College of Computing and Informatics

Computer Science Department

University of Sharjah

Fall 2025/2026

Development of a Web-Based Course Allocation System for Faculty Members

by

Mahra AlKuhoori U22105880
Dana AlSafadi U22200606
Amna AlShamsi U2114536

Asma AlAli U22100601

Supervised by

Isam Mashhour Hasan Al Jawarneh

Committee Member Names

Committee Member's First Name Middle Initial Last Name

Committee Member's First Name Middle Initial Last Name

November 2025

ACKNOWLEDGEMENT

We would like to sincerely thank the University of Sharjah for giving us the academic setting, materials, and ongoing assistance that allowed us to complete this project effectively. We would also want to express our gratitude to Dr. Isam Mashhour Al Jawarneh, our supervisor, whose advice, knowledge, and helpful criticism were vital at every turn. His dedication, support, and wise guidance were crucial to the growth and improvement of our project.

This project would not have reached this level of completeness without the support and contributions of the University community to whom we express our deepest gratitude.

UNDERTAKING

This is to declare that the project entitled “Development of a Web-Based Course Allocation System for Faculty Members” is an original work done by undersigned, in partial fulfillment of the requirements for the degree “Bachelor in Computer Science and IT Multimedia” at the Computer Science Department, College of Computing and Informatics, University of Sharjah, UAE.

All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

Signed,

Dana Al Safadi

Mahra Al Kuhoori

Amna Al Shamsi

Asma Al Ali

ABSTRACT

This project provides a Web-Based Course Allocation System that aims to automate and simplify the process of assigning courses to university faculty members. The method reduces the amount of manual work for the Head of Department (HoD) while ensuring fairness, transparency, and correctness in course distribution. Faculty members can submit their teaching preferences and CVs, which are then extracted and aligned with course requirements using an AI-based system.

The system offers user dashboards for the Head of Department, faculty members, and administrators, each with its own set of features, including course offering management, preference submission, allocation approval, user administration, and database backup. UML diagrams were used, such as sequence diagrams, state diagrams, and class diagrams to describe behavioral, structural, and data elements. The implementation focuses on usability, security, scalability, and dependability.

The goal of the end outcome is to have a system which is flexible, focused on user interaction, and efficient for managing academic workloads and making intelligent course distribution decisions

Contents

CHAPTER 1: Introduction

1.1 Overview

Course allocation systems are necessary for supporting faculty members in order to reduce the workload distribution challenges they face in working life. In academic settings, Head of Departments frequently encounter issues in distributing courses fairly, along with the burden of time-consuming manual coordination. Therefore, this system provides essential tools to ensure a fair and transparent workload for faculty members, and allows them to focus on academic responsibilities rather than procedural duties.

1.2 Project Motivation

Manual course allocation is error-prone, time-consuming, and lacks transparency. Over the years, errors and frequent changes in distribution of courses have been noticed, and these mistakes have sometimes affected both students and instructors. This system addresses these difficulties by centralizing data, using automation, and incorporating AI-based matching to make more accurate and equitable decision-making solution.

1.3 Problem Statement

The current approaches of assigning courses often leads to inefficiencies, subjective decisions, and inconsistent distributions. Faculty expertise and preferences are often overlooked due to scattered information. Through the digitization and centralization of our course allocation process, the system we are proposing aims to resolve these issues and ensure a more transparent workflow.

1.4 Project Aim and Objectives

The goal of our system is to provide a user-friendly, safe, and flexible experience to faculty members.

- Develop an intelligent web-based system that automates course allocation.
- Enable faculty to submit teaching preferences and upload CVs.
- Provide automated course recommendations to HoDs.
- Allow admins to manage users and logs.

1.5 Project Scope

Modules for Head of Departments and faculty members are all included in this project. It includes functionalities that provides HoDs with tools to manage course offerings, view system-generated allocation recommendations, and approve or modify final course assignments. The faculty members are allowed to create profiles, submit their teaching preferences, and upload CVs. The system also supports the submission and processing of course change requests, as well as the generation of allocation reports, notifications, and system logs.

1.6 Project Software and Hardware Requirements

Software:

- HTML, CSS, JavaScript, MySQL

Hardware:

- PC/Laptop for development.
- Database server.
- Standard web server.

Tools:

- Git, VS Code

1.7 Project Limitations

- Limited to University of Sharjah.
- Only for HoDs and faculty members, not students.
- Web-based, no mobile applications.
- Accuracy relies on CV quality.

1.8 Project Expected Output

Our expected output for this phase is a complete system design and fully developed GUI prototypes that demonstrates how AI-assisted recommendation will function in the next implementation stage. The AI component is conceptual and will be developed in the future phase.

1.9 Project Schedule

Activities	Start Date	End Date	Deliverables
Requirements gathering	Sep 20, 2025	Oct 01, 2025	Stakeholders analysis and faculty handbook revisions
System Design & Modeling	Oct 02, 2025	Oct 15, 2025	Diagrams of use cases, activities, sequences, and classes
Interface Design	Oct 16, 2025	Nov 01, 2025	Wireframes for each role
Back-End & Front-End Implementation	Nov 02, 2025	Nov 20, 2025	Basic system features
Final Report & Presentation Preparation	Nov 21, 2025	Nov 25, 2025	Final report + Presentation slides

1.10 Project, product, and schedule risks

The project faces several possible risks that need to be considered. The complexity of modeling the system may result in time constraints, affecting the project schedule. Furthermore, user acceptance to the new system may be challenging, as effective training will be necessary to ensure the system is used correctly. There is also a risk of technical difficulties in implementation such as authentication procedures.

CHAPTER 2: Related Existing System

2.1 Introduction

Course allocation is a crucial administrative responsibility in academic institutions, requiring careful matching of faculty expertise with departmental course requirements. Over time, colleges have built a variety of technologies to aid with this process, ranging from simple spreadsheets to basic digital portals. These current systems provide a basis for collecting faculty data and documenting course assignments, but they frequently lack intelligence, automation, and structured workflows. Analyzing these systems helps identify their strengths, flaws, and gaps, which influences the development of a more efficient and transparent course allocation strategy. By examining how current systems work, it becomes clear what areas we should focus on to improve, such as AI Integration, centralized data administration, and traceable decision-making, in order to optimize faculty course assignment processes.

2.2 Existing Systems

Several systems are currently used to manage course allocation and faculty assignments in universities, though most have limitations in implementing AI-driven expertise-based assignment.

Below are some examples of systems:

- **UniTime:** It is an open-source academic scheduling system that handles course scheduling and faculty availability while resolving conflicts using constraint-solving. However, it does not evaluate faculty skills to ensure appropriate course pairing.

Existing systems offer effective scheduling and job management capabilities, but there are gaps in AI-assisted skill assessment, organized change handling, and transparent logging. The suggested Course Allocation System tries to fill these gaps.

2.3 Overall Problems of Existing Systems

The limitations of current course allocation approaches include:

- **Lack of Automation:** Manual assignment requires significant work and is prone to human error.
- **No AI-Based Expertise Assessment:** Existing systems do not automatically assess faculty qualifications or teaching expertise in order to optimize course distribution.
- **Complex Interfaces:** Most systems interfaces are complex and lack flexibility.
- **Limited Transparency:** Allocation decisions are sometimes difficult to trace, reducing accountability.

2.4 Overall Solution Approach

The system we are proposing is designed to overcome the limitations mentioned above. The solution is to combine AI-assisted faculty assessment, organize workflows, and centralize data management. AI algorithms assess faculty expertise based on qualifications, previous teaching records, and departmental requirements, and make intelligent course allocation recommendations to heads of departments. Structured workflows enable consistent change management, while centralized logging ensures openness and accountability. Backup techniques provide additional protection against data loss, making the allocation process more dependable and efficient.

The suggested solution automates and structures the allocation process, ensuring that faculty members are allocated courses that match their expertise, workloads are balanced, and HoDs can make educated, traceable decisions, all without relying on manual work.

CHAPTER 3: Requirement Engineering and Analysis

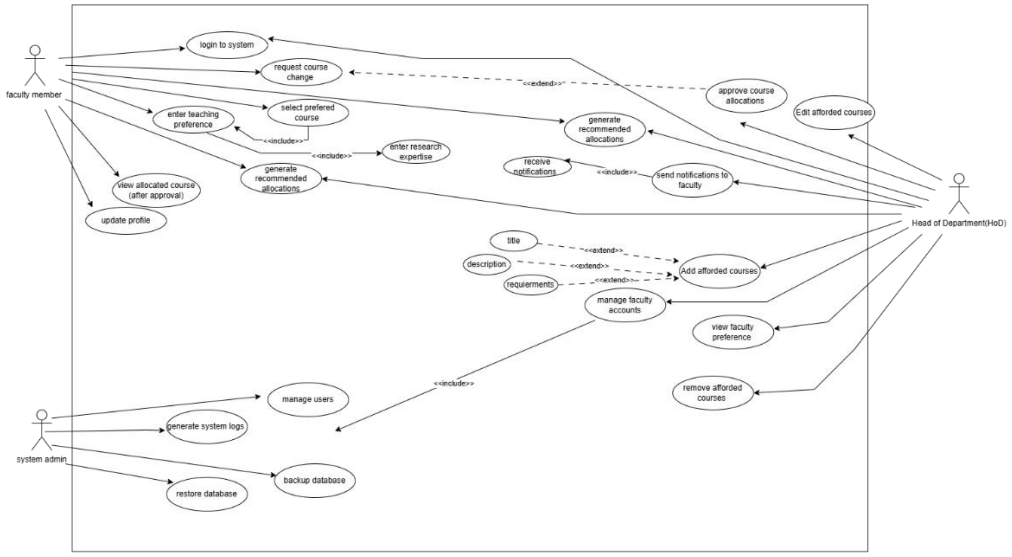
3.1 Stakeholders

- Head of Department (Primary)
- Faculty Members (Primary)
- System Administrator (Primary)
- University Management (Secondary)

3.2 Use Case Diagram

3.2.1 Use Case Section

Figure 3.2-1: Use Case Diagram



This diagram illustrates a course management system with three main actors and their interactions with the various system functions.

Actors:

1. **Faculty Member**

   The primary user with the most extensive permissions, including:

   - **Account Management:** Login to system, request course change, update profile.
   - **Course Operations:** Enter teaching preference, generate schedule and allocations, and view allocated courses (after approval).

- **Research & Faculty Development:** Enter research expertise, browse allocations, receive notifications about allocations and faculty accounts.

2. **Head of Department(s) (HoD)**
   Administrative role with approval and oversight capabilities:
   - **Approval Functions:** Approve course allocations, add allocated courses, manage faculty accounts, and view faculty conference details.
   - **Administrative Tasks:** Generate schedules, view allocations, manage course preferences, and remove allocated courses.
3. **System Admin**
   Technical administrator responsible for:
   - **System Management:** Generate system logs.
   - **Database Operations:** Restore database, backup database.
   - **User Administration:** Manage users.

Key Workflows

Course Allocation Process

The diagram shows a workflow in which faculty members enter preferences, which feeds into schedule development. The HoD then approves allocations, allowing faculty to access their assigned courses.

Notification System

Multiple notification paths connect multiple functions to keep faculty informed of approvals and system upgrades.

Database Management

This system administrator manages the technical infrastructure using backup, restore, and logging operations that run independently of the course management workflows.

3.3 Non-functional requirements

Non-Functional Requirements
1.Usability 1.1 The interface should be user-friendly and accessible on multiple devices. 1.2 The design should be responsive and intuitive.
2. Security 2.1 Secure authentication and role-based access control. 2.2 Data encryption for sensitive information.
3. Reliability 3.1 Regular data backups. 3.2 High uptime and fault tolerance.

3.4 Constraints

The following constraints establish the bounds and limitations of the system's development and deployment:

- **Web-based Only:** The initial version of the system will not include any separate desktop or mobile applications, due to time and resource limitations. The system will only be accessed via web browsers on desktop and mobile devices.

- **AI Requires Internet Connection:** Any AI-driven features such as automated courses recommendations or instructor-course matching require a stable internet connection. The system will not work offline, especially when communicating with external APIs or cloud-based AI services.

- **Limited Development Time:** The project is being developed throughout an academic year. Limiting the breadth of advanced features. Only key modules, such as login, course assignment, instructor profiles, and allocation logic, will be fully built in the first version.

- **Resource Constraints:** The system will initially focus on core functionalities only. Advanced features will then be considered in future stages.

CHAPTER 4: Architecture and Design

4.1 Overview

The Web-Based Course Allocation System's architecture is multi-tiered, which ensures scalability, adaptability, and safe interaction between system components. The architecture includes a web-based user interface, a layered application logic tier, a database management tier, and an external AI processing module that handles CV-based expertise extraction.

To clearly show the system's design, this chapter gives the logical, process, and physical viewpoints, followed by detailed descriptions of the primary software components. This chapter also contains diagrams like sequence diagrams, class diagrams, ER diagrams, and state transition diagrams

4.2 Software design

4.2.1 UML sequence/communication diagram

1. **Course Allocation Process (Sequence Diagram):** This diagram represents the entire workflow for course allocation, from initial setup to approval and notification. The dashed lines show the return messages from the system, showing the asynchronous nature of certain processes like displaying results.

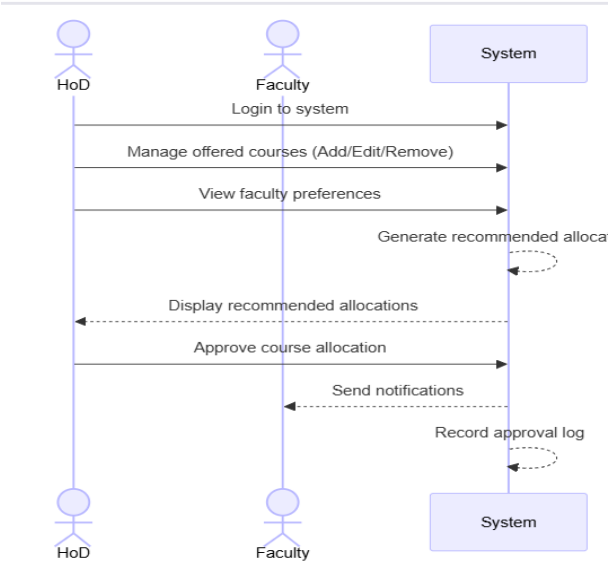


Figure 4.3.1-1 Course Allocation Process Diagram

Process Flow

Phase1: Setup & Configuration

- Both the HoD and the faculty log in to the system.
- The HoD manages the course catalog by adding, updating, or removing offered courses.
- The HoD considers faculty preferences that have been submitted.

Phase2: Allocation Generation

- The system creates recommended course allocations based on faculty preferences and available courses.
- Recommendations are displayed back to the HoD for review.

Phase3: Approval & Recording

- HoD reviews and approves course allocations.
- The system delivers messages to appropriate faculty members on their authorized courses.
- The System logs the permission for auditing and tracking purposes.

2. **Faculty Preference Submission Process (Sequence Diagram):** This figure shows the way faculty members submit their teaching preferences into the system.

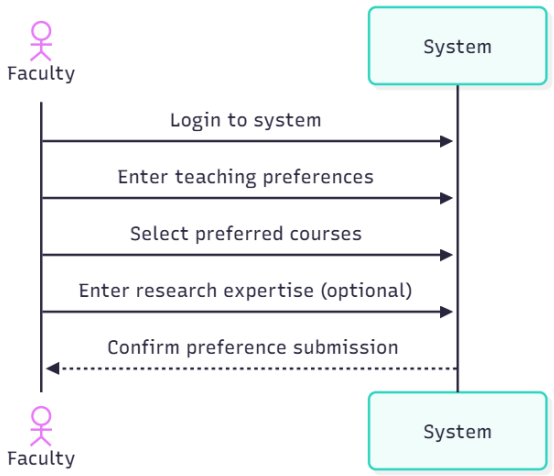


Figure 4.3.1-2 Faculty Preference Submission Diagram

Process Flow

Step-by-Step Submission

- **Login:** Faculty member authenticates with the system.
- **Enter Teaching Preferences:** Faculty defines their general teaching availability and limitations.
- **Select Preferred Courses:** Faculty chooses certain courses they want to teach.
- **Enter Research Expertise (optional):** Faculty can contribute information about their research topics to aid in course matching.
- **Confirmation:** System validates the successful submission of preferences.

3. **System Administration Process (Sequence Diagram):** This figure shows the administrative and maintenance responsibilities carried out by system administrators. It is crucial for both system maintenance and security.

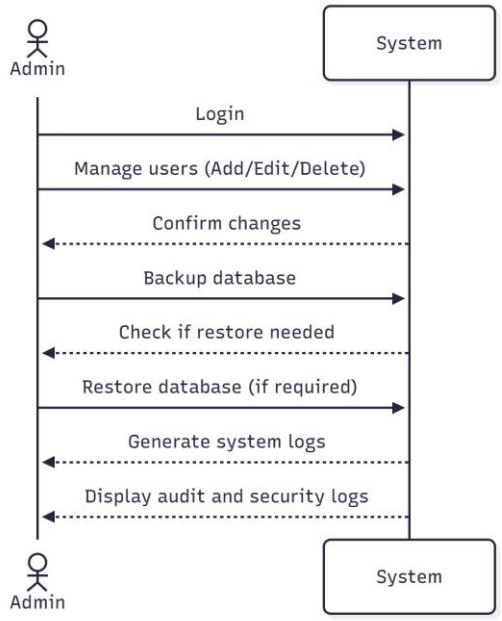


Figure 4.3.1-3 System Administration

Process Flow

- **User Management:** Admin logs into the system, and performs user administration actions (add, edit, and delete users). The system then confirms the changes applied.
- **Database Maintenance:** The Admin starts database backup; system then checks if restoration is needed.
- **Audit & Monitoring:** Admin requests system log generation. The system generates logs that contain audit and security information. The admin reads these logs to monitor system health and security.

4. Course Change Request Process (Sequence Diagram): This Figure shows the workflow faculty members and HoDs, when faculty requests a change to their course allocation after the initial assignment.

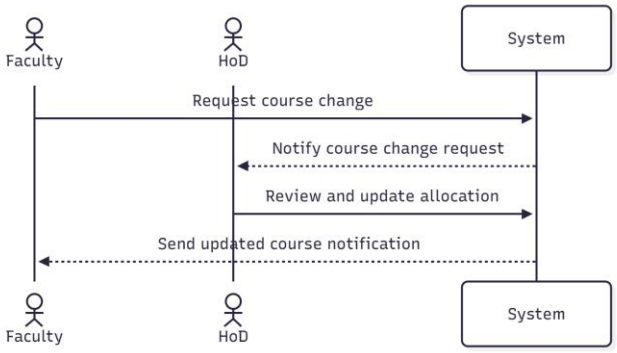


Figure 4.3.1-4 Course Change Request

Process Flow

- **Initiation:** Faculty sends a request for course changes using the system.
- **Notification & Review:** The system notifies the HoD of the pending course change request. The HoD then reviews the request and updates the allocation accordingly.
- **Confirmation:** The system sends an updated course notification back to the faculty member. Faculty receives confirmation for their modified course.

4.2.2 Class diagram

This diagram shows the system's object-oriented structure, with User as the abstract base class, which is inherited by Faculty, Department (HoD), and SystemAdmin.

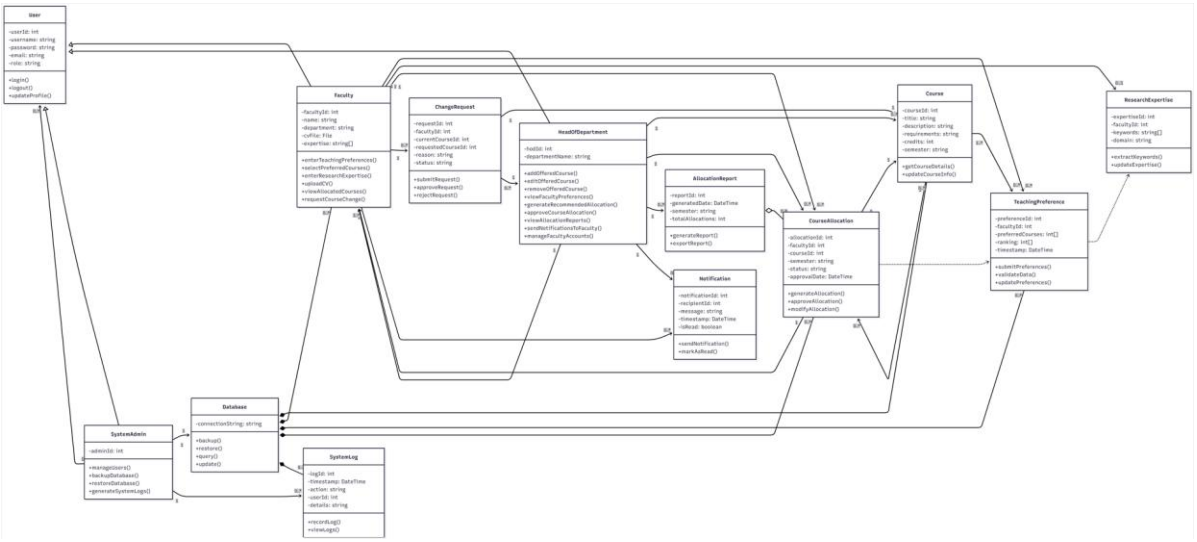


Figure 4.3.2-1 Class Diagram

Key Classes:

- **Faculty:** Manages teaching preferences, reviews, allocations, and requests changes.
- **Department:** Approves allocations, manages courses and faculty.
- **Course:** Stores course information (ID, name, credits, semester).
- **TeachingPreference:** Identifies faculty course preferences and research focus
- **CourseAllocation:** Connects faculty to courses with approval status.
- **Notification:** Sends messages regarding allocations and changes.
- **Systemlogs:** Record all system activities for auditing.

Relationships:

- Faculty members belong to Departments, create TeachingPreferences for Courses, receive CourseAllocations approved by the HoD, and receive notifications. Systemlogs are managed by system admins.

4.2.3 ER diagram

This ER diagram shows the database structure for the course allocation system, including entities and relationships. It ensures complete allocation lifecycle management with data integrity.

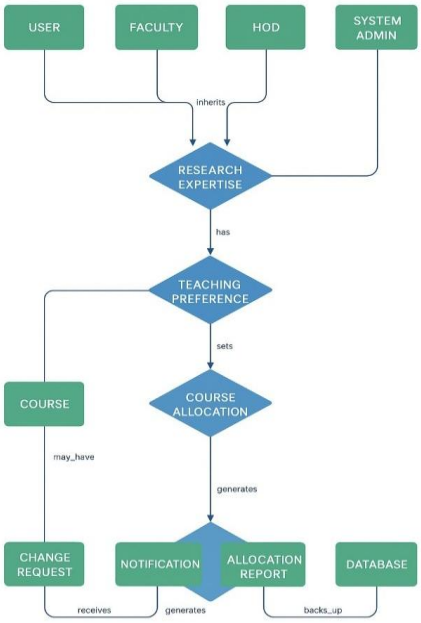


Figure 4.3.3-1 ER Diagram

Key Components

- **User Hierarchy:** USER which is the base entity is inherited by FACULTY, HOD, and SYSTEM ADMIN, implementing role-based access.

Main Relationships

- **RESEARCH EXPERTISE → TEACHING PREFERENCE ("has")**
  Links faculty research areas to course preferences
- **TEACHING PREFERENCE → COURSE ALLOCATION ("sets")**
  Preferences determine course assignments
- **COURSE → CHANGE REQUEST ("may_have")**
  Allows post-allocation changes
- **COURSE ALLOCATION**
  The central entity that generates the notification and the allocation report
- **CHANGE REQUEST → NOTIFICATION ("receives")**
  Alerts for approvals
- **ALLOCATION REPORT → DATABASE ("back_ups")**
  Maintains data persistence

4.2.4 State transition diagram

1. **Faculty Authentication and Profile Maintaining:** The user logs in, views the dashboard, edits their profile, saves the changes, returns to the dashboard, and logs out. Error states: Invalid credentials or missing data lead to an error screen with a retry option.

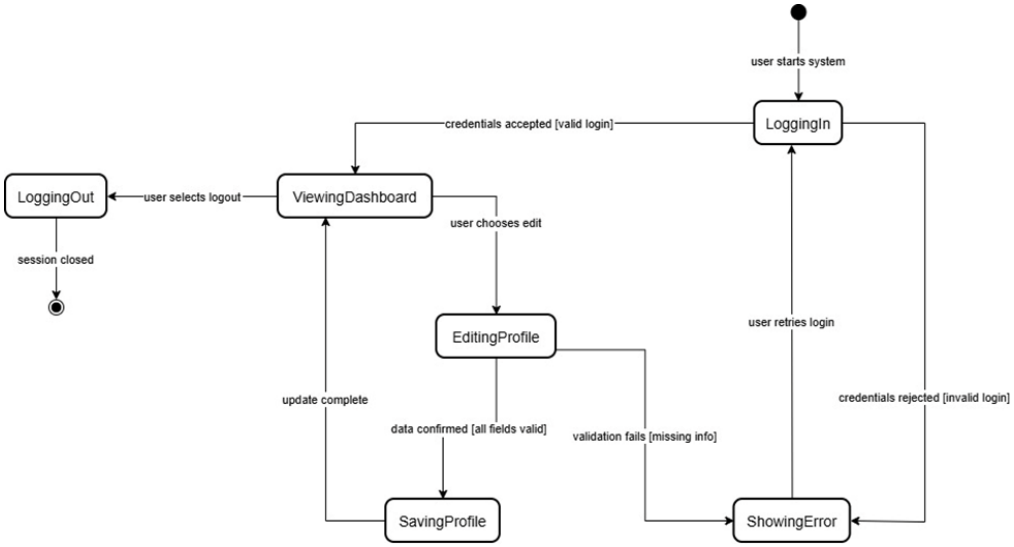


Figure 4.3.4-1 Faculty authentication and profile maintaining

2. **Faculty Preference Submitting:** The user can view preferences, update them, upload a CV for auto-extraction, review the data, submit it, and receive confirmation. Alternative: Manual entry without CV uploading.

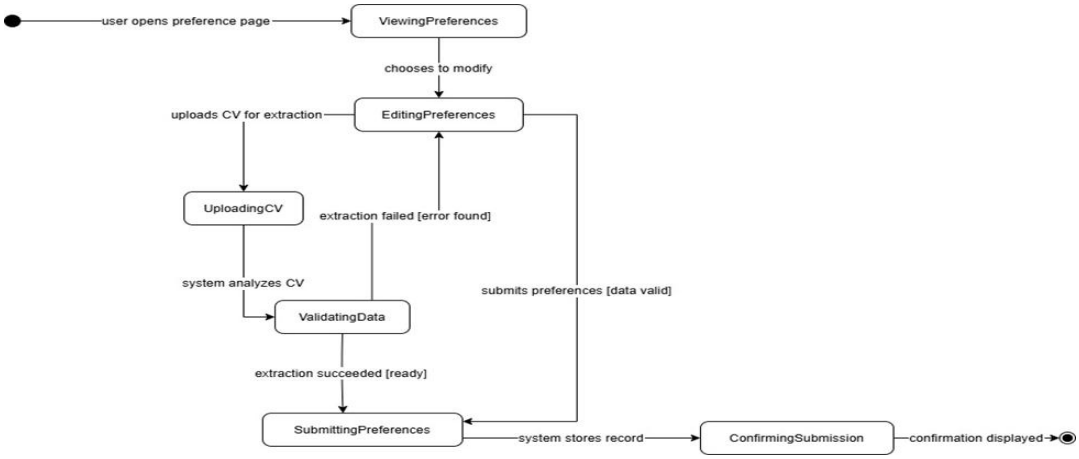


Figure 4.3.4-2 Faculty Preference Submitting

3. **Course Changing Request (faculty):** Faculty selects a course, submits a change request, and awaits the HoD's decision.

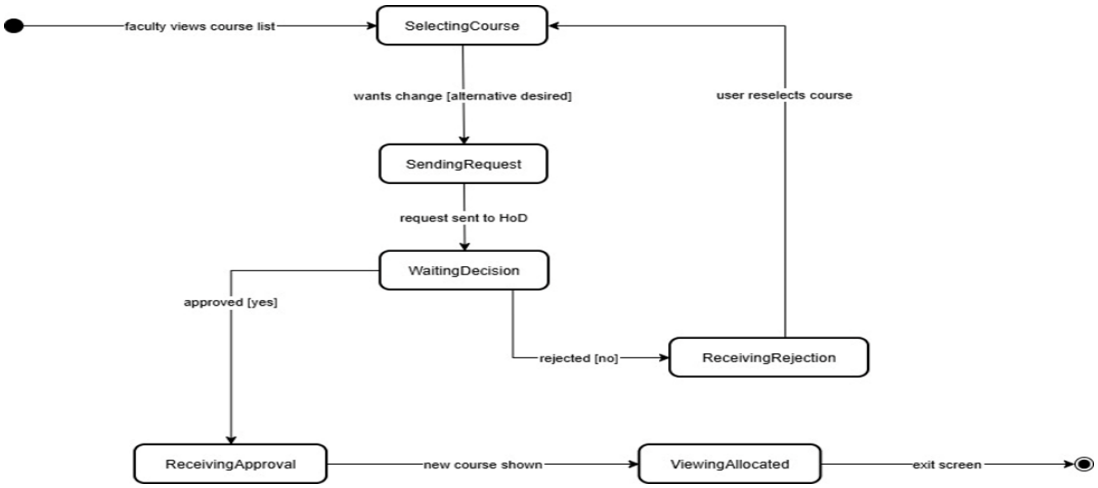


Figure 4.3.4-3 Course Changing Request

4. **Allocation Generating & Approving (System & HoD):** System gathers preferences, generates suggestions, discovers disagreements, resolves issues, reviews findings, receives HoD approval, displays allocations, notifies faculty, and archives records.

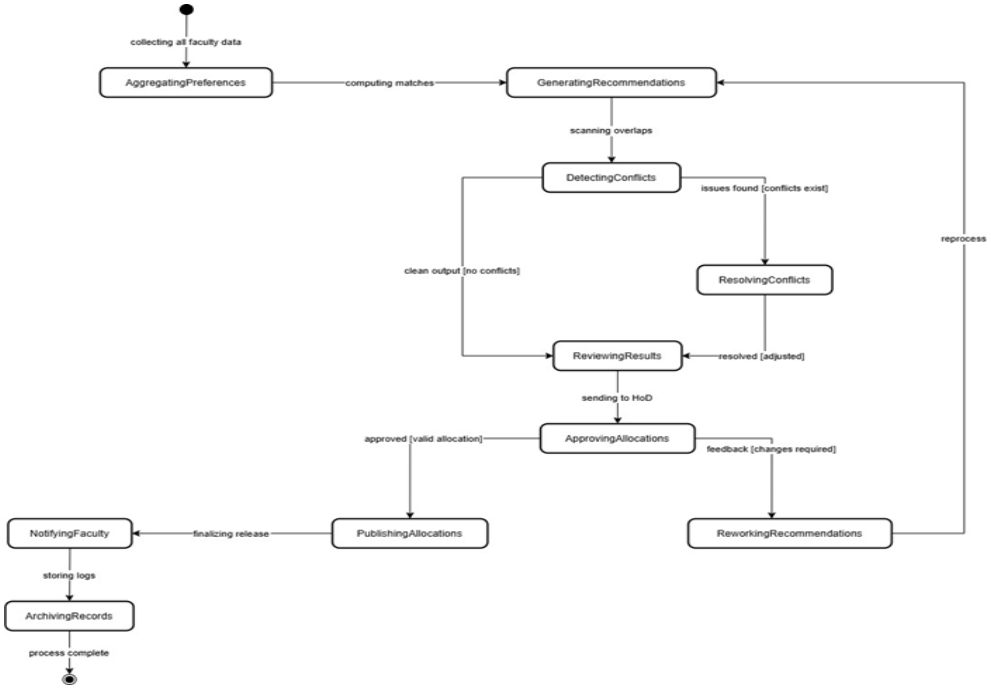


Figure 4.3.4-4 Allocation generating and approving

5. **Course Offering Managing (HoD):** View courses and select Add (Input details, validate, save), Edit (edit, validate), or Remove (confirm deletion). All actions update the course list and return to the viewing mode.

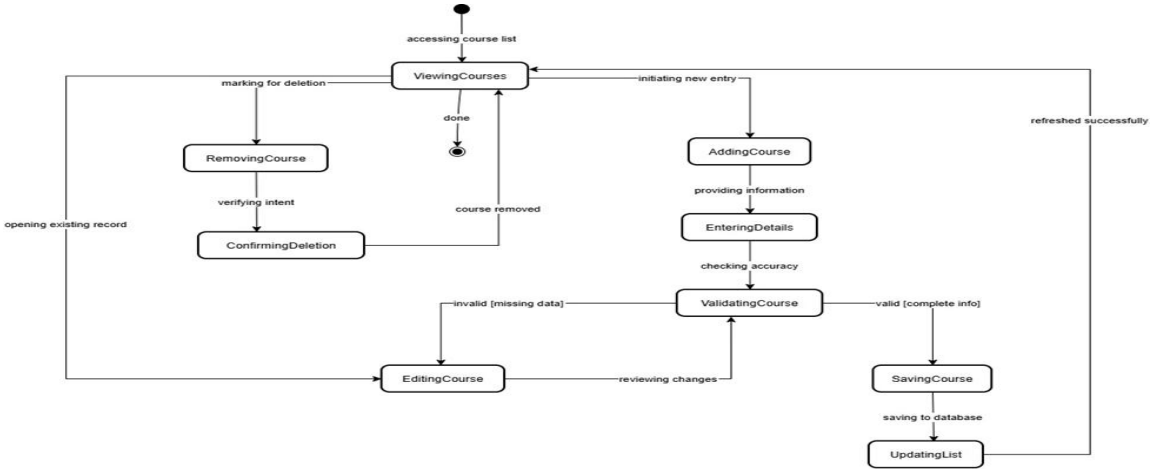


Figure 4.3.4-5 Course Offering Managing

6. **Admin Backup:** Admin schedules backups, performs them, compresses the data, confirms their integrity, and logs the activity. Error handling: Failed verification results in an automated retry.

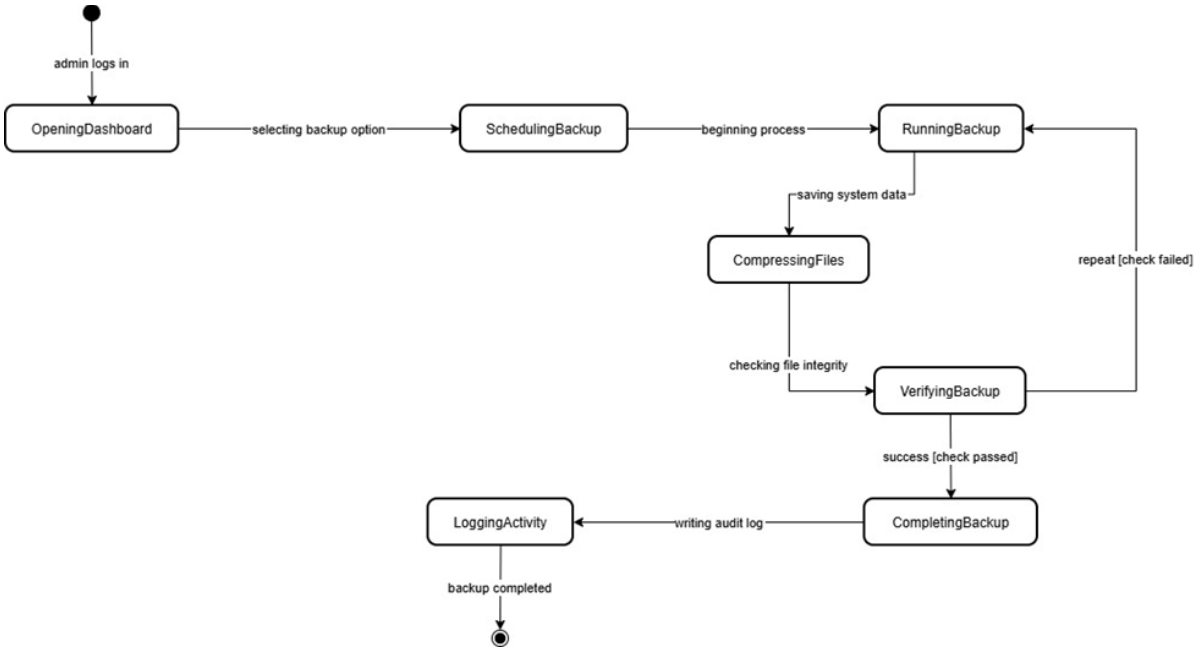
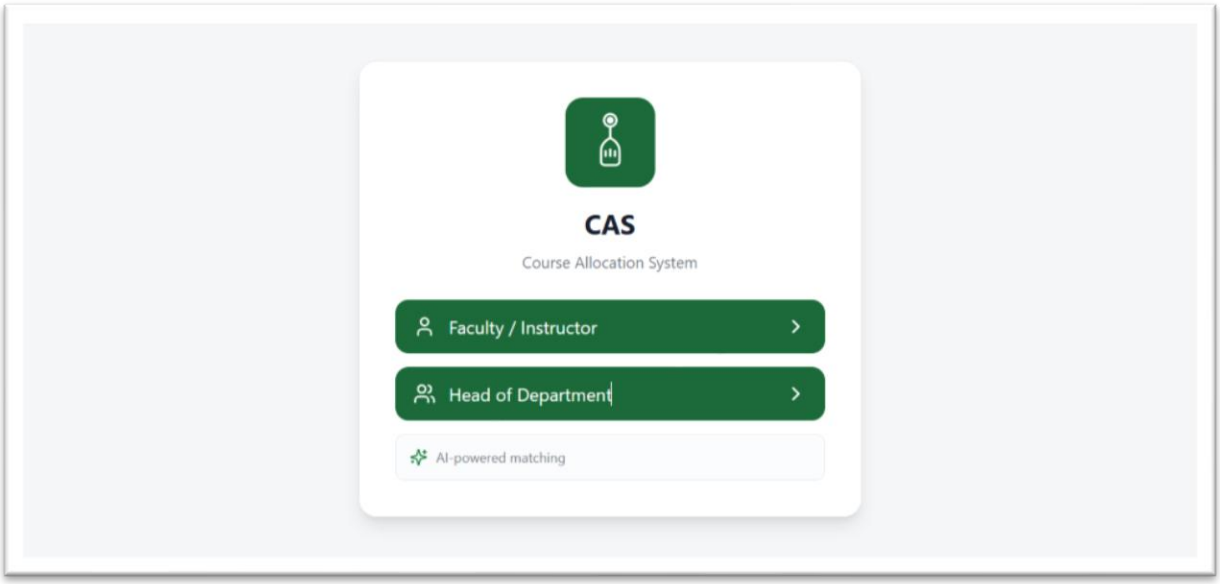


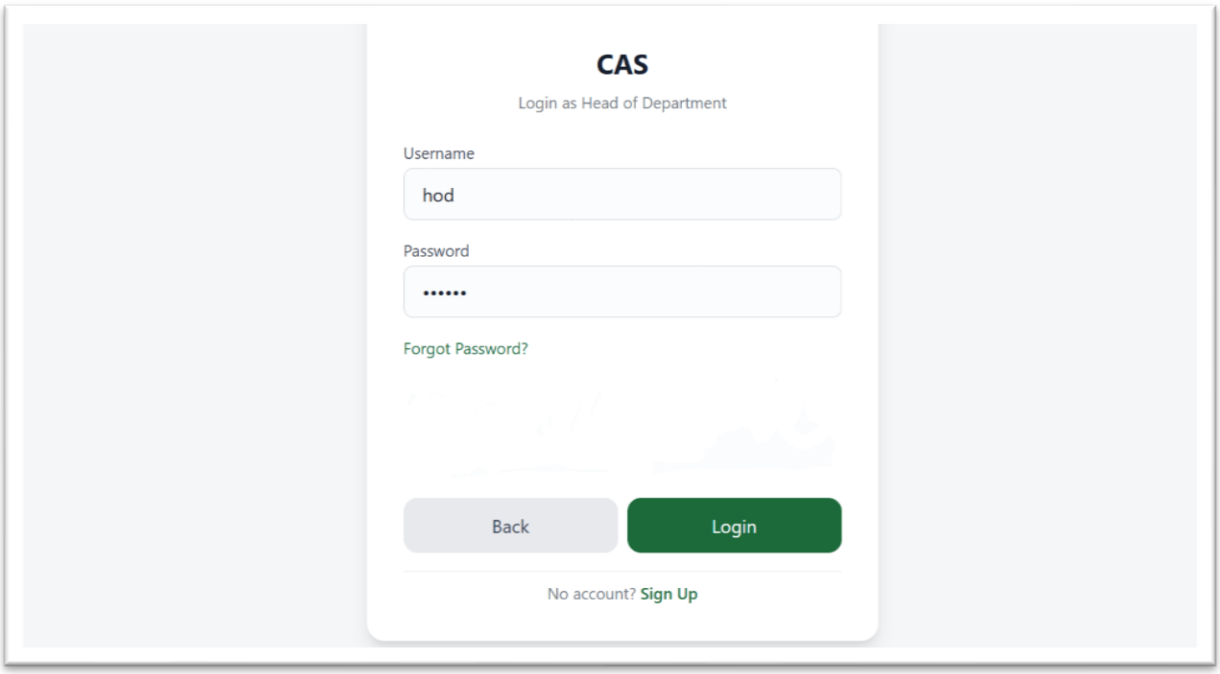
Figure 4.3.4-6 Admin Backup

4.3 User interface design

1. Landing Page & Login Page

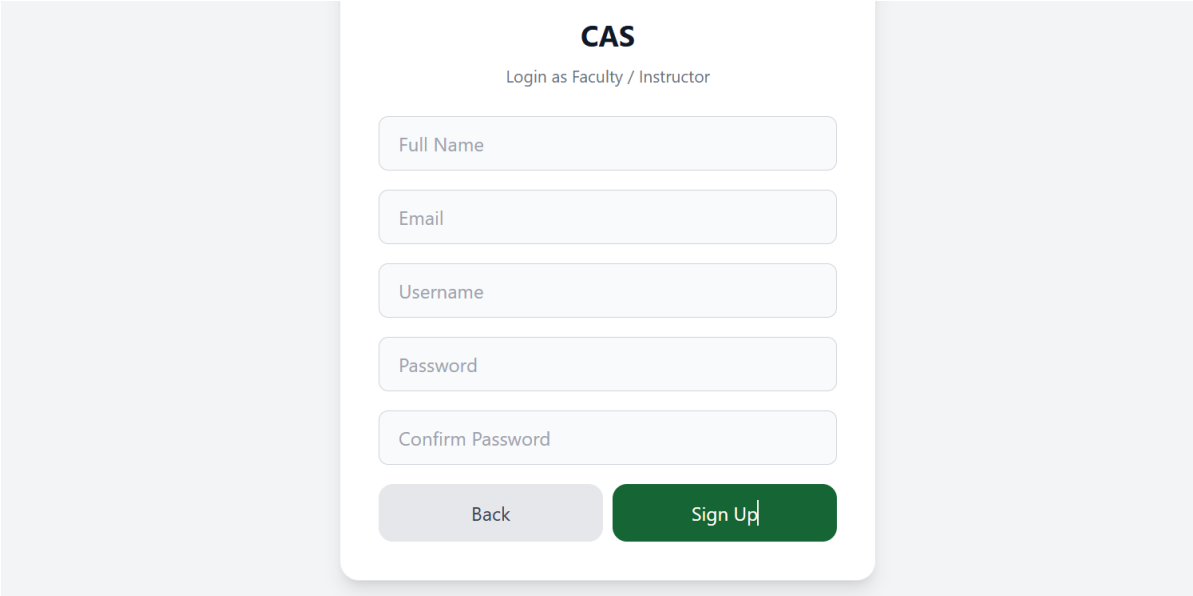


- Main entry screen displays two role options: Faculty/Instructor & HoD. The logo is displayed and the AI feature is highlighted.

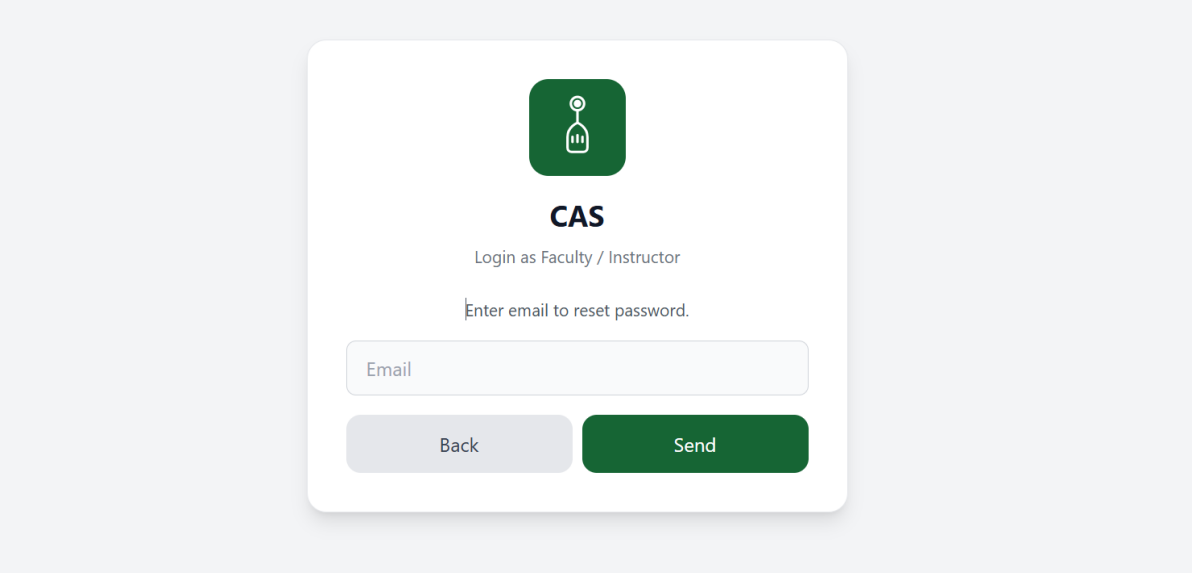


- Login interface featuring username/password boxes. “Forgot Password?” link, and “Sign Up” option. Includes Back and login buttons for navigation.

2. Sign Up Page & Password Reset Page

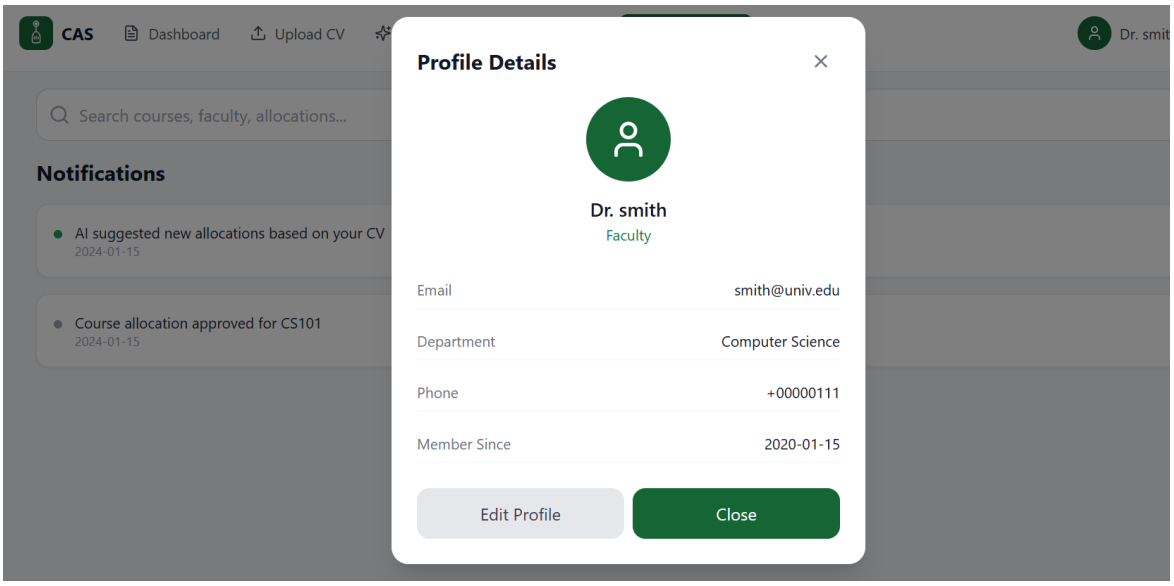


- The registration form for new users includes fields for Full Name, Email, Username, Password. Back and Sign up buttons for navigation.

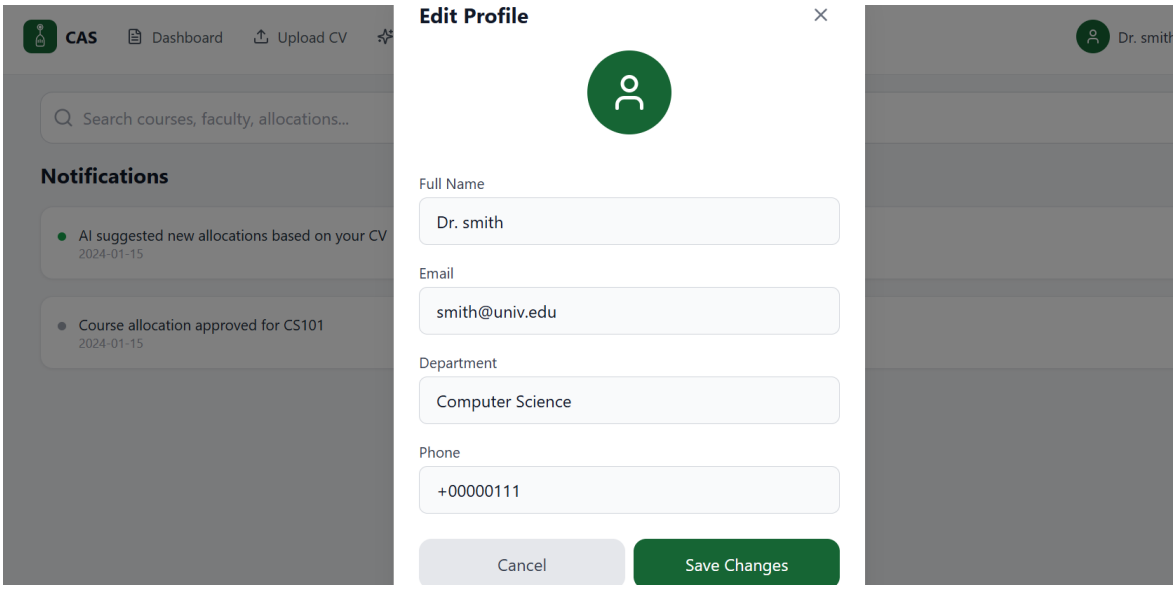


- The password recovery interface allows users to input their email address to obtain reset instructions. Back and Submit buttons for navigation.

3. Profile Details Panel & Edit Profile Panel

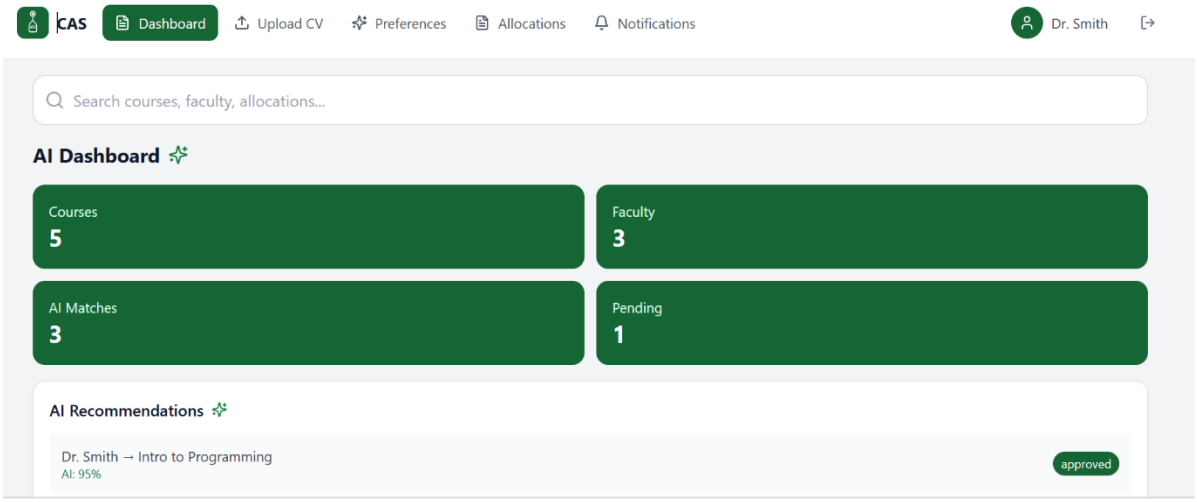


- This panel displays profile details of either HoD or Faculty. Information such as email, department and phone, Membership period. Also, Edit Profile and Close buttons for navigation.



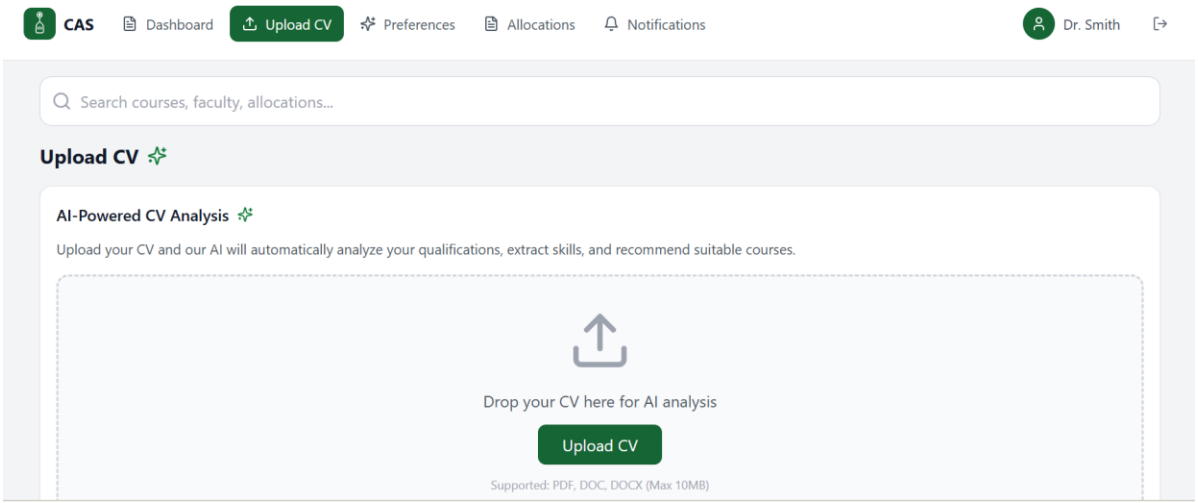
- This panel displays the options for editing the profile. The user can edit the Full Name or Email in case of any required changes. Cancel and Save Changes buttons for navigation. (Improvements such as security verifications for certain changes, will be implemented in next stages)

4. Faculty Main Dashboard



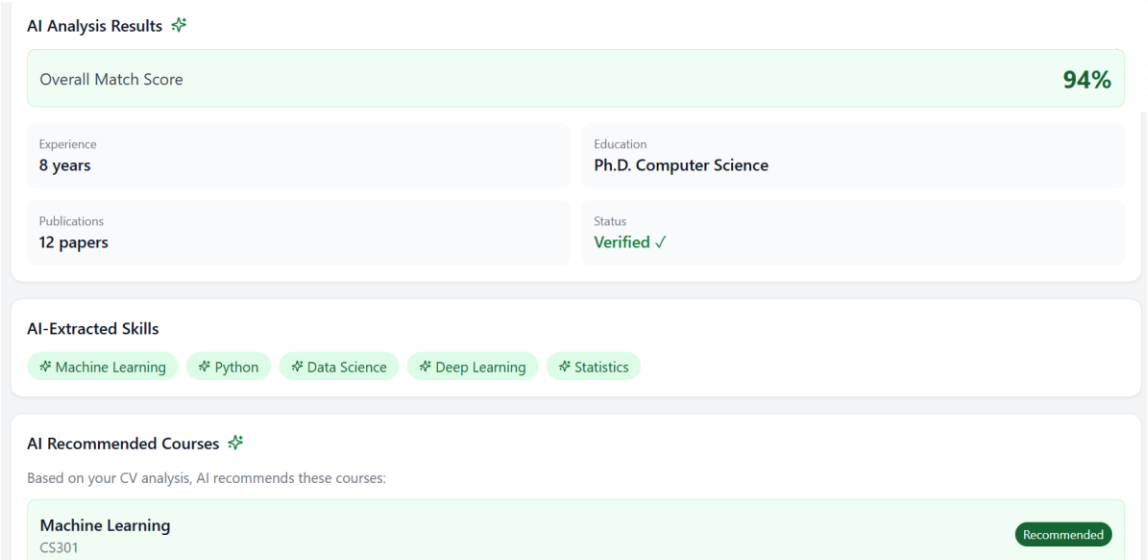
- Main dashboard for faculty displaying key metrics such as Courses, and AI Matches. Top navigation which includes options the user can navigate to.

5. Upload CV Page (Faculty)



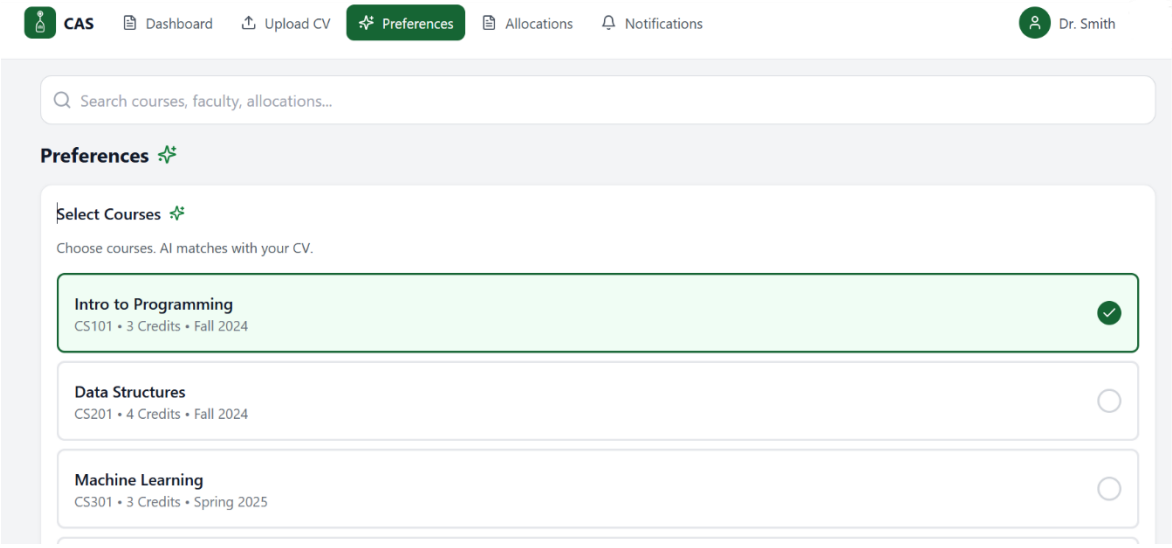
- This page allows the user to upload their CV to receive the AI-powered analysis. The system saves extracted information and update it in any case.

6. AI Analysis Page (Faculty)



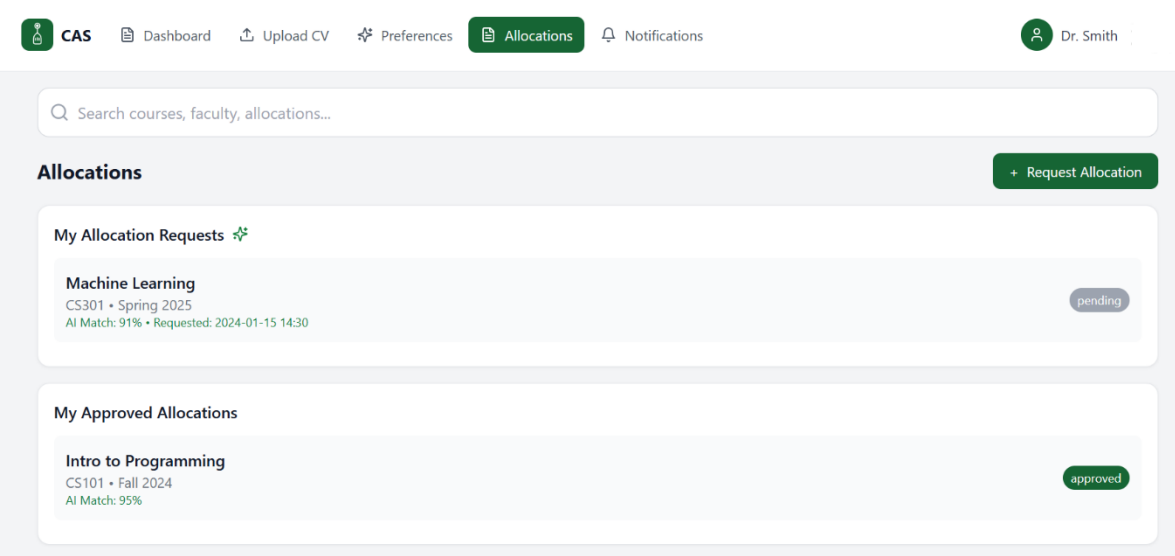
- This page displays AI analysis outcome along with extracted information such as AI Recommended Courses, and Skills.

7. Preferences Page (Faculty)



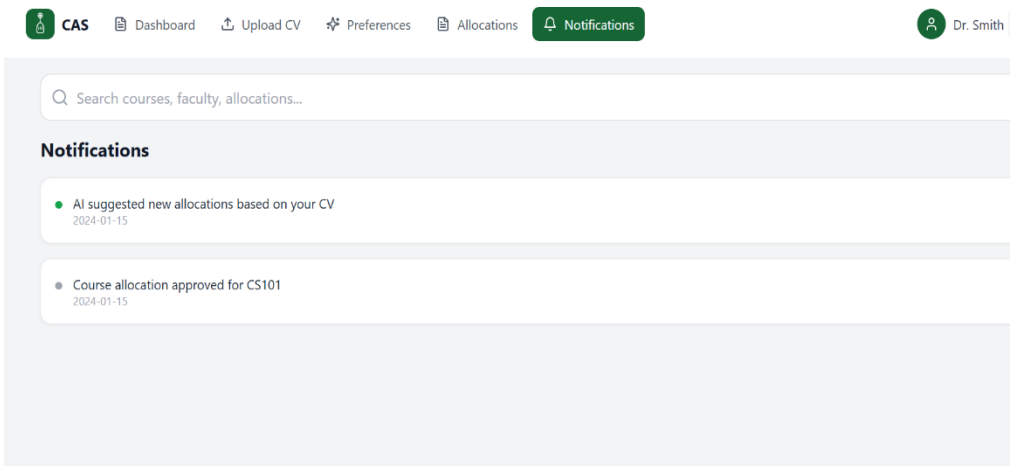
- In this page, faculty can select courses they prefer based on AI-CV Analysis.

8. Allocations Page (Faculty)



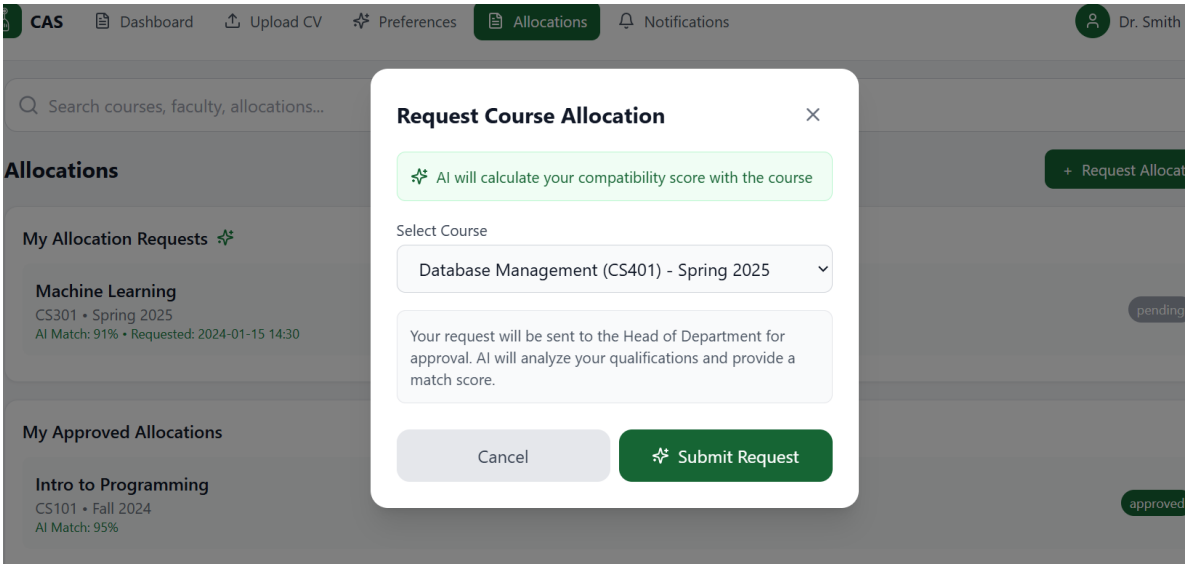
- This page shows two sections: My Allocation Requests and My Approved Allocations.
- Allocation requests displays the pending requests to be then approved by HoD.
- Approved allocations display requests approved by HoD.
- Top right button for extra requests.

9. Notifications Page (Faculty)



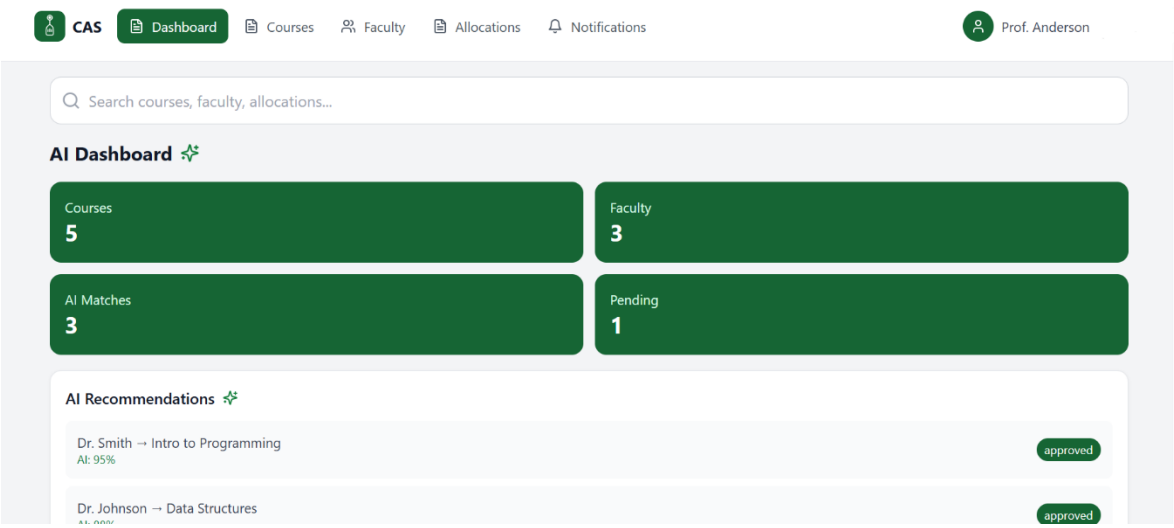
- Lists system notifications with timestamps.

10. Request Course Allocation Panel (Faculty)



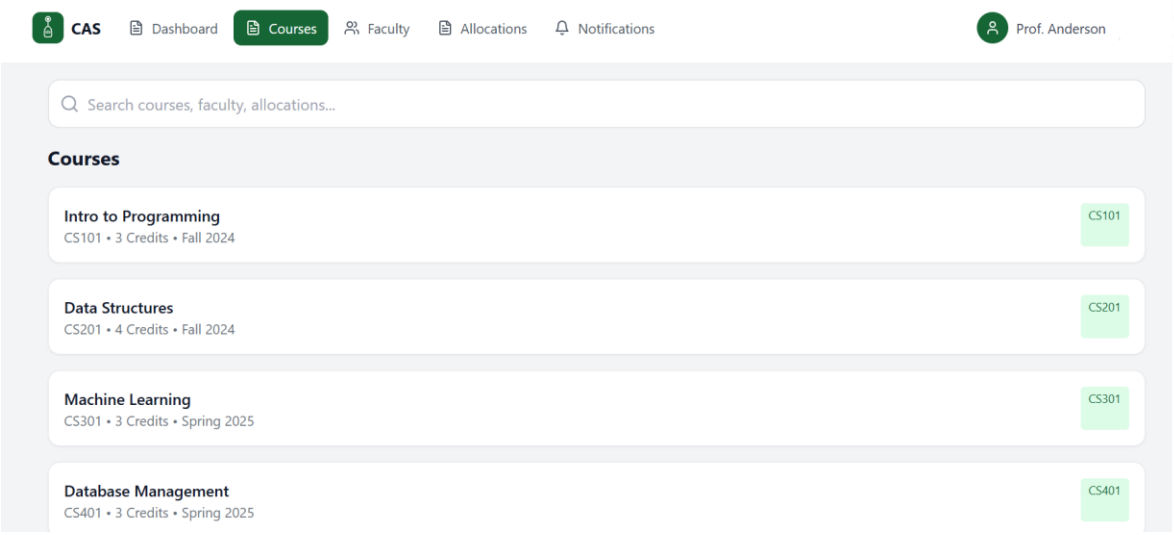
- Panel for submitting new allocation requests. Dropdown for displaying courses.

11. Main Dashboard (HoD)



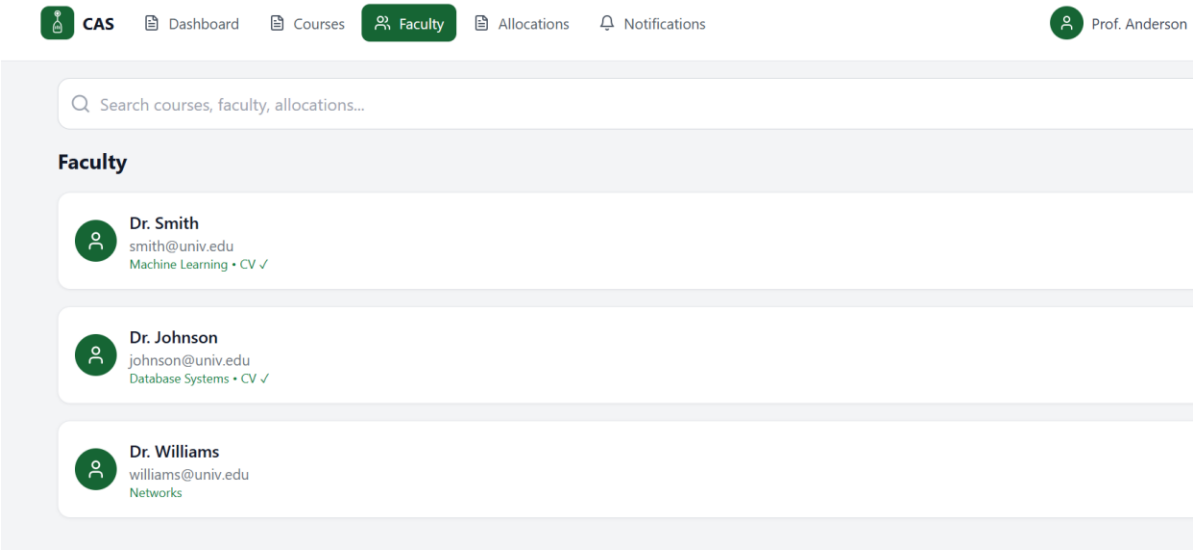
- This page shows the main dashboard for the HoD along with AI Recommendations of courses and matching instructors + status.

12. Courses Page (HoD)



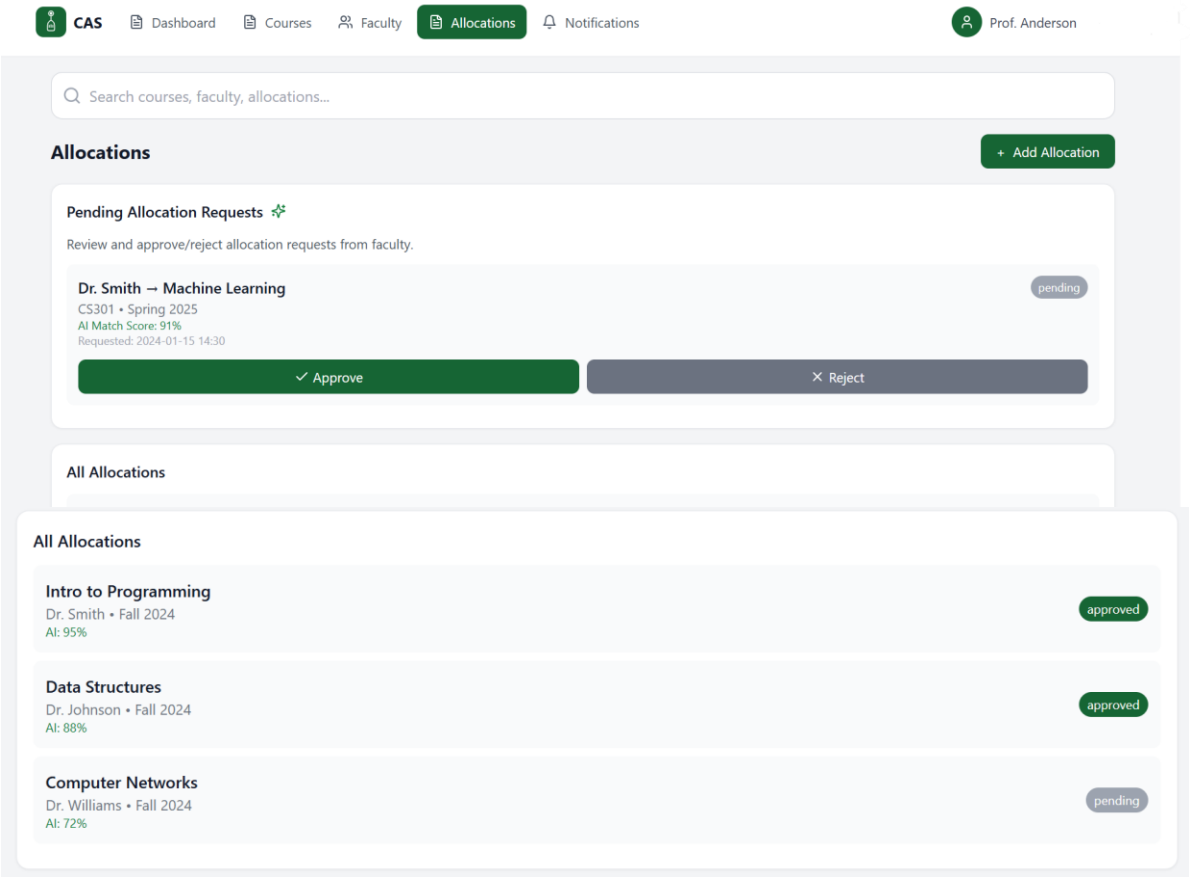
- This page lists all available courses with details.

13. Faculty Page (HoD)



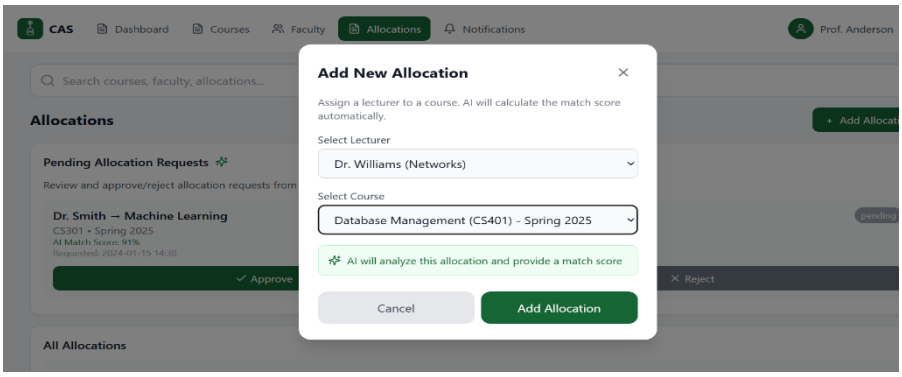
- This page displays faculty members showing name, email, expertise area, and CV verification status.

14. Allocations Page (HoD)



- This page displays two sections: Pending Allocation Requests and All Allocations.
- Allocation requests shows requests awaiting approval and All allocations lists completed allocations.
- Top right button for manual assignments.

15. Add New Allocation Panel (HoD)



- Panel for creating course allocations. Dropdowns for selecting instructors and course.

CHAPTER 5: Implementation Plan

5.1 Description of Implementation

In this Phase, the implementation focuses on creating the system's graphical user interfaces and laying the groundwork for the Course Allocation System. The purpose is to graphically demonstrate the intended features and user processes rather than to develop a fully functional system at this time.

Using HTML, CSS, and basic JavaScript, the team produced high-fidelity interface prototypes for all system users, including faculty, HoDs, and system admins. These prototypes mimic navigation, and form layouts, allowing the system's design and functions to be demonstrated.

Completed implementation tasks include:

- Designing all main GUI screens for Faculty, HoD, and Admin.
- Creating static HTML/CSS layouts and basic JavaScript interactions.
- Producing UML diagrams (Sequence, state, class) to model system behavior.
- Designing the MySQL database schema and relationships.

5.2 Programming language and technology

The following languages and technologies will be used in the implementation phase:

- Front-End: HTML5, CSS, JavaScript
- Tools for development: Visual Studio Code (VS Code)
- Database: MySQL
- GitHub
- Testing: Web browsers (Edge, Chrome, etc.)

CHAPTER 6: Experiments

6.1 Black-box

Case ID	Test Scenario	Input	Expected Output	Status
BB-01	Login with valid credentials	Valid email + correct password	User is redirected to dashboard	Pass/Fail
BB-02	Login with invalid credentials	Wrong email/password	System displays “Invalid login credentials”	Pass/Fail
BB-03	Faculty preference submission	Complete preference form	System accepts & saves submission	Pass/Fail
BB-04	HoD course approval	HoD clicks “Approve allocation”	Allocation status has been updated to approved	Pass/Fail
BB-05	Course change request	Faculty submits a change request	System forwards request to HoD for review	Pass/Fail
BB-06	Notification delivery	Trigger event	Faculty receives instant notification	Pass/Fail
BB-07	All allocation accuracy	Faculty CV + course requirements	System generates suggested matches	Pass/Fail

Table 6.1-1 Black-Box Testing

6.2 Testing automation

In this phase, we used testing automation mainly to check how the interfaces we created behaved, as we had not yet implemented the backend part. Our goal was to ensure that the prototypes we designed respond correctly when users interact with them, rather than having to manually perform the same operations each time.

To do this, we relied on simple automation technologies such as, Selenium to simulate basic user interactions. These automated scripts allow us to test things like opening different pages, clicking buttons, and submitting requests.

This helped us ensure that the navigation flow in our system is fluid, reliable, and efficient.

CHAPTER 7: Conclusion and Results

In this project, we concentrated on designing the structure of our Web-Based Course Allocation System and preparing the resources required for complete development in the upcoming phase. We identified the main issues with the present manual allocation method and turned them into clear system requirements and workflow.

During this phase, we created the necessary system diagrams, designed the MySQL database schema, and created GUI prototypes for each user role. These designs helped us understand how the system would function and ensured that the entire architecture was well-planned prior to deployment.

Overall, this stage enabled us to establish a solid basis for the implementation stage of the project.

CHAPTER 8: References

[1] Müller, T., Kopecký, J., & Vlk, M. (2020). *UniTime: Open-source university timetabling system.* Purdue University. https://www.unitime.org/

[2] Alhassan, M., & Anwar, M. (2020). Automated faculty-course assignment system using rule-based matching. *International Journal of Advanced Computer Science,* 11(5), 112–121.

[3] Sharma, S., & Katarya, R. (2021). AI-driven decision-making in academic course allocation. *Journal of Educational Technology Systems,* 50(1), 45–64.

[4] Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.

[5] Pressman, R. S., & Maxim, B. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill.

[6] Wiegers, K., & Beatty, J. (2013). *Software requirements* (3rd ed.). Microsoft Press.

[7] Ambler, S. (2005). *The elements of UML 2.0 style.* Cambridge University Press.

[8] Fowler, M. (2004). *UML distilled: A brief guide to the standard object modeling language* (3rd ed.). Addison-Wesley.

[9] W3C. (2024). *HTML5 and web standards documentation.* https://www.w3.org/

[10] https://api.sharjah.ac.ae/Flipbooks/MediaPublicationsDocument/FacGuide/Faculty_Handbook/4/index.html

[11] SeleniumHQ. (2023). Selenium Documentation. Retrieved from https://www.selenium.dev/documentation