IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151120

StudyMate: A Multi-Agent Academic Assistance System

S. L. Tambe[1], Harshada Vinod Jagdale[2], Mrunal Madhukar Bharat[3], Jidnyasa Deepak Patil[4], Srushti Mahesh Patil[5]

Professor, Department of Artificial Intelligence & Machine Learning, R. C. Patel Institute of Technology, Shirpur, India[1]

Student, Department of Artificial Intelligence & Machine Learning, R. C. Patel Institute of Technology, Shirpur, India[2,3,4,5]

Abstract: The rapid expansion of academic content and digital learning resources requires centralized, secure, and intelligent platforms to assist engineering students in their daily studies and placement preparations. Traditional academic portals often suffer from fragmented file management, rigid access controls, lack of personalized branch-wise filtering, and absence of real-time AI assistance. This paper presents StudyMate: A Multi-Agent Academic Assistance System, a modern, web-based platform tailored exclusively for students and administrators of the R. C. Patel Institute of Technology (RCPIT). StudyMate consolidates all essential academic resources into a centralized, lightning-fast Single Page Application (SPA). The frontend is engineered using React.js, Vite, and Tailwind CSS, offering a modern, responsive user interface. The backend is powered by a robust Python Flask RESTful API interacting with a cloud-hosted MongoDB Atlas database. Security is enforced through JSON Web Tokens (JWT) for stateless session management and Werkzeug for cryptographic password hashing. Furthermore, StudyMate integrates advanced functional modules including a dynamic branch-and-semester resource filter, an interactive digital library, placement aptitude mock tests, a curated Final Year project repository, and a 24/7 AI-driven doubt-solving Hub. To ensure smooth institutional management, the platform features a strict Exclusive Admin Architecture incorporating credential tracking and real-time student account impersonation. Comprehensive testing confirms the stable, secure, and rapid execution of all client-server workflows.

Keywords: Academic Resource Platform, React.js, Python Flask, MongoDB Atlas, JSON Web Tokens (JWT), Role-Based Access Control, AI Doubt Solver.

I. INTRODUCTION

Artificial Intelligence, cloud-native web frameworks, and advanced database management systems are increasingly being used to develop intelligent educational ecosystems that can analyze student data, streamline resource management, and provide useful academic recommendations. Academic success and efficient exam preparation are important milestones for engineering students and job seekers, but many individuals face difficulty in accessing verified materials because of fragmented file storage, lack of personalized branch-wise filtering, and improper academic planning.

Traditional academic portals generally depend on static file downloads, manual administrator uploads, and general folder structures. Such approaches may require considerable time and may not consider the specific semester, branch, and individual profile requirements of an engineering student. A student's enrolled branch, current semester, technical book references, project interests, and placement aptitude scores can provide useful information for identifying tailored learning pathways and career opportunities.

StudyMate: A Multi-Agent Academic Assistance System is proposed to address these challenges. The system centralizes all essential academic resources using a modern decoupled client-server architecture. Important curriculum data, lecture notes, question papers, and project documentation are organized dynamically, while an integrated AI Hub provides round-the-clock conceptual doubt solving. The complete platform is designed as a web-based Single Page Application (SPA) so that users can access different academic guidance services through a single, highly responsive interface.

A. Problem Statement

Engineering students and academic administrators often face difficulty in locating verified, branch-specific academic resources because they may not know where official syllabi, notes, and previous year question papers are stored across scattered platforms. They may also lack a unified environment for technical placement preparation, digital library

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151120

browsing, and instant conceptual doubt solving. Traditional educational portals provide general file access and may require manual searching through disorganized folders. Therefore, there is a need for an intelligent system that can automatically authenticate users, filter resources by branch and semester, deliver aptitude mock tests, provide 24/7 AI-driven doubt solving, and maintain secure administrative oversight over student records and session states.

B. Objectives

The main objectives of the proposed system are:

1. To provide secure user authentication and stateless session management using JSON Web Tokens (JWT) and Werkzeug password hashing.
2. To dynamically filter academic resources including syllabi, lecture notes, and question papers according to the student's registered branch and semester.
3. To deliver comprehensive placement preparation modules including quantitative aptitude, logical reasoning, and verbal ability mock tests with instant score tracking.
4. To curate an interactive digital library of books and final-year project ideas complete with architectural summaries and technology stacks.
5. To provide a simple, highly responsive, and interactive Single Page Application (SPA) web interface using React.js and Tailwind CSS.
6. To support students and institutional faculty members in improving academic productivity and placement readiness.

C. Applications

The system can be used in educational institutions, department libraries, Training and Placement (T&P) cells, student study circles, campus coaching centers, recruitment and academic management departments, skill-development platforms, and professional engineering study planning.

II. LITERATURE REVIEW

Modern web development frameworks, secure token-based authentication protocols, and NoSQL cloud databases have become important technologies for building scalable academic resource platforms and intelligent student assistance systems. Research in these areas focuses on decoupling client-server architectures, optimizing document retrieval, and securing administrative workflows. The review below is organized into four themes that directly shape the design of StudyMate.

A. Decoupled Client-Server Architectures in Education

A modern academic platform contains both structured presentation layers and dynamic data endpoints. Course notes, syllabi, project archives, and user profiles are normally managed as JSON documents. Decoupled architectures separate the frontend single-page application from backend micro-framework APIs [3], [4]. React.js with Vite provides fast component rendering and hot-module replacement, while Python Flask offers a lightweight RESTful backend [6]. In the proposed system, frontend and backend communicate seamlessly over HTTP to process user requests and render academic data dynamically.

B. Secure Authentication and Stateless JSON Web Tokens

Web applications handling student credentials and administrative data require robust security models. Traditional cookie sessions can introduce scaling challenges in cloud deployments, whereas JSON Web Tokens (JWT) provide a cryptographically signed, stateless mechanism for session validation. The proposed system utilizes Werkzeug security utilities for cryptographic password hashing and PyJWT for generating and verifying access tokens, ensuring that unauthorized users cannot bypass role restrictions or access protected administrative routes.

C. NoSQL Database Management with MongoDB Atlas

Relational databases can sometimes introduce rigid schema constraints when handling diverse educational metadata such as multi-unit syllabi, variable project technology tags, and nested quiz questions. NoSQL document databases offer flexible, schema-less storage ideal for academic asset management [5]. The proposed system uses MongoDB Atlas to store collections for users, syllabus files, lecture notes, question papers, books, projects, and aptitude questions, allowing rapid data retrieval and seamless cloud scalability.

D. Role-Based Access Control and Administrative Oversight

Academic portals require strict separation between student users and institutional administrators. Role-Based Access Control (RBAC) ensures that standard users can only access their authorized branch resources, while administrators retain specialized privileges. The proposed system enforces a strict single-admin policy and introduces advanced administrative

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151120

capabilities such as credential tracking and session impersonation, allowing administrators to troubleshoot student views safely.

E. Research Gap

Existing academic portals can provide static document repositories, basic user logins, or general department folders. However, project evaluations identify limitations such as lack of dynamic branch-and-semester filtering, absence of integrated AI doubt solvers, weak administrative security models, lack of session impersonation for support, and disjointed placement aptitude modules. The proposed system combines secure JWT authentication, dynamic resource filtering, an interactive digital library, placement aptitude tests, an AI doubt-solving Hub, and an exclusive administrative architecture into a single unified platform. Table I summarizes how StudyMate addresses each identified limitation.

TABLE I COMPARISON OF EXISTING PORTALS AND STUDYMATE

Aspect	Typical existing academic portal	StudyMate (proposed)
Resource access	Static downloads, generic folders	Dynamic branch-and-semester filtering
Authentication	Basic login or cookie sessions	Stateless JWT with Werkzeug hashing
Doubt solving	Not available or manual	24/7 AI Hub (LLM backend)
Placement prep	Disjointed or absent	Integrated aptitude mock tests
Admin security	Weak role separation	Single-admin policy with RBAC
Support tools	None	Session impersonation for troubleshooting

III. PROPOSED METHODOLOGY

The proposed methodology of StudyMate consists of secure user authentication, role-based access control, dynamic branch-and-semester resource filtering, placement aptitude testing, digital library curation, an AI doubt-solving hub, and an exclusive administrative oversight architecture.

A. System Overview

The user first registers into the system by specifying their full name, institutional email, password, academic branch (e.g., Artificial Intelligence and Machine Learning, Computer Engineering), and current semester. The backend intercepts the registration details, and stores the record in the cloud-hosted MongoDB Atlas database under the default student role. Depending on the user's navigation choice, the system dynamically delivers tailored content through five modules:

- **Academic Resources Hub:** Automatically filters curriculum files, lecture notes, and previous year question papers (PYQs) based on the user's authenticated branch and semester.
- **Aptitude & Placement Prep:** Provides interactive mock tests for quantitative reasoning, logical deduction, and verbal ability with instant score tabulation.
- **Books & Projects Hub:** Displays curated digital library catalogs and final-year project architectures categorized by emerging domains.
- **AI Hub:** Connects students to an intelligent LLM backend for 24/7 conceptual doubt solving.
- **Exclusive Admin Panel:** Accessible solely via authorized administrative credentials (admin@studymate.com), providing visibility into student records and session impersonation capabilities.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151120

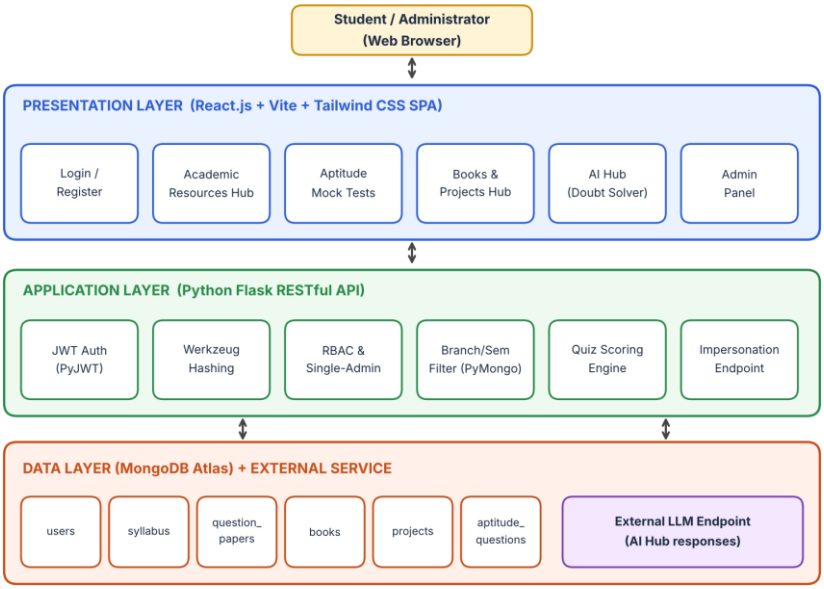


Fig. 1. Proposed system architecture of StudyMate

B. Authentication and Dynamic Resource Filtering

The academic resource filtering process ensures that students only view verified materials matching their exact curriculum:

1. The authenticated user requests course materials (e.g., syllabus or lecture notes).
2. The Flask backend extracts the branch and semester claims from the incoming JWT token header.
3. Database queries are dynamically constructed using PyMongo to filter collections matching the user's specific branch and semester parameters.
4. Filtered document URLs and metadata are returned as JSON responses to the React frontend.
5. Integrated HTML iframe PDF viewers render the selected study materials directly within the browser interface.

C. Aptitude and Project Curation Modules

The placement preparation and project recommendation modules operate through structured database models:

1. **Aptitude Module:** Retrieves categorized multiple-choice questions from the aptitude_questions collection, manages timer states on the client side, and calculates final scores upon quiz submission.
2. **Projects Hub:** Pulls curated final-year project titles, technology stacks, domain tags (such as AI/ML, Cloud Computing, and Web3), and architectural summaries from the projects collection to inspire and guide students.

D. AI Hub (Doubt Solver Module)

The AI Hub module serves as an interactive assistant for engineering students:

1. The student submits a technical query or coding doubt through the React frontend interface.
2. The Flask backend receives the prompt and securely routes it to the configured LLM endpoint.
3. The AI model generates a contextual, step-by-step technical explanation or code snippet.
4. The response is formatted and rendered in real time on the student's dashboard, providing round-the-clock academic support.

E. Exclusive Admin Architecture and Session Impersonation

To maintain strict institutional security and streamlined user support, StudyMate implements an advanced administrative architecture:

1. **Single Admin Policy:** All standard frontend registrations are hardcoded to the student role on the backend, making unauthorized administrative privilege escalation mathematically and logically impossible.
2. **Admin Command Center:** Unlocks a dedicated dashboard for the sole administrator, displaying comprehensive rosters of all registered students, their branches, semesters, and stored credentials.
3. **Session Impersonation:** The administrator can trigger an impersonation endpoint (POST /api/admin/impersonate/<student_id>), which generates a temporary, scoped JWT token allowing the administrator to view and navigate the platform exactly as that specific student sees it for troubleshooting and support purposes.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151120

F. Database Schema and Collections (studymate_rcpit)

The persistent data of StudyMate is organized as document collections in MongoDB Atlas. The main collections are listed in Table II.

TABLE II MONGODB COLLECTIONS IN STUDYMATE_RCPIT

Collection Name	Description
users	Stores user identification, hashed passwords, branch, year, semester, and role.
syllabus	Stores official curriculum data including unit names, subject codes, branches, semesters.
question_papers	Stores previous year question papers (PYQs) indexed by subject and semester.
books	Maintains the digital library catalog including titles, authors, categories.
projects	Contains curated final-year project titles, domains, technologies used.

G. Exploratory Resource and User Analytics

Exploratory analysis of platform usage and academic resource distribution helps maintain institutional efficiency:

- Branch-wise distribution analytics for monitoring student engagement across departments.
- Syllabus and note frequency tracking to ensure comprehensive coverage across semesters.
- Aptitude score performance distributions for evaluating student placement readiness.
- Project domain popularity analysis to align final-year suggestions with current industry trends.

IV. SYSTEM IMPLEMENTATION

The proposed system, StudyMate: A Multi-Agent Academic Assistance System, is implemented as a modern web-based application utilizing a decoupled client-server architecture with React.js and Python Flask.

A. Technology Stack

TABLE III TECHNOLOGY STACK

Technology	Role in StudyMate
React.js & Vite	Dynamic, responsive Single Page Application (SPA) frontend.
Tailwind CSS	Utility-first CSS framework for modern, mobile-responsive user interface design.
Lucide React	Clean vector icons across all hubs and dashboards.
React Router	Client-side navigation between different academic views.
Python Flask	Lightweight micro-framework serving the backend RESTful API endpoints.
PyMongo	Python driver for seamless NoSQL interactions.
PyJWT & Werkzeug	Secure JWT-based stateless authentication and cryptographic password hashing.
MongoDB Atlas	Cloud-hosted NoSQL database storing user profiles, syllabi, notes, question papers, books, projects, and aptitude questions.

B. User Registration and Login Module

The registration and login module provides strict access control for students and administrators. A new user enters registration details—specifying their full name, email, password, branch, and semester—which are securely processed and stored in the database (Fig. 2). Registered users log in using their email and password (Fig. 3). Flask verifies the entered credentials using Werkzeug hashing. If valid, the user is granted access to their personalized dashboard via a signed JWT token; invalid credentials are automatically rejected. Project module testing confirms that user registration and login authentication operate successfully.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151120

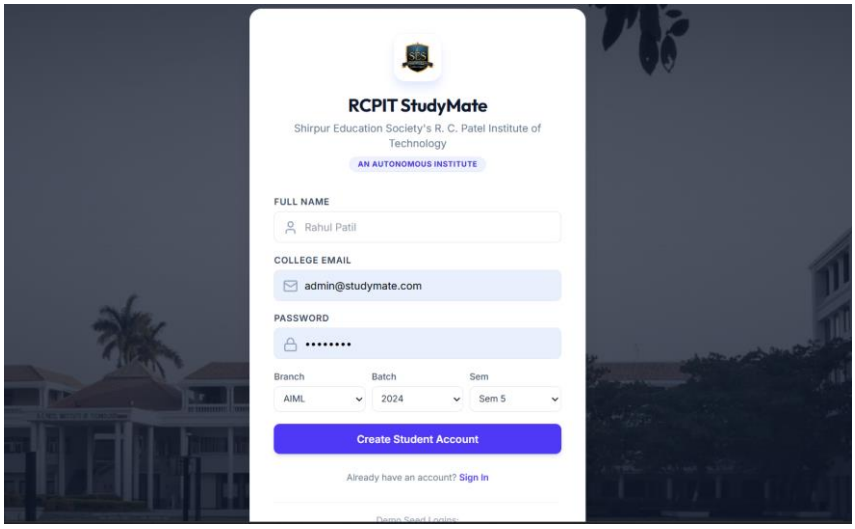


Fig. 2. User register page

The registration page allows users to create an account.

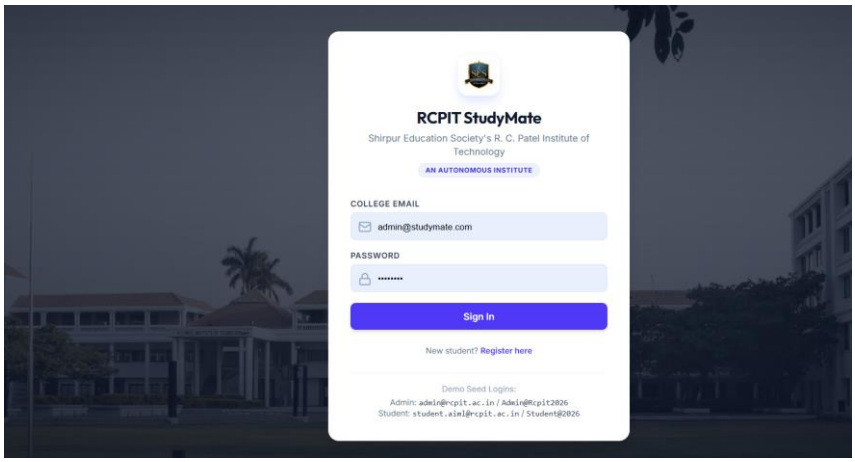


Fig. 3. User login page

The login page allows registered users to securely access their StudyMate dashboard.

C. Dynamic Academic Resources Hub

The academic resource module allows students to access official university syllabus files, lecture notes, and previous year question papers (PYQs) filtered automatically by their registered branch and semester. The automated workflow is structured as follows:

User Profile Claims → JWT Verification → Branch/Semester Query Filtering → MongoDB Atlas → Dynamic HTML iFrame PDF Rendering

This automated filtering reduces manual searching and ensures students view precisely what is relevant to their curriculum (Fig. 4).

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151120

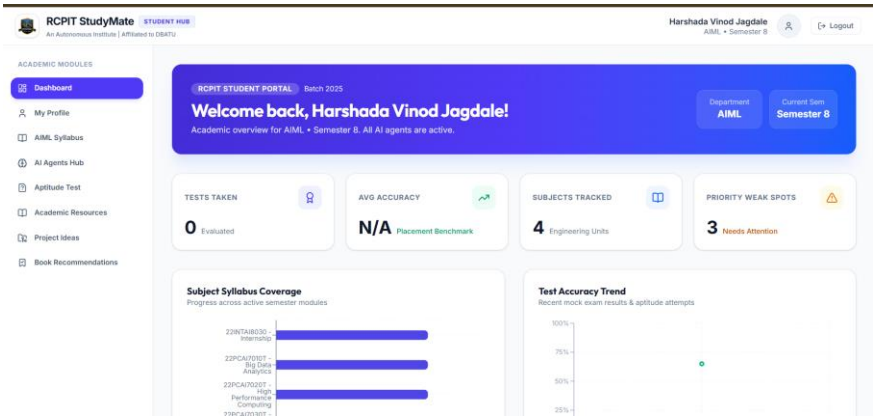


Fig. 4. Student academic dashboard and branch/semester resource filter

D. Aptitude Prep, Digital Library, and AI Hub Modules

The platform integrates several specialized auxiliary features (Fig. 5):

- **Placement Aptitude Module:** Tests quantitative reasoning, logical deduction, and verbal skills with automated scoring.
- **Books & Projects Hubs:** Provides an interactive digital book catalog and curated final-year project archives complete with technology stack summaries.
- **AI Hub:** Integrates 24/7 AI-driven doubt solving for engineering concepts.

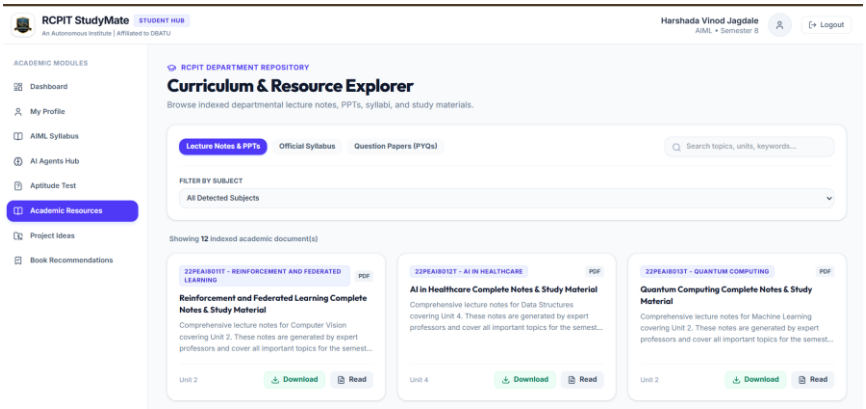


Fig. 5. Aptitude mock test and AI Hub interface

V. RESULTS AND DISCUSSION

The developed StudyMate system was tested module by module and after complete integration. Project testing indicates that all major components successfully achieved their expected outcomes.

A. Overall Module Testing

TABLE IV OVERALL MODULE TESTING

Test Case	Expected Result	Status
User Registration	Student account created with branch and semester profile	Passed
User Login & JWT	Secure JWT generated; dashboard access granted	Passed
Branch/Semester Filtering	Curriculum files filtered dynamically by user profile	Passed
PDF Document Viewer	Syllabus and lecture notes rendered inline correctly	Passed
Aptitude Mock Test	Quantitative/logical tests submitted and scored instantly	Passed
AI Doubt Solver	Contextual explanations generated for engineering queries	Passed

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151120

Test Case	Expected Result	Status
Admin Access Control	Standard registration restricted strictly to student role	Passed
Admin Impersonation	Temporary token generated for student view simulation	Passed

The testing results indicate successful operation of the major system modules.

B. Dynamic Resource Filtering Results

The academic resource module successfully accepted student branch and semester parameters from encrypted JWT tokens and queried the MongoDB Atlas database. Curriculum files, lecture notes, and question papers were filtered instantly, reducing manual searching effort and providing immediate access to relevant study materials.

C. Aptitude and Project Hub Results

The placement aptitude module successfully delivered mock tests across quantitative, logical, and verbal domains, providing instant score tabulation. Additionally, the projects repository supplied curated final-year project architectures and technology stacks to guide students effectively.

D. Administrative Architecture Results

The exclusive administrative module successfully enforced the single-admin security policy, preventing unauthorized privilege escalation. Furthermore, the credential panel and session impersonation features operated smoothly, allowing the administrator to inspect student rosters and simulate student portal views for troubleshooting.

E. AI Hub Doubt-Solving Results

The AI doubt-solving module successfully processed technical engineering queries submitted by students and returned immediate, structured explanations. This integration provides valuable 24/7 academic assistance outside traditional classroom hours.

F. Discussion

The implementation demonstrates how a decoupled React-Flask architecture paired with MongoDB Atlas and JWT authentication can deliver a high-performance, secure academic resource ecosystem. Centralizing curriculum materials, placement tests, digital books, project ideas, and an AI doubt solver eliminates resource fragmentation. The main contribution of StudyMate is the integration of multiple institutional functions:

Secure Authentication + Dynamic Resource Filtering + Placement Aptitude Prep + Digital Library + Final-Year Projects + AI Doubt Solver + Exclusive Admin Oversight

This integrated platform provides significantly more comprehensive support than isolated document storage or generic web portals. However, the overall effectiveness depends on regular updates to curriculum metadata and course files. Future enhancements will focus on multi-agent AI tutoring, real-time study rooms, and native mobile applications.

VI. CONCLUSION AND FUTURE SCOPE

StudyMate: A Multi-Agent Academic Assistance System was successfully designed, engineered, and deployed as a comprehensive, web-based platform tailored exclusively for the students and faculty of the R. C. Patel Institute of Technology (RCPIT). The system combines modern web technologies, secure authentication protocols, dynamic curriculum filtering, placement preparation modules, and artificial intelligence to centralize academic resources and streamline student learning.

The platform securely authenticates users using JSON Web Tokens (JWT) and Werkzeug password hashing, storing persistent records in a cloud-hosted MongoDB Atlas database. It automatically extracts branch and semester profile claims to filter official university syllabus files, lecture notes, and previous year question papers (PYQs), providing instant access via integrated HTML PDF viewers.

Furthermore, the platform delivers interactive quantitative and logical aptitude mock tests, a curated digital library catalog, final-year project architecture summaries, and a 24/7 AI Hub for engineering doubt solving. To ensure robust institutional management, StudyMate incorporates an exclusive administrative architecture enforcing a single-admin security policy alongside real-time credential tracking and session impersonation capabilities. Testing performed across all major components confirmed successful operation for user registration, JWT token generation, resource filtering,



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151120

mock testing, AI assistance, and administrative oversight. The proposed system effectively eliminates resource fragmentation, helping engineering students understand their curriculum requirements, enhance placement readiness, and improve academic productivity.

In future work, the platform will be extended with multi-agent AI tutoring, real-time study rooms, and native mobile applications, and its curriculum metadata and course files will be updated regularly to keep the content current.

ACKNOWLEDGEMENT

The authors sincerely thank Prof. S. L. Tambe for valuable guidance, continuous support, and technical suggestions throughout the development of this project. The authors also sincerely thank the Head of the Department, Prof. Dr. U. M. Patil, the Director, Prof. Dr. J. B. Patil, and all faculty members of the Department of Artificial Intelligence and Machine Learning, R. C. Patel Institute of Technology, Shirpur, for their encouragement and support.

REFERENCES

[1]. T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in Proc. International Conference on Learning Representations (ICLR), 2013.
[2]. J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. NAACL-HLT, 2019, pp. 4171–4186.
[3]. React Documentation, “React – A JavaScript library for building user interfaces,” Meta Platforms.
[4]. Flask Documentation, “Flask web framework documentation,” Pallets Projects.
[5]. MongoDB Documentation, “MongoDB Atlas cloud database administration and PyMongo driver guide.”
[6]. L. Richardson and S. Ruby, RESTful Web Services, O’Reilly Media, 2007.

 This work is licensed under a Creative Commons Attribution 4.0 International License