



JEPPIAAR
ENGINEERING COLLEGE

JEPPIAAR ENGINEERING COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ONLINE LEARNING PLATFORM USING MERN

Team Leader : Renuga S (310821104078)
Team Member 1 : Visalakshi AN (310821104114)
Team Member 2 : Anovah Sherin H (310821104010)
Team Member 3 : Clarrieous Athiraj B (310821104021)
Team Member 4 : Swarna Latha V (310821104097)

Supervisor : Mrs. D. Jeevitha M.E.,
Assistant Professor,
Department of CSE






INTRODUCTION


Problem Statement

With the growing demand for online education, many platforms struggle to provide an efficient and user-friendly experience. Existing systems lack seamless course management, personalized user access, and effective tools, leading to difficulties in managing courses, enrolling students, and ensuring smooth interactions.

Proposed Solution



Coursara, the Online Learning Platform, addresses these limitations by utilizing the MERN stack (MongoDB, Express, React, Node.js) for an efficient, secure, and scalable solution. The platform enhances the learning experience with a seamless, intuitive interface, ensuring ease of management for administrators and a robust system for all users.



PROJECT OVERVIEW

Purpose:

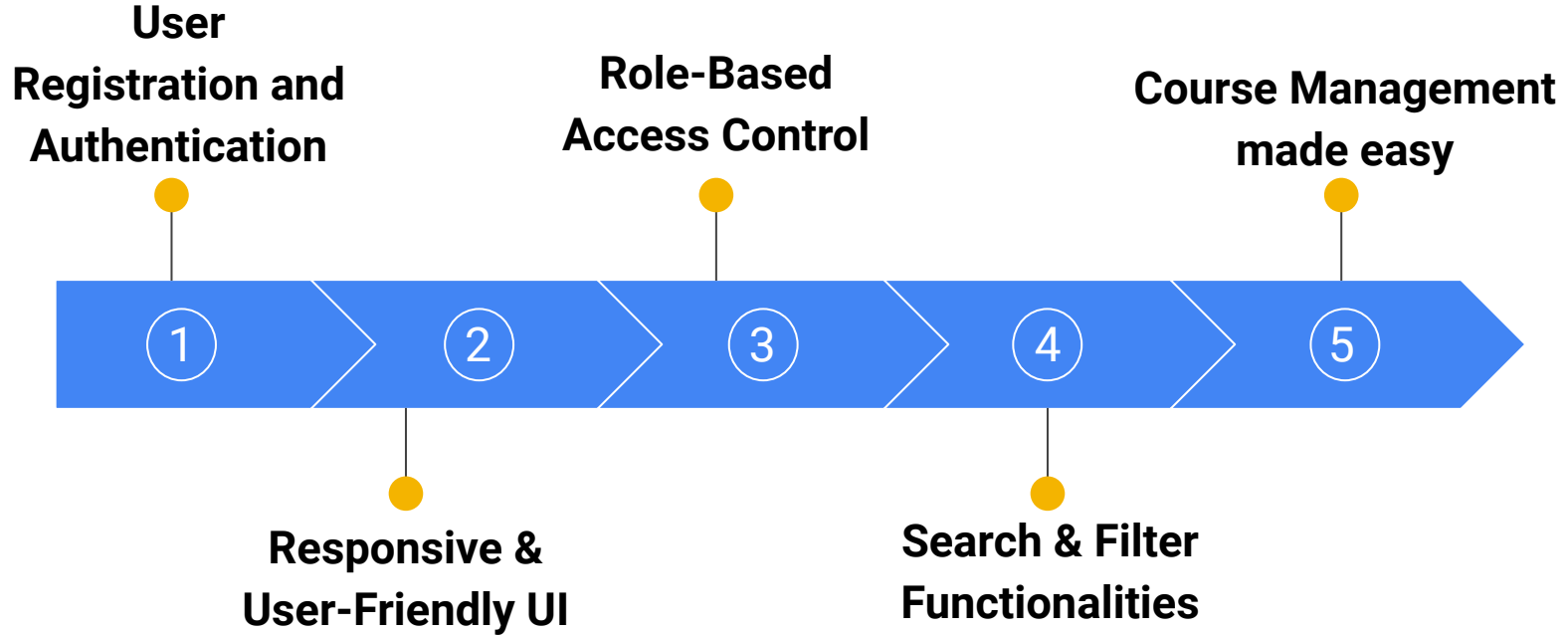
- ❑ Flexible & accessible platform to enhance skills.
- ❑ Teachers share expertise.
- ❑ Smooth and efficient operations.
- ❑ Makes education flexible, interactive.

Goals:

- ❑ Simplify managing and accessing online courses.
- ❑ Robust user management system for Admins.
- ❑ Seamless course creation and enrollment.
- ❑ User-friendly and interactive learning.

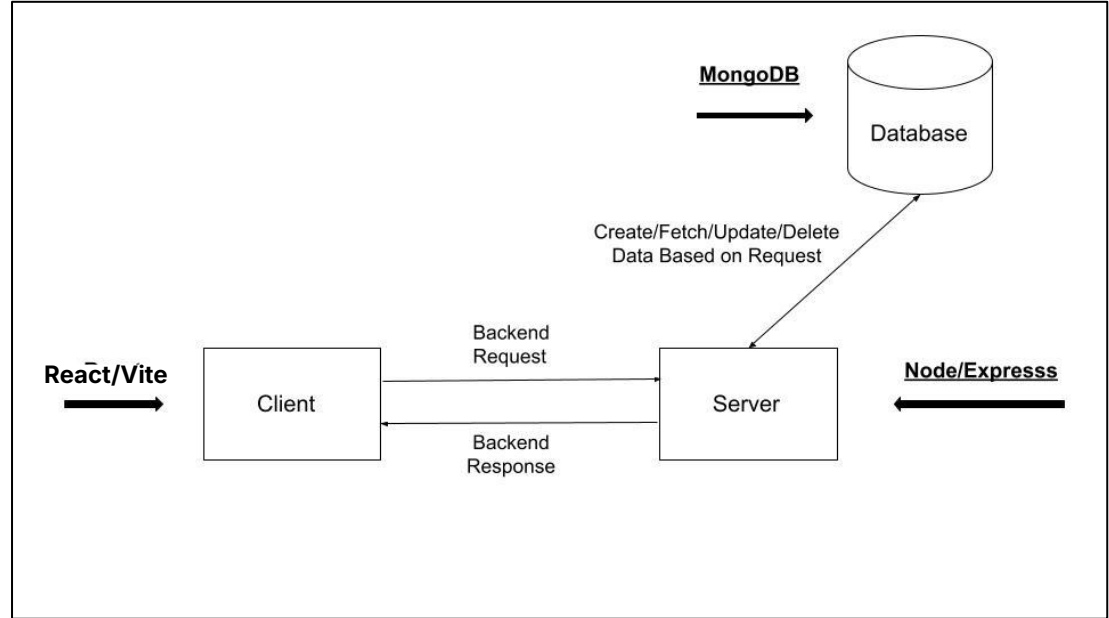
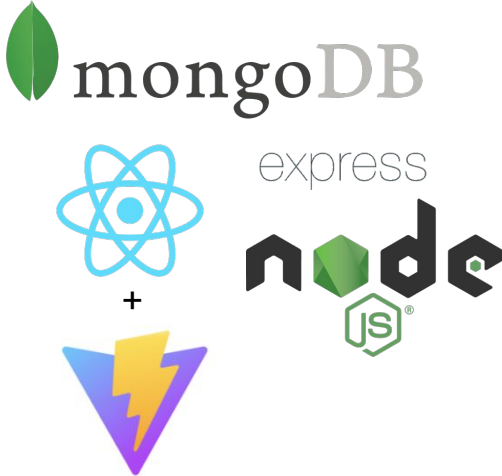


FEATURES



ARCHITECTURE

Tech Stack



API Testing

TESTING

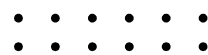
Role	HTTP Method	Endpoint
Student	POST	<code>/api/user/register</code>
Student	POST	<code>/api/user/login</code>
Student	GET	<code>/api/user/getallcourses</code>
Teacher	POST	<code>/api/user/addcourse</code>
Teacher	DELETE	<code>/api/user/deletecourse/:id</code>
Admin	GET	<code>/api/admin/getallusers</code>
Admin	DELETE	<code>/api/admin/deleteuser/:id</code>
Admin	GET	<code>/api/admin/getallcourses</code>

TESTING (contd..)

AUTHENTICATION

Feature	Description
User Authentication Flow	Registration with hashed passwords and JWT-based login sessions.
Token-Based Authentication	JWT generated on login; validated for secure API access.
Role-Based Access Control	Admin: Manage users and courses Teacher: Create/manage courses Student: Enroll/access courses





TESTING(contd..)



UI Testing:

Conducted cross-browser testing to ensure consistent functionality and layout across browsers like Chrome, Edge.

Error Handling:

Verified proper responses for invalid actions, such as 403 for unauthorized access, 404 for missing resources, and 500 for server errors.



Course Management:

Validated course enrollment, tracking, and certification generation processes for students.

Tested course creation, and publishing features for teachers.



KNOWN ISSUES



- ❑ Courses Cannot Be Edited Once Created
- ❑ No Support for Live Classes
- ❑ Lack of a Review System
- ❑ No User Analytics and Tracking
- ❑ Inadequate Notification System

FUTURE ENHANCEMENTS

- ❑ Offline Access to Courses
- ❑ AI-Powered Recommendations
- ❑ Peer-to-Peer Learning Communities
- ❑ Mobile Application
- ❑ Advanced Search and Filtering
- ❑ Integrate Payment Gateway for Paid Courses



PROJECT SETUP

Create Project Folder

Frontend
Backend



Backend Setup

Initialize Node.js project
Install necessary Libraries



Frontend Setup

Install React and all necessary
libraries



API and env variables

Setup API routes
Add all env variables in .env file

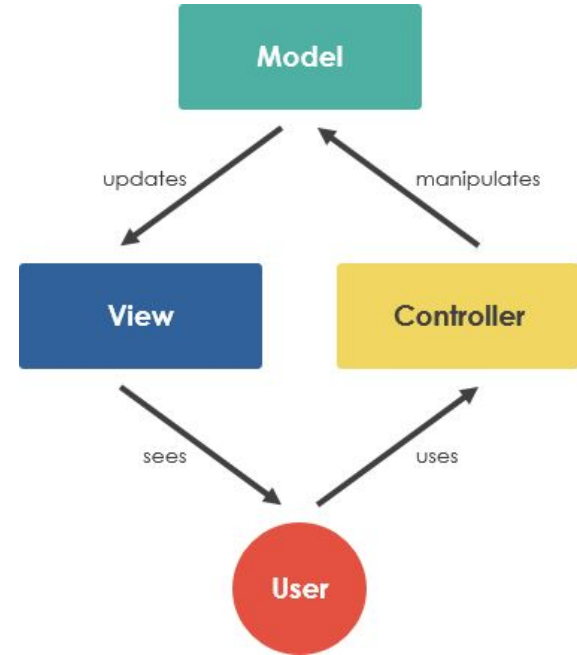


FOLDER STRUCTURE

BACKEND

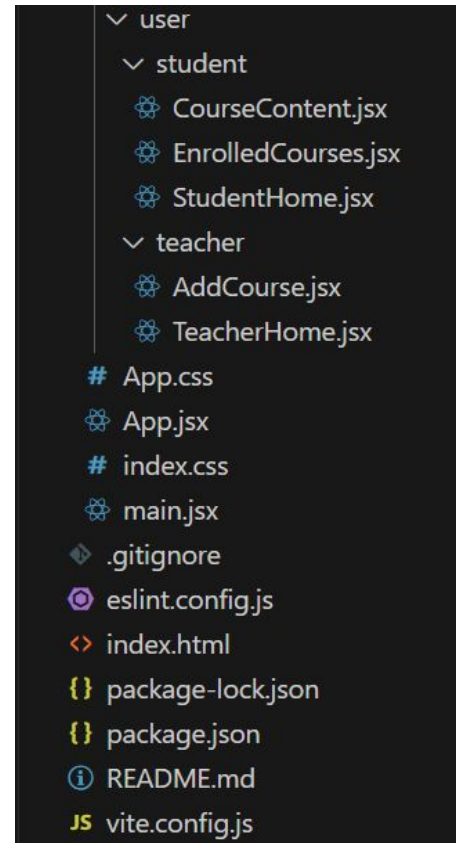
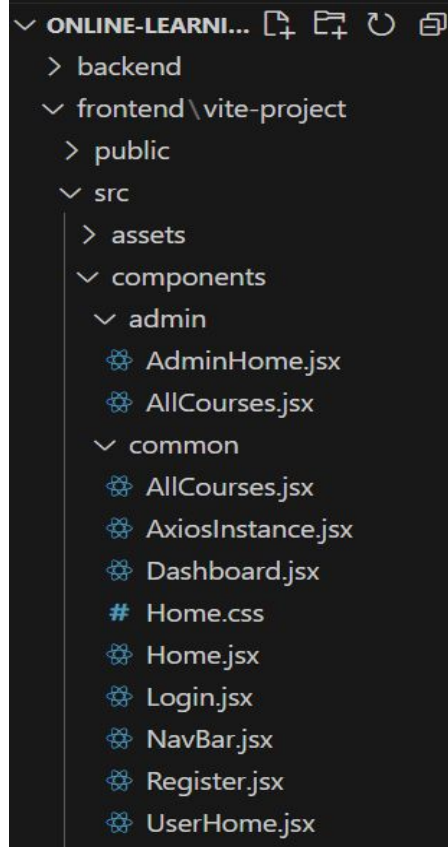
```
▼ ONLINE-LEARNI... [Icons]
  ▼ backend
    ▼ config
      JS connect.js
    ▼ controllers
      JS adminController.js
      JS userControllers.js
    ▼ middlewares
      JS authMiddleware.js
    ▼ routers
      JS adminRoutes.js
      JS userRoutes.js
    ▼ schemas
      JS courseModel.js
      JS coursePaymentModel.js
      JS enrolledCourseModel.js
      JS userModel.js
    > uploads
    .gitignore
    JS index.js
    {} package-lock.json
    {} package.json
```

MVC



FOLDER STRUCTURE(contd..)

FRONTEND



⋮

RUNNING THE APPLICATION

npm run dev

✓ TERMINAL

```
^CTerminate batch job (Y/N)? y
PS C:\Projects\LearningPla\backend> npm run dev

> backend@1.0.0 dev
> nodemon index.js

[nodemon] 3.1.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
(node:19664) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option
for version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:19664) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated
MongoDB Driver version 4.0.0 and will be removed in the next major version
running on 5000
Connected to MongoDB
█
```

```
PS C:\Users\swarn\OneDrive\Desktop\OLP\frontend\vite-project> npm run dev

> vite-project@0.0.0 dev
> vite

VITE v5.4.10 ready in 908 ms

→ Local:    http://localhost:5173/
→ Network:  use --host to expose
→ press h + enter to show help
```



PROJECT

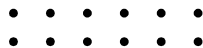
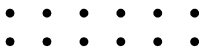
DEMONSTRATION



ATION

N





THANK YOU

