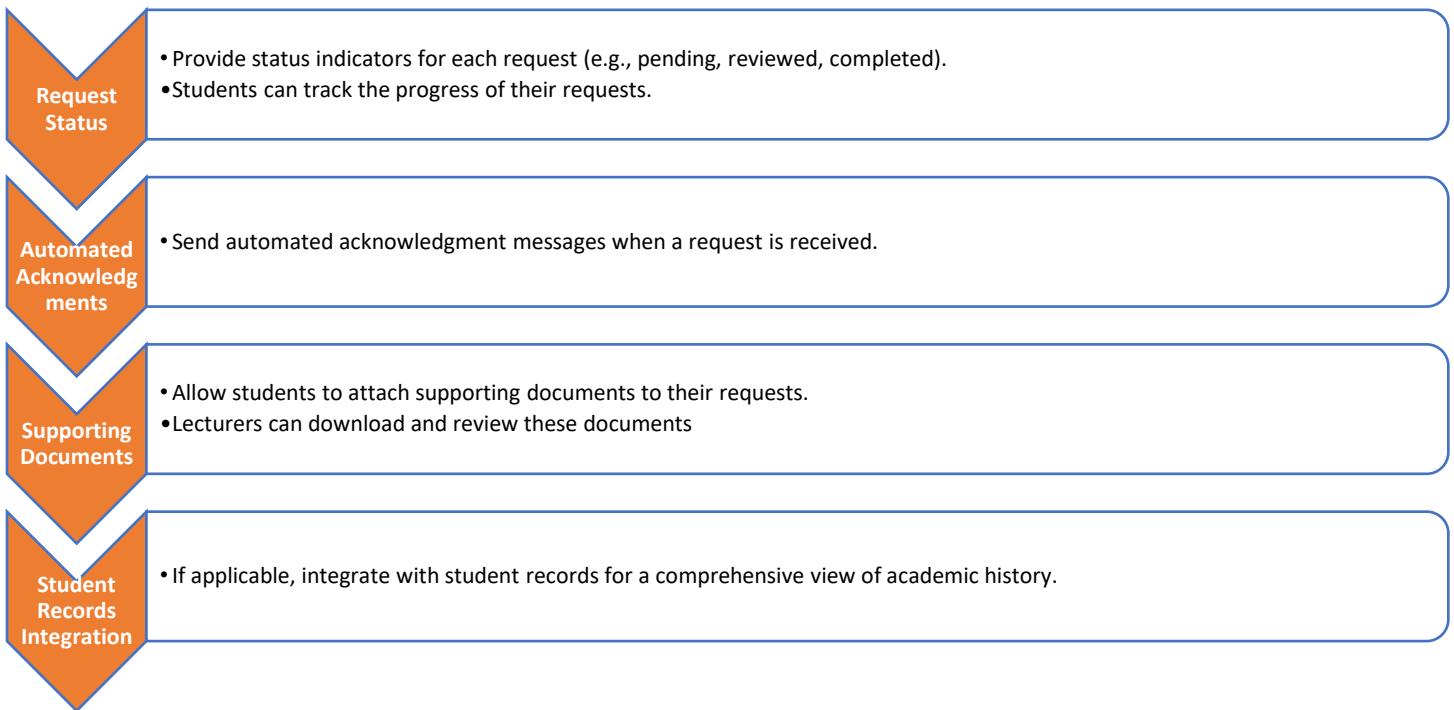
Chat to Communicate with Students:



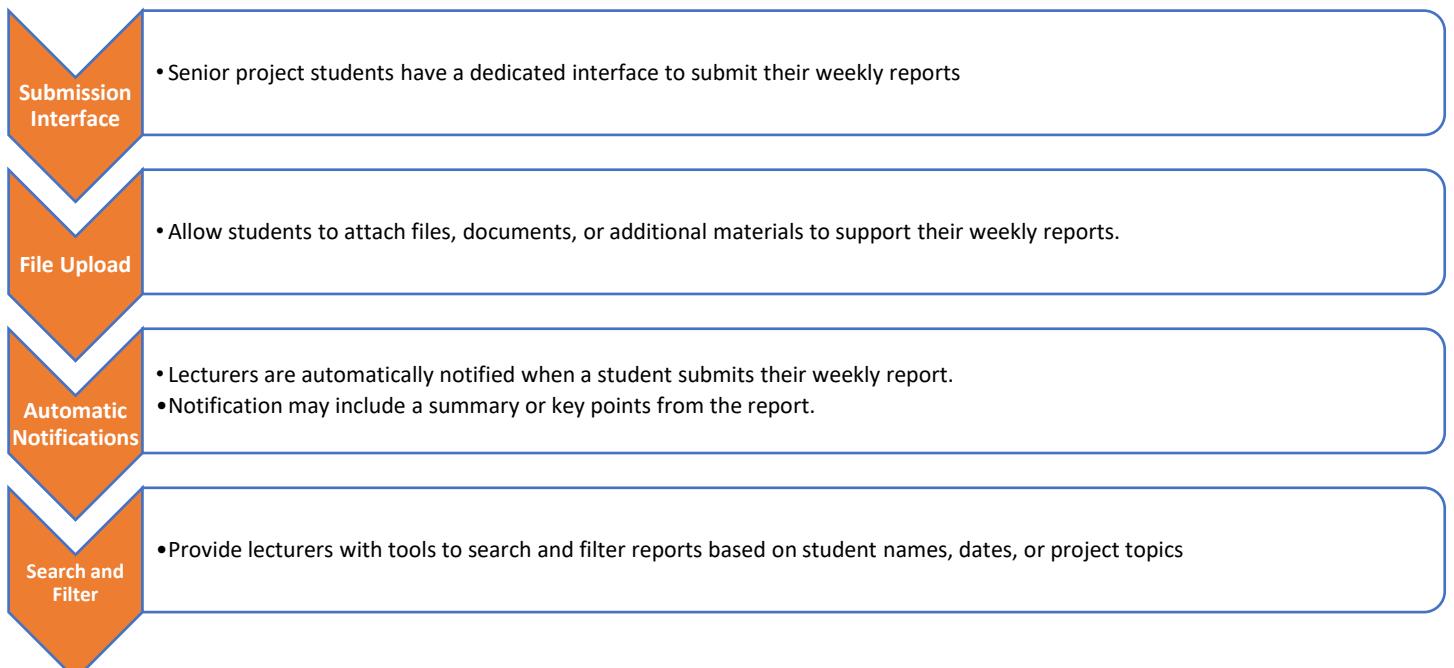
Privacy and Security	<ul style="list-style-type: none">• Implement secure and private channels for communication.• Ensure that sensitive information is handled securely.
Notifications	<ul style="list-style-type: none">• Lecturers receive notifications for incoming student messages.• Optional student notifications for lecturer responses.
Message Archiving	<ul style="list-style-type: none">• Archive chat messages to maintain a record of communication history.• Accessible for future reference.
Search Functionality	<ul style="list-style-type: none">• Lecturers can search through chat history to quickly find past conversations.
Filter by Student	<ul style="list-style-type: none">• Ability to filter conversations by student to streamline communication management.
Online/Offline Status	<ul style="list-style-type: none">• Display online or offline status for both lecturers and students.• Provide an indication of when the lecturer is available for chat.
Typing Indicators	<ul style="list-style-type: none">• Show when someone is typing to indicate an incoming message.
Announcements Feature	<ul style="list-style-type: none">• Ability for lecturers to send announcements to a group of students.• Announcements can include important updates, reminders, or general information.
Read Receipts	<ul style="list-style-type: none">• Implement read receipts to track whether students have read group announcements.

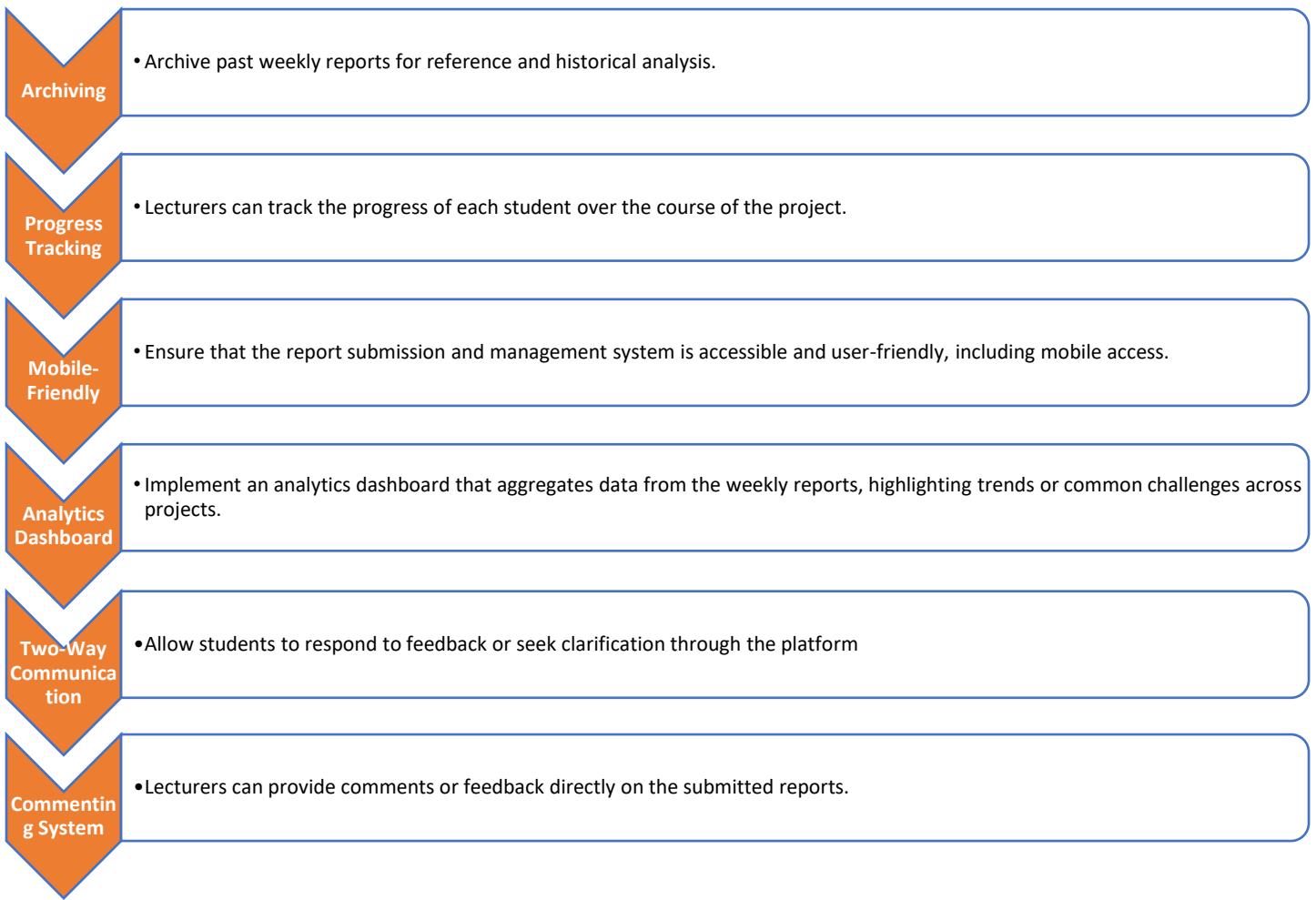
Receiving Student Requests and Replies:



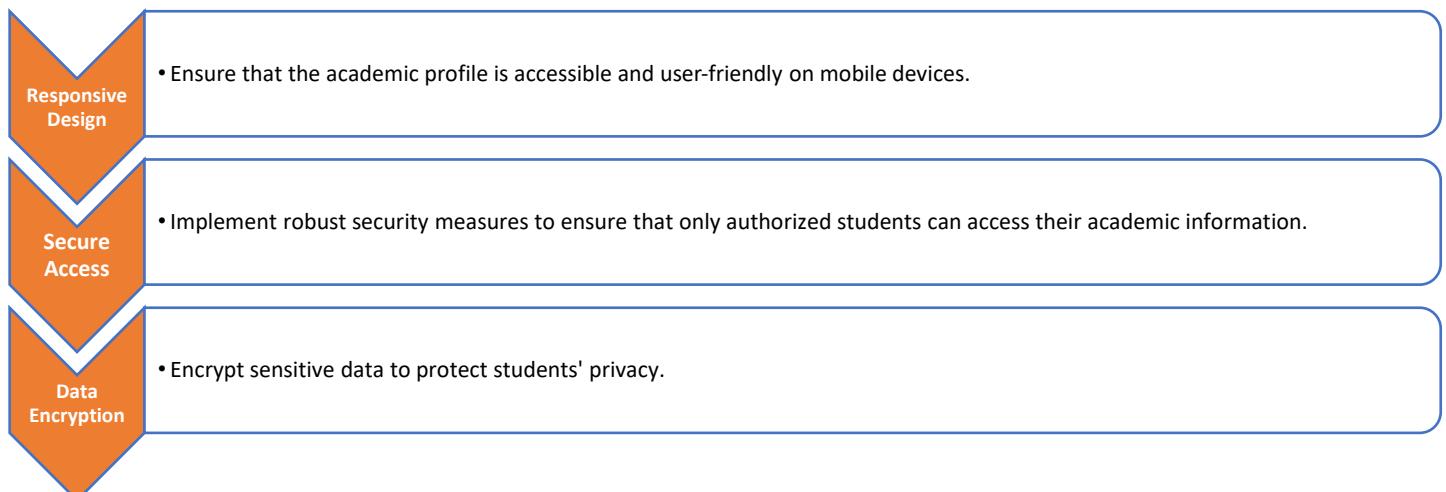


Senior Project Receiving Reports



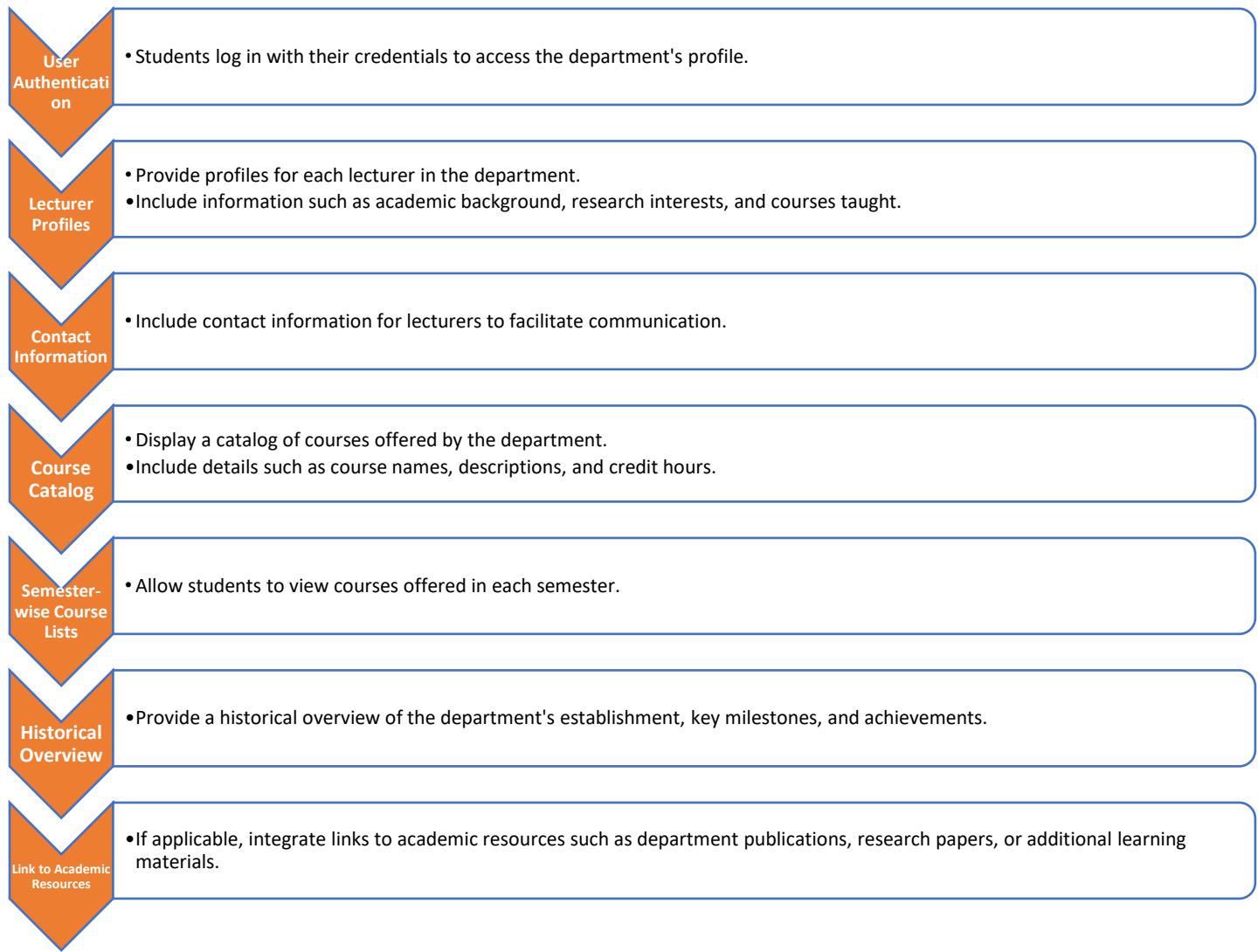


Student Academic Information:

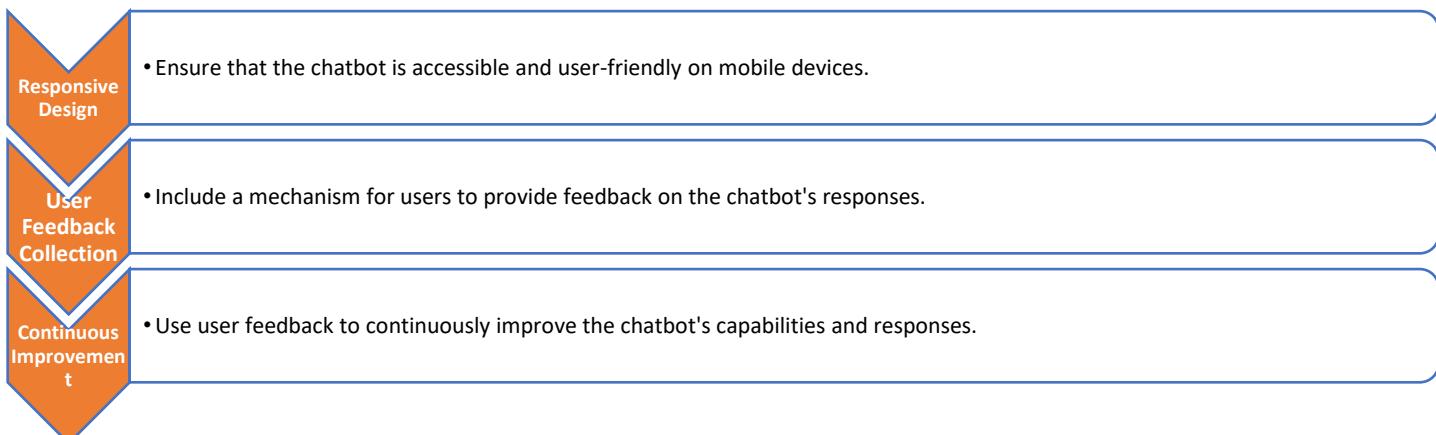




Department Profile:

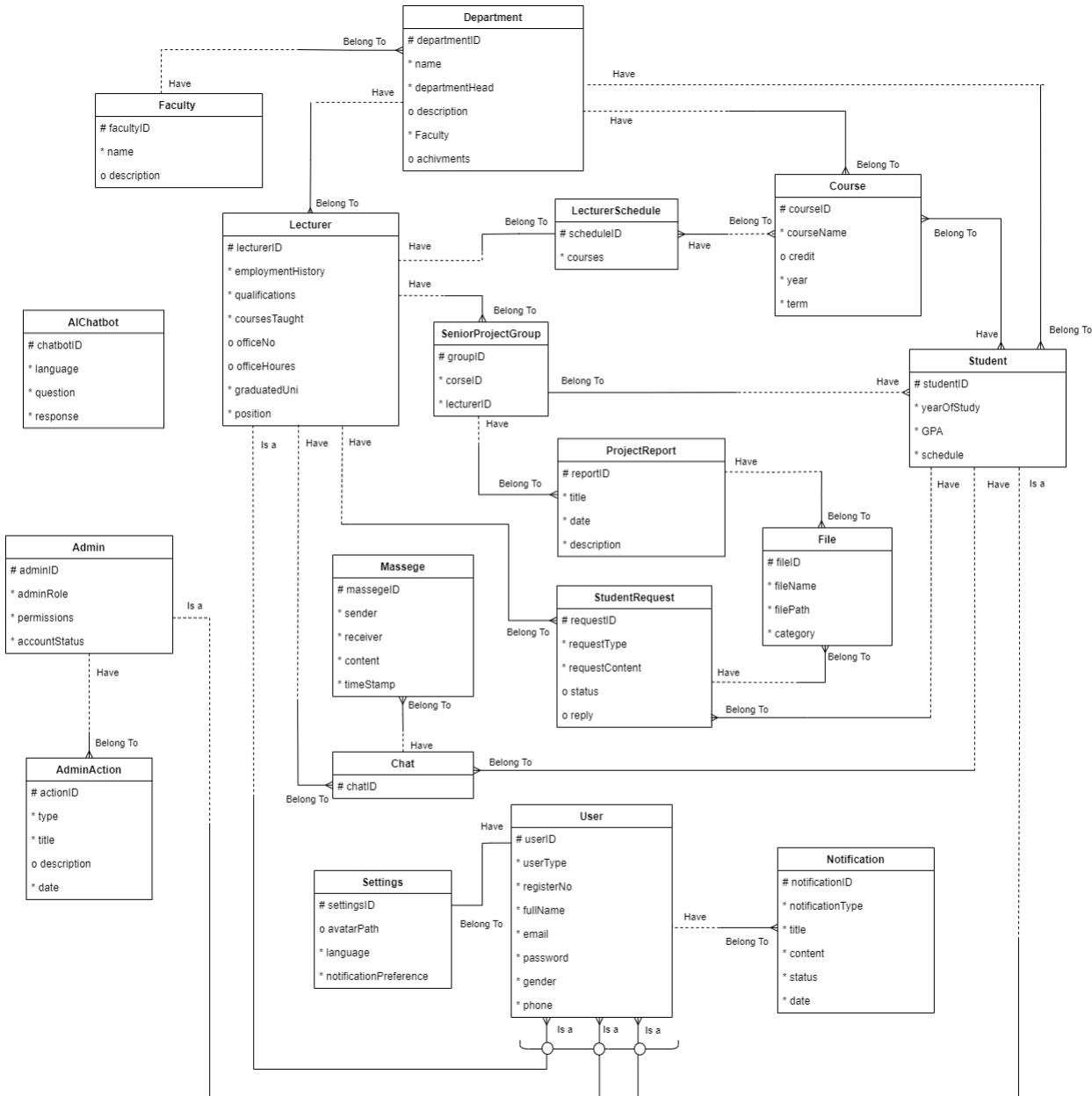


AI Chatbot





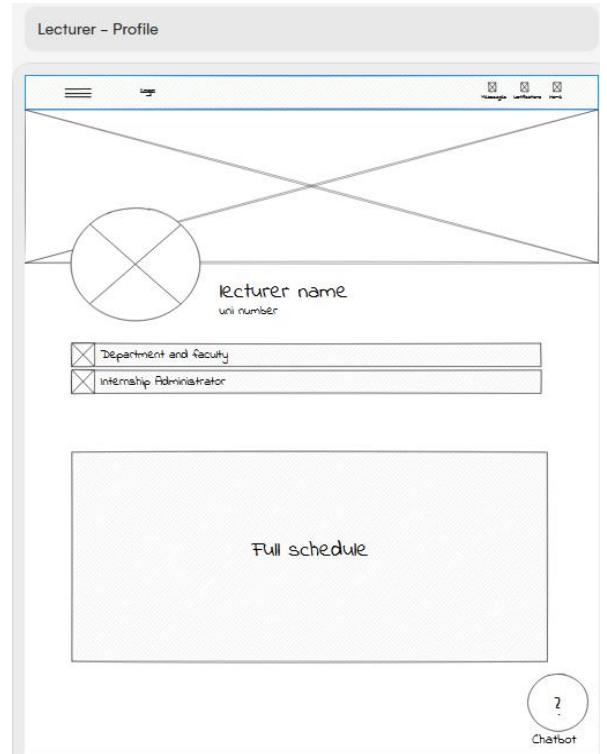
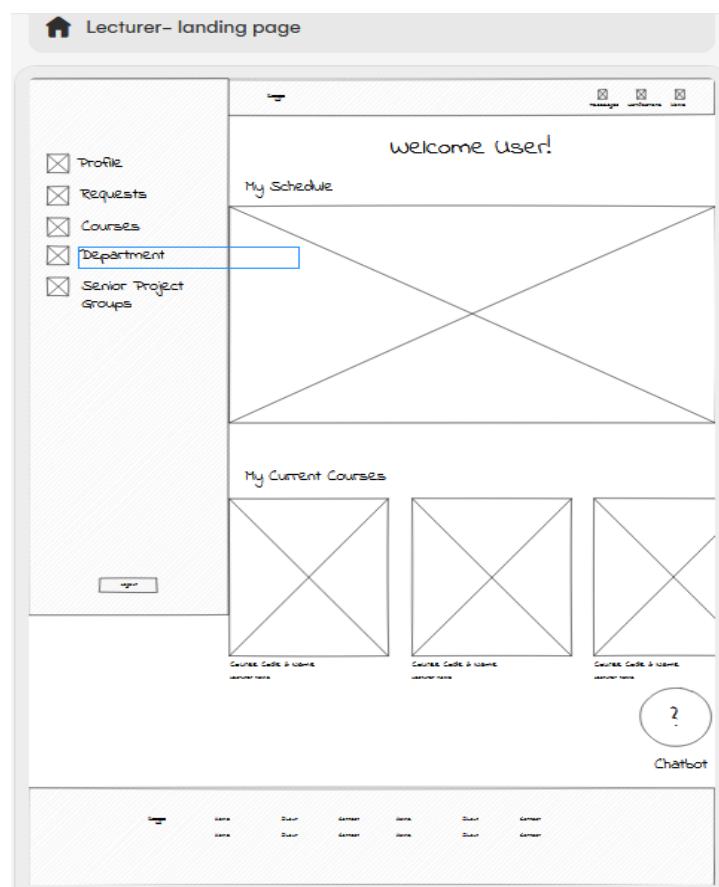
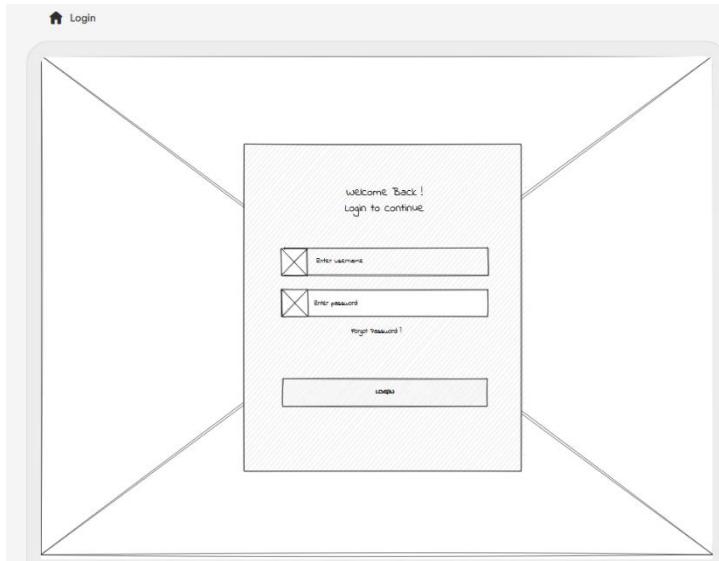
Database Design



User Interface Design

Low-Fidelity Wireframes

1. Web Page Screens



This wireframe shows a requests management screen. The header has a "Logout" button and links for "Messages", "Profile", "Home", and "Logout". The main content is titled "Requests". It features two tables: "Current Requests" and "Past Requests".

Current Requests		
Receiver	Type	Action
Drilhami	internship	<input type="button" value="Edit"/> <input type="button" value="Cancel"/>

Past Requests		
Receiver	Type	Status
Drilhami	internship	Approved

Courses

Courses

My Current Courses

Course Code & Name
User Name

My Past Courses

Course Code & Name
User Name

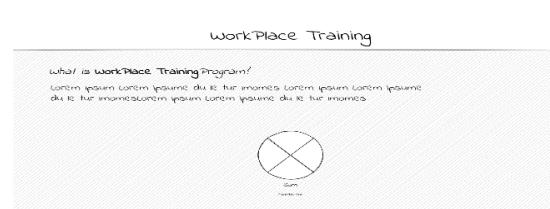
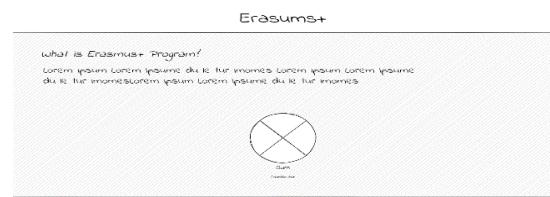
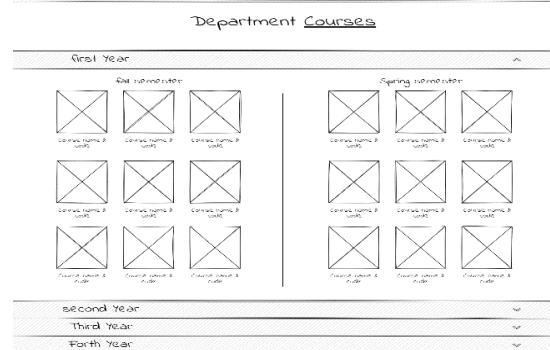
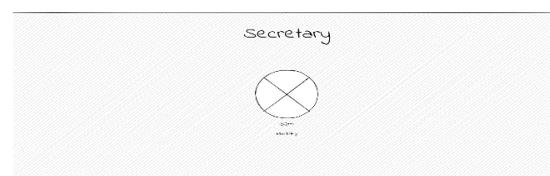
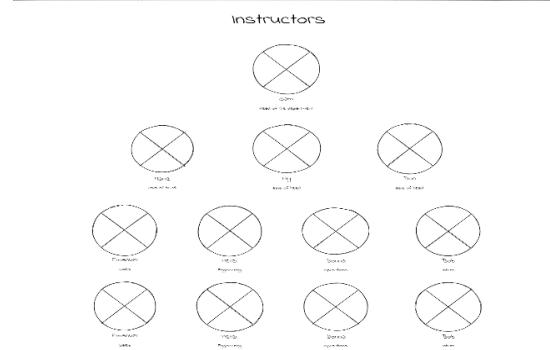
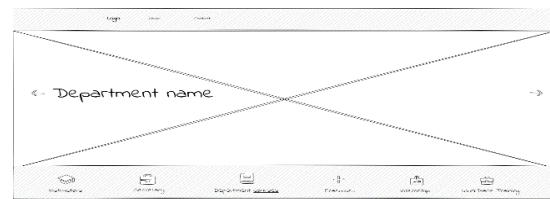
Senior Project groups

Senior Project groups

+ Add new group filter by:

Group Name View Details Edit Delete	Group Name View Details Edit Delete
Group Name View Details Edit Delete	Group Name View Details Edit Delete
Group Name View Details Edit Delete	Group Name View Details Edit Delete

Logo Home About Contact





Notifications

A list of three notifications. Each notification card includes a checkbox, a timestamp, and a brief description.

- Recent notification**
Sender name: weight 4 hours ago
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
- sender name**
weight 4 hours ago
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
- sender name**
weight 4 hours ago
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

[View more](#)

A sidebar for filtering notifications. It has sections for 'Filter' (with date ranges like 'Today', 'Yesterday', 'Past 1 Days', 'This Month', 'Past 4 Months') and 'From' (with checkboxes for 'Teachers', 'Students', 'Administration', and 'Everyone').

Filter
 Today
 Yesterday
 Past 1 Days
 This Month
 Past 4 Months

From
 Teachers
 Students
 Administration
 Everyone
[Reset all filters](#)



Student - landing page

The main content area of the student landing page. It includes:

- A sidebar with checkboxes for 'Profile', 'Requests', 'Courses', 'Department', and 'Senior Project Groups'.
- A central area with a 'Welcome User!' message and four placeholder cards labeled 'Subject name'.
- A section titled 'Today's Schedule' which is crossed out with a large 'X'.
- A bottom section titled 'My Current Courses' showing four placeholder cards labeled 'Course Code & Name'.

My Current Courses

Four placeholder cards for current courses, each containing the text 'Course Code & Name' and 'Available now'. A question mark icon and a 'Chatbot' button are at the bottom right.

Course Code & Name
Available now

?

Chatbot

notes uploading -lecturer

Notes & Related Links

A grid of nine notes cards, each with a placeholder 'file name file type date'. Navigation icons are at the bottom right.

category **Search** **FILTER** **AUDIO** **VIDEO**

file name file type date
file name file type date

...

A modal window for uploading files. It has a title 'Upload a file', a placeholder 'drag and drop your file here to start uploading', and a 'UPLOAD' button.

category **Search** **FILTER** **AUDIO** **VIDEO**

file name file type date
file name file type date

...

Upload a file:
drag and drop your file here to start uploading
UPLOAD

Student - Profile

The main content area of the student profile page. It includes:

- A large circular placeholder for the student's profile picture.
- A section for 'Student name' and 'student number'.
- Input fields for 'Department and faculty', 'Class 1', 'Avn 1', and 'Ago 1'.
- A large empty box for 'Full schedule'.
- A question mark icon and a 'Chatbot' button at the bottom right.

Logo

Messages Profile Home Logout

Requests

Current Requests		
From	Type	Action
Mohamed Zubi	internship	<button>Show</button>

Past Requests		
From	Type	Status
Mohamed Zubi	internship	Approved

Student - course

Logo

Messages Profile Home Logout

Course Name & Code

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat."

Student name

View profile

Notes & Related Links

category

Search

Notes

Logout

2 / 2 → 3

?

Chatbot

Logo

Messages Profile Home Logout

Courses

Courses		
Category	Title	Date
	Course Name	2023-09-15
	Course Name	2023-09-15
	Course Name	2023-09-15

My Current Courses

Course Code & Name
 Student name

My Past Courses

Course Code & Name
 Student name

2. Mobile App Screens

The image displays a grid of 12 wireframe screens for a mobile application, arranged in three rows and four columns. Each screen shows a different part of the app's interface, including:

- Row 1:**
 - Screen 1: Welcome Back! Login to continue.
 - Screen 2: Welcome User. My Schedule (with a large red X over the card), My Current Courses (with three cards, each with a red X over the card), and a section for Code & name (lec. name) with three cards, each with a red X over the card).
 - Screen 3: Logo, Profile requests, Courses, Senior Project Groups, Department, and a LOG OUT button.
 - Screen 4: Student Name, student Number, Department and Faculty (checkboxes for Class 1, Aivo 1, Aivo 2), Full Schedule (with a large red X over the card), and a question mark icon.
- Row 2:**
 - Screen 1: Requests. Current Requests table (From: Mohammad Abu, Type: internship, Action: Show) and Past Requests table (From: Mohammad Abu, Type: internship, Action: Approved, Declined). Both tables have a red X over them.
 - Screen 2: Courses. Current courses section with four cards, each with a red X over the card, and a Past courses section with four cards, each with a red X over the card.
 - Screen 3: Course name & code. Includes a search bar, recent notifications, and a list of messages from Sender Name (Subject: toh) with a red X over them.
 - Screen 4: Similar to Screen 3, showing a list of messages from Sender Name (Subject: toh) with a red X over them, and a LOAD MORE button.
- Row 3:**
 - Screen 1: Messages. Includes a filter section (choosing Today, Yesterday, Past 7 Days, Past Month, Past 4 Years) and a list of messages from Sender Name (Subject: toh) with a red X over them.
 - Screen 2: Senior Project groups. Shows a list of group members (4. Mohammad Zulfi, 2. Nareen Bouts) and their reports (Report 1, 12/12/2023) with a red X over them.
 - Screen 3: Similar to Screen 2, showing a list of group members and their reports with a red X over them.

Logo

Instructors

Secretary

Course name & code

Requests

Department Courses

Erasumst+

First Year

Second Year

Third Year

Forth Year

Footer

Erasmst+

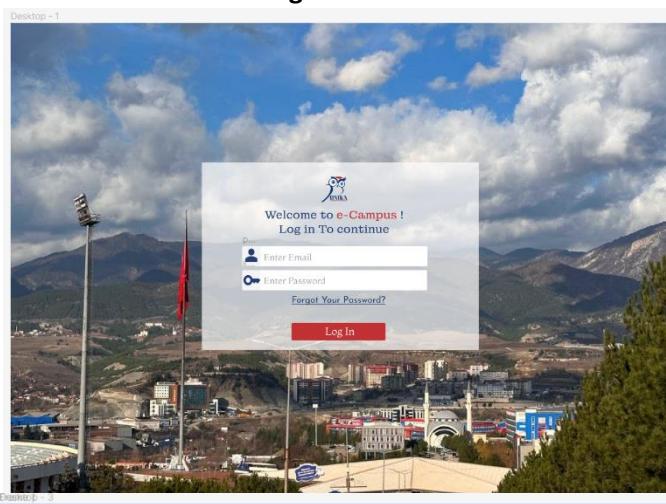
what is Erasmst+ Program?

Course name & code

Peter Coordinator

Experience Design (Mockups & High-Fidelity Prototypes)

1. Web Page Screens



Desktop - 1

X e-Campus

Welcome Nisreen!

Profile

Requests

Courses

Senior Project Groups

Department

PRG209 Programming Dr. Alex Sam

CPE323 Microprocessors Dr. Alex Sam

CPE309 Data Communication Dr. Alex Sam

CHM149 Chemistry Dr. Alex Sam

About Us Who Are we? What Is e-Campus? Contact Us info@karabuk.com +9076492572092 Social Media facebook instagram Copyright © KBU KBU

Desktop - 6

e-Campus

Courses

Current Courses

MAT209 Mathematics Dr. Alex Sam

PHY209 Physics Dr. Alex Sam

PRG209 Programming Dr. Alex Sam

PHY209 Physics Dr. Alex Sam

MAT209 Mathematics Dr. Alex Sam

Past Courses

MAT209 Mathematics Dr. Alex Sam

PHY209 Physics Dr. Alex Sam

PHY209 Physics Dr. Alex Sam

PHY209 Physics Dr. Alex Sam

MAT209 Mathematics Dr. Alex Sam

About Us Who Are we? What Is e-Campus? Contact Us info@karabuk.com +9076492572092 Social Media facebook instagram Copyright © KBU KBU

Desktop - 2

e-Campus

Welcome Nisreen!

Today's Schedule

March 25 Monday

Time Line	
10:00 MAT209 Mathematics	14:00 MAT209 Mathematics
11:00 MAT209 Mathematics	09:00 MAT209 Mathematics
12:00 MAT209 Mathematics	10:00 MAT209 Mathematics
13:00 MAT209 Mathematics	08:00 MAT209 Mathematics

Lessons

MAT209 Mathematics

PHY209 Physics

PRG209 Programming

My Current Courses

MAT209 Mathematics Dr. Alex Sam

PHY209 Physics Dr. Alex Sam

PRG209 Programming Dr. Alex Sam

CPE323 Microprocessors Dr. Alex Sam

CPE309 Data Communication Dr. Alex Sam

CHM149 Chemistry Dr. Alex Sam

About Us Who Are we? What Is e-Campus? Contact Us info@karabuk.com +9076492572092 Social Media facebook instagram Copyright © KBU KBU

Desktop - 3

e-Campus

Nisreen BOUTA 201010101010

Computer Engineering - Faculty of Engineering

Class 4 ANO 4.00 AGNO 4.00

My Time Table

Mon	Tue	Wed	Thu	Fri
14:00 MAT209 Mathematics	10:00 MAT209 Mathematics			
09:00 MAT209 Mathematics	08:00 MAT209 Mathematics			
08:00 MAT209 Mathematics				

About Us Who Are we? What Is e-Campus? Contact Us info@karabuk.com +9076492572092 Social Media facebook instagram Copyright © KBU KBU

Frame 7

e-Campus

About Us Who Are we? What Is e-Campus? Contact Us info@karabuk.com +9076492572092 Social Media facebook instagram Copyright © KBU KBU

Desktop - 5

e-Campus

Requests

Active Requests

Receiver Prof İlhami ORAK

Request Type Internship

Action show

Receiver Prof İlhami ORAK

Request Type Internship

Action show

Receiver Prof İlhami ORAK

Request Type Internship

Action show

Receiver Prof İlhami ORAK

Request Type Internship

Action show

Receiver Prof İlhami ORAK

Request Type Internship

Action show

Receiver Prof İlhami ORAK

Request Type Internship

Action show

Receiver Prof İlhami ORAK

Request Type Internship

Action show

Receiver Prof İlhami ORAK

Request Type Internship

Action show

Receiver Prof İlhami ORAK

Request Type Internship

Action show

Past Requests

Receiver Prof İlhami ORAK

Request Type Internship

Action Approved

Receiver Prof İlhami ORAK

Request Type Internship

Action Approved

Receiver Prof İlhami ORAK

Request Type Internship

Action Approved

Receiver Prof İlhami ORAK

Request Type Internship

Action Approved

Receiver Prof İlhami ORAK

Request Type Internship

Action Approved

About Us Who Are we? What Is e-Campus? Contact Us info@karabuk.com +9076492572092 Social Media facebook instagram Copyright © KBU KBU

Desktop - 4

e-Campus

Dr. Ilhami Muharram
19432400023
Computer Engineering - Faculty of Engineering

Full Schedule

Mon	Tue	Wed	Thu	Fri
16:00 DAC209 Data Communication Group A		10:00 Senior Project Group B		
09:00 DAC209 Data Communication Group B				
08:00 Senior Project Group A				

About Us Contact Us Social Media

e-Campus Who Are we? info@karabuk.com +9076492572092 facebook instagram

Copyright © KBU
KBU

Desktop - 7

e-Campus

PRG209 Programming

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris

Notes & Related Links

Category name	Search	Sort By	Filter
pdf	Week 3 Exercises.pdf 1/12/2023	...	Week 3 Notes.docx 1/12/2023
PPTX	Week 3 Notes.ppt 1/12/2023	...	txt
XLS	Week 3 Notes.xls 1/12/2023	...	Week 3 Notes.txt 1/12/2023

About Us Contact Us Social Media

e-Campus Who Are we? info@karabuk.com +9076492572092 facebook instagram

Copyright © KBU
KBU

Desktop - 11

e-Campus

Welcome Nisreen

Today's Schedule

March 25 Monday	Time Line
MAT209 Mathematics	10:00 MAT209 Mathematics
PHY209 Physics	11:00 MAT209 Mathematics
PRG209 Programming	12:00 MAT209 Mathematics

Current Courses

MAT209 Mathematics Dr. Alex Sam	PHY209 Physics Dr. Alex Sam	PRG209 Programming Dr. Alex Sam	CPE323 Microprocessors Dr. Alex Sam	CPE309 Data Communication Dr. Alex Sam	CHM149 Chemistry Dr. Alex Sam
------------------------------------	--------------------------------	------------------------------------	--	---	----------------------------------

About Us Contact Us Social Media

e-Campus Who Are we? info@karabuk.com +9076492572092 facebook instagram

Copyright © KBU
KBU

Frame 10

e-Campus

Senior Project Groups

Add New Group Filter By

Group Name View Details	Edit	Delete	Group Name View Details	Edit	Delete
Group Name View Details	Edit	Delete	Group Name View Details	Edit	Delete
Group Name View Details	Edit	Delete	Group Name View Details	Edit	Delete

About Us Contact Us Social Media

e-Campus Who Are we? info@karabuk.com +9076492572092 facebook instagram

Copyright © KBU
KBU

Desktop - 11

e-Campus

Senior Project Groups

Members

- Nisreen Bouda
- Nisreen Bouda
- Nisreen Bouda

Notes & Related Links

Reports & Files	Search	Add File +	Filter
pdf	Week 3 Exercises.pdf 1/12/2023	Edit	Delete
pdf	Week 3 Exercises.pdf 1/12/2023	Edit	Delete
dox	Week 3 Notes.docx 1/12/2023	Edit	Delete

About Us Contact Us Social Media

e-Campus Who Are we? info@karabuk.com +9076492572092 facebook instagram

Copyright © KBU
KBU

Desktop - 12

e-Campus

Computer Engineering

- [Instructors](#)
- [Secretary](#)
- [Courses](#)
- [Erasmus](#)
- [Internship](#)
- [Workplace Training](#)

Instructors

	Prof. İlhami Head of the department
	Dr. Lilly Assis. Head of the department
	Dr. Sam Assis. Head of the department
	Dr. Sam Assis. Head of the department
	Dr. Sara Assis. Head of the department
	Dr. Lilly Assis. Head of the department
	Dr. Sam Assis. Head of the department
	Dr. Sara Assis. Head of the department

Secretary

Alex
Secretary of the department

Department Courses

First Year

Fall Semester	Spring Semester
MAT209 Mathematics	MAT209 Mathematics
MAT209 Mathematics	MAT209 Mathematics
MAT209 Mathematics	MAT209 Mathematics

Second Year

Third Year

Fourth Year

Erasmus +

What is Erasmus+ program?

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



Alex

Erasmus Program Coordinator

Workplace Training

What is Workplace Training?

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



Alex

Workplace Coordinator

Internship

What is Internship?

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



Alex

Internship Coordinator

?



e-Campus

About Us

Who Are we?
What Is e-Campus?

Contact Us

info@karabuk.com
+9076492572092

Social Media

facebook
instagram

Copyright © KBÜ
KBÜ

2. Mobile App Screens



Instructors



Prof. Ilhami
Head of the department



Dr. Lilly
Assis. Head of the department



Dr. Sam
Assis. Head of the department



Dr. Sam
Assis. Head of the department



Dr. Lilly
Assis. Head of the department



Dr. Sam
Assis. Head of the department



Dr. Sara
Assis. Head of the department



Dr. Lilly
Assis. Head of the



Dr. Sam
Assis. Head of the



Dr. Sara
Assis. Head of the

Secretary

Alex
Secretary of the department

Department Courses

First Year

Fall Semester

MAT209
Mathematics

MAT209
Mathematics

MAT209
Mathematics

Spring Semester

MAT209
Mathematics

MAT209
Mathematics

MAT209
Mathematics

Second Year

Third Year

Forth Year

Forth Year

Erasmus +

What is Erasmus+ program?

Lore ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Alex
Erasmus+ Program Coordinator

Internship

What is Internship?

Lore ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Alex
Internship Coordinator

Workplace Training

What is Workplace Training?

Lore ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Alex
Workplace training Coordinator



The complete source on Figma: [Figma Link](#)

Ai Chat Bot

Process: the chat bot will be developed in python, a Pdf loader will be used to load the pdfs that contains the rules.. After that a text splitter will be used to split each pdf contents into chuncks for example (700 letters). After we will convert the Text Chunks into Embeddings and Create a FAISS Vector Store that will store the chunks in vector database. After that a template text will be prepared to guide the ai model how to deal with the prompts and the pdf files. For exampe :

template=""">Use the following pieces of information to answer the user's question.

If you dont know the answer, just say you don't know, don't try to make up an answer.

Context: {context}

Question: {question}

Only return the helpful answer below and nothing else

Helpful answer

"""

After that we will integrate a large language model such as chatgpt or llama, etc..

We will be using the model to read the pdfs from the embeddings and compare the user prompt (question) to the list of embeddings we have in the vector db. Based on that the model will use the template to know how to handle the response. So, It will read and compare and answer based on the pdf contents, and not make up a response from itself.

We will be using the sin similarity algorithm to compare the embeddings to the prompt.

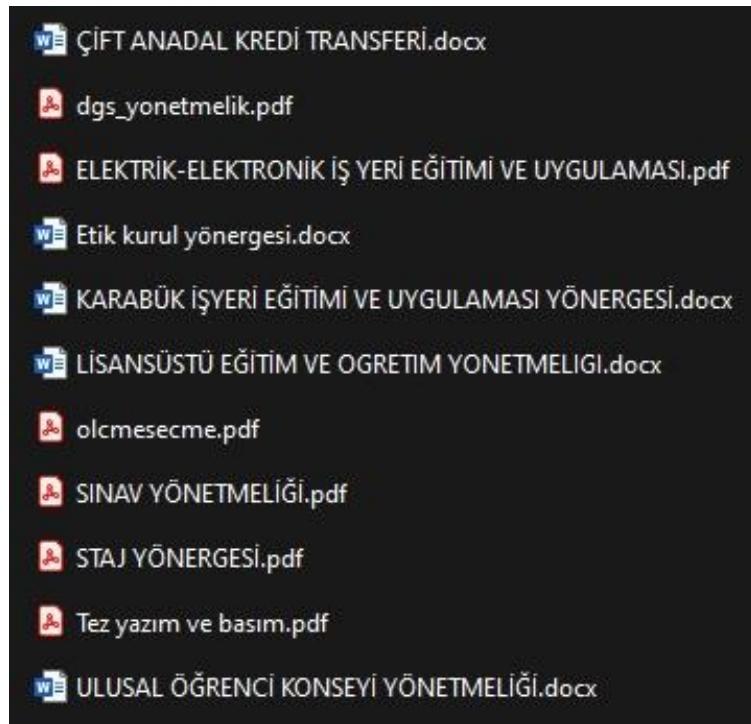
Code example for the process:

```
while True:
    user_input=input("prompt:")
    if user_input=='prompt:exit':
        print('Exiting')
        sys.exit()
    if user_input=="prompt:":
        continue
    // will send the user prompt to the function to do the comparement
    result=chain({'query':user_input})
    // will retrieve the answer (result) from the function
    print(f"Answer:{result['result']}")
print("done")
```

Integration:

- The Flask web framework will be used to make an Api end point for the chatbot.
- An Api to get the user input (Prompt). And use it in the process. And return the response (Answer).

Collected PDFs (Rules):



Web Code Implementation

Overview:

The university automation system is implemented using the MERN stack, which includes MongoDB as the database, Express.js as the backend framework, React.js for the frontend, and Node.js for server-side scripting. The system aims to streamline various university processes, providing functionalities such as student enrollment, course management, and grading.

The complete source code for the project can be found on GitHub. The repository includes both frontend and backend code, along with necessary documentation.

GitHub Repository Link: <https://github.com/MOHAMAD-ZUBI/MERN-Student-Automation-System>

Follow these steps to run the application locally on your machine.

1. Before you begin, ensure you have the following installed on your machine:

- Node.js (with npm)
- MongoDB (Make sure the MongoDB server is running)

2. Clone the Repository

```
git clone https://github.com/MOHAMAD-ZUBI/MERN-Student-Automation-System
cd your-repository
```

3. Install Dependencies

```
cd server
npm i
cd client
npm i
```

4. Configure Environment Variables

Create a **.env** file in the **backend** directory and provide the following variables:

```
MONGODB_URI=your_mongodb_connection_string
PORT=3001
```

5. Run the Application

Backend

```
cd server
npm run server
```

Frontend

```
cd client
npm start
```

File Structure:

Backend (Node.js/Express)

```
| -- controllers/  
| -- models/  
| -- routes/  
| -- .gitignore  
| -- package.json  
| -- server.js  
| -- .env
```

- **controllers/**: This directory contains controller files that handle business logic.
- **models/**: This directory includes data models.
- **routes/**: This directory holds route files.
- **package.json**: Configuration file for Node.js projects, including dependencies and scripts.
- **README.md**: Documentation file for the backend part of your project.
- **server.js**: The main entry point for your Express.js application.
- **.env**: Configuration file for environment variables used in your frontend.
- **.gitignore**: A file specifying which directories to ignore on the Git repository.

Frontend (React)

```
| -- public/  
|   | -- index.html  
| -- src/  
|   | -- components  
|   | -- Pages  
|   | -- App.jsx  
|   | -- index.jsx  
| -- .gitignore  
| -- package.json  
| -- .env
```

- **public/**: This directory contains the index.html file, the entry point for the React.
- **src/**: This directory holds all the source code for your React application.
- **components/**: This directory contains React components.
- **App.jsx**: The main component that serves as the entry point for the React application.
- **index.jsx**: The file responsible for rendering the React app into the HTML document.
- **package.json**: Configuration file for Node.js projects, including dependencies and scripts.

Code Snippets:

These code snippets provide a glimpse of the project's structure and functionality. For a more in-depth understanding, please refer to the complete codebase available on the GitHub repository.

Backend (Node.js/Express)

```
/* server.js */
const express = require("express");
const mongoose = require("mongoose");

// Routers
const auth = require("./Routes/AuthRouter");
const studentRouter = require("./Routes/StudentRouter");
...

// Middlewares
const { isStudentUser } = require("./Middlewares/Roles");
const { isAdminUser } = require("./Middlewares/Roles");
...

const app = express();

app.use(express.json());
mongoose
  .connect("mongodb://localhost:27017/Senior")
  .then(() => {
    app.listen(3060, () => {
      console.log("server is on 3060");
    });
  })
  .catch((err) => {
    console.log(err.message);
  });

// Routes
app.use("/api/auth", auth);
app.use("/api/student", isStudentUser, studentRouter);
...
```

```
/* userModule.js */
const userSchema = new Schema(
  {
    firstName: {
      type: String,
      required: true,
    },
    lastName: {
      type: String,
      required: true,
    },
    ...
    permissions: {
      type: [
        {
          type: String,
          enum: Roles,
          default: [Roles.Student],
        },
      ],
    },
  );
);

module.exports = mongoose.model("User", userSchema);
```

Frontend (React)

```
/* App.js */
import {HomeLayout, Landing, Register, ...} from "./pages";

import { action as registerAction } from "./pages/Register";
import { action as loginAction } from "./pages/Login";
import { loader as dashboardLoader } from "./pages/DashboardLayout";
...

const router = createBrowserRouter([
  {
    path: "/",
    element: <HomeLayout />,
    errorElement: <Error />,
    children: [
      {
        index: true,
        element: <Landing />,
      },
      {
        path: "register",
        element: <Register />,
        action: registerAction,
      },
      ...
    ],
  },
],
);

function App() {
  return (
    <>
      <RouterProvider router={router} />
    </>
  );
}
```

Mobile Application Implementation

Overview:

The Mobile application is developed using React Native and is designed to streamline and enhance various aspects of university operations. The app aims to provide students, faculty, and staff with convenient access to essential university services and information, all from their mobile devices.

The complete source code for the project can be found on GitHub.

GitHub Repository Link: <https://github.com/nisreenbauta/E-campus/tree/master>

Follow these steps to run the application locally on your device.

1- Install Required Dependencies:

- Make sure you have Node.js and npm installed on your system.
- Install Expo CLI globally by running the following command in your terminal:

```
npm install -g expo-cli
```

2- Navigate to your project directory:

```
cd E-campus
```

3- Install Expo Go App:

- Install the Expo Go app on your iOS or Android device from the App Store or Google Play Store, respectively.

4- Run Your App:

- Open your project directory in Visual Studio Code.
- In the terminal within Visual Studio Code, run the following command to start the development server:

```
npm start
```

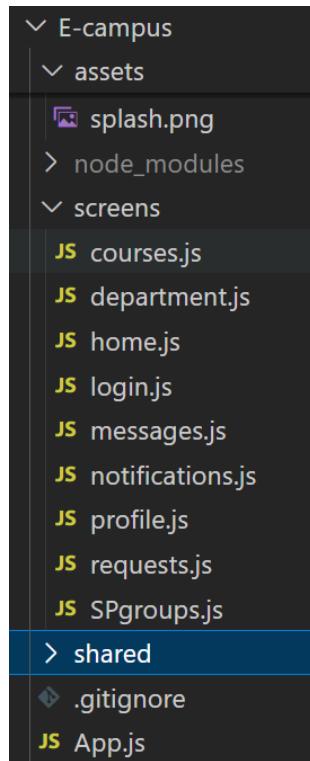
- This command will start the Metro bundler, which compiles your JavaScript code and provides a QR code to load your app.

5- Connect Your Device:

- Open the Expo Go app on your mobile device.
- Scan the QR code displayed in the terminal using the Expo Go app. This will load your React Native app on your device.

File Structure:

The ‘screens’ folder contains the main screens in the application.



Code Snippets:

These code snippets provide a glimpse of the project's structure and functionality. For a more in-depth understanding, please refer to the complete codebase available on the GitHub repository.

Login.js:

handleLogin function is used for authenticating the user’s credentials. For the meantime, it only takes dummy data since the backend is not connected with project yet.

```
const Login = ({navigation}) => {
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');

  const handleLogin = () => {
    //implement your authentication logic
    console.log('Username:', username);
    console.log('Password:', password);
    //clear the input fields after logging in
    setUsername('');
    setPassword('');

    navigation.navigate('Home');
  };
}
```

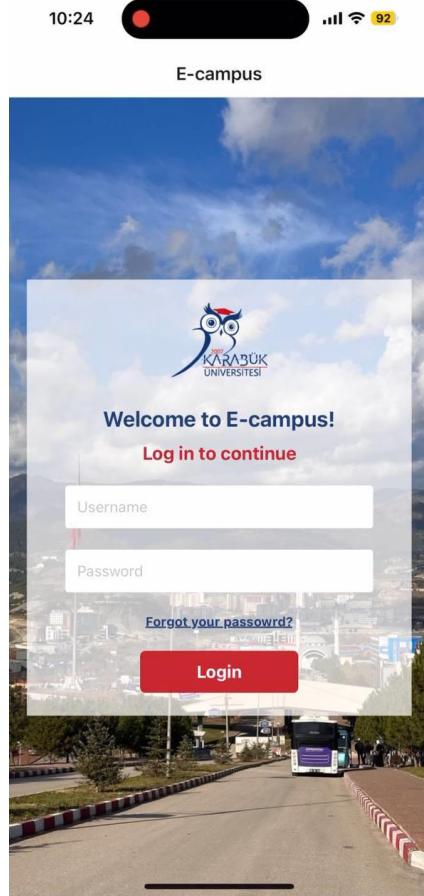
```

return (
  <ImageBackground
    source={require('../assets/login_background.jpg')}
    style={styles.backgroundImage}
  >

    <View style={styles.container}>
      <Image source={require('../assets/kbu_logo.png')} style={styles.logo}>
      />

      <Text style={styles.title}>Welcome to E-campus!</Text>
      <Text style={styles.subtitle}>Log in to continue</Text>
      <TextInput
        style={styles.input}
        placeholder="Username"
        value={username}
        onChangeText={setUsername}
      />
      <TextInput
        style={styles.input}
        placeholder="Password"
        secureTextEntry={true}
        value={password}
        onChangeText={setPassword}
      />
      <Text style={styles.subsubtitle}>Forgot your password?</Text>
      <TouchableOpacity style={styles.button} onPress={handleLogin}>
        <Text style={styles.buttonText}>Login</Text>
      </TouchableOpacity>
    </View>
  </ImageBackground>
);

```



Home.js:

```

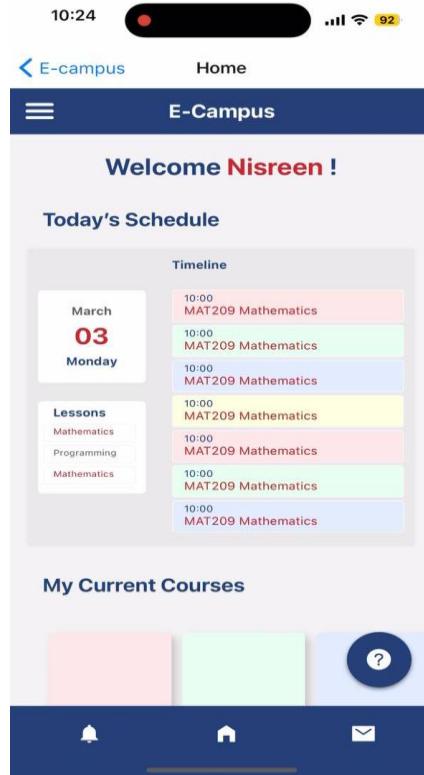
const Home = () => {
  const navigation = useNavigation();
  const [isDrawerOpen, setIsDrawerOpen] = useState(false);

  const toggleDrawer = () => {
    setIsDrawerOpen(!isDrawerOpen);
  };

  const handleLogout = () => {
    navigation.navigate('E-campus');
  };

  const handleDrawerItemPress = (screenName) => {
    navigation.navigate(screenName);
    setIsDrawerOpen(false); // Close the drawer after navigating
  };
  const handleBottomNavBar = (screenName) => {
    navigation.navigate(screenName);
  };
}

```



```

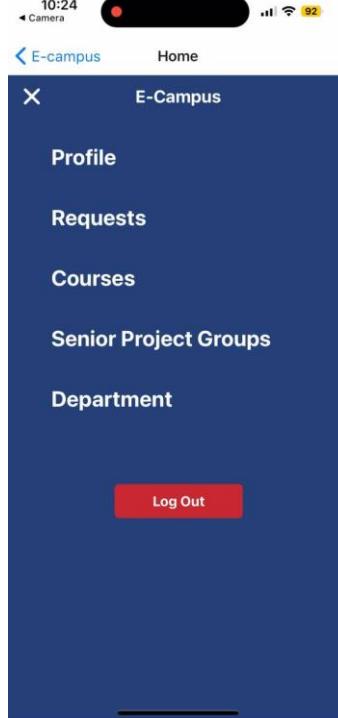
<ScrollView>
  <Text style={styles.mainText}>Welcome <Text style={styles.subtext}>Nisreen</Text> !</Text>
  <Text style={styles.heading}>Today's Schedule</Text>

  <View style={styles.card}>
    <View style={styles.smallCardContainer}>
      <View style={styles.smallCard}>
        <Text style={styles.month}>March</Text>
        <Text style={styles.day}>03</Text>
        <Text style={styles.weekday}>Monday</Text>
      </View>
    <View style={styles.smallCardLessons}>
      <ScrollView>
        <Text style={styles.lessons}>Lessons</Text>
        <View style={styles.whiteRec}><Text style={styles.lessonRed}>Mathematics</Text></View>
        <View style={styles.whiteRec}><Text style={styles.lessonGray}>Programming</Text></View>
        <View style={styles.whiteRec}><Text style={styles.lessonRed}>Mathematics</Text></View>
        <View style={styles.whiteRec}><Text style={styles.lessonGray}>Programming</Text></View>
      </ScrollView>
    </View>
  </View>

  <View style={styles.recsContainer}>
    <Text style={styles.subHead}>Timeline</Text>
    <ScrollView>
      <View style={styles.redRec}><Text style={styles.subSubText}>10:00</Text><Text style={styles.subSubRedText}>MAT209 Mathematics</Text></View>
      <View style={styles.greenRec}><Text style={styles.subSubText}>10:00</Text><Text style={styles.subSubRedText}>MAT209 Mathematics</Text></View>
      <View style={styles.blueRec}><Text style={styles.subSubText}>10:00</Text><Text style={styles.subSubRedText}>MAT209 Mathematics</Text></View>
      <View style={styles.yellowRec}><Text style={styles.subSubText}>10:00</Text><Text style={styles.subSubRedText}>MAT209 Mathematics</Text></View>
      <View style={styles.redRec}><Text style={styles.subSubText}>10:00</Text><Text style={styles.subSubRedText}>MAT209 Mathematics</Text></View>
      <View style={styles.greenRec}><Text style={styles.subSubText}>10:00</Text><Text style={styles.subSubRedText}>MAT209 Mathematics</Text></View>
      <View style={styles.blueRec}><Text style={styles.subSubText}>10:00</Text><Text style={styles.subSubRedText}>MAT209 Mathematics</Text></View>
      <View style={styles.yellowRec}><Text style={styles.subSubText}>10:00</Text><Text style={styles.subSubRedText}>MAT209 Mathematics</Text></View>
    </ScrollView>
  </View>
</View>
<Text style={styles.heading}>My Current Courses</Text>

```

In each screen the nav bar , bottom nav bar and drawer are the same and in a fixed place.Below are the code snippets for each one:



The screenshot shows the E-Campus mobile application interface. At the top, there is a navigation bar with a camera icon, signal strength, and battery level. Below the navigation bar is a header with a back arrow, the text "E-campus", and a "Home" button. A drawer is open, displaying a list of menu items: "Profile", "Requests", "Courses", "Senior Project Groups", "Department", and "Log Out". The "Log Out" button is highlighted with a red background.

```

<View style={styles.navigationBar}>
  <TouchableOpacity onPress={toggleDrawer} style={styles.menuButton}>
    <Ionicons name={isDrawerOpen ? 'close' : 'menu'} size={35} color="white" />
  </TouchableOpacity>
  <Text style={styles.title}>E-Campus</Text>
</View>

/* Drawer Content Goes Here */
{isDrawerOpen && (
  <View style={styles.drawer}>
    <TouchableOpacity onPress={() => handleDrawerItemPress('Profile')}>
      <Text style={styles.drawerContent}>Profile</Text>
    </TouchableOpacity>
    <TouchableOpacity onPress={() => handleDrawerItemPress('Requests')}>
      <Text style={styles.drawerContent}>Requests</Text>
    </TouchableOpacity>
    <TouchableOpacity onPress={() => handleDrawerItemPress('Courses')}>
      <Text style={styles.drawerContent}>Courses</Text>
    </TouchableOpacity>
    <TouchableOpacity onPress={() => handleDrawerItemPress('SPgroups')}>
      <Text style={styles.drawerContent}>Senior Project Groups</Text>
    </TouchableOpacity>
    <TouchableOpacity onPress={() => handleDrawerItemPress('Department')}>
      <Text style={styles.drawerContent}>Department</Text>
    </TouchableOpacity>

    <TouchableOpacity style={styles.button} onPress={handleLogOut}>
      <Text style={styles.buttonText}>Log Out</Text>
    </TouchableOpacity>
  </View>
)
}

```

```

<View style={styles.bottomNavBar}>

  <TouchableOpacity onPress={() => handleBottomNavBar('Notifications')}>
    <MaterialIcons name="notifications" size={24} color="white" style={styles.bottomNavBarContent}/>
  </TouchableOpacity>
  <TouchableOpacity onPress={() => handleBottomNavBar('Home')}>
    <MaterialCommunityIcons name="home-variant" size={24} color="white" style={styles.bottomNavBarContent}/>
  </TouchableOpacity>
  <TouchableOpacity onPress={() => handleBottomNavBar('Messages')}>
    <Ionicons name="mail-sharp" size={24} color="white" style={styles.bottomNavBarContent}/>
  </TouchableOpacity>
</View>
</View>

```

Profile.js:

This code defines a function `renderTimetable` which renders a timetable based on the selected day (either Monday or Tuesday) with hardcoded schedule items for each day.

```

const renderTimetable = () => {
  // Logic to render timetable based on selected day
  switch (selectedDay) {
    case 'Mon':
      return (
        <>
        {/* Timetable for Monday */}

        <View style={styles.redrec}><Text style={styles.subsubtext}>10:00</Text><Text style={styles.subsubrec}>
        <View style={styles.greenrec}><Text style={styles.subsubtext}>10:00</Text><Text style={styles.subsubrec}>
        <View style={styles.bluerec}><Text style={styles.subsubtext}>10:00</Text><Text style={styles.subsubrec}>
        <View style={styles.yellowrec}><Text style={styles.subsubtext}>10:00</Text><Text style={styles.subsubrec}>
        <View style={styles.redrec}><Text style={styles.subsubtext}>10:00</Text><Text style={styles.subsubrec}>
        <View style={styles.greenrec}><Text style={styles.subsubtext}>10:00</Text><Text style={styles.subsubrec}>
        <View style={styles.bluerec}><Text style={styles.subsubtext}>10:00</Text><Text style={styles.subsubrec}>
        <View style={styles.yellowrec}><Text style={styles.subsubtext}>10:00</Text><Text style={styles.subsubrec}>
      </>

    );
    case 'Tue':
      return (
        <>
        {/* Timetable for Tuesday */}

        <View style={styles.redrec}><Text style={styles.subsubtext}>10:00</Text><Text style={styles.subsubrec}>
        <View style={styles.greenrec}><Text style={styles.subsubtext}>10:00</Text><Text style={styles.subsubrec}>
        <View style={styles.bluerec}><Text style={styles.subsubtext}>10:00</Text><Text style={styles.subsubrec}>

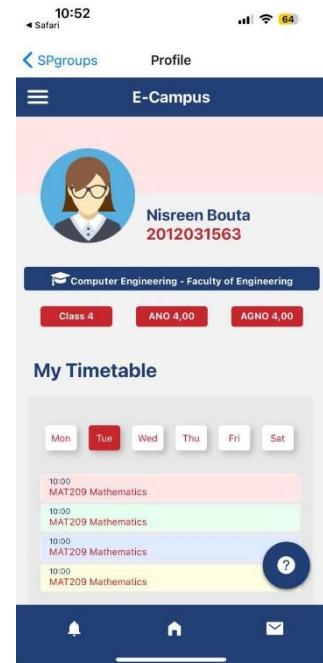
```

This code creates a UI component displaying a "My Timetable" heading followed by selectable weekdays, where the selected day's background color changes upon selection. It renders a timetable based on the selected day using the `renderTimetable()` function.

```

<Text style={styles.heading}>My Timetable</Text>
<View style={styles.card}>
  <View style={styles.weekdaysContainer}>
    {[ 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat' ].map((day) => (
      <TouchableOpacity
        key={day}
        style={[
          styles.weekdayRec,
          selectedDay === day && { backgroundColor: '#C8272E' },
        ]}
        onPress={() => setSelectedDay(day)}
      <Text style={[styles.weekdayText, selectedDay === day && { color: '#FFFFFF' }]}>
        {day}
      </Text>
    )})
  </View>
  /* Render timetable based on the selected day */
  <ScrollView>
    <View style={styles.recscontainer}>
      {renderTimetable()}
    </View>
  </ScrollView>
</View>

```



Courses.js:

This code defines a React component CourseDetailsScreen that takes route parameters (courseId, courseName, and instructor) and renders them within a View. It displays the details of a specific course including its ID, name, and instructor, centered within the view.

```

esktop\E-campus\E-campus\screens\department.js ) => {
  // EXTRACT COURSE DETAILS FROM route.params
  const { courseId, courseName, instructor } = route.params;

  return (
    <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
      <Text>Course ID: {courseId}</Text>
      <Text>Course Name: {courseName}</Text>
      <Text>Instructor: {instructor}</Text>
    </View>
  );
};

```

This code renders a section titled "Current Courses" followed by a horizontally scrollable view of course cards. Each course card contains information such as course ID, name, and instructor. Tapping on a course card navigates to a screen displaying detailed information about that course.

```
<Text style={styles.heading}>Courses</Text>

<View style={styles.headingRec}>
    <Text style={styles.headingRecText}> Current Courses</Text>
</View>
<ScrollView
    horizontal
    style={{ flex: 1 }}
    contentContainerStyle={{ paddingHorizontal: 20 }}>
    <View style={styles.cardContainer}>

        <TouchableOpacity
            style={styles.courseCardRed}
            onPress={() =>
                navigation.navigate('CourseDetails', {
                    courseId: 'MAT202',
                    courseName: 'Mathematics',
                    instructor: 'Dr. Alex',
                })
            }
        >
            <Text style={styles.cardGrayText}>MAT202</Text>
            <Text style={styles.cardBlueText}>Mathematics</Text>
            <Text style={styles.lessonred}>Dr. Alex</Text>
        </TouchableOpacity>
    </View>

```

