University of Dhaka

Department of Computer Science and Engineering

CSE-4113 : Internet Programming Lab

Project Title: Website for CSEDU

Requirement Analysis Document

Submission Date: March 18, 2025

Submitted to:

Dr. Mamun Or Rashid
Redwan Ahmed Rizvee
Md. Tanvir Alam

Submitted by:

Meherun Farzana (Roll - 05)
Sarower Jahan Rafin (Roll - 09)
Md. Mahmudul Hasan (Roll - 10)
Md. Emon Khan (Roll - 30)
Diptajoy Mistry (Roll - 34)
Md. Rasel Hossen (Roll - 39)
Md. Mushiur Rahman (Roll - 58)
Mahmudul Hasan (Roll - 60)
Sabrina Hossain (Roll - 61)

Contents

1 Introduction

The *Department of Computer Science and Engineering* (CSE) at the *University of Dhaka* is a leading academic institution in Bangladesh. It offers a wide range of programs and resources for students, faculty, and staff. To support its academic and administrative activities, there is a need for a centralized, interactive, and efficient application platform. This platform will serve as a digital hub for accessing academic resources, managing administrative tasks, and enabling communication among participants. The goal of this project is to design and develop a scalable and user-friendly system that fulfills the needs of the department, using modern tools and technologies.

1.1 Purpose of the System

The purpose of this system is to create a digital platform that improves the academic and administrative experience for students, faculty, and staff within the *Department of Computer Science and Engineering.* The system will provide academic resources, administrative processes, and better communication and collaboration.

1.2 Scope of the System

The scope of this project includes the following key features:

- **User Management:** Support for multiple user roles (students, faculty, admin) with role-based access control.
- **Academic Resources:** A repository for course materials, syllabi, lecture notes, and research papers.
- **Announcements:** Updates on important events, deadlines, and announcements.
- **Event Management:** Scheduling and management of departmental events, seminars, workshops, and meetings.
- **Interactive Interface:** A modern, responsive, and user-friendly interface accessible on both desktop and mobile devices.
- **Database Integration:** Secure storage and retrieval of data using a relational database system.

1.3 Objective and Success Criteria of the System

The main objectives of this project are:

- **Improve User Experience:** Develop an intuitive and responsive interface to ensure ease of use and accessibility for all users.
- **Automate Processes:** Automate administrative tasks to reduce manual effort, minimize errors, and improve efficiency.
- **Scalability:** Design the system to accommodate future growth in terms of users, data, and features.
- **Improve Communication:** Establish communication between students, faculty, and staff.
- **Data Security:** Implement powerful security measures to protect sensitive information to ensure data privacy.

The success criteria of the system are as follows:

- **Functional Completeness:** All planned features (user management, academic resources, announcements, etc.) are fully implemented and operational.
- **User Satisfaction:** High usability scores and positive feedback from students, faculty, and staff.
- **Performance and Scalability:** Fast response times (< 2 seconds) and ability to handle peak loads and future growth.
- **Reliability and Stability:** No critical failures, downtime, or data loss during normal usage.
- **Security and Data Protection:** Secure data storage, role-based access control, and compliance with security standards.
- **Adoption and Usage:** High adoption rates and active usage of key features.
- **Maintainability and Extensibility:** Well-documented, modular code base
- **Timely Delivery and Budget Compliance:** Completed within the agreed timeline and budget.
- **User Feedback and Impact:** Positive feedback and acknowledgment of the system's impact on improving workflows.

1.4 Tools and Technologies

The following tools and technologies will be used to design and develop the system:

Frontend:

- ***ReactJS***: A ***JavaScript*** library for building interactive and dynamic user interfaces.
- ***HTML***/***CSS***: For structuring and styling the web pages.

Backend:

- ***FastAPI***: A modern, fast (high-performance) web framework for building APIs with ***Python***.
- ***PostgreSQL***: A powerful, open-source relational database system for data storage and management.

Other Tools:

- Git/GitHub: For version control and collaborative development.
- Docker: For containerization and deployment.
- Swagger: For API documentation, testing and debugging.

1.5 References

References

[1] React Documentation. *React – A JavaScript Library for Building User Interfaces*. Available at: https://reactjs.org/docs/getting-started.html.

[2] FastAPI Documentation. *FastAPI – Modern, Fast (high-performance) Web Framework for Python*. Available at: https://fastapi.tiangolo.com/.

[3] PostgreSQL Documentation. *PostgreSQL – The World's Most Advanced Open Source Relational Database*. Available at: https://www.postgresql.org/docs/.

[4] Docker Documentation. *Docker – Develop, Ship, and Run Any Application, Anywhere*. Available at: https://docs.docker.com/.

[5] Postman Documentation. *Postman – The Collaboration Platform for API Development*. Available at: https://learning.postman.com/.

1.6 Overview

The Department of Computer Science and Engineering (CSE) at the University of Dhaka requires a modern, centralized software platform to automate academic and administrative processes, provide better communication, and improve accessibility to resources. This project aims to design and develop a scalable and user-friendly software system to meet the needs of students, faculty, and staff.

The system will include features such as **user management, academic resource repositories, real-time announcements, event scheduling, and administrative task automation.** It will be built with ***ReactJS*** for the frontend, ***FastAPI*** for the backend, and ***PostgreSQL*** for database management. The platform will offer a responsive and interactive interface accessible on both desktop and mobile devices.

The project focuses on improving user experience, ensuring data security, and enabling scalability for future growth. This system will serve as a tool for enhancing academic and administrative operations within the Department of CSE.

2 Overall Description

2.1 Product Perspective

The Department of Computer Science and Engineering at the University of Dhaka needs a modern software platform to handle academic and admin tasks, improve communication, and make resources easier to access.

The system will manage users, store academic materials, send announcements, schedule events, and automate admin work. It runs independently but can connect to university databases if needed.

Built with ***ReactJS*** (frontend), ***FastAPI*** (backend), and ***PostgreSQL*** (database), it's designed for security, scalability, and future upgrades. The goal is to boost efficiency and make life easier for students, faculty, and staff.

2.2 Product Functions

The system will include the following major functions:

User Management:

- Account Creation: Faculty and admin accounts can be created with verification.

- Password Recovery: Users can reset their passwords via email if forgotten.

Academic Resources:

- Resource Repository: Access to course materials, syllabi, lecture notes, and research papers.
- Resource Upload: Faculty can upload and update academic resources.

Announcements:

- Event Scheduling: Faculty and admins can schedule and manage departmental events, seminars, and workshops.

2.3 User Profiles

Administrators:

- Oversee the entire system, manage user accounts, and access analytics.
- Responsible for system maintenance and updates.

Faculty:

- Upload and manage academic resources.
- Schedule events, track attendance, and manage grades.
- Access administrative tools for course management.

All(General) Users:

- Access academic resources, view grades, and submit forms.
- View different aspects, goals, achievements and opportunities within the department

2.4 Constraints

Technology:

- The system is designed as a web application and is compatible with modern browsers (Chrome, Firefox, Edge, etc.).
- Mobile compatibility is ensured through responsive design, but native mobile apps are not part of the initial release.

Time Constraints:

- Due to time limitations, advanced features such as AI-based recommendations or integration with external university systems may be deferred to future updates.

Resource Constraints:

- The development team must work within the allocated budget and resources, focusing on main functionalities for the initial release.

2.5 Assumptions and Dependencies

Assumptions:

- **User Familiarity:** Users (students, faculty, and staff) are assumed to have basic familiarity with web applications and online systems.
- **Technical Feasibility:** The chosen technology stack (***ReactJS***, ***FastAPI***, ***PostgreSQL***) is assumed to be capable of supporting the required functionalities.
- **Adoption:** It is assumed that users will adopt the system positively and transition from existing processes to the new platform.
- **Security:** Users trust the system to handle their data securely, and it is assumed that the platform complies with the data protection standards.

Dependencies:

- **Database Management System (*PostgreSQL*):** The system relies on ***PostgreSQL*** for secure and efficient data storage and retrieval.
- **Frontend and Backend Frameworks:** The functionality of the system depends on the effective integration of ***ReactJS*** (frontend) and ***FastAPI*** (backend).
- **Third-Party Services:** Integration with email services for password recovery is essential for user management.
- **Deployment Infrastructure:** The system depends on reliable hosting and deployment infrastructure (e.g., Docker, cloud services) for smooth operation.

3 Proposed System

3.1 Functional Requirements

ID	3.2.1
Name	User and Administration
Description	• The system should allow the ***Super Admin*** to create and manage teacher and admin accounts through institutional mails. • Teachers can update their account information. • Allow password resets for all accounts. • Admins should be allowed to access, manipulate, and publish any data from the database into the website
Priority	High
Reference	

ID	3.2.2
Name	Navbar
Description	• Visualize University logo along with the department name • Include quick links to pages, such as, Homepage, About, Admission, Teachers and Staff, Students Corner, Research, Academics, Events, Facilities and Alumni. • Include quick links to login, logout pages • Necessary quick links for admin
Priority	High
Reference	

ID	3.2.3
Name	Homepage
Description	• Should display a welcome message. • Show the department's mission and vision. • Provide announcements and news updates. • Display upcoming events.
Priority	High
Reference	

ID	3.2.4
Name	About Us
Description	• Provide an overview of the department. • Include a message from the Head of the Department. • Showcase department history and achievements. • Show contact information and location on a map.
Priority	Medium
Reference	

ID	3.2.5
Name	Academics
Description	• Provide information about undergraduate, postgraduate, and PhD programs. • Display course structures, syllabi, and credit details. • Include class schedules and timetables. • Provide an academic calendar with important dates.
Priority	High
Reference	

ID	3.2.5
Name	Course Management
Description	• A part of academics • Teachers should be able able to manage their ongoing courses by sharing resources
Priority	High
Reference	

ID	3.2.6
Name	Faculty and Staff
Description	• Maintain a faculty directory with names, designations, and research backgrounds. • Provide links to faculty members' LinkedIn and Google Scholar profiles. • Display faculty and staff details, office room numbers, and office hours.
Priority	High
Reference	

ID	3.2.7
Name	Research and Innovation
Description	• Provide information on research areas and groups. • Show research publications and ongoing projects. • Display collaboration opportunities and funding options.
Priority	Medium
Reference	

ID	3.2.8
Name	Admissions
Description	• Provide admission criteria and process details. • Allow applicants to download and submit application forms. • Display scholarship and financial aid information. • Provide an FAQ section related to admissions.
Priority	High
Reference	

ID	3.2.9
Name	Student Corner
Description	• Display information on student clubs and their activities, and other organizations. • Highlight student achievements. • Provide access to academic resources such as lecture notes and past question papers.
Priority	Medium
Reference	

ID	3.2.10
Name	Notices and Circulars
Description	• Allow admins to publish official announcements. • Provide department policies, rules, and regulations. • Display exam schedules, results, and other academic notices.
Priority	High
Reference	

ID	3.2.11
Name	Events and Conferences
Description	• Display information on upcoming workshops and seminars. • Provide details on hackathons and coding competitions. • Show details on departmental events such as covocations, parties, and picnics etc • Provide event participation details and allow registration
Priority	Medium
Reference	

ID	3.2.12
Name	Alumni
Description	• Showcase alumni success stories. • Provide information on alumni networks and events.
Priority	Low
Reference	

ID	3.2.13
Name	Facilities
Description	• Provide information on departmental facilities such as labs, libraries, and auditoriums.
Priority	Medium
Reference	

ID	3.2.14
Name	AI Assistant
Description	• Implement an AI assistant to promptly answer frequently asked questions. • Provide information on admission, courses, faculty, and events. • Allow users to ask questions, and get answers in both Bengali and English.
Priority	Medium
Reference	

3.2 Non Functional Requirements

1. **Performance Requirements**

    - The system should respond to user requests within **2 seconds** under normal load.
    - It should support at least **1000 concurrent users** without performance degradation.

    - Database queries should be optimized to execute within **500 milliseconds** on average.
    - The system should be able to handle a **10,000-record dataset** efficiently.

2. **Security Requirements**
    - User authentication should be enforced using **OAuth 2.0 or JWT-based authentication**.
    - Institutional email verification should be required for teacher and admin accounts.
    - Passwords should be **hashed and stored securely** using bcrypt or Argon2.
    - The system should prevent **SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF)**.
    - Role-based access control (RBAC) should be implemented to restrict access to different sections of the system.

3. **Usability Requirements**
    - The user interface should be **intuitive and accessible** to users of different technical backgrounds.
    - The system should follow **WCAG 2.1 Level AA** accessibility standards.
    - Mobile responsiveness should be ensured across devices with different screen sizes.
    - The system should support **Bengali and English languages**.

4. **Availability Requirements**
    - The system should have **99.5% uptime** except for planned maintenance.
    - In case of a failure, the system should be **recoverable within 30 minutes**.
    - Automated database backups should be scheduled **daily** and retained for at least **30 days**.

5. **Maintainability and Scalability**

- The system should follow a **modular architecture** for easy maintenance and updates.
- The backend should be designed using **RESTful API principles**, and **Software Design Best Practices and Patterns**.
- The system should be **horizontally scalable** to accommodate an increase in users.
- The system should be **vertically scalable** to accommodate an increase in data (such as, resources) volume.
- Logging and monitoring should be implemented using **ELK stack or Prometheus**.

6. **Data Integrity and Consistency**
   - The database should maintain **ACID compliance** for transactions.
   - The system should ensure that **no duplicate user accounts** are created.
   - All updates to the database should follow **proper validation and constraints**.
7. **Compatibility Requirements**
   - The frontend should be **compatible with modern browsers** (Chrome, Firefox, Edge, Safari).
   - The backend should be deployable on **Linux-based cloud servers**.
   - The database should support either **SQL-based relational models (PostgreSQL/MySQL/SQLite)** or **no SQL models (MongoDB/Redis)**.
8. **Logging and Monitoring**
   - All system events should be **logged with timestamps and user activity**.
   - Error logs should be stored and categorized based on severity.
   - A monitoring dashboard should be available for **system health checks**.
9. **Disaster Recovery and Backup**

- The system should have a **disaster recovery plan** to restore services quickly.
- Backups should be **stored in a separate location** to avoid single-point failure.
- Regular **testing of backup restoration** should be performed.

3.3 System Models

3.3.1 Scenarios

1. **View Homepage**

   Flow of Events:

   - The user visits the website homepage.
   - They see a welcome message, department mission & vision.
   - They check announcements, upcoming events, and news.
   - They use quick links to navigate to specific sections (Admissions, Research, Faculty, etc.).

2. **View About Us**

   Flow of Events:

   - The user selects the ”About Us” section.
   - They read about the department’s history and achievements.
   - They view a message from the Head of the Department.
   - They can download the ”Department Information Booklet.”
   - They check contact details if they need further information.
   - They can follow the Department on Social Media.

3. **Browse Academic Programs**

   Flow of Events:

   - The user selects ”Academics” from the menu.
   - They choose between Undergraduate or Postgraduate.
   - They view course structures, syllabi, and credit details.
   - They view Program Educational Objectives (PEO) & Program Outcomes (PO).

- They check academic calendars and class schedules.

4. **View Faculty & Staff Directory**

   Flow of Events:

   - The user visits the "Faculty & Staff" section.
   - They search for a faculty member by name or department.
   - They view faculty details, including designation and specialization.
   - They check office hours and contact information.

5. **Explore Research & Innovation**

   Flow of Events:

   - The user selects "Research & Innovation" from the menu.
   - They browse research areas and ongoing projects.
   - They view details on research groups and publications.
   - They check collaboration opportunities and funding details.

6. **Apply for Admission**

   Flow of Events:

   - The prospective student visits the "Admissions" page.
   - They read about admission criteria and process.
   - They download the application form.
   - They fill out and submit the form along with required documents.
   - They receive a confirmation email.

7. **View Student Corner**

   Flow of Events:

   - The student selects "Student Corner" from the menu.
   - They browse student clubs and organizations.
   - They check job opportunities.
   - They explore student achievements and project showcases.
   - They download academic resources such as lecture notes and study materials.

8. **Register for an Event**

   Flow of Events:

   - The user visits the "Events & Conferences" section.
   - They browse workshops, seminars, and hackathons.
   - They click on a specific event to view details.
   - They fill out the registration form.
   - They receive a confirmation email with event details.

9. **Check the Alumni Network**

   Flow of Events:

   - The alumni visit the "Alumni" section.
   - They read alumni success stories.
   - They receive updates on alumni events and mentorship programs.

10. **Access Resources & Facilities**

    Flow of Events:

    - The user selects "Resources & Facilities" from the menu.
    - They browse information about labs, libraries, and computing resources.
    - They find access instructions for digital resources.
    - They download department policies if needed.

11. **Check Notices & Circulars**

    Flow of Events:

    - The user selects "Notices & Circulars" from the menu.
    - They browse a list of official announcements and updates.
    - They check details about upcoming exams and policies.
    - They download any important circulars.

12. **Contact the Department**

    Flow of Events:

    - The user visits the "Contact Us" page.
    - They find the department's phone number and email.

- They fill out an inquiry form with their question.
- The system sends an email confirmation and assigns the inquiry to a staff member.

13. **Customize Profiles**

    Flow of Events:

    - Faculty logs into their account.
    - Navigate to ”Profile” or ”Account Settings.”
    - View current profile information.
    - Select option to edit profile.
    - Update information (e.g., change picture, modify contact details).
    - Set notification preferences.
    - Confirm updates by clicking ”Save.”
    - Receive confirmation message of successful update.
    - User can log out or navigate elsewhere.

14. **Maintain Course Resources**

    Flow of Events:

    - The faculty logs into the system.
    - They navigate to the ”Course Resources” or ”Learning Materials” section.
    - They select the relevant course.
    - They upload new course materials (lecture notes, study materials, presentations, etc.).
    - They can edit or delete existing materials if necessary.
    - The system saves the updates.
    - Users can now view and download the materials.

15. **Chatbot Interaction**

    Flow of Events:

    - The user clicks on the chatbot icon to initiate a conversation.
    - The chatbot greets the user and offers options (e.g., ”How can I help you today?”).

- The user selects a category or types a query (e.g., ”How do I apply for admission?”).
- The chatbot processes the query and provides relevant responses based on predefined FAQs and database information.
- If the chatbot cannot answer the question:
  - It provides alternative resources (e.g., links to relevant pages).
  - It offers to connect the user with a human representative (if available).
- The user continues the conversation or exits the chatbot.

16. **Scenarios for Admin Panel (For Department Admins Only):**

    (a) **Manage Announcements & Notices**
    Flow of Events:
    - The admin logs into the system.
    - They select ”Notices & Circulars” or ”Announcements.”
    - They create a new notice and enter the details.
    - They publish the notice, making it visible to users.

    (b) **Update Faculty Directory**
    Flow of Events:
    - The admin logs in to the admin panel.
    - They navigate to ”Faculty & Staff” management.
    - They add a new faculty member or update existing details.
    - They save changes, updating the public directory.

    (c) **Create Faculty Account**
    Flow of Events:
    - The admin logs into the admin panel.
    - They navigate to the ”Faculty Management” section.
    - They select the option to ”Create New Faculty Account.”
    - They enter faculty details (name, email, designation, etc.).
    - They generate login credentials (or allow automatic email generation).
    - They confirm and submit the details.
    - The system creates the faculty account.
    - The faculty member receives an email with login credentials and setup instructions.

(d) **Manage Events & Conferences**

Flow of Events:

- The admin logs into the admin panel.
- They select ”Event Management.”
- They add event details (title, date, description).
- They publish the event, making it available for users.

3.3.2 Use Cases

Name	Use Case 1: View Homepage
Actor	Visitor
Flow of Events	• The user visits the website homepage. • They see a welcome message, department mission & vision. • They check announcements, upcoming events, and news. • They use quick links to navigate to specific sections (Admissions, Research, Faculty, etc.).
Entry Condition	The user accesses the website through a browser.
Exit Condition	The user successfully finds the information they need or navigates to another section.

Name	Use Case 2: View About Us
Actor	Visitor
Flow of Events	• The user selects the ”About Us” section. • They read about the department’s history and achievements. • They view a message from the Head of the Department. • They can download the ”Department Information Booklet” also. • They check contact details if they need further information. • They can follow the Department on Social Media.
Entry Condition	The user selects the ”About Us” section from the navigation menu.
Exit Condition	The user learns about the department’s background and contact details.

Name	Use Case 3: Browse Academic Programs
Actor	Student, Visitor
Flow of Events	• The user selects ”Academics” from the menu. • They choose between Undergraduate and Postgraduate programs. • They view course structures, syllabi, and credit details. • They view Program Educational Objectives (PEO) & Program Outcomes (PO). • They check academic calendars and class schedules.
Entry Condition	The user selects ”Academics” from the menu.
Exit Condition	The user gathers program details or downloads relevant documents.

Name	Use Case 4: View Faculty & Staff Directory
Actor	Visitor
Flow of Events	• The user visits the ”Faculty & Staff” section. • They search for a faculty member by name or department. • They view faculty details, including designation and specialization. • They check office hours and contact information.
Entry Condition	The user selects ”Faculty & Staff” from the menu.
Exit Condition	The user successfully finds faculty information.

Name	Use Case 5: Explore Research & Innovation
Actor	Student, Faculty, Researcher, Visitor
Flow of Events	• The user selects ”Research & Innovation” from the menu. • They browse research areas and ongoing projects. • They view details on research groups and publications. • They check collaboration opportunities and funding details.
Entry Condition	The user selects ”Research & Innovation” from the menu.
Exit Condition	The user learns about department research and collaboration options.

Name	Use Case 6: Apply for Admission
Actor	Prospective Student
Flow of Events	• The prospective student visits the ”Admissions” page. • They read about admission criteria and process. • They download the application form. • They fill out and submit the form along with required documents. • They receive a confirmation email.
Entry Condition	The user selects ”Admissions” and is interested in applying.
Exit Condition	The user submits the application successfully.

Name	Use Case 7: View Student Corner
Actor	Student, Visitor
Flow of Events	• The student selects ”Student Corner” from the menu. • They browse student clubs and organizations. • They check job opportunities. • They explore student achievements and project showcases. • They download academic resources.
Entry Condition	The student selects ”Student Corner” from the menu.
Exit Condition	The student finds relevant information or downloads resources.

Name	Use Case 8: Register for an Event
Actor	Student, Faculty, Visitor
Flow of Events	• The user visits the ”Events & Conferences” section. • They browse workshops, seminars, and hackathons. • They click on a specific event to view details. • They fill out the registration form. • They receive a confirmation email with event details.
Entry Condition	The user selects ”Events & Conferences” from the menu.
Exit Condition	The user successfully registers for the event.

Name	Use Case 9: Check the Alumni Network
Actor	Alumni
Flow of Events	• The alumni visit the ”Alumni” section. • They read alumni success stories. • They receive updates on alumni events and mentorship programs.
Entry Condition	The user selects ”Alumni” and wants to connect with the network.
Exit Condition	The alumni are registered and receive updates on events.

Name	Use Case 10: Access Resources & Facilities
Actor	Student, Faculty
Flow of Events	• The user selects ”Resources & Facilities” from the menu. • They browse information about labs, libraries, and computing resources. • They find access instructions for digital resources. • They download department policies if needed.
Entry Condition	The user selects ”Contact Us” from the menu.
Exit Condition	The user successfully submits an inquiry.

Name	Use Case 11: Check Notices & Circulars
Actor	Student, Faculty
Flow of Events	• The user selects "Notices & Circulars" from the menu. • They browse a list of official announcements and updates. • They check details about upcoming exams and policies. • They download any important circulars.
Entry Condition	The admin logs into the system.
Exit Condition	The notice is successfully posted.

Name	Use Case 12: Contact the Department
Actor	Visitor
Flow of Events	• The user visits the "Contact Us" page. • They find the department's phone number and email. • They fill out an inquiry form with their question. • The system sends an email confirmation and assigns the inquiry to a staff member.
Entry Condition	The admin logs into the admin panel.
Exit Condition	The faculty directory is successfully updated.

Name	Use Case 13: Customize Profiles
Actor	Faculty
Flow of Events	• User logs into their account. • Navigate to ”Profile” or ”Account Settings.” • View current profile information. • Select option to edit profile. • Update information (e.g., change picture, modify contact details). • Set notification preferences. • Confirm updates by clicking ”Save.” • Receive confirmation message of successful update. • User can log out or navigate elsewhere.
Entry Condition	• The user must be logged into their account. • The system must have an existing user profile associated with the account.
Exit Condition	• The profile information is successfully updated in the system. • The system displays a confirmation message. • The user is redirected to their profile page or another section of the site.

Name	Use Case 14: Maintain Course Resources
Actor	Faculty
Flow of Events	• The faculty logs into the system. • They navigate to the ”Course Resources” or ”Learning Materials” section. • They select the relevant course. • They upload new course materials (lecture notes, assignments, presentations, etc.). • They can edit or delete existing materials if necessary. • They organize the resources into categories (e.g., Week-wise, Topic-wise). • They set access permissions (e.g., public, only enrolled students). • The system saves the updates and notifies students if a new resource is added. • Students can now view and download the materials.
Entry Condition	• The faculty member must be logged into their account. • The course must be assigned to the faculty member. • The system must allow file uploads and content management.
Exit Condition	• The course resources are successfully uploaded, updated, or deleted. • The system logs all changes for tracking.

Name	Use Case 15: Chatbot Interaction
Actor	Visitor
Flow of Events	• The user clicks on the chatbot icon to initiate a conversation. • The chatbot greets the user and offers options (e.g., ”How can I help you today?”). • The user selects a category or types a query (e.g., ”How do I apply for admission?”). • The chatbot processes the query and provides relevant responses based on predefined FAQs and database information. • If the chatbot cannot answer the question: – It provides alternative resources (e.g., links to relevant pages). – It offers to connect the user with a human representative (if available). • The user continues the conversation or exits the chatbot. • If applicable, the chatbot logs the conversation for future improvements or admin review.
Entry Condition	• The user visits the CSE Department website • The chatbot logs the interaction for analytics and improvement.
Exit Condition	• The user receives the necessary information or is redirected to a human representative. • The chatbot widget is available and active.

Name	Use Case 16: Manage Announcements & Notices
Actor	Department Admin
Flow of Events	• The admin logs into the system. • They select ”Notices & Circulars” or ”Announcements.” • They create a new notice and enter the details. • They publish the notice, making it visible to users.
Entry Condition	The admin logs into the admin panel.
Exit Condition	The event is successfully posted on the website.

Name	Use Case 17: Update Faculty Directory
Actor	Department Admin
Flow of Events	• The admin logs in to the admin panel. • They navigate to ”Faculty & Staff” management. • They add a new faculty member or update existing details. • They save changes, updating the public directory.
Entry Condition	The admin logs into the admin panel with valid credentials.
Exit Condition	The faculty directory is successfully updated and visible to all users.

Name	Use Case 18: Create Faculty Account
Actor	Department Admin
Flow of Events	• The admin logs into the admin panel. • They navigate to the ”Faculty Management” section. • They select the option to ”Create New Faculty Account.” • They enter faculty details (name, email, designation, etc.). • They generate login credentials (or allow automatic email generation). • They confirm and submit the details. • The system creates the faculty account. • The faculty member receives an email with login credentials and setup instructions.
Entry Condition	• The admin must be logged into the system. • The system must have the necessary permissions for the admin to create faculty accounts
Exit Condition	• A new faculty account is successfully created. • The faculty member receives login details • The admin sees a confirmation message

Name	Use Case 19: Manage Events & Conferences
Actor	Department Admin
Flow of Events	• The admin logs into the admin panel. • They select ”Event Management.” • They add event details (title, date, description). • They publish the event, making it available for users.
Entry Condition	• The admin logs into the admin panel with valid credentials.
Exit Condition	• The event is successfully added, updated, or removed on the website. • Users can register or view updated event details.

3.3.3 Use Case Model

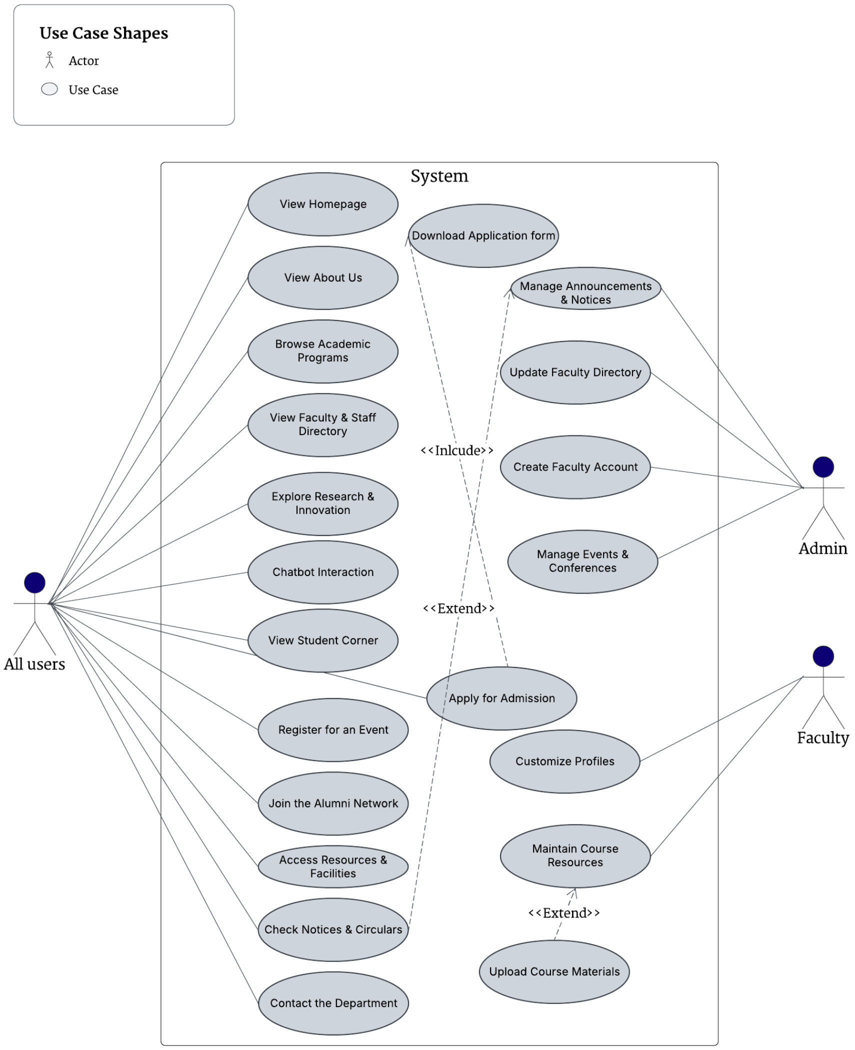

Figure 1: Use case Diagram for TestCraft

3.3.4 Dynamic Model

- **Sequence Diagram**

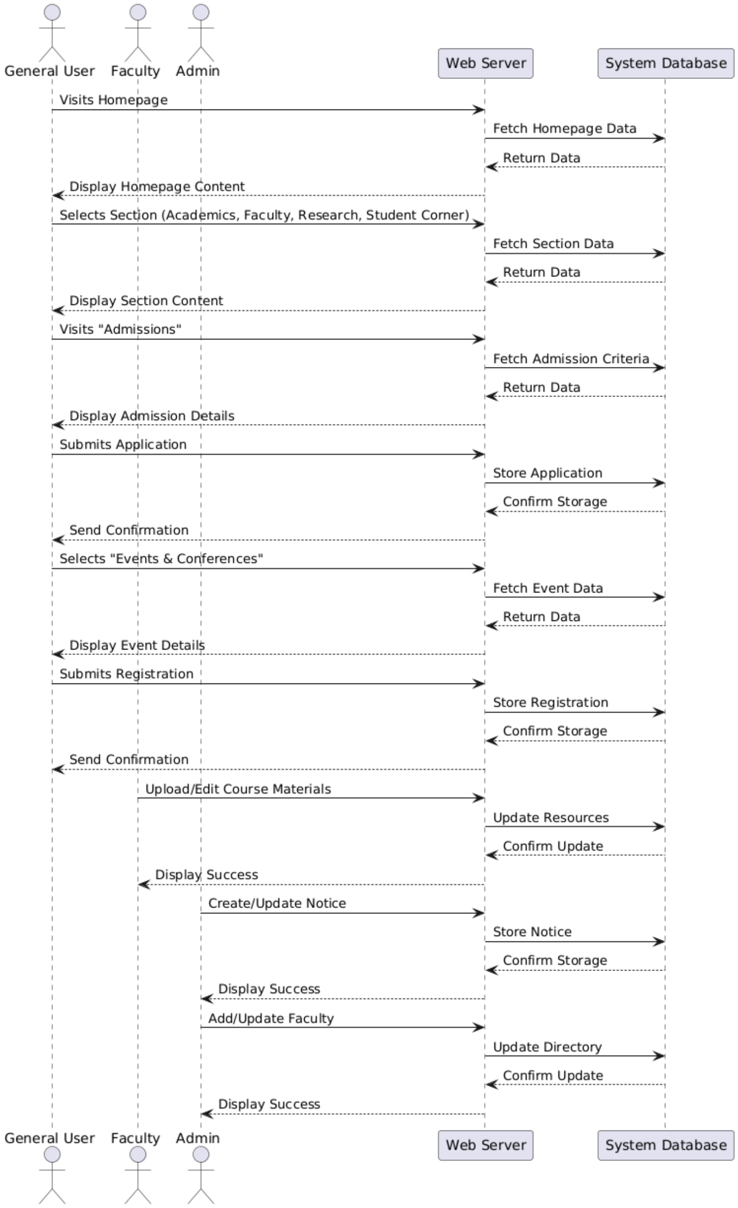

Figure 2: Sequence Diagram

- **Activity Diagram**

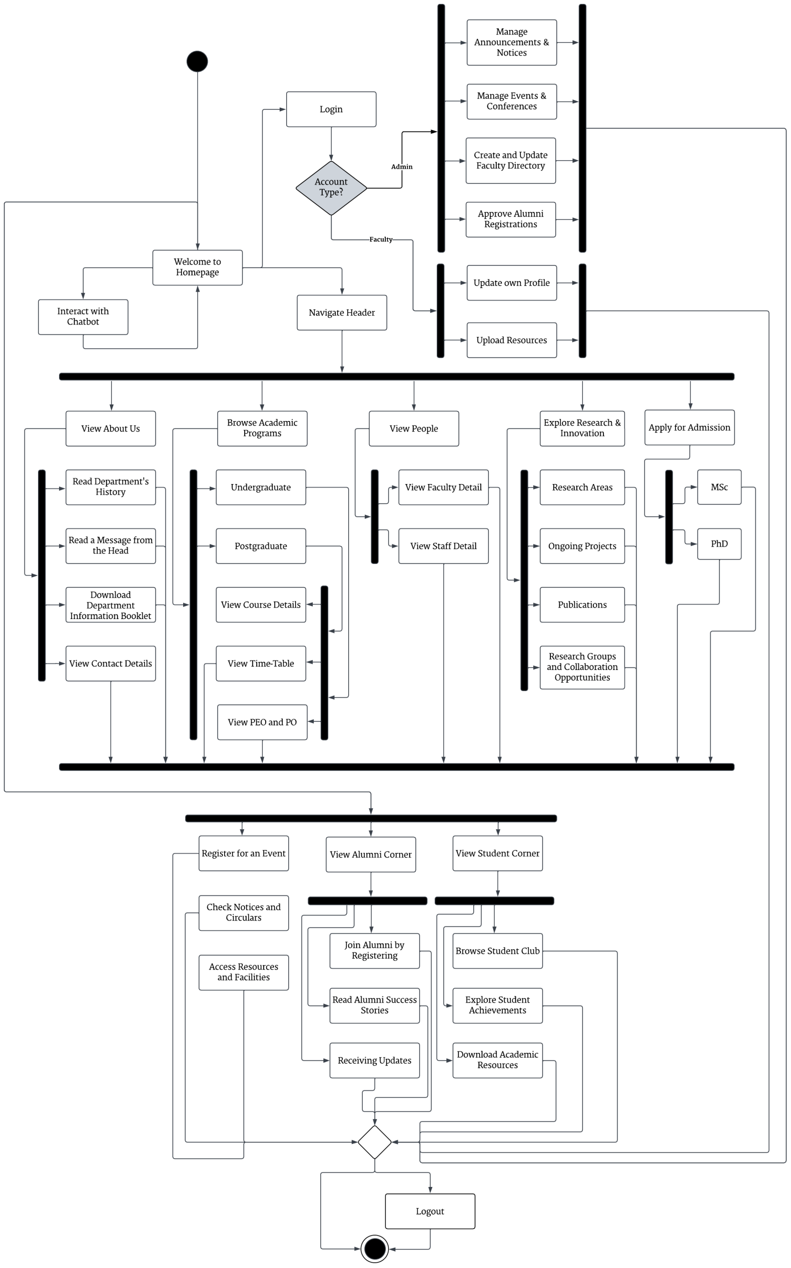

Figure 3: Activity Diagram

- **State Diagrams**

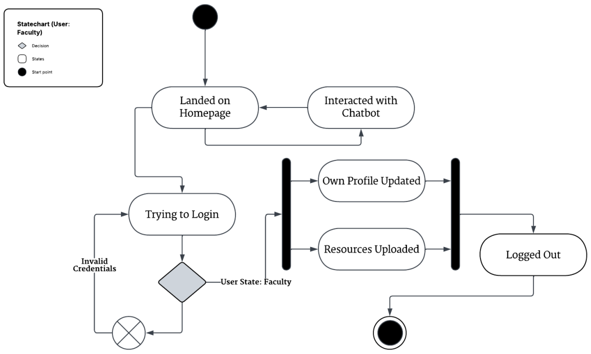

Figure 4: Faculty state diagram

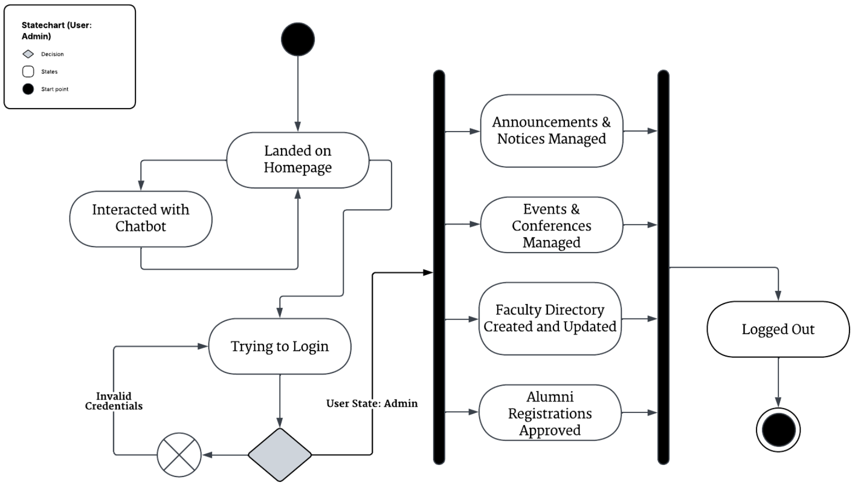

Figure 5: Admin state diagram

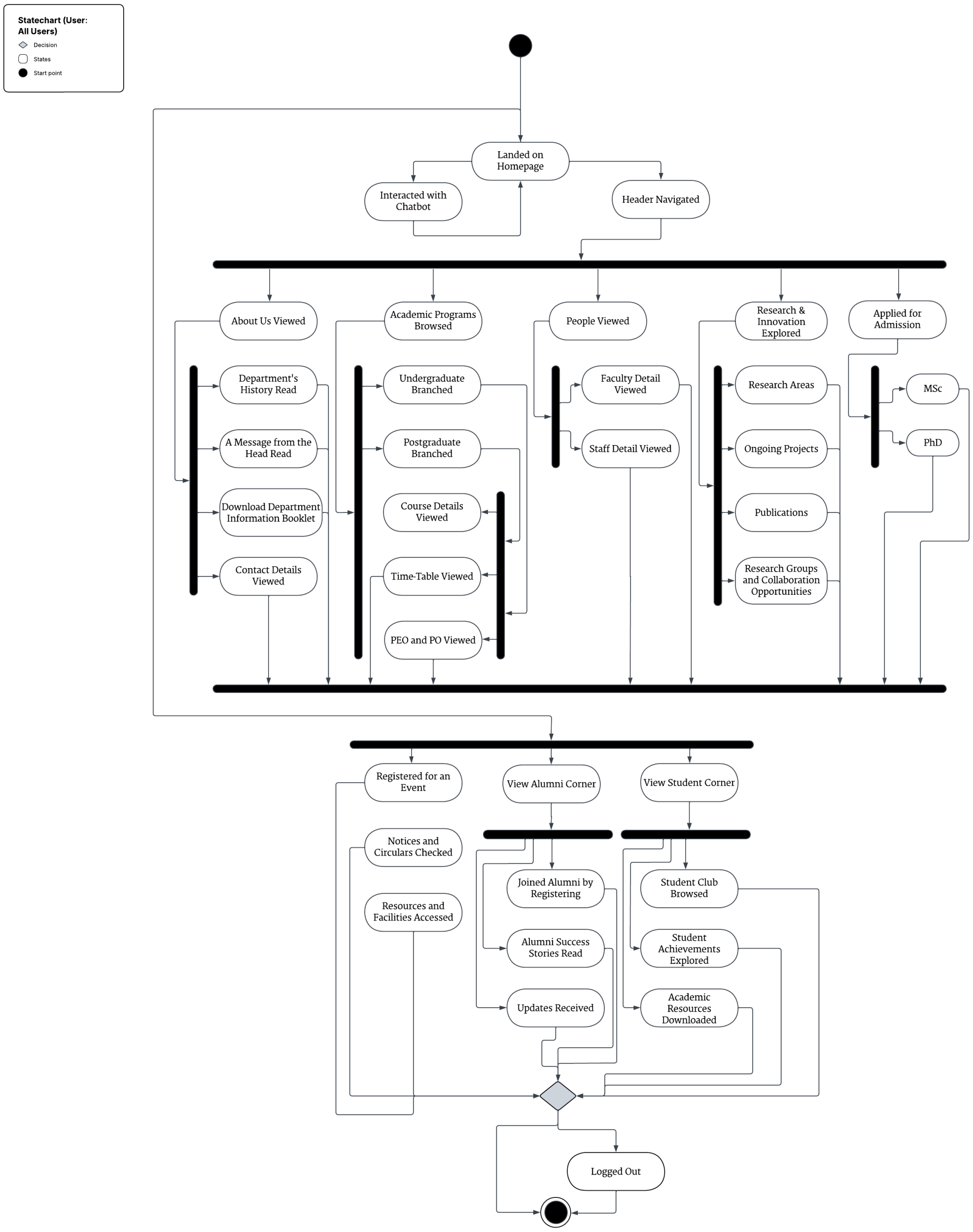

Figure 6: All users state diagram

3.3.5 User Interface

The prototype for this application was designed while maintaining a consistent color scheme and typography throughout the interface. The application is a web-based platform, and its responsiveness has been thoroughly tested across various devices, including desktops, laptops, iPads, and mobile phones. For development, we will use ***ReactJS*** for the frontend, providing a dynamic and efficient user experience, while ***FastAPI*** will be used for the backend, ensuring fast performance and scalability. The overall User Interface workflow is described below:

1. **Homepage Workflow:**
    - The user lands on the homepage.
    - The homepage displays:
        - Welcome message
        - Department mission & vision
        - Announcements & news
        - Upcoming events
        - Quick links (admissions, faculty, research, etc.)
    - Users can navigate to different sections from here.
2. **About Us Workflow:**
    - User selects ”About Us” from the navigation menu.
    - Overview of the department is displayed.
    - Users can read the message from the Head of the Department.
    - History and achievements are accessible.
3. **Academics Workflow:**
    - User selects ”Academics.”
    - A list of academic programs is displayed, including:
        - Undergraduate Programs (course structure, syllabus, credit details)
        - Postgraduate Programs
        - PhD Programs
    - Users can access:
        - Course Catalog

    - Class schedules & timetables
    - Academic Calendar

4. **Faculty & Staff Workflow:**
    - User selects ”Faculty & Staff.”
    - Faculty directory displays names, designations, specializations, contact details, and research backgrounds.
    - Users can access faculty members’ LinkedIn and Google Scholar profiles.
    - Staff details and office hours are available.

5. **Research & Innovation Workflow:**
    - User selects ”Research & Innovation.”
    - Research areas are displayed with detailed descriptions.
    - Publications are available for browsing or download.
    - Research groups, labs, and ongoing projects are listed.
    - Collaboration opportunities and research funding information are accessible.

6. **Admissions Workflow:**
    - User selects ”Admissions.”
    - Admission criteria and process for MSc and PhD programs are outlined.
    - Application forms are available for download.
    - Scholarship and financial aid details are provided.
    - Users can browse FAQs for common queries.

7. **Student Corner Workflow:**
    - User selects ”Student Corner.”
    - Student clubs and organizations are listed with descriptions and join options.
    - Student achievements are showcased.
    - Academic resources, including lecture notes and previous question papers, are accessible.

8. **Events & Conferences Workflow:**
   - User selects "Events & Conferences."
   - Upcoming workshops, seminars, hackathons, and coding competitions are displayed.
   - Users can register for events directly from this section.
   - Past event highlights include images and videos.
9. **Alumni Workflow:**
   - User selects "Alumni."
   - Success stories are displayed.
   - Alumni network and event details are accessible.
10. **Resources & Facilities Workflow:**
    - User selects "Resources & Facilities."
    - Information about laboratories, equipment, and computing facilities is available.
    - Library and digital resources are listed.
    - Department policies can be viewed or downloaded.
11. **Notices & Circulars Workflow:**
    - User selects "Notices & Circulars."
    - Official announcements and examination notices are displayed.
    - Departmental policies and updates are listed.
12. **Contact Us Workflow:**
    - User selects "Contact Us."
    - Department location and an interactive map are displayed.
    - Email and phone contact details are provided.
    - An inquiry form allows users to submit questions directly.

3.3.6 Software Interface

Front-End (Presentation Layer)

- **Web Framework (*ReactJS*)**: Handles the UI with reusable components and client-side rendering.
- **Styling (*CSS*)**: Styles the user interface, including layout, fonts, and colors.
- **Programming Language (*JavaScript*)**: Enhances interactivity and dynamic behavior of the web application.

Back-End (Logic Layer)

- **Programming Language (*Python*)**: Manages server-side logic, API requests, and data processing.
- **Web Framework (*FastAPI*)**: Provides a fast and efficient backend with API routing, authentication, and database interactions.
- **Database (*PostgreSQL*/*MongoDB*)**: Stores all application data, including user information, department details, research data, event details, and notices.

Data Flow

- **Incoming Data:**
  - **User Input**: Users provide data through web forms in the React frontend, such as sign-up details, event registrations, and feedback.
  - **API Requests (Optional)**: If integrating with external *API*s (e.g., research publications, funding details, or alumni networks), *FastAPI* will handle incoming data from these sources.
- **Outgoing Data:**
  - **API Responses**: *FastAPI* sends structured *JSON* responses to the React frontend, which dynamically updates the UI.
  - **Database Interactions**: *FastAPI* communicates with *PostgreSQL/MongoDB* to store and retrieve data, such as faculty details, academic records, research projects, and student information.

  - **Emails**: The backend can send automated emails for password change requests
  - **API Interactions (Optional)**: If needed, *FastAPI* can send data to external services for research data integration, collaboration tools, or alumni networking.

Communication Services

- **Database Management System (*PostgreSQL* or *MongoDB*)**: The chosen database management system provides functionalities for storing, retrieving, and manipulating data within the database.

Implementation Constraints

- **Data Security**: Sensitive user data (passwords, personal information) should be stored securely using hashing algorithms. Research and academic records must be protected from unauthorized access.
- **API Authentication**: If external APIs are used, proper authentication methods will be established to securely exchange data.
- **Role-Based Access Control**: Faculty, students, and administrators should have different access permissions to manage data effectively.

3.3.7 Hardware Interface

As per the plan for this project, the hardware parts include the servers for hosting the app and the devices clients use to access it online. The way the software connects with the hardware involves supporting different web browsers like Chrome, Firefox, and Safari, and making sure the devices have internet access to reach the app. The physical aspects of this connection include the network linking client devices and servers, the specifications of server hardware, and how much data the server can store. To communicate between client devices and servers, the system will use HTTP over the internet, with SSL encryption for safety. The hardware needs to meet certain requirements to run the app, outlined in the installation and deployment guides. However, since this is a web app, it can work on any device with an internet connection.

4 Supporting Information

- **Installation**

  - **Frontend:** If ***NodeJS*** is already installed, run the following commands in the terminal to set up and start the ***ReactJS*** frontend:

    ```
    npm install
    npm run start
    ```

  - **Backend:** If ***Python*** is already installed, run the following commands in the terminal to set up and start the ***FastAPI*** backend:

    ```
    pip install -r requirements.txt
    uvicorn main:app --reload
    ```

- **Deployment**

  - **Method 1:** Deploy frontend server into ***Vercel*** from the git repository pretty easily. See `https://vercel.com/` for more information

    And then deploy backend server into any cloud service (***Google Cloud, AWS*** etc)

  - **Method 2:** Containerize the application using ***Docker*** and host it from a ***Linux*** based cloud server. Or manually set up a custom ***Linux*** server and host under the domain ***du.ac.bd***