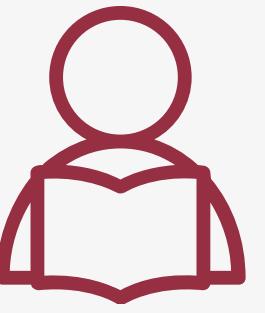


Academic and Administrative Website

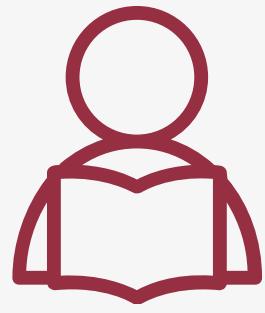


A Project Presentation
CSEN 275-Object Oriented Analysis,
Design & Programming

TEAM MEMBERS



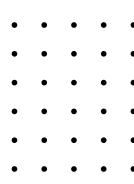
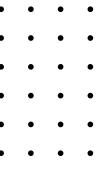
Kinsu Patel



Tanay
Mayee



Ruthu
Rajendra



Overview

Our project involves the development of an end-to-end Academic and Administrative website (AAW), designed to streamline the educational experience for students, faculty, and administrators. The AAW provides a centralized platform where users can manage courses, assignments, grades, and announcements, fostering an interactive and organized learning environment. The system is akin to platform like MySCU Portal, ensuring ease of use and comprehensive functionality for all user roles.



Project Scope and Importance of AAW in educational setting



Students can access course materials, submit assignments, take quizzes, and view grades in a structured and organized manner, enhancing their learning experience.



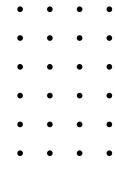
Faculty members can easily manage course content, track student progress, and communicate with students, reducing administrative workload and allowing them to focus on teaching.



Administrators can efficiently oversee course assignments, track faculty and student activities, and ensure the smooth operation of the educational institution.



The AAW can be accessed from anywhere, at any time, providing flexibility for both students and faculty to engage with the course material according to their schedules.



(Continued ...)

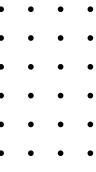
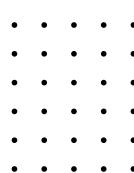


An AAW provides a single platform for managing all aspects of the educational process, from course content delivery to grading and communication.

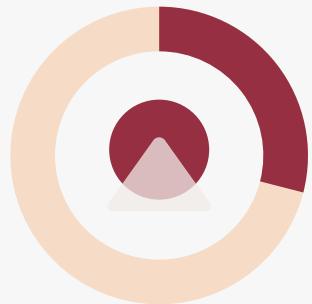


The AAW can generate reports and analytics, providing valuable insights into student performance and engagement, aiding in informed decision-making.

Tech-Stack Used: Next.js, HTML, Tailwind CSS, Flask, MySQL

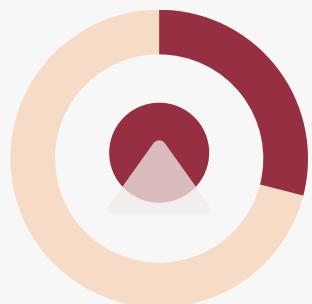


Faculty functionality



- View Homepage with list of courses (current and previous semesters)
- Manage courses (published/unpublished)
- Add content to syllabus
- View student lists and grades
- Assign grades
- Add assignments and quizzes
- Post announcements

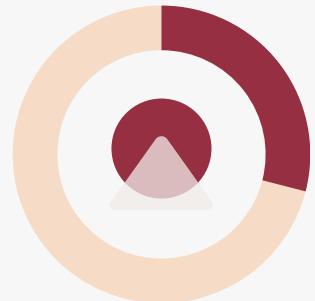
Admin functionality



- Admin Functionality
- View courses by Faculty by semester
- Assign courses to Faculty for new semesters
- View student lists (without grades)



Student functionality



- View enrolled courses (current and previous semesters)
- Access course content if published
- View quizzes and assignments
- View personal grades
- Set profile information and notifications



OOP Principles



Encapsulation

Course Class: Contains attributes like course ID, name, syllabus, and methods to add content, view student list, and assign grades. These details are hidden from other parts of the application to prevent unauthorized access or modifications.



Polymorphism

ViewCourses Method: Both Faculty and Student classes can have a viewCourses method, but the implementation differs based on the role. For Faculty, it shows courses they teach; for Students, it shows courses they are enrolled in.



Inheritance

- User Class: A base class with common attributes like user ID, name, email, and methods for authentication.
- Derived Classes: Faculty, Student, and Admin classes inherit from User and add specific attributes and methods relevant to their roles.

⋮⋮⋮

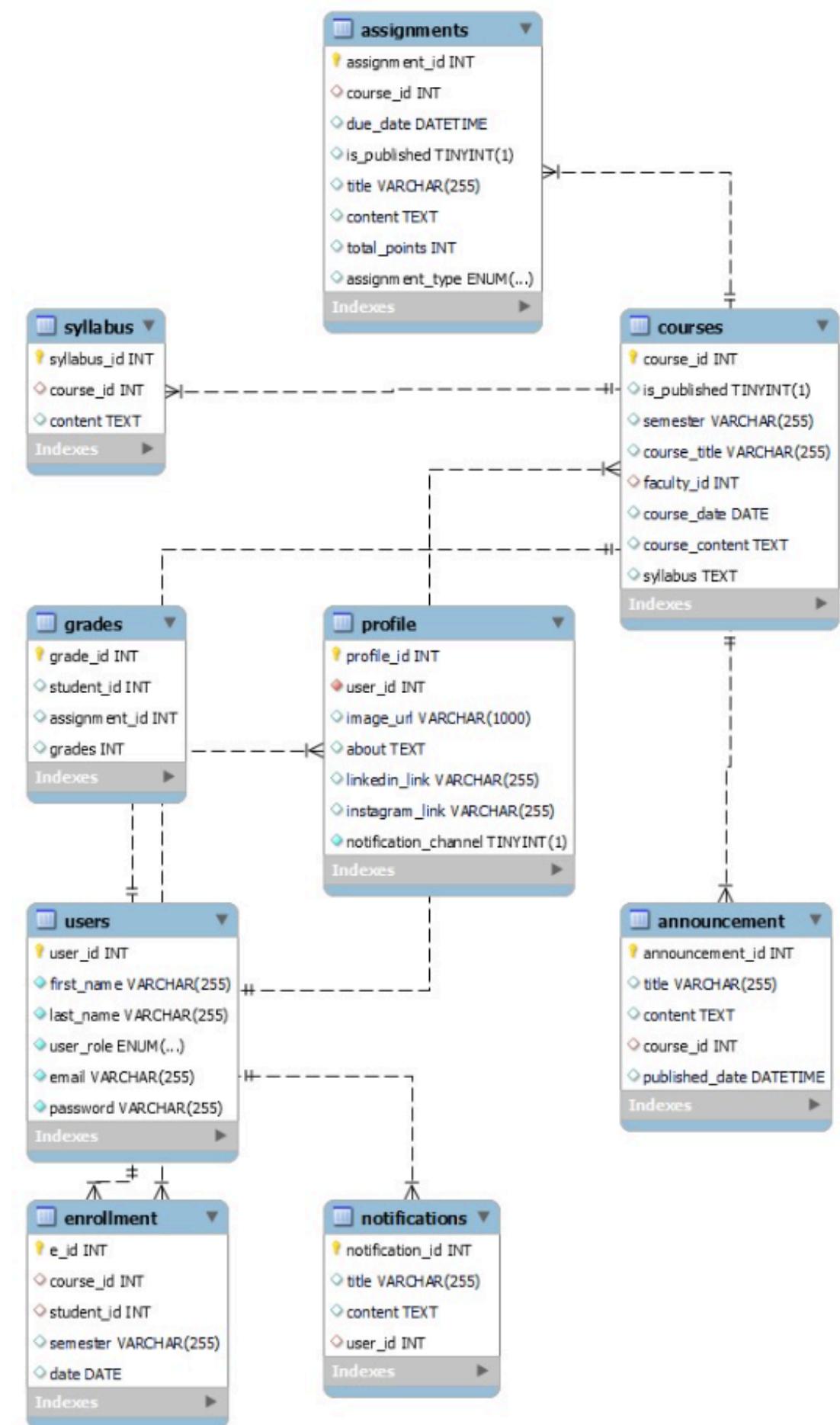


Abstraction

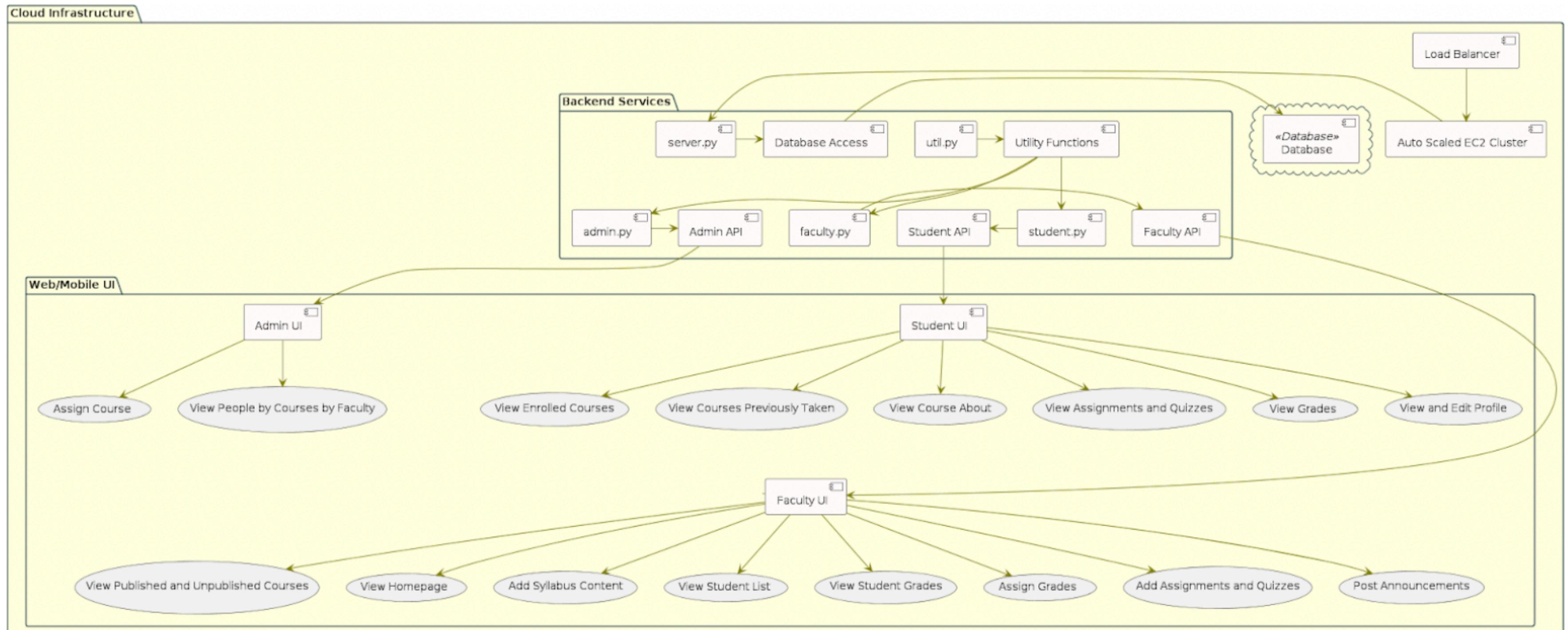
Course Management Interface: Provides a simplified way for faculty to add content and manage grades without exposing the underlying data structures and algorithms.

⋮⋮⋮

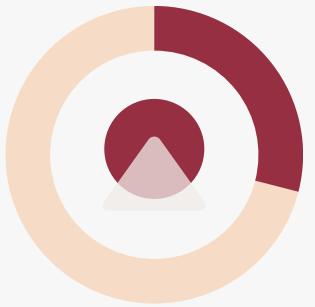
ER Diagram



Component Diagram



MVC Framework in the AAW Project



- Model: Manages the core data of the AAW, including courses, users (faculty, students, and admins), assignments, and grades. It handles data validation, database interactions, and business rules.
- View: Provides the graphical user interface for users to interact with the system. This includes dashboards for faculty, students, and admins, forms for inputting data, and pages for displaying course content and grades.
- Controller: Interprets user inputs (e.g., adding a new assignment, viewing grades), processes these inputs via the Model, and updates the View accordingly. It ensures the correct flow of data and user interactions.

Overview of Middleware or Main Logic



Definition: Middleware bridges the front-end user interface and the back-end database, processing user requests, enforcing business rules, and managing data flow.

Key Algorithms and Processes

1. User Authentication and Authorization:
 - Process: Verifies credentials, assigns roles (Faculty, Student, Admin), manages sessions.
2. Course Management:
 - Process: CRUD operations for courses; Faculty manage courses, students view enrolled courses.
3. Assignment and Quiz Management:
 - Process: Faculty create and grade assignments/quizzes, students submit and view them.
4. Grade Management:
 - Process: Faculty assign grades, students view their grades in real-time.
5. Announcement System:
 - Process: Faculty/Admin post announcements, students receive notifications.

Middleware Interactions

1. Front-End Interaction:
 - APIs: Middleware exposes APIs for front-end actions (viewing courses, submitting assignments).
 - Data Flow: User actions trigger API calls, middleware processes and updates UI.
2. Back-End Interaction:
 - Database Operations: Middleware handles fetching, updating, storing data.
 - Error Handling: Validates interactions, handles errors, provides feedback.

Middleware Components

- Controllers: Manage API endpoints, handle requests.
- Services: Implement business logic, validate and manipulate data.
- Repositories: Encapsulate database access, interface for services.

UI Mockup



Santa Clara University

User ID

Password

Sign In

By signing up, I agree to the SCU Privacy Policy and Terms of Service.

Login / SignUp Page

UI Mockup

Account

Dashboard

History

Logout

About

Assignments

Announcements

Quizzes

Grade

Assignment for 107

Assignment 1
Due by: Tue, 07 May 2024 23:59:59 GMT Total Points: 100

Assignment 2
Due by: Tue, 14 May 2024 23:59:59 GMT Total Points: 100

Assignment 3
Due by: Tue, 21 May 2024 23:59:59 GMT Total Points: 100

Assignment 4
Due by: Tue, 28 May 2024 23:59:59 GMT Total Points: 100

Assignment 5
Due by: Tue, 04 Jun 2024 23:59:59 GMT Total Points: 100

testing
Due by: Total Points: 150

Student Page

UI Mockup

The image shows a user interface mockup for a student grades application. On the left, there is a vertical sidebar with a light gray background containing navigation links. The links are organized into two columns:

Account	Assignments
Dashboard	Announcements
History	About
Logout	Quizzes
	People
	To Grade
	Grades

On the right, the main content area has a white background. At the top, it displays the title "Student Grades". Below the title is a table with a green header bar. The table has two columns: "Student Name" and "Score (%)".

Student Name	Score (%)
Alex Johnson (401)	92.80
Emma Smith (402)	93.17
Isabella Garcia (406)	90.17
Ava Davis (408)	78.40
Noah Taylor (411)	84.80
Charlotte Moore (412)	85.60
Olivia White (416)	93.75
Lucas Martin (418)	88.50
Neel Desai (721)	92.83
Samarth Sharma (722)	87.17

At the bottom right of the main content area, there is a vertical ellipsis consisting of three dots.

Student can see assigned grades

UI Mockup

The screenshot shows a web browser window with a URL of `ec2-18-144-82-123.us-west-1.compute.amazonaws.com:3000/admin/tobeassigned`. The page title is "Courses to be Assigned". On the left, there is a sidebar with links: "Account", "Faculties", "To Be Assigned" (which is the active tab), and "Logout". The main content area displays three course entries, each with a "Select Faculty" dropdown and an "Assign" button:

- Information System Strategy and Policy
Semester: Fall 24
Select Faculty ▾ Assign
- Object Oriented Programming
Semester: Fall 24
Select Faculty ▾ Assign
- Software Project Management
Semester: Fall 24
Select Faculty ▾ Assign

Admin can assign Faculty



Thank You!!

