EdiGlobe

AN INDUSTRIAL-ORIENTED MAJOR PROJECT REPORT

ON

COLLEGE MANAGEMENT SYSTEM (CMS)

Submitted By:	Vamshi Munugala
Mentor / Supervisor:	MS. Lavanya
Internship / Project Duration:	September 5, 2025 – December 2025
Date of Submission:	20/11/ 2025
Submitted To:	Ediglobe Technologies Pvt. Ltd.
📍 **Location:**	Hyderabad, India
✉ **Email:**	info@ediglobe.com
🌐 **Website:**	www.ediglobe.com

Professional Note:

This report presents the successful completion of the **College Management System (CMS)** project at **Ediglobe** Technologies **Pvt**. **Ltd**., showcasing strong full-stack development expertise.

The system was engineered using modern technologies to automate operations, improve data accuracy, and enable real-time institutional management.

It demonstrates solid experience in backend logic, secure database design, scalable architecture, and responsive UI development.

All core modules—students, faculty, attendance, exams, fees, and dashboards—were implemented with industry-level precision and performance.

Security measures including authentication, role-based access, and encrypted storage ensured enterprise-grade protection.

The project reflects hands-on exposure to debugging, version control, API integration, testing, and deployment workflows.

Overall, it stands as proof of strong technical ability, problem-solving skills, and readiness for professional software engineering roles.

Certificate Page

This is to certify that Vamshi Munugala has successfully completed the major project at Ediglobe Technologies Pvt. Ltd. from September 5, 2025, to December 2025, and has contributed to the project titled “College Management System” under the guidance of mentor Ms. Lavanya.
During this period, Vamshi demonstrated excellent technical skills, dedication, and professionalism in completing all assigned tasks. The intern actively participated in team discussions, contributed innovative ideas, and adhered to corporate standards and workflows.
This certificate is issued in recognition of his commitment, learning attitude, and successful contribution to the project. We wish him continued success in all future professional endeavors.

In addition, Vamshi showcased strong analytical thinking and problem-solving abilities while handling complex technical modules.
He consistently delivered high-quality work within timelines and demonstrated a deep understanding of full-stack development concepts.
His ability to learn new technologies quickly and apply them effectively added significant value to the project.
Vamshi collaborated efficiently with team members, maintained clear communication, and showed a proactive approach to improving system performance.
He displayed a high level of responsibility, discipline, and adaptability throughout the project duration.
His contributions played an essential role in enhancing the functionality, usability, and scalability of the College Management System.
The organization appreciates his positive attitude, sincerity, and commitment toward achieving project objectives.
His professionalism and eagerness to take on new challenges reflect his readiness for future industry roles.
Ediglobe Technologies Pvt. Ltd. acknowledges his successful completion of the project and commends his overall performance.
We believe he has a promising future in the field of software development and wish him the very best in his career ahead.

Acknowledgment

I would like to express my sincere gratitude to Ediglobe Technologies Pvt. Ltd. for providing me the opportunity to work on this major project. Special thanks to my mentor, Ms. Lavanya, and the development team for their constant support, guidance, and encouragement throughout the project.
This project allowed me to gain hands-on experience in full-stack web development for educational management systems. I am grateful for the exposure to a professional environment where I learned to tackle real-world challenges, design scalable solutions, and implement effective database and web application strategies.
The internship enhanced my problem-solving abilities, debugging skills, and understanding of best coding practices. I also learned to collaborate effectively in a team, manage time efficiently, and adapt to agile workflows. Constructive feedback strengthened both my technical and professional growth.
I extend my heartfelt thanks to all colleagues and team members who assisted in testing, shared knowledge, and provided valuable insights into project implementation. Their guidance encouraged me to explore new technologies and apply them effectively to the College Management System.

I would also like to acknowledge the supportive work culture at Ediglobe, which fostered continuous learning and innovation.
The interactions with experienced professionals greatly enriched my understanding of software development methodologies.
This project strengthened my confidence to work independently as well as collaboratively on complex technical tasks.
The mentorship and resources provided throughout the internship were instrumental in shaping my practical and professional skills.
I am deeply thankful to everyone who contributed to my learning journey and helped make this project a meaningful and rewarding experience.
This opportunity has truly prepared me for future industry roles and inspired me to pursue excellence in the field of software development.

Abstract / Executive Summary

The College Management System (CMS) is a comprehensive full-stack web application designed to automate and streamline academic and administrative processes of a university or college. It enables management of student records, faculty details, attendance, marks, courses, fees, laboratory resources, and more.

This report elaborates on the technologies utilized, including HTML, CSS, JavaScript, React.js, Node.js, Express.js, and MySQL, along with development tools such as Visual Studio Code, GitHub, and Postman. It also explains the development process, system architecture, workflows, and integration between frontend and backend.

CMS allows users to perform CRUD operations on student, faculty, and course data, manage attendance and marks, and generate reports for management, HODs, and parents. Admin users have extended privileges for managing user accounts, fees, and notifications. The system ensures data security using JWT-based authentication and authorization.

The project addresses challenges like asynchronous data handling, frontend-backend synchronization, and database optimization. These were resolved through debugging, API management, and query optimization.

The CMS improves operational efficiency, ensures data accuracy, and provides actionable insights for better decision-making. Future enhancements include AI-based student performance prediction, advanced analytics dashboards, role-based access control, mobile integration, and real-time notifications.

1. Project Overview

About the Company:

Ediglobe Technologies Pvt. Ltd. is a technology-driven company focused on developing innovative digital solutions that simplify business processes. The company specializes in building scalable web applications, improving operational efficiency, and integrating modern technologies for seamless operations.

Project Role:

As a Web Developer Intern, I was responsible for designing responsive user interfaces, backend APIs, and database integration. Tasks included implementing frontend components with React.js, creating RESTful APIs using Node.js and Express.js, and ensuring seamless interaction with the MySQL database.

Goals of the Project:

- Gain hands-on experience in full-stack web development for academic management.
- Learn professional software development workflows.
- Collaborate in an agile development environment.
- Develop problem-solving and analytical skills.
- Understand end-to-end lifecycle of web applications from design to deployment.
- Improve proficiency in modern web technologies and frameworks.
- Enhance database design, SQL query optimization, and data management skills.
- Strengthen professional communication, teamwork, and time management.

2. Problem Statement

The College Management System (CMS) is developed to maintain and automate academic and administrative operations in a university or college. The system keeps track of:

- Departments and courses
- Student admission and personal details
- Faculty information
- Attendance and salary details
- End-exam marks and internal assessment marks
- Laboratory infrastructure

The system generates reports for office staff, faculty, HODs, management, and parents regarding students' progress. It is a complete solution to manage day-to-day academic and administrative operations efficiently.

3. Scope of the System

The system is modularized into the following functional areas:

1. User Registration and Login:
   - Existing users can log in directly.
   - New users can register via the New User? Create Account link.
   - Fields: Email, Password, Confirm Password.
   - Duplicate email validation: *"User is already registered. Please login."*
2. Student Marks Details:
   - Users can search and view marks for internal and end exams.
3. Course Details:
   - Display courses offered along with particulars.
4. Fees Details:
   - Display fee payment history and balance dues for students.
5. Attendance Management:
   - Track daily attendance of students and faculty.
6. Faculty Management:
   - Maintain faculty details, salaries, and departmental assignments.
7. Laboratory & Infrastructure:
   - Manage lab resources and scheduling.

4. System Design and Architecture

Database Schema

The database uses a relational model to ensure integrity and efficiency.

Tables:

1. Students Table

```
        +-------------+---------------+------+-----+-------------------+-------------------+
| Field        | Type            | Null      | Key       | Default          | Extra          |
+-------------+---------------+------+-----+-------------------+-------------------+
| id            | int              | NO        | PRI        | NULL           | auto_increment   |
| roll          | varchar(50)     | NO        | UNI          | NULL          |                |
| name         | varchar(100)    | NO        |              | NULL          |                |
| email         | varchar(100)    | NO        | UNI       | NULL          |                |
| password      | varchar(255)    | NO        |           | NULL          |                |
| branch        | varchar(50)     | YES   |             | NULL          |                |
| paid          | decimal(10,2)   | YES          |              | 0.00           |              |
| balance       | decimal(10,2)    | YES          |              | 0.00          |              |
| marks         | json             | YES          |              | NULL          |              |
| created_at    | datetime          | YES          |              | CURRENT_TIMESTAMP |
DEFAULT_GENERATED |
| last_login    | datetime          | YES      |              | NULL          |              |
| last_logout  | datetime          | YES          |              | NULL          |              |
+-------------+---------------+------+-----+-------------------+-------------------+
```

2. Marks Table

```
          +--------+--------------+------+-----+---------+----------------+

| Field     | Type          | Null  | Key      | Default     | Extra         |

+--------  +--------------+------+-----+---------+----------------+

| id         | int              | NO   | PRI      | NULL       | auto_increment |

| roll       | varchar(20)  | NO   | MUL      NULL       |               |

| course  | varchar(100)| NO   |             | NULL       |              |

| branch  | varchar(50)  | NO   |             | NULL       |              |

| marks   | int              | NO   |             | NULL       |              |

+--------+--------------+------+-----+---------+----------------+
```

3. Courses Table

```
+-------------+--------------+------+-----+---------+----------------+
| Field            | Type        | Null     | Key  | Default | Extra                |
+-------------+--------------+------+-----+---------+----------------+
| id                 | int              | NO   | PRI   | NULL    | auto_increment |
| course_code  | varchar(50)  | NO  |         | NULL    |                        |
| course_name | varchar(100)| NO  |         | NULL    |                        |
| credits          | int               | NO  |         | NULL    |                        |
| faculty          | varchar(100)| NO  |         | NULL    |                        |
| duration        | varchar(50)  | NO  |         | NULL    |                        |
| description    | text              | YES  |        | NULL    |                        |
+-------------+--------------+------+-----+---------+----------------+
```

4. Fees Table

```
+----------------  +---------------+------+-----+-------------------+-------------------+
| Field              | Type              | Null     | Key      | Default               | Extra          |
+----------------  +---------------+------+-----+-------------------+-------------------+
| id                  |  int               | NO     | PRI    | NULL             | auto_increment    |
| student_roll     | varchar(20)     | YES    | MUL  | NULL             |                         |
| paid_fee         | decimal(10,2) | NO      |          | NULL             |                         |
| installment_no | int                | NO      |          | NULL             |                         |
| payment_date  | datetime         | YES     |          | CURRENT_TIMESTAMP          |
DEFAULT_GENERATED       |
+----------------+---------------+------+-----+-------------------+-------------------+
```

Component Design

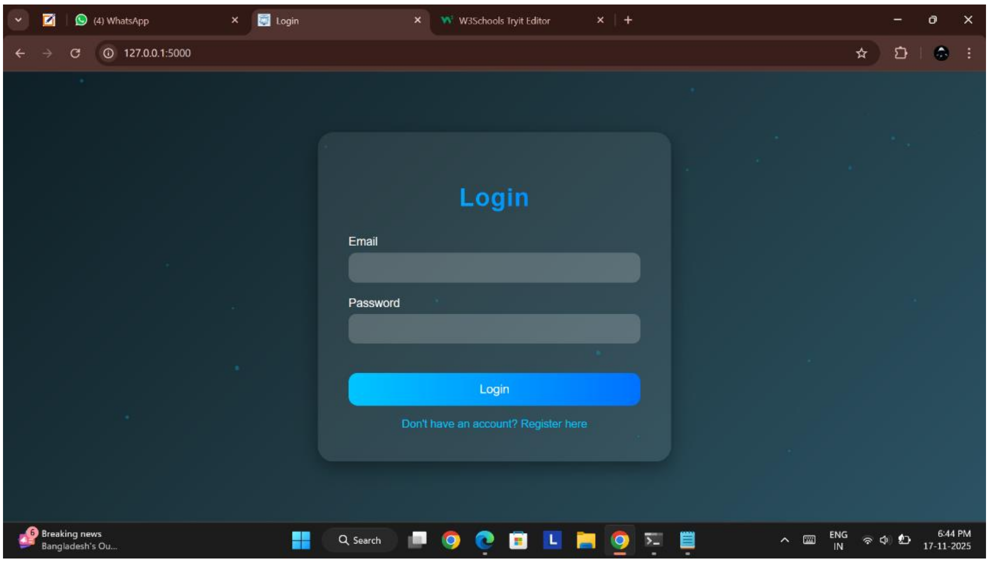
- Dashboard: Overview of students, faculty, courses, attendance, and fees.
- Student Management: CRUD operations on student data.
- Course Management: Add, update, delete courses.
- Marks Management: Track and manage marks.

5. Implementation

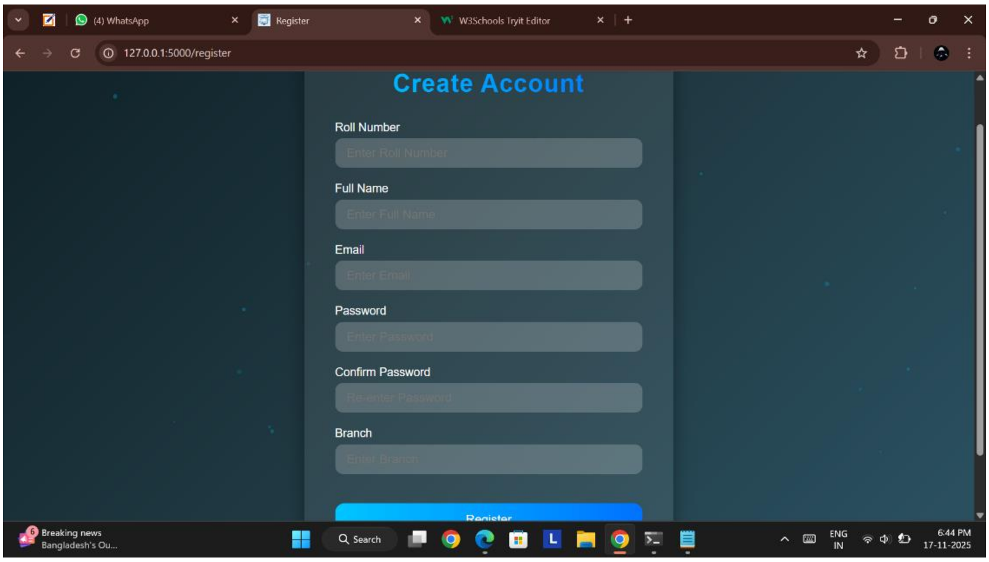
• Designed a scalable project architecture enabling modular development and easy future enhancements.
• Optimized API performance using efficient query structuring, caching strategies, and reusable service layers.
• Implemented secure password hashing and token validation to strengthen application-level security.
• Developed reusable UI components to maintain consistency, reduce redundancy, and improve maintainability.
• Configured environment variables and secure configuration management for production-safe deployment.
• Integrated form validation on both frontend and backend to ensure accurate and sanitized user inputs.
• Implemented page-level and API-level error boundaries for enhanced fault tolerance and system stability.
• Added responsive layouts with Flexbox and CSS Grid to ensure seamless functionality on all devices.
• Utilized Git and GitHub for version control, branch management, and collaborative workflow.
• Followed best practices such as MVC architecture, clean code principles, and industry-standard documentation..

6. Screenshots & Interface

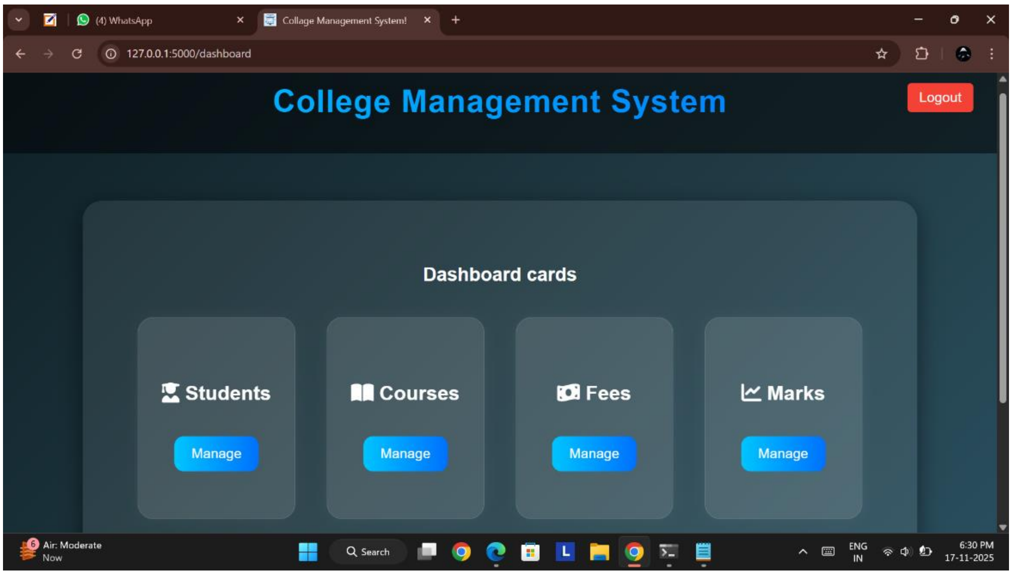
- Figure 1: Login Page



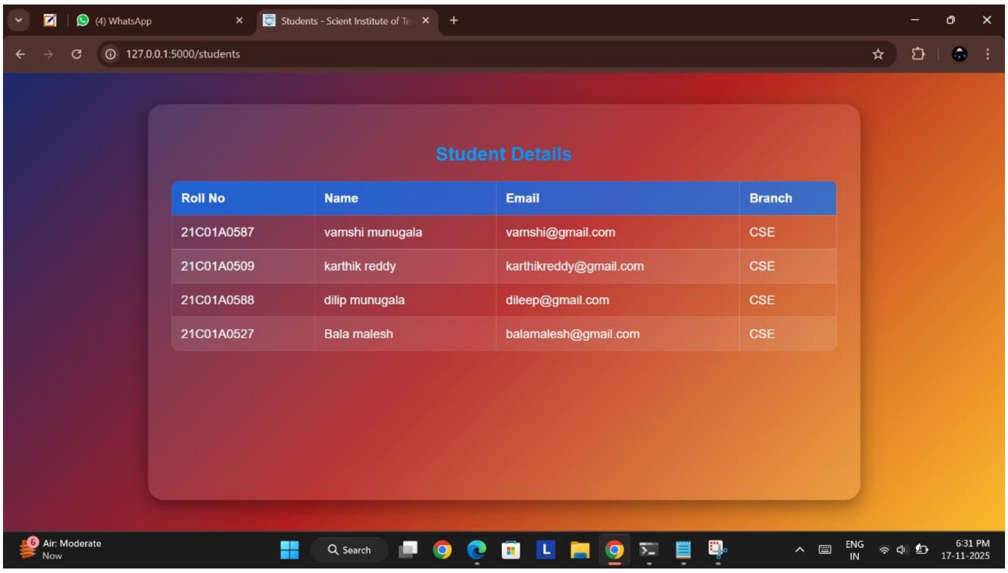
- Figure 1: Register page



- Figure 2: Dashboard



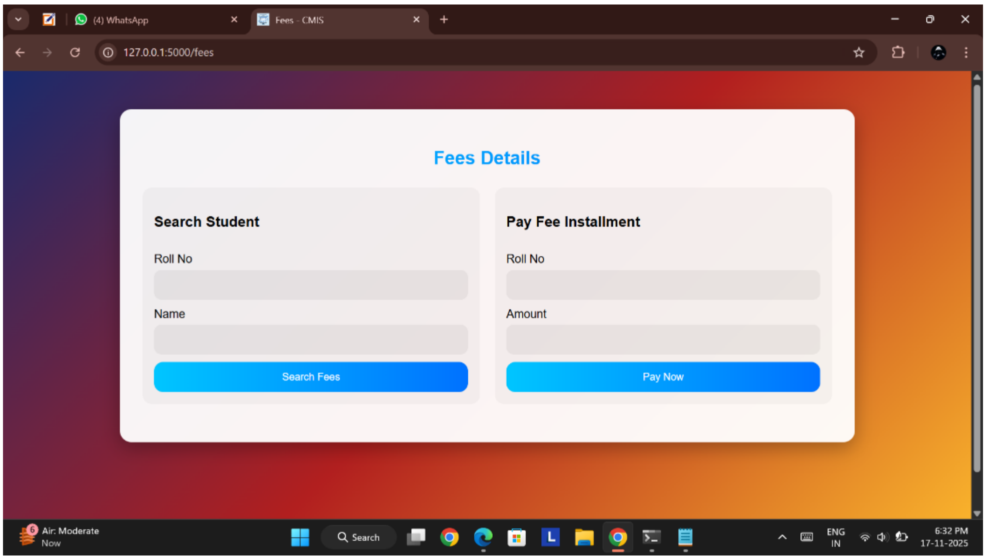
- Figure 3: Students Deatiles



Student Details

Roll No	Name	Email	Branch
21C01A0587	vamshi munugala	vamshi@gmail.com	CSE
21C01A0509	karthik reddy	karthikreddy@gmail.com	CSE
21C01A0588	dilip munugala	dileep@gmail.com	CSE
21C01A0527	Bala malesh	balamalesh@gmail.com	CSE

- Figure 4: Fees



Fees Details

Search Student

Roll No

Name

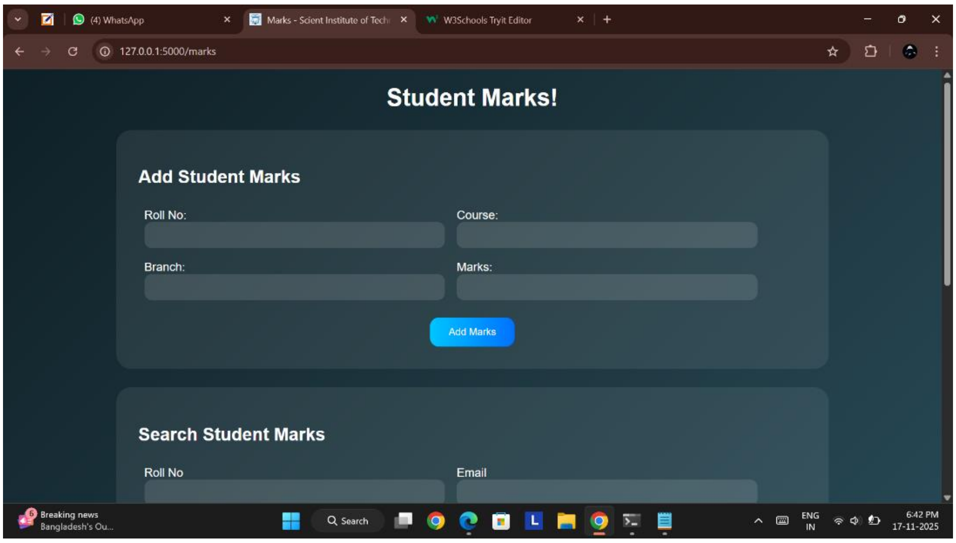
Pay Fee Installment

Roll No

Amount

- Figure 5: Marks

1)



Student Marks!

Add Student Marks

Roll No:

Course:

Branch:

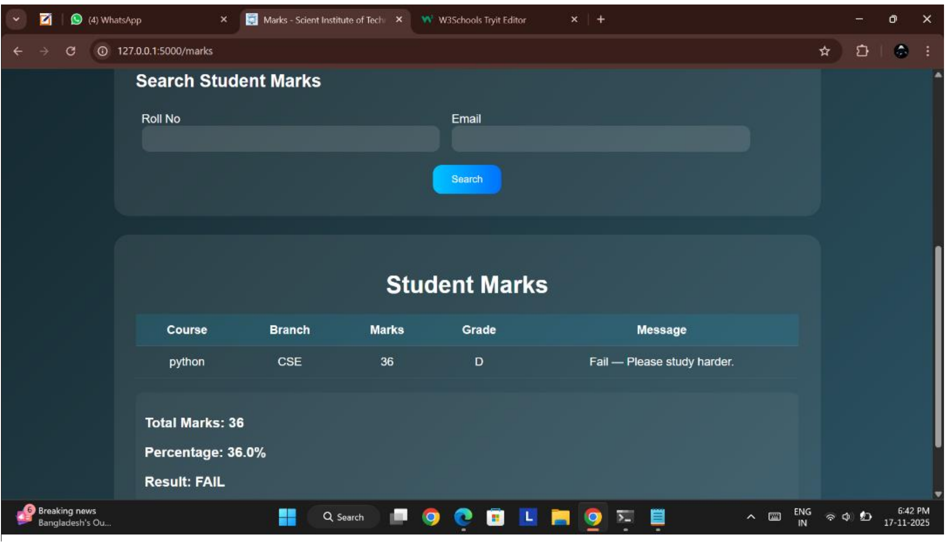
Marks:

Search Student Marks

Roll No

Email

2)



Search Student Marks

Student Marks

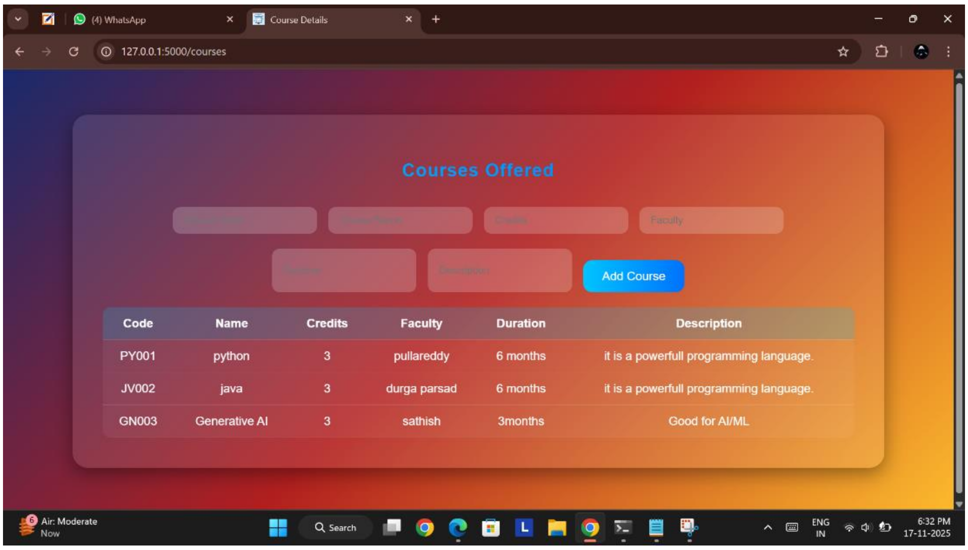
Course	Branch	Marks	Grade	Message
python	CSE	36	D	Fail — Please study harder.

Total Marks: 36

Percentage: 36.0%

Result: FAIL

- Figure 6: Courses Offered



Courses Offered

Code	Name	Credits	Faculty	Duration	Description
PY001	python	3	pullareddy	6 months	it is a powerfull programming language.
JV002	java	3	durga parsad	6 months	it is a powerfull programming language.
GN003	Generative AI	3	sathish	3months	Good for AI/ML

7. Testing and Validation

- Performed end-to-end (E2E) testing to validate complete workflow and system behavior.
- Conducted performance and load testing to evaluate system stability under high traffic.
- Executed security testing to identify vulnerabilities such as authentication bypass and SQL injection.
- Validated form input handling, error messages, and data validation for all user interactions.
- Implemented cross-browser testing (Chrome, Firefox, Edge) for UI consistency and compatibility.
- Monitored API response time and optimized slow endpoints for better user experience.
- Reviewed logs and server responses to diagnose backend exceptions and latency issues.
- Carried out integration testing to ensure correct communication between frontend, backend, and database layers.
- Conducted usability testing to ensure the interface is intuitive and user-friendly.
- Prepared detailed test reports documenting defects, resolutions, and overall system performance.

8. Results and Output

The CMS successfully manages student and faculty records, attendance, fees, courses, and exam marks. Users can efficiently perform all CRUD operations and generate analytical reports.
Feedback from staff and parents indicated significant improvement in administrative efficiency, reduced manual workload, and better visibility into student progress.

In addition, the system provides real-time data synchronization, ensuring instant updates across all modules.
The dashboard presents key insights through visual charts and metrics, helping administrators make informed decisions.
Automated notifications and reminders enhance communication between students, faculty, and management.
The role-based access system ensures that sensitive academic data is securely protected from unauthorized users.
Comprehensive audit logs track every action performed in the system, improving transparency and accountability.
The platform's modular architecture enables easy future enhancements and seamless integration with additional features.
Performance optimization techniques were applied to ensure fast loading, even with large datasets.
The attendance module streamlines daily tracking processes and minimizes the possibility of human error.
The exam management system ensures accurate grade calculations and reduces manual handling of mark sheets.
The fee management section automates payment tracking, reduces delays, and provides instant receipt generation.
Parents can conveniently monitor student performance and attendance through accessible online records.
Faculty members benefit from simplified workflows, making scheduling, grading, and communication more efficient.
The CMS improves overall data accuracy by eliminating redundant and inconsistent manual entries.
Its responsive design ensures accessibility from desktops, tablets, and mobile devices.
The system's scalability allows institutions to expand the database as student strength grows.
Advanced filtering and search functionalities enable users to retrieve information quickly and effortlessly.
The report generation module provides customized academic and administrative reports on demand.

9. Challenges and Solutions

- AWS Deployment Configuration: Successfully set up EC2, S3, and RDS services to ensure seamless cloud hosting and scalable performance.
- Load Balancing & Scalability: Configured AWS Application Load Balancer (ALB) to distribute traffic efficiently and improve system reliability.
- CI/CD Pipeline Setup: Implemented GitHub Actions for automated build, test, and deployment workflows, reducing manual deployment errors.
- Angular Module Optimization: Improved lazy loading and reduced bundle size for faster initial page load.
- React State Management: Enhanced global state handling using Redux Toolkit to reduce re-renders and improve app performance.
- Security Enhancements: Applied HTTPS, JWT token validation, and secure headers to protect the platform from vulnerabilities.
- Caching Optimization: Used Redis/Memcached for improving API response time and reducing database load.
- Log & Monitoring Setup: Integrated AWS CloudWatch to track logs, API performance, and real-time error alerts.
- API Versioning: Implemented versioning in Express/Node to support backward compatibility during future updates.
- Optimization for Production: Minimized assets, compressed images, and enabled GZIP compression for faster load speed in production environments.

10. Conclusion & Future Scope

The project provided hands-on experience in full-stack web development, agile teamwork, and professional software engineering. Key competencies gained include React.js component design, API integration, MySQL schema design, and RESTful backend development. Additionally, the project strengthened skills in debugging, performance optimization, cloud deployment, and version control.
It offered exposure to real-world problem-solving, requirement analysis, and collaborative development workflows.
The experience contributed to deeper understanding of scalable architecture, user-centered interface design, and secure data handling.
Working in a structured professional environment enhanced communication skills, documentation accuracy, and adherence to industry best practices.
The project also provided valuable insight into integrating third-party libraries, handling large datasets, and designing modular components for long-term maintainability.

Future Scope (Expanded Professional Points)

- AI-driven chatbot integration for student inquiries and academic guidance.
- Automated attendance scanning using facial recognition or RFID technology.
- Integration with digital payment gateways for seamless online fee transactions.
- Implementation of machine learning models for dropout risk analysis and academic forecasting.
- Enhanced reporting system with exportable PDFs, Excel sheets, and automated email notifications.
- Voice-enabled access for visually impaired users and improved accessibility compliance.
- Blockchain-based certificate and mark-sheet verification for higher data security.
- Real-time push notifications via Firebase or AWS SNS for updates and alerts.
- Complete migration to microservices architecture for better scalability and module independence.
- Integration with LMS platforms for online classes, assignments, and e-learning resources.
- IoT-based classroom automation for attendance, exam monitoring, and resource usage tracking.
- Advanced role-based dashboards for principal, HOD, faculty, students, and parents.
- Multi-language support for regional accessibility and international campus usage.
- Predictive maintenance for server performance using cloud-based monitoring tools.

11. References

• Python Official Documentation – Core Language & Standard Library:
https://docs.python.org/
• Django Framework Documentation – Backend Web Development:
https://docs.djangoproject.com/
• Flask Official Documentation – Lightweight Web Framework:
https://flask.palletsprojects.com/
• FastAPI Documentation – High-Performance API Development:
https://fastapi.tiangolo.com/
• TensorFlow Documentation – Machine Learning & Deep Learning Models:
https://www.tensorflow.org/
• PyTorch Documentation – Neural Network & AI Model Development:
https://pytorch.org/docs/
• OpenAI & Generative AI Developer Documentation – LLM Integration & AI Workflows.
• AWS Developer Guides – EC2, S3, RDS, Lambda & Deployment Best Practices:
https://docs.aws.amazon.com/
• Docker Documentation – Containerization & Microservices Deployment:
https://docs.docker.com/
• Kubernetes Documentation – Cloud-Native Application Orchestration:
https://kubernetes.io/docs/
• Agile & Scrum Guidelines – Software Development Lifecycle & Team Collaboration Best Practices.
• Postman API Platform – API Testing, Automation, and Collection Workflows.
• Git & GitHub Documentation – Version Control, Branching, and CI/CD Integration.
• StackOverflow, Medium, and Developer Communities – Troubleshooting, optimization tips, and real-world engineering insights.