

BVC EduPay—Smart College Fee Payment System

Perabattula Ramesh¹, Rayudu Venkata Satya Sai Kumar², Kandabhattu S.S.V.K. Satish Raju³, Puppala Vamsi Surya Manikanta Kumar⁴, Mahadasa Sri Swathi⁵, Kusuma Rajasekhar^{6,*}

Abstract

Fee management has become a crucial component of modern educational institutions, with increasing reliance on digital platforms to handle financial transactions and administrative processes. However, students and staff often encounter challenges such as scattered fee information, manual verification procedures, delayed payment updates, and limited transparency across departments. To address these issues, BVC EduPay—a smart college fee payment system—has been developed as a centralized digital platform that consolidates academic fees, examination fees, hostel charges, transport fees, library dues, and placement-related payments into a unified system. The platform simplifies the fee payment process by providing structured and real-time information tailored to user roles, along with features such as automated eligibility checking, secure online transactions, and real-time payment status updates. The system is implemented using modern full-stack MongoDB, Express.js, React, and Node.js (MERN) architecture, with React for the frontend to ensure an interactive user experience, Node.js and Express.js for backend processing, MongoDB for efficient data storage, and Razorpay integrated as a secure payment gateway. Its modular and scalable design supports future extensions such as mobile app integration, advanced analytics dashboards, automated reminders, and connectivity with institutional enterprise resource planning (ERP) systems. Overall, BVC EduPay enhances transparency, reduces administrative workload, and significantly improves the efficiency of fee management within educational environments.

Keywords: Fee management system, MERN stack, Razorpay, digital payments, EDUTech, full-stack application

INTRODUCTION

The digital transformation of educational institutions has accelerated significantly with the widespread use of online platforms, web-based applications, and automated management systems. In earlier years, students depended heavily on manual procedures, physical visits to administrative offices, and paper-based receipts to complete college fee payments. This traditional approach is time-consuming, requires repeated verification, and often causes unnecessary delays during critical academic periods, such as examination registrations and admissions. With the adoption of digital systems across various sectors, students and parents now expect faster, more reliable, and more transparent fee payment processes.

*Author For Correspondence

Kusuma Rajasekhar
E-mail: krsece.bvce@bvcegroup.in

¹⁻⁵Scholar, Department of Electronics and Communication Engineering, Bonam Venkata Chalamayya Engineering College (Autonomous), Odalarevu, Andhra Pradesh, India

⁶Professor & Training and Placement Officer, Department of Electronics and Communication Engineering, Bonam Venkata Chalamayya Engineering College (Autonomous), Odalarevu, Andhra Pradesh, India

Received Date: March 31, 2026

Accepted Date: April 11, 2026

Published Date: April 30, 2026

Citation: Perabattula Ramesh, Rayudu Venkata Satya Sai Kumar, Kandabhattu S.S.V.K Satish Raju, Puppala Vamsi Surya Manikanta Kumar, Mahadasa Sri Swathi, Kusuma Rajasekhar. BVC EduPay—Smart College Fee Payment System. Journal of Open Source Developments. 2026; 13(1): 24–34p.

Conventional fee management methods require students to visit multiple departments, including academics, examinations, transport, hostel, and library, to understand their dues, confirm their eligibility, and complete payments. These fragmented systems create confusion and inefficiency, as students must navigate between

different counters to verify outstanding dues, submit receipts, and obtain clearance. The lack of integration among these departments often leads to miscommunication, delayed updates, human errors, and difficulties in retrieving historical payment data. These limitations highlight the need for a centralized, streamlined, and transparent digital platform for fee management.

To overcome these challenges, BVC EduPay was developed as a unified web-based system that integrates all fee-related services into a single platform. The primary objective of BVC EduPay is to simplify college fee payments and improve user convenience by providing a centralized environment where students can view fee details, make secure payments, track their payment history, and receive instant confirmation. Through this platform, students can access information regarding academic fees, examination fees, transport charges, hostel dues, library fines, and other institutional payments in an organized and intuitive manner.

The system provides a user-friendly environment that allows users to complete fee payments based on real-time eligibility, dynamically calculated fee structures, and secure online transactions. By integrating multiple fee-related processes into one application, BVC EduPay eliminates the need for physical visits to administrative blocks and reduces the dependency on manual verification. This integration improves transparency, enhances user satisfaction, and helps both students and staff make informed decisions during fee management.

From a technical perspective, BVC EduPay follows a robust full-stack architecture built using the MERN stack, which integrates React for frontend development, Node.js and Express.js for backend processing, and MongoDB for efficient data management. Razorpay was incorporated as a secure payment gateway to handle online transactions with real-time verification. The frontend ensures a responsive and interactive user experience, whereas the backend manages business logic, role-based authentication, payment workflows, and communication with the database. MongoDB stores all essential records, including student profiles, fee structures, transaction details, department-wise dues, and payment histories.

This architectural approach ensures scalability, security, and maintainability as the number of users and transactions increases. Security mechanisms, such as JWT-based authentication, data encryption, and server-side payment validation, safeguard sensitive user and financial information. The modular structure also allows for future enhancements, including analytics dashboards, notification systems, and mobile application integration.

Overall, BVC EduPay aims to enhance the digital fee payment experience by offering a comprehensive, transparent, and efficient platform that supports seamless financial transactions, reduces the administrative burden, and promotes streamlined coordination across various college departments.

LITERATURE SURVEY

The development of modern web-based financial management systems has been significantly shaped by established software engineering principles, structured development models, and architectural best practices. Pressman and Maxim highlighted that systematic software engineering approaches contribute to building reliable, scalable, and maintainable software applications, especially in domains requiring high accuracy and data consistency, such as fee management systems [1]. Similarly, Sommerville emphasized that well-defined software architectures improve system performance, maintainability, and long-term sustainability, which are essential for applications that must process large volumes of transactions and sensitive financial data [2].

These foundational principles are particularly relevant to the development of centralized digital fee payment platforms, such as BVC EduPay.

Existing fee management systems in educational institutions often rely on manual or semi-digital processes that lack integration across departments, such as academics, transport, examinations, and hostel administration. Multiple studies have pointed out that fragmented systems result in inefficiencies, delays, and data inconsistencies. Fowler discusses the importance of enterprise architectural patterns that promote modularity and unify diverse services within a single platform, thereby addressing the problems caused by isolated fee-processing systems [3]. Implementing these patterns enables seamless communication between institutional departments and ensures faster and more transparent financial workflows.

Modern web application development is strongly influenced by the evolution of client-server models and advances in dynamic web technology. JavaScript is a fundamental technology for building interactive and responsive user interfaces. Flanagan highlights that JavaScript's flexibility and widespread adoption make it a core component of modern web solutions [4], whereas Crockford emphasizes that efficient design principles in JavaScript development significantly enhance application performance and user experience [5]. These characteristics are crucial for systems such as BVC EduPay, where users depend on real-time updates and smooth fee-processing interfaces.

The introduction of component-based frontend frameworks has further simplified the development of interactive web platforms. React allows developers to construct modular, reusable UI components and manage complex state-driven interactions efficiently. Its virtual document object model (DOM) capabilities support faster rendering and improved responsiveness, making it an ideal choice for applications that require frequent updates, such as fee status checks, transaction notifications, and dashboard analytics [6].

On the backend, technologies such as Node.js and Express.js have been widely adopted to implement scalable, event-driven server-side applications. Their non-blocking architecture enables the concurrent processing of multiple client requests, which is essential for handling high transaction loads in financial management systems. Modern backend design relies heavily on REST-based architecture. Fielding's REST principles remain fundamental to creating distributed and maintainable web services, enabling smooth interaction between the frontend and backend layers of systems such as BVC EduPay [7, 8].

Data management plays an equally important role in web application development. As educational institutions handle large sets of structured and semi-structured fee-related data, robust database technologies are required to manage them.

MongoDB, a NoSQL database, provides a flexible schema design and efficient storage for evolving data models. Silberschatz, Korth, and Sudarshan highlight the importance of efficient data retrieval, indexing, and integrity in database systems, concepts that directly apply to storing student information, fee records, and transaction histories [9].

Secure and reliable digital payment integration is another critical aspect of modern financial systems. Payment gateways, such as Razorpay, implement encryption, transaction validation, and secure APIs to ensure reliable online payments. Industry research emphasizes the importance of secure communication channels, identity verification, and tamper-proof transaction workflows to protect sensitive user data and prevent financial fraud [10, 11]. These mechanisms form the backbone of secure fee-processing in BVC EduPay.

Furthermore, service-oriented and modular system designs remain central to building scalable enterprise applications. Erl explains that service-oriented architecture (SOA) supports the integration of multiple services in a flexible, interoperable environment, allowing systems to scale and evolve as institutional needs grow [12]. This approach is highly relevant for a fee management system that must integrate academic departments, administrative modules, and third-party payment gateways.

The web standards defined by the World Wide Web Consortium (W3C) for HTML5, CSS, and accessibility ensure consistent user experiences across devices and platforms. These standards guide developers in creating responsive, accessible, and reliable user interfaces suitable for students, staff, and administrators using various devices [13].

Overall, the literature indicates that modern fee management systems benefit significantly from integrated architectures, secure payment mechanisms, dynamic user interfaces, and scalable backend frameworks.

These findings collectively reinforce the need for a centralized, robust, and user-friendly system, such as BVC EduPay, to address the limitations of traditional fee payment processes.

PROPOSED SYSTEM

The proposed system, BVC EduPay, is designed to provide a centralized digital platform for managing college fee payment activities. The system addresses the limitations of traditional manual processes by integrating various fee-related components, such as academic fees, examination fees, hostel charges, transport fees, and library dues, into a single unified system. By combining these services, the platform simplifies the overall fee payment process and allows students to complete transactions efficiently without having to visit multiple departments.

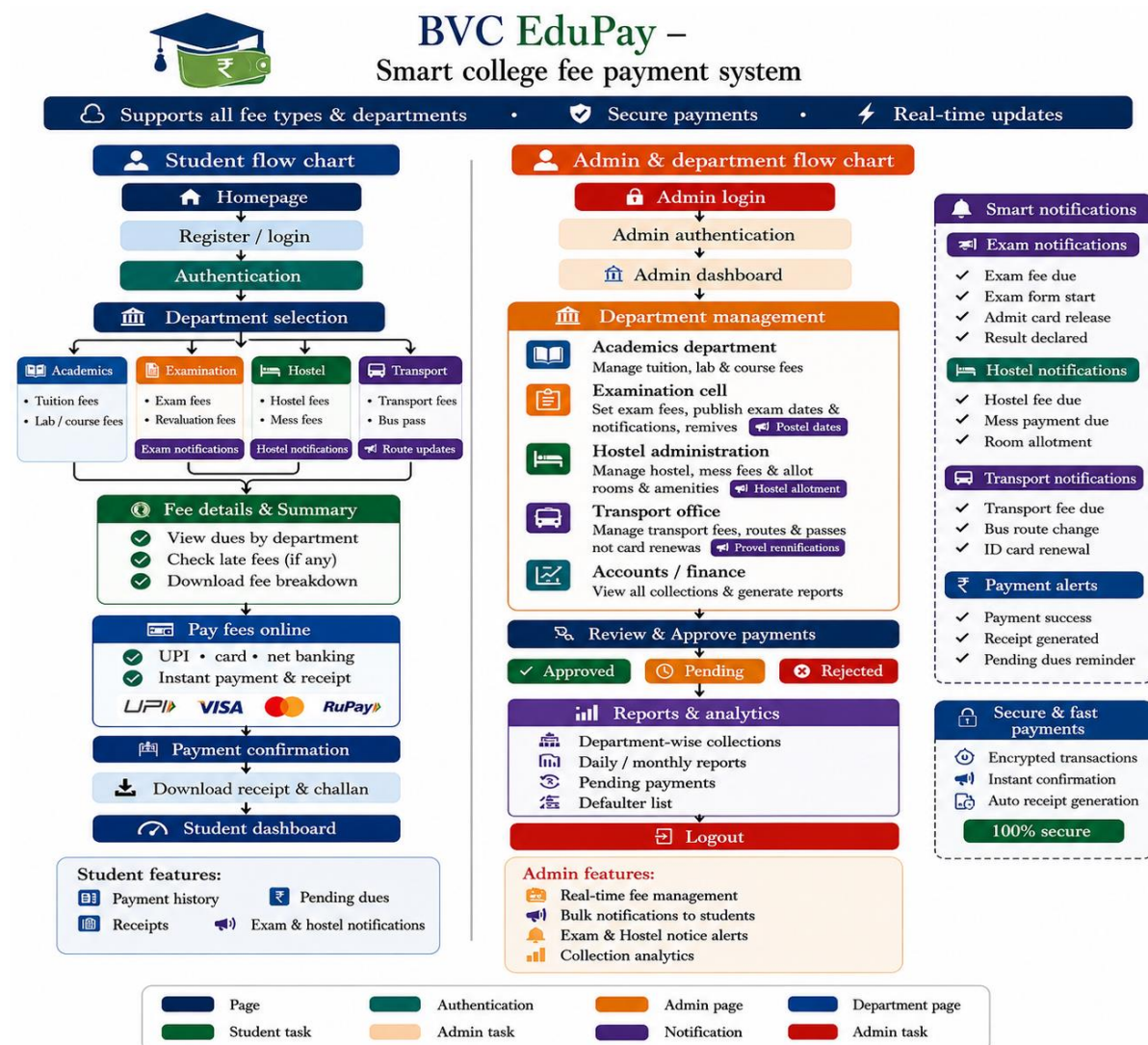


Figure 1. Flow chart.

BVC EduPay was developed using modern web technologies and follows a modular and scalable architecture. The system supports secure interactions between students and administrators while ensuring reliable data management. Each component of the system performs specific tasks, such as fee calculation, payment processing, and verification, while communicating smoothly with other modules. This structure ensures efficient service delivery and enhances the overall transparency and usability of the fee management process (Figure 1).

BVC EduPay System Architecture

The BVC EduPay application follows a three-tier architecture that separates the system into three distinct layers: frontend, backend, and database. This architectural approach improves system organization, simplifies maintenance, and enhances scalability by separating the presentation logic, application logic, and data management.

Frontend Layer

The frontend layer is responsible for providing an interactive and user-friendly interface through which students and administrators interact with the system. This layer was developed using React.js, a modern JavaScript library used to build dynamic and responsive web interfaces. React allows the creation of reusable UI components and supports efficient rendering, making the application fast and intuitive to use.

Through the frontend interface, students can view their academic fees, examination fees, hostel and transport charges, library dues, and payment history. They can also receive notifications regarding fee deadlines, exam fee announcements, and payment confirmations. Administrators can access department dashboards, verify payments, update the fee structures, and manage user roles. The use of React ensures a smooth and consistent experience across different devices and browsers.

Backend Layer

The backend layer handles the core application logic and manages communication between the frontend and database. The backend of the BVC EduPay system was implemented using Node.js and Express.js, which provide a powerful environment for building scalable server-side applications.

This layer processes student requests, manages role-based authentication and authorization, calculates fee amounts, validates payment details, and interacts with the Razorpay payment gateway for secure transactions. The backend also coordinates departmental workflows, such as exam fee approvals, hostel fee verifications, and transportation registration updates. The modular nature of Node.js and Express.js ensures efficient development, easy debugging, and smooth integration with external APIs and other services.

Database Layer

The database layer is responsible for storing and managing all the data required for system operations. The BVC EduPay system uses MongoDB, a NoSQL document-oriented database that supports flexible data storage and rapid retrieval.

The database stores various types of information, including:

- Student registration and profile details
- Department-wise fee structures
- Academic, exam, hostel, library, and transport fee records
- Payment transactions and Razorpay responses
- Admin and department head roles
- Notifications and payment status updates
- Fee receipts and payment history

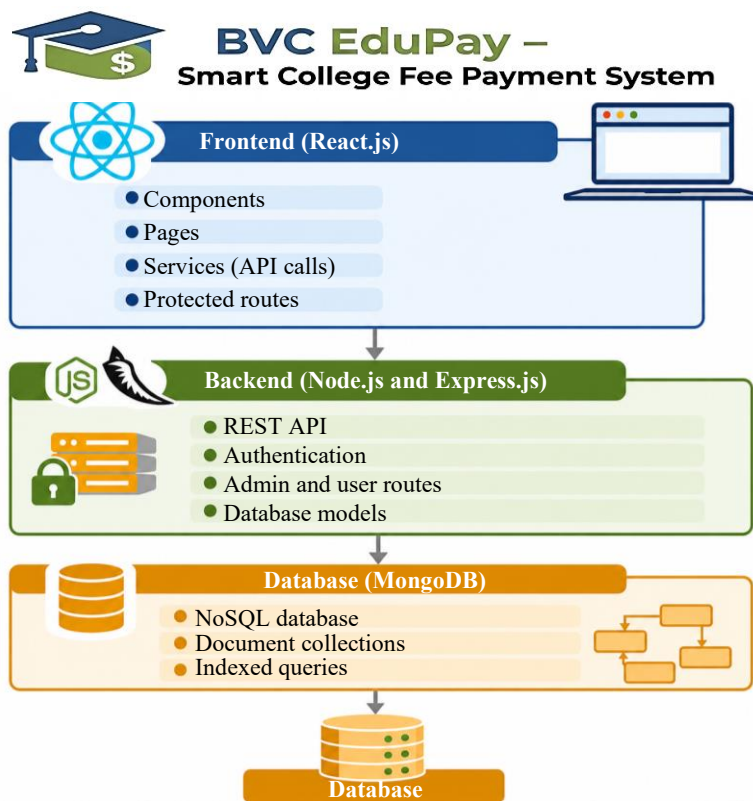


Figure 2. Project architecture.

Using a flexible database structure, the system maintains data integrity, ensures fast queries, and efficiently handles large volumes of fee-related data. This layered architecture ensures that BVC EduPay remains modular, scalable, and secure, allowing developers to maintain and extend the system easily in the future (Figure 2).

System Design

The BVC EduPay system is designed with multiple functional modules that manage different aspects of student fee management and academic recordkeeping. The two primary modules of the system are the Student and Admin Modules. These modules interact with the system architecture to perform various operations related to student payment and administration.

Student Module

The student module allows students to access the system and manage their fee payments and academic details as per their needs. This module provides an intuitive interface through which students can interact with the platform and perform the multiple operations required for educational management.

Students must first register and log in to the system using secure authentication. Once logged in, they can view pending fees for tuition, library, and other services. The system provides detailed information about the fee structure, enabling students to choose the appropriate payment methods.

After selecting a payment method, students can complete online transactions through unified payments interface (UPI), net banking, or card payments. The system also allows students to download fee receipts, track their payment history, and view their academic progress.

Once all actions are completed, the system automatically updates the student account. Students can also submit feedback regarding the fee payment process, which helps improve system quality and reliability.

Admin Module

The admin module provides administrative control over the system and allows system administrators to manage student- and fee-related information efficiently. Administrators play a critical role in maintaining the accuracy and availability of fee records and student details on the platform.

Through the admin interface, administrators can manage student profiles, including enrollment details, courses, and personal information of the students. They can also add, update, or remove fee structures and payment options to ensure that students have access to current and accurate information.

In addition, administrators can monitor fee payments, generate reports, and track pending or completed transactions. The admin module also allows administrators to review student queries and requests, ensuring proper validation and timely resolutions.

Fee Management Workflow

The Fee Management Workflow defines the step-by-step process through which students complete their payments and administrators manage financial records using the BVC EduPay platform. This structured workflow simplifies the fee payment process and ensures that students complete all the required steps before finalizing payments.

The workflow begins when a student logs into the system using valid authentication credentials to access the system. After accessing the system dashboard, students view their pending fees for the current semester. The student then chooses their preferred payment method to complete the transaction.

Once the payment is made, the system updates the student account and generates an official receipt. Subsequently, administrators can verify transactions, update records, and generate reports as required. Students can also provide feedback on the payment process, which can contribute to system improvement.

RESULTS AND DISCUSSION

The BVC EduPay system was successfully implemented using modern web technologies, including React.js for the frontend, Node.js/Express for the backend, and MySQL for the database. The system was tested in a web environment to evaluate its functionality, usability, and efficiency in managing fee transactions and student record management.

The results demonstrate that the platform effectively integrates multiple student and administrative services into a single system, thereby simplifying the overall fee management process.

The implemented system provides several functional modules, including student registration, login authentication, fee overview, payment processing, fee record management, and administrative control. Each module was tested to verify the correct interaction between the frontend interface, backend application logic, and database operations.

Student Registration and Authentication

The system provides a secure student registration interface that allows new users to create an account by entering personal details, such as name, email, and student ID. Once registered, the students can log in to the system using their credentials. The login module verifies student information through backend authentication processes before granting access to the fee management interface.

This authentication mechanism ensures that only authorized students can access the system features and submit fee payments. The results indicate that the login and registration modules operate efficiently with appropriate validation and error handling.

Fee Overview Interface

After a successful login, students can access the fee overview interface, where pending and paid fees are displayed in an organized manner. The interface allows students to browse detailed information about tuition, library, and other fees. Each fee category provides relevant details to help students effectively plan their payments.

The system dynamically retrieves fee data from the database through backend APIs, ensuring that students receive accurate and up-to-date information. This feature improves usability and reduces the time required for fee reviews.

Payment Processing and Receipt Generation

The core functionality of the system is the payment processing module, where students can select payment methods and complete their transactions. Once a payment is made, the system automatically updates the student account and generates a fee receipt.

The system displays a structured summary of the payment, allowing students to review the details before confirmation. This automated processing improves accuracy and helps students make informed financial decisions.

Fee Record Management

After completing a payment, students can view updated records on their dashboard under the “My Dashboard” section. Students can track their payment history, view outstanding fees, and download the receipts.

The dashboard provides a convenient way to manage financial records and review the previously submitted transactions. The system also provides confirmation messages upon successful payment processing.

Administrative Control Panel

The system includes an admin dashboard that enables administrators to manage the platform’s overall operation. Administrators can monitor registered students, manage the fee structure, review payments, and generate reports.

Through the admin interface, administrators can track pending payments, approve student requests, and manage notifications about upcoming fee deadlines. This centralized administrative control ensures that student and financial data remain accurate and up-to-date.

Student Feedback and System Completion

At the end of the payment process, students are provided with a feedback option, where they can submit reviews about their experience with the system. The system also includes a confirmation interface displaying a “Thank You for Using BVC EduPay” message after a successful payment submission.

The feedback feature allows administrators to analyze student satisfaction and to improve system services over time.

System Performance Evaluation

The testing results indicate that the BVC EduPay platform successfully integrates student and administrative operations into a single workflow. The system demonstrates efficient communication between the frontend, backend, and database layers, ensuring smooth data retrieval and processing (Figure 3).

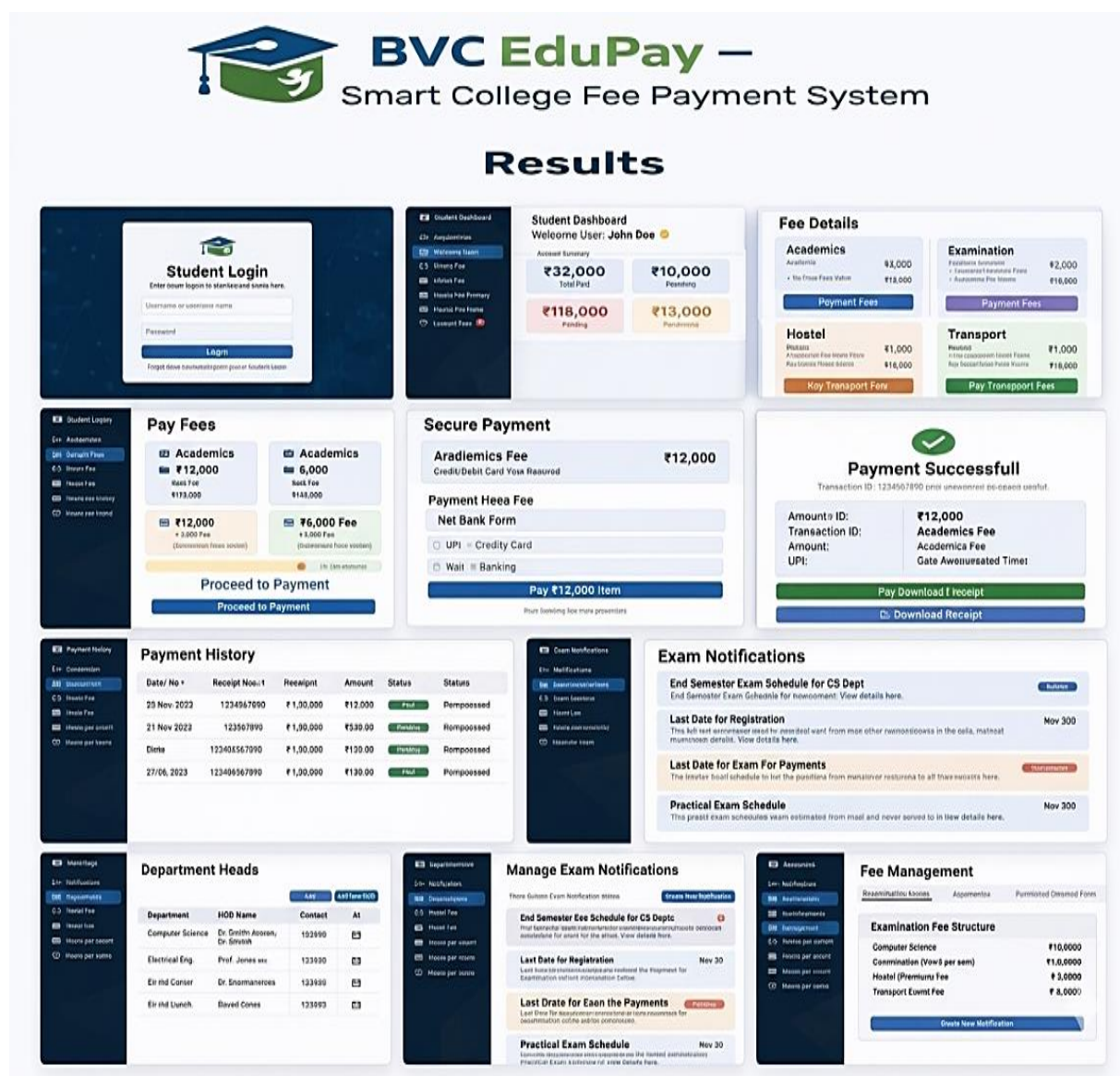


Figure 3. Results.

APPLICATIONS

- *Online fee management:* Helps students view and pay tuition, library, and other fees through a single platform.
- *Educational institution platforms:* Useful for colleges and universities to manage student financial data digitally.
- *Student record management:* This allows administrators to maintain accurate student profiles, course enrollments, and payment histories.
- *Administrative management:* It helps administrators track payments, monitor defaulters, and generate reports efficiently.
- *Smart payment calculation:* Automatically updates fee records and generates receipts upon successful transactions.
- *Student feedback services:* Enable students to provide feedback on payment processes and system usability.
- *Notifications and reminders:* Sends alerts about upcoming fee deadlines and pending payments to students.
- *Educational and research projects:* It can be used as a case study for full-stack web development

learning.

- *Mobile fee applications*: These can be extended into Android or iOS applications for mobile fee payment and management.
- *Analytics and reporting tools*: Provide insights into student payment patterns and administrative performance.

CONCLUSION AND FUTURE SCOPE

Conclusion

The BVC EduPay system was developed to provide an integrated and user-friendly platform for efficient student fee management and recordkeeping. Traditional fee management methods often require students to visit multiple offices or use offline systems, which increases complexity and may lead to delayed or inaccurate recordkeeping.

The proposed BVC EduPay platform addresses these challenges by integrating multiple student and administrative services into a single, centralized system. The application allows students to register, log into the system, view pending fees, select payment methods, complete transactions, download receipts, and submit feedback using a structured workflow. The system automatically updates the fee records and generates confirmations, thereby improving efficiency and transparency.

The system was successfully implemented using React.js for the frontend interface, Node.js/Express for backend processing, and MySQL for database management. The three-tier architecture ensures proper separation between the presentation, application logic, and data storage layers, thereby enhancing system scalability, security, and maintainability.

The experimental results demonstrate that the BVC EduPay platform simplifies fee management and provides students with a convenient digital environment for payment handling. The system also provides an administrative dashboard that allows administrators to manage student profiles, monitor transactions, and maintain overall operational efficiencies.

Overall, the BVC EduPay system provides a comprehensive and reliable solution for digital student fee management and academic recordkeeping, improving both user experience and administrative efficiency.

Future Scope

Although the current version of BVC EduPay provides essential fee management functionalities, several enhancements can be implemented in the future to improve its performance and usability. One possible improvement is the integration of online payment gateways that support multiple banking options, enabling real-time fee transactions.

Another enhancement could be the implementation of AI-based notifications and recommendations that remind students of upcoming deadlines, suggest optimal payment plans, and highlight pending dues. The system can also be extended to support mobile applications for Android and iOS platforms, enabling students to access fee management services on smartphones, thereby improving accessibility and convenience.

Furthermore, advanced analytics and reporting tools can be included for administrators to monitor student payment trends, track defaulters, and optimize resource planning. With these enhancements, BVC EduPay has the potential to evolve into a fully intelligent and comprehensive student management system that supports a wide range of academic and financial operations.

Funding Information

This project did not receive any specific financial support or grants from any funding agency in the public, commercial, or non-profit sectors. This work was carried out as part of the academic requirements for the Bachelor of Technology degree.

Ethics Statement

This study did not involve human or animal testing. The project is a software-based educational payment management system developed for academic and research purposes. Therefore, ethical approval was not required for this study.

Statement of Conflict of Interests

The authors declare no conflicts of interest regarding the development and publication of this project. This work was conducted solely for academic purposes.

REFERENCES

1. Pressman RS. Software Engineering: A Practitioner's Approach. 6th ed. New York (NY): Palgrave Macmillan; 2005.
2. Sommerville I. Software Engineering. 10th ed. Harlow: Pearson Education; 2016. p. 1–808.
3. Fowler M. Patterns of Enterprise Application Architecture. Boston (MA): Addison-Wesley Professional; 2012.
4. Flanagan D. JavaScript: The Definitive Guide. 6th ed. Sebastopol (CA): O'Reilly Media; 2011.
5. Crockford D. JavaScript: The Good Parts. Sebastopol (CA): O'Reilly Media; 2008.
6. Nikolov N. Modern technologies for building graphical user interfaces on the Internet. HR Technol. 2023;2:90–104.
7. Grinberg M. Flask Web Development. 2nd ed. Sebastopol (CA): O'Reilly Media; 2018.
8. Deshpande Y, Hansen S. Web engineering: creating a discipline among disciplines. IEEE Multimed. 2001;8(2):82–87. doi:10.1109/93.917974.
9. Silberschatz A, Korth HF, Sudarshan S. Database System Concepts. 4th ed. New York (NY): McGraw-Hill; 2002.
10. Kovvuri V. Optimizing real-time web applications in 2025: a performance and scalability study of Node.js backend with Angular frontend architectures. J Data Anal Crit Manag. 2025;1(02):41–44.
11. Widenius M, Axmark D. MySQL Reference Manual: Documentation from the Source. Sebastopol (CA): O'Reilly Media; 2002.
12. Sprott D, Wilkes L. Understanding service-oriented architecture. Architect J. 2004;1(1):10–17.
13. Sikos LF. Web standards: mastering HTML5, CSS3, and XML. New York (NY): Apress; 2014.