

"Mentora – A Mentorship Platform for Learners and Professionals"

*A
project report
submitted
in partial fulfillment
for the award of the Degree of
Bachelor of Computer Application
in Department of IT&CS*



Submitted To:
Dr. Avinash Panwar

Submitted By:
Nishit Jain
Enrolment No.: 227793284

Department of IT & CS
University College of Science

Mohanlal Sukhadia University

Month, 2025

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in the project, entitled **Mentora – A Mentorship Platform for Learners and Professionals** in partial fulfilment for the award of Degree of Bachelor of Computer Application in Deptt. of **Informatics & Computational Sciences, and submitted to the Department of IT & CS, University College of Science**, Mohanlal Sukhadia University is a record of my own investigations carried under the Guidance of Mr. Jitesh Kumar, Department of Informatics & Computational Sciences, University College of Science .

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

Nishit Jain

Bachelor's of Computer Application

Enrolment No.: 227793284

University College of Science, Udaipur

Counter Signed by

Mr. Mahesh Choudhary

.....

CERTIFICATE

This is to certify that Nishit Jain of Semester VI B.C.A. 2024-25 has presented a major project titled "**Mentora – A Mentorship Platform for Learners and Professionals**" in partial fulfilment for the award of the degree of Bachelor of Computer Application under Mohanlal Sukhadia University, Udaipur.

Date:

Mr. Mahesh Choudhary - Project Co-ordinator Supervisor

ACKNOWLEDGEMENT

I take this opportunity to express my gratitude to all those people who have been directly and indirectly with me during the completion of this project. I pay thank to <supervisor name> who has given guidance and a light to me during this major project. His versatile knowledge about title name has eased me in the critical times during the span of this major project.

I acknowledge here out debt to those who contributed significantly to one or more steps. I take full responsibility for any remaining sins of omission and commission.

Nishit Jain B.C.A. Semester VI

CONTENTS

Candidate's Declaration	i
Certificate	ii
Acknowledgement	iii
Contents	iv
List of Figures.....	vi
List of Tables	vii
Chapter 1: Introduction.....	1
1.1 Introduction of Project	1
1.2 Aim of Project.....	1
1.3 Objectives of Projects	2
1.4 Hardware and Software requirements.....	4
Chapter 2: Technology Used	6
2.1 Frontend Technology	6
2.2 Backend Technology	8
2.3 Frontend and Backend Tools	9
Chapter 3: System Analysis and System Study	13
3.1 Project analysis	13
3.2 Feasibility study.....	14
3.3 Cost Benefit Analysis	16
Chapter 4: Project Design	18
4.1 Data Dictionary	18

4.2 Data Flow Diagrams	26
4.3 ER Diagram	29
4.4 Database Diagram	31
4.4 Snapshot.....	32
Chapter 5: Mechanisms Used	49
5.1 Testing mechanism	49
5.2 Implementation mechanism.....	50
Chapter 6: Future Scope of the Project	51
Chapter 7: References and Bibliography	57
Chapter 8: Application Code	58

LIST OF FIGURES

Fig. 4.2.1: Level 0 Data Flow Diagram – Overall System Architecture.	26
Fig. 4.2.2: Level 1 Data Flow Diagram – Detailed System Architecture.	26
Fig. 4.2.3: Level 2 Data Flow Diagram – User Module Workflow.	27
Fig. 4.2.4: Level 2 Data Flow Diagram – Learner Module Workflow.	28
Fig. 4.2.5: Level 2 Data Flow Diagram – Mentor Module Workflow.	28
Fig. 4.3: Entity Relation Diagram.....	29
Fig. 4.4: Database Schema Diagram.	31
Fig. 4.5.1: Landing page screenshot.	32
Fig. 4.5.2: Browse Mentor Page.	34
Fig. 4.5.3: About Page	36
Fig. 4.5.4: Contact Page.	37
Fig. 4.5.5.1: Login Page.	38
Fig. 4.5.5.2: Registration Page.	39
Fig. 4.5.6.1: Learner Dashboard.	40
Fig. 4.5.6.2: Learner's Session Page.	41
Fig. 4.5.6.3: Learner's Favorites Page.	42
Fig. 4.5.6.4: Learner's Settings Page.	43
Fig. 4.5.6.5: Book Session Page.	45
Fig. 4.5.7.1: Mentor Dashboard Page	46
Fig. 4.5.7.2: Mentor Sessions Page.....	47
Fig. 4.5.7.3: Mentor Earnings Page	48

LIST OF TABLES

Table 3.3.1 Cost Analysis Table	17
Table 3.3.2 Benefits Analysis Table	17
Table 4.1.1 Table Mentee (Learner) User Data Dictionary	18
Table 4.1.2 Table Mentor User Data Dictionary	18
Table 4.1.3 Table SocialLinks Data Dictionary	19
Table 4.1.4 Table work_experience Data Dictionary	20
Table 4.1.5 Table education Data Dictionary	20
Table 4.1.6 Table expertise Data Dictionary	21
Table 4.1.7 Table reviews Data Dictionary	21
Table 4.1.8 Table bookings Data Dictionary	22
Table 4.1.9 Table payments Data Dictionary	23
Table 4.1.10 Table schedules Data Dictionary	24

Chapter 1

INTRODUCTION

1.1 Introduction of the Mentora Project:

In today's fast-paced and competitive world, young learners often struggle to find personalized guidance to shape their careers and academic paths. **Mentora** is a web-based mentorship platform designed to bridge this gap by connecting learners with experienced mentors from diverse domains. The platform offers an intuitive and accessible interface where students can search for mentors, book one-on-one sessions, manage bookings, and leave feedback. Mentors, in turn, can manage their availability, handle upcoming sessions, and update their profiles.

Initially developed as a college-level project, Mentora is built with scalability in mind, with future plans to integrate features like real-time chat, video conferencing, and personalized AI-driven mentor recommendations. The platform aims to democratize access to mentorship by making expert guidance more structured, convenient, and available to everyone.

1.2 Aim of the Project :-

The aim of this project is to design and develop a responsive, user-centric mentorship platform called **Mentora** that connects students or young learners with experienced mentors from various domains. The platform intends to bridge the gap between those seeking guidance and those willing to provide it by offering features such as user registration, mentor discovery, session booking, and profile management. Mentora focuses on providing an intuitive and accessible interface while ensuring the system is scalable, secure, and ready for future enhancements like real-time communication and intelligent mentor suggestions.

1.3 Objectives of the Project:

1.3.1 Primary Development Objectives

- **To develop a comprehensive user authentication and authorization system** that supports secure registration, login, and role-based access control for three distinct user types: learners seeking guidance, mentors offering expertise, and administrators managing the platform ecosystem.
- **To create an intuitive and personalized dashboard interface** that enables learners to efficiently view upcoming mentorship sessions, track session history, manage booking preferences, provide detailed feedback and ratings for completed sessions, and monitor their learning progress over time.
- **To implement an advanced mentor discovery and search system** featuring intelligent filtering capabilities based on multiple criteria including domain expertise, professional experience level, availability windows, user ratings and reviews, pricing ranges, and specialized skills or certifications.
- **To establish a flexible session booking and scheduling system** that allows learners to seamlessly reserve mentorship sessions by selecting from available time slots, choosing session durations, specifying session objectives, and handling scheduling conflicts or modifications efficiently.
- **To provide mentors with a comprehensive profile management platform** enabling them to maintain detailed professional profiles, update biographical information and credentials, showcase their expertise and achievements, set availability schedules, manage pricing structures, and track their mentoring statistics.

1.3.2 Technical and Design Objectives

- **To implement a fully responsive and accessible user interface design** that delivers consistent user experiences across desktop computers, tablets, and mobile devices while adhering to web accessibility guidelines (WCAG) and modern UI/UX principles.

- **To architect a modular, scalable, and maintainable codebase** using modern web development frameworks and design patterns that facilitate future feature integration such as real-time messaging, video conferencing capabilities, AI-driven mentor recommendations, and advanced analytics.
- **To ensure optimal user experience through consistent design systems** implementing professional visual aesthetics, intuitive navigation patterns, clear information hierarchy, and user-friendly interaction flows that minimize learning curves for all user types.
- **To develop a robust backend infrastructure with efficient data management** incorporating secure database design, optimized query performance, reliable session handling, automated backup systems, and scalable architecture to support growing user bases.
- **To implement comprehensive security and privacy protection measures** ensuring secure user authentication, encrypted data transmission, GDPR compliance for data handling, protection against common web vulnerabilities, and transparent privacy policies.

1.3.3 Functional and User Experience Objectives

- **To create an effective feedback and rating system** that enables learners to provide structured feedback on mentorship sessions, rate mentor performance across multiple dimensions, and help maintain quality standards while assisting future learners in mentor selection.
- **To develop efficient notification and communication systems** that keep users informed about upcoming sessions, booking confirmations, schedule changes, system updates, and important platform announcements through multiple channels.
- **To implement comprehensive session management features** including session preparation tools, resource sharing capabilities, session notes and summaries, follow-up task tracking, and progress monitoring to maximize the value of mentorship interactions.

- **To establish quality assurance and content moderation systems** ensuring mentor profile authenticity, maintaining appropriate session content standards, handling dispute resolution, and fostering a safe and professional learning environment.
- **To create analytics and reporting capabilities** providing users with insights into their mentorship journey, session effectiveness, skill development progress, and platform usage patterns while offering administrators comprehensive platform performance metrics.

1.4 Hardware and Software Requirements:

Hardware Requirements

For Developers (Development Environment):

- **Processor:** Intel i5 or equivalent AMD Ryzen 5 (minimum)
- **RAM:** 8 GB (minimum), 16 GB recommended
- **Storage:** SSD with at least 50 GB of free space
- **Display:** 13" or larger screen with at least 1080p resolution
- **Internet Connectivity:** Stable broadband connection for API access, version control, and package installation

For End-Users (Web Application Usage):

- **Device:** Laptop, desktop, tablet, or smartphone with modern browser support
- **Processor:** Dual-core processor or higher
- **RAM:** Minimum 4 GB
- **Internet Connectivity:** Stable internet connection (preferably 2 Mbps or above)
- **Screen Resolution:** Minimum 720p resolution recommended for proper UI rendering

Software Requirements

Development Environment:

- **Operating System:** Windows 10/11, macOS, or Linux
- **Code Editor:** Visual Studio Code (or any modern IDE)
- **Runtime Environment:**
 - Node.js (v18 or above)
 - npm (v9 or above)
- **Version Control:** Git with access to GitHub or other repository platforms
- **Database:** MongoDB (local or cloud-hosted via MongoDB Atlas)
- **API Client:** Postman for API testing
- **Package Managers & Tools:**
 - Tailwind CSS (via PostCSS setup)
 - ReactJS for frontend
 - Express.js for backend
 - dotenv for environment management
 - Mongoose for MongoDB interactions

End-User Environment:

- **Web Browser:** Chrome, Firefox, Edge, or Safari (latest versions recommended)
- **No installation needed:** The app is designed to run on browsers, so no local software installations are required for users.

Chapter 2

TECHNOLOGY USED

The Mentora platform is built using a modern full-stack web development approach with React.js, Express.js, and MongoDB. This technology stack ensures scalability, maintainability, and optimal user experience while following industry best practices.

2.1 Frontend Technology

2.1.1 React.js (JavaScript Library)

React.js serves as the core frontend framework, providing:

- **Component-Based Architecture:** Reusable UI components for consistent design across the platform, including standardized mentor cards, session booking forms, and navigation elements that maintain visual coherence throughout the application.
- **Virtual DOM:** Efficient rendering and performance optimization that ensures smooth user interactions, particularly important for real-time features like live mentor availability updates and dynamic filtering of search results.
- **Rich Ecosystem:** Extensive library support and community resources that accelerate development and provide battle-tested solutions for common functionality requirements.
- **React Router DOM:** Seamless client-side routing and navigation management that enables:
 - **Single Page Application (SPA) Experience:** Smooth transitions between different platform sections without full page reloads
 - **Dynamic Route Handling:** Parameterized routes for mentor profiles, session details, and user dashboards
 - **Protected Routes:** Authentication-based access control for mentor and learner-specific areas

- **Nested Routing:** Hierarchical navigation structure supporting complex dashboard layouts with sub-sections
- **Browser History Management:** Proper back/forward button functionality and bookmarkable URLs for improved user experience
- **State Management Integration:** Seamless integration with Context API for maintaining route-specific state and user session persistence across navigation.

Implementation: Interactive dashboards with multi-section navigation, dynamic session booking with step-by-step routing, real-time profile management with nested route structures, responsive mentor discovery system with filtered URL parameters, and protected authentication flows with redirect handling.

2.1.2 Tailwind CSS

Tailwind CSS enables rapid UI development through:

- **Utility Classes:** Pre-built styling components for faster development, eliminating the need for custom CSS files while maintaining design consistency across mentor cards, booking interfaces, and dashboard components.
- **Responsive Design:** Mobile-first approach with built-in breakpoints that automatically adapts the platform for desktop, tablet, and mobile devices, ensuring optimal user experience across all screen sizes for both mentors and learners.
- **Customization:** Easy theme implementation and brand consistency through:
 - **Custom Color Palette:** Platform-specific branding colors for buttons, backgrounds, and accent elements
 - **Component Variants:** Standardized styling for different button types, form inputs, and card layouts
- **Design System Integration:** Facilitates creation of a cohesive design language with reusable component patterns for navigation bars, mentor profiles, session booking flows, and dashboard analytics displays.

Implementation Benefits: Rapid prototyping of new features, consistent spacing and typography across all platform sections, seamless responsive behavior for mentor discovery filters and booking calendars, and maintainable styling architecture that scales with platform growth.

2.1.3 Context API (State Management)

Purpose: Global state management across React components

The Context API enables:

- **Centralized State:** Manages user authentication, session data, and application state
- **Prop Drilling Prevention:** Eliminates the need to pass props through multiple component layers
- **Performance Optimization:** Selective re-rendering of components based on state changes
- **Scalability:** Easy integration with additional state management solutions if needed

2.2 Backend Technology

The backend of Mentora provides the server-side logic, API endpoints, and database interactions necessary for the application's functionality.

2.2.1 Node.js

Purpose: Server-side JavaScript runtime

- **High Performance:** Event-driven, non-blocking I/O model for handling concurrent requests
- **JavaScript Everywhere:** Unified development experience across frontend and backend
- **Rich Package Ecosystem:** Access to npm packages for extended functionality

- **Scalability:** Excellent performance for I/O-intensive applications
- **Real-time Capabilities:** Support for WebSocket connections and real-time features

2.2.2 Express.js

Purpose: Backend API development and server management

- **Minimalist Framework:** Lightweight and flexible web application framework
- **Middleware Support:** Extensive middleware ecosystem for authentication, logging, and security
- **RESTful API Development:** Easy creation of REST endpoints for frontend communication
- **Error Handling:** Robust error handling and debugging capabilities
- **Template Engine Support:** Integration with various view engines if needed

2.2.3 MongoDB (Database)

Purpose: Primary data storage and management

- **Document-Based Storage:** Flexible schema design suitable for evolving data requirements
- **Horizontal Scalability:** Built-in sharding and replication capabilities
- **JSON-Like Documents:** Native JavaScript object handling for seamless integration
- **Indexing:** Advanced indexing capabilities for optimized query performance
- **Aggregation Framework:** Powerful data processing and analytics capabilities

2.3 Frontend and Backend Tools

2.3.1 Visual Studio Code

Visual Studio Code serves as the primary integrated development environment (IDE) for MessMate development, offering:

- **Extensions:** Key extensions utilized include:
 - Prettier - For consistent code formatting
 - React Developer Tools - For React component debugging
 - MongoDB - For database connection and navigate database
 - GitLens - For enhanced Git integration
- **Integrated Terminal:** The built-in terminal facilitates running npm scripts, database commands, and Git operations without leaving the editor.

2.3.2 API Testing Tools

- **Postman:** Postman served as the main API development and testing platform throughout the project. This industry-standard tool was utilized to build comprehensive test suites for all backend endpoints, helping ensure reliable API functionality across the application.
- **Thunder Client:** As a lightweight alternative, Thunder Client (a Visual Studio Code extension) was employed for quick API testing directly within the development environment. This tool provided convenience for rapid testing during active development sessions without requiring context switching.

2.3.3 Version Control

Git is implemented as the version control system for MessMate, with a straightforward approach:

- **Branching Strategy:** A simple, single-branch workflow was employed for this project:
 - main branch served as the central branch for all development work
 - Direct commits to main were made for both features and fixes
 - This approach was selected to simplify the workflow for the project's scale and team size

- **Commit Approach:** Standard Git commits were utilized without specific formatting conventions:
 - Descriptive commit messages were encouraged
 - Changes were generally grouped by feature or fix type
 - Focus was placed on functional milestones rather than categorical commit types
- **GitHub:** GitHub was used as the remote repository host, providing:
 - Central code storage and backup
 - Basic project tracking capabilities
 - Collaboration features for code sharing and review

2.3.4 Development Workflow Tools

Additional tools that support the development workflow include:

- **Nodemon:** Used during backend development to automatically restart the Node.js server when file changes are detected.
- **dotenv:** Manages environment-specific configuration through .env files without committing sensitive information to version control.

2.3.5 Database Management Tools

MongoDB Compass provided a visual interface for database management and monitoring:

- **Visual Query Builder:** Intuitive interface for creating and testing database queries without writing complex MongoDB syntax, particularly useful for analyzing mentor performance metrics and session booking patterns.
- **Real-time Performance Monitoring:** Live insights into database performance, query execution times, and resource utilization to ensure optimal platform responsiveness during peak booking periods.

- **Schema Visualization:** Interactive exploration of document structures and data relationships, helping developers understand the connections between users, sessions, reviews, and availability collections.
- **Index Management:** Visual tools for creating, analyzing, and optimizing database indexes to improve query performance for frequently accessed data like mentor searches and session filtering.

Chapter 3

SYSTEM ANALYSIS AND SYSTEM STUDY

3.1 Product Analysis

Mentora, the system has been designed to address the growing need for accessible, personalized mentorship by connecting students with professionals across various domains. The platform aims to simplify the process of finding and scheduling one-on-one sessions with mentors, thereby enhancing the learning and development experience for young learners.

The system primarily supports two types of users: **Learners** and **Mentors**. Learners, typically students or early-career individuals, can register on the platform, log in, search for mentors based on specific criteria such as area of expertise or availability, and book one-on-one sessions. They also have access to a personalized dashboard to track upcoming bookings and submit reviews after each session. Mentors, on the other hand, can create and manage their profiles, including updating personal details, professional experience, and available time slots. They can view a list of their scheduled sessions and prepare accordingly. While an admin role has been considered for future versions—particularly to oversee mentor onboarding and platform moderation—there is currently no administrative functionality implemented in the present system.

From a functional perspective, the core modules of the system include role-based authentication, mentor search and filter, booking management, availability scheduling, session history tracking, and review submission. These modules are supported by a robust front-end interface built using **ReactJS** and styled with **Tailwind CSS**, providing a responsive and consistent user experience. State management is handled using **Context API**, enabling smooth data flow across components without prop drilling.

The backend of the application is developed using **Express.js**, which provides RESTful APIs for handling user authentication, session management, and data operations. The backend is connected to a **MongoDB** database, which stores and organizes entities such as users, bookings, availability slots, and reviews. MongoDB's flexible schema design supports the dynamic nature of the application while allowing for scalability and easy future modifications.

The current stage of development reflects significant progress, with most of the frontend components completed and initial backend APIs functional. Learner and mentor dashboards are operational, and essential features such as session booking, time slot selection, and review systems are integrated and undergoing refinement. Some advanced features like in-app messaging, notification systems, or calendar synchronization are not yet implemented but may be included in later versions depending on user feedback and project scope expansion.

In summary, the system analysis of *Mentora* demonstrates that the project is well-aligned with its objectives, offering a structured, user-friendly platform to facilitate meaningful interactions between mentors and learners. The system architecture supports future scalability while the current implementation effectively addresses the core use cases. The absence of admin functionality does not hinder current operations, but its potential addition remains a viable direction for future enhancement.

3.2 Feasibility Study

A feasibility study was conducted to determine whether the project can be successfully developed and deployed. It includes various aspects:

3.2.1 Technical Feasibility

Mentora is built using widely adopted and accessible technologies:

- **Frontend:** ReactJS with Tailwind CSS ensures responsive, modern UI development.
- **Backend:** Express.js offers a lightweight and scalable REST API framework.

- **Database:** MongoDB is well-suited for the flexible, document-based data model required for user roles and session bookings.
- **State Management:** Context API is sufficient for the current scope; scalable to Redux or Zustand if needed.
- **Infrastructure:** Deployment is feasible using platforms like Vercel (frontend), Render/Heroku (backend), and MongoDB Atlas (cloud database).

All chosen technologies are proven, community-supported, and free to use with generous free tiers — making the project technically feasible.

3.2.2 Operational Feasibility

- **User Roles:** The system cleanly separates features for learners, mentors, and admins.
- **Core Functionalities:** Session booking, profile management, reviews, and mentor availability are clear, actionable features.
- **User Experience:** The interface is being designed with accessibility and clarity in mind, especially for a student-focused audience.
- **Maintenance:** Easily maintainable due to modular component structure and well-separated backend routes.

Conclusion: With simple onboarding and logical flows, Mentora is operationally feasible for real-world use and user testing.

3.2.3 Economic Feasibility

- **Cost:** The project has no major monetary costs for now, thanks to open-source tools and student-tier hosting services.
- **Value:** High return in terms of skills, resume value, and future project expansion potential (e.g., turning it into a SaaS platform).

Conclusion: The cost-benefit ratio is strongly positive, with minimal financial investment and long-term personal/professional gain.

3.2.4 Schedule Feasibility

Progress So Far:

- UI design: Largely completed
- Basic backend routes: Implemented
- Additional features (like advanced filtering, availability logic, reviews): In progress

Remaining Tasks: Testing, polishing, edge-case handling, and integration refinements

3.3 Costs-Benefit Analysis

Costs

Category	Description	Estimated Cost (if monetized)
Development Time	Time spent by the team to build frontend (ReactJS, Tailwind) and backend (Express.js, MongoDB)	₹0 (college project), but ~₹30,000+ if outsourced
Design	UI/UX design efforts including wireframes, visual identity, and responsiveness	₹0 (self-designed), ~₹10,000 if outsourced
Hosting & Domain	Costs for deploying backend APIs (e.g., Render, Railway), frontend (e.g., Vercel), and a domain name	₹0 (free tier) to ₹5,000 annually
Database	MongoDB Atlas free tier used for development	₹0 (initially), scalable paid plan later
Learning & Research	Time spent learning technologies, understanding user needs	Intangible, but time-intensive
Maintenance	Future updates, bug fixes, and feature	Depends on scale, ₹5,000–

Category	Description	Estimated Cost (if monetized)
	enhancements	₹15,000 monthly if scaled

Table 3.3.1 Cost Analysis Table

Benefits

Category	Description
Real-World Learning	Practical experience in full-stack development, design systems, REST APIs, MongoDB schemas, etc.
Portfolio Value	Strong addition to team members' resumes/portfolios
Scalability	Modular architecture allows easy addition of features like payments, chat, video sessions
Social Impact	Helps students connect with real-world mentors and receive personalized career guidance
Monetization Potential	Can be scaled into a SaaS product or community platform for student-mentor connections
Skill Demonstration	Demonstrates proficiency in modern tech stack (MERN), role-based access control, and clean UI design

Table 3.3.2 Benefits Analysis Table

Conclusion

Mentora has minimal financial cost due to free-tier tools and in-house development but provides **high educational and strategic value**. It can evolve into a full-fledged mentorship product with further investment and has potential for both **social impact** and **commercial viability**.

Chapter 4

PROJECT DESIGN

4.1 - Data Dictionary

4.1.1 Table: mentee_user

Purpose: Stores information about users who seek mentorship services.

Column Name	Data Type	Constraints	Description
_userId	string	Primary Key	Unique identifier for the mentee user
username	string	-	Unique username for login and identification
name	string	-	Full name of the mentee
profileImage	string	-	URL or path to the user's profile image
number	number	-	Contact phone number
email	string	-	Email address for communication and login
password	string	-	Encrypted password for authentication
createdAt	timespan	-	Timestamp when the account was created
updatedAt	timespan	-	Timestamp of the last account update

Table 4.1.1 Table Mentee (Learner) User Data Dictionary

4.1.2 Table: mentor_user

Purpose: Stores information about users who provide mentorship services.

Column Name	Data Type	Constraints	Description
_userId	string	Primary Key	Unique identifier for the mentor user

Column Name	Data Type	Constraints	Description
username	string	-	Unique username for login and identification
name	string	-	Full name of the mentor
profileImage	string	-	URL or path to the user's profile image
number	number	-	Contact phone number
email	string	-	Email address for communication and login
password	string	-	Encrypted password for authentication
createdAt	timespan	-	Timestamp when the account was created
updatedAt	timespan	-	Timestamp of the last account update

Table 4.1.2 Table Mentor User Data Dictionary

4.1.3 Table: socialLinks

Purpose: Stores social media profiles and external links associated with users (primarily mentors).

Column Name	Data Type	Constraints	Description
_id	number	Primary Key	Unique identifier for the social link entry
userId	string	Foreign Key	References mentor_user._userId
platform	string	-	Name of the social media platform (e.g., LinkedIn, Twitter)
url	string	-	Full URL to the user's profile on the platform
createdAt	timespan	-	Timestamp when the link was added
updatedAt	timespan	-	Timestamp of the last link update

Table 4.1.3 Table SocialLinks Data Dictionary

4.1.4 Table: work_experience

Purpose: Stores professional work history information for users (primarily mentors).

Column Name	Data Type	Constraints	Description
_id	string	Primary Key	Unique identifier for the work experience entry
userId	string	Foreign Key	References mentor_user._userId
company	string	-	Name of the company or organization
jobProfile	string	-	Job title or role designation
startingDate	timespan	-	Date when the job started
endingDate	string	-	Date when the job ended (or “Present” for current job)
aboutJob	string	-	Description of job responsibilities and achievements
createdAt	timespan	-	Timestamp when the entry was created
updatedAt	timespan	-	Timestamp of the last entry update

Table 4.1.4 Table work_experience Data Dictionary

4.1.5 Table: education

Purpose: Stores educational background information for users (primarily mentors).

Column Name	Data Type	Constraints	Description
_id	string	Primary Key	Unique identifier for the education entry
userId	string	Foreign Key	References mentor_user._userId
college	string	-	Name of the educational institution
course	string	-	Name of the degree or course studied
CGPA	number	-	Cumulative Grade Point Average or equivalent score

Column Name	Data Type	Constraints	Description
startingDate	timespan	-	Date when the education started
endingDate	string	-	Date when the education ended
createdAt	timespan	-	Timestamp when the entry was created
updatedAt	timespan	-	Timestamp of the last entry update

Table 4.1.5 Table education Data Dictionary

4.1.6 Table: expertise

Purpose: Stores areas of expertise and skill levels for mentors.

Column Name	Data Type	Constraints	Description
_id	string	Primary Key	Unique identifier for the expertise entry
userId	string	Foreign Key	References mentor_user._userId
expertise	string	-	Area of expertise or skill (e.g., “JavaScript”, “Project Management”)
level	string	-	Proficiency level (e.g., “Beginner”, “Intermediate”, “Expert”)
createdAt	timespan	-	Timestamp when the entry was created
updatedAt	timespan	-	Timestamp of the last entry update

Table 4.1.6 Table expertise Data Dictionary

4.1.7 Table: reviews

Purpose: Stores feedback and ratings provided by mentees about their mentors.

Column Name	Data Type	Constraints	Description
_id	string	Primary Key	Unique identifier for the review
menteeId	string	Foreign Key	References mentee_user._userId

Column Name	Data Type	Constraints	Description
mentorId	string	Foreign Key	References mentor_user._userId
rating	number	-	Numerical rating (typically 1-5 scale)
reviewText	string	-	Written feedback and comments
createdAt	timespan	-	Timestamp when the review was submitted
updatedAt	timespan	-	Timestamp of the last review update

Table 4.1.7 Table reviews Data Dictionary

4.1.8 Table: bookings

Purpose: Manages mentorship session bookings between mentees and mentors.

Column Name	Data Type	Constraints	Description
_id	string	Primary Key	Unique identifier for the booking
menteeId	string	Foreign Key	References mentee_user._userId
mentorId	string	Foreign Key	References mentor_user._userId
scheduleId	string	Foreign Key	References schedules._id
status	string	Enum	Current booking status: “pending”, “confirmed”, or “cancelled”
paymentId	number	-	Reference to associated payment transaction
meetingLink	string	-	URL for the online meeting (Zoom, Google Meet, etc.)
notes	string	-	Additional notes or special instructions
createdAt	timespan	-	Timestamp when the booking was created

Column Name	Data Type	Constraints	Description
updatedAt	timespan	-	Timestamp of the last booking update

Table 4.1.8 Table bookings Data Dictionary

4.1.9 Table: payments

Purpose: Tracks payment transactions for mentorship bookings.

Column Name	Data Type	Constraints	Description
_id	string	Primary Key	Unique identifier for the payment record
bookingId	string	Foreign Key	References bookings._id
amount	string	-	Payment amount (stored as string to handle currency formatting)
status	string	Enum	Payment status: "success", "failed", or "pending"
razorpayPaymentId	string	-	Payment ID from Razorpay payment gateway
razorpayOrderId	string	-	Order ID from Razorpay payment gateway
paymentMethod	string	-	Method used for payment (card, UPI, wallet, etc.)
createdAt	timespan	-	Timestamp when the payment was initiated
updatedAt	timespan	-	Timestamp of the last payment update

Table 4.1.9 Table payments Data Dictionary

4.1.10 Table: schedules

Purpose: Manages available time slots that mentors offer for booking.

Column Name	Data Type	Constraints	Description
_id	string	Primary Key	Unique identifier for the schedule slot
mentorId	string	Foreign Key	References mentor_user._userId
date	timestamp	-	Date of the available slot
startTime	timespan	-	Start time of the available slot
endTime	timespan	-	End time of the available slot
isBooked	boolean	-	Flag indicating if the slot is already booked
createdAt	timespan	-	Timestamp when the schedule was created
updatedAt	timespan	-	Timestamp of the last schedule update

Table 4.1.10 Table schedules Data Dictionary

Relationships Summary

One-to-Many Relationships

- *mentor_user → socialLinks*: One mentor can have multiple social media links
- *mentor_user → work_experience*: One mentor can have multiple work experiences
- *mentor_user → education*: One mentor can have multiple educational qualifications
- *mentor_user → expertise*: One mentor can have expertise in multiple areas
- *mentor_user → schedules*: One mentor can create multiple available time slots

Many-to-Many Relationships

- *mentor_user ↔ bookings ↔ mentee_user*: Mentors and mentees can have multiple bookings with each other
- *mentor_user ↔ reviews ↔ mentee_user*: Mentors can receive multiple reviews from different mentees

One-to-One Relationships

- *bookings → payments*: Each booking has one associated payment
- *bookings → schedules*: Each booking is associated with one specific schedule slot

Business Rules & Constraints

- **User Authentication**: Both mentees and mentors require unique usernames and email addresses
- **Booking Flow**: A booking must reference valid mentee, mentor, and schedule records
- **Payment Integration**: Payments are processed through Razorpay gateway with transaction tracking
- **Schedule Management**: Time slots become unavailable (`isBooked = true`) when booked
- **Review System**: Only mentees who have completed bookings can leave reviews for mentors
- **Data Integrity**: All timestamp fields track creation and modification times for audit purposes.

4.2 – Data Flow Diagram

Level 0 - Context Diagram

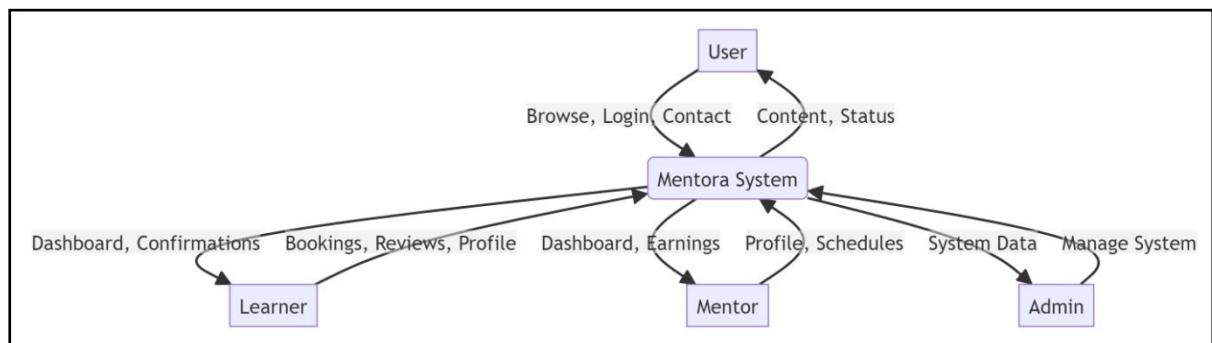


Fig. 4.2.1: Level 0 Data Flow Diagram – Overall System Architecture

Level 1 - Overview DFD

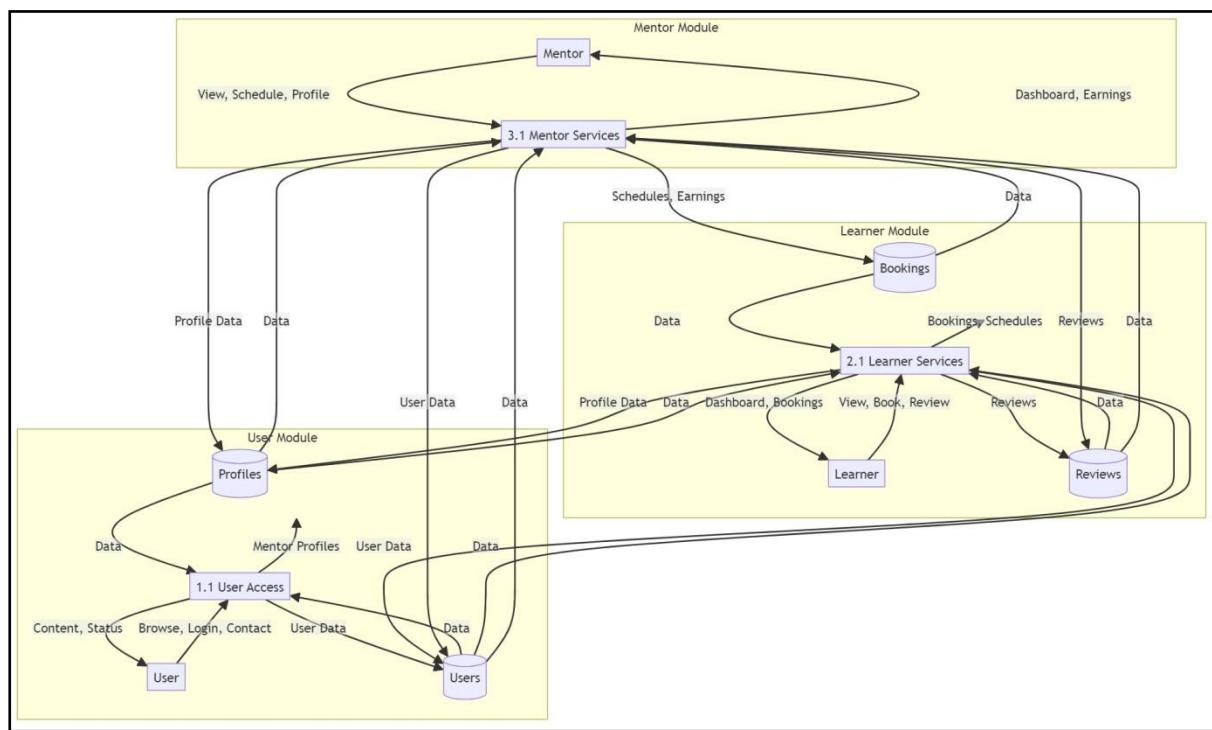


Fig. 4.2.2: Level 1 Data Flow Diagram – Detailed System Architecture

Level 2 - Detailed Process Breakdowns

User Module

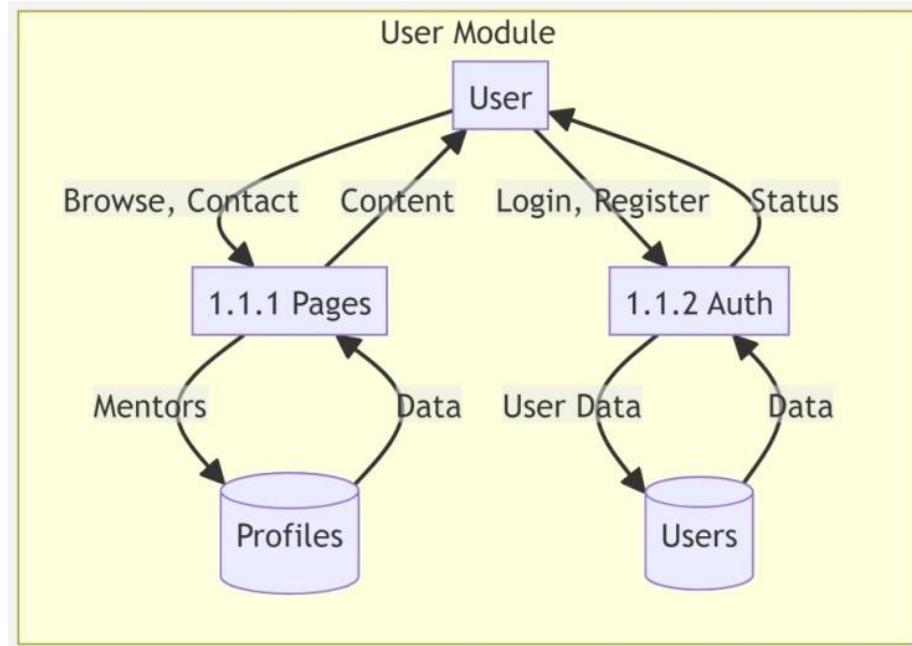


Fig. 4.2.3: Level 2 Data Flow Diagram – User Module Workflow

Learner Module

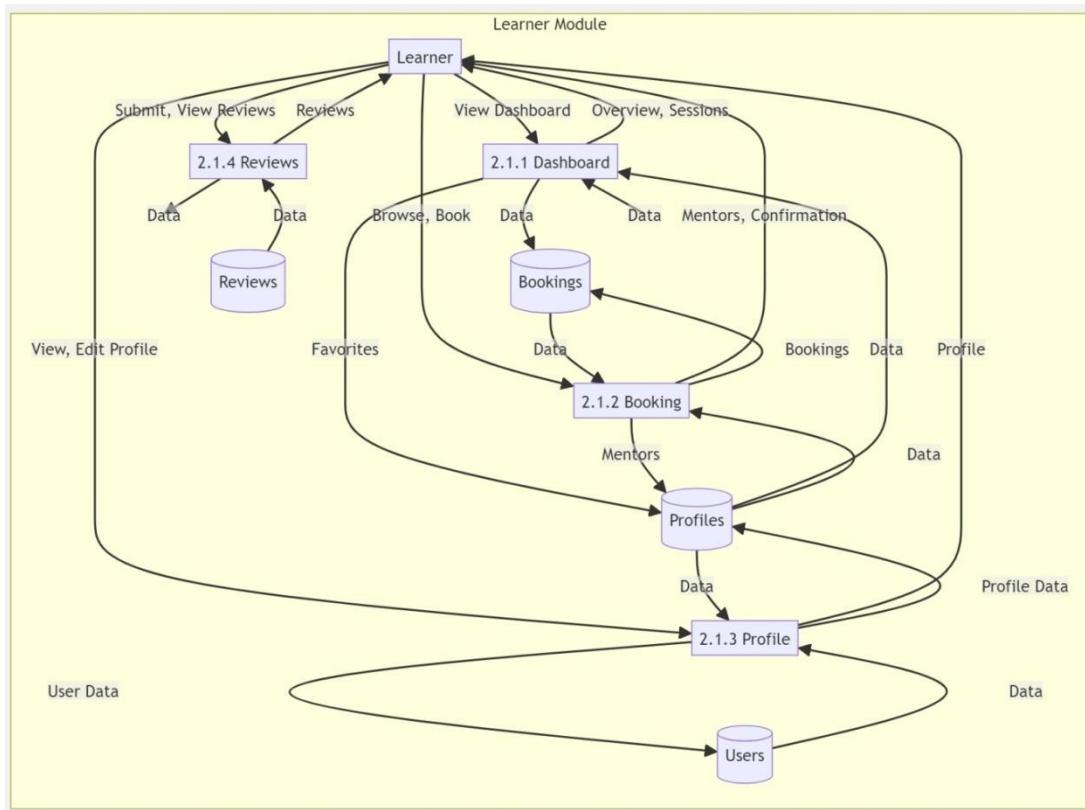


Fig. 4.2.4: Level 2 Data Flow Diagram – Learner Module Workflow

Mentor Module

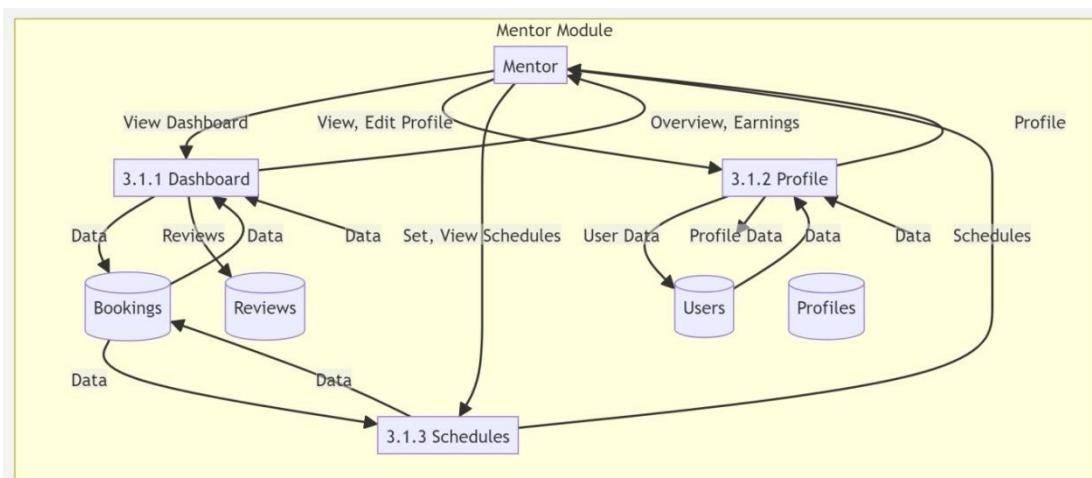
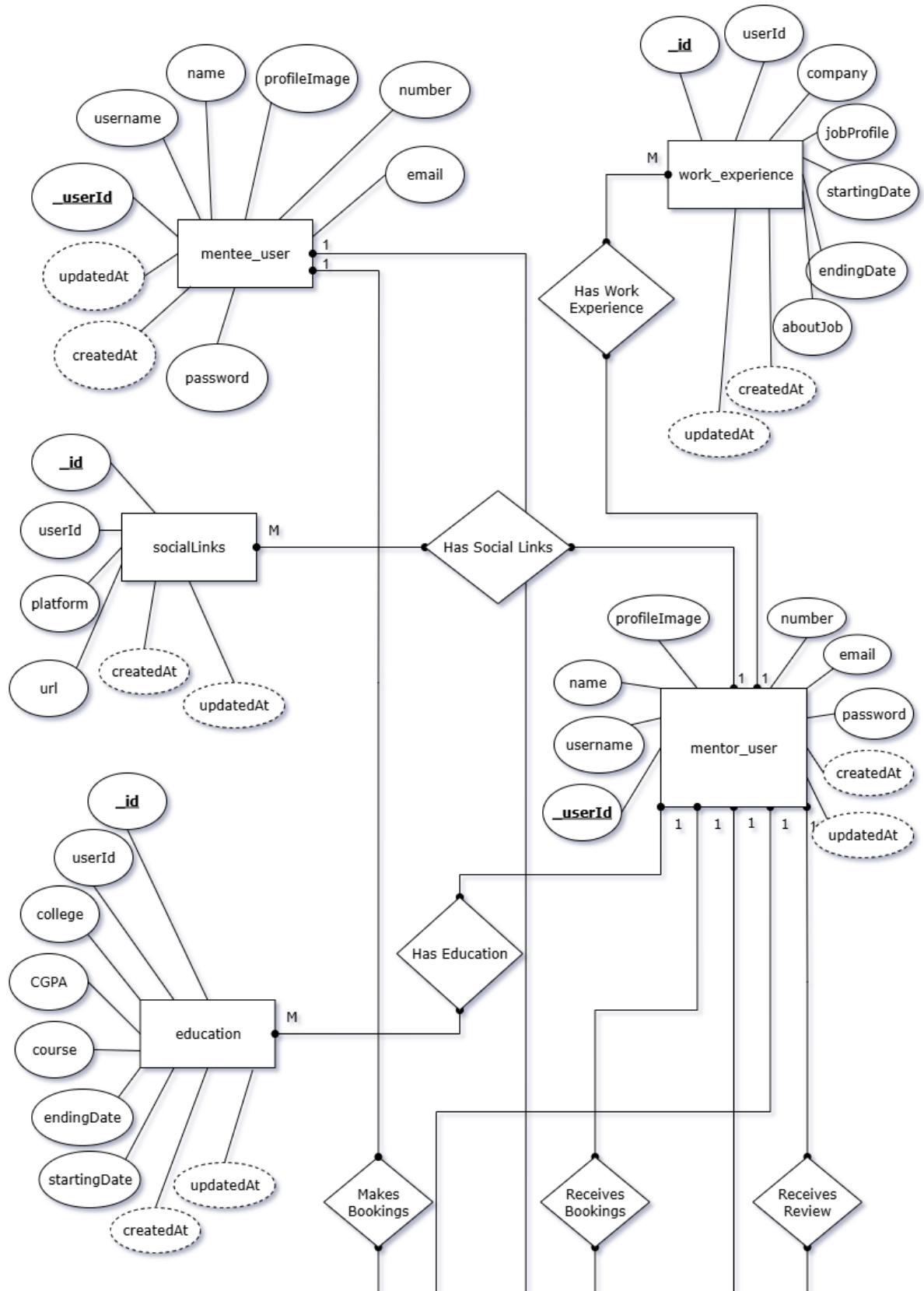


Fig. 4.2.5: Level 2 Data Flow Diagram – Mentor Module Workflow

4.3 – Entity Relation Diagram



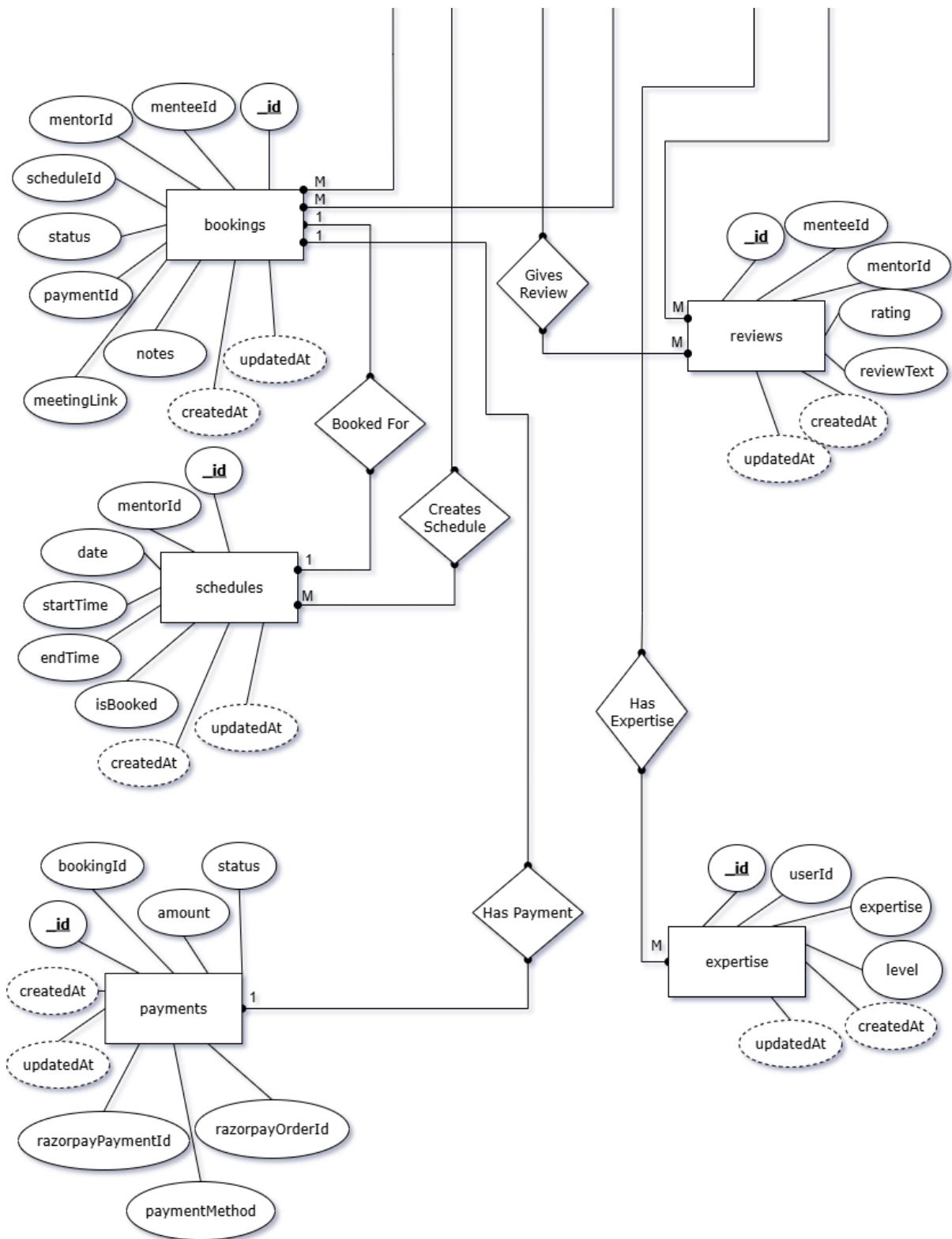


Fig. 4.3: Entity Relation Diagram

4.4 Database Diagrams



Fig. 4.4: Database Schema Diagram

4.5 - Snapshot (screenshot of all pages)

4.5.1 Landing Page

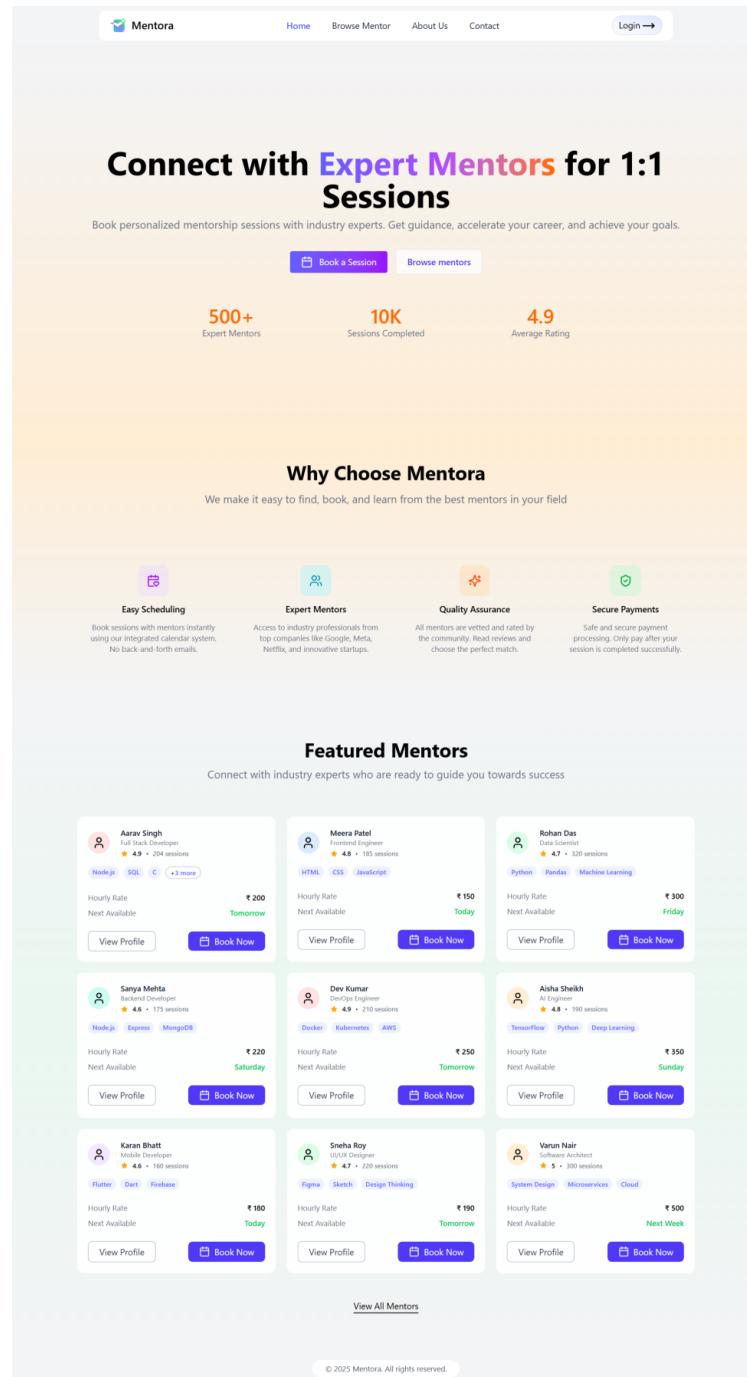


Fig. 4.5.1: Landing page screenshot
Pg. 29

The main entry point showcasing Mentora's value proposition with an intuitive hero section, featured mentors, and platform benefits. Includes a prominent call-to-action for registration and easy navigation to key sections. The responsive design ensures optimal viewing across all devices.

Navigation Bar:

- **Login/Register:** Secure authentication system with role selection (Learner/Mentor)
- **Browse Mentors:** Public access to mentor profiles and expertise areas
- **About:** Platform mission, vision, and team information
- **Contact:** Direct communication channel for inquiries and support

4.5.2 Browse Mentor Page

The screenshot shows the Mentorora website's browse mentor section. At the top, there's a navigation bar with links for Home, Browse Mentor, About Us, Contact, and Login. Below the navigation is a large banner with the heading "Find Your Mentor" and a subtext: "Connect with industry leaders and experienced professionals who can guide your career journey and help you achieve your goals." Three statistics are displayed in colored boxes: "500+ Expert Mentors" (blue), "10k Sessions Completed" (green), and "4.9 Average Rating" (orange). Below the banner is a search bar with placeholder text "Search by name, skill, company, or expertise..." and a list of trending searches: React, Data, Cloud, Home Work.

Available Mentors

30 mentor ready to help you succeed

Mentor Profile	Skills	Hourly Rate	Next Available	Action Buttons
Aarav Singh Full Stack Developer ★ 4.9 • 204 sessions	Node.js, SQL, C, +3 more	₹ 200	Tomorrow	View Profile Book Now
Meera Patel Frontend Engineer ★ 4.8 • 185 sessions	HTML, CSS, JavaScript	₹ 150	Today	View Profile Book Now
Rohan Das Data Scientist ★ 4.7 • 320 sessions	Python, Pandas, Machine Learning	₹ 300	Friday	View Profile Book Now
Sanya Mehta Backend Developer ★ 4.6 • 175 sessions	Node.js, Express, MongoDB	₹ 220	Saturday	View Profile Book Now
Dev Kumar DevOps Engineer ★ 4.9 • 210 sessions	Docker, Kubernetes, AWS	₹ 250	Tomorrow	View Profile Book Now
Aisha Sheikh AI Engineer ★ 4.8 • 190 sessions	TensorFlow, Python, Deep Learning	₹ 350	Sunday	View Profile Book Now
Karan Bhatt Mobile Developer ★ 4.6 • 160 sessions	Flutter, Dart, Firebase	₹ 180	Today	View Profile Book Now
Sneha Roy UI/UX Designer ★ 4.7 • 220 sessions	Figma, Sketch, Design Thinking	₹ 190	Tomorrow	View Profile Book Now
Varun Nair Software Architect ★ 5 • 300 sessions	System Design, Microservices, Cloud	₹ 500	Next Week	View Profile Book Now

Previous Page 1 of 4 Next © 2025 Mentorora. All rights reserved.

Fig. 4.5.2: Browse Mentor Page

A comprehensive directory displaying all available mentors with filtering and search capabilities. Users can view mentor profiles, expertise areas, ratings, and availability without requiring authentication. Each mentor card shows key information including specialization, experience level, and average rating.

4.5.3 About Us Page

About Mentora

We believe everyone deserves access to great mentorship. Our platform connects ambitious individuals with experienced professionals who are passionate about sharing their knowledge.

500+ Active Mentors	2000+ Success Stories	10000+ Experts sessions	5000+ Happy mentees
------------------------	--------------------------	----------------------------	------------------------

Our Mission

Democratizing Access to Quality Mentorship

Traditional mentorship has often been limited to those with existing networks. We're changing that by creating a platform where anyone can find the guidance they need to succeed.

Our carefully vetted mentors come from diverse backgrounds and industries, ensuring that every mentee can find someone who understands their unique journey and challenges.

© 2025 Mentora. All rights reserved.

Fig. 4.5.3: About Page
Pg. 33

Dedicated section highlighting Mentora's mission to democratize learning through personalized mentorship. Features the platform's core values, team background, success stories, and future roadmap. Includes testimonials and statistics demonstrating platform impact.

4.5.4 Contact Page

Fig. 4.5.4: Contact Page

Professional contact interface with a structured inquiry form allowing users and potential business partners to submit queries. Includes categorized inquiry types, estimated response times, and alternative contact methods for urgent matters.

4.5.5 Authentication System

Unified authentication interface providing secure access to the Mentora platform with dual-path user onboarding:

4.5.5.1 Login Section

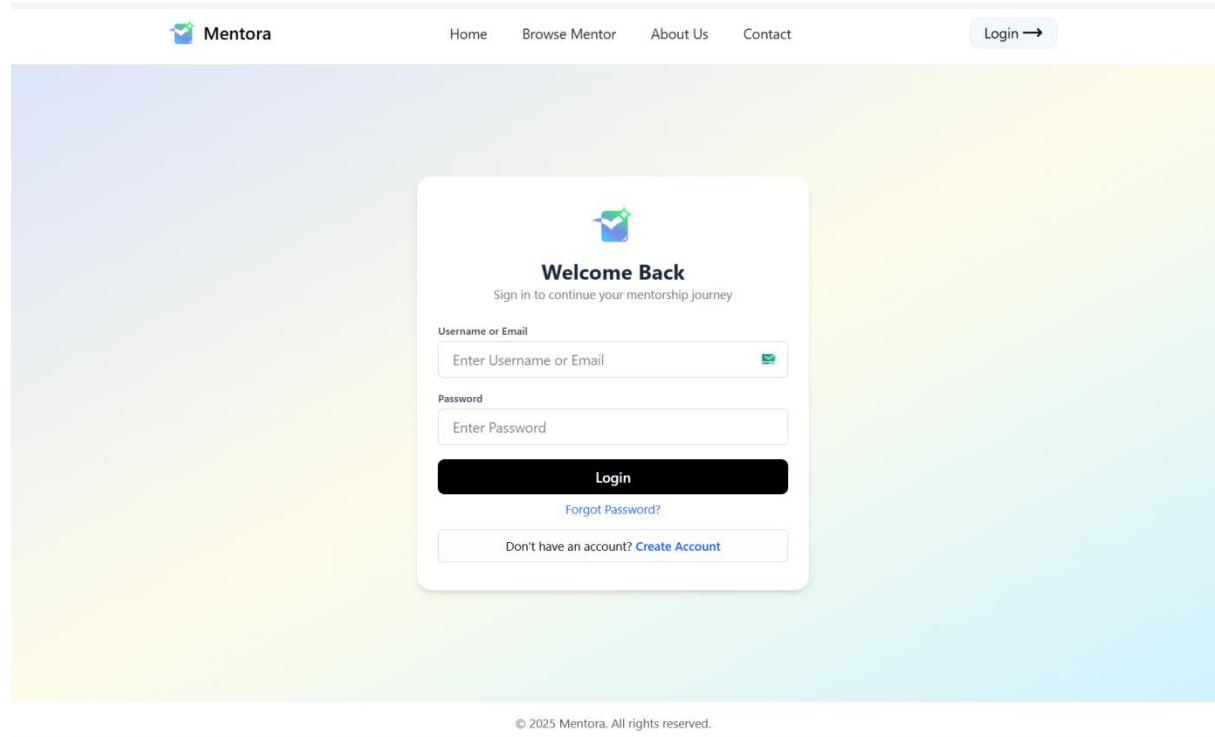


Fig. 4.5.5.1: Login Page

Enables existing users to login as learner or mentor.

4.5.5.2 Registration Section

The screenshot shows the 'Create Your Account' page for the Mentora platform. At the top, there is a navigation bar with links for Home, Browse Mentor, About Us, Contact, and a 'Login →' button. The main form is titled 'Create Your Account' with the sub-instruction 'Join our community and start your mentorship journey'. The form fields include 'Full Name' (placeholder: 'Enter your full name'), 'Email' (placeholder: 'Enter your email'), 'Password' (placeholder: 'Create a password'), 'Confirm Password' (placeholder: 'Re-enter your password'), and a dropdown menu for 'I am' (selected option: 'Mentee seeking mentorship'). A large 'Create Account' button is at the bottom of the form, and a link 'Already have an account? [login](#)' is located just below it. The footer of the page contains the copyright notice '© 2025 Mentora. All rights reserved.'

Fig. 4.5.5.2: Registration Page

Enables new users to register themselves on the platform as Learner or Mentor.

4.5.6 Learner Dashboard (Post-Login)

4.5.6.1 Dashboard Overview

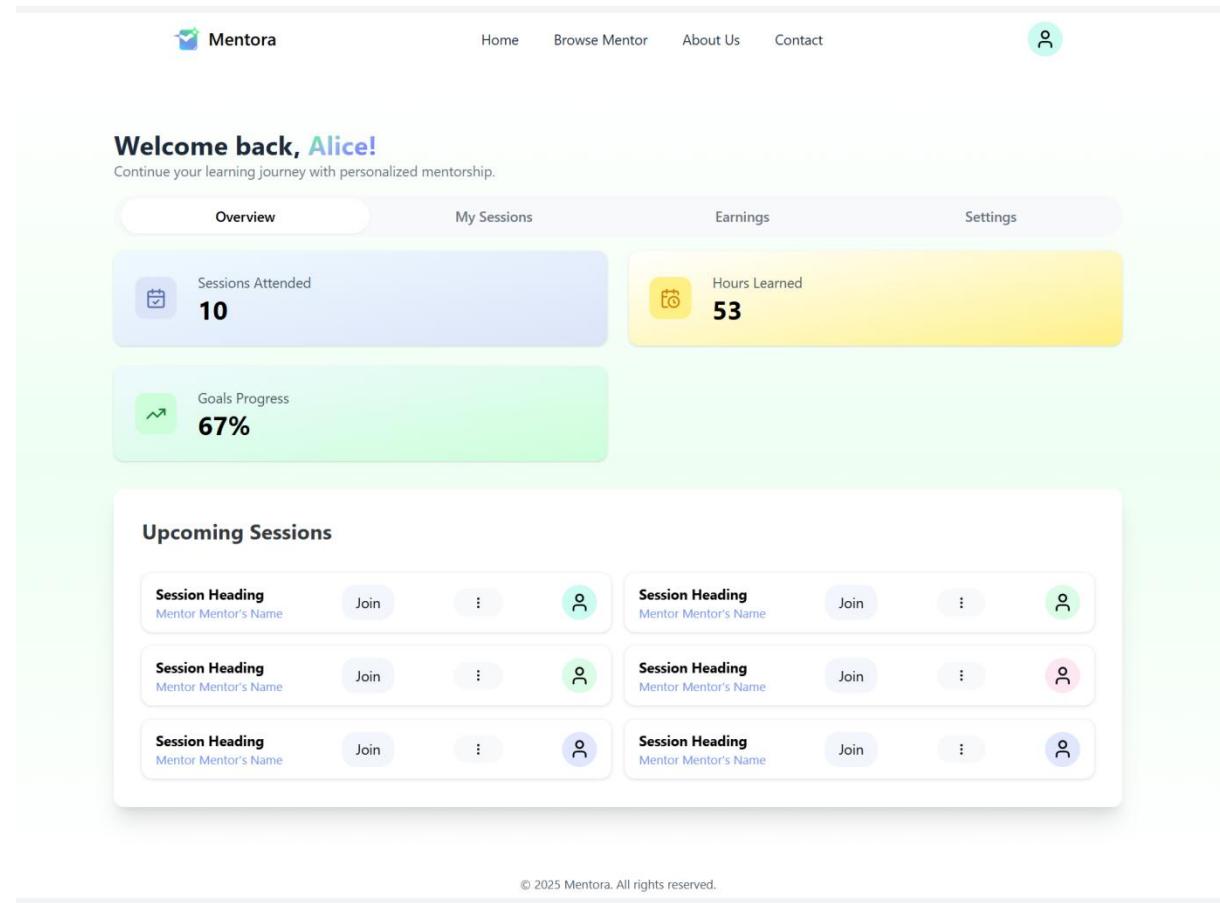


Fig. 4.5.6.1: Learner Dashboard

Comprehensive learning analytics hub displaying key metrics in an intuitive dashboard format. Features visual representations of learning progress, session statistics, total hours invested, and achievement milestones. Includes quick access buttons to frequently used features.

Key Metrics Displayed:

- Total sessions completed

- Learning hours accumulated
- Favorite mentors count
- Upcoming session alerts

4.5.6.2 Sessions Management

The screenshot displays the Learner's Session Page on the Mentora platform. At the top, there is a navigation bar with links for Home, Browse Mentor, About Us, Contact, and a user profile icon. Below the navigation bar, a welcome message says "Welcome back, steve!" followed by a subtext "Continue your learning journey with personalized mentorship." The main content area is divided into two sections: "Upcoming Sessions" and "Past Sessions".

Upcoming Sessions: This section lists five sessions, each with a card containing the session heading, mentor's name, a "Join" button, a three-dot menu icon, and a person icon. The person icons are colored blue, green, yellow, and red.

Session Heading	Mentor Mentor's Name	Join	⋮	Person Icon
Session Heading	Mentor Mentor's Name	Join	⋮	Blue
Session Heading	Mentor Mentor's Name	Join	⋮	Green
Session Heading	Mentor Mentor's Name	Join	⋮	Yellow
Session Heading	Mentor Mentor's Name	Join	⋮	Red

Past Sessions: This section lists four sessions, each with a card containing the session heading, mentor's name, a "Leave Review" button, a three-dot menu icon, and a person icon. The person icons are colored purple, orange, yellow, and purple. Each card also includes a rating star icon below the person icon.

Session Heading	Mentor Mentor's Name	Leave Review	⋮	Person Icon
Session Heading	Mentor Mentor's Name	Leave Review	⋮	Purple
Session Heading	Mentor Mentor's Name	Leave Review	⋮	Orange
Session Heading	Mentor Mentor's Name	★★★☆☆	⋮	Yellow
Session Heading	Mentor Mentor's Name	★★★☆☆	⋮	Purple

At the bottom of the page, a copyright notice reads "© 2025 Mentora. All rights reserved."

Fig. 4.5.6.2: Learner's Session Page

Complete session lifecycle management interface displaying:

- **Upcoming Sessions:** Schedule view with join links, mentor details, and session preparation materials
- **Past Sessions:** Historical record with session outcomes, duration, and integrated rating system

- **Review System:** Streamlined interface for providing mentor feedback and ratings post-session

4.5.6.3 Favourites

The screenshot shows the 'Favourites Mentor' section of the Learner's Favourites Page. At the top, there is a navigation bar with links: Home, Browse Mentor, About Us, Contact, and a user profile icon. Below the navigation bar, a welcome message reads 'Welcome back, steve!' followed by a sub-message: 'Continue your learning journey with personalized mentorship.' The main content area displays five mentor profiles, each consisting of a small circular icon with a person symbol, the mentor's name, and their title 'Full Stack Developer'. The 'Favorites Mentor' tab is highlighted with a green background and white text. The other tabs are 'Overview', 'My Sessions', and 'Settings'. At the bottom of the page, a copyright notice reads '© 2025 Mentora. All rights reserved.'

Fig. 4.5.6.3: Learner's Favourites Page

Curated list of preferred mentors with quick access to their profiles and booking systems. Allows learners to track their preferred mentors and receive notifications about their availability or new offerings.

4.5.6.4 Setting

The screenshot shows the 'Settings' page for a learner on the Mentora platform. At the top, there is a navigation bar with links for Home, Browse Mentor, About Us, Contact, and a 'Login →' button. Below the navigation bar, a welcome message reads 'Welcome back, ! Continue your learning journey with personalized mentorship.' The main content area is divided into two sections: 'Profile Management' and 'Security & Privacy'.

Profile Management

- Personal Information**: Update basic information, contact details, and bio. Includes an 'Update' button.
- Education & Qualifications**: Add educational background and certifications. Includes an 'Update' button.
- Social Links**: Connect social media profiles and websites. Includes an 'Update' button.
- Skills & Expertise**: Showcase skills, experience, and areas of expertise. Includes an 'Update' button.

Security & Privacy

Change Password: Fields for Current Password, New Password, and Confirm New Password.

Email Notifications: A toggle switch is turned on, with the description 'Receive updates about your account activity'.

Public Profile: A toggle switch is turned on, with the description 'Make your profile visible to others'.

Show Email Publicly: A toggle switch is turned off, with the description 'Display your email address on your profile'.

Two-Factor Authentication: A toggle switch is turned off, with the description 'Add an extra layer of security to your account'.

A large blue button at the bottom of the section is labeled 'Save All Changes'.

Danger Zone: A red warning box contains the text 'Once you deactivate your account, there is no going back. Please be certain.' and a red button labeled 'Deactivate Account'.

At the bottom of the page, a copyright notice reads '© 2025 Mentora. All rights reserved.'

Fig. 4.5.6.4: Learner's Settings Page

Pg. 40

Comprehensive account management center providing learners with full control over their platform experience:

- **Account Security:** Password change functionality with strong password requirements and two-factor authentication options
- **Social Media Integration:** Management of connected social profiles (LinkedIn, Twitter, GitHub) for enhanced networking and profile credibility
- **Privacy Controls:** Granular privacy settings for profile visibility, learning activity sharing, and mentor interaction preferences
- **Account Management:** Critical account actions including temporary account deactivation with data retention options and permanent account deletion with clear data handling policies

4.5.6.5 Book Session

Mentora

Home Browse Mentor About Us Contact

Book a Session

Schedule your personalized mentorship session

Sarah Johnson ★ 4.9 (247)
Senior Software Engineer

₹75 15 minutes

Available: 9:00 AM - 7:00 PM

Select Date
25-07-2025

Duration *
15 minutes

- 15 min +

Session Type
Video

Session Details

Session Title *

e.g., React Architecture Review, Career Guidance...

Description / Notes

Describe what you'd like to discuss...

Available Time Slots * (36 available)

10:00 AM	10:15 AM	10:30 AM	10:45 AM
11:00 AM	11:15 AM	11:30 AM	11:45 AM
12:00 PM	12:15 PM	12:30 PM	12:45 PM
1:00 PM	1:15 PM	1:30 PM	1:45 PM
2:00 PM	2:15 PM	2:30 PM	2:45 PM
3:00 PM	3:15 PM	3:30 PM	3:45 PM
4:00 PM	4:15 PM	4:30 PM	4:45 PM
5:00 PM	5:15 PM	5:30 PM	5:45 PM
6:00 PM	6:15 PM	6:30 PM	6:45 PM

12:15 PM - 12:30 PM (15 minutes)

Session Summary

Date: Fri Jul 25 2025
Time: 12:15 PM - 12:30 PM
Duration: 15 minutes
Type: Video Session
Total Price: ₹75

Book Session - ₹75

© 2025 Mentora. All rights reserved.

Fig. 4.5.6.5: Book Session Page

Gives learners an interface to choose a available time slot and book session accordingly

4.5.7 Mentor Dashboard (Post-Login)

4.5.7.1 Dashboard Overview

The screenshot shows the Mentor Dashboard Overview page for a user named Alice. At the top, there is a navigation bar with links for Home, Browse Mentor, About Us, and Contact, along with a profile icon showing the number 8. Below the navigation bar, a welcome message says "Welcome back, Alice!" followed by a subtext "Continue your learning journey with personalized mentorship." There are four main performance metrics displayed in cards: "Sessions Attended" (10), "Hours Learned" (53), "Goals Progress" (67%), and a fourth card that is partially visible. Below these metrics, there is a section titled "Upcoming Sessions" which lists five sessions, each with a "Join" button, a three-dot menu, and a user icon. At the bottom of the page, a copyright notice reads "© 2025 Mentora. All rights reserved."

Fig. 4.5.7.1: Mentor Dashboard Page

- Business intelligence hub for mentors featuring:
- Monthly earnings summary with trend analysis
- Session statistics and completion rates
- Average rating and review highlights
- Performance metrics and growth indicators

4.5.7.2 Sessions Management

Welcome back, Alice!

Continue your learning journey with personalized mentorship.

Overview **My Sessions** Earnings Settings

Upcoming Sessions

- Session Heading
Mentor Mentor's Name

Past Sessions

- Session Heading
Mentor Mentor's Name

© 2025 Mentora. All rights reserved.

Fig. 4.5.7.2: Mentor Sessions Page

Comprehensive session administration panel:

- **Upcoming Sessions:** Scheduled appointments with learner information and preparation notes
- **Session History:** Complete record of conducted sessions with outcomes and feedback
- **Availability Management:** Calendar interface for setting available time slots and session types

4.5.7.3 Earnings Dashboard

The screenshot shows the 'Earnings' dashboard for a mentor named Komal Sharma. At the top, there's a navigation bar with links for Home, Browse Mentor, About Us, Contact, and a profile icon. Below the navigation is a welcome message: 'Welcome back, Komal Sharma!' followed by a sub-instruction: 'Manage your mentoring sessions and track your impact'. A horizontal navigation bar below this includes Overview, My Sessions, Earnings (which is highlighted), and Settings.

The main section is titled 'Earnings Overview' with the subtitle 'Track your financial performance'. It displays three key metrics in cards:

- This Month:** \$4,200 (with a note: +15% from last month)
- Total Earned:** \$12,450 (with a note: Since joining)
- Avg per Session:** \$150 (Hourly rate)

Below this, there's a section titled 'Recent Payments' which lists two transactions:

User	Date	Description	Amount	Status
Emma Davis	2024-01-10	Strategy Planning	\$150	Paid
Mike Chen	2024-01-08	Career Pivot Advice	\$150	Paid

At the bottom of the page, a copyright notice reads: '© 2025 Mentora. All rights reserved.'

Fig. 4.5.7.3: Mentor Earnings Page

Financial management interface providing:

- Detailed earning history with filtering options
- Payment status tracking and payout schedules
- Revenue analytics with monthly/quarterly breakdowns
- Tax documentation and financial reporting tools

Chapter 5

MECHANISM USED

This chapter describes the testing and implementation mechanisms employed during the development of the MessMate Report application. It outlines how the system was rigorously validated and how each component was effectively implemented to ensure a smooth and efficient user experience.

5.1- Testing Mechanism

Testing was integrated at different stages of the development lifecycle to ensure functionality and stability of the application. The key mechanisms used include:

- **Manual Testing:** The application was initially tested manually at various stages to identify UI inconsistencies, logic errors, and navigation flow issues. Form validations, booking flows, and state management were manually checked on different devices and screen sizes.
- **Unit Testing:** Selected components, particularly those handling form input, session data, and context-based logic, were tested in isolation to verify individual functionalities.
- **API Testing:** Postman was used to test backend Express.js endpoints independently. Each route, including session booking, user login, and data fetching, was tested with different payloads and error scenarios.
- **Integration Testing (Planned):** Although not yet fully implemented, future iterations plan to integrate automated testing using tools such as Jest and React Testing Library to ensure combined frontend-backend functionality remains intact after code changes.

5.2 - Implementation Mechanism

- The project's implementation was guided by modular design and best practices to ensure clarity, reusability, and scalability. Below are the core mechanisms that defined the implementation process:
- **Frontend Development:** The frontend of the application was developed using **ReactJS** with **Tailwind CSS** for styling. Context API was utilized for state management, especially in handling user roles and session data. Components were designed to be reusable and responsive.
- **Backend Development:** The backend was built using **Express.js** and **MongoDB**. RESTful APIs were created for managing users, mentor sessions, reviews, and booking information. Middleware functions handled authentication and request validation.
- **Routing & Role-Based Access:** Protected routes were set up both on the client and server side to ensure that pages and actions are accessible only to authenticated users, and role-based rendering (mentor vs learner) was implemented.
- **Session Booking Logic:** A dynamic system was implemented to allow learners to view mentor availability and book sessions accordingly. Session conflicts were prevented via availability checks in the backend before confirmation.
- **Version Control:** Git was used for version control throughout the development, with branches created for different feature sets to allow smooth collaboration and integration.
- **Deployment (Planned):** While still in development, the application is intended to be deployed using platforms such as **Vercel** (for frontend) and **Render** or **Railway** (for backend), ensuring CI/CD compatibility and environment configuration flexibility.

Chapter 6

FUTURE SCOPE OF THE PROJECT

While **Mentora** is currently being developed as a foundational mentorship platform with core functionalities like registration, mentor discovery, session booking, and profile management, it is designed to be **highly scalable** and capable of integrating advanced features in future phases. The platform's modular architecture and robust backend infrastructure provide a solid foundation for implementing sophisticated enhancements that will transform the mentorship experience. Below are some potential enhancements that will significantly improve user experience, engagement, and functionality:

6.1 Integrated Video Calling

- Direct integration of secure video conferencing solutions (e.g., Jitsi Meet, WebRTC, Agora.io, or Zoom SDK) with end-to-end encryption for privacy protection.
- Eliminates the need for third-party tools like Google Meet, creating a seamless in-platform experience.
- Enables scheduled and spontaneous one-on-one mentorship sessions with features like screen sharing, whiteboard collaboration, and session recording (with consent).
- Advanced call quality optimization with automatic bandwidth adjustment and connection stability monitoring.
- Integration with calendar systems for automated session scheduling and timezone management.

6.2 AI-Based Mentor Recommendations

- Sophisticated machine learning algorithms that analyze learner profiles, career goals, past session feedback, domain expertise requirements, and learning preferences.
- Dynamic recommendation engine that considers mentor availability, teaching style compatibility, and success rates with similar learners.

- Uses natural language processing to match learners with mentors based on specific skill requirements and career aspirations mentioned in their profiles.
- Continuous learning system that improves recommendations based on session outcomes, ratings, and long-term learner progress tracking.
- Smart filtering options that consider factors like industry experience, company backgrounds, and specialized certifications.

6.3 AI-Powered Session Summaries

- Advanced speech-to-text technology with multi-language support and accent recognition for accurate transcription of mentorship sessions.
- Intelligent extraction of key discussion points, action items, resources mentioned, and follow-up tasks using natural language processing.
- Automated generation of structured learning notes with categorized insights, skill development areas, and progress milestones.
- Integration with learner's personal knowledge base and goal-tracking system for comprehensive learning journey documentation.
- Customizable summary formats based on session type (technical skills, career guidance, project reviews, etc.).

6.4 In-App Chat & Messaging System

- Feature-rich real-time messaging platform with support for text, voice messages, file attachments, and code snippet sharing.
- Advanced message organization with conversation threads, message search functionality, and conversation history archiving.
- Offline message delivery with push notifications and read receipts for improved communication reliability.
- Integration with session scheduling allowing direct booking requests and confirmations through chat.

- Smart message filtering and professional communication guidelines to maintain platform quality.

6.5 Notifications and Reminders

- Multi-channel notification system supporting email, SMS, push notifications, and in-app alerts with user preference customization.
- Intelligent reminder scheduling with multiple touchpoints: booking confirmation, pre-session reminders, and post-session follow-ups.
- Automated no-show prevention system with gentle nudges and easy rescheduling options.
- Personalized notification content based on user behavior patterns and engagement preferences.
- Integration with popular calendar applications for seamless schedule management.

6.6 Community and Group Sessions

- Comprehensive group learning infrastructure supporting webinars, workshops, and collaborative learning experiences.
- Flexible pricing models for group sessions including free community events, paid masterclasses, and subscription-based learning circles.
- Interactive features like live polls, Q&A sessions, breakout rooms, and collaborative whiteboards for enhanced engagement.
- Community forums and discussion boards organized by domains, skill levels, and interests.
- Mentor-led learning paths and structured curriculum delivery for specialized topics.

6.7 Performance Tracking and Analytics

- Comprehensive learner dashboards displaying session history, skill development progress, goal achievement metrics, and learning velocity analysis.

- Advanced mentor analytics including session effectiveness scores, learner satisfaction trends, booking patterns, and revenue insights.
- Predictive analytics to identify learning gaps, suggest improvement areas, and recommend optimal session frequency.
- Integration with professional development frameworks and certification tracking systems.
- Exportable progress reports for performance reviews and career development documentation.

6.8 Enhanced Admin Dashboard

- Sophisticated mentor verification system with background checks, credential validation, and ongoing performance monitoring.
- Advanced content moderation tools using AI-powered detection for inappropriate content and automated flag resolution.
- Comprehensive dispute resolution system with structured mediation processes and escalation protocols.
- Real-time platform health monitoring with user engagement metrics, system performance analytics, and business intelligence reporting.
- Automated compliance checking for data protection regulations and platform policy enforcement.

6.9 Multilingual & Regional Support

- Comprehensive localization supporting major Indian languages (Hindi, Bengali, Tamil, Telugu, Marathi, Gujarati, Kannada, Malayalam, Punjabi, and others) with cultural context adaptation.
- Regional mentor discovery based on location preferences, local industry expertise, and cultural familiarity.
- Localized payment methods supporting regional banking systems, digital wallets, and preferred transaction methods.

- Time zone optimization for cross-regional mentorship connections and automated scheduling adjustments.
- Cultural sensitivity features ensuring appropriate communication styles and professional etiquette standards.

6.10 Premium Features and Monetization

- Tiered subscription models offering priority booking, extended session durations, exclusive mentor access, and premium support services.
- Integrated digital wallet system with secure payment processing, automatic invoicing, and transparent fee structures.
- Revenue sharing models for mentors with performance-based incentives and loyalty programs.
- Corporate packages for organizations seeking bulk mentorship services for employee development programs.
- Certification and badge systems for completed learning paths with industry recognition and LinkedIn integration.

6.11 Advanced Matching & Compatibility

- Sophisticated personality and learning style assessments to optimize mentor-learner compatibility.
- Industry-specific expertise matching with deep-dive skill requirement analysis.
- Success prediction algorithms based on historical data and user interaction patterns.
- Dynamic re-matching suggestions when learning goals evolve or initial pairings prove suboptimal.

6.12 Data-Driven Insights & Reporting

- Advanced analytics platform providing insights into learning trends, skill demand patterns, and market opportunities.

- Predictive modeling for career path recommendations and skill development prioritization.
- Integration with industry job market data to align mentorship focus with employment opportunities.
- Comprehensive ROI tracking for both learners and mentors to demonstrate platform value.

Chapter 7

REFERENCES AND BIBLIOGRAPHY

<https://nodejs.org/docs/latest/api/>

<https://github.com/GithubNexon/React>

https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Frameworks_libraries/React_getting_started

<https://www.postgresql.org/docs/current/>

<https://expressjs.com/en/guide/routing.html>

<https://tailwindcss.com/docs/installation/using-vite>

Chapter 8

APPLICATION CODE

Location - \frontend\src

Main.jsx

```
import { StrictMode } from "react";

import { createRoot } from "react-dom/client";

import "./index.css";

import Header from "./components/Header";

import About from "./pages/about/About";

import Contact from "./pages/contact/Contact";

import Browse_mentor from "./pages/browse-mentor/Browse_mentor";

import { BrowserRouter, Route, Router, Routes } from "react-router-dom";

import Footer from "./components/Footer";

import Home from "./pages/Home/Home";

import Login from "./pages/Login/Login";

import Register from "./pages/register/Register";

import SearchResult from "./pages/browse-mentor/components/SearchResult";

import SearchSection from "./pages/browse-mentor/components/SearchSection";
```

```
import ViewMentor from './pages/ViewMentor/ViewMentor'

import Dashboard from './pages/LearnerModule/Dashboard'

// Mentor Imports

import MentorLayout from "./pages/MentorModule/MentorLayout";

import MentorDashboard from
"./pages/MentorModule/MentorDashboard/MentorDashboard";

//import contexts and providers

import AuthProvider from './context/auth.context'

import FavoriteMentors from "./pages/LearnerModule/components/FavoriteMentors";

import Settings from "./pages/LearnerModule/components/Settings";

import MySession from "./pages/LearnerModule/components/MySession";

import Overview from "./pages/LearnerModule/components/Overview";

import SetupProfile from "./pages/SetupProfile/SetupProfile";

import PersonalInfo from "./pages/SetupProfile/Components/PersonalInfo";

import WorkExperience from "./pages/SetupProfile/Components/WorkExperience";

import EducationInfo from "./pages/SetupProfile/Components/EducationInfo";

import SetYourRate from "./pages/SetupProfile/Components/SetYourRate";

import WeeklyAvailability from "./pages/SetupProfile/Components/WeeklyAvailability";
```

```
import SocialLinks from "./pages/SetupProfile/Components/SocialLinks";
import Expertise from "./pages/SetupProfile/Components/Expertise";
createRoot(document.getElementById("root")).render(
  <StrictMode>
    <AuthProvider>
      <BrowserRouter>
        <Header />
        <Routes>
          <Route path="/" element={<Home />}></Route>
          <Route path="/about-us" element={<About />}></Route>
          <Route path="/contact" element={<Contact />}></Route>
          <Route path="/browse-mentor" element={<Browse_mentor />}>
            <Route path="all-mentors" element={<SearchResult />}></Route>
            <Route path="search-mentors" element={<SearchSection />}></Route>
          </Route>
          <Route path="/login" element={<Login />}></Route>
          <Route path="/register" element={<Register />}></Route>
        </Routes>
      </BrowserRouter>
    </AuthProvider>
  </StrictMode>
)
```

```

<Route path="/view-mentor/:username" element={<ViewMentor />}></Route>

{ /* setup profile */}

<Route path="/setup-profile" element={<SetupProfile />}>

<Route path="personal-info" element={<PersonalInfo />} />

<Route path="work-experience" element={<WorkExperience />} />

<Route path="education-info" element={<EducationInfo />} />

<Route path="expertise-info" element={<Expertise />} />

<Route path="social-link" element={<SocialLinks />} />

<Route path="availability-info" element={<WeeklyAvailability />} />

<Route path="rates-info" element={<SetYourRate />} />

</Route>

{ /* Mentee module */}

<Route path="/profile/username" element={<Dashboard />}>

<Route index element={<Overview />} />

<Route path="my-session" element={<MySession />} />

<Route path="my-favorite-mentor" element={<FavoriteMentors />} />

<Route path="settings" element={<Settings />} />

```

```
</Route>

{ /* Mentor module */}

<Route path="/mentor" element={<MentorLayout />} >

<Route index element={<MentorDashboard />} />

<Route path="/mentor/view-mentor" element={<ViewMentor />} />

</Route>

<Route path="/" element={<h2>404 - Page Not Found </h2>} />

</Routes>

<Footer />

</BrowserRouter>

</AuthProvider>

</StrictMode>

);
```

Index.css

```
@import "tailwindcss";
```

)

Location - (\frontend\src\pages\about)

About.jsx

```
import Hero from './components/Hero'

import Mission from './components/Mission';

import Success from './components/Success'

function About() {

  return (
    <>
    <Hero/>

    <Success/>

    <Mission/>

  </>
);

}

export default About
```

Location - (/frontend/src/pages/about/components)

Hero.jsx

```

import React from "react";

function Hero() {

  return (
    <section data-aos-delay="0" className="w-full">
      <div className="mx-5 md:mx-28 flex justify-center items-center h-screen">
        <div className="w-full">
          <div className="flex items-center justify-center">
            <p className="font-semibold text-green-500 bg-green-100 border border-green-400 px-6 py-1 rounded-full my-5">
              About Mentora
            </p>
          </div>
        <div className="text-center ">
          <h1 className="text-3xl md:text-6xl font-bold text-black">
            Empowering <span className="bg-gradient-to-r from-gray-400 via-green-400 to-slate-500 bg-clip-text text-transparent">Growth Through Meaningful</span> Connections
          </h1>
        </div>
      </div>
    </section>
  );
}

export default Hero;

```

```
<br />

<p className="text-gray-500 text-lg">
    We believe everyone deserves access to great mentorship. Our
    platform connects ambitious individuals with experienced
    professionals who are passionate about sharing their knowledge.
</p>

</div>

</div>

</section>

);

}

export default Hero;
```

Mission.jsx

```
import React from 'react'
```

```
import aboutImg from '../../assets/aboutImg.png'

function Mission() {

  return (
    <></>

    <section data-aos-delay="0" className='w-full py-10 bg-gradient-to-b from-green-100/70 to-transparent'>

      <div className='mx-5 md:mx-28 flex justify-center items-center h-fit'>

        <div className='w-full'>

          <p className='text-black text-center text-4xl font-bold mb-10'>Our Mission</p>

          <div className='md:flex md:justify-between md:items-center gap-10'>

            <div>

              <h1 className='text-2xl font-semibold'>Democratizing Access to Quality Mentorship</h1>

              <p className='mt-4 text-lg text-gray-500'>Traditional mentorship has often been limited to those with existing networks. We're changing that by creating a platform where anyone can find the guidance they need to succeed.</p>

              <p className='mt-4 text-lg text-gray-500'>Our carefully vetted mentors come from diverse backgrounds and industries, ensuring that every mentee can find someone who understands their unique journey and challenges.</p>

            </div>

          </div>

        </div>

      </div>

    </section>
  )
}
```

Success.jsx

```
import React from 'react'  
  
import {User, Target, Award, Heart} from 'lucide-react'  
  
function Success() {
```

```
return (



    <>

    <section data-aos-delay="0" className='w-full py-10 bg-gradient-to-t from-green-100/70 to-transparent'>

        <div className='mx-5 md:mx-28 flex justify-center items-center'>

            <div className='w-full'>

                <div className='md:flex md:items-center md:justify-center gap-10'>

                    <div className='border border-indigo-50 p-10 rounded-lg bg-green-200 text-center m-3 md:m-0'>

                        <div className='w-fit mx-auto mb-3'>

                            <User size={30} className='text-green-500'/'>

                        </div>

                        <p className='font-bold'>500+</p>

                        <p className='text-black'>Active Mentores</p>

                    </div>

                    <div className='border border-indigo-50 p-10 rounded-lg bg-pink-200 text-center m-3 md:m-0'>

                        <div className='w-fit mx-auto mb-3'>

                            <Target size={30} className='text-pink-500'/'>

                        </div>

                    </div>

                </div>

            </div>

        </div>

    </section>

)
```

```
<p id='users' className='fount-bold'>2000+</p>

<p className='text-black'>Success Stories</p>

</div>

<div className='border border-indigo-50 p-10 rounded-lg bg-yellow-200 text-center m-3 md:m-0'>

  <div className='w-fit mx-auto mb-3'>

    <Award size={30} className='text-yellow-500' />

  </div>

  <p id='session' className='font-bold'>10000+</p>

  <p className='text-black'>Experts sessions</p>

  </div>

<div className='border border-indigo-50 p-10 rounded-lg bg-orange-200 text-center m-3 md:m-0'>

  <div className='w-fit mx-auto mb-3'>

    <Heart size={30} className='text-orange-500' />

  </div>

  <p id='happy' className='font-bold'>5000+</p>

  <p className='text-black'>Happy mantees</p>

  </div>

</div>
```

```
</div>

</div>

</section>

</>

)

}

export default Success
```

Location - (/frontend/src/pages/browse-mentor)

Browse_mentor.jsx

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

import Hero from './components/Hero'

import SearchSection from './components/SearchSection'

import SearchResult from './components/SearchResult'

import { Mentors } from '../../sample/MentorCard'

import MentorCard from '../../components/MentorCard'

function Browse_mentor() {

  const [filteredMentors, setFilteredMentors] = useState(Mentors);

  return (
```

```

<>

<Hero/>

<SearchSection setFilteredMentors={setFilteredMentors} />

<SearchResult filteredMentors={filteredMentors}/>

</>

);

}

export default Browse_mentor;

```

Location - (/frontend/src/pages/about/browse-mentor/components)

Hero.jsx

```

import React from 'react'

import {UsersRound, TrendingUp ,Sparkles} from 'lucide-react'

function Hero() {

return (
<section className='w-full bg-gradient-to-br from-transparent to-purple-100'

```

```

<div className='mx-5 md:mx-28 flex justify-center items-center'>

  <div className='w-full h-fit'>

    {/* section heading text */}

    <div className='mt-20 text-center'>

      <h1 className='text-3xl md:text-6xl font-bold'>Find <span className='bg-gradient-to-r
from-yellow-300 via-pink-300 to-rose-300 bg-clip-text text-transparent'>Your
Mentor</span></h1>

      <p className='text-gray-600 text-xl my-3'>Connect with industry leaders and experienced
professionals who can guide your career journey and help you achieve your goals.</p>

    </div>

    {/* platform overview content */}

    <div className='text-center w-full mx-auto md:flex md:justify-center md:items-center
md:gap-20 mt-14'>

      <div className='text-center p-10 m-2 rounded-lg overflow-hidden relative group
transition-transform duration-300 ease-in-out hover:scale-105'>

        {/* Blur + Gradient Overlay on hover */}

        <div className='absolute inset-0 bg-gradient-to-br via-cyan-300/30 to-blue-300/10
rounded-xl backdrop-blur-lg opacity-100 transition-opacity duration-300 z-0' />

      <div className='relative'>

        <div className='w-fit mx-auto mb-3'><UsersRound className=' text-pink-500' /></div>

```

```
<h1 className='text-black text-2xl font-bold z-1'>500+</h1>

<span className='text-gray-500 my-2'>Expert Mentors</span>

</div>

</div>

<div className='text-center p-13 m-2 rounded-lg overflow-hidden relative group transition-transform duration-300 ease-in-out hover:scale-105'>

 {/* Blur + Gradient Overlay on hover */}

 <div className="absolute inset-0 bg-gradient-to-br via-green-300/30 to-blue-300/10 rounded-xl backdrop-blur-lg opacity-100 transition-opacity duration-300 z-0" />

<div className='relative '>

 <div className='w-fit mx-auto mb-3'><TrendingUp className=' text-emerald-500' /></div>

<h1 className='text-black text-2xl font-bold z-1'>10k</h1>

<span className='text-gray-500 my-2'>Sessions Completed</span>

</div>

</div>

 {/* Blur + Gradient Overlay on hover */}
```

```

<div className="absolute inset-0 bg-gradient-to-br via-orange-300/30 to-blue-300/10
rounded-xl backdrop-blur-lg opacity-100 transition-opacity duration-300 z-0" />

<div className='relative '>

  <div className='w-fit mx-auto mb-3'><Sparkles className=' text-violet-500' /></div>

  <h1 className='text-black text-2xl font-bold z-1'>4.9</h1>

  <span className='text-gray-500 my-2'>Average Rating</span>

</div>

</div>

</div>

</div>

</div>

</section>

)

}

export default Hero

```

SearchResult.jsx

```

import React,{useEffect, useState} from 'react'

import MentorCard from '.././../components/MentorCard'

```

```
function SearchResult({filteredMentors = []}) {  
  
  const [currentPage, setCurrentPage] = useState(1);  
  
  const itemsPerPage = 9;  
  
  // Calculate total pages  
  
  const totalPages = Math.ceil(filteredMentors.length / itemsPerPage);  
  
  // Get paginated data  
  
  const indexOfLast = currentPage * itemsPerPage;  
  
  const indexOfFirst = indexOfLast - itemsPerPage;  
  
  const currentMentors = filteredMentors.slice(indexOfFirst, indexOfLast);  
  
  // Reset page when new search is made  
  
  useEffect((() => {  
  
    setCurrentPage(1);  
  
  }), [filteredMentors]);  
  
  const renderMentors = ()=>{  
  
    if(currentMentors.length === 0){  
  
      return (<p className='text-center text-gray-600'>No mentors found.</p>)  
  
    }else{  

```

```

return currentMentors.map((Mentor, index) => (
  <MentorCard      key={index}      name={Mentor.name}      title={Mentor.title}
    rating={Mentor.rating}      sessions={Mentor.sessions}      skills={Mentor.skills}
    hourlyRate={Mentor.hourlyRate} nextAvailable={Mentor.nextAvailable}/>
));
}

return (
  <section className='w-full'>

    <div className='mx-5 md:mx-28 flex justify-center items-center mt-5'>

      <div className='w-full'>

        <div className='header'>

          <div>

            <h1 className='text-3xl font-bold '>Available Mentors</h1>

            <p className='text-sm text-gray-400'><span className='text-2xl font-bold bg-gradient-to-br from-yellow-500/30 via-green-400/90 to-pink-500/30 bg-clip-text text-transparent'>{filteredMentors.length > 999 ? '999+' : filteredMentors.length}</span> mentor ready to help you succeed</p>

          </div>

        </div>

      </div>

    <div className='md:grid md:grid-cols-3 md:grid-rows-2 md:gap-1 mt-14'>

```

```
{renderMentors()}

</div>

</div>

</div>

<div className='flex justify-center items-center mt-4 gap-4'>

<button

  onClick={() => setCurrentPage(prev => Math.max(prev - 1, 1))}

  disabled={currentPage === 1}

  className='px-4 py-1 bg-gray-300 rounded disabled:opacity-50'

>

  Previous

</button>

<span className='text-sm'>

  Page {currentPage} of {totalPages}

</span>

<button

  onClick={() => setCurrentPage(prev => Math.min(prev + 1, totalPages))}

  disabled={currentPage === totalPages}

  className='px-4 py-1 bg-blue-300 rounded disabled:opacity-50'
```

```
>

    Next

</button>

</div>

</section>

)

}

export default SearchResult
```

SearchSection.jsx

```
import React, { useRef } from 'react';

import { UserSearch } from 'lucide-react';

import {Mentors} from '.././../sample/MentorCard'

function SearchSection({setFilteredMentors}) {

  const inputRef = useRef(null);

  const filterMentors = () => {

    const query = inputRef.current.value.toLowerCase();

    const result = Mentors.filter((mentor) => {
```

```

const nameMatch = mentor.name.toLowerCase().includes(query);

const titleMatch = mentor.title.toLowerCase().includes(query);

const skillMatch = mentor.skills.some(skill =>

  skill.toLowerCase().includes(query)

);

return nameMatch || titleMatch || skillMatch;

});

setFilteredMentors(result);

};

const addInput = (tag) => {

  inputRef.current.value = tag;

  filterMentors(); // update after setting value

};

return (
  <section className='w-full bg-gradient-to-t from-transparent to-purple-100 border-b-2 border-b-gray-200 py-10'>

    <div className='mx-5 md:mx-28 flex justify-center items-center'>

      <div className='w-full md:w-[60%] mx-auto'>

        <div className='p-3 md:p-8 bg-white rounded-2xl shadow-lg border border-indigo-300'>

```

```
<div className='flex justify-left gap-3 items-center p-3 rounded-2xl border-2 border-transparent transition-all ease-in-out duration-300 focus-within:border-black'>

    <UserSearch className='text-gray-500' />

    <input
        id='searchInput'
        ref={inputRef}
        onKeyUp={filterMentors}
        className='text-sm placeholder:text-gray-400 bg-transparent w-full focus:outline-none'
        type="text"
        placeholder='Search by name, skill, company, or expertise...'
    />

</div>

<hr className='mb-5 mt-3 text-gray-200' />

<div className='mt-6'>
    <span className='text-gray-400 text-sm'>Trending searches:</span>
    <div className='flex flex-wrap mt-2'>
        {["React", "Data", "Cloud", "Home World"].map((tag, index) => (
            <span>
```

```
key={index}

        className='bg-gradient-to-br from-gray-100 to-transparent text-xs font-
semibold rounded-full px-2 py-0.5 mr-2 mb-2 cursor-pointer'

        onClick={() => addInput(tag)}

    >

    {tag}

</span>

))}

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

);

}

export default SearchSection;
```

Location - (/frontend/src/pages/contact)

Contact.jsx

```
import React from 'react'

import Hero from './Components/Hero'

import ConectSrc from './Components/ConectSrc'

import Message from './Components/Message'

function Contact() {

  return (
    <>
    <Hero/>

    <ConectSrc/>

    <Message/>

    </>
  )
}

export default Contact
```

Location - (/frontend/src/pages/login)

Login.jsx

```
import React, { useState, useRef, useEffect, useContext } from 'react';
```

```
import { NavLink, useNavigate } from 'react-router-dom';

import axios from 'axios';

import Cookies from "js-cookie";

import Logo from '../../components/Logo';

import Loading from '../../components>Loading';

import { AuthContext } from '../../context/auth.context';

function Login() {

  const {login} = useContext(AuthContext)

  const navigate = useNavigate();

  const [formData, setFormData] = useState({ 

    usernameORemail: "",

    password: "",

  });

  const [errorCount, setErrorCount] = useState(0);

  const [timer, setTimer] = useState(0);

  const [disabled, setDisabled] = useState(false);

  const [showErrors, setShowErrors] = useState(false);

  const [isLoading, setLoading] = useState(false);

  const intervalRef = useRef(null);
```

```
const handleRedirect = ()=>{

    navigate("/");

    // window.location.href = '/';

}

// Handle Input Change

const handleChange = (e) => {

    setFormData({

        ...formData,
        [e.target.name]: e.target.value,
    });

}

// Basic Validation

const validateFields = () => {

    return formData.usernameORemail.trim() !== "" && formData.password.trim() !== "";

};

// Handle Lockout Countdown

useEffect(() => {

    if (timer > 0) {

        intervalRef.current = setInterval(() => {

```

```
setTimer((prev) => {  
  
    if (prev <= 1) {  
  
        clearInterval(intervalRef.current);  
  
        setDisabled(false);  
  
        setErrorCount(0);  
  
        return 0;  
  
    }  
  
    return prev - 1;  
  
});  
  
, 1000);  
  
}  
  
return () => clearInterval(intervalRef.current);  
  
}, [timer]);  
  
// Handle Login Submission  
  
const handleLogin = async () => {  
  
    setShowErrors(false);  
  
    if (!validateFields()) {  
  
        setShowErrors(true);  
  
        return;  
    }  
}
```

```
}

 setLoading(true);

try {

const response = await axios.post('http://localhost:3000/api/auth/login', formData,{

  withCredentials:true

});

if (response.status === 200) {

  alert(response.data.message);

  console.log(response.data.user)

  login(response.data.user)

  handleRedirect();

} else {

  alert(response.data.message || 'Invalid credentials');

  throw new Error(response.data.message || 'Invalid credentials');

}

} catch (error) {

  console.error(`Login error: ${error}`);

  setShowErrors(true);

  setErrorCount((prev) => prev + 1);
```

```
if (errorCount + 1 >= 3) {  
  
    setDisabled(true);  
  
    setTimer(Math.floor(Math.random()*100));  
  
}  
  
} finally {  
  
    setLoading(false);  
  
}  
  
};  
  
return (  
  
<section className="w-full py-10 min-h-screen flex items-center justify-center">  
  
<div className="w-full max-w-md bg-white rounded-2xl shadow-md px-6 py-8 mx-4  
bg-gradient-to-br from-indigo-100 via-yellow-50 to-pink-100">  
  
<div className="text-center mb-6">  
  
<div className="flex justify-center mb-3">  
  
<span className="w-12 h-12">  
  
<Logo />  
  
</span>  
  
</div>  
  
<h1 className="text-2xl font-bold text-gray-800">Welcome Back</h1>
```

```
<p className="text-sm text-gray-500">Sign in to continue your mentorship  
journey</p>  
  
</div>  
  
<form  
    className="space-y-4"  
    onSubmit={(e) => {  
  
        e.preventDefault();  
  
        handleLogin();  
  
    }}>  
  
>  
  
<div>  
  
<label htmlFor="usernameORemail" className="block text-xs text-gray-700 font-  
medium mb-1">  
  
    Username or Email  
  
</label>  
  
<input  
    id="usernameORemail"  
    type="text"  
    name="usernameORemail"  
    value={formData.usernameORemail} />
```

```
onChange={handleChange}

      className="w-full px-4 py-2 border border-gray-300 rounded-md focus:outline-
none focus:ring-2 focus:ring-black"

      placeholder="Enter Username or Email"

    />

  {showErrors && !formData.usernameORemail && (
    <p className="text-xs text-red-500 mt-1">Invalid username or email</p>
  )}

</div>

<div>

  <label htmlFor="password" className="block text-xs text-gray-700 font-medium
mb-1">

    Password

  </label>

  <input
    id="password"
    type="password"
    name="password"
    value={ formData.password }
    onChange={handleChange}
  >

```

```
    className="w-full px-4 py-2 border border-gray-300 rounded-md focus:outline-  
none focus:ring-2 focus:ring-black"  
  
    placeholder="Enter Password"  
  
/>  
  
{showErrors && !formData.password && (  
  
    <p className="text-xs text-red-500 mt-1">Invalid password</p>  
  
)}  
  
</div>  
  
<button  
    type="submit"  
  
    className={`w-full py-2 rounded-lg text-white font-semibold transition ${  
  
        disabled  
  
    ? 'bg-gray-200 cursor-not-allowed'  
  
    : 'bg-black hover:bg-black/80'  
  
}`}  
  
disabled={disabled}  
  
>  
  
{isLoading ? <Loading /> : 'Login'}  
  
</button>
```

```
{disabled && (  
  
  <p className="text-xs text-gray-500 text-center mt-2">  
  
    You can try again after <span>{timer}</span> seconds  
  
  </p>  
  
)}  
  
</form>  
  
<div className="text-center mt-1">  
  
  <NavLink to="/reset" className="text-blue-600 text-sm hover:underline">  
  
    Forgot Password?  
  
  </NavLink>  
  
</div>  
  
<div className="mt-3 flex items-center justify-center gap-1 text-sm bg-gray-50/30  
border border-gray-300 py-2 rounded-md">  
  
  <p>Don't have an account?</p>  
  
  <NavLink to="/register" className="text-blue-600 font-medium hover:underline">  
  
    Create Account  
  
  </NavLink>  
  
</div>  
  
</div>
```

```
</section>

);

}

export default Login;
```

Location - (/frontend/src/pages/MentorModule)

MentorLayout.jsx

```
import React from 'react'

import { Outlet } from 'react-router-dom'

const MentorLayout = () => {

  return (

    <div>

      <main className='p-16 bg-white min-w-screen'>

        <Outlet />

      </main>

    </div>
  )
}

export default MentorLayout
```

Location - (/frontend/src/pages/MentorModule/MentorDashboard)

MentorDashboard.jsx

```
import { useState } from "react";

import React from "react";

import OverviewCards from "./components/OverviewCards";

import MentorReviews from "./components/MentorReviews";

const MentorDashboard = (props) => {

  const [activeTab, setActiveTab] = useState("Overview");

  const tabs = ["Overview", "Sessions", "Requests", "Earnings"];

  const receivedData = { };

  const renderBottomCard = () => {

    switch (activeTab) {

      case 'Overview':

        return <OverviewCards/>

      case 'Sessions':

        return <MentorReviews />

      case 'Availability':

        return <MentorAvailability />
    }
  }
}
```

default:

```
    return <p>Tab content not found</p>

}

}

return (

<>

<div className="page-background bg-white w-auto min-h-screen">

<div className="center-it p-12 flex flex-col gap-4">

<p className="text-4xl font-bold">

Welcome back,{ " " }

<span className="bg-gradient-to-r from-secondary-500 to-primary-600 text-transparent bg-clip-text">

{props.Mentor_name || "John"}

</span>

</p>

/* Page subtext */

<p className=" text-m-gray-500">

Manage your mentoring sessions and track your impact

</p>
```

```
<div className="tabs-container">

  <div className="flex p-1 bg-gray-50 rounded-full overflow-x-auto ">

    {tabs.map((tab) => (

      <button

        key={tab}

        onClick={() => setActiveTab(tab)}

        className={`${px-4 py-2 w-full font-medium transition rounded-full ${

          activeTab === tab

            ? "bg-white text-black"

            : "text-gray-500 hover:text-black"

        }}`}

      >

        {tab}

      </button>

    ))}

  </div>

</div>

<div className="">

  {renderBottomCard() }


```

```
</div>

</div>

</div>

</>

);

};

export default MentorDashboard;
```

Location -

(/frontend/src/pages/MentorModule/MentorDashboard/components)

MentorReviews.jsx

```
import React from 'react'

import MentorReviewCard from '../../../../../components/MentorReviewCard'

const MentorReviews = () => {

  return (

    <div>

      <MentorReviewCard />

      <MentorReviewCard />

    </div>
  )
}

export default MentorReviews;
```

```
<MentorReviewCard />

<MentorReviewCard />

<MentorReviewCard />

<MentorReviewCard />

</div>

)

}

export default MentorReviews
```

OverviewCards.jsx

```
import React from 'react'

import { CalendarCheck, IndianRupee, Star, Users } from "lucide-react";

const OverviewCards = (receivedData) => {

  return (
    <>
      {/* Dashboard cards Container */}

      <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-2 gap-6">
        {/* Dashboard Card 2 */}
    
```

```
<div className="dashboard-card shadow-sm p-6 bg-gradient-to-tl from-secondary-200 to bg-secondary-300 to-tertiary-300 w-full rounded-2xl h-full flex items-center gap-6 ">

  {/* Dashboard Card icon wrapper */}

  <div className="p-3 w-fit h-fit rounded-2xl bg-secondary-200">

    <IndianRupee className="w-[2rem] h-[2rem] stroke-secondary-700" />

  </div>

  {/* Dashboard Card text */}

  <div className="dashboard-card-1-text flex flex-col h-auto">

    <p className="dashboard-card-1-heading text-base text-m-gray-700">

      Earnings this month

    </p>

    <p className="dashboard-card-1-content text-3xl text-black font-bold">

      {receivedData.sessionsAttended || 0}

    </p>

  </div>

</div>

<div className="dashboard-card shadow-sm p-6 bg-gradient-to-tl from-primary-200 to bg-primary-300 to-tertiary-100 w-full rounded-2xl h-full flex items-center gap-8 sm:p-3">

  {/* Dashboard Card icon wrapper */}

  <div className="p-3 w-fit h-fit rounded-2xl bg-primary-100">
```

```
<Users className="w-[2rem] h-[2rem] stroke-primary-600" />

</div>

{/* Dashboard Card text */}

<div className="dashboard-card-1-text flex flex-col h-auto">

  <p className="dashboard-card-1-heading text-base text-m-gray-700">

    Sessions

  </p>

  <p className="dashboard-card-1-content text-3xl text-black font-bold">

    {receivedData.sessions || 0}

  </p>

</div>

</div>

<div className="dashboard-card shadow-sm p-6 bg-gradient-to-tl from-orange-200 to bg-orange-300 to-white w-full rounded-xl h-full flex items-center gap-8 ">

  {/* Dashboard Card icon wrapper */}

  <div className="p-3 w-fit h-fit rounded-2xl bg-orange-100">

    <Star className="w-[2rem] h-[2rem] stroke-orange-400" />

  </div>

  {/* Dashboard Card text */}
```

```

<div className="dashboard-card-1-text flex flex-col h-auto">

  <p className="dashboard-card-1-heading text-base text-m-gray-700">
    Avg. Rating
  </p>

  <p className="dashboard-card-1-content text-3xl text-black font-bold">
    {receivedData.average_rating || 0}
  </p>

  </div>

</div>

</>

)
}

export default OverviewCards

```

Location - (/frontend/src/pages/ProfileModule/MenteeProfile)

Dashboard.jsx

```

import React, { useContext } from 'react';

import { NavLink, Outlet } from 'react-router-dom';

```

```

import { AuthContext } from '../../../../../context/auth.context';

function DashboardLayout() {

  const { user } = useContext(AuthContext);

  const tabs = [
    { name: "Overview", path: "" },
    { name: "My Sessions", path: 'my-session' },
    { name: "Favorites Mentor", path: 'my-favorite-mentor' },
    { name: "Settings", path: 'settings' }
  ];

  return (
    <section className="w-full bg-gradient-to-t from-transparent via-green-50 to-transparent py-10">
      <div className="mx-5 md:mx-28 flex justify-center items-center">
        <div className="w-full">
          <h1 className="text-3xl text-gray-800 font-bold">
            Welcome back, <span className='bg-gradient-to-br from-rose-500 to-indigo-500 bg-clip-text text-transparent'>{user.name}</span>
          </h1>
          <p className="text-gray-500">
            Continue your learning journey with personalized mentorship.
          </p>
        </div>
      </div>
    </section>
  );
}

```

```

</p>

{ /* Tabs */}

<div className="w-4/6 flex p-1.5 bg-white rounded-2xl overflow-x-auto my-10">

  {tabs.map((tab) => (
    <NavLink
      to={tab.path}
      key={tab.name}
    end
    className={({ isActive }) =>
      `px-4 py-2 w-full text-center transition rounded-xl ${

        isActive
        ? "bg-gray-200 font-semibold"
        : "text-gray-500 hover:text-black"
      }`}

    }
  >
    {tab.name}
  </NavLink>
))}
```

```

        </div>

        {/* Dynamic Content based on Route */}

        <Outlet />

    </div>

    </div>

</section>

);

}

export default DashboardLayout;

```

Location - (/frontend/src/pages/contact/components)

ContactSrc.jsx

```

import React from 'react'

import {Mail, Phone, MapPin, Clock} from 'lucide-react'

function ContactSrc() {

    return (

        <section data-aos-delay="0" className='w-full bg-gradient-to-b from-pink-100 to-transparent'>

        <div className='mx-5 md:mx-28 flex justify-center items-center'>

            <div className='w-full md:flex md:justify-center md:items-center md:gap-5 h-fit py-10'>

```

```
<div className='relative before:content-[""] before:absolute before:left-1/2 before:-translate-x-1/2 before:w-[100px] before:h-[100px] before:rounded-full before:bg-yellow-200 z-0 m-5 md:m-0'>

<div className='p-10 border border-green-200 rounded-lg backdrop-blur-2xl text-center'>

<div className='w-fit mx-auto mb-3'>

<Mail />

</div>

<p className='text-sm font-semibold'>Email Us</p>

<p className='text-gray-500 text-xs mt-1'>mentor@mentor.com</p>

</div>

</div>

<div className='relative before:content-[""] before:absolute before:left-1/2 before:-translate-x-1/2 before:w-[100px] before:h-[100px] before:rounded-full before:bg-pink-200 z-0 m-5 md:m-0'>

<div className='p-10 border border-green-200 rounded-lg backdrop-blur-2xl text-center'>

<div className='w-fit mx-auto mb-3'>

<Phone />

</div>

<p className='text-sm font-semibold'>Call Us</p>

<p className='text-gray-500 text-xs mt-1'>+91 55555-33333</p>
```

```
</div>

</div>

<div className='relative before:content-[""] before:absolute before:left-1/2 before:-translate-x-1/2 before:w-[100px] before:h-[100px] before:rounded-full before:bg-green-200 z-0 m-5 md:m-0'>

<div className='p-10 border border-green-200 rounded-lg backdrop-blur-2xl text-center'>

<div className='w-fit mx-auto mb-3'>

<MapPin />

</div>

<p className='text-sm font-semibold'>Visit Us</p>

<p className='text-gray-500 text-xs mt-1'>GB Road Near Naresh House, Dehli</p>

</div>

</div>

<div className='relative before:content-[""] before:absolute before:left-1/2 before:-translate-x-1/2 before:w-[100px] before:h-[100px] before:rounded-full before:bg-indigo-200 z-0 m-5 md:m-0'>

<div className='p-10 border border-green-200 rounded-lg backdrop-blur-2xl text-center'>

<div className='w-fit mx-auto mb-3'>

<Clock />

</div>
```

```
<p className='text-sm font-semibold'>Business Hours</p>

<p className='text-gray-500 text-xs mt-1'>Mon-Fri: 9Am - 5Pm IST</p>

</div>

</div>

</div>

</div>

</section>

)

}

export default ContextSrc
```

Hero.jsx

```
import React from "react";

function Hero() {

  return (
    <section
      data-aos-delay="0"
      className="w-full py-10 bg-gradient-to-b from-transparent to-pink-100">
    </section>
  );
}
```

```
<div className="mx-5 md:mx-28 flex justify-center items-center">

  <div className="w-full text-center">

    <span className="text-pink-600 border border-pink-400 rounded-full bg-pink-100
    py-1 px-5 font-semibold text-sm">

      Get In Touch

    </span>

    <h1 className="bg-gradient-to-r from-yellow-300 via-pink-400 to-red-400 bg-clip-
    text text-transparent font-bold text-3xl md:text-6xl mt-5">

      We'd Love to Hear From You

    </h1>

    <p className="mt-3 text-gray-500 text-lg">

      Have questions, feedback, or need support? Our team is here to help

      You make the most of your mentorship journey.

    </p>

    </div>

  </div>

</section>

);

}

export default Hero;
```

Message.jsx

```
import React from "react";  
  
function Message() {  
  return (  
    <>  
    <section  
      data-aos-delay="0"  
      className="w-full bg-gradient-to-b from-transparent via-green-100 to-transparent"  
    >  
    <div className="mx-5 md:mx-28 flex justify-center items-center h-fit">  
      <div className="w-full md:flex md:justify-center md:items-center md:gap-10">  
        <div className="mb-10 md:mb-0">  
          <h1 className="text-black font-bold text-3xl mb-5">  
            Send Us A Message  
          </h1>  
          <p className="text-gray-600">  
            Whether you're a mentee looking for guidance, a mentor wanting  
          </p>  
        </div>  
      </div>  
    </div>  
  );  
}  
  
export default Message;
```

to join our platform, or a business interested in

partnerships, we're here to help.

</p>

```
<div className="mt-6 flex justify-left items-start gap-5">
```

```
  <div className="w-[40px] h-[25px] md:w-[40px] flex justify-center items-center bg-indigo-100 rounded-full mt-1.5">
```

```
    <p className="md:text-lg font-semibold text-indigo-700">1</p>
```

</div>

<div>

```
  <p className="font-semibold text-lg">Quick Response</p>
```

```
  <p className="text-sm text-gray-600 mt-0.5">
```

We are Typically response to all the inquiries within 24 hours

during business days{ " "}

</p>

</div>

</div>

```
<div className="mt-6 flex justify-left items-start gap-5">
```

```
  <div className="w-[40px] h-[25px] md:h-[40px] flex justify-center items-center bg-yellow-100 rounded-full mt-1.5">
```

```
    <p className=" md:text-lg font-semibold text-yellow-700">2</p>
```

```
</div>

<div>

  <p className="font-semibold text-lg">Personalized Support</p>

  <p className="text-sm text-gray-600 mt-0.5">

    Every message is read by our team to ensure you get the most relevant help.

  </p>

</div>

</div>

<div className="mt-6 flex justify-left items-start gap-5">

  <div className="w-[33px] md:w-[40px] h-[25px] md:h-[40px] flex justify-center items-center bg-red-100 rounded-full mt-1.5">

    <p className=" md:text-lg font-semibold text-red-700">3</p>

  </div>

<div>

  <p className="font-semibold text-xl">Follow Up</p>

  <p className="text-sm text-gray-600 mt-0.5">

    We'll follow up to make sure your questions are fully answered.

  </p>

</div>
```

```
</div>

</div>

<form action="" className="p-5 rounded-lg bg-white">

  <h2 className="mb-5 font-bold text-2xl">Contact From</h2>

  <div className="mb-3 flex justify-between items-center gap-5 w-full overflow-hidden">

    <div>

      <label

        htmlFor="name"

        className="text-xs font-medium text-gray-700 mb-1"

      >

        Name

      </label>

      <input

        type="text"

        id="name"

        className="border border-gray-200 py-1.5 px-4 rounded-lg w-40"

        placeholder="Your Name"

      />
```

```
</div>

<div>

    <label
        htmlFor="name"
        className="text-xs font-medium text-gray-700 mb-1">
        Name
    </label>

    <input
        type="text"
        id="name"
        className="border border-gray-200 py-1.5 px-4 rounded-lg w-40"
        placeholder="Your Name"
    />

</div>

</div>

<div className="mb-3">
```

<label

```
htmlFor="email"

className="text-xs font-medium text-gray-700 mb-1"

>

Email

</label>

<input

type="email"

id="email"

className="border border-gray-200 py-1.5 px-4 rounded-lg w-full"

placeholder="Example@gmail.com"

/>

</div>

<div className="mb-3">

<label

htmlFor="subject"

className="text-xs font-medium text-gray-700 mb-1"

>

Subject

</label>
```

```
<input  
    type="text"  
    id="subject"  
  
    className="border border-gray-200 py-1.5 px-4 rounded-lg w-full"  
    placeholder="What is this about"  
/>  
  
</div>  
  
<div className="mb-3">  
  
    <label  
        htmlFor="message"  
        className="text-xs font-medium text-gray-700 mb-1"  
    >  
  
        Message  
    </label>  
  
    <textarea  
        id="message"  
        className="border border-gray-200 py-1.5 px-4 rounded-lg w-full"  
        rows={4}  
        placeholder="Tell us more about how can help you...">  
    </textarea>
```

```
></textarea>

</div>

<button

  type="submit"

    className="text-white p-2.5 border rounded-lg w-full bg-black hover:bg-
black/80"

>

  Send Message

</button>

<br />

</form>

</div>

</div>

</section>

</>

);

}

export default Message;
```

Location - (/frontend/src/pages/Home)

Home.jsx

```
import React, {useEffect} from 'react'

import HeroSection from './components/HeroSection'

import WhyMentora from './components/WhyMentora'

import FeaturedMentors from './components/FeaturedMentors'

import AOS from 'aos';

import 'aos/dist/aos.css';

function Home() {

  useEffect(() => {

    AOS.init({

      easing: 'ease-in-out',

    });

  }, []);

  return (

    <main>

    {/* Add hero section */}

    <HeroSection/>

    {/* Add Why chouse us section*/}

  )
}
```

```
<WhyMentora/>

{ /* add Featured Mentors */}
```

```
<FeaturedMentors/>
```

```
</main>
```

```
)
```

```
}
```

```
export default Home
```

Location - (/frontend/src/pages/Home/components)

FeautredMentors.jsx

```
import React from 'react'
```

```
import {useNavigate} from 'react-router-dom'
```

```
import MentorCard from '../.../components/MentorCard'
```

```
import { Mentors } from '../../.../sample/MentorCard';
```

```
function FeaturedMentors() {
```

```
  const navigate = useNavigate();
```

```
  const renderMentors = ()=>{
```

```
    const newSlice = Mentors.slice(0,9);
```

```
    return newSlice.map((Mentor, index) => (
```

```

        <MentorCard      key={index}      name={Mentor.name}      title={Mentor.title}
rating={Mentor.rating}      sessions={Mentor.sessions}      skills={Mentor.skills}
hourlyRate={Mentor.hourlyRate} nextAvailable={Mentor.nextAvailable}/>
);

}

const AllMentors = ()=>{

  navigate('/browse-mentor/all-mentors')

}

return (

<section data-aos-delay="0" className='w-full py-10 bg-gradient-to-t from-indigo-transparent via-green-100/40 to-transparent'>

<div className='mx-5 py-10 md:mx-28 flex justify-center items-center'>

<div className='w-full'>

<div className='text-center'>

<h1  className='font-bold text-3xl md:text-4xl'  data-aos="fade-up"  data-aos-delay="100">Featured Mentors</h1>

<p  className='my-3 text-gray-500 text-xl'  data-aos="fade-up"  data-aos-delay="200">Connect with industry experts who are ready to guide you towards success</p>

</div>

<div className='md:grid md:grid-cols-3 md:grid-rows-2 md:gap-1 mt-14'>

{renderMentors()}

```

```
</div>

<div className='text-center'>

  {Mentors.length>9?(<button onClick={AllMentors} className='underline underline-offset-6 mt-10 cursor-pointer transition-all ease-in-out duration-300 hover:scale-104'><span>View All Mentors</span></button>

):""}

</div>

</div>

</div>

</section>

)

}

export default FeaturedMentors
```

HeroSection.jsx

```
import React from 'react'

import { useNavigate } from 'react-router-dom';

function HeroSection() {

  const navigate = useNavigate();

  const AllMentors = ()=>{
```

```

navigate('/browse-mentor/search-mentors')

}

return (
  <section data-aos="fade-in" data-aos-duration="1000" data-aos-delay="100"
  className='w-full bg-gradient-to-b from-transparent to-orange-100' >

    <div className='mx-5 md:mx-28 flex justify-center items-center h-screen'>

      <div className='w-full'>

        {/* section heading text */}

        <div className='text-center'>

          <h1 className='text-3xl md:text-6xl font-bold' >Connect with <span className='bg-
gradient-to-r from-indigo-500 via-purple-500 to-orange-500 bg-clip-text text-
transparent'>Expert Mentors</span> for 1:1 Sessions</h1>

          <p className='text-gray-500 text-xl my-3' >Book personalized mentorship sessions with
industry experts. Get guidance, accelerate your career, and achieve your goals.</p>

        </div>

        {/* section CTA buttons */}

        <div className='flex justify-center items-center gap-2 md:gap-4 mt-8'>

          <button className='flex justify-center items-center gap-1 md:gap-3 bg-gradient-to-r from-
indigo-500 to-purple-600 py-2 px-3 md:px-5 rounded text-white cursor-pointer transition
ease-in-out duration-300 hover:bg-gradient-to-tr hover:from-indigo-500 hover:to-orange-
300'>

```

```

<svg xmlns="http://www.w3.org/2000/svg" width="22" height="22" viewBox="0 0 24 24"
fill="none" stroke="currentColor" strokeWidth="2" strokeLinecap="round"
strokeLinejoin="round" className="lucide lucide-calendar-icon lucide-calendar"><path
d="M8 2v4"/><path d="M16 2v4"/><rect width="18" height="18" x="3" y="4"
rx="2"/><path d="M3 10h18"/></svg>

<span>Book a Session</span>

</button>

<button onClick={AllMentors} className='border border-indigo-100 border-1 bg-white/70
py-2.5 px-3 md:px-5 rounded text-indigo-600 cursor-pointer hover:bg-indigo-50 font-
semibold'>

<span>Browse mentors</span>

</button>

</div>

/* platform overview content */

<div className='text-center w-full flex justify-between md:justify-center items-center
md:gap-40 mt-14'>

<div className='items-center'>

<h1 className='text-orange-500 text-4xl font-semibold'>500+</h1>

<span className='text-gray-500 my-2'>Expert Mentors</span>

</div>

<div className='items-center'>

<h1 className='text-orange-500 text-4xl font-semibold'>10K</h1>

```

```
<span className='text-gray-500 my-2'>Sessions Completed</span>

</div>

<div className='items-center'>

  <h1 className='text-orange-500 text-4xl font-semibold'>4.9</h1>

  <span className='text-gray-500 my-2'>Average Rating

  </span>

</div>

</div>

</div>

</div>

</section>

)

}

export default HeroSection
```

WhyMentora.jsx

```
import React from 'react'
```

```
function WhyMentora() {
```

```
  return (
```

```

<section data-aos-delay="0" className='w-full py-10 bg-gradient-to-b from-orange-100 to-transparent'>

<div className='mx-5 md:mx-28 flex justify-center items-center'>

<div className='w-full'>

<div className='text-center'>

<h1 className='text-3xl font-bold md:text-4xl' data-aos="fade-up" data-aos-delay="100">Why Choose Mentora</h1>

<p className='my-4 text-xl text-gray-500' data-aos="fade-up" data-aos-delay="200">We make it easy to find, book, and learn from the best mentors in your field</p>

</div>

<div className='md:flex justify-between items-center mt-20'>

<div data-aos="fade-up" data-aos-delay="400" className='text-center p-5 m-2 rounded-xl overflow-hidden relative group transition-transform duration-300 ease-in-out hover:scale-105'>

{/* Blur + Gradient Overlay on hover */}

<div className="absolute inset-0 bg-gradient-to-br from-purple-300/10 via-purple-300/10 to-blue-300/10 rounded-xl backdrop-blur-lg opacity-0 group-hover:opacity-100 transition-opacity duration-300 z-0" />

<div className='relative'>

<div className='w-fit p-4 bg-purple-200/40 rounded-xl text-purple-600 mx-auto'>

<svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor" strokeWidth="2" strokeLinecap="round">

```

```

strokeLinejoin="round"   className="lucide lucide-calendar-heart-icon lucide-calendar-heart">

<path d="M3 10h18V6a2 2 0 0 0-2-2H5a2 2 0 0 0-2 2v14a2 2 0 0 0 2 2h7" />

<path d="M8 2v4" />

<path d="M16 2v4" />

<path d="M21.29 14.7a2.43 2.43 0 0 0-2.65-.52c-.3.12-.57.3-.8.53l-.34.34-.35-.34a2.43
2.43 0 0 0-2.65-.53c-.3.12-.56.3-.79.53-.95.94-1 2.53.2 3.74L17.5 22l3.6-3.55c1.2-1.21 1.14-
2.8.19-3.74Z" />

</svg>

</div>

<h2 className='text-black my-3 text font-semibold'>Easy Scheduling</h2>

<p className='text-gray-500 text-sm my-2'>Book sessions with mentors instantly using our integrated calendar system. No back-and-forth emails.</p>

</div>

</div>

<div data-aos="fade-up" data-aos-delay="800" className='text-center p-5 m-2 rounded-xl overflow-hidden relative group transition-transform duration-300 ease-in-out hover:scale-105'>

/* Blur + Gradient Overlay on hover */

<div className="absolute inset-0 bg-gradient-to-br from-purple-300/10 via-cyan-300/10 to-blue-300/10 rounded-xl backdrop-blur-lg opacity-0 group-hover:opacity-100 transition-opacity duration-300 z-0" />

```

```

<div className='relative'>

  <div className='w-fit p-4 bg-cyan-200/40 rounded-xl text-cyan-600 mx-auto'>

    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24"
      fill="none" stroke="currentColor" strokeWidth="2" strokeLinecap="round"
      strokeLinejoin="round" className="lucide lucide-users-icon lucide-users"><path d="M16
      21v-2a4 4 0 0 0-4-4H6a4 4 0 0 0-4 4v2"/><path d="M16 3.128a4 4 0 0 1 0 7.744"/><path
      d="M22 21v-2a4 4 0 0 0-3-3.87"/><circle cx="9" cy="7" r="4"/></svg>

  </div>

  <h2 className='text-black my-3 text font-semibold'>Expert Mentors</h2>

  <p className='text-gray-500 text-sm my-2'>Access to industry professionals from top
  companies like Google, Meta, Netflix, and innovative startups.</p>

</div>

</div>

<div data-aos="fade-up"

  data-aos-delay="1200" className='text-center p-5 m-2 rounded-xl overflow-hidden
  relative group transition-transform duration-300 ease-in-out hover:scale-105'>

  {/* Blur + Gradient Overlay on hover */}

  <div className="absolute inset-0 bg-gradient-to-br from-purple-300/10 via-orange-300/10
  to-blue-300/10 rounded-xl backdrop-blur-lg opacity-0 group-hover:opacity-100 transition-
  opacity duration-300 z-0" />

  <div className='relative'>

    <div className='w-fit p-4 bg-orange-200/40 rounded-xl text-orange-600 mx-auto'>

```

```
<svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24"
fill="none" stroke="currentColor" strokeWidth="2" strokeLinecap="round"
strokeLinejoin="round" className="lucide lucide-sparkles-icon lucide-sparkles"><path
d="M9.937 15.5A2 2 0 0 0 8.5 14.063l-6.135-1.582a.5.5 0 0 1 0-.962L8.5 9.936A2 2 0 0 0
9.937 8.5l1.582-6.135a.5.5 0 0 1 .963 0L14.063 8.5A2 2 0 0 0 15.5 9.937l6.135 1.581a.5.5 0
0 1 0 .964L15.5 14.063a2 2 0 0 0-1.437 1.437l-1.582 6.135a.5.5 0 0 1-1.963 0z"/><path
d="M20 3v4"/><path d="M22 5h-4"/><path d="M4 17v2"/><path d="M5 18H3"/></svg>
```

</div>

Quality Assurance

`<p className='text-gray-500 text-sm my-2'>All mentors are vetted and rated by the community. Read reviews and choose the perfect match.</p>`

</div></div>

<div data-aos="fade-up">

>

```
{/* Blur + Gradient Overlay on hover */}
```

```
<div className="absolute inset-0 bg-gradient-to-br from-purple-300/10 via-green-300/10  
to-blue-300/10 rounded-xl backdrop-blur-lg opacity-0 group-hover:opacity-100 transition-  
opacity duration-300 z-0" />
```

```
<div className='relative'>
```

```
<div className='w-fit p-4 bg-green-200/40 rounded-xl text-green-600 mx-auto'>
```

```
<svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24"
fill="none" stroke="currentColor" strokeWidth="2" strokeLinecap="round"
strokeLinejoin="round" className="lucide lucide-shield-check-icon lucide-shield-"
```

```
check"><path d="M20 13c0 5-3.5 7.5-7.66 8.95a1 1 0 0 1-67-.01C7.5 20.5 4 18 4 13V6a1 1 0 0 1 1-1c2 0 4.5-1.2 6.24-2.72a1.17 1.17 0 0 1 1.52 0C14.51 3.81 17 5 19 5a1 1 0 0 1 1z"/><path d="m9 12 2 2 4-4"/></svg>

</div>

<h2 className='text-black my-3 text font-semibold'>Secure Payments</h2>

<p className='text-gray-500 text-sm my-2'>Safe and secure payment processing. Only pay after your session is completed successfully.</p>

</div></div>

</div>

</div>

</div>

</div>

</section>

)

}

export default WhyMentora
```