CERTIFICATE

ProjectTitle:"**LearnHub:YourCenterForSkillEnhancement"is**abonafidework carried out by the following students:

- **TEAMID:** LTVIP2025TMID56186

1. **Havilasa Cheeli(223C1A0508)**
2. **Mohammad Firdose (Team Leader)**
3. **Karani Yamineeswari (TeamMember)**
4. Ejju Vijaya Durga (TeamMember))

Date Of Submission:30-06-2025

TABLEOFCONTENTS

1. INTRODUCTION

OnlineLearningPlatformusingMERN

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, especiallyinrecentyears,astheyofferflexibilityandaccessibilityforlearnersofallagesandbackgrounds. 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 interfacethatmakesiteasyforlearners,regardlessoftheirtechnicalproficiency,tonavigateand access the content.

→**CourseManagement:** Instructorsorcoursecreatorscanupload,organize,andmanagecourse materials. Learners can enroll in courses and track their progress.

→**Interactivity:**Manyplatformsincludeinteractiveelementslikediscussionforums,chatrooms,and live webinars, which foster communication and collaboration among learners and instructors.

→**Certification:**Learnerscanearncertificatesorbadgesuponcompletingcoursesormeetingcertain criteria, which can be valuable for employment or further education.

→**Accessibility:**Contentisoftenaccessibleonvariousdevices,includingcomputers,tablets,and smartphones, making learning possible from anywhere with an internet connection.

→**Self-PacedLearning:** Learnerscantypicallyaccesscoursematerialsattheirownpace.Thisflexibility allows for learning that fits into individual schedules and preferences.

→**PaymentandSubscriptionOptions**:Theremaybefreecourses,butsomecontentmayrequire payment or a subscription. Platforms often offer multiple pricing models.

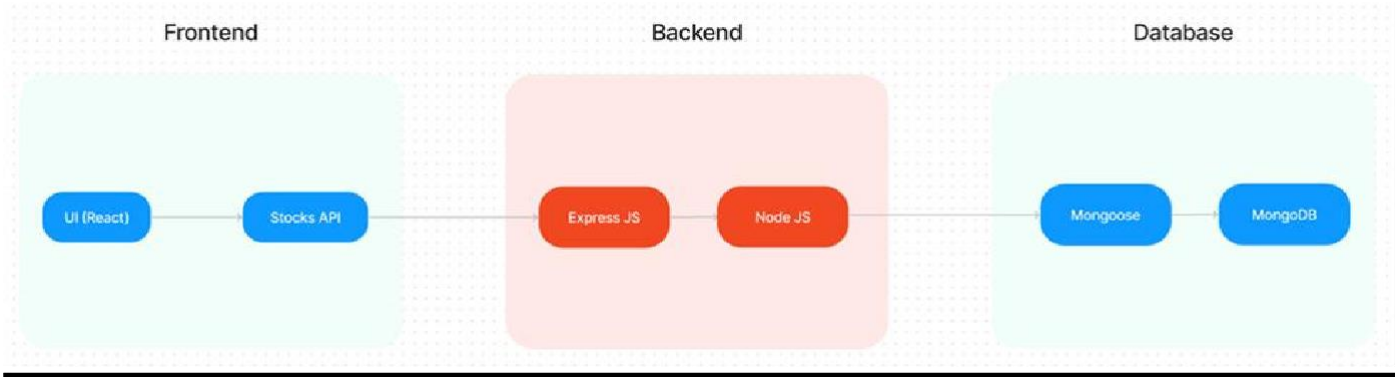
2. PROJECTOVERVIEW

→Scenario-basedCaseStudy:

- ✓ Scenario:LearningaNew Skill
- ✓ **User Registration:** Sarah, a student interested in learning web development, visits the OnlineLearningPlatformandcreatesanaccount.Sheprovidesheremailandchoosesa password.
- ✓ **BrowsingCourses:**Uponloggingin,Sarahisgreetedwithauser-friendlyinterface displaying various courses categorized by topic, difficulty level, and popularity.
- ✓ Shenavigatesthroughthecoursecatalog,filteringcoursesbynameandcategoryuntil she finds a "Web Development Fundamentals" course that interests her.
- ✓ **EnrollinginaCourse:**Sarahclicksonthecourseandreadsthecoursedescription, instructor details, and syllabus. Impressed, she decided to enroll in the course.
- ✓ Afterenrolling,Sarahcanaccessthecoursematerials,includingvideolectures,reading materials, and assignments.
- ✓ **LearningProgress:**Sarahstartsthecourseandproceedsthroughthemodulesatherown 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 elementssuchasdiscussionforumsandlivewebinarswhereshecanaskquestionsand interact with the instructor and other learners.
- ✓ **CourseCompletionandCertification:**Aftercompletingallthemodulesandassignments, Sarah takes the final exam. Upon passing,shereceives adigitalcertificateof completion, which she can download and add to her portfolio.
- ✓ **Paid Courses:** Sarah discovers an advanced web development course that requires payment.Shepurchasesthecourseusingtheplatform'spaymentsystemandgains access to premium content.
- ✓ **Teacher's Role:** Meanwhile, John, an experienced web developer, serves as a teacher on theplatform.Hecreatesanduploadsnewcoursesonadvancedwebdevelopmenttopics, adds sections to existing courses, and monitors course enrollments.
- ✓ **Admin Oversight:** The admin oversees the entire platform, monitoring user activity, managingcourselistings,andensuringsmoothoperation.Theykeeptrackofenrolled students, handle any issues that arise, and maintain the integrity of the platform.

3. ARCHITECTURE

TECHNICALARCHITECTURE:



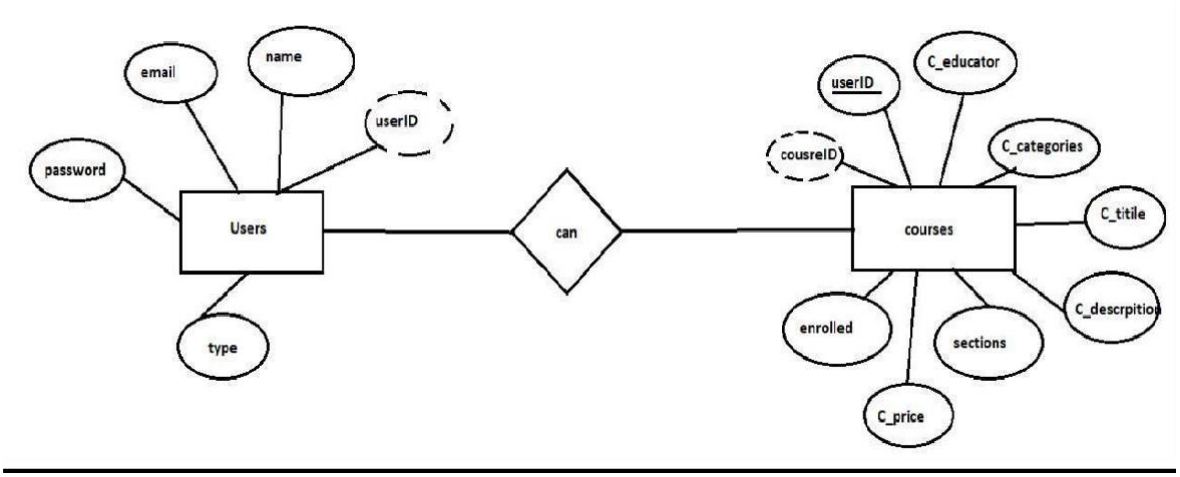
Thetechnical architecture ofOLP appfollowsaclient-servermodel,wherethe frontend servesasthe client andthebackendactsastheserver.Thefrontendencompassesnotonlytheuserinterfaceandpresentation but also incorporates the axios library to connect with backend easily by using RESTful Apis.

ThefrontendutilizesthebootstrapandmaterialUIlibrarytoestablish areal-timeandbetterUIexperience for any user.

On thebackendside,weemployExpress.jsframeworkstohandletheserver-sidelogicandcommunication. 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 comprehensivetechnicalarchitectureforourOLP app.Thisarchitectureenablesreal-timecommunication, efficient data exchange, and seamless integration, ensuring a smooth and immersive blogging experience for all users.

ER DIAGRAM:



Herethereare2collectionsnamelyusers,courses thathavetheirownfieldsin

Users:

1. _id:(MongoDBcreatesbyuniquedefault)
2. name
3. email
4. password
5. type

Courses:

1. userID:(canactasaforeignkey)
2. _id:(MongoDBcreatesbyuniquedefault)
3. C_educator
4. C_categories
5. C_title
6. C_description
7. sections
8. C_price
9. enrolled

4. SET INSTRUCTIONS

PRE-REQUISITES/INSTALLATION:

Here arethe keyprerequisitesfordevelopingafull-stackapplicationusingNode.js,Express.js,MongoDB, and React.js:

Vite:

Viteisanewfrontendbuildtoolthataimstoimprovethedeveloperexperiencefordevelopmentwiththe 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.

```
npmcreatevite@latest
```

Node.jsand npm:

Node.jsisapowerfulJavaScriptruntimeenvironmentthatallowsyoutorunJavaScriptcodeonthe server side. It provides a scalable and efficient platform for building network applications.
InstallNode.jsandnpmon yourdevelopmentmachine,astheyarerequiredtorunJavaScriptonthe server side.
Download:https://nodejs.org/en/download/
Installationinstructions:https://nodejs.org/en/download/package-manager/

```
npminit
```

Express.js:

Express.js is a fast and minimalist web application framework for Node.js. It simplifies the process of creatingrobustAPIsandwebapplications,offeringfeatureslikerouting,middlewaresupport,andmodular architecture.
InstallExpress.js,awebapplicationframeworkforNode.js,whichhandlesserver-siderouting,middleware, and API development.
Installation:Openyourcommandpromptorterminalandrunthefollowingcommand:

```
npminstallexpress
```

MongoDB:

MongoDB is a flexible and scalable NoSQL database that stores data in a JSON-like format. It provides high performance,horizontalscalability,andseamlessintegrationwithNode.js,makingitidealforhandlinglarge 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
Installationinstructions:https://docs.mongodb.com/manual/installation/

React.js:

React.js is a popular JavaScript library for building user interfaces. It enables developers to createinteractiveandreusableUIcomponents,makingiteasiertobuilddynamicandresponsivewebapplications. Install React.js, a JavaScript library for building user interfaces. Followtheinstallationguide:https://reactjs.org/docs/create-a-new-react-app.html

HTML,CSS,andJavaScript:Basicknowledgeof HTMLforcreatingthestructureofyourapp,CSSforstyling, and JavaScript for client-side interactivity is essential.

Database Connectivity:Use a MongoDB driver or an Object-Document Mapping (ODM) library like MongoosetoconnectyourNode.jsserverwiththeMongoDBdatabaseandperformCRUD(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/

InstallDependencies:

- Navigateintotheclonedrepositorydirectory: cd containment-zone
- Installtherequireddependenciesbyrunningthefollowingcommands:
  cd frontend
  npm
  installcd../ba
  ckend npm
  install

StarttheDevelopmentServer:

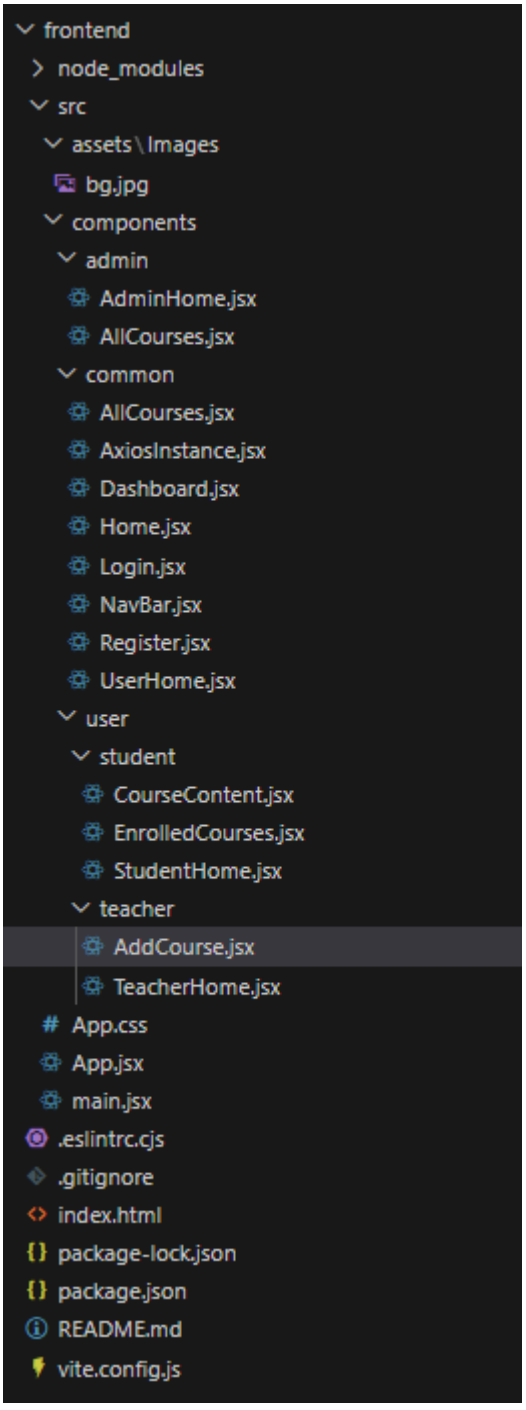
- Tostartthedevelopmentserver,executethefollowingcommand:
  npm start
- TheOLPappwillbeaccessibleathttp://localhost:5172

YouhavesuccessfullyinstalledandsetuptheOnline learningapponyourlocalmachine.Youcannow proceed with further customization, development, and testing as needed.

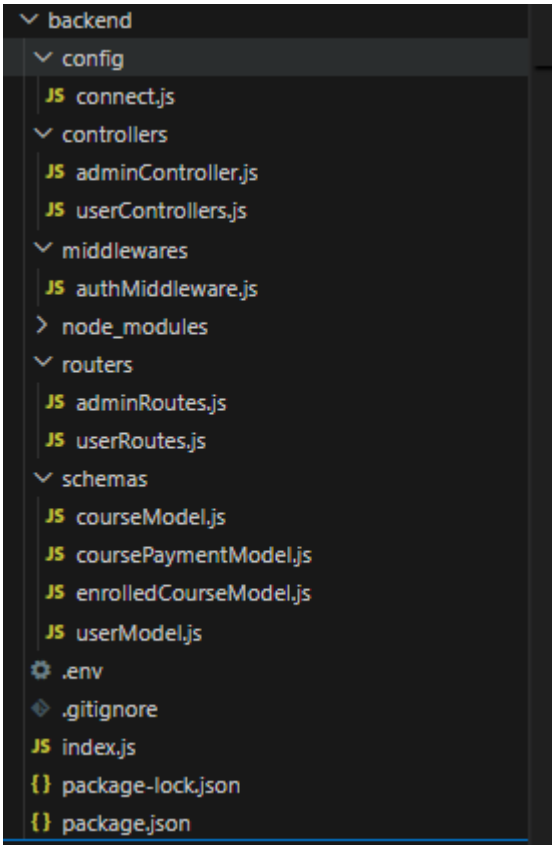
5.PROJECTSTRUCTURE

CreateaProjectfolderthatcontainsfilesasshownbelow:

1. Frontend:



2. Backend:



- Thefirstimageisoffrontendpartwhichisshowingallthefilesandfoldersthathavebeenusedin UI development

- ThesecondimageisofBackendpartwhichisshowingallthefilesandfoldersthathavebeen used in backend development

6. APIDOCUMENTATION

ApplicationFlow:

Theprojecthasausercalled–teacherandstudentand theotherwillbeAdminwhichtakescare of all the users. The roles and responsibilities of these users can be inferred from the API endpoints defined in the code. Here is a summary:

Teacher:

- Canaddcoursesforthestudent.
- Also,deletethecourseifnostudentenrolledinitorforanyother reasons.
- Also,addsectionstocourses.

Student:

- Canenrollinanindividualormultiple courses.
- Canstartthecoursewhereithas stopped.
- Oncethecourseiscompleted,theycandownloadtheircertificateofcompletionofthecourse.
- Forapaidcourse,theyneedtopurchaseitandthentheycanstartthecourse.
- Theycanfilteroutthecoursebysearchingbyname,category,etc

Admin:

- heycanalterallthecoursesthatarepresentintheapp.
- Watchoutforallkindsofusersinthe app.
- Recordalltheenrolledstudentsthatareenrolledinthecourse.

Setup&configuration:

→**Foldersetup:**

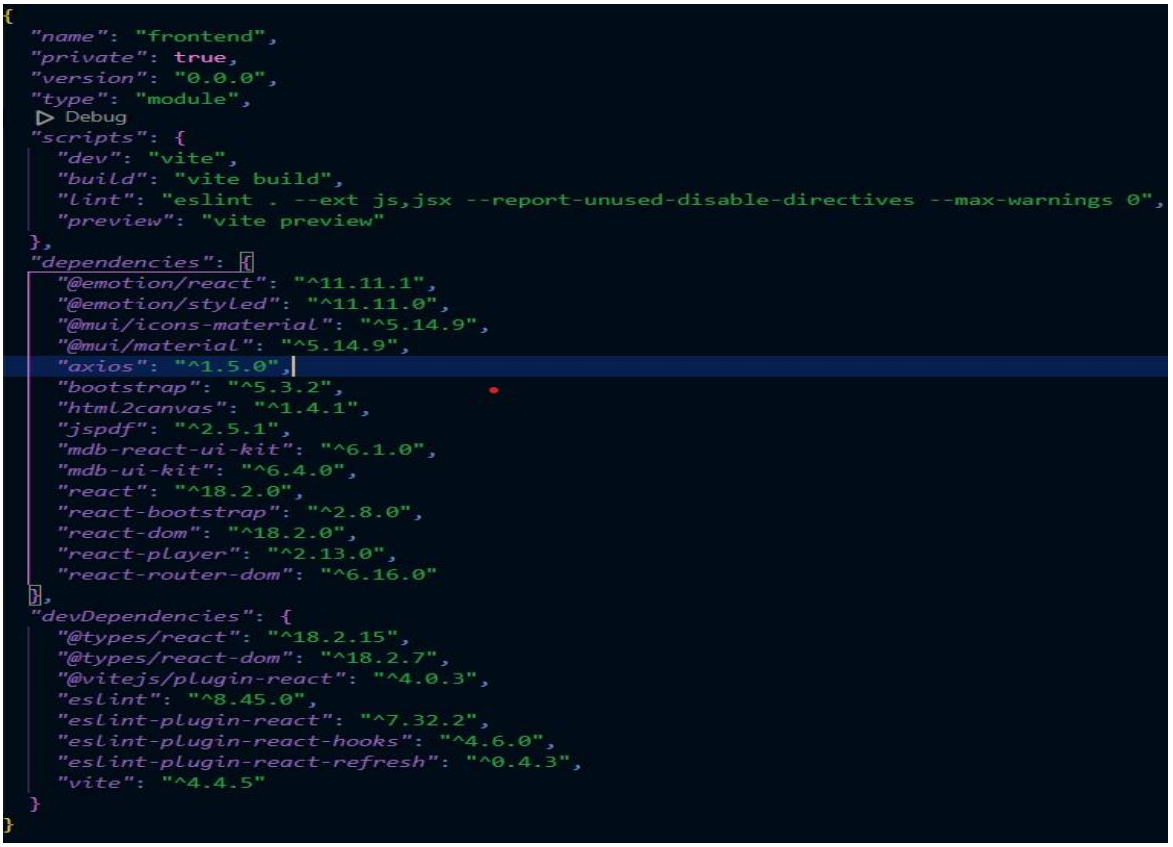
- ✓ Createfrontendand
- ✓ Backendfolders

Openthebackendfoldertoinstall thenecessarytools For backend, we use:

- ✓ cors
- ✓ bcryptjs
- ✓ express

- ✓ dotenv
- ✓ mongoose
- ✓ Multer
- ✓ Nodemon
- ✓ jsonwebtoken

→Aftertheinstallationofallthelibraries,thepackage.jsonfilesforthefrontendlooksliketheone mentioned below:



→Aftertheinstallationofallthelibraries,thepackage.jsonfilesforthebackendlooksliketheone mentioned below.

BackendDevelopment:

- **Setupexpressserver**
  - ✓ Createindex.jsfileintheserver(backendfolder).
  - ✓ Definetheportnumber,MongoDB connectionstring, andJWTkeyin theenvfileto access it.
  - ✓ Configuretheserverbyaddingcors,andbody-parser.
- **Addauthentication:**f
  - ✓ Youneedtomake amiddlewarefolderandinthatmakeauthMiddleware.jsfilefor the authentication of the projects and can use in.

DatabaseDevelopment:

- **ConfigureMongoDB**
  - ✓ Importmongoose.
  - ✓ Adddatabaseconnectionfromconfig.jsfilepresentintheconfigfolder

- ✓ Create amodelfoldertostoreall theDBschemas.

FrontendDevelopment:

→**Installationofrequiredtools:**

- Forfrontend,weuse:
  - ✓ React
  - ✓ Bootstrap
  - ✓ Material UI
  - ✓ Axios
  - ✓ Antd
  - ✓ mdb-react-ui-kit
  - ✓ react-bootstrap

7. AUTHENTICATION

➢ OVERVIEW

LearnHub uses **JWT (JSON Web Tokens)** for secure, stateless authentication and implements**role-basedaccesscontrol**toensureusersonlyaccessfeaturesrelevant to their roles.

➢ USER ROLES:

Student–Canregister,browsecourses,enroll,andinteractviacoursed discussion/chat.
Instructor–Cancreateandmanagecourses,uploadcontent,andrespondtostudent queries.
Admin–Hasfullprivilegestomanageusers,courses,andsystemsettings.

➢ REGISTRATION:

Userssignupviathe/registerendpoint.
Requiredfields:**name**,**email**,**password**,and**role**.
Passwordsare**hashedusingbcrypt**beforebeingstoredinthedatabasetoensure secure credential storage.

➢ LOGIN:

Usersloginusing/loginendpoint.
Ifcredentialsarevalid,a**JWTtoken**isgeneratedandreturned. This token is required for accessing protected routes.

➢ TOKEN-BASEDAUTHENTICATION:

JWTtokensarestoredclient-side(typicallyinlocalStorageorsessionStorage). For secure API access, the token is attached to the **Authorization header** as:

```
Authorization: Bearer <token>
```

➢ MIDDLEWARE PROTECTION:

Backendroutesaresecuredusing**authenticationmiddleware**.
Middleware verifies the token and extracts user details.
Ifinvalidormissing,a**401Unauthorized**responseis returned.

- **ROLE-BASEDACCESSCONTROL:**

  - ✓ Afterauthentication,user**rolesareverified**tocontrolaccess.
    - For example:
      - Only**Admins**canaccesstheAdminDashboard.
      - Only**Instructors**cancreate/editcourses.
  - ✓ Preventsunauthorizedaccesstoprivilegedendpointsandfeatures.

- **LOGOUT:**

  Logoutishandledontheclientsideby **removingtheJWTtoken**fromstorage. Since sessions are stateless, no server-side cleanup is required.

- **PASSWORDSECURITY:**

→LearnHub ensuresstrongpasswordpractices:

  - ✓ Passwordsare**hashedusingbcrypt**.
  - ✓ **Salting**isusedtopreventrainbowtableandbrute-forceattacks.
  - ✓ Theoriginalpasswordis**neverstored**inthedatabase.

- **SESSIONSECURITYANDTOKENEXPIRY:**

  - ✓ JWTtokensincludean**expirytime**(e.g.,1hour).
  - ✓ Onceexpired,usersmust **loginagain**toreceiveanew token.
  - ✓ Thisreducesriskoflong-termtokenmisuseandimprovesoverallsystem security.

➢ FRONTENDTOKENHANDLING:

- Tokensarestoredin**localStorage**or**sessionStorage**ontheclientside.
- Duringlogout,thetokenisclearedfromstoragetopreventunauthorizedaccess.
- Thefrontendincludeschecks todetectexpiredormissingtokensandredirectusers to the login page.

➢ ERRORHANDLING:

→Learn-Hubprovidesclearandmeaningfulerrormessagesincaseof:

- Invalidlogincredentials(wrongemail/password)
- Expiredormissingtoken
- Unauthorizedaccesstorestrictedroutes
- Improperuserroles

Thishelpsbothusersanddevelopersquicklyidentifyandresolveissues.

➢ SECURITYBESTPRACTICES:

Toenhancetheoverallsecurityoftheauthenticationsystem:

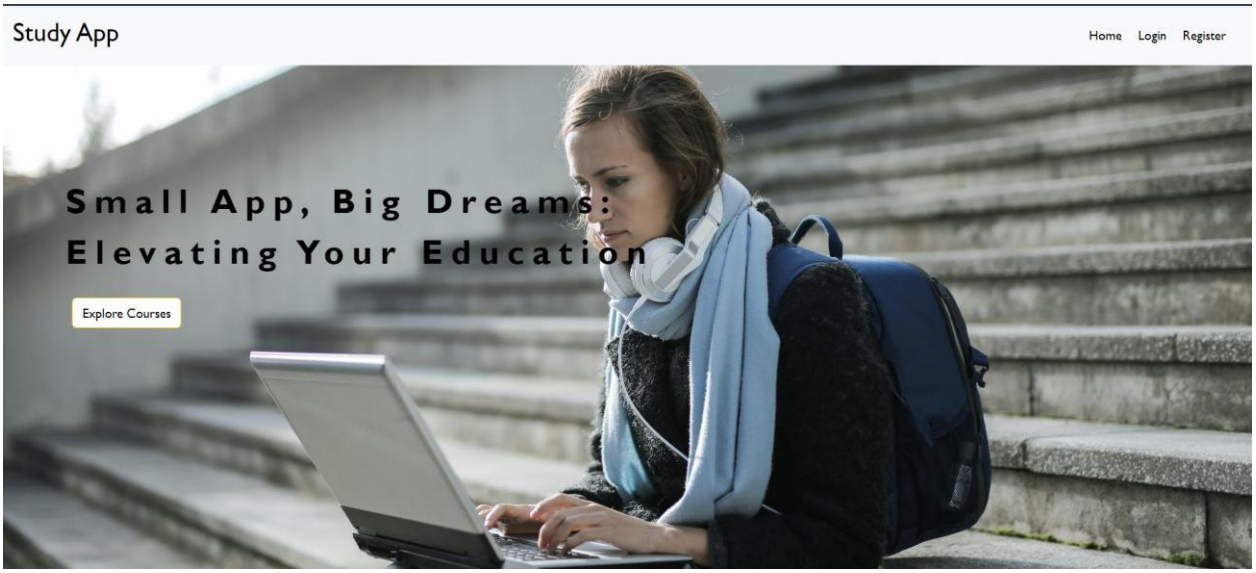
- HTTPSisusedtoencryptallclient-servercommunication.
- Inputfieldsarevalidatedonbothclientandservertopreventinjectionattacks.
- Ratelimitingandbrute-forceprotectionmechanismscanbeaddedtologinAPIs.

8. USERINTERFACE

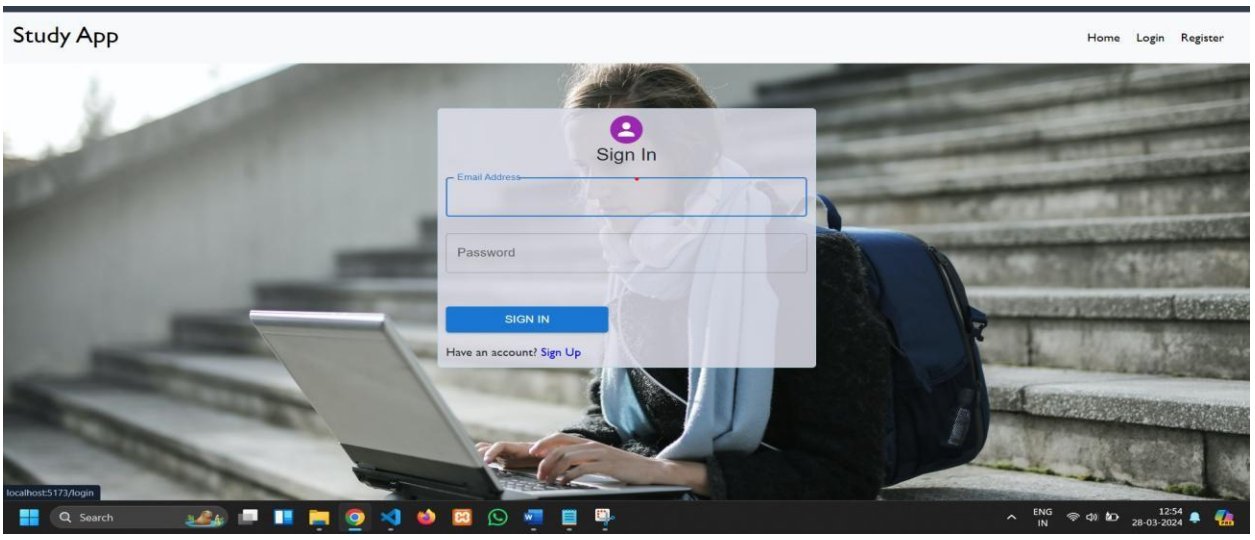
ProjectImplementation:

On completingthedevelopmentpart,wethen rantheapplicationonelasttimeto verifyallthe functionalities and look for any bugs in it. The user interface of the application looks a bit like the one's provided below.

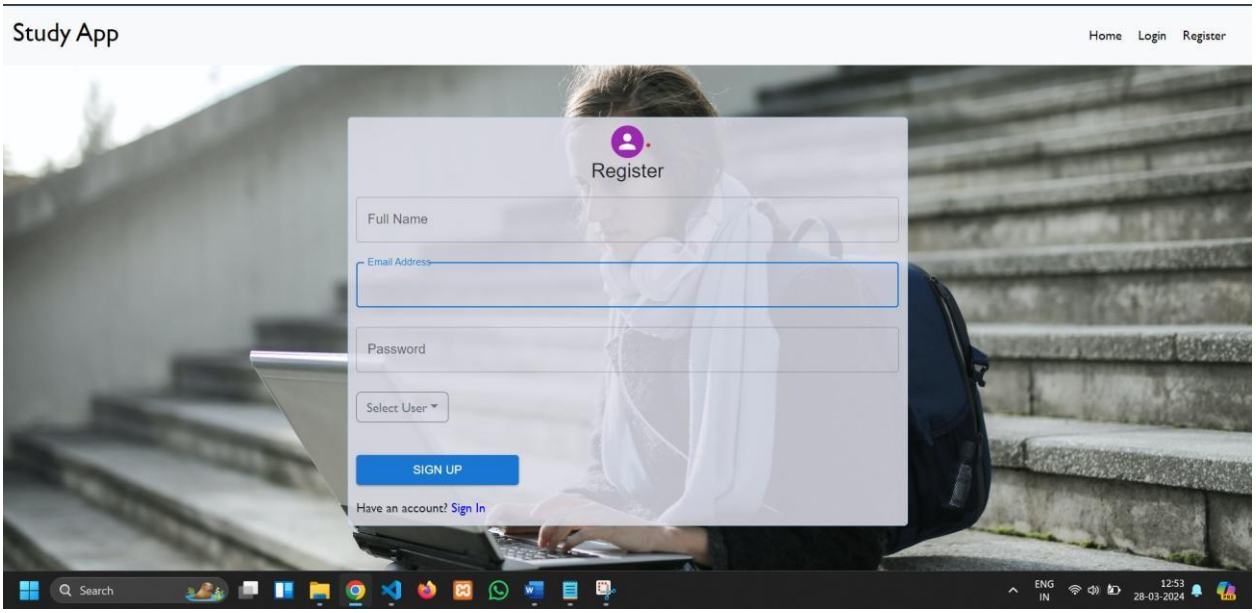
- **LANDINGPAGE:**



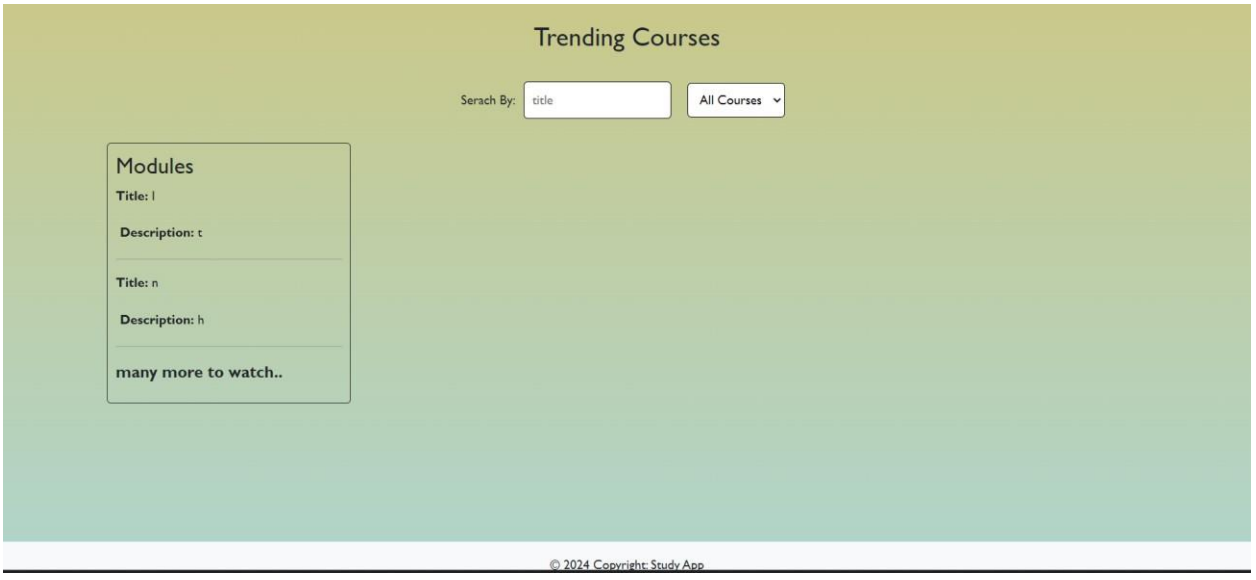
- **LOGINPAGE:**



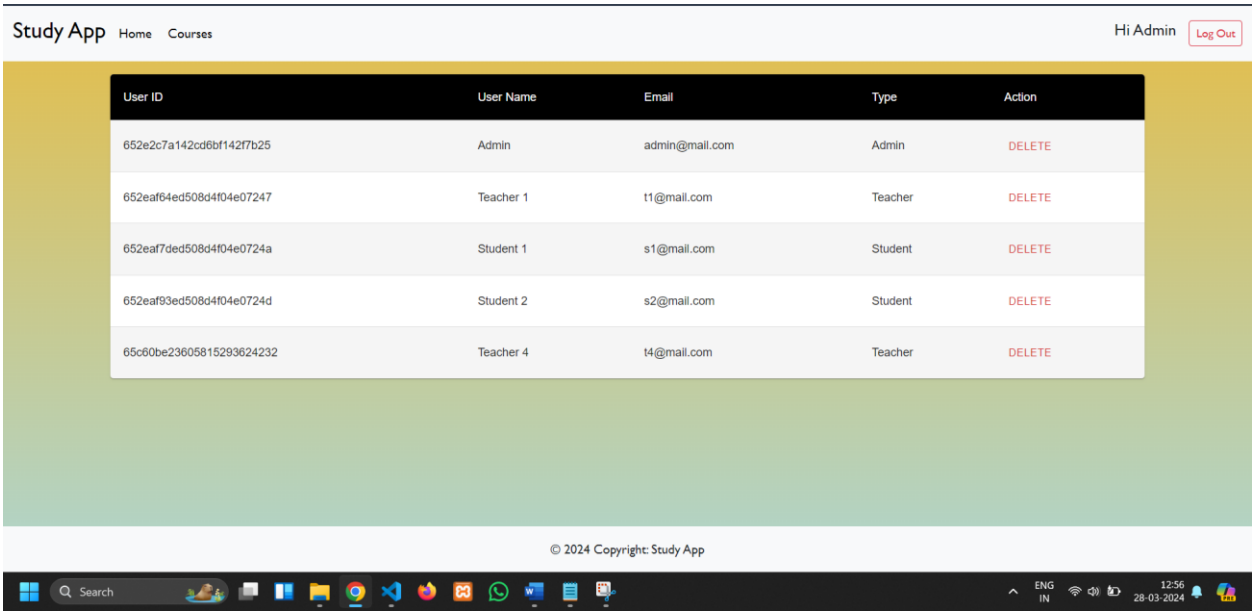
- **REGISTRATIONPAGE:**



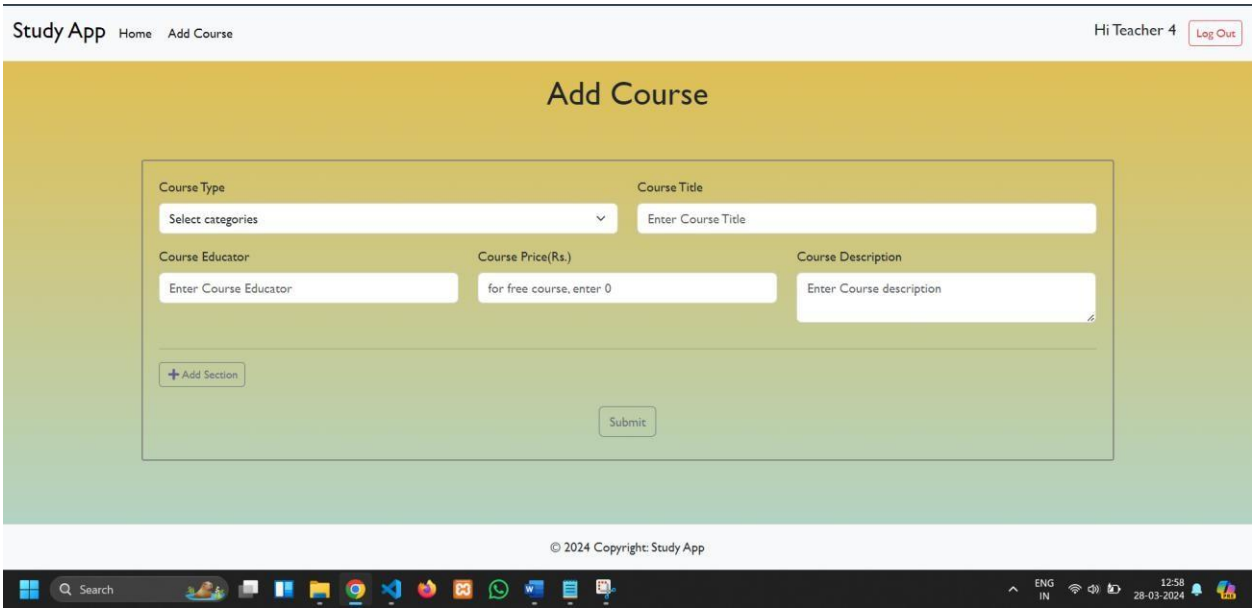
- **COURSESPAGE:**



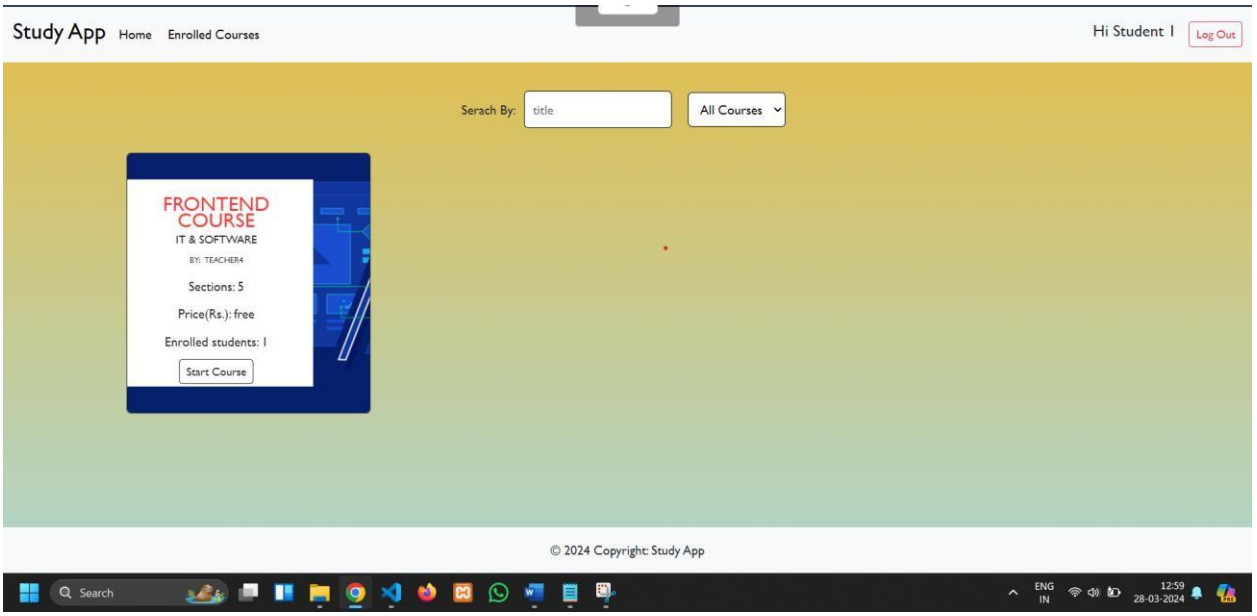
- **ADMINDASHBOARD:**

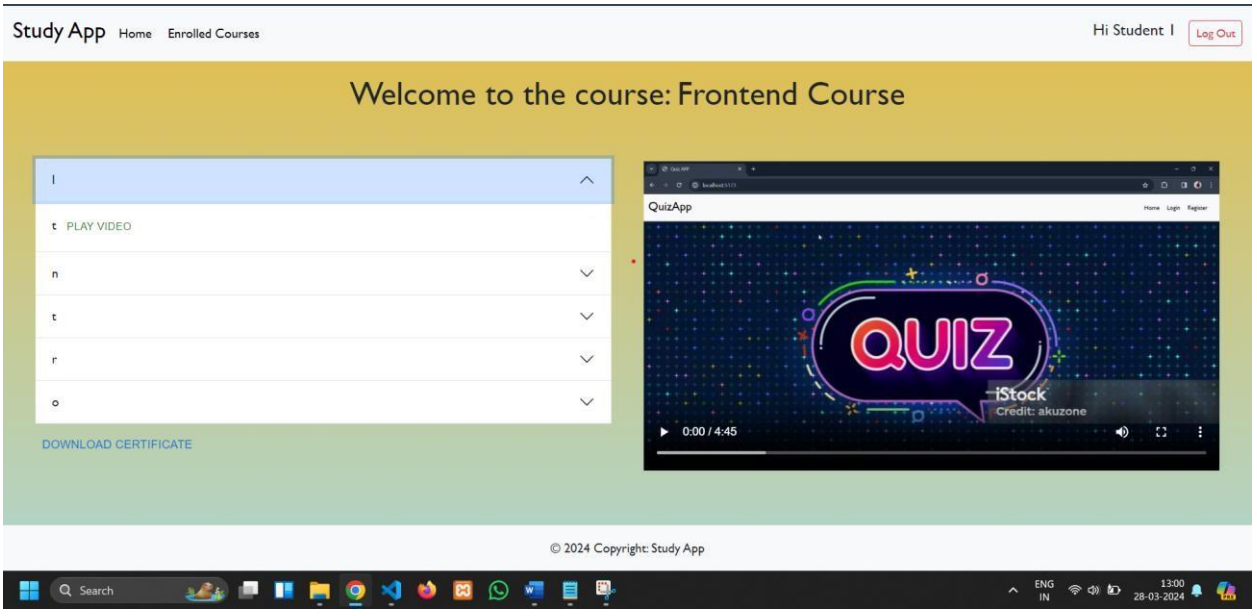


- **TEACHERDASHBOARD:**



- **STUDENTDASHBOARD:**





9. TESTING

- ManualTesting
  - ✓ Allmajormoduleswereманuallytestedusingrealuserscenarios.
  - ✓ Corefeatureslikeuserregistration,login,coursecreation,enrollment, video playback, and discussion/chat were verified for correctness.
- APITestingwithPostman
  - ✓ BackendAPIsweretestedindependentlyusingPostman.
  - ✓ Endpointssuchas:
    - POST/register
    - POST/login
    - POST/courses/add
    - GET /courses/:id
    - POST/enroll
    - POST/messages
- UI Testing
  - ✓ Alluserinterfacecomponentslike buttons,forms,coursecards,andchat inputs were tested for responsiveness and proper behavior.
  - ✓ UIwastestedacrossvariousscreensizesincluding mobile,tablet,and laptop, to ensure responsive design and cross-device compatibility.
  - ✓ Formvalidations(e.g.,requiredfields,emailformat)wereverified.

10. KNOWNISSUES

- ✓ **Noemailverification**isimplementedafteruserregistration,whichmayaffect account authenticity.
- ✓ **Coursediscussionordoubt-solvingchatdoesnotauto-refresh**;usersneedto manually reload the page to see new messages.
- ✓ **Fileorassignmentuploads** arenotcurrentlysupportedincoursecontentor submissions.
- ✓ **Chatfeatureusespollinginsteadofreal-timesockets**,whichmaycausedelaysin message delivery.
- ✓ **Roleassignment(student/teacher/admin)**ismanuallyhandled;there'snouser interface for admin role management.
- ✓ **Someformslackclient-sidevalidation**,leadingtopoorUXwhenusersforgettofill required fields like email or password.

11. FUTUREENHANCEMENTS

- ✓ **Real-time Doubt Solving via Chat (Socket.IO):**
  Enableinstantmessagingbetweenstudentsand instructors for live query resolution.

- ✓ **Email&SMSNotifications:**
  Notifylearnersaboutcourseupdates,newcontent, assignment deadlines, or feedback.

- ✓ **LearningHistory&ProgressAnalytics:**
  Allowadminsandeducatorstoviewdetailedanalyticson course completion rates, student performance, and engagement trends.

- ✓ **FileUpload &Assignment Submission:**
  Supportforstudentstouploadhomework,projectfiles,or essays directly through the platform.

- ✓ **MultilingualSupport:**
  Offerregionallanguageoptionstoimproveaccessibility and inclusiveness for diverse learners.

- ✓ **PushNotifications:**
  Provideinstantbrowser/mobilealertsaboutnewcourse releases, instructor announcements, or reminders.

- ✓ **Instructor&AdminRoleManagementPanel:**
  Simplifiedinterfaceformanaginguserroles,permissions, and responsibilities across courses.

- ✓ **MobileApp Version:**
  Developacross-platform**ReactNative**apptoenable learning on-the-go for Android and iOS users.

RunningtheApplication:

A. FRONTEND:

- Torunthe**Reactfrontend**-
  - ✓ Openterminalandnavigatetothefrontendfolder:

```
cd frontend
```

  - ✓ Install dependencies:

```
npm install
```

  - ✓ Startthe frontend:

```
npm run dev
```

  - ✓ Openbrowserandvisit:

    http://localhost:5173

B. BACKEND:

- Toruntheb**Node.js+Express backend**:
  - ✓ Openanotherterminalandnavigatetothebackendfolder:

```
cd backend
```

  - ✓ Installdependencies:

```
npm install
```

  - ✓ Startthebackend server:

```
npm start
```

  - ✓ Server runs at:

    http://localhost:8000

DEMOLINKS

- VideoDemo Link:

  https://drive.google.com/drive/folders/1W7iWq-6_-Kuc_1hwYQDEBCAve10TrBAO

- Drive Link:

  https://drive.google.com/drive/folders/15pjLvgXIpNMN9EdLx1FSAx4V3seb_QYC