MENTOR CONNECT

A

PROJECT REPORT

Submitted by,

CHITRIKA M - 20211CBD0007

ANUPAMA HARIDAS -20211CBD0014

ATHMAKURU DEEPTHI - 20211CBD0053

KALLA SEETHA THANOOJA - 20211CBD0054

Under the guidance of,

Dr. SRINIVASAN T R

in partial fulfillment for the award of the

degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND TECHNOLOGY (BIG DATA)

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

MAY 2025

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the Project report **"MENTOR CONNECT"** being submitted by "CHITRIKA M", "ANUPAMA HARIDAS", "ATHMAKURU DEEPTHI", "KALLA SEETHA THANOOJA**"** bearing roll numbers 20211CBD0007, 20211CBD0014, 20211CBD0053, 20211CBD0054 respectively in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Technology (Big Data) is a Bonafide work carried out under my supervision.

Dr. SRINIVASAN T R	**Dr. PRAVINTHRAJA**
PROFESSOR	PROFESSOR and HOD
School of CSE&IS	School of CSE&IS
Presidency University	Presidency University

Dr. MYDHILI NAIR	**Dr. SAMEERUDDIN KHAN**
Associate Dean	Pro-VC School of Engineering
School of CSE	Dean - School of CSE&IS
Presidency University	Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **"MENTOR CONNECT"** in partial fulfillment for the award of Degree of **Bachelor of Technology** in Computer Science and Technology (Big Data), is a record of our own investigations carried under the guidance of **Dr. Srinivasan T R,** Professor, School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

NAME	ROLL NUMBER	SIGNATURE
Chitrika M	20211CBD0007	
Anupama Haridas	20211CBD0014	
Athmakuru Deepthi	20211CBD0053	
Kalla Seetha Thanooja	20211CBD0054	

ABSTRACT

The traditional mentor-mentee relationship has been significantly altered by the development of digital communication technologies, particularly in the aftermath of the COVID-19 pandemic. Effective and user-friendly digital mentorship solutions are becoming more and more necessary as professional development and remote learning gain traction. A comprehensive web-based mentoring platform based on the MERN stack (MongoDB, Express, React, and Node.js) is presented in this project. Its purpose is to help mentors and mentees communicate in an organized and efficient manner.

In addition to automated procedures like appointment scheduling and attendance tracking, the platform offers real-time video conferencing and chat features to facilitate interesting sessions. The system's ability to track mentee attendance and notify mentors when it falls below 75% is a crucial feature that promotes proactive engagement and accountability.

With a focus on user experience, the platform offers mentees and mentors an intuitive interface for scheduling meetings, communicating effectively, and regularly monitoring learning progress. While the frontend ensures responsive and accessible user engagement across a range of devices, the backend ensures secure data management.

Using previous research on e-mentoring techniques and online learning environments, this report examines the system's architecture, design, and implementation. The platform demonstrates how modern web technologies can be used to develop scalable, interactive, and significant mentorship systems that promote academic and professional development in a post-pandemic setting by utilizing the MERN stack.

Using current research on e-mentoring techniques and digital learning environments, this paper examines the system's architecture, structure, and implementation. By utilizing the MERN stack's capabilities, the platform serves as an example of how modern web technologies can be used to build scalable, interactive, and successful mentorship programs that support professional and academic growth in the wake of a pandemic. Potential improvements might include analytics dashboards to monitor performance, AI-powered mentor recommendations, and mobile application integration to increase user engagement and accessibility even more.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC, School of Engineering and Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Dean **Dr. Mydhili Nair,** School of Computer Science Engineering & Information Science, Presidency University, and **Dr. S. Pravinthraja**, Head of the Department, School of Computer Science Engineering & Information Science, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Dr. SRINIVASAN T R,** Professor and Reviewer **Mr. Rajan T** Assistant Professor, School of Computer Science Engineering & Information Science, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the CSE-7301 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman,** department Project Coordinators **Ms. SUMA N G** and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

Chitrika M - 20211CBD0007

Anupama Haridas - 20211CBD0014

Athmakuru Deepthi - 20211CBD0053

Kalla Seetha Thanooja - 20211CBD0054

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

1.1 Background

A key element of both professional and personal development, mentoring is essential for education, career progression, and the acquisition of necessary skills. Mentoring has traditionally taken place in-person, which, although beneficial, can be limited by logistical challenges, time constraints, and geographic barriers. E-mentoring is a modern method that enables mentors and mentees to interact, work together, and grow virtually. It was made possible by the quick development of digital communication technologies in recent years.

There is a need for platforms that enable meaningful interactions between mentors and mentees as a result of the increased reliance on digital resources for professional networking and education. Throughout the mentoring process, these platforms ought to offer structured guidance, continuous tracking, and accountability in addition to facilitating smooth communication. In order to overcome the drawbacks of conventional mentoring, this study offers a web-based mentoring platform with key features like integrated video conferencing, real-time messaging, automated scheduling, and an integrated attendance monitoring system. This system's ability to alert mentors if a mentee's attendance falls below 75% is a noteworthy feature that encourages early intervention and guarantees both parties' active participation.

1.2 Approach

Utilizing the MERN (MongoDB, Express.js, React.js, Node.js) stack, this initiative develops a web-based e-mentoring platform in order to address the problems associated with traditional and fragmented mentoring systems. Establishing an organized, scalable, and user-friendly environment that promotes seamless interactions between mentors and mentees is the goal of this strategy.

In order to identify common constraints, user expectations, and best practices, the development phase begins with a thorough analysis of current e-mentoring platforms and pertinent scholarly research. Based on these conclusions, the platform is designed to have key components like: Real-time video conferencing were combined to facilitate efficient virtual communication. Automated scheduling to minimize the need for manual coordination and expedite appointment management a system for tracking attendance that alerts mentors when a mentee's count falls below 75%, so enabling prompt interventions.

Access particular for mentors, mentees, and managers to supervise exchanges and track development. A user-focused design methodology is utilized to guarantee that the interface is intuitive and accessible, while the backend architecture ensures data security, real-time communication, and efficient performance. To increase involvement and extend mentoring support, the platform also allows for features including alumni participation and twin mentoring assignments.

1.3 Problem statement

Conventions of traditional mentoring are sometimes constrained by things like geographical distances, a lack of suitable mentors, and trouble planning and tracking mentee development. Especially in situations of education and professional development where continuous guidance and responsibility are absolutely essential, these obstacles hinder regular and effective communication. Accelerated by the COVID-19 epidemic, the global shift toward digital platforms has underlined the need of a complete virtual mentoring solution guaranteeing accessibility, engagement, and organization. Still, modern e-mentoring systems usually lack a complete spectrum of capabilities including real-time communication, automated scheduling, attendance tracking, and role-based interactions. This results in jagged mentoring experiences that reduce the general success of the mentor-mentee relationship.

1.4 Project Objectives

Establish a centralized system that facilitates easy communication between mentors and mentees. To guarantee safe and effective mentoring procedures, implement role-specific access levels (mentor, mentee, admin). To boost communication and collaboration and to facilitate easy communication between mentors and mentees, we have included real-time messaging and video conferencing tools to assure a user-friendly interface for effortless navigation.

Automate Session Management and Scheduling provides a calendar-integrated automated scheduling tool to reduce the need for manual coordination. To reduce the possibility of missed sessions, send out reminder notifications (by SMS or email). Put Attendance

Monitoring & Responsibility into Practice Create an automated system to track mentee engagement and attendance. If a mentee's attendance falls below 75%, mentors should be notified so that prompt action can be taken. Assure Protection and Scalability.

1.5 Requirements and Specifications

COMPONENT	REQUIREMENT	SPECFICATION
Memory	4GB minimum	8GB recommended
Storage	100GB SSD	512 recommended
Processing	Multi-Core	8 cores minimum
Processing	High bandwidth	1Gbps minimum
Security	HIPAA compliant	MD5 -SHAI encryption

1.6 Project Scope

1.6.1 User Management and Role-Based Access

There are three main user roles: Mentor, Mentee, and Admin. Secure authentication is provided through login/signup with email verification. Mentors and mentees can manage their profiles (including bio, skills, and availability).

1.6.2 Real-Time Communication Tools

Mentors and mentees can instantly message each other. Video conferencing is integrated using APIs such as WebRTC or Zoom. Automated Scheduling System, Attendance Tracking & Alerts Session attendance is logged automatically. Alerts are triggered based on thresholds (for example, if a mentee's attendance is below 75%).

1.6.3 Admin Dashboard & Oversight

User management includes the ability to approve or remove mentors and mentees. System analytics cover usage statistics and session logs. There are moderation tools available for handling disputes or policy violations.

CHAPTER-2
LITERATURE SURVEY

2.1 Introduction

The body of studies on mentoring, e-mentoring programs, and the technical underpinnings of virtual mentoring is examined in this literature review. Emphasizing the shortcomings in present e-mentoring systems and the need of a cohesive platform including automated scheduling, tracking attendance, and enabling real-time communication, it follows the evolution of mentoring from conventional face-to-face forms to digital options.

2.2 Essential Elements of Successful E-Mentoring Sites

Successful e-mentoring websites prioritize a user-friendly and secure interface to ensure seamless access and interaction. This includes an intuitive design, mobile responsiveness, and secure user authentication. Clear role-based registration for mentors and mentees, along with verified profiles, fosters trust and builds a safe community for professional growth.

Another key element is an effective matching and communication system. Automated or manual matching features based on interests, skills, and goals help connect suitable mentor-mentee pairs. Integrated communication tools - such as chat, video calls, and scheduling calendars enable consistent and productive engagement. Goal-setting, progress tracking, and feedback mechanisms further support structured mentorship.

2.3 Instantaneous Communication

Face-to-face communication is made possible through video conferencing (integration with Zoom, Microsoft Teams).

2.3.1 Automatic Scheduling Calendar Integration: This feature synchronizes with programs such as Outlook or Google Calendar (Dabbagh & Kitsantas, 2012). Reminder systems that send out email or SMS notifications can help cut down on missed appointments.

2.3.2 Monitoring Attendance and Accountability : Engagement Analytics: Tracks participation and attendance in sessions (Parker et al., 2020). Automated Alerts: Notifies mentors when a mentee's attendance drops below a predetermined level, such as 75%.

2.3.3 Access Control Based on Roles: Dashboards for Mentors and Mentees interfaces specifically designed for different user roles. It also allows the feature for the administrators to monitor the progress of mentoring.

2.4 Technical Structures for Online Mentoring

2.4.1 MERN Stack (Node.js, Express.js, React.js, and MongoDB)

Scalability which includes the ability to efficiently handle sizable user bases.We have also utilised WebSocket integration for real-time notifications and messaging. The site also provides data protection through encryption (SHA-1, MD5).

2.5 Gaps in Existing E-Mentoring Systems

Despite technological progress, current platforms often exhibit: Comprehensive Attendance Monitoring – Few systems actively track and respond to low engagement. Seamless Scheduling – Manual booking processes still dominate. Structured Mentorship Workflows – Many platforms lack defined mentorship processes.

Sl.no.	Paper Title	Proposed Model	Results	Drawbacks
1.	Post Covid Scenario Effective E-Mentoring System in Higher Education(2023)	Development of a web-based e-mentoring application to facilitate communication between university students and mentors, focusing on matching, interaction, and support in a post-pandemic educational	Enhanced mentor-mentee engagement, improved academic performance, and better adaptation to online learning modalities.	Challenges may include ensuring data privacy, user adoption resistance, and maintaining effective communication in a virtual setting.

		environment.		
2	FREEDM ERC Precollege Programs: Motivating Careers in the Electric Power Industry" by Holbert et al. (2020)	Implementation of precollege programs, including teacher workshops, student research experiences, and renewable energy camps, to engage middle and high school students in STEM, particularly in electric power engineering.	Increased interest and enrollment in STEM fields among precollege students, with a focus on diversifying the future workforce in the electric power industry.	Potential challenges include sustaining long-term student interest, securing continuous funding, and effectively measuring program impact.
3	Mentoring in Lower-Level Engineering Courses" by Vasquez et al. (2024)	Integration of mentoring components into lower-level engineering courses to support student learning and retention, possibly through peer mentoring or	Improved student understanding of course material, increased retention rates in engineering programs, and enhanced sense of community among student	Challenges may involve resource allocation for mentoring programs, ensuring mentor quality, and balancing mentoring activities with

		faculty-student interactions.		academic workloads.
4	Work-in-Progress: Emergent Themes from 'High Impact' Role Model and Mentor Narratives" by Trenshaw et al. (2023)	Qualitative analysis of narratives from impactful role models and mentors to identify themes that contribute to effective mentoring in engineering education.	Identification of key characteristics and practices of successful mentors, providing insights for developing effective mentoring programs.	Limitations may include subjective interpretations of narratives and the challenge of generalizing findings across diverse educational contexts.
5	Peer-Mentoring in Design Projects in Project-Based Learning (PBL) at First-Year Engineering Course" by Gadad et al. (2022)	Implementation of peer-mentoring structures within project-based learning environments for first-year engineering students, facilitating collaborative learning and skill development.	Enhanced student engagement, improved problem-solving skills, and smoother transition into engineering education.	Potential issues include varying mentor competencies, possible dependency on peers, and the need for effective oversight to ensure positive outcomes.
6	A Web Platform to Support Mentoring Programs in Higher Education" by Andrade et al. (2024)	Development of a web-based platform designed to facilitate and manage mentoring relationships in higher education,	Streamlined mentor-mentee matching process, improved accessibility to mentoring resources, and	Challenges may include ensuring user engagement, maintaining data security, and providing technical support

		incorporating features like matching algorithms, communication tools, and progress tracking	enhanced tracking of mentee progress	for platform users.
7	Study by Hafiz Iqbal (2024)	While specific details are unavailable, the study likely explores educational strategies or interventions, possibly focusing on mentoring or instructional methods.	Insights into effective educational practices, potentially leading to improved teaching methodologies or student outcomes.	Without specific information, potential limitations could involve contextual applicability and the need for empirical validation.
8	Exploring E-Mentoring: Co-Designing & Un-Platforming" by Alhadlaq et al. (2019)	Investigation of e-mentoring practices through co-design sessions with stakeholders, aiming to develop flexible mentoring approaches beyond traditional platforms.	Development of adaptable e-mentoring models that cater to diverse needs and contexts, fostering more personalized mentoring experiences.	Challenges may include coordinating co-design efforts, ensuring scalability of un-platformed solutions, and addressing technological barriers.
9	Building and Sustaining Mentor	Examination of strategies for	Identification of best practices for	Potential limitations

	Interactions as a Mentee" by Sarabipour et al. (2022)	mentees to effectively initiate and maintain mentoring relationships, possibly through surveys or interviews with successful mentees.	mentees, leading to more fruitful and sustained mentoring interactions.	include the variability of mentor availability and the mentee's ability to implement suggested strategies effectively.
10	Online Mentoring: Programs and Practices" by Susan S. de Janusz and Shelley D. Godshalk(2023)	Comprehensive review of various online mentoring programs and practices, analyzing their structures, outcomes, and impacts on professional development.	Compilation of effective online mentoring models and identification of key factors contributing to successful mentoring relationships in virtual environments.	Challenges discussed may include technological limitations, maintaining engagement in online settings, and ensuring the quality of mentor-mentee interactions.

Table 2.1 Literature Survey

2.6 Conclusion

The literature emphasizes the necessity of an integrated e-mentoring platform that combines role-based access, automated scheduling, real-time communication, and attendance tracking. These gaps are filled by the suggested MERN-based system, which offers an organized, scalable, and intuitive way to conduct successful virtual mentoring.

CHAPTER 3

RESEARCH GAPS OF EXISTING METHODS

Existing e-mentoring platforms still have a number of drawbacks and research gaps despite advancements in digital mentoring solutions. Acknowledging these shortcomings emphasizes how important the suggested system is.

3.1. Lack of Integrated Attendance Tracking and Engagement Monitoring

Most platforms don't keep track of attendance automatically or notify users right away when a mentee's participation starts to wane. As a result, mentors might not notice a mentee's lack of interest until it's too late.

Research Deficit: Automated systems for attendance-based intervention in e-mentoring have not been extensively studied.

3.2. Inadequate Automated Scheduling and Session Management Problem

A lot of platforms still rely on manual coordination (like email exchanges) to set up sessions.As a result, efficiency is reduced and the administrative burden is increased.

Research Deficit: Little is known about using AI to optimize mentorship program scheduling.

3.3. Absence of Organized Accountability Systems

In many mentoring relationships, there are no official mechanisms in place to monitor progress, which leads to irregular follow-ups. As a result, mentees might not receive timely feedback, which reduces the program's efficacy.

Research Deficit: Not many platforms include goal-setting frameworks with automated reminders.

3.4. Inadequate Integration of Tools for Real-Time Communication Problem

Some platforms use different tools (Zoom + Slack + Email, for example), which makes the experience fragmented. As a result, users' cognitive demands are increased and usability is reduced.

Research Deficit: The topic of integrative communication dashboards in mentoring systems has not received much attention.

3.5. Insufficient Tool Integration for Real-Time Communication Issue

Some platforms employ multiple tools (e.g., Zoom + Slack + Email), which fragments the experience. Consequently, usability is decreased and users' cognitive demands are raised.

Research Deficit: Not much has been written about integrative communication dashboards in mentoring systems.

Research Gap	Proposed Solution
No attendance tracking	Automated system with 75% threshold alerts
Manual scheduling	Integrated calendar sync
Poor scalability	MERN stack for flexible, cloud-based deployment
No analytics	Automated reports + engagement trends

Table 3.1 RESEARCH GAPS OF EXISTING METHODS

CHAPTER-4
PROPOSED METHODOLOGY

The methodology for developing the Mentor-Connect platform focuses on systematically addressing the identified gaps while ensuring a robust and user-friendly system.

1. Research and Needs Analysis: Understanding user needs and the shortcomings of the current system is the aim.
2. Literature Assessment: Examine academic publications, pertinent case studies, and well-known online mentoring services.
3. Stakeholder Discussions: Get opinions from administrative staff, mentees, and mentors.
4. Evaluation of Competitors: Examine the benefits and drawbacks of similar platforms.
5. List both functional and non-functional requirements, such as attendance tracking and instant messaging.

4.1 Architecture and Design of the System

The main objective is to create a system framework that is secure, scalable, and easy to use. Below is the architecture of the proposed project, explaining its workflow.

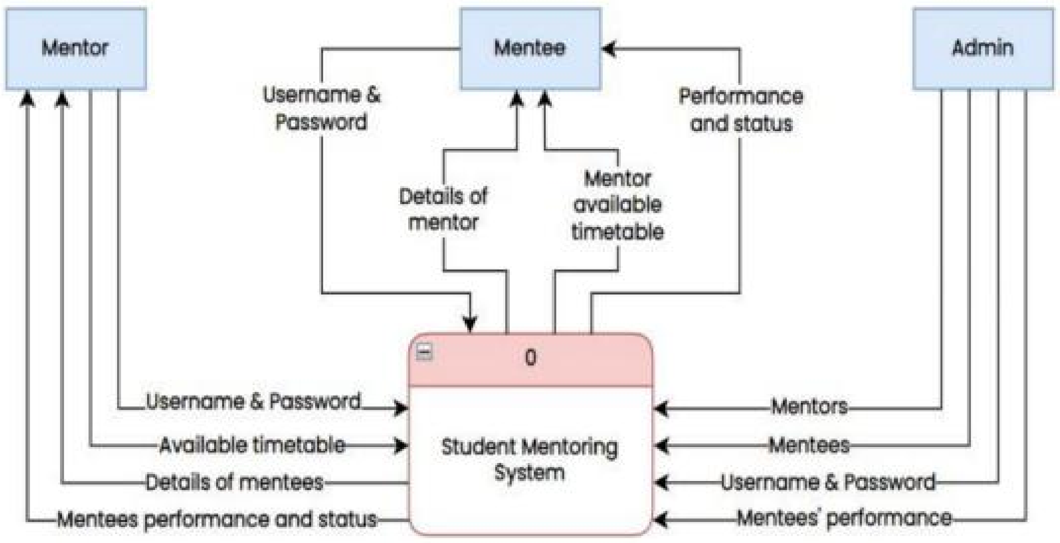


Fig - 4.1 Design and Architecture

4.1.1 General Architecture (MERN Stack)

Dashboards based on user roles are part of the frontend (React.js) user interface.

Business logic is handled by RESTful APIs in the backend (Node.js + Express.js).

MongoDB is a NoSQL database that allows for flexible data storage (session records, user profiles).

4.1.2 Design of Database Schema

i. User Model (Mentor, Mentee, Administrator)

ii. Session Model (Timetable, Participation, Feedback)

iii. Analytics Model (Engagement metrics, Analytics reports)

4.1.3 UI/UX

i. Mentor Dashboard – Scheduling sessions, tracking the progress of mentees.

ii. Mentee Dashboard – Setting goals, viewing session history.

iii. Admin Panel – Managing users, analyzing system data.

4.2. Development Phase Objective

4.2.1 User authentication (JWT-based login system) and backend development (Node.js + Express.js)

API Endpoints (CRUD functions for users, sessions, and attendance)

Connectivity to Third-Party APIs (Google Calendar, Zoom)

4.2.2 React.js Frontend Development

Role-Based Interfaces (Admin, Mentee, and Mentor Dashboards)

Instant Messaging (Socket.io)

The Automated Scheduling Function

4.2.3 MongoDB Atlas Database Implementation

Administration of User Information

Recording Sessions and Tracking Attendance

4.3.Key Features

4.3.1.Role-Based Access Control: Specific user roles (Mentor, Mentee, Admin) featuring customized dashboards and functions.

4.3.2. Video Conferencing Integration: Implementation via API with services such as Zoom or WebRTC for real-time video interactions.

4.3.3 Automated Scheduling: Mentors and mentees can plan and manage their sessions using a calendar-integrated scheduling system that includes automated email and SMS reminders.

4.3.4 Attendance Tracking: Auto-logging of attendance with threshold-based alert system (notifies mentors if a mentee's attendance drops below 75%).

4.3.5 Analytics Dashboard: A graphical display of attendance, session metrics, and engagement rates for both mentors and administrators.

4.3.6 Admin Moderation Resources User acceptance/removal, management of policy breaches, and analysis of system usage statistics.

4.4.Feature Implementation

4.4.1 Role Management and User Authentication Backend:

Node.js and Express.js were used in its development. When a user logs in, JWT tokens are created, and bcrypt is used to safely store passwords. During the registration process, OTPs and confirmation emails are sent using Nodemailer. Depending on the payload of the token, middleware was implemented to secure role-specific endpoints.

4.4.2 Real-Time Messaging

This feature is made possible by Socket.io and allows for continuous chat and effective video call conferencing.

Video Conference:

Option 1: Use the available SDKs and OAuth to integrate Zoom.

Option 2: Use WebRTC to facilitate peer-to-peer communication directly (suitable for self-hosted video calls).

4.4.3 Notifications and Scheduling

Scheduling UI: Built with React-Calendar and Full Calendar components.

Backend Integration: Schedule information is stored in MongoDB with REST APIs for managing bookings.

4.4.4 UI/UX and Dashboards with React.js Frontend

Mentor Dashboard: Progress tracker for mentees, calendar view.

Mentee Dashboard: Goal tracker, session history.

Admin Dashboard: Analytics charts and a user management interface.

Style: Tailwind CSS is used for flexible design and component reuse.

4.4.5 Database Architecture (MongoDB) User Schema

Holds login information, availability, profile information, and user role. Date, time, mentor and mentee IDs, and attendance details are all included in the session schedule.
Notification Schema: Controls engagement notifications, reminders, and system alerts.
Analytics Schema: Tracks session counts, attendance trends, and login frequency.

4.5.METHODOLOGY

4.5.1. Requirement Analysis

Identify the needs of mentors, mentees, and administrators. Gather insights from existing mentoring systems and user expectations.
key features:

- Video conferencing and chat functionality.
- Attendance tracking and automated alerts.
- Automated scheduling and session booking.

4.5.2. System Design & Architecture

Frontend Development: Use React.js or Angular for an interactive, responsive UI. Ensure a user-friendly dashboard with session scheduling, notifications, and attendance tracking.
Backend Development: Develop using Node.js with Express. Implement a RESTful API for smooth communication between frontend and backend.
Database Management: Use MongoDB for unstructured data storage.Maintain user profiles, session details, and attendance records.

Communication & Scheduling Services: Integrate WebRTC or Zoom API for video calls.Use Firebase or Twilio for real-time chat. Implement calendar-based automated booking with Google Calendar API.

4.5.3. Implementation & Features

User Authentication: Secure login using OAuth, JWT, or Firebase Authentication.

Automated Attendance Tracking: If a mentee's attendance drops below 75%, an alert is triggered to notify the mentor. Attendance logs stored in a database for real-time tracking.

Notification System: Use email/SMS notifications for upcoming meetings and attendance alerts.

4.5.4. Testing & Deployment

Unit Testing: Validate each module (video, chat, attendance, scheduling).

Integration Testing: Ensure seamless interaction between frontend, backend, third-party APIs.

User Acceptance Testing (UAT): Conduct trials with students and mentors to gather feedback.

4.5.5. Maintenance & Future Enhancements

Regular Monitoring: Improve system stability and fix bugs.

Feedback-Based Enhancements: Collect user input for iterative improvements.

Advanced AI Features: Introduce sentiment analysis to assess mentee engagement.

CHAPTER-5
OBJECTIVES

5.1.Primary Objectives

5.1.1 Establish a Structured E-Mentoring System

The main goal is to create a centralized digital platform that allows effective interactions between mentors and mentees while maintaining organizational structure. Role-based access control will be part of the system, therefore enabling various functions for mentors, mentees, and administrators. While mentees will have access to tailored development resources, mentors will be armed with tools to track the progress of several mentees. Administrative functions will enable program coordinators to generate institutional reports and monitor general engagement statistics. This structured method ensures that every participant has the appropriate access levels and tools created for their particular responsibilities in the mentoring process.

5.1.2. Provide a Student-Friendly and Accessible Platform

Giving students, the platform's primary users, an enjoyable and simple experience is one of its main goals. The user interface will be carefully crafted to guarantee simple navigation and minimize complexity. For students to get started on the platform quickly, the registration and login procedures will be optimized. Students can monitor and manage their mentorship journey on a customized dashboard after logging in.

The platform will have tools including:

i. Students can add details on their academic background, interests, and aspirations, so improving the matching of mentors. Improved search and filtering tools will help students find mentors depending on industry, skill set, location, language preference, or another criteria. Students will be able to schedule sessions straight on the platform and check mentor availability.

ii. Access to a detailed record of previous interactions, session summaries, and shared resources. These tools are meant to let students, via guided mentoring, take charge of their education and career development.

5.1.3. Give mentors useful tools for managing mentees

Mentors have a significant impact on students' educational experiences. Mentor Connect will provide a full suite of tools to help mentors in their endeavours, with the goal of simplifying and optimizing their workflows.

The platform will offer:

i. Availability and Scheduling: Mentors can manage appointments, set their weekly availability, and get alerts for future meetings.
ii. Mentee Progress Monitoring: A dedicated dashboard where mentors can review mentee profiles, evaluate progress, and offer feedback.
iii. Resource and Document Sharing: Integrated tools for distributing files, notes, and reference materials before or after sessions.
iv. Mentor Coordination: In situations where dual mentorship is in effect, mentors can collaborate with one another to provide consistent guidance to shared mentees.

5.1.4. Give administrators the tools they need to monitor and report

Strong administrative oversight is also necessary for a mentorship platform to be effective. System administrators will be able to oversee the entire platform with Mentor Connects centralized control panel.

The following will be important administrative features:

i. Real-time Attendance Monitoring: The system will automatically record session attendance and send out notifications in the event of low participation or recurring absences.
ii. Mentor-Mentee Pairing: Administrators can pair students with mentors manually or automatically based on alignment, availability, and preferences.
iii. Role-Based Access Control: Users will only be able to view information relevant to their roles thanks to secure access management.
iv. Comprehensive Analytics and Reporting: Administrators will be able to generate reports on mentor performance, mentorship outcomes, user engagement, and platform usage in general.

5.1.5. Use Secure and Smooth Video Conferences for Online Mentoring

The Mentor Connect platform will have a safe, integrated video conferencing feature to facilitate virtual mentoring. This will ensure that all mentorship interactions take place within the platform and remove the need for third-party tools.

Important features for video conferences will include:

i. Superior Audio and Video: Designed to ensure unambiguous communication, even with limited bandwidth.

ii. Screen and File Sharing: Enabling real-time collaboration between mentors and mentees through the use of shared documents or visual aids.

5.2 Success Metrics for Objectives

5.2.1. Enabling Seamless Connections Between Mentors and Mentees

Success in this area will be evaluated based on how quickly and effectively students can find appropriate mentors. Ideally, the average time for a student to connect with a mentor should be under 5 minutes. A positive matching experience would also be shown through user feedback, with at least 90% of students indicating satisfaction with their assigned mentors. Furthermore, the number of successful pairings and ongoing relationships will be monitored to evaluate long-term engagement.

5.2.2 Mentor Active Participation

In this ecosystem, mentors' active participation is essential. In order to succeed, mentors must regularly communicate with their mentees, show up for meetings, and provide value through involvement. We expect mentors to meet with their mentees at least five times a month and receive an average feedback score of 4.5 out of 5. At least 90% of sessions are completed, which is considered healthy and shows that both mentors and mentees are committed to the mentorship program.

5.2.3 User Friendly Platform

The platform's usability will be essential to its uptake and general success. In order to achieve an average platform usability rating of at least 4.3 out of 5, we will regularly conduct user satisfaction surveys. Our goal is to keep the drop-off rate below 10%, so the onboarding process will be closely watched to make sure mentors and students aren't dropping out after registering. Additionally, to demonstrate the platform's user-friendliness, it is anticipated that each user will only generate one support ticket per month.

5.2.4 . Providing High-Quality Video Conferencing

Since the majority of mentoring sessions will take place virtually, seamless video conferencing is crucial. At least 95% of sessions must be finished without any technical problems in order to be successful in this area. Calls should be at least 20 minutes long on average, and post-session evaluations should receive at least a 4.5 out of 5 rating. Since features like screen sharing and chat make for a more interesting mentorship experience, we will also evaluate how well they are used during calls.

5.2.5 Executing Effective Dual Mentorship

Each student is paired with two mentors under the dual mentorship model in order to provide a variety of viewpoints. For this approach to be successful, at least 85% of students should be assigned dual mentors, and a satisfaction rating of at least 80% should be achieved for this model. Getting input from mentors and mentees will help improve the structure and operation of dual mentorship.

5.3 Secondary Objectives

5.3.1 To Establish a Safe System for User Authentication

Use JSON Web Tokens (JWT) and encrypted passwords to create a dependable login and registration process for mentors and students. This protects user confidentiality and data integrity by ensuring that only authenticated users can access the platform.

5.3.2 To Make Real-Time Reservations and Scheduling Possible

Provide a user-friendly interface so that students can schedule sessions and see if mentors are available. This streamlines the mentorship booking process by providing options for cancellation, reminders, and status tracking (e.g., confirmed, pending, completed).

5.3.4 To Promote Easy Video Communication

Include a video conferencing function (e.g., via WebRTC or third-party APIs) that allows mentors and students to have virtual meetings within the platform, reducing the need for outside tools and guaranteeing session continuity.

5.3.5 To Create an Admin Dashboard That Is Easy to Use

Provide a centralized management panel that allows administrators to monitor users (mentors and students), add new accounts, monitor platform activity, and collect analytical data.

5.3.6 To Put Mentor and Student Profile Management Into Practice

Permit users to add important details to their profiles, like their training, areas of expertise, and mentoring experience. This promotes greater matching between mentors and students and increases transparency.

5.3.7 To Include a Session Feedback and Rating System

Allow students to offer feedback and assess the mentoring session following each one. This helps evaluate mentors' performance and encourages quality control.

5.3.8 To keep thorough session logs and analytics

Record all session metadata, such as the date, time, length, and feedback. Display this information graphically using dashboards to track participation, find well-liked mentors, and assess the success of sessions.

5.3.9 To guarantee responsive design and cross-platform compatibility:

Using responsive design principles, create a platform that works flawlessly on desktop, tablet, and mobile devices, improving the user experience on all screen sizes.

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

The comprehensive web-based program "Mentor Connect" was developed to streamline and enhance the mentoring process in both academic and professional settings. The platform facilitates live video conferencing, efficient session scheduling, comprehensive attendance tracking, and simple communication for three different user groups: administrators, mentors, and students. The Mentor Connect platform, which is constructed with the MERN stack (MongoDB, Express.js, React.js, and Node.js), is described in detail in this document.

6.1 System Architecture

6.1.1 Overview

The architecture has a distinct division of responsibilities and is multi-tiered. Every tier communicates via secure API endpoints and plays a distinct role.

Frontend (React.js): Form management, routing, interaction logic, and user interface elements.

Backend (Node.js & Express.js): middleware configurations, business logic, and API endpoint definitions.
Database (MongoDB): Mongoose ODM for long-term data management and storage.

Third-Party Services: Real-time communication through integrated video conferencing APIs (e.g., Jitsi/Zoom/WebRTC).

6.1.2 Component Dissection

React-router-dome manages client-side routing to provide smooth navigation.

State Management: Use State is used locally, and context is used globally when needed.

Authentication: a secure token system for session management based on JWT.

6.2 Implementation of Backend

6.2.1 Configuring the Server

Express.js has been used to configure the server. CORS, express. Json, and a custom authentication middleware are examples of middleware components.

The following make up the API routing hierarchy:

/auth is in charge of the registration and login procedures.

/admin: used for administrative duties like assigning mentors.

Endpoints for mentor dashboard data are provided by /mentor.

/student – helps students interact with mentors and sessions.

6.2.2 Authorization & Authentication

User identity is encoded using JWT in client-side tokens.

Passwords are securely hashed using bcryptjs.

Middleware confirms that the user's role matches the route's access requirements.

6.3. Structure of Databases

6.3.1 Schemas and Collections

i. Schema for Users

ii. Name, email, password, role, department, and assigned mentors are among the attributes.

iii. Schema for Mentoring Sessions

iv. Features: topic, timetable, feedback, mentorId, and studentId

v. Schema for Attendance

vi. Features: month, total, and studentIdDays and leavesMeetings takenattended

vii. Schema for Video Sessions

viii. Features: recordingLink, startTime, endTime, and sessionId

ix. Object references (ObjectId) are used to create connections.

6.4. Features of the Frontend

6.4.1 Dashboard for Students

View the mentor or mentors that have been assigned to you.

Engage in mentoring sessions via integrated video calls.

Keep track of individual attendance and feedback.

Modular elements and CSS modules were used in the interface's design.

6.4.2 The Mentor Dashboard

See the list of mentees who have been assigned.

Set up meetings and keep track of attendance.

Access meeting minutes and feedback history.

6.4.3 The Administrator's Dashboard

Handle user account creation, reading, updating, and deletion.

Assign students to mentors.

Keep an eye on departmental attendance summaries and session analytics.

6.5 Analytics & Attendance Module

6.5.1 Tracking Attendance

Every session's attendance is recorded.
It makes use of the OverallStudent.js form submission interface.
Quick metrics are provided by utility functions such as calculate Monthly Attendance Percentage and calculate Overall Attendance Percentage.

6.5.2 Attendance Submission Example

```
Await axios.post(`/update/student/attendance/${studentId}`, attendanceData)
```

6.5.3 Visualization of Data

dynamic bars that show percentage metrics are displayed.
Traffic light color indicators: Red denotes missing records, and green indicates up-to-date records.

6.6 Integration of Video Conferencing

6.6.1 Integration Strategy

To create peer-to-peer connections, WebRTC is suggested.
As an alternative, session scheduling data is used in conjunction with Zoom or Jitsi APIs.
Along with session information, video links are integrated and stored.

6.6.2 Features That Are Available

a) Live video calls that happen right on the platform.

b) Reminders that are scheduled.

c) Feedback gathering following sessions.

6.7. Security Procedures

Using JWT tokens to control sessions

Role-based access control implementation

Secure transmission of data through HTTPS

Passwords are stored using bcrypt in an encrypted format.

6.8. Effectiveness and Growth

Requests from asynchronous APIs to preserve non-blocking I/O

MongoDB indexing to improve read operation speed

React components can be loaded slowly to increase rendering efficiency.

6.9 Implementation Session Booking

6.9.1.Retrieving Data

i. Axios is used to retrieve reservations from the backend API.

ii. From localStorage, the student ID and authentication token are obtained.

iii. In the event that the token is invalid, it handles token expiration and reroutes to the login page.

6.9.2.Features of the User Interface

i. Use the mentor's name to look for reservations.

ii. Sort sessions according to their status (confirmed, pending, all, etc.).

iii. Present booking cards with pertinent information on them.

iv. Once the status has been verified, allow session joining.

v. Use a modal popup to view session information.

6.9.3.Reusable Components

Booking Details Modal for showing session data.

spinner loading while the data is being retrieved.

6.10 Implementation of Admin Dashboard

Overall Mentors & Students

Each displays:

- Total number
- User table (name, department, email, etc.)
- The ability to amend or remove records
- Search and filter options

Examples of user interface functions:

- The search bar
- Filters by year or department
- Modify or remove icons
- The use of pagination

6.11 Backend Implementation

6.11.1Admin JWT Authentication System

JWT-based authentication system for admin users using Node.js, Express, MongoDB, and bcrypt for password hashing. The authentication flow supports both admin registration and admin login, providing secure access to protected routes via a custom middleware.

Technologies Used:

i. Node.js (server environment)

ii. The web application framework Express.js

iii. MongoDB is a NoSQL database system.

iv. Dotenv (environmental variable management)

v. JSONWebToken (JWT) for user verification

vi. bcrypt (password hashing)

vii. Cross-origin resource sharing, or CORS

Using Node.js, Express, MongoDB, and JWT (JSON Web Tokens), a backend authentication framework for administrator users has been created. The main goals are to protect backend routes via token-based authentication and to make it easier for administrators to register and log in securely. This system uses MongoDB for data storage, specifically storing admin user information in a collection called admin within the studentMentor database, and is built on Express.js for handling routing and HTTP requests. Admin registration and admin login are the two main features of the system. The server expects to receive an email, password, secret key, and a type field that reads "register" during the registration process. It verifies the secret key against a secure environment variable and makes sure all required fields are present. The server saves the credentials in the database after hashing the password with bcrypt if the administrator doesn't already exist.

The client enters their email address, password, and a type field labeled "login" to complete the login process. The server verifies the administrator's identity and determines if the password entered corresponds to the hashed version that is kept in the database. A JWT token is created and returned if the credentials match.

6.11.2 Student Login Backend

Students can safely log into the mentoring platform using the student login API (/student/login). The system checks to see if the roll number is in the database when a student enters it along with their password. An error message appears if it cannot be located. The entered password is compared to the stored (hashed) password using bcrypt if the roll number is found. If the password is incorrect, an error message is displayed. A secure token (JWT) is created using the student's ID if the password matches, enabling the user to stay logged in for the duration of their session. The system displays the token and student ID in a success message and removes the password from the response for security.

6.11.3 Working Of Backend

The system is designed to use JWT (JSON Web Tokens) as a token-based approach to implement a strong authentication mechanism. The system's backend, which is based on Node.js and Express, allows administrative users to register and log in. The database used to store administrator credentials and associated information is MongoDB. The system securely encrypts passwords using the bcrypt library and confirms that the administrator's email address is unique during the registration process. It verifies user information during the login process and provides a signed JWT token, which the client can use to gain access to resources that are protected**.**

6.12. Modules

User Management Module : Secure user login and access control based on roles (OAuth, JWT). Profile creation including details for both mentees and mentors (bio, expertise, availability). Pairing of mentors and mentees.

Video Conferencing & Chat Module : Live video communication (WebRTC, Zoom API, or Jitsi). Instant messaging (Firebase/Twilio). Ability to share files for study materials, notes, or assignments.

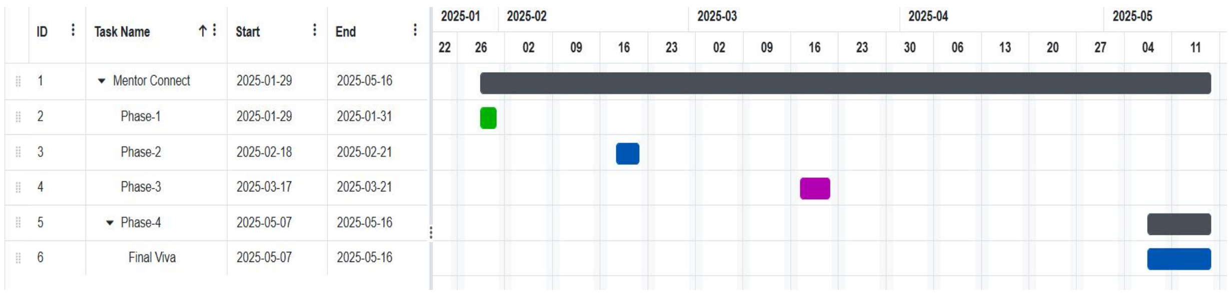
Module Attendance Tracking & Monitoring : Mentees record their attendance through the platform. If attendance drops below 75%, a notification is sent to the mentor. Attendance records are kept in the database for tracking and reporting purposes.

Automated Scheduling & Appointment Module : Integration with calendars (Google Calendar API) for scheduling purposes. Automatic booking of meetings based on mentor availability. Reminder alerts for upcoming sessions.

Dashboard & Analytics Module : Mentors can view mentee progress, attendance records, and session documentation. Admins can oversee overall platform engagement and statistics.

Notification & Alerts Module : Email/SMS reminders for sessions and attendance notifications. Instant chat alerts for new messages.

CHAPTER-7
TIMELINE FOR EXECUTION OF PROJECT
(GANTT CHART)



ID	Task Name	Start	End
1	Mentor Connect	2025-01-29	2025-05-16
2	Phase-1	2025-01-29	2025-01-31
3	Phase-2	2025-02-18	2025-02-21
4	Phase-3	2025-03-17	2025-03-21
5	Phase-4	2025-05-07	2025-05-16
6	Final Viva	2025-05-07	2025-05-16

Fig 7.1 Project Timeline

The image is a Gantt chart showing the timeline of tasks for a program called "Mentor Connect," running from January 29 to May 16, 2025. It includes four phases and a final viva, each represented by colored bars indicating their durations. Phase-1 to Phase-3 occur sequentially from January to March, while Phase-4 and the Final Viva take place in May. The chart visually tracks progress and scheduling of each task across the months.

CHAPTER-8
OUTCOMES

8.1. Enhanced Mentor-Mentee Communication

Real-time video conferencing and chat will improve interaction and accessibility for mentees.
Seamless file sharing will enable effective knowledge exchange.

8.2. Improved Attendance Monitoring

Automated attendance tracking will ensure mentees maintain regular participation.
Mentor notifications for attendance below 75% will help in early intervention.

8.3. Streamlined Appointment Scheduling

Automated session booking will eliminate scheduling conflicts.
Integration with Google Calendar or in-app scheduling will make it easy to manage mentorship sessions.

8.4. Personalized Mentorship & Learning Experience

AI-powered mentor-mentee matching based on expertise and preferences (if implemented).
Progress tracking will help mentors identify mentee strengths and weaknesses.

8.5. Data-Driven Insights & Performance Monitoring

Mentors and admins will have access to analytics dashboards with engagement statistics.
Sentiment analysis (future enhancement) will help assess mentee engagement and satisfaction.

8.6. Increased Accountability & User Engagement

Mentees will be more accountable due to attendance tracking and performance monitoring.
Push notifications and email alerts will keep users engaged.

8.7. Secure Login System

Tokenization and encryption were used to create a safe and dependable login system for students. After being hashed with bcrypt, passwords are safely stored, and to ensure session integrity, a JWT (JSON Web Token) is created upon successful login. This method creates the application's first layer of security by ensuring that student accounts stay safe and are only accessible with legitimate credentials.

8.8 Booking Dashboard For Student

Students receive a customized dashboard with all of their scheduled mentoring sessions. Statuses such as "pending," "confirmed," "completed," and "rejected" are used to arrange the sessions. Students can easily track past and upcoming mentoring sessions and understand the current status of their bookings thanks to this organization.

8.9 Admin Dashboard

After confirming their reservations, students can immediately access mentoring sessions through an integrated video call feature. By removing distance barriers and facilitating virtual interactions between mentors and students, this live capability makes mentoring easier.

8.10 Adding New Student and Mentor

Students can classify sessions based on their current statuses using the platform's easy-to-use filters. Furthermore, by allowing users to search for particular mentors by name, a real-time search bar enhances functionality. Together, these elements produce a seamless and user-friendly reservation process.

8.11 Resposive User InterFace

Modern UI principles and responsiveness were considered throughout the platform's design. The layout adapts to provide a clear and consistent experience whether viewed on a desktop computer or a mobile device. The interface is more effective and engaging when it has user-friendly elements, animations, and icons.

8.12 Loading and Error Handling

Appropriate loading indicators are shown while data is being fetched to improve user experience; error messages are shown when a problem occurs (e.g., invalid login, server errors). This system of feedback guarantees that users are always aware of what is going on behind the scenes and never left perplexed**.**

CHAPTER-9
RESULTS & DISCUSSIONS

9.1.System for Student Authentication

Both valid and invalid cases were thoroughly tested for the login feature. The system successfully verified student credentials by using JWT for token creation and bcrypt for secure password validation. Students were only allowed access after correctly entering their roll number and password. Clear error messages were given when credentials were entered incorrectly. By ensuring that only verified students could use the platform's features, this secure authentication method reduced the possibility of unwanted access.

9.2 Controlling Session Reservations and Filtering

Essential features for scheduling sessions are included in the student dashboard. Students are presented with a list of their forthcoming mentoring sessions when they log in. These sessions can be sorted by users according to their current status, which includes rejected, completed, confirmed, and pending. By updating the session view in real-time without requiring a page reload, the filter functionality boosts user experience and performance. Additionally, users can find specific mentors by name thanks to the booking list's seamless integration with the search tool.

9.3 Modal for Session Information and Joining Features

Students can view specific session information in a modal window by selecting the "View Details" button. This window contains important information like the mentor's name, the session's scheduled time, and its booking status. The modal also has a "Join Session" button for confirmed sessions. Students who click this button are taken straight to the session's specific video call link. To avoid any confusion or unwanted entries, the procedure was carefully tested to make sure that only sessions marked as confirmed could be joined.

9.4 Assessment of the Admin Dashboard

The admin interface was designed to act as a central location for platform management. Administrators can easily switch between views, add new users, and keep an eye on all enrolled students and mentors. Every section—students, mentors, add student, add mentor—loaded properly with forms and user-specific information, and tab navigation was seamless.

Additionally, the admin interface required proper role validation for access and maintained user authentication across tabs.

9.5 Mechanisms for Error Management and Reaction

Building a dependable system requires strong error management. This application uses loading indicators, inline error notifications, and alert pop-ups to provide real-time feedback. Examples include notifications for failed reservations, missing session links, or unsuccessful login attempts. These mechanisms provided clarity during slower network responses and helped users during errors. Using appropriate fallback messages made sure that the platform was always easy to use.

9.6 Database Efficiency and Backend API

MongoDB was used as the database and Node.js was used to build the backend. It supported all basic CRUD (create, read, update, and delete) functions for mentors, students, and session scheduling. Postman was used to evaluate APIs in order to verify data formats, response statuses, and routes. Even with multiple concurrent requests, MongoDB queries produced accurate and quick results. Every protected route included token verification, and appropriate error handling for missing or invalid tokens was implemented.

9.7 User Interface Adaptability

React and CSS were used to create a user interface that is both accessible and flexible. The layout ensures consistent usability across desktops, laptops, tablets, and smartphones. Input fields, buttons, icons, and status indicators are examples of visual components that have been improved for clarity and accessibility. Icons (such as FaSearch and FaVideo) improved user interaction and gave the site a more contemporary look. Chrome DevTools and device simulations were used to test the UI's responsiveness.

9.8 Instantaneous Performance Evaluation

During the use of the booking dashboard and session management system, real-time responsiveness was evaluated. Axios was used to retrieve all data asynchronously, which reduced waiting times. Filtering and searching were instantaneous, and booking lists and modals loaded rapidly. As long as there was a working link, the video calling feature guided users with ease. Additionally, the admin panel efficiently supported CRUD operations, allowing changes to be reflected instantly without the need for a page refresh.

9.10 Summary of Findings and Observations

In conclusion, the project's goal of creating a strong system for managing and communicating between mentors and students was accomplished. Every essential feature, including bookings, administration, and login, was developed and tested with positive outcomes. The system is responsive, safe, and scalable. While real-time chat and calendar synchronization are absent from the current version, the framework is adaptable enough to accommodate such future additions. The results validate the effectiveness and dependability of the design decisions made during the development phase.

CHAPTER 10

CONCLUSION

The aim of this project was to create a comprehensive platform for booking and managing student-mentor interactions that ensures effective communication and scheduling between students and mentors. This concluding chapter summarizes the entire process of project execution, the insights gained, the challenges overcome, and the potential for future enhancements. During development, it became increasingly clear that there was a need for a streamlined and user-friendly platform to handle mentorship interactions. Conventional mentorship systems frequently experience challenges such as inadequate scheduling tools, inefficient communication interfaces, and problems with tracking mentorship effectiveness. The solution provided here resolves these issues by offering a modern web-based interface with role-specific functionalities for administrators, students, and mentors.

This system's scalable and modular architecture is a noteworthy achievement. The MERN stack - Express.js for server-side logic, React.js for an engaging frontend, Node.js for the runtime environment, and MongoDB for a NoSQL backend—was used to build the application. Cross-platform compatibility, real-time updates, and a seamless development process were all made possible by this stack.

Creating intuitive user flows received a lot of attention from the standpoint of user experience. Logging in, browsing mentors, making appointments, and managing bookings are all simple tasks for students. In turn, mentors are able to monitor their forthcoming meetings and give prompt updates. The database is fully accessible to administrators, who can add, edit, or remove mentors and users as well as use dashboards to examine statistical insights.

Security was another important issue that the project addressed. Only verified users will be able to access role-specific resources thanks to JWT-based authentication. Sensitive tokens are securely stored in localStorage/sessionStorage, and passwords are hashed using bcrypt. This stops unwanted access and protects data integrity.

This platform's integrated video conferencing feature, which allows mentors and students to join sessions straight from the application interface, is a noteworthy feature. This encourages participation and lessens dependency on outside resources. To guarantee that only authorized users take part, the join link is only made available for sessions that have been approved.

In order to deliver a flawless user experience, error handling and responsiveness were top priorities. By providing appropriate user feedback through alerts and fallback user interfaces, the system efficiently handles problems like inaccurate login credentials, server errors, and empty data sets.

In order to replicate real-world situations and preserve relational integrity among components like users, reservations, and sessions, the database schema was meticulously designed. Managing large datasets is made flexible and easy with the use of MongoDB collections for students, mentors, and reservations.

The platform effectively demonstrated all required features, such as mentor-student interaction, session scheduling and status tracking, dynamic content rendering according to user roles, user authentication, and administrative oversight. The system is presently in use in a testing environment and will soon be deployed in a live setting, where user input will help it become even more capable.

The project also identified some drawbacks, including the need for internet access to manage sessions, the lack of a specific mobile application, and the absence of real-time chat capabilities. The platform's usability will be improved in the future by addressing these issues.

A recommendation system for matching mentors, calendar synchronization capabilities with programs like Google Calendar, and the addition of feedback forms following sessions for continuous quality monitoring are some possible future improvements. Furthermore, applying AI to the analysis of data on student-mentor interactions may yield ideas for improving the mentorship process.

In conclusion, the project achieved its objectives and created a strong platform that can help mentorship programs become more modern. Its implementation has yielded incredibly instructive lessons that provide important insights into agile project management, system integration, full-stack development, and user-centered design.

REFERENCES

1. Chiranmai, S. M. Rajesh, G. Meghana, S. Rose and N. Jayapandian, "Post Covid Scenario Effective E-Mentoring System in Higher Education," 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), Uttarakhand, India, 2023
2. K. E. Holbert, L. L. Grable, A. Overbay and B. Nzekwe, "FREEDM ERC precollege programs: Motivating careers in the electric power industry," 2023 IEEE Power & Energy Society General Meeting, Vancouver, BC, Canada, 2024
3. K. Trenshaw, D. Rushton, E. E. Miskioğlu and P. Asare, "Work-in-Progress: Emergent Themes from "High Impact" Role Model and Mentor Narratives," 2020 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, 2022
4. H. Vasquez, A. A. Fuentes and J. Mouhamad, "Mentoring in Lower-Level Engineering Courses," 2022 IEEE Frontiers in Education Conference (FIE), Covington, KY, USA, 2022, pp..
5. J. Gadad, V. Talageri, P. Baligar and G. Joshi, "Peer-Mentoring in Design Projects in Project-Based Learning (PBL) at First-Year Engineering Course," 2021 IEEE Frontiers in Education Conference (FIE), Lincoln.
6. C. Andrade, P. Alves, J. E. Fernandes and F. Coutinho, "A web platform to support mentoring programs in higher education," 2020 15th Iberian Conference on Information Systems and Technologies (CISTI), Seville, Spain, 2020.
7. Author(s): Hafiz Iqbal,Subject(s): Education, Published by: Szkoła Główna Handlowa w Warszawie, Fundacja Promocji i Akredytacji Kierunków Ekonomicznych
8. Alhadlaq, A., Kharrufa, A., & Olivier, P. (2022). Exploring e-mentoring: co-designing & un-platforming. Behaviour & Information Technology.
9. Sarabipour, S., Hainer, S. J., Arslan, F. N., Furlong, E., Bielczyk, N., Jadavji, N. M., Shah, A. P., & Davla, S. (2022). Building and sustaining mentor interactions as a mentee. The FEBS Journal, 289(6), 1374-1384.
10. Online Mentoring: Programs and Practices" by Susan S. de Janusz and Shelley D. Godshalk

APPENDIX-A
PSUEDOCODE

```
import React, { useState, useEffect } from 'react';
import './overallMentors.css';
import DepartmentNavbar from '../Navbar/departmentNavbar';
import Search from '../Search/Search';
import { departments } from '../../../../utils';
import axios from 'axios';
import AssignStudentsModal from './modal';
const OverallMentors = () => {
  const [selectedDepartment, setSelectedDepartment] = useState(departments[0]);
  const [mentors, setMentors] = useState([]);
  const [loading, setLoading] = useState(false);
  const [isModalOpen, setIsModalOpen] = useState(false);
  const [selectedMentor, setSelectedMentor] = useState(null);
  useEffect(() => {
    fetchMentorsByDepartment(selectedDepartment.name);
  }, []);

  // Mock function to fetch mentors by department
  const fetchMentorsByDepartment = async (departmentName) => {
    try {
      setLoading(true);
      const response = await
axios.get(http://localhost:9000/get/all/mentors/${departmentName}, {
        headers: {
          "x-auth-token": localStorage.getItem('adminToken')
        }
      })
      if(response.data.error === "jwt must be provided") {
        alert("Please login to continue");
        navigate('/admin');
      }
      if(response.status === 200) {
        setTimeout(() => {
          setLoading(false);
          setMentors(response.data.mentors);
        }, 1000);
      }}
catch (error) {
      alert(error?.response?.data?.message ?? "Error fetching mentors");
      console.log(error)
    }
  };
  const handleDepartmentSelect = (department) => {
    setSelectedDepartment(department);
    fetchMentorsByDepartment(department.name);
  };
  console.log(mentors, "mentors");
  return (
    <div className="admin-mentors-container">
      <DepartmentNavbar
        departments={departments}
        selectedDepartment={selectedDepartment}
        handleDepartmentSelect={handleDepartmentSelect}
      />
      {selectedDepartment && (
        <section className="admin-mentors-section">
          <Search
            selectedDepartment={selectedDepartment}
            fetchStudentsByDepartment={fetchMentorsByDepartment}
            setStudents={setMentors}
            students={mentors}
            title="Mentor"
          />
    {loading ? (
     <div className="admin-mentors-loading">
       <div className="spinner"></div>
       <p>Loading mentors...</p>
     </div>
    ) : mentors && mentors?.length > 0 ? (
     <div className="admin-mentors-grid">
       {mentors?.map(mentor => (
         <div key={mentor.id} className="admin-mentor-card">
           <div className="admin-mentor-header">
             <img src={mentor.photo} alt={mentor.name} className="admin-mentor-
photo" />
             <div className="admin-mentor-basic-info">
               <h3>{mentor.name}</h3>
               <p><strong>Department:</strong> {mentor.department}</p>
             </div>
           </div>

           <div className="admin-mentor-details">
             <p><strong>Email:</strong> {mentor.email}</p>
             <p><strong>Phone:</strong> {mentor.phone}</p>
             <p><strong>Students Assigned:</strong> {mentor.studentsAssigned?.length ??
0}</p>
             <p><strong>Experience:</strong> {mentor.experience} years</p>
             <p><strong>Specialization:</strong> {mentor.specialization}</p>
             <button
               className="assign-students-btn"
               onClick={() => {
                 setSelectedMentor(mentor);
                 setIsModalOpen(true);
               }}
             >
               <i className="fas fa-user-plus"></i> Assign Students
             </button>
```

```
              </div>
            </div>
          ))}
        </div>
      )
 (
        <div className="admin-mentors-no-mentors">
          <p className='admin-mentors-no-mentors-text'>No mentors found</p>
        </div>
      )}
    </section>
   )}
   {isModalOpen && (
    <AssignStudentsModal
      isOpen={isModalOpen}
      onClose={() => setIsModalOpen(false)}
      mentor={selectedMentor}
    />
   )}
  </div>
 );
};
export default OverallMentors;

import './departmentNavbar.css';
const DepartmentNavbar = ({ departments, selectedDepartment, handleDepartmentSelect })
=> {
 return (
  <div className="admin-students-departments-section">
  <h2 className="admin-students-departments-title">Departments</h2>
  <div className="admin-students-departments-grid">
   {departments.map(department => (
    <div
      key={department.id}
```

```
        className={admin-students-department-card ${selectedDepartment?.id ===
department.id ? 'selected' : ''}}
        onClick={() => handleDepartmentSelect(department)}
       >
        <h3 className='admin-students-department-card-title'>{department.name}</h3>
       </div>
     ))}
    </div>
  </div>
 );
};
export default DepartmentNavbar;

import React, { useState } from 'react';
import axios from 'axios';

function Login() {
 const [email, setEmail] = useState('');
 const [password, setPassword] = useState('');

 const handleLogin = async (e) => {
  e.preventDefault();
  try {
   const res = await axios.post('http://localhost:5000/api/auth/login', { email,
password });
   alert('Login successful');
  } catch (error) {
   alert('Login failed');
  }
 };
```

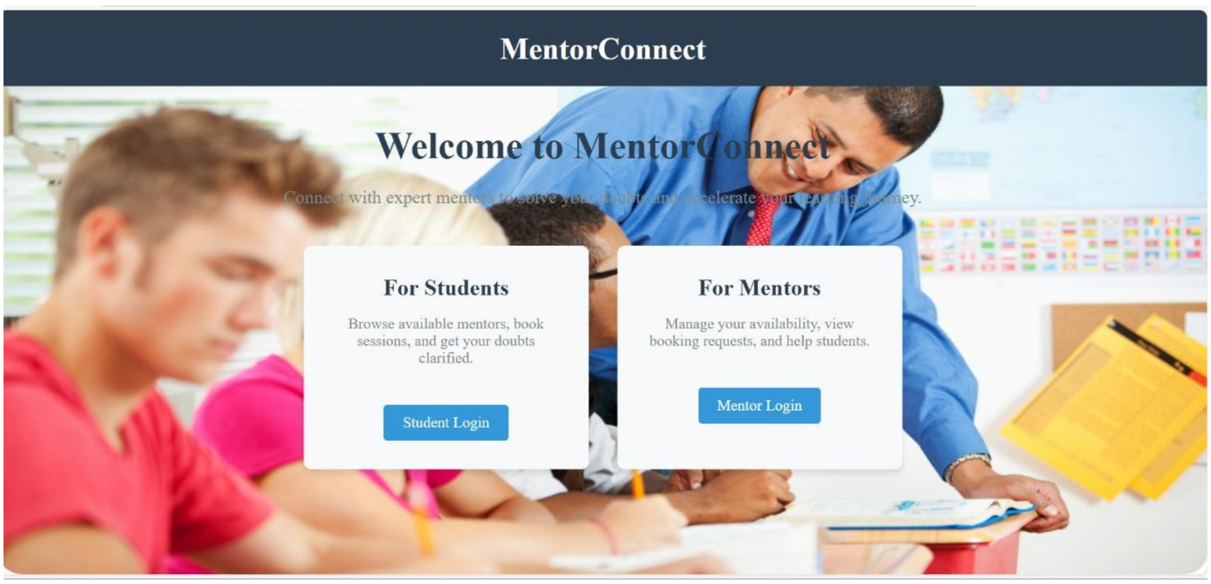
```
  return (
    <form onSubmit={handleLogin}>
      <h2>Login</h2>
      <input type="email" placeholder="Email" onChange={e =>
setEmail(e.target.value)} />
      <input type="password" placeholder="Password" onChange={e =>
setPassword(e.target.value)} />
      <button type="submit">Login</button>
    </form>
  );
}
export default Login;

import React, { useEffect, useState } from 'react';
import axios from 'axios';
import MentorCard from '../components/MentorCard';

function Mentors() {
  const [mentors, setMentors] = useState([]);

  useEffect(() => {
    axios.get('http://localhost:5000/api/mentors')
      .then(res => setMentors(res.data))
      .catch(err => console.error(err));
  }, []);

  return (
    <div>
```

```
      <h2>Available Mentors</h2>
      <div>
        {mentors.map(mentor => (
          <MentorCard key={mentor._id} mentor={mentor} />
        ))}
      </div>
    </div>
  );
}
export default Mentors;

import React, { useState, useEffect } from 'react';
import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-
dom';
import axios from 'axios';

function App() {
  return (
    <Router>
      <Navbar />
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/login" element={<Login />} />
        <Route path="/register" element={<Register />} />
        <Route path="/mentors" element={<Mentors />} />
      </Routes>
    </Router>
  );
}
```

```
// Navbar Component
function Navbar() {
  return (
    <nav style={styles.nav}>
      <h2 style={{ color: 'white' }}>MentorConnect</h2>
      <ul style={styles.navList}>
        <li><Link to="/" style={styles.link}>Home</Link></li>
        <li><Link to="/mentors" style={styles.link}>Mentors</Link></li>
        <li><Link to="/login" style={styles.link}>Login</Link></li>
        <li><Link to="/register" style={styles.link}>Register</Link></li>
      </ul>
    </nav>
  );
}

// Home Page
function Home() {
  return (
    <div style={styles.container}>
      <h1>Welcome to MentorConnect</h1>
      <p>Connect with mentors to shape your future!</p>
    </div>
  );
}

// Login Page
function Login() {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const handleLogin = async (e) => {
    e.preventDefault();
    try {
      await axios.post('http://localhost:5000/api/auth/login', { email, password });
      alert('Login successful!');
    } catch (error) {
      alert('Login failed.');
    }
  };

  return (
    <form onSubmit={handleLogin} style={styles.form}>
      <h2>Login</h2>
      <input type="email" placeholder="Email" value={email} onChange={e =>
setEmail(e.target.value)} required />
      <input type="password" placeholder="Password" value={password}
onChange={e => setPassword(e.target.value)} required />
      <button type="submit">Login</button>
    </form>
  );
}

// Register Page
function Register() {
  const [formData, setFormData] = useState({ name: '', email: '', password: '' });

  const handleRegister = async (e) => {
    e.preventDefault();
    try {
      await axios.post('http://localhost:5000/api/auth/register', formData);
      alert('Registration successful!');
    } catch (err) {
      alert('Registration failed.');
    }
  };

  return (
    <form onSubmit={handleRegister} style={styles.form}>
      <h2>Register</h2>
      <input type="text" placeholder="Name" value={formData.name}
onChange={e => setFormData({ ...formData, name: e.target.value })} required
/>
      <input type="email" placeholder="Email" value={formData.email}
onChange={e => setFormData({ ...formData, email: e.target.value })} required
/>
      <input type="password" placeholder="Password"
value={formData.password} onChange={e => setFormData({ ...formData,
password: e.target.value })} required />
      <button type="submit">Register</button>
    </form>
  );
}

// Mentors Page
function Mentors() {
  const [mentors, setMentors] = useState([]);
  useEffect(() => {
    axios.get('http://localhost:5000/api/mentors')
      .then(res => setMentors(res.data))
      .catch(err => console.error(err));
  }, []);

  return (
    <div style={styles.container}>
      <h2>Available Mentors</h2>
      {mentors.map(mentor => (
        <div key={mentor._id} style={styles.card}>
          <h3>{mentor.name}</h3>
          <p>Expertise: {mentor.expertise}</p>
          <p>Email: {mentor.email}</p>
        </div>
      ))}
    </div>
  );
}
```

APPENDIX-B

SCREENSHOTS


MentorConnect

Welcome to MentorConnect

Connect with expert mentors to solve your doubts and accelerate your learning journey.

For Students

Browse available mentors, book sessions, and get your doubts clarified.

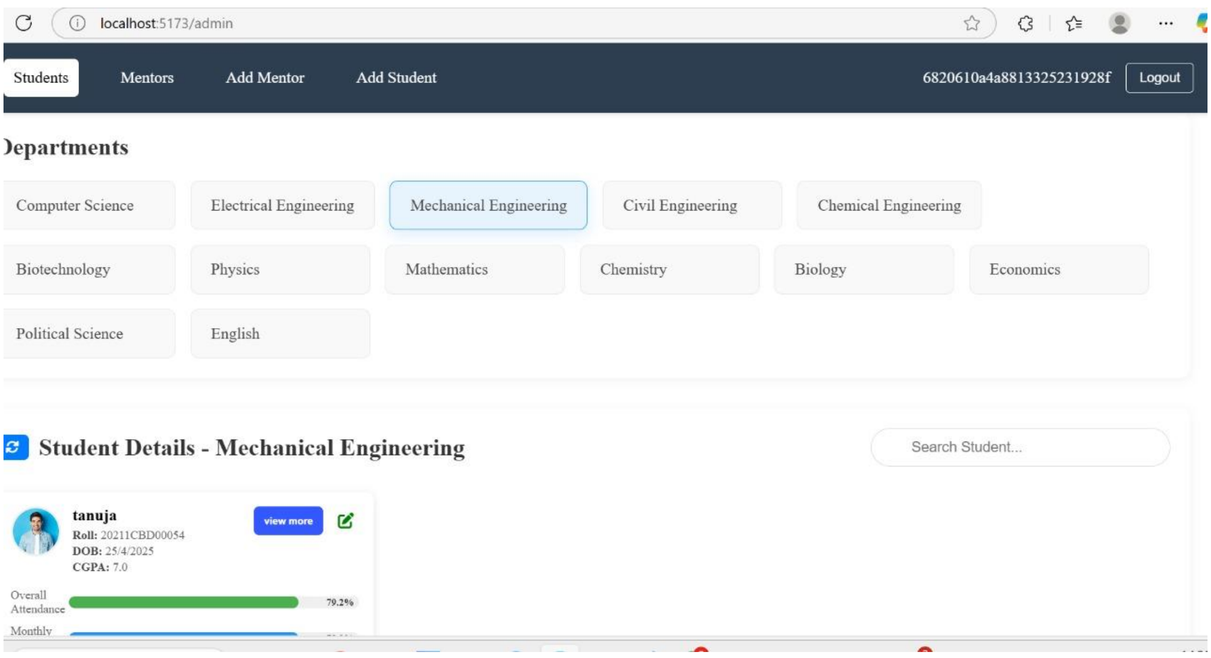
Student Login

For Mentors

Manage your availability, view booking requests, and help students.

Mentor Login

Fig - Home page of the mentor connect website


localhost:5173/admin

Students Mentors Add Mentor Add Student 6820610a4a8813325231928f Logout

Departments

Computer Science | Electrical Engineering | Mechanical Engineering | Civil Engineering | Chemical Engineering

Biotechnology | Physics | Mathematics | Chemistry | Biology | Economics

Political Science | English

Student Details - Mechanical Engineering

Search Student...

tanuja

Roll: 20211CBD00054

DOB: 25/4/2025

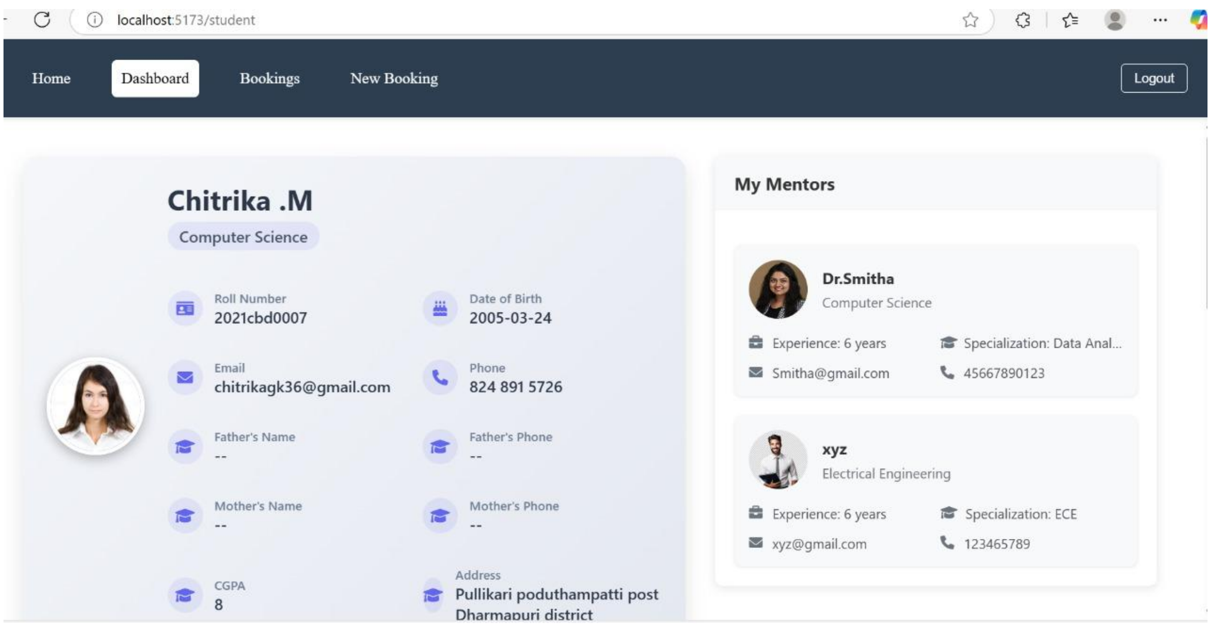
CGPA: 7.0

view more

Overall Attendance 79.2%

Monthly

Fig - List of students along with their departments displayed on the mentor connect website



Chitrika .M
Computer Science

Roll Number
2021cbd0007

Date of Birth
2005-03-24

Email
chitrikagk36@gmail.com

Phone
824 891 5726

Father's Name
--

Father's Phone
--

Mother's Name
--

Mother's Phone
--

CGPA
8

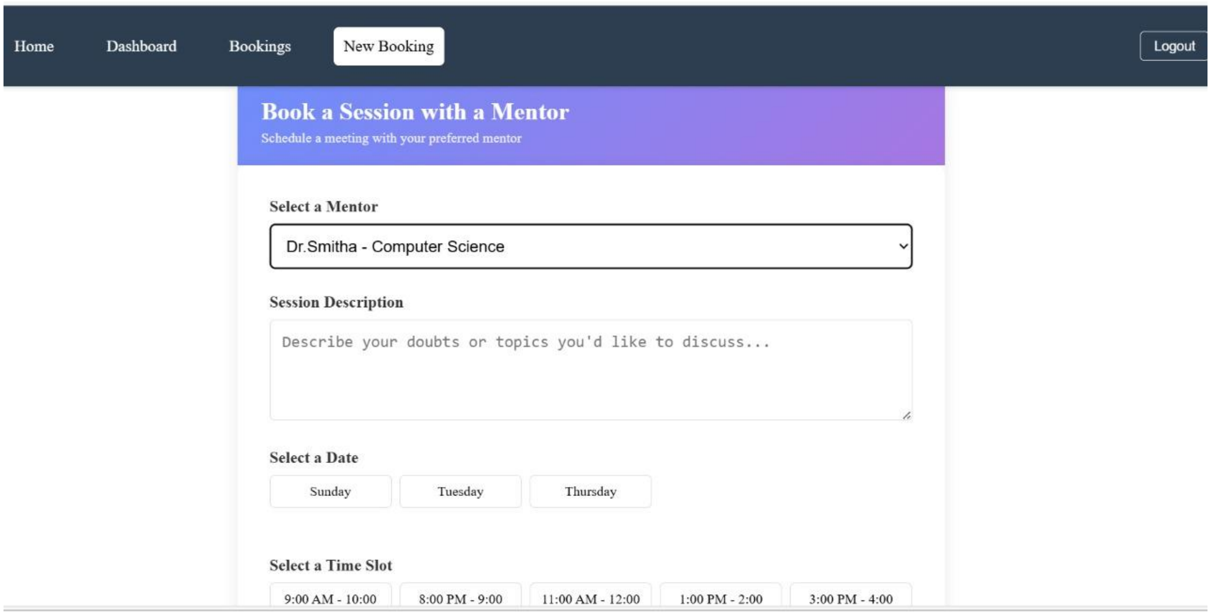
Address
Pullikari poduthampatti post
Dharmapuri district

My Mentors

Dr.Smitha
Computer Science
Experience: 6 years
Specialization: Data Anal...
Smitha@gmail.com
45667890123

xyz
Electrical Engineering
Experience: 6 years
Specialization: ECE
xyz@gmail.com
123465789

Fig - Student details along with the assigned mentor



Book a Session with a Mentor
Schedule a meeting with your preferred mentor

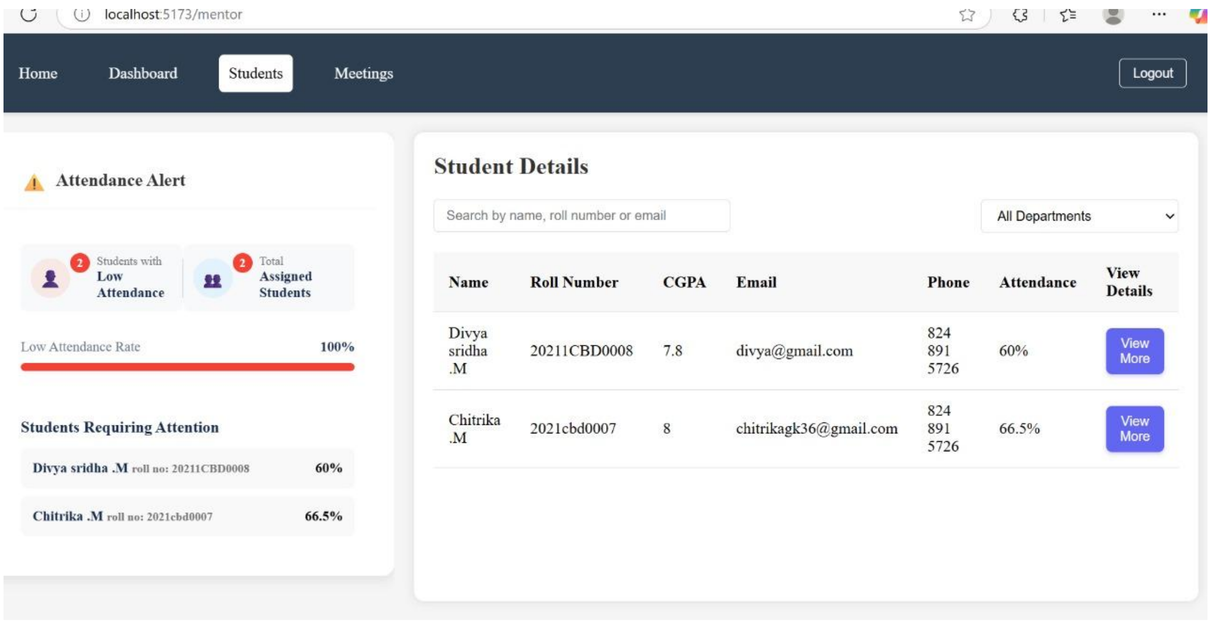
Select a Mentor
Dr.Smitha - Computer Science

Session Description
Describe your doubts or topics you'd like to discuss...

Select a Date
Sunday
Tuesday
Thursday

Select a Time Slot
9:00 AM - 10:00
8:00 PM - 9:00
11:00 AM - 12:00
1:00 PM - 2:00
3:00 PM - 4:00

Fig - Drop down menu for the mentors and mentee to book the meeting slot, day and time of the session


localhost:5173/mentor

Home | Dashboard | Students | Meetings | Logout

Attendance Alert

2 Students with Low Attendance

2 Total Assigned Students

Low Attendance Rate 100%

Students Requiring Attention

Divya sridha .M roll no: 20211CBD0008 60%

Chitrika .M roll no: 2021cbd0007 66.5%

Student Details

Search by name, roll number or email

All Departments

Name	Roll Number	CGPA	Email	Phone	Attendance	View Details
Divya sridha .M	20211CBD0008	7.8	divya@gmail.com	824 891 5726	60%	View More
Chitrika .M	2021cbd0007	8	chitrikagk36@gmail.com	824 891 5726	66.5%	View More

Fig - List of students and their particulars displayed to the mentor after logging in with the credentials

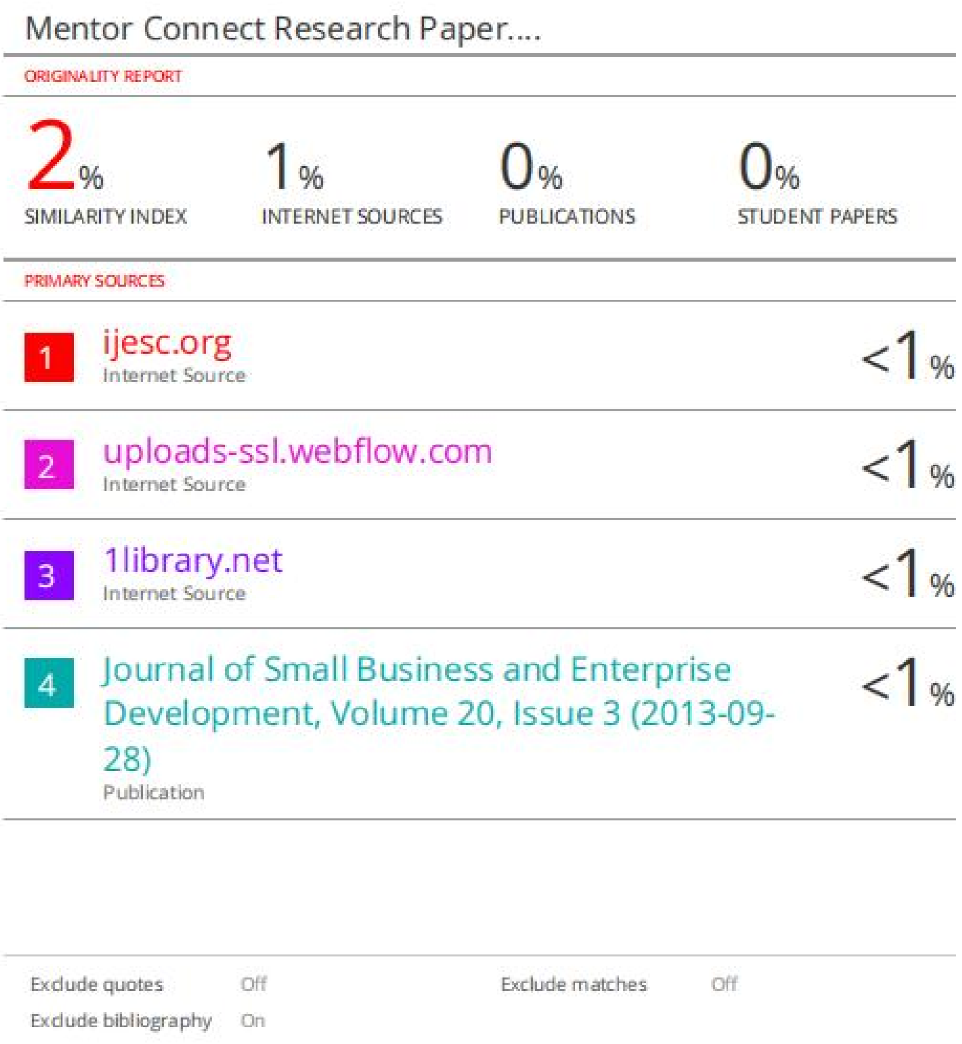
APPENDIX-C

ENCLOSURES

1. Paper if formulated and Communicated.

2. Similarity Index / Plagiarism Check report clearly showing the Percentage (%). No need for a page-wise explanation.

PLAGARISM REPORT OF PUBLISHED RESEARCH PAPER



Mentor Connect Research Paper....

ORIGINALITY REPORT

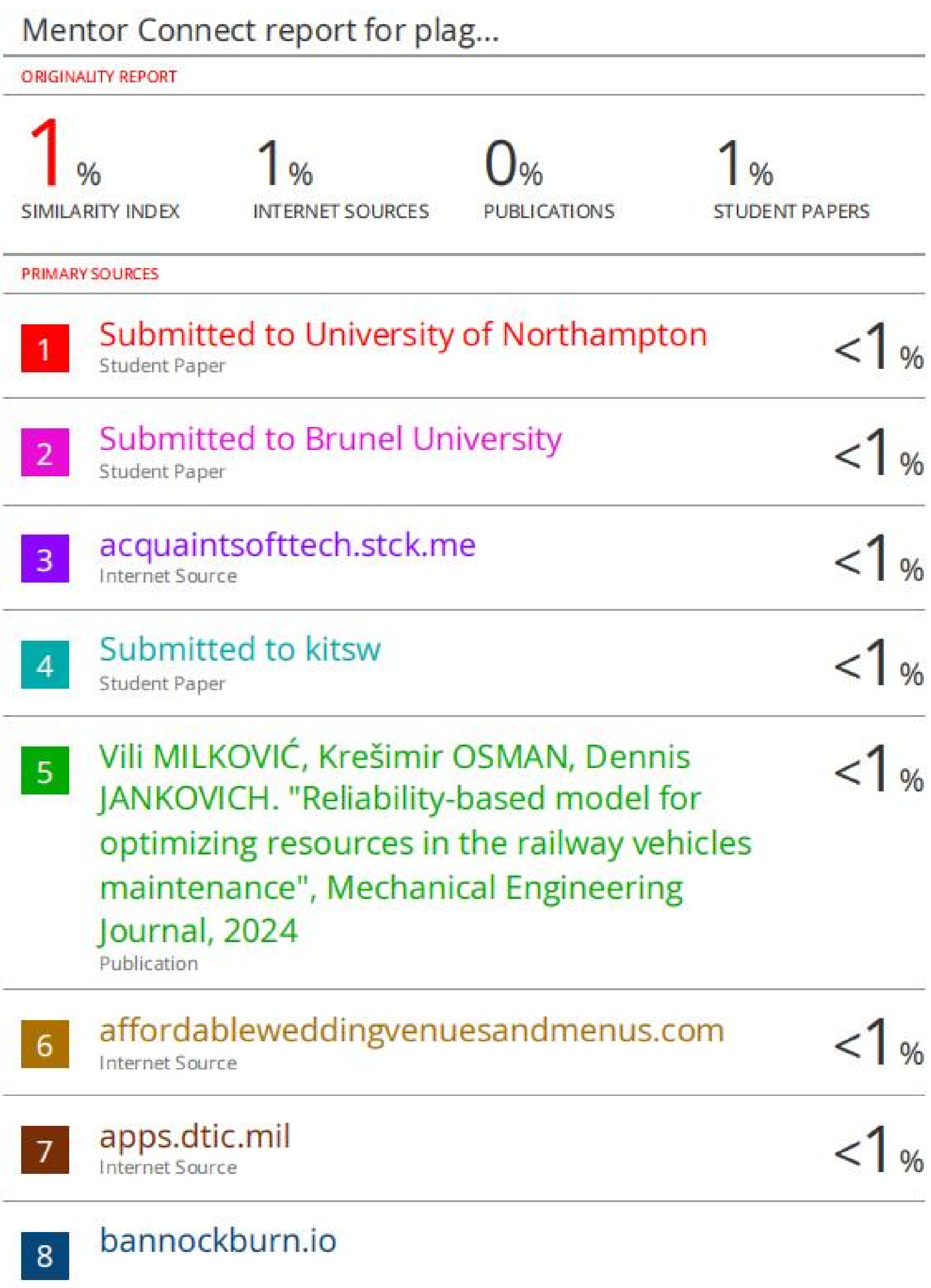
2%	1%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ijesc.org Internet Source	<1%
2	uploads-ssl.webflow.com Internet Source	<1%
3	1library.net Internet Source	<1%
4	Journal of Small Business and Enterprise Development, Volume 20, Issue 3 (2013-09-28) Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

PLAGARISM REPORT OF PROJECT REPORT



Mentor Connect report for plag...

ORIGINALITY REPORT

1 %	1 %	0 %	1 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Northampton Student Paper	<1 %
2	Submitted to Brunel University Student Paper	<1 %
3	acquaintsofttech.stck.me Internet Source	<1 %
4	Submitted to kitsw Student Paper	<1 %
5	Vili MILKOVIĆ, Krešimir OSMAN, Dennis JANKOVICH. "Reliability-based model for optimizing resources in the railway vehicles maintenance", Mechanical Engineering Journal, 2024 Publication	<1 %
6	affordableweddingvenuesandmenus.com Internet Source	<1 %
7	apps.dtic.mil Internet Source	<1 %
8	bannockburn.io	

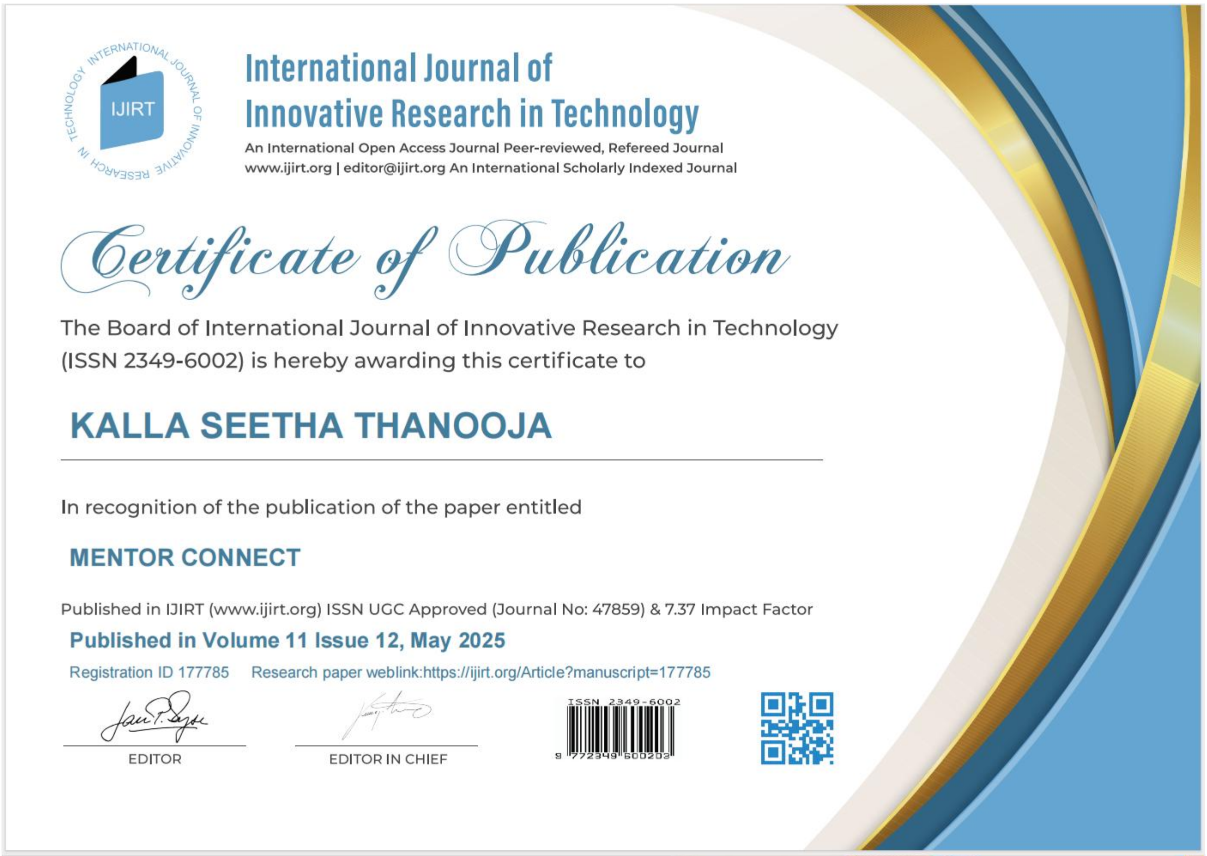
MAPPING THE PROJECT TO SDG MAPPING

SUSTAINABLE DEVELOPMENT GOALS
17 GOALS TO TRANSFORM OUR WORLD
1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE, JUSTICE AND STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS
SUSTAINABLE DEVELOPMENT GOALS

The Mentor Connect Project directly supports **“SDG - 4 Quality Education”** by enabling learners to access personalized guidance and skill development through experienced mentors, enhancing both academic and professional growth.

It also aligns with **“SDG - 8 Decent Work and Economic Growth”** by fostering career readiness, improving employability, and supporting entrepreneurship through targeted mentorship programs. By bridging the gap between education and industry, the platform ensures that individuals are better prepared for evolving job markets. Overall, it empowers users with knowledge and opportunities for sustainable economic advancement.

PUBLICATION CERTIFICATES



International Journal of Innovative Research in Technology

An International Open Access Journal Peer-reviewed, Refereed Journal
www.ijirt.org | editor@ijirt.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of International Journal of Innovative Research in Technology (ISSN 2349-6002) is hereby awarding this certificate to

KALLA SEETHA THANOOJA

In recognition of the publication of the paper entitled

MENTOR CONNECT

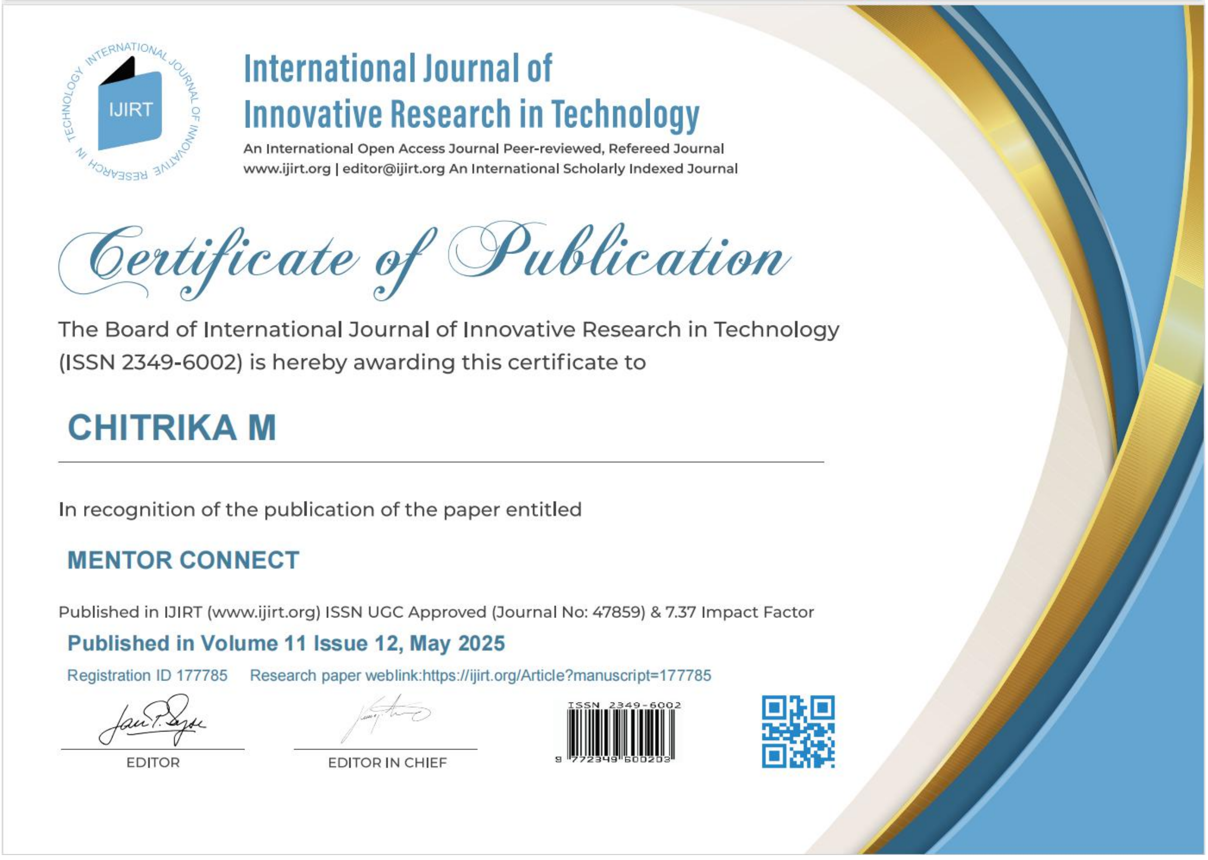
Published in IJIRT (www.ijirt.org) ISSN UGC Approved (Journal No: 47859) & 7.37 Impact Factor

Published in Volume 11 Issue 12, May 2025

Registration ID 177785 Research paper weblink:https://ijirt.org/Article?manuscript=177785

EDITOR EDITOR IN CHIEF

ISSN 2349-6002



International Journal of Innovative Research in Technology

An International Open Access Journal Peer-reviewed, Refereed Journal
www.ijirt.org | editor@ijirt.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of International Journal of Innovative Research in Technology (ISSN 2349-6002) is hereby awarding this certificate to

CHITRIKA M

In recognition of the publication of the paper entitled

MENTOR CONNECT

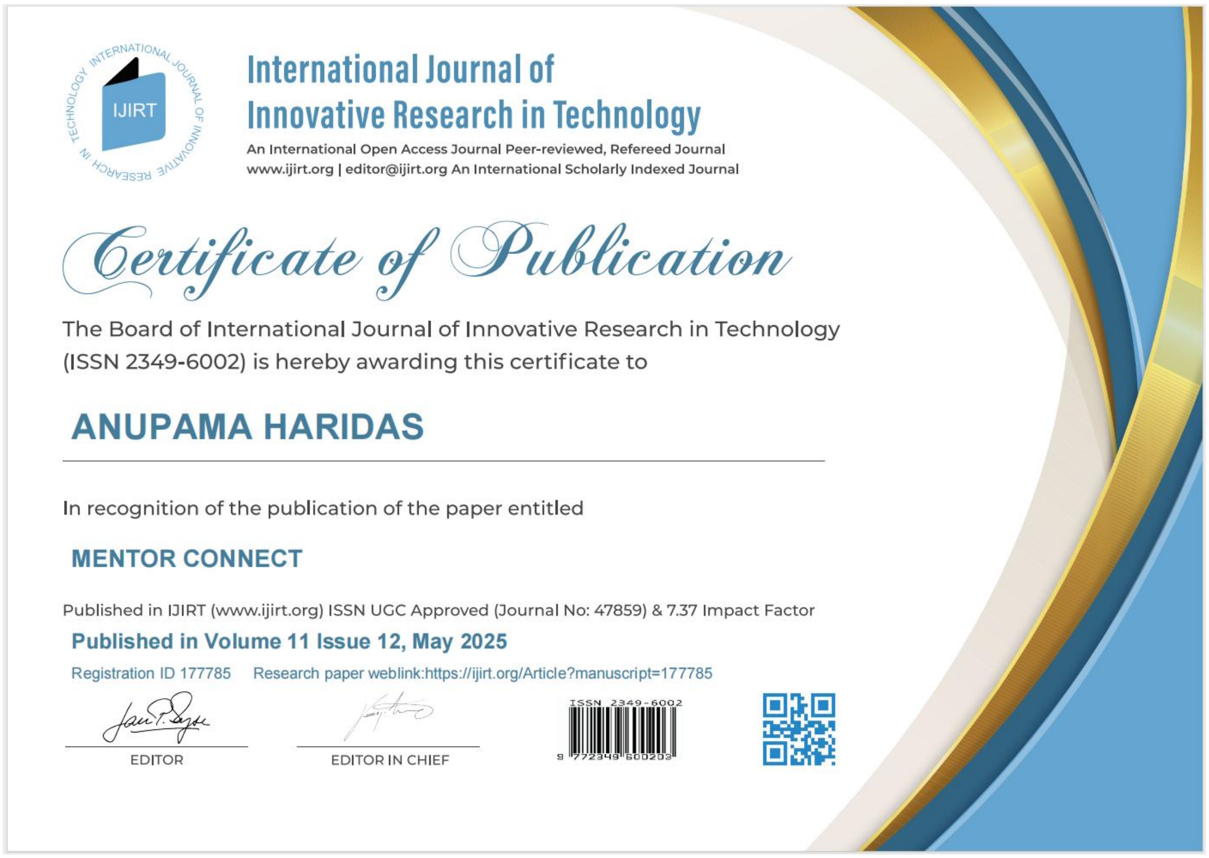
Published in IJIRT (www.ijirt.org) ISSN UGC Approved (Journal No: 47859) & 7.37 Impact Factor

Published in Volume 11 Issue 12, May 2025

Registration ID 177785 Research paper weblink:https://ijirt.org/Article?manuscript=177785

EDITOR EDITOR IN CHIEF

ISSN 2349-6002



International Journal of
Innovative Research in Technology

An International Open Access Journal Peer-reviewed, Refereed Journal
www.ijirt.org | editor@ijirt.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of International Journal of Innovative Research in Technology (ISSN 2349-6002) is hereby awarding this certificate to

ANUPAMA HARIDAS

In recognition of the publication of the paper entitled

MENTOR CONNECT

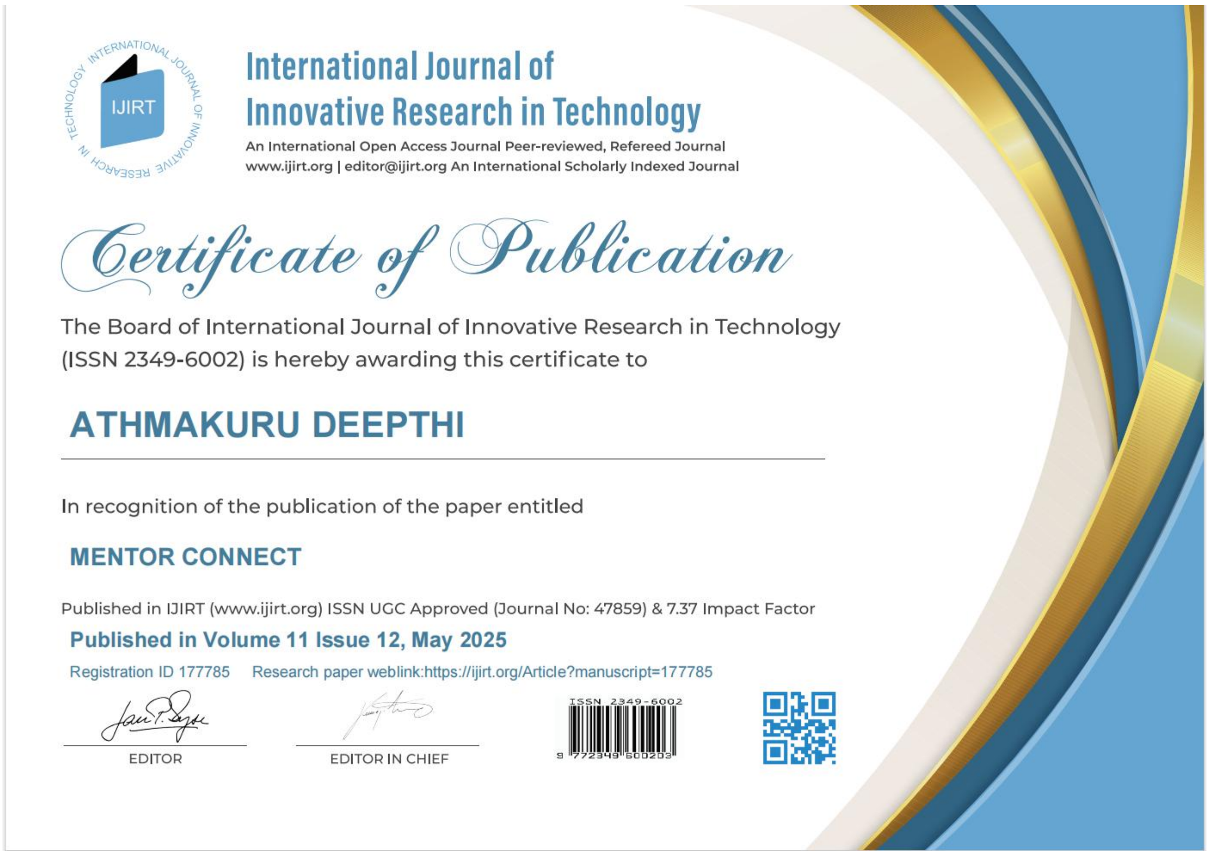
Published in IJIRT (www.ijirt.org) ISSN UGC Approved (Journal No: 47859) & 7.37 Impact Factor

Published in Volume 11 Issue 12, May 2025

Registration ID 177785 Research paper weblink:https://ijirt.org/Article?manuscript=177785

EDITOR EDITOR IN CHIEF

ISSN 2349-6002



International Journal of
Innovative Research in Technology

An International Open Access Journal Peer-reviewed, Refereed Journal
www.ijirt.org | editor@ijirt.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of International Journal of Innovative Research in Technology (ISSN 2349-6002) is hereby awarding this certificate to

ATHMAKURU DEEPTHI

In recognition of the publication of the paper entitled

MENTOR CONNECT

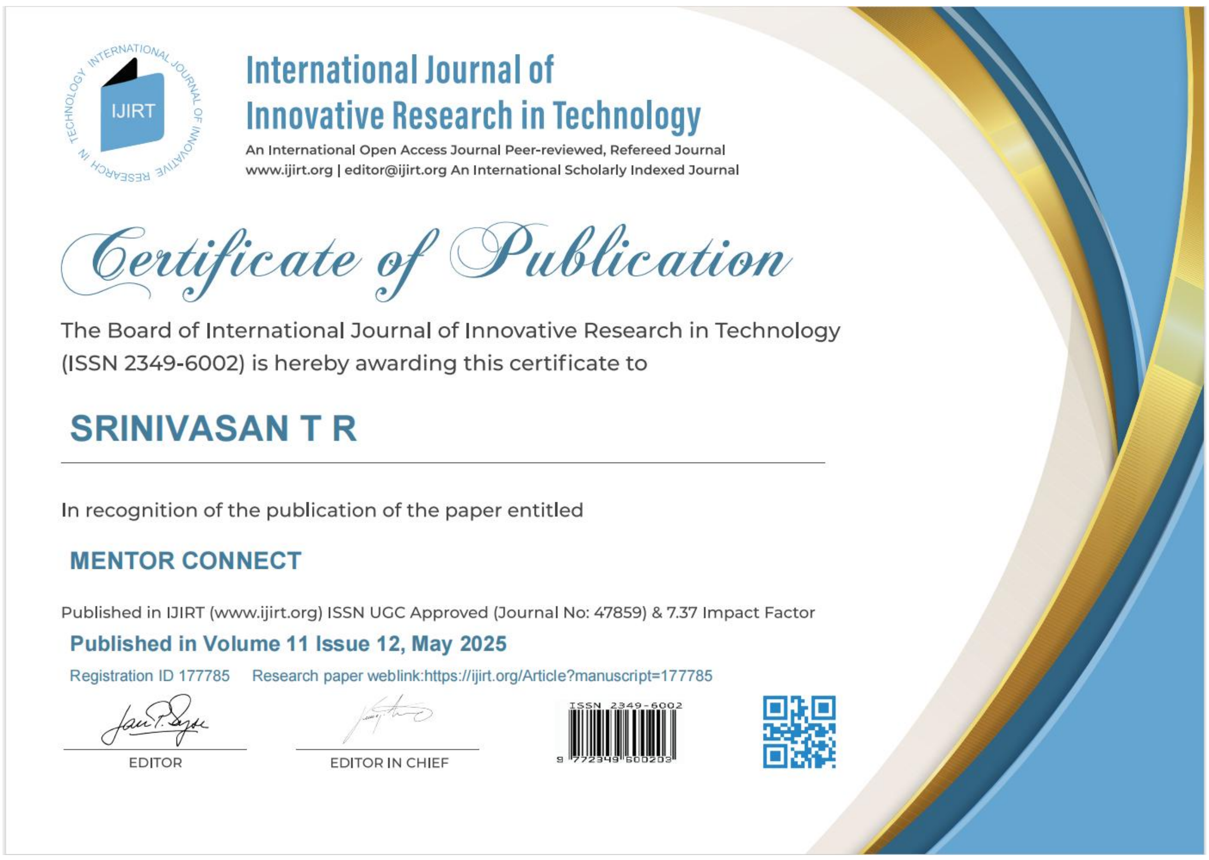
Published in IJIRT (www.ijirt.org) ISSN UGC Approved (Journal No: 47859) & 7.37 Impact Factor

Published in Volume 11 Issue 12, May 2025

Registration ID 177785 Research paper weblink:https://ijirt.org/Article?manuscript=177785

EDITOR EDITOR IN CHIEF

ISSN 2349-6002



International Journal of
Innovative Research in Technology

An International Open Access Journal Peer-reviewed, Refereed Journal
www.ijirt.org | editor@ijirt.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of International Journal of Innovative Research in Technology (ISSN 2349-6002) is hereby awarding this certificate to

SRINIVASAN T R

In recognition of the publication of the paper entitled

MENTOR CONNECT

Published in IJIRT (www.ijirt.org) ISSN UGC Approved (Journal No: 47859) & 7.37 Impact Factor

Published in Volume 11 Issue 12, May 2025

Registration ID 177785 Research paper weblink:https://ijirt.org/Article?manuscript=177785

[illegible]
EDITOR

[illegible]
EDITOR IN CHIEF

ISSN 2349-6002