| | |
|---|---|
| **Date** | **19/02/2026** |
| **Team ID** | **LTVIP2026TMIDS71135** |
| **Project Name** | **LearnHub – Online Learning Platform** |

# Project Setup and Configuration

This document explains the complete project flow of LearnHub (Online Learning Platform) including project setup, backend development, database configuration, frontend development, and final execution. It also includes screenshots of the application UI and package configurations.

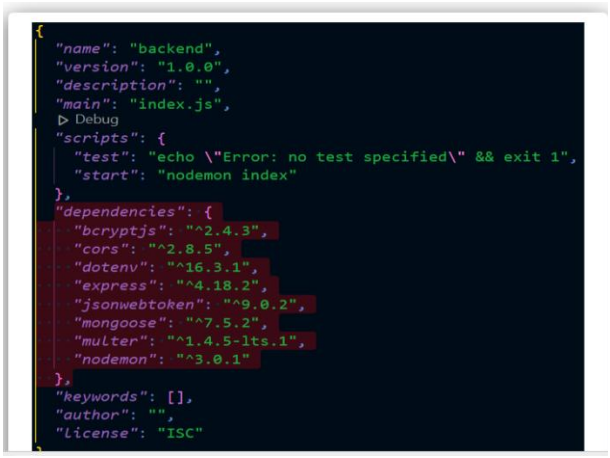
### 1. Project Setup and Configuration

The LearnHub project is divided into two major parts: frontend and backend. During the initial setup, separate folders are created for both components to maintain clean architecture.

Folder Setup Steps:

- Create two folders: frontend and backend.
- Open the backend folder and install the required backend dependencies.
- Open the frontend folder and install the required frontend dependencies.

**Backend package configuration (package.json):**



```
{
  "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"
}
```

### 2. Backend Development

The backend of LearnHub is developed using Node.js and Express.js.

Backend dependencies used in this project include:

- cors
- bcryptjs
- express
- dotenv
- mongoose
- multer
- nodemon
- jsonwebtoken

Backend Implementation Steps:

• Create an index.js file in the backend folder.

• Configure Express server and define the PORT.

• Store MongoDB connection string and JWT secret key in the .env file.

• Add middleware such as cors and body-parser for request handling.

• Implement authentication using authMiddleware.js inside the middlewares folder.

**3. Database Development (MongoDB)**

MongoDB is used as the primary database for LearnHub. Mongoose is used as the ODM library to define schemas and interact with the database.

Database Setup Steps:

• Configure MongoDB locally or using MongoDB Atlas.

• Import mongoose in the project.

• Add database connection logic inside config/config.js.

• Create schemas/models in the schemas folder (e.g., userModel, courseModel, enrolledCourseModel).

**4. Frontend Development**

The frontend of LearnHub is developed using React.js with Vite. The UI is designed using Bootstrap and Material UI, and Axios is used for API communication.

Frontend libraries used in this project include:

• React

• Bootstrap

• Material UI

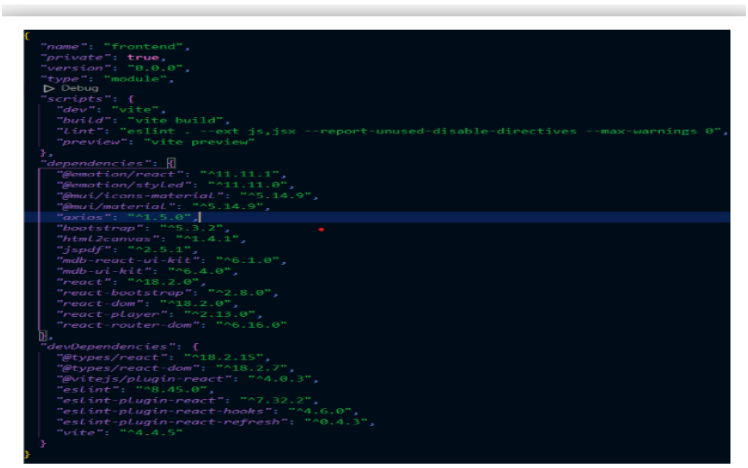
• Axios

• Ant Design (antd)

• mdb-react-ui-kit

• react-bootstrap

**Frontend package configuration (package.json):**



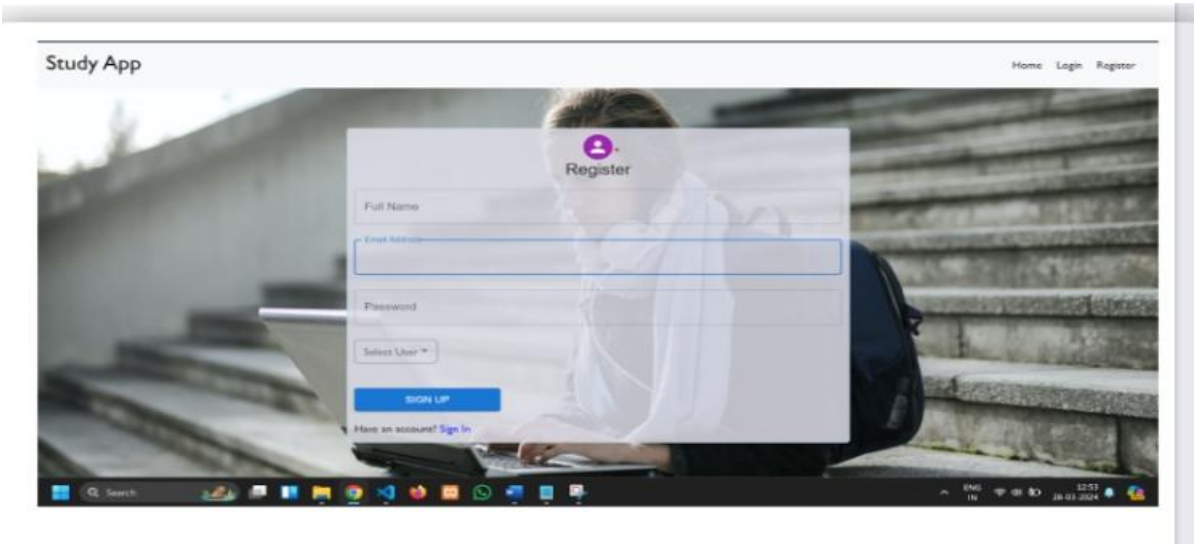
**5. Project Implementation and Execution**

**After completing development, the LearnHub application is executed locally to verify all major features and ensure that there are no bugs. The following screenshots show the user interface of the application.**

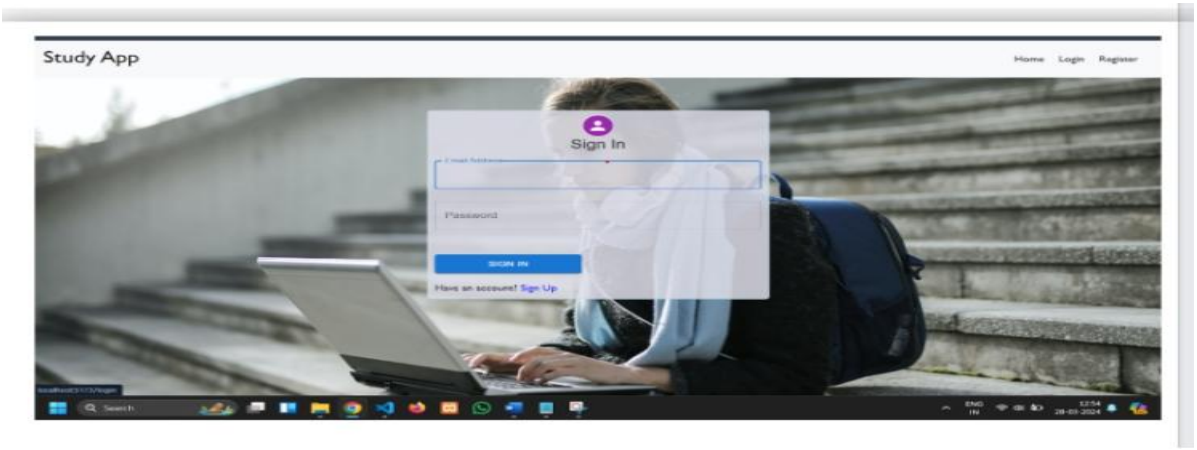
**Landing Page**


**Trending Courses Page**

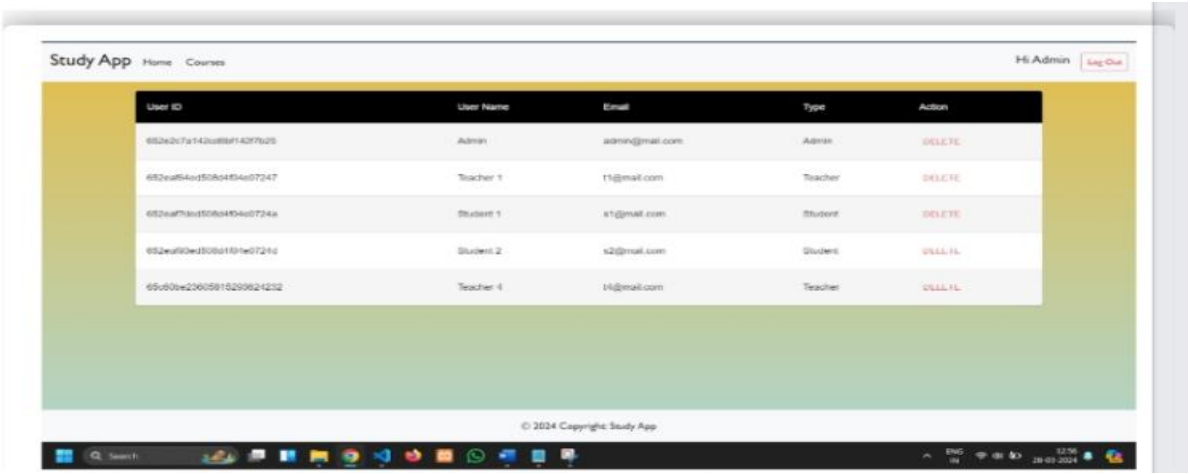

**Register Page**



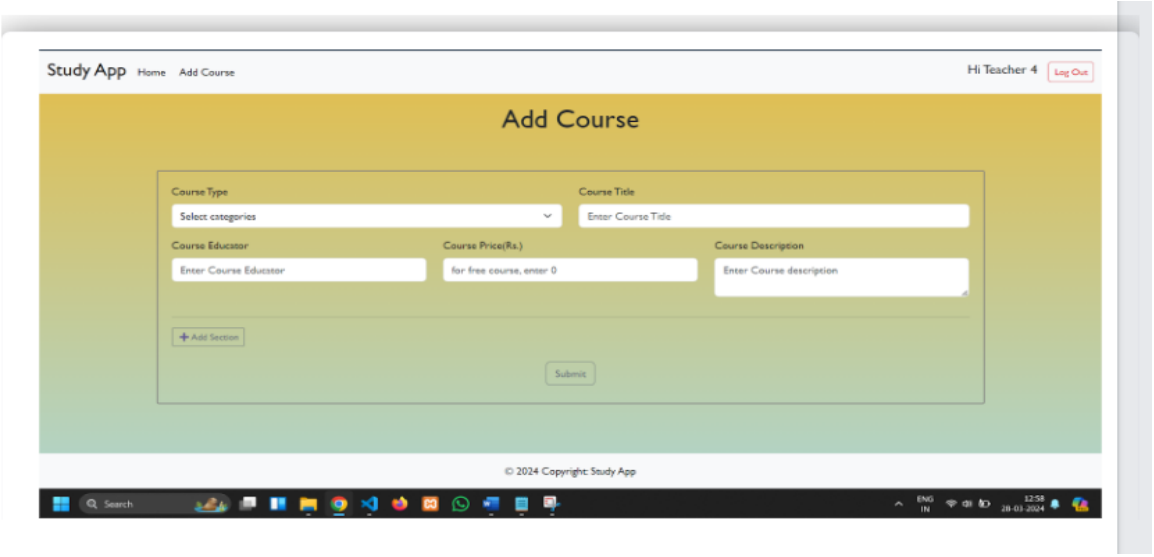
**Login Page**



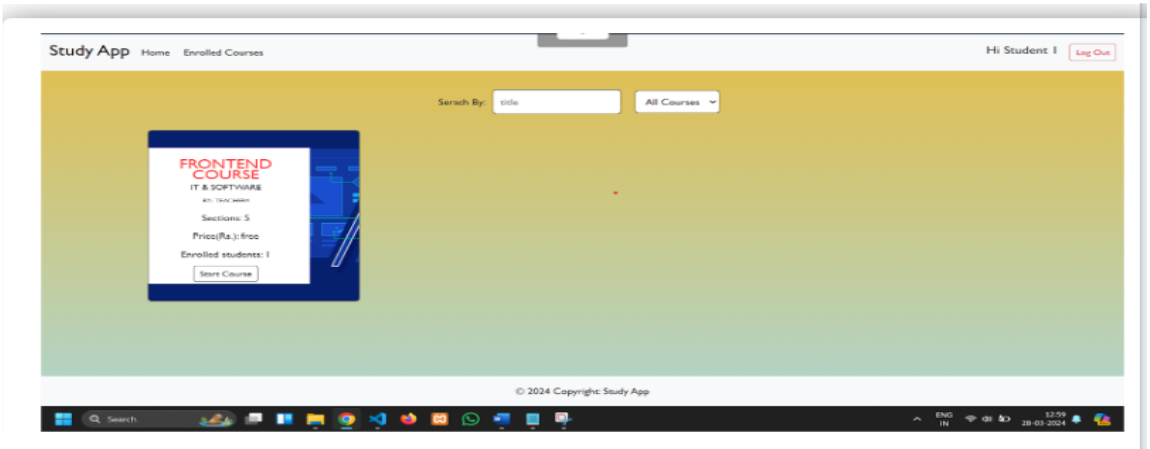

**Admin Dashboard**

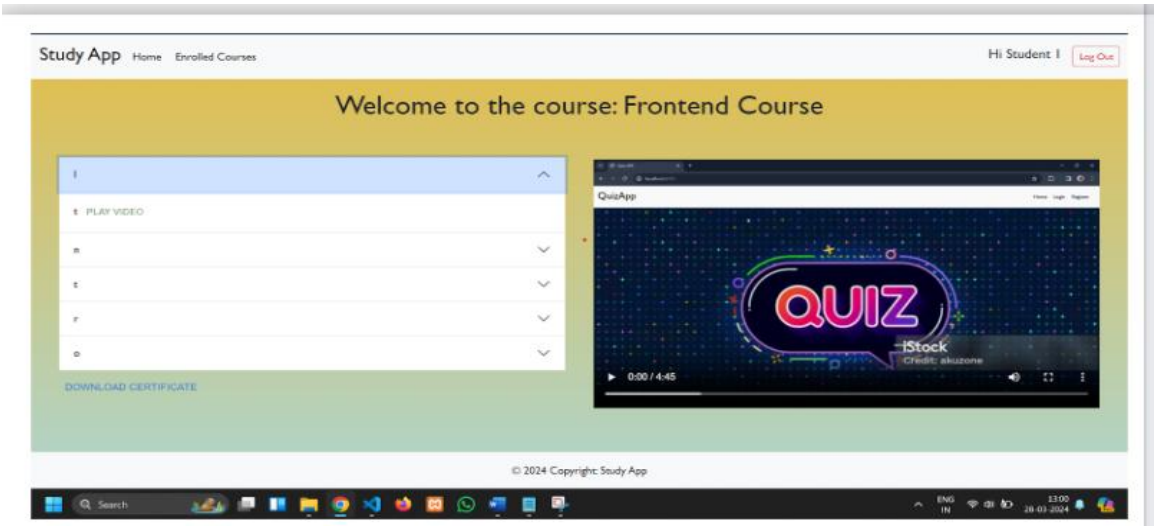



**Teacher Dashboard:**



**Student Dashboard:**




**6. Reference**

**For the complete source code drive and demo video, use the following link:**

**https://drive.google.com/drive/folders/1tDDOZqnWSgRxzXw20gyimiAPJI7WNRaz**