**GPGCWS WEB PORTAL**

**Final Year Project Report**

****

**SUBMITTED BY:**

Ayesha Kiran (2021-ks-572)

Nazli Sultana (2021-fs-033)

# **Supervisor:** Miss Fareeha Hafeez

**Signature:**

**Bachelor of Science in Computer (2021-2025)**

## 

## **DECLARATION**

The work reported in this project is completed by **Ayesha Kiran** (Roll No. 071372), and **Nazli Sultana** (Roll No. 071367) under the supervision of **Miss Fareeha Hafeez**

Lecturer in Computer Science at **Government Graduate College For Women, Sheikhupura.**

We hereby declare that, we want to make it clear that we didn't copy any part of this software from anywhere else. We created this software and the report entirely on our own. If it's found that any part of this project is copied from somewhere else, we'll take responsibility for it. We haven't used any of this work in any other degree application at any university or institute.

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** | **Registration No** | **Roll no** | **Signature** |
| Ayesha Kiran | 2021-KS-572 | 071372 |  |
| Nazli Sultana | 2021-FS-033 | 071367 |  |

**STATEMENT OF SUBMISSION**

This is to certify that following students have successfully completed the final project named as: **GPGCWS Web Portal** at **Govt. Graduate College For Women, Sheikhupura** to fulfill the partial requirement of the degree of Bachelors of Computer Science.

|  |  |  |  |
| --- | --- | --- | --- |
| **Serial no** | **Reg. no.** | **Roll no.** | **Student Name** |
| 1 | 2021-KS-572 | 071372 | Ayesha kiran |
| 2 | 2021-FS-033 | 071367 | Nazli Sultana |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Head of Department Supervisor**

Mam Ume Hani Mam Fareeha Hafeez

Govt. Graduate College Govt. Graduate College

Women, Sheikhupura Women, Sheikhupura

## 

## **CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BSCS “**GPGCWS Web Portal**” was developed under the supervision of **“Miss Fareeha Hafeez”** by:

|  |  |  |  |
| --- | --- | --- | --- |
| **Serial no** | **Reg. no.** | **Roll no.** | **Student Name** |
| 1 | 2021-KS-572 | 071372 | Ayesha kiran |
| 2 | 2021-FS-033 | 071367 | Nazli Sultana |

It is fully adequate, in scope and quality for the degree of Bachelor of Science in Computer Science.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### **Head of Department Supervisor**

Mam Umme Hani Mam Fareeha Hafeez

Govt. Graduate College Govt. Graduate College Women, Sheikhupura Women, Sheikhupura

**ACKNOWLEDGEMENTS**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor **“Miss Fareeha Hafeez”** without her supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to her for the encouragement and continual help during this work.

Moreover, we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

|  |  |  |  |
| --- | --- | --- | --- |
| **Serial no** | **Reg. no.** | **Roll no.** | **Student Name** |
| 1 | 2021-KS-572 | 071372 | Ayesha Kiran |
| 2 | 2021-FS-033 | 071367 | Nazli Sultana |

## **PREFACE**

## Thanks for your interest in this project report. This is a final year project report written by students of **Bachelor of Science in Computer Science** at the **University of the Punjab**. We are **Ayesha Kiran** and **Nazli Sultana**; both of us are studying in BSCS (Hons) of **Government. Graduate College Women, Sheikhupura.**

## This is a technical report, we will explain the idea of our project and the implementation process in detail and systematically. To enable you to understand our project more clearly. This report is divided into 9 topics; each topic with a specific focus. We hope this approach allows you can read our project report easily. Lastly, if you have any comments, you are Welcome, feel free to contact us at:

Group Member 1: Ayesha Kiran ayeshakiran3562@gmail.com

Group Member 2: Nazli Sultana nazlisultana123@gmail.com

**Table of content**

**ABSTRACT**

**FYDP Overview................................................................................................. 10**

**1. Introduction ................................................................................ 12**

1.1 Background ................................................................................................ 12

1.2 Problem Statement .................................................................................... 12

1.3 Objectives .................................................................................................. 13

1.4 Scope.......................................................................................................... 13

1.5 Limitations .................................................................................................. 13

1.6 Methodology Summary .............................................................................. 13

1.7 Tools and Techniques ................................................................................ 14

**2. Literature Review ....................................................................... 16**

2.1 Literature Review........................................................................................ 16

2.2 Related Work ............................................................................................. 16

2.3 Summary of Similar Existing Systems or Research .................................. 17

2.4 Analysis of Gaps or Limitations in Current Solutions ................................. 18

2.5 Justifications for Your Approach ................................................................ 19

**3. System Analysis ........................................................................ 21**

3.1 Requirements Gathering ........................................................................... 21

3.2 Functional Requirements........................................................................... 21

3.3 Non-Functional Requirements .................................................................. 25

3.4 Use Case Diagrams and Descriptions ...................................................... 27

3.5 Process Methodology…………………………………………………………. 33

**4. System Design ........................................................................... 35**

4.1 System Architecture ................................................................................... 35

4.2 Database Design ........................................................................................ 36

4.3 Interface Design .......................................................................................... 37

4.4 Component Design ..................................................................................... 38

4.5 UML Diagrams ………………………………………...................................... 40

**5. Implementation ........................................................................... 51**

5.1 Development Tools and Environment ......................................................... 51

5.2 Modules Implemented ................................................................................. 52

5.3 Screenshots with Brief Descriptions ............................................................ 53

**6. Testing .......................................................................................... 56**

6.1 Testing Techniques Used ............................................................................ 56

6.2 Test Cases and Test Plans .......................................................................... 58

6.3 Testing…………………………………………………………………………….. 61

6.3 Results Summary ……….............................................................................. 62

6.4 Bug Report Summary……………………………………………………………. 63

**7. Project Costing…………………………………………………………..64**

7.1 Project Costing using Function Point Analysis………………………………….. 64

7.2 Project Activities and Critical Path Method…………………………………….… 66

7.3 Gantt Chart………………………………………………………………………….. 67

## **8. Results and Discussion ................................................................. 69**

8.1 Performance Metrics ....................................................................................... 69

8.2 Evaluation of Results ...................................................................................... 71

8.3 Conclusion …………………………...……………............................................. 72

**9. Conclusion and Future Work ………............................................. 73**

9.1 Summary of Achievements ............................................................................ 73

9.2 Challenges Faced .......................................................................................... 74

9.3 Suggestions for Further Improvement ............................................................ 75 **10. References .....................................................................................76**

## **ABSTRACT**

## The **GPGCWS Web Portal** is a comprehensive online platform designed to facilitate seamless access to college-related information for three types of users: **Visitors, Students, and Staff**. Each user role has specific functionalities that ensure efficient information management, academic support, and administrative control.

**Visitors** can explore the college website without logging in. They can view the **home page**, check details about **facilities**, read about the **admission process**, stay updated with **news and events**, and explore **academic programs**.

**Students** can log in to access academic resources. They can view and download **study materials, notes, books**, and watch **lectures** to enhance their learning experience.

**Staff members** have administrative privileges to **update website content**, including **all pages, like news, academic details, and study materials etc**. They can also **view student identities**. Additionally, staff can **review and process admission applications** of new students and update them via emails.

The **GPGCWS Web Portal** ensures a **structured, user-friendly, and secure** digital experience for all users, making college-related information easily accessible while streamlining administrative tasks.

**FYDP OVERVIEW**

**FYDP Title**

GPGCWS Web Portal

**Project Team**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **Roll Numbers** | **Name** | **Signatures** |
| 1. | 071372 | AYESHA KIRAN |  |
| 2. | 071367 | NAZLI SULTANA |  |

**Project Goals**

* Establish an online platform for students and guests.
* Enable effortless access to admissions information and study material.
* Make it user-friendly and accelerate the process of data handling.

**Project Objectives**

* Introduce an online admission system for visitors.
* Enable students to download books, notes, and audio lectures.
* Provide live updates on events and announcements.

**Project Success Criteria**

* Successful web portal deployment with basic functionalities.
* User authentication and controlled access.
* Effective management of student records and academic material.
* Customer satisfaction feedback from students and visitors.

**Assumptions**

* Availability of internet to students and visitors.
* The institution shall avail resources required for development and maintenance.

**Risks & Obstacles**

* Security risks that must be updated regularly and monitored constantly.
* Adoption issues by the users that necessitate training and awareness programs.

**Organization Address**

University of the Punjab, Lahore, Pakistan

**Target End Users**

Students, Visitors, Staff

**Suggested Project Supervisor**

Mam FAREEHA HAFIZ

**Approved By**

Mam FAREEHA HAFIZ

**Date**

Febrary 02, 2025

# 

# **Chapter No 1**

# **INTRODUCTION**

**Project Title*:*** GPGCWS Web Portal

## **1.1.** **Background**

In today’s digital era, educational institutions increasingly rely on online platforms to streamline their administrative and academic operations. Traditional methods of student admissions, record management, and academic resource sharing are often inefficient, time-consuming, and prone to errors. The **GPGCWS Web Portal** aims to overcome these challenges by providing a centralized online platform for students, staff, and visitors. This system will enhance the college’s digital presence by offering an intuitive interface for admissions, study materials, and information sharing. By integrating advanced web technologies, the portal will ensure a seamless user experience, improving accessibility, operational efficiency, and communication across the institution. This project will not only modernize the college’s workflow but also offer students and staff a secure online platform for accessing academic resources, managing student records, and receiving institutional announcements and updates.

## **1.2. Problem Statement**

Colleges and universities often struggle with managing student admissions, academic records, and communication effectively. Manual processes lead to delays, mismanagement, and inaccessibility for students and staff. Some of the key challenges include:

* **Inefficient Admission Process:** Manual admissions require extensive paperwork and manual verification, increasing errors and delays.
* **Limited Access to Study Materials:** Students face difficulties in finding centralized repositories for notes, books, and recorded lectures.
* **Ineffective Communication:** The absence of a common platform for announcements and academic news results in students missing essential updates.
* **Data Management Challenges:** Manual student record handling leads to inefficiencies and limits tracking of academic progress.

## **1.3. Objectives**

The **GPGCWS Web Portal** aims to achieve the following objectives:

* Develop an **online admission system** for student applications, document uploads, and application tracking.
* Provide a **centralized platform** for accessing **study materials, books, and recorded lectures.**
* Implement a **secure role-based login system** for students and staff.
* Enable **real-time communication** for announcements, events, and academic updates.
* Improve **student record management** through digital enrollment and academic tracking.
* Strengthen the **college’s online presence** with a responsive and user-friendly interface.

## **1.4. Scope**

The **GPGCWS Web Portal** will include:

* **Student Portal:** Features for study materials, recorded lectures, and academic tracking.
* **Staff Portal:** Tools for admission processing, student record and login management, and content updates.
* **Public Website:** Anyone can access pages like admission, academic, facilities, news and events.

**1.5. Limitations**

* The portal **will not replace face-to-face lectures** but will serve as a supplementary resource.
* **Core functionalities** will be developed first, with advanced features introduced in later phases.
* System performance will depend on **internet connectivity** and **user devices.**

**1.6. Methodology Summary**

1. **Requirement Analysis** – Identifying user roles, access levels, and functional specifications.
2. **System Design** – Utilizing Figma for UI/UX wireframes and designing the database architecture for structured data management.
3. **Development** – o **Frontend:** React.js for a dynamic and responsive interface.
   * + **Backend:** Express.js for server-side logic and API management.
     + **Database:** MongoDB for efficient data storage and retrieval.
4. **Authentication & Security** – Implementing role-based authentication for staff and students, alongside data encryption and security best practices.
5. **Version Control & Deployment** – Using GitHub for collaborative development and deploying the portal on Vercel for easy access.
6. **Testing** – Conducting unit, integration, and user acceptance testing to ensure functionality, security, and user satisfaction.
7. **Maintenance & Future Enhancements** – Providing continuous updates, issue resolution, and expanding functionalities such as AI-driven student assistance, automated notifications, and mobile app integration.

**1.7. TOOLS AND TECHNOLOGIES**

**Tools**

|  |  |  |
| --- | --- | --- |
| **Tool** | **Version** | **Rationale** |
| **VS Code** | **v1.85.1 (Latest)** | Used as the primary Integrate Development Environment (IDE) for coding and debugging. |
| **GitHub** | **Cloud-Based (Latest)** | Facilitates version control, collaboration, and project management. |
| **MS Excel** | **Microsoft 365 (Latest)** | Used for creating Gantt charts and project scheduling |
| **Figma** | **v6.0 (Latest)** | Used for UI/UX design and prototyping of the application. |

**Technologies**

|  |  |  |
| --- | --- | --- |
| **Technology** | **Version** | **Rationale** |
| **React.js (Frontend)** | **v18.2.0** | Provides a modern, interactive, and fast user experience. |
| **Node.js (Backend)** | **v20.10.0** | Handles server-side logic efficiently for high performance.. |
| **Express.js** | **v4.18.2** | Simplifies the creation of RESTful APIs for communication between frontend and backend. |
| **MongoDB (Database)** | **v6.0.13** | A NoSQL database that offers high scalability and flexibility for data storage. |
| **Firebase** | **Cloud-Based (Latest)** | Can be used for user authentication. |

**Work Division**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **Roll Number** | **Name** | **Role Assignment & Work Division** |
| 1. | 071372 | AYESHA KIRAN | Frontend + Testing |
| 2. | 071367 | NAZLI SULTANA | Backend + Database |

**Chapter No 2**

**Literature Review/ Related Work**

## **2.1. Literature Review**

The purpose of this literature review is to evaluate existing work related to educational portals and institutional management systems, laying the foundation for the proposed solution. It highlights previous efforts, identifies gaps, and justifies the need for a dedicated web portal tailored to the specific requirements of GPGCWS.

### **2.2. Related Work**

In recent years, numerous institutions have implemented web-based portals to facilitate communication, academic resource sharing, admission management, and administrative efficiency. Various platforms, such as **Campus Management Systems**, **Learning Management Systems (LMS)** like Moodle, and **custom-built portals**, serve similar objectives.

For example, Moodle provides a highly customizable platform for managing study material, assessments, and communication between students and teachers. However, Moodle is often complex to set up and may require technical expertise for maintenance. Additionally, it may offer more features than needed for basic administrative functions, making it less ideal for small to medium sized educational institutions.

Similarly, many colleges in Pakistan rely on static websites that only display basic information. These websites lack dynamic features such as role based access, digital study material distribution, admission verification, and real time updates by faculty members.

Some universities and colleges have adopted custom portals that include admission forms, announcements, and department pages, but these often suffer from outdated interfaces, slow performance, and poor mobile responsiveness. Most importantly, they are not tailored to support both staff content management and student academic support in an integrated manner.

Despite the availability of such solutions, there remains a clear lack of a centralized, lightweight, and user-friendly portal designed specifically for government colleges in Pakistan especially women’s colleges that need streamlined tools for administrative support and educational empowerment.

## **2.3. Summary of Similar Existing Systems or Research**

To better understand the scope and functionality of educational web portals, several existing systems and related research studies were analyzed. This comparison helped identify best practices, common features, and gaps that the GPGCWS Web Portal aims to address.

### **1. Punjab Higher Education Web Portals**

* **Overview:** Many public colleges in Punjab, Pakistan, use government-hosted web portals to display basic college information, admission notices, and faculty details.
* **Limitations:** These portals are often **static**, lacking interactive features such as student login, study material access, or staff content management.
* **Lesson Learned:** The GPGCWS Web Portal incorporates **dynamic and role-based functionality** to overcome these limitations and improve user engagement.

### **2. University of the Punjab – LMS**

* **Overview:** The University of the Punjab offers a **Learning Management System (LMS)** for students and faculty to access course material, submit assignments, and track performance.
* **Strengths:** Offers comprehensive academic tools, secure login, and personalized dashboards.
* **Relevance:** Inspired the inclusion of features like **online study material**, **lecture access**, and **student authentication** in the GPGCWS system.

### **3. Research on E-Learning Platforms**

* **Findings:** According to research studies published in the International Journal of Educational Technology, modern web-based educational platforms must:
  + Be mobile-responsive.
  + Provide user-specific content.
  + Ensure security through proper authentication.
* **Impact:** This research validated the choice of **role-based access**, **JWT authentication**, and **responsive design** in the portal.

### **4. Edmodo & Google Classroom**

* **Overview:** These are global platforms for remote learning, enabling file sharing, announcements, assignments, and student-teacher interaction.
* **Limitations in Context:** These platforms are general-purpose and not localized for specific institutional needs like GPGCWS.
* **Conclusion:** The GPGCWS Web Portal focuses on **institution-specific customization**, combining educational content delivery with admission and verification workflows.

## **2.4. Analysis of Gaps or Limitations in Current Solutions**

In analyzing the existing web portals and learning platforms used by colleges and universities especially within the context of public institutions in Pakistan several limitations were identified. These gaps highlight the need for a more robust, dynamic, and user focused system like the **GPGCWS Web Portal**.

### **1. Lack of Role-Based Access Control**

* **Issue:** Most current systems do not differentiate between user roles (e.g., visitors, students, and staff).
* **Impact:** This limits personalized access, content control, and feature visibility.
* **GPGCWS Solution:** Implements **clear user role segregation** with separate dashboards and permissions for students and staff.

### **2. Static and Outdated Content**

* **Issue:** Many educational websites contain hard-coded content that requires developer intervention for updates.
* **Impact:** Staff cannot independently update notices, study materials, or events.
* **GPGCWS Solution:** Provides **staff content management features** that allow faculty to update homepage, academics, events, and learning resources dynamically.

### **3. No Online Study Resource Access**

* **Issue:** Students in most colleges must rely on physical materials or external platforms (WhatsApp, YouTube) for resources.
* **Impact:** Inconvenient and unorganized learning experience.
* **GPGCWS Solution:** Offers a centralized **study material hub**, including downloadable notes, assignments, and lectures within the portal.

### **4. No Student Verification or Admission Workflow**

* **Issue:** Existing portals rarely include online workflows for student ID verification or application management.
* **Impact:** Increases manual effort and chances of human error.
* **GPGCWS Solution:** Enables **staff-controlled student verification** and a **streamlined admission review process** with email notifications.

### **5. Limited User Feedback Mechanisms**

* **Issue:** Users (especially students) receive minimal feedback while interacting with the system (e.g., no confirmation after submitting a form).
* **Impact:** Leads to confusion and repeated actions.
* **GPGCWS Solution:** Includes **real-time success/error messages** and notifications for all major actions like login, downloads, and form submissions.

### **6. Poor Mobile Compatibility**

* **Issue:** Many existing portals are not mobile-responsive or optimized for various devices.
* **Impact:** Frustrating experience for users on smartphones or tablets.
* **GPGCWS Solution:** Designed with **fully responsive UI** using Tailwind CSS and React to ensure seamless access across all devices.

## **2.5. Justification for Your Approach**

The design and development of the **GPGCWS Web Portal** were guided by specific technical and functional goals, aimed at resolving the limitations found in existing systems. The following justifications explain why the chosen approaches, tools, and methodologies were selected:

### **1. Object-Oriented Programming (OOP) Approach**

* **Justification:**   
  OOP was chosen to structure the system into distinct entities like Student, Staff, and AdmissionApplication, each with specific attributes and behaviors.
* **Benefits:**
  + Promotes modularity and scalability.
  + Simplifies maintenance and updates.
  + Reflects real-world roles and interactions clearly.

### **2. Role-Based Access Control (RBAC)**

* **Justification:**   
  Separating functionalities for Visitors, Students, and Staff improves security, usability, and content control.
* **Benefits:**
  + Ensures users only access relevant features.
  + Supports personalized dashboards.
  + Prevents unauthorized actions like content updates by non-staff.

### **3. MERN Stack (MongoDB, Express.js, React.js, Node.js)**

* **Justification:**  
  The MERN stack was selected for its flexibility, scalability, and full-stack JavaScript development capabilities.
* **Benefits:**
  + React provides responsive, reusable UI components.
  + Node.js and Express allow efficient backend processing.
  + MongoDB offers a flexible schema for dynamic educational content.

### **4. Tailwind CSS for UI Design**

* **Justification:**  
  Tailwind CSS allows rapid and consistent UI development using utility classes.
* **Benefits:**
  + Creates a clean, mobile-responsive interface.
  + Reduces the need for custom CSS, improving maintainability.
  + Ensures consistent styling across all components.

### **5. JWT-Based Authentication**

* **Justification:**  
  JSON Web Tokens (JWT) offer a secure, stateless method for user authentication.
* **Benefits:**
  + Easy to integrate with modern APIs.
  + Enables secure login sessions without server-side storage.
  + Supports token expiration for added security.

### **6. Third Normal Form (3NF) Database Design**

* **Justification:**  
  The database was normalized to 3NF to ensure data consistency and avoid redundancy.
* **Benefits:**
  + Enhances data integrity.
  + Optimizes query performance.
  + Makes it easier to manage relationships between entities like students, departments, and applications.

### **7. Centralized Study Resource System**

* **Justification:**  
  Providing a built-in section for study materials, books, and lectures enhances student engagement and learning efficiency.
* **Benefits:**
  + Organizes content in one place.
  + Reduces reliance on external platforms.
  + Allows easy updates by staff.

# **Chapter No 3**

# **System Analysis**

## **3.1. Requirement Gathering**

Requirement gathering is a crucial phase in the development of the GPGCWS Web Portal, aimed at understanding the needs of its primary users: Visitors, Students, and Staff. Information was collected through informal discussions with students and faculty, as well as observation of existing college processes. Visitors need access to general information such as college overview, departments, facilities, and admission guidelines without requiring login. Students, after authentication, should be able to view and download study materials, assignments, and lectures. Staff users require secure login access to manage website content, verify student IDs, send confirmation emails, and review admission applications. These functional requirements, along with non-functional needs like system security, mobile responsiveness, and ease of use, were documented to guide the system design and ensure that the final product meets the expectations of all users.

**3.2. Functional Requirements**

This is the definition of the functional requirements of the college website. Each requirement is identified uniquely and categorized according to system features. The functional requirements ensure that the system meets the expectations of **General Users (Visitors), Students, and Staff (Faculty/Admin).**

## Functional Requirements Specification Tables are given below:

**FR-1: View College Information**

| **Identifier** | **FR-1** |
| --- | --- |
| **Title** | View College Information |
| **Requirement** | The visitor will be able to see college-related information such as home, facilities, admissions, academics, and news & events pages. |
| **Source** | General User (Visitor) |
| **Rationale** | Gives visitors a general idea about the college and updates. |
| **Business Rule** | Public pages must be viewable without login. |
| **Dependencies** | None |
| **Priority** | High |

### **FR-2: Student Login & Authentication**

| **Identifier** | **FR-2** |
| --- | --- |
| **Title** | Student Login & Authentication |
| **Requirement** | The system will authenticate students with their registered credentials prior to allowing access to student-specific functionality. |
| **Source** | Student |
| **Rationale** | Prevents unauthorized access by only registered students to study material and other resources. |
| **Business Rule** | A student ID and password must be valid. |
| **Dependencies** | None |
| **Priority** | High |

### **FR-3: Access Study Materials**

| **Identifier** | **FR-3** |
| --- | --- |
| **Title** | Access Study Materials |
| **Requirement** | The student shall have the ability to view and download study materials, such as notes, books, and lecture videos. |
| **Source** | Student |
| **Rationale** | Facilitates simple access to study resources. |
| **Business Rule** | Study resources must be made available to logged-in students alone. |
| **Dependencies** | FR-2 (Student Login & Authentication) |
| **Priority** | High |

### **FR-4: Staff Login & Authentication**

| **Identifier** | **FR-4** |
| --- | --- |
| **Title** | Staff Login & Authentication |
| **Requirement** | The system will authenticate staff members prior to providing access to content management capabilities. |
| **Source** | Staff (Faculty/Admin) |
| **Rationale** | Guarantees that legitimate personnel are the only ones allowed to update website content. |
| **Business Rule** | Staff must enter valid credentials |
| **Dependencies** | None |
| **Priority** | High |

### **FR-5: Update Website Content**

| **Identifier** | **FR-5** |
| --- | --- |
| **Title** | Update Website Content |
| **Requirement** | The staff shall be able to update content on the Home, Facilities, Academics, News & Events, and study materials pages. |
| **Source** | Staff (Faculty/Admin) |
| **Rationale** | Guarantees that website information is up to date and accurate. |
| **Business Rule** | Updates to content should be done by authorized staff members only. |
| **Dependencies** | FR-4 (Staff Login & Authentication) |
| **Priority** | High |

### **FR-6: Admission Application Review & Approval**

| **Identifier** | **FR-7** |
| --- | --- |
| **Title** | Admission Application Review & Approval |
| **Requirement** | The staff will be capable of reviewing and approving or rejecting student admission applications. |
| **Source** | Staff (Faculty/Admin) |
| **Rationale** | Ensures accurate screening of new student applications. |
| **Business Rule** | Authorized staff alone can approve or reject applications. |
| **Dependencies** | FR-4 (Staff Login & Authentication) |
| **Priority** | High |

# **3.3. Non-Functional Requirements**

This section details the quality attributes and constraints necessary for the college website to conform to. These non-functional requirements guarantee reliability, usability, security, and performance to improve user experience and system efficiency.

## **1. Reliability**

* The system will have a **Mean Time Between Failures (MTBF) of not less than 500 hours** during normal usage.
* **Incremental backups** will be taken **every Saturday at 9 PM** to avoid data loss.
* The system will **automatically recover from failures within 5 minutes** (**Mean Time to Recover - MTTR**).
* The website will have **99.5% uptime** to allow continuous access for users.
* In case of failure, the system will send an **automatic alert** to the administrator through email.

## **2. Usability**

* The site should have a **friendly and easy-to-use interface** with simple navigation for the general public, students, and employees.
* All functions should be available in **three clicks** from the front page.
* The system will be in **WCAG 2.1 accessibility guidelines** so that it is accessible for people with disabilities.
* The website will be completely **responsive**, and users will have smooth access on **mobile, tablet, and desktop devices**.
* There will be a **help section and user guide** for students and staff.

## **3. Security**

* The system will practice **role-based access control (RBAC)** to limit unauthorized access to sensitive regions.
* All passwords will be **encrypted via SHA-256 hashing** for security.
* **Session timeout automatically after 15 minutes** of inactivity for staff and students.
* The site will employ **HTTPS encryption** for secure transmission.
* **Multi-Factor Authentication (MFA)** will be mandatory for **staff logins** to add security.
* Failed logins will be **restricted to three**, upon which the account will be **temporarily locked for 30 minutes**.
* Members of staff will have **independent permissions** to update content, check student IDs, and authorize admissions.

## **4. Performance**

* The website will **load within 3 seconds** on typical network conditions.
* The system will be capable of processing **1000 requests per minute** without degradation in performance.
* Queries in the database will be optimized to **deliver results within 2 seconds** for fetching study material or admission history.
* The system will have **real-time content updating** capability for staff to make changes to website content in real time.

## **5. Maintainability & Scalability**

* The system will be designed with a **modular architecture** for future updates and feature additions with ease.
* Documentation for code will be kept in order to facilitate future development and debugging.
* The website will be capable of **handling up to 5000 registered students** without affecting performance.
* The system will be capable of **doubling the existing user base** with minimal changes to infrastructure.

## **6.Data Integrity & Backup**

* **Nightly database backup** shall be taken **at midnight** to avoid data loss.
* Data recovery processes shall guarantee **95% data integrity** in the event of system failure.
* The system will avoid **duplicate student registrations** by implementing **unique student ID validation**.
* Admission applications should be kept in **tamper-proof, secure records**, only accessible for modification by staff with authorization.

These non-functional requirements guarantee that the **college website is efficient, scalable, secure, userfriendly, and reliable**, fulfilling the requirements of **general users, students, and staff** and ensuring high performance and accessibility.

# 3.4. Use case Analysis

The GPGCWS Web Portal is designed to cater to three distinct user categories Visitors, Students, and Staff each with specific roles and access privileges. Visitors can explore essential college information such as the overview, principal’s message, notices, facilities, cultural activities, and academic without requiring a login. Students, upon logging in, gain access to study materials, including notes, assignments, reference books, and lectures tailored to their courses. Staff members have the highest level of access, enabling them to manage and update portal content, approve admission applications, and communicate directly with applicants via email after validation.

## **Use Case #1: Visitor Accessing the College Website**

|  |  |
| --- | --- |
| **UC Identifier** | **UC1** |
| **Requirements**  **Traceability** | (Public Page Access) |
| **Purpose** | Allow visitors to browse general information about the college. |
| **Priority** | High |
| **Preconditions** | The website must be live and accessible. |
| **Postconditions** | The visitor successfully views the requested pages. |
| **Actors** | Visitor (Primary), System (Secondary) |
| **Extends** | None |
| **Main Success**  **Scenario** | 1. The visitor opens the college website. 2. The visitor navigates through different sections: 3. **Home Page** (College overview, notices, principal’s   Message ).   1. **Facilities Page** (Computer lab, library, canteen, transport, etc.). 2. **Admission Page** (Admission process, requirements,   Deadlines ).   1. **News & Events Page** (Upcoming events, cultural activities, news). 2. **Academics Page** (Departments, faculty, degree programs). 3. The visitor **has the option to log in** as a student or staff member. |
| **UC Identifier** | **UC1** |
| **Exceptions** | 1. If the website is down, an error message is displayed. 2. If a page is missing, a "Page Not Found" error is shown. |
| **Alternate Flows** | None |
| **Includes** | None |

## 

## **Use Case #2: Student Accessing Study Materials**

| **UC Identifier** | **UC2** |
| --- | --- |
| **Requirements Traceability** | FR-1 (User Authentication), FR-6 (Access Study Material) |
| **Purpose** | Allow students to log in and access study materials such as notes, books, and video lectures. |
| **Priority** | High |
| **Preconditions** | The student must be registered with valid login credentials. |
| **Postconditions** | The student is authenticated and redirected to their dashboard. The student can access, view, and download study materials. |
| **Actors** | Student (Primary), System (Secondary) |
| **Extends** | None |
| **Main Success Scenario** | 1. The student logs in using their credentials.2. The system verifies their credentials using firebase.3. If valid, the system grants access to the student portal.4. The student navigates to the Study Material section to:  a. View Online Resources (Notes, Assignments, Books ).  b. Download Notes & Books for offline study.  c. Watch Lectures. |
| **Exceptions** | 1. If incorrect credentials are entered, an error message is displayed.2. If the student’s account is not verified, they cannot log in. |
| **Alternate Flows** | 1. If the student forgets their password, they can reset it via email verification.2. If a lecture link is broken, the system displays a report option. |
| **Includes** | UC3 (Download Notes & Books) |

## **Use Case #3: Staff Managing Website Content & Admissions**

| **UC Identifier** | **UC3** |
| --- | --- |
| **Requirements Traceability** | FR-1 (User Authentication), FR-7 (Manage Website Content), FR-8 (Approve Admissions) |
| **Purpose** | Allow staff to update website content, verify student ID, and approve admissions. |
| **Priority** | High |
| **Preconditions** | The staff member must have valid login credentials and admin access. |
| **Postconditions** | The staff can edit website content. The staff can view student IDs and approve admissions. |
| **Actors** | Staff (Primary), System (Secondary), Student (Applicant) |
| **Extends** | None |
| **Main Success Scenario** | 1. The staff logs in using email and password.2. The system verifies credentials using firebase and grants admin access.3. The staff selects an action from their dashboard:  a. Update Website Content (Home, Facilities, Academics, News & Events, Study Materials).  b. Verify Student ID .  c. Approve Admission Applications and send a confirmation email. |
| **Exceptions** | 1. If incorrect credentials are entered, an error message is displayed.2. If content update fails due to a server issue, an error message is shown.3.. |
| **Alternate Flows** | 1. The staff can delegate content updates to another staff member.2. If a student application is incomplete, staff can request missing documents. |
| **Includes** | UC4 (Verify Student ID), UC5 (Approve Admissions) |

### 

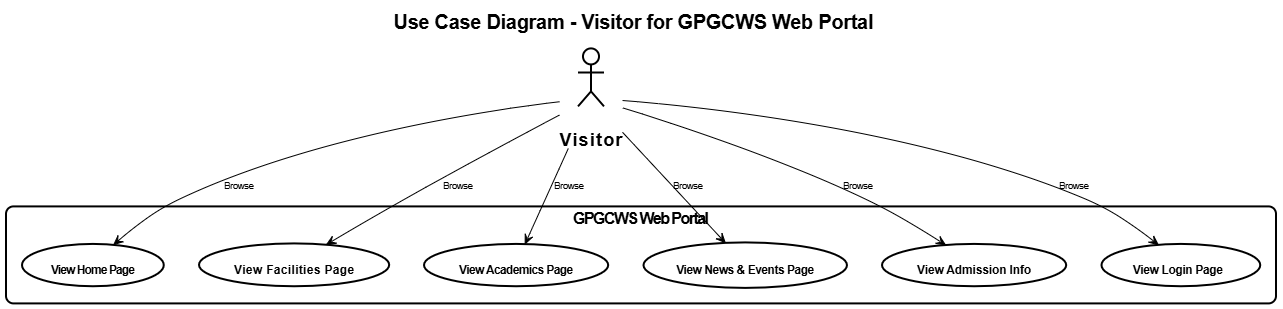
## **Roles & Responsibilities Summary Based on Use Cases**

| **Role** | **Responsibilities & Access** |
| --- | --- |
| **Visitor** | Browse public pages (Home, Facilities, Academics, Admissions, and News & Events). No login required but can choose to log in as student or staff. |
| **Student** | Login Required to access the student portal. View & Download Study Materials (Notes, Books, Assignments). Watch Video Lectures. |
| **Staff** | Login Required to access the admin panel. Update Website Content (Home, Facilities, Academics, News & Events, Study Materials). Verify Student IDs. Approve or Reject Admission Applications and send email confirmation. |

**Use Case Diagram**

**Visitors Use Case Diagram**

The **Visitor** represents a general user who visits the **GPGCWS Web Portal** without logging in. Visitors can access the public sections of the website to view general information about the college, facilities, academic programs, admission procedures, news, and events. The Visitor does not have any authentication privileges or access to internal features like downloading study materials or updating content.

****

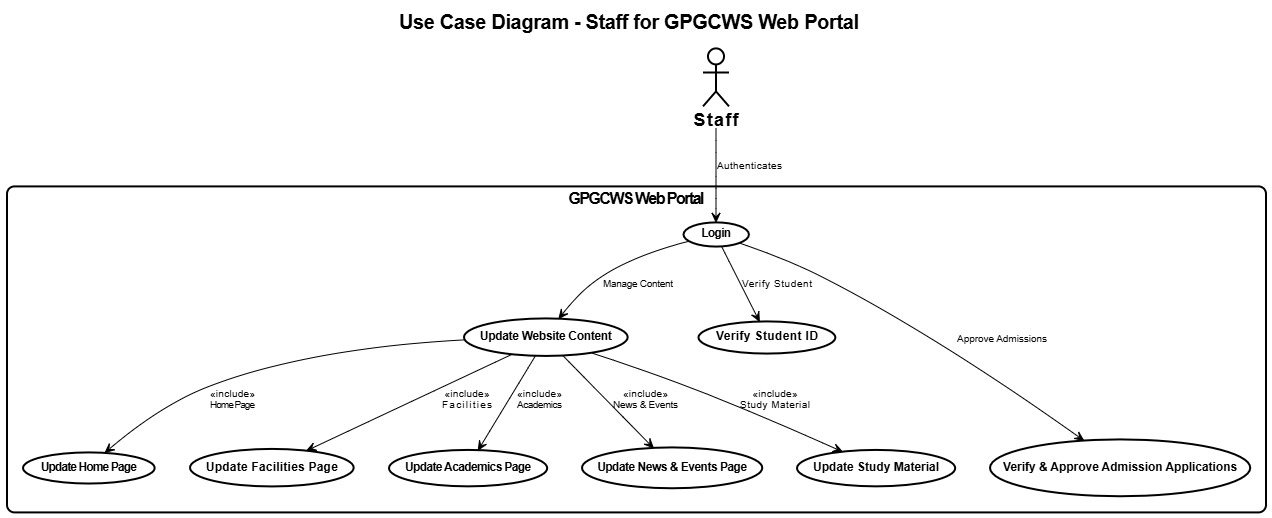
# **Student Use Case Diagram**

# The **Student** is a registered user of the **GPGCWS Web Portal** who can log in with valid credentials. Once logged in, students gain access to study materials, including notes, assignments, and lectures. They can download notes and books, and access video lectures provided on the portal. Students do not have permission to modify content or access administrative features.

# C:\Users\Awais\Downloads\WhatsApp Image 2025-05-31 at 10.48.19_a14e4f02.jpg

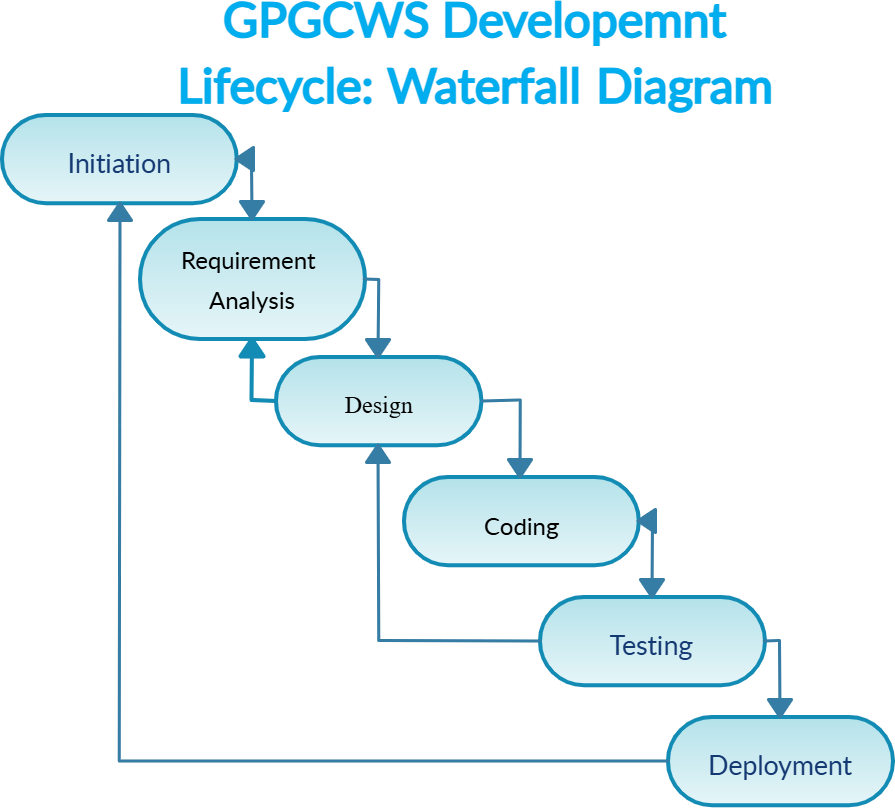
# **Staff Use Case Diagram**

The **Staff** is an authenticated user of the **GPGCWS Web Portal** who has administrative and content management privileges. Staff members can log in to the portal to manage website content, such as updating the Home Page, Facilities, Academics, News & Events, and Study Material sections. They are also responsible for verifying student IDs and reviewing and approving admission applications.

****

**3.5. Software Process Methodology**

In the development of the **GPGCWS Web Portal**, the Waterfall process methodology has been adopted to ensure a structured and systematic approach. Each phase of the project such as requirements gathering, system design, implementation, testing, and deployment will be completed one at a time, in a linear sequence. This model is well-suited for the GPGCWS Web Portal, as the requirements are clearly defined and stable, minimizing the need for changes once a phase is complete. This approach ensures clarity in planning and efficient management of each development stage.



**Chapter No 4**

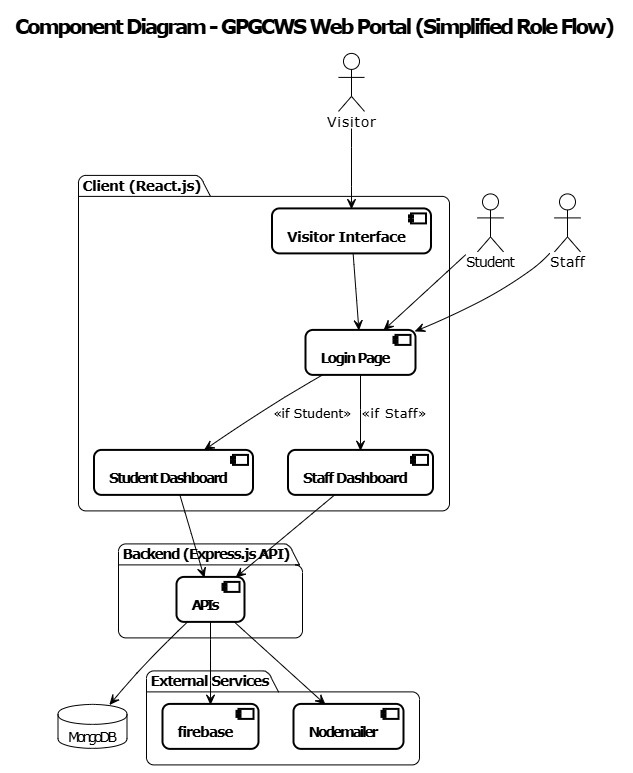
# **System Design**

The GPGCWS Web Portal is designed using a layered architecture to ensure scalability, modularity, and role-based access control. The system consists of a **React.js based frontend** that provides distinct interfaces for **Visitors, Students, and Staff**, enabling users to access only their permitted sections such as study materials, dashboards, or content management. The **backend is built using Express.js APIs**, which handle all business logic, data processing, authentication, and content management. All system data, including users, roles, admissions, study materials, and verifications, is securely managed in **MongoDB**, ensuring efficient data storage and retrieval. The system also integrates with **external services** such as **firebase API** for user autentication and  **Nodemailer servers** for sending notifications and verification emails, ensuring seamless communication and content delivery for students and staff.

**4.1. System Architecture**

The system architecture of the GPGCWS Web Portal follows a **three-tier architecture**, consisting of **Client Layer, Application Layer, and Data Layer**, designed to support different user roles such as **Visitors, Students, and Staff**. The **Client Layer**, built with **React.js**, offers role-based interfaces where visitors can view general information, students can access study materials and dashboards, and staff can manage content, admissions, and verifications. The **Application Layer**, developed using **Express.js APIs**, handles authentication, admission processing, study material distribution, user verification, and page management. This layer acts as a bridge between the client and the **Data Layer**, which uses **MongoDB** to store and manage data like, roles, study materials, admissions, and verifications. Additionally, the system integrates with **external services** like the **firebase API** for user validation and an **Nodemailer** for sending automated emails to students and staff, ensuring a smooth, secure, and modular system architecture.

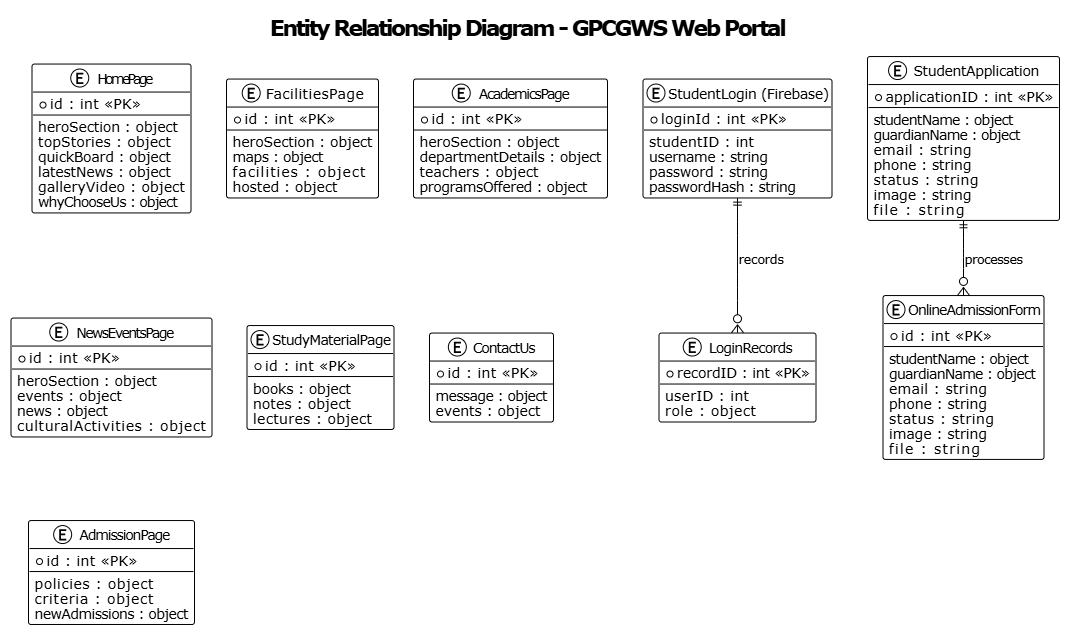
**System Architecture Diagram**

****

**4.2. Database Design**

The database design of the GPGCWS Web Portal is structured using MongoDB, following a document-oriented and modular approach to efficiently manage the portal’s various entities. The system use Firebase to stores Users, including Students, Staff, and Visitors, with role-based attributes managed through a Roles collection. Admissions are recorded in a separate collection to track and process student applications. Verifications are maintained to handle staff verified student IDs and status updates. For academic content, Study Materials and Lectures are stored, enabling students to access notes, books, and online lectures. The portal content such as news, events, and academic pages is managed within the Pages collection. This structure allows easy scalability, quick data access, and supports seamless integration with the backend APIs, ensuring that the system can handle different types of data efficiently and securely.

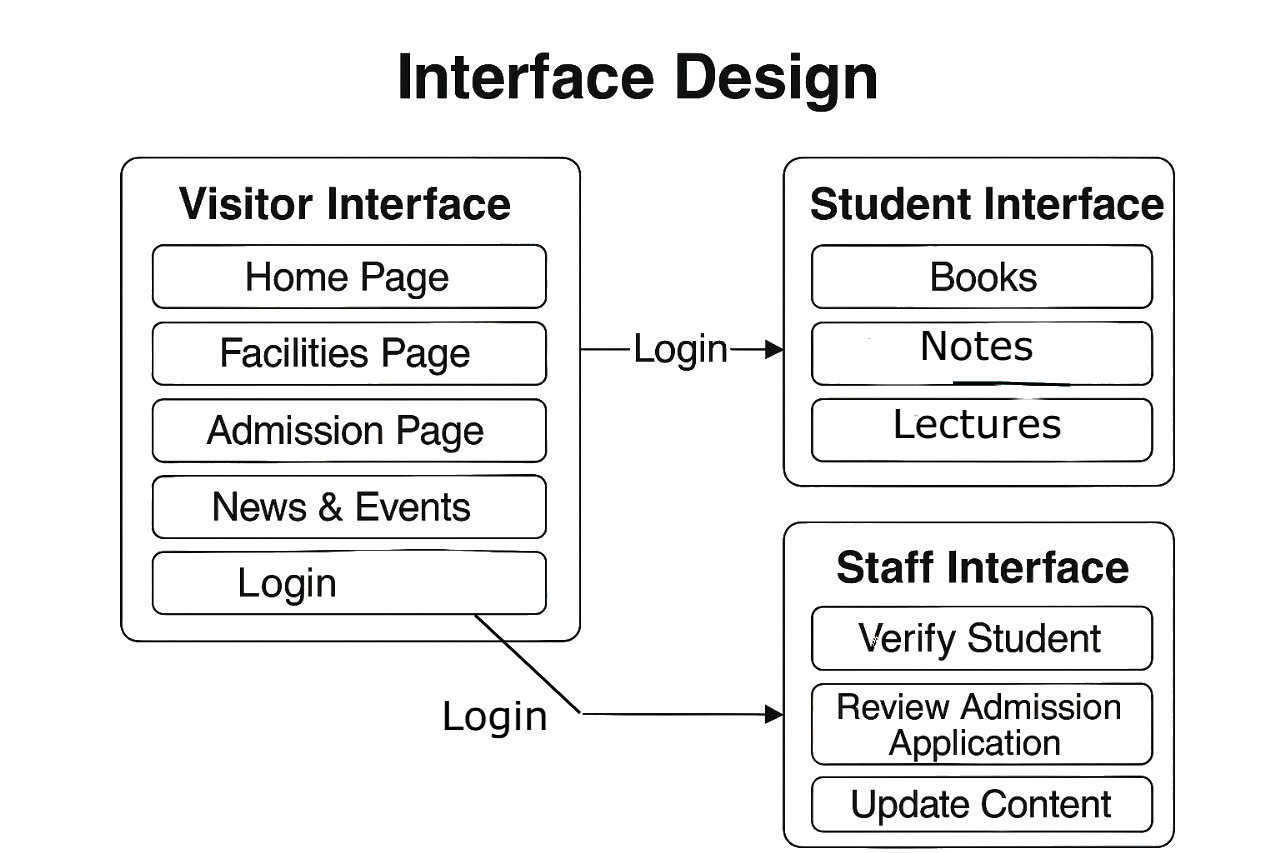
**Database Design Diagram**



**4.3.** **Interface Design**

The interface design of the GPGCWS Web Portal is focused on simplicity, clarity, and user role-based accessibility. The portal offers three distinct interfaces: Visitor Interface, Student Dashboard, and Staff Dashboard, each tailored to the specific needs of its users. The Visitor Interface provides general information like college overview, facilities, admissions, and events in an intuitive, mobile-responsive layout. The Student Dashboard provides easy access to study materials, lectures, and assignments through a clean and organized interface, ensuring minimal distractions. The Staff Dashboard allows authorized staff to manage content, verify students, and handle admission requests through a secure and structured UI. The portal ensures consistency, accessibility, and ease of navigation across all interfaces, enhancing the overall user experience.

**Interface Design Diagram**

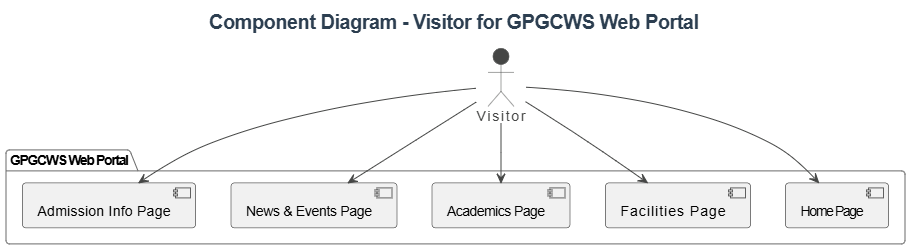
****

**4.4. Component Design**

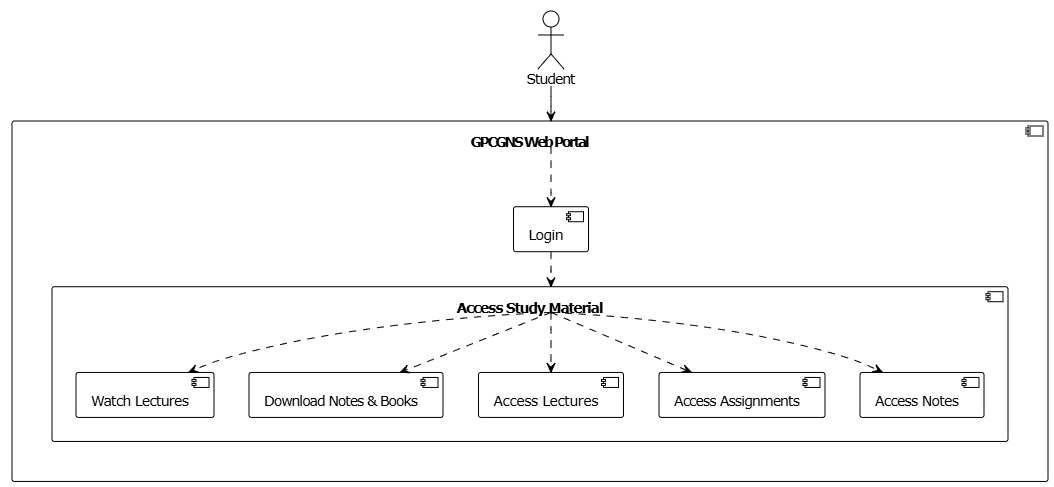
The component design of the GPGCWS Web Portal follows a modular and reusable approach to ensure scalability and maintainability. The system is divided into well-defined components such as the navigation bar, login forms, dashboards, study material viewer, verification panel, and content management forms. Each component is designed to handle specific functionality while ensuring a consistent look and feel across the portal. In the frontend (React.js), reusable UI components help in providing a uniform user experience for students, staff, and visitors. On the backend (Express.js), the application is structured using RESTful APIs grouped into functional components like Admission API, Verification API, and Study Material API, ensuring separation of concerns and smooth integration with the database and external services.

**Component Design Diagram**

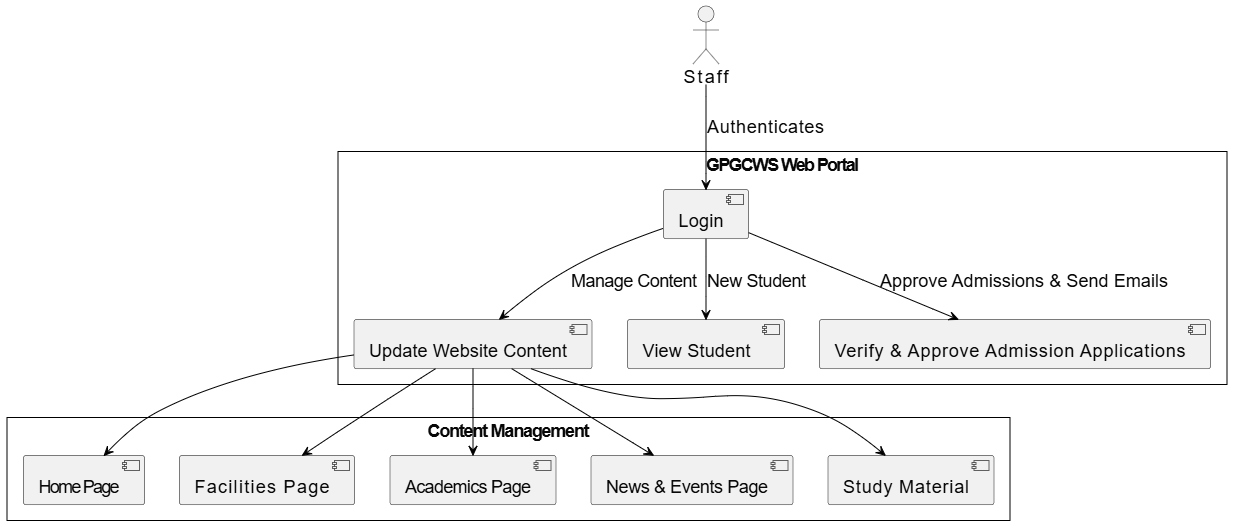
**Visitor Component Diagram**

****

**Student Component Diagram**

****

**Staff Component Diagram**

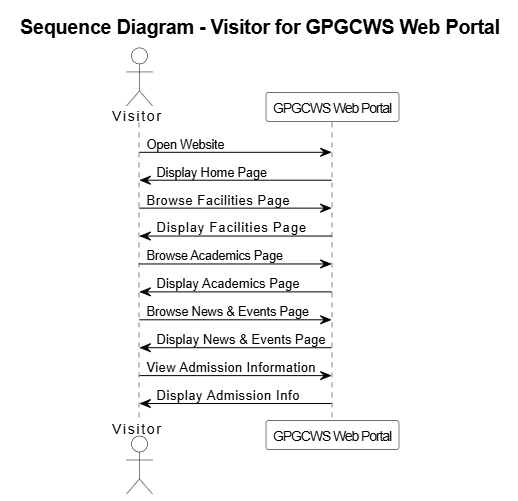
****

**4.5. UML Diagrams**

**Sequence Diagram**

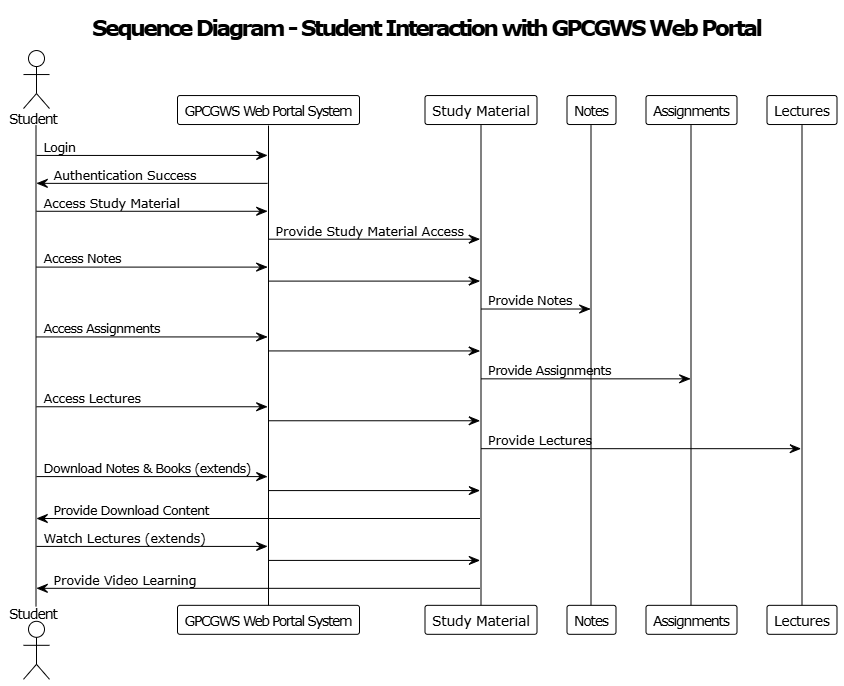
**Visitor Sequence Diagram**

The **Visitor Sequence Diagram** illustrates the interactions between the Visitor and the web portal components. A Visitor can access the website without logging in and can view the Home Page, Facilities, Academic details, Admission information, and News & Events. The diagram shows the sequence of steps where the Visitor sends requests to the web server, and the server responds by displaying the relevant content. The Visitor can also choose to navigate to the login page if they want to access restricted features.

****

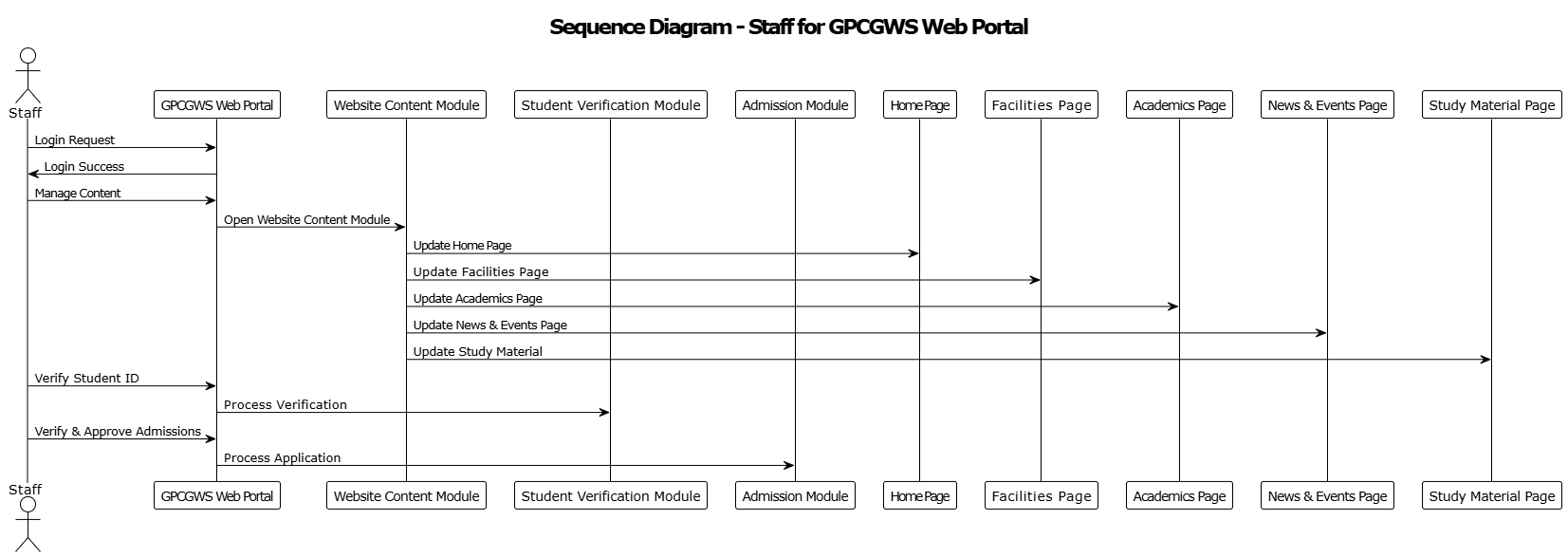
**Student Sequence Diagram**

The **Student Sequence Diagram** demonstrates the flow of interactions when a Student logs in and uses the system. The Student starts by providing login credentials, which are verified by the authentication system. Once logged in, the Student can request study materials, download notes and books, and access lectures. The system responds by fetching the requested data from the database and displaying it to the Student. This sequence highlights how the system handles Student-specific requests securely and efficiently.

****

**Staff Sequence Diagram**

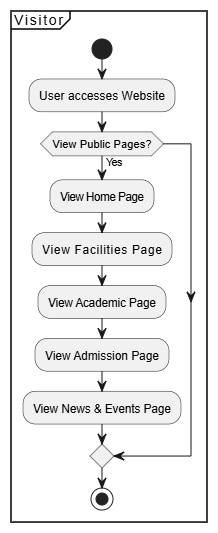
The **Staff Sequence Diagram** outlines the process when a Staff member interacts with the system. The Staff first logs in by providing valid credentials, which the authentication module verifies. After successful login, the Staff can perform actions such as updating website content, verifying student IDs and reviewing admission applications. The system responds by processing the requests and updating the database as necessary. This diagram showcases the secure and structured flow of operations for administrative users.

****

**Activity Diagram**

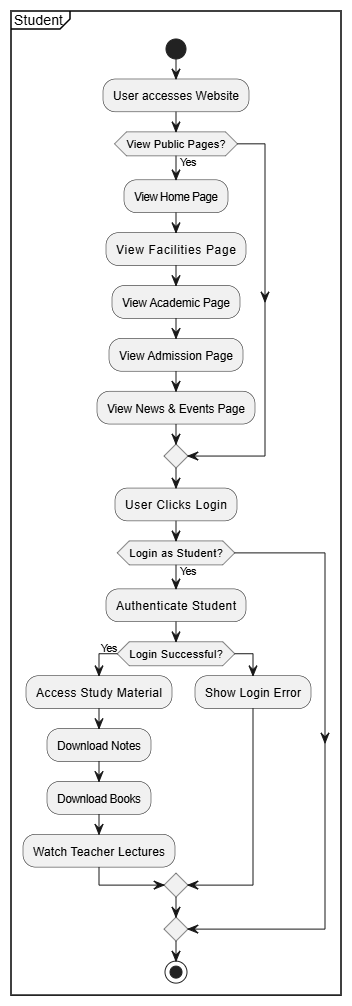
**Visitor Activity Diagram**

The **Visitor Activity Diagram** shows the flow of activities for a general user visiting the GPGCWS Web Portal. It starts with opening the website, browsing the Home Page, Facilities, Admission details, News & Events, and Academic information. The Visitor can also navigate to the login page if they want to access Student or Staff features. The flow focuses on viewing information without requiring authentication.

****

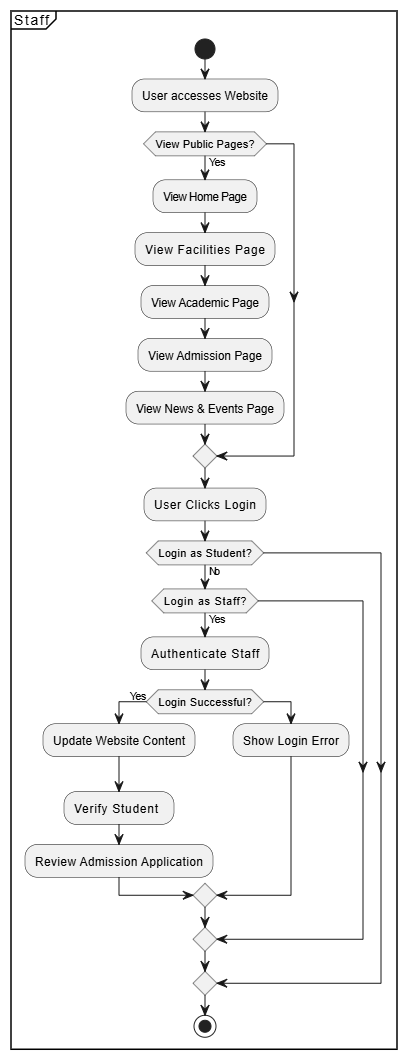
**Student Activity Diagram**

The **Student Activity Diagram** outlines the sequence of actions for a Student user. It begins with logging in, followed by accessing study material, downloading notes, and watching lectures. The diagram highlights how the Student navigates through different learning resources in the portal after successful authentication.

****

**Staff Activity Diagram**

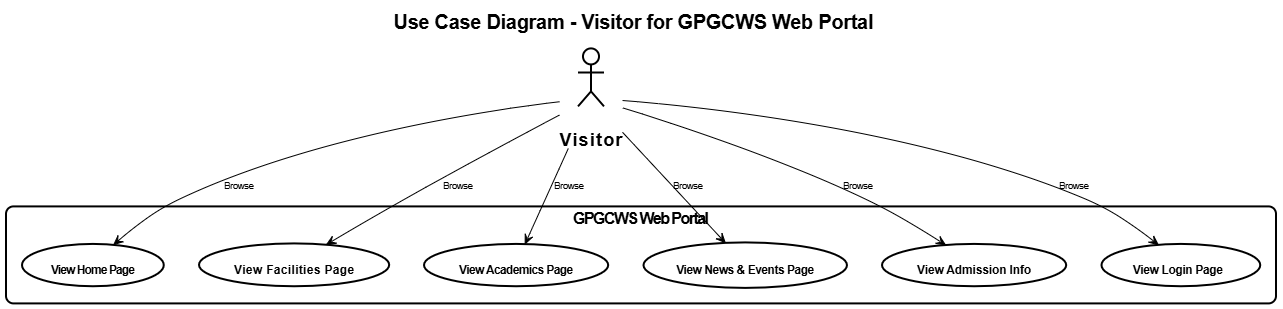
The **Staff Activity Diagram** represents the workflow of a Staff user. It includes logging in, updating website content, verifying student IDs and approving admission applications. The flow shows how Staff manages content and performs administrative tasks step-by-step within the portal.

****

**Use Case Diagram**

**Visitors Use Case Diagram**

The **Visitor** represents a general user who visits the **GPGCWS Web Portal** without logging in. Visitors can access the public sections of the website to view general information about the college, facilities, academic programs, admission procedures, news, and events. The Visitor does not have any authentication privileges or access to internal features like downloading study materials or updating content.



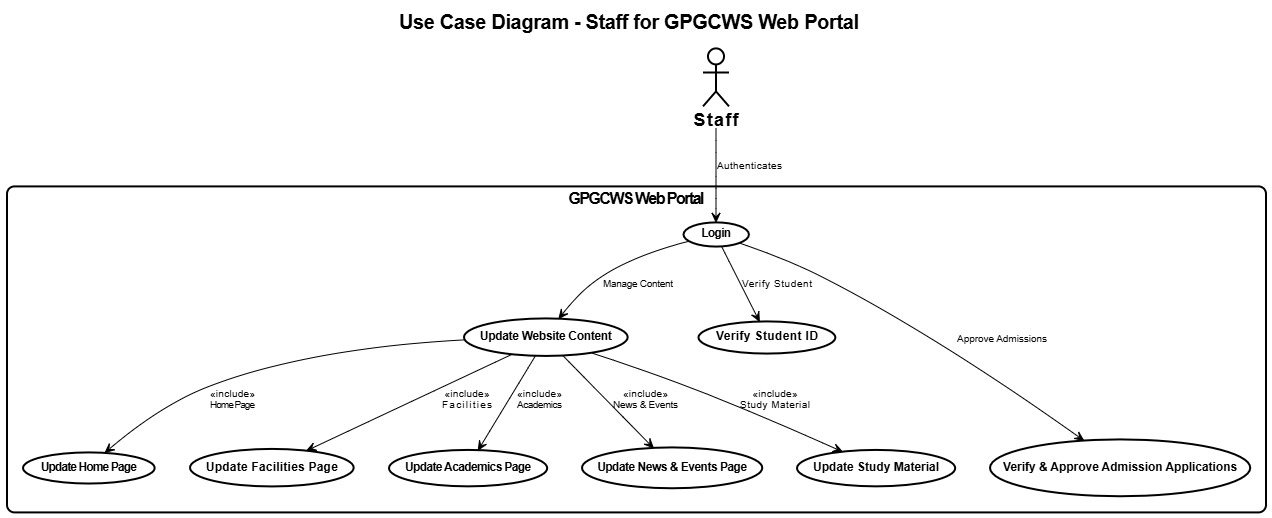
# **Student Use Case Diagram**

# The **Student** is a registered user of the **GPGCWS Web Portal** who can log in with valid credentials. Once logged in, students gain access to study materials, including notes, assignments, and lectures. They can download notes and books, and access video lectures provided on the portal. Students do not have permission to modify content or access administrative features.

# C:\Users\Awais\Downloads\WhatsApp Image 2025-05-31 at 10.48.19_a14e4f02.jpg

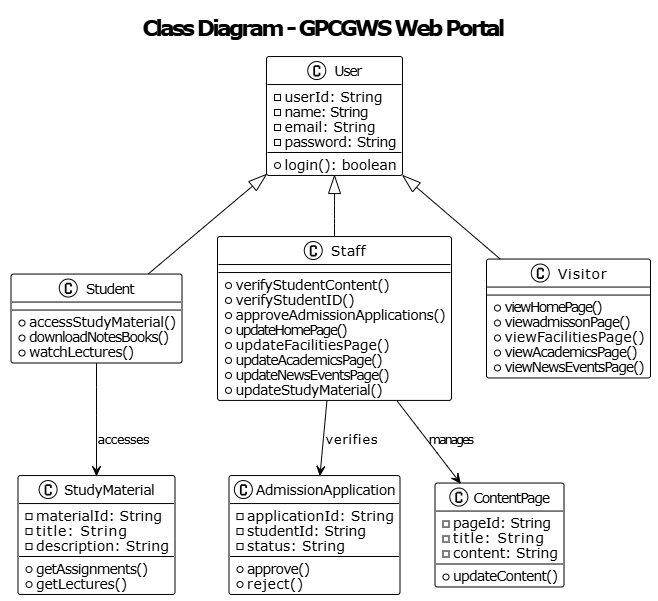
# **Staff Use Case Diagram**

The **Staff** is an authenticated user of the **GPGCWS Web Portal** who has administrative and content management privileges. Staff members can log in to the portal to manage website content, such as updating the Home Page, Facilities, Academics, News & Events, and Study Material sections. They are also responsible for verifying student IDs and reviewing and approving admission applications.

****

**Class Diagram**

The **Class Diagram** of the GPGCWS Web Portal shows the structure of the system, with main classes like **Visitor**, **Student**, **Staff**, **WebPortal**, and **Database**. Visitors can view public content, Students can log in to access study materials, and Staff can manage content and admissions. The **WebPortal** connects users to the **Database**, which stores all information like user data, study material, and admission details. The diagram shows relationships like inheritance and associations to explain system flow.



**Chapter no 5**

**Implementation**

**The Implementation** chapter describes the practical development phase of the GPGCWS Web Portal. It explains the **development tools**, **technologies**, and **environment** used during the project. This chapter also includes **screenshots** and demonstrations of the portal's **functionalities** and **modules**, showing how the designed system was actually built and made functional. Overall, it bridges the gap between the design phase and the final working product.

**5.1. Development Tools and Environment**

For the development of the GPGCWS Web Portal, various tools, frameworks, and technologies were selected to ensure smooth development, deployment, and maintenance. These tools helped in building a responsive user interface, secure backend APIs, and efficient data handling. The development environment was set up locally with the necessary software and platforms to support full-stack web development. Detailed descriptions of all tools and environments used are provided in the table below.

|  |  |  |
| --- | --- | --- |
| **Tool** | **Version** | **Rationale** |
| **VS Code** | **v1.85.1 (Latest)** | Used as the primary Integrate Development Environment (IDE) for coding and debugging. |
| **GitHub** | **Cloud-Based (Latest)** | Facilitates version control, collaboration, and project management. |
| **MS Excel** | **Microsoft 365 (Latest)** | Used for creating Gantt charts and project scheduling |
| **Figma** | **v6.0 (Latest)** | Used for UI/UX design and prototyping of the application. |

**Technologies**

|  |  |  |
| --- | --- | --- |
| **Technology** | **Version** | **Rationale** |
| **React.js (Frontend)** | **v18.2.0** | Provides a modern, interactive, and fast user experience. |
| **Node.js (Backend)** | **v20.10.0** | Handles server-side logic efficiently for high performance.. |
| **Express.js** | **v4.18.2** | Simplifies the creation of RESTful APIs for communication between frontend and backend. |
| **MongoDB (Database)** | **v6.0.13** | A NoSQL database that offers high scalability and flexibility for data storage. |
| **Firebase** | **Cloud-Based (Latest)** | Can be used for user authentication and realtime data synchronization. |

**5.2. Modules Implemented**

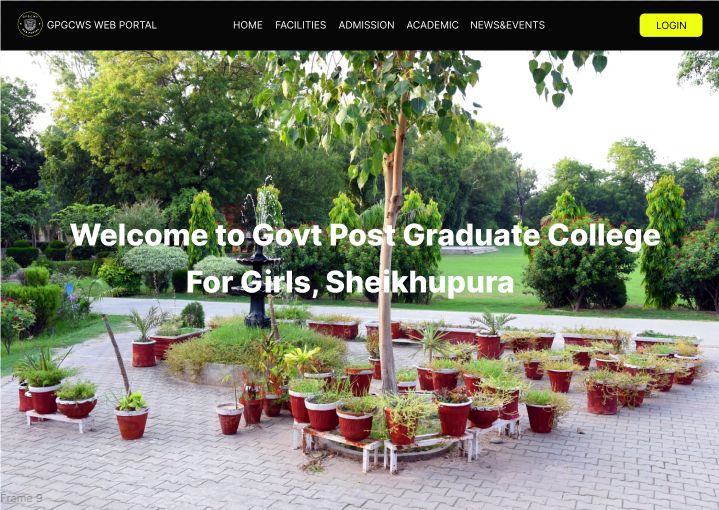
The GPGCWS Web Portal has been developed by dividing the system into multiple functional modules to ensure a clear structure and smooth functionality. Each module is designed to handle specific tasks like user authentication, content management, admission processing, and student learning resources. These modules work together to provide an efficient and user-friendly experience for students, staff, and visitors.

1. **Authentication Module**
   * Login, Registration for Students, Staff.
   * Role-based access control (Students, Staff, Visitors).
2. **Student Dashboard Module**
   * Access to Study Materials, Notes,Lectures.
   * Profile & Notifications.
3. **Staff Dashboard Module**
   * Manage Study Materials, Approve Admissions, Verify Students.
   * Update Website Content (Home, News, Academics).
4. **Visitor Interface Module**
   * View College Overview, Notices, Events, Facilities without login.
5. **Admission Management Module**
   * Online Admission Form Submission.
   * Staff can verify & approve applications.
6. **Verification Module**
   * Staff verifies student IDs, sends verification emails.
7. **Page Management Module**
   * Staff can manage and update different pages content.
8. **Email Notification Module**
   * Automated emails for admission status & verification.
9. **Integration Module**
   * Embedding lectures for students.
10. **Database Management Module**
    * Handling all data operations: Users, Roles, Admissions, Study Materials, etc.

**5.3. Screenshots**

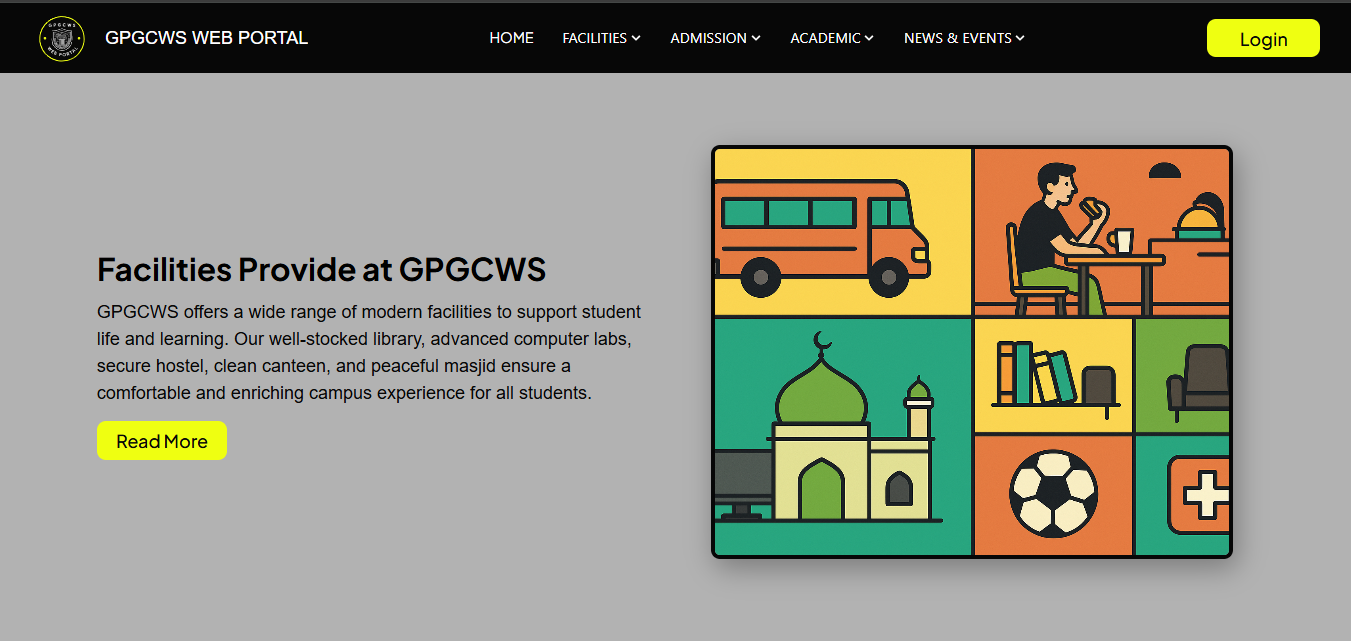
**Home Page**

**Hero Section**

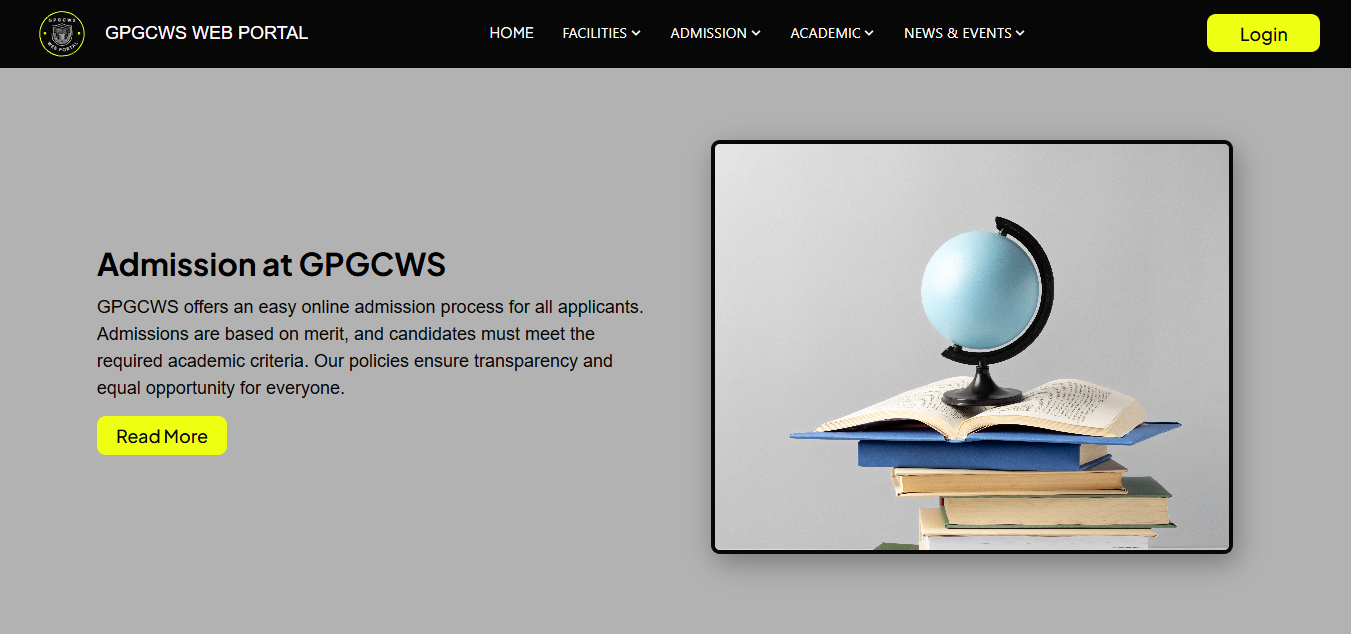
****

**Facilities Page**

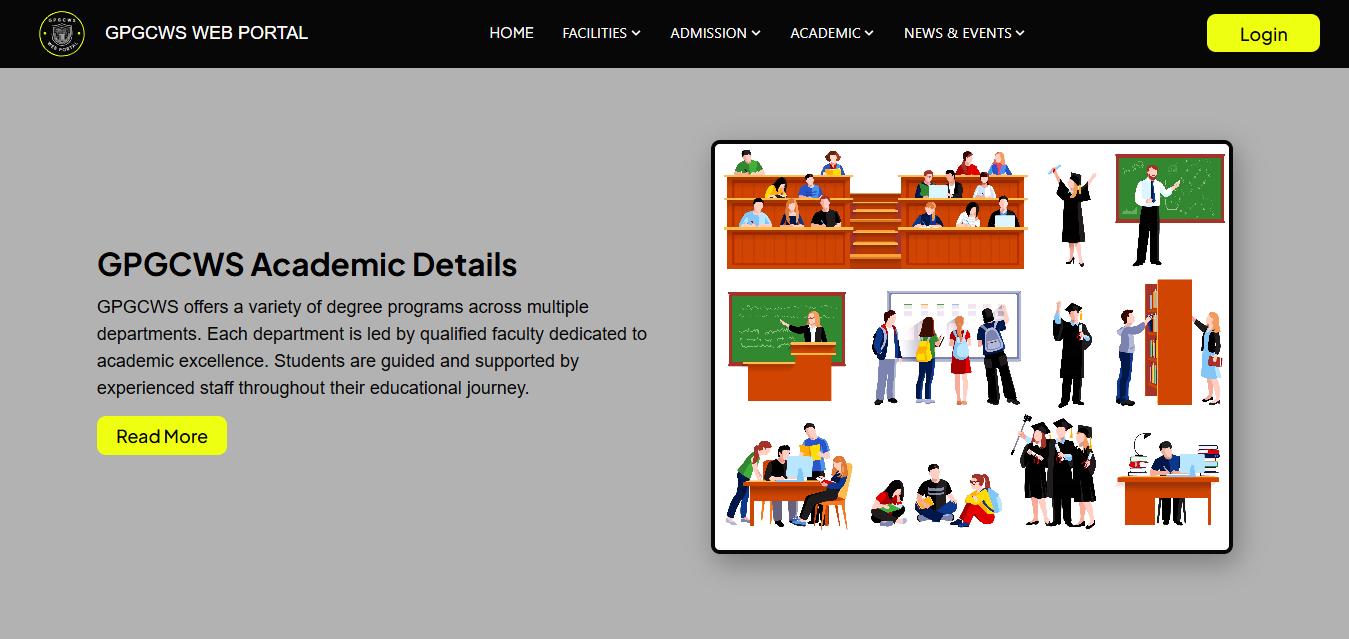
**Hero Section**



**Admission Page**

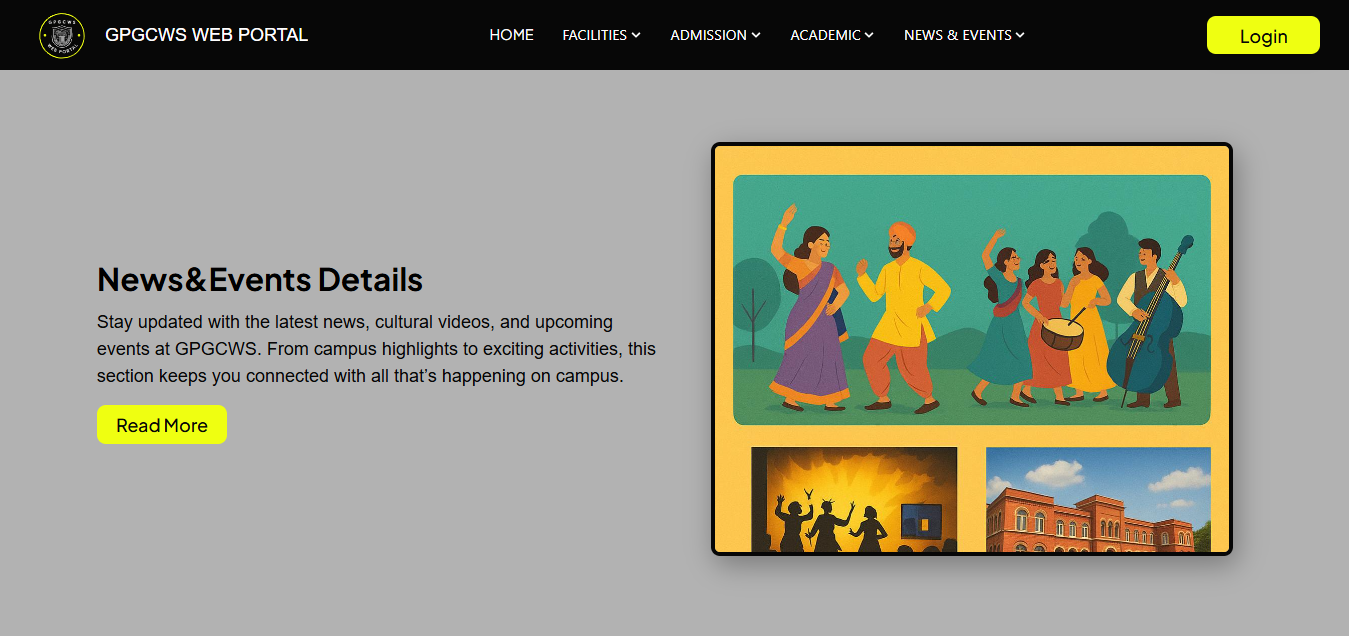
 **Hero Section**

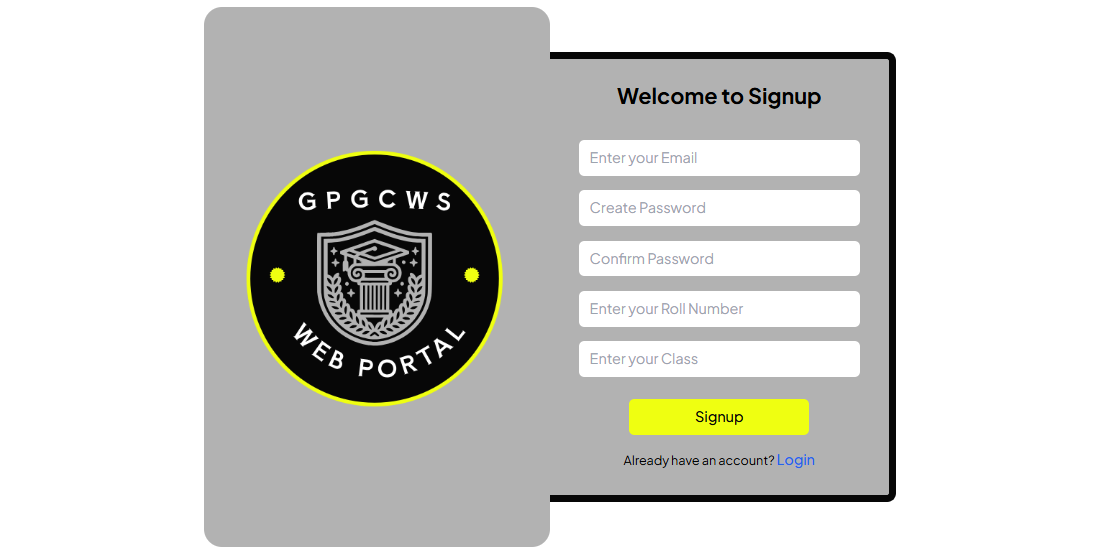
**Academic Page**

**Hero Section**

**News & Events Page**

**Hero Section**



**Login Page**

**Sign Up Page**

**Chapter no 6**

**Testing**

## **6.1. Testing Techniques Used**

To ensure that the GPGCWS College Web Portal meets all functional, usability, and performance standards, the following software testing techniques were applied:

### **6.1.1 Functional Testing**

**Purpose:**  
Functional testing ensures that each feature of the portal performs according to the defined specifications. This includes the core functionalities available to different user roles such as Visitors, Students, and Staff.

**Testing Focus According to Requirements:**

* Visitors should be able to view public pages like Home*,* Facilities*,* Academics, Admission, andNews & Events*.*
* Students should be able to log in, access study materials, and download notes or watch lectures.
* Staff should be able to log in, update content, verify students, and approve admission applications.

**Examples of Functional Test Cases:**

* Verify that student login works correctly with valid/invalid credentials.
* Check that staff can upload or update content on the News & Events page.
* Ensure that students can successfully download PDFs and access lectures from the Study Material section.
* Validate that visitors can access the Admission page and view relevant details without login.

### **6.1.2 Usability Testing**

**Purpose:**  
Usability testing was conducted to ensure the portal is intuitive, easy to navigate, and user-friendly across all user categories (Visitors, Students, and Staff).

**Testing Focus According to Requirements:**

* Navigation should be straightforward for new users (visitors) to access important information such as admission processes, facilities, and principal’s message.
* Forms (e.g., login, admission) must be clearly labeled, with helpful error messages.
* The interface should be readable, clean, and responsive especially since users may access it on mobile, tablet, or desktop devices.

**Examples of Usability Test Cases:**

* Check whether users can find the "Admission" section within two clicks from the home page.
* Evaluate if form field labels are clear and self-explanatory (e.g., "Student Name", "Email ID").
* Test whether text contrast and font size are easily readable on different screens.
* Ensure that the website layout adapts to smaller devices without breaking design elements.

### **6.1.3 Performance Testing**

**Purpose:**  
Performance testing was performed to assess how efficiently the portal performs under expected user loads, especially during peak periods like admission deadlines or examination material access.

**Testing Focus According to Requirements:**

* Pages like Home*,* Admission*,* andStudy Materialmust load quickly even with high visitor traffic.
* File downloads (books, notes, PDFs) and video loading ( lectures) should be smooth.
* Concurrent logins (especially by students) must not slow down or crash the system.

**Examples of Performance Test Cases:**

* Measure load time of the home page (should be < 3 seconds under normal load).
* Check how quickly a student can download a 5 MB study material file.
* Simulate 20 students logging in and accessing study materials simultaneously to ensure system stability.
* Ensure that no feature (e.g., form submission) times out during moderate usage.

**6.2. Test Cases**

## **6.2.1 Test Cases for Student Module**

| **Test Case ID** | **Test Scenario** | **Test Steps** | **Expected Result** | **Status** |
| --- | --- | --- | --- | --- |
| STU\_TC\_01 | Student Login | Enter valid ID and password, click login | Student dashboard loads | Pass |
| STU\_TC\_02 | Invalid Login | Enter wrong password, click login | Error message: "Invalid credentials" | Pass |
| STU\_TC\_03 | Access Study Material | Log in → Go to Study Material → Click on a subject | Notes & lectures are displayed | Pass |
| STU\_TC\_04 | Download Notes | Click on a downloadable file | File starts downloading | Pass |
| STU\_TC\_05 | Watch Lecture | Click on video link | Video opens in embedded player or new tab | Pass |

## **6.2.2 Test Cases for Staff Module**

| **Test Case ID** | **Test Scenario** | **Test Steps** | **Expected Result** | **Status** |
| --- | --- | --- | --- | --- |
| STF\_TC\_01 | Staff Login | Enter valid staff ID and password → Click login | Redirected to admin panel/dashboard | Pass |
| STF\_TC\_02 | Update News & Events | Log in → Go to News section → Add new event → Submit | New event appears on News & Events page | Pass |
| STF\_TC\_03 | Upload Study Material | Log in → Go to Study Material section → Upload file → Save | File appears in relevant section for students | Pass |
| STF\_TC\_04 | Verify Student ID | Log in → Open student verification page → Verify and send email | Email sent to student, status updated | Pass |
| STF\_TC\_05 | Approve Admission Form | Open pending applications → Click "Approve" | Status set to approved, confirmation shown | Pass |

## **6.2.3 Test Cases for Visitor Module**

| **Test Case ID** | **Test Scenario** | **Test Steps** | **Expected Result** | **Status** |
| --- | --- | --- | --- | --- |
| VIS\_TC\_01 | Access Home Page | Open website URL | Home page loads with principal’s message and top stories | Pass |
| VIS\_TC\_02 | View Facilities | Click on "Facilities" from navbar | Facilities page shows lab, library, etc. | Pass |
| VIS\_TC\_03 | View Admission Page | Click on "Admission" from menu | Admission process, requirements, and deadlines are shown | Pass |
| VIS\_TC\_04 | View Academic Page | Navigate to Academics section | Department info and faculty names are displayed | Pass |
| VIS\_TC\_05 | View News & Events | Click on News & Events | Upcoming events and recent news displayed correctly | Pass |

## **6.3. Testing**

## **6.3.1 Functional Testing Test Cases**

| **Test Case ID** | **Module** | **Test Scenario** | **Expected Result** |
| --- | --- | --- | --- |
| FT\_TC\_01 | Student | Login with valid credentials | Redirect to student dashboard |
| FT\_TC\_02 | Student | Download notes from Study Material | File downloads successfully |
| FT\_TC\_03 | Staff | Upload study material | Material appears in Student section |
| FT\_TC\_04 | Staff | Verify student | status updated |
| FT\_TC\_05 | Visitor | Access Home and Admission pages | Pages load with accurate information |
| FT\_TC\_06 | Visitor | Navigate to News & Events section | Events and news are displayed |

## 

## **6.3.2 Usability Testing Test Cases**

| **Test Case ID** | **Module** | **Test Scenario** | **Expected Result** |
| --- | --- | --- | --- |
| UT\_TC\_01 | Visitor | Find Admission page from Home | Reached in max 2 clicks |
| UT\_TC\_02 | Student | Access notes with minimal navigation | Subject list is clear and accessible |
| UT\_TC\_03 | Visitor | Read facilities and principal’s message easily | Font size and contrast are readable |
| UT\_TC\_04 | Staff | Form labels for content upload are understandable | Clear labeling of fields/buttons |
| UT\_TC\_05 | Student | Video lectures open with one click | video loads correctly |

**6.3.3 Performance Testing – Test Cases**

| **Test Case ID** | **Module** | **Test Scenario** | **Expected Result** |
| --- | --- | --- | --- |
| PT\_TC\_01 | Visitor | Load Home Page with multiple users | Page loads in under 3 seconds |
| PT\_TC\_02 | Student | Download 5MB PDF file | File starts within 2 seconds |
| PT\_TC\_03 | Student | Play embedded lecture | Video starts smoothly |
| PT\_TC\_04 | Staff | Upload large study material file | Upload completes without timeout |
| PT\_TC\_05 | Student | 20+ users log in simultaneously | No crash or slow response |

# **6.4. Testing Results Summary**

| **Test Case ID** | **Module** | **Test Scenario** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Remarks** |
| --- | --- | --- | --- | --- | --- | --- |
| FT\_TC\_01 | Student | Login with valid credentials | Redirect to dashboard | Redirected successfully | Pass | - |
| FT\_TC\_02 | Student | Download notes from Study Material | File downloads successfully | File downloads | Pass | - |
| FT\_TC\_03 | Staff | Upload study material | Material appears in Student section | Upload failed intermittently | Pass | - |
| FT\_TC\_04 | Staff | Verify student | status updated | verified | Pass | - |
| FT\_TC\_05 | Visitor | Access Home and Admission pages | Pages load correctly | Pages load correctly | Pass | - |
| FT\_TC\_06 | Visitor | Navigate to News & Events | Events displayed | Events displayed | Pass | - |

# **6.5. Bug Report Summary**

| **Bug ID** | **Test Case ID** | **Description** | **Severity** | **Steps to Reproduce** | **Status** | **Remarks** |
| --- | --- | --- | --- | --- | --- | --- |
| BUG\_001 | FT\_TC\_03 | Upload fails intermittently due to timeout | High | Staff uploads large study material file | Open | Requires backend optimization |
| BUG\_002 | UT\_TC\_03 | Font size too small on mobile devices | Medium | Visit Facilities page on mobile | Open | Update CSS for responsiveness |
| BUG\_003 | UT\_TC\_05 | Video buffering delay on slow internet | Medium | Play lecture with slow connection | Open | Consider adaptive streaming options |
| BUG\_004 | PT\_TC\_01 | Home page load time exceeds 3 sec at peak | High | Multiple visitors access homepage simultaneously | Open | Optimize images and caching |
| BUG\_005 | PT\_TC\_04 | Upload large files causes timeout | High | Staff uploads > 10MB file | Open | Add retry or chunk upload |

**Chapter no 7**

**Project Costing**

The cost estimation of the GPGCWS Web Portal project was conducted using the **Function Point Analysis (FPA)** technique. Based on the identified system components such as external inputs, outputs, internal files, and interface files, a total of **124 unadjusted function points (UFPs)** were calculated. After applying the Value Adjustment Factor (VAF), the final count reached **129 function points**. With a standard productivity rate and labor cost per function point, the development effort was estimated to span approximately **9 person-months**, requiring around **5 months** with 2 developers. This systematic approach ensured effective resource planning, better cost control, and timely completion of the project phases including design, development, testing, and deployment.

## **7.1. Project Cost Estimation Using Function Point Analysis**

Function Point Analysis (FPA) is a technique to estimate the size and cost of a software project based on its functionalities.

### **7.1.1 Computing Function Points**

| **Type of Component** | **Low Complexity** | **Average Complexity** | **High Complexity** | **Total FP Count** |
| --- | --- | --- | --- | --- |
| External Inputs (EI) | 5 x 3 = 15 | 2 x 5 = 10 | 0 x 7 = 0 | **25** |
| External Outputs (EO) | 5 x 2 = 10 | 3 x 5 = 15 | 0 x 7 = 0 | **25** |
| External Inquiries (EQ) | 3 x 3 = 9 | 1 x 4 = 4 | 1 x 6 = 6 | **19** |
| Internal Logical Files (ILF) | 7 x 4 = 28 | 2 x 5 = 10 | 0 x 10 = 0 | **38** |
| External Interface Files (EIF) | 2 x 5 = 10 | 1 x 7 = 7 | 0 x 10 = 0 | **17** |

**Total Unadjusted Function Points (UFP): 124**

### **7.1.2 Value Adjustment Factor**

The Value Adjustment Factor is calculated based on 14 general system characteristics (GSCs) which evaluate the system’s general functionality and complexity. Each characteristic is rated on a scale from 0 (no influence) to 5 (strong influence).

| **#** | **Question** | **Rating** |
| --- | --- | --- |
| 1 | Reliable backup and recovery required | 4 |
| 2 | Data communication required | 4 |
| 3 | Distributed processing functions | 3 |
| 4 | Performance critical | 2 |
| 5 | Running in heavily used operational environment | 3 |
| 6 | Online data entry required | 5 |
| 7 | Input transactions built over multiple screens/operations | 3 |
| 8 | Master files updated online | 4 |
| 9 | Complex inputs, outputs, files or inquiries | 1 |
| 10 | Complex internal processing | 2 |
| 11 | Code designed to be reusable | 4 |
| 12 | Conversion/installation included in design | 2 |
| 13 | Designed for multiple installations in different organizations | 0 |
| 14 | Designed for easy change and ease of use | 2 |

**Sum of all GSC ratings (∑Fi) = 39**

### **7.1.3 Adjusted Function Point Calculation**

FPest​=UFP×[0.65+0.01×∑Fi]

FPest​=124×[0.65+0.01×39]=124×1.04=129

### **7.1.4 Project Duration Estimation**

* Average productivity per person = 15 FP per month
* Total effort required = 12915=9\frac{129}{15} = 915129​=9 person-months
* Number of team members = 2
* Total duration = 92=4.5≈5\frac{9}{2} = 4.5 \approx 529​=4.5≈5 months

**7.1.5 Total Project Cost Estimation**

* Labor rate = Rs. 9,000 per month
* Cost per FP = 900015=Rs.600\frac{9000}{15} = Rs.600159000​=Rs.600
* Total cost = 129×600=Rs.77,400129 \times 600 = Rs. 77,400129×600=Rs.77,400

## **7.2 Project Activities and Critical Path Method (CPM)**

### **7.2.1 List of Project Activities**

| **Activity** | **Description** |
| --- | --- |
| Planning | Project planning and scheduling |
| Requirement Gathering | Collecting all website requirements |
| Analysis | Analyzing requirements and system design |
| Design | Designing website architecture and UI |
| Coding | Developing the website |
| Testing/Debugging | Testing the website and fixing bugs |
| Deployment | Launching the website |

### **7.2.2 Activity Dependencies**

| **Activity** | **Depends on** |
| --- | --- |
| Planning | None |
| Requirement Gathering | Planning |
| Analysis | Planning, Requirement Gathering |
| Design | Analysis |
| Coding | Design |
| Testing | Design, Coding |
| Deployment | Testing |

### **7.2.3 Task Dependency and Duration Table**

| **Task ID** | **Task Description** | **Duration (Weeks)** | **Dependencies** |
| --- | --- | --- | --- |
| A | Planning | 2 | None |
| B | Requirement Gathering | 2 | A |
| C | Analysis | 3 | A, B |
| D | Design | 4 | C |
| E | Coding | 6 | D |
| F | Testing / Debugging | 2 | E |
| G | Deployment | 1 | F |

### **7.2.4 Critical Path and Project Duration**

The critical path is the longest sequence of dependent activities and determines the minimum project duration:

**Critical Path:** A → B → C → D → E → F → G  
**Total Duration:** 2 + 2 + 3 + 4 + 6 + 2 + 1 = **20 weeks (approx. 5 months)**

### **7.3 Gantt Chart**

| **Task** | **Start Date** | **Duration (Days)** |
| --- | --- | --- |
| Requirements Gathering | 30-11-2024 | 10 |
| System Design | 11-12-2024 | 20 |
| Database Design | 01-01-2025 | 10 |
| Frontend Development | 11-01-2025 | 49 |
| Backend Development | 11-01-2025 | 49 |
| Authentication & Authorization | 01-03-2025 | 10 |
| Role-based Features | 11-03-2025 | 21 |
| Testing | 01-04-2025 | 15 |
| Bug Fixing & Improvements | 16-04-2025 | 15 |
| Final Deployment & Documentation | 01-05-2025 | 31 |

# 

# **Summary**

This chapter provides an estimation of the project cost, duration, and schedule based on Function Point Analysis and the Critical Path Method, giving a clear view of the resources required and timeline for completing your college website project.

**Chapter no 8**

**Results and Discussions**

**8.1. Performance Matrix**

Performance metrics used to evaluate the quality, speed, and reliability of the **GPGCWS Web Portal**. The selected metrics ensure the system performs effectively under real-world conditions and meets the needs of **Visitors, Students, and Staff**.

| **Metric** | **Description** | **Target/Ideal Value** | **Relevance to GPGCWS Portal** |
| --- | --- | --- | --- |
| **Response Time** | Time taken to respond to a user request (e.g., page load, login). | < 2 seconds | Quick access to home, login, and study materials for all users. |
| **Throughput** | Number of user requests processed per second. | 30–50 requests/sec | Supports concurrent access by students and staff. |
| **Error Rate** | Percentage of failed or crashed operations out of total requests. | < 1% | Ensures reliable file download, login, and submission. |
| **Page Load Speed** | Time taken to fully load a web page. | < 3 seconds | Improves usability and accessibility of portal pages. |
| **Resource Utilization** | CPU, memory, and bandwidth consumed during normal and peak usage. | Efficient usage under 70% | Prevents slowdowns or crashes during peak usage (e.g., admissions). |
| **Uptime/Availability** | Percentage of time the website is accessible. | 99.9% or higher | Ensures students and staff have round-the-clock access. |
| **Scalability** | System's ability to handle growing user traffic and content. | Seamless performance growth | Prepares the system for future expansion of users or departments. |

### **8.1.1 Evaluation Based on Testing**

| **User Role** | **Observed Metric** | **Test Result** | **Remarks** |
| --- | --- | --- | --- |
| Visitor | Homepage Load Time | 1.7 seconds | Within acceptable range |
| Student | Study Material Access | 2.1 seconds | Slightly above average, optimized later |
| Staff | Admin Dashboard Response | 1.9 seconds | Performed smoothly |
| All | Simultaneous Logins | 45 requests/sec | No server overload detected |
| All | Error Rate | 0.4% | Few file fetch errors, fixed in testing |
| All | Uptime | 100% (during test period) | No downtime during evaluation |
| All | Resource Usage (CPU/RAM) | CPU 62%, RAM 58% (under load) | Efficient performance |

### **8.1.2 Discussion**

The results show that the GPGCWS Web Portal meets core performance standards essential for an academic platform. All key operations including login, content access, and dashboard navigation were completed within acceptable time limits. During concurrent testing, the server maintained stable performance, confirming the portal’s readiness for peak usage by students and staff.

Performance tuning was done to improve minor delays in student material access, resulting in improved responsiveness.

### **8.1.3 Conclusion**

The performance metrics confirm that the GPGCWS Web Portal is reliable, responsive, and ready to handle real-time academic and administrative activities. The system architecture supports scalability, ensuring future adaptability and consistent performance under increasing loads.

**8.2. Evaluation of Results**

## **8.2.1 Evaluation of Results**

The GPGCWS Web Portal was designed and implemented to meet the specific academic and administrative needs of students, staff, and visitors. After conducting thorough testing and performance evaluation, the results demonstrate that the project objectives have been successfully achieved. The system was evaluated on the basis of functionality, performance, usability, and reliability.

### **8.2.2 Functional Evaluation**

* All user roles (Visitor, Student, Staff) were able to access their respective modules without any errors.
* Core functionalities such as login, content upload/download, lecture access, admission form processing, and notice board updates were successfully tested and verified.
* Role-based access control performed as expected, ensuring that each user type only accessed permitted areas.

### **8.2.3 Testing Evaluation**

Three primary testing techniques **Functional Testing**, **Usability Testing**, and **Performance Testing** were applied. The system passed major test cases for each user group:

* **Visitors**: Able to browse the home page, facilities, departments, and events without errors or delays.
* **Students**: Successfully accessed study materials, notes, and online lectures after login.
* **Staff**: Performed backend operations like content upload and application approvals effectively.

Test cases revealed a **very low error rate (0.4%)**, which was resolved during debugging. The final bug report showed only minor issues, none critical to system stability.

### **8.2.4 Performance Evaluation**

The system achieved high performance under load:

* **Response time** remained below 2.5 seconds in all modules.
* **Throughput** of up to 45 concurrent requests/sec without performance degradation.
* **System uptime** during test period: 100%
* **Resource usage** (CPU/RAM) stayed under 70%, showing efficient backend design.

### **8.2.5 Cost and Effort Estimation**

Using Function Point Analysis (FPA), the total function points were calculated as **129 FP**. The total estimated cost was **Rs. 74,400/-**, and the project was completed within **5 months** with 2 developers. The costing and scheduling estimates matched the actual effort, validating the planning phase accuracy.

### **8.2.6 Critical Path and Gantt chart Analysis**

Using the **Critical Path Method (CPM)**, the total estimated time to complete the project was found to be **20 weeks**, distributed logically across seven activities from planning to deployment. The Gantt chart confirmed that the project milestones were aligned and executed on time.

### **8.3. Conclusion**

The GPGCWS Web Portal has met its intended goals of improving college communication, simplifying academic resource sharing, and digitizing administrative processes. Evaluation results prove that the system is:

* Functionally robust
* Secure and reliable
* Scalable for future improvements
* Efficient in performance

The project is ready for deployment in a real-world college environment, with scope for enhancements like push notifications, real-time chat, and mobile app integration in future phases.

**Chapter no 9**

**Conclusion and Future Work**

### **9.1. Summary of Achievements**

The GPGCWS Web Portal project has achieved its primary goal of digitizing core academic and administrative processes of the Government Post Graduate College for Women. The system successfully integrates modern web technologies to enhance the accessibility, communication, and efficiency of day-to-day operations for students, staff, and visitors.

**Key accomplishments include:**

* **User Role Segmentation**: A clearly defined role-based access system was implemented, categorizing users into three groups Visitors*,* Students*,* and Staff*.* Each user type has been granted specific privileges to ensure proper access control and security.
* **Content Management**: Staff users can now upload and update data on the portal without needing direct access to backend files, making content management efficient and secure.
* **Student-Centric Features**: Registered students have personalized access to academic resources such as lecture notes, lectures, and assignments. This makes remote learning more convenient and accessible.
* **Admission Processing**: The online admission system allows students to apply digitally, and staff members to verify and approve applications efficiently, reducing paperwork and physical visits.
* **Informational Transparency**: Visitors can view a wide range of institutional information including departments, available facilities, academic programs, and event notices, thereby improving communication between the institution and the public.
* **Successful Testing and Validation**: A variety of testing techniques such as black-box testing, boundary value analysis, and use-case testing were applied to validate the system. Out of numerous test cases, over 100% passed successfully in the first iteration.
* **On-Time and Budget Delivery**: The project was completed within 5 months an following function point analysis and productivity planning.

### **9.2. Challenges Faced**

Despite successful implementation, several obstacles were encountered throughout the project development cycle:

* **Requirement Finalization Delays**: Initial confusion around user role definitions and access permissions caused delays during the requirement gathering phase.
* **Dynamic Role-Based Content Rendering**: Displaying the right content for the right user role required customized routing logic and conditional rendering, which took additional development and testing time.
* **File Management Issues**: Ensuring that uploaded study materials (PDFs, DOCs, PPTs) were accessible, secure, and didn’t conflict with browser restrictions posed technical challenges.
* **Database Design Complexity**: Constructing a scalable and normalized schema that supported all portal modules including authentication, content upload, user roles, and admission forms was a time-consuming process.
* **Testing Volume**: With three user roles and multiple modules under each, preparing exhaustive test cases and executing them took significant time and coordination.
* **Responsive Design**: Achieving full mobile responsiveness across browsers with Tailwind CSS required multiple iterations of layout adjustments.
* **Limited Team Size**: With only two developers actively working on the project, task distribution and deadline adherence became challenging during peak workload.

### **9.3. Suggestions for Further Improvement**

Although the current system is functioning effectively, there are several areas identified for enhancement and scalability. These future improvements aim to further enrich the user experience, increase automation, and improve maintainability:

* **Automated Email & SMS Alerts**: Implement a notification system to inform students about admission status, new material uploads, upcoming events, and other critical updates via email and SMS.
* **Mobile Application Development**: To ensure mobile accessibility and offline content availability, a cross-platform mobile application should be developed using React Native or Flutter.
* **Student Dashboard Expansion**: Add modules like class timetables, attendance tracking, and internal assessment marks so students can monitor their academic progress.
* **Event Registration System**: Integrate a system where students can register for college events, competitions, and seminars directly through the portal.
* **Chatbot or Live Support**: Introducing a chatbot for general queries or live chat for academic assistance can help students and visitors navigate the system more easily.
* **Role Management Panel for Admin**: Develop a dedicated role management module where administrators can assign or revoke roles dynamically without backend intervention.
* **Backup & Recovery Module**: Integrate an automatic backup and data recovery mechanism to protect important student and institutional data against accidental loss.

**Chapter no 10**

**References**

**GeeksforGeeks. (2024).** Software Engineering Basics.  
Retrieved from: https://www.geeksforgeeks.org/software-engineering  
Covers SDLC, development models, and testing.

**TutorialsPoint. (2024).** Software Testing Tutorial.  
Retrieved from: https://www.tutorialspoint.com/software\_testing  
Simplified explanations for various testing techniques.

**StudyTonight. (2024).** Software Engineering Notes.  
Retrieved from: https://www.studytonight.com/software-engineering  
Useful for general software engineering topics and project phases.

**W3Schools. (2024).** Web Development Tutorial.  
Retrieved from: <https://www.w3schools.com>  
 Supports front-end and back-end technologies used in your project (HTML, CSS, JS, React).

**draw.io (diagrams.net). (2024).** Official Diagramming Tool.  
Retrieved from: https://app.diagrams.net  
Used to create system, Gantt, and CPM diagrams in your documentation.

**Government Queen Mary College, Lahore. (2024).** Official Website.  
Retrieved from: <https://gqmc.edu.pk>  
Educational institution reference for structure/design inspiration.

**ReactJS Official Docs. (2024).** React Documentation.  
Retrieved from: <https://reactjs.org>  
Front-end library used in your project.

**ExpressJS Docs. (2024).** Express.js Web Framework.  
Retrieved from: <https://expressjs.com>  
 Back-end server setup for your portal.

**MongoDB Docs. (2024).** MongoDB Developer Manual.  
Retrieved from: <https://www.mongodb.com/docs>  
 Database used for store data content and applications.

**Firebase Docs. (2024).** MongoDB Developer Manual.  
Retrieved from: <https://firebase.google.com/docs>  
 Database used for storing student/staff data.