KESHAV MEMORIAL ENGINEERING COLLEGE

A unit of Keshav Memorial Technical Educational Society (KMTES)
(Approved by AICTE, New Delhi & Affiliated to Osmania University, Hyderabad)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(CSE – C)

Mini Project Presentation on

BYTECLASS

by

Bolla Divya Sai Mounika – 245522733138
Manaswini K – 245522733172

Guide:
Mrs. SWAPNA VANGURU

(Assistant Professor)

Index

Abstract

- Byte Class is a comprehensive Learning Management System (LMS) designed to support both learners and academicians through a modern, responsive, and user-friendly web platform. It falls under the domain of Educational Technology and Web Technology, aiming to enhance online education through efficient and accessible tools. The platform enables educators to create and monetize their courses, while students can browse, enroll, and learn through video-based lessons. Secure user authentication and integration of payment to ensure protected access for all users. Some of the Key features include real-time progress tracking, performance analytics dashboards, educators analytics for tracking revenue and student engagement. Byte Class offers a seamless experience across devices such as mobiles, tablets, and desktops. It brings together essential educational features in a cohesive system that promotes effective digital learning.

- **Keywords:** Ed-Tech platform, MERN Stack, RESTful APIs, User Authentication, Payment Integration.

1. Introduction

ByteClass – Learning Management System (LMS)

ByteClass is a full-stack Learning Management System web application built using the **MERN stack** (MongoDB, Express.js, React.js, Node.js). It offers a responsive, intuitive platform for online teaching and learning.

Key Features:

- Course creation with video lessons
- Real-time progress tracking and analytics dashboards
- **Role-Based Access Control (RBAC):**
- **Students** can enroll, learn, and participate in discussions
- **Educators** can create, publish, and monetize courses

•Secure **JWT Authentication**

•Integrated **Payment Gateway** for course purchases

•**RESTful API** architecture ensuring modularity and scalability

Scope:

Enables digital learning through robust course hosting, learner engagement, and educator revenue tracking. Scalable for use in academic institutions, training centers, or independent educators.

2. Literature Survey

“Navigating the E-Learning Platform with MongoDB, Express, React, and Node” (Kulkarni et al., Mar 2024)
Analyzes UX/UI, modular content delivery, and backend setup in a React-based LMS prototype. Explores challenges in handling multimedia learning content efficiently.
“Front-End Learning Management Design System with Agile & React JS” (Sep 2024)
A case study (Alterra Academy) detailing implementation of React in LMS front-end using Agile methodology. Covers UI modules like login/logout, class management, notifications, admin/instructor dashboards
“**Revolutionizing EdTech: Building the Future of Learning with the MERN Stack”** (May 2025)
Presents a MERN-based LMS/blog hybrid with React on the front-end, focusing on responsive, modular, and scalable UI. Highlights seamless CRUD workflows and dynamic component integration
“Learning Management System”
JWT-based authentication, role-based dashboards, course content delivery (text, video, downloads), quizzes/assignments with auto-grading, and real-time notifications—all managed with Redux Toolkit and React Router.

3. Problem Statement

Gaps in Existing LMS Platforms:

- Lack of detailed role segregation (limited distinction between students and educators)
- No integrated payment or revenue tracking for educators
- Static course structures without dynamic content organization
- Limited real-time progress tracking and learner analytics
- Poor mobile responsiveness and inconsistent cross-device experience

Proposed Solution – ByteClass LMS:

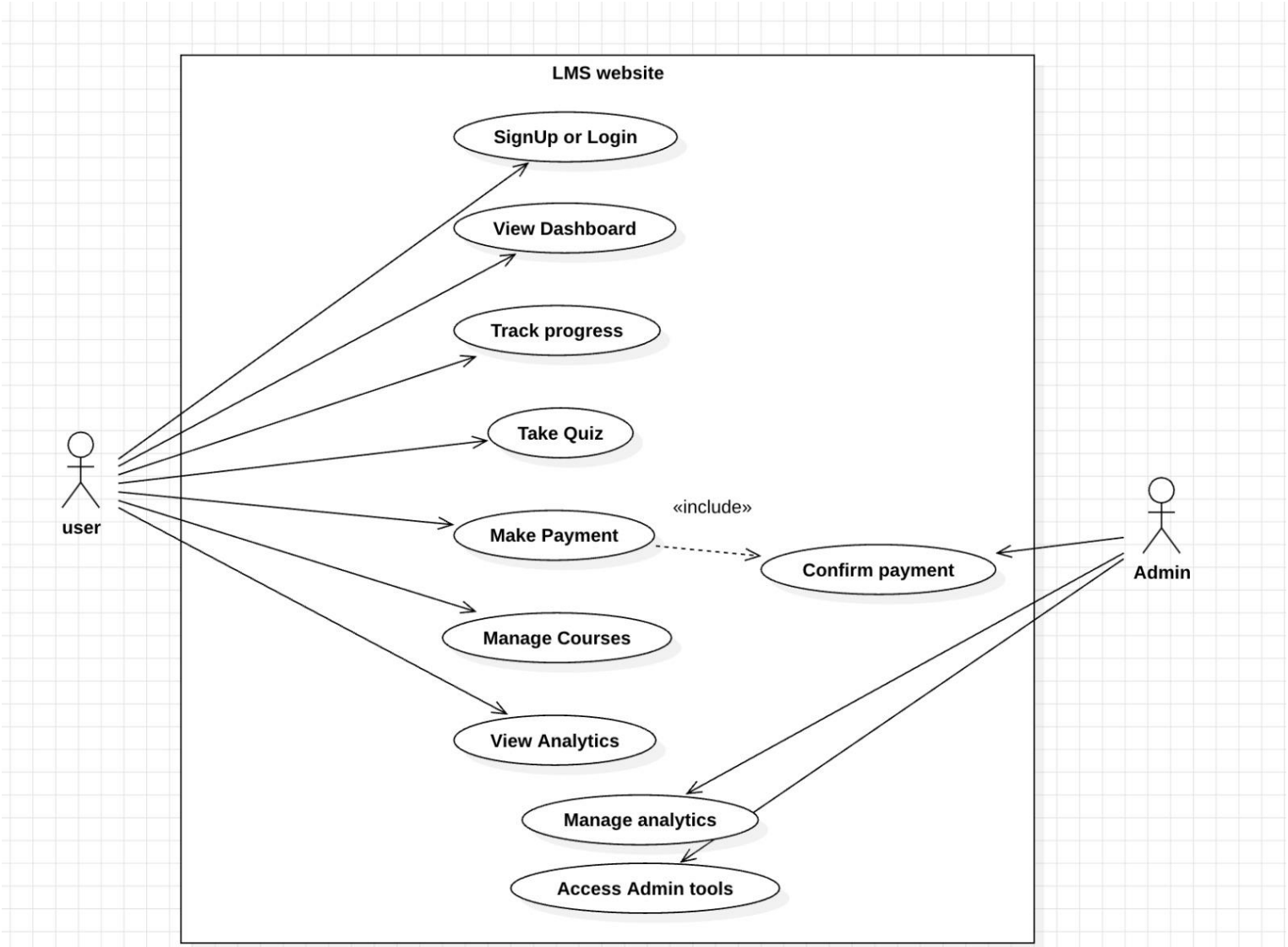
- Clearly defined Role-Based Access Control (RBAC) for students, educators, and admins
- Integrated payment gateway for monetizing courses and tracking educator revenue
- Flexible course management with dynamic modules and downloadable resources
- Real-time progress tracking and personalized performance dashboards
- Responsive design for seamless access on mobile, tablet, and desktop

4. Methodology

Client-Server Separation: The React.js frontend communicates with the Express.js backend via RESTful APIs. The backend is built with Node.js and connects to MongoDB to manage data storage and business logic.
RESTful API Design: Core functionalities are exposed through RESTful APIs for user management, authentication, progress tracking, discussions, and payments.
Schema Modeling: Mongoose models define the structure for application data, including Users, Courses, Lessons, Quizzes, Progress Reports, and Payments
Middleware Usage: The backend uses middleware for JSON parsing (express.json()), CORS handling (cors()), JWT-based authentication, role-based access control
Authentication & Authorization: A secure JWT authentication system is implemented, providing login, signup, and protected route access. Role-Based Access Control (RBAC) is enforced to distinguish between Students, Educators, and Admins.
Payment Integration: The system includes secure payment gateway integration (e.g., Razorpay or Stripe) to enable educators to monetize courses. Payment data is validated and stored with transaction records in MongoDB.
Frontend Integration: React.js components are managed using useState, useEffect and routing
Analytics & Dashboards: Real-time learner analytics and educator dashboards are powered by API-driven data aggregation. Educators can track student engagement and earnings, while students can view progress and quiz performance using visual charts.

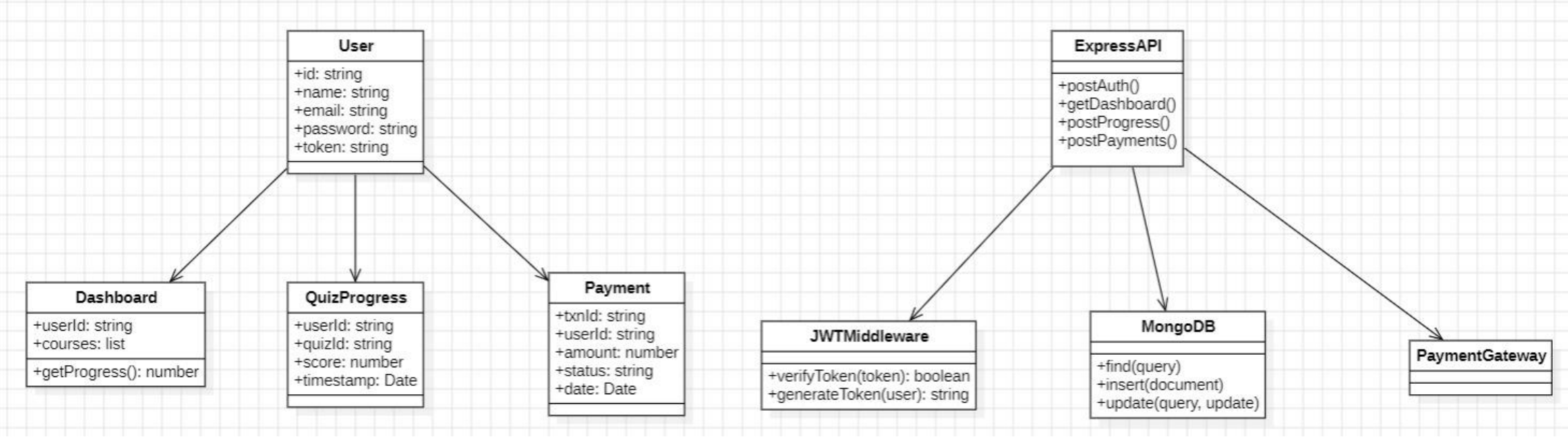
5. Designing

Use case diagram:



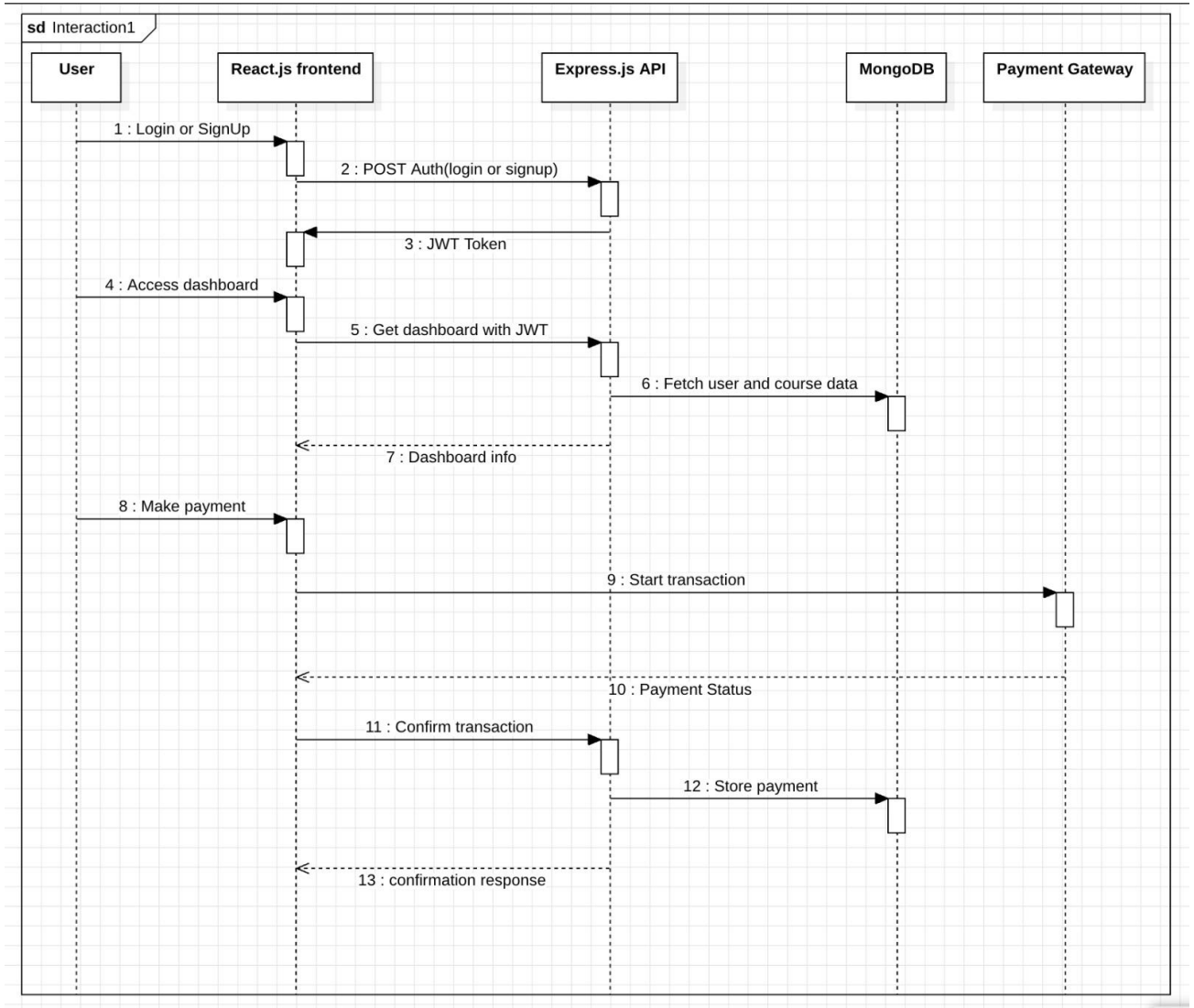
5. Designing

Class diagram:



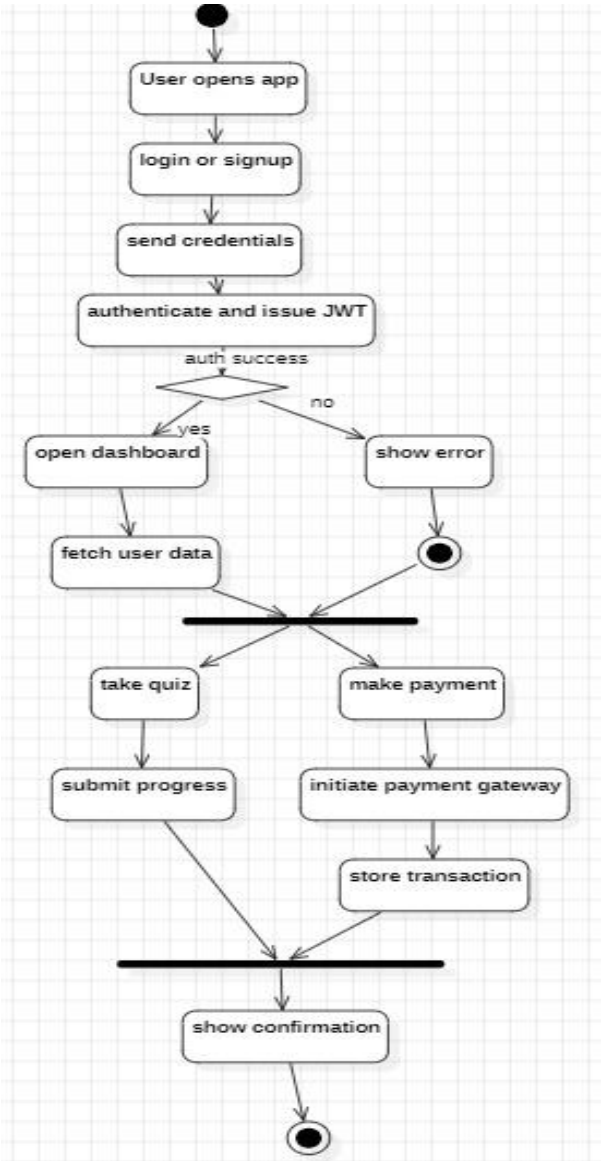
5. Designing(contd.)

Sequence Diagram:



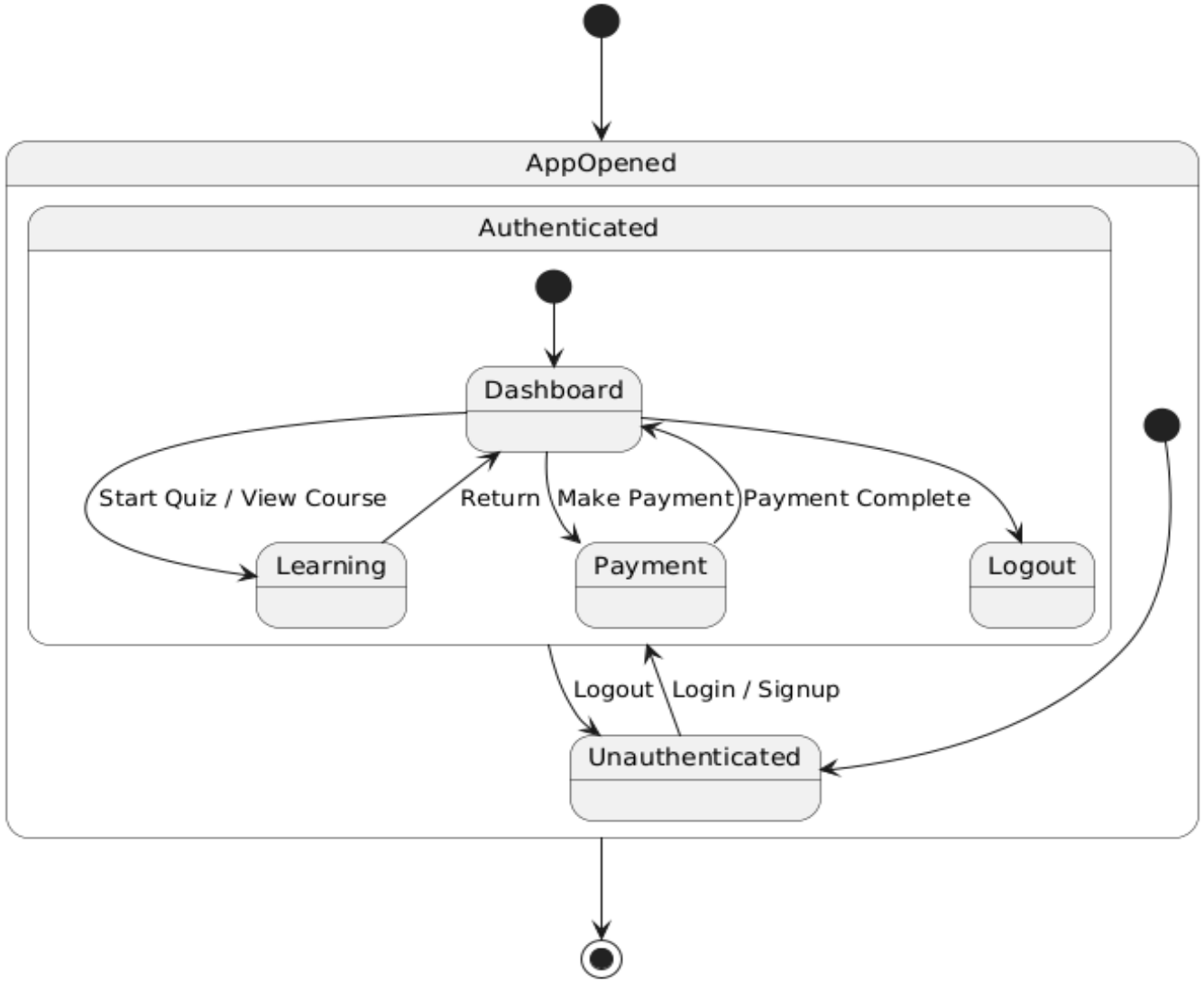
5. Designing(contd.)

Activity Diagram:



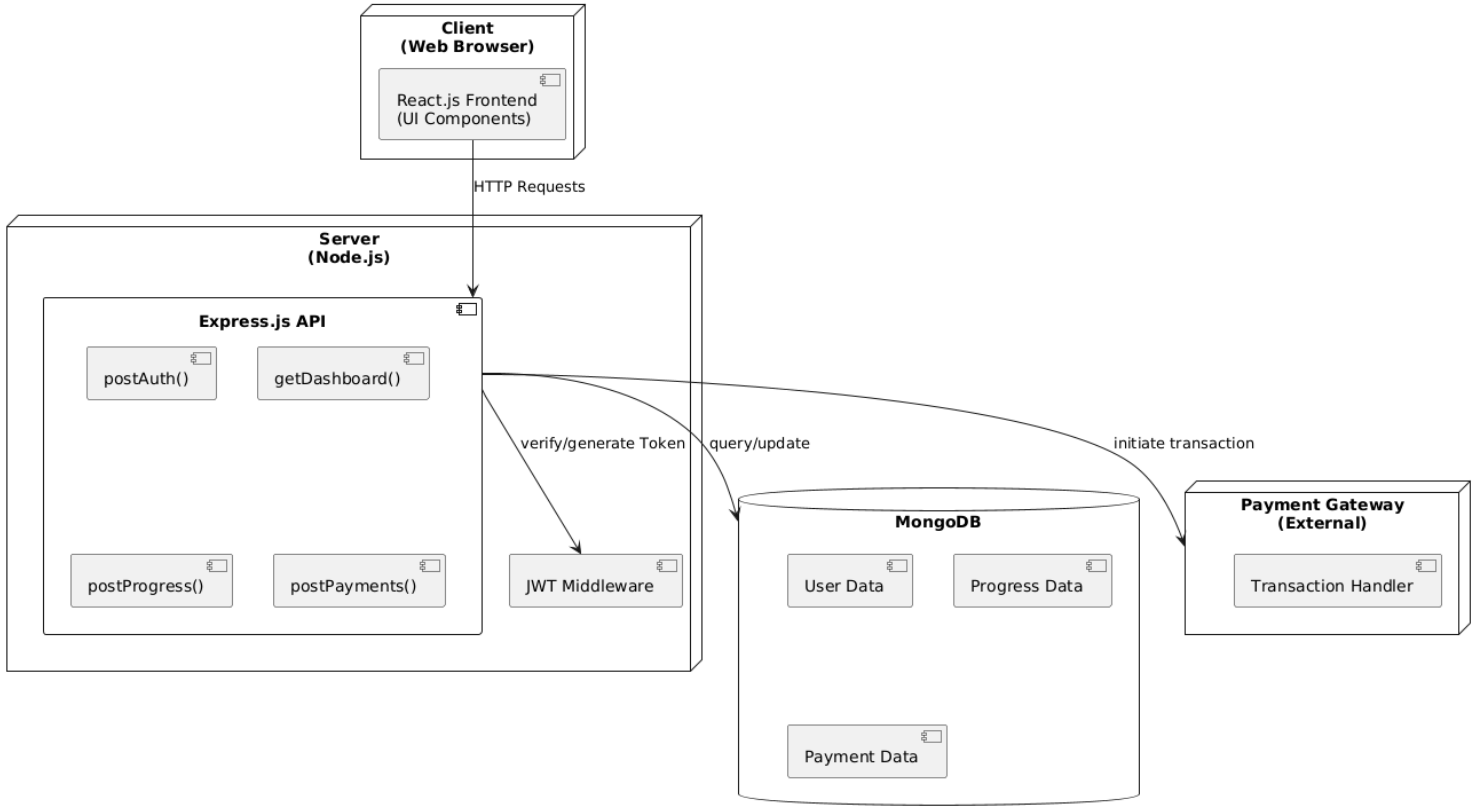
5. Designing(contd.)

State chart Diagram:



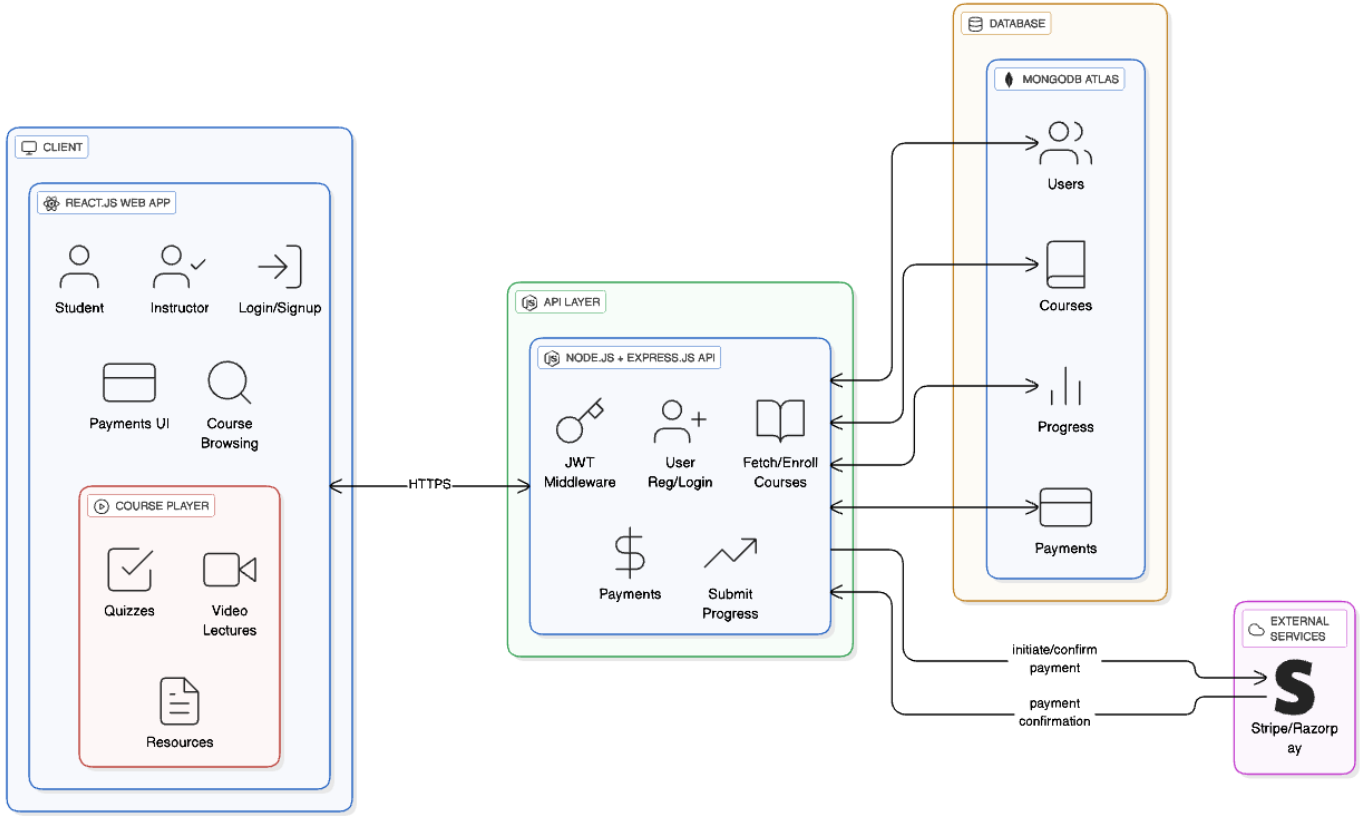
5. Designing(contd.)

Deployment Diagram:



5. Designing(contd.)

Architecture Diagram·



6. Software Requirements

Frontend Technologies: React.js, Axios, HTML5, CSS3

Backend Technologies: Node.js with Express.js

Database: MongoDB (NoSQL)

Authentication: JSON Web Token (JWT)

Image Upload Service: Cloudinary

Version Control: Git and GitHub

Code Editor (IDE): Visual Studio Code

Supported Browsers: Google Chrome, Mozilla Firefox, Microsoft Edgearchitecture

Operating System Compatibility: Windows / Linux / macOS

Internet Requirement: Required for hosting, media uploads, and API access

Hardware Requirements

Hardware Requirements for ByteClass LMS

Processor: Intel Core i5 (8th Gen or newer) or AMD Ryzen 5 (recommended for smooth development)
RAM: Minimum 8 GB (required for running backend server, database, browser, and code editor simultaneously)
Storage:
Project Files & Local DB: Minimum 1 GB free
Display: Minimum 14” screen with 1366×768 resolution (1080p recommended for comfortable UI/UX testing)
Graphics: Integrated graphics sufficient (no GPU required)
Internet Connection: Stable broadband (required for using APIs, uploading media to Cloudinary, and deploying on Vercel/Heroku)
Peripherals: Keyboard, mouse, webcam

7.Implementation

Project Setup

- The ByteClass LMS is implemented using the **MERN stack**: MongoDB, Express.js, React.js, and Node.js for full-stack JavaScript development.
- Integrated **Clerk** for user authentication, role-based access, and session handling.
- Included media management with **Cloudinary**, payment integration via **Stripe**, and file uploads using **Multer**.

Database Design

- **MongoDB Atlas** used for scalable, cloud-hosted NoSQL storage with high availability.
- **Mongoose** defines schemas for key entities like Courses, Users, and Enrollments to ensure structured data.
- Collections are role-specific and include course metadata, user roles, and enrollment relationships.

API Development

- RESTful APIs are built using **Express.js** (or **Next.js API routes**) to handle server-side logic and data processing.
- These APIs manage core operations such as course creation, user enrollment, and data retrieval.
- **JWT tokens** and **Clerk middleware** are used to ensure secure access and route protection.
- All data is exchanged in **JSON format**, allowing seamless communication with the frontend.
- APIs are optimized for scalability, enabling fast response times and easy integration with React via Axios.

7.Implementation(contd.)

Frontend Development

•Built using **React.js** with **Tailwind CSS** for responsive design and utility-based styling.
•Integrated **Clerk** for handling sign-up, login, session persistence, and role enforcement.
•Navigation managed via **React Router**, with **Protected Routes** for authenticated users only.

Backend Development

•Developed using **Node.js + Express.js** or **Next.js API routes** to handle business logic and data operations.
•Integrated **Multer** for file uploads and **Cloudinary** to manage and serve media content.
•**Stripe** is used for secure payment processing, educator payouts, and checkout flows.

7.Implementation(contd.)

Backend Integration

•APIs consumed in the frontend using **Axios** for fetching and manipulating dynamic course data.
•Clerk tokens are validated across client and server to maintain secure sessions.
•Access control ensures that educators manage only their courses, while learners access enrolled content.

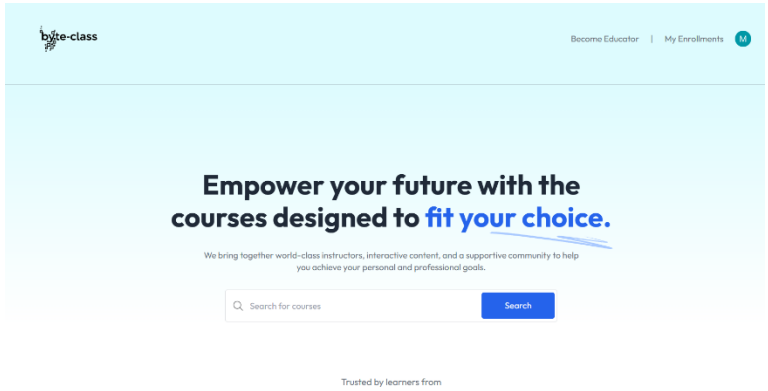
Authentication & Authorization

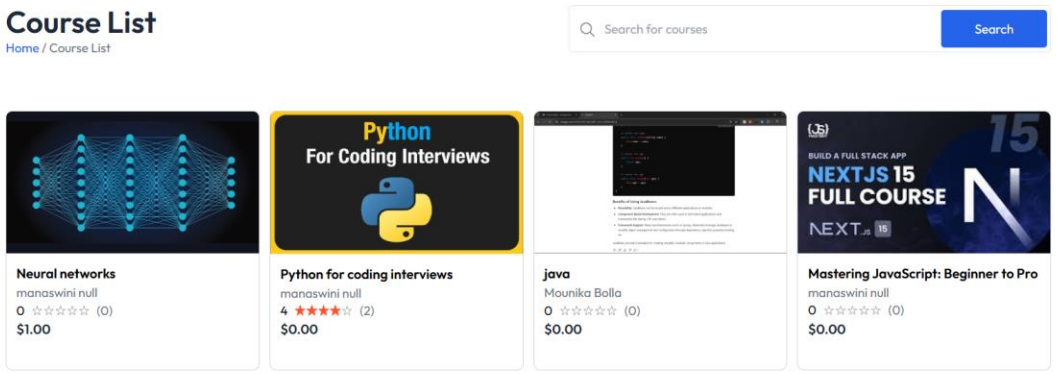
•Implemented using **Clerk**, which supports Google/email login, session management, and role-based access.
•Only authenticated users can access restricted routes like the dashboard or course management.
•Role checks enforce educator and learner-specific permissions across the platform.

Deployment

•**Frontend** is deployed using **Vercel**, with automatic builds and global CDN delivery.
•**Backend** is hosted on **Render** or **Vercel Functions**, depending on the API framework used.
•**MongoDB Atlas** serves as the managed cloud database for reliability and scalability.

9.Result


byte-class
Become Educator | My Enrollments
Empower your future with the courses designed to fit your choice.
We bring together world-class instructors, interactive content, and a supportive community to help you achieve your personal and professional goals.
Search for courses
Search
Trusted by learners from


Course List
Home / Course List
Search for courses
Search
Neural networks
manaswini null
0 (0)
$1.00
Python
For Coding Interviews
Python for coding interviews
manaswini null
4 (2)
$0.00
java
Mounika Bolla
0 (0)
$0.00
BUILD A FULL STACK APP
NEXTJS 15
FULL COURSE
Mastering JavaScript: Beginner to Pro
manaswini null
0 (0)
$0.00

9.Conclusion & Future Scope

Conclusion:

ByteClass is a full-stack Learning Management System (LMS) built using the MERN stack.

It provides a seamless platform for educators to create and monetize courses, while learners can enroll, track progress, and interact with instructors.

The system features secure JWT-based authentication, role-based access control (Student, Educator, Admin), integrated payments, and media support via Cloudinary.

Analytics dashboards help track student performance and educator earnings.

ByteClass delivers a responsive, scalable, and user-friendly learning experience across devices.

9.Conclusion & Future scope(contd.)

Future Scope:

•Real-time chat between learners and instructors

•AI-based quiz generation and adaptive learning paths

•Mobile application for Android/iOS platforms

•Certificate generation on course completion

•Voice/video lecture support with recording tools

•Admin-level analytics on course performance and platform usage

•Multilingual content support for regional inclusivity

•Offline access for downloaded course materials

10. References

[1] Ayush Kulkarni, Himanshu Jain, and Vikas Sharma; Navigating the E-Learning Platform with MongoDB, Express, React, and Node; International Journal of Innovative Science and Research Technology (IJISRT), Vol. 9, Issue 3, March 2024, pp. 1375–1382.

[2] Ani Amaliyah and Intan Meilanie Nugraha; Front-End Learning Management Design System with Agile & React JS (Case Study: PT Marka Kreasi Persada – Alterra Academy); JURETI e-Journal, September 2024, pp. 18–30.

[3] Raghavendra Patil G E; Revolutionizing EdTech: Building the Future of Learning with the MERN Stack; ResearchGate Preprint, May 2025, pp. 1–11.

[4] Unnamed Authors; Learning Management System (MERN stack, JWT-based auth, role-based dashboards, content delivery, quizzes/assignments, auto-grading, real-time notifications); Journal of Emerging Technologies and Innovative Research (JETIR), April 2025, Paper ID: JETIR2504043, pp. 67–75.

[5] Nisam A., Sheik S., and Murugan P.; Learn-It: An E-Learning Web Application Using MERN Stack; International Journal of Fundamental and Multidisciplinary Research (IJFMR), Vol. 6, Issue 1, January 2024, pp. 1132–1138.

[6] Jadhao A. A., Patil M. S., and Gawande M. V.; A MERN Stack-Based Learning Management System: Development and Evaluation; International Research Journal of Modernization in Engineering Technology and Science (IRJMETS), Vol. 6, Issue 4, April 2025, pp. 283–290.

THANK YOU