

**Final Project: Developing a Comprehensive GUI-Based College Management System**  
**CSAS2124 – Object Oriented Programming – II**  
**Spring 2024**

Due: May 10, 2024

Presentation time – 12.20 pm to 2.20 pm

**Objective:**

The main objective of this project is to design and develop an application-oriented GUI-based College Campus Management System (CMS). This system will incorporate a range of functionalities, including administrative task management, networking capabilities, concurrency features, data visualization tools, and web scraping capabilities. This project will provide an opportunity for you to apply your knowledge of Python programming, OOPs concepts, GUI development, networking protocols, concurrency mechanisms, data visualization techniques, web scraping methodologies, etc. By implementing this user-friendly application, you'll be able to demonstrate proficiency in various aspects of software engineering, thereby enhancing your understanding and expertise in these domains.

**Project Requirements:**

1. Module 1: User Authentication:

- The User Authentication module is responsible for managing user access to the CMS. It ensures that only authorized users, including administrators, faculty members, and students, can log in and access the system's functionalities. This module implements a secure login system with authentication mechanisms based on user roles and permissions.
  - Create a user database and store user credentials, including usernames and passwords, securely.
  - Implement a login Functionality, that allows users to log in using their credentials.
  - Assign specific roles to users based on Role-Based Access Control (RBAC). E.g., administrators, faculty members, students, and grants appropriate access.

2. Module2: Graphical User Interfaces:

- Design visually appealing user interfaces for administrators, faculty members, and students using the GUI development library Tkinter.
- Include features such as forms, buttons, dropdown menus, and input fields to facilitate user interaction.

3. Module 3: Functionalities for Administrators:

- Allow administrators to manage various administrative tasks, including student enrollments, course registrations, and faculty assignments.
- Student Management:

- Make the administrator privilege to manage student enrollments, including adding, editing, or removing student records.
  - Provide only view and update privilege functionality for viewing and updating student information such as personal details, academic status, and enrollment status.
  - Course Management:
    - Give privileges to administrators to manage course registrations by adding new courses, updating existing ones, removing courses, updating course schedules, etc.
  - Faculty Management:
    - Enables administrators to update faculty profiles, including contact information, academic qualifications, and teaching assignments.
  - Academic Schedule Management:
    - Make administrators create and manage academic schedules for courses, exams, and other academic events.
    - Provides functionality for scheduling classes, labs, exams, and other academic activities, taking into account faculty availability and student preferences.
4. Module 4: Functionalities for Faculty Members:
- This module is to provide functionalities to faculty members within the CMS, enabling them to effectively manage their courses, interact with students, and track academic progress.
    - Course management: Allows faculty members to access course materials such as syllabi, lecture slides, and reading materials.
    - Make the faculty members track student progress by viewing grades, assignment submissions, and attendance records.
    - Give a facility for the creation, posting, and grading of assignments and assessments.
    - (optional) Allows faculty members to provide feedback on assignments and communicate grades to students, faculty members to post announcements, etc.
5. Module5: Functionalities for Students:
- This module encourages students to view courses, register for classes, and access course materials.
  - Provide features for students to submit assignments, view grades, and communicate with faculty members(optional), etc.
6. Module6: Data Management:
- The module integrates a robust DBMS such as SQLite3 or MySQL to efficiently store and manage various types of data critical to academic operations. This includes student records, course information, academic schedules, and administrative tasks.
  - Users can add new records to the database, such as enrolling students, creating new courses, or scheduling academic events.
  - Users can retrieve and view existing data records stored in the database. This includes accessing student profiles, course details, class schedules, etc.

- Authorized users can modify existing data records as needed. This includes updating student information (e.g., contact details, enrollment status), adjusting course details (e.g., schedule changes, instructor assignments), etc.
- Users with appropriate permissions(admin) can delete data records from the database.

#### 7. Module7: Networking module:

- The networking module enables secure communication between the client and server components of the CMS.
- Clients send requests to the server component, such as retrieving course information or submitting assignments.
- The Networking Module processes these requests locally, without the need for external networks or internet connections.
- It handles the flow of data between the client interface and the server components within the local environment.

#### 8. Module8: Concurrency and Parallelism:

In software development, using concurrency and parallelism techniques is really important. These techniques help make programs faster and more responsive. This part of our project is all about using these techniques to make the College Management System work better. Let's see how we can use concurrency and parallelism to deal with different problems at the same time.

- Use concurrency and parallelism techniques to enhance the performance of data processing tasks, such as fetching data from the server, processing large datasets, or handling multiple client requests simultaneously.
- When students, faculty, or administrators access the CMS to retrieve data from the server (such as course information, student records, etc.), concurrency techniques like multithreading can be employed.
- Multithreading allows the CMS to initiate multiple threads, each responsible for fetching data asynchronously. This ensures that the user interface remains responsive while data retrieval operations are in progress.

#### 9. Module 9: Data Visualization:

Data visualization is a powerful tool that helps us understand complex information by presenting it in a visual format. In our CMS, we can use data visualization to analyze various aspects of academic data like student performance, course popularity, faculty, etc.

- Analyze academic data to generate insights into student performance, course popularity, faculty workload, and other relevant metrics.
- Visualize relevant data using data visualization libraries such as Matplotlib or Plotly for creating various types of visualizations, including charts, graphs, and dashboards.

#### 10. Module 10 :Web Scraping:

Web scraping is important for the CMS as it automatically gathers important data from external sources. This feature helps users have access to the latest information, such as course details, faculty profiles, student resources, academic calendars, course reviews, etc. This web scraping helps to keep the system up-to-date and informative for users.

- Integrate web scraping functionality to gather external data sources such as course information, faculty profiles, student resources, academic calendars, and course reviews from relevant educational websites.
- We'll use web scraping to gather data exclusively from SHU and its subdomain websites, such as various department pages, online course platforms, and academic forums. This focused approach ensures that the College Management System provides users with comprehensive and insightful information directly from SHU's online resources.
- We'll follow to ethical guidelines and terms of service by refraining from scraping data from external websites. Only data from SHU sources will be collected, ensuring compliance with ethical standards and terms.
- Implement error handling and validation mechanisms to handle inconsistencies or changes in scraped data.

#### 11. Error Handling and Validation:

- Implement error handling and validation mechanisms to ensure data integrity and prevent unauthorized access or misuse of the CMS.

#### 12. Project Report:

- Provide a comprehensive project report for your CMS, comprising:
  - Write an objective and introduction to the College Management System
  - Draw an overall system architecture.
  - Module Descriptions: Write detailed comprehensions into each module's functionalities and implementation specifics.
  - Implementation Details: Include information on the technologies, modules, and other components used in system development.
  - Data Management, Database Schema, Networking, Concurrency, and Data Visualization: Explain these aspects thoroughly.
  - Web Scraping Implementation: Describe the approach used for implementation.
  - Challenges and Solutions: Discuss challenges faced during implementation and the corresponding solutions.
  - Individual Contributions: Explain the contributions made by each team member.
  - Summary of Achievements and Future Plans: Provide a summary of project accomplishments and outline future plans.
  - References: Include citations for external resources and documentation used.

**Notes**

- The project is a team effort, with a maximum of three members allowed in each team.
- The final exam has been scheduled for May 10 from 12:20 PM to 2:20 PM.
- During the final exam presentation, every team member is expected to participate and contribute to the live demonstration and PowerPoint presentation.
- Additionally, the final exam documents are required to be submitted on Canvas.
- As mentioned, the best project will be rewarded after the finals.

**Evaluation Criteria:**

- (35%) Completeness and functionality of the GUI-based College Management System meets all specified requirements and functions as intended.
- (35%) Applying Python programming concepts, Object-Oriented Programming (OOP) principles, and other relevant programming techniques.
  - Assessing how well these concepts are leveraged to enhance the system's functionality, efficiency, and usability.
- (20%) Evaluating the approach used for implementing web scraping functionality.
- (10%) Examining the project report and documentation for clarity, comprehensiveness, and organization to effectively convey the project's objectives, methodologies, outcomes, and future plans.
- (5%) Conducting tests and ensuring adherence to all design principles.

Good luck with your project!