FULL STACK MERN DEVELOPMENT

Submitted in accordance with the requirement for the degree of

BACHELOR OF TECHNOLOGY IN

COMPUTER SCIENCE AND ENGINEERING

Under the faculty Guideship of

DR.G.VENKATESWARA RAO SIR,PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY

(Autonomous)

Approved by AICTE- New Delhi, Accredited by NAAC with 'A' Grade

Permanently Affiliated to JNTUK, Kakinada

NH-5, Chowdavaram, Guntur - 522019.

2021 - 2025

FULL STACK DEVELOPMENT WITH MERN

PROJECT DOCUMENTATION

PROJECT TITLE:

LearnHub : Your Centre for Skill Enhancement

TEAM MEMBERS

K. Ramya

(218x1a0535)

B.Sireesha

(218x1a0506)

B.Venkata Raghavendra

(228x5a0501)

B.Surendra Babu

(218x1a0502)

TABLE OF CONTENT:

1. PROJECT OVERVIEW:

LearnHub: Your Center for Skill Enhancement

1.1 INTRODUCTION:

An online learning platform(OLP) is a digital platform that provides a variety of tools and resources to facilitate learning and education over the internet. These platforms have become increasingly popular, especially in recent years, as they offer flexibility and accessibility for learners of all ages and backgrounds. Here are some key features and a description of an online learning platform:

User-Friendly Interface:

Online learning platforms typically have an intuitive and user-friendly interface that makes it easy for learners, regardless of their technical proficiency, to navigate and access the content.

Course Management:

Instructors or course creators can upload, organize, and manage course materials. Learners can enroll in courses and track their progress.

Interactivity:

Many platforms include interactive elements like discussion forums, chat rooms, and live webinars, which foster communication and collaboration among learners and instructors.

Certification:

Learners can earn certificates or badges upon completing courses or meeting certain criteria, which can be valuable for employment or further education.

Accessibility:

Content is often accessible on various devices, including computers, tablets, and smartphones, making learning possible from anywhere with an internet connection.

Self-Paced Learning:

Learners can typically access course materials at their own pace. This flexibility allows for learning that fits into individual schedules and preferences.

Payment and Subscription Options:

There may be free courses, but some content may require payment or a subscription. Platforms often offer multiple pricing models.

Scenario-based Case Study:

Scenario: Learning a New Skill

User Registration:

Sarah, a student interested in learning web development, visits the Online Learning Platform and creates an account. She provides her email and chooses a password.

Browsing Courses:

Upon logging in, Sarah is greeted with a user-friendly interface displaying various courses categorized by topic, difficulty level, and popularity.

She navigates through the course catalog, filtering courses by name and category until she finds a "Web Development Fundamentals" course that interests her.

Enrolling in a Course:

Sarah clicks on the course and reads the course description, instructor details, and syllabus. Impressed, she decided to enroll in the course.

After enrolling, Sarah can access the course materials, including video lectures, reading materials, and assignments.

Learning Progress:

Sarah starts the course and proceeds through the modules at her own pace. The platform remembers her progress, allowing her to pick up where she left off if she needs to take a break.

Interaction and Support:

Throughout the course, Sarah engages with interactive elements such as discussion forums and live webinars where she can ask questions and interact with the instructor and other learners.

Course Completion and Certification:

After completing all the modules and assignments, Sarah takes the final exam. Upon passing, she receives a digital certificate of completion, which she can download and add to her portfolio.

Paid Courses:

Sarah discovers an advanced web development course that requires payment. She purchases the course using the platform's payment system and gains access to premium content.

Teacher's Role:

Meanwhile, John, an experienced web developer, serves as a teacher on the platform. He creates and uploads new courses on advanced web development topics, adds sections to existing courses, and monitors course enrollments.

Admin Oversight:

The admin oversees the entire platform, monitoring user activity, managing course listings, and ensuring smooth operation. They keep track of enrolled students, handle any issues that arise, and maintain the integrity of the platform.

1.2 PURPOSE:

The objective of the "Online Learning Platform" is to revolutionize the field of education by harnessing digital technology. It aims to make high-quality educational resources accessible to learner's worldwide, breaking down geographical barriers. The platform focuses on creating interactive, adaptable, and personalized learning experiences, catering to the diverse needs of students. Ultimately, the goal is to ensure inclusivity, affordability, and lifelong learning opportunities, fostering individual growth and contributing to the advancement of society in the digital age.

1.3 FEATURES:

1. Personalized Learning Paths

- **Customized Curriculum:** Tailor learning journeys based on user preferences, current skills, and career goals.
- **Skill Assessments:** Offer initial assessments to identify strengths and gaps, providing personalized recommendations.
- **Progress Tracking:** Real-time tracking of courses completed, skills gained, and areas needing improvement.

2. Interactive Courses

- **Video Lessons & Webinars:** High-quality instructional videos covering a wide range of skills (technical, creative, soft skills, etc.).
- **Hands-On Projects:** Interactive projects that allow users to apply what they've learned in real-world scenarios.
- **Quizzes & Challenges:** Frequent assessments, quizzes, and challenges to ensure users retain knowledge.

3. Peer Collaboration and Networking

- **Discussion Forums:** Community spaces where users can engage in discussions, ask questions, and share insights.
- **Skill Certifications:** Provide verified certifications upon completion of courses, allowing users to showcase their skills to employers.
- **Gamified Badges:** Implement a badge system to recognize achievements, motivate users, and make learning fun.

4. Certifications & Badges

- **Group Projects:** Opportunities for users to collaborate on projects, fostering teamwork and real-world problem-solving.
- **Mentorship Programs:** Allow experienced professionals to mentor learners, offering guidance and feedback on career development.

2.ARCHITECTURE:

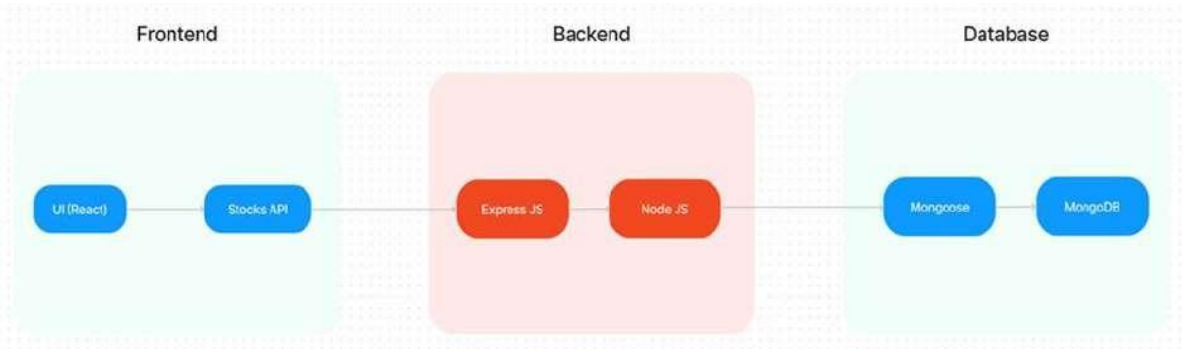


Fig: Architecture of OLP

The technical architecture of OLP app follows a client-server model, where the frontend serves as the client and the backend acts as the server. The frontend encompasses not only the user interface and presentation but also incorporates the axios library to connect with backend easily by using RESTful Apis.

The front end utilizes the bootstrap and material UI library to establish a real-time and better UI experience for any user.

On the backend side, we employ Express.js frameworks to handle the server-side logic and communication.

For data storage and retrieval, our backend relies on MongoDB. MongoDB allows for efficient and scalable storage of user data and necessary information about the place.

Together, the frontend and backend components, along with Express.js, and MongoDB, form a comprehensive technical architecture for our OLP app. This architecture enables real-time communication, efficient data exchange, and seamless integration, ensuring a smooth and immersive blogging experience for all users.

SYSTEM ANALYSIS:

Existing System:

The current system lacks a structured platform for seniors to assist juniors, leading to fragmented support and missed opportunities for knowledge sharing. Mentoring often occurs informally, hindering consistent guidance. The absence of a centralized system results in juniors facing challenges alone, while seniors' expertise remains underutilized. This necessitates the development of an organized online mentorship platform to enhance academic support and collaboration within the college community.

Disadvantages:

1.Lack of Consistency: In the absence of a structured platform, mentoring interactions between seniors and juniors may lack consistency. This inconsistency can lead to varying levels of support and guidance for juniors, resulting in gaps in their academic development.

2. Limited Access to Expertise: Without a centralized system, juniors may struggle to identify and connect with seniors possessing the specific expertise they require. This limitation can hinder juniors' ability to access comprehensive guidance and assistance for their academic endeavors.

3. Inefficient Communication: Informal mentoring often relies on ad-hoc communication methods, such as in-person meetings or email exchanges, which can be inefficient and time-consuming. Without streamlined communication channels, juniors may experience delays in receiving timely support and responses to their queries.

Proposed System:

An online mentorship platform connecting junior and senior students through domain-specific profiles, facilitating academic guidance, project assistance, and mental support. Integrated communication channels including real-time chat, forums, and video conferencing ensure engagement. Token-based feedback mechanisms incentivize mentorship, while also reinforcing seniors' foundational knowledge. Ethical considerations and data privacy are prioritized throughout development and deployment.

Advantages:

1.Enhanced Academic Support:

The proposed mentorship platform provides juniors with direct access to senior students possessing expertise in various academic domains. This facilitates tailored guidance, assistance, and clarification of doubts, thereby enhancing juniors' understanding of coursework and project requirements.

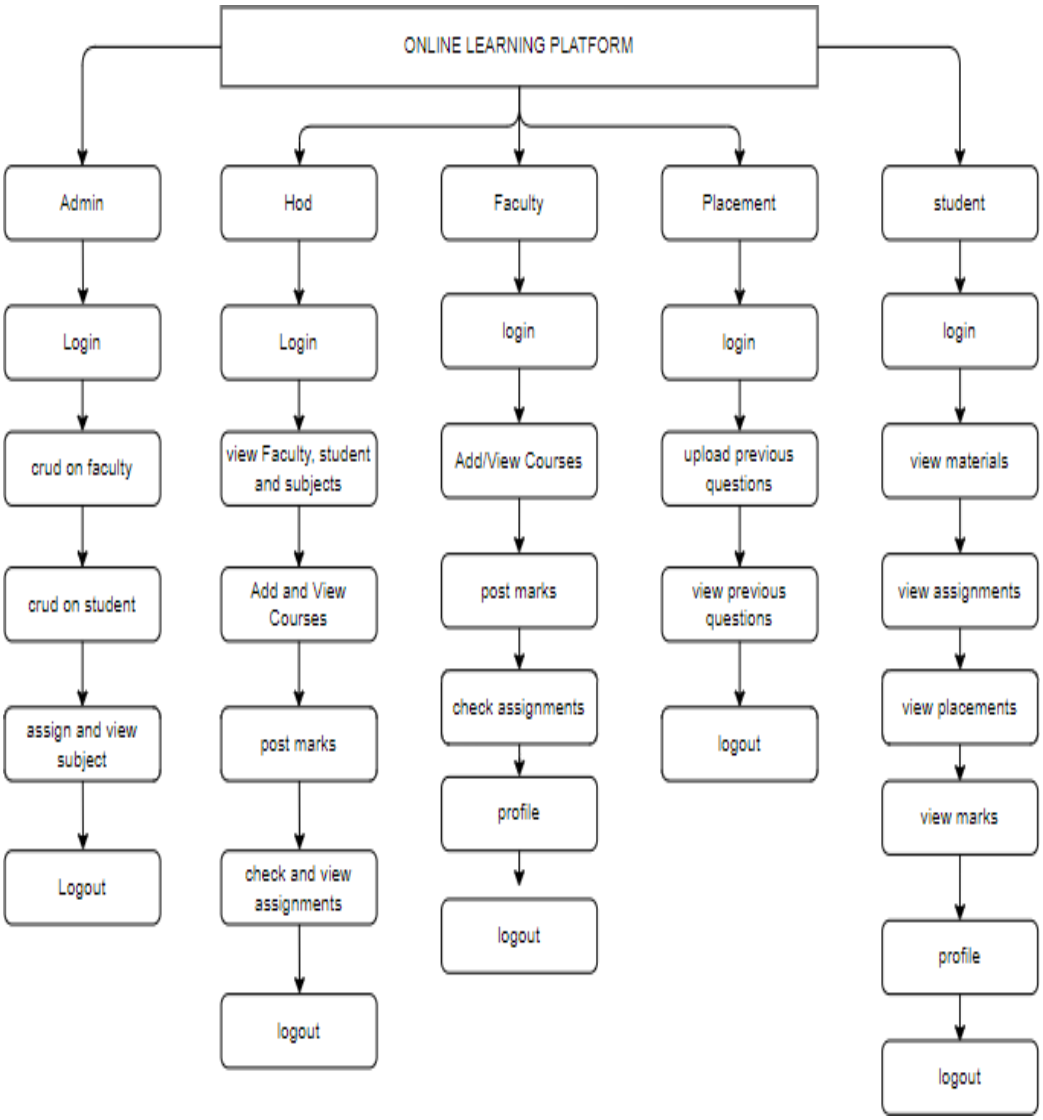
1.Fostered Mentorship Relationships:

By incentivizing mentorship through token-based feedback mechanisms, the platform encourages seniors to actively engage with juniors, fostering meaningful mentorship relationships. This not only benefits juniors in their academic pursuits but also allows seniors to leave a positive impact on their peers' educational journey.

3.Strengthened Academic Community:

Through the integration of communication channels and personalized progress tracking, the platform fosters a sense of camaraderie within the college community. By facilitating collaboration and support between juniors and seniors, the platform contributes to a strengthened academic environment where students can thrive and learn from each other's experiences.

Work Flow of Proposed System:



3.SETUP INSTRUCTIONS:

3.1 Pre-Requisites:

Here are the key prerequisites for developing a full-stack application using Node.js, Express.js, MongoDB, and React.js:

Vite:

Vite is a new frontend build tool that aims to improve the developer experience for development with the local machine, and for the build of optimized assets for production (go live). Vite (or ViteJS) includes a development server with ES _native_ support and Hot Module Replacement; a build command based on rollup.

npm create vite@latest

Node.js and npm:

Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the server side. It provides a scalable and efficient platform for building network applications.

Install Node.js and npm on your development machine, as they are required to run JavaScript on the server side.

Download: https://nodejs.org/en/download/

Installation instructions: https://nodejs.org/en/download/package-manager/

npm init

Express.js:

Express.js is a fast and minimalist web application framework for Node.js. It simplifies the process of creating robust APIs and web applications, offering features like routing, middleware support, and modular architecture.

Install Express.js, a web application framework for Node.js, which handles server-side routing, middleware, and API development.

Installation: Open your command prompt or terminal and run the following command:

npm install express

MongoDB:

MongoDB is a flexible and scalable NoSQL database that stores data in a JSON-like format. It provides high performance, horizontal scalability, and seamless integration with Node.js, making it ideal for handling large amounts of structured and unstructured data.

Set up a MongoDB database to store your application's data.

Download: https://www.mongodb.com/try/download/community

Installation instructions: https://docs.mongodb.com/manual/installation/

React.js:

React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.

Install React.js, a JavaScript library for building user interfaces.

Follow the installation guide: https://reactjs.org/docs/create-a-new-react-app.html

HTML, CSS, and JavaScript: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.

Database Connectivity: Use a MongoDB driver or an Object-Document Mapping (ODM) library like Mongoose to connect your Node.js server with the MongoDB database and perform CRUD (Create, Read, Update, Delete) operations. To Connect the Database with Node JS go through the below provided link:

https://www.section.io/engineering-education/nodejs- mongoosejs-mongodb/

3.2 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

Functional Requirements:

1. User Registration and Profile Creation: Allow seniors and juniors to create profiles specifying their academic expertise and needs respectively.

2. Real-Time Communication: Provide features such as real-time chat, discussion forums, and video conferencing options for seamless interaction between seniors and juniors.

3. Progress Tracking: Enable tracking of mentorship progress and outcomes to assess the effectiveness of the mentorship relationships.

4. Feedback Mechanism: Implement a token-based feedback system where juniors can provide feedback to seniors based on the assistance received, with weightage given to the quality of help provided.

Non-Functional Properties:

1. User-Friendly Interface: Ensure the platform is easy to navigate and intuitive for both seniors and juniors to use.

2. Security and Data Privacy: Implement robust security measures to protect user data and ensure compliance with data privacy regulations.

3. Scalability: Design the platform to accommodate a growing number of users and interactions over time.

4. Reliability: Ensure high uptime and availability of the platform to facilitate continuous mentorship interactions.

5. Performance: Optimize platform performance to provide seamless communication and interaction experiences for users.

6. Ethical Considerations: Embed ethical guidelines within the platform to promote fair and respectful interactions between seniors and juniors.

7. Integration: Ensure seamless integration of various communication channels and feedback mechanisms within the platform for enhanced user experience.

SOFTWARE REQUIREMENTS:

- Operating System : Windows95/98/2000/XP
- Application Serve : Tomcat 7.0
- Front End : React JS
- Scripts : JavaScript.
- Backend Language : Express Js
- Database : Mongo DB
- IDE : Visual Studio Code

HARDWARE REQUIREMENTS:

- Processor : Intel i3
- RAM : 4GB
- Hard Disk : 500GB

3.3 SYSTEM DESIGN:

Introduction of Input Design:

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties −

- It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user's attention, consistency, and simplicity.
- All these objectives are obtained using the knowledge of basic design principles regarding −
    - What are the inputs needed for the system?
    - How end users respond to different elements of forms and screens.

Objectives for Input Design:

The objectives of input design are −

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.

- To use validation checks and develop effective input controls.

Output Design:

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

Objectives of Output Design:

The objectives of input design are:

- To develop output design that serves the intended purpose and eliminates the production of unwanted output.
- To develop the output design that meets the end user's requirements.
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- able on time for making good decisions.

3.4 UML Diagrams:

Use Case Diagram:

To make the output availA use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

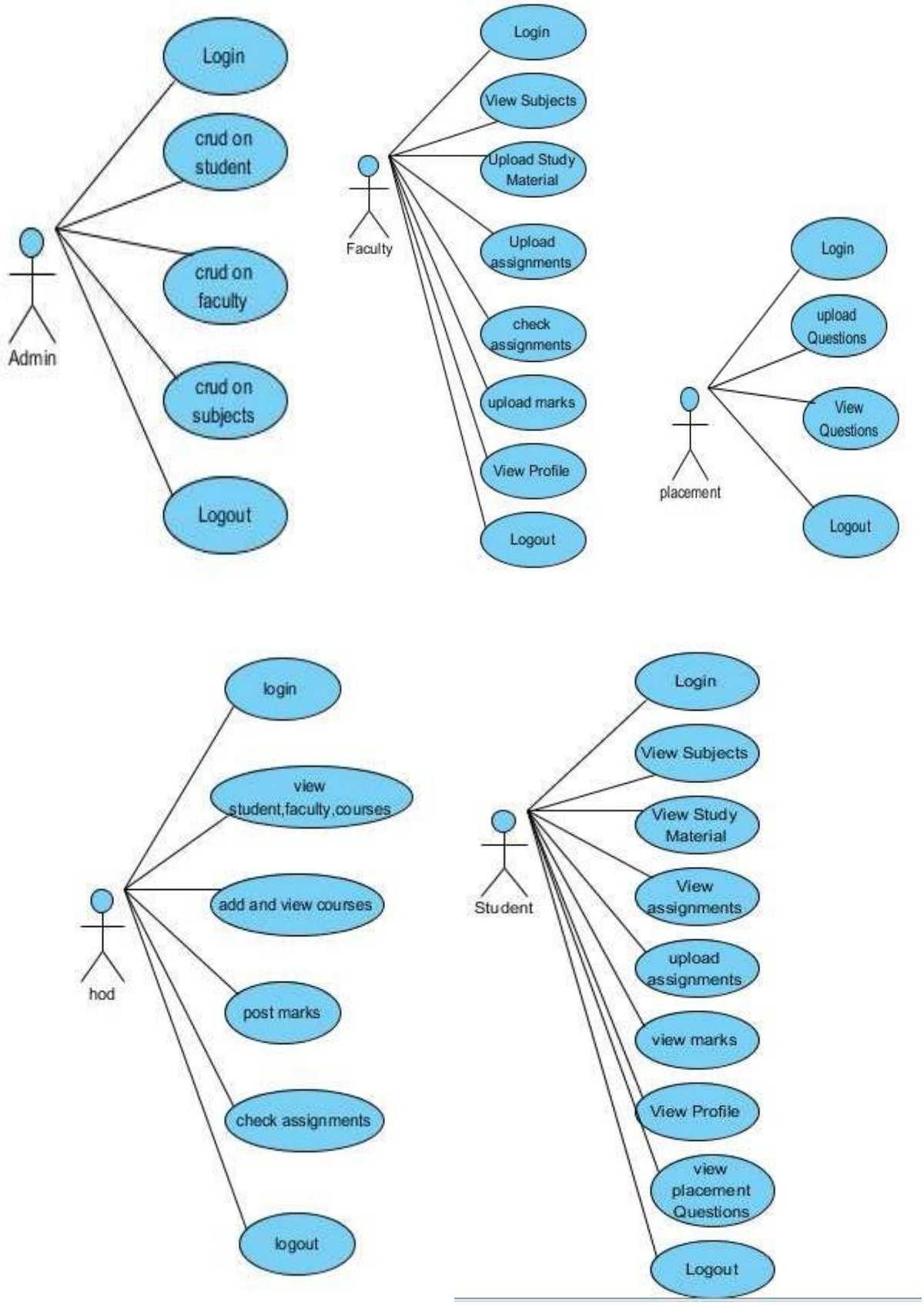


Fig:Use case diagram for Uml diagram

Class Diagram:

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

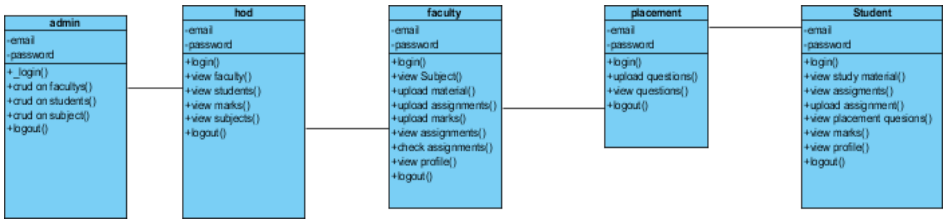


Fig:Class diagram for Uml diagram

Sequence Diagram:

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

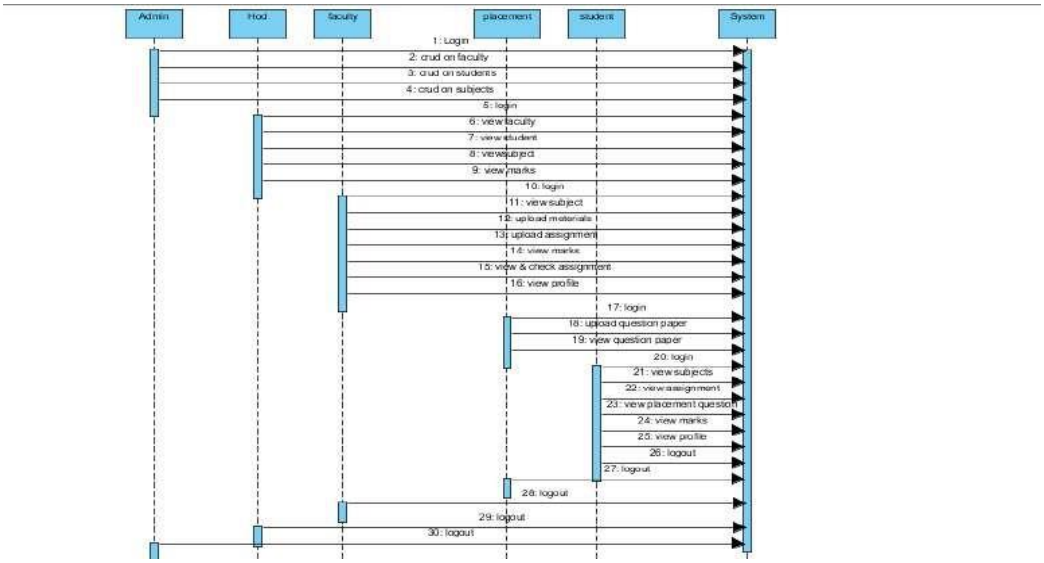


Fig: Sequence diagram for Uml diagram

Collaboration Diagram:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

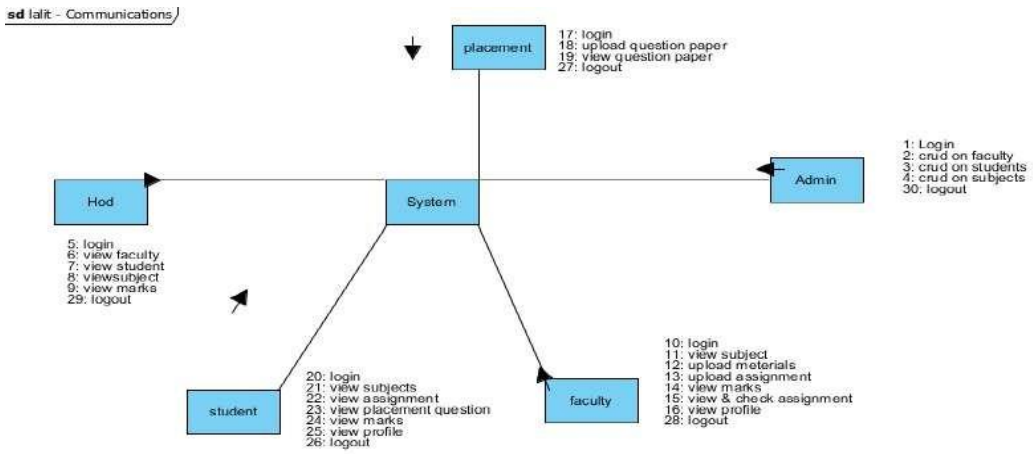


Fig: Collaboration diagram for Uml diagram

Deployment Diagram:

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.

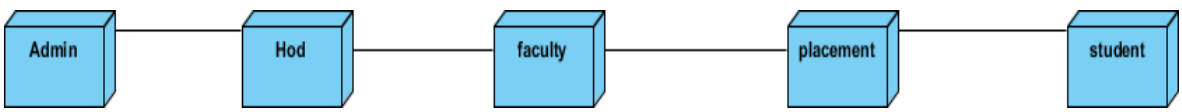
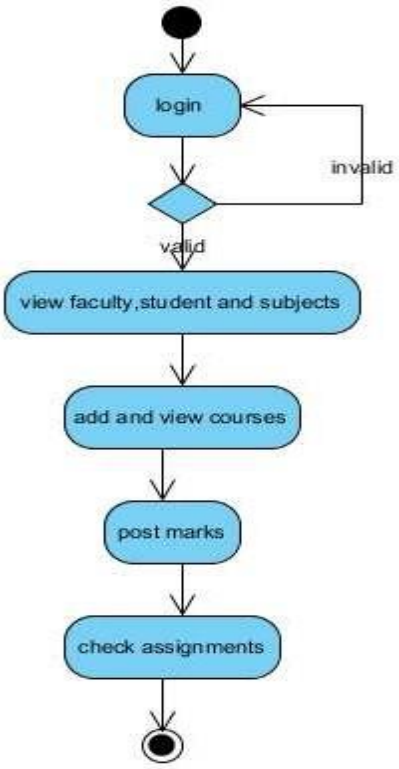
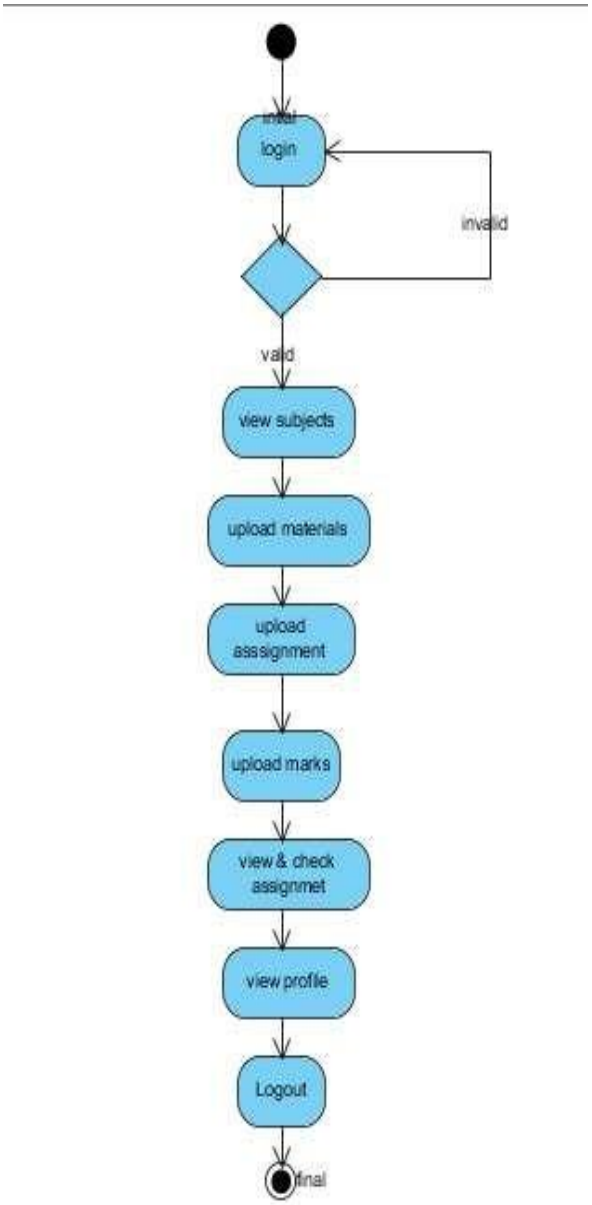


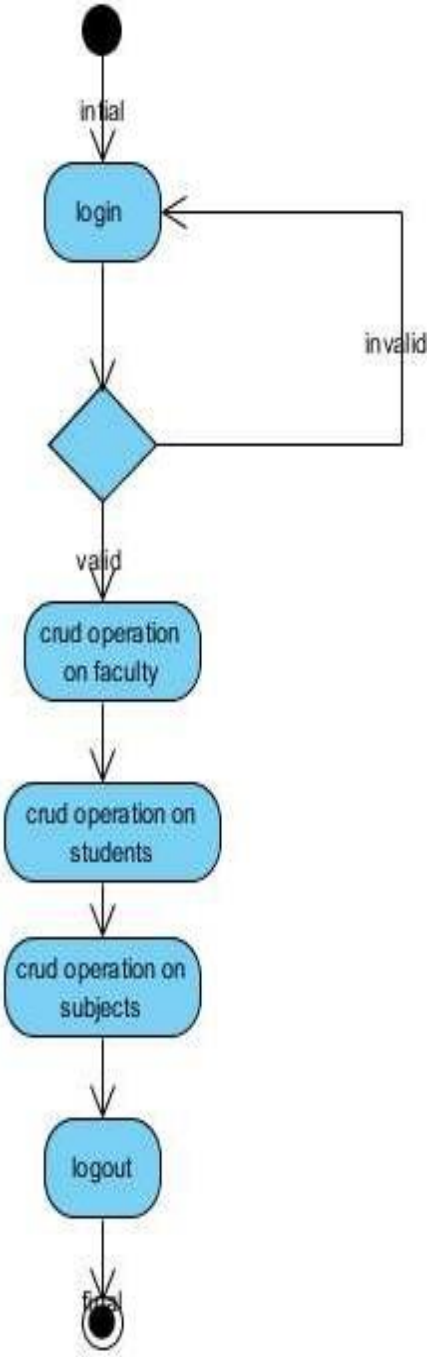
Fig : Deployment diagram for Uml diagram

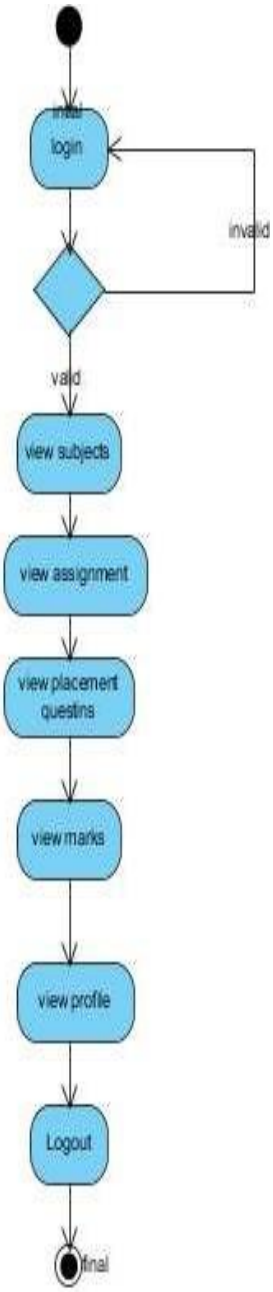
Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.






initial
login
invalid
valid
crud operation on faculty
crud operation on students
crud operation on subjects
logout
final


login
invalid
valid
view subjects
view assignment
view placement questins
view marks
view profile
Logout
final

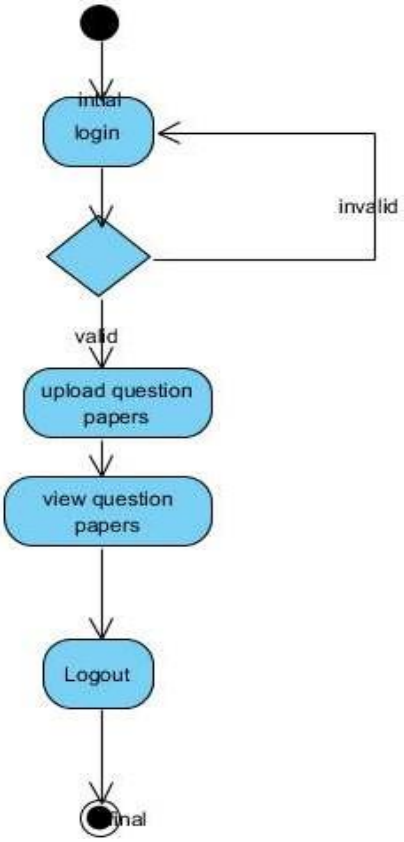


Fig : Activity diagram for Uml diagram

Component Diagram:

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

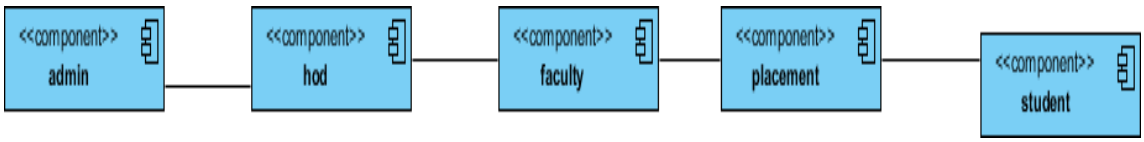


Fig : Component diagram for Uml diagram

ER Diagram:

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram to understand this concept.

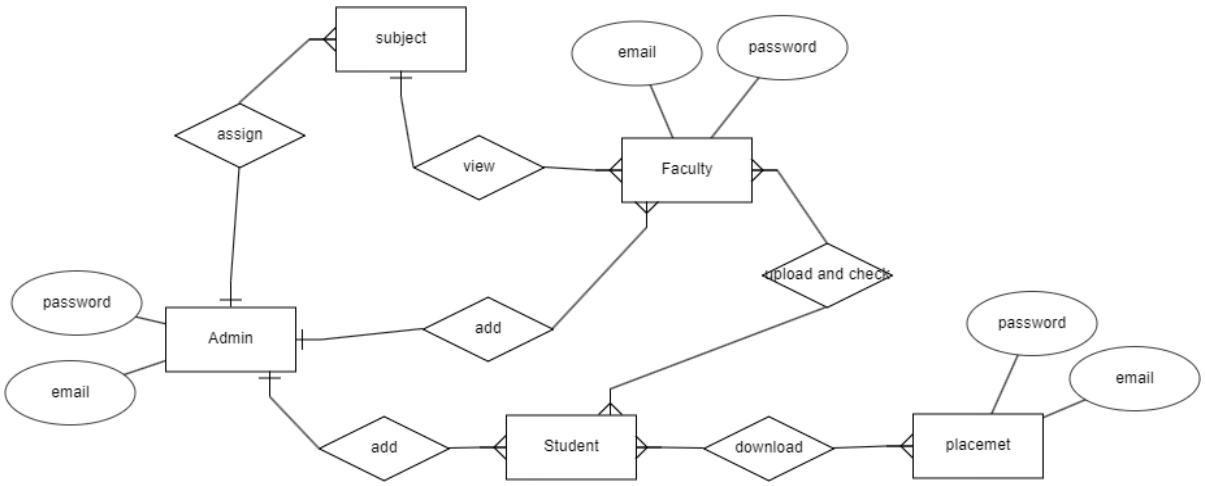
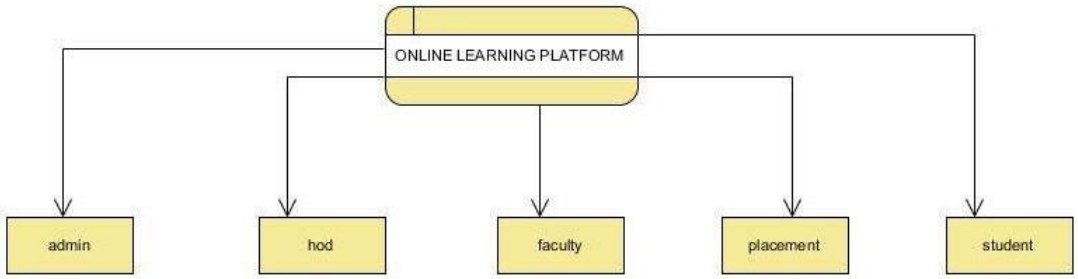


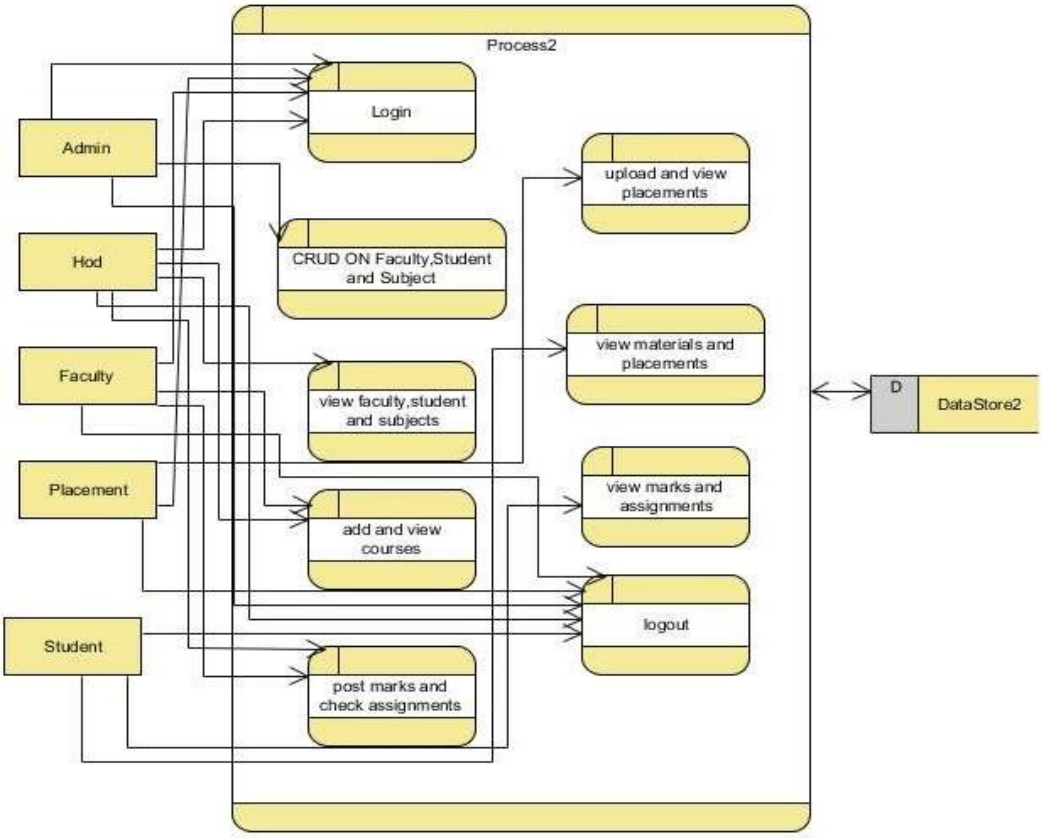
Fig: ER diagram

DFD Diagram:

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole.



Data Flow Diagram:



3.5 INSTALLATION:

Install Dependencies:

• Navigate into the cloned repository directory:

cd containment-zone

• Install the required dependencies by running the following commands:

cd frontend

npm install

cd ../backend

npm install

Start the Development Server:

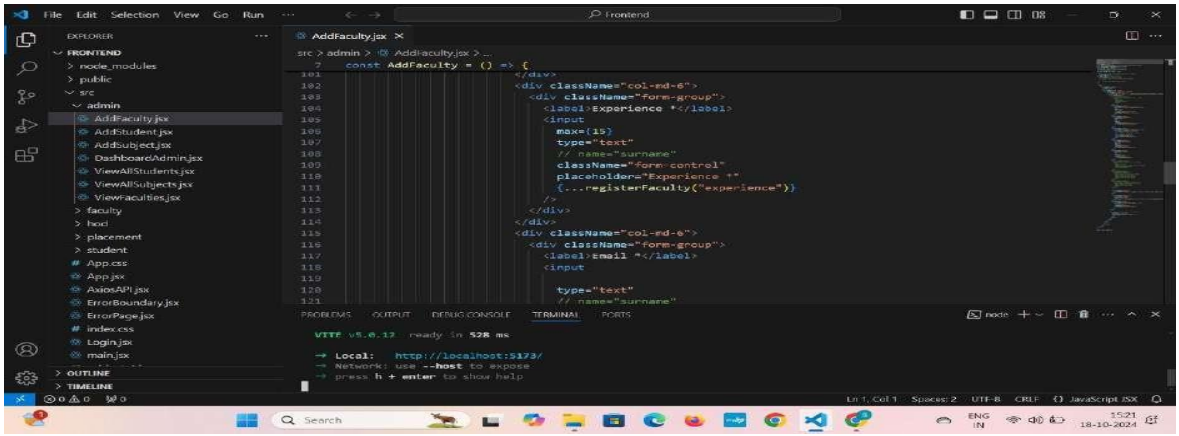
• To start the development server, execute the following command:

npm start

• The OLP app will be accessible at http://localhost:5172

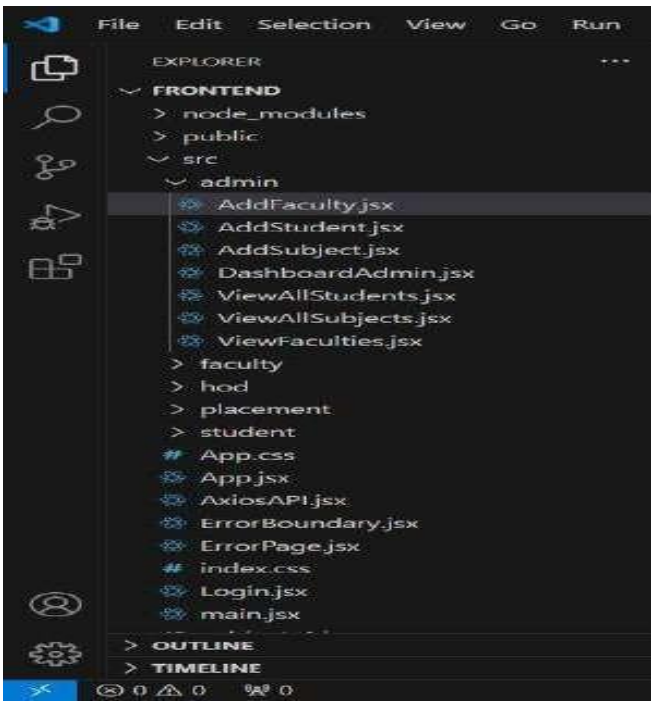
You have successfully installed and set up the Online learning app on your local machine. You can now proceed with further customization, development, and testing as needed.

4.FOLDER STRUCTURE:



4.1 CLIENT DIRECTORY:

The below directory structure represents the directories and files in the client folder (front end) where, react js is used along with Api's such as socket.io and agora.



4.2 SEVER DIRECTORY:

The below directory structure represents the directories and files in the server folder (back end) where, node js, express js and mongo db are used along with socket.io Api.

4.3 IMPLEMENTATION AND RESULTS

Process:

ADMIN:

Login: ADMIN will login with his default credentials.

Faculty: ADMIN will manage (add, view, update) faculty.

Student: ADMIN will manage (add, view and update) students

Subject: ADMIN will assign subjects to faculty,delete and view them.

Logout: ADMIN will logout from application.

HOD:

Login: HOD will login with his default credentials.

Faculty: HOD will view faculty.

Student: HOD will view students

Subject: HOD will view Subjects.

Courses: Hod will add the courses.

Assignments: Hod will check the assignments

Logout: HOD will logout from application.

Faculty:

Login: Faculty will login with his credentials.

Assigned Subject: Faculty will view assigned subjects. Then he upload videos, materials and assignments.

View Assignments: Faculty check the assignments get from students.

Profile: Faculty will update his profile.

Logout: Faculty will logout from application.

Placement:

Login: Placement will login with his credentials.

Add Placements: placement will add previous placements in his college.

Add Question papers: He will add previous question paper in application

Profile: Placement will update his profile.

Logout: Placement will logout from application.

Student:

Login: Student will login with his credentials.

View Subject: Student view his subjects (here he can view videos, download materials and download assignments).

Send Assignments: He will upload the assignments in certain amount of time.

View Placements: he will view the previous placements of his college

View Marks: he will view the marks.

Profile: student will update his profile.

Logout: student will logout from application

5.RUNNING THE APPLICATION:

Provide command to start the Frontend and Backend servers locally.

FRONTEND:

npm start in the client directory.

BACKEND:

npm start in the server directory

6. API DOCUMENTATION:

Backend Development

- **Setup express server**

1. Create index.js file in the server (backend folder).
2. define the port number, MongoDB connection string, and JWT key in the env file to access it.
3. Configure the server by adding cors, and body-parser.

- **Add authentication:** for this,

1. You need to make a middleware folder and in that make authMiddleware.js file for the authentication of the projects and can use in.

Database

- **Configure MongoDB**

1. Import mongoose.
2. Add database connection from config.js file present in the config folder
3. Create a model folder to store all the DB schemas.

Front-end Development

- **Installation of required tools:**
- For frontend, we use:

1. React
2. Bootstrap
3. Material UI
4. Axios
5. Antd
6. mdb-react-ui-kit
7. react-bootstrap

7.AUTHENTICATION:

1. Multi-Factor Authentication (MFA):

- **Two-Factor Authentication (2FA):** In addition to a password, require users to verify their identity using a second method like a one-time code sent via SMS, email, or an authentication app.
- **Biometric Authentication:** Support biometric methods such as fingerprint or facial recognition for users accessing the platform via mobile devices.

2. Social Login:

- **OAuth Integration:** Allow users to sign up or log in using their social accounts like Google, LinkedIn, Facebook, or GitHub. This provides convenience and reduces the need to remember another set of credentials.
- **SSO (Single Sign-On):** For corporate or institutional users, implement Single Sign-On, enabling them to access LeranHub using their existing credentials from platforms like Microsoft Azure AD, Okta, or Google Workspace.

3. Email/Password Authentication

- **Standard Login:** Offer a traditional email and password sign-up and login option, allowing users to create accounts directly on the platform.
- **Password Requirements:** Enforce strong password requirements (e.g., a combination of letters, numbers, and special characters) to ensure security.

4. Passwordless Authentication

- **Magic Links:** Allow users to log in without a password by sending a secure, time-sensitive login link to their email. This enhances convenience while maintaining security.
- **SMS Codes:** For mobile users, provide an option to log in by entering their phone number and receiving a one-time code via SMS.

5. Account Verification

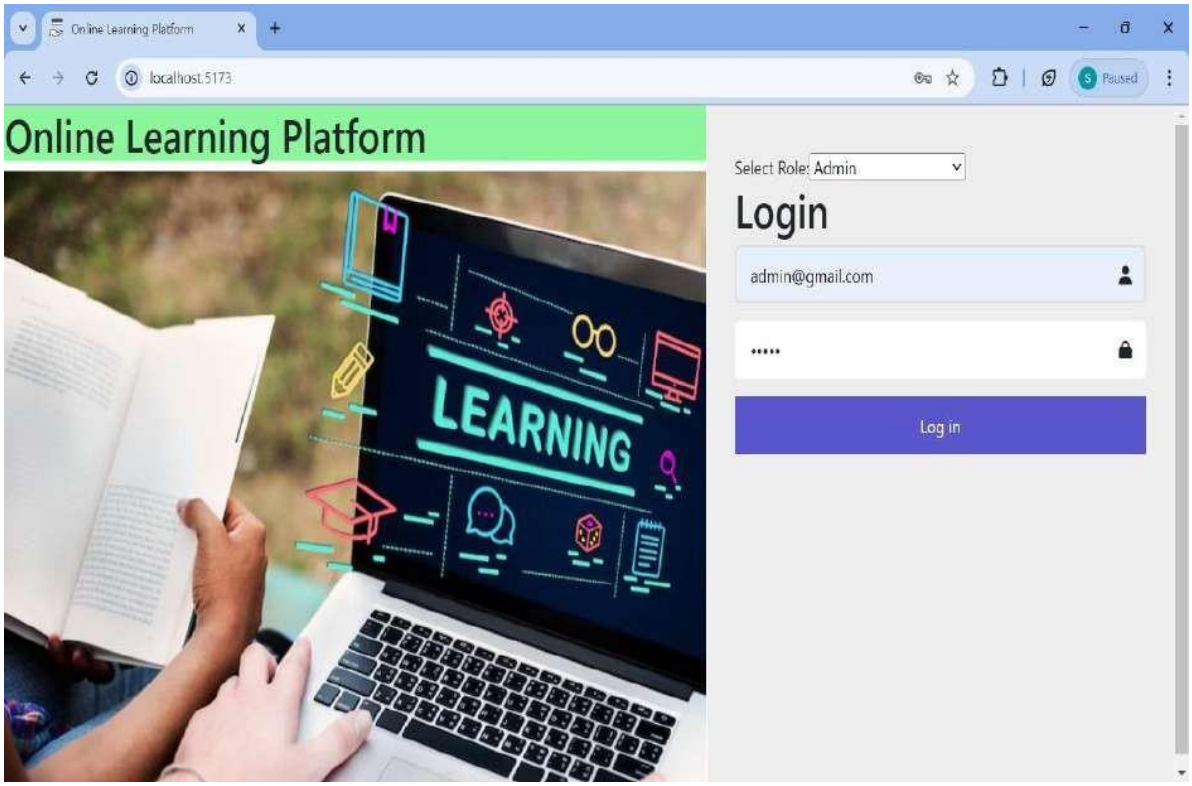
- **Email Confirmation:** Require users to verify their email address by clicking a link sent during the registration process, ensuring they have access to the email they signed up with.

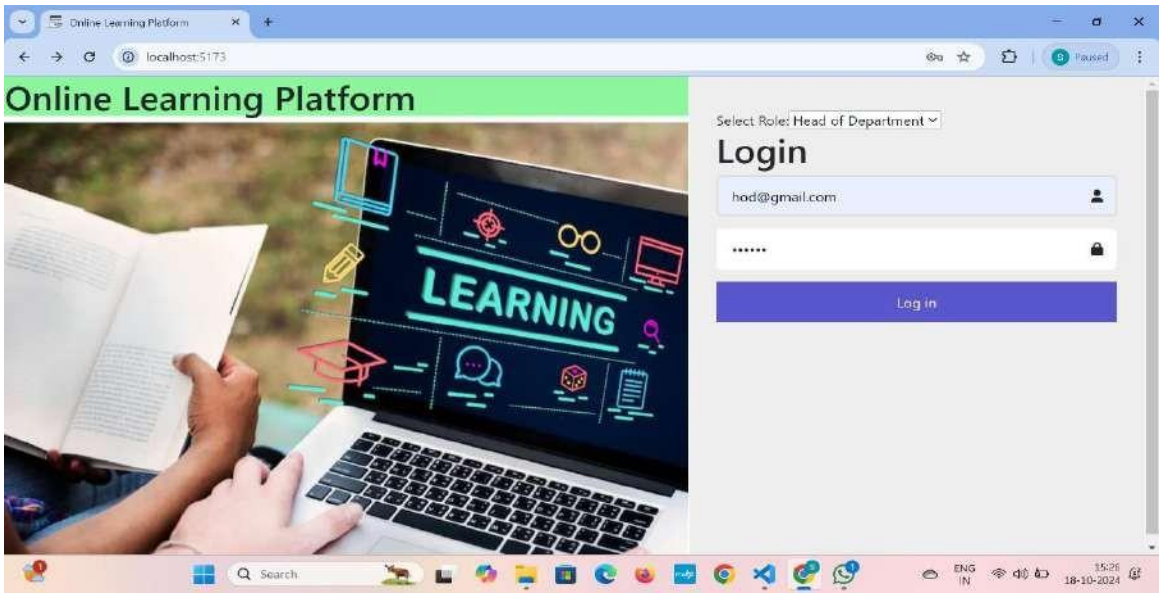
- **Phone Number Verification:** For enhanced security, especially for mobile users, include an optional or mandatory phone number verification step.

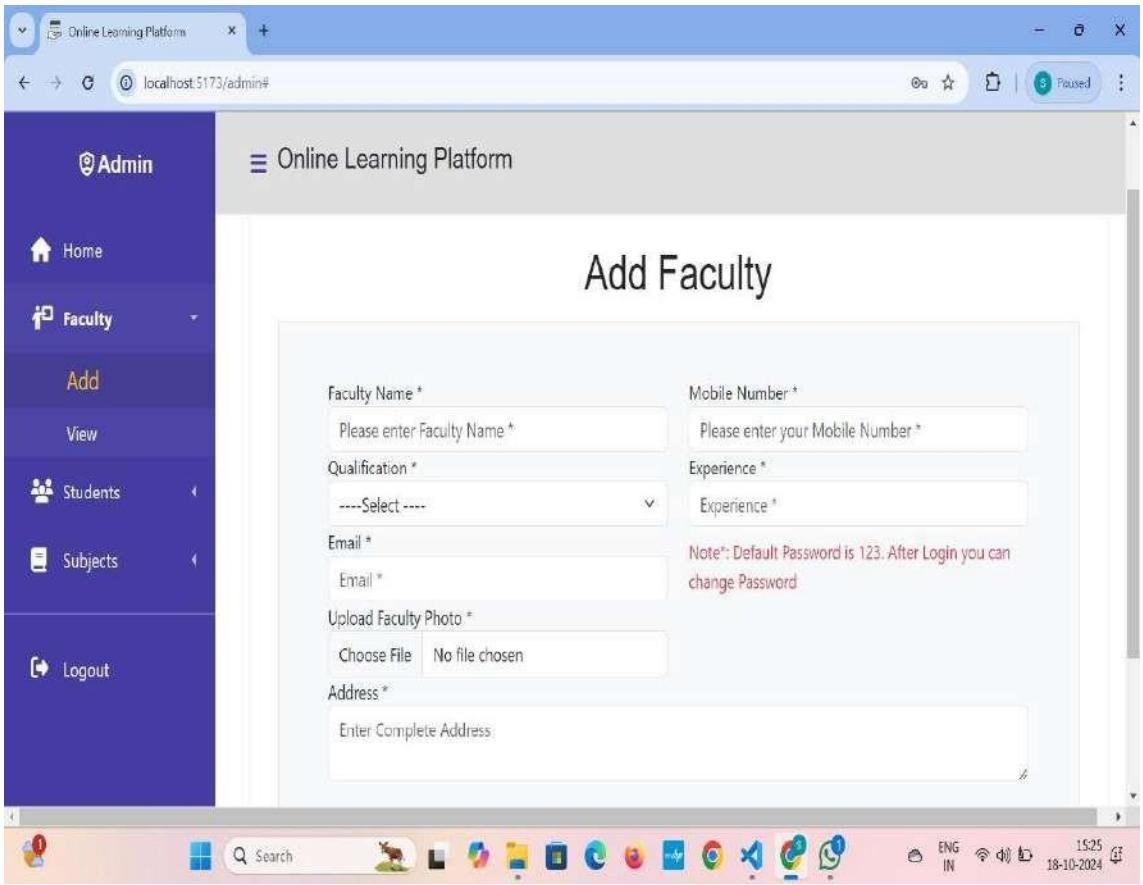
6. Account Recovery Options

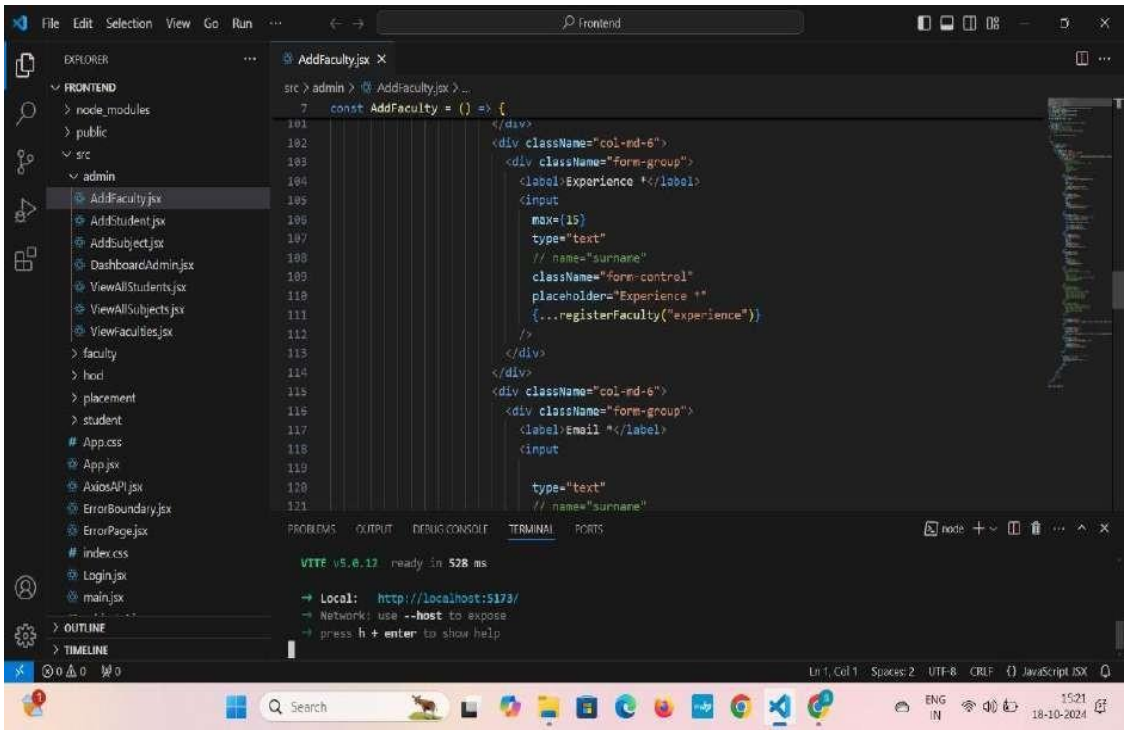
- **Forgot Password:** Provide a simple and secure password recovery process where users can request a password reset link sent to their email.
- **Backup Codes:** Offer backup codes that users can generate and store in case they lose access to their primary authentication method, especially when MFA is enabled.

8. USER INTERFACE



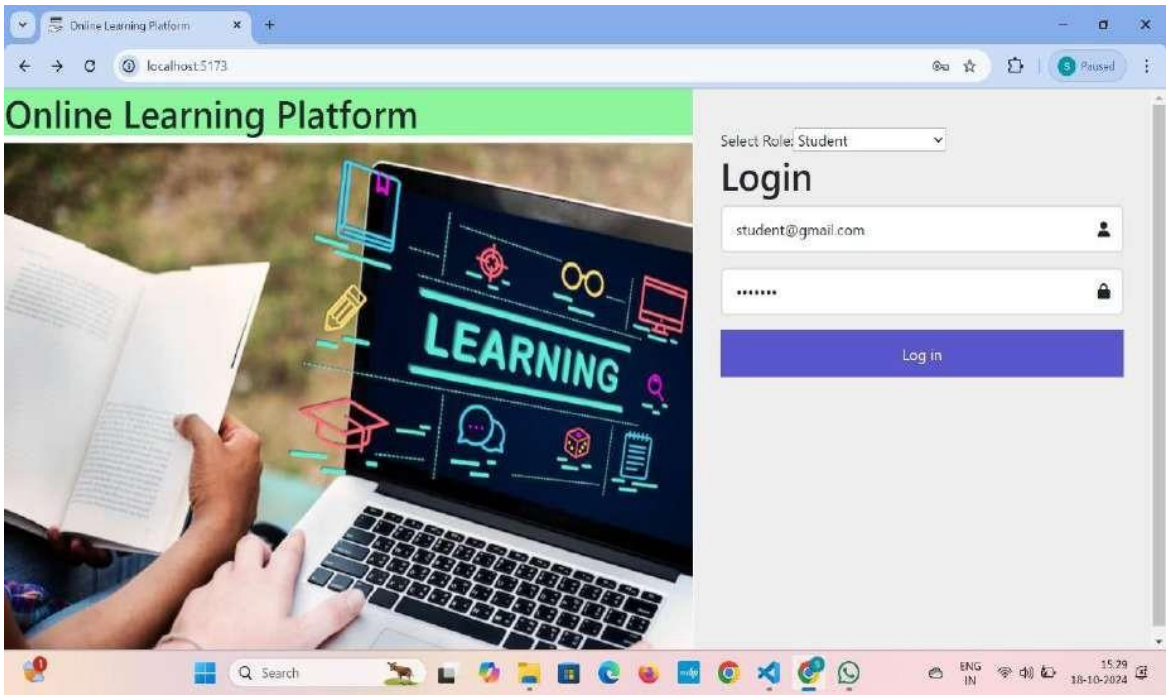

Online Learning Platform
localhost:5173
Online Learning Platform
LEARNING
Select Role: Head of Department
Login
hod@gmail.com
Log in

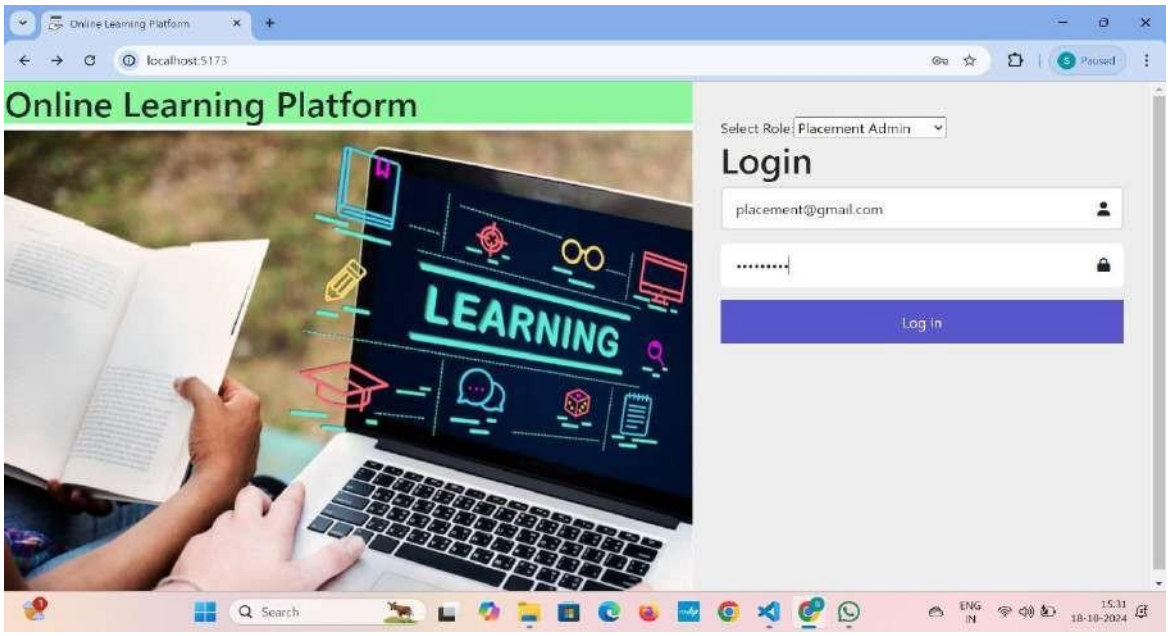

Online Learning Platform
localhost:5173/admin#
Admin
Online Learning Platform
Home
Faculty
Add
View
Students
Subjects
Logout
Add Faculty
Faculty Name *
Please enter Faculty Name *
Mobile Number *
Please enter your Mobile Number *
Qualification *
----Select ----
Experience *
Experience *
Email *
Email *
Note*: Default Password is 123. After Login you can change Password
Upload Faculty Photo *
Choose File
No file chosen
Address *
Enter Complete Address


File Edit Selection View Go Run
Frontend
EXPLORER
AddFaculty.jsx
FRONTEND
node_modules
public
src
admin
AddFaculty.jsx
AddStudent.jsx
AddSubject.jsx
DashboardAdmin.jsx
ViewAllStudents.jsx
ViewAllSubjects.jsx
ViewFaculties.jsx
faculty
hod
placement
student
App.css
App.jsx
AxiosAPI.jsx
ErrorBoundary.jsx
ErrorPage.jsx
index.css
Login.jsx
main.jsx
OUTLINE
TIMELINE
src > admin > AddFaculty.jsx > ...
const AddFaculty = () => {

<label>Experience *</label>
<input
max={15}
type="text"
// name="surname"
className="form-control"
placeholder="Experience *"
{...registerFaculty("experience")}
/>

<label>Email *</label>
<input
type="text"
// name="surname"
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
node
VITE v5.0.12 ready in 528 ms
Local: http://localhost:5173/
Network: use --host to expose
press h + enter to show help
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF {} JavaScript JSX
Search
ENG
IN
15:21
18-10-2024


Online Learning Platform
localhost:5173
Paused
Online Learning Platform
Select Role: Student
Login
student@gmail.com
Log in
LEARNING
Search
ENG
IN
15:29
18-10-2024



9. SYSTEM STUDY AND TESTING

7.1 Feasibility Study:

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- Economic feasibility
- Technical feasibility
- Social feasibility

Economic Feasibility:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed

system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical Feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social Feasibility:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

System Testing:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Types of Tests:

Unit testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive.

Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Functional testing:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process

flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

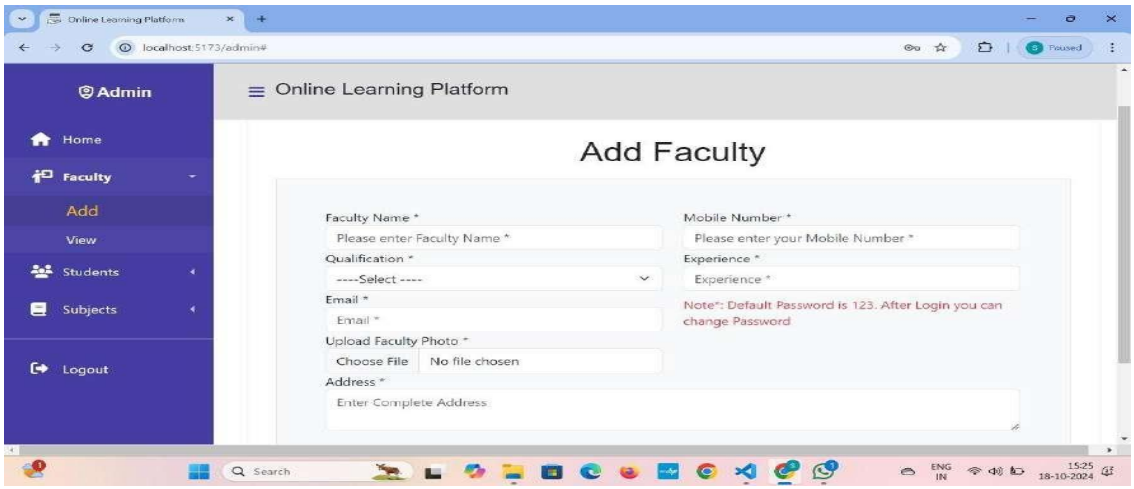
Test objectives:

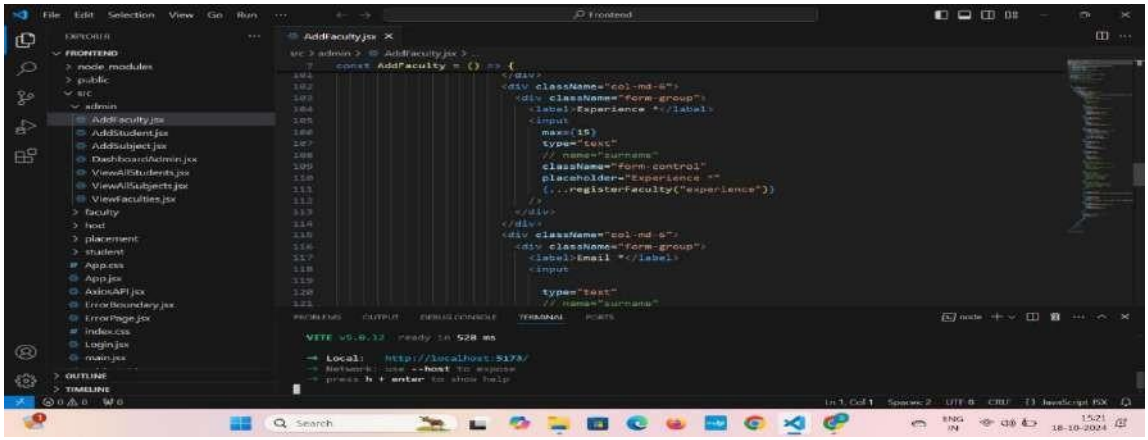
- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

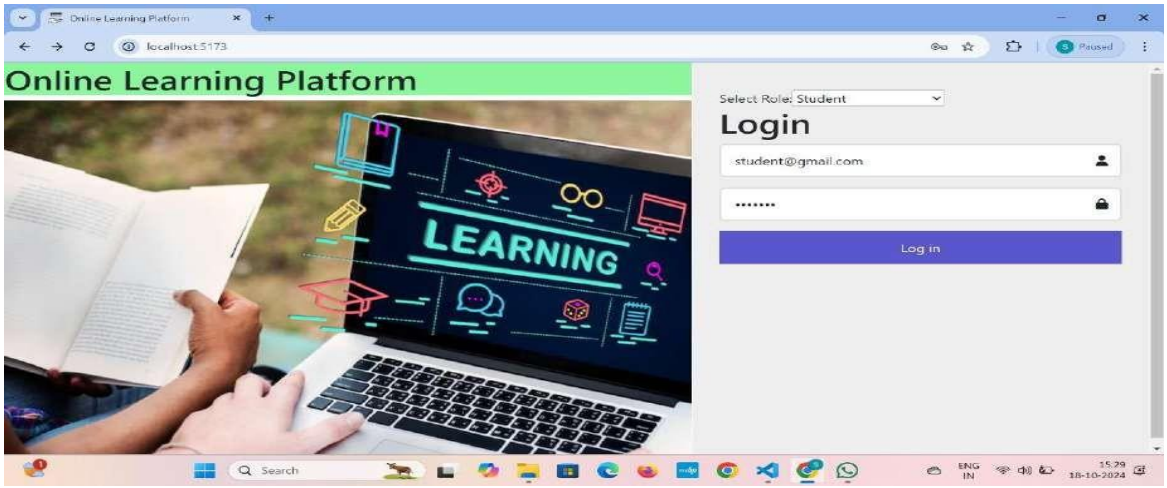
Features to be tested:

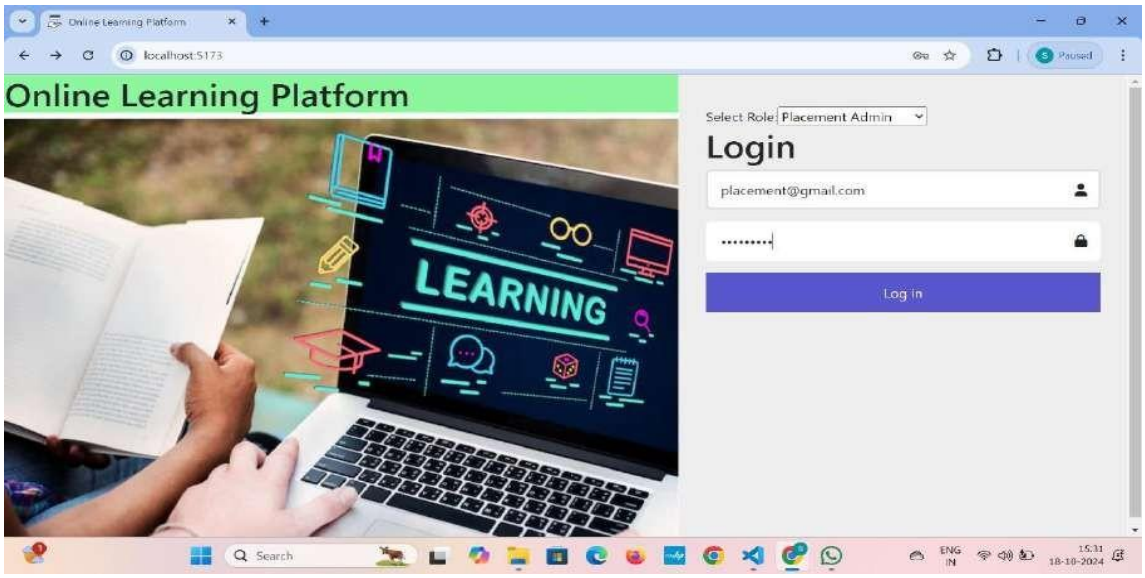
- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page**.**

10. SCREENSHOTS AND DEMO


Online Learning Platform
localhost:5173/admin#
Admin
Online Learning Platform
Home
Faculty
Add
View
Students
Subjects
Logout
Add Faculty
Faculty Name *
Please enter Faculty Name *
Mobile Number *
Please enter your Mobile Number *
Qualification *
----Select ----
Experience *
Experience *
Email *
Email *
Note*: Default Password is 123. After Login you can change Password
Upload Faculty Photo *
Choose File
No file chosen
Address *
Enter Complete Address


AddFaculty.jsx
FRONTEND
node_modules
public
src
admin
AddFaculty.jsx
AddStudent.jsx
AddSubject.jsx
DashboardAdmin.jsx
ViewAllStudents.jsx
ViewAllSubjects.jsx
ViewFaculties.jsx
faculty
host
placement
student
App.css
App.jsx
AxiosAPI.jsx
ErrorBoundary.jsx
ErrorPage.jsx
index.css
Login.jsx
main.jsx
const AddFaculty = () =>
<label>Experience *</label>
<label>Email *</label>
VITE v5.0.12 ready in 528 ms
Local: http://localhost:5173/


Online Learning Platform
localhost:5173
Online Learning Platform
LEARNING
Select Role: Student
Login
student@gmail.com
Log in



11. KNOWN ISSUES

1. Technical Issues:

Internet Connectivity: Learners and instructors may experience problems due to unreliable or slow internet connections, especially in remote or under served areas, leading to interruptions in learning.

2. Assessment Integrity:

- **Cheating and Plagiarism**: Ensuring academic integrity is a major concern for online platforms. With unsupervised assessments, students may resort to dishonest practices such as cheating or using unauthorized resources.
- **Proctoring Issues**: While online proctoring tools exist, they come with limitations like false positives, privacy concerns, and potential technological errors that may not fully prevent cheating.
- **Verification of Learner Identity**: It can be difficult to confirm whether the student taking the exam or completing the course work is the one registered for the course, leading to concerns about the validity of the learning outcomes.

3. Time-Zone Differences:

- **Scheduling Conflicts**: Online learning platforms with global users may face challenges in scheduling live sessions, discussions, or exams due to time zone differences, making synchronous learning difficult for some students.

12. FUTURE ENHANCEMENT:

Future enhancements could include the integration of AI-driven recommendation systems to match juniors with the most suitable senior mentors based on their academic needs and preferences. Gamification elements can be introduced to incentivize active participation and engagement, such as leaderboard rankings and achievement badges. Furthermore, expanding the platform to offer career guidance and networking opportunities beyond academic support could enhance its utility and long-term impact on students' personal and professional development.

13. CONCLUSION:

In conclusion, our innovative online platform aims to foster a culture of collaboration and support within our college community by facilitating academic interaction between seniors and juniors. By leveraging the expertise of seniors, we seek to empower juniors with guidance and assistance, ultimately strengthening the academic environment for all. Through personalized feedback mechanisms and ethical considerations, we aspire to create a sustainable mentorship ecosystem benefiting students at every level of their educational journey.