An Najah National University
Faculty of Engineering & Information Technology
Department of Computer Engineering

Software Graduation Project 1

Jamati Platform

Prepared by:

Malik Tayseer Ali

Shaheen Ali Abbas

Supervised by:

Dr. Ashraf Armoush

June 2025

Acknowledgment

We extend our sincere gratitude to everyone who supported the successful creation of *Jamati Platform*, a E-Learning website and mobile app.
Special thanks to our academic supervisor, Dr.Ashraf Armoush , for his valuable guidance and encouragement throughout the project.
As the developers, we take pride in building this platform to support and uplift freelancers in our occupied Palestine.
We are especially grateful to our families and friends for their unwavering motivation and support.
Finally, we appreciate everyone who offered feedback and encouragement during our development journey.

Abstract

Jamati is a comprehensive e-learning platform developed to empower the education and communication between universities students where the students can share **posts** and images and interact with posts and add comments and also provide chatting system between the students, the second main feature is the **Projects** where the students can share projects and upload files and links to support the project also the other students can take this files and take benefit from at also they can rate the projects and filter it by colleges, authors, title and issue date ,the third main feature **courses and payment** where the courses managed by the instructors how have the Eligibility to publish courses where the course has list of videos that cover the course content the students can rate the course and write feedbacks also there is a real time groups chatting system for each course . there is three user roles **System Admin, Instructor** and **Student.** We also implement user **Authentication and Authorization** and email code verification to guarantee the Credibility.

Jamati is built using **MongoDB, REST Express Nodejs Framework** for the backend, **React Framework** for web and **Expo React Native** for mobile app.

Table of Contents

Chapter 1: Introduction

1.1 Statement of the Problem

During our time at the university, we noticed that many students face difficulties in finding diverse project ideas and writing reports in a correct and effective manner. Additionally, they often struggle to find suitable courses that align with their academic majors. Therefore, we developed **jamati** platform to serve as a comprehensive space where students can access all the resources they need to enhance and develop their academic skills.

1.2 Objectives of the Work

Jamati aims to develop a full e-learn platform accessible via a mobile application and web portal, enabling An Najah National University Students to communicate and sharing projects with each other's and take courses that fit t Their educational path. The platform provides role-based access for Student, Admin and Instructor, courses, projects and posts creation and categorization,

real-time chat, email verification, secure payments via Stripe, and customizable profiles. The backend is powered by Nodejs and MongoDB for robust and saleable data handling, while the frontend is built with React and React Native for a responsive and cross-platform experience. By leveraging modern technologies, *Jamati* seeks to empower students and make the university life easier.

1.3 Scope of the Work

The project includes the design and development of a cross-platform system featuring structured posts publishing, projects categories, payment for courses, messaging, profile customization, and secured financial transactions. The backend leverages expressjs and expressjs REST Framework for scalable API development, while MongoDB serves as the non-relational database. The frontend is implemented in React and React Native, offering a consistent Web and Mobile experience with intuitive UI components. Features such as email verification, role-based UI adjustments, profile management, and in-app communication ensure a seamless workflow. Stripe is used to handle checkout and payment confirmations. The system is designed with scalability and user-friendliness in mind.

1.4 Significance of the Work

Jamati addresses a critical gap among An-Najah University students by offering a comprehensive e-learning platform tailored to their specific needs. It empowers students by enhancing their academic and practical skills. With a role-based design, the platform ensures a user-friendly experience for both students and instructors.

Features such as real-time messaging, group chats for courses, and integrated payment systems facilitate seamless communication and collaboration.

1.5 Organization of the Report

This report is structured into several chapters. The first chapter introduces the project, its purpose, scope, and significance. Chapter two discusses the project's constraints, standards, and the foundational coursework that informed its development. Chapter three presents the literature review, comparing existing e-learn platforms and identifying gaps. Chapter four details the design and implementation, covering database schema, backend logic, and frontend development. Chapter five outlines testing results, challenges, and final outcomes. The final chapter summarizes conclusions and proposes future enhancements. Supporting documents and code references are included in the appendices.

Chapter 2: Constraints, Standards/Codes, and Earlier Coursework

2.1 Constraints and Limitations

- Internet & Infrastructure Limitations: In occupied Palestine, unstable internet and limited access to payment gateways can restrict real-time communication, deployment, and testing across different environments.
- Stripe Limitations for Region: Stripe is not natively supported in Palestine, requiring workaround configurations or testing in simulated environments for payment processing.
- Time Constraints: Developing a full-stack mobile app and web site and backend within the given academic timeline required careful planning and prioritization of core features over optional enhancements.

- Real-Time Features Complexity: Implementing secure, scalable chat functionality and notification systems required deeper research into WebSocket protocols, express, and frontend message handling.
- Frontend Mobile Optimization: Ensuring a consistent and user-friendly experience across different screen sizes and Android devices presented UI/UX and performance challenges.

2.2 Standards and Codes

2.2.1 Backend – Nodejs & Nodejs REST Framework

The backend was developed using Nodejs, a high-level java script web framework known for its scalability and security. Nodejs REST Framework was used to create RESTful APIs that handle authentication, user roles, service listings, and real-time data processing. JWT tokens.

2.2.2 Database – MongoDB

MongoDB was chosen as the non-relational database system due to its scalability, support for complex queries and large, and wide compatibility with Nodejs. The database schema includes multiple interrelated tables—users, roles, posts, comments, projects, messages, chats, reports, courses and videos —designed using reference objectId constraints and indexes for performance.

2.2.3 Frontend – React (Web) & React Native (Mobile)

The frontend of *Jamati* is developed using two powerful JavaScript frameworks: **React** for the web interface and **React Native** for the mobile application.

- The **web version**, built with React and CRA, offers fast development cycles, modular component architecture, and responsive design. It utilizes modern libraries such as **Tailwind CSS** for clean, consistent

styling and reusable UI components. Navigation is managed using **Modern React Router**, and role-based routing ensures different UI experiences for System Admin Instructor and Student.

- The **mobile version**, developed using React Native, brings a native-like experience on Android devices with shared logic from the web platform. It features smooth navigation using **React Navigation**, sidebars, and screen transitions. State is managed using React hooks and Context API to ensure consistent user sessions and interaction across the app.

  Together, both interfaces maintain design consistency, role-based access control, and seamless integration with the Nodejs backend APIs, delivering a unified user experience across platform

2.2.4 Payment Integration

Strip Integration: The inclusion of the stripe dependency signifies the

integration of payment processing functionalities. This allows the API to handle transactions, subscriptions, or other financial operations securely, leveraging Stripe's robust payment gateway.

2.3 Earlier Coursework

Our studies in the Computer Engineering Department provided a strong foundation for developing the *jamati* platform. Key courses and experiences that contributed to this project include:

- **Web Programming**: Focused on modern tools such as React and Tailwind CSS, which enabled us to build responsive and modular interfaces for the web version of the platform.

- **Database Systems:** Covered relational database design, normalization, and MongoDB, all of which were essential for structuring the database and managing relationships between users, and collection's.
- **Software Engineering:** Provided insights into UML diagrams, software development methodologies (such as Agile), and project planning—skills that guided our team's development workflow and documentation.
- **Computer Networks**: Introduced core concepts such as HTTP, WebSocket's, and RESTful API communication, which we applied while implementing real-time chat and API-based frontend-backend integration.

  Beyond coursework, we expanded our skills through self-directed learning in react native, Nodejs REST Framework, and third-party integrations like Strip, which we used to handle secure user authentication and payments in a regionally compliant manner.

Chapter 3: Literature Review

Several platforms exist to support student learning, project sharing, and course delivery in both academic and professional contexts. However, these platforms often address specific features in isolation, without providing a fully integrated academic experience tailored to local university environments.

One of the most relevant platforms to our work is the **ANNU Digital Library**, which allows students at An-Najah National University to upload and access academic projects. While useful for storing and retrieving student work, the platform lacks interactive features such as project rating, detailed filtering, and user-friendly navigation. In **Jamati**, we enhanced this concept by enabling students to **rate and review projects**, **filter them by college, author, title, and issue date**, and **easily browse** through the repository using a modern interface.

These improvements make it significantly easier for users to discover high-quality academic content and benefit from peer contributions.

For the course-related features, our platform drew inspiration from **Udemy**, a popular online learning platform offering a wide range of courses in various domains. While Udemy is designed for a global audience with diverse interests, it does not cater specifically to the academic needs of university students. In contrast, **Jamati focuses on university-level courses**, particularly those relevant to An-Najah students. Instructors who are verified members of the university community can **publish structured video-based courses**, and students can **rate and provide feedback** on the courses. Furthermore, we incorporated **real-time group chatting** for each course to encourage collaboration and discussion, a feature not commonly available in most standard learning platforms.

What sets Jamati apart is its holistic approach to academic life. It **integrates project sharing, academic social networking, and course management** into a single user-friendly system. With clearly defined user roles (System Admin, Instructor, and Student), secure **user authentication and email verification**, and a **mobile-first approach** using modern frameworks such as **React, React Native (Expo), Node.js, Express, and MongoDB**, Jamati ensures a secure and seamless experience tailored to the needs of local students.

In summary, while existing platforms provide individual features for either project sharing or course delivery, **Jamati bridges the gap by offering an all-in-one platform** that enhances academic communication, content

Chapter 4: Methodology

4.1 Tools, Methods, and Programming Languages

4.1.1 Tools

- **Visual Studio Code:** Used as the primary IDE for writing and debugging both backend (Nodejs) and frontend (React/React Native) code.
- **Android Studio:** Employed for testing and running the React Native mobile app on emulators and physical devices.
- **MongoDB Tools (MongoDB Server):** Used for managing the database.
- **GitHub:** Used for version control, collaboration, and managing code through branches and pull requests.
- **Postman:** Helped with documentation, task testing the backend.
- **Canva:** Used for designing wireframes, user interfaces, and flow diagrams before implementation.
- **Cloudinary:** Used for store images, videos and files

4.1.2 Programming Languages & Frameworks

Frontend Development:

- **Web App:** Built using **React**, a JavaScript library for building responsive, component-based UIs. Styled using **Tailwind CSS**.
- **Mobile App:** Developed using **React Native**, which allows us to write cross-platform mobile apps using JavaScript and native modules for Android support.

Backend Development:

- **Nodejs (JavaScript):** The backend framework used to create RESTful APIs, handle business logic, and manage user roles and authentication. Nodejs REST Framework (expressjs) helped structure and serialize the APIs.
- **JWT Authentication:** Used for secure user login and session management, integrated with Nodejs.

Third-Party Integration:

- **Strip:** Integrated for payment confirm and represent a real payment process.
- **lucide-react:** for icons in react framework.

4.1.3 Database

- **MongoDB:** A scalable non-relational database used to manage structured data including users, projects, courses, payment, order, messages, feedback and posts.
- The database schema is normalized and includes objectId constraints to preserve data integrity.
- **Atlas:** Used to visually manage and monitor the database during development and testing.

Chapter 5: Results and Discussion

In this chapter, we present the results of the *jamati platform* project by showcasing its main features, supported by screenshots and explanations. The system was developed with separate interfaces for web and mobile, and it includes role-based functionality for admin, instructors and students. During development, differences in design and technical decisions between web and mobile versions led to some variations. In such cases, we adopted the most effective implementation. This chapter outlines both shared and role-specific

features to provide a comprehensive overview of the platform's capabilities and performance.

5.1 Web and Mobile Features

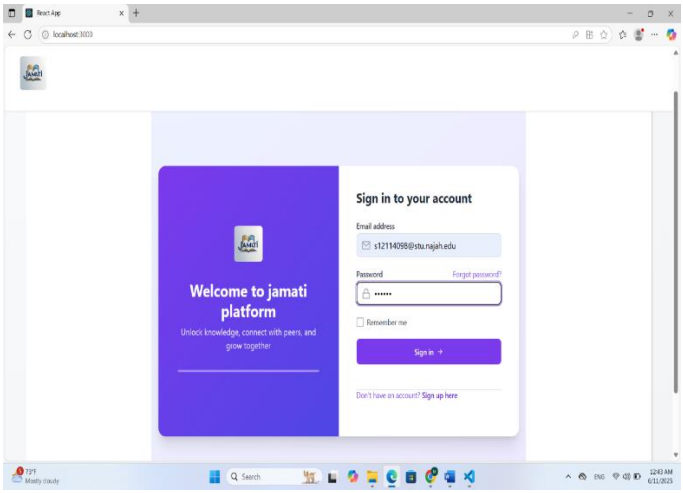
5.1.1 Common Features Among Users

These features are available to all users regardless of their role (Student or Admin or Instructor), and they help create a unified and intuitive experience within the mobile app.

5.1.1.1 Login Screen

The **login screen** is the gateway to accessing the *Jamati* mobile application and Website. It is designed with clarity, simplicity, and user accessibility in mind. Users are required to input a **valid email address** and **password** to sign in. If incorrect credentials are entered, the system displays an **error alert** to guide the user. Additional features include:

- A **"Forgot your password?"** link to initiate the password recovery process.
- A **"Sign up"** link for users who do not have an existing account.

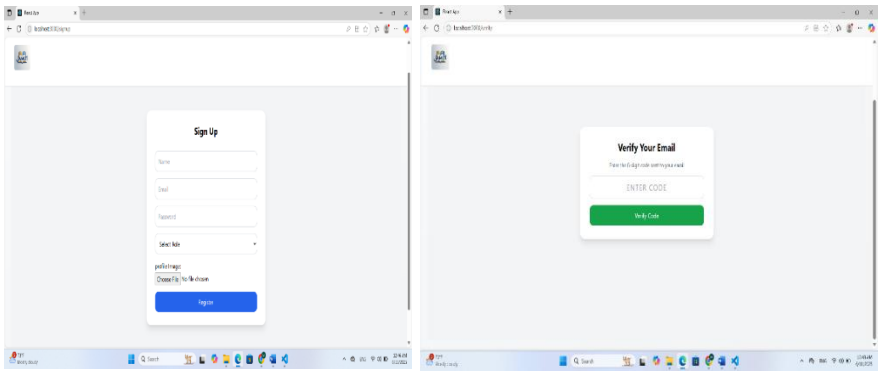

Welcome to jamati platform
Unlock knowledge, connect with peers, and grow together
Sign in to your account
Email address
s12114098@stu.najah.edu
Password
Forgot password?
Remember me
Sign in
Don't have an account? Sign up here

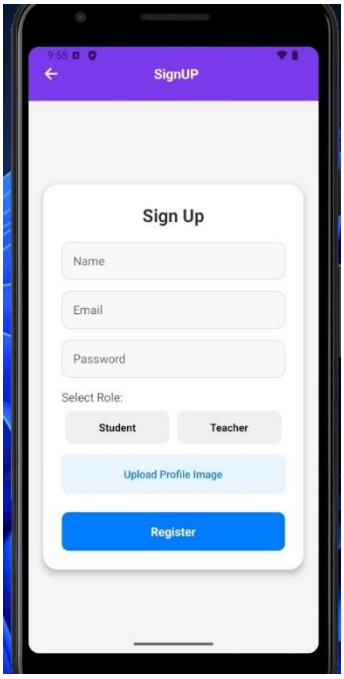

9:50
Login
JAMATI
Welcome to Jamati Platform
Login
Email
Password
Forgot Password?
Login
Don't have an account? Sign Up

5.1.1.2 Register

The **registration screen** (Sign Up) enables new users to create an account by entering their **username**, **email address**, **password**, **role, personal photo** and confirming it.. Once completed, clicking **"Create account"** registers the user and prepares them for email verification and access to the platform.

This intuitive flow ensures that all users—whether new or returning—can access the platform quickly, securely, and with role-specific functionality from the very beginning.



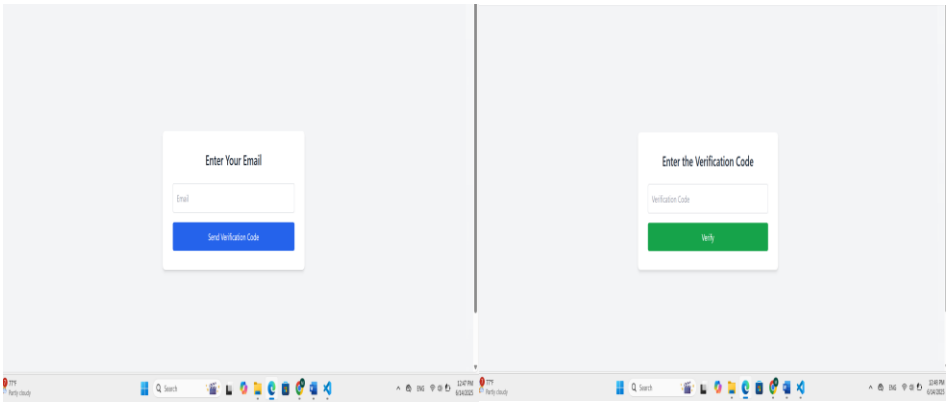


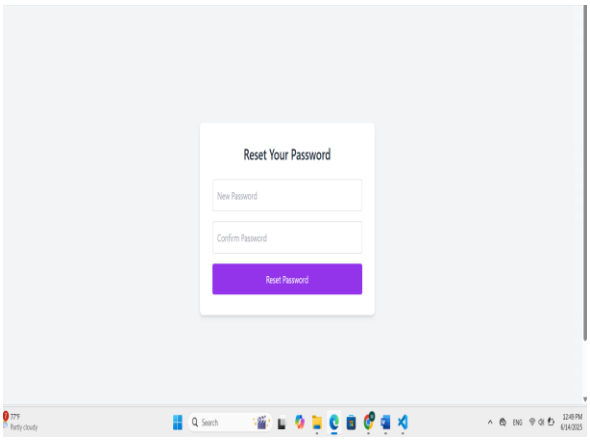
5.1.1.3 Forgot Password

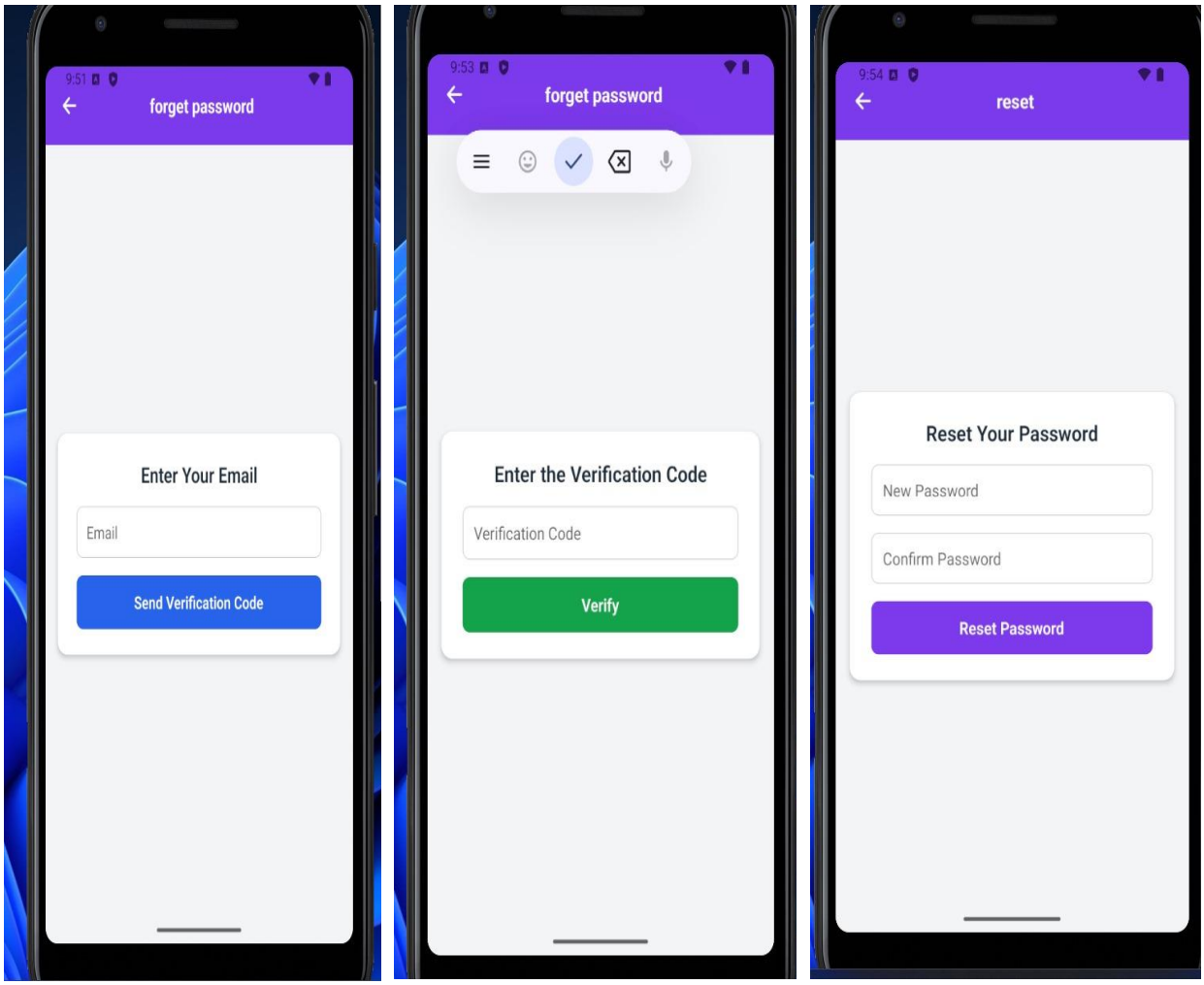
If a user forgets their password, the application provides a straightforward recovery process to regain access.

- If the entered email is found in the system, the application sends a **verification code** to that email address.
- This code is then used in the next step to **reset the password securely**.

This feature ensures account recovery is both secure and user-friendly, using email verification as a reliable method of identity confirmation.





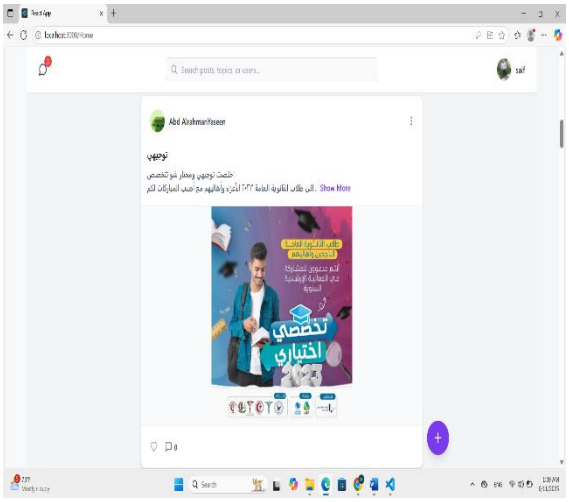


5.1.1.4 Home Page & Navigation Sidebar

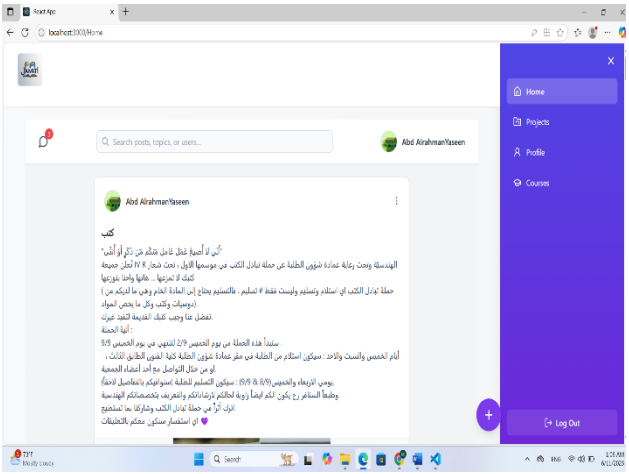
After login, all users are directed to the home page, which serves as the main hub for exploring Posts and community between users. At the top, a search bar allows users to quickly find specific posts according to the title or description and also the chatting button to go to the chatting page also user name and personal photo.

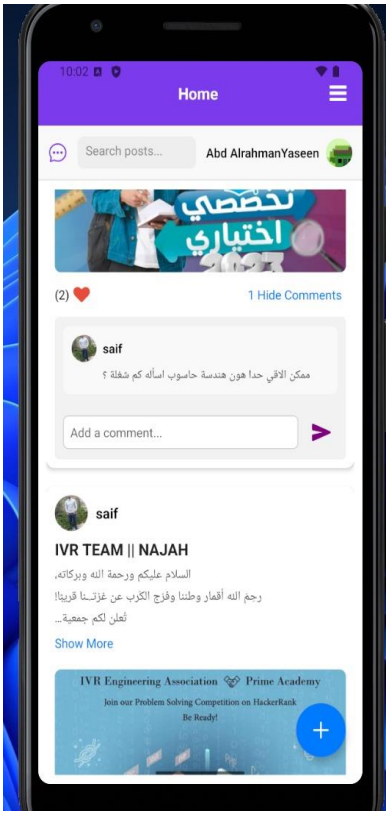
Below it, users can scroll vertically to view posts **comments** and interact with it, each displaying essential details like title, image, and description.

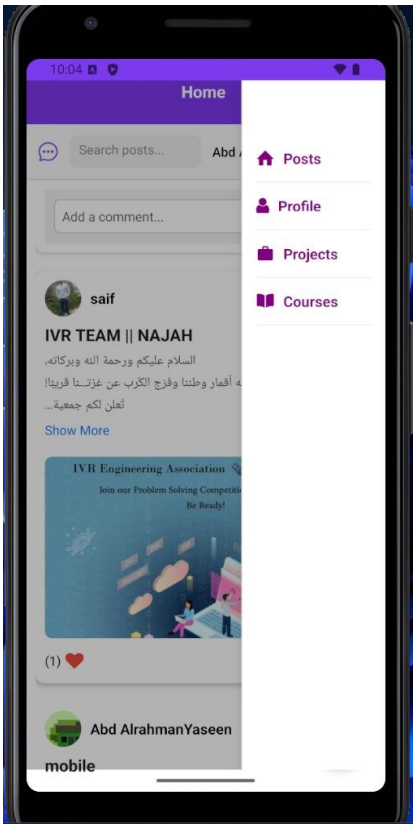
A fixed **navigation sidebar** at the right provides quick access to four main sections: Home, Projects, Profile, and Courses—ensuring smooth navigation throughout the app.


Abd AlrahmanYaseen
توجيهي
خلصت توجيهي ومعدل شو تخصص
الى طلاب الثانوية العامة 2025 المتراجد وأهاليهم مع اصعب امتحانات لكم
Show More
تخصصي
اختياري
saif


1
saif
ممكن الاقي حدا هون هندسة حاسوب اسأله كم شغلة ؟
Write a comment...


Home
Projects
Profile
Courses
Log Out
Abd AlrahmanYaseen
كتب
"اني لا أضيع عمل عامل منكم من ذكر أو أنثى"
الهندسة وتحت رعاية عمادة شؤون الطلبة على حملة تبادل الكتب في موسمها الاول ، تحت شعار " لا تحملي جميعها
كتبك لا تعرفها .. هالها واحنا بوزعها
حملة تبادل الكتب اي استلام وتسليم وليست فقط تسليم . فالتسليم يخلع إلى العادة الخام وهي ما لديكم من)
(روسيات وكتب وكل ما يخص المواد
تفضل عنا وجب كتبك القديمة لتفيد غيرك
أتية الحملة
سنبدأ هذه الحملة من يوم الخميس 2/9 لتنتهي في يوم الخميس 9/9
أيام الخميس والسبت والاحد : سيكون استلام من الحملة في مقر عمادة شؤون الطلبة كلية الفنون التطبيقي الدائر .
او من خلال التواصل مع أحد أعضاء اللجنة العضية
يومي الاربعاء والخميس(8/9 & 9/9) : سيكون التسليم للحملة إستوانيكم بالتفاصيل لاحقاً
وطبعاً الساغر رح يكون اكم ايضاً زوية لحالكم تارضاءاكم والتعريف بتخصصاتكم الهندسية
اذرك أثراً في حملة تبادل الكتب وشاركنا بما تستطيع
اي استمرار سنكون معكم بالتطبيقات





5.1.1.5 Add New Post Page

This page allows the users to insert a new post to the community and there is tow type of posts: with image and just text post

Create a New Post

Image Post | Text Post

Upload Image

Choose File No file chosen

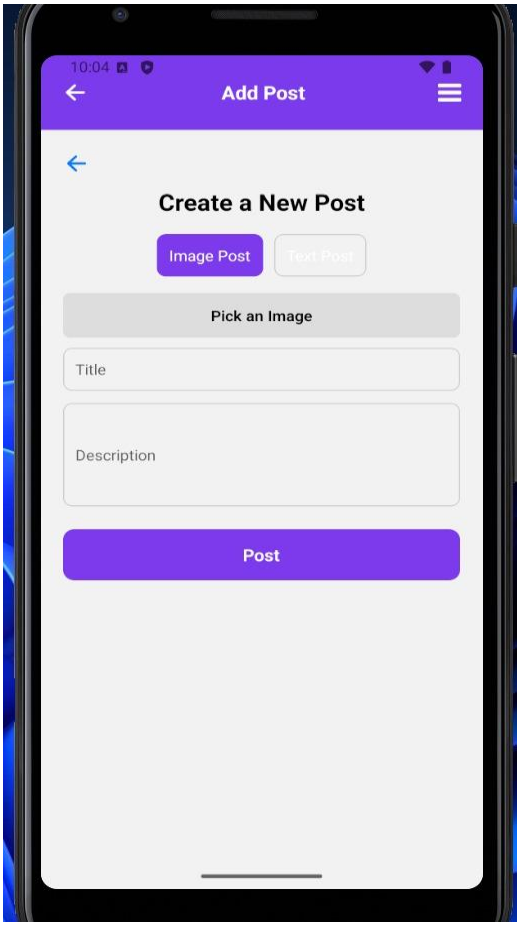
Title

Enter post title

Description

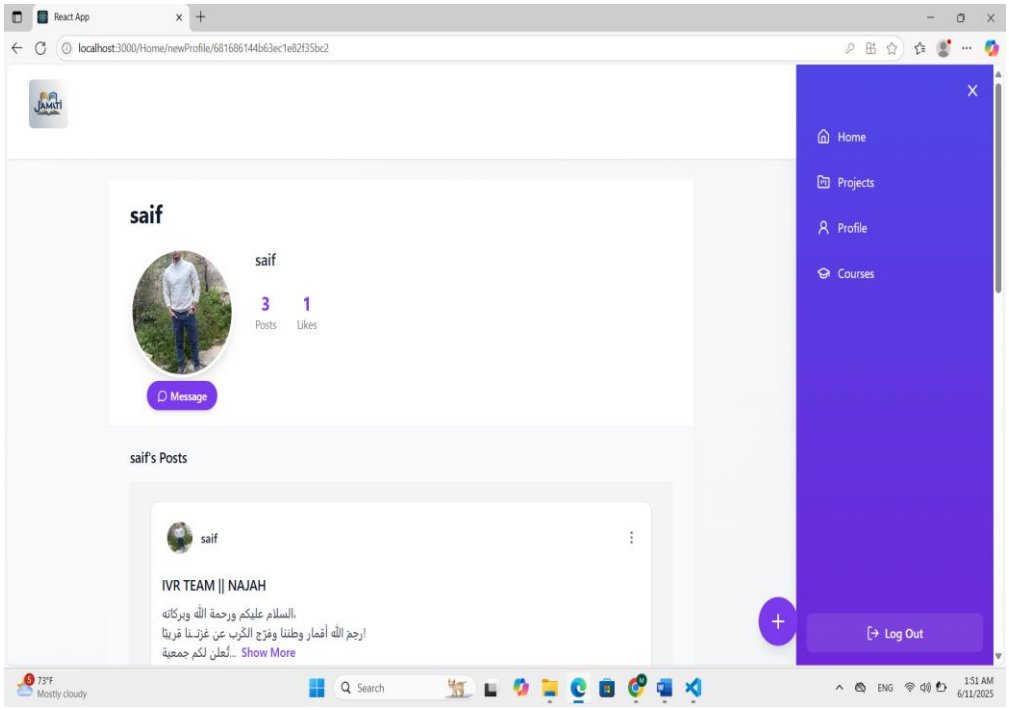
Enter post description

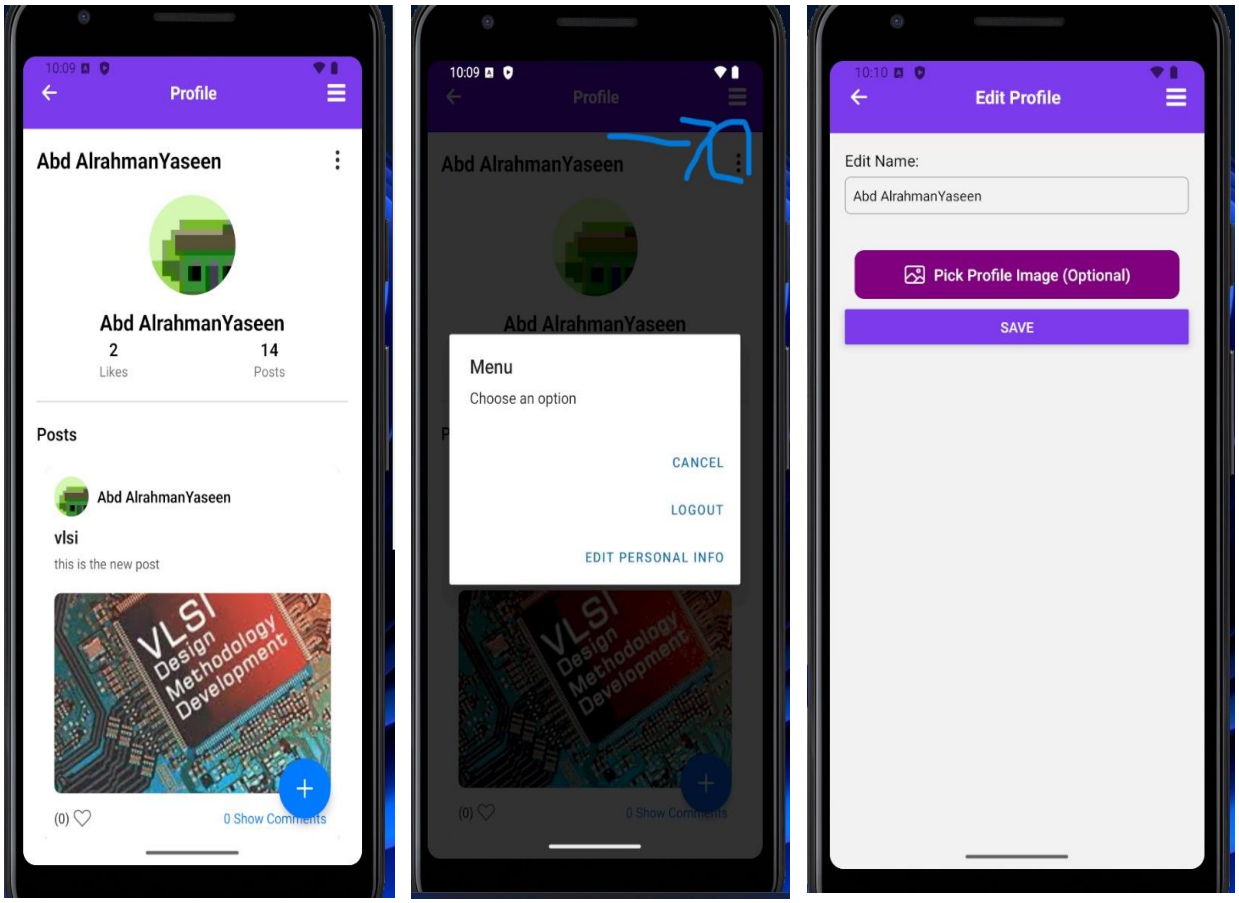
Post



5.1.1.6 Profile Page

Each user has personal profile page contained his posts and the other users can browse each other profiles and see posts comment and add reaction to the post also they can open new chat with each other by click on message button.



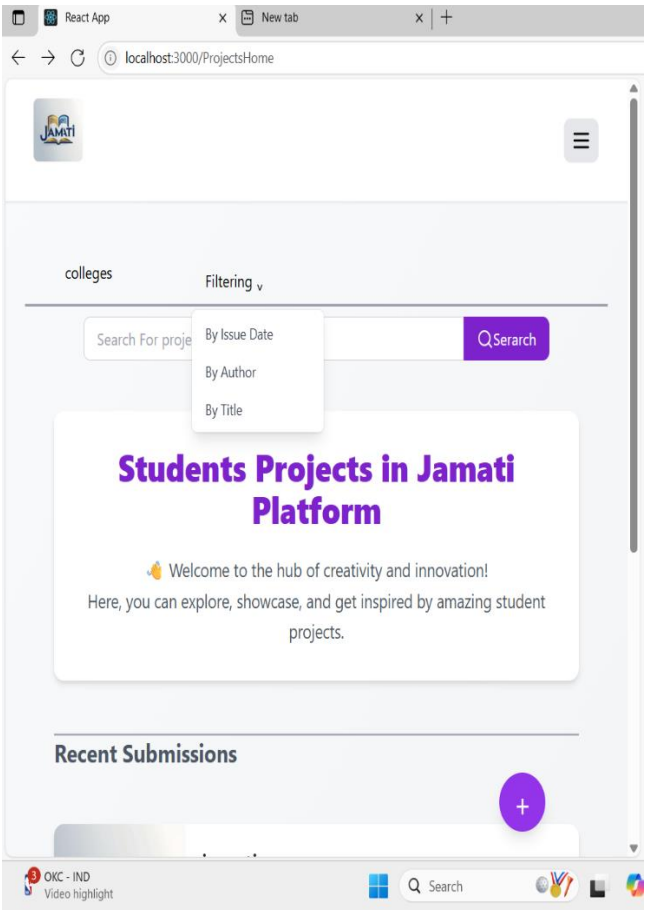


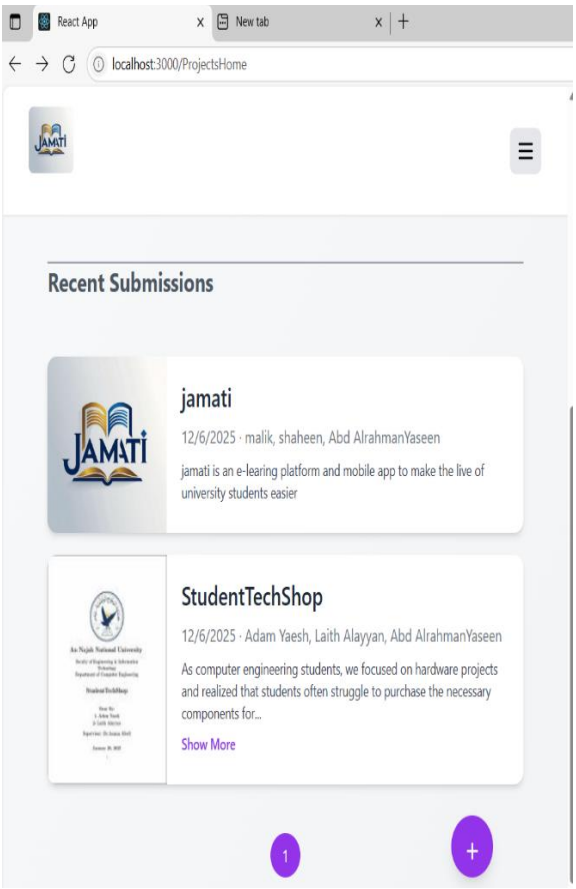
5.1.1.7 Main Projects Page

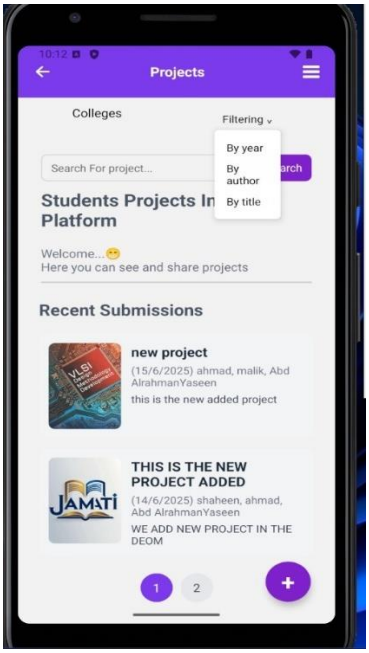
This page has a list of the projects that added by the students where in the main page the projects display with order if the submitting data (last added first)

- **Each project poster has an image, description, Authors, created at and title of the project**.
- **In the top navbar we can filter and search of specific post title or description.**
- **The Colleges button show the list of colleges that have a project**.
- **add project button that navigate to add project page**
- **list of projects and pagination where the user and browse between pages**

this is the page:



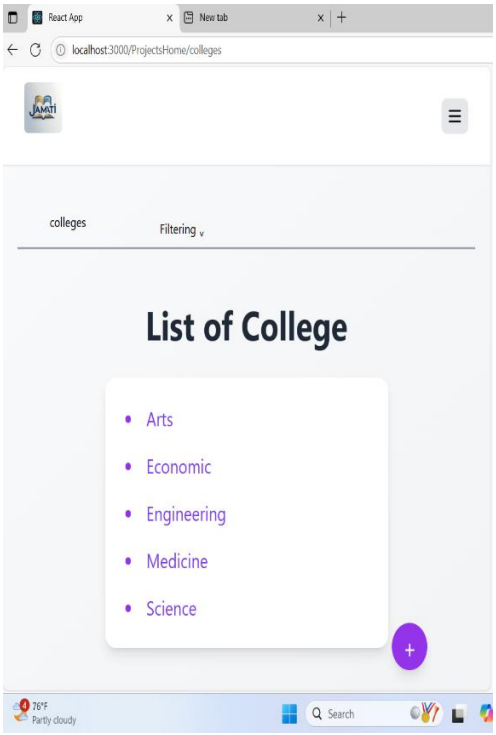


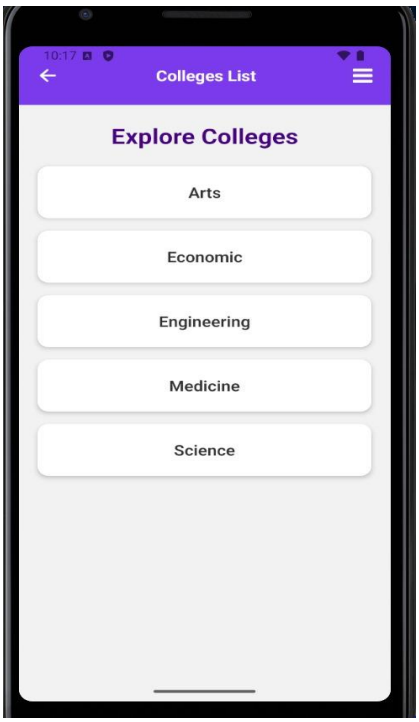


5.1.1.8 Colleges Page

The College page display list of colleges in the data base that have projects so if a specific college doesn't have any project, it's not display.

Economic. So, when choose I college then **filter** the projects according to this colleg

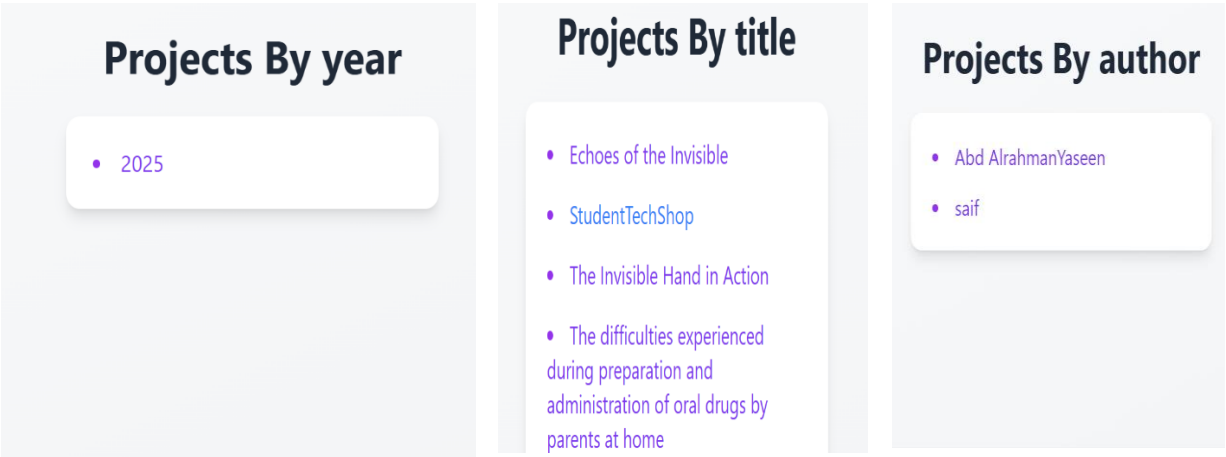




5.1.1.10 Filter:

There are three types of filtering:

- **By title: shows all titles of all projects to make the navigate on a certain title easier**
- **By issue date: filter the projects according to the year that the project added.**
- **By other : get a list of authors how was added a project.**

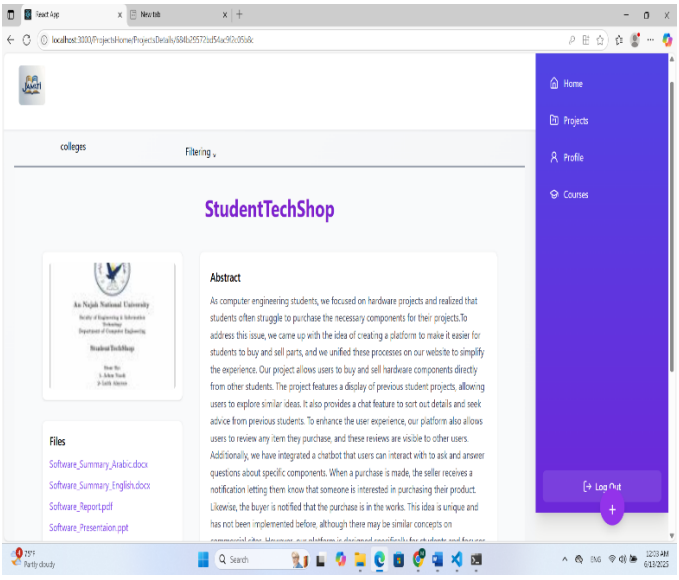


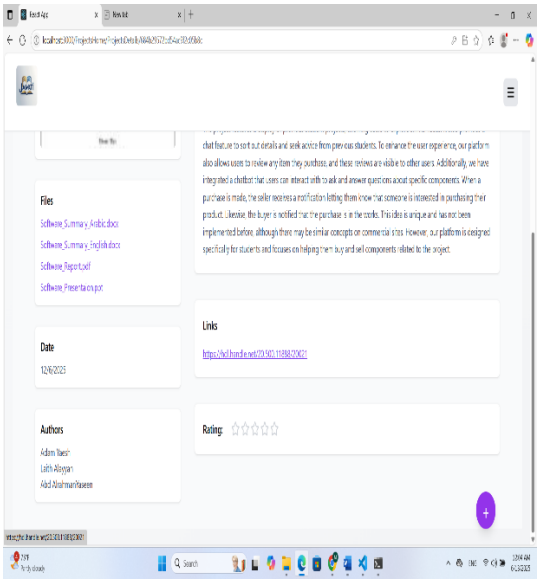
5.1.1.11 Project Details and Content Page:

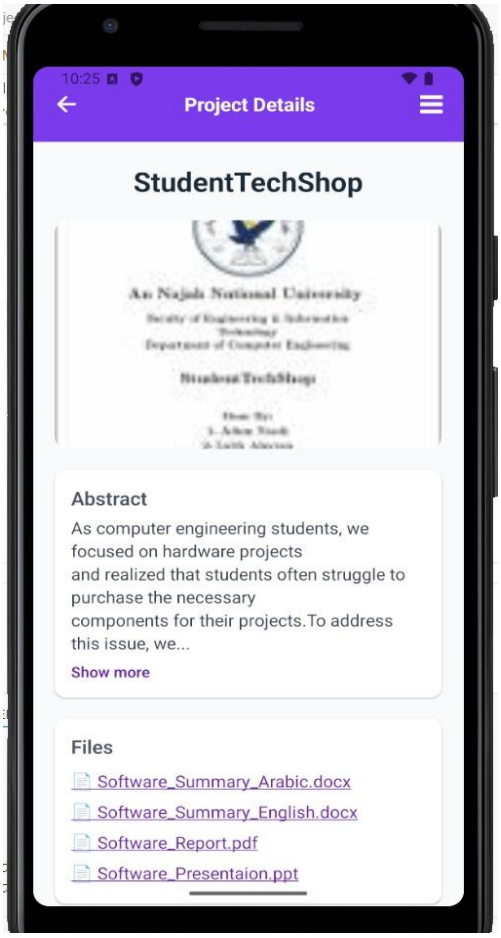
This page displays all project fields and content which is:

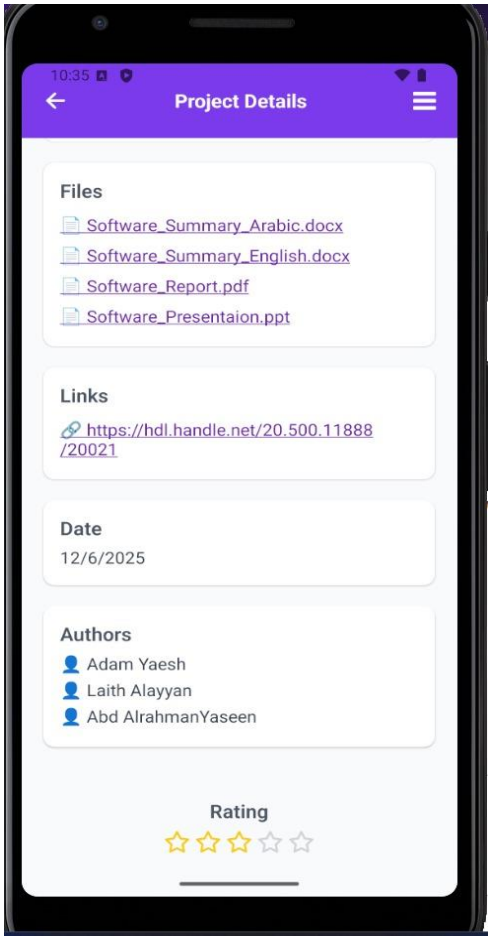
- Image poster
- Title
- Description
- List of files for the project
- List of links
- Date of published
- Authors names
- Rating of the project

The student can download or view the projects file and browse the links provided.

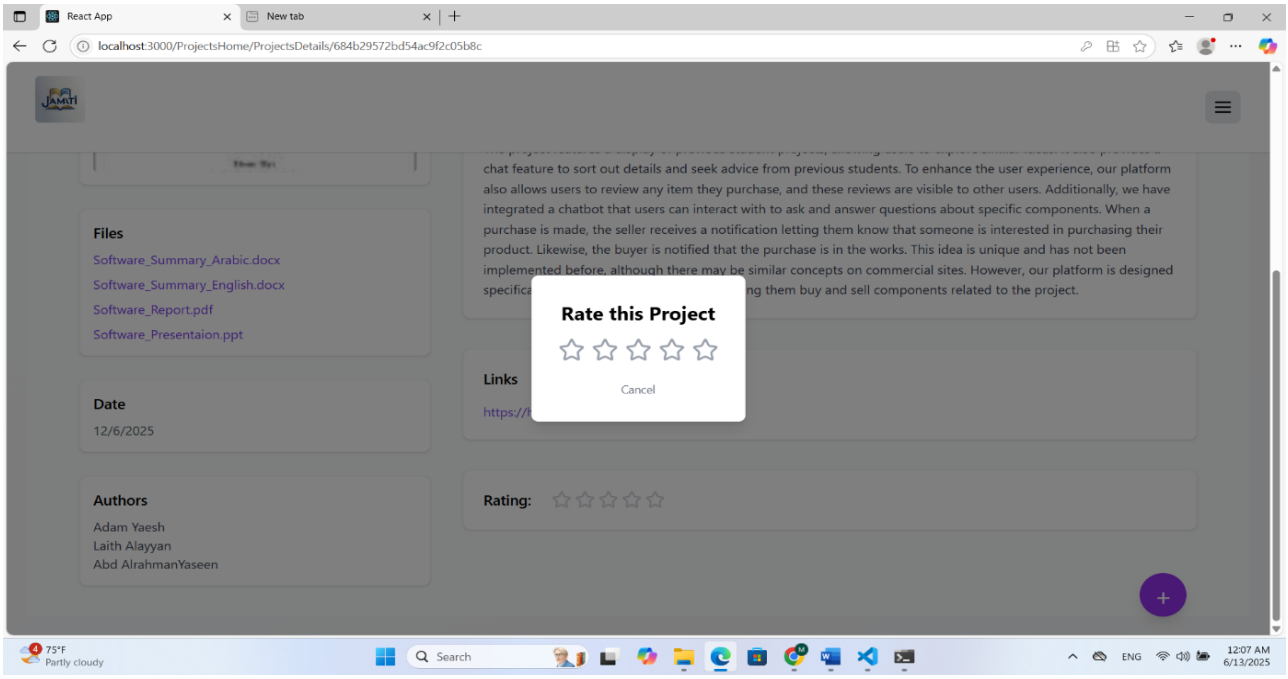

colleges
Filtering
StudentTechShop
Abstract
As computer engineering students, we focused on hardware projects and realized that students often struggle to purchase the necessary components for their projects.To address this issue, we came up with the idea of creating a platform to make it easier for students to buy and sell parts, and we unified these processes on our website to simplify the experience. Our project allows users to buy and sell hardware components directly from other students. The project features a display of previous student projects, allowing users to explore similar ideas. It also provides a chat feature to sort out details and seek advice from previous students. To enhance the user experience, our platform also allows users to review any item they purchase, and these reviews are visible to other users. Additionally, we have integrated a chatbot that users can interact with to ask and answer questions about specific components. When a purchase is made, the seller receives a notification letting them know that someone is interested in purchasing their product. Likewise, the buyer is notified that the purchase is in the works. This idea is unique and has not been implemented before, although there may be similar concepts on
Files
Software_Summary_Arabic.docx
Software_Summary_English.docx
Software_Report.pdf
Software_Presentaion.ppt
Home
Projects
Profile
Courses
Log Out

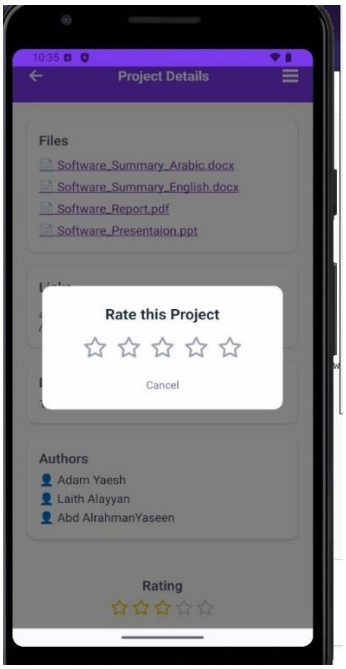

Files
Software_Summary_Arabic.docx
Software_Summary_English.docx
Software_Report.pdf
Software_Presentaion.ppt
Date
12/6/2025
Authors
Adam Yaesh
Laith Alayyan
Abd AlrahmanYaseen
Links
Rating:


10:25
Project Details
StudentTechShop
An Najah National University
Abstract
As computer engineering students, we focused on hardware projects and realized that students often struggle to purchase the necessary components for their projects.To address this issue, we...
Show more
Files
Software_Summary_Arabic.docx
Software_Summary_English.docx
Software_Report.pdf
Software_Presentaion.ppt


10:35
Project Details
Files
Software_Summary_Arabic.docx
Software_Summary_English.docx
Software_Report.pdf
Software_Presentaion.ppt
Links
https://hdl.handle.net/20.500.11888/20021
Date
12/6/2025
Authors
Adam Yaesh
Laith Alayyan
Abd AlrahmanYaseen
Rating

And in the rate of the project the user can rate the project when he click on the stares then a modal to set rate will be shown the user can rate just one time and if he rerates the project then overwrite on the previous rate .

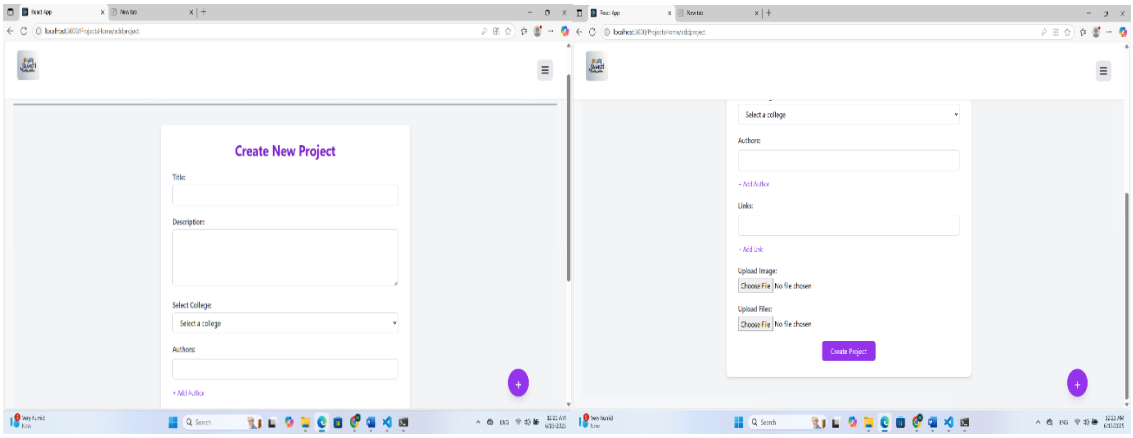


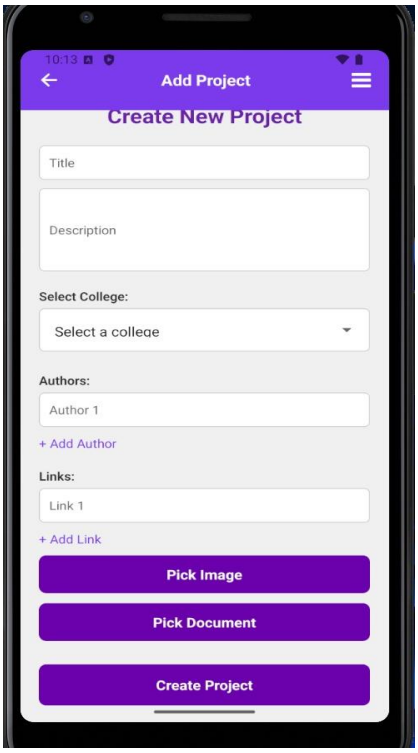


5.1.1.12 Add Project:

All students can add projects to the platform and this project can be seen by all user the admin also can view these projects and delete it

- The student can add list of students also how cooperate in building and making this project
- Can also add list of documents (pdf, text, word, PowerPoint)
- He can also select the college that the project belongs to it
- project title and description and image poster.





5.1.1.13 Courses Feature:

Jamati provide courses to the students in the system but this courses not free the instructor adds the course and also the price of the course so the students must buy the course to be able to view its content and witch the course video

5.1.1.14 Course Main Screen:

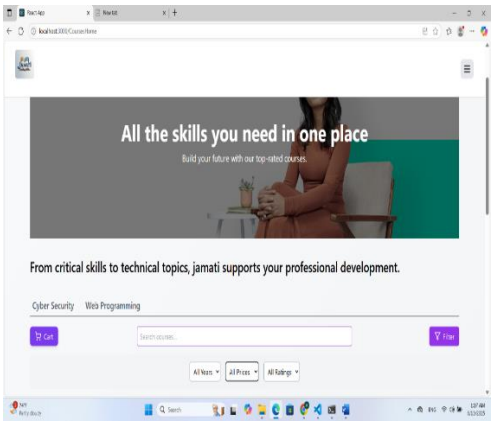
This page displays the category and filtering and search navbar, also the container of courses that display courses posters according the above.

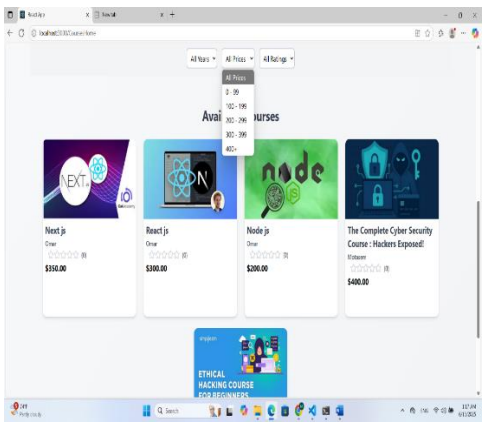
There is a search input to search on title or description of courses and filter according to three things:

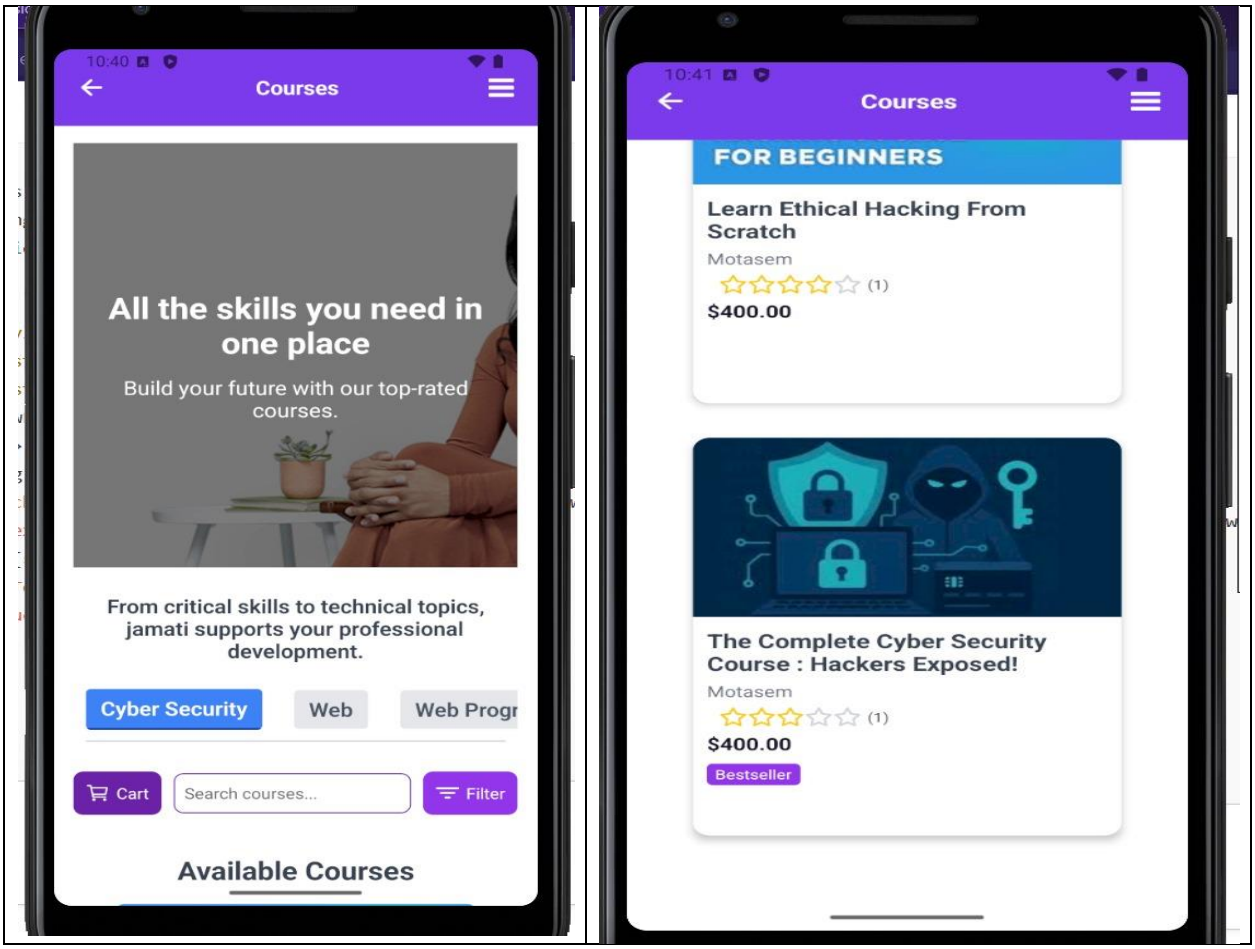
- **Year**
- **Prices Range**
- **Rate (1-5)**

Course Poster has the following:

- Image Poster
- Title of the course
- Price and discount, free word if the course has discount 100%
- Average rating of the course and number of how rate the course

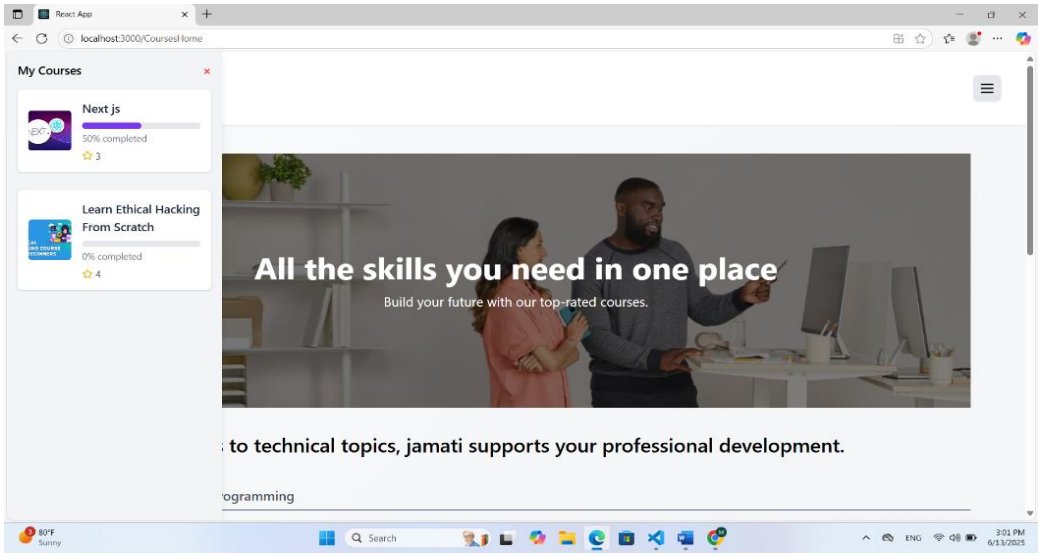


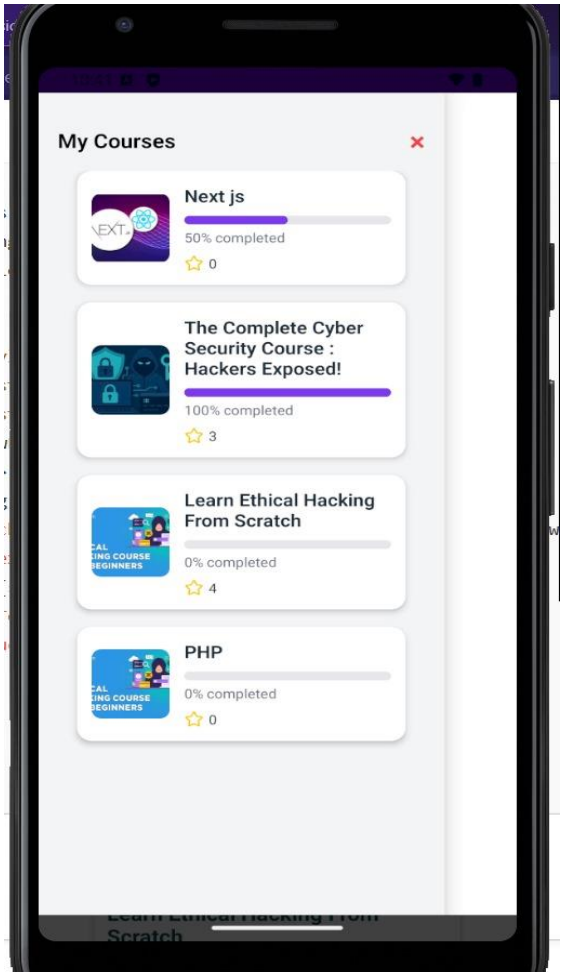




5.1.1.15 Courses Cart

To show all courses that the student bought and also the progress reached in each course:


React App
localhost:3000/Courses#home
My Courses
Next js
50% completed
3
Learn Ethical Hacking From Scratch
0% completed
4
All the skills you need in one place
Build your future with our top-rated courses.
to technical topics, jamati supports your professional development.
ogramming
80°F
Sunny
Search
ENG
3:01 PM
6/13/2025


My Courses
Next js
50% completed
0
The Complete Cyber Security Course : Hackers Exposed!
100% completed
3
Learn Ethical Hacking From Scratch
0% completed
4
PHP
0% completed
0

5.1.1.16 Course Details page

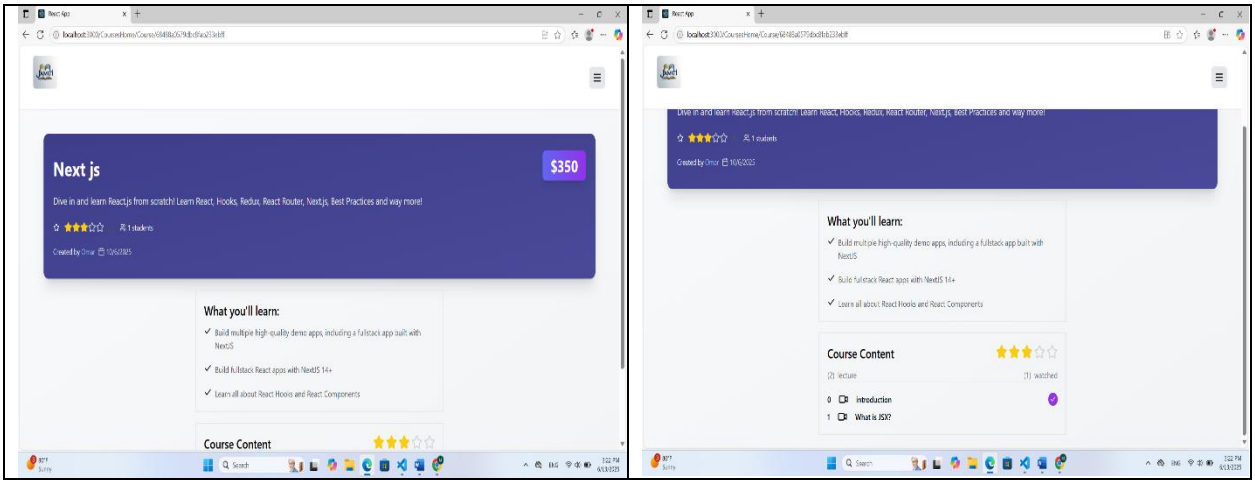
In this page we show all the course content and details:

The header displays the following: course name, description, average rate, number of students bought the course the instructor's name and when he crate the post, the price of the course.

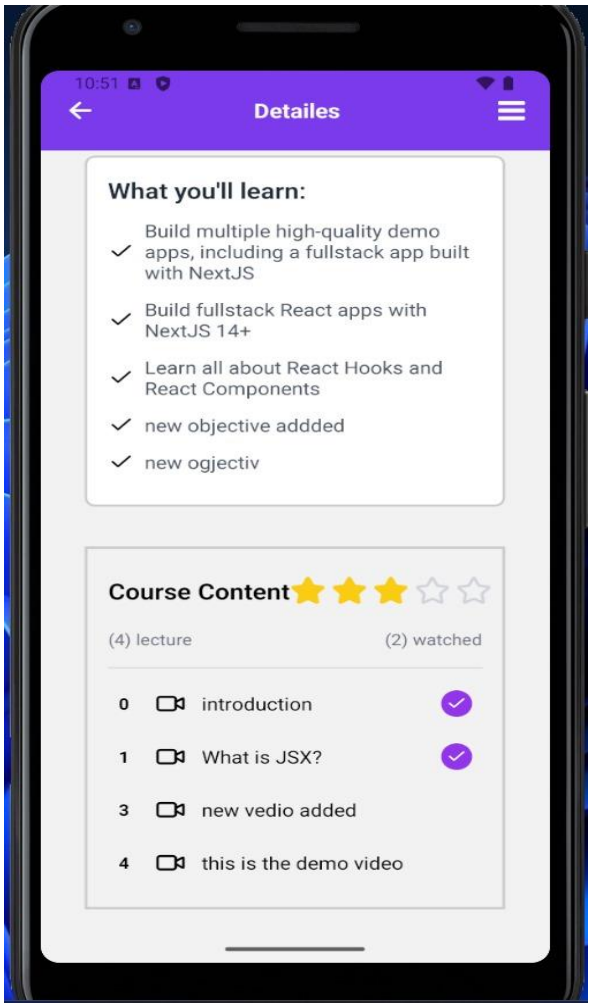
The objective expected from this course.

Course Content: in this part there is the current student info about the course like his rate and count of videos and count of watched videos (progress)

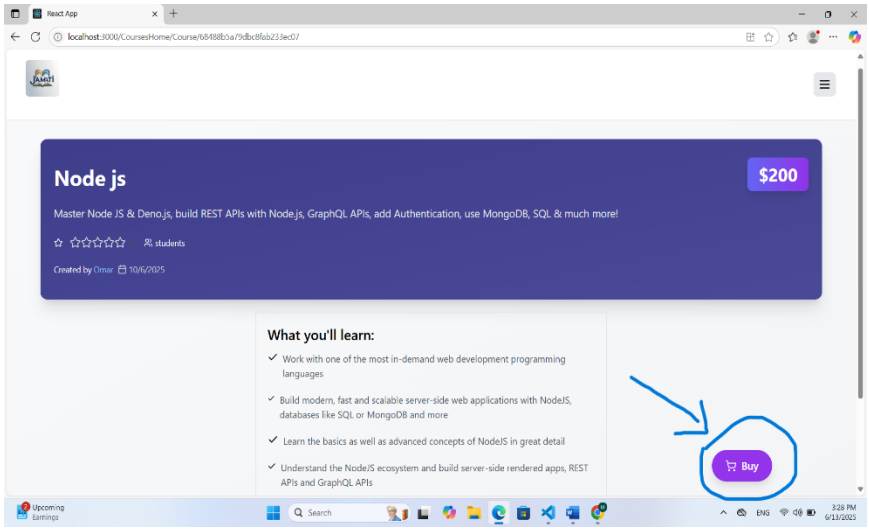
And also, the list of videos for this course if the student already watched the video, then there is a ✔tic will show save the current index that the student reached in the course. But this info will show just if the student is already bought this course if not just the list of videos name will show (no access to the course videos and the student not able to rate the course or give feedback about the course)




10:50
Detailes
Learn Ethical Hacking From Scratch
$400
Become an ethical hacker that can hack like black hat hackers and secure systems like cybersecurity experts
1 students
Created by Motasem
10/6/2025
What you'll learn:
Hack & secure WiFi & wired networks.
Create backdoors & Hack Windows.
Hack cloud servers.
Learn Network Hacking / Penetration Testing.
Course Content

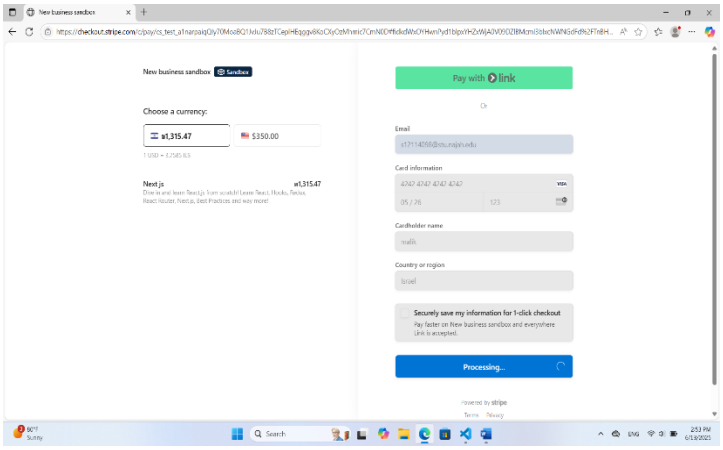

10:51
Detailes
What you'll learn:
Build multiple high-quality demo apps, including a fullstack app built with NextJS
Build fullstack React apps with NextJS 14+
Learn all about React Hooks and React Components
new objective addded
new ogjectiv
Course Content
(4) lecture
(2) watched
0 introduction
1 What is JSX?
3 new vedio added
4 this is the demo video

If the course not bought yet then Buy button will display in the page :



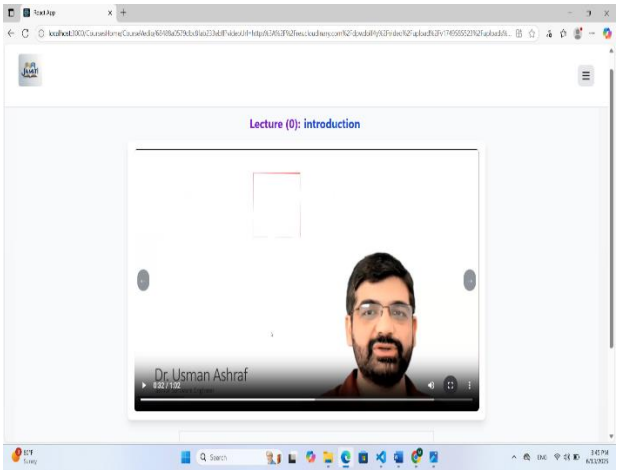
5.1.1.17 Payment Strip Page:

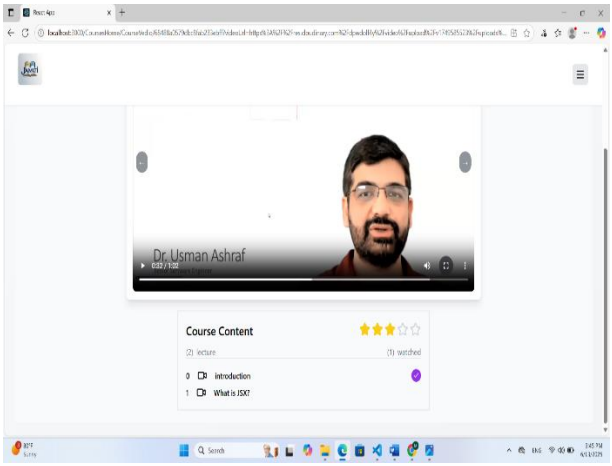
This page shows the payment process and the info that should be entered by the user to confirm the payment in this feature we use an external strip fake method but its store the payment info in a table, to preserve the rights of the instructor and student:

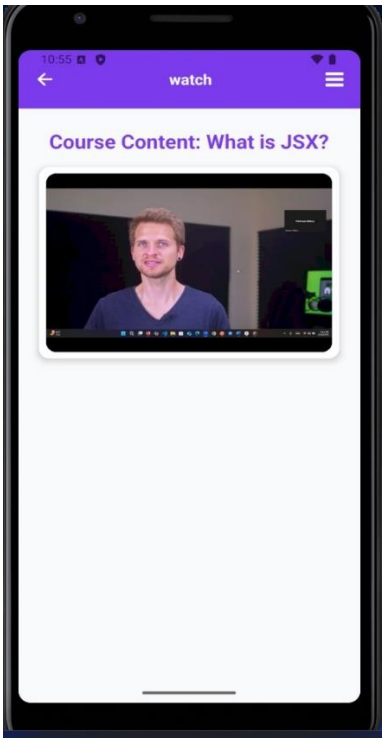


5.1.1.18 Watch Page:

When the student clicks on a video name then it goes to the watch page where the video display and also the video content to it easy to navigate between videos there is tow arrow (next, previous) which going to the next and previous video and also if the video end, then atomically open the next video.

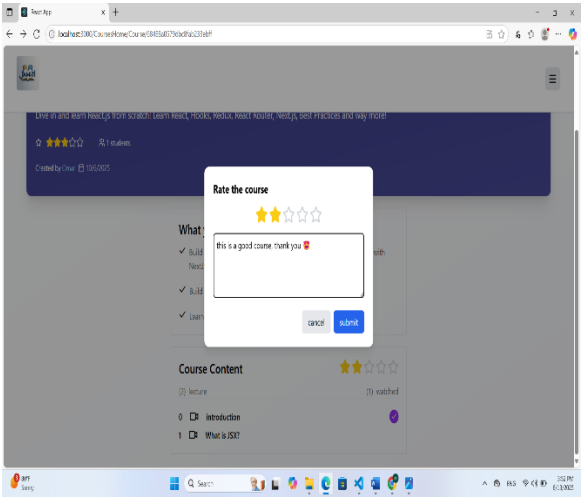


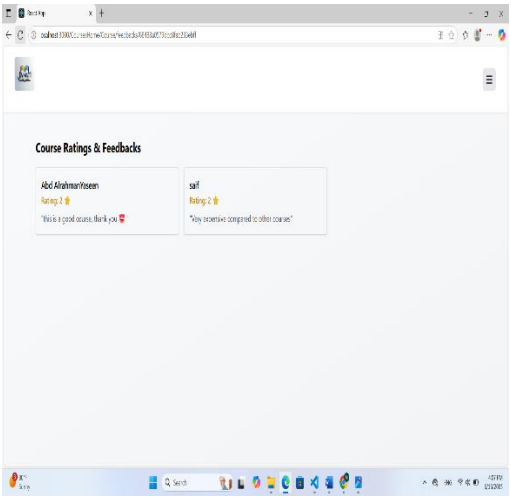


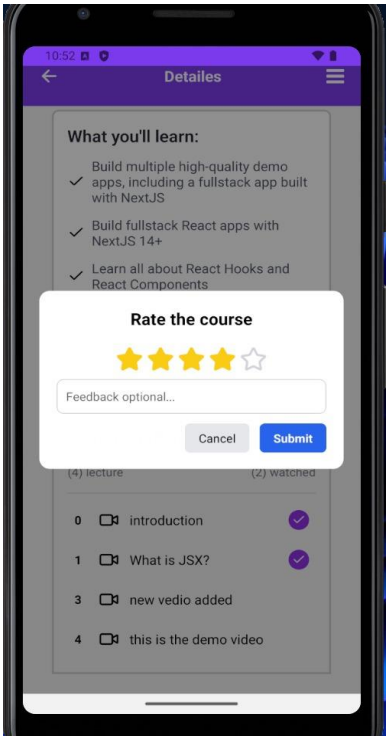


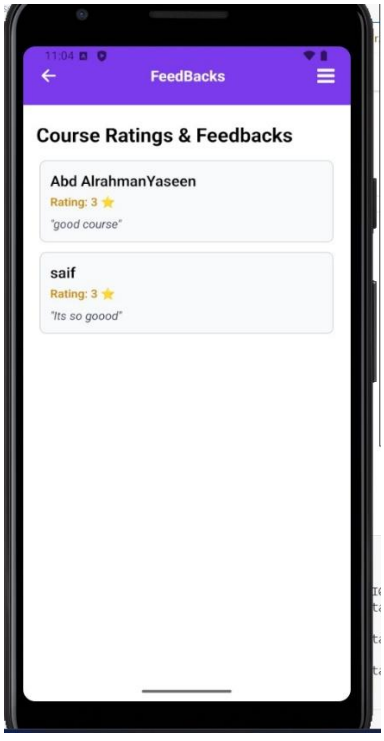
5.1.1.19 Rate course and feedback Page:

The student can rate the course and give optionally feedback if and only if he buys the course, this rate and feedback can be seen by the instructor and the other student also of they not bought the course yet.



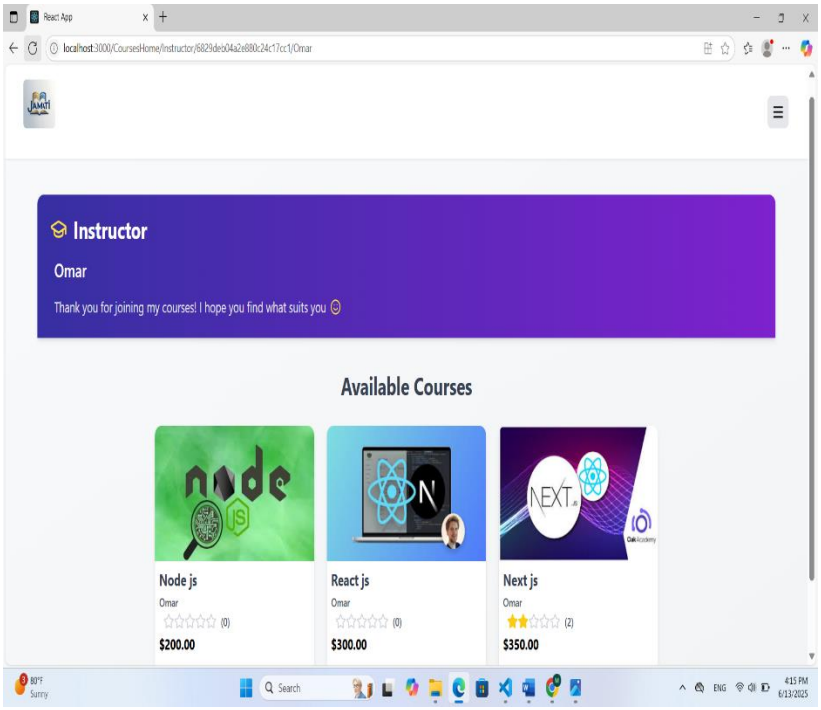


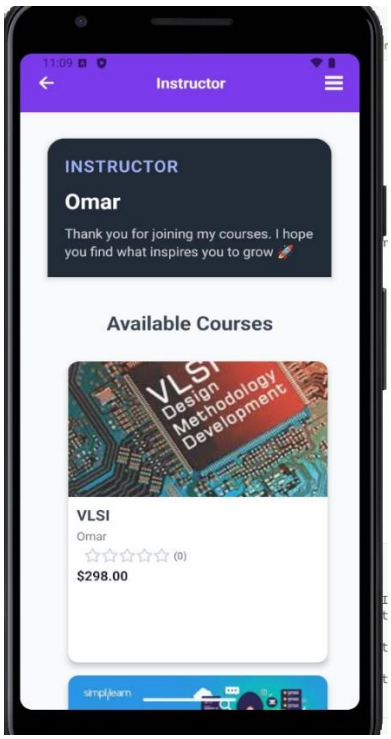




5.1.1.20 Instructor Courses:

When the student clicks on instructor name when he views a course details then navigate to the instructor courses page where all this instructor courses posters shown and instructor information:





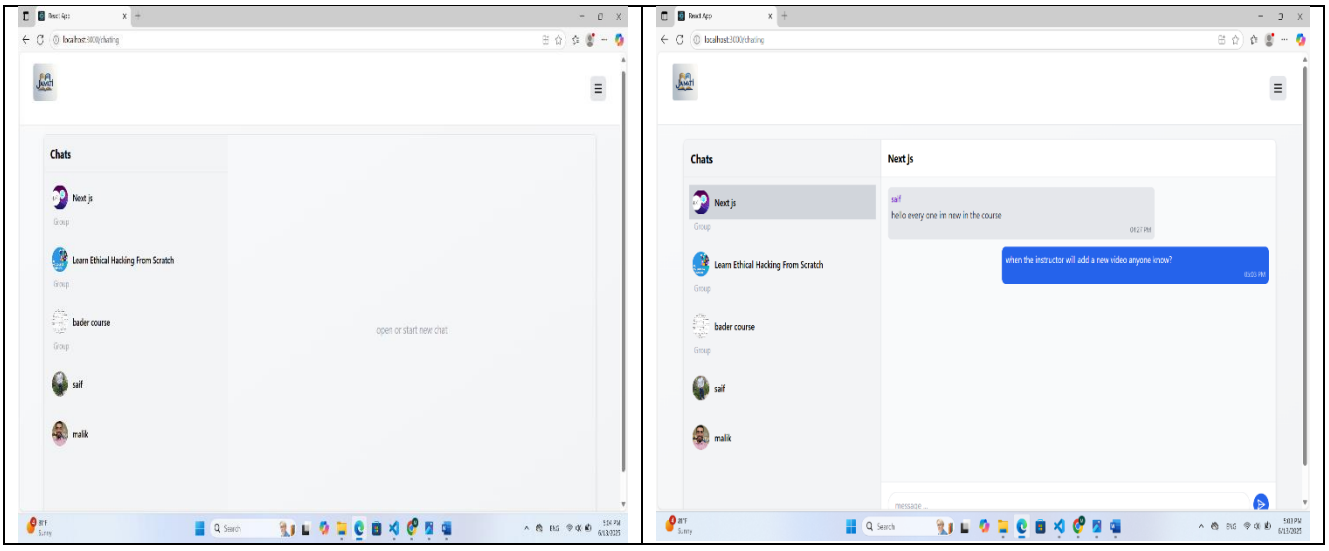
5.1.1.21 Chatting Feature:

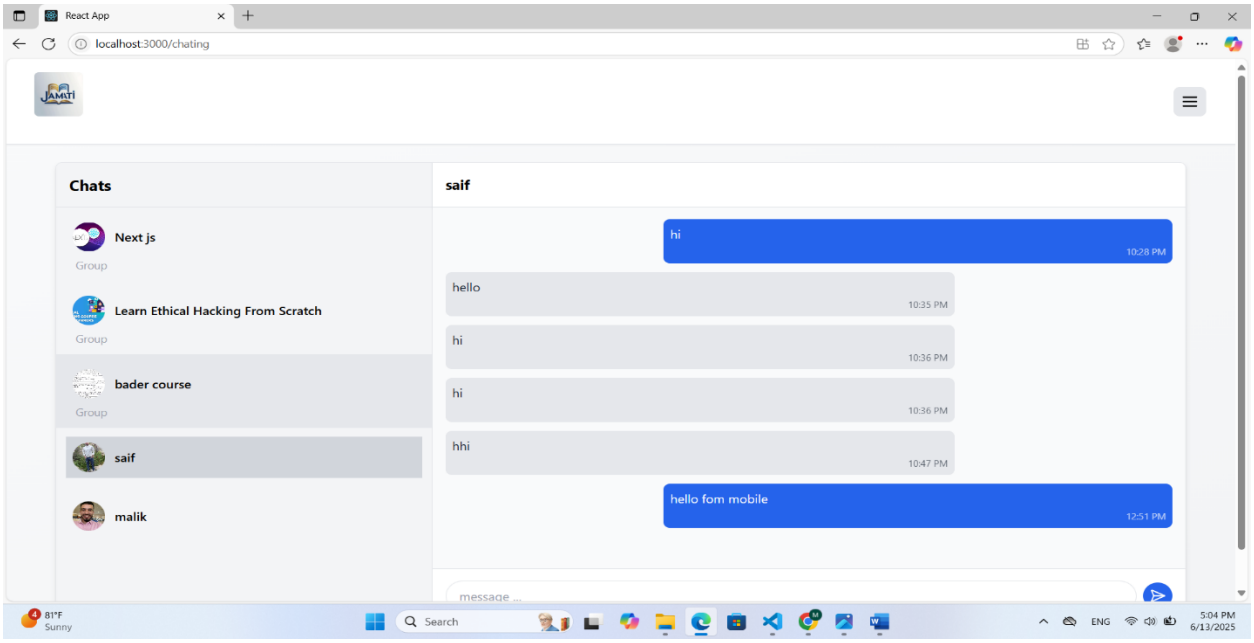
This feature provides a comprehensive real-time communication system within the platform, enabling seamless interaction between users. The chat system supports:

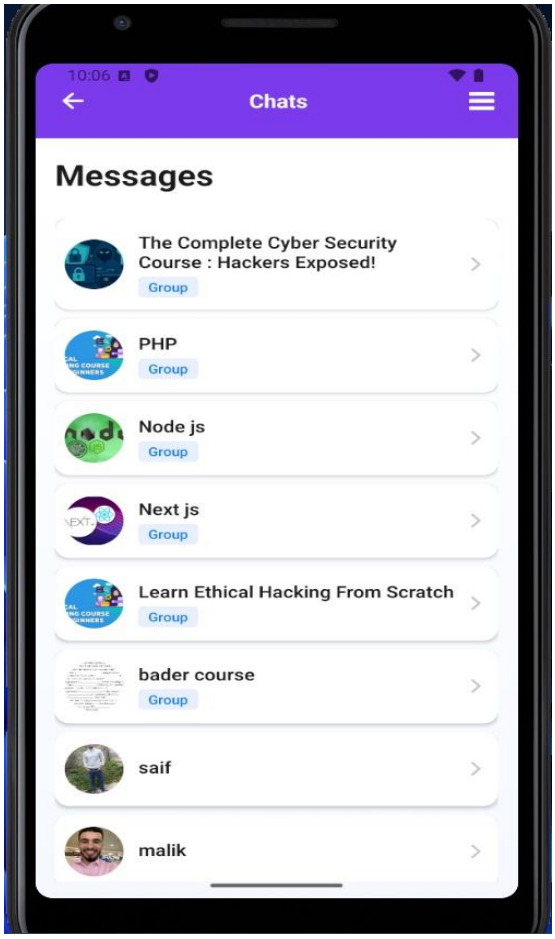
- **One-to-One Chat**: Allows direct messaging between students, as well as between instructors and administrators. This facilitates private and focused conversations.
- **Group Chat**: When an instructor creates a new course, a dedicated group chat is automatically created using the course's name and image. Any student who joins the course is automatically added to the corresponding group. This group chat enables students and the instructor to communicate collectively, ask questions, share ideas, and stay updated.

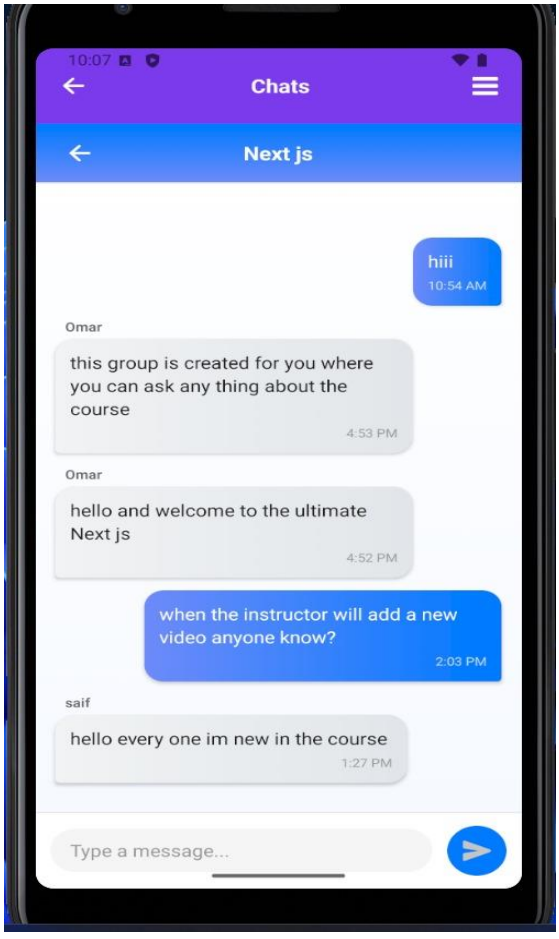
When a user navigates to the chat page, they are presented with a list of all chats they are part of—whether personal or group chats.

To ensure real-time communication, the chat system is built using **WebSocket technology**, which allows instant message delivery without the need for page refreshes or manual updates




React App
localhost:3000/chating
Chats
saif
Next js
Group
Learn Ethical Hacking From Scratch
Group
bader course
Group
saif
malik
hi
10:28 PM
hello
10:35 PM
hi
10:36 PM
hi
10:36 PM
hhi
10:47 PM
hello fom mobile
12:51 PM
message ...
Search
ENG
5:04 PM
6/13/2025


10:06
Chats
Messages
The Complete Cyber Security Course : Hackers Exposed!
Group
PHP
Group
Node js
Group
Next js
Group
Learn Ethical Hacking From Scratch
Group
bader course
Group
saif
malik

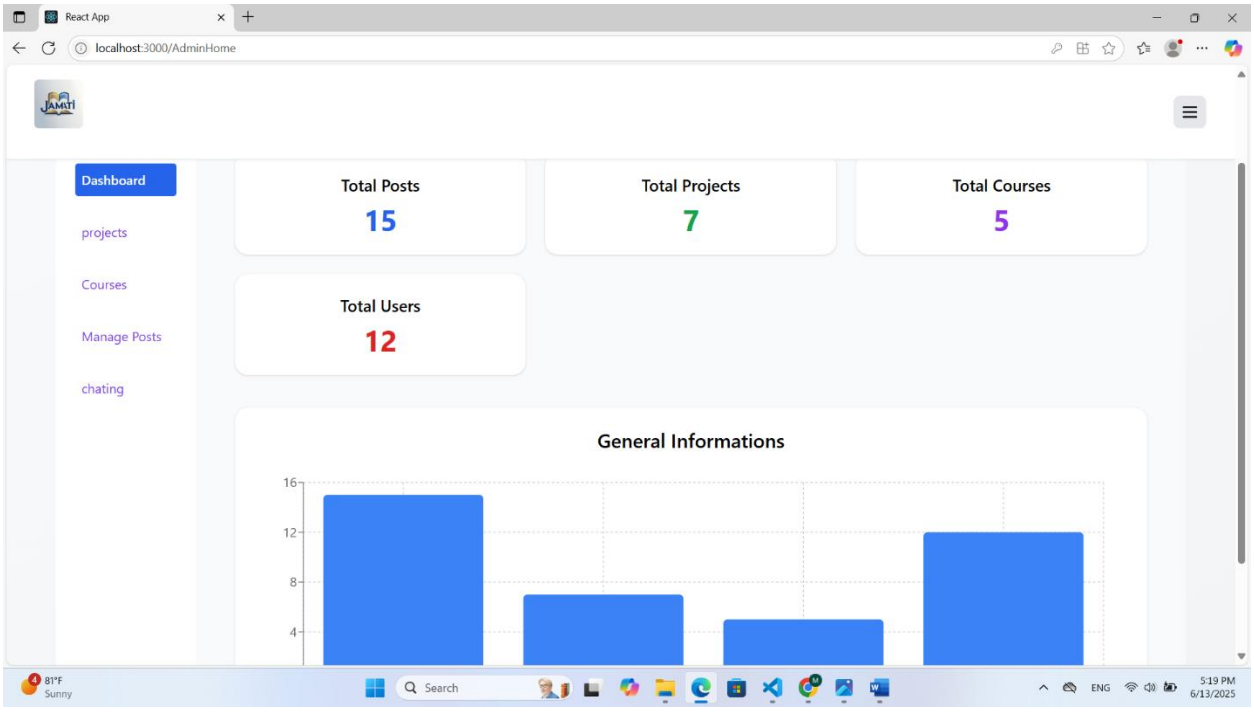

10:07
Chats
Next js
hiii
10:54 AM
Omar
this group is created for you where you can ask any thing about the course
4:53 PM
Omar
hello and welcome to the ultimate Next js
4:52 PM
when the instructor will add a new video anyone know?
2:03 PM
saif
hello every one im new in the course
1:27 PM
Type a message...

6.1 Admin Dashboard:

The **Admin Dashboard** is accessible through the web and mobile, as the control center for *Jamati*. It enables administrators to:

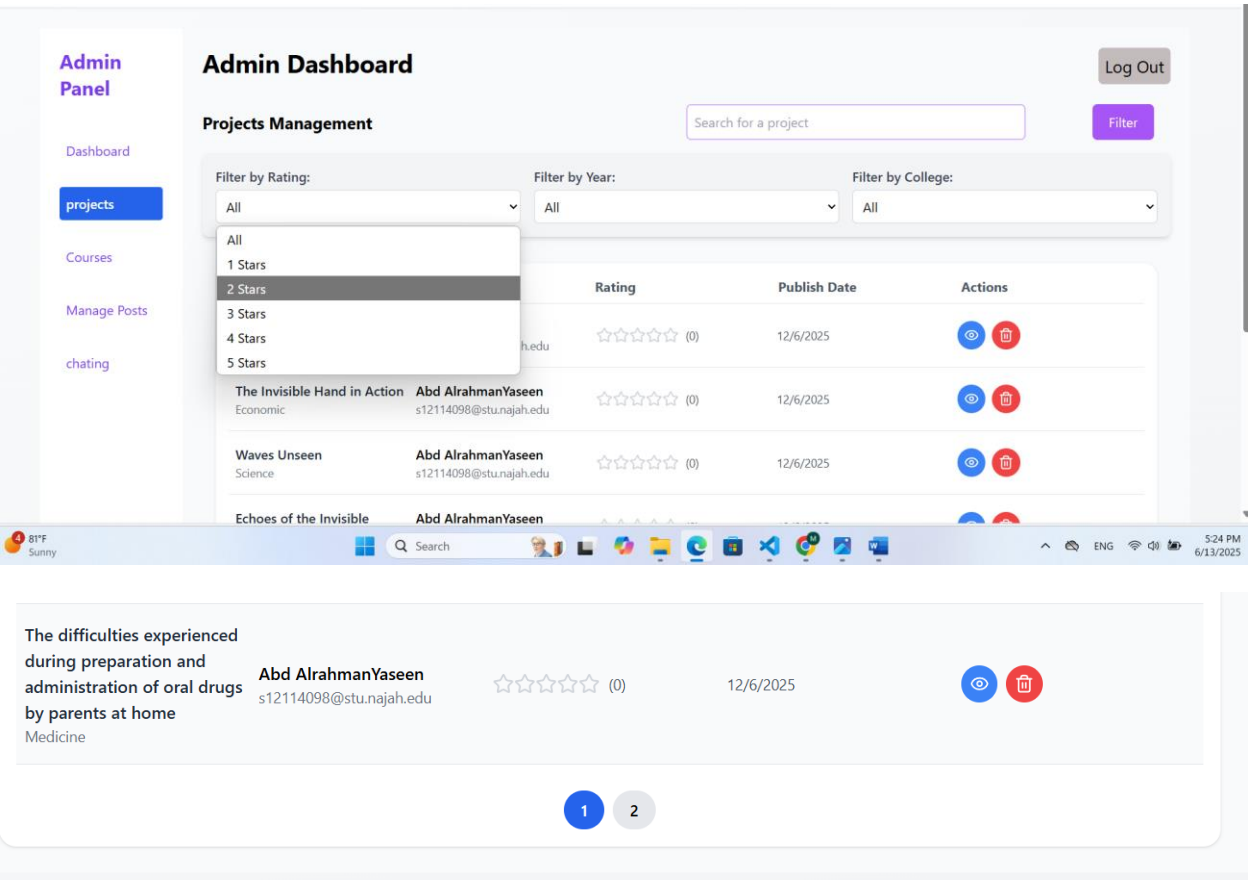
- Monitor core stats: total users, total posts, courses and projects.
- Manage the projects and he has the ability of deleting it
- Monitor post rated and selling five courses
- Manages instructors: delete in instructor viewing documents that the instructor uploads and reject or approve it
- **Communicate directly** with instructors
- **Manage and deleted reported posts or remove it from the list of repots**

This dashboard provides essential tools for maintaining platform integrity and supporting users efficiently.



6.1.2: Manage Projects:

In the page the admin can view the projects and filter then by rating, Year and College. Also, he can search on a project title, each page shows up to 5 project there is a pagination to make the fetch of the data faster and user friendly.

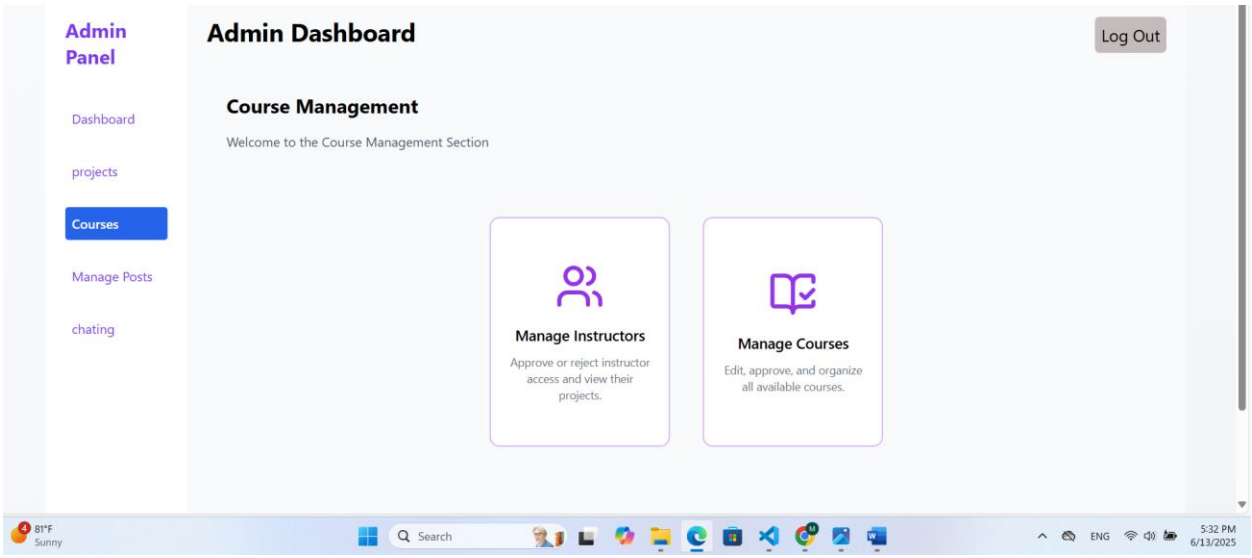


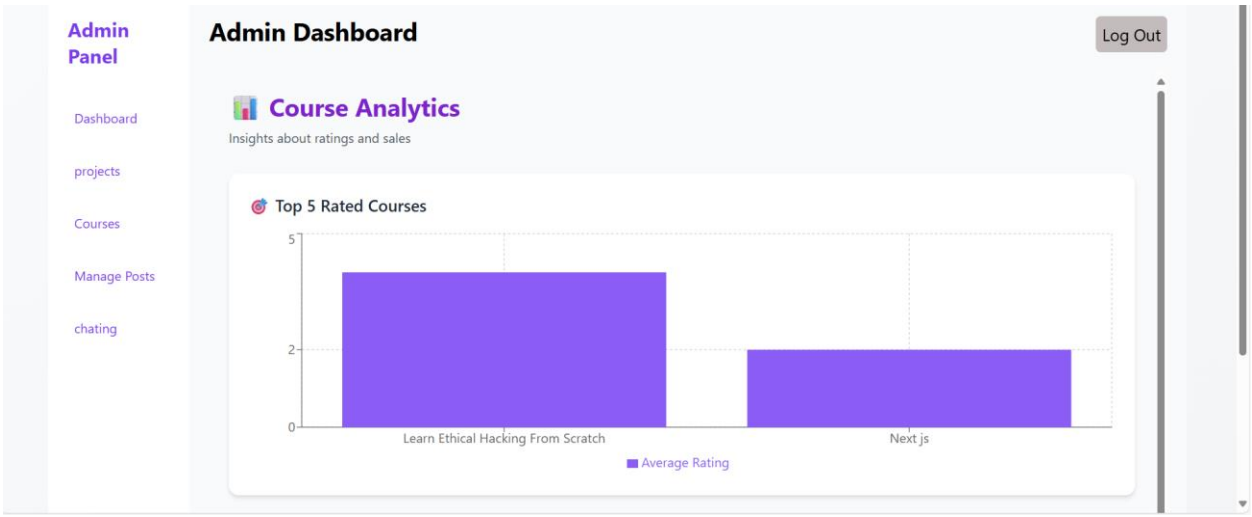
6.1.3Manage Courses:

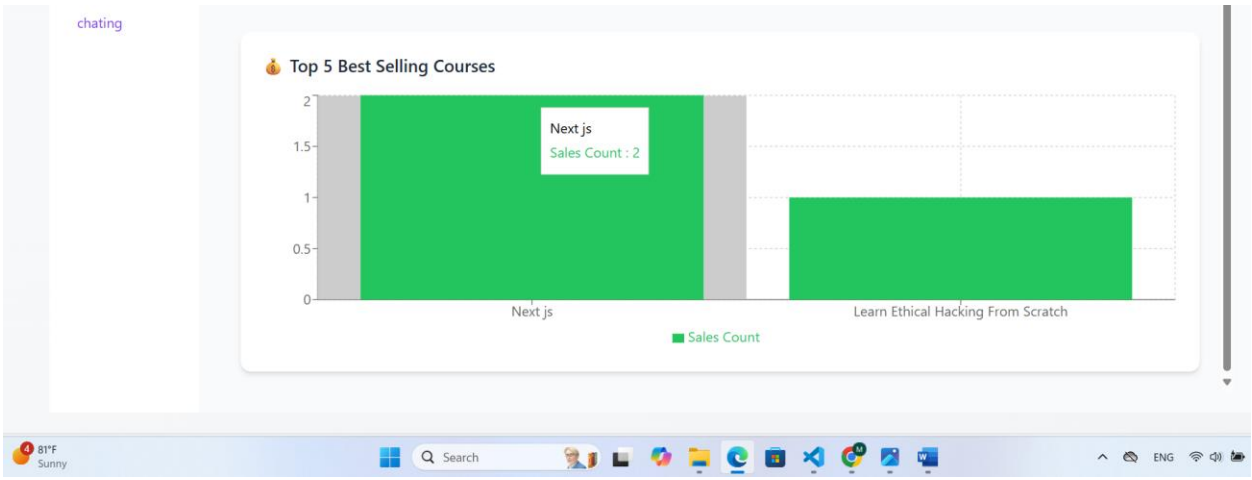
This page enables the admin tow manage the instructor or monitor the **top rated** or **Bestselling courses**.

In managing instructor the admin can view the document that the instructor uploads when he sign up to the system and he can't log in until the admin approve on that.

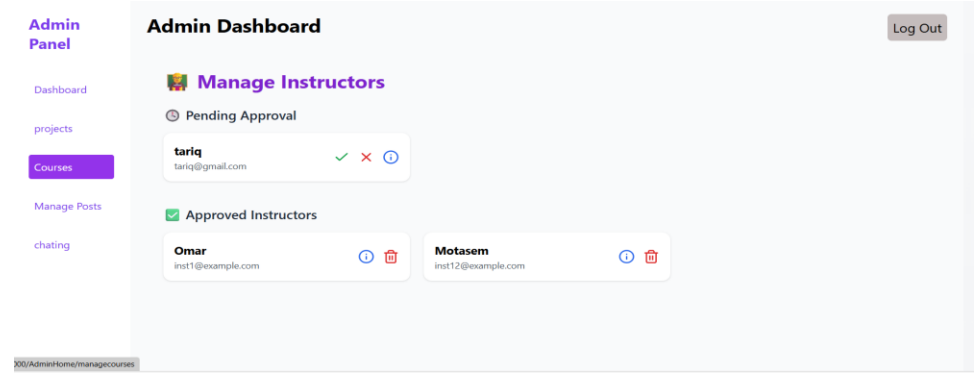
.

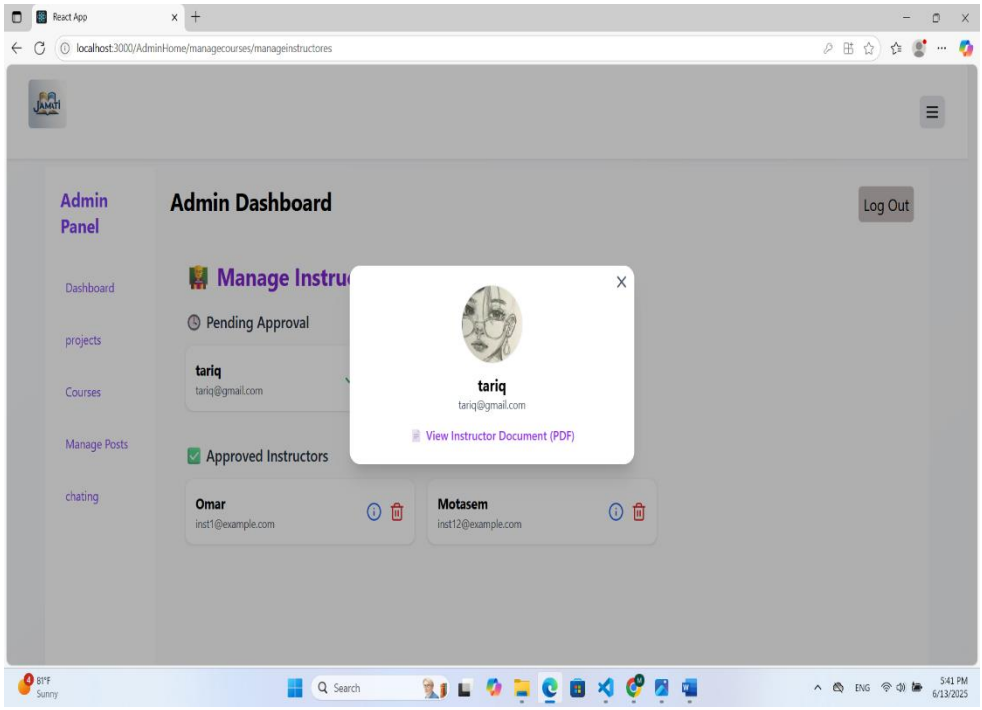

Admin Panel
Admin Dashboard
Log Out
Dashboard
projects
Courses
Manage Posts
chating
Course Management
Welcome to the Course Management Section
Manage Instructors
Approve or reject instructor access and view their projects.
Manage Courses
Edit, approve, and organize all available courses.
81°F
Sunny
Search
ENG
5:32 PM
6/13/2025


Admin Panel
Admin Dashboard
Log Out
Dashboard
projects
Courses
Manage Posts
chating
Course Analytics
Insights about ratings and sales
Top 5 Rated Courses
5
2
0
Learn Ethical Hacking From Scratch
Next js
Average Rating


chating
Top 5 Best Selling Courses
2
1.5
1
0.5
0
Next js
Sales Count : 2
Next js
Learn Ethical Hacking From Scratch
Sales Count
81°F
Sunny
Search
ENG

Manage instructor’s page:

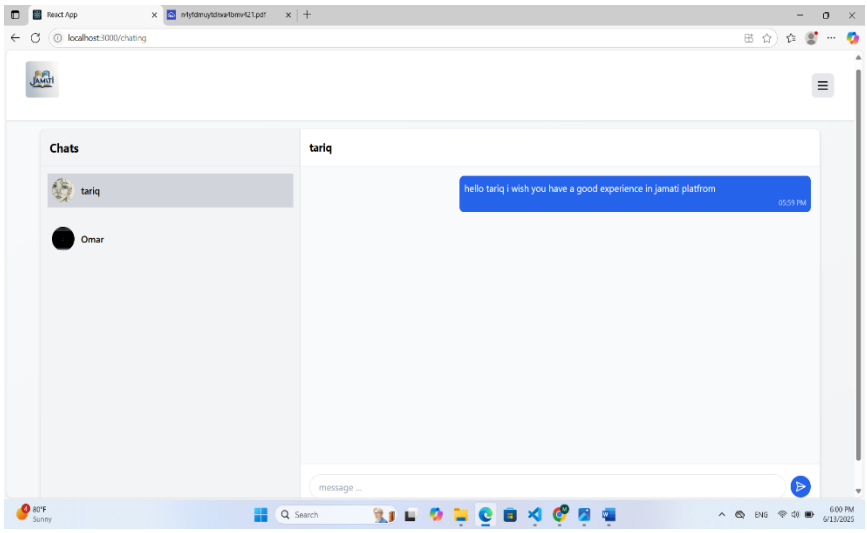




When click on the document its open in the browser.

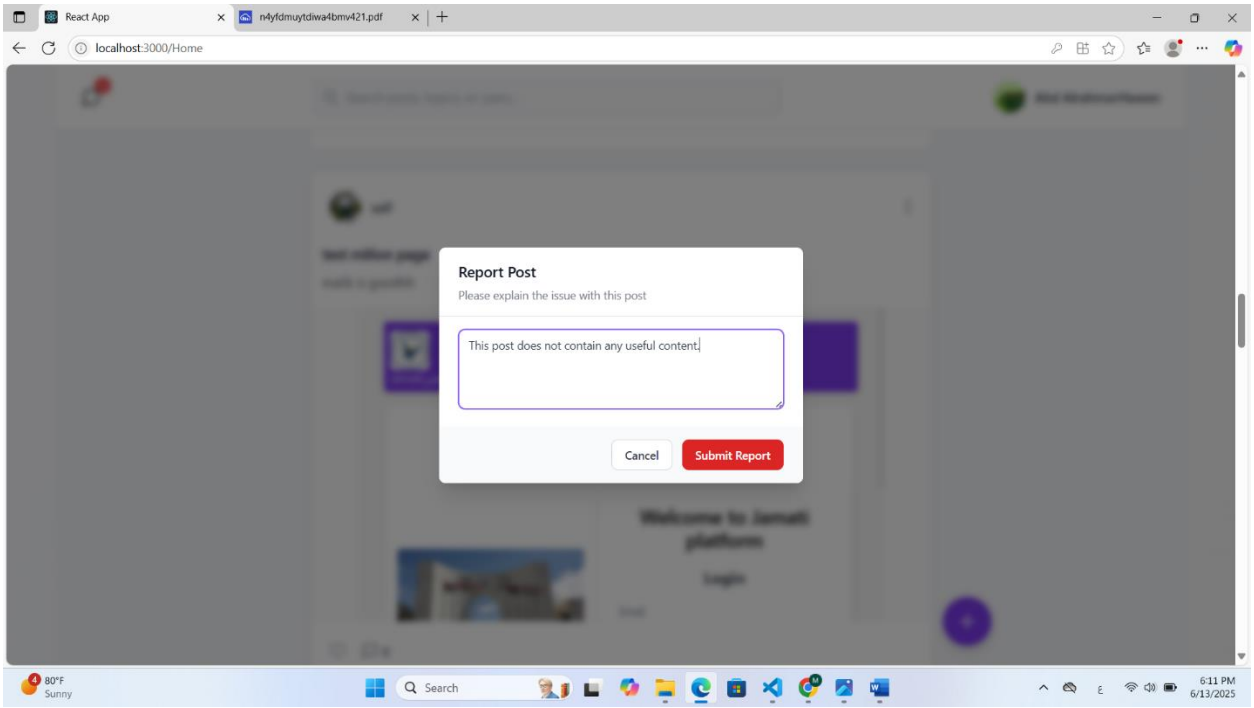
6.1.4 Admin chatting:

The admin can message the instructors only:

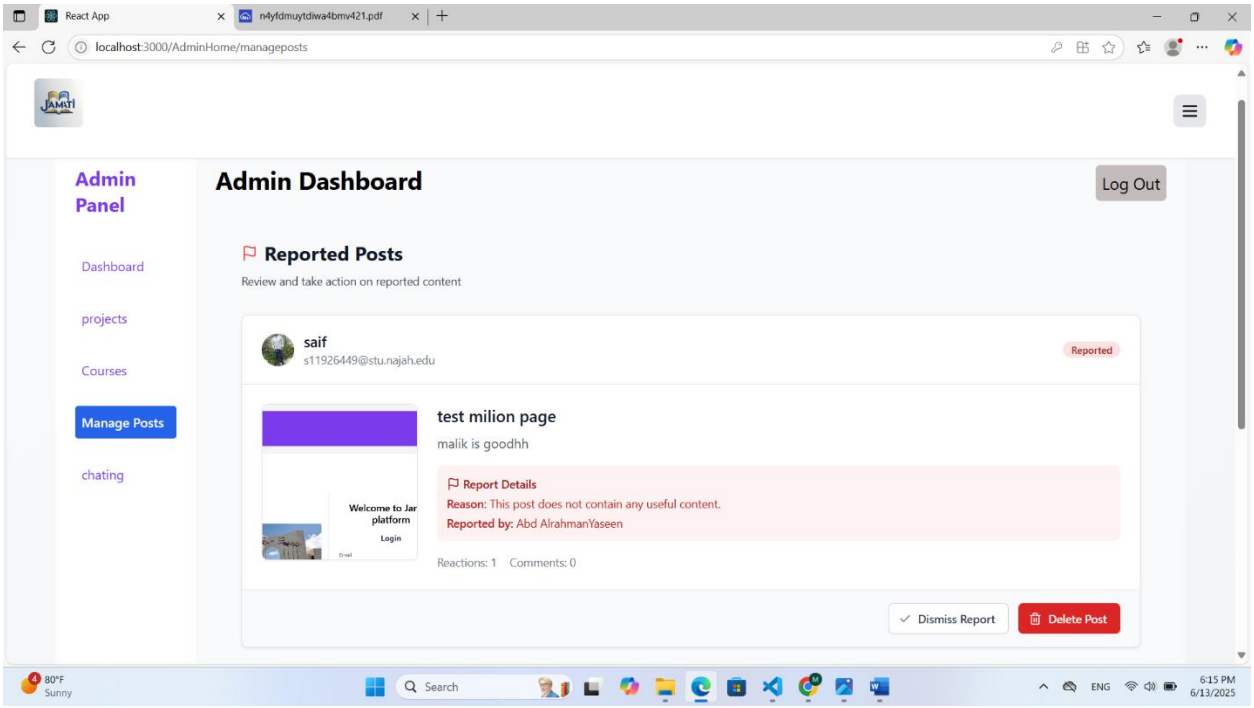


6.1.5 manage reported posts:

The student can report a post and also write the reason for report:



The admin views the reported posts and he can delete the post or remove it from the list of reported pots:



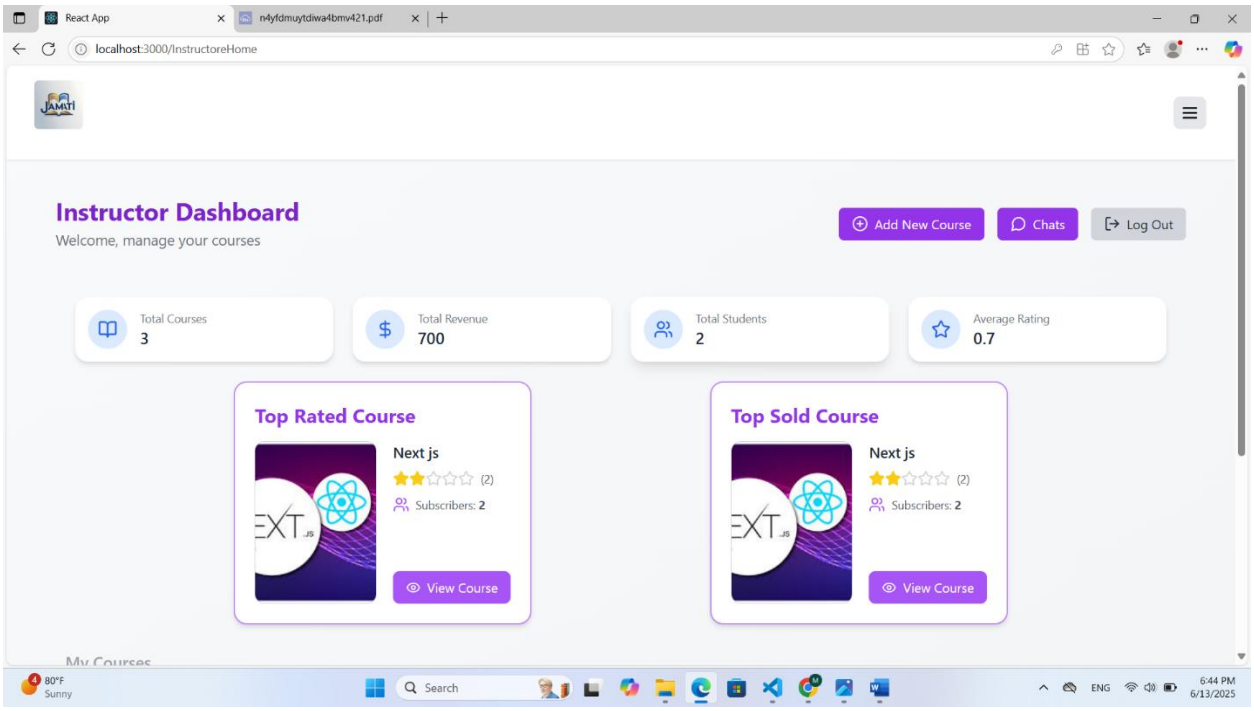
7.1 Instructor Views:

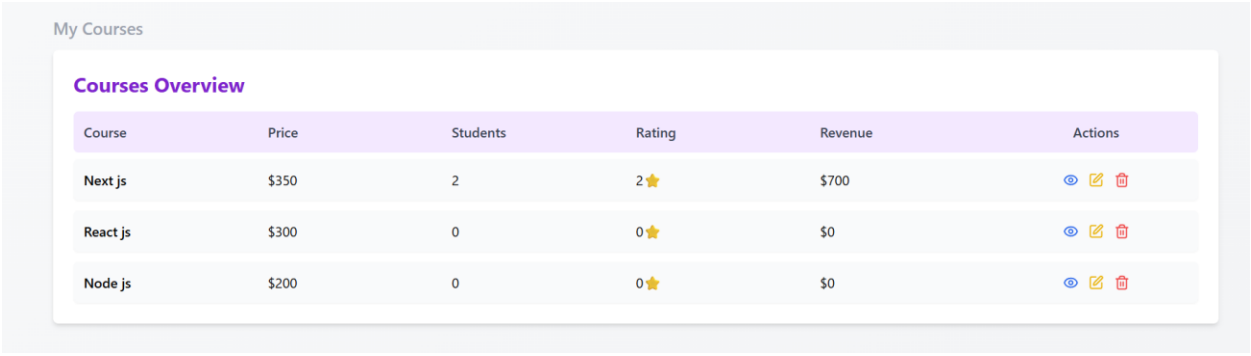
track performance, and interact with users efficiently. It includes the following key features:

- Navigation Bar: Positioned at the top of the dashboard, the navbar provides quick access to essential actions:
    - Add Course: Allows the instructor to create and publish a new course.
    - Chatting: Redirects to the chat system where the instructor can engage with students and admins.
    - Logout: Enables the instructor to securely exit the system.
- Instructor Statistics: This section displays insightful metrics related to the instructor's activity and performance, including:
    - Total Number of Courses the instructor has created.

    - Average Rating across all the instructor's courses.
    - Number of Students who have purchased any of the instructor's courses.
    - Total Revenue generated from course sales.
- Top Performing Courses:
    - Top Rated Course: The course with the highest average rating.
    - Top Sold Course: The course with the highest number of purchases.
- Course Management Table: A list of all the instructor's courses is presented, each with options to:
    - Edit course details.
    - View the course as a student.
    - Delete the course.

It is important to note that when an instructor deletes a course, the course is not removed entirely from the system. This is to ensure that students who previously purchased the course still have access to it, thus preserving a positive user experience. Instead, deleted courses are hidden from new students and no longer displayed in public listings.



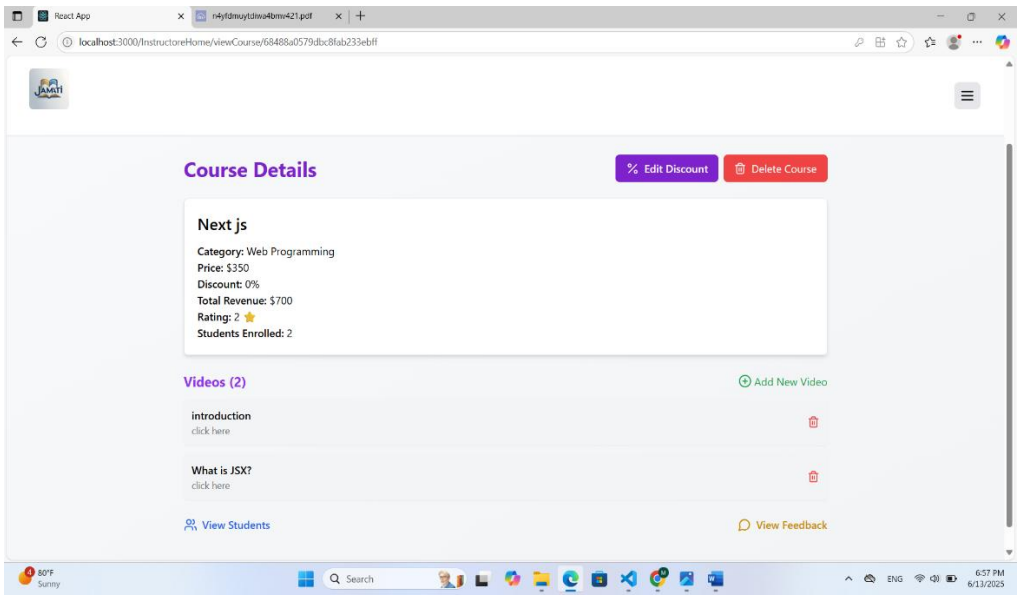


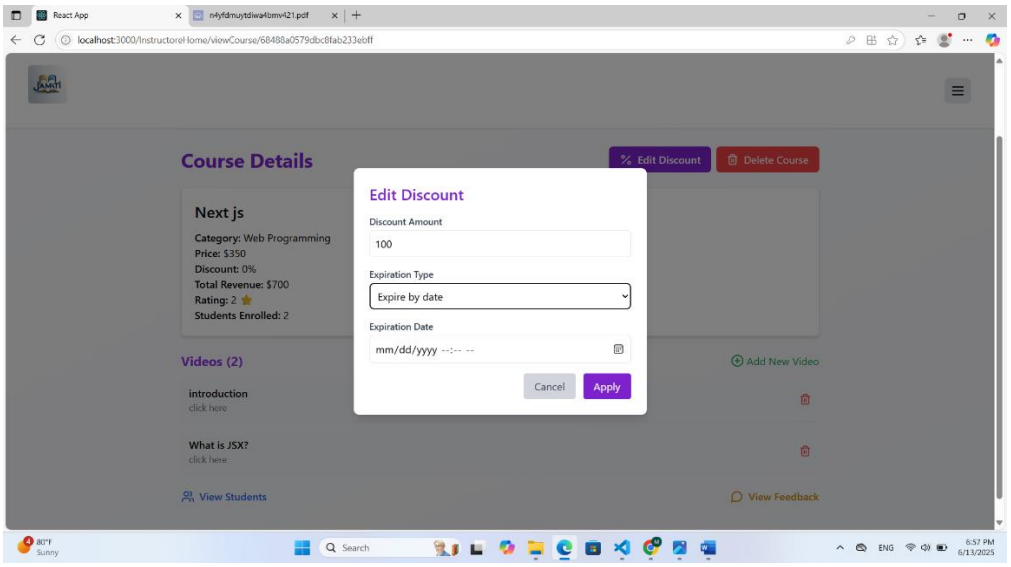
7.1.1 Instructor View Course Content:

In this page the instructor can view its course content and videos the instructor can do the following:

- Make a discount on the course for a specific interval of time or for an count of hours.
- Delete the course (from this data) which means the old students still able to see the course content

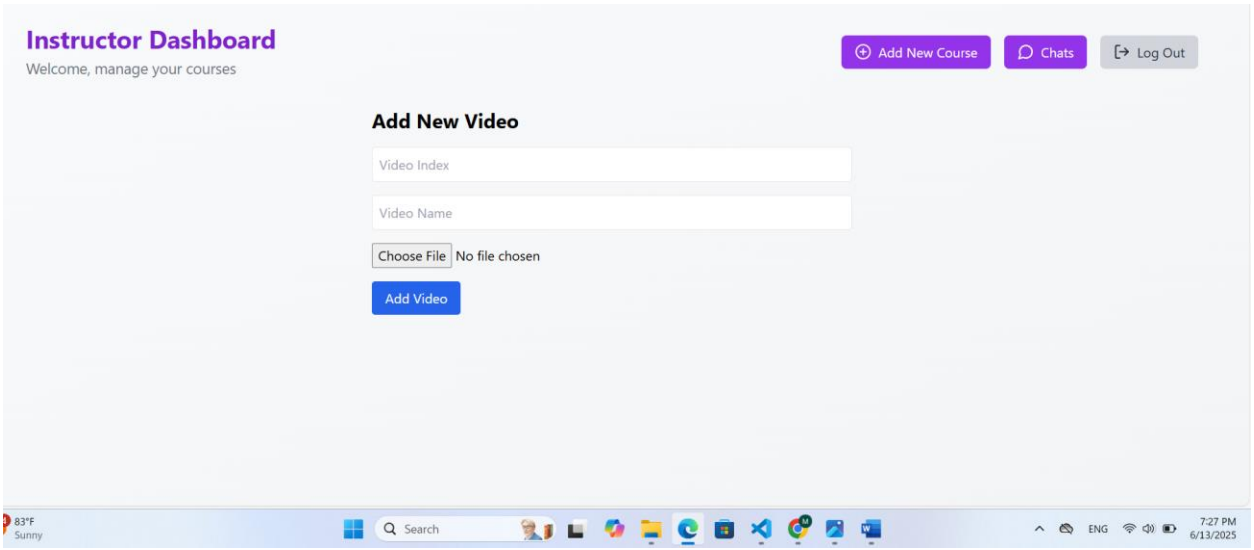
- Add new video to the course
- Delete video from the course
- View student how bought this course
- View feedbacks and he can delete it



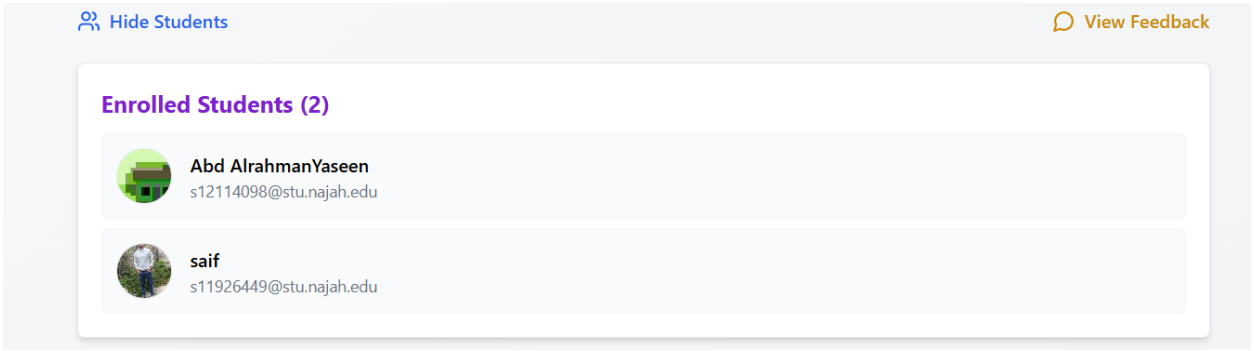


Add new video:

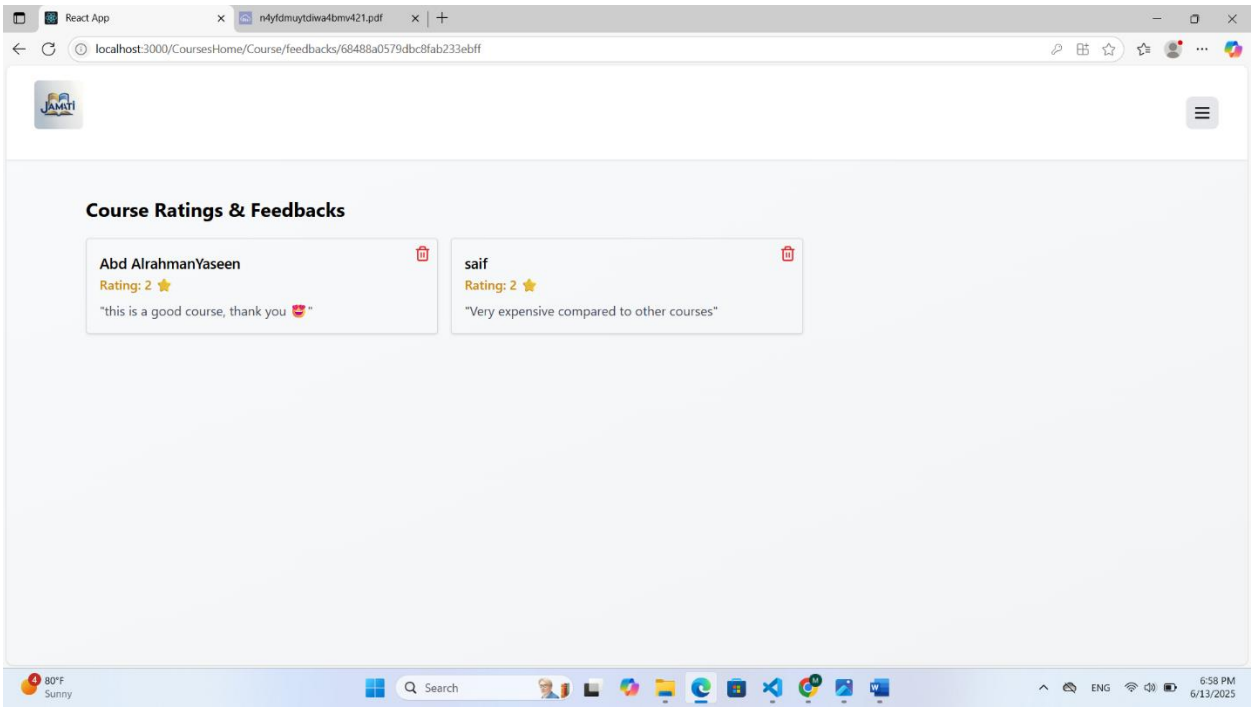
The instructor can add a new video and its index to manage where to set it in the list:



View students:

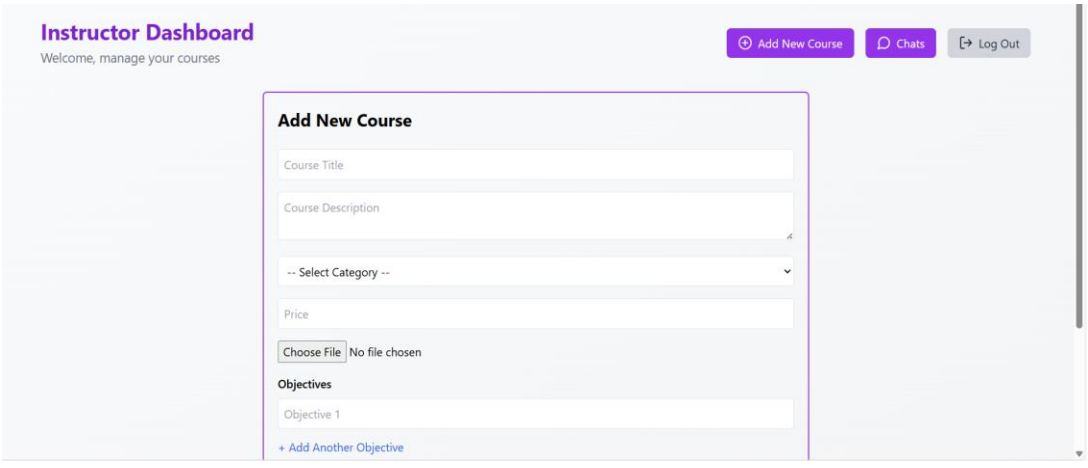


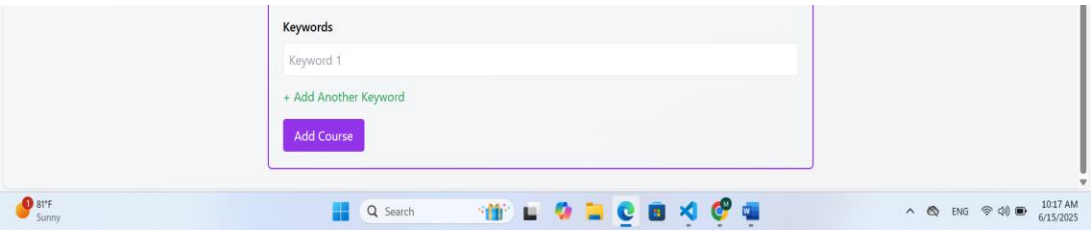
View feedbacks:



7.1.2 Add new course:

The instructor can add new course and he can add array of keywords to the course where is array used to help the students to show related courses to what they buy,after that new chatting group automatically will be created with instructor as first member in the group

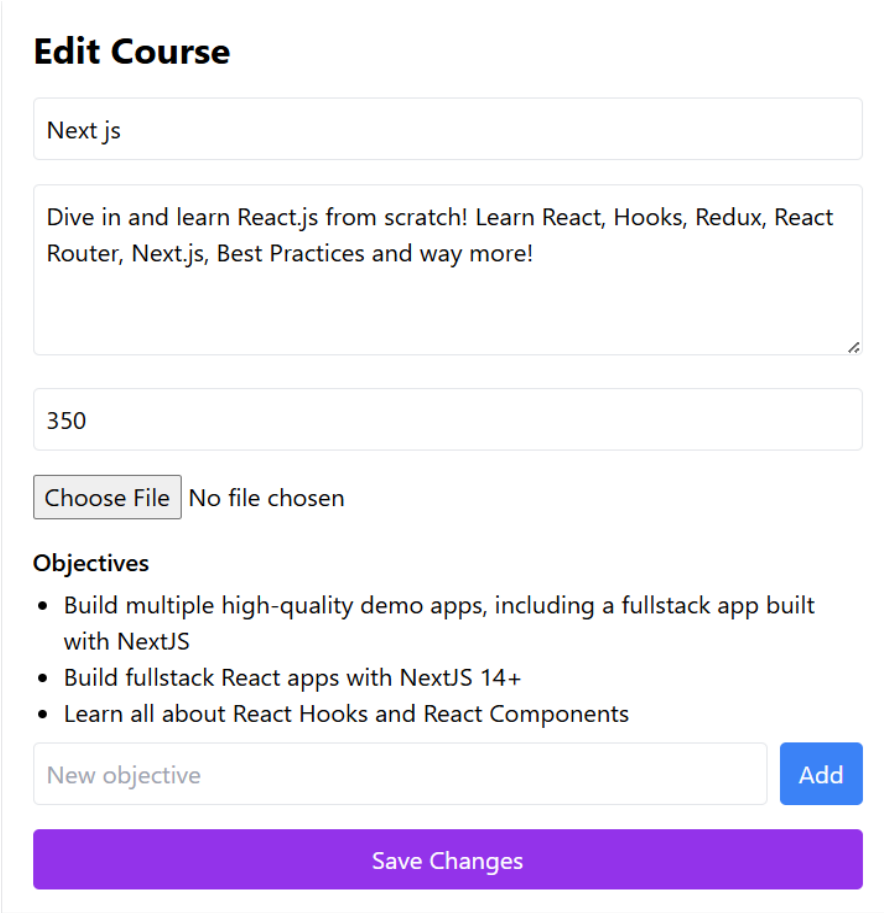




7.1.3 edit course:

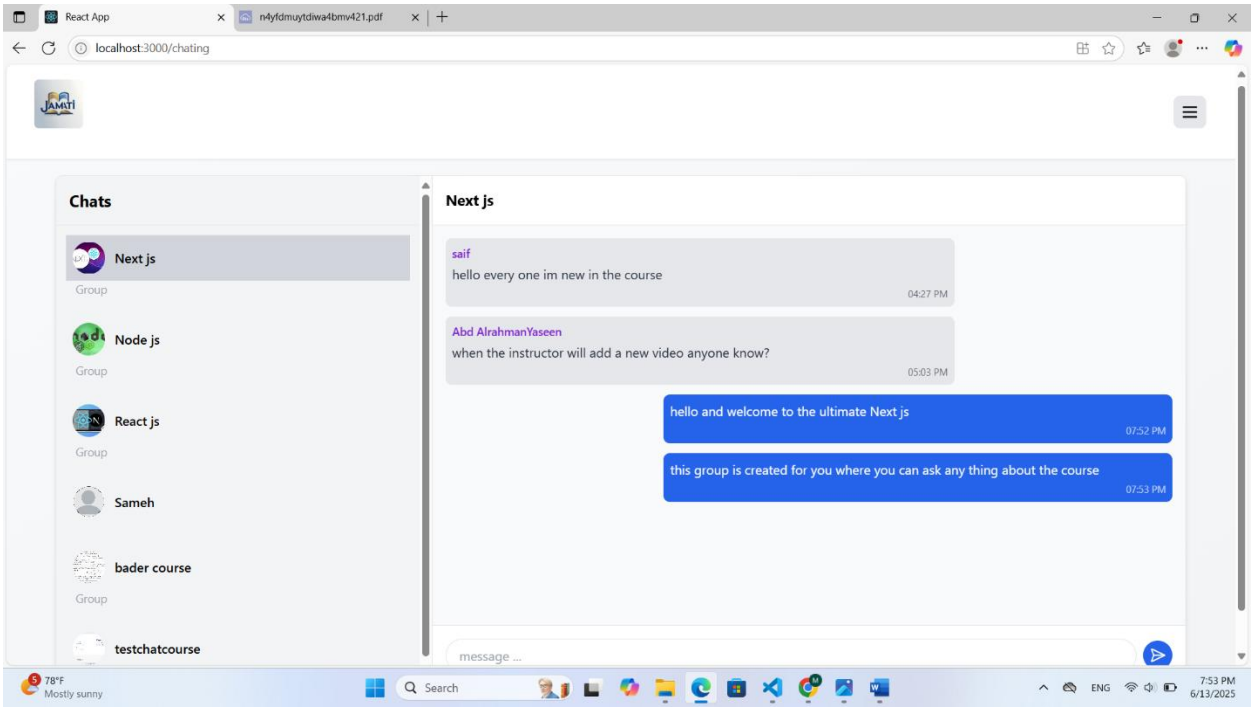
The instructor can edit the course information

- Name
- Description
- Base price
- Poster Image
- Add new objectives



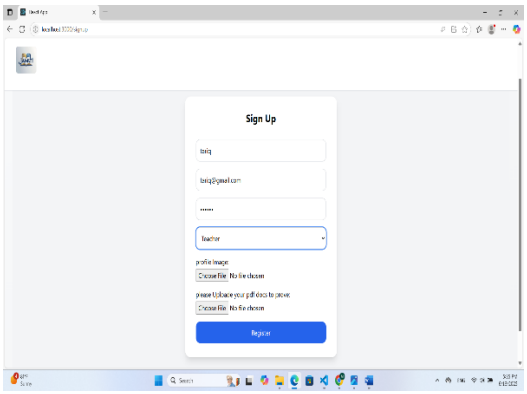
7.1.4 chatting:

as we mention the instructor can only communicate with the admin (**one to one messages**) and with students via the groups of its courses.



7.1.5 instructor Registration:

When the instructor signs up, he must upload a document to verify that he is a real instructor and to make then System admin accessibility to see this document then the admin decides to approve or reject him.

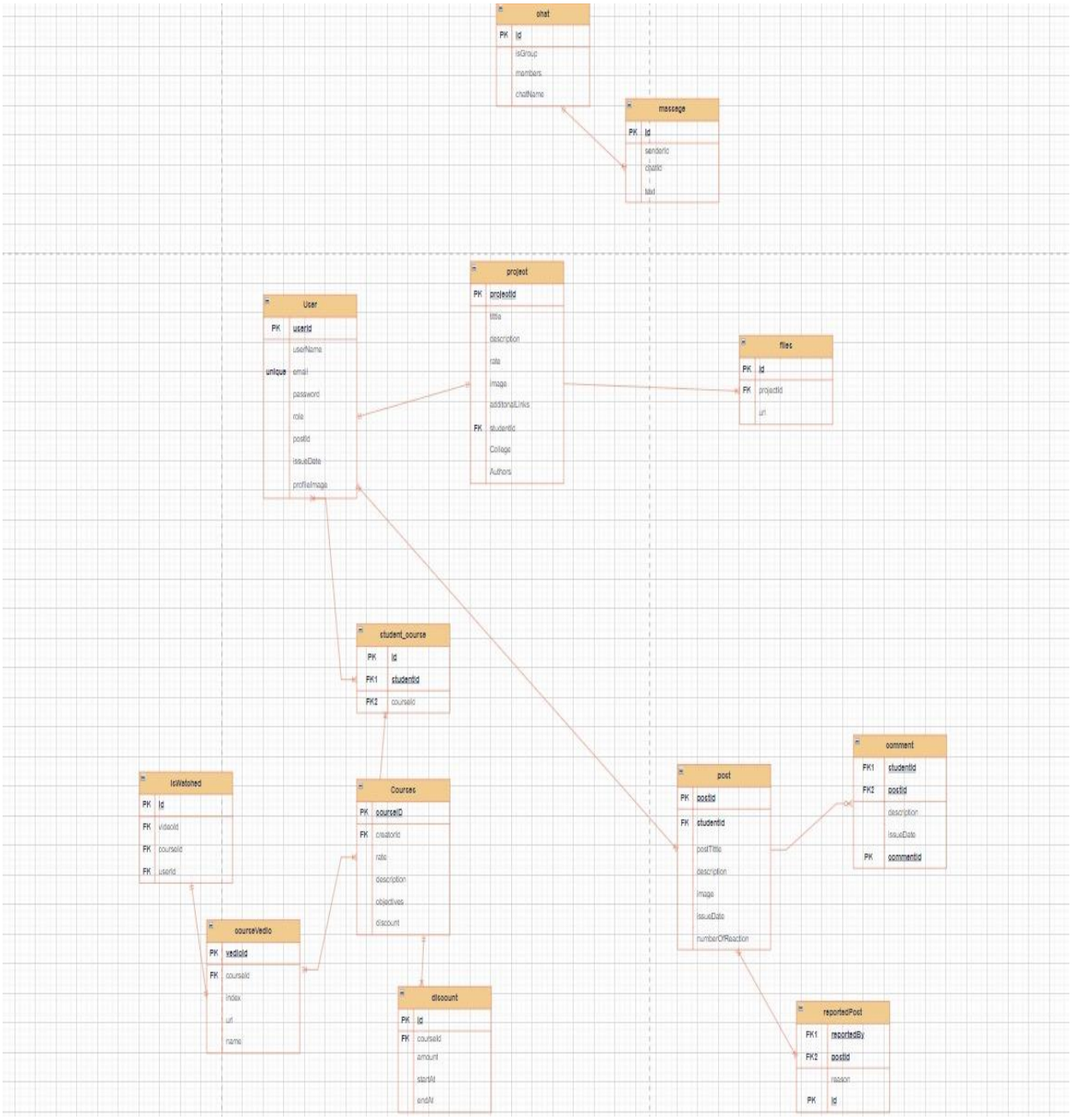


8 Database:

In the ***Jamati*** project, we use **MongoDB** as the primary database to store and manage all application data. MongoDB is a **NoSQL document-oriented database** designed for flexibility, scalability, and high performance, making it an excellent fit for modern web applications.

Why MongoDB?

- **Schema Flexibility**: MongoDB stores data in JSON-like documents (BSON), allowing dynamic and flexible schemas. This is ideal for a system like *Jamati*, where different entities (users, posts, courses, etc.) may have diverse structures and evolve over time.
- **High Scalability**: MongoDB supports horizontal scaling and can handle large volumes of data and concurrent users efficiently.
- **Real-time Performance**: With features like indexing and fast queries, MongoDB ensures responsive performance, crucial for real-time features like chatting and content interaction.
- **Native Support for Relationships**: While being a NoSQL database, MongoDB supports references and embedded documents, which allows modeling both relational and hierarchical data efficiently.


chat
PK Id
isGroup
members
chatName
massage
PK Id
senderId
chatId
text
projeot
PK projectId
title
description
role
image
additionalLinks
FK studentId
College
Authors
User
PK userId
userName
unique email
password
role
postId
issueDate
profileImage
files
PK Id
FK projectId
url
student_sourse
PK Id
FK1 studentId
FK2 courseId
comment
FK1 studentId
FK2 postId
description
issueDate
PK commentId
IciWatched
PK Id
FK videoId
FK courseId
FK userId
Courses
PK courseID
FK creatorId
role
description
objectives
discount
post
PK postId
FK studentId
postTitle
description
image
issueDate
numberOfReaction
courseVedio
PK vedioId
FK courseId
index
url
name
discount
PK Id
FK courseId
amount
startAt
endAt
reportedPost
FK1 reportedBy
FK2 postId
reason
PK Id

9 Discussion

The development of *Jamati* aimed to provide an all-in-one academic platform that supports project sharing, course delivery, and student communication in a structured and scalable way. This section discusses how the system met its objectives, the effectiveness of the implemented features, challenges faced during development, and the rationale behind key design and technology choices.

9.1 Achievement of Objectives:

Through the integration of features such as user roles, project repositories, course management, and real-time chat, *Jamati* successfully delivers a unified academic experience. The system enables students to share and explore academic content, communicate with peers and instructors, and enroll in structured video-based courses—all within a mobile and web environment. Each user role (Student, Instructor, Admin) experiences a tailored interface, ensuring accessibility and clarity

9.2 Feature Effectiveness:

One of the most significant features is the real-time chat system, which supports both private and group communication. The decision to use WebSocket technology enabled smooth and instantaneous message delivery. Group chats linked to courses are automatically created when an instructor publishes a new course, and students are auto-added upon enrollment—streamlining collaboration and academic interaction.

Another major feature is the project repository, where students can upload, search, and rate academic projects. The inclusion of filters (college, author, date, etc.) made it easier to navigate and find relevant work, improving academic resource accessibility. The course module complements this by allowing instructors to manage content via a video-based curriculum, with real-time feedback and ratings

from students. Furthermore, integrating Stripe for secure and verifiable payments ensured a trustworthy and transparent purchase process for paid courses

9.3Project Impact:

Jamati stands out by offering a holistic solution tailored to the academic culture of An-Najah National University. Unlike global platforms that separate project sharing from course delivery, Jamati brings these together in one localized, secure, and user-centered environment. The ability for students to not only learn but also **contribute**, **collaborate**, and **communicate** in real-time creates a richer and more engaging academic experience

10 Conclusions and Recommendations

In this section, we present a summary of the project outcomes along with suggestions for future enhancements.

10.1 Summary

In summary, the *Jamati* platform successfully delivers an integrated academic environment tailored to the needs of An-Najah National University students. By combining project sharing, structured course delivery, real-time communication, and secure payments, the system addresses multiple educational challenges in a unified solution. The thoughtful use of modern technologies such as MongoDB, WebSocket, React, and Node.js ensured scalability, responsiveness, and real-time interactivity across web and mobile. The implementation of role-based access, email verification, and payment protection further enhanced user trust and security. Through overcoming technical and user-experience challenges, *Jamati* demonstrates its value as a scalable and future-ready academic platform

10.2 Future Works

To further improve the platform and expand its capabilities, the following future enhancements are proposed:

- **AI-Based Course Recommendation System**: Implement a recommendation engine using machine learning to suggest courses to students based on their interests, activity history, academic background, and previously rated courses.

- **Modern filtering for projects**: enable student to filter and navigate in more user-friendly methods.
- **Upload files and voice messages:** Add a feature that allows users to upload files and images in the chat and also enable recorded voice messages.
- **Subcategories courses**: add subcategories to the courses to make navigation and filtering more friendly and easy.

References

1-Node.js: Node.js Foundation. (n.d.). *Node.js Documentation*. Retrieved from: https://nodejs.org/docs/latest/api/

2-Express.js: Express.js. (n.d.). *Express – Node.js web application framework*. Retrieved from: https://expressjs.com

2-MongoDB: MongoDB Inc. (n.d.). *MongoDB Manual – Official Documentation*. Retrieved from: https://www.mongodb.com/docs/

3-ReactJS: Team. (2024). React – A JavaScript library for building user interfaces. Retrieved from https://reactjs.org

4-Meta Platforms: Inc. (2024). React Native – Build native apps using React. Retrieved from https://reactnative.dev

5-Google LLC. (2024). Android Studio – The Official IDE for Android Development. Retrieved from https://developer.android.com/studio

6-Cloudinary. Cloudinary. (n.d.). *Cloudinary Documentation*. Retrieved from: https://cloudinary.com/documentation

7-Stripe. Stripe, Inc. (n.d.). *Stripe Documentation – Payment Integration Guide*. Retrieved from: https://stripe.com/docs

8-chatgbt: https://chatgpt.com/