

# **PSTU PGS Management System**

**Submitted by:**

Ziad

Registration No: 09594  
Student ID: 2002077

**Supervised By:**

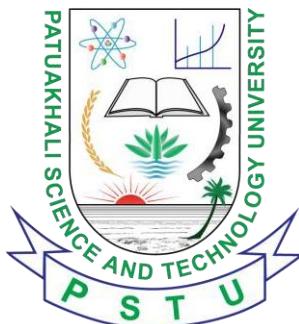
Dr. Md Samsuzzaman

Professor

Department of Computer and Communication Engineering

Faculty of Computer Science and Engineering

A PROJECT IN THE FACULTY OF COMPUTER SCIENCE  
AND ENGINEERING PRESENTED IN PARTIAL  
FULFILLMENT OF THE REQUIREMENTS FOR THE  
DEGREE OF BACHELOR OF SCIENCE IN COMPUTER  
SCIENCE AND ENGINEERING



**Faculty of Computer Science and Engineering  
Patuakhali Science and Technology University**

Patuakhali, Bangladesh

February 2026

## DECLARATION

This is to certify that the work presented in this project, titled “PSTU PGS(Postgraduate Studies) Management System”, is the outcome of the investigation and research carried out by us under the supervision of Prof. Dr. Md. Samsuzzaman.

It is also declared that neither this project nor any part thereof has been submitted anywhere else for the award of any degree, diploma, or other qualifications.

.....  
Ziad  
2002077

## **CERTIFICATION**

This project, titled “PSTU PGS Management System,” submitted by Ziad (ID: 2002077), has been accepted as satisfactory in partial fulfillment of the requirements for the Degree of Bachelor of science in Computer Science and Engineering degree, in February 2026.

Approval of the Supervisor

---

**Dr. Md. Samsuzzaman**

Professor

Department of Computer and Communication Engineering

Faculty of Computer Science and Engineering

## **Board of Examiners**

Signature : \_\_\_\_\_

Name of the Supervisor : Dr. Md.Samsuzzaman  
Professor, Department of Computer and Communication Engineering

Signature : \_\_\_\_\_

Chairman of the CCE & Internal Member: Golam Md. Muradul Bashir  
Professor, Department of Computer and Communication Engineering

Signature : \_\_\_\_\_

Name of Internal Member : Sarna Majumder  
Associate Professor, Department of Computer and Communication Engineering

Signature : \_\_\_\_\_

Name of Internal Member : Arpita Howlader  
Assistant Professor, Department of Computer and Communication Engineering

Signature : \_\_\_\_\_

Name of External Member : Dr. Md. Abdul Masud  
Professor, Department of Computer Science and Information Technology

## **ACKNOWLEDGEMENTS**

First and foremost, I express my heartfelt gratitude to the Almighty for granting me the strength, patience, and determination to complete this project successfully.

I would like to extend my deepest appreciation to my supervisor, Prof. Dr. Md. Samsuzzaman, for his invaluable guidance, continuous support, and insightful feedback throughout the development of this project. His encouragement and expertise have been instrumental in shaping the direction and quality of this work.

I am also thankful to the faculty members of the Faculty of Computer Science and Engineering, Patuakhali Science and Technology University, for equipping me with the foundational knowledge and technical skills necessary to undertake this project.

Finally, I am deeply grateful to my family and friends for their constant encouragement, emotional support, and understanding throughout the entire journey.

With Best Regards,  
Ziad

## **Abstract**

This project is a significant component of the PSTU PGS Management System, which has been specifically designed and developed for managing Postgraduate Studies (PGS) at Patuakhali Science and Technology University (PSTU). The primary objective of this system is to digitize, secure, and automate the postgraduate admission process of the university.

The system includes multiple highly secured panels, mainly the Admin Panel and Users Panel. As the current phase of the project focuses on the admission application process, only the panels required for this workflow have been implemented. The Users Panel currently consists of separate role-based panels for Supervisor, Chairman, and Dean, all of which are fully secured and access-controlled.

To ensure system integrity and security, all critical user accounts are created and managed by the Admin. The replacement of officials such as Chairman or Dean after the completion of their tenure is also handled by the Admin, which significantly reduces the possibility of role mismatches and data inconsistencies within the system. Furthermore, the authentication mechanism of these panels is highly secure and includes OTP-based verification, particularly for password recovery and additional access protection.

Through this system, applicants can easily submit their admission applications online. One of the most challenging parts of the admission process application fee payment is simplified through an integrated online payment mechanism, eliminating unnecessary hassle. Applicants receive real-time updates via email notifications regarding application status, including approval, rejection, or placement on a waiting list. Once the application is finally approved, the applicant will be able to complete the enrollment process within the same system, which is currently under development.

Overall, the PSTU PGS Management System provides a secure, efficient, and scalable solution that enhances transparency, reduces manual workload, and modernizes the postgraduate admission process of the university.

# Table of Contents

DECLARATION .....	i
CERTIFICATION .....	ii
Board of Examiners .....	iii
ACKNOWLEDGEMENTS .....	iv
Abstract.....	v
<b>Chapter 1: Introduction.....</b>	<b>5</b>
1.1 Background .....	5
1.2 Problem Statement.....	5
1.3 Objectives.....	6
1.4 Motivation.....	7
1.5 Summary .....	8
<b>Chapter 2: Literature Review .....</b>	<b>9</b>
2.1 Introduction .....	9
2.2 Existing Systems .....	9
2.3 Comparison with PSTU PGS Management System .....	10
2.4 Summary .....	12
<b>Chapter 3: Methodology .....</b>	<b>13</b>
3.1 Introduction .....	13
3.2 Fact-Finding Techniques.....	13
3.3 Software Development Life Cycle (SDLC).....	14
3.3.1 Software Process Model.....	15
3.3.2 Feasibility Study .....	15
3.4 Database Design.....	16
3.5 Key Collections and Their Schemas .....	16
3.5.1 Department Schema.....	16
3.5.2 Admin Schema .....	16
3.5.3 OTP Schema .....	17
3.5.4 Academic Record Schema.....	17
3.5.5 Applied Subject Course Schema .....	17
3.5.6 Approval Log Schema.....	17
3.5.7 Admission Application Schema.....	18
3.5.8 Admission Payment Schema .....	19
3.5.9 Admission Season Schema .....	19

3.5.10 Department Registration Range Schema.....	19
3.5.11 Notice Schema .....	20
3.5.12 Student Enrollment Schema .....	20
3.5.13 User Schema .....	20
3.5.14 User Tenure Schema .....	21
3.6 Database ER Diagram.....	22
3.6.1 Key Entities.....	23
3.7 Summary .....	25
<b>Chapter 4: PSTU PGS Management System Design .....</b>	<b>26</b>
4.1 Introduction .....	26
4.2 Overall System Design.....	26
4.2.1 System Architecture .....	27
4.2.2 Infrastructure Services .....	32
4.2.3 Four-Tier Architectural Diagram .....	33
4.2.4 User Activity Diagram.....	34
4.2.5 Data Flow Diagram.....	39
4.3 Application Features .....	43
4.3.1 Admin And User Management.....	43
4.3.2 Admission Season And Range Management .....	43
4.3.3 Admission Application Management .....	43
4.3.4 Academic Record And Document Management .....	44
4.3.5 Payment Processing.....	44
4.3.6 Application Review And Approval Workflow .....	44
4.3.7 Notification and Communication System.....	45
4.3.8 Security And Data Integrity.....	45
4.3.9 Notice Management.....	45
4.4 User Interface (UI) Design .....	46
4.4.1 Home Page .....	46
4.4.2 Login Page.....	47
4.4.3 Recover Password Page .....	47
4.4.4 Admin Dashboard.....	49
4.4.5 Application Form .....	52
4.5 Implementation.....	55
4.5.1 Backend.....	55
4.5.2 Frontend.....	58

<b>Chapter 5: Testing &amp; Security .....</b>	60
5.1 Introduction .....	60
5.2 Security Strategies.....	60
5.2.1 Authentication and Authorization.....	60
5.2.2 Sandboxed Code Execution.....	60
5.2.3 Rate Limiting and API Throttling .....	61
5.2.4 Data Encryption.....	61
5.3 XP Practices .....	61
5.3.1 Test-Driven Development(TDD) .....	62
5.3.2 Continuous Integration (CI).....	62
5.3.3 XP Testing .....	62
5.3.4 Unit Testing.....	63
5.3.5 Integration Testing .....	63
5.3.6 End-to-End (E2E) Testing.....	63
5.4 Summary .....	64
<b>Chapter 6: Conclusion .....</b>	65
6.1 Introduction .....	65
6.2 Project Outcomes.....	65
6.3 Future Works .....	66
6.4 Summary .....	67
<b>References.....</b>	68

# List of Figures

3.1	Database ER Diagram .....	22
4.1	System Architecture of PSTU PGS Management System.....	31
4.2	Code Execution Workflow .....	32
4.3	Code Execution In Four Layer.....	33
4.4	Authentication Workflow.....	34
4.5	Admin Create Users Workflow.....	35
4.6	Update Users Workflow.....	35
4.7	Academic Record Process Workflow.....	36
4.8	Document Upload Workflow.....	36
4.9	Payment Workflow.....	36
4.10	Application Submission Workflow.....	37
4.11	Selection And Approval Workflow.....	37
4.12	Notice Management Workflow.....	38
4.13	Level-0 Data Flow Diagram .....	40
4.14	Level-1 Data Flow Diagram .....	42
4.15	Home Page.....	46
4.16	Login Page.....	47
4.17	Send Mail For Getting OTP Page .....	48
4.18	OTP Set Page.....	48
4.19	Update Password Page.....	48
4.20	Admin Dashboard .....	49
4.21	Create New Department and Department List Page.....	49
4.22	Create User Page .....	50
4.23	User's List Page.....	50
4.24	Tenure History Page.....	50
4.25	Create Season And Department Range Page.....	51
4.26	Season's Details List Page.....	51
4.27	Create Notice Page.....	51
4.28	Application Form Fillup Page.....	52
4.29	Payment Page .....	53
4.30	Upload File Section.....	54

# **Chapter 1: Introduction**

## **1.1 Background**

At Patuakhali Science and Technology University (PSTU), the management of postgraduate admission processes and also entire full system has traditionally relied on manual and semi-manual methods, involving paperwork, spreadsheets, and fragmented record-keeping across different departments. Applicants, faculty members, and administrative personnel depend heavily on physical documents and email-based communication to process admission applications, which increases the risk of human error and makes it difficult to maintain accurate and up-to-date records. From collecting application forms and academic documents to verifying eligibility, approving applications, and handling payments, the absence of a centralized digital system has resulted in inefficiencies, delays, and limited transparency throughout the admission workflow.

In the existing manual system, each department and administrative unit maintains its own records, leading to data inconsistencies, redundancy, and communication gaps between stakeholders. For example, application approval decisions may be delayed due to missing documents or unclear status updates, and applicants often face uncertainty regarding payment confirmation, approval status, or waiting list placement. Without real-time tracking and automated notifications, monitoring application progress becomes challenging for both applicants and administrators, increasing workload and reducing accountability.

The traditional admission and post-enrollment management approach, while previously sufficient, is no longer adequate for the growing academic and administrative demands of PSTU. Therefore, there is a critical need for a secure, centralized, and automated postgraduate management system that ensures efficient application processing, role-based access control, real-time status updates, and reliable data management. Implementing such a system will enhance transparency, reduce manual effort, improve decision-making, and modernize the postgraduate management process of the university.

## **1.2 Problem Statement**

The existing manual and semi-digital postgraduate management process at PSTU faces several significant challenges that hinder its efficiency and reliability:

- Manual Record-Keeping:

The heavy reliance on paper-based documents, spreadsheets, and email communication increases the risk of human error. Incomplete forms, misplaced documents, and inconsistent data entries often result in discrepancies between application records and actual applicant information, making the admission process unreliable and difficult to manage.

- Inefficiencies in Application Processing:

Due to the absence of a centralized system, administrative staff and academic authorities must manually verify documents, track application progress, and coordinate approvals. This leads to unnecessary delays in application evaluation, payment confirmation, and decision-making, ultimately affecting applicants and institutional timelines.

- Time-Consuming and Hassle-Prone Admission Process:

During peak admission periods, both applicants and university staff experience significant time consumption and administrative hassle. Applicants often need to submit documents physically, follow up repeatedly for status updates, and wait extended periods for approvals, while staff must handle large volumes of applications manually, increasing workload and stress.

- Lack of Accessibility and Transparency:

Since application data and approval statuses are not managed through a unified digital platform, accessing real-time information becomes challenging. Applicants are often required to contact departments repeatedly to check their application status, while administrators face difficulties in monitoring progress across departments, reducing transparency and accountability.

- Difficulties in Reporting and Monitoring:

Generating reports related to applications, approvals, payments, and departmental performance is time-consuming and prone to errors in a manual system. The lack of automated reporting tools makes it difficult for university authorities to analyze admission trends, track workloads, and plan resources effectively.

In light of these challenges, there is an urgent need for a centralized, automated, and secure postgraduate management system at PSTU. Such a system would streamline application processing, improve transparency, reduce administrative workload, and support informed decision-making across all stakeholders involved in the postgraduate system.

### **1.3 Objectives**

The main objective of this project is to design and develop a digital, secure, and scalable Postgraduate Management System for Patuakhali Science and Technology University (PSTU), replacing the existing manual and semi-digital admission and post-admission processes. By introducing a centralized and automated solution, the system aims to:

- Automate the Admission Application Process:

Eliminate manual handling of admission applications by enabling applicants to submit application forms, academic information, and required documents online. This ensures accurate data collection, reduces human error, and speeds up the overall admission workflow.

- Centralized Data Management:  
Provide a single, centralized platform for managing all admission-related data, including applications, academic records, documents, payments, and approval statuses. Authorized users across different departments can access real-time and consistent information from the system.
- Implement Role-Based Access Control:  
Establish a secure role-based access mechanism for different users such as Admin, Supervisor, Chairman, and Dean. Each role is granted specific permissions to ensure controlled access, accountability, and data integrity throughout the admission process.
- Ensure Secure Authentication and Authorization:  
Enhance system security through OTP-based authentication and secure login mechanisms, particularly for administrative users, to protect sensitive data and prevent unauthorized access.
- Streamline Application Review and Approval:  
Facilitate a structured and transparent application review process by allowing supervisors, chairmen, and deans to evaluate applications, record decisions, and maintain approval logs within the system.
- Enable Online Payment and Notification Services:  
Integrate an online payment system to simplify application fee submission and provide real-time email notifications to applicants regarding application status, approval decisions, and waiting list updates.

## 1.4 Motivation

The motivation for developing the PSTU PGS Management System originates from personal observation and experience with the overall postgraduate (PGS) academic and administrative processes at Patuakhali Science and Technology University (PSTU). Currently, most postgraduate activities—starting from admission and continuing through enrollment, thesis submission, approval processes, stipend management, and other academic procedures—largely depend on manual documentation, email-based communication, and fragmented record management across different departments. As a result, applicants, departmental members, and administrative staff frequently face delays, confusion, and increased workload during the admission period as well as throughout the postgraduate lifecycle.

Through interactions with postgraduate students and involvement in academic and administrative activities, it became evident that the absence of a centralized and automated system creates multiple challenges. These include difficulty in tracking application and academic status, delays in approval and verification processes, and inefficiencies in handling documents, payments, and communication. Students often need to follow up repeatedly to receive updates, while administrators and academic authorities struggle to manage large volumes of data and processes within limited time frames.

These challenges highlight the need for a secure, integrated, and scalable digital solution capable of managing the entire postgraduate study lifecycle. Although the current development phase of this

project focuses primarily on the postgraduate admission process, it has been designed with the long-term goal of expanding into a complete PGS management platform. The motivation behind this project is to reduce administrative burden, minimize human error, enhance transparency, and provide a smooth and user-friendly experience for all stakeholders. By implementing the PSTU PGS Management System, the university can significantly improve operational efficiency and modernize its postgraduate academic and administrative management.

## 1.5 Summary

This chapter provided an overview of the background, problem statement, and objectives behind the development of the PSTU PGS Management System. It highlighted the limitations and inefficiencies of the existing manual and semi-digital postgraduate admission and academic management processes at Patuakhali Science and Technology University (PSTU), along with the challenges faced by students, faculty members, and administrative authorities. The chapter also outlined the key objectives of the proposed system, focusing on automation, security, role-based access control, transparency, and real-time information updates. Additionally, the motivation behind the project was discussed, emphasizing the need to replace fragmented and outdated processes with a centralized, secure, and scalable digital solution capable of managing the postgraduate study lifecycle. While the current phase of development primarily addresses the admission process, the system is designed with future expansion in mind to support enrollment, thesis management, stipend processing, and other postgraduate academic activities.

In the subsequent chapters, the system design, architecture, and technical implementation of the PSTU PGS Management System will be discussed in detail, providing insights into how the proposed solution addresses existing challenges and improves efficiency, transparency, and overall management of postgraduate studies at the university.

# **Chapter 2: Literature Review**

## **2.1 Introduction**

This chapter reviews existing postgraduate admission and academic management systems and their features related to online application processing, user authentication, document management, payment handling, and approval workflows. It examines the strengths and limitations of current systems used by universities and academic institutions both locally and globally, highlighting the challenges faced by administrators, faculty members, and students in managing postgraduate studies efficiently.

Furthermore, this chapter explores how many existing systems struggle with issues such as manual or semi-automated processes, lack of centralized data management, limited real-time updates, and inadequate role-based access control. These limitations often result in delays, reduced transparency, and increased administrative workload during critical academic processes such as admissions, approvals, and enrollment.

By comparing these existing solutions with the design goals of the PSTU PGS Management System, this chapter illustrates how the proposed system aims to address these gaps by providing a secure, centralized, and user-friendly platform. The system is specifically tailored to meet the postgraduate study management requirements of Patuakhali Science and Technology University (PSTU), with a focus on automation, security, scalability, and improved communication among all stakeholders.

## **2.2 Existing Systems**

Over the years, various postgraduate admission and academic management systems have been developed to assist universities in handling student applications, approvals, and academic workflows. These systems differ significantly in terms of functionality, scalability, security, and integration capabilities. Below is an overview of some existing systems and approaches relevant to this project:

### **1. Manual and Spreadsheet-Based Admission Systems: Simple but Error-Prone**

The most commonly used approach in many universities, including PSTU, is managing postgraduate admission processes through manual documentation, spreadsheets, and email communication. While this method allows basic record-keeping, it is highly prone to human error, data inconsistency, and document loss—especially when handling a large number of applications. It lacks centralized data management, real-time status tracking, automated notifications, and secure access control, making the admission process inefficient and time-consuming.

## **2. Legacy University Admission Systems: Traditional but Inflexible**

Some institutions use older desktop-based or locally hosted admission systems developed years ago. These systems often support limited automation and require manual intervention for application verification, approvals, and updates. Due to outdated architectures, they offer poor scalability, limited integration with payment gateways or notification services, and are not user-friendly for modern web-based access. Customization and maintenance of such systems can also be challenging.

## **3. Cloud-Based Admission Platforms: Accessible but Not Fully Tailored**

Several cloud-based admission and student management platforms provide online application submission, basic tracking, and reporting features. These systems offer better accessibility and real-time data access compared to manual methods. However, most of them are designed as generic solutions and lack customization for specific postgraduate workflows, such as supervisor assignment, multi-level approval processes, tenure-based role management, and institution-specific policies like those followed at PSTU.

## **4. Open-Source University Management Systems: Flexible but Complex**

Open-source academic management solutions provide customizable features for handling admissions, academic records, and user management. While these systems offer flexibility and cost advantages, they often require significant technical expertise to configure, deploy, and maintain. The complexity of setup and ongoing maintenance can make them impractical for institutions without dedicated technical teams, particularly when institution-specific postgraduate rules and workflows must be implemented.

## **5. Enterprise Resource Planning (ERP) Systems: Comprehensive but Overly Complex**

Large universities sometimes adopt full-scale ERP solutions that include admission and academic management as part of a broader system. Although these systems provide robust functionality, security, and reporting capabilities, they are often expensive, complex to implement, and require extensive training. For institutions like PSTU, many ERP features may be unnecessary for postgraduate admission management, making such solutions costly and inefficient for targeted needs.

### **2.3 Comparison with PSTU PGS Management System**

The PSTU PGS Management System has been developed to directly address the limitations identified in existing postgraduate admission and academic management systems, particularly in the areas of application processing, approval workflows, security, and communication. Below is a comparison of key features between existing approaches and the proposed system:

## 1. Real-Time Application Tracking vs. Manual Status Updates:

In many existing systems, especially those relying on manual records or email-based communication, application status updates are handled manually, leading to delays and uncertainty for applicants. The PSTU PGS Management System provides real-time application tracking, allowing applicants and administrators to monitor the progress of applications instantly, from submission to final approval or waiting list placement.

## 2. Centralized Admission Management vs. Fragmented Record-Keeping:

Traditional admission processes often require different departments to maintain separate records, resulting in data inconsistencies and redundancy. In contrast, the PSTU PGS Management System centralizes all admission-related data, including applications, academic records, documents, payments, and approval logs, within a single platform. This centralized approach ensures consistency, transparency, and efficient coordination among departments.

## 3. Role-Based Access Control vs. Limited Security Mechanisms:

Many existing systems provide minimal access control, increasing the risk of unauthorized data access or modification. The PSTU PGS Management System implements strict role-based access control, ensuring that users such as Admins, Supervisors, Chairmen, Deans, Provosts, PGS Specialists, and Registrars can only access system functions relevant to their responsibilities. This significantly enhances system security and accountability.

## 4. Structured Approval Workflow vs. Informal Decision Processes:

In traditional admission systems, approval decisions are often communicated informally through emails or physical documents, which can be inefficient and difficult to track. The PSTU PGS Management System introduces a structured, multi-level approval workflow, enabling supervisors and academic authorities to record decisions digitally, maintain approval histories, and ensure transparency throughout the admission and enrollment processes.

## 5. Automated Notifications vs. Manual Communication:

Existing systems often rely on applicants to repeatedly follow up for updates. In contrast, the PSTU PGS Management System automatically sends real-time email notifications for key events such as application submission, payment confirmation, approval decisions, waiting list updates, and enrollment status, significantly improving communication and user experience.

## 6. Scalable Design vs. Limited Future Expansion:

Many current solutions are designed to handle only a specific phase of the admission process and are difficult to extend. The PSTU PGS Management System is designed with scalability in mind, allowing future expansion to include enrollment management, thesis submission, stipend processing, academic progress tracking, and other postgraduate academic activities.

## **2.4 Summary**

This chapter reviewed existing postgraduate admission and academic management systems, including manual and semi-digital processes, legacy university systems, cloud-based platforms, open-source solutions, and enterprise-level ERP systems, highlighting their strengths and limitations. While many of these systems provide partial automation and basic functionality, they often fall short in delivering centralized data management, real-time application tracking, secure role-based access control, structured approval workflows, and effective communication mechanisms.

By addressing these gaps, the PSTU PGS Management System offers a secure, scalable, and user-friendly solution tailored to the specific requirements of Patuakhali Science and Technology University (PSTU). The system introduces automated application processing, multi-level approval workflows, OTP-based authentication, online payment integration, and real-time notification services, ensuring transparency, efficiency, and accountability throughout the postgraduate admission process.

Furthermore, the proposed system is designed with future expansion in mind, enabling the integration of additional postgraduate academic functionalities such as enrollment management, thesis submission, stipend processing, and academic progress tracking. The following chapters will present the system design, architecture, and implementation details of the PSTU PGS Management System, demonstrating how it provides a modern, efficient, and secure approach to managing postgraduate studies at PSTU.

# **Chapter 3: Methodology**

## **3.1 Introduction**

The methodology for developing the PSTU PGS Management System is based on established software engineering principles and tailored to meet the academic and administrative requirements of Patuakhali Science and Technology University (PSTU). This chapter outlines the methods, tools, and processes applied throughout the system development lifecycle to ensure a structured, secure, and efficient solution for managing postgraduate admission and academic processes.

The selected methodology emphasizes comprehensive requirement analysis, systematic system design, modular implementation, and rigorous testing to develop a reliable and user-friendly management system. Particular attention has been given to security, role-based access control, data integrity, and scalability, as the system handles sensitive academic and administrative information.

By following recognized best practices in software development, this project aims to deliver a system that effectively supports the postgraduate admission workflow while remaining adaptable for future expansion to include enrollment management, thesis submission, stipend processing, and other postgraduate academic activities. The methodology is organized into distinct sections, including requirement gathering techniques, the Software Development Life Cycle (SDLC), database design, system architecture, and integration strategies, each contributing to the successful development of the PSTU PGS Management System.

## **3.2 Fact-Finding Techniques**

To develop an effective PSTU PGS Management System, various fact-finding techniques were employed during the requirements analysis phase. These techniques were essential for gaining a comprehensive understanding of the postgraduate admission and academic processes, as well as identifying the needs, expectations, and challenges of different stakeholders involved in the system.

In-depth discussions were conducted with postgraduate students, department chairmans, supervisors, and administrative at Patuakhali Science and Technology University (PSTU) to gather qualitative insights. These discussions enabled a detailed examination of the existing manual and semi-digital processes related to admission, application review, approval workflows, and communication. Through these interactions, critical pain points such as delays, lack of transparency, and difficulty in tracking application status were identified.

Additionally, observational studies were carried out to analyze real-world admission and administrative workflows during peak admission periods. This approach helped in understanding how applications are processed, how approvals are managed across departments, and how documents and payments are currently handled. By observing these processes directly, several inefficiencies and process gaps were identified that were not always apparent through interviews alone.

The insights gathered through these fact-finding techniques played a crucial role in defining system requirements, designing role-based workflows, and ensuring that the PSTU PGS Management System effectively addresses the practical challenges faced by Applicants, Supervisors, Department Chairman, and Administrative officers such as: Vice-Chancellor, Register, Hall-Provost and related other officers and staff.

### **3.3 Software Development Life Cycle (SDLC)**

The Software Development Life Cycle (SDLC) is a structured approach essential for the systematic development of the PSTU PGS Management System. The following phases outline the key steps involved in the SDLC:

1. **Planning**: In this initial phase, project goals, scope, and timelines were defined. Stakeholder meetings were conducted to outline the project's vision and ensure alignment with the needs of PSTU's PGS Office.
2. **Requirements Gathering**: This phase utilized the fact-finding techniques previously discussed to compile a comprehensive list of user needs. Interviews and focus groups were instrumental in capturing both quantitative and qualitative data on user expectations and PGS office's requirements.
3. **Design**: The design phase focused on creating the system architecture, user interfaces, and database schemas based on the gathered requirements. Emphasis was placed on ensuring a user-friendly experience for non-technical staff while maintaining robustness and security.
4. **Implementation**: During the implementation phase, the development of the backend and frontend was carried out. Modern technologies were employed to create a responsive, efficient, and secure platform. A RESTful API was developed for user management, application form process with payment and file upload system, review that application and organized according to merit by academic users and real-time updates.
5. **Testing**: The testing phase was critical for verifying that the system functioned as intended. It included:
  - Unit Testing: Testing individual components for functionality.
  - Integration Testing: Ensuring that different modules work together seamlessly.
  - User Acceptance Testing (UAT): Validating the system with actual users to confirm it meets their needs and expectations.
6. **Deployment**: In this phase, the system was deployed on the localhost remain but hopefully very soon will deployed in our University's local server, ensuring that all data related to inventory tracking remained secure and accessible only by authorized users.

7. Maintenance and Evaluation: After deployment, ongoing evaluation and maintenance processes were established to address user feedback, implement updates, and ensure that the system continued to meet the evolving needs of PSTU.

### **3.3.1 Software Process Model**

The software development process for the PSTU PGS(Postgraduate Studies) Management System follows a structured approach, starting from requirements analysis and design, leading to development, testing, deployment, and maintenance. The system utilizes React.js, React-Redux, Bootstrap, HTML and CSS for the frontend and Node.js, Express.js and MongoDB for the backend. This combination ensures a scalable, responsive, and secure automation system solution that aligns with the needs of PSTU.

### **3.3.2 Feasibility Study**

The feasibility study assesses the viability of the PSTU Inventory Management System by examining technical, operational, financial, and scheduling aspects:

1. Technical Feasibility:

The project utilizes a reliable technology stack, including React.js, Node.js, Express.js, MongoDB with Mongoose, and React-Redux(MERN Stack). The use of a RESTful API ensures scalability and modularity, making the project technically feasible. The stack's stability and strong community support ensure long-term sustainability and ease of future enhancements.

2. Operational Feasibility:

The system addresses the operational needs of PSTU by providing a web-based platform for inventory management. Features such as real-time stock updates, role-based access control, and automated report generation streamline the lab management and procurement processes for both instructors and administrators. The user-friendly interface ensures the system is easy to use, even for non-technical staff.

3. Financial Feasibility:

Development costs are minimized by using open-source tools and technologies. Currently, the system has been developed and tested in a local environment. For future deployment, it is planned to host the system on PSTU's existing server infrastructure, which will help reduce hosting and maintenance costs and ensure long-term sustainability.

#### **4. Schedule Feasibility:**

The project follows an iterative development model with well-defined milestones. The timeline ensures that resources are allocated efficiently, and risk management strategies are in place to meet project deadlines. The structured plan allows for timely completion, ensuring the system will be ready for deployment within the planned timeframe.

### **3.4 Database Design**

The database design is fundamental to ensuring data integrity, performance, and scalability for the PSTU PGS Automation/Management System. The system utilizes MongoDB as its NoSQL database, which is structured into collections tailored to meet the university's PGS automation and user management requirements. Below is a detailed explanation of the key collections and their schemas

### **3.5 Key Collections and Their Schemas**

#### **3.5.1 Department Schema**

The Department schema stores information about each department within the PGS Faculty in Patuakhali Science and Technology University. The fields include:

- name: The department's name.
- offeredSubjects: if multiple subjects offered by a specific department.
- Timestamps: Automatically stores the created and updated timestamps.

#### **3.5.2 Admin Schema**

The Admin schema represents who control the whole system and only one admin can stay at a time. The fields include:

- email: Email of the controller or Admin.
- firstName: First Part of the name of Admin.
- lastName: Last Part of the name of Admin.
- mobile: Mobile number of Admin.
- password: Set's password during registered at admin panel.
- photo: Set photo of admin but this is not mandatory.

Timestamps: Automatically stores the created and updated timestamps.

### **3.5.3 OTP Schema**

The OTP schema stores OTP number during forget password. The fields include:

- email: which email used at the admin/users created time.
- otp: Send otp number to that email.
- staaus: Just indicates that otp sends or not.If 1,means send otherwise not send.

Timestamps: Automatically stores the created and updated timestamps.

### **3.5.4 Academic Record Schema**

The Academic Record schema stores academic information of Applicants during application time.

The fields include:

- examLevel: Name of the exam's which applicant's need to pass with minimum GPA/CGPA.
- institution: Institution's name corresponding to that exam names.
- passingYear: Year of the pass that level's exam.
- cgpa/gpa: Total gpa/cgpa that exam's.
- cgpaScale: Which GPA/CGPA gathered by that applicant.
- isFinal: Last decision take that those information are ok and that stores.

Timestamps: Automatically stores the created and updated timestamps.

### **3.5.5 Applied Subject Course Schema**

The Applied Subject Course schema stores information about applied subject/Department's BSc/Hon's level's result(This Schema applied for MS/MBA/LLM students). The fields include:

- courseCode: BSc/BBA/LLB level's courses code.
- courseTitle: BSc/BBA/LLB level's courses title.
- creditHour: BSc/BBA/LLB level's total courses credit hours.
- gp\*ch: calculated results store here.

Timestamps: Automatically stores the created and updated timestamps.

### **3.5.6 Approval Log Schema**

The Approval Log schema stores the applicant's application approved or reject by specific users(Supervisor,Chairman and Dean). The fields include:

- role: The user's role.
- approvedBy: Approved by each User in every step.
- approvedByName,approvedBymail,ApprovedByRoleAtThatTime: These three fields are for

snapshot that is stack for specific application. This is used for if in future Chairman/Dean replace or change, that time this snapshot's details stack for that application.

- decision: Decision by User's Either Selected or Rejected or In waiting.
- remarks: Application Stage Justify.

Timestamps: Automatically stores the created and updated timestamps.

### **3.5.7 Admission Application Schema**

The Addmission Application schema stores all information about the applicant.

The fields include:

- program: Which Program(MS/MBA/LLM/PhD) choose by applicant for admission.
- admissionSeason: Admission Season Schema.
- academicYear: Wants to admit under which year(Ex: 2026).
- department: Select Department under selected Program.
- supervisor: Selected Supervisor by Applicant under selected Department.
- applicantName: The applicant's name.
- fatherName,motherName,dateOfBirth,nationality,maritalStatus,sex: The applicant's personal details.
- presentaddress,permanentAddress: Applicant's address details.
- mobile: Applicant's Contact Number.
- email: Applicant's Email address.
- academicRecords: Academic Record Schema(6)
- appliedSubjectCourses: Applied Subject Course Schema.
- totalCreditHourBachelor: Total bachelor level's credit hour.
- totalCreditHourAppliedSubject: In applied subject's/department's bachelor level's credit hour.
- calculatedCGPA: In applied subject's/department's bachelor level's counted CGPA.
- isEligibleByCGPA: Is CGPA is enough for applied or not.
- academicQualificationPoints: Gives point according to previous result calculation and make auto meritlist.
- isInService: Applicant's is in any job or in servive or not.
- serviceInfo: If is in any service, then that's full information.
- numbersOfPublications: Publication numbers if have any.
- publications: If publication have then attach the publication's link.
- declarationAccepted: To submit the application, need to accept the declaration.
- payment: Admission Payment Schema.
- documents: Need upload required documents.
- applicationNo: After submission, Application No auto generated to help find the application.
- applicationStatus: After Submit, real time status show.
- supervisorRank: Per Supervisor's under Applicant's serial by meritwise after Supervisor Approve.
- approvalLog: Approval Log Schema.

Timestamps: Automatically stores the created and updated timestamp.

### **3.5.8 Admission Payment Schema:**

The Admission Payment schema stores information about payment status and details during application time. The fields include:

- email: Applicant's email address.
- program: Applicant's selected admission program.
- role: The user's role in the system.
- admissionSeason: Admission Season Schema.
- amount: Given amount for Application.
- transactionId: Reference to the payment clearance.
- status: Is payment success or failed.

Timestamps: Automatically stores the created and updated timestamps.

### **3.5.9 Admission Season Schema**

The Admission Season schema is created by admin for admission runs in which season that defines. The fields include:

- seasonName: Wants to admit under which season(Ex: January-June).
- academicYear: Wants to admit under which year(Ex: 2026).
- isLock: If lock then that season's under can't applied anyone.
- createdBy: Clarify that this created by Admin.

Timestamps: Automatically stores the created and updated timestamps.

### **3.5.10 Department Registration Range Schema**

The Department Registration Range schema indicated that in specific season's under allotted registration No's range fixed by admin. The fields include:

- admissionSeason: Admission Season Schema.
- action: The action performed on the item.
- department: Department set for allotted registration No's range.
- startRegNo: For specific range's start no.
- endRegNo: For specific range's end no.
- currentRegNo: Which Registration No is last allotted for a department's student who enroll under allotted department's registration no's one specific number.
- createdBy: This created by Admin.

Timestamps: Automatically stores the created and updated timestamps.

### **3.5.11 Notice Schema**

The Notice schema stores notice which show everyone. The field include:

- title: Title of Notice .
- description: Description of Notice.
- attachment: File attach.
- isPublic: If admin make a specific notice public then public can show,otherwise not.
- isLocked: If lock then can't edit that notice.
- expireAt: Can set expire date,in that date automatic delete that notice.
- createBy: Notice Created By Admin.

Timestamps: Automatically stores the created and updated timestamps.

### **3.5.12 Student Enrollment Schema**

The Student Enrollment schema stores final enrollment details. The field include:

- application: Admission Application Schema .
- registrationNumber: Auto allotted registration number for specific applicant.
- studentId: Auto allotted ID number for specific applicant.
- approvedBy: Enrollment approved by Hall-provost, Register, VC.
- enrollmentStatus: Real time enrollment status show.
- enrolledAt: Create enrollment Date.

Timestamps: Automatically stores the created and updated timestamps

### **3.5.13 User Schema**

The User schema stores information related to all system users involved in the PSTU PGS Management System. This includes academic authorities and students who interact with the system. The schema supports role-based access control and secure authentication.

The fields include:

- name: Full name of the user.
- nameExtension: Additional name information or title (if applicable).
- phone: Contact number of the user.
- email: Unique email address used for login and communication.
- password: Encrypted password for secure authentication.
- role: Defines the user role such as *Dean, Chairman, Supervisor, or Student*.
- isActive: Indicates whether the user account is currently active.
- department: Reference to the department associated with the user (mandatory for all roles except Dean,Hall-Provost,Register,PGS Office and VC).

- **subject:** Subject specialization assigned to supervisors.
- **createdBy:** Reference to the admin who created the user account.
- **isSelfRegistered:** Identifies whether the user registered themselves(Only Students).
- **isEnrolled:** Indicates whether the user has completed final enrollment.
- **isFirstLogin:** Tracks whether the user is logging in for the first time.
- **passwordResetToken:** Token used for password reset functionality.
- **passwordResetExpires:** Expiration time for the password reset token.
- **createdAt:** Stores the account creation date.

**Timestamps:** Automatically stores the user creation timestamp.

### 3.5.14 User Tenure Schema

The User Tenure schema is designed to manage the appointment duration and role history of key academic authorities within the PSTU PGS Management System. This schema ensures proper tracking of tenure periods without supervisors, maintaining accountability and historical accuracy within the system.

The fields include:

- **user:** Reference to the user assigned to the role during a specific tenure.
- **role:** Defines the academic role held during the tenure, such as *Dean* or *Chairman*.
- **department:** Reference to the department associated with the role (if applicable).
- **nameSnapshot:** Snapshot of the user's name at the time of appointment, preserved for historical consistency.
- **emailSnapshot:** Snapshot of the user's email address at the time of appointment.
- **startDate:** The date on which the tenure begins.
- **endDate:** The date on which the tenure ends (null if the tenure is currently active).
- **appointedBy:** Reference to the admin who appointed the user to the role.
- **remarks:** Additional notes or comments related to the appointment.

**Timestamps:** Automatically stores the creation and update timestamps for each tenure record.

### 3.6 Database ER Diagram

To visualize the Entity-Relationship (ER) Diagram for PSTU PGS Management System, I design an ER diagram based on the collections and relationships within database, focusing on the connections between admin, users, departments, application systems, students, notice management and other entities.

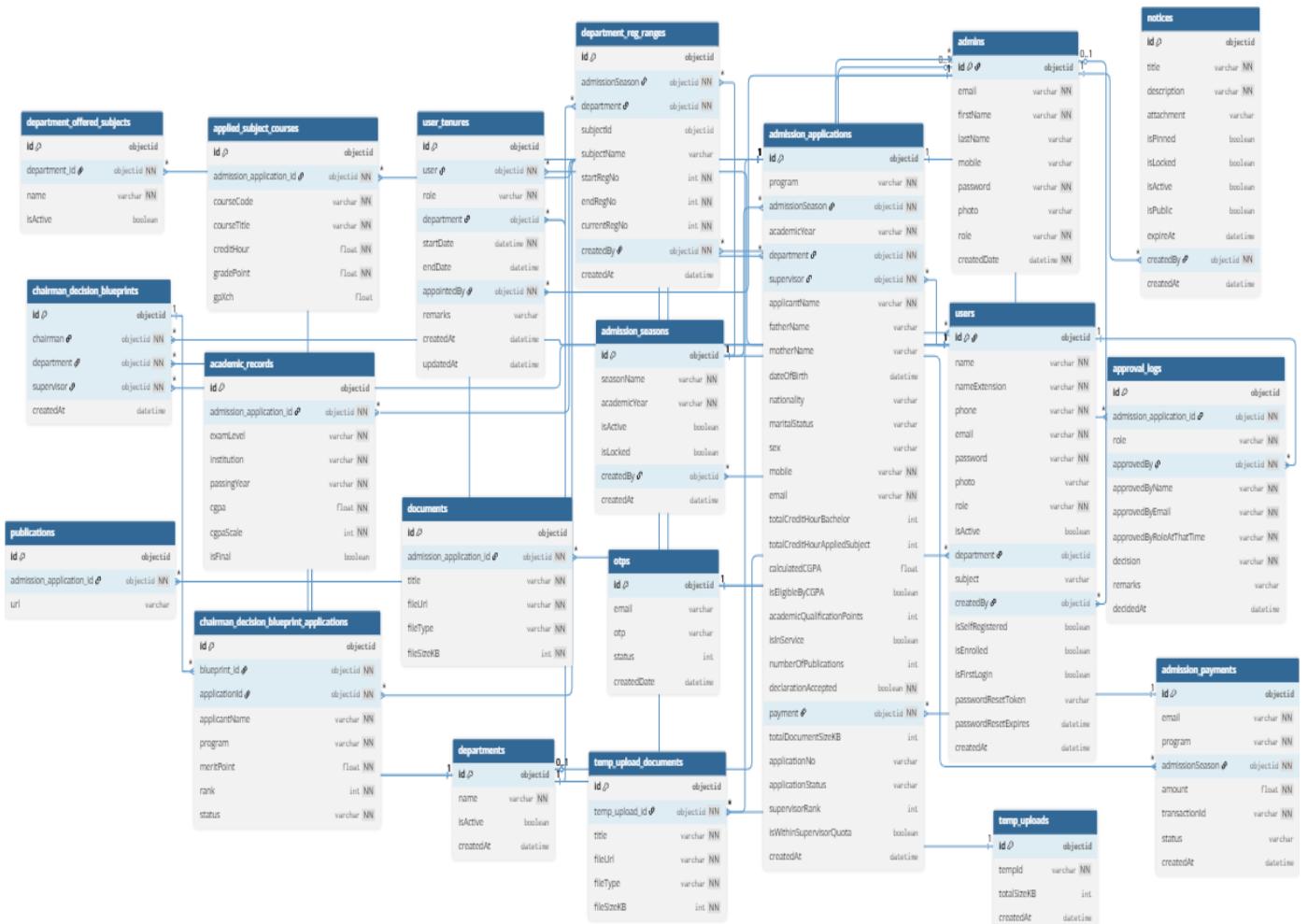


Fig 3.1. Database ER Diagram

### **3.6.1 Key Entities:**

**1. Admin:** This entity stores information about system administrators. It includes like:

- `admin_id`, `email`, `firstName`, `lastName`, `mobile`, `role`, `createdDate`
- Relationships: Admins are associated with users, admission seasons, notices, department registration ranges, and user tenure appointments.

**2. OTP:** Represents one-time passwords used for authentication. It includes like:

- `otp_id`, `email`, `otp`, `status`, `createdDate`
- Relationships: OTPs are linked with admin and user entities for secure authentication.

**3. Admission Season:** Represents an admission cycle for postgraduate studies. It includes like:

- `season_id`, `seasonName`, `academicYear`, `isActive`, `isLocked`
- Relationships: Connected to admission applications and department registration ranges.

**4. Department:** Represents various departments related to the PGS system. It includes like:

- `department_id`, `name`, `isActive`
- Relationships: Departments are associated with users, applications, offered subjects, and registration ranges.

**5. User:** Represents system users such as students and academic authorities. It includes like:

- `user_id`, `name`, `email`, `role`, `department`, `isActive`
- Relationships: Users participate in applications, approvals, supervision, and tenure records.

**6. Admission Application:** Stores postgraduate admission applications. It includes like:

- `application_id`, `program`, `academicYear`, `applicationStatus`, `applicationNo`
- Relationships: Connected with users, departments, payments, academic records, documents, and approvals.

**7. Academic Record:** Represents applicant academic qualifications. It includes like:

- `record_id`, `examLevel`, `institution`, `cgpa`, `passingYear`

- **Relationships:** Linked to admission applications.

**8. Document:** Stores uploaded application documents. It includes like:

- `document_id`, `title`, `fileUrl`, `fileType`, `fileSizeKB`

- **Relationships:** Documents are associated with admission applications.

**9. Admission Payment:** Tracks application fee payments. It includes like:

- `payment_id`, `email`, `amount`, `transactionId`, `status`

- **Relationships:** Payments are linked to admission applications and enrollment.

**10. Approval Log:** Stores approval and decision history. It includes like:

- `approval_id`, `role`, `decision`, `approvedBy`, `decidedAt`

- **Relationships:** Connected to admission applications and user entities.

**11. User Tenure:** Tracks tenure history of academic roles. It includes like:

- `tenure_id`, `role`, `startDate`, `endDate`, `remarks`

- **Relationships:** Linked to users, departments, and admins.

**12. Notice:** Represents official notices and announcements. It includes like:

- `notice_id`, `title`, `description`, `isActive`, `expireAt`

- **Relationships:** Created and managed by admins.

**13. Decision Blueprint:** Defines supervisor allocation logic. It includes like:

- `blueprint_id`, `chairman`, `dean`, `department`, `supervisor`

- **Relationships:** Associated with blueprint applications and the admission process.

**14. Student Enrollment:** This entity stores final enrollment information of approved postgraduate applicants. It includes like:

- enrollment\_id, application, user, admissionSeason, department, supervisor
- registrationNumber, studentId, enrollmentStatus, enrolledAt
- Relationships: Student Enrollment is linked with admission applications, users (students), admission seasons, departments, and supervisors. Each admission application can have only one enrollment record, ensuring a one-to-one relationship between application and enrollment.

### 3.7 Summary

This chapter presented a comprehensive overview of the methodology followed throughout the development of the PSTU PGS Management System, covering the selected development model, system architecture, database design, feasibility analysis, and testing strategies. A structured and systematic approach, guided by established software engineering best practices, was adopted to ensure the development of a secure, efficient, and user-friendly postgraduate automation system for Patuakhali Science and Technology University (PSTU).

Particular emphasis was placed on database design to ensure scalability, data integrity, and optimal system performance. MongoDB was chosen as the NoSQL database due to its flexibility and suitability for handling complex and evolving postgraduate admission workflows. The database was organized into well-defined collections supporting user and role management, admission applications, academic records, document handling, payment processing, approval workflows, enrollment management, and notification services. Additionally, a feasibility study was conducted to evaluate the technical, operational, financial, and scheduling viability of the system, confirming that the proposed solution is practical, sustainable, and appropriate for deployment within the university environment. The subsequent chapter will focus on the system design details, including architectural diagrams, interface specifications, and the interaction between system components, illustrating how the PSTU PGS Management System delivers a cohesive and efficient solution for managing postgraduate academic and administrative processes.

# **Chapter 4: PSTU PGS Management System Design**

## **4.1 Introduction**

This chapter presents a comprehensive system design for the PSTU PGS Management System, a secure and web-based platform developed to streamline and automate postgraduate academic and administrative processes at Patuakhali Science and Technology University (PSTU). The primary objective of the system is to provide a centralized, scalable, and secure environment for managing postgraduate admissions, user roles, application workflows, approvals, enrollment processes, and academic communications within the university.

The system design emphasizes modularity, scalability, and reliability, ensuring that different functional components such as authentication, application processing, document management, payment handling, approval workflows, and enrollment management operate independently yet cohesively. Modern software engineering principles are incorporated to support role-based access control, real-time status updates, and secure data handling. The architecture is designed to accommodate future extensions, including thesis management, stipend processing, and postgraduate academic progress tracking.

This chapter elaborates on the overall system architecture, core modules, frontend and backend design considerations, database structure, and integration strategies. Together, these design elements provide a clear and detailed blueprint for the development and deployment of the PSTU PGS Management System, ensuring an efficient, transparent, and user-friendly solution for managing postgraduate studies at PSTU.

## **4.2 Overall System Design**

The PSTU PGS Management System is architecturally designed to ensure scalability, modularity, and a seamless user experience while prioritizing strong security and data integrity. The system integrates multiple functional modules that operate cohesively to support efficient postgraduate admission and academic management workflows. It adheres to modern software engineering standards and is capable of accommodating diverse user roles, including applicants, administrators, and academic authorities. The system is designed to replace fragmented and manual postgraduate admission processes by automating application submission, approval workflows, enrollment procedures, and communication mechanisms. Through centralized data management and real-time updates, the PSTU PGS Management System improves operational efficiency, enhances transparency, and ensures accurate and reliable handling of postgraduate academic information across the university.

## 4.2.1 System Architecture

The PSTU PGS Management System adopts a modular client–server architecture, organized into well-defined layers to ensure separation of concerns, scalability, security, and ease of maintenance. This layered design enables independent development and efficient interaction between system components. The major architectural layers are described below:

### 4.2.1.1 Frontend Layer:

The frontend layer provides a dynamic, responsive, and user-friendly interface for all stakeholders, including applicants, administrators, and academic authorities. This layer is responsible for user interaction, data visualization, and communication with backend services through secure APIs. The core components of the frontend include:

- **Interactive Dashboards:**  
Provides role-specific dashboards for Admin, Supervisor, Chairman. Each dashboard presents relevant information such as application status, approval queues and notifications.
- **Admission Application Interface:**  
Allows applicants to submit applications, upload academic records and documents, track application status, and receive real-time updates throughout the admission process.
- **User and Role Management Interface:**  
Enables administrators to create and manage users, assign roles, control access permissions, and manage tenure-based academic roles within the system.
- **Approval Interface:**  
Provides academic authorities with interfaces to review applications, approve or reject submissions, manage approvals, and monitor waiting lists efficiently.
- **Notification and Status Tracking:**  
Displays real-time system notifications and application status updates, ensuring transparency and effective communication between users and administrators.
- **Responsive Design:**  
The frontend is developed using React.js for component-based UI development, Bootstrap for responsive styling, and React-Redux for centralized state management. This ensures consistent performance, maintainable code structure, and accessibility across desktop and mobile devices.

#### **4.2.1.2 Backend Layer:**

The backend of the PSTU PGS Management System exposes a RESTful API that acts as an intermediary between the frontend and the underlying services. The backend is developed using Node.js and Express.js, ensuring efficient request handling, scalability, and secure communication. The backend is responsible for managing the following core functionalities:

- User Authentication and Authorization:

Implements secure login mechanisms and role-based access control using JWT-based authentication. OTP verification and password recovery mechanisms are integrated to enhance system security.

- Application and Enrollment Management:

Handles the creation, retrieval, update, and processing of postgraduate admission applications, enrollment records, and student activation workflows, ensuring real-time status updates across the system.

- Database Operations:

Manages all CRUD operations related to users, roles, admission applications, academic records, documents, payments, approvals, and enrollment data through MongoDB.

- Payment Processing (SSLCommerz Sandbox):

Integrates the SSLCommerz sandbox payment gateway to simulate and validate online payment workflows for application and enrollment fees. This allows secure testing of transaction initiation, payment confirmation, and status handling without processing real financial transactions.

- Approval Workflow and Notification Services:

Controls multi-level approval processes involving supervisors, chairmen, deans, provosts, PGS specialists, and registrars, while maintaining approval logs. Automated email notifications are triggered for payment confirmation, approval decisions, waiting list updates, and enrollment status changes.

#### **4.2.1.3 Infrastructure Services:**

Infrastructure services form the backbone of the PSTU PGS Management System, ensuring scalability, security, and data integrity across all system operations.

- **Server Environment and Deployment:**

The system is designed to operate within a controlled server environment, supporting secure deployment and efficient handling of concurrent user requests. The architecture allows future integration with container-based deployment if required.

- **Database Services:**

A MongoDB NoSQL database is used to store and manage all postgraduate-related data, including users, roles, admission applications, academic records, documents, payments, approvals, enrollment records, and notifications. MongoDB's flexible schema and scalability support the complex and evolving requirements of postgraduate academic workflows.

- **Payment Gateway Integration:**

The system integrates the SSLCommerz sandbox payment service to simulate secure online transactions for application and enrollment fees. This ensures reliable payment handling and validation during development and testing.

- **Email and Notification Services:**

Email services are integrated to automatically notify applicants and academic authorities regarding application submission, payment confirmation, approval decisions, waiting list updates, and enrollment status. This improves communication, transparency, and user engagement throughout the postgraduate admission process.

#### **4.2.1.4 Communication Protocols:**

- **HTTP/HTTPS:**

The system uses HTTP/HTTPS protocols for secure communication between the frontend and backend layers. All client requests, including authentication, application submission, approval actions, payment processing, and data retrieval, are handled through RESTful APIs over HTTPS to ensure data confidentiality and integrity.

- Request–Response Based Updates:

Application status updates, approval decisions, and notification data are retrieved through API-based request–response mechanisms. This approach ensures reliable data synchronization between the client and server without the need for persistent socket connections.

#### **4.2.1.5 Security:**

The system implements best practices for data security and access control, including:

- JWT-Based Authentication:

Ensures secure user login and session management through token-based authentication, preventing unauthorized access to system resources.

- Data Protection and Password Security:

Sensitive information such as user passwords is securely protected using encryption and hashing techniques, ensuring confidentiality and data integrity.

- Role-Based Access Control (RBAC):

Access to system features and data is strictly controlled based on user roles such as Admin, Applicant, Supervisor, Chairman, Dean, Provost, PGS Specialist, Registrar, and Student. This ensures that only authorized users can view or modify sensitive postgraduate admission, approval, and enrollment information.

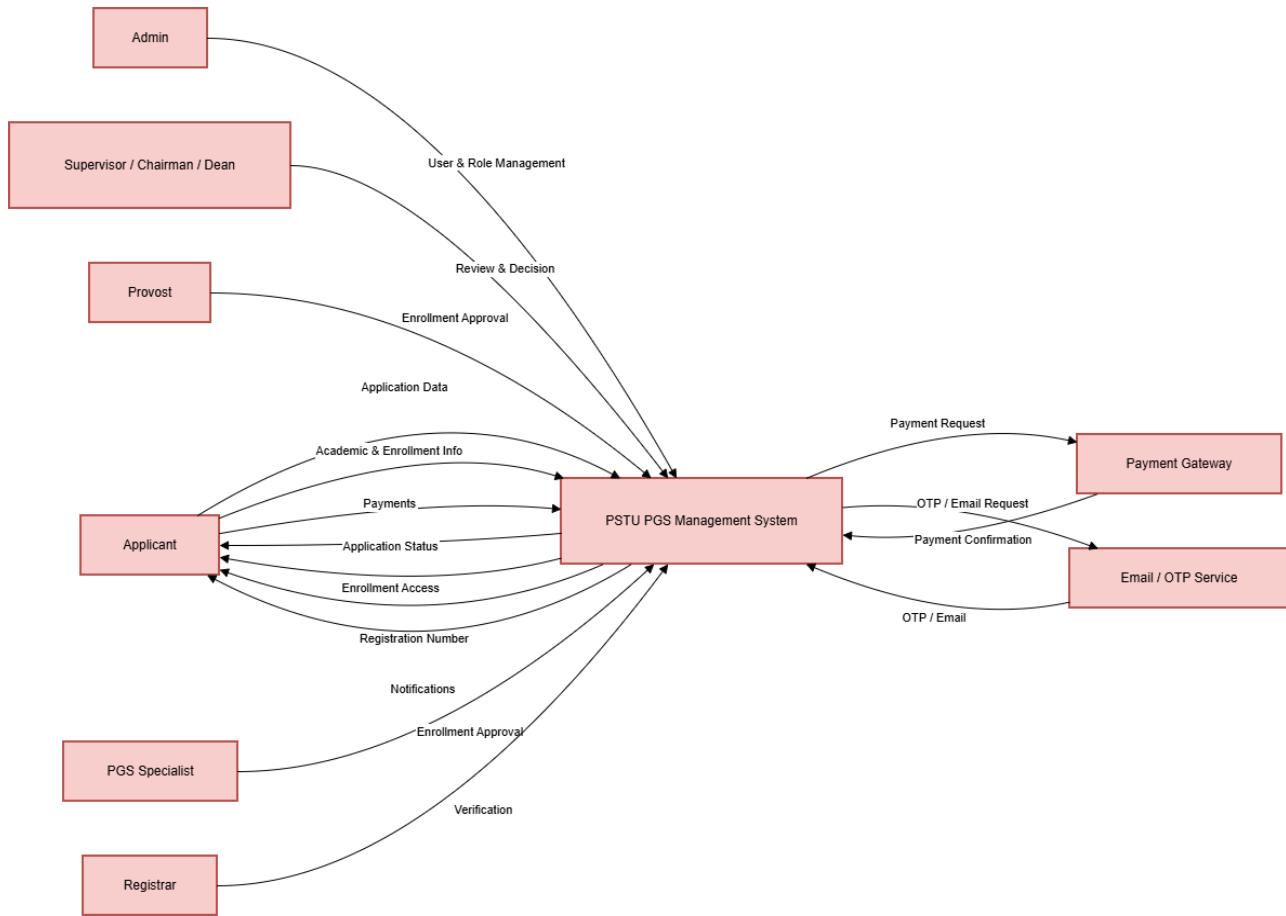


Fig 4.1. System Architecture of PSTU PGS Management System(Specially Admission and Enrollment System)

#### 4.2.2 Infrastructure Services

The PSTU PGS Management System includes several key infrastructure services that support critical system operations, ensuring reliability, security, and smooth workflow execution. These services are designed to handle communication, transaction processing, and academic workflow management in an integrated manner.

One of the core infrastructure components is the Payment Processing Service, which manages online payment workflows for application and enrollment fees. The system integrates the SSLCommerz sandbox payment gateway to initiate transactions, verify payment status, and securely record transaction details. This service ensures that payment confirmation is properly validated before allowing applicants to proceed to subsequent stages such as application submission or enrollment.

The Email and Notification Service plays a vital role in system communication by automatically sending email notifications to applicants and academic authorities. Notifications are generated for key events including application submission, payment confirmation, approval decisions, waiting list updates, temporary enrollment access, and final enrollment confirmation, ensuring transparency and reducing manual follow-ups.

Additionally, the Application and Enrollment Processing Service manages postgraduate admission and enrollment workflows, including multi-level approvals by supervisors, chairmen, deans, provosts, PGS specialists, and registrars. This service ensures that each process step is executed in the correct sequence and accurately recorded in the system database.

Together, these infrastructure services enable the PSTU PGS Management System to operate efficiently during peak admission periods while maintaining data consistency, transaction reliability, and effective communication across all stakeholders.

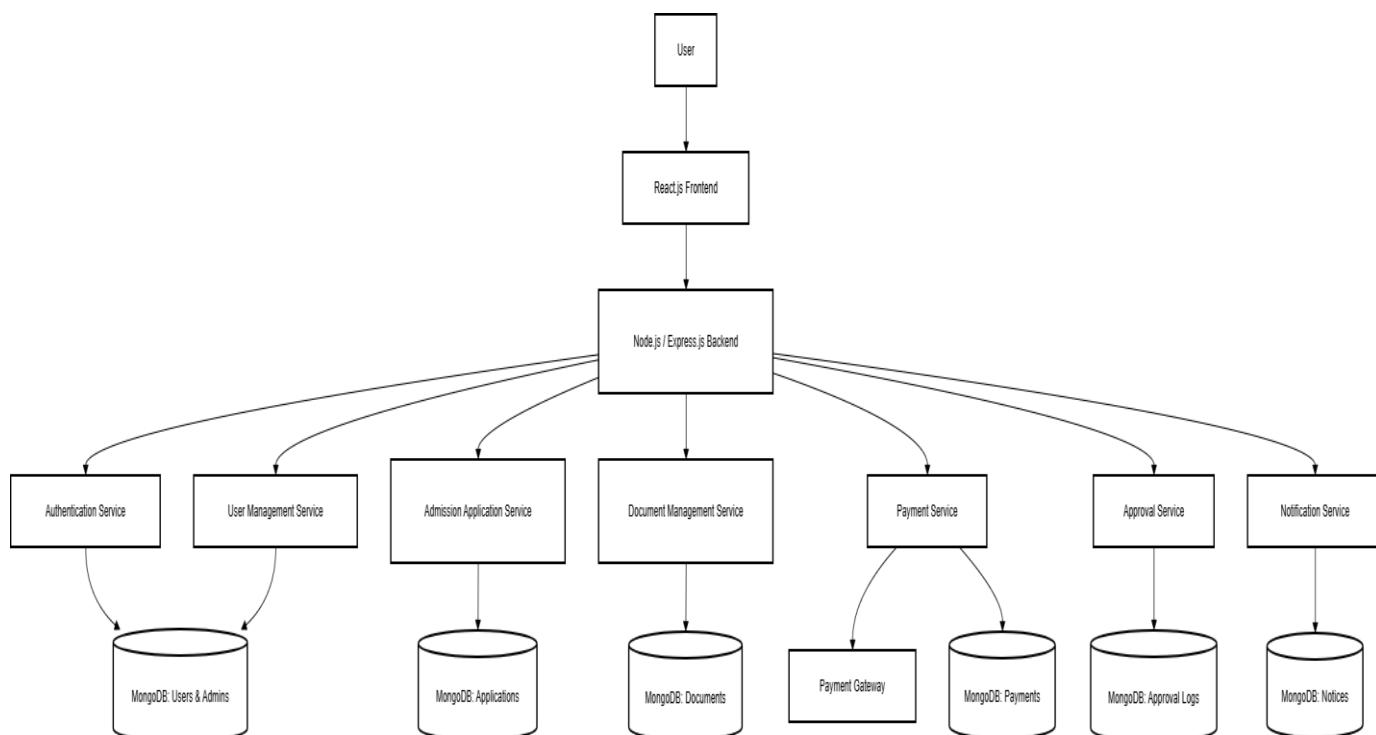


Fig 4.2. Code Execution Workflow

#### **4.2.3 Four-Tier Architecture Diagram**

The PSTU PGS Management System follows a four-tier architectural model to ensure modularity, scalability, security, and ease of maintenance. Each tier is responsible for a specific set of functions, allowing independent development and efficient interaction among system components.

##### **1. User Layer:**

This layer represents all end users of the system, including applicants, administrators, supervisors, chairmen, deans, and other academic authorities. Users interact with the system through web-based interfaces to perform tasks such as application submission, approval processing, user management, and status tracking.

##### **2. Presentation Layer:**

The presentation layer provides role-based user interfaces for Applicants, Admins, and Supervisors. It handles user interactions, form submissions, data visualization, and input validation. This layer ensures a responsive and user-friendly experience while securely communicating user requests to the application layer.

##### **3. Application Layer:**

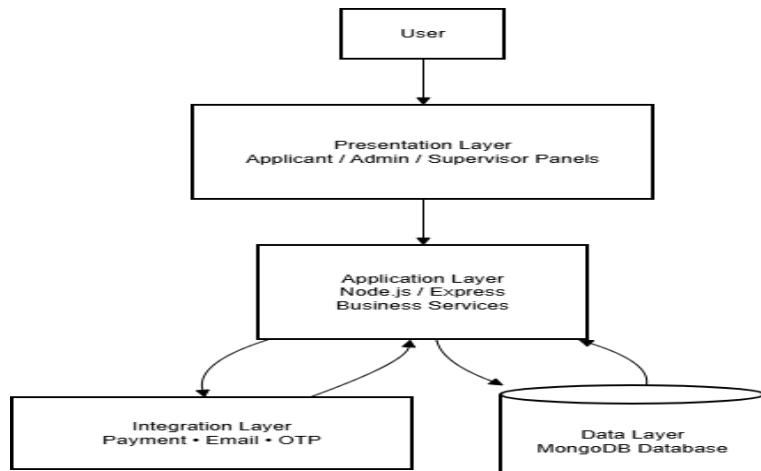
The application layer contains the core business logic of the system and is implemented using Node.js and Express.js. It processes user requests, enforces business rules, manages authentication and authorization, and coordinates workflows such as application processing, document validation, payment verification, and approval handling. This layer acts as the central controller of the system.

##### **4. Integration Layer:**

The integration layer is responsible for connecting the system with external services such as payment gateways, email services, and OTP verification mechanisms. It enables secure communication with third-party services for payment processing, notification delivery, and additional security features.

##### **5. Data Layer:**

The data layer uses MongoDB as the primary database to store and manage all system data, including user accounts, admission applications, academic records, documents, payments, approval logs, and notifications. This layer ensures data consistency, persistence, and efficient retrieval.



*Fig 4.3. Code Execution in Four Layer*

#### 4.2.4 User Activity Diagram

The User Activity Diagram for the PSTU PGS Management System illustrates the primary interactions performed by different users within the system, presenting a clear sequence of actions starting from authentication and progressing through application processing, approval workflows, enrollment, and notification handling.

##### Login Process:

Users begin by securely logging into the system using their registered credentials. Authentication is handled by the backend through RESTful API calls with JWT-based authorization, ensuring that only authorized users can access the system. Based on the authenticated user's role—such as Applicant, Admin, Supervisor, Chairman, Dean, Provost, PGS Specialist, Registrar, or Student—the system redirects the user to a role-specific dashboard that provides access to relevant functionalities and information.

Once logged in, users can perform actions according to their assigned roles, such as submitting applications, reviewing and approving applications, managing users, processing payments, completing enrollment, or monitoring application and academic status updates. Throughout these interactions, the system maintains secure data handling and provides timely notifications to ensure transparency and smooth workflow execution.

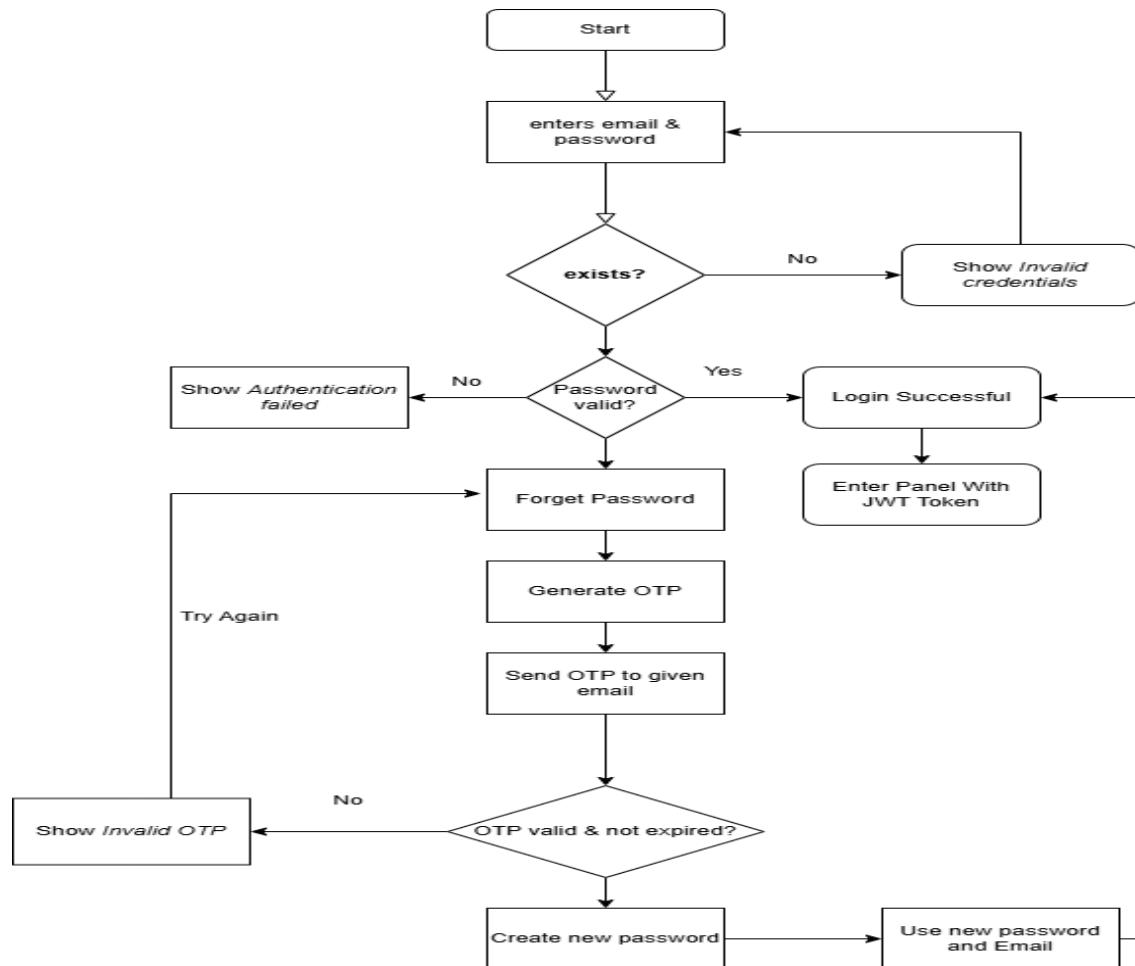


Fig 4.4. Authentication Workflow

## **User Creation Process:**

The user creation process begins when an administrator accesses the Admin Panel to add a new system user. The admin first checks whether the provided email address is already registered in the system. If the email already exists, the process is terminated to prevent duplicate accounts.

If the email is not registered, the admin assigns an appropriate role to the user. For roles such as Chairman or Supervisor, the admin is required to assign a corresponding department to ensure proper academic association. After role and department assignment, the system creates the user account and generates a user tenure record to track role validity and appointment history.

Once the user account and tenure information are successfully created, the system sends login credentials or a confirmation message to the newly created user. The process then ends.

## **User Update and Replacement Process:**

The user update process is initiated when an administrator accesses the Admin Panel and views the existing users list. The system verifies whether the operation is being performed by an authorized admin. If not, the process is terminated.

The admin selects a user and checks whether the user holds a Supervisor role. If the user is not a supervisor, the system does not allow replacement and the process ends. If the user is a supervisor, the admin proceeds to update user information such as name, email, and mobile number.

After updating the user details, the system updates the corresponding user tenure record to maintain accurate historical and role-related information. Finally, a confirmation message or updated credentials are sent to the affected user, and the process is completed.

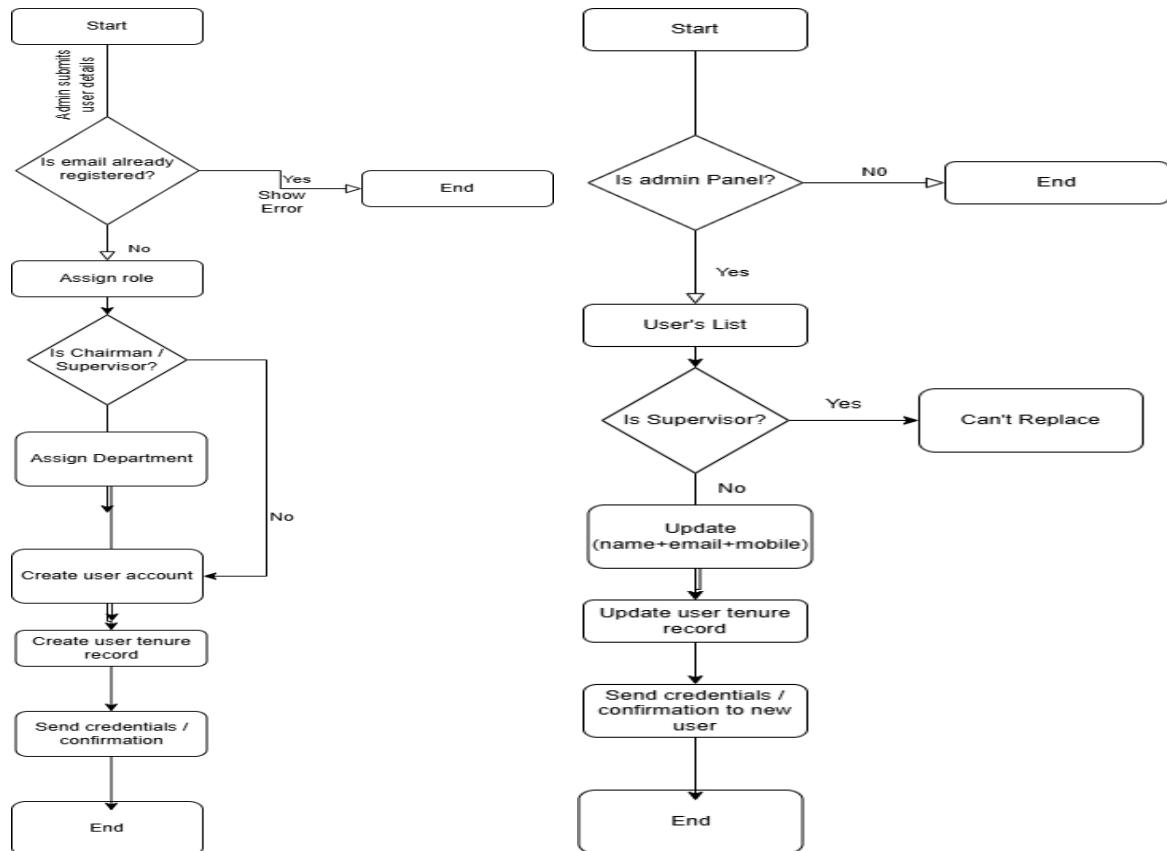


Fig 4.5. Admin Create Users Workflow

Fig 4.6. Update Users Workflow

## **Academic Record Processing:**

The process begins when the applicant enters their academic records by providing details such as examination level, institution name, passing year, CGPA, and CGPA scale. After submission, the system validates whether the entered CGPA meets the minimum eligibility criteria defined by the admission rules. If the CGPA is found to be invalid or below the required threshold, the applicant is marked as ineligible and the process ends.

If the CGPA is valid, the system calculates the final CGPA along with the corresponding academic qualification points based on predefined rules. Once the calculation is completed, the academic records and calculated results are securely stored in the database, and the process proceeds to the next stage of the admission workflow.

## **Document Upload Process:**

The document upload process begins when the applicant uploads the required documents through the system interface. The system first validates the file type to ensure it matches the allowed formats. If the file type is invalid, the upload is rejected. If the file type is valid, the system then checks whether the file size is within the permitted limit. Files exceeding the size limit are rejected. When both validations are successful, the document is temporarily stored, and upon confirmation, it is moved to the permanent document storage, completing the upload process.

## **Payment Processing:**

The payment process starts when the applicant initiates the payment for application or enrollment fees. The system sends the payment request to the payment gateway for processing. If the payment is unsuccessful, the transaction is marked as failed. If the payment is successful, the transaction details are securely stored in the system, and the payment is linked to the corresponding admission application. The process then ends, allowing the applicant to proceed to the next stage.

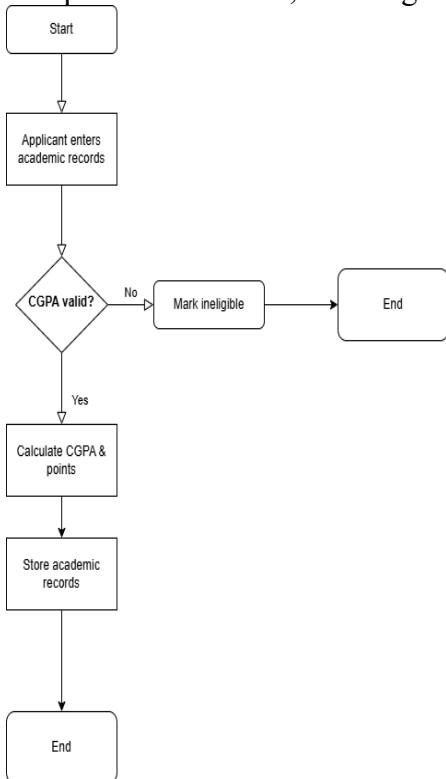


Fig 4.7. Academic Record Process Workflow

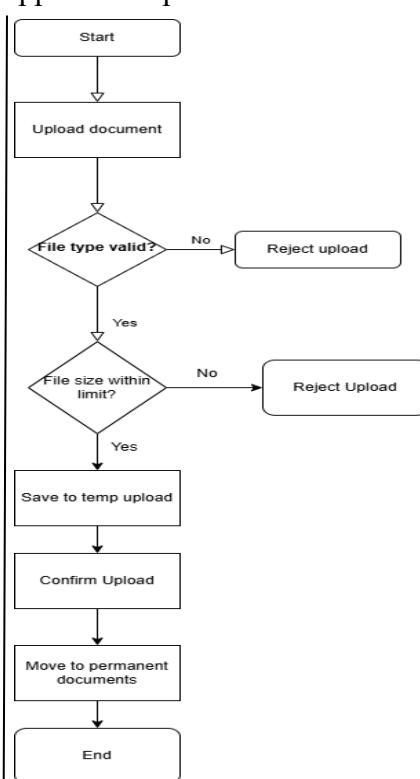


Fig 4.8. Document Upload Workflow

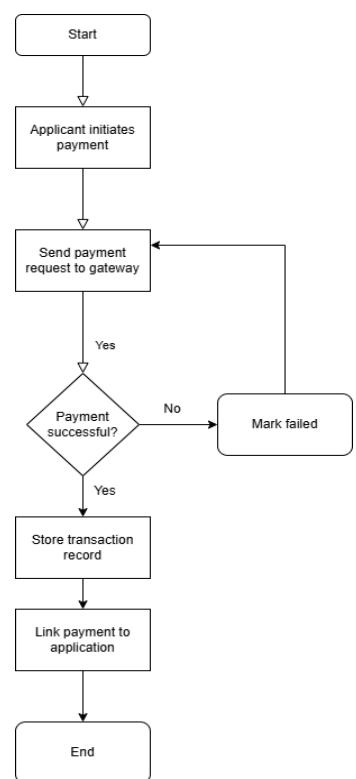


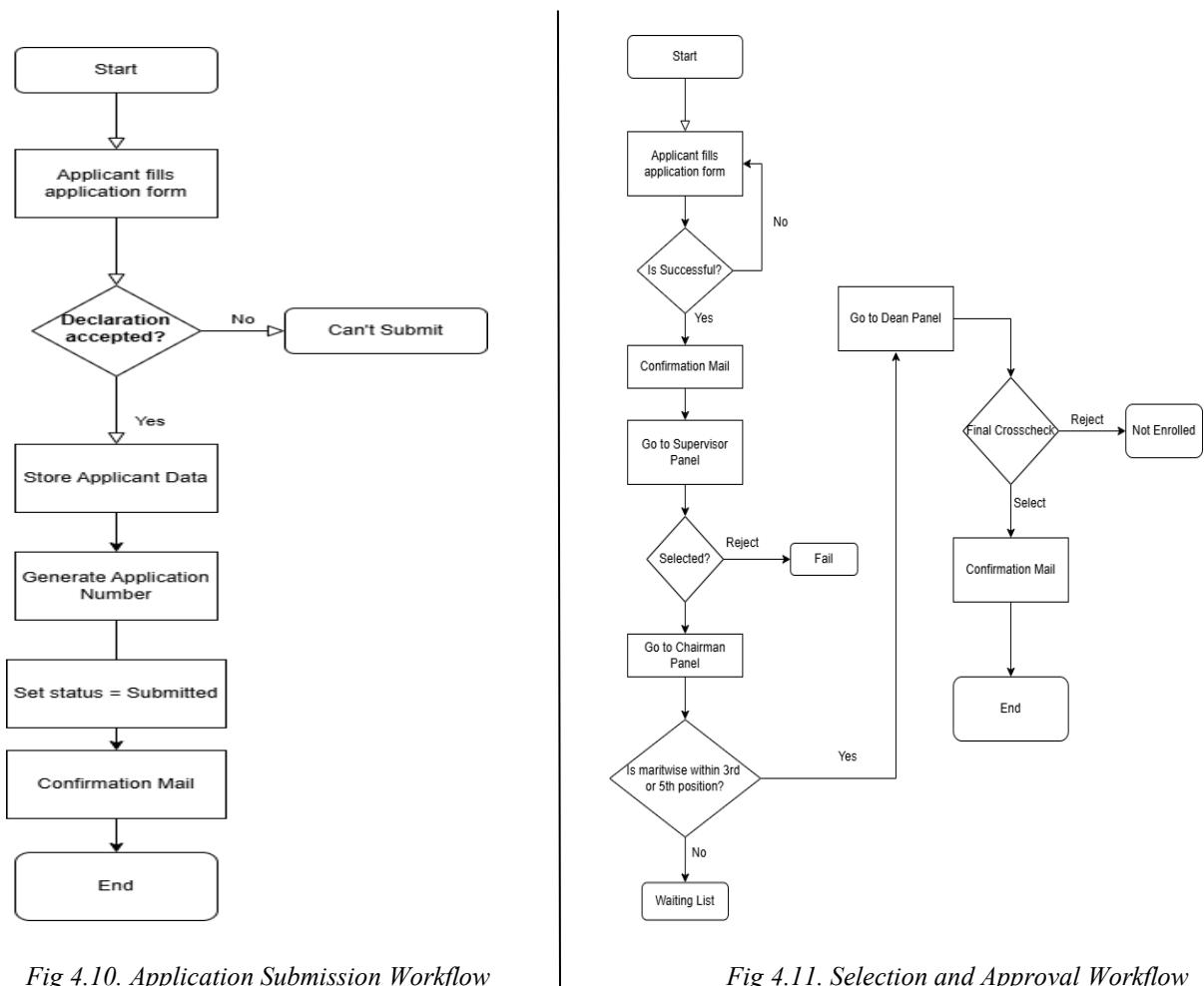
Fig 4.9. Payment Workflow

## Application Submission Process:

The application submission process begins when the applicant fills out the online admission application form. Before submission, the system verifies whether the applicant has accepted the mandatory declaration. If the declaration is not accepted, the system does not allow the application to be submitted. Once the declaration is accepted, the applicant's data is securely stored in the database, and a unique application number is generated. The application status is then set to “Submitted”, and a confirmation email is sent to the applicant, marking the successful completion of the submission process.

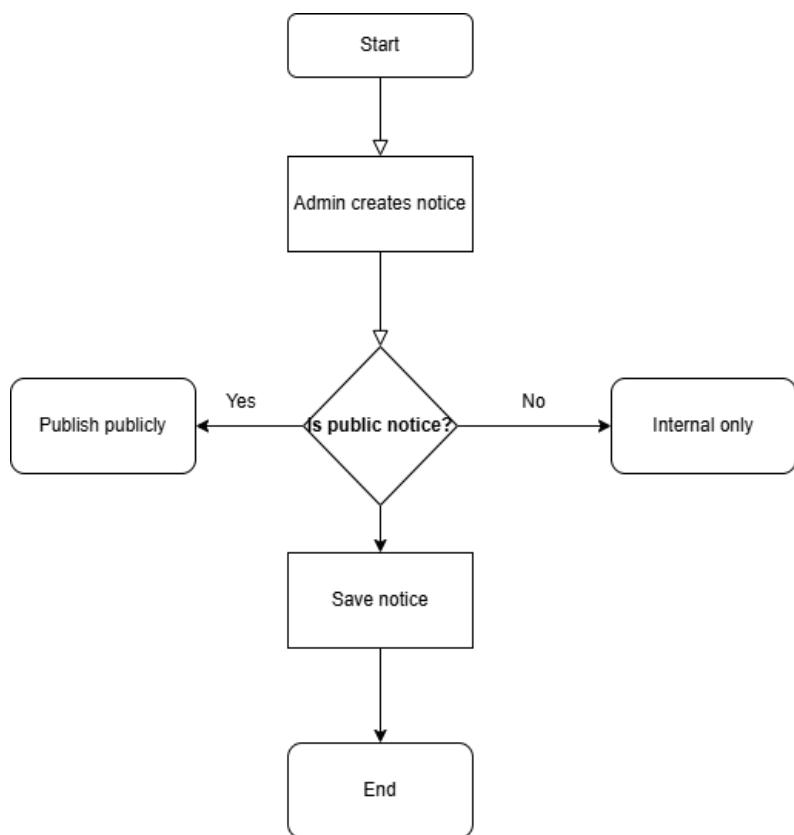
## Application Selection and Approval Process:

After successful submission, the application enters the selection and approval workflow. The application is first forwarded to the Supervisor Panel for initial review. If the applicant is not selected at this stage, the process is marked as failed. Selected applications proceed to the Chairman Panel, where merit position is evaluated. If the applicant falls within the waiting list range, the application is placed on the Waiting List. Otherwise, the application moves to the Dean Panel for final crosschecking. If the application passes the final crosscheck, a confirmation email is sent to the applicant; otherwise, the applicant is marked as Not Enrolled, and the process ends.



## **Notice Management Process:**

The notice management process is handled through the Admin Panel, where authorized administrators can create, update, publish, or deactivate official notices related to postgraduate admissions and academic activities. While creating a notice, the admin provides necessary information such as title, description, attachments, visibility status, and expiration date. Once published, notices are made available to relevant users based on access settings, ensuring timely dissemination of important information. The system also supports pinning and locking notices to control visibility and prevent unauthorized modifications, ensuring accuracy, transparency, and effective communication across the platform.



*Fig 4.12. Notice Management Workflow*

#### **4.2.5 Data Flow Diagram**

The Data Flow Diagram (DFD) for the PSTU PGS Management System visually represents how data moves through the system, illustrating the flow, processing, and transformation of information among different system components. It provides a clear overview of the system's core processes and their interactions with external entities, data stores, and outputs.

The DFD helps in understanding how various functional modules—such as user authentication, admission application processing, document management, payment handling, approval workflows, enrollment management, and notification services—communicate and exchange data. This diagram plays a vital role in identifying system processes, clarifying data dependencies, and ensuring that all components operate cohesively to deliver a secure, efficient, and transparent postgraduate admission management solution.

##### **4.2.5.1 Level 0 - Context Diagram**

At the highest level, the Context Diagram provides an overall view of the PSTU PGS Management System. It represents the system as a single unified process and illustrates how external entities interact with it. This diagram helps to understand the system boundaries and the flow of information between users and the system.

##### **External Entities:**

###### **Applicant:**

Applicants interact with the system to submit postgraduate admission applications, upload required documents, make online payments, and receive application status updates and notifications.

###### **Admin:**

The admin manages system operations, including user creation, role assignment, admission seasons, notices, registration ranges, and overall system configuration.

###### **Users(Dean/Supervisor/Chairman):**

These academic authorities interact with the system to review applications, evaluate candidates, approve or reject applications, manage supervision decisions, and perform academic crosschecks.

###### **Payment Gateway:**

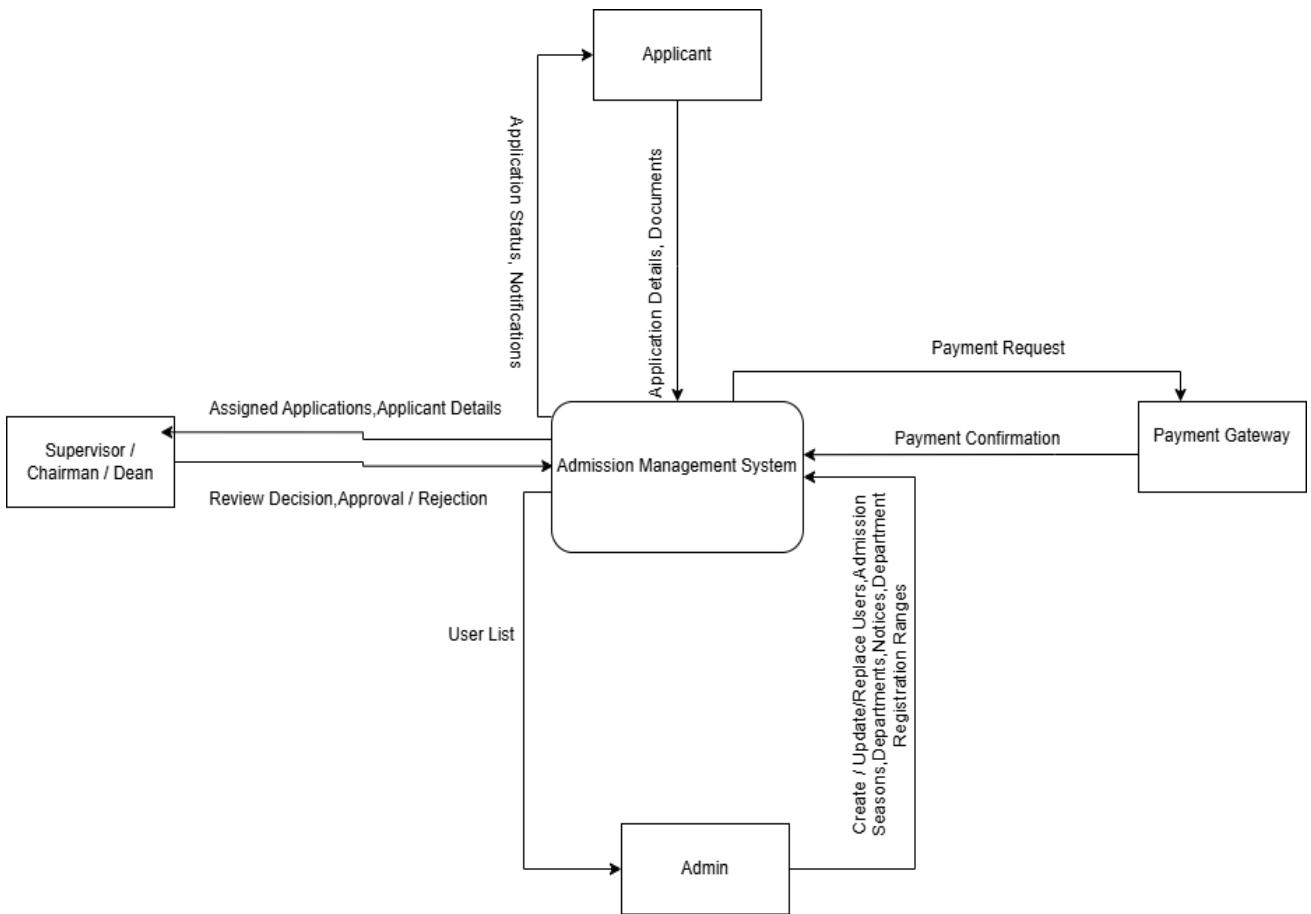
The payment gateway facilitates secure online payment processing for application and enrollment fees and sends transaction status updates to the system.

###### **System(PGS Management System):**

The central system manages postgraduate admission workflows, user authentication, application processing, document handling, payment verification, approval workflows, enrollment management, and notification delivery.

## **Core Processes Represented:**

- User Authentication: Verifies user credentials and provides role-based access to the system.
- Admission Application Management: Handles application submission, validation, storage, and status tracking.
- Payment Processing: Manages online fee transactions and links payment status with applications.
- Approval and Decision Workflow: Coordinates multi-level academic review and approval processes.
- Notification Management: Sends automated email notifications for submissions, approvals, payments, and enrollment updates.



*Fig 4.13. Level-0 Data Flow Diagram*

#### **4.2.5.2 Level 1 - Decomposition of Major Processes**

In Level 1, the diagram decomposes the PSTU PGS Management System into its major functional processes related to postgraduate admission application handling and final decision making. These processes describe how data flows between users, system components, and data stores during the application lifecycle.

##### **1. Process 1: User Authentication & Authorization**

- Input: User credentials (email, password, OTP if required)
- Output: Authentication token (JWT), authenticated user profile, role-based access permissions
- Data Store: User Database (MongoDB), OTP Database

##### **2. Process 2: Admission Application Management**

- Input: Application details (program, academic year, personal information, declaration acceptance)
- Output: Stored application data, unique application number, application status (Submitted)
- Data Store: Admission Applications Database (MongoDB)

##### **3. Process 3: Academic Record & Document Processing**

- Input: Academic information (exam level, institution, CGPA) and uploaded documents
- Output: Validated academic records, document storage confirmation, eligibility status
- Data Store: Academic Records Database, Documents Database

##### **4. Process 4: Payment Processing**

- Input: Application fee payment request
- Output: Payment confirmation or failure status linked to the application
- Data Store: Admission Payments Database
- External System: SSLCommerz Sandbox Payment Gateway

##### **5. Process 5: Application Review & Final Approval**

- Input: Submitted applications, academic records, payment confirmation
- Output: Application decision (Approved, Rejected, or Waiting List)
- Data Store: Approval Logs Database, Admission Applications Database

##### **6. Process 6: Notification Service**

- Input: System events (application submission, payment confirmation, final decision)
- Output: Email notifications to applicants and academic authorities
- External System: Email Service (SMTP / Email API)

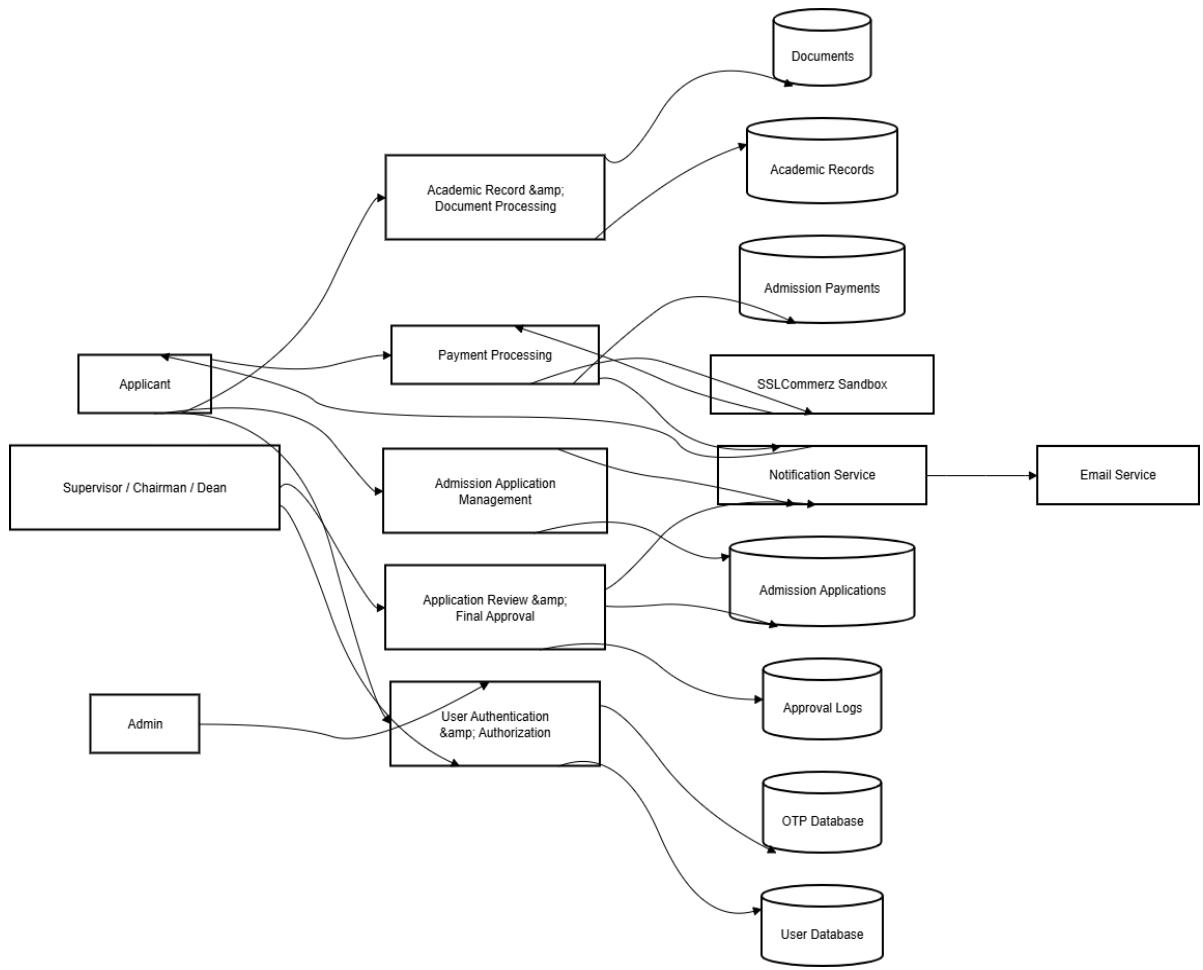


Fig 4.14. Level-1 Data Flow Diagram

## **4.3 Application Features**

The PSTU PGS Management System is designed to provide a comprehensive and secure platform for managing postgraduate admission and academic administrative processes at Patuakhali Science and Technology University (PSTU). The system ensures transparency, efficiency, and real-time information flow among applicants, administrators, and academic authorities. The major features of the application are described below:

### **4.3.1 Admin and User Management**

The system provides centralized control for administrators to manage users and academic roles:

- User Creation by Admin: Admins can create user accounts for academic authorities such as Supervisors, Chairmen, Deans, Provosts, PGS Specialists, Registrars, and Students.
- Role and Department Assignment: Each user is assigned a specific role and, where applicable, a department to ensure role-based access control.
- Tenure Management: Time-bound academic roles (Ex: Chairman, Dean) are managed using tenure records to ensure accurate replacement and prevent role conflicts.

### **4.3.2 Admission Season and Range Management**

This feature ensures structured and real-time control over admission cycles:

- Real-Time Admission Season Listing: Admins can create, activate, lock, or deactivate admission seasons, which are displayed in real time across the system.
- Department-wise Registration Number Range: Admins can define and manage registration number ranges for each department and admission season, ensuring unique and sequential registration number allocation.
- Controlled Admission Flow: Locking mechanisms prevent unauthorized modifications once an admission season is finalized.

### **4.3.3 Admission Application Management**

This feature supports the complete postgraduate admission application process:

- Online Application Submission: Applicants can fill out and submit admission applications through an online form.
- Application Number Generation: Each submitted application is assigned a unique application number for tracking.
- Application Status Tracking: Applicants can monitor their application status in real time.

#### **4.3.4 Academic Record and Document Management**

The system ensures accurate evaluation and secure handling of applicant data:

- Academic Record Entry: Applicants can submit academic qualifications, including CGPA and institution details.
- CGPA Validation and Calculation: The system validates CGPA eligibility and calculates academic points based on predefined rules.
- Secure Document Upload: Required documents are validated by file type and size before being stored securely in the system.

#### **4.3.5 Payment Processing**

The application integrates an online payment mechanism to simplify fee submission:

- Online Fee Payment: Applicants can pay application fees through the SSLCommerz sandbox payment gateway.
- Payment Verification: Payment status is verified and securely linked to the corresponding application.
- Transaction Records: All payment transactions are stored for transparency and audit purposes.

#### **4.3.6 Application Review and Approval Workflow**

A structured multi-level approval process ensures fair and transparent decision-making:

- Supervisor Review: Applications are initially reviewed by supervisors for academic suitability.
- Chairman Evaluation: Shortlisted applications are evaluated based on merit ranking and departmental quotas.
- Dean Final Approval: Final academic crosschecking is performed before confirming application decisions.
- Waiting List Management: Eligible applicants may be placed on a waiting list based on merit position.

#### **4.3.7 Notification and Communication System**

The system ensures timely communication among all stakeholders:

- Automated Email Notifications: Applicants receive notifications for application submission, payment confirmation, approval decisions, and waiting list updates.
- Authority Alerts: Academic authorities are notified when applications require review or approval.
- System Announcements: Important notices and updates are communicated through the platform.

#### **4.3.8 Security and Data Integrity**

Security is a fundamental aspect of the PSTU PGS Management System:

- Role-Based Access Control (RBAC): Access to system features is strictly controlled based on user roles.
- Password Protection: User passwords are securely hashed and protected.
- Audit and Approval Logs: All critical actions, including approvals and decisions, are logged to ensure accountability and transparency.

#### **4.3.9 Notice Management**

The system includes a dedicated notice management feature for effective communication:

- Notice Creation by Admin: Admins can create and manage official notices related to admissions and academic activities.
- Public Visibility: Notices marked as public are visible to all users, including applicants and guests.
- Notice Control: Admins can pin, lock, activate, or set expiration dates for notices to control visibility and relevance.

## 4.4 User Interface (UI) Design

The user interface (UI) of the PSTU PGS Management System is developed using React.js and styled with Bootstrap CSS framework, prioritizing simplicity, responsiveness, and ease of use. The UI design aims to provide an intuitive and seamless experience for all users, including students, departments, and administrative staff. Whether interacting with the system for application submission, approval and decision-making workflows, online payment processing, or real-time status tracking, users can navigate the platform smoothly with minimal complexity.

### 4.4.1 Home Page

The home page of the PSTU Inventory Management System serves as the primary entry point for admin,users, offering a simple and intuitive interface. Upon accessing the platform,admin and users are presented with clear options to log in or learn more about the system's features.On the navbar,here's a notice sliding section where latest notice's headline sliding always and can show anyone all public notices.Also have Application button,where Application time for admission,a applicant can applied for admission easily. Here users can't register personally because they are registered by Admin,so they can only login. The design is clean and uncluttered, guiding users effortlessly to their next actions. With responsive design built using React.js and Bootstrap, the home page adapts seamlessly to various devices, from desktops to tablets and smartphones, ensuring a smooth and consistent experience across all platforms. Whether users are new or returning, they can easily navigate to the registration form or log in with a single click, providing a user-friendly and accessible start to their journey on the platform.

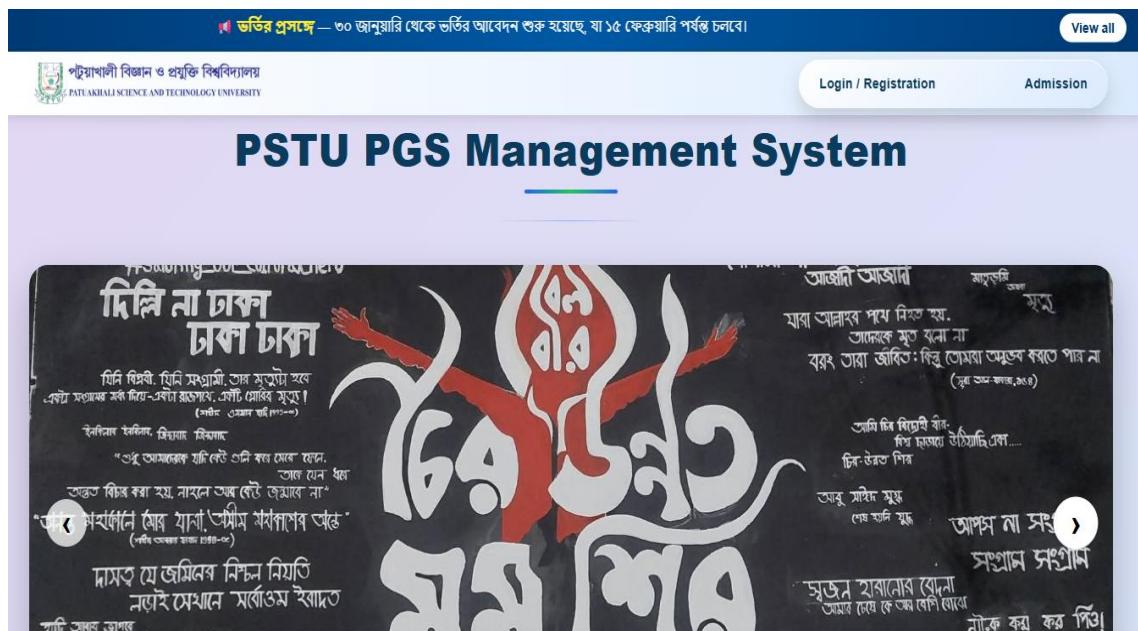


Fig 4.15. Home Page

#### 4.4.2 Login Page

The login page of the PSTU PGS Management System provides a simple and secure interface for users to access their accounts. It includes input fields for email/username and password, allowing users to easily log in. The page is designed with minimal distractions and includes options for password recovery in case of forgotten credentials. It is fully responsive, ensuring users can log in seamlessly from any device, whether on a desktop, tablet, or smartphone, offering a smooth and consistent experience across all platforms.

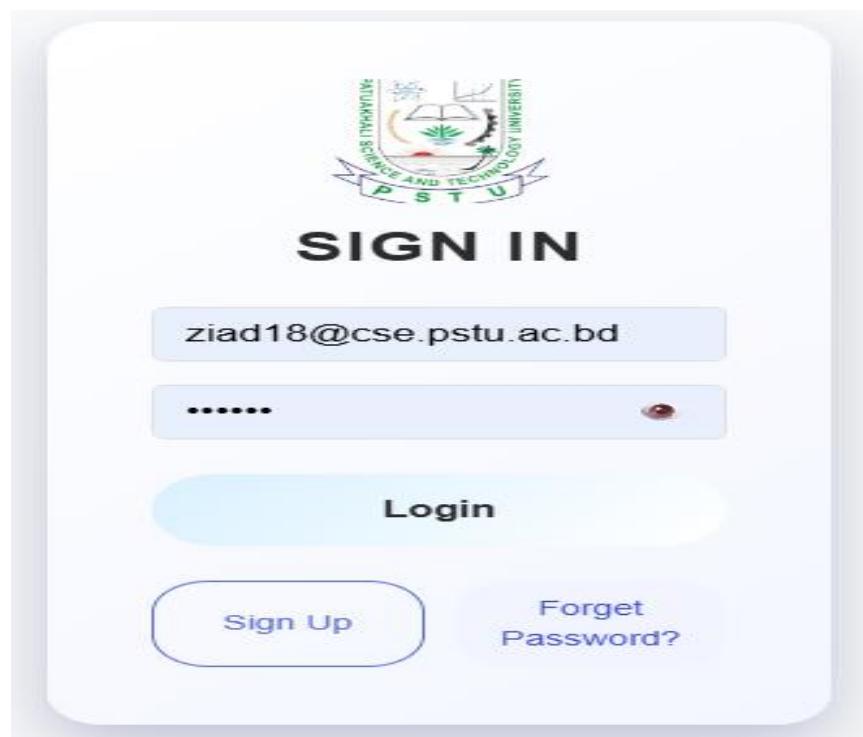


Fig 4.16. Login Page

#### 4.4.3 Recover Password Page

If forget password, then need to email that he/she used, after given email then get OTP, which expire after specific time, suppose:5 min. After given correct OTP, he/she can set new password.



### Email Verification

Enter your registered email address to receive an OTP

Fig 4.17. Send mail for getting OTP

### OTP VERIFICATION

A 6 Digit verification code has been sent to your email address.

Next

Fig 4.18. OTP Set page



### Set New Password

Please create a new secure password

Fig 4.19. Update Password Page

#### 4.4.4 Admin Dashboard

The Admin (PGS) Dashboard of the PSTU PGS Management System provides a centralized and role-based interface for managing postgraduate admission and academic administrative activities. Through this dashboard, administrators and academic authorities can efficiently oversee critical system components such as user management, department management, admission seasons, registration number ranges, and notice management.

The dashboard presents key summary information at a glance, including the total number of chairmen and supervisors, along with department-wise enrollment statistics. This enables administrators to quickly assess the overall status of postgraduate activities across departments. An organized navigation panel allows seamless access to different management modules, ensuring smooth and efficient administrative workflows.

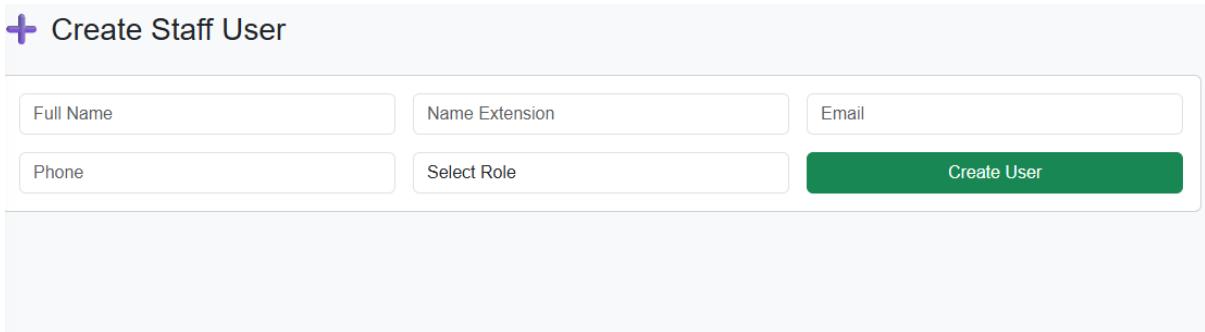
Designed with a clean and intuitive layout, the dashboard emphasizes usability and clarity. It is fully responsive, allowing administrators to manage and monitor system operations effectively from desktop, tablet, or mobile devices. This centralized dashboard significantly improves administrative efficiency, transparency, and real-time decision-making within the PSTU PGS Management System.

The screenshot shows the Admin Dashboard with a purple header. On the left is a sidebar with icons for Dashboard, Notice Management, Department, Season, and User. The main area displays a user profile for 'Dean (PGS) CHayon daschayon26@gmail.com', a count of 'Total Chairmen' (1), and 'Total Supervisors' (4). Below this is a section titled 'Department-wise Final Enrollment' with a table showing 'No enrollment data'.

Fig 4.20. Admin Dashboard

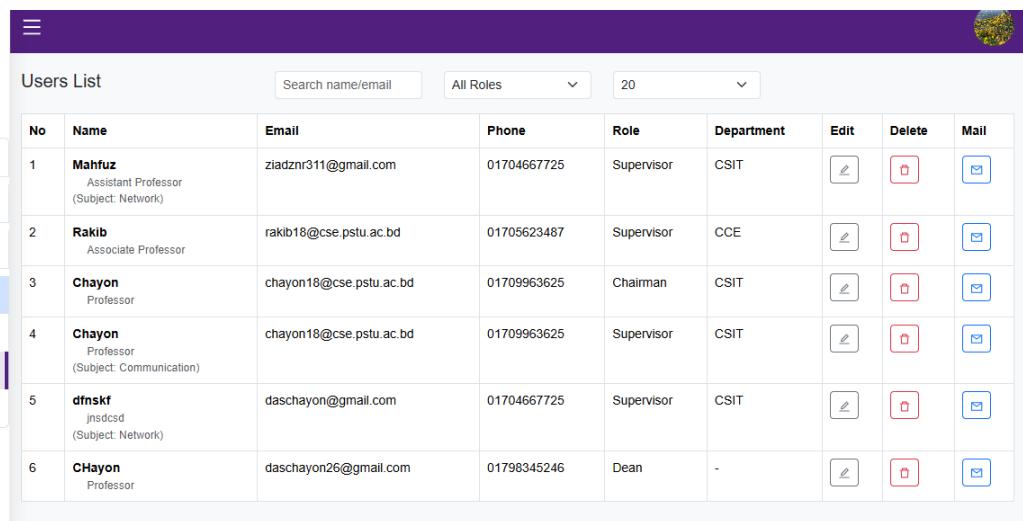
The screenshot shows two pages side-by-side. The left page is 'Create Department' with fields for Faculty (PGS), Department Name, and Offered Subjects (with an 'Add' button). The right page is 'Department List' showing a table with two entries: CSIT (2 subjects, created 1/23/2026) and CCE (0 subjects, created 1/23/2026). Both rows have 'Subjects' and 'Edit' buttons.

Fig 4.21. Create New Department and Department List Page



The screenshot shows a form titled "Create Staff User". It contains five input fields: "Full Name", "Name Extension", "Email", "Phone", and "Select Role". A green "Create User" button is located to the right of the "Select Role" field.

*Fig 4.22. Create User Page*

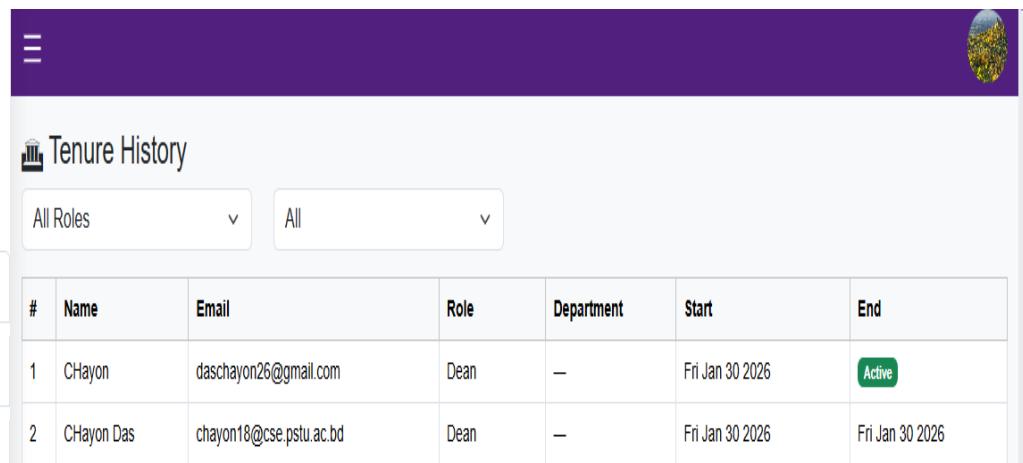


The screenshot shows a table titled "Users List" with columns: No, Name, Email, Phone, Role, Department, Edit, Delete, and Mail. The table lists six users:

No	Name	Email	Phone	Role	Department	Edit	Delete	Mail
1	Mahfuz Assistant Professor (Subject: Network)	ziadznr311@gmail.com	01704667725	Supervisor	CSIT			
2	Rakib Associate Professor	rakib18@cse.pstu.ac.bd	01705623487	Supervisor	CCE			
3	Chayon Professor	chayon18@cse.pstu.ac.bd	01709963625	Chairman	CSIT			
4	Chayon Professor (Subject: Communication)	chayon18@cse.pstu.ac.bd	01709963625	Supervisor	CSIT			
5	dfnskf jnsdcsd (Subject: Network)	daschayon@gmail.com	01704667725	Supervisor	CSIT			
6	CHayon Professor	daschayon26@gmail.com	01798345246	Dean	-			

*Fig 4.23. User's List Page*

Here, At the edit time, if Admin edit at the same time any user's (without Supervisor) mobile no and name change that means replace the user.



The screenshot shows a table titled "Tenure History" with columns: #, Name, Email, Role, Department, Start, and End. The table lists two tenure entries:

#	Name	Email	Role	Department	Start	End
1	Chayon	daschayon26@gmail.com	Dean	-	Fri Jan 30 2026	
2	Chayon Das	chayon18@cse.pstu.ac.bd	Dean	-	Fri Jan 30 2026	Fri Jan 30 2026

*Fig 4.24. Tenure History page*

Select Admission Season

January-June (2026)

Create New Season

Season Academic Year

Select Create Season

Department Registration Range

Department Start Reg No End Reg No

Select Add

Fig 4.25. Create Season and Department Range Page

January-June (2026) Lock Collapse

No	Department	Subject	Start	End	Current	Action
1	CSIT	Network	1000	1020	1000	<button>Edit</button> <button>Delete</button>
2	CCE	—	1021	1050	1021	<button>Edit</button> <button>Delete</button>
3	CSIT	Communication	2000	2060	2000	<button>Edit</button> <button>Delete</button>

Fig 4.26. Season's Details List Page

Create Notice

Notice title Notice description Choose File No file chosen  Public Add

All Notices

#	Title	Status	Attachment	Actions
1	ভর্তির প্রস্তাব	Public	<button>View</button>	<button>Edit</button> <button>Private</button> <button>Pin</button> <button>Lock</button> <button>Delete</button>
2	ass	Public	<button>View</button>	<button>Edit</button> <button>Private</button> <button>Pin</button> <button>Lock</button> <button>Delete</button>
3	cholen	Public	-	<button>Edit</button> <button>Private</button> <button>Pin</button> <button>Lock</button> <button>Delete</button>
4	Cholo jai	Public	-	<button>Edit</button> <button>Private</button> <button>Pin</button> <button>Lock</button> <button>Delete</button>
5	Admission Start	Private	-	<button>Edit</button> <button>Public</button> <button>Pin</button> <button>Lock</button> <button>Delete</button>
6	Admission Start	Private	-	<button>Edit</button> <button>Public</button> <button>Pin</button> <button>Lock</button> <button>Delete</button>

Fig 4.27. Create Notice Page

#### 4.4.5 Application Form

Here need to some steps to fill up the form,that are:

**Postgraduate Admission Application**

[Exit](#)

**1. Program**  
Select Program

**2. Admission Season**  
Select Season

**3. Department**  
Select Department

**4. Supervisor**  
Select Supervisor

**5. Personal Information**

Applicant Name
Father's Name
Mother's Name
Date of Birth
Nationality
Marital Status
Sex

**6. Present Address**

Village / Street
Post Office (P.O)
Postal Code
Sub-District (Upazila)
District

Permanent address same as present

**7. Permanent Address**

Village / Street
Post Office (P.O)
Postal Code
Sub-District (Upazila)
District

**8. Contact Information**

Mobile Number
Email Address

**Course-wise GPA Calculation**

Course code format: CSE101, EEE205  
Minimum required CGPA: 2.75

Bachelor (BSc/BBA/LLB)	Total Credit Hour	Applied Subject	Total Credit Hour
165	4.50		
CCE321	Networking	3	3.25
CCE322	Socket IO	1.5	3.5

[Remove](#)

[+ Add Course](#)

**9. Academic Information**

**SSC Information**

Institute Name
Passing Year
CGPA (out of 5)
Maximum allowed CGPA: 5.00

**HSC Information**

Institute Name
Passing Year
CGPA (out of 5)
Maximum allowed CGPA: 5.00

**BSc Information**

Institute Name
Passing Year
CGPA (out of 4)
Maximum allowed CGPA: 4.00

**Credit Summary**

- Bachelor Credit Hour: 165
- Applied Subject Credit Hour: 4.50

Calculated CGPA: 3.33  
✓ You meet the minimum CGPA requirement

Fig 4.28. Application Form Fillup Page

**Required Documents**

Documents are saved temporarily and will be attached when you submit the application.

Choose Files No file chosen

**Declaration**

I do hereby declare that:

(a) The information mentioned in this application is true and correct to the best of my knowledge. If any information is found to be false, my candidature shall automatically be cancelled;

(b) I shall abide by all the rules and regulations of the University if admitted; and

(c) I shall not take part in any activity subversive of the University, the State, or of discipline.

I have read and agree to the above declaration.

**Pay Application Fee (৳100)**

**Submit Application**

Demo

Support FAQ Offers Login

CARDS MOBILE BANKING NET BANKING

PAY 100 BDT

### OTP Page

**Do not press browser back or forward button while you are in payment page**

Payment Summary	
Please review the following detail for this transaction:	
Amount:	100.00
Invoice number:	260131204653MXArsOpGS14UCdk
Description:	Products

Enter Card Information	
OTP: <input type="text"/> <div style="display: flex; justify-content: space-around; width: 100%;"> <span>Success</span> <span>Failed</span> </div> <div style="display: flex; justify-content: space-around; width: 100%;"> <span>Success with risk</span> </div>	<p>Your entered card information could not be corrupted or become known to the third party, as all transmitted data is encrypted by the SSL protocol.</p> <p><b>Note</b></p> <ol style="list-style-type: none"> <li>For VISA and MC, look at the back side of your Card to find 3-digit CVV2/ CVC2. For AMEX, look at the upper right corner of the front side of your Card to find 4-digit CSC.</li> <li>The cardholder's name should be entered just as it's written on the card.</li> </ol>

SSL COMMERCIAL GATEWAY. NO CARD INFORMATION WILL BE SAVED AND DUMMY.

Fig 4.29. Payment Page

**Required Documents**

Upload each required document in its respective field.  
**Note:** For SSC & HSC, upload either certificate or mark sheet.

**SSC Level** (Any one required)

SSC Certificate

Choose File No file chosen

SSC Mark Sheet

Choose File No file chosen

Upload at least one document from this section

**HSC Level** (Any one required)

HSC Certificate

Choose File No file chosen

HSC Mark Sheet

Choose File No file chosen

Upload at least one document from this section

**Mandatory Documents**

**Bachelor Certificate**

Choose File No file chosen

Not uploaded

**Bachelor Transcript**

Choose File 2002058-Rakib.pdf

Uploaded (184 KB)

**Nationality Certificate**

Choose File PSTU Inventory Management System .pdf

Uploaded (2572 KB)

**Testimonial**

Choose File 1726555846\_Student List\_20240917\_0001.pdf

Uploaded (5914 KB)

**Passport Photo**

Choose File full.drawio.png

Uploaded (112 KB)

Total Uploaded Size: 8.58 MB (Max 100 MB)

Fig 4.30.. Upload File Section

## 4.5 Implementation

The PSTU PGS Management System has been developed using a combination of modern technologies and development tools to ensure scalability, security, and maintainability. Below is a detailed explanation of the backend and frontend technologies, along with the implementation strategies used for building the system.

### 4.5.1 Backend

The backend of the PSTU PGS Management System is developed using a robust and secure technology stack to ensure scalability, reliability, and maintainability. The backend is responsible for handling authentication, admission applications, academic records, document uploads, payment processing, approval workflows, and notification services. It is designed to efficiently manage concurrent requests and ensure secure communication between the frontend and database.

Below is a detailed description of the backend technologies used in the system:

- Node.js:
  - Node.js is used as the runtime environment for the backend. Its non-blocking, event-driven architecture enables the system to handle multiple user requests simultaneously, making it suitable for high-traffic scenarios during admission periods.
  - Version: 16.x
- Express.js:
  - Express.js is used as the core web framework for building RESTful APIs. It manages routing, middleware, request handling, and response processing between the frontend and backend components of the system.
  - Version: 4.x
- MongoDB:
  - MongoDB is used as the primary NoSQL database for storing and managing system data, including users, admission applications, academic records, documents, payments, approval logs, enrollment records, and notices. Its flexible schema design supports the complex and evolving data requirements of postgraduate admission workflows.
  - Database Type: NoSQL
  - Version: 5.x
- Mongoose:
  - Mongoose is used as an Object Data Modeling (ODM) library to interact with MongoDB. It provides schema definitions, data validation, and an abstraction layer for performing CRUD operations, ensuring data consistency and integrity.
  - Version: 9.x

- **JWT (JSON Web Tokens):**
  - JWT is used for secure user authentication and authorization. Upon successful login, a token is generated and used to manage user sessions and enforce role-based access control throughout the system.
  - Version: 9.x
- **Postman:**
  - Postman is used during the development phase to test and validate RESTful API endpoints of the PSTU PGS Management System. It allows developers to simulate HTTP requests, verify request-response behavior, test authentication mechanisms, and ensure that backend services function correctly before integration with the frontend.
  - Purpose: API testing and validation
  - Version: 8.x

#### **Password Encryption (bcrypt / bcryptjs):**

User passwords are securely hashed using bcrypt and bcryptjs before being stored in the database. This ensures strong protection against unauthorized access and password-related vulnerabilities.

#### **File Upload Handling (Multer):**

Multer is used to handle multipart/form-data requests, enabling applicants to upload academic documents securely. File type and size validation are enforced to ensure system security and data integrity.

#### **Email Service (Nodemailer):**

Nodemailer is used to send automated email notifications to applicants and academic authorities. Emails are triggered for events such as application submission, payment confirmation, approval decisions, waiting list updates, and enrollment notifications.

#### **Payment Gateway Integration (SSLCommerz Sandbox):**

The system integrates the SSLCommerz sandbox payment gateway to process application and enrollment fees. It handles payment initiation, transaction verification, and secure storage of payment records.

#### **Security Middleware:**

Several middleware packages are used to enhance system security:

- **Helmet:** Protects against common HTTP vulnerabilities
- **CORS:** Controls cross-origin resource sharing
- **Express Rate Limit:** Prevents brute-force and excessive requests
- **Mongo Sanitize:** Protects against NoSQL injection
- **XSS Clean & HPP:** Prevent cross-site scripting and HTTP parameter pollution.

## **Key Backend Functionalities:**

### User Registration and Account Management:

Users such as applicants and academic authorities can access the system through secure accounts created by the admin or through the application workflow. User credentials are securely stored using bcrypt/bcryptjs password hashing to protect sensitive information. The backend manages user roles, access permissions, and tenure-based academic positions to ensure proper authorization across the system.

### Admission Application Processing:

The backend handles the complete admission application lifecycle, including storing applicant information, validating academic records, generating unique application numbers, and updating application statuses. All application data is securely stored and managed through MongoDB collections.

### Payment Processing and Verification:

Applicants can initiate application fee payments through the integrated SSLCommerz sandbox payment gateway. The backend processes payment requests, verifies transaction status, stores payment records, and links successful payments to corresponding admission applications before allowing further processing.

### Application Review and Approval Workflow:

The backend coordinates the multi-level approval process involving Supervisors, Chairmen, and Deans. It records decisions at each stage, updates application statuses, and maintains detailed approval logs to ensure transparency and accountability throughout the decision-making process.

### Email Notification Services:

Automated email notifications are sent to applicants and academic authorities for key events such as application submission, payment confirmation, approval decisions, waiting list updates, and enrollment-related communications. Email delivery is handled using Nodemailer, ensuring timely and reliable communication without disrupting core system operations.

#### **4.5.2 Frontend**

The frontend of the PSTU PGS Management System is developed using modern web technologies to deliver a responsive, interactive, and user-friendly interface for applicants, administrators, and academic authorities. The frontend focuses on usability, role-based navigation, and real-time interaction with backend services, ensuring a smooth experience throughout the postgraduate admission process.

- React.js:
  - React.js is used as the core frontend library to build a component-based user interface. Its declarative architecture allows efficient rendering of dynamic data such as application status, approval progress, notices, and dashboards for different user roles.
  - Version: 17.x
- Vite:
  - Vite is used as the frontend build tool, providing fast development startup, hot module replacement, and optimized production builds. It significantly improves development speed and overall performance.
  - Version: 7.x
- Redux Toolkit & React-Redux:
  - Redux Toolkit, along with React-Redux, is used for global state management. It manages shared application state such as authenticated user information, application data, admission seasons, notices, and dashboard statistics, ensuring consistency across different components.
  - Redux Toolkit Version: 2.x
  - React-Redux Version : 9.x
- React Router DOM:
  - React Router DOM is used for client-side routing, enabling smooth navigation between different panels such as Applicant Panel, Admin Panel, Supervisor Panel, Chairman Panel, and Dean Panel without full page reloads.
  - Version: 7.x
- Bootstrap CSS:
  - Bootstrap is used for responsive UI design and layout styling, while React-Bootstrap provides reusable UI components such as forms, tables, modals, and navigation elements. This ensures consistency, responsiveness, and ease of use across devices.
  - Bootstrap Version: 5.x
  - React-Bootstrap Version: 2.x
- Axios:
  - Axios is used for handling asynchronous HTTP requests between the frontend and backend REST APIs. It manages secure data exchange for user authentication, application submission, payment verification, approval workflows, and real-time updates.
  - Version: 1.x

## **Key Frontend Components:**

### **1. User Registration & Login:**

The system provides a secure authentication mechanism for all users, including applicants, administrators, and academic authorities. User accounts are created by the admin or through the admission workflow, and login is handled using JWT-based authentication. Passwords are securely hashed before storage, and OTP-based verification is supported for password recovery. Upon successful login, users are redirected to their respective role-based dashboards.

### **2. Admission Application Interface:**

The admission application interface allows applicants to submit postgraduate admission applications by providing personal, academic, and program-related information. Applicants can upload required documents, enter academic records, and track application progress. Each submission is validated and stored securely, and a unique application number is generated for tracking purposes.

### **3. Admin Dashboard:**

The admin dashboard serves as a centralized control panel for managing the postgraduate admission system. Through this interface, administrators can create and manage users, configure admission seasons, define department-wise registration number ranges, publish notices, and monitor overall system activity. The dashboard provides summary statistics and real-time insights to support efficient administrative decision-making.

### **4. Application Submission and Tracking**

This feature enables applicants to monitor the status of their submitted applications throughout the admission process. Applicants can view updates such as submission confirmation, payment status, approval decisions, or placement on the waiting list. Automated email notifications ensure that applicants remain informed at every stage of the process.

# **Chapter 5: Testing & Security**

## **5.1 Introduction**

This chapter focuses on the testing strategies and security measures implemented in the PSTU PGS Management System. It provides sensitive postgraduate admission data, user credentials, academic records, document uploads, and online payment information. This chapter outlines the security strategies, testing methodologies, and best practices followed during the design, development, and testing phases to maintain data integrity, prevent unauthorized access, and provide a secure and trustworthy user experience throughout the postgraduate admission process.

## **5.2 Security Strategies**

To maintain a secure environment, several strategies were employed throughout the design and development of the PSTU PGS Management System:

### **5.2.1 Authentication and Authorization**

**JWT (JSON Web Tokens):** JWT is used for user authentication in the system. When users log in, they receive an access token, which is used to verify their identity during subsequent requests. This ensures that only authorized users can access specific features and data.

**Role-Based Access Control (RBAC):** RBAC is implemented to restrict access to sensitive features based on the user's role. For instance, admin users have full access to the system's features, while department heads and staff have restricted access based on their responsibilities.

### **5.2.2 Sandboxed Code Execution**

The system leverages Postman to create a secure environment for handling sensitive data and admission and related data handling. This ensures that errors, invalid requests, or unauthorized actions are properly handled without affecting the stability of the overall system. By isolating and validating each operation through role-based access control and secure API handling, the system maintains data integrity and prevents unintended system behavior.

This approach provides a safe and reliable environment for managing sensitive postgraduate admission data, ensuring that academic records, payment information, and approval decisions are processed securely and consistently throughout the system lifecycle.

### **5.2.3 Rate Limiting and API Throttling**

Rate Limiting: To prevent abuse and ensure system stability, rate limiting is applied to critical API endpoints within the PSTU PGS Management System. This mechanism restricts the number of requests a user can make within a specific time frame, protecting the system from excessive login attempts, repeated application submissions, or unauthorized request flooding. By controlling request rates, the system safeguards server resources and maintains reliable performance for all users.

API Throttling: API throttling is implemented to ensure that no user exceeds a predefined number of requests within a given time period. This mechanism helps protect the PSTU PGS Management System from potential denial-of-service (DoS) attacks and prevents misuse of system resources. By regulating request frequency, the system ensures fair resource distribution, maintains consistent performance, and provides a stable experience for all users.

### **5.2.4 Data Encryption**

Encryption of Sensitive Data: Sensitive information such as user passwords, authentication tokens, and personal data is securely protected before being stored in the database. Passwords are hashed using strong cryptographic algorithms, ensuring that original credentials cannot be retrieved even in the event of unauthorized access. This approach helps safeguard user data, maintain confidentiality, and uphold the overall security of the PSTU PGS Management System.

HTTPS: All communications between the frontend and backend are encrypted using HTTPS, ensuring that data is securely transmitted between users and the server without being intercepted by malicious third parties.

## **5.3 XP Practices**

The development of the PSTU PGS Management System followed an iterative and incremental development approach, emphasizing continuous testing, close collaboration, and regular feedback throughout the development process. This approach ensured that system features were developed, tested, and refined in small cycles, allowing issues to be identified early and new requirements to be incorporated efficiently. As a result, the system remains flexible, maintainable, and capable of adapting to future enhancements in postgraduate academic and administrative processes.

### **5.3.1 Test-Driven Development (TDD)**

TDD was employed to write tests before implementing new features. This approach allowed developers to identify potential issues early in the development process, ensuring that new features were fully tested and worked as expected.

Tests were written for both the backend and frontend components of the system, covering scenarios such as:

- User authentication and role management
- Admission application submission and status updates
- Approval workflow processing by supervisors, chairmen, and deans
- Payment verification and transaction handling
- Database interactions and CRUD operations
- Data validation for academic records, documents and application inputs

### **5.3.2 Continuous Integration (CI)**

CI pipelines were integrated to automate the testing process whenever new code was pushed to the repository. This ensured that any new code met the required quality standards before being deployed to the production environment.

Automated tests included:

Unit, integration, and user interface testing were conducted to ensure the reliability of the PSTU PGS Management System. Unit tests verified individual backend functions such as authentication logic, application validation, CGPA calculation, and approval handling. Integration tests ensured proper interaction between API endpoints, including application submission, payment verification, approval workflows, and notification services. Additionally, UI testing validated frontend components such as login forms, application submission interfaces, dashboards, and status tracking views to ensure a smooth and error-free user experience.

### **5.3.3 XP Testing**

XP testing practices were employed to ensure that the PSTU PGS Management System met high standards of quality and functionality. Key aspects of the testing process included.

### **5.3.4 Unit Testing**

Unit tests were developed for the backend functions using Jest to ensure that each module performed as expected. These tests validated the core features, such as:

User authentication

Admission application submission and processing

Database CRUD operations

For frontend using React testing Library to ensure that user interface elements such as forms, buttons, validation messages and navigation flows behaved correctly under different scenarios.

### **5.3.5 Integration Testing**

Integration tests were conducted to ensure that different parts of the system worked together seamlessly. For example: Testing the interaction between the frontend and backend, such as user registration, login, and admission application submission and processing. Verifying that the API endpoints correctly processed requests, interacted with the database, and returned the appropriate responses.

### **5.3.6 End-to-End (E2E) Testing**

End-to-End (E2E) testing was conducted using Cypress to simulate real user interactions with the system. E2E tests helped verify the overall workflow of the system, including:

- Admission application submission and approval workflows
- Application status updates and payment verification
- User login and role-based dashboard interactions

These tests helped identify issues that could only be detected when testing the entire workflow, ensuring that users had a smooth and error-free experience when interacting with the system.

## **5.4 Summary**

This chapter presented the security strategies and testing methodologies implemented in the PSTU PGS Management System to ensure a secure, reliable, and robust platform for postgraduate admission management. Security mechanisms such as JWT-based authentication, role-based access control, rate limiting, and secure password hashing were applied to protect sensitive user credentials, academic records, and payment-related data. An iterative development approach inspired by Agile practices, including Test-Driven Development (TDD), was followed to emphasize quality and early issue detection. Comprehensive testing—covering unit testing, integration testing, and user interface testing—ensured that system components functioned correctly and interacted seamlessly. Together, these practices contribute to a secure, stable, and user-friendly system that effectively supports the postgraduate admission process at Patuakhali Science and Technology University.

# **Chapter 6: Conclusion And Future Work**

## **6.1 Introduction**

This chapter provides an overview of the outcomes achieved through the development of the PSTU PGS Management System, along with insights into potential future enhancements. The project was designed to digitize, secure, and streamline the postgraduate admission process at Patuakhali Science and Technology University (PSTU) by replacing fragmented and manual procedures with a centralized, automated platform. This chapter discusses the overall results of the system implementation, highlights key achievements, and outlines possible directions for future development to further enhance postgraduate academic and administrative management.

## **6.2 Project Outcomes**

The PSTU PGS Management System has successfully achieved its primary objectives by providing a secure, centralized, and automated platform for managing postgraduate admission processes at Patuakhali Science and Technology University (PSTU). The key outcomes of the project are summarized below:

### Automated Admission Management:

The system automates the postgraduate admission workflow, including application submission, document handling, payment processing, and approval tracking. This significantly reduces manual effort, paperwork, and administrative delays while minimizing human error.

### Real-Time Application Status Tracking:

Applicants can track their admission status in real time, including submission confirmation, payment verification, approval decisions, and waiting list updates. Automated email notifications further enhance transparency and communication throughout the process.

### Role-Based Access Control:

The implementation of role-based access control ensures that system functionalities are securely restricted according to user roles such as Admin, Supervisor, Chairman, Dean, and Applicant. This maintains data integrity, confidentiality, and accountability across all admission-related operations.

#### Centralized Administrative Control:

Administrators can efficiently manage users, admission seasons, department-wise registration number ranges, notices, and overall system configuration from a single dashboard. This centralized control improves coordination and operational efficiency.

#### User-Friendly and Responsive Interface:

The system features an intuitive and responsive interface developed using modern frontend technologies, making it easy for applicants and academic authorities to navigate, submit data, review applications, and perform administrative tasks with minimal learning effort.

#### Secure and Scalable Solution:

Built using open-source technologies and modern security practices, the system provides a cost-effective, scalable, and secure solution suitable for long-term deployment within the university environment. It establishes a strong foundation for future expansion to cover additional postgraduate academic processes.

### **6.3 Future Works**

Although the PSTU PGS Management System currently focuses on automating the postgraduate admission process, several important extensions and enhancements are planned for future development to make the system a complete postgraduate academic management platform.

One major area of future work is the full implementation of the student enrollment module. After final admission approval, the system will support a comprehensive enrollment workflow involving temporary student login, hall approval by the Provost, verification by the Registrar, PGS Specialist, and Registrar Office, and final confirmation as an enrolled postgraduate student. Upon successful enrollment, the system will automatically generate and assign a registration number, enabling students to access the dedicated Student Panel.

Another significant future enhancement includes the development of post-enrollment postgraduate study (PGS) management features. These will cover essential academic processes such as thesis proposal submission, thesis progress tracking, supervisor approvals, final thesis submission, defense scheduling, and result publication. In addition, modules for stipend management, research grants, and academic notifications can be integrated to support students throughout their postgraduate journey.

The system can also be further enhanced through advanced reporting and analytics, providing insights into admission trends, enrollment statistics, department-wise performance, and academic outcomes. Integration with existing university systems and expansion of mobile-friendly and performance-optimized features will improve accessibility and usability.

Overall, the PSTU PGS Management System has strong potential for future expansion into a complete, end-to-end postgraduate academic management solution, significantly improving efficiency, transparency, and digital transformation at Patuakhali Science and Technology University.

#### **6.4 Summary**

The PSTU PGS Management System has successfully established a secure, efficient, and user-friendly digital platform for managing postgraduate admission processes at PSTU. By automating application submission, payment handling, approval workflows, and real-time status tracking, the system has effectively addressed the limitations of manual and fragmented admission procedures. It has reduced administrative workload, improved transparency, and enhanced communication among applicants, academic authorities, and administrators.

While there remain opportunities for further expansion—particularly in full enrollment management and post-enrollment postgraduate academic processes—the current implementation represents a significant step toward the digital transformation of postgraduate studies at PSTU. The proposed future work provides a clear roadmap for system enhancement, ensuring long-term scalability and alignment with the evolving academic and administrative needs of the university.

## References

- [1] JavaScript Info, “*JavaScript Guide*,” 2024. [Online]. Available: <https://javascript.info/>
- [2] React Documentation, “*React – A JavaScript Library for Building User Interfaces*,” 2024. [Online]. Available: <https://react.dev/>
- [3] Vite Documentation, “*Vite: Next Generation Frontend Tooling*,” 2024. [Online]. Available: <https://vitejs.dev/>
- [4] Redux Toolkit Documentation, “*Redux Toolkit Official Documentation*,” 2024. [Online]. Available: <https://redux-toolkit.js.org/>
- [5] React Router Documentation, “*React Router*,” 2024. [Online]. Available: <https://reactrouter.com/>
- [6] Bootstrap Documentation, “*Bootstrap CSS Framework*,” 2024. [Online]. Available: <https://getbootstrap.com/>
- [7] Axios Documentation, “*Axios HTTP Client*,” 2024. [Online]. Available: <https://axios-http.com/>
- [8] Node.js Documentation, “*Node.js Official Documentation*,” 2024. [Online]. Available: <https://nodejs.org/en/docs>
- [9] Express.js Documentation, “*Express – Node.js Web Framework*,” 2024. [Online]. Available: <https://expressjs.com/>
- [10] MongoDB Documentation, “*MongoDB Manual*,” 2024. [Online]. Available: <https://www.mongodb.com/docs/>
- [11] Mongoose Documentation, “*Mongoose ODM for MongoDB*,” 2024. [Online]. Available: <https://mongoosejs.com/docs/>
- [12] JSON Web Token Documentation, “*JWT Introduction*,” 2024. [Online]. Available: <https://jwt.io/introduction/>
- [13] SSLCommerz Documentation, “*SSLCommerz Payment Gateway*,” 2024. [Online]. Available: <https://developer.sslcommerz.com/>
- [14] Nodemailer Documentation, “*Nodemailer Email Library*,” 2024. [Online]. Available: <https://nodemailer.com/>
- [15] Postman Documentation, “*Postman API Platform*,” 2024. [Online]. Available: <https://learning.postman.com/>