CSE471: System Analysis and Design
Project Report
Project Title: Thesis Portal

Group No: 01, CSE471 Lab Section: 05, Fall 2024	
ID	**Name**
21201532	Tasmin Ahmed Oni
22101881	Afsana Akter Mim
21201757	Riba Rahman
21201241	Nirvik Shaha Rudra

Submission Date:02/09/2025

Table of Contents

1. System Request

Business need

The Thesis Portal is designed to streamline and digitize the entire thesis management process within academic institutions. This platform provides a centralized, interactive, and secure platform where students, supervisors, and administrators can collaborate seamlessly. It ensures smooth submission tracking, supervisor-student coordination, structured evaluation, and transparent progress monitoring throughout the thesis lifecycle.

Business requirements

Using the Thesis Portal, students, supervisors, and administrators will be able to manage the entire thesis process in one centralized platform. The specific functionality that the system should have includes the following:

- Develop a secure and role-based platform for student, supervisor, and administrator access with profile management.
- Enable thesis submission tracking for Registration, P1, P2, and P3, including supervisor approval, feedback, and grading.
- Provide group formation and supervisor assignment features to simplify collaboration and workload distribution.
- Support real-time notifications, messaging, and progress tracking to ensure smooth communication and timely updates.
- Maintain a repository of previous research documents with search, filtering, and bookmarking tools for academic reference.
- Equip supervisors and administrators with dashboards for monitoring submissions, evaluating work, and managing users.
- Offer additional tools such as one-click thesis templates and AI-powered report summarization to enhance efficiency.

Business value

We expect that the Thesis Portal will significantly improve the efficiency and transparency of the thesis process by providing a centralized platform for students, supervisors, and administrators. The portal will streamline submissions, supervisor assignments, feedback collection, and evaluation, reducing delays and miscommunication common in manual processes.

The system’s user-friendly design, real-time notifications, and progress tracking features will ensure students stay on schedule while supervisors can manage and evaluate multiple groups effectively. By reducing delays and miscommunication, the portal enhances academic quality and collaboration. Its scalable design ensures adaptability to institutional growth and evolving academic needs, supporting long-term usability and value.

Special issues or constraints

1. **Data Privacy and Security**
   Since the portal handles sensitive student and faculty data (like grades, personal info, and research papers), it must have strong security to prevent hacking, data leaks, or unauthorized access.

2. **Device Compatibility**
   Although the system is responsive, it may not look or work the same on all devices (especially older phones or browsers). Testing on different screen sizes is needed.

3. **AI Features Limitations**
   The AI summarizer may not always generate accurate summaries or suggestions, especially for complex or poorly formatted papers. Users should not rely on it 100%.

4. **Scalability Issues**
   If too many users (students, teachers, admins) use the system at the same time, performance might slow down unless the system is optimized for high traffic.

5. **Maintenance and Updates**
   The portal will need regular maintenance, bug fixes, and updates. If not updated, it might face compatibility issues or security risks in the future.

6. **Group Management Complexity**
   Group formation, supervisor requests, and coordination between students can get complex, especially if users don't follow proper steps. It may lead to confusion without clear instructions.

1. Functional Requirements

➔ **Role-based Authentication**

The system supports different user roles (Student, Teacher, and Admin) with appropriate permissions.

➔ **User Registration**

Students, teachers, and administrators can create accounts by providing details such as name, email, ID, and password.

➔ **User Login/Logout**

Users can securely log in and log out using their registered credentials.

➔ **Change/Reset Password**

Users can reset or update their passwords securely.

➔ **Profile Page**

Students and teachers each have a profile page displaying academic information such as CGPA, department, supervisor expertise, and profile photo.

➔ **Edit Profile**

Students and teachers can update their profile information when necessary.

➔ **Dashboard**

Dashboards allow students and teachers to view their personalized information, thesis submissions, and activities.

➔ **Admin Control**

Admins can manage users, assign or update marks for P1, P2, and P3 submissions, upload and update pre-formatted thesis templates (P1, P2, P3) for student download, adjust supervisor group limits to balance workload, and delete inappropriate posts from the platform.

➔ **Report Submission**

Students can upload Registration, P1, P2, and P3 reports in PDF format along with metadata (title, submission date, and group details).

➔ **Supervisor Review**

Supervisors can approve or reject reports and provide feedback.

➔ **Grading System**

Supervisors and admins can assign marks for P1, P2, and P3 submissions.

➔ **Progress Tracking**

Students can view the timeline of their thesis progress, submission status, and marks.

➔ **Notifications**

Students receive notifications about submission status, supervisor feedback, and marks.

➔ **Previous Research Repository**

A searchable database of past thesis documents is available for students. Faculty can upload new research papers to the repository.

➔ **Bookmarking**

Students can bookmark and remove bookmarks for important resources from the repository.

➔ **AI Summarizer**

The system provides an AI-based summarizer for submitted reports, highlighting strengths, weaknesses, and suggestions.

➔ **Group Formation**

Students can form groups of 1–4 members by sending or accepting group requests.

➔ **Supervisor Selection**

Students can request supervisors or co-supervisors, while supervisors can accept or reject based on availability.

➔ **Supervisor Assignment**

Once confirmed, all group members are automatically linked to the same supervisor.

➔ **Group Posts**

Students can create posts to find group members, with other users able to comment or like.

➔ **Feedback & Rating System**

Students can rate and leave feedback for supervisors after thesis completion.

➔ **Search & Filter**

Students can search for supervisors by name, department, or email, and filter submissions by status, group, or academic year.

➔ **Real-Time Chat**

Students and supervisors can exchange messages in real time.

➔ **Chat Search**

Users can search for specific contacts or messages in the chat system.

➔ **Chat Alerts**

Users are notified when a new message is received.

➔ **Notification System**

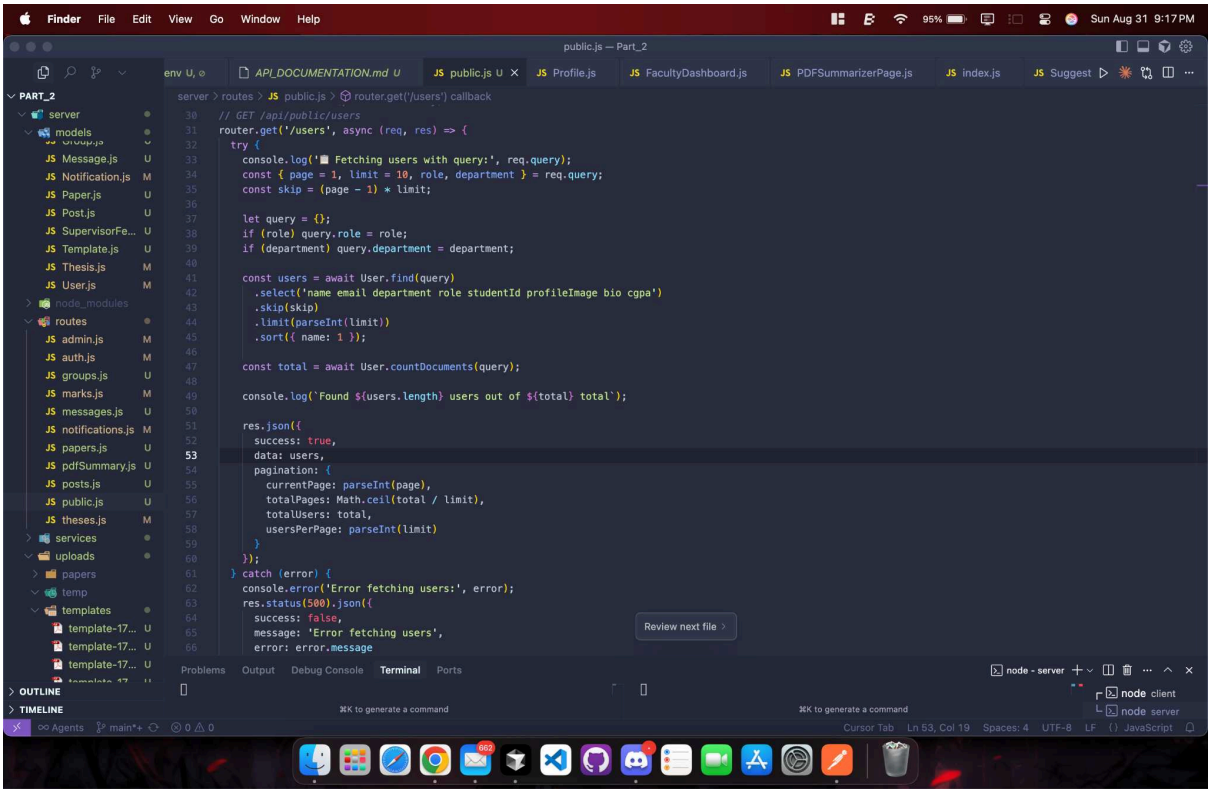
Students are notified about submission status, feedback, marks, and supervisor assignments, while supervisors receive notifications for new requests or submissions.

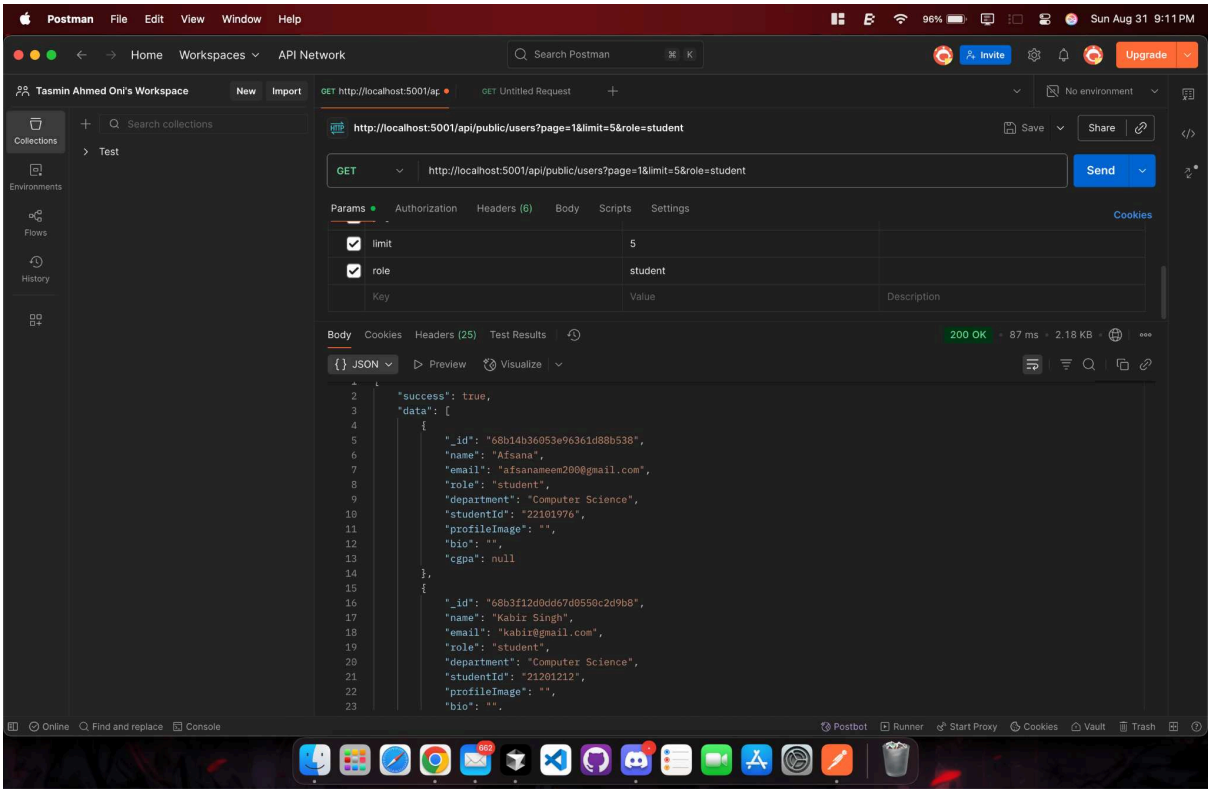
2. Technology (Framework, Languages)

➔ Language - JavaScript
➔ Frontend- React
➔ Backend- NodeJS and ExpressJS
➔ Database - MongoDB

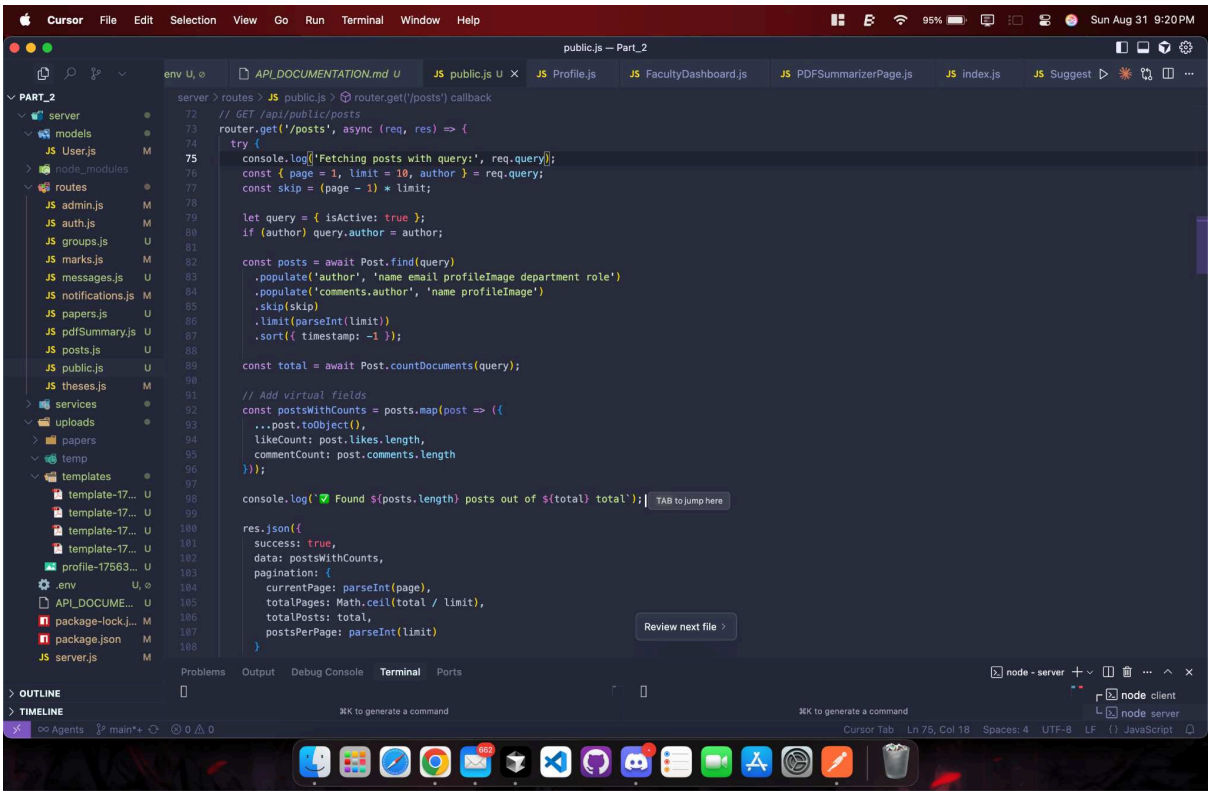
3. Backend Development

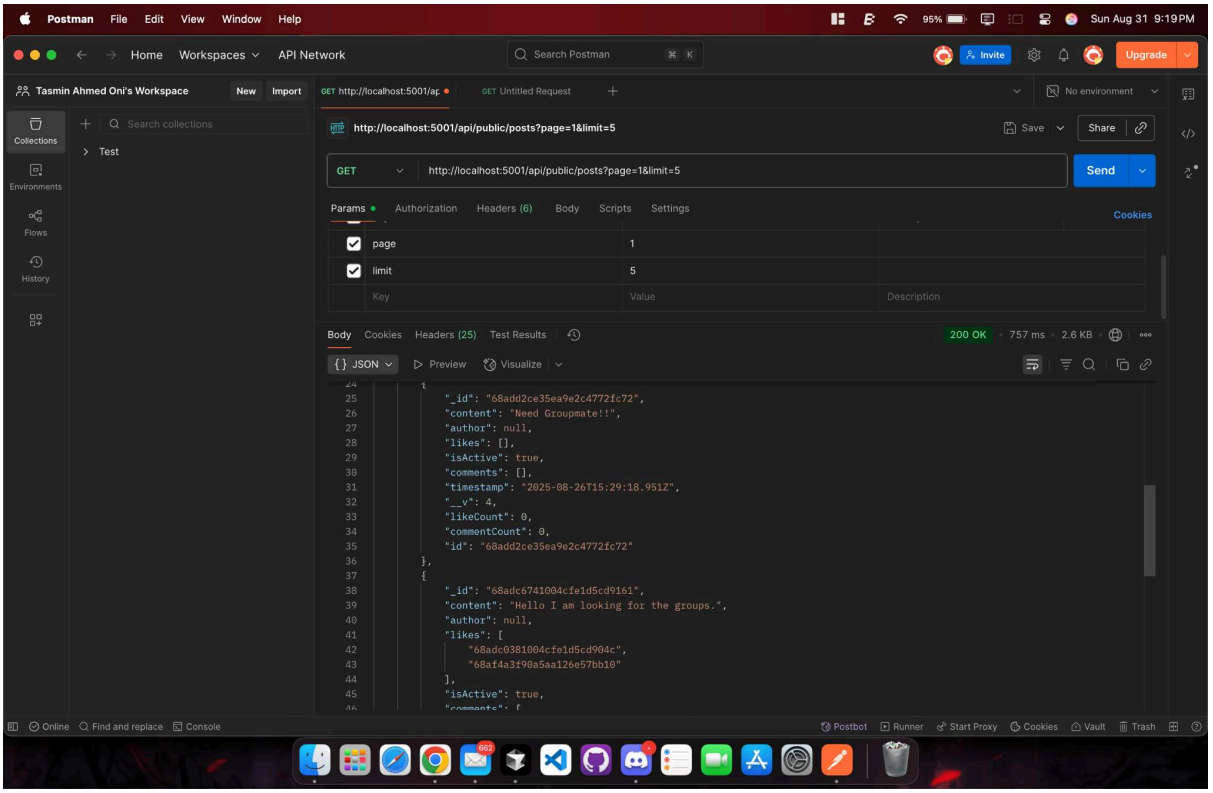
1. User Info:



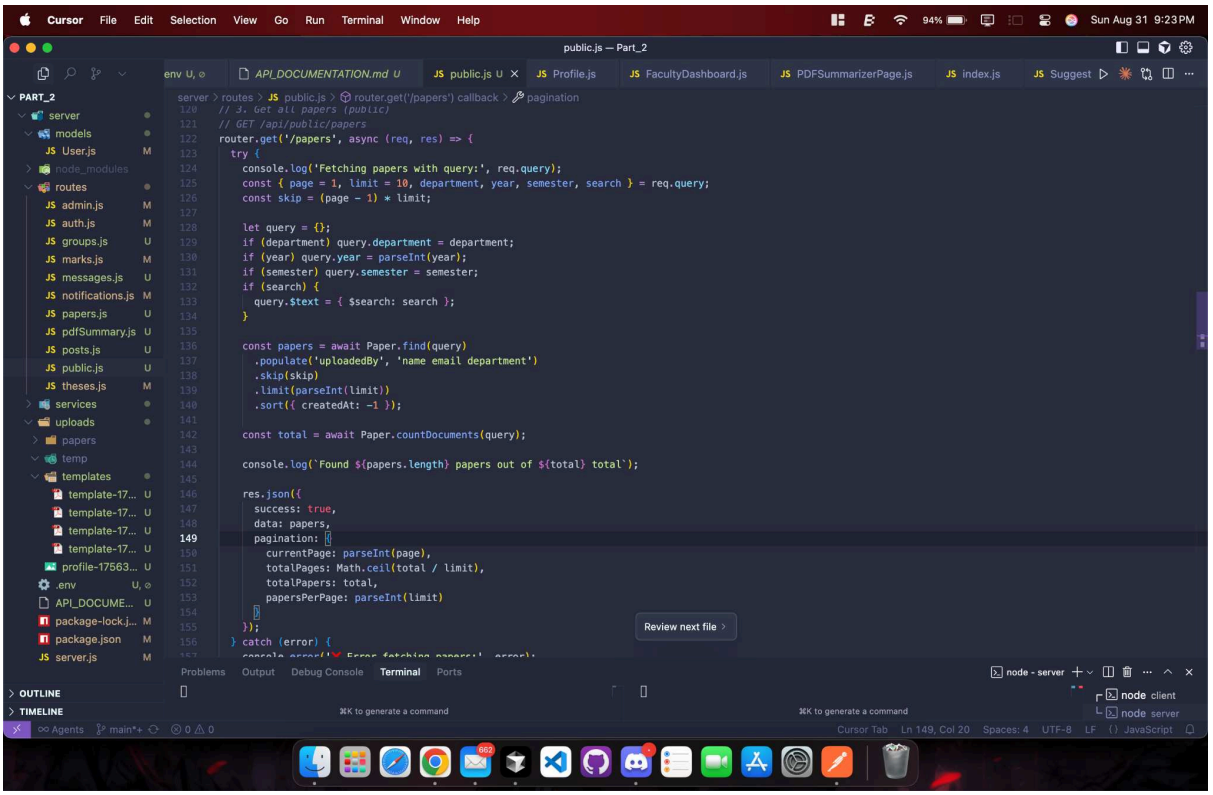


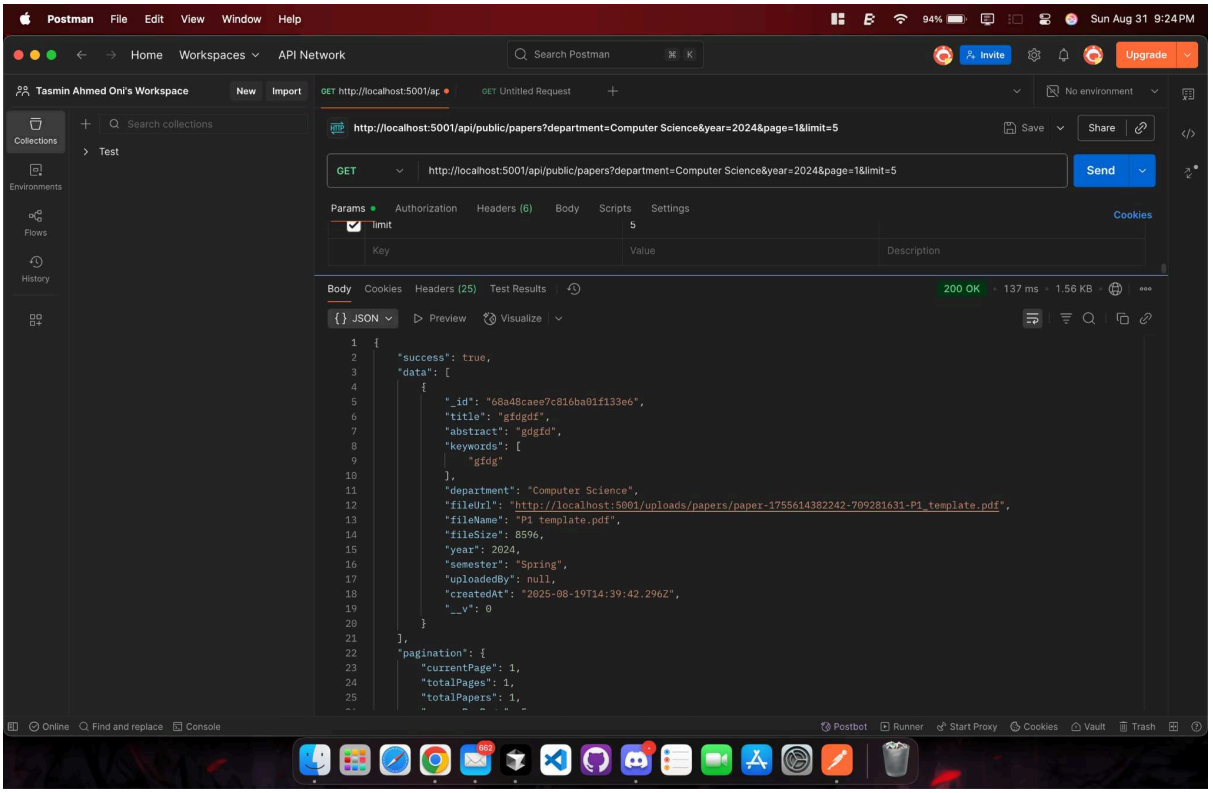
2. Users Posts:



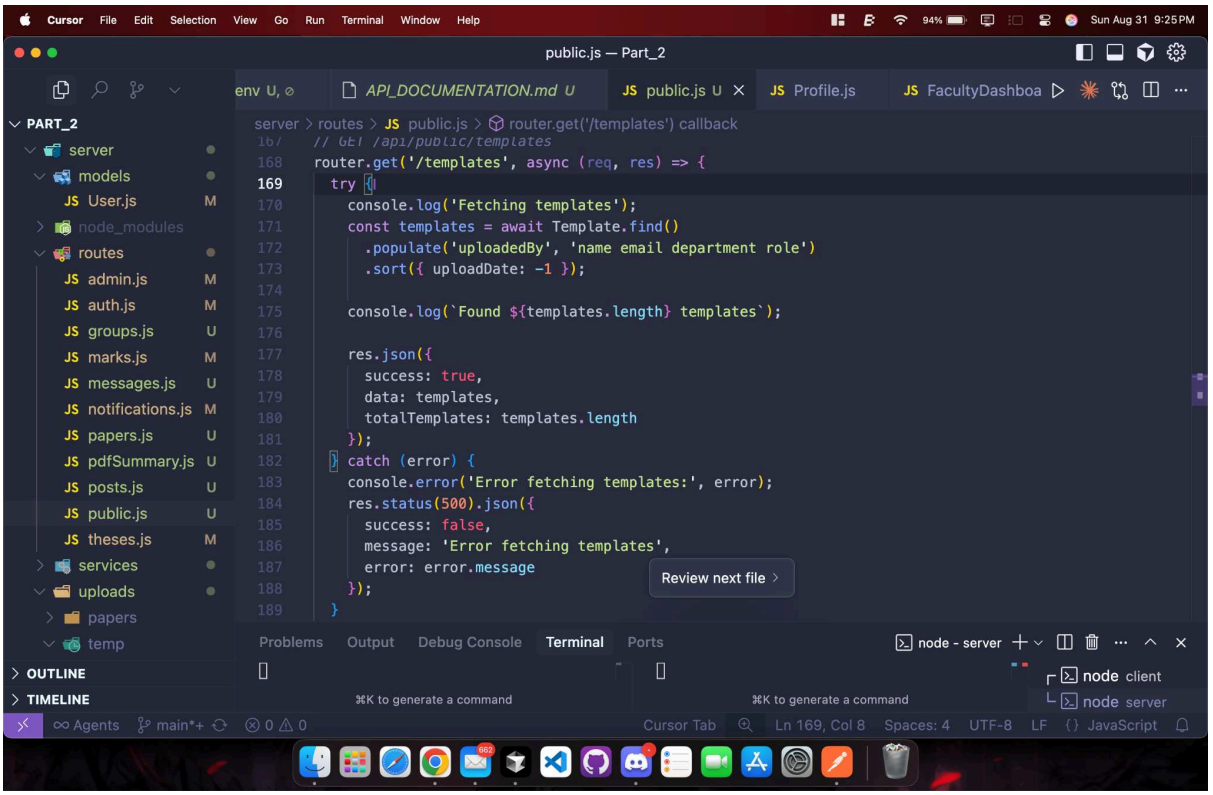

Postman File Edit View Window Help
Sun Aug 31 9:19 PM
Home Workspaces API Network
Search Postman
Invite Upgrade
Tasmin Ahmed Oni's Workspace
New Import
Collections Environments Flows History
Test
GET http://localhost:5001/api/public/posts?page=1&limit=5
Save Share
GET http://localhost:5001/api/public/posts?page=1&limit=5
Send
Params Authorization Headers (6) Body Scripts Settings
Cookies
page 1
limit 5
Key Value Description
Body Cookies Headers (25) Test Results
200 OK 757 ms 2.6 KB
{ } JSON Preview Visualize
{
"_id": "68add2ce35ea9e2c4772fc72",
"content": "Need Groupmate!!",
"author": null,
"likes": [],
"isActive": true,
"comments": [],
"timestamp": "2025-08-26T15:29:18.951Z",
"__v": 4,
"likeCount": 0,
"commentCount": 0,
"id": "68add2ce35ea9e2c4772fc72"
},
{
"_id": "68adc6741004cfe1d5cd9161",
"content": "Hello I am looking for the groups.",
"author": null,
"likes": [
"68adc0381004cfe1d5cd904c",
"68af4a3f90a5aa126e57bb10"
],
"isActive": true,
Online Find and replace Console
Postbot Runner Start Proxy Cookies Vault Trash

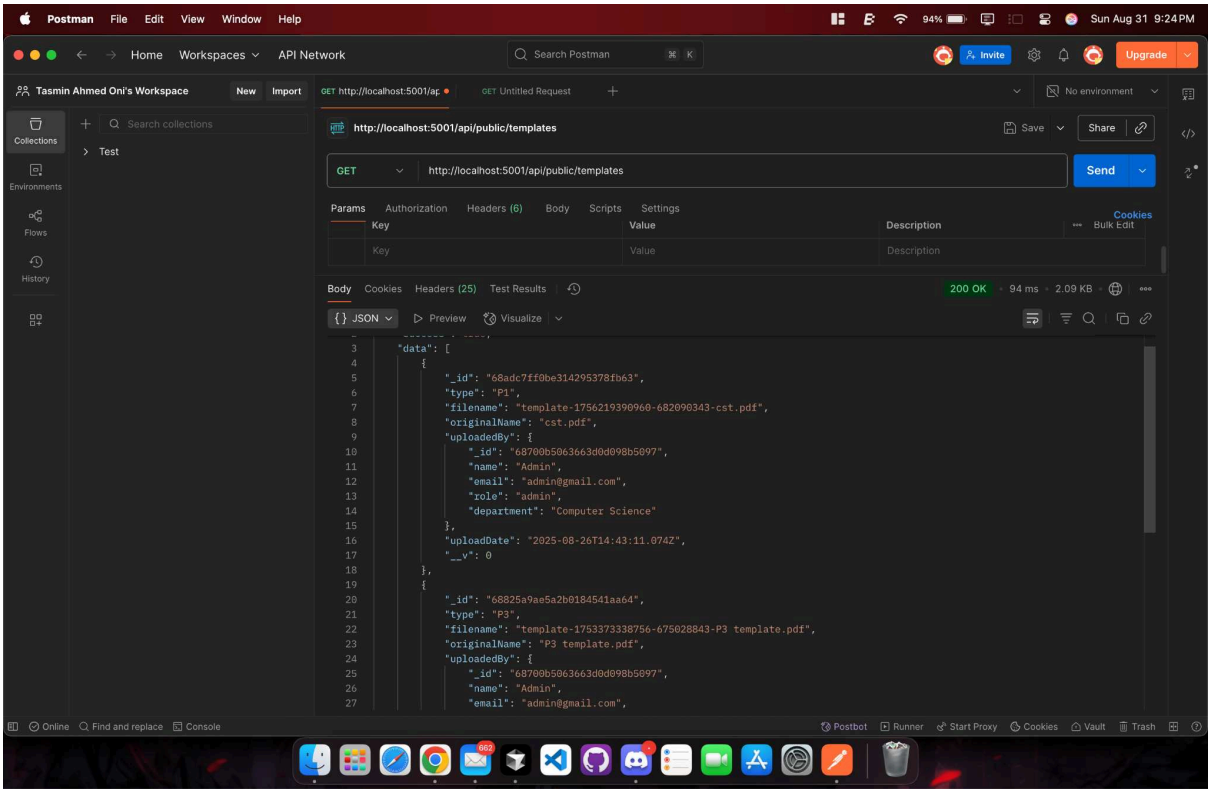
3. All Papers:


Cursor File Edit Selection View Go Run Terminal Window Help
Sun Aug 31 9:23 PM
public.js — Part_2
env U, ⊘ API_DOCUMENTATION.md U public.js U Profile.js FacultyDashboard.js PDFSummarizerPage.js index.js Suggest
PART_2
server
models
User.js M
node_modules
routes
admin.js M
auth.js M
groups.js U
marks.js M
messages.js U
notifications.js M
papers.js U
pdfSummary.js U
posts.js U
public.js U
theses.js M
services
uploads
papers
temp
templates
template-17... U
template-17... U
template-17... U
template-17... U
profile-17563... U
.env U, ⊘
API_DOCUME... U
package-lock.j... M
package.json M
server.js M
OUTLINE
TIMELINE
server > routes > public.js > router.get('/papers') callback > pagination
// 3. Get all papers (public)
// GET /api/public/papers
router.get('/papers', async (req, res) => {
try {
console.log('Fetching papers with query:', req.query);
const { page = 1, limit = 10, department, year, semester, search } = req.query;
const skip = (page - 1) * limit;
let query = {};
if (department) query.department = department;
if (year) query.year = parseInt(year);
if (semester) query.semester = semester;
if (search) {
query.$text = { $search: search };
}
const papers = await Paper.find(query)
.populate('uploadedBy', 'name email department')
.skip(skip)
.limit(parseInt(limit))
.sort({ createdAt: -1 });
const total = await Paper.countDocuments(query);
console.log(`Found ${papers.length} papers out of ${total} total`);
res.json({
success: true,
data: papers,
pagination: {
currentPage: parseInt(page),
totalPages: Math.ceil(total / limit),
totalPapers: total,
papersPerPage: parseInt(limit)
}
});
} catch (error) {
Review next file
Problems Output Debug Console Terminal Ports
node - server
node client
node server
⌘K to generate a command
⌘K to generate a command
Cursor Tab Ln 149, Col 20 Spaces: 4 UTF-8 LF JavaScript

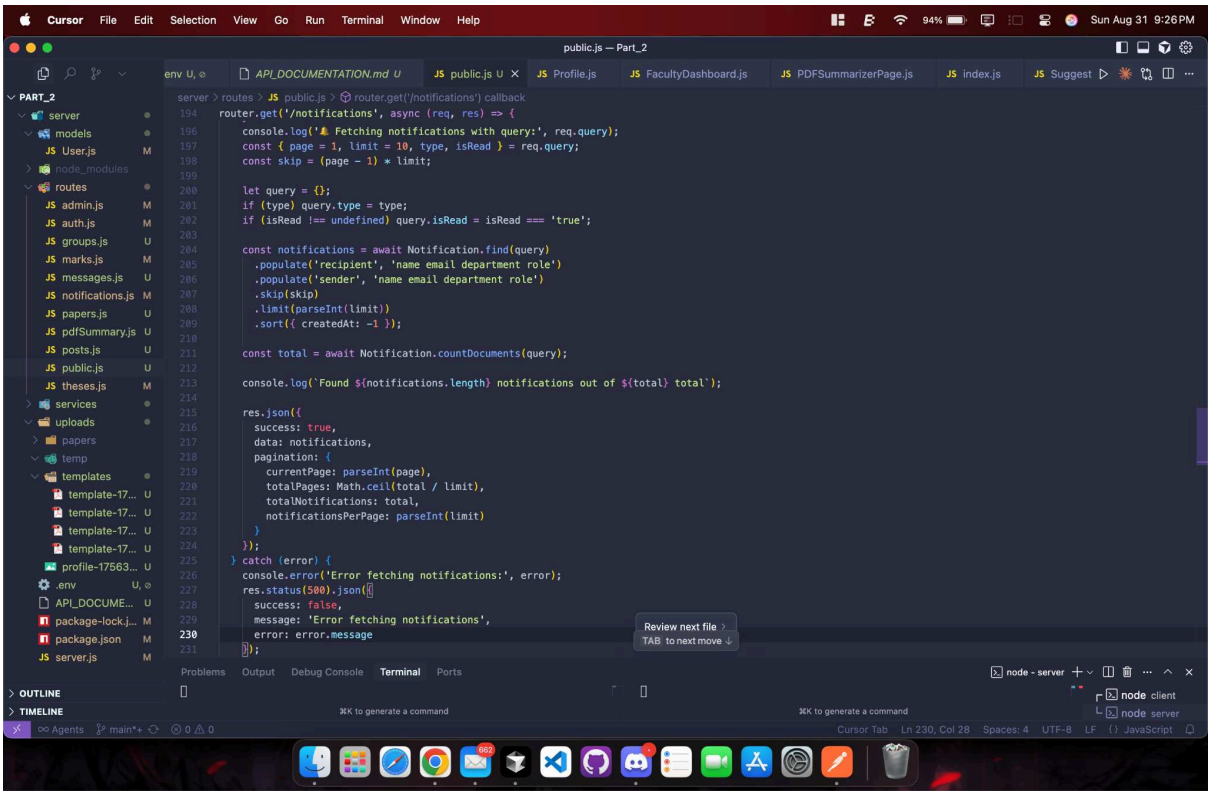

Postman File Edit View Window Help
Sun Aug 31 9:24 PM
Home Workspaces API Network
Search Postman
Invite
Upgrade
Tasmin Ahmed Oni's Workspace
New Import
Collections
Environments
Flows
History
Test
GET http://localhost:5001/ap
GET Untitled Request
No environment
http://localhost:5001/api/public/papers?department=Computer Science&year=2024&page=1&limit=5
Save
Share
GET
http://localhost:5001/api/public/papers?department=Computer Science&year=2024&page=1&limit=5
Send
Params
Authorization
Headers (6)
Body
Scripts
Settings
Cookies
limit
5
Key
Value
Description
Body
Cookies
Headers (25)
Test Results
200 OK
137 ms
1.56 KB
{ } JSON
Preview
Visualize
{
"success": true,
"data": [
{
"_id": "68a48caee7c816ba01f133e6",
"title": "gfdgdf",
"abstract": "gdgfd",
"keywords": [
"gfdg"
],
"department": "Computer Science",
"fileUrl": "http://localhost:5001/uploads/papers/paper-1755614382242-709281631-P1_template.pdf",
"fileName": "P1 template.pdf",
"fileSize": 8596,
"year": 2024,
"semester": "Spring",
"uploadedBy": null,
"createdAt": "2025-08-19T14:39:42.296Z",
"__v": 0
}
],
"pagination": {
"currentPage": 1,
"totalPages": 1,
"totalPapers": 1,
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash

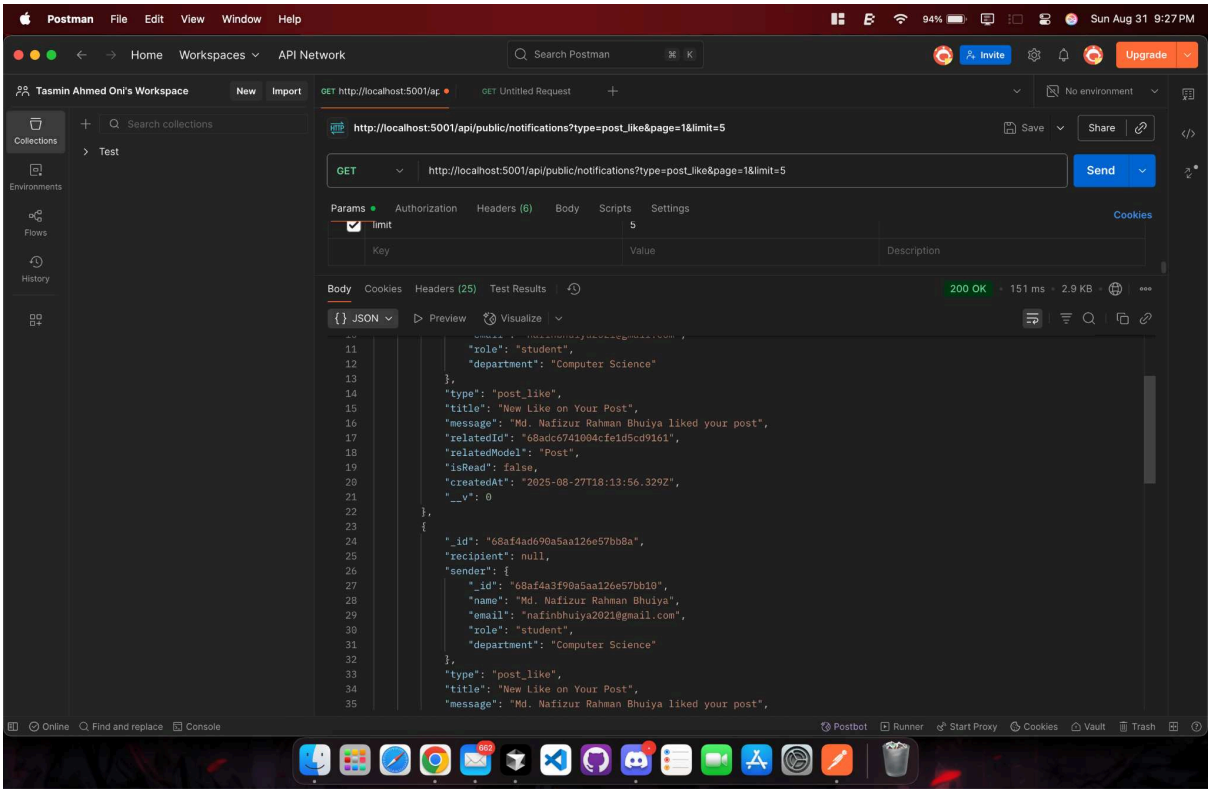
4. Templates:


Cursor File Edit Selection View Go Run Terminal Window Help
Sun Aug 31 9:25 PM
public.js — Part_2
env U,
API_DOCUMENTATION.md U
public.js U
Profile.js
FacultyDashboa
PART_2
server
models
User.js M
node_modules
routes
admin.js M
auth.js M
groups.js U
marks.js M
messages.js U
notifications.js M
papers.js U
pdfSummary.js U
posts.js U
public.js U
theses.js M
services
uploads
papers
temp
OUTLINE
TIMELINE
server > routes > public.js > router.get('/templates') callback
// GET /api/public/templates
router.get('/templates', async (req, res) => {
try {
console.log('Fetching templates');
const templates = await Template.find()
.populate('uploadedBy', 'name email department role')
.sort({ uploadDate: -1 });
console.log(`Found ${templates.length} templates`);
res.json({
success: true,
data: templates,
totalTemplates: templates.length
});
} catch (error) {
console.error('Error fetching templates:', error);
res.status(500).json({
success: false,
message: 'Error fetching templates',
error: error.message
});
}
Review next file
Problems
Output
Debug Console
Terminal
Ports
node - server
node client
node server
⌘K to generate a command
⌘K to generate a command
Agents
main*+
0 0
Cursor Tab
Ln 169, Col 8
Spaces: 4
UTF-8
LF
JavaScript



5.Notification:

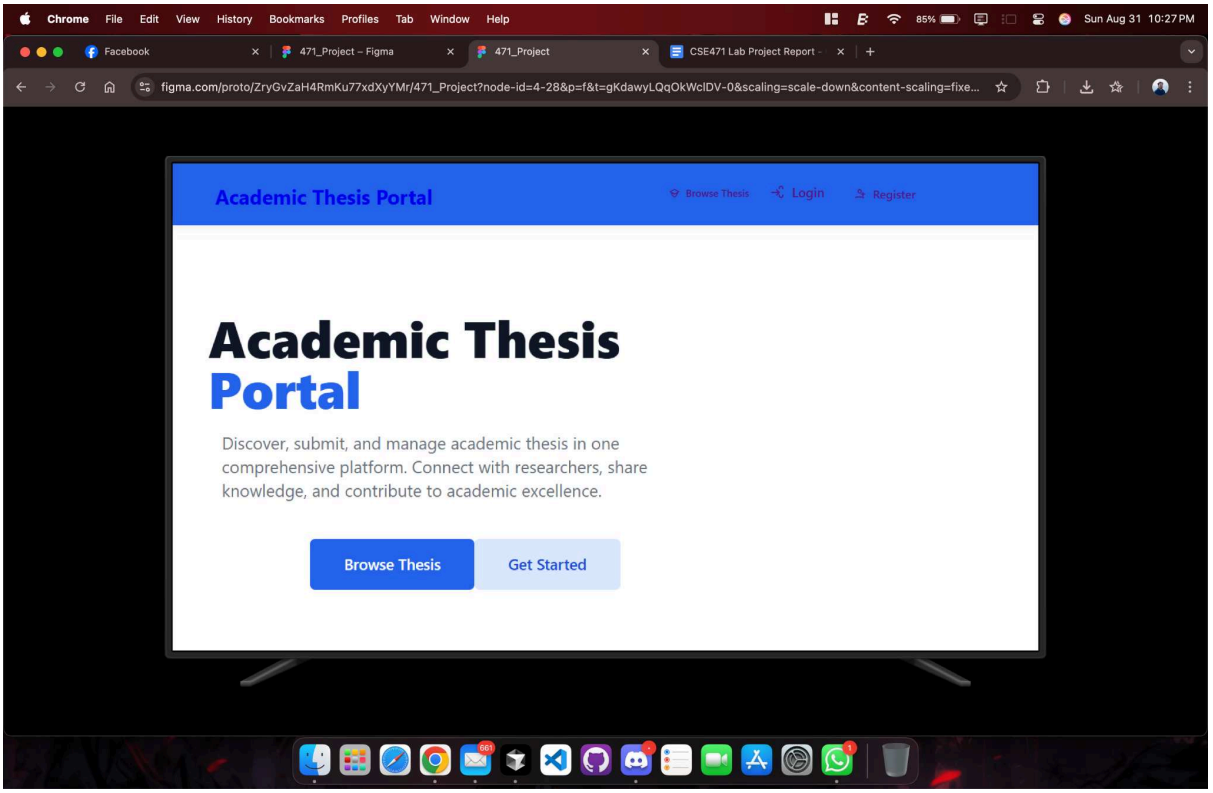


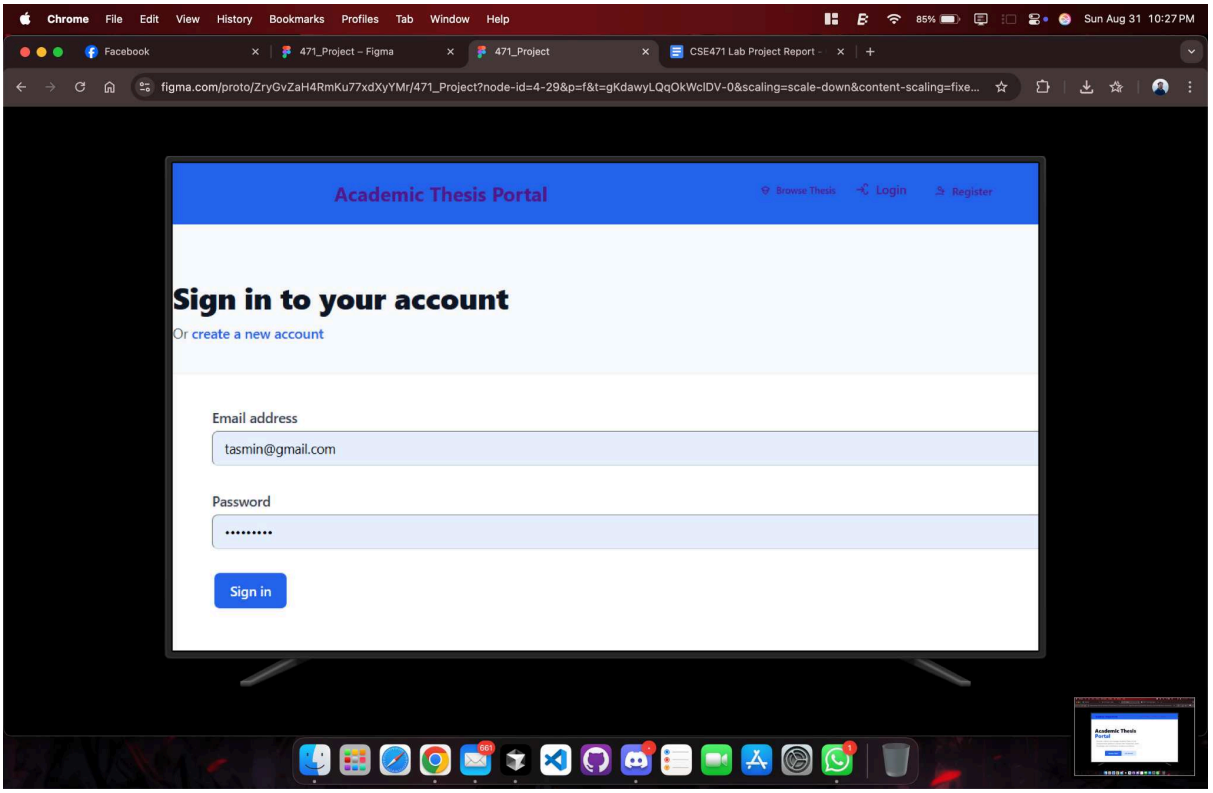


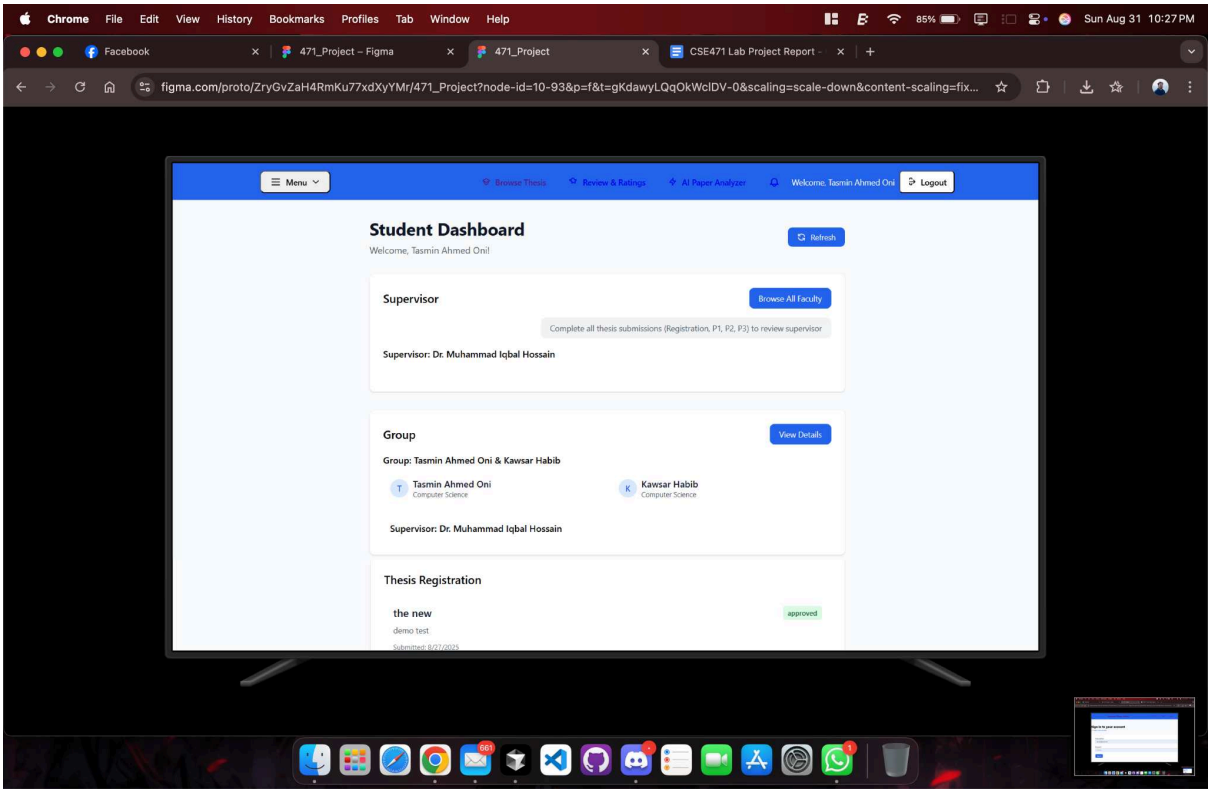
4. User Interface Design

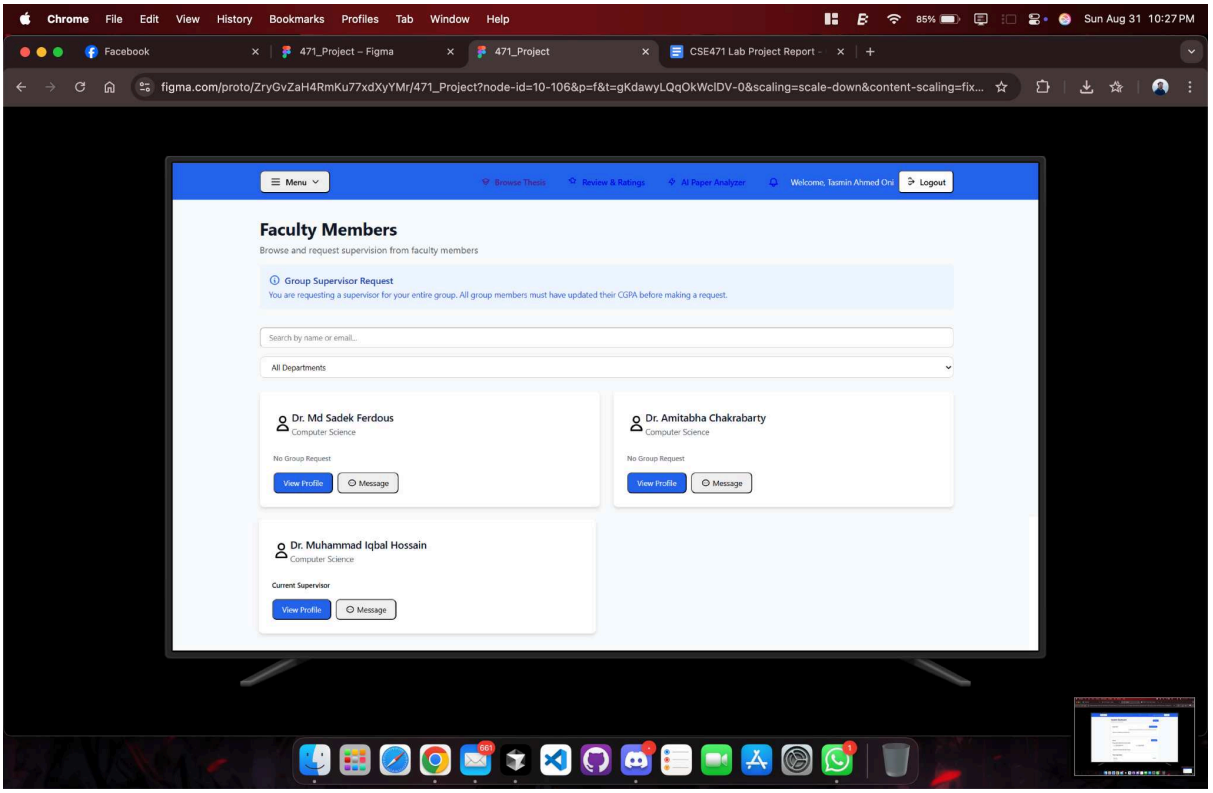
Figma Link:
https://www.figma.com/design/ZryGvZaH4RmKu77xdXyYMr/471_Project?node-id=0-1&p=f

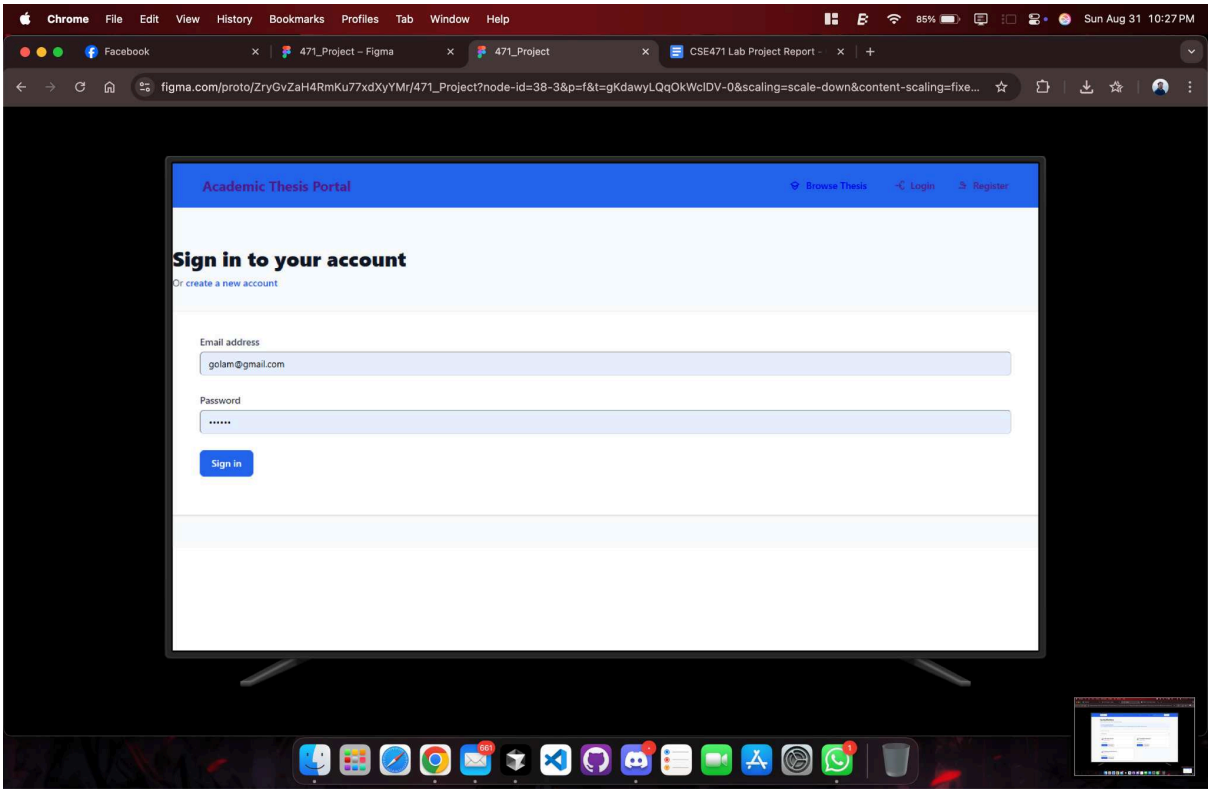
Prototype Link:
https://www.figma.com/proto/ZryGvZaH4RmKu77xdXyYMr/471_Project?node-id=4-28&t=ecd1g8ZD6oz0kEE5-1&starting-point-node-id=4%3A28

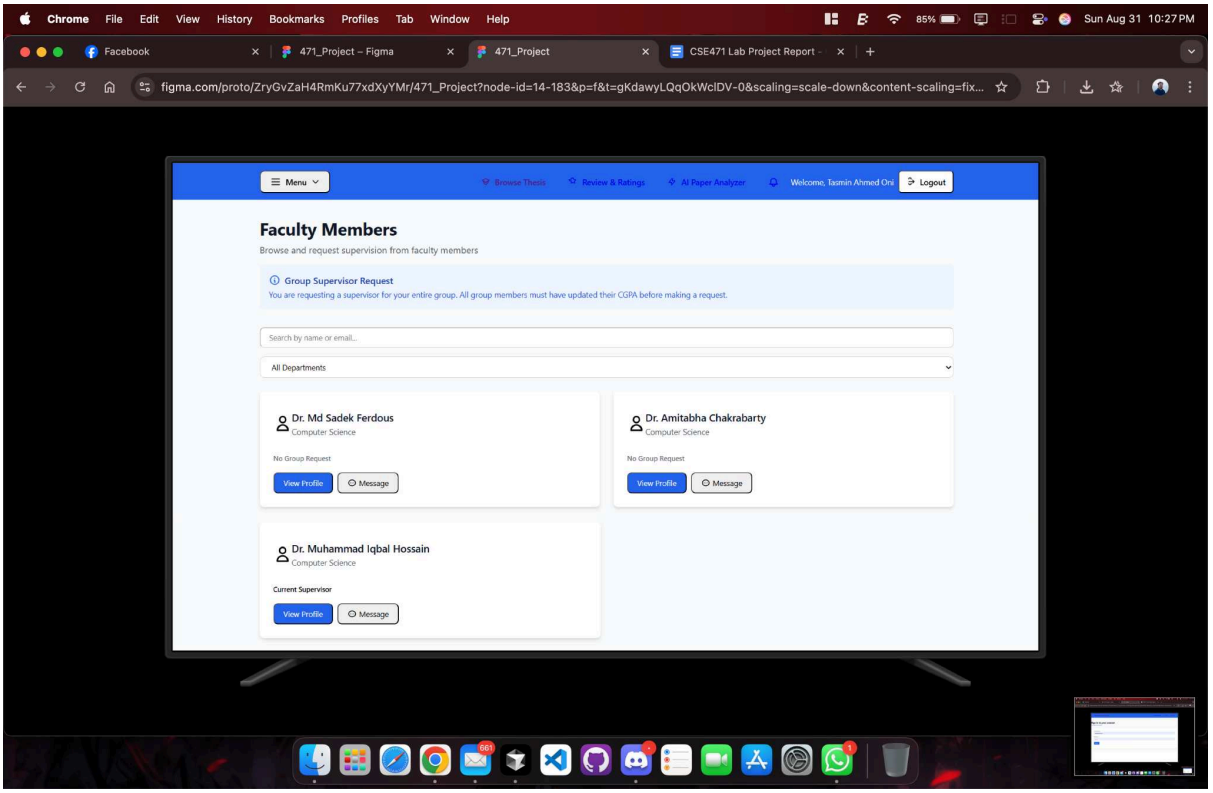

Academic Thesis Portal
Browse Thesis
Login
Register
Academic Thesis
Portal
Discover, submit, and manage academic thesis in one comprehensive platform. Connect with researchers, share knowledge, and contribute to academic excellence.
Browse Thesis
Get Started


Academic Thesis Portal
Browse Thesis
Login
Register
Sign in to your account
Or create a new account
Email address
tasmin@gmail.com
Password
Sign in


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun Aug 31 10:27 PM
Facebook
471_Project – Figma
471_Project
CSE471 Lab Project Report -
figma.com/proto/ZryGvZaH4RmKu77xdXyYMr/471_Project?node-id=10-93&p=f&t=gKdawyLQqOkWclDV-0&scaling=scale-down&content-scaling=fix...
Menu
Browse Thesis
Review & Ratings
AI Paper Analyzer
Welcome, Tasmin Ahmed Oni
Logout
Student Dashboard
Welcome, Tasmin Ahmed Oni!
Refresh
Supervisor
Browse All Faculty
Complete all thesis submissions (Registration, P1, P2, P3) to review supervisor
Supervisor: Dr. Muhammad Iqbal Hossain
Group
View Details
Group: Tasmin Ahmed Oni & Kawsar Habib
Tasmin Ahmed Oni
Computer Science
Kawsar Habib
Computer Science
Supervisor: Dr. Muhammad Iqbal Hossain
Thesis Registration
the new
approved
demo test


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun Aug 31 10:27 PM
Facebook
471_Project – Figma
471_Project
CSE471 Lab Project Report -
figma.com/proto/ZryGvZaH4RmKu77xdXyYMr/471_Project?node-id=10-106&p=f&t=gKdawyLQqOkWclDV-0&scaling=scale-down&content-scaling=fix...
Menu
Browse Thesis
Review & Ratings
AI Paper Analyzer
Welcome, Tasmin Ahmed Oni
Logout
Faculty Members
Browse and request supervision from faculty members
Group Supervisor Request
You are requesting a supervisor for your entire group. All group members must have updated their CGPA before making a request.
Search by name or email...
All Departments
Dr. Md Sadek Ferdous
Computer Science
No Group Request
View Profile
Message
Dr. Amitabha Chakrabarty
Computer Science
No Group Request
View Profile
Message
Dr. Muhammad Iqbal Hossain
Computer Science
Current Supervisor
View Profile
Message

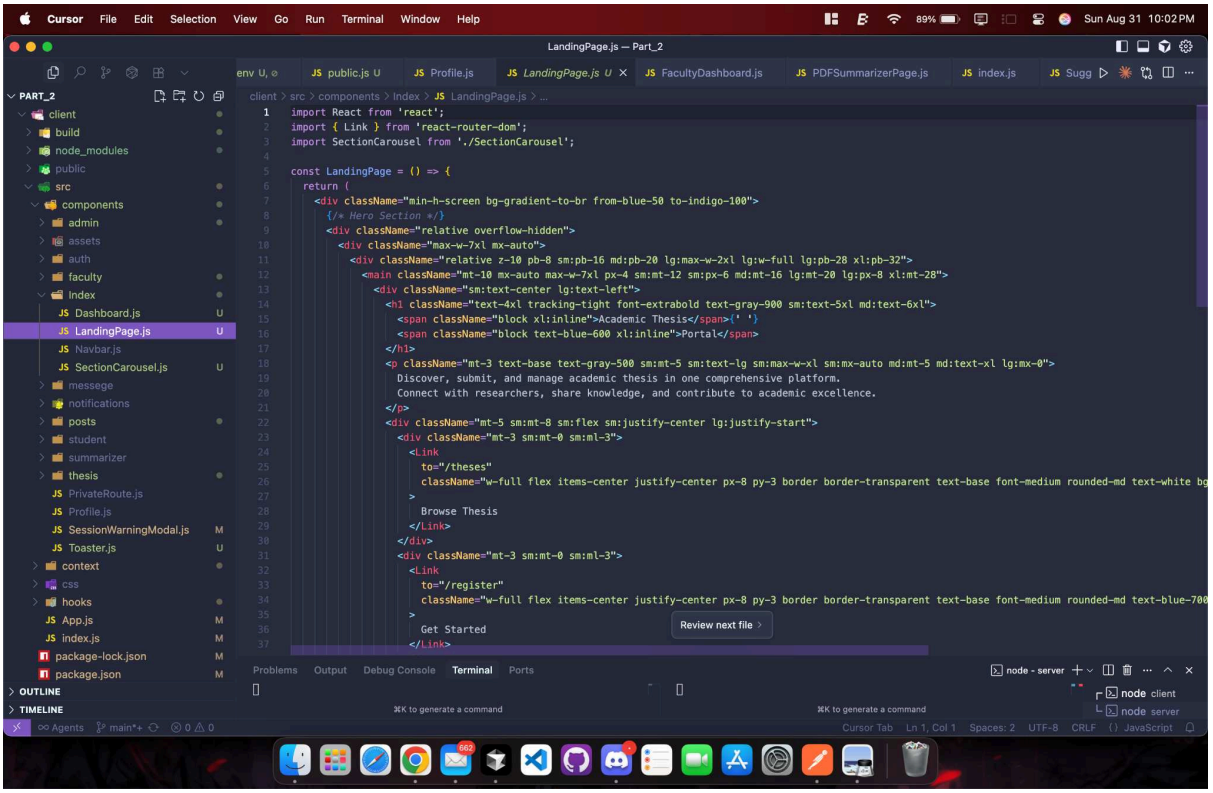


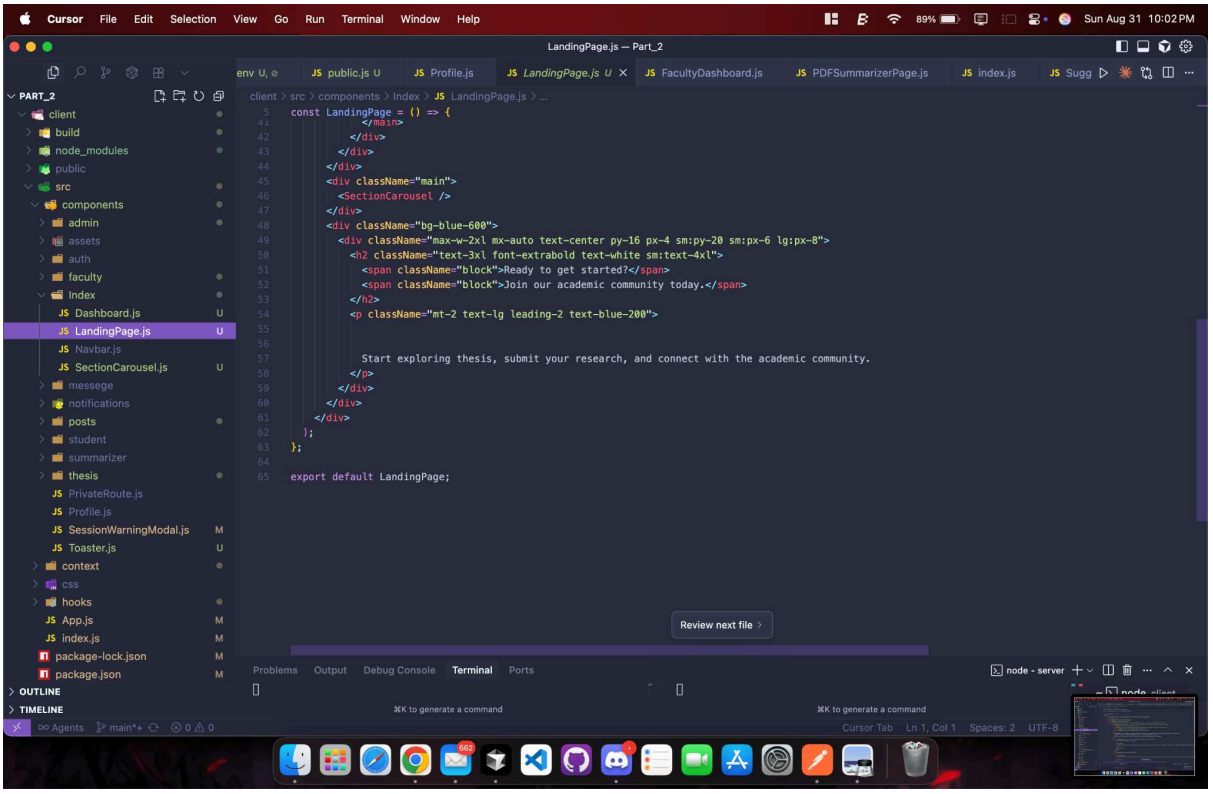


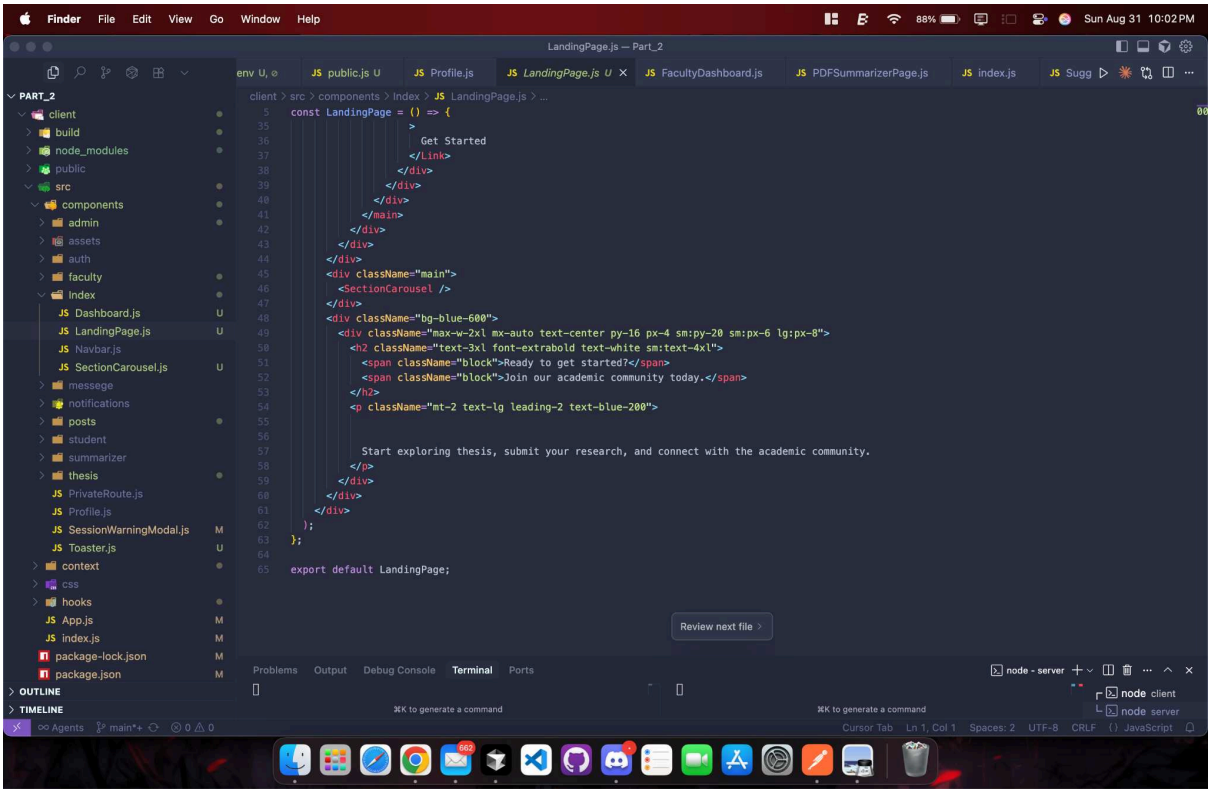
5. Frontend Development

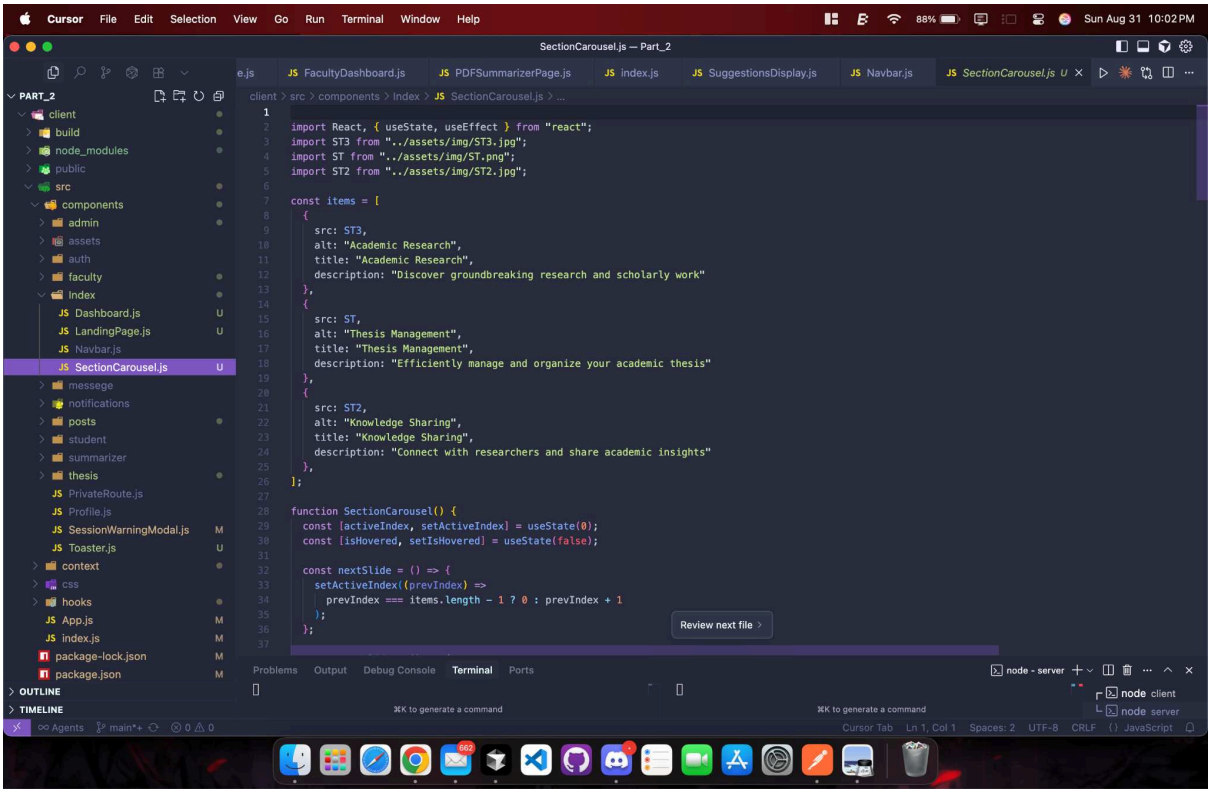
1.Landing Page: The landing page is the main entry point and introduction to the Academic Thesis Portal, designed to welcome visitors and provide an overview of the platform's features. When users first visit the site, they see a visually appealing hero section with

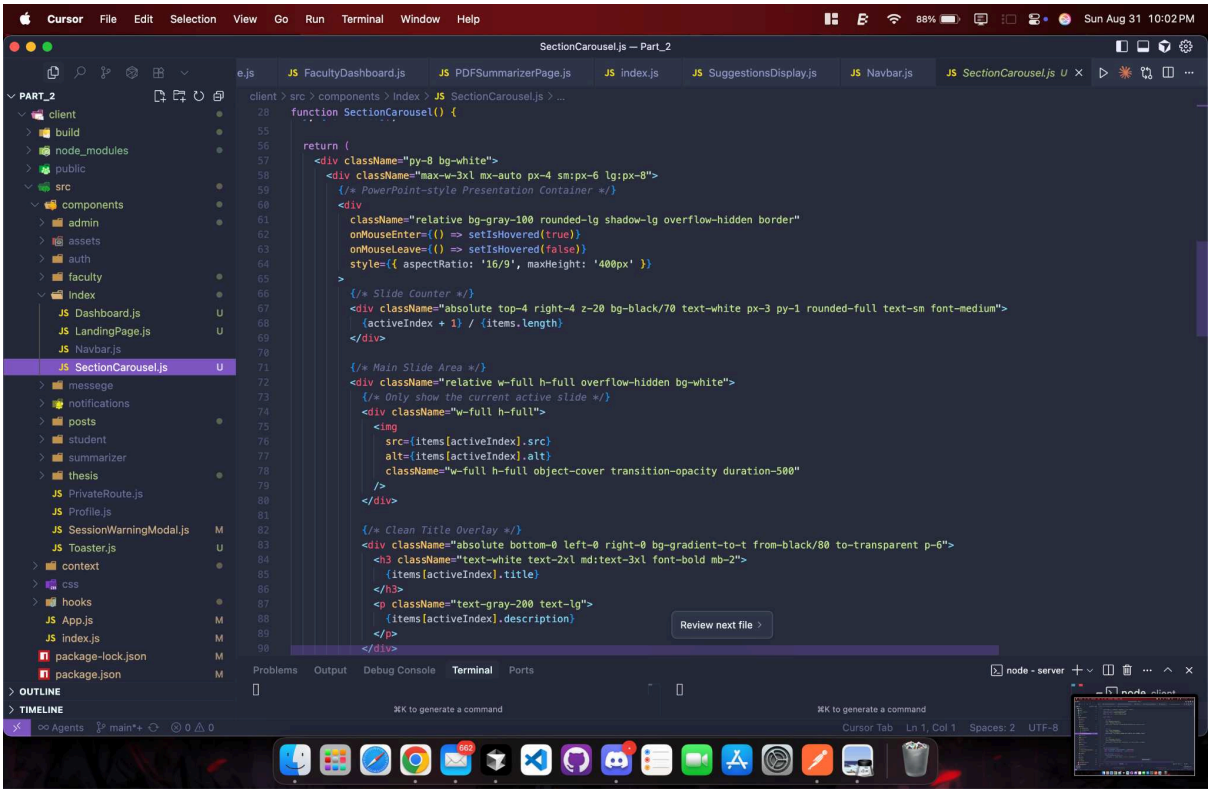
compelling headlines, descriptive text about the portal's purpose, and call-to-action buttons that guide them to either register for a new account or log in to an existing one.






LandingPage.js — Part_2
PART_2
client
build
node_modules
public
src
components
admin
assets
auth
faculty
Index
Dashboard.js
LandingPage.js
Navbar.js
SectionCarousel.js
messege
notifications
posts
student
summarizer
thesis
PrivateRoute.js
Profile.js
SessionWarningModal.js
Toaster.js
context
css
hooks
App.js
index.js
package-lock.json
package.json
const LandingPage = () => {
Get Started
Ready to get started?
Join our academic community today.
Start exploring thesis, submit your research, and connect with the academic community.
export default LandingPage;
Review next file
Problems
Output
Debug Console
Terminal
Ports
node - server
node client
node server
OUTLINE
TIMELINE


SectionCarousel.js — Part_2
PART_2
client
build
node_modules
public
src
components
admin
assets
auth
faculty
Index
Dashboard.js
LandingPage.js
Navbar.js
SectionCarousel.js
messege
notifications
posts
student
summarizer
thesis
PrivateRoute.js
Profile.js
SessionWarningModal.js
Toaster.js
context
css
hooks
App.js
index.js
package-lock.json
package.json
import React, { useState, useEffect } from "react";
import ST3 from "../assets/img/ST3.jpg";
import ST from "../assets/img/ST.png";
import ST2 from "../assets/img/ST2.jpg";
const items = [
src: ST3,
alt: "Academic Research",
title: "Academic Research",
description: "Discover groundbreaking research and scholarly work"
src: ST,
alt: "Thesis Management",
title: "Thesis Management",
description: "Efficiently manage and organize your academic thesis"
src: ST2,
alt: "Knowledge Sharing",
title: "Knowledge Sharing",
description: "Connect with researchers and share academic insights"
];
function SectionCarousel() {
const [activeIndex, setActiveIndex] = useState(0);
const [isHovered, setIsHovered] = useState(false);
const nextSlide = () => {
setActiveIndex((prevIndex) =>
prevIndex === items.length - 1 ? 0 : prevIndex + 1
);
};
Review next file
Problems
Output
Debug Console
Terminal
Ports
node - server
node client
node server
OUTLINE
TIMELINE


Cursor File Edit Selection View Go Run Terminal Window Help
Sun Aug 31 10:02 PM
SectionCarousel.js — Part_2
PART_2
client
build
node_modules
public
src
components
admin
assets
auth
faculty
Index
Dashboard.js
LandingPage.js
Navbar.js
SectionCarousel.js
messege
notifications
posts
student
summarizer
thesis
PrivateRoute.js
Profile.js
SessionWarningModal.js
Toaster.js
context
css
hooks
App.js
index.js
package-lock.json
package.json
OUTLINE
TIMELINE
FacultyDashboard.js
PDFSummarizerPage.js
index.js
SuggestionsDisplay.js
Navbar.js
SectionCarousel.js
client > src > components > Index > SectionCarousel.js > ...
function SectionCarousel() {
return (

{/* PowerPoint-style Presentation Container */}
<div
className="relative bg-gray-100 rounded-lg shadow-lg overflow-hidden border"
onMouseEnter={() => setIsHovered(true)}
onMouseLeave={() => setIsHovered(false)}
style={{ aspectRatio: '16/9', maxHeight: '400px' }}
>
{/* Slide Counter */}

{activeIndex + 1} / {items.length}

{/* Main Slide Area */}

{/* Only show the current active slide */}

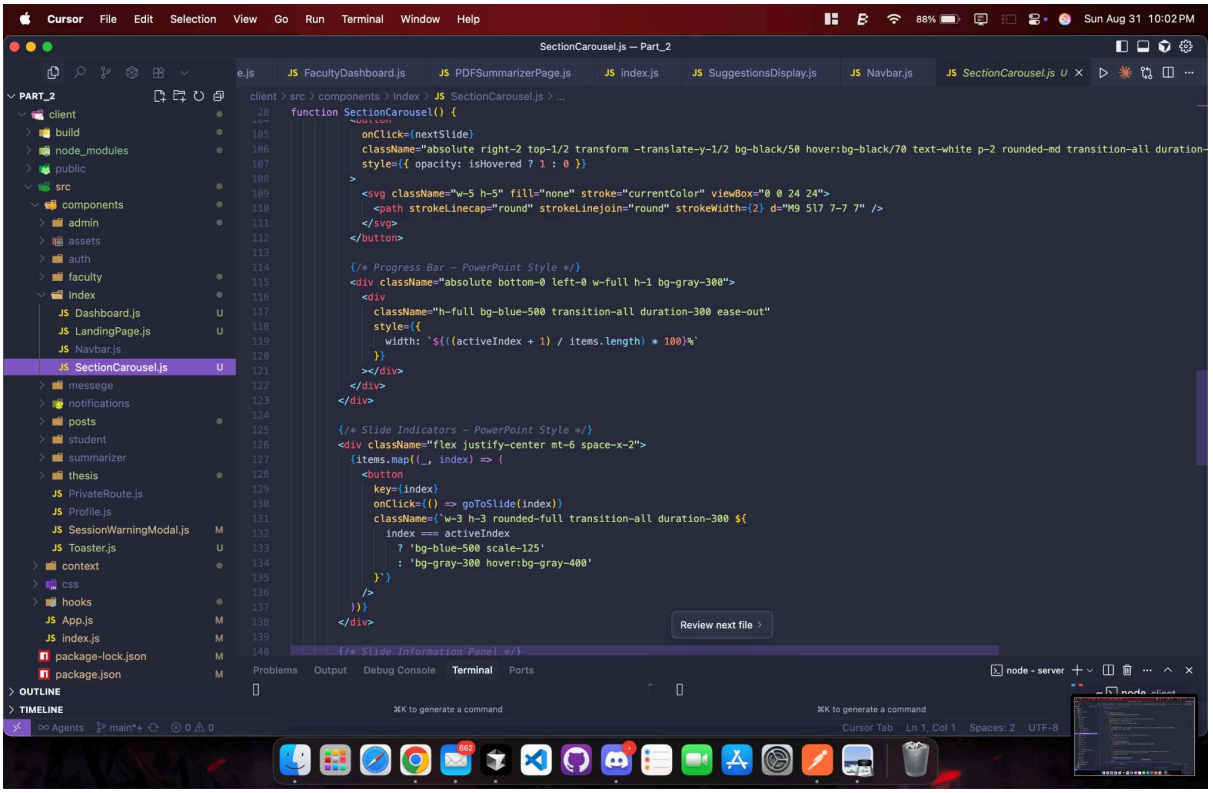
<img
src={items[activeIndex].src}
alt={items[activeIndex].alt}
className="w-full h-full object-cover transition-opacity duration-500"
/>

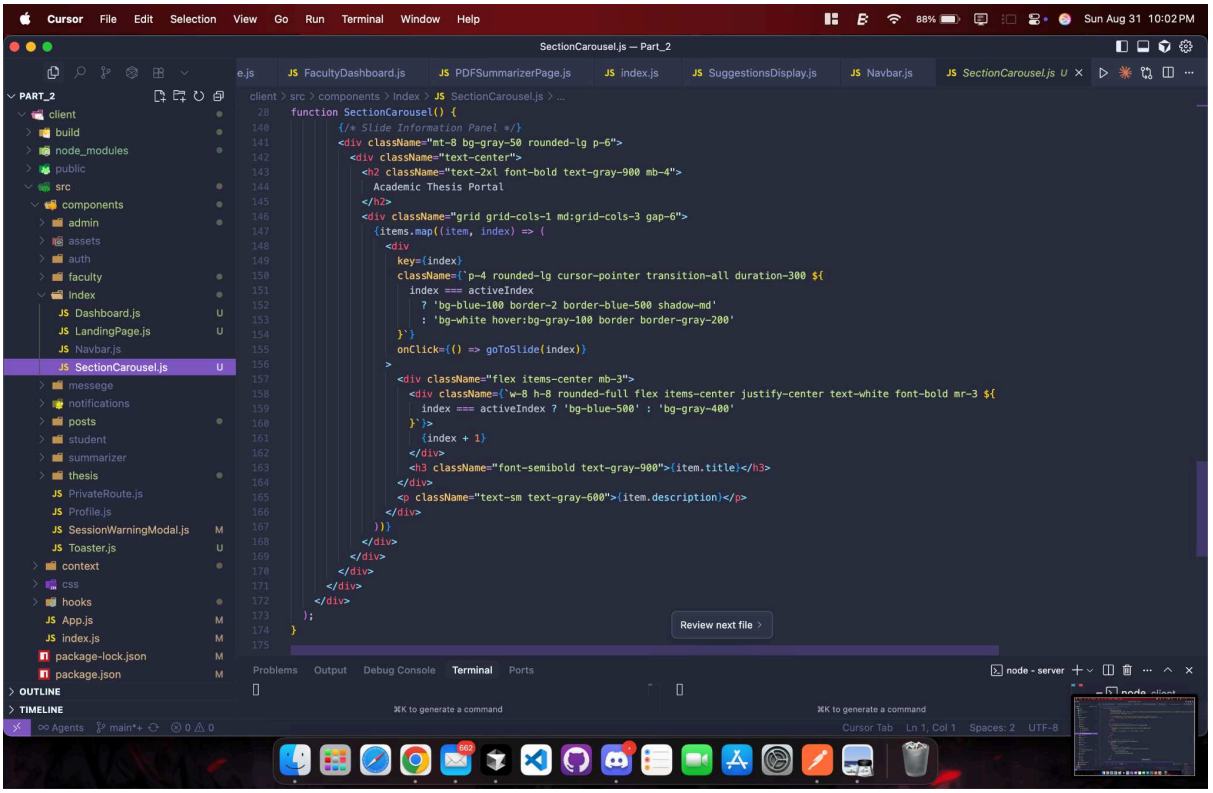
{/* Clean Title Overlay */}

<h3 className="text-white text-2xl md:text-3xl font-bold mb-2">
{items[activeIndex].title}
</h3>

{items[activeIndex].description}

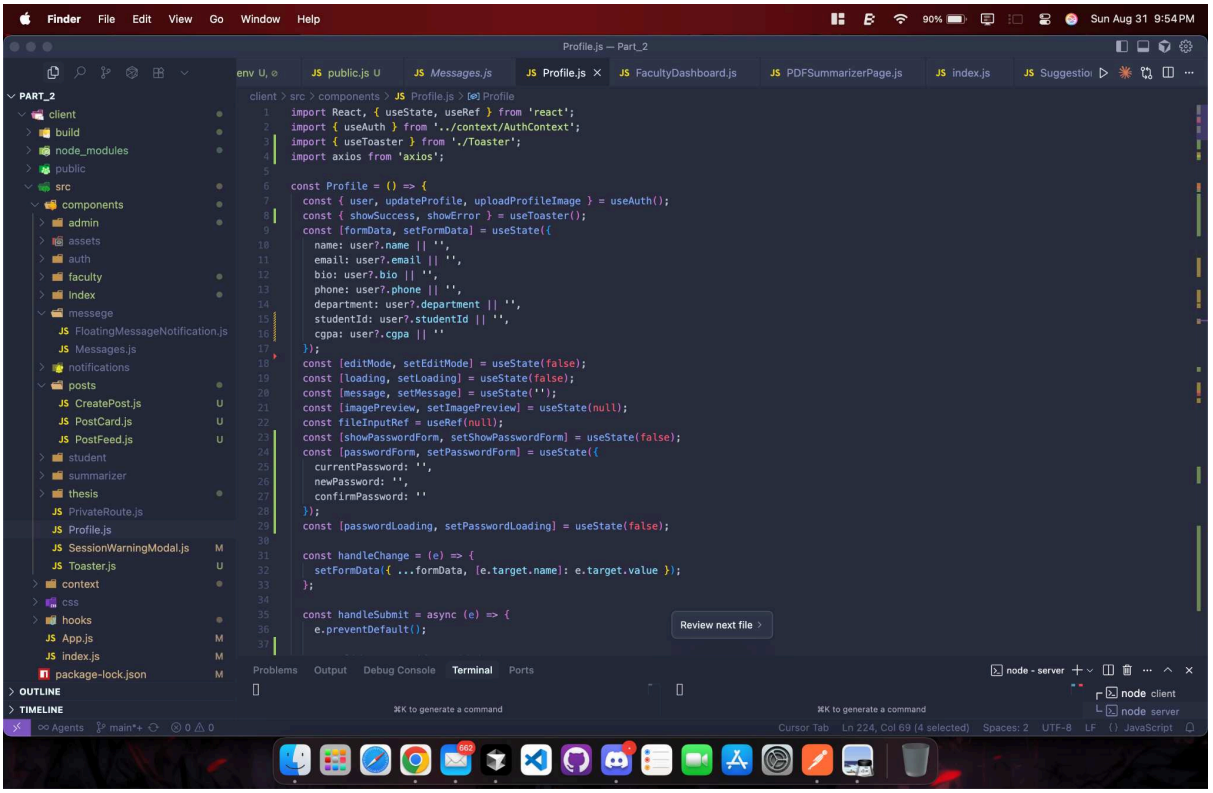
Review next file
Problems Output Debug Console Terminal Ports
node - server
⌘K to generate a command
⌘K to generate a command
Agents main*+ 0 0
Cursor Tab Ln 1, Col 1 Spaces: 2 UTF-8

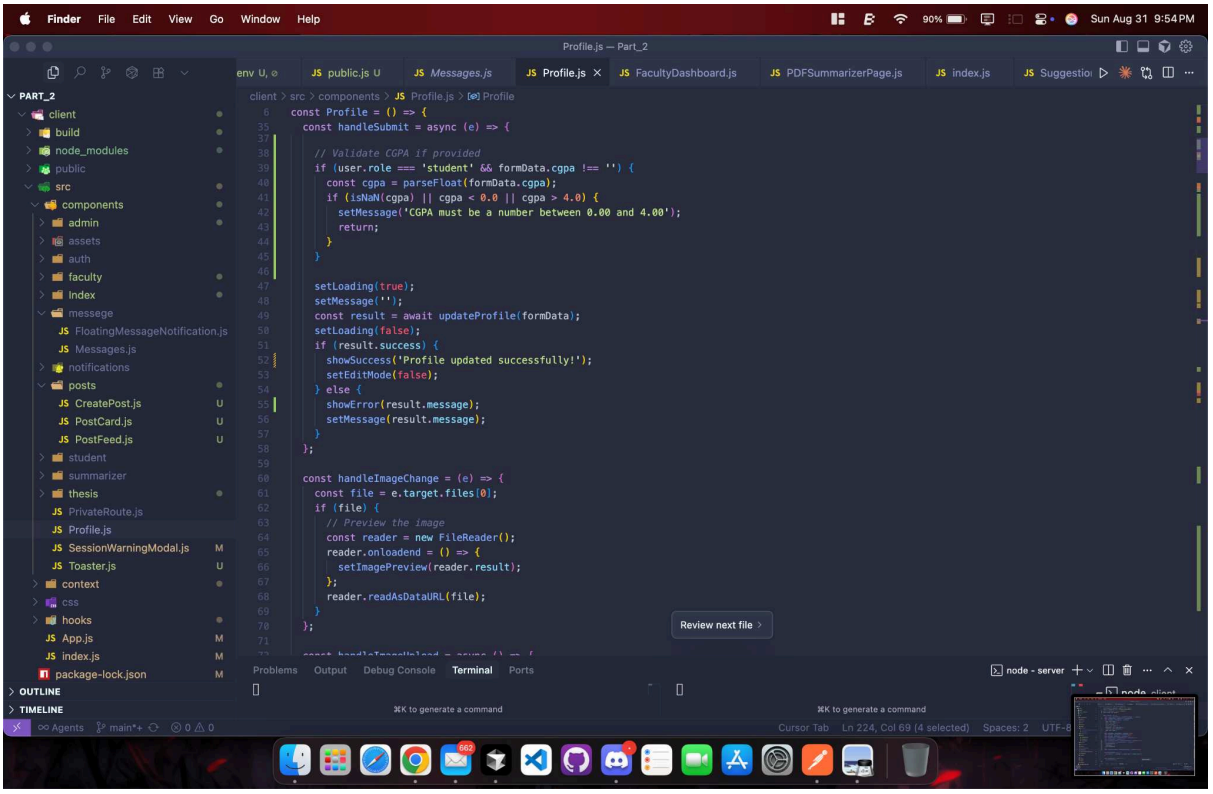


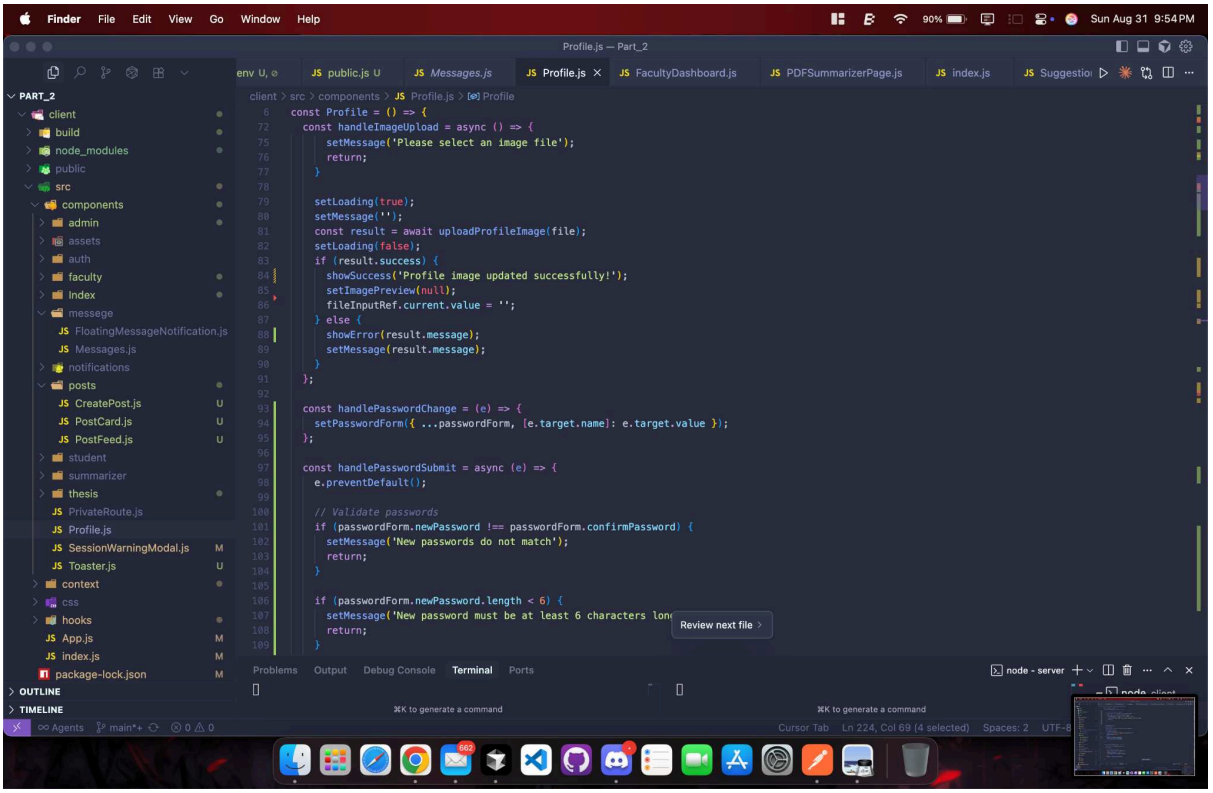


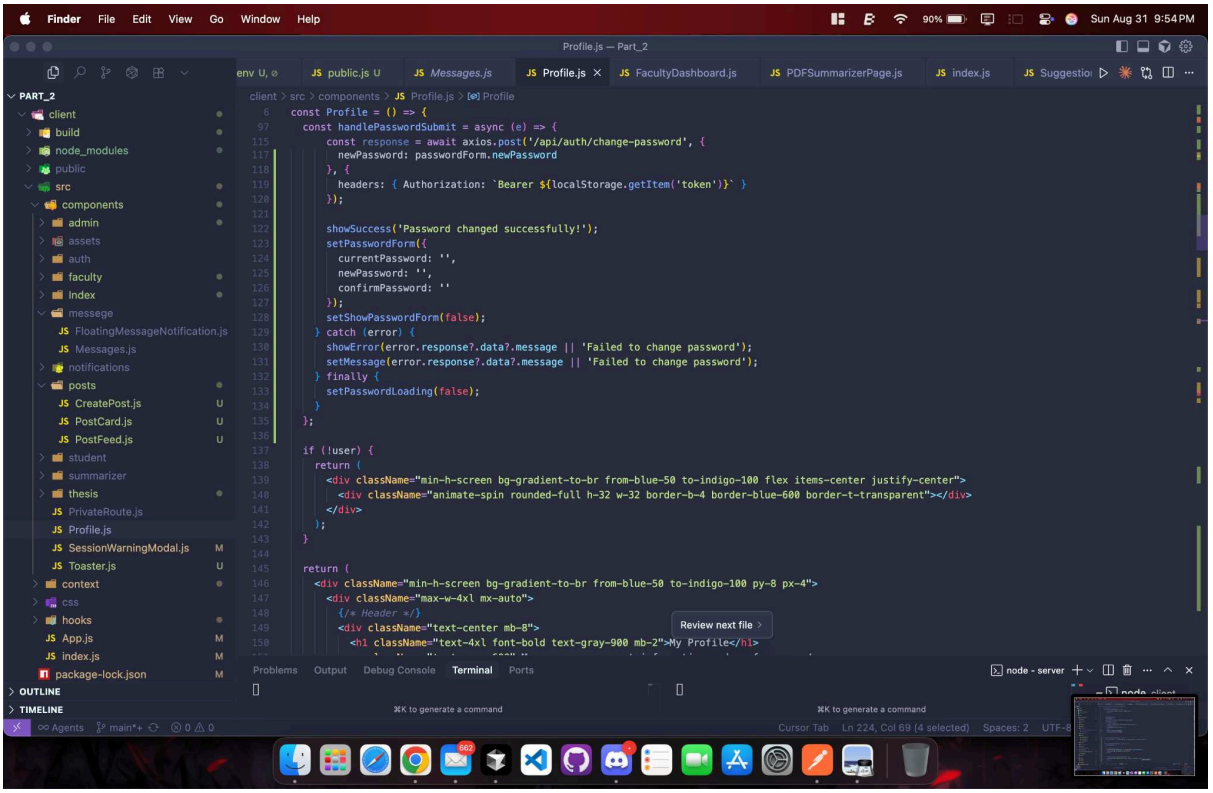
2.User Profile: The user profile display and editing interface, where users can view and modify their personal information. When a user navigates to their profile page, the frontend fetches their current profile data (including name, email, department, role, profile image, bio, and other details) from the backend API and displays it in a structured format. Users can edit their information by clicking edit buttons, which transforms the display into editable form

fields where they can update their bio, upload new profile pictures, change contact information, or modify other profile attributes. The frontend handles image uploads with preview functionality, validates form inputs, and sends updated data to the backend when users save changes.





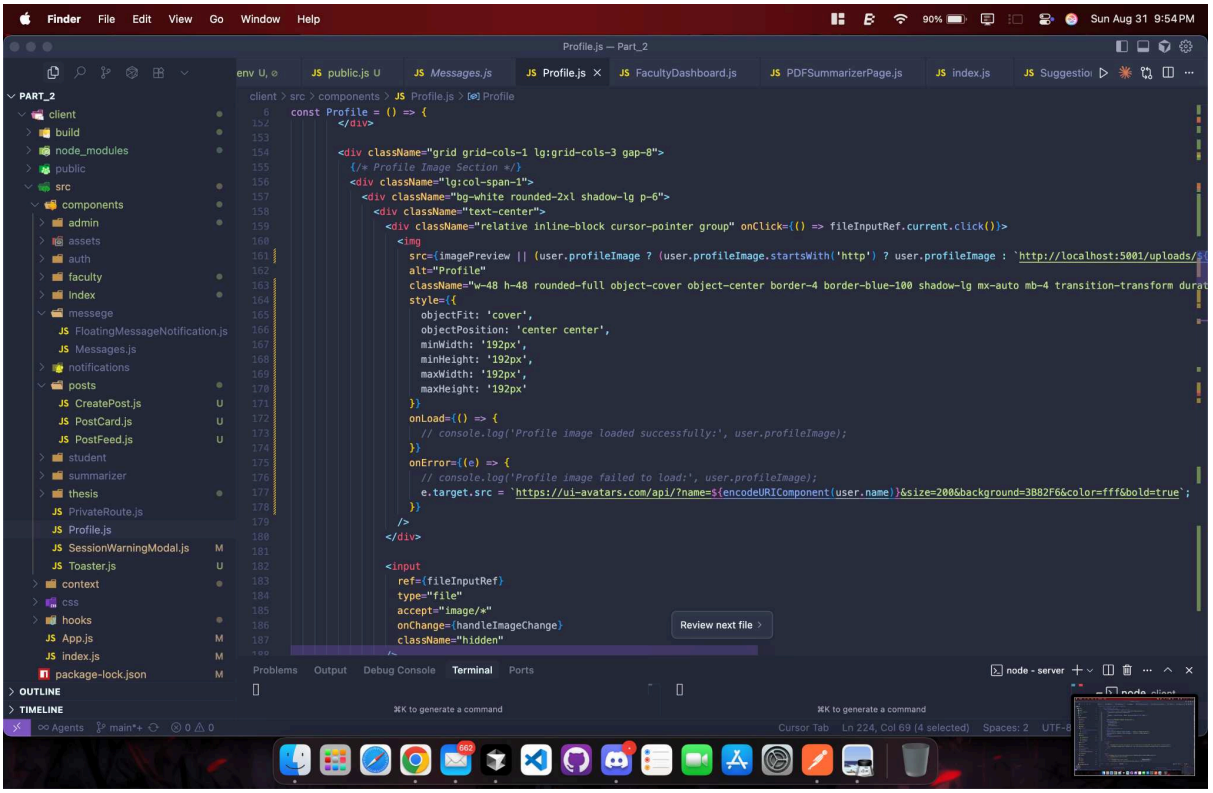

Finder File Edit View Go Window Help
Sun Aug 31 9:54 PM
Profile.js — Part_2
PART_2
client
build
node_modules
public
src
components
admin
assets
auth
faculty
Index
messege
FloatingMessageNotification.js
Messages.js
notifications
posts
CreatePost.js
PostCard.js
PostFeed.js
student
summarizer
thesis
PrivateRoute.js
Profile.js
SessionWarningModal.js
Toaster.js
context
css
hooks
App.js
index.js
package-lock.json
OUTLINE
TIMELINE
public.js U
Messages.js
Profile.js
FacultyDashboard.js
PDFSummarizerPage.js
index.js
client > src > components > Profile.js > Profile
const Profile = () => {
const handleImageUpload = async () => {
setMessage('Please select an image file');
return;
}
setLoading(true);
setMessage('');
const result = await uploadProfileImage(file);
setLoading(false);
if (result.success) {
showSuccess('Profile image updated successfully!');
setImagePreview(null);
fileInputRef.current.value = '';
} else {
showError(result.message);
setMessage(result.message);
}
};
const handlePasswordChange = (e) => {
setPasswordForm({ ...passwordForm, [e.target.name]: e.target.value });
};
const handlePasswordSubmit = async (e) => {
e.preventDefault();
// Validate passwords
if (passwordForm.newPassword !== passwordForm.confirmPassword) {
setMessage('New passwords do not match');
return;
}
if (passwordForm.newPassword.length < 6) {
setMessage('New password must be at least 6 characters lon
return;
}
Review next file
Problems Output Debug Console Terminal Ports
node - server
⌘K to generate a command
Cursor Tab Ln 224, Col 69 (4 selected) Spaces: 2 UTF-8


Finder File Edit View Go Window Help
Sun Aug 31 9:54 PM
Profile.js — Part_2
PART_2
client
build
node_modules
public
src
components
admin
assets
auth
faculty
Index
messege
FloatingMessageNotification.js
Messages.js
notifications
posts
CreatePost.js
PostCard.js
PostFeed.js
student
summarizer
thesis
PrivateRoute.js
Profile.js
SessionWarningModal.js
Toaster.js
context
css
hooks
App.js
index.js
package-lock.json
OUTLINE
TIMELINE
public.js U
Messages.js
Profile.js
FacultyDashboard.js
PDFSummarizerPage.js
index.js
client > src > components > Profile.js > Profile
const Profile = () => {
const handlePasswordSubmit = async (e) => {
const response = await axios.post('/api/auth/change-password', {
newPassword: passwordForm.newPassword
}, {
headers: { Authorization: `Bearer ${localStorage.getItem('token')}` }
});
showSuccess('Password changed successfully!');
setPasswordForm({
currentPassword: '',
newPassword: '',
confirmPassword: ''
});
setShowPasswordForm(false);
} catch (error) {
showError(error.response?.data?.message || 'Failed to change password');
setMessage(error.response?.data?.message || 'Failed to change password');
} finally {
setPasswordLoading(false);
}
};
if (!user) {
return (

);
}
return (

{/* Header */}

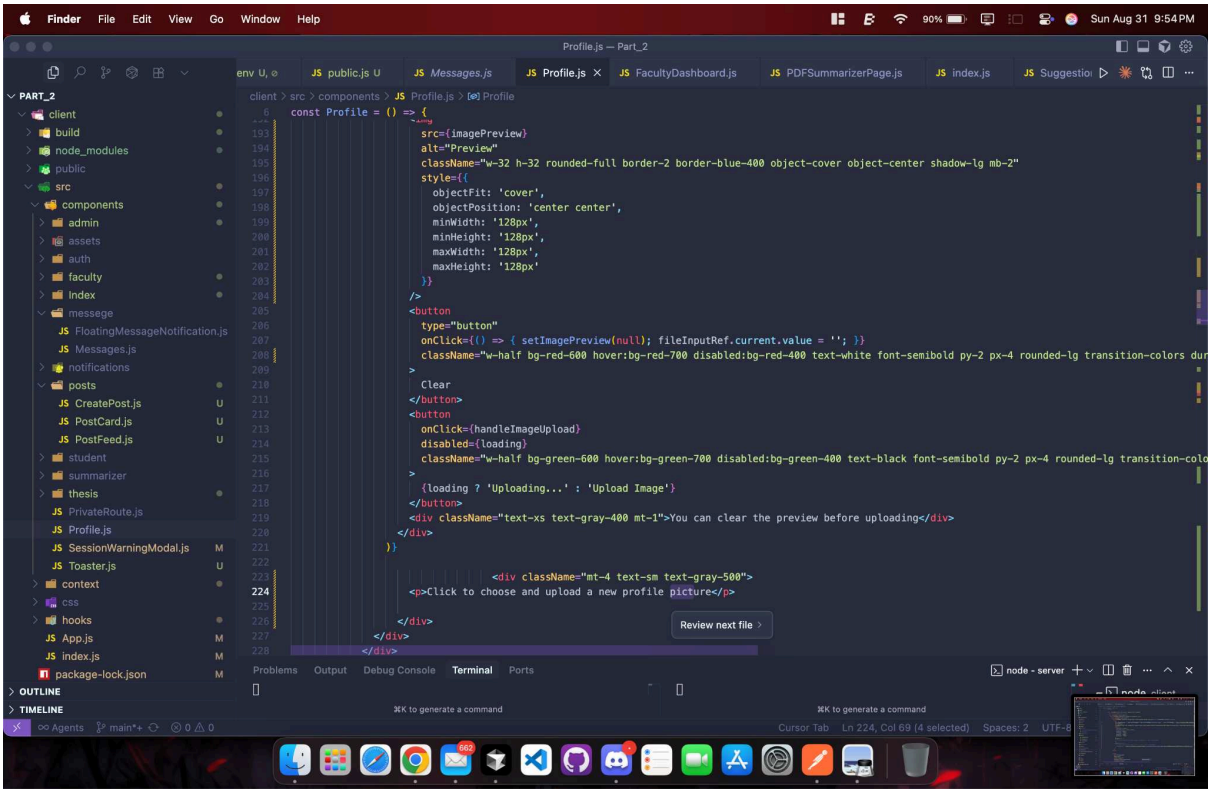
<h1 className="text-4xl font-bold text-gray-900 mb-2">My Profile</h1>
Review next file
Problems Output Debug Console Terminal Ports
node - server
⌘K to generate a command
Cursor Tab Ln 224, Col 69 (4 selected) Spaces: 2 UTF-8


Profile.js — Part_2
PART_2
client > src > components > Profile.js > Profile
const Profile = () => {

{/* Profile Image Section */}

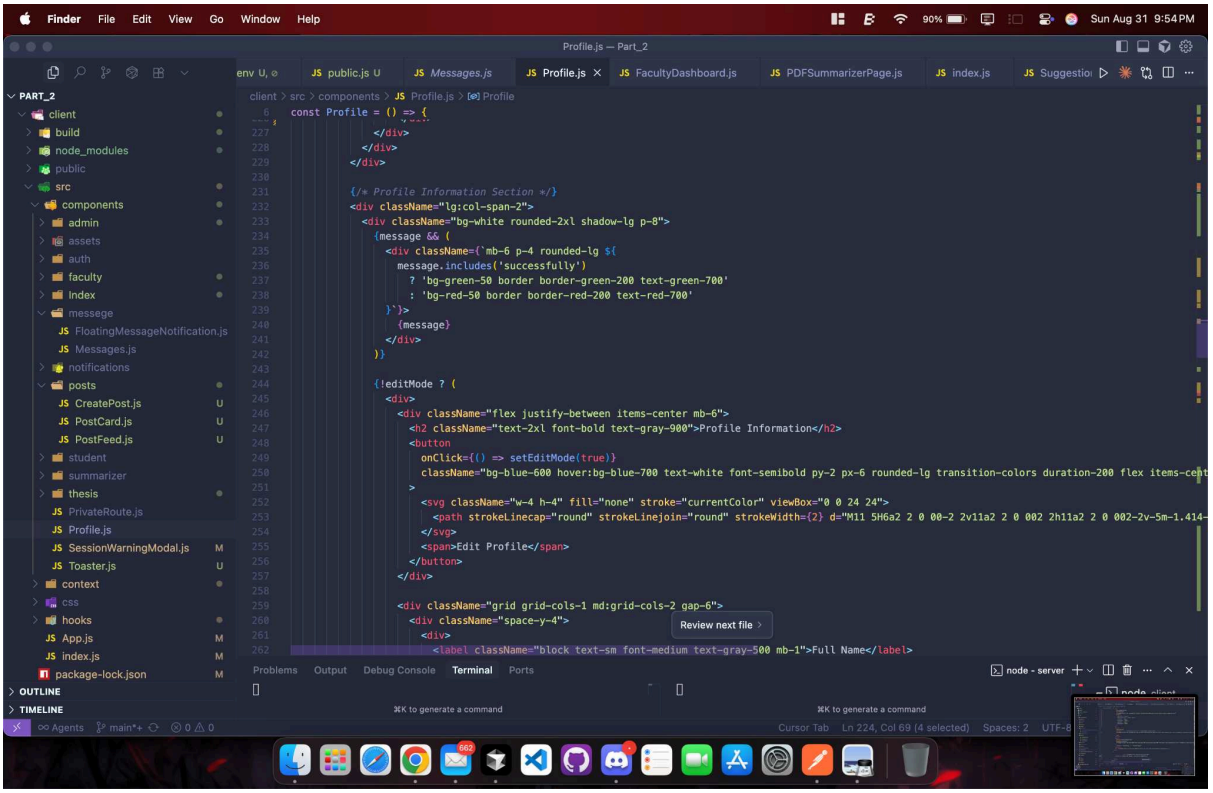
<div className="relative inline-block cursor-pointer group" onClick={() => fileInputRef.current.click()}>
<img
src={imagePreview || (user.profileImage ? (user.profileImage.startsWith('http') ? user.profileImage : `http://localhost:5001/uploads/
alt="Profile"
className="w-48 h-48 rounded-full object-cover object-center border-4 border-blue-100 shadow-lg mx-auto mb-4 transition-transform durat
style={{
objectFit: 'cover',
objectPosition: 'center center',
minWidth: '192px',
minHeight: '192px',
maxWidth: '192px',
maxHeight: '192px'
}}
onLoad={() => {
// console.log('Profile image loaded successfully:', user.profileImage);
}}
onError={(e) => {
// console.log('Profile image failed to load:', user.profileImage);
e.target.src = `https://ui-avatars.com/api/?name=${encodeURIComponent(user.name)}&size=200&background=3B82F6&color=fff&bold=true`;
}}
/>

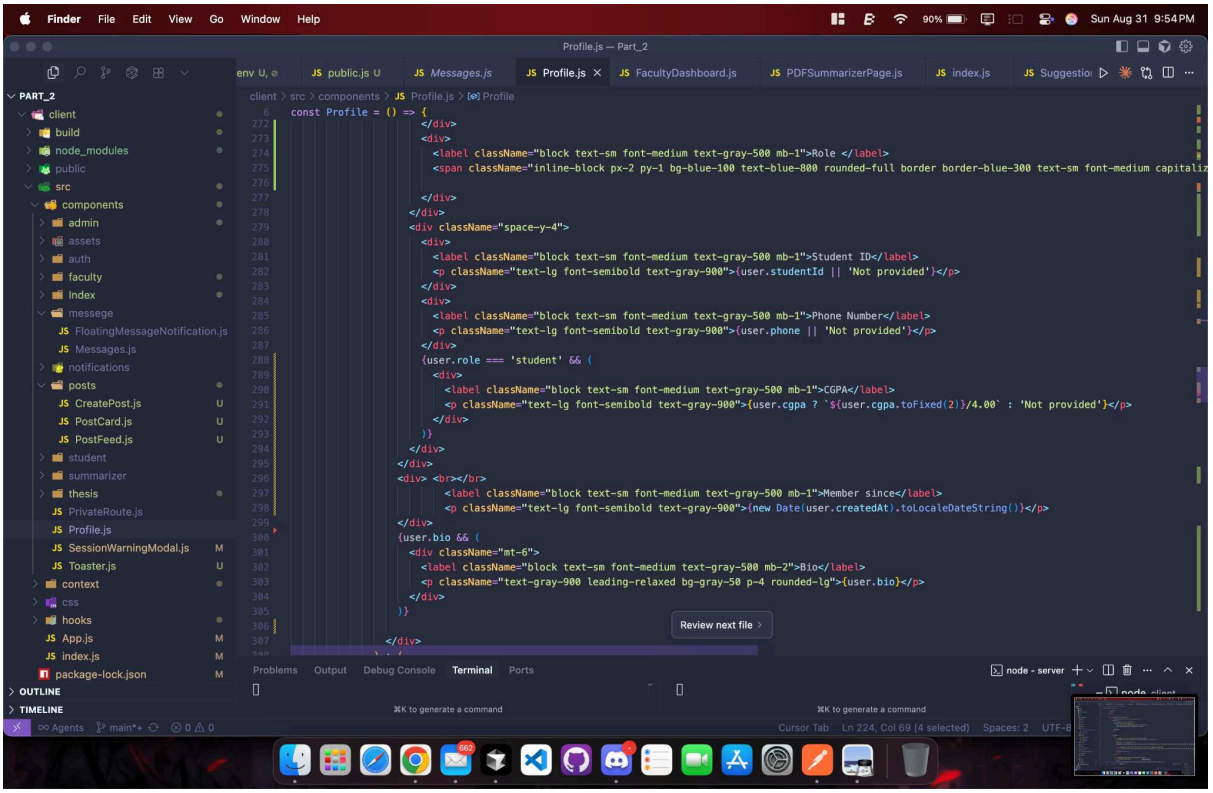
<input
ref={fileInputRef}
type="file"
accept="image/*"
onChange={handleImageChange}
className="hidden"
Review next file
Problems Output Debug Console Terminal Ports
node - server
⌘K to generate a command
⌘K to generate a command
Cursor Tab Ln 224, Col 69 (4 selected) Spaces: 2

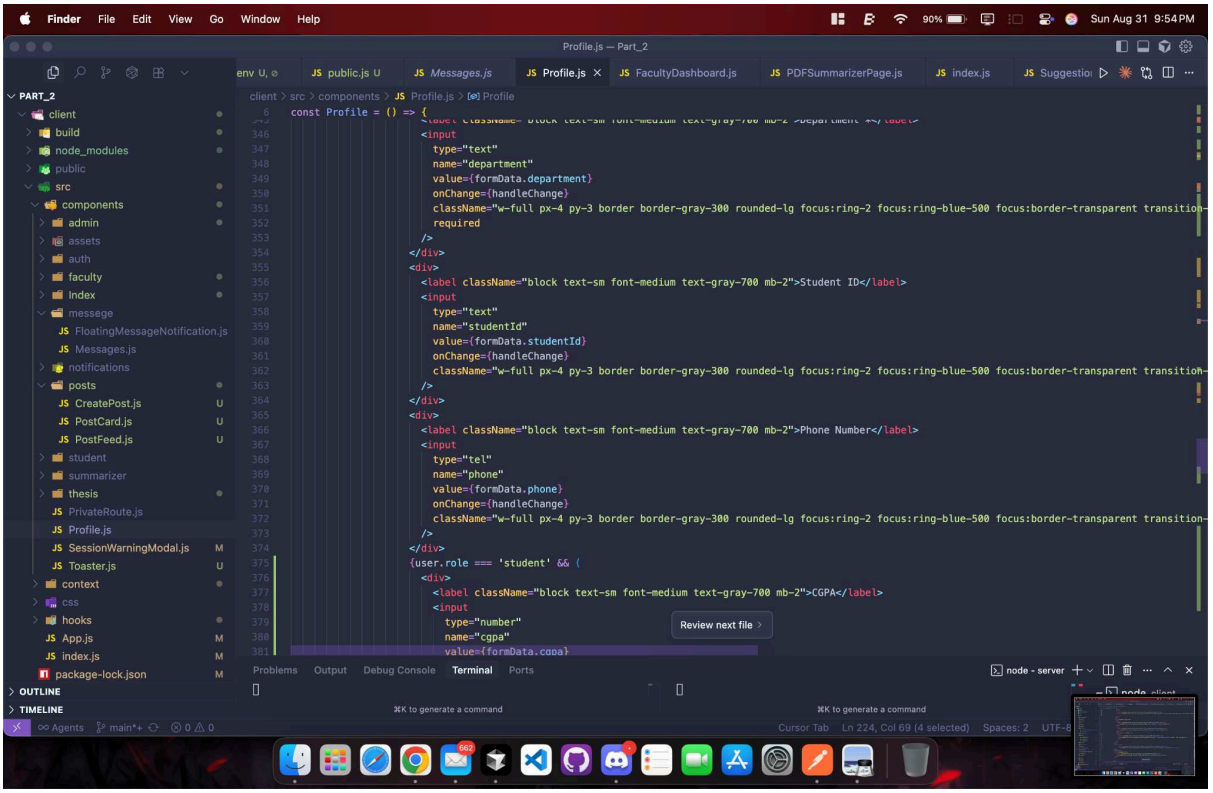

Profile.js — Part_2
PART_2
client > src > components > Profile.js > Profile
const Profile = () => {
src={imagePreview}
alt="Preview"
className="w-32 h-32 rounded-full border-2 border-blue-400 object-cover object-center shadow-lg mb-2"
style={{
objectFit: 'cover',
objectPosition: 'center center',
minWidth: '128px',
minHeight: '128px',
maxWidth: '128px',
maxHeight: '128px'
}}
/>
<button
type="button"
onClick={() => { setImagePreview(null); fileInputRef.current.value = ''; }}
className="w-half bg-red-600 hover:bg-red-700 disabled:bg-red-400 text-white font-semibold py-2 px-4 rounded-lg transition-colors dur
>
Clear
</button>
<button
onClick={handleImageUpload}
disabled={loading}
className="w-half bg-green-600 hover:bg-green-700 disabled:bg-green-400 text-black font-semibold py-2 px-4 rounded-lg transition-colo
>
{loading ? 'Uploading...' : 'Upload Image'}
</button>
You can clear the preview before uploading

)}

Click to choose and upload a new profile picture
Review next file
Problems Output Debug Console Terminal Ports
node - server
⌘K to generate a command
⌘K to generate a command
Cursor Tab Ln 224, Col 69 (4 selected) Spaces: 2

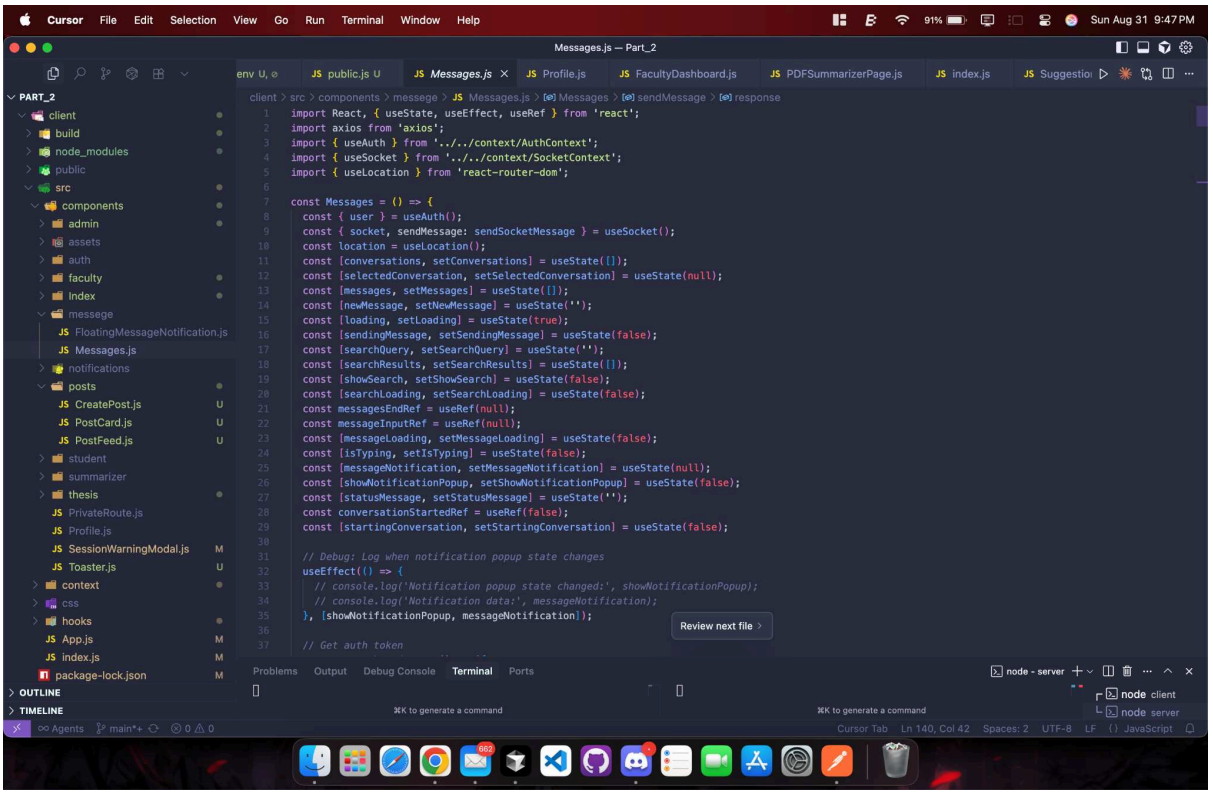

Finder File Edit View Go Window Help
Sun Aug 31 9:54 PM
Profile.js — Part_2
PART_2
client
build
node_modules
public
src
components
admin
assets
auth
faculty
Index
messege
FloatingMessageNotification.js
Messages.js
notifications
posts
CreatePost.js
PostCard.js
PostFeed.js
student
summarizer
thesis
PrivateRoute.js
Profile.js
SessionWarningModal.js
Toaster.js
context
css
hooks
App.js
index.js
package-lock.json
OUTLINE
TIMELINE
public.js U
Messages.js
Profile.js
FacultyDashboard.js
PDFSummarizerPage.js
index.js
client > src > components > Profile.js > Profile
const Profile = () => {
{/* Profile Information Section */}
<h2 className="text-2xl font-bold text-gray-900">Profile Information</h2>
Edit Profile
Review next file
Problems Output Debug Console Terminal Ports
node - server
⌘K to generate a command
Cursor Tab Ln 224, Col 69 (4 selected) Spaces: 2

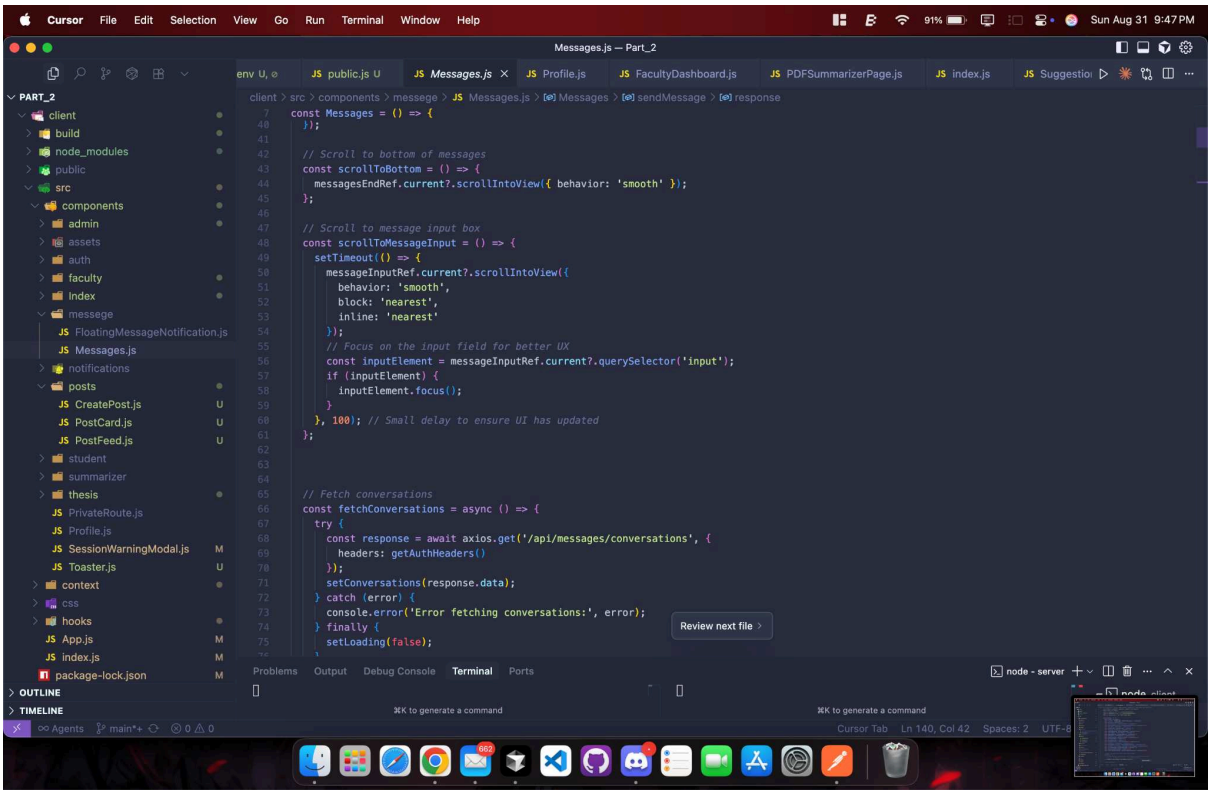


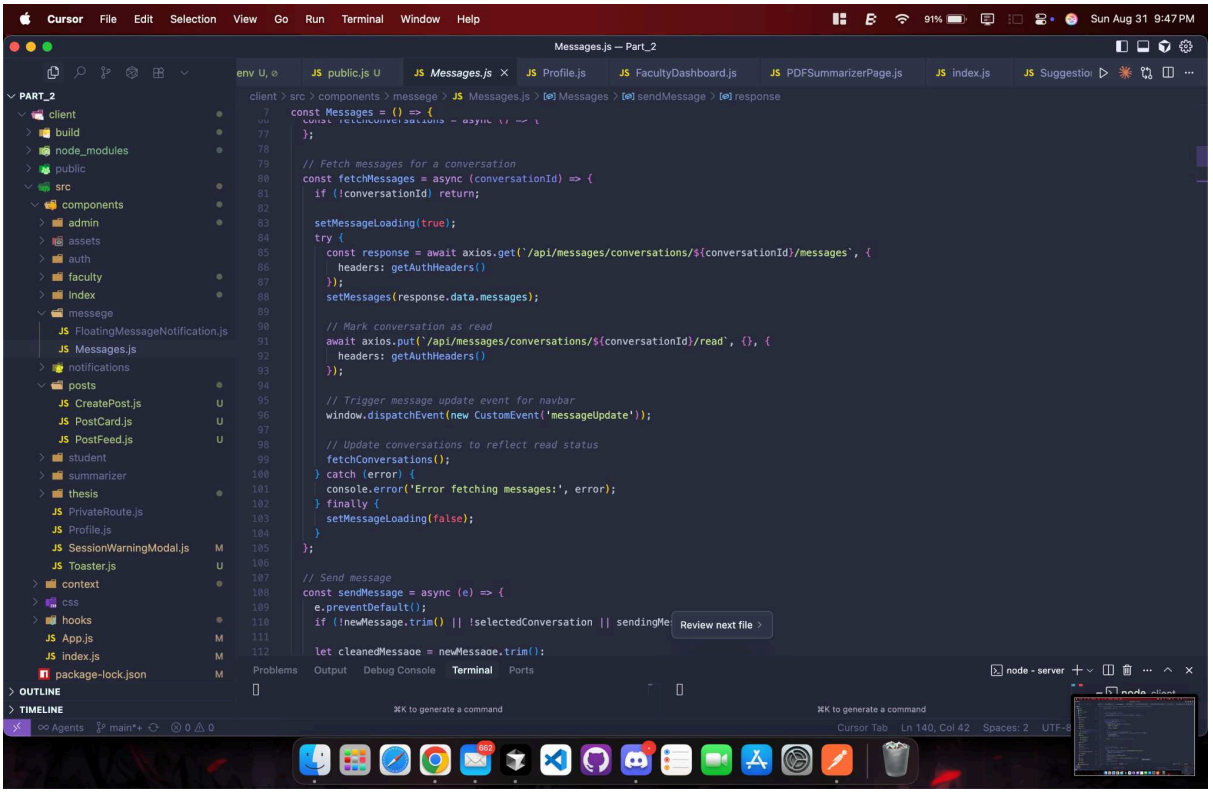


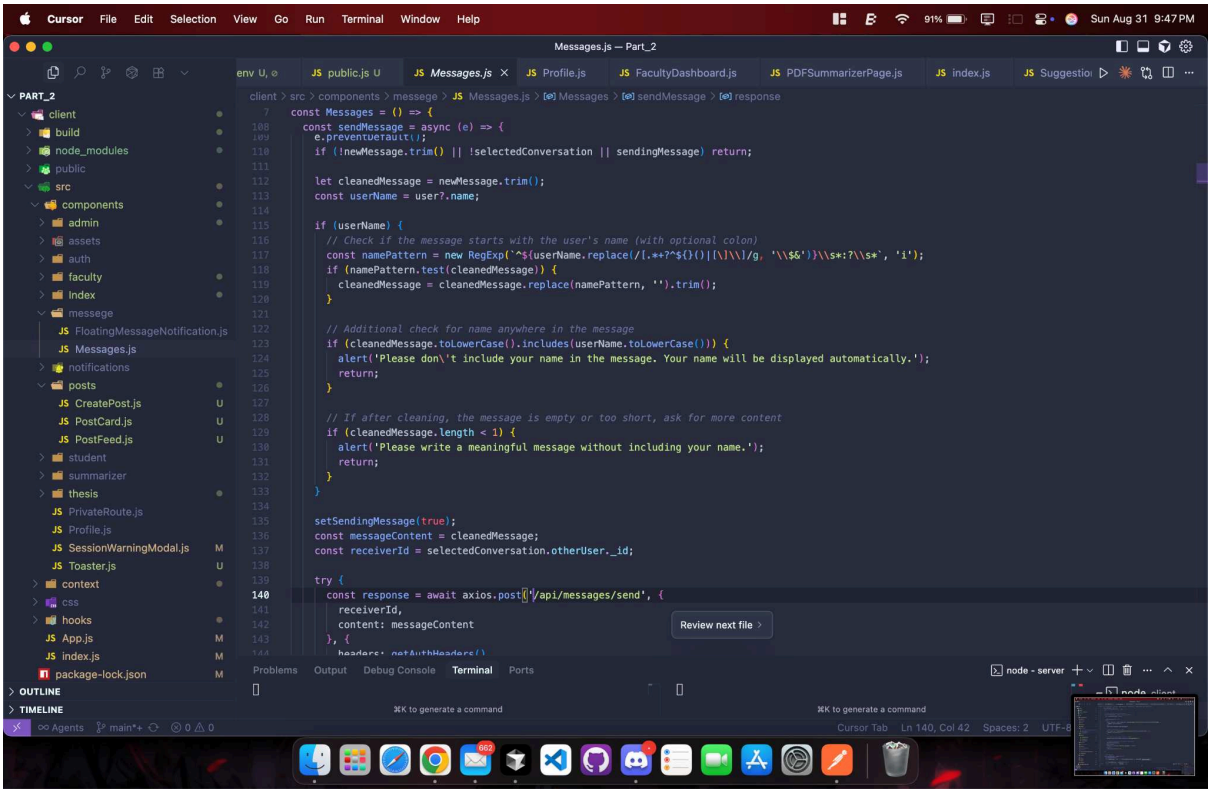
3.Messege: The messaging system works through a real-time chat interface where users can send and receive messages with other users. When a user types a message and sends it, the frontend captures the input, displays it immediately in the chat window, and sends the message data to the backend API. The system uses WebSocket connections (via Socket.io) to provide real-time updates, so when other users send messages, they appear instantly without

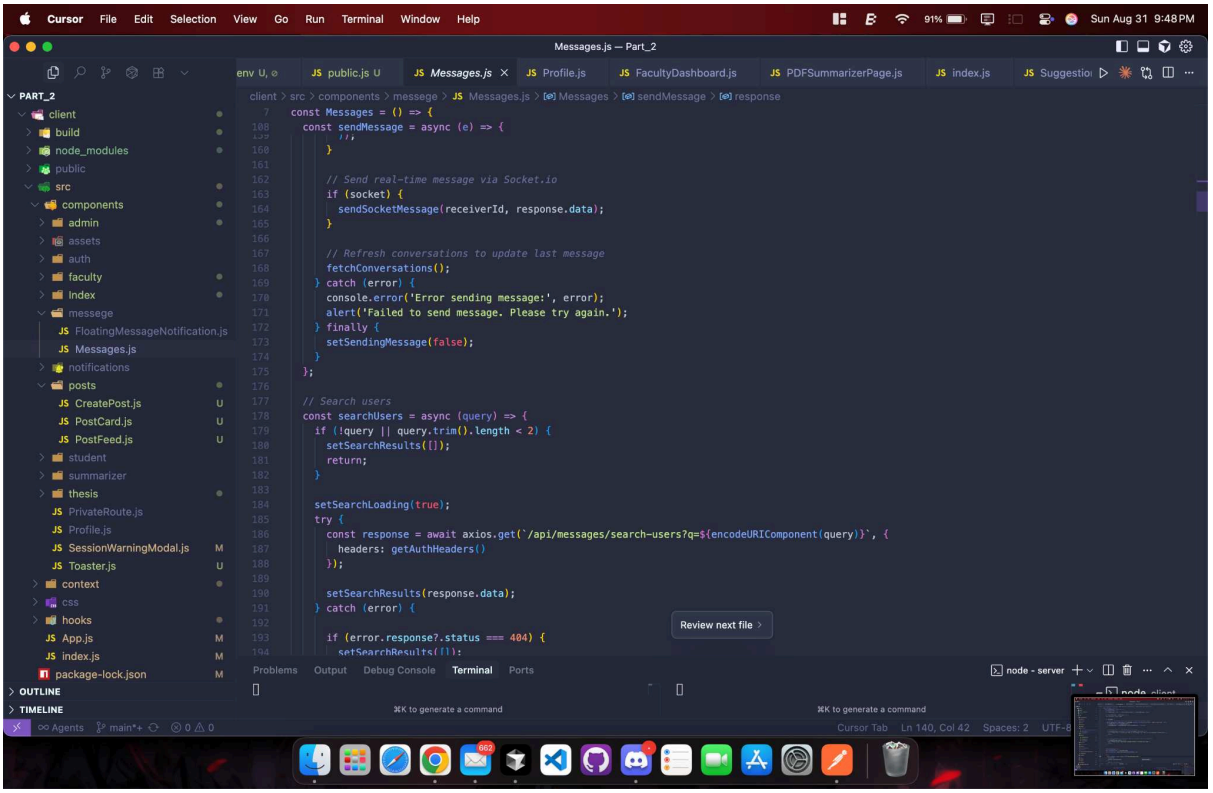
needing to refresh the page. The frontend handles message display with timestamps, user avatars, and different styling for sent vs. received messages. Users can view their message conversations, search other users.

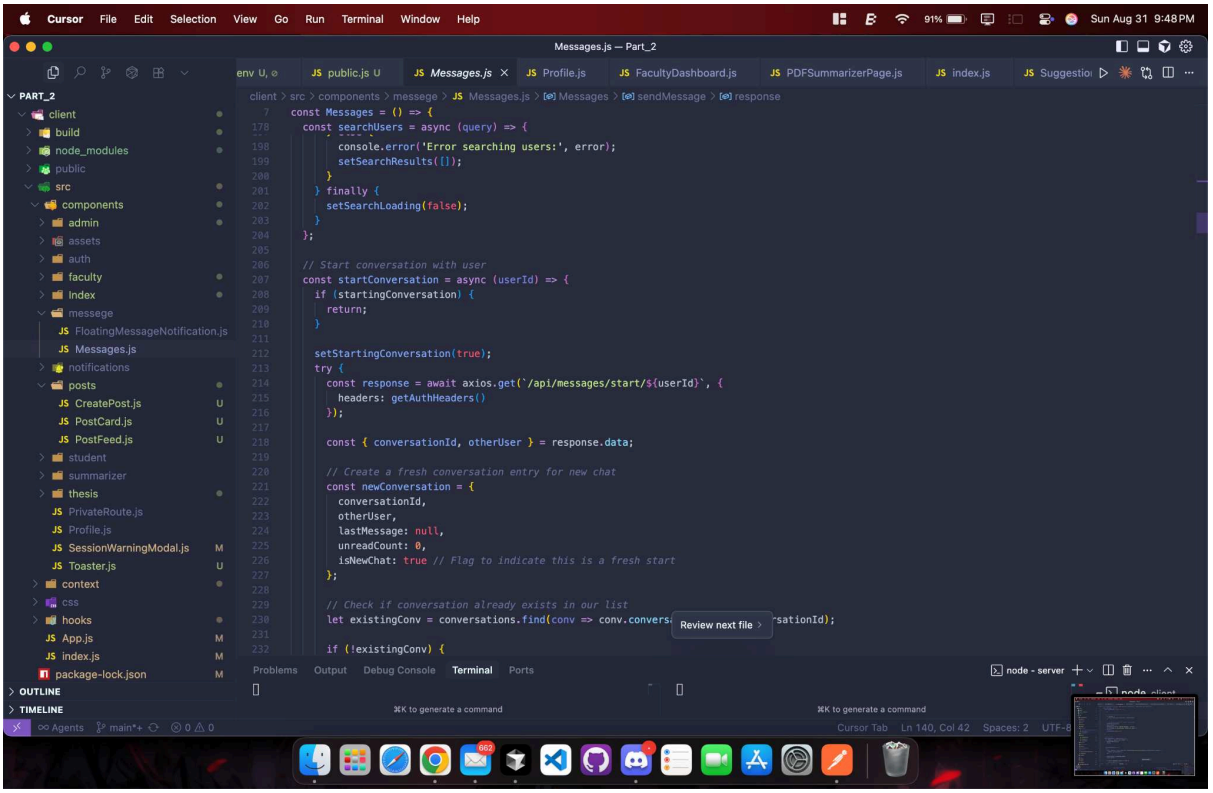


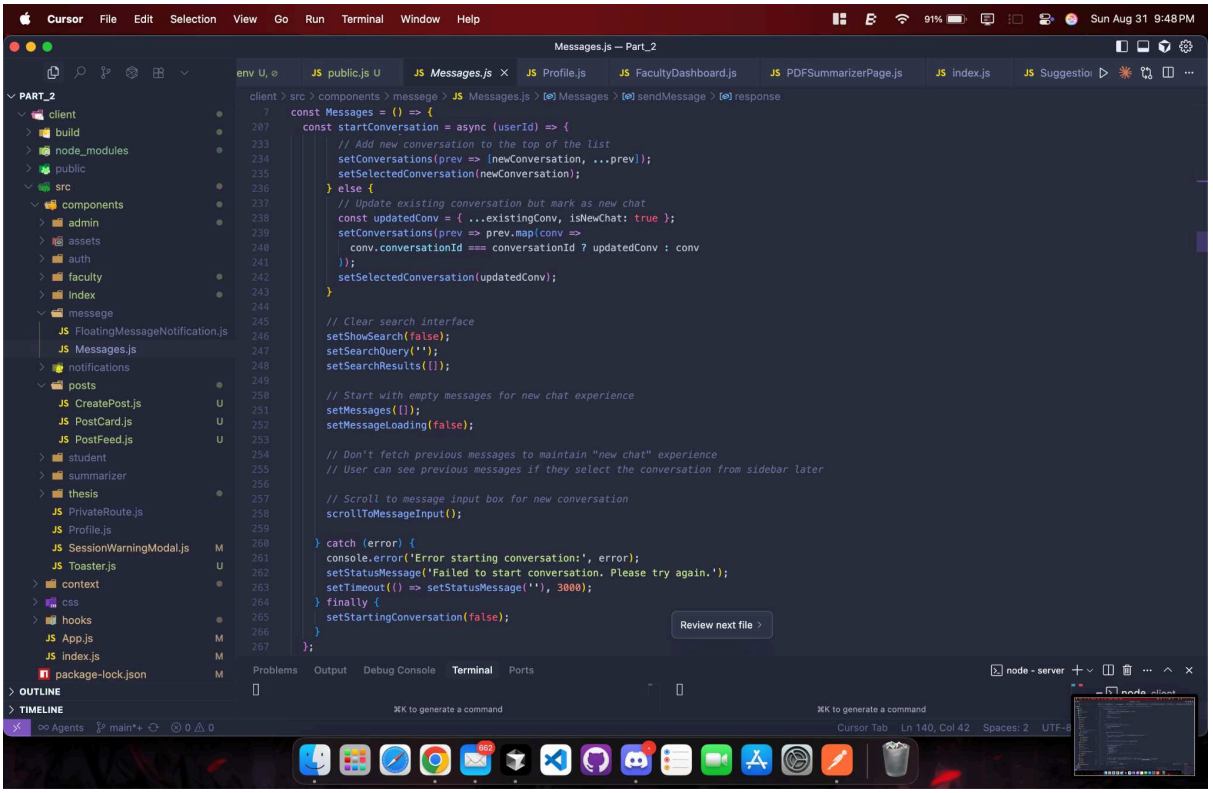


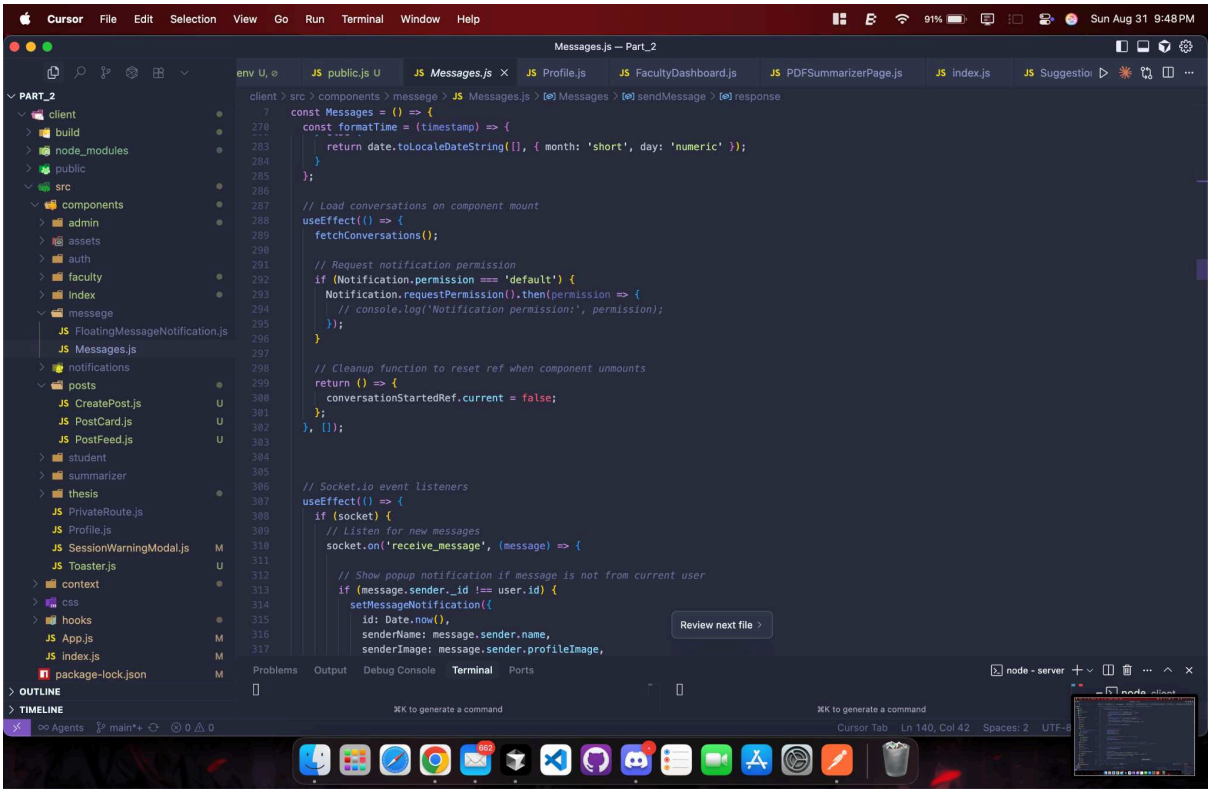

Cursor File Edit Selection View Go Run Terminal Window Help
Sun Aug 31 9:47 PM
Messages.js — Part_2
PART_2
client
build
node_modules
public
src
components
admin
assets
auth
faculty
index
messege
FloatingMessageNotification.js
Messages.js
notifications
posts
CreatePost.js
PostCard.js
PostFeed.js
student
summarizer
thesis
PrivateRoute.js
Profile.js
SessionWarningModal.js
Toaster.js
context
css
hooks
App.js
index.js
package-lock.json
OUTLINE
TIMELINE
const Messages = () => {
};
// Fetch messages for a conversation
const fetchMessages = async (conversationId) => {
if (!conversationId) return;
setMessageLoading(true);
try {
const response = await axios.get(`/api/messages/conversations/${conversationId}/messages`, {
headers: getAuthHeaders()
});
setMessages(response.data.messages);
// Mark conversation as read
await axios.put(`/api/messages/conversations/${conversationId}/read`, {}, {
headers: getAuthHeaders()
});
// Trigger message update event for navbar
window.dispatchEvent(new CustomEvent('messageUpdate'));
// Update conversations to reflect read status
fetchConversations();
} catch (error) {
console.error('Error fetching messages:', error);
} finally {
setMessageLoading(false);
}
};
// Send message
const sendMessage = async (e) => {
e.preventDefault();
if (!newMessage.trim() || !selectedConversation || sendingMe
Review next file
let cleanedMessage = newMessage.trim();
Problems Output Debug Console Terminal Ports
node - server
⌘K to generate a command
Cursor Tab Ln 140, Col 42 Spaces: 2 UTF-8

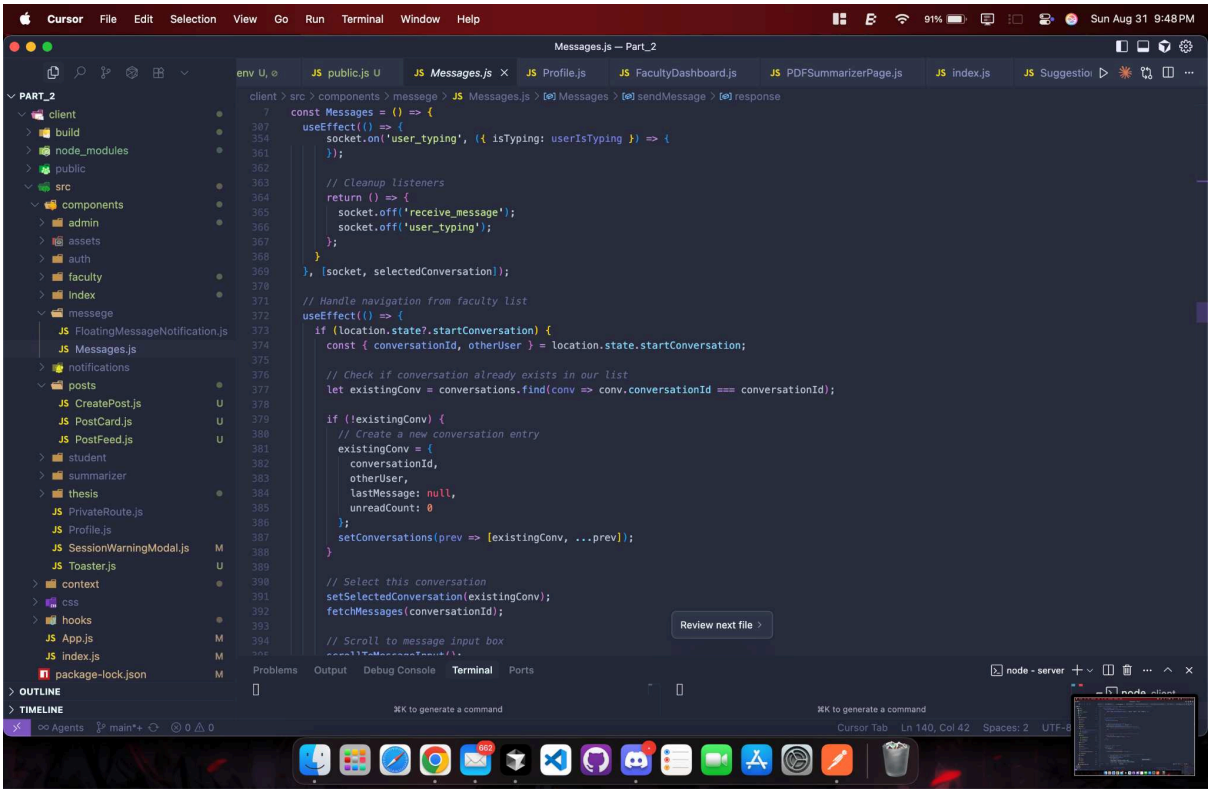

Cursor File Edit Selection View Go Run Terminal Window Help
Sun Aug 31 9:47 PM
Messages.js — Part_2
PART_2
client
build
node_modules
public
src
components
admin
assets
auth
faculty
index
messege
FloatingMessageNotification.js
Messages.js
notifications
posts
CreatePost.js
PostCard.js
PostFeed.js
student
summarizer
thesis
PrivateRoute.js
Profile.js
SessionWarningModal.js
Toaster.js
context
css
hooks
App.js
index.js
package-lock.json
OUTLINE
TIMELINE
const Messages = () => {
const sendMessage = async (e) => {
e.preventDefault();
if (!newMessage.trim() || !selectedConversation || sendingMessage) return;
let cleanedMessage = newMessage.trim();
const userName = user?.name;
if (userName) {
// Check if the message starts with the user's name (with optional colon)
const namePattern = new RegExp(`^${userName.replace(/[.*+?^${}()|[\]\\]/g, '\\$&')}\\s*:?\\s*`, 'i');
if (namePattern.test(cleanedMessage)) {
cleanedMessage = cleanedMessage.replace(namePattern, '').trim();
}
// Additional check for name anywhere in the message
if (cleanedMessage.toLowerCase().includes(userName.toLowerCase())) {
alert('Please don\'t include your name in the message. Your name will be displayed automatically.');
return;
}
// If after cleaning, the message is empty or too short, ask for more content
if (cleanedMessage.length < 1) {
alert('Please write a meaningful message without including your name.');
return;
}
}
setSendingMessage(true);
const messageContent = cleanedMessage;
const receiverId = selectedConversation.otherUser._id;
try {
const response = await axios.post('/api/messages/send', {
receiverId,
content: messageContent
Review next file
}, {
Problems Output Debug Console Terminal Ports
node - server
⌘K to generate a command
Cursor Tab Ln 140, Col 42 Spaces: 2 UTF-8

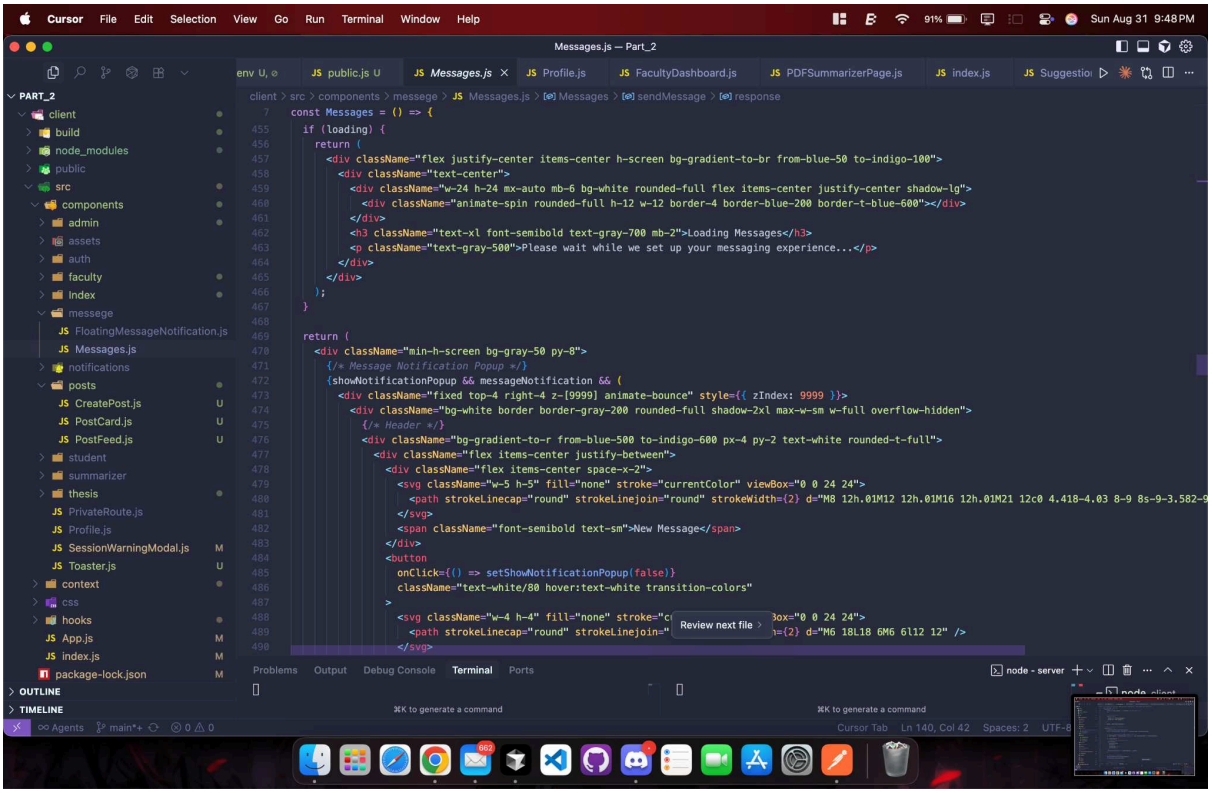

Messages.js — Part_2
PART_2
client
build
node_modules
public
src
components
admin
assets
auth
faculty
Index
messege
FloatingMessageNotification.js
Messages.js
notifications
posts
CreatePost.js
PostCard.js
PostFeed.js
student
summarizer
thesis
PrivateRoute.js
Profile.js
SessionWarningModal.js
Toaster.js
context
css
hooks
App.js
index.js
package-lock.json
OUTLINE
TIMELINE
const Messages = () => {
const sendMessage = async (e) => {
// Send real-time message via Socket.io
if (socket) {
sendSocketMessage(receiverId, response.data);
// Refresh conversations to update last message
fetchConversations();
} catch (error) {
console.error('Error sending message:', error);
alert('Failed to send message. Please try again.');
} finally {
setSendingMessage(false);
// Search users
const searchUsers = async (query) => {
if (!query || query.trim().length < 2) {
setSearchResults([]);
return;
setSearchLoading(true);
try {
const response = await axios.get(`/api/messages/search-users?q=${encodeURIComponent(query)}`, {
headers: getAuthHeaders()
setSearchResults(response.data);
} catch (error) {
if (error.response?.status === 404) {
Review next file
Problems
Output
Debug Console
Terminal
Ports
node - server
⌘K to generate a command
Cursor Tab
Ln 140, Col 42
Spaces: 2

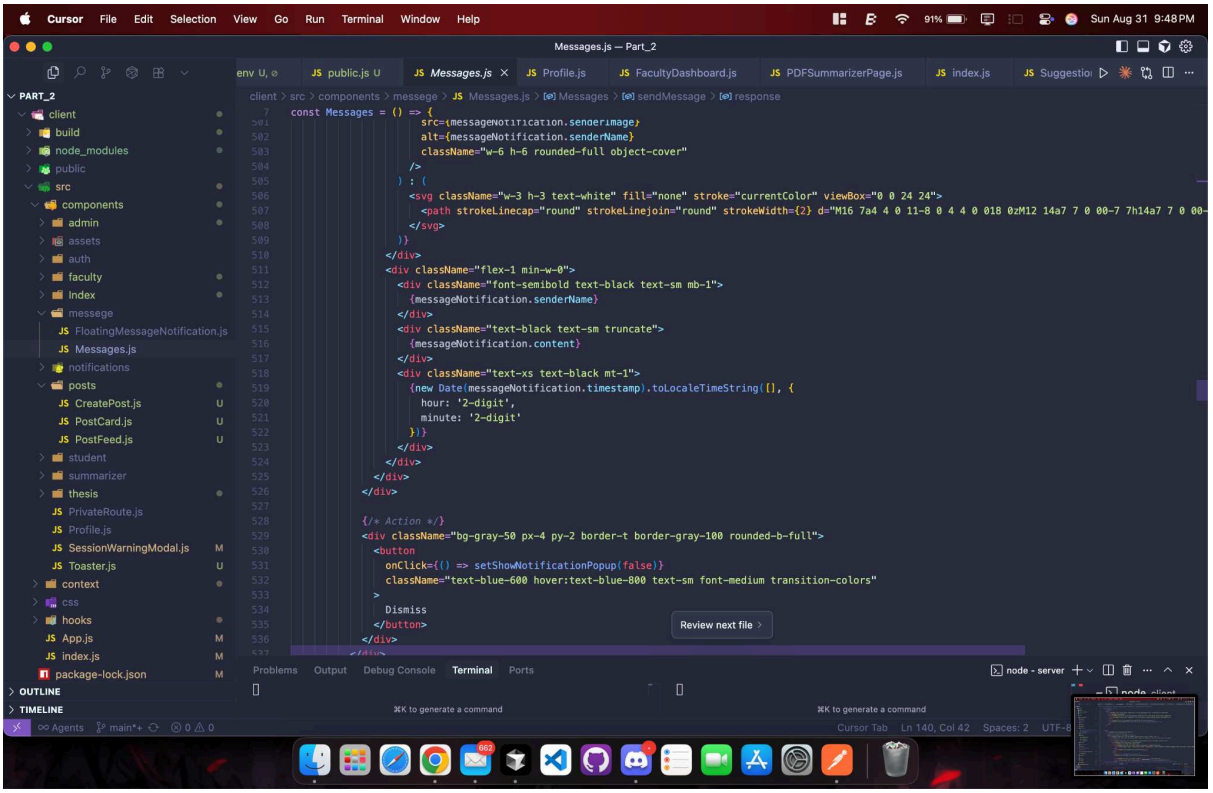

Messages.js — Part_2
PART_2
client
build
node_modules
public
src
components
admin
assets
auth
faculty
Index
messege
FloatingMessageNotification.js
Messages.js
notifications
posts
CreatePost.js
PostCard.js
PostFeed.js
student
summarizer
thesis
PrivateRoute.js
Profile.js
SessionWarningModal.js
Toaster.js
context
css
hooks
App.js
index.js
package-lock.json
OUTLINE
TIMELINE
const Messages = () => {
const searchUsers = async (query) => {
console.error('Error searching users:', error);
setSearchResults([]);
} finally {
setSearchLoading(false);
// Start conversation with user
const startConversation = async (userId) => {
if (startingConversation) {
return;
setStartingConversation(true);
try {
const response = await axios.get(`/api/messages/start/${userId}`, {
headers: getAuthHeaders()
const { conversationId, otherUser } = response.data;
// Create a fresh conversation entry for new chat
const newConversation = {
conversationId,
otherUser,
lastMessage: null,
unreadCount: 0,
isNewChat: true // Flag to indicate this is a fresh start
// Check if conversation already exists in our list
let existingConv = conversations.find(conv => conv.conversa
sationId);
if (!existingConv) {
Review next file
Problems
Output
Debug Console
Terminal
Ports
node - server
⌘K to generate a command
Cursor Tab
Ln 140, Col 42
Spaces: 2


Messages.js — Part_2
PART_2
client
build
node_modules
public
src
components
admin
assets
auth
faculty
Index
messege
FloatingMessageNotification.js
Messages.js
notifications
posts
CreatePost.js
PostCard.js
PostFeed.js
student
summarizer
thesis
PrivateRoute.js
Profile.js
SessionWarningModal.js
Toaster.js
context
css
hooks
App.js
index.js
package-lock.json
OUTLINE
TIMELINE
const Messages = () => {
const startConversation = async (userId) => {
// Add new conversation to the top of the list
setConversations(prev => [newConversation, ...prev]);
setSelectedConversation(newConversation);
} else {
// Update existing conversation but mark as new chat
const updatedConv = { ...existingConv, isNewChat: true };
setConversations(prev => prev.map(conv =>
conv.conversationId === conversationId ? updatedConv : conv
));
setSelectedConversation(updatedConv);
}
// Clear search interface
setShowSearch(false);
setSearchQuery('');
setSearchResults([]);
// Start with empty messages for new chat experience
setMessages([]);
setMessageLoading(false);
// Don't fetch previous messages to maintain "new chat" experience
// User can see previous messages if they select the conversation from sidebar later
// Scroll to message input box for new conversation
scrollToMessageInput();
} catch (error) {
console.error('Error starting conversation:', error);
setStatusMessage('Failed to start conversation. Please try again.');
setTimeout(() => setStatusMessage(''), 3000);
} finally {
setStartingConversation(false);
}
};
Review next file
Problems
Output
Debug Console
Terminal
Ports
node - server
⌘K to generate a command
Cursor Tab
Ln 140, Col 42
Spaces: 2
UTF-8


Messages.js — Part_2
PART_2
client
build
node_modules
public
src
components
admin
assets
auth
faculty
Index
messege
FloatingMessageNotification.js
Messages.js
notifications
posts
CreatePost.js
PostCard.js
PostFeed.js
student
summarizer
thesis
PrivateRoute.js
Profile.js
SessionWarningModal.js
Toaster.js
context
css
hooks
App.js
index.js
package-lock.json
OUTLINE
TIMELINE
const Messages = () => {
const formatTime = (timestamp) => {
return date.toLocaleDateString([], { month: 'short', day: 'numeric' });
}
};
// Load conversations on component mount
useEffect(() => {
fetchConversations();
// Request notification permission
if (Notification.permission === 'default') {
Notification.requestPermission().then(permission => {
// console.log('Notification permission:', permission);
});
}
// Cleanup function to reset ref when component unmounts
return () => {
conversationStartedRef.current = false;
};
}, []);
// Socket.io event listeners
useEffect(() => {
if (socket) {
// Listen for new messages
socket.on('receive_message', (message) => {
// Show popup notification if message is not from current user
if (message.sender._id !== user.id) {
setMessageNotification({
id: Date.now(),
senderName: message.sender.name,
senderImage: message.sender.profileImage,
Review next file
Problems
Output
Debug Console
Terminal
Ports
node - server
⌘K to generate a command
Cursor Tab
Ln 140, Col 42
Spaces: 2
UTF-8


Messages.js — Part_2
Review next file


Messages.js — Part_2
Review next file


Cursor File Edit Selection View Go Run Terminal Window Help
Sun Aug 31 9:48 PM
Messages.js — Part_2
PART_2
client
build
node_modules
public
src
components
admin
assets
auth
faculty
Index
messege
FloatingMessageNotification.js
Messages.js
notifications
posts
CreatePost.js
PostCard.js
PostFeed.js
student
summarizer
thesis
PrivateRoute.js
Profile.js
SessionWarningModal.js
Toaster.js
context
css
hooks
App.js
index.js
package-lock.json
OUTLINE
TIMELINE
const Messages = () => {
src={messageNotification.senderImage}
alt={messageNotification.senderName}
className="w-6 h-6 rounded-full object-cover"
/>
) : (
<svg className="w-3 h-3 text-white" fill="none" stroke="currentColor" viewBox="0 0 24 24">
<path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M16 7a4 4 0 11-8 0 4 4 0 018 0zM12 14a7 7 0 00-7 7h14a7 7 0 00-
</svg>
)}

{messageNotification.senderName}

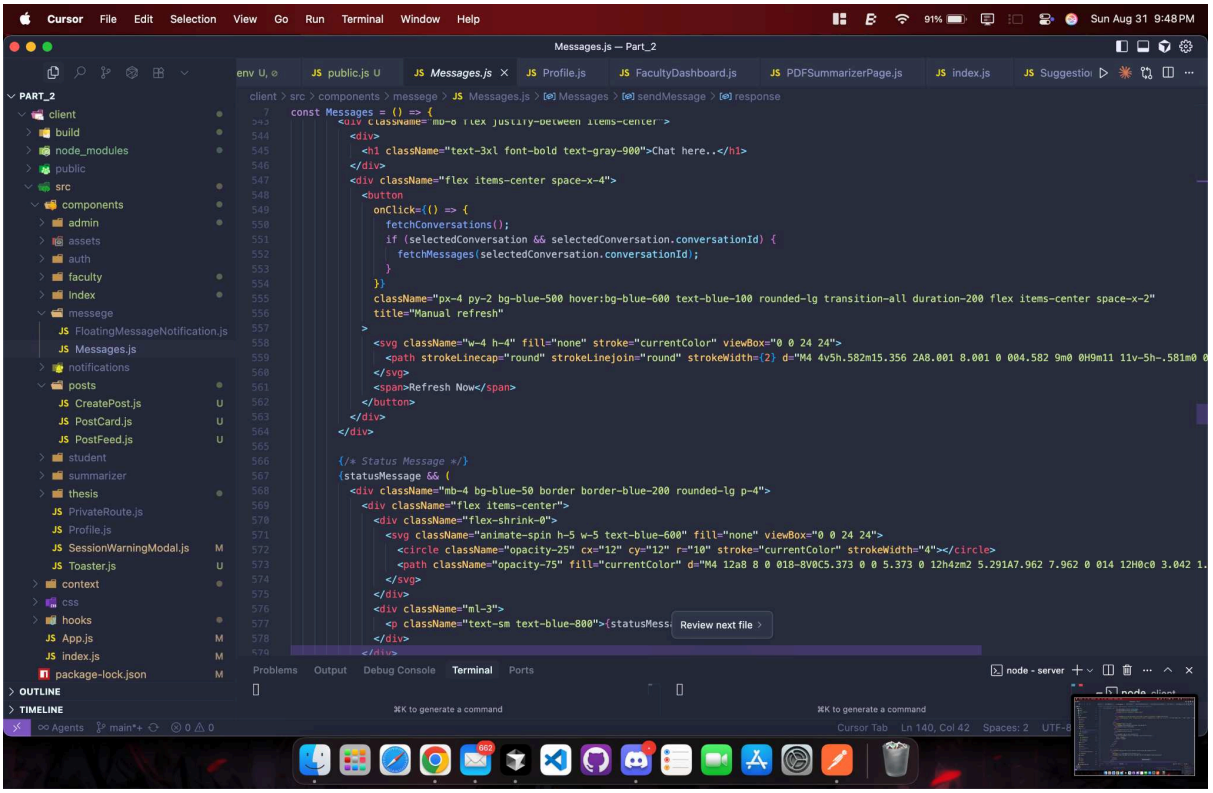
{messageNotification.content}

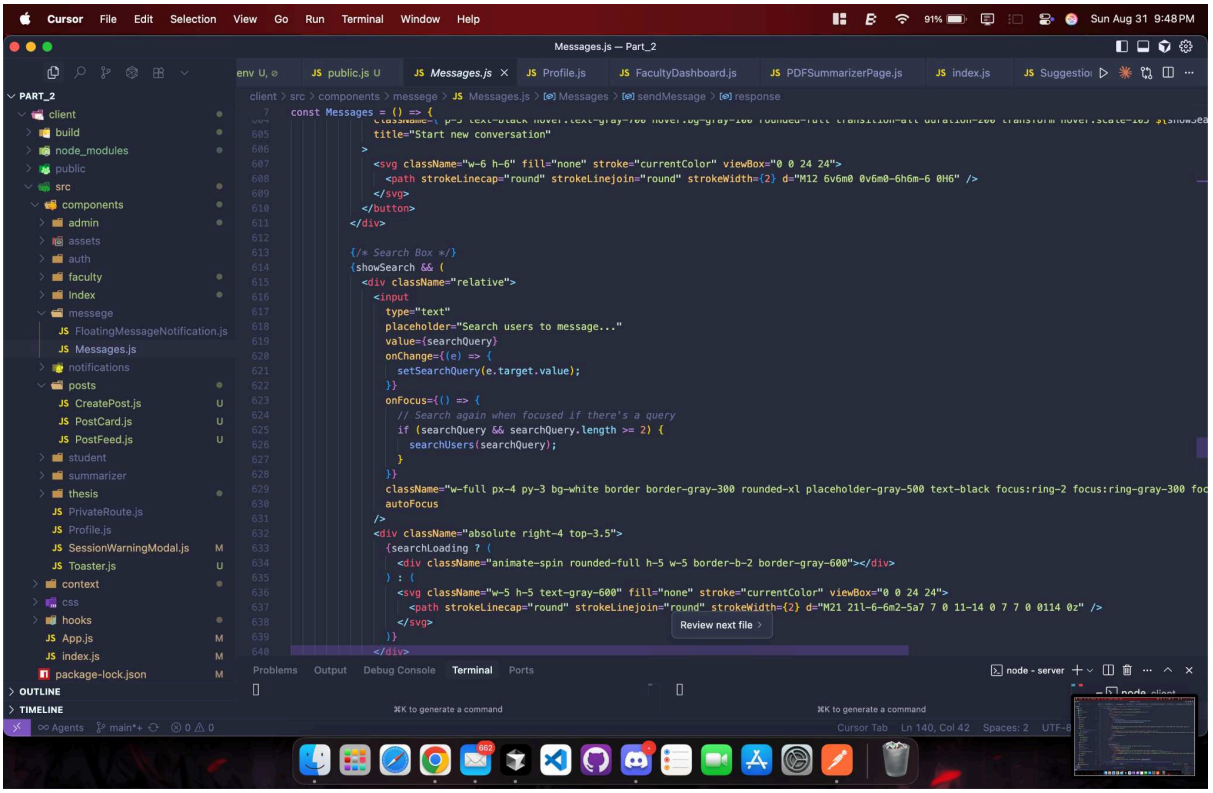
{new Date(messageNotification.timestamp).toLocaleTimeString([], {
hour: '2-digit',
minute: '2-digit'
})}

{/* Action */}

<button
onClick={() => setShowNotificationPopup(false)}
className="text-blue-600 hover:text-blue-800 text-sm font-medium transition-colors"
>
Dismiss
</button>

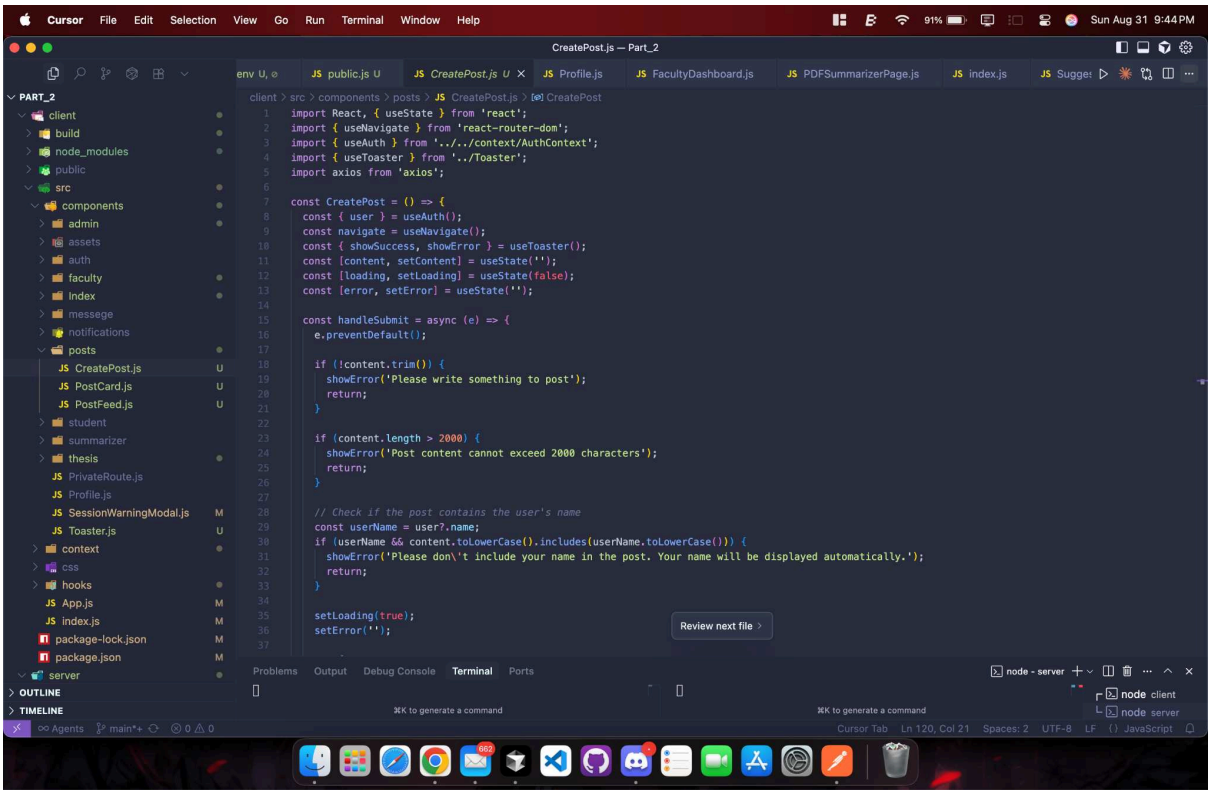
Review next file
Problems Output Debug Console Terminal Ports
node - server
⌘K to generate a command
⌘K to generate a command
Agents main*+ 0 0
Cursor Tab Ln 140, Col 42 Spaces: 2 UTF-8

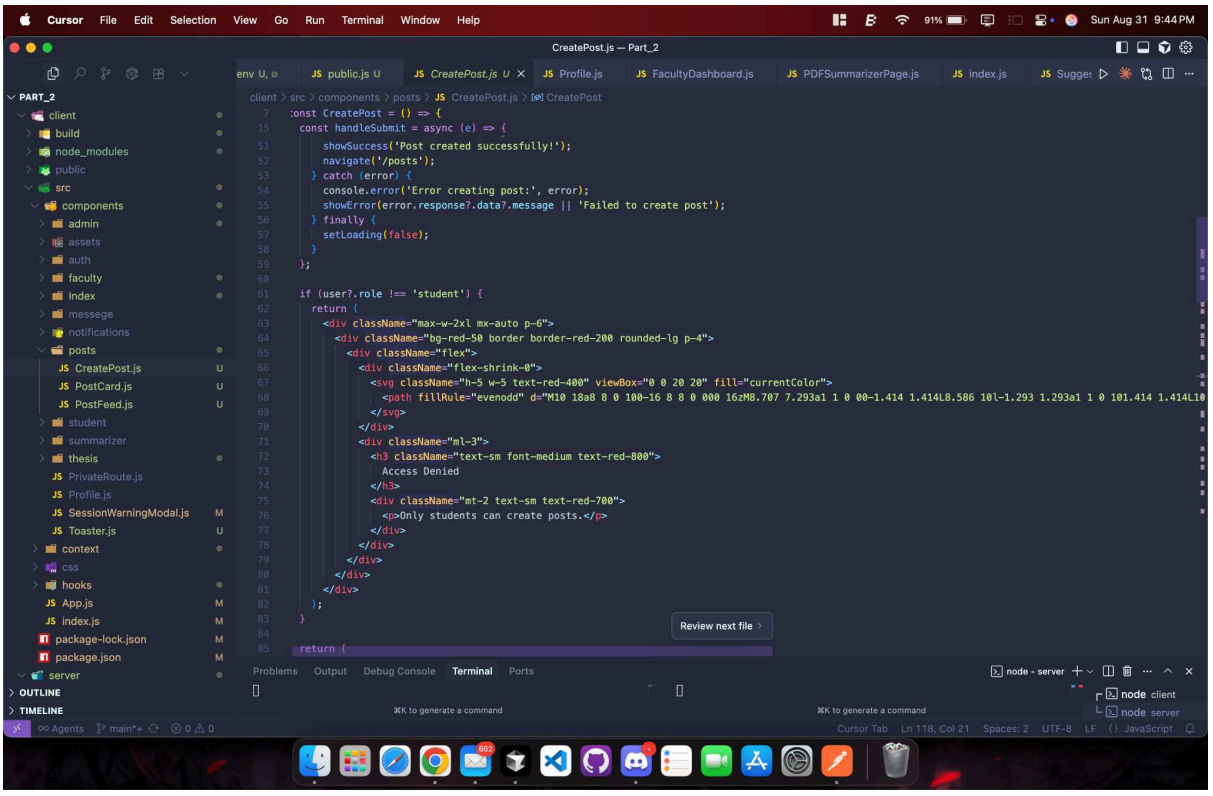


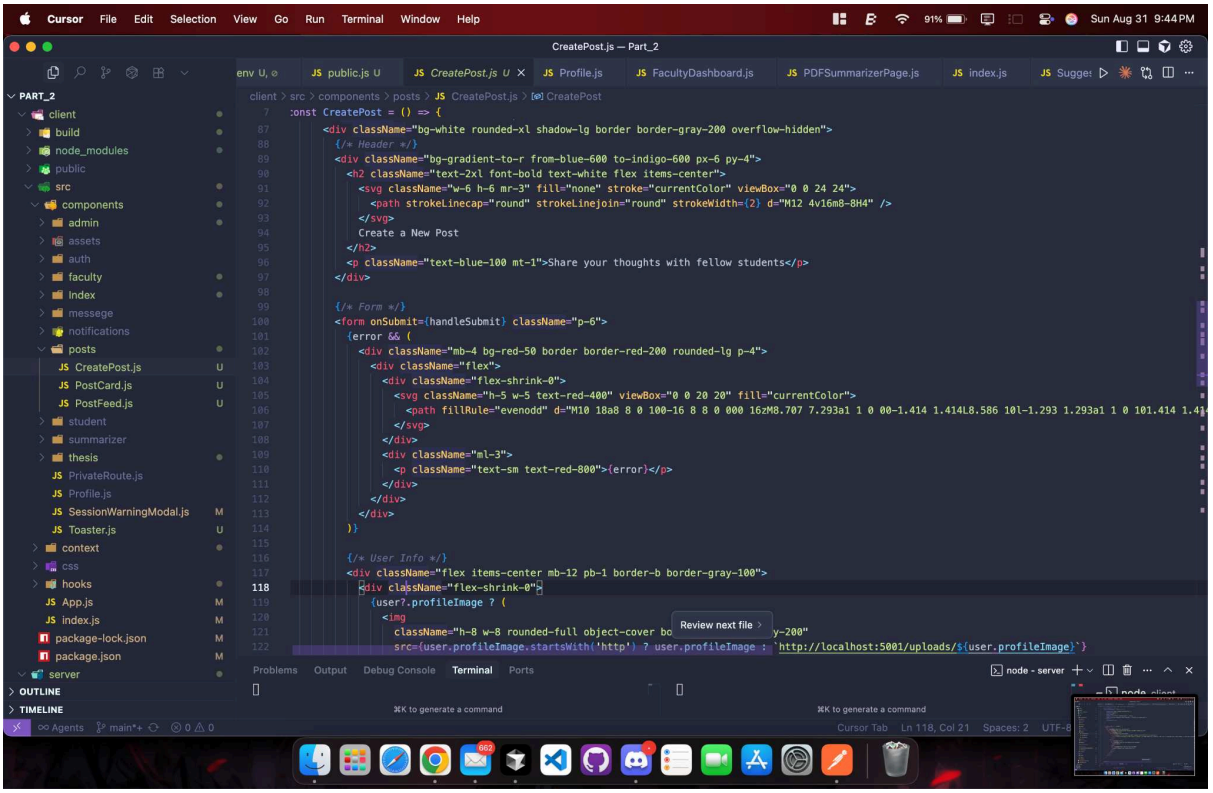


4.Post: The post works through a form-based interface where users can input text content and optionally attach images. When a user types in the post content area and clicks the submit button, the frontend captures the form data, validates the input, and then sends a POST request to the backend API endpoint (typically `/api/posts`) with the post data. The frontend handles the user experience by showing loading states during submission, displaying

success/error messages via the toaster system, and automatically refreshing the post feed to show the newly created post. The component also manages image previews, character limits, and form validation to ensure users can only submit valid posts with proper content.






Cursor File Edit Selection View Go Run Terminal Window Help
Sun Aug 31 9:44 PM
CreatePost.js — Part_2
PART_2
client
build
node_modules
public
src
components
admin
assets
auth
faculty
Index
messege
notifications
posts
CreatePost.js
PostCard.js
PostFeed.js
student
summarizer
thesis
PrivateRoute.js
Profile.js
SessionWarningModal.js
Toaster.js
context
css
hooks
App.js
index.js
package-lock.json
package.json
server
OUTLINE
TIMELINE
public.js
CreatePost.js
Profile.js
FacultyDashboard.js
PDFSummarizerPage.js
index.js
client > src > components > posts > CreatePost.js > CreatePost
const CreatePost = () => {

{/* Header */}

<h2 className="text-2xl font-bold text-white flex items-center">
<svg className="w-6 h-6 mr-3" fill="none" stroke="currentColor" viewBox="0 0 24 24">
<path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M12 4v16m8-8H4" />
</svg>
Create a New Post
</h2>
Share your thoughts with fellow students

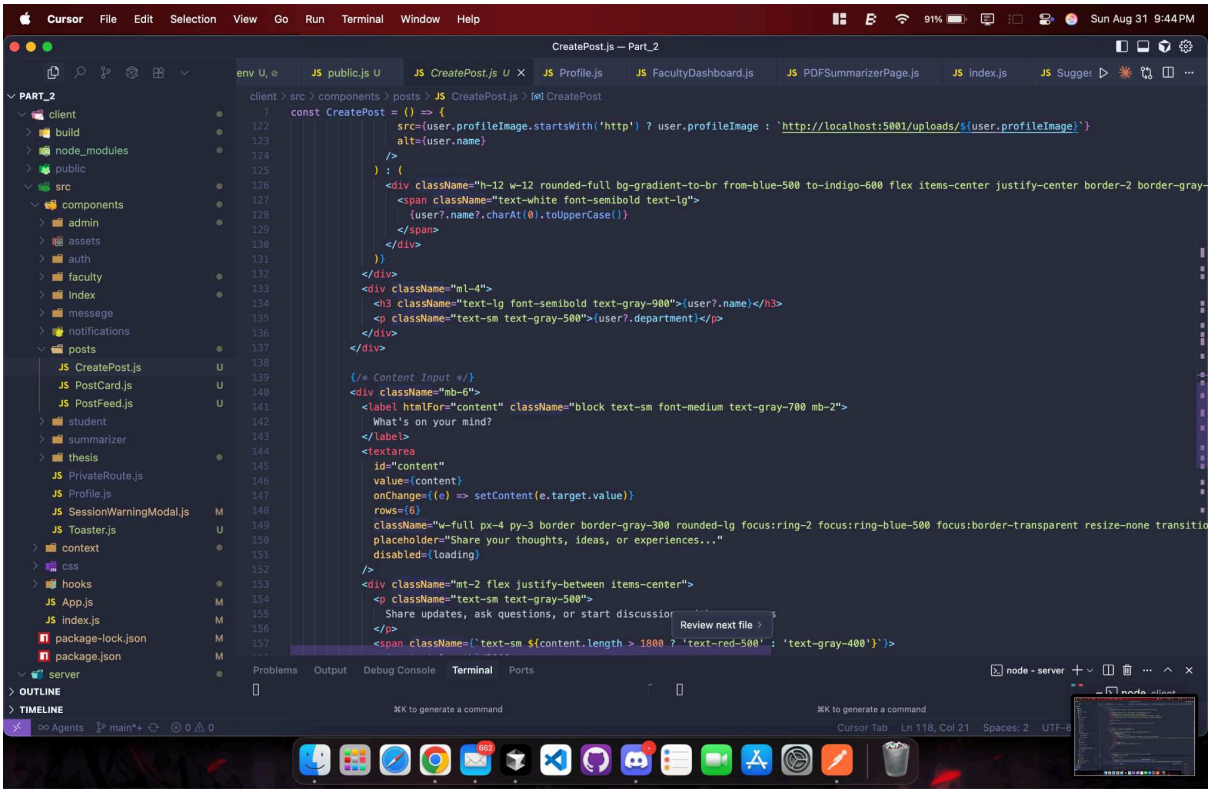
{/* Form */}
<form onSubmit={handleSubmit} className="p-6">
{error && (

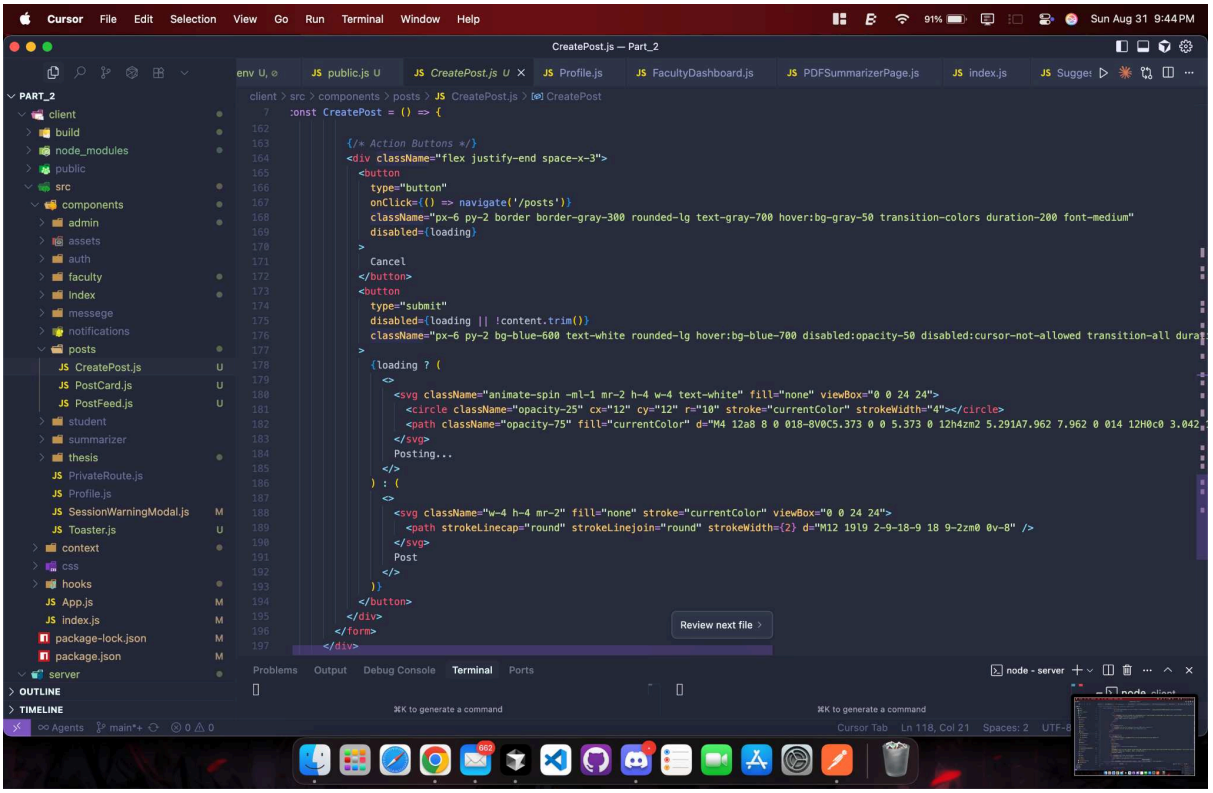
<svg className="h-5 w-5 text-red-400" viewBox="0 0 20 20" fill="currentColor">
<path fillRule="evenodd" d="M10 18a8 8 0 100-16 8 8 0 000 16zM8.707 7.293a1 1 0 00-1.414 1.414L8.586 10l-1.293 1.293a1 1 0 101.414 1.414
</svg>

{error}

)}
{/* User Info */}

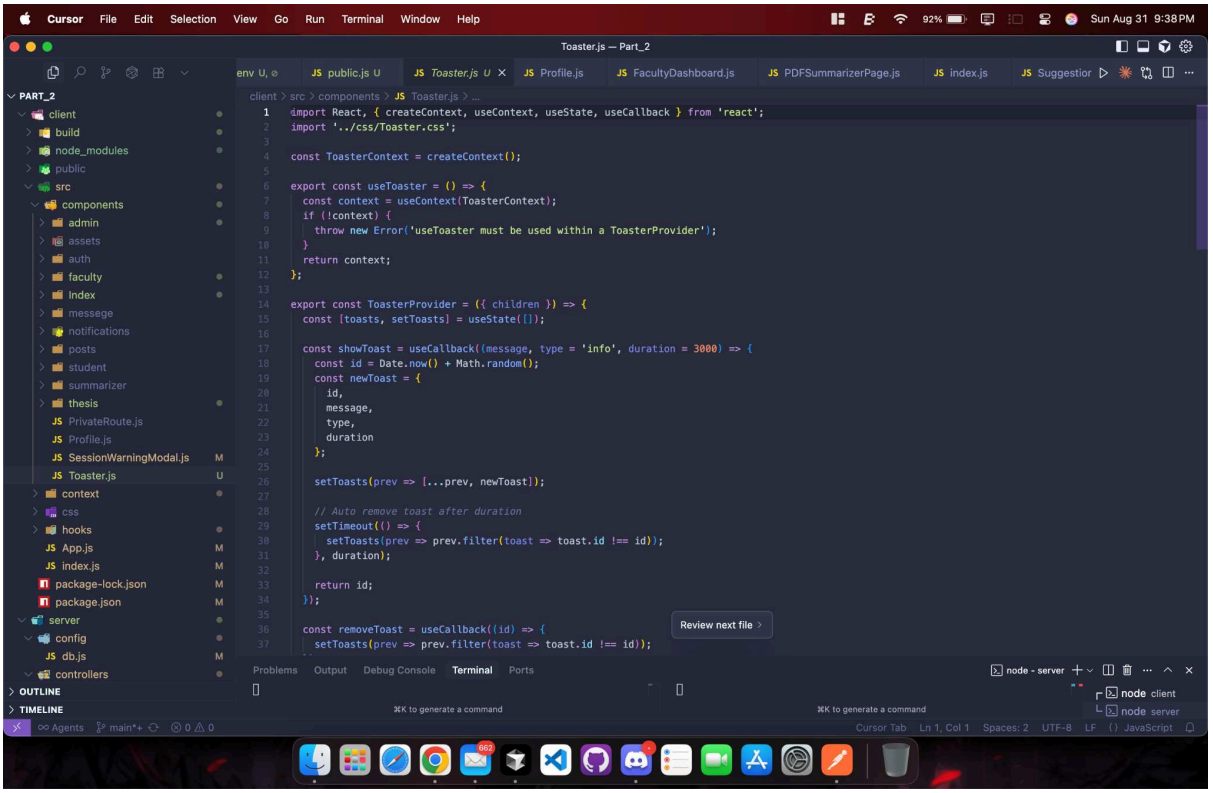
{user?.profileImage ? (
<img
className="h-8 w-8 rounded-full object-cover bo
y-200"
src={user.profileImage.startsWith('http') ? user.profileImage : `http://localhost:5001/uploads/${user.profileImage}`}
Review next file
Problems Output Debug Console Terminal Ports
node - server
⌘K to generate a command
⌘K to generate a command
Agents main*+ 0 0
Cursor Tab Ln 118, Col 21 Spaces: 2 UTF-8

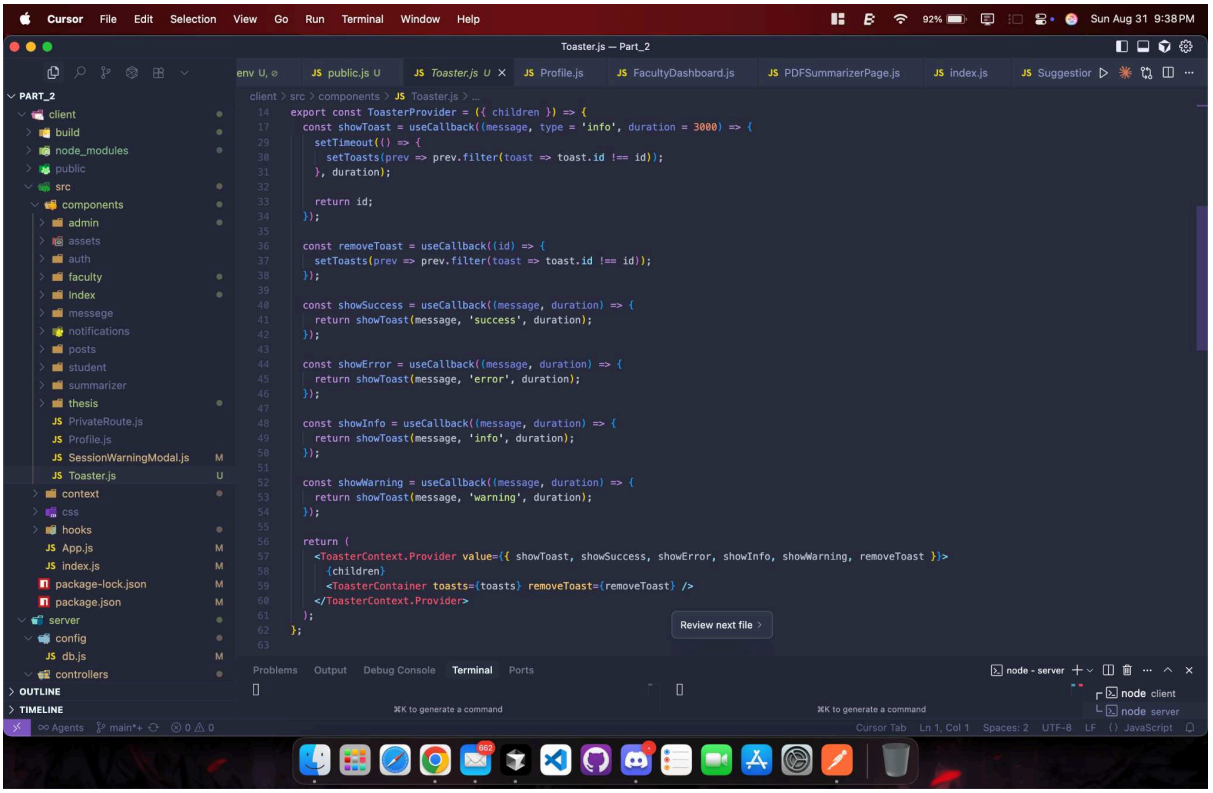


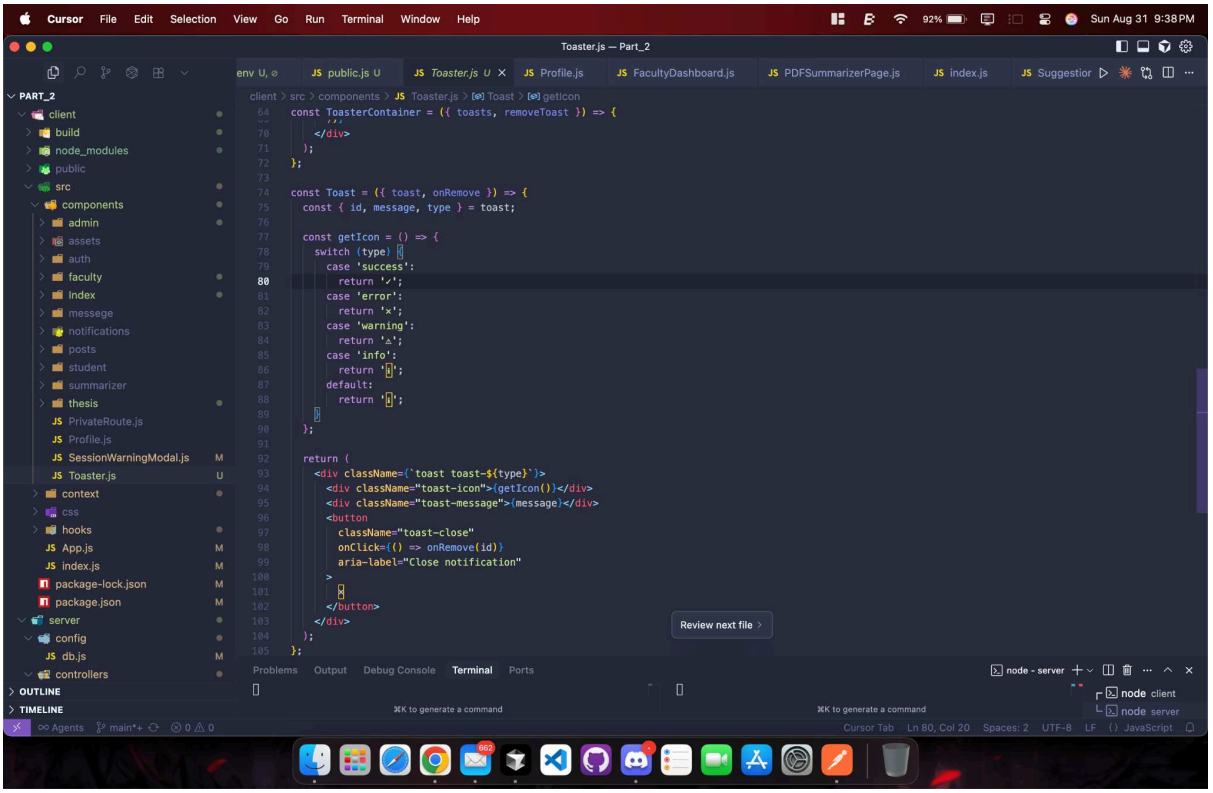


5.Notification: The toaster component is used here to provide user feedback and notifications throughout the application. It serves as a centralized notification system that displays temporary messages (like success confirmations, error alerts, or informational updates) to users in a non-intrusive way. When users perform actions such as logging in/out, submitting forms, or when system events occur, the toaster automatically shows relevant messages at the

top or corner of the screen, ensuring users are always informed about the status of their actions without disrupting their workflow or requiring them to navigate to separate notification pages.


Cursor File Edit Selection View Go Run Terminal Window Help
Sun Aug 31 9:38 PM
Toaster.js — Part_2
PART_2
client
build
node_modules
public
src
components
admin
assets
auth
faculty
index
messege
notifications
posts
student
summarizer
thesis
PrivateRoute.js
Profile.js
SessionWarningModal.js
Toaster.js
context
css
hooks
App.js
index.js
package-lock.json
package.json
server
config
db.js
controllers
OUTLINE
TIMELINE
import React, { createContext, useContext, useState, useCallback } from 'react';
import '../css/Toaster.css';
const ToasterContext = createContext();
export const useToaster = () => {
const context = useContext(ToasterContext);
if (!context) {
throw new Error('useToaster must be used within a ToasterProvider');
}
return context;
};
export const ToasterProvider = ({ children }) => {
const [toasts, setToasts] = useState([]);
const showToast = useCallback((message, type = 'info', duration = 3000) => {
const id = Date.now() + Math.random();
const newToast = {
id,
message,
type,
duration
};
setToasts(prev => [...prev, newToast]);
// Auto remove toast after duration
setTimeout(() => {
setToasts(prev => prev.filter(toast => toast.id !== id));
}, duration);
return id;
});
const removeToast = useCallback((id) => {
setToasts(prev => prev.filter(toast => toast.id !== id));
Review next file
Problems Output Debug Console Terminal Ports
node - server
node client
node server
⌘K to generate a command
Cursor Tab Ln 1, Col 1 Spaces: 2 UTF-8 LF JavaScript


Cursor File Edit Selection View Go Run Terminal Window Help
Sun Aug 31 9:38 PM
Toaster.js — Part_2
PART_2
client
build
node_modules
public
src
components
admin
assets
auth
faculty
index
messege
notifications
posts
student
summarizer
thesis
PrivateRoute.js
Profile.js
SessionWarningModal.js
Toaster.js
context
css
hooks
App.js
index.js
package-lock.json
package.json
server
config
db.js
controllers
OUTLINE
TIMELINE
export const ToasterProvider = ({ children }) => {
const showToast = useCallback((message, type = 'info', duration = 3000) => {
setTimeout(() => {
setToasts(prev => prev.filter(toast => toast.id !== id));
}, duration);
return id;
});
const removeToast = useCallback((id) => {
setToasts(prev => prev.filter(toast => toast.id !== id));
});
const showSuccess = useCallback((message, duration) => {
return showToast(message, 'success', duration);
});
const showError = useCallback((message, duration) => {
return showToast(message, 'error', duration);
});
const showInfo = useCallback((message, duration) => {
return showToast(message, 'info', duration);
});
const showWarning = useCallback((message, duration) => {
return showToast(message, 'warning', duration);
});
return (
<ToasterContext.Provider value={{ showToast, showSuccess, showError, showInfo, showWarning, removeToast }}>
{children}
<ToasterContainer toasts={toasts} removeToast={removeToast} />
</ToasterContext.Provider>
);
};
Review next file
Problems Output Debug Console Terminal Ports
node - server
node client
node server
⌘K to generate a command
Cursor Tab Ln 1, Col 1 Spaces: 2 UTF-8 LF JavaScript



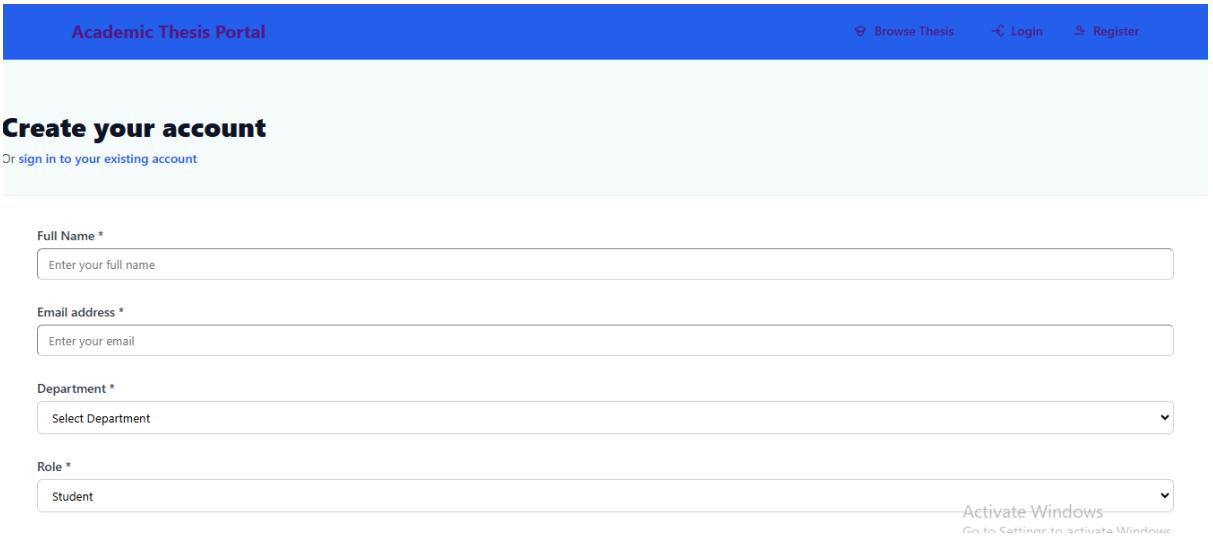
6. User Manual

For new users, this user manual offers a thorough overview of how to use and navigate the Thesis Portal system. To guarantee clarity, it is separated into sections according to user roles (Administrator, Supervisor/Teacher, and Student). Each section contains important features, detailed instructions, and placeholders for screenshots (which can be taken from prototypes or the live system). The portal is designed to be intuitive, with a responsive UI that works on PCs, tablets, and mobile devices.

1. Getting Started: Registration and Login

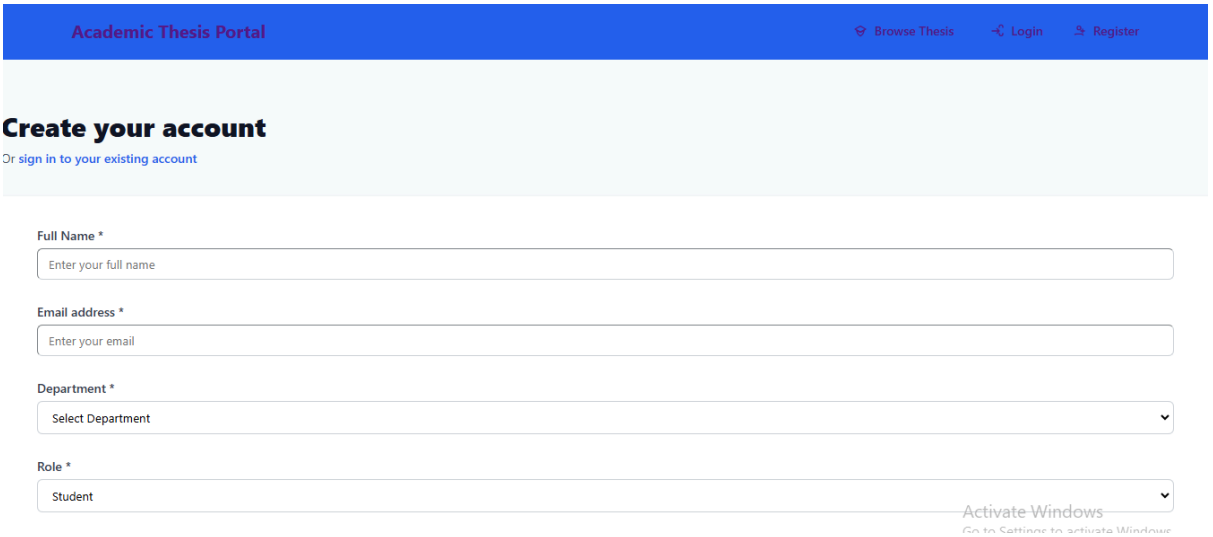
1.1 User Registration

New users (students, teachers, or admins) must create an account to access the portal. Go to the homepage of the Thesis Portal (for example, by using the URL for the login page). The "Register" button in the upper-right corner should be clicked. Enter the necessary information: Name, Email (preferably an institutional email), Department *, student ID (optional). The password needs to have a minimum of six characters. Choose the role of the user: Student, Instructor, or Administrator. "Submit" will create the account.



1.2 User Login

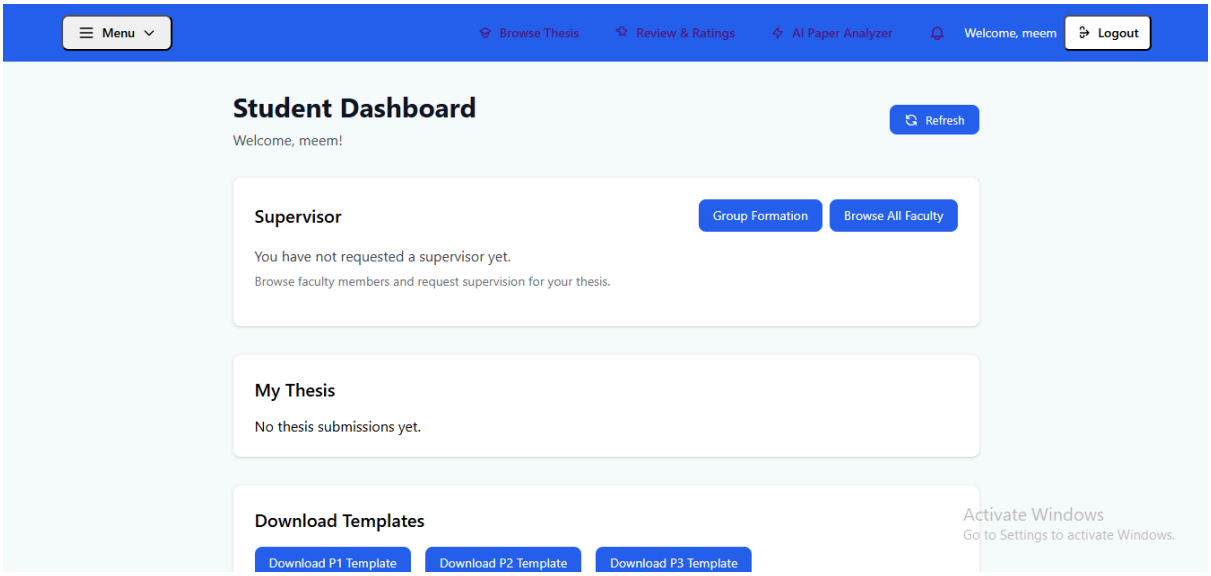
Go to the homepage and click "Login." Enter your registered email or User ID and password. Click "Login" to access your dashboard.



2. Student User Guide

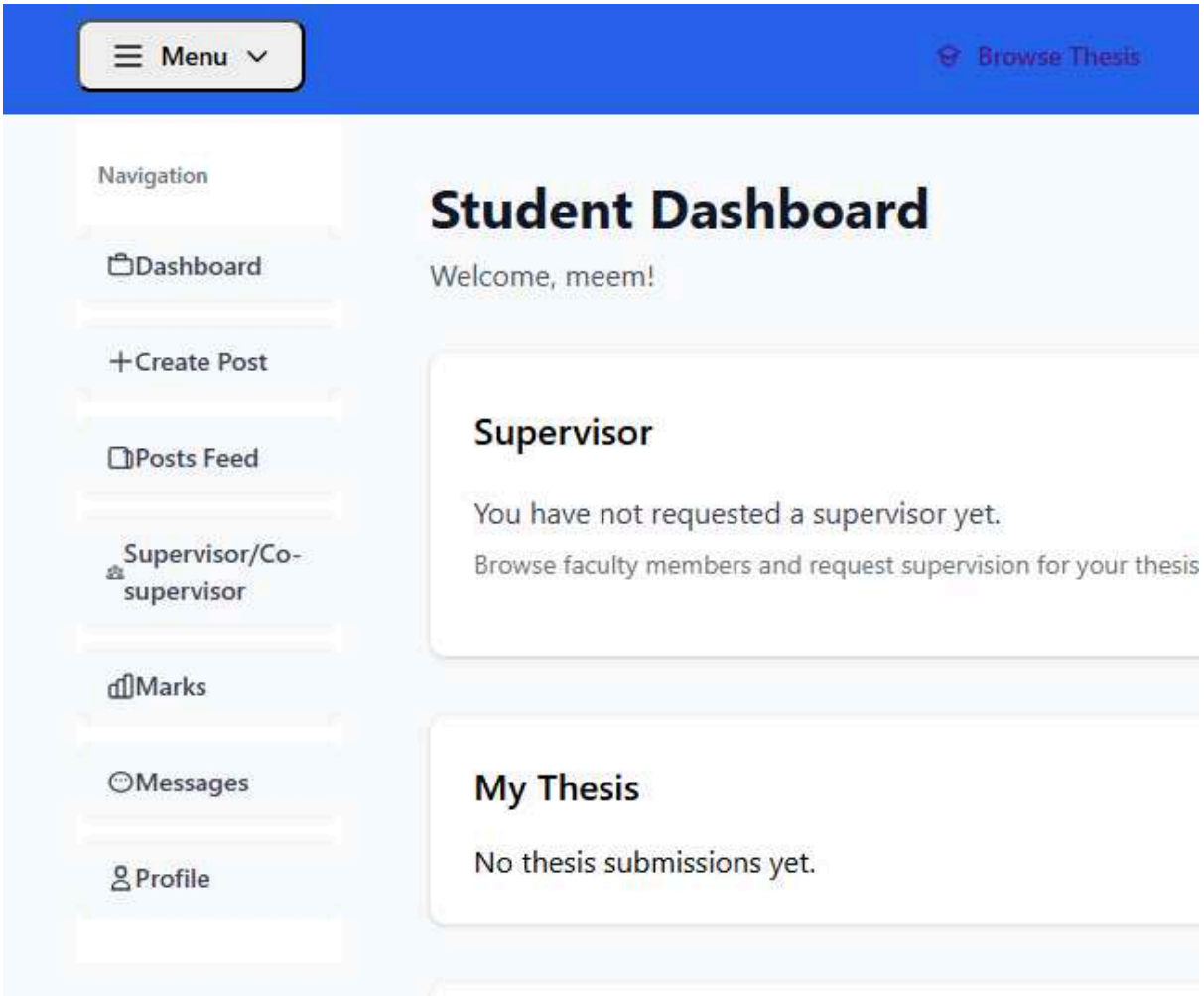
2.1 Student Dashboard

Upon logging in as a student, you will be directed to your personalized Student Dashboard. This is the central hub for managing your thesis process. The dashboard provides quick access to key features and displays relevant information tailored to your current status.



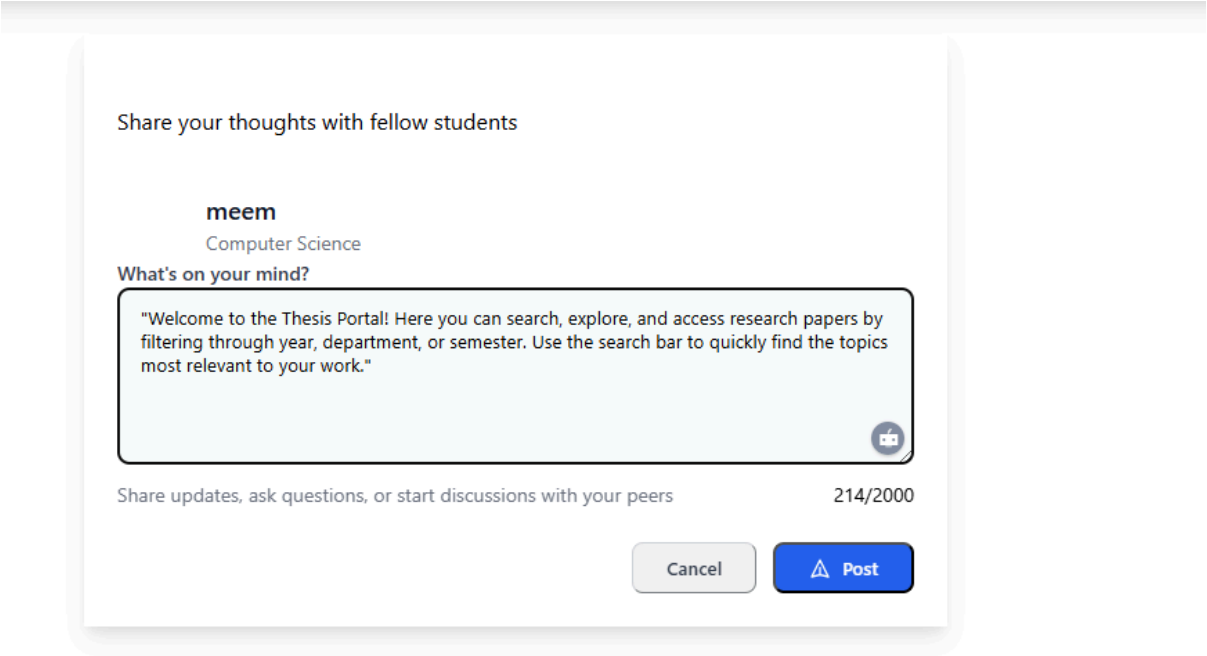
Navigating the Dashboard Features

Menu: Click the "Menu" button (three horizontal lines) to open a sidebar with additional navigation links (e.g., Dashboard, Profile, Create post, Supervisor/co-supervisor, Marks, Messages).

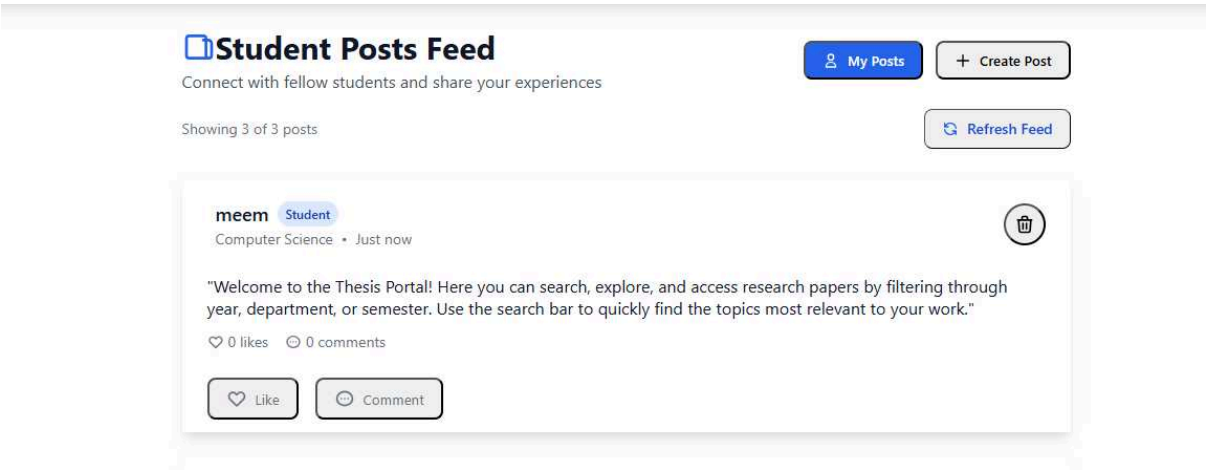


New Post Create:

Upon selecting the "Create Post" option from the menu bar, the Chat interface appears, enabling students to post messages and engage in real-time communication. The page features a clean layout with a "Refresh Now" button to update the conversation, ensuring the latest messages are visible. A sample exchange displays a student's message in the chat window, with a text input field below for typing new posts, allowing for direct and immediate interaction

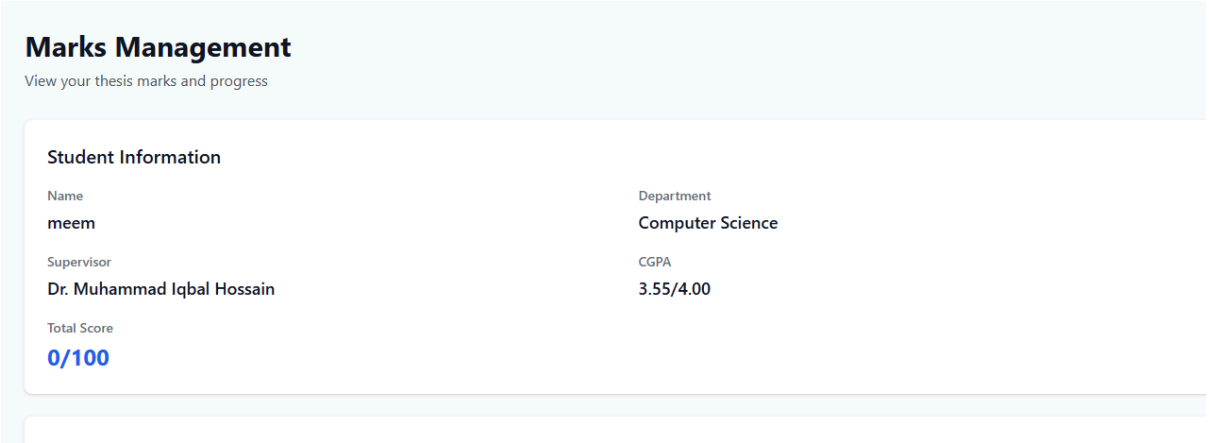


The Student Posts Feed page allows students to connect and share experiences, featuring a feed with posts shared just now, welcoming users, and explaining research access features. Options such as "My Posts," "Create Post," and "Refresh Feed" enable users to manage and update their content, with interactive elements like "Like" and "Comment" buttons enhancing engagement.

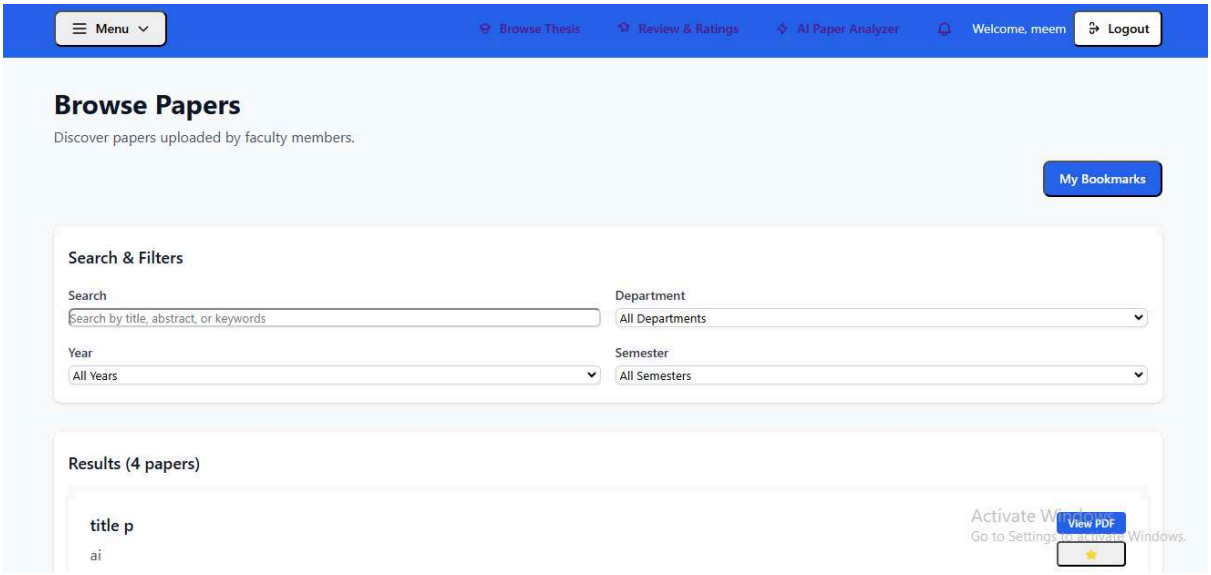


Marks Management:

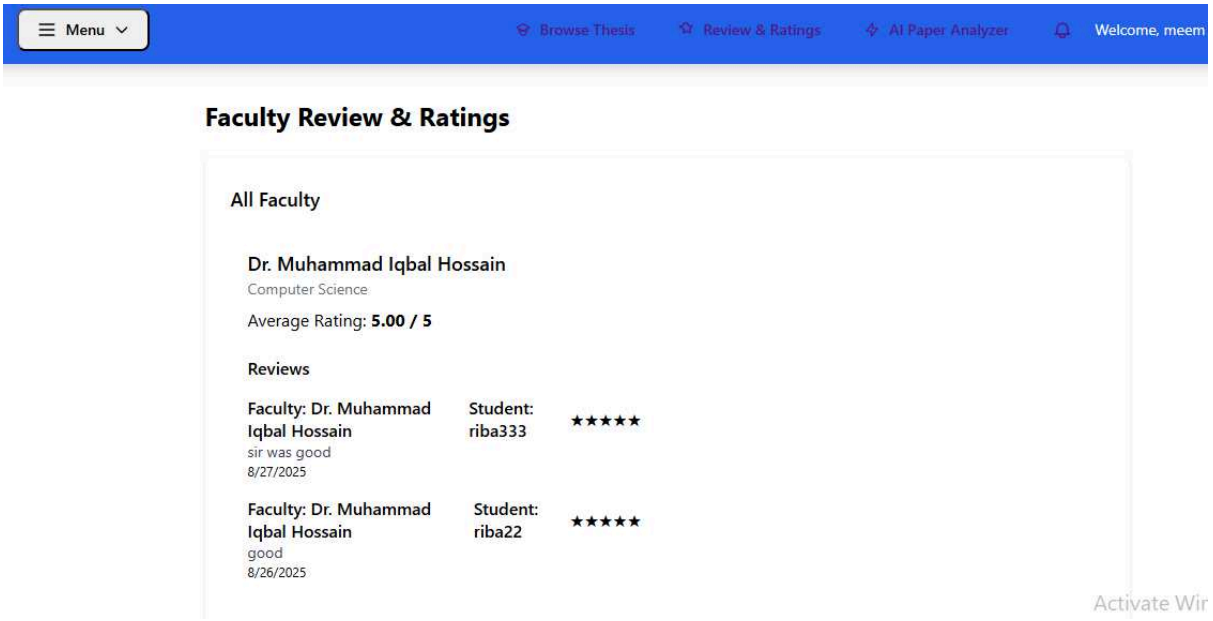
The Marks Management page provides a clear interface for viewing thesis marks and progress, displaying student information and total scores. It includes a section for the supervisor and CGPA details, with the total score highlighted for easy reference.



Browse Thesis: Click "Browse Thesis" in the navbar to go to the Browse Papers page after logging in. You can look through a searchable database of previous thesis documents that faculty members have posted for academic reference in this section. You can easily find relevant research papers using the search and filter options. Simply enter a keyword, title, or abstract in the search field and press Enter to locate papers of interest. To refine your results, you can filter by year by selecting a specific option from the dropdown menu, or choose a department to narrow down the papers by faculty. Additionally, you may select a semester from the dropdown to filter by academic term. By adjusting these filters and searching again, you can quickly locate the most relevant research papers. Click "My Bookmarks" to see a list of saved papers. Remove bookmarks by unselecting the star icon next to a paper in the bookmarks list.

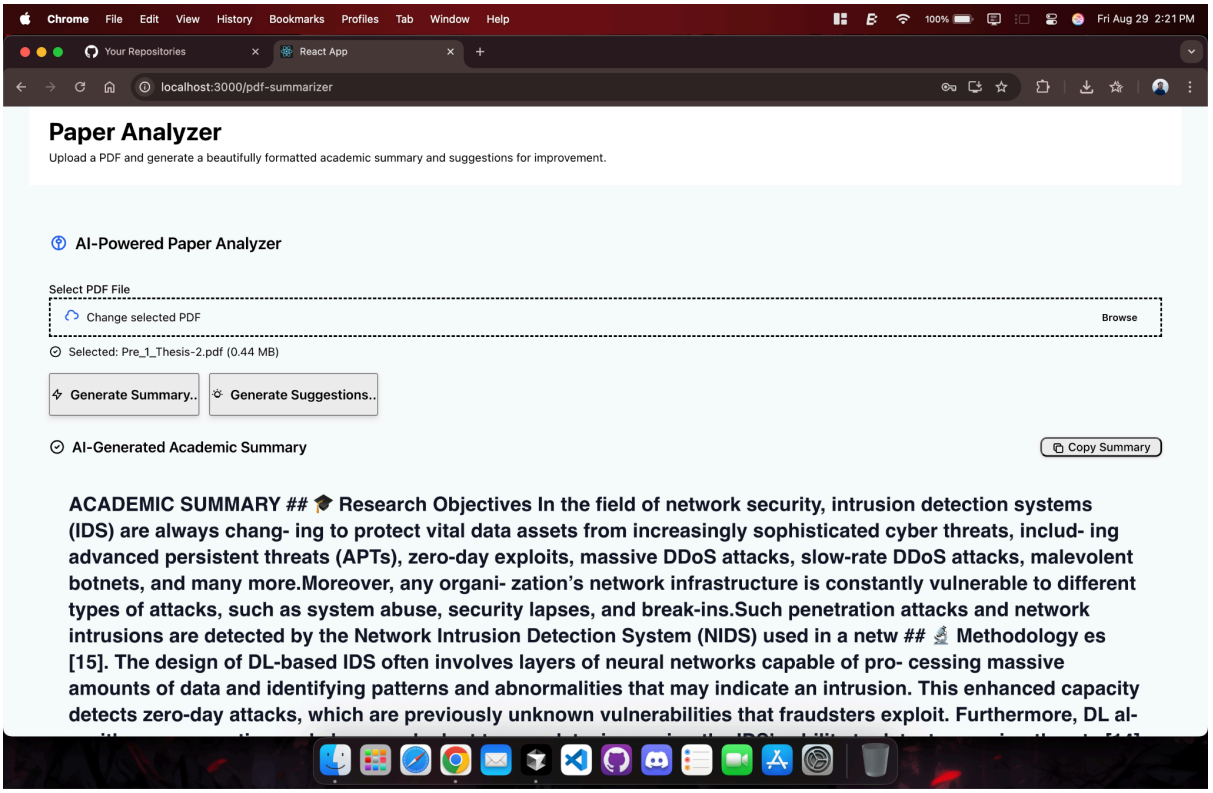


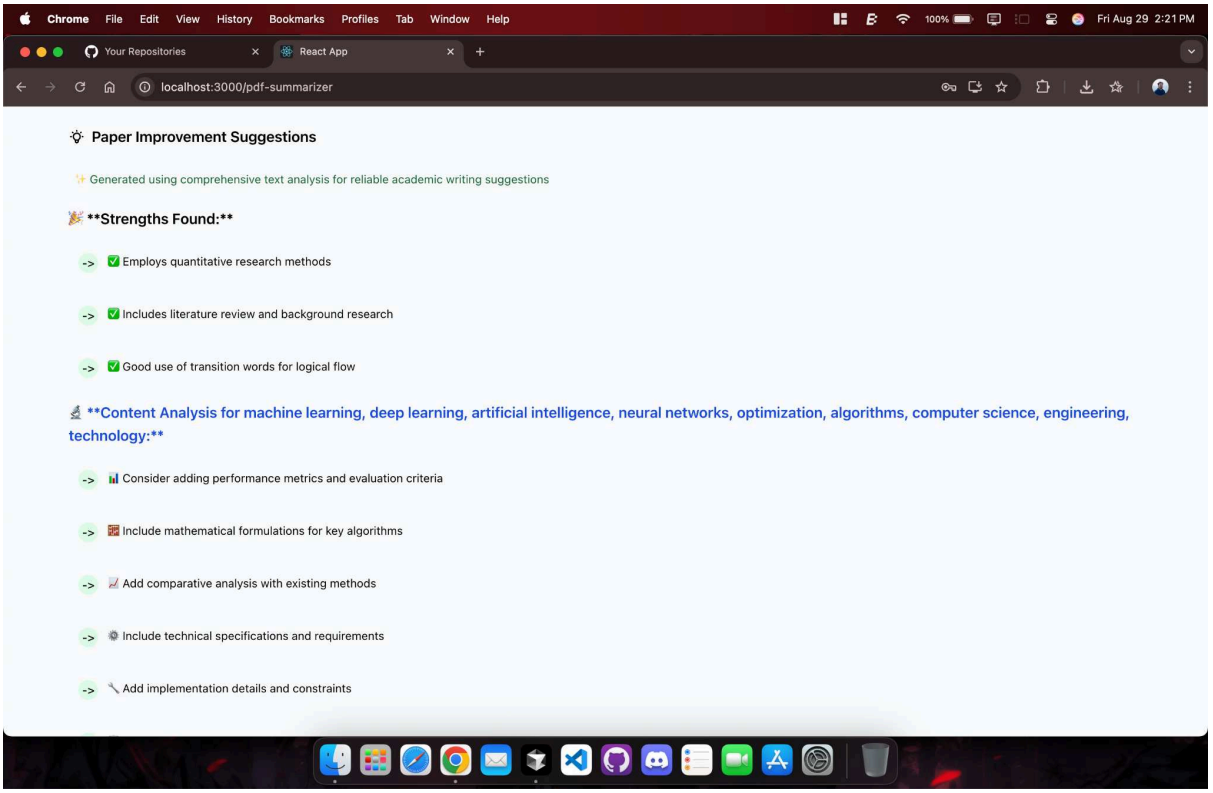
Review & Ratings: Navigate to view supervisor feedback and ratings (visible only after thesis completion).

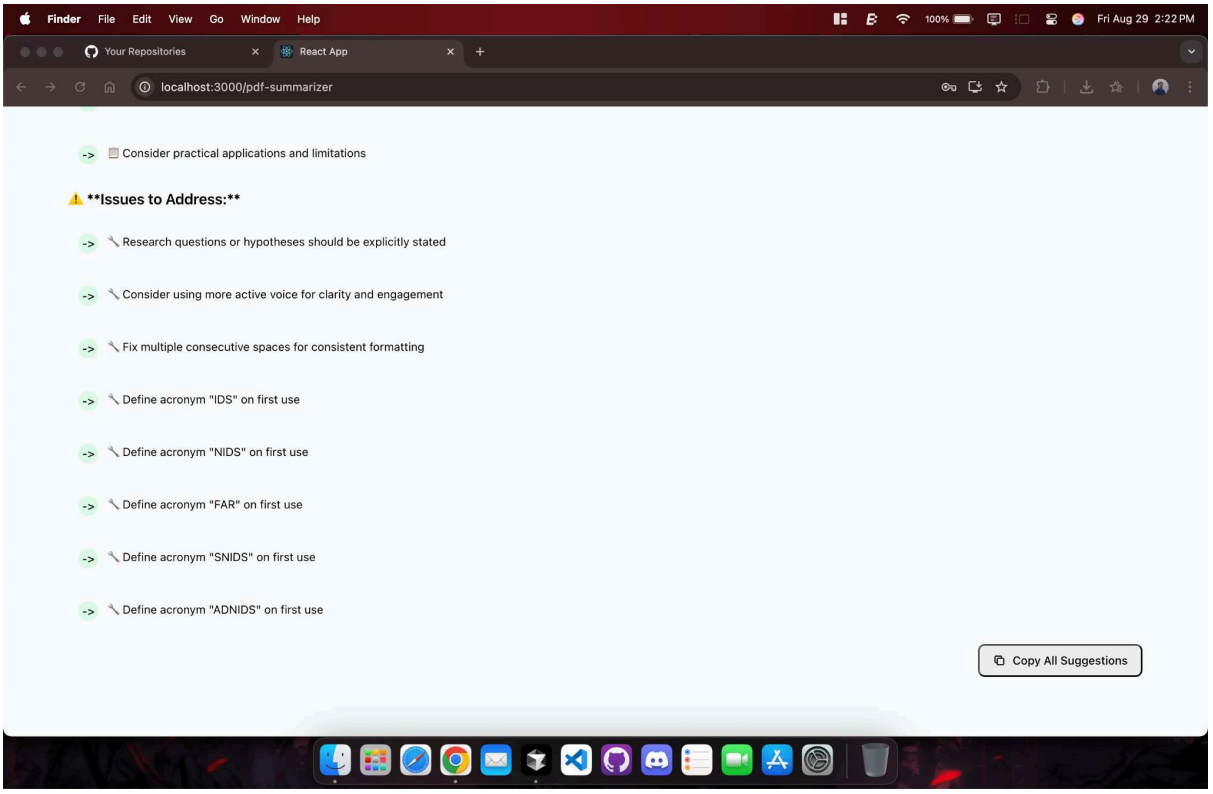


AI Paper Analyzer: Access the AI-powered summarizer to analyze submitted reports (available after submission).

Upload a paper then click summarizer it will generate a brief summary and if clicked on suggestions it will generate suggestions for the uploaded paper.

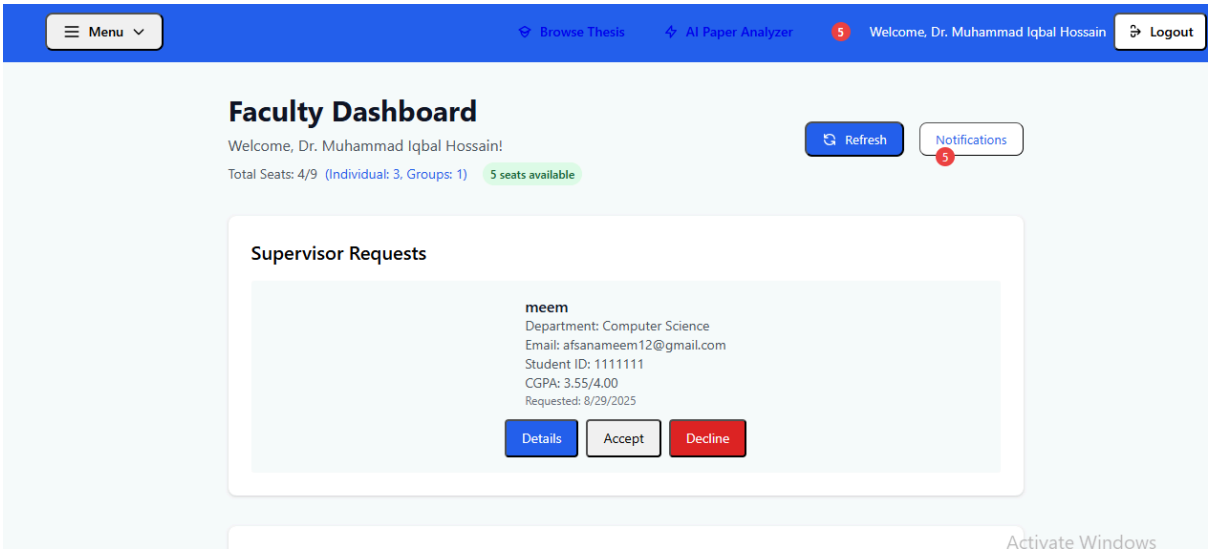




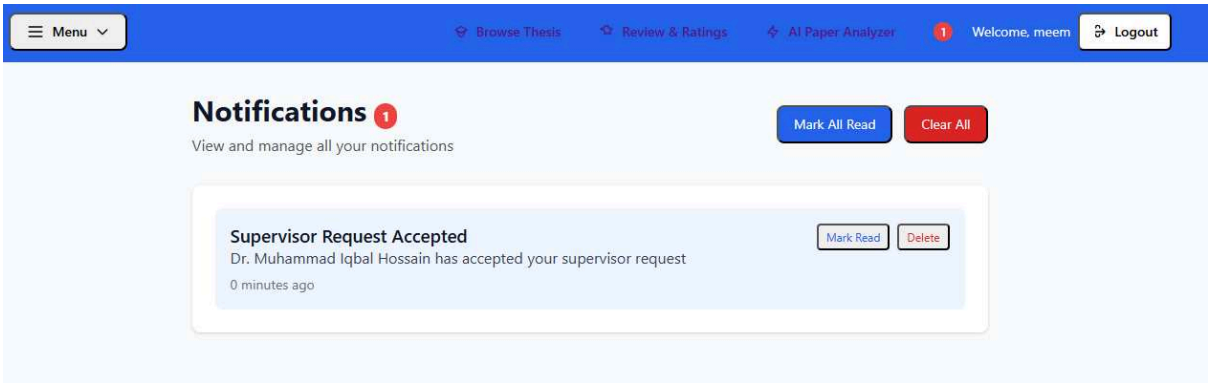


New Supervisor Request

Upon sending a supervisor request from the student side, a notification alert appears on the faculty member's dashboard, indicated by the red "5" badge on the **Notifications** bell icon. The provided image shows the **Faculty Dashboard** for Dr. Muhammad Iqbal Hossain, where the dashboard displays a welcoming message, the total seats available (4 out of 9, with 3 individual and 1 group seat), and a "Supervisor Requests" section. The faculty member can review the request, click "Details" for more information, "Accept" to approve, or "Decline" to reject, ensuring an efficient process for managing supervision assignments.

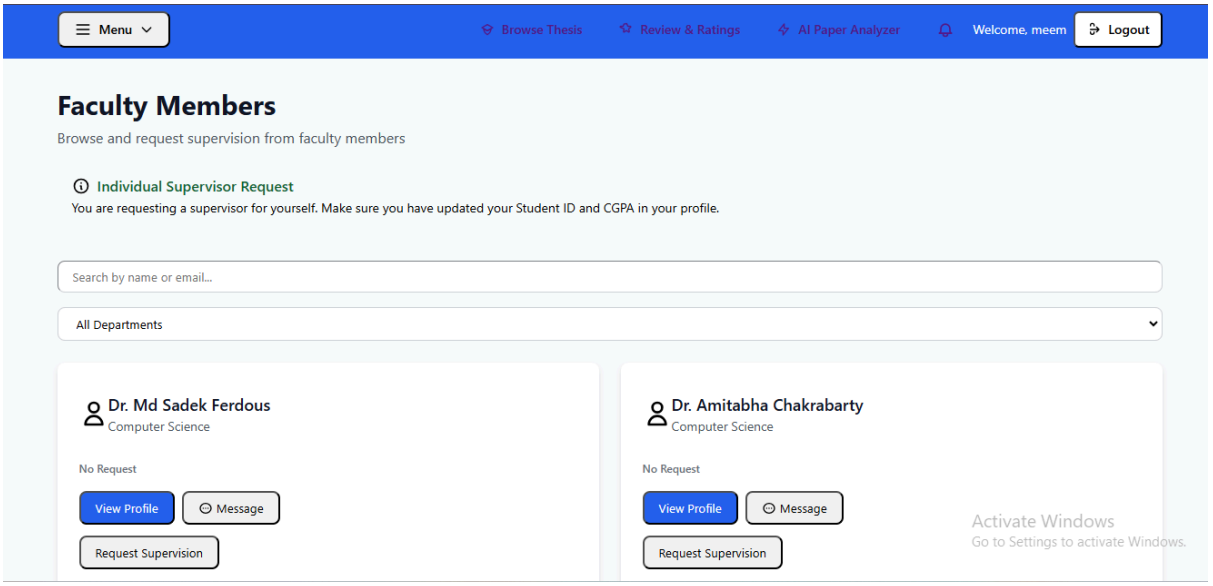


The supervisor has accepted the request.



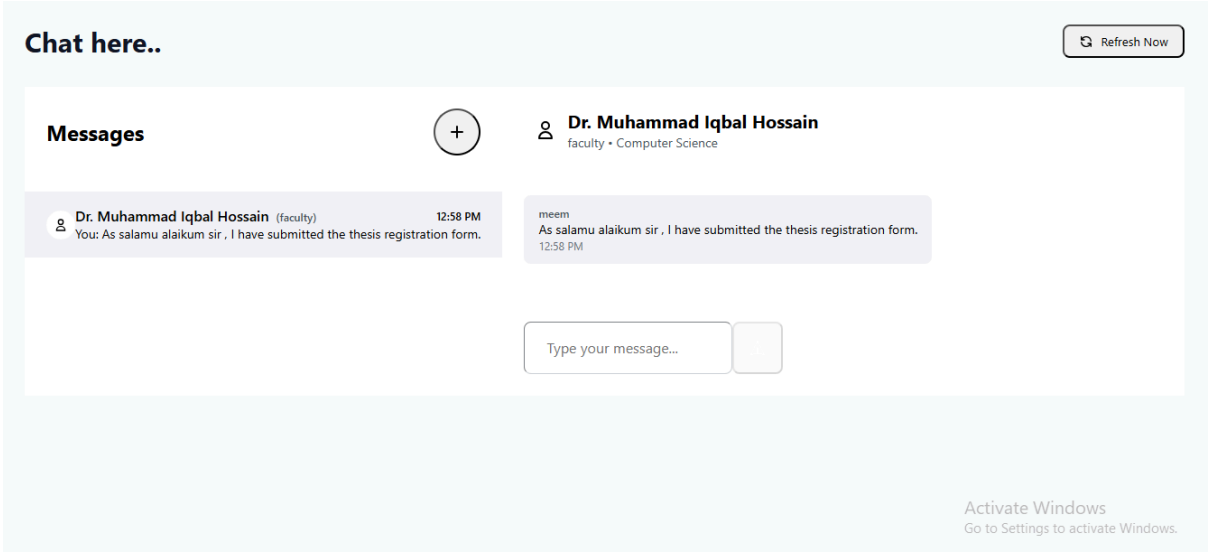
2.2 Browse All Faculty

You can browse and request supervision from available faculty members on the Faculty Members tab after selecting "Browse All Faculty" on the Student Dashboard. This portal offers an easy-to-use interface for departmental filtering, faculty search by name or email, and seeing comprehensive biographies of possible supervisors, including Dr. Amitabha Chakrabarty and Dr. Md. Sadek Ferdous from the Computer Science department. With a clear indication of the current request status (e.g., "No Request"), each faculty listing offers the ability to check their profile, send a message, or request supervision. To ensure a seamless procedure for individual supervision assignments, a helpful remark advises you to update your Student ID and CGPA in your profile prior to submitting a supervision request.



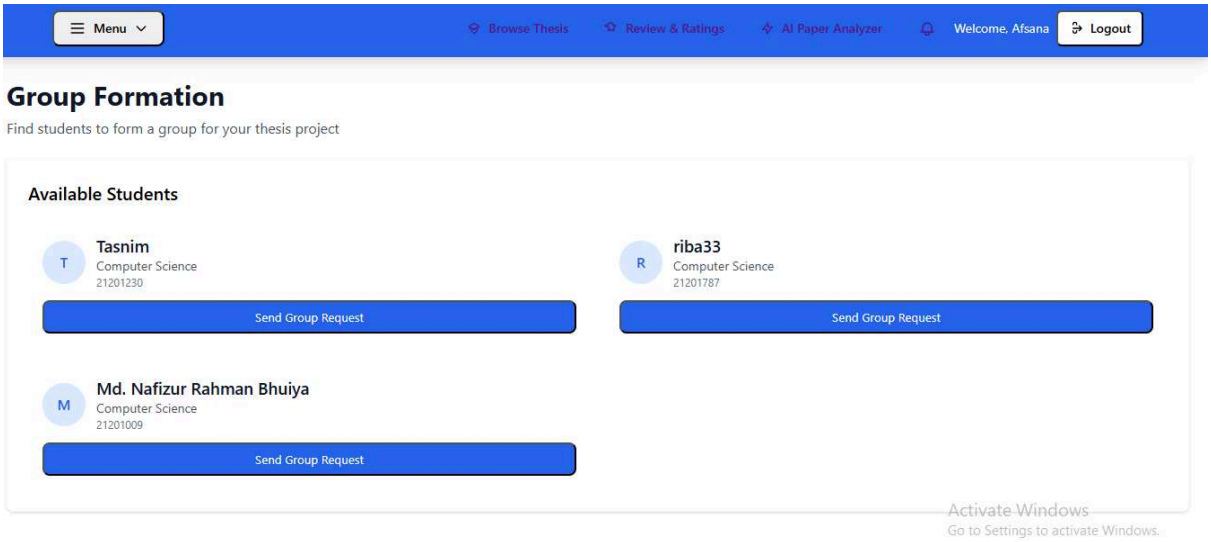
2.3 New Message for Faculty

Upon clicking the "Message" option next to a faculty member's profile on the "Browse All Faculty" page, the Chat interface appears, allowing students to communicate directly with faculty members such as Dr. Muhammad Iqbal Hossain from the Computer Science department.



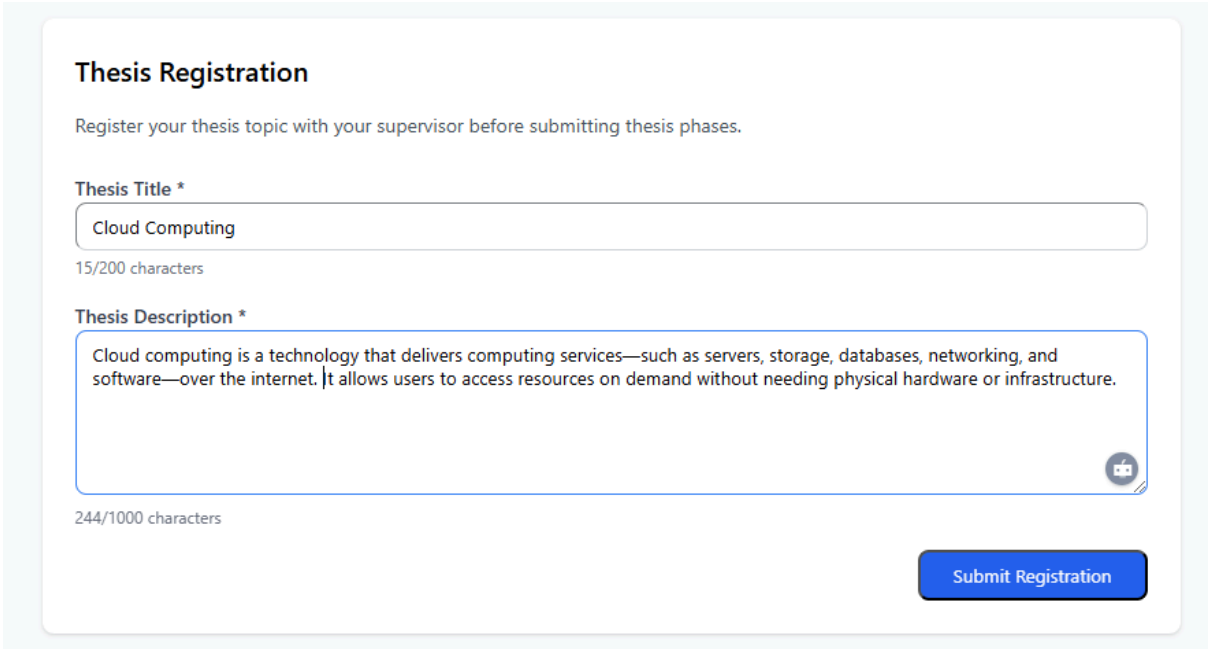
2.4 Group Formation Update

Upon clicking "Group Formation" on the Student Dashboard, the Group Formation page appears, offering a platform to find and collaborate with other students for a thesis project. The page displays a list of available students under the "Available Students" section, showcasing profiles with names, IDs, and departments, such as Computer Science students. Each profile includes a "Send Group Request" button, allowing the user to initiate group formation by sending requests to potential members.



2.5 Thesis Registration

Users are given a simple interface to register their thesis subject with their supervisor before submitting the following stages when they visit the Thesis Registration page. The page includes a form where users can enter a 200-character Thesis Title and a 1000-character Thesis Description field that can hold in-depth explanations (for example, a description of cloud computing as a technology that provides services like servers, storage, and networking over the internet). A "Submit Registration" button allows users to complete their registration.

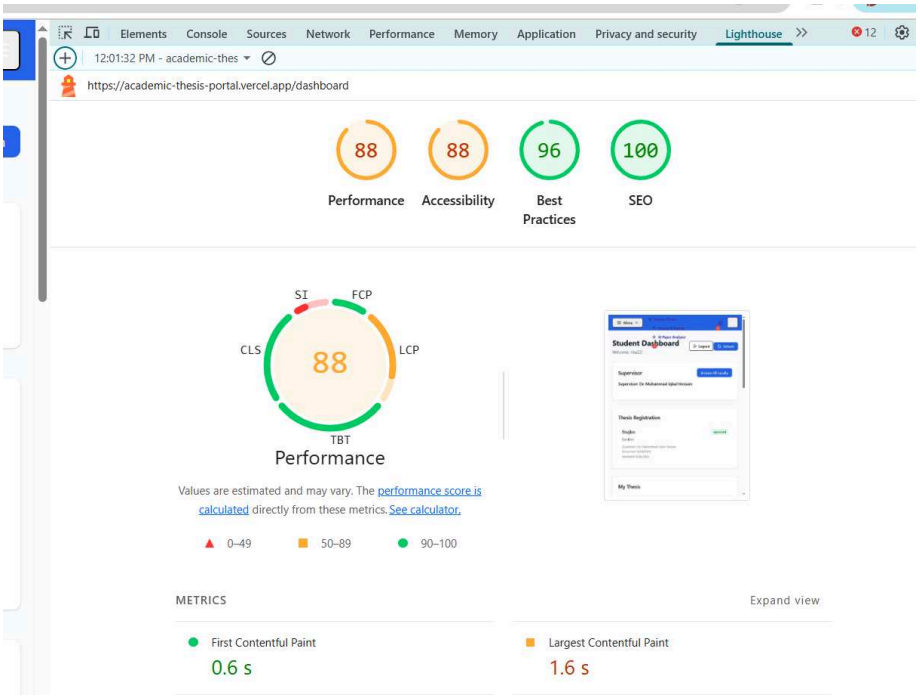


Logout: Click to securely log out of the system.

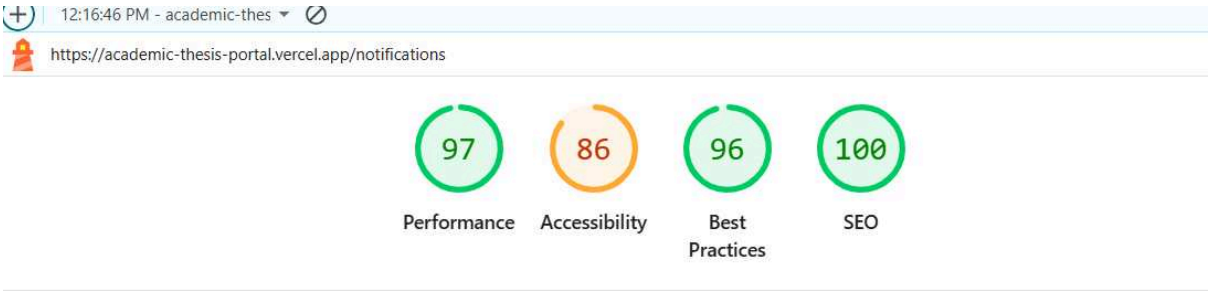
7. Performance and Network Analysis

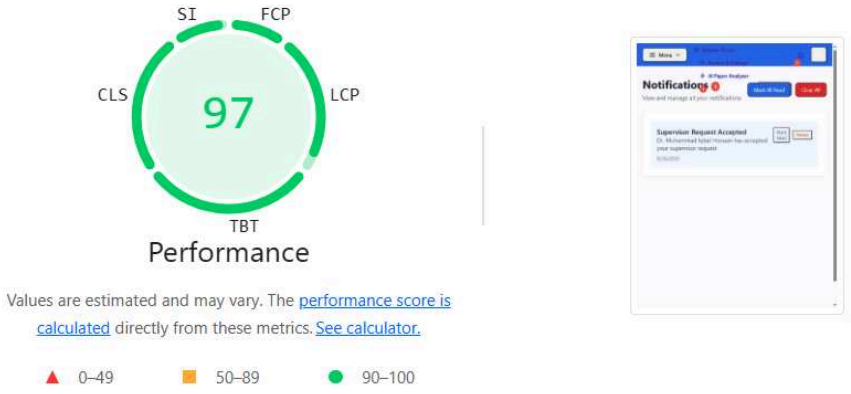
Desktop Analysis

Student Dashboard

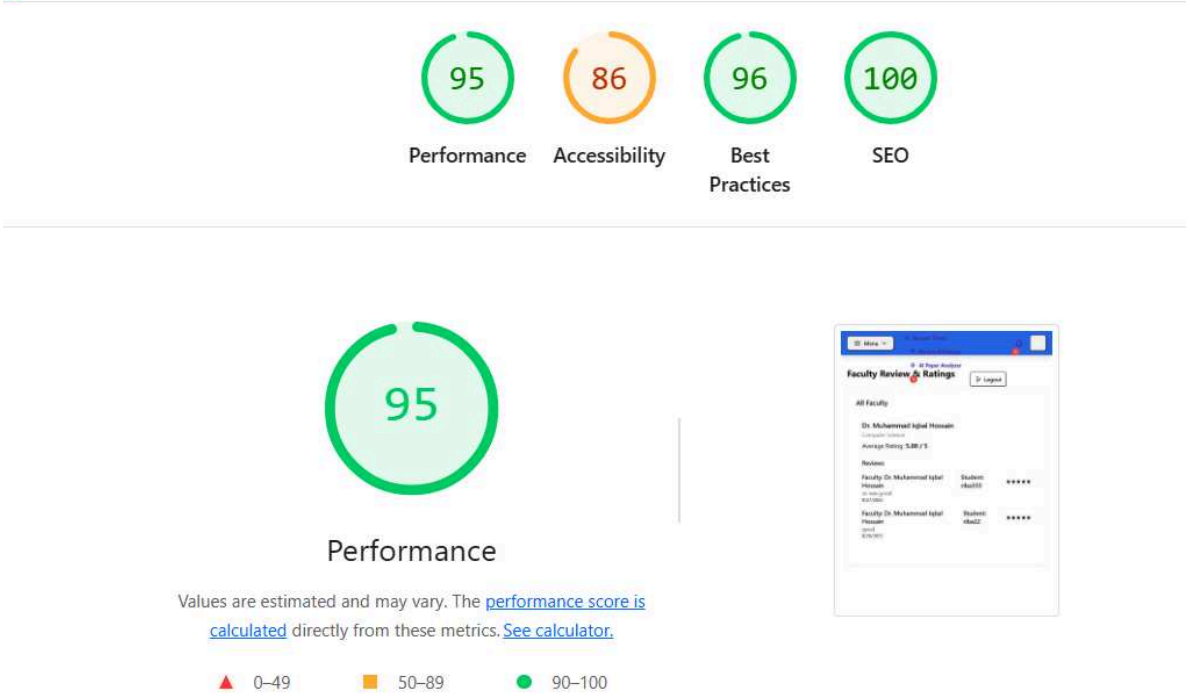


Notification Page



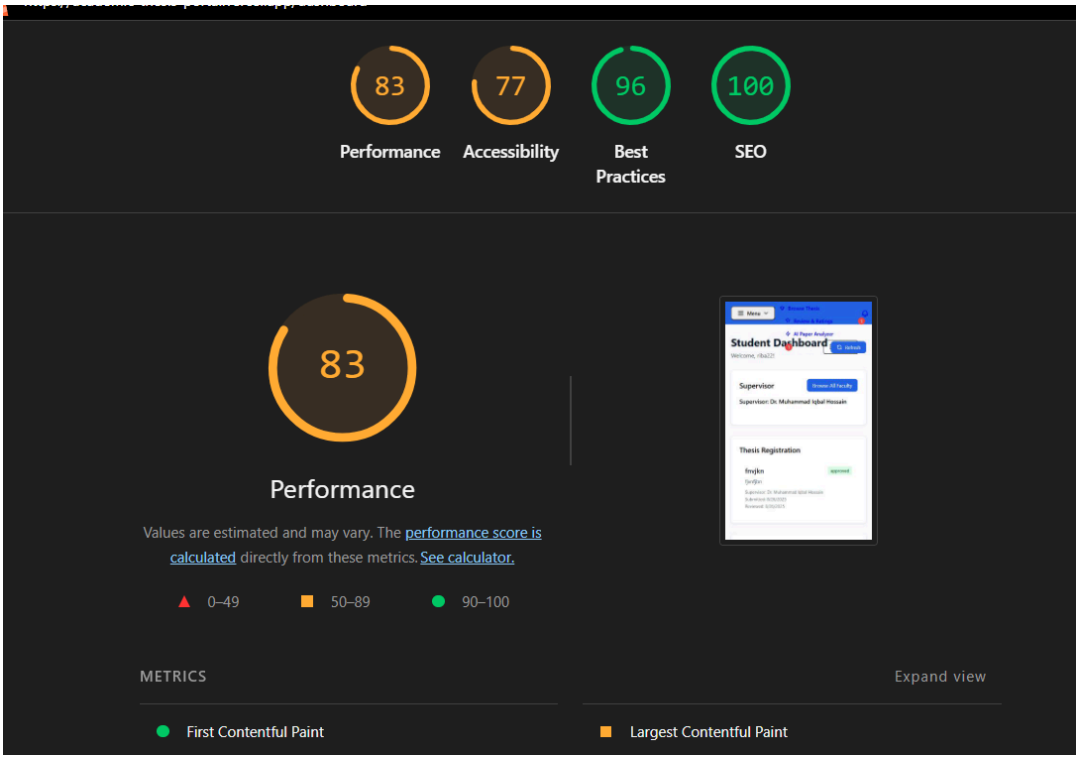


Faculty Review & Rating Page

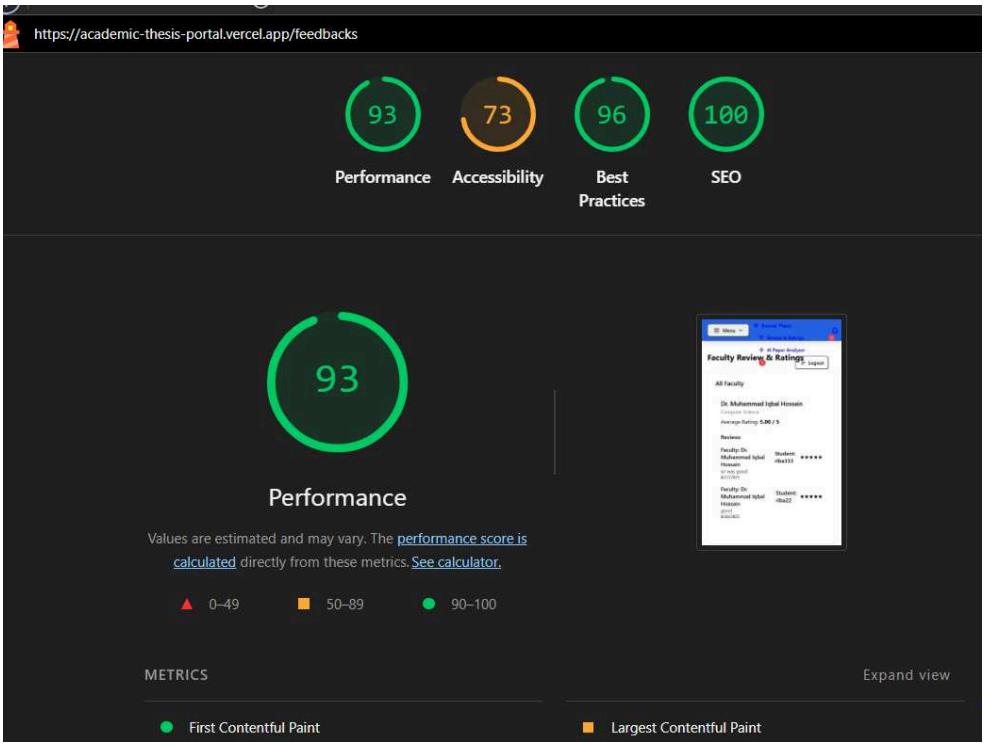


Mobile Analysis

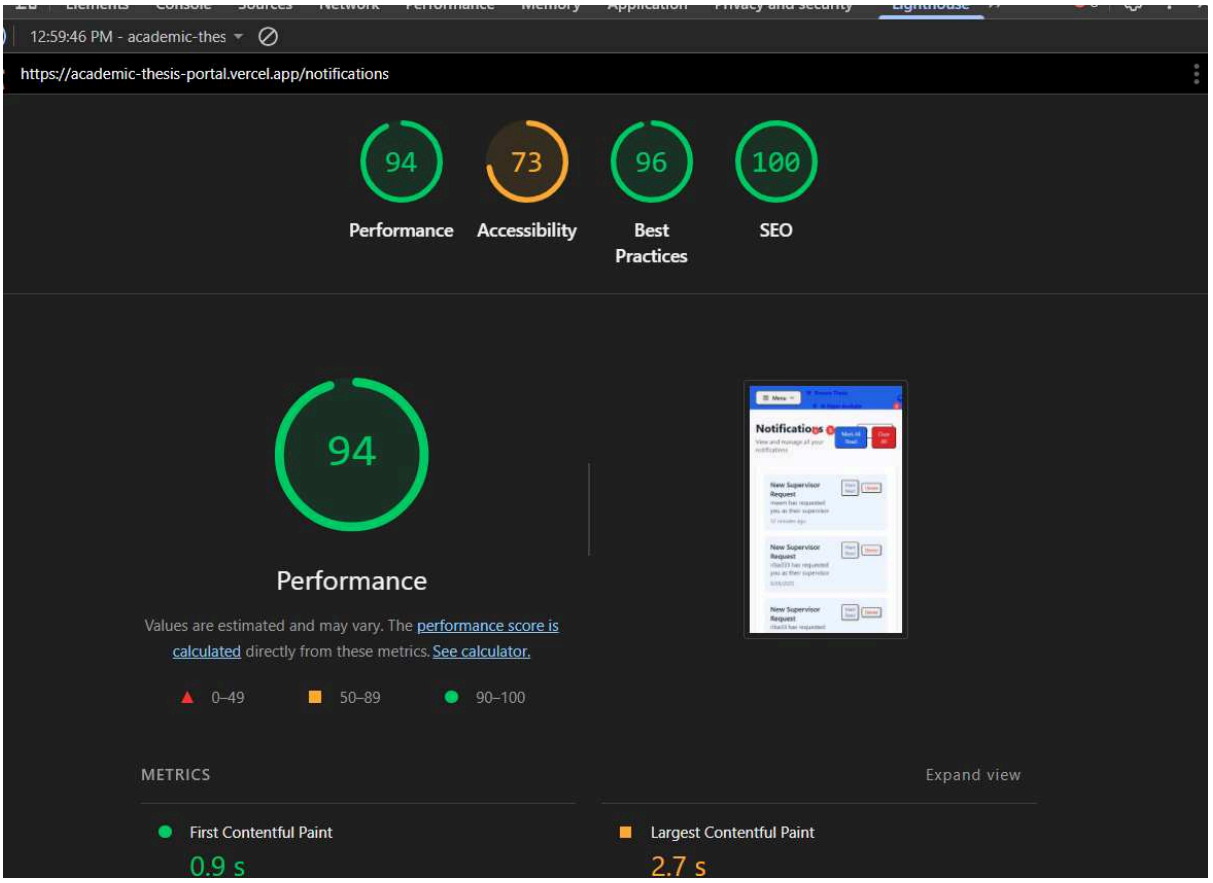
Student Dashboard



Faculty Review & Rating



Notification



8. Github Repo [Public] Link-

https://github.com/tasminoni/Thesis-Portal

9. Link of Deployed Project -

https://academic-thesis-portal.vercel.app/

10. Individual Contribution

Group member - 01	
Name:Tasmin Ahmed Oni	Student ID:21201532
Functional Requirements that are developed by this member:	
1.**Chat/Messaging System**	
2. **Progress Tracking**	
3. **Group Formation**	
4. **Summarize using AI and provide detailed suggestions**	

Group member - 02	
Name: Afsana Akter Mim	Student ID:22101881
Functional Requirements that are developed by this member:	
1. **Notification System**	
2. **Thesis Submission**	
3. **Search, Filter, and Categorization**	
4. **One-Click Report Template Download**	

Group member - 03	
Name: Riba Rahman	Student ID:21201757
Functional Requirements that are developed by this member:	
1. **User Authentication and Authorization**	
2. **Teacher Dashboard/ Student Dashboard**	
3. **Supervisor Selection**	
4. **Feedback and Rating System**	

Group member - 04

Name: Nirvik Shaha Rudra	Student ID:21201241
Functional Requirements that are developed by this member:	
1. **Student Profile and Teacher Profile**	
2. **Documents on Previous Research**	
3. **Bookmarking Tool**	
4. **Admin Dashboard**	

11. References

N/A