# ONLINE LEARNING PLATFORM USING MERN STACK

## Project Report

*Submitted by*

Brein Austin J - 513421104004 Abishek Sam R - 513421104703 Selvaraj S - 513421104038 Dhanush K - 513421104007

# BACHELOR OF ENGINEERING

*in*

# COMPUTER SCIENCE AND ENGINEERING

**UNIVERSITY COLLEGE OF ENGINEERING KANCHIPURAM**

**ONLINE LEARNING PLATFORM USING MERN STACK**

# ABSTRACT

ACQUEL is an innovative online learning platform developed using the MERN (MongoDB, Express.js, React.js, Node.js) stack. The platform bridges the gap between educators and students by providing comprehensive digital learning solutions. Key features include course management, video lecture streaming, interactive assessments, real-time discussion forums, and automated certificate generation. The system employs JWT-based authentication for security and offers robust analytics for tracking student progress. Built with a responsive design, the platform supports multiple user roles and delivers a seamless learning experience across devices. Future enhancements will include AI-driven recommendations and mobile application development.

# INTRODUCTION

Acquel is a state-of-the-art online learning platform developed by a cross-functional team of experts in web development and educational technology.

The platform aims to revolutionize digital education by providing an intuitive, feature-rich environment for both educators and students. Built on the MERN stack (MongoDB, Express.js, React.js, Node.js), EduConnect offers comprehensive course management, interactive learning tools, and real- time collaboration features. The system emphasizes user experience,

accessibility, and scalability to meet the growing demands of modern online education.

# PURPOSE

Acquel is a comprehensive online learning platform designed to bridge the gap between educators and students. The platform facilitates remote learning through interactive courses, real-time assessments, and collaborative tools.

# KEY FEATURES

* Course creation and management
* Video lecture streaming
* Interactive quizzes and assignments
* Real-time discussion forums
* Progress tracking and analytics
* Virtual classroom sessions
* Certificate generation upon course completion

# ARCHITECTURE

## Frontend:

* Built using React.js with functional components and hooks
* State management using Redux for global state
* Material-UI for consistent design components
* React Router for client-side routing
* Axios for API integration

## Backend:

* Node.js with Express.js framework
* RESTful API architecture
* JWT-based authentication
* WebSocket integration for real-time features
* Middleware for request validation and authentication

## Database:

// MongoDB Schema Examples

// User Schema

{

username: String, email: String, password: String,

role: ['student', 'instructor', 'admin'], enrolledCourses: [CourseID], createdAt: Date

}

// Course Schema

{

title: String, description: String,

instructor: UserID, lessons: [LessonID], category: String, enrolledStudents: [UserID]

}

// Lesson Schema

{

title: String, content: String, videoUrl: String, duration: Number, courseId: CourseID,

}

# SET UP INSTRUCTION

## Prerequisites:

* Node.js (v14.0.0 or higher)
* MongoDB (v4.4 or higher)
* npm or yarn package manager

## Installation

# Clone the repository

git clone https://github.com/educonnect/learning-platform.git # Frontend setup

cd client npm install

cp .env.example .env # Backend setup

cd ../server npm install

cp .env.example .env

# Configure environment variables # .env file should include:

MONGODB\_URI=your\_mongodb\_uri JWT\_SECRET=your\_jwt\_secret PORT=5000

# FOLDER STRUCTURE

## Client:

├── public/

└── src/

├── assets/

├── components/

├── context/

├── pages/

│ ├── admin/

│ ├── instructor/

│ └── user/

├── App

├── Index

└── main

## Server:

├── config/

├── controller/

├── middlewares/

├── models/

├── node\_modules/

├── routes/

├── services/

├── utils/

├── index

├── package.json

└── package-lock.json

# RUNNING APPLICATION

# Start frontend (from client directory) npm run dev

# Runs on [http://localhost:3000](http://localhost:3000/)

# Start backend (from server directory) npm start

# Runs on [http://localhost:5000](http://localhost:5000/)

# API DOCUMENTATION

**Authentication endpoints** POST /api/auth/register POST /api/auth/login

GET /api/auth/profile

**Course Endpoints** GET /api/courses POST /api/courses GET /api/courses/:id PUT /api/courses/:id

DELETE /api/courses/:id

## Lesson Endpoints

GET /api/lessons/:courseId POST /api/lessons

PUT /api/lessons/:id DELETE /api/lessons/:id

# AUTHENTICATION

Acquel implements a secure JWT (JSON Web Token) based authentication system. Upon login, users receive an access token valid for 24 hours and a refresh token for extended sessions. The system employs role- based access control (RBAC) with three distinct user types: students, instructors, and administrators.

## Security features include:

* Password hashing using bcrypt
* HTTP-only cookies for token storage
* JWT verification middleware for protected routes
* Access control based on user roles
* Automatic token refresh mechanism
* Session management with MongoDB
* Rate limiting for login attempts
* CORS protection and XSS prevention

# TESTING

Acquel implements a comprehensive testing strategy across all application layers to ensure reliability and performance:

## Frontend Testing:

* Jest and React Testing Library for component testing
* Cypress for end-to-end testing
* User interface testing for responsiveness
* Snapshot testing for UI components

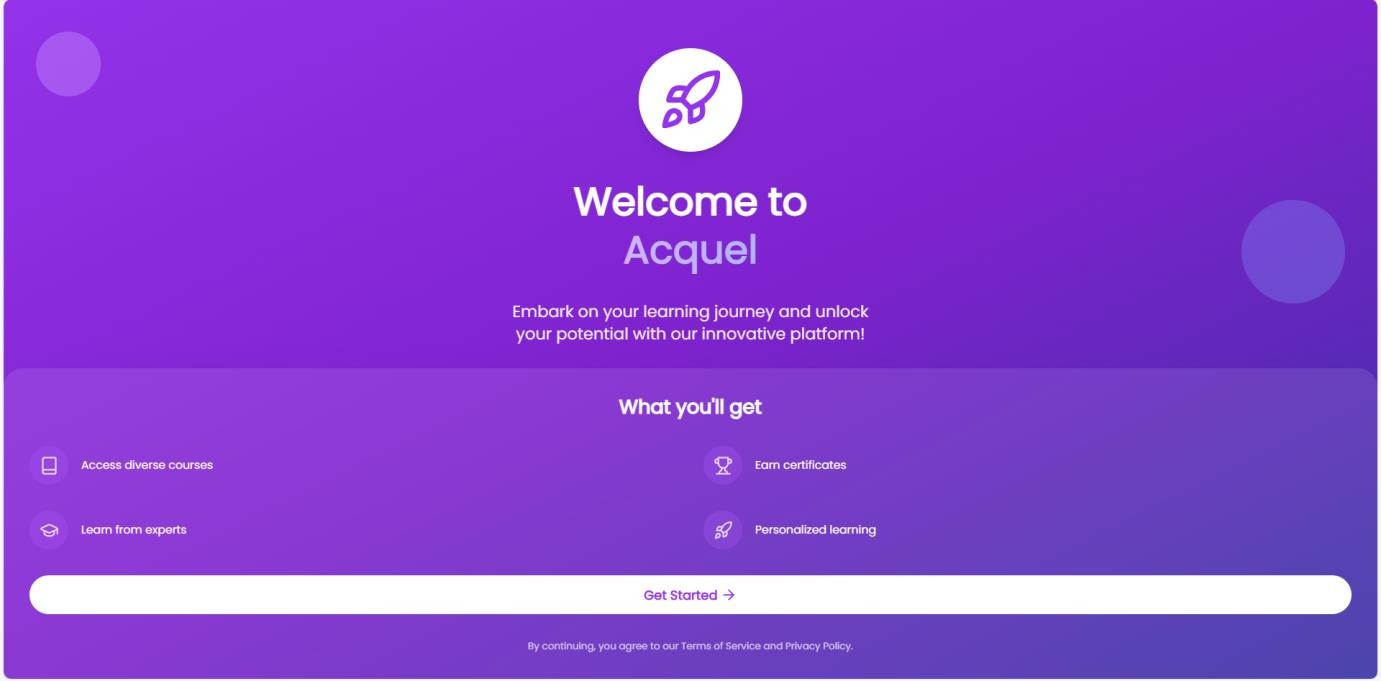
## Backend Testing:

* Mocha and Chai for API endpoint testing
* Unit tests for controllers and services
* Integration tests for database operations
* Load testing using Artillery

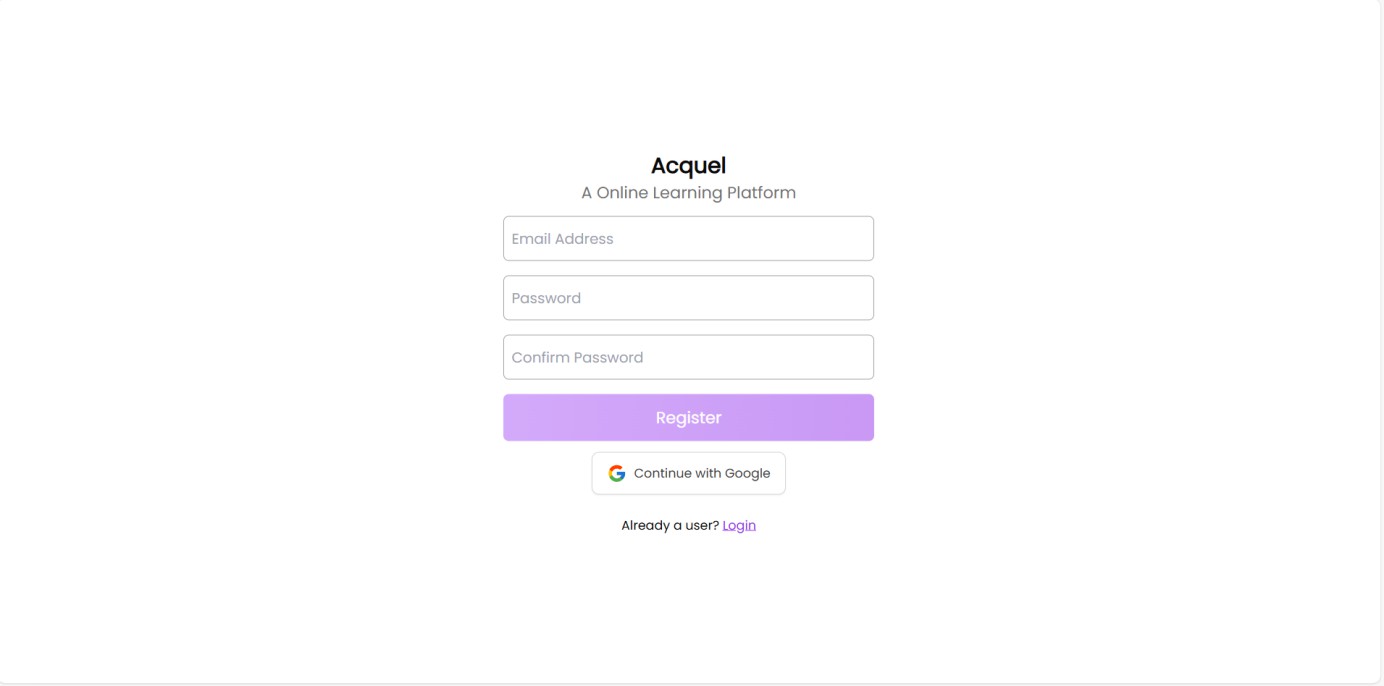
## Additional Testing:

* Postman for API documentation and testing
* Security testing with OWASP guidelines
* Performance testing for video streaming
* Cross-browser compatibility testing

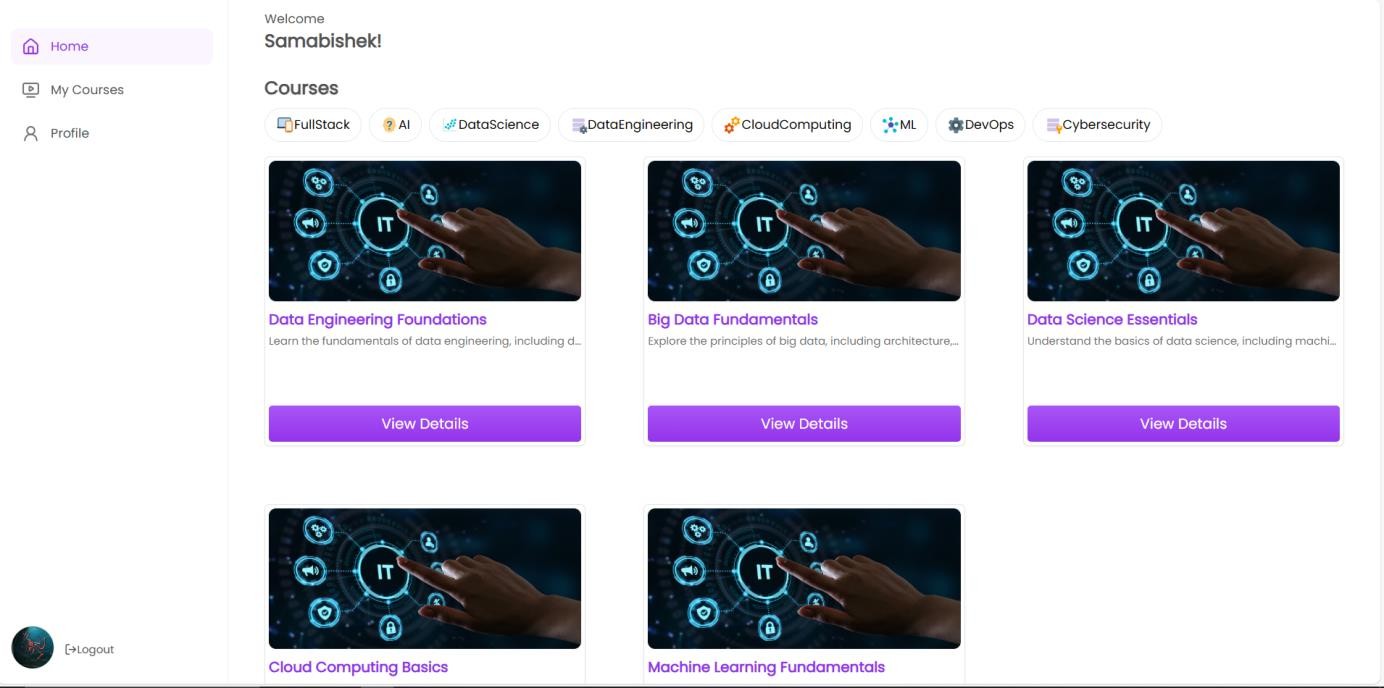
**SCREENSHOTS AND DEMO**

**Landing Page:**

**Login Page:**

****

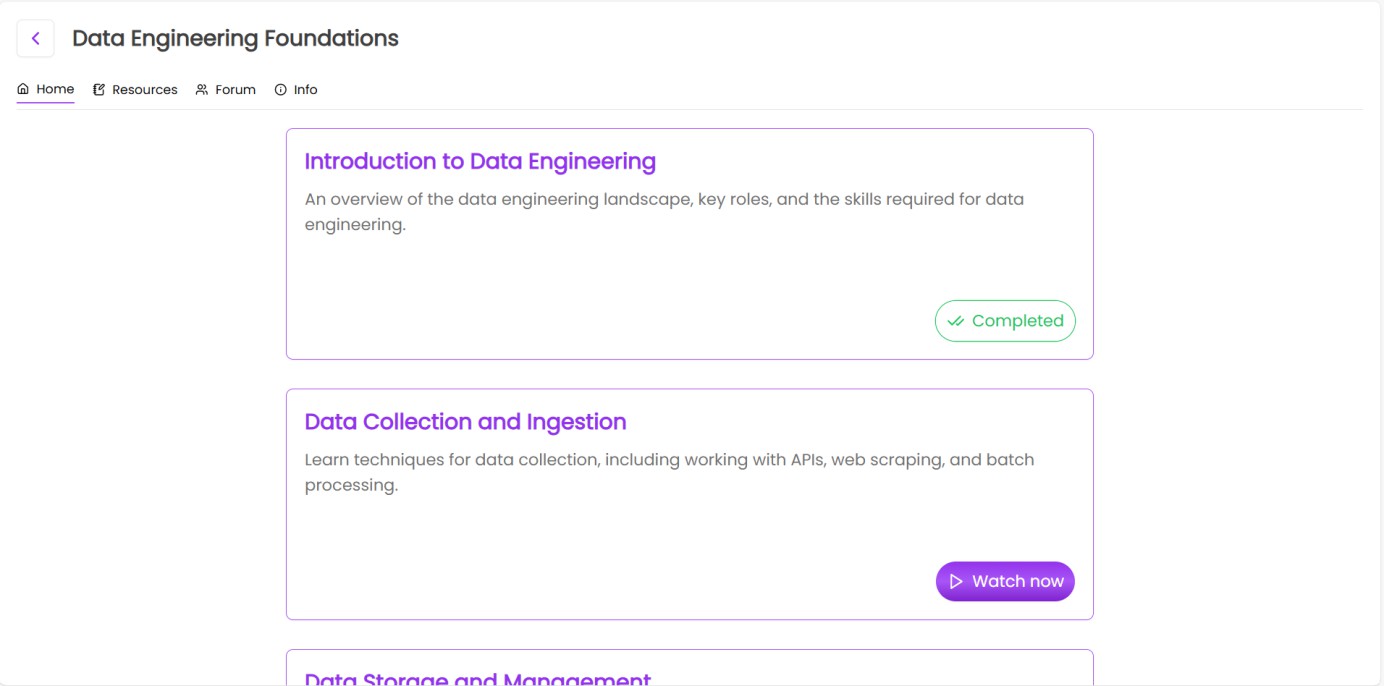
**Home Page:**

****

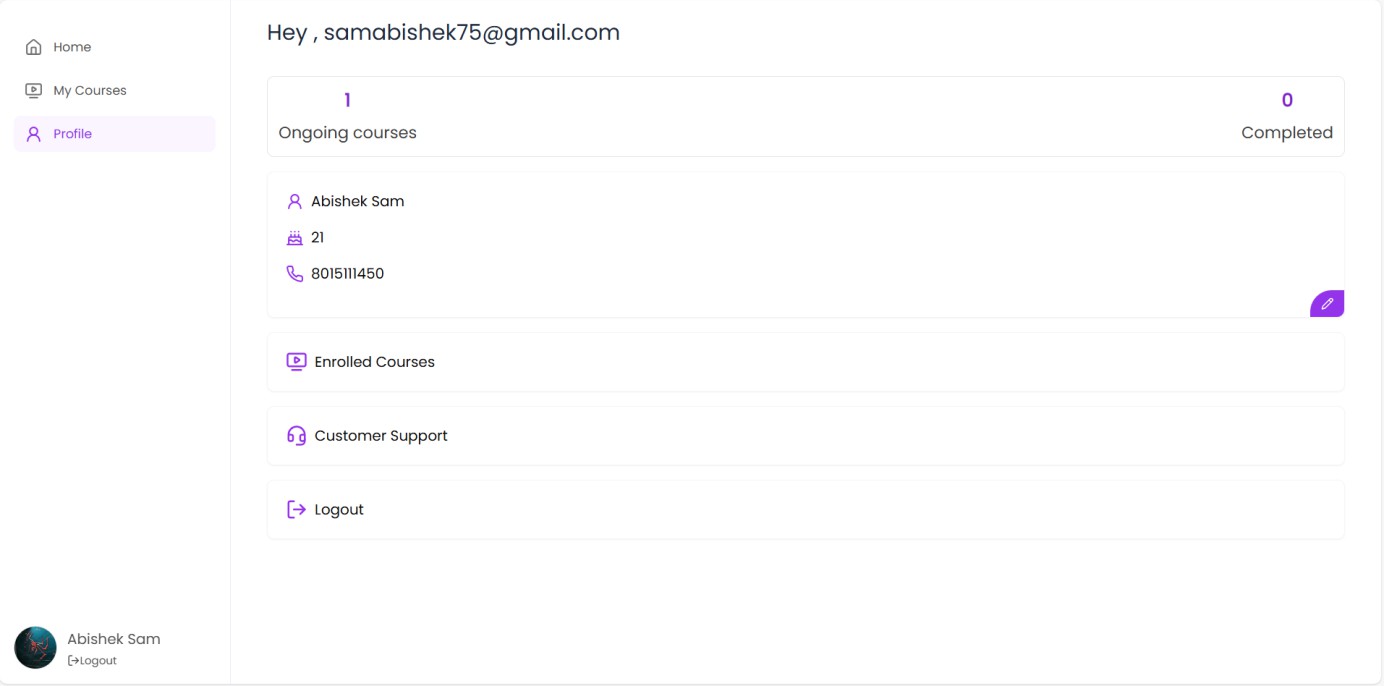
**Video Page:**

****

**Course Page:**

****

**User Profile Page:**

****

**Demo Video Link:**

A live demo of the Acquel website is available at:

<https://drive.google.com/file/d/11uao-dmEFO2-OhVpWpI-3_eyMhFgdT_C/view?usp=drivesdk>

# ISSUES AND ENHANCEMENT

The Acquel platform, while robust, has identified several technical challenges that impact user experience. Performance issues primarily affect video streaming and quiz functionalities, particularly during high-traffic periods. Mobile responsiveness requires optimization for certain features, and real-time communication tools show limitations with large user groups.

## Current Known Issues:

* Video playback buffers on slow internet connections (<10 Mbps)
* Mobile responsiveness issues on complex course layouts
* Chat feature experiences latency with >100 concurrent users

## Planned Future Enhancements:

* AI-powered course recommendations based on learning patterns
* Mobile application for iOS and Android platforms
* Integration with major payment gateways
* Live streaming capabilities for real-time lectures
* Peer-to-peer tutoring system
* Advanced analytics dashboard with predictive insights
* Offline mode for downloaded content
* Gamification features to increase engagement

# CONCLUSION

Acquel represents a significant advancement in online education technology, successfully implementing a comprehensive learning solution using the MERN stack. The platform effectively bridges the gap between educators and students through its robust feature set, including video streaming, interactive assessments, and real-time collaboration tools. Despite some technical challenges, the system demonstrates strong potential for scalability and improvement.

Looking forward, Acquel is positioned to evolve with planned enhancements in AI integration, mobile accessibility, and advanced analytics. The platform's architecture provides a solid foundation for future growth, ensuring its ability to meet the ever-changing demands of digital education while maintaining security, performance, and user satisfaction.