ONLINE LEARNING PLATFORM USING MERN STACK

Project Title: **Learnhub: Your Center For Skill Enhancement**

TEAM INFO:-

Team ID : LTVIP2025TMID5710
Team Size 5
Team Leader : CHARISHMA YARRAMOTHU
Team member : SRIVALLI MULIPIRI
Team member : SUSMITHA UNNAM
Team member : ASHAJYOTHI MUSUNURI
Team member : LAKSHMI PRASANNA PASUPULETI

1.INTRODUCTION

Learnhub:Your Center For Skill Enhancement

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.

2. PROJECT OVERVIEW

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.

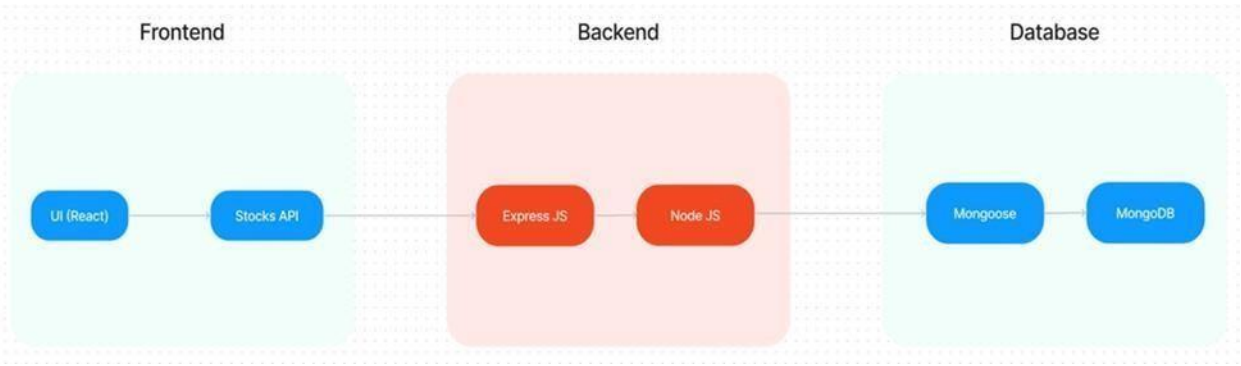
Paid Courses: Sarah discovers an advanced web development course that requires payment. She purchases the course using the platform's payment system and gains access to premium content.

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.

3. ARCHITECTURE

TECHNICAL ARCHITECTURE:



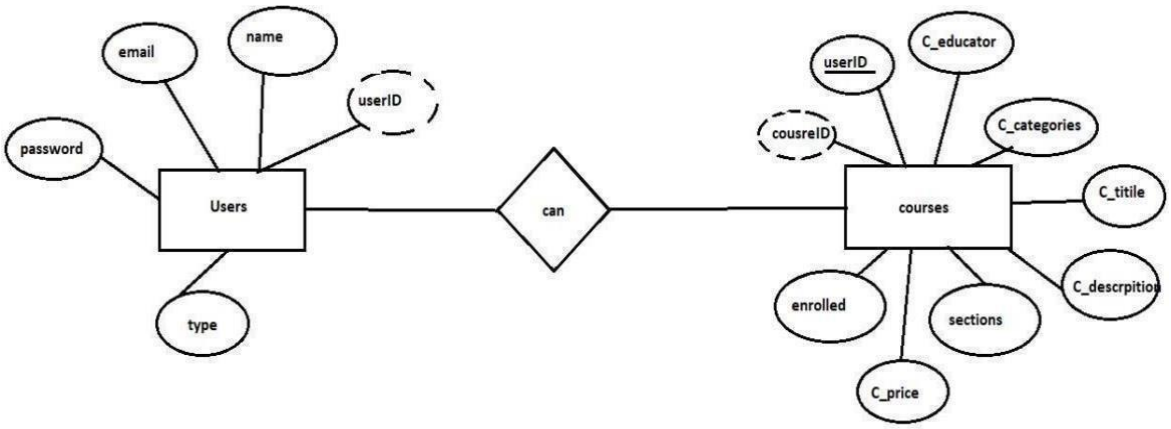
The technical architecture of OLP app follows a client-server model, where the frontend serves as the client and the backend acts as the server. The frontend encompasses not only the user interface and presentation but also incorporates the axios library to connect with backend easily by using RESTful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

ER Diagram:



Here there are 2 collections namely users, courses that have their own fields in

Users:

1. _id: (MongoDB creates by unique default)
2. name
3. email
4. password
5. type

Courses:

1. userID: (can act as a foreign key)
2. _id: (MongoDB creates by unique default)
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 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/

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
```

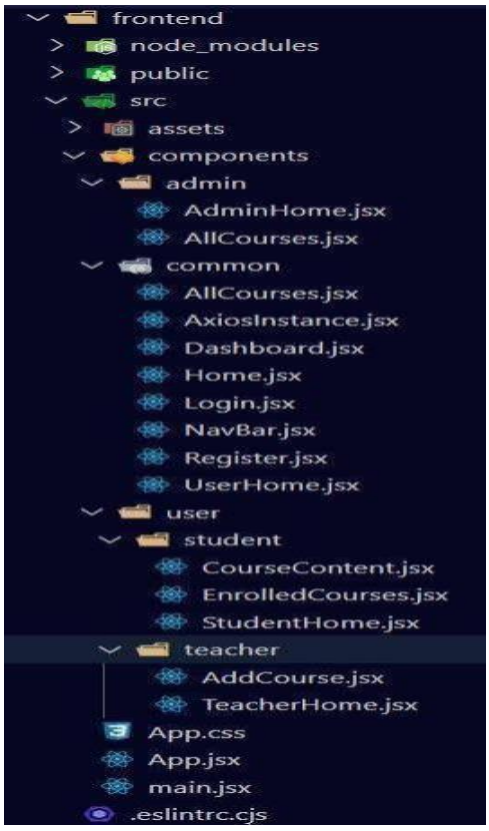
- The OLP app will be accessible at http://localhost:5173

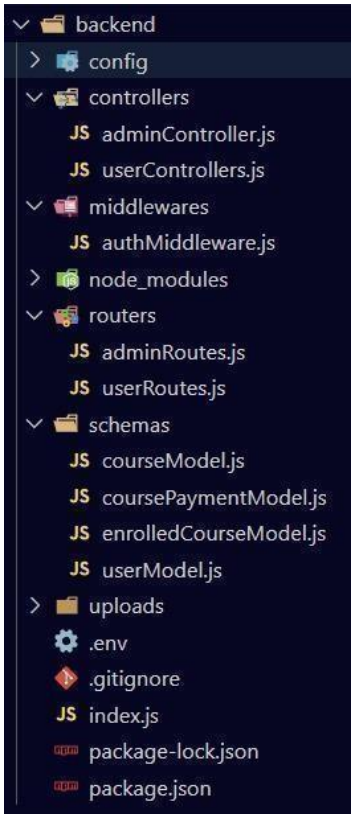
You have successfully installed and set up the Online learning app on your local machine. You can now proceed with further customization, development, and testing as needed.

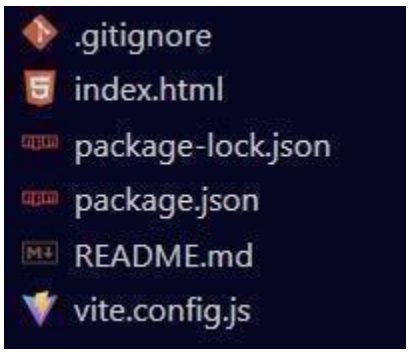
5. PROJECT STRUCTURE

The first image is of the front part which shows all the files and folders that have been used in UI development

The second image is of the Backend part which shows all the files and folders that have been used in the backend development







Roles & Responsibilities:

Application Flow: The project has a user called– teacher and student and the other will be Admin which takes care 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:

1. Can add courses for the student.
2. Also, delete the course if no student enrolled in it or for any other reasons.
3. Also, add sections to courses.

Student:

4. Can enroll in an individual or multiple courses.
5. Can start the course where it has stopped.
6. Once the course is completed, they can download their certificate of completion of the course.
7. For a paid course, they need to purchase it and then they can start the course.
8. They can filter out the course by searching by name, category, etc

Admin:

9. They can alter all the courses that are present in the app.
10. Watch out for all kinds of users in the app.
11. Record all the enrolled students that are enrolled in the course.

6. API DEVELOPMENTATION

Milestone 1- Setup & configuration

- **Folder setup:**

1. Create frontend and
2. Backend folders

. Open the backend folder to install the necessary tools

For backend, we use:

- cors
- bcryptjs
- express
- dotenv
- mongoose
- Multer
- Nodemon
- jsonwebtoken

```
{
  "name": "backend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  ▷ Debug
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "nodemon index"
  },
  "dependencies": {
    "bcryptjs": "^2.4.3",
    "cors": "^2.8.5",
    "dotenv": "^16.3.1",
    "express": "^4.18.2",
    "jsonwebtoken": "^9.0.2",
    "mongoose": "^7.5.2",
    "multer": "^1.4.5-lts.1",
    "nodemon": "^3.0.1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Milestone 2- 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.

REF(BACKED):https://drive.google.com/file/d/19nPuDsC3FVJzEb_fjoNOAm_zw2yxSB1k/view?usp=drivesdk

Milestone 3- Database

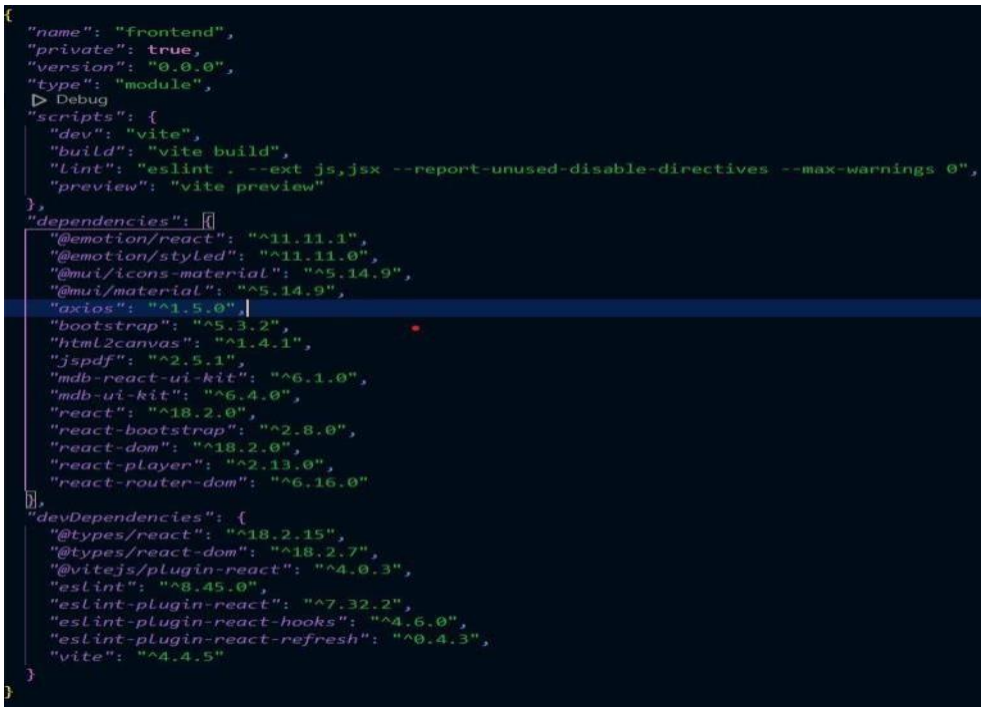
- **Configure MongoDB**

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.
   ref: database.mp4

Milestone 4- Frontend 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



REF(FRONTEND):https://drive.google.com/file/d/19o9dSZKDmhvB5hdQdw1hAQuaHcLN-PIW/view?usp=drivesdk

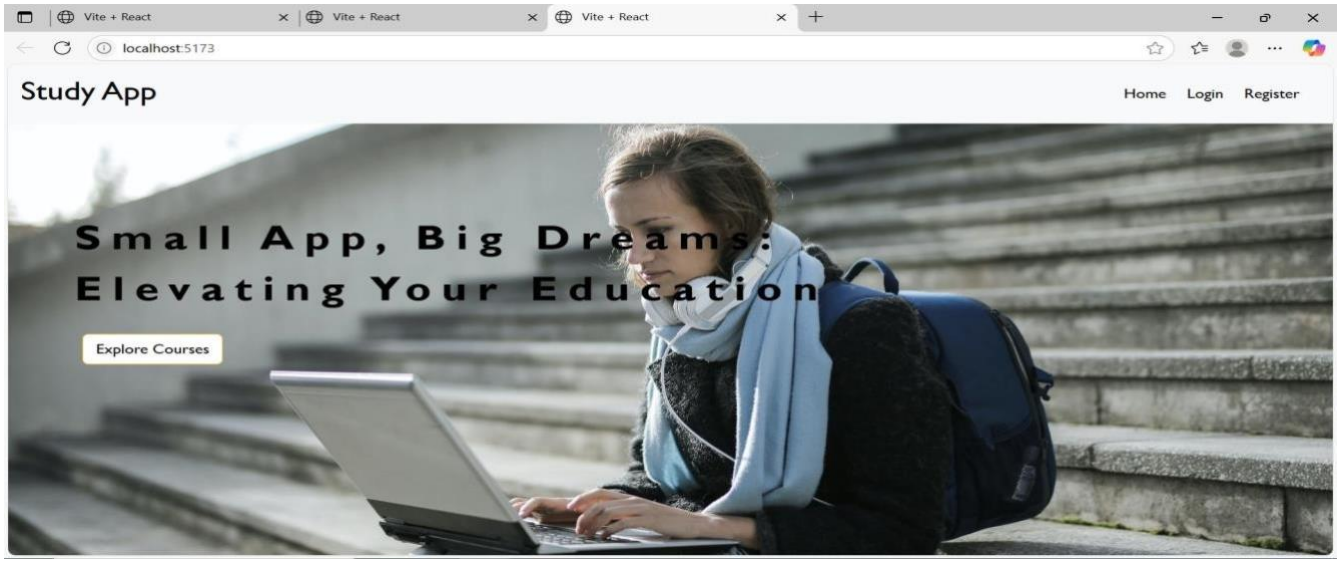
7.USER INTERFACE

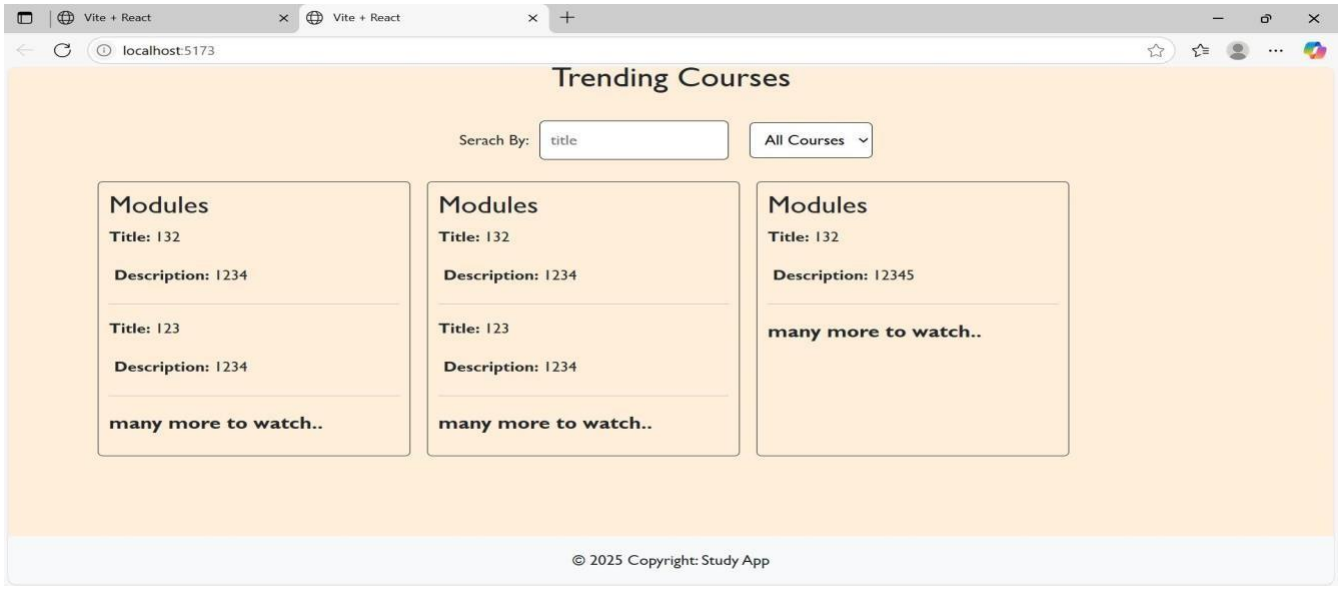
Milestone 5: Project Implementation:

On completing the development part, we then ran the application one last time to verify all the functionalities and look for any bugs in it. The user interface of the application looks a bit like the one's provided below.

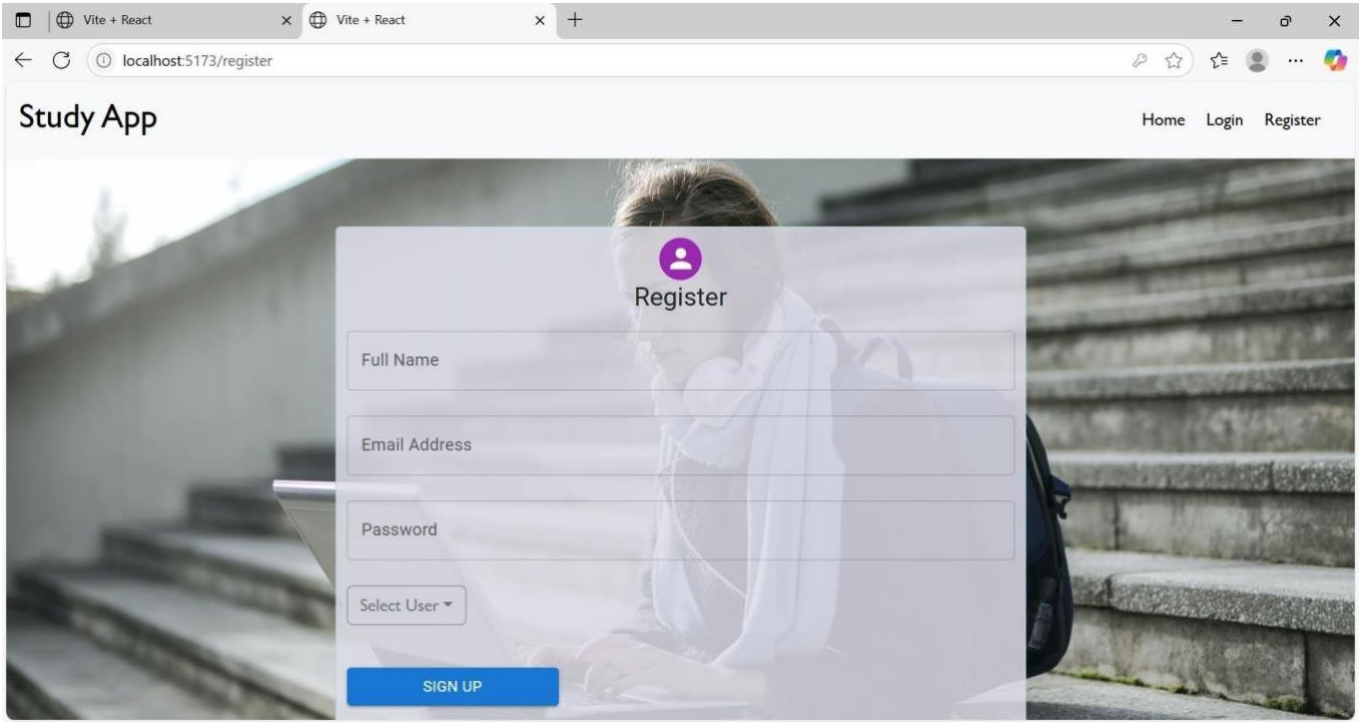
SCREENSHOTS:-

Landing page:

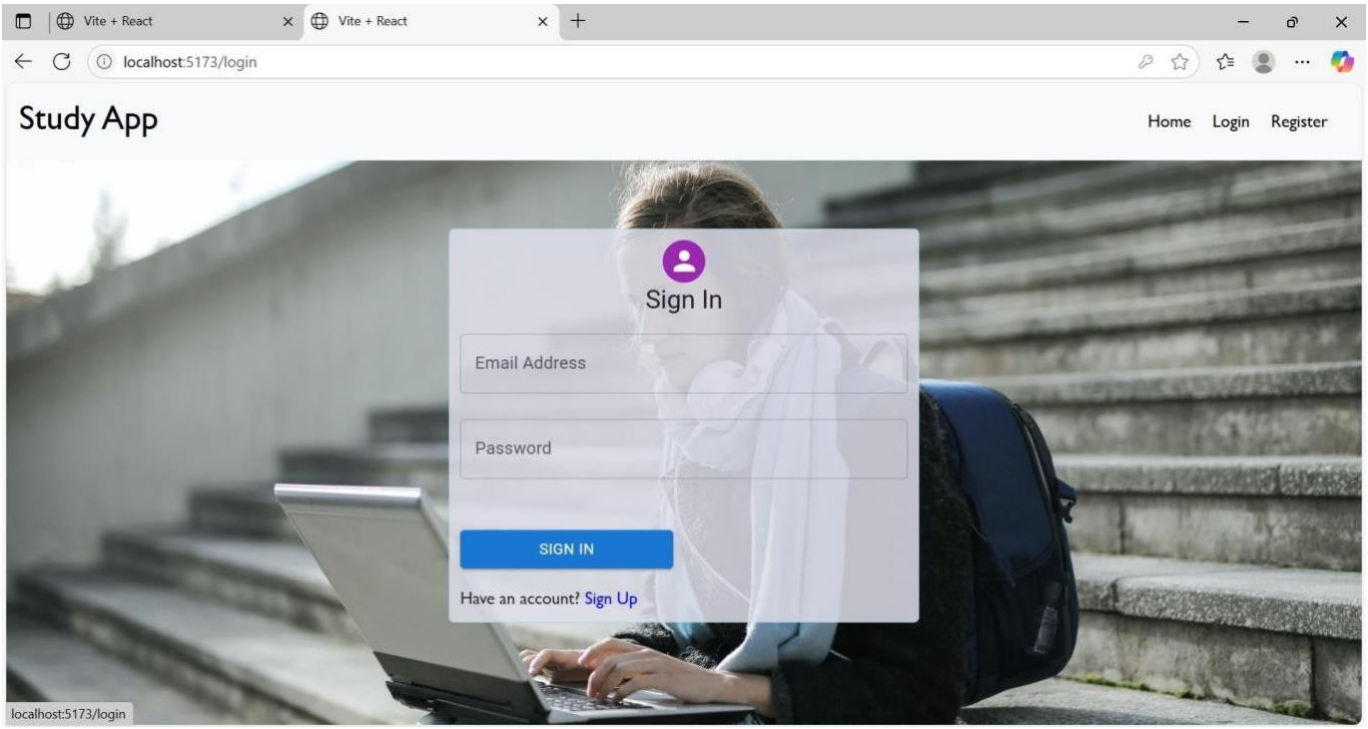




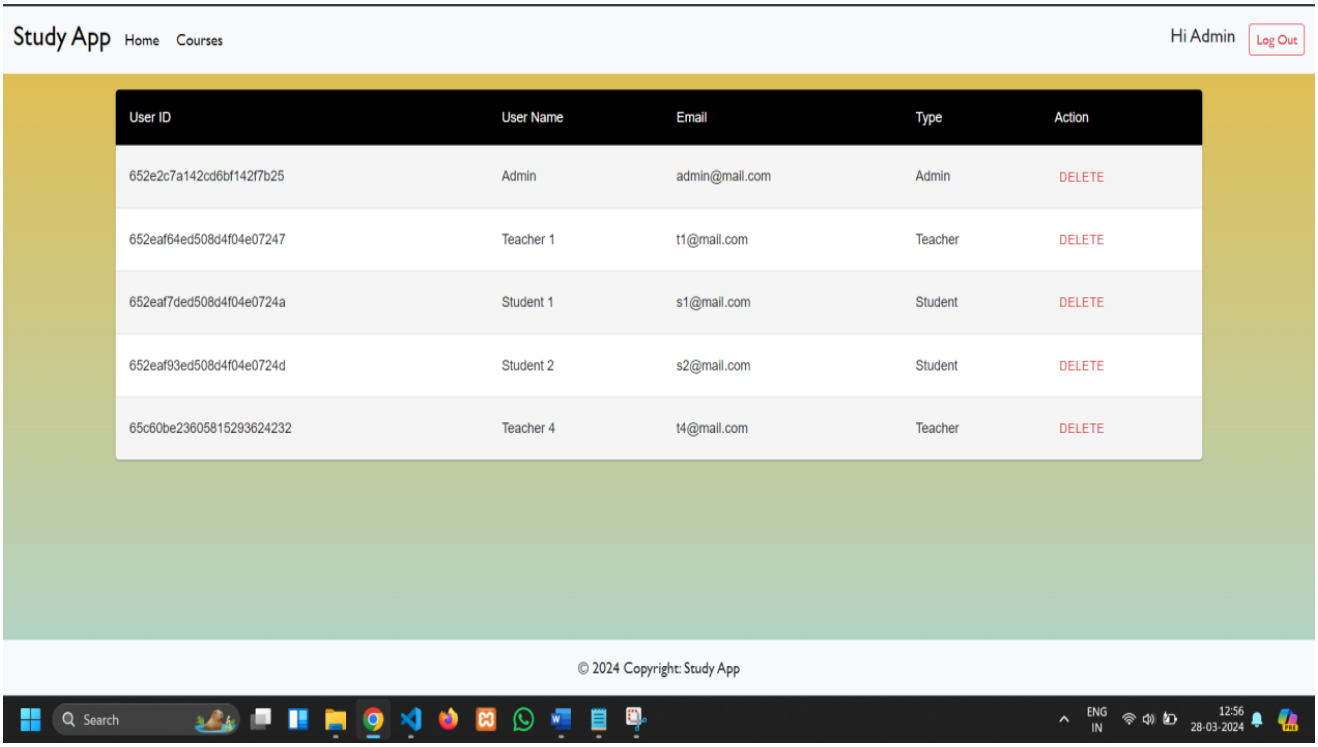
. **Register page**:



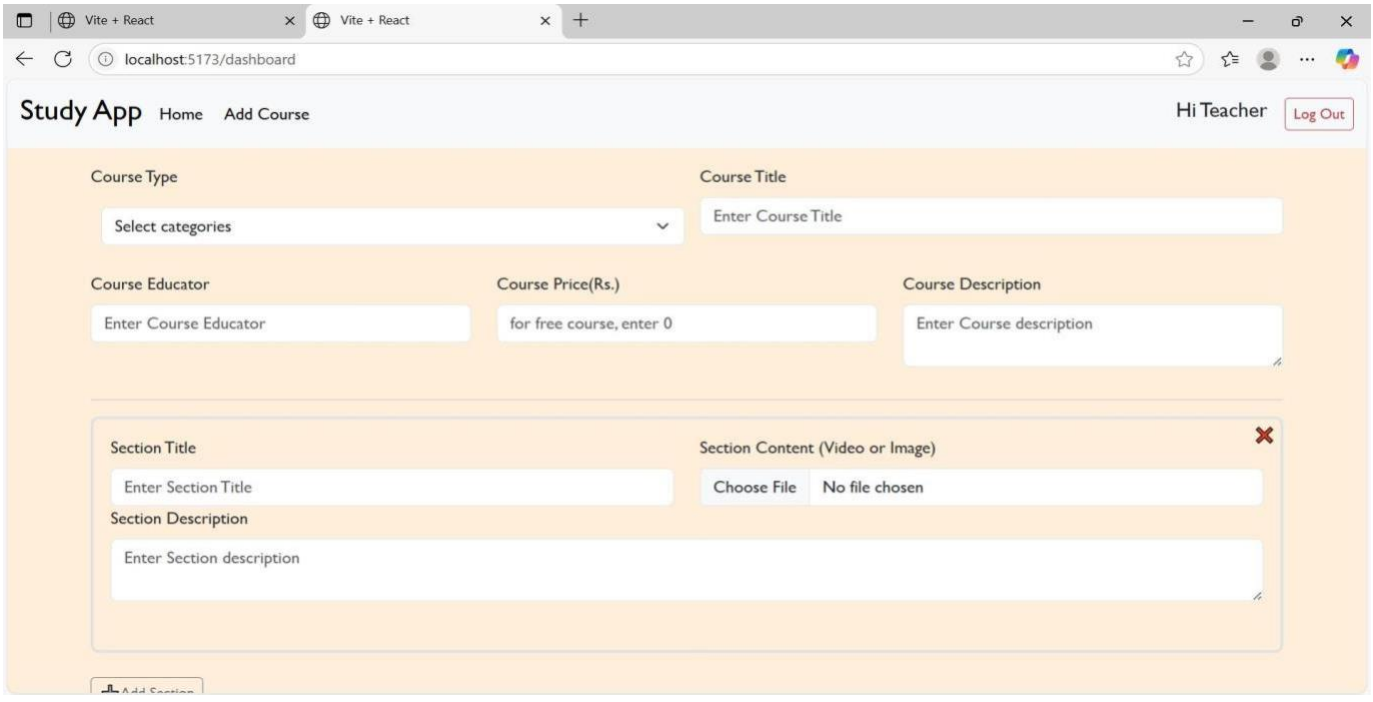
Login Page:



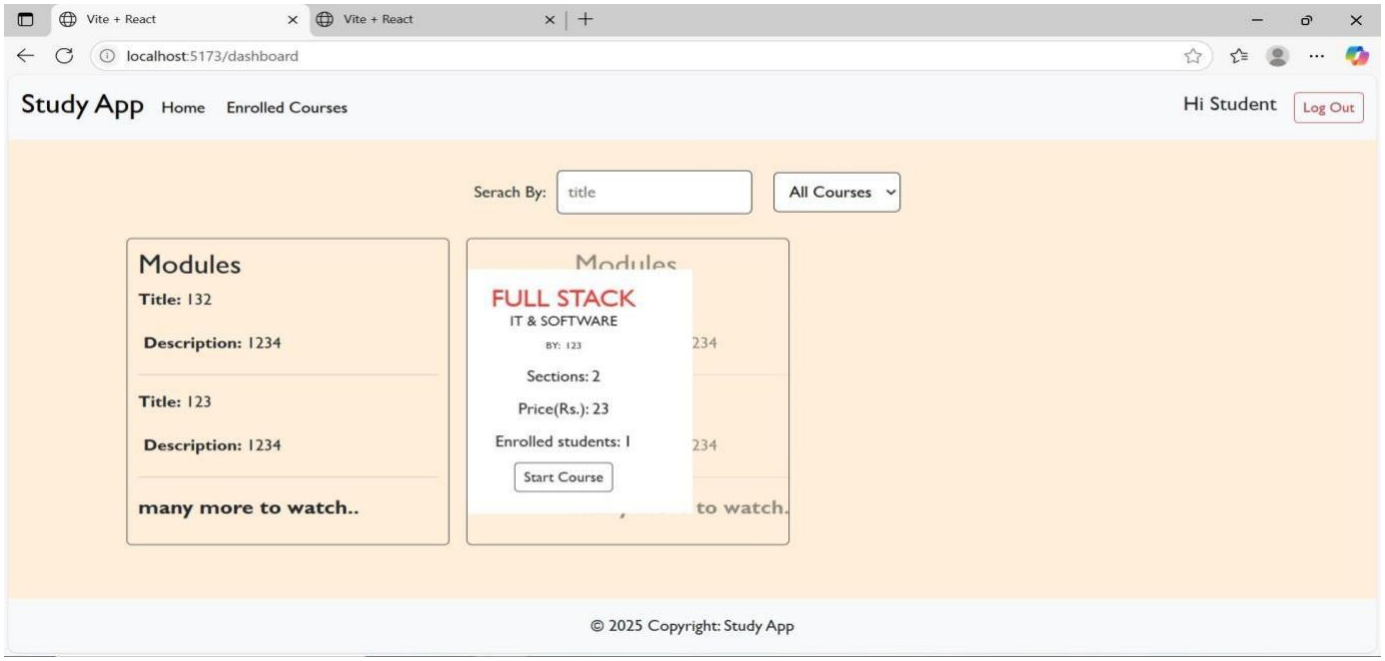
Admin Dashboard:

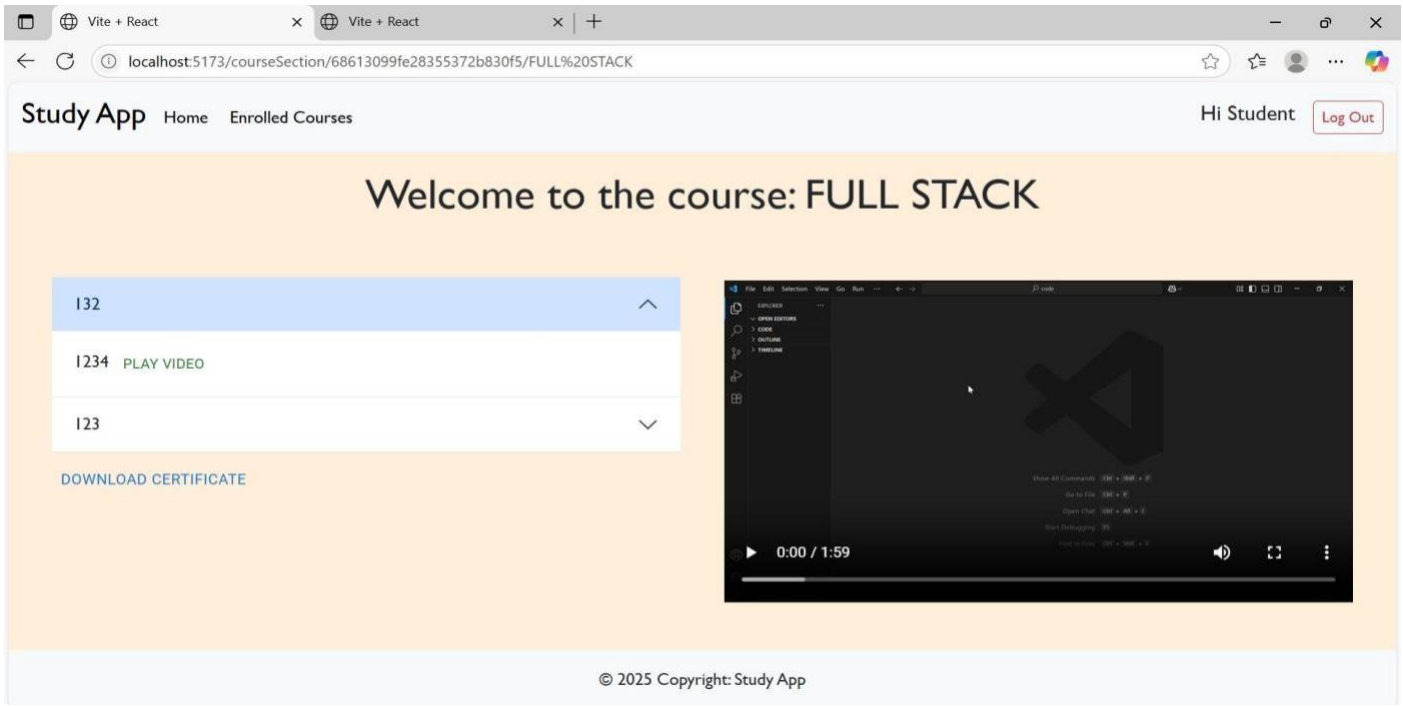


Teacher Dashboard:



- **Student Dashboard:**





8. AUTHENTICATION

1. User Registration & Secure Password Storage

- Users register with their **email, password, and role**.
- Passwords are:
  - **Hashed using bcrypt** before saving to the database.
  - **Salted** to prevent rainbow table attacks.
- Email uniqueness is enforced to prevent duplicate accounts.

2. Login & Token-Based Session Management

- On successful login:
  - A **JWT (JSON Web Token)** is generated and returned to the frontend.
  - This token includes:
    - User ID
    - Role (e.g., student, instructor, admin)
    - Token expiry
- Tokens are stored:
  - **Frontend:** in HttpOnly cookies or local storage.
  - **Backend:** optionally saved in a blacklist for logout and session control.

3. Access & Refresh Token Strategy

- **Access Token:**
  - Short-lived (e.g., 15 mins – 1 hour)
  - Used to authenticate user API requests.
- **Refresh Token:**
  - Long-lived (e.g., 7–30 days)
  - Stored securely (HttpOnly cookie)
  - Used to re-issue new access tokens without forcing user logout.

4. Logout Functionality

- On logout:
  - Tokens are **invalidated or blacklisted** server-side (if implemented).
  - Refresh tokens are deleted from storage.
  - Client state is cleared (e.g., Redux store, localStorage, cookies).

5. Email Verification

- After registration, an **email verification link** is sent.
- The link contains a **JWT with limited validity**.
- The user must verify their email before they can log in.

6. Forgot Password / Reset Flow

- Users who forget their password:
  - Request a password reset.
  - Receive a **secure token link via email.**
  - Can set a new password after verifying token validity and expiration.

9. KNOWN ISSUES

1. Login & Access Issues

- **Issue:** *"Invalid Credentials"* error on correct password.
  Cause: Session timeout, case-sensitive input, or server-side delay.
  Fix: Clear browser cache, check Caps Lock, or reset password.
- **Issue:** Two-Factor Authentication (2FA) not working.
  Cause: Delayed OTP delivery or wrong time sync.
  Fix: Use authenticator app instead of SMS; sync device time.

2. Course-Related Issues

- **Issue:** Course videos not playing.
  Cause: Unsupported browser, slow internet, or expired session.
  Fix: Use updated browser (Chrome/Edge), check network speed, refresh session.
- **Issue:** Incomplete or missing modules.
  Cause: Course update in progress or content access misconfiguration.
  Fix: Contact support with course name and ID.

3. Assessment & Certification Problems

- **Issue:** Quiz not submitting or freezing.
  Cause: Browser incompatibility or network drop.
  Fix: Use stable connection; avoid back button or refresh mid-quiz.
- **Issue:** Certificate not generated after course completion.
  Cause: Final assessment not marked as passed or system bug.
  Fix: Recheck if all lessons are marked complete; then retry download or contact support.

4. Payment and Subscription Errors

- **Issue:** Paid course still showing as locked.
  Cause: Payment gateway delay in confirming transaction.
  Fix: Log out and back in; contact support with payment receipt.
- **Issue:** Subscription auto-renewed without notification.
  Cause: Auto-renew toggle was enabled by default.
  Fix: Contact billing@learnhub.com for refund and cancel subscription settings.

5. Mobile App Issues (if applicable)

- **Issue:** App crashes on older Android versions.
  Fix: Update to latest app version or switch to web version.
- **Issue:** Notifications not received.
  Fix: Check phone settings to allow push notifications for Learnhub.

10. FUTURE ENHANCEMENT

Future Enhancements for Learnhub

To keep Learnhub at the forefront of online skill development, the following enhancements are planned for future releases:

1. AI-Powered Personalized Learning Paths

- Use AI and machine learning algorithms to recommend courses based on learner's goals, past progress, and skill gaps.
- Dynamic adjustment of learning materials according to user performance and feedback.

2. Gamification & Rewards System

- Introduce badges, points, leaderboards, and achievement milestones to motivate learners.
- Reward consistent engagement and course completions with certificates, discounts, or digital trophies.

3. Enhanced Mobile Experience

- Develop native mobile apps with offline learning capabilities.
- Push notifications for course updates, reminders, and personalized messages.

4. Live Virtual Classrooms & Webinars

- Integration of live streaming and interactive webinars for real-time instructor-learner engagement.
- Support for breakout rooms, polls, and Q&A sessions.

5. Expanded Social Learning Features

- Learner discussion forums, peer reviews, and collaborative projects.
- Networking features to connect learners with similar interests.

6. Advanced Analytics Dashboard

- Provide learners with insights into their learning habits, strengths, and areas to improve.
- Instructors and admins get detailed reports on course effectiveness and user engagement.

7. Multi-language Support

- Expand platform accessibility by adding multiple language options for UI and course content.

8. Integrations with Industry Tools

- Connect with popular tools like LinkedIn Learning, GitHub, and certification bodies for seamless skill validation.

9. Improved Security & Privacy

- Continuous updates to authentication methods, data encryption, and compliance with evolving data protection laws.

10. Accessibility Enhancements

- Compliance with WCAG standards for users with disabilities, including screen reader support, subtitles, and alternative input methods.

Note: For GitHub Repository and Demo video:-click on below links

Github Repository Links Are Below Of All This Team Members k :-

YARRAMOTHU CHARISHMA:https://github.com/CharishmaYarramothu/Learnhub-Your-Center-For-Skill-Enhancementt.git
MULIPIRI SRIVALLI:https://github.com/srivallimulipiri/Learnhub-Your-Center-For-Skill-Enhancementt.git
UNNAM SUSMITHA:https://github.com/susmitha6300/Learnhub-Your-Center-For-Skill-Enhancementt.git
. **MUSUNURI ASHAJYOTHI:**https://github.com/Asha-2004/Learnhub-Your-Center-For-Skill-Enhancementt.git
PASUPULETI LAKSHMI PRASANNA:https://github.com/Prasanna-3008/Learnhub-Your-Center-For-Skill-Enhancementt.git

Demo Video: Click https://drive.google.com/file/d/1Y7oRCmTmcbg_HgXaxCbcDNUZ1Rp7m-If/view?usp=drivesdk