ADAPTIVE COLLABORATIVE CODE LEARNING LAB

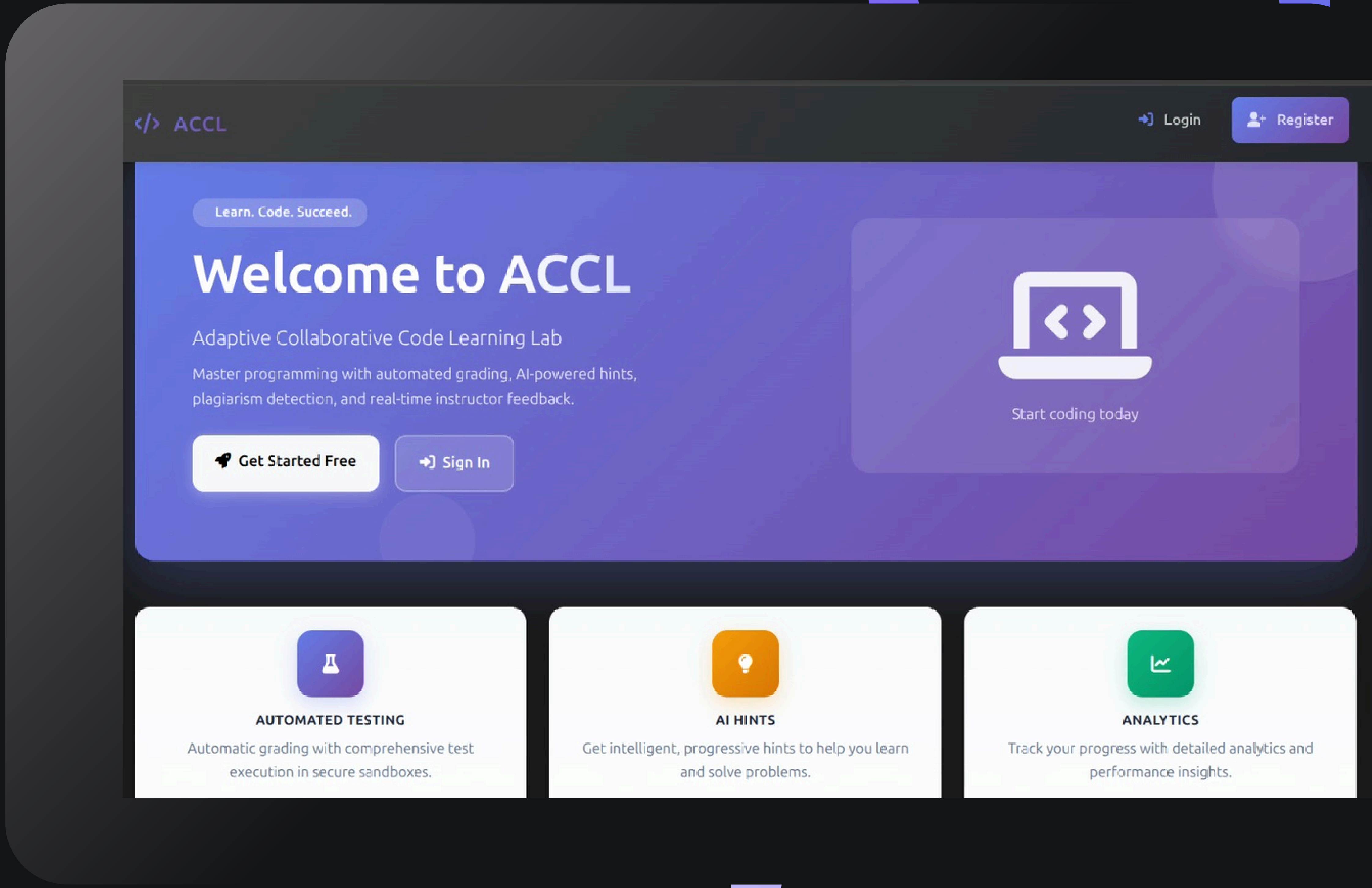
ACCL

Team Number: Team 18
Team Members:
Amr Yasser 202301043
Omar Hazem Ahmed 202300800
Abdelrahman Mohamed 202301645
Hady Emad Saeed 202301707

About us

The Adaptive Collaborative Code Learning Lab (ACCL) is a web-based platform that enables instructors to publish programming assignments and test cases, and allows students to submit, run, and iterate on code in an isolated sandbox. The system automates grading by executing instructor-provided tests, records per-test results, generates progressive AI-assisted hints using pre-trained models, and maintains versioned submission history with diffs. Instructor features include similarity detection (plagiarism advisories), peer review workflows, dashboards for analytics, and CSV export for grading.

ACCL follows a modular, service-oriented design: a web application (Flask with Blueprints) implements the MVC layers and user-facing UI, a background worker system (Redis/queue) manages sandbox jobs that run in Docker containers, and a persistence layer (SQLAlchemy-backed database and object storage) stores submissions, test results, hints, and audit logs. An AI service wrapper provides cached hint generation and an extensible similarity engine supports token/embedding hybrid comparisons; repositories and service layers encapsulate business logic for maintainability and testability.



Our Services

1. ACCL provides a secure sandbox that runs student code in isolation. Each submission is executed with strict limits on time, memory, and network access to ensure safety.

2. ACCL automatically grades submissions using instructor-defined test cases. It records per-test results and scores consistently to ensure fair and repeatable evaluation.

3. ACCL generates contextual learning hints when tests fail. These hints guide students toward solutions without revealing answers, supporting learning and improvement.

4. ACCL offers instructors dashboards for analytics, re-grading, and monitoring. These tools provide transparency, control, and efficient management of coursework.

01 ACCL Mission

Our mission is to provide a secure, fair, and intelligent learning platform that enhances programming education through automated evaluation, meaningful feedback, and collaborative learning. We aim to support students in developing strong problem-solving skills while empowering instructors with reliable tools for assessment, insight, and academic integrity. Our vision is to become a trusted educational environment where technology reinforces transparency, continuous improvement, and personalized learning, enabling both students and educators to achieve better outcomes through adaptive and data-driven code education.

Objectives

Ensure Secure and Fair Evaluation

Provide a safe, isolated execution environment that guarantees consistent, unbiased, and auditable grading of programming submissions.

Enhance Learning Through Feedback

Support student learning by delivering timely, contextual feedback, progressive hints, and clear performance insights without revealing solutions.

Promote Academic Integrity and Collaboration

Maintain originality through similarity analysis while encouraging constructive collaboration via peer review workflows.

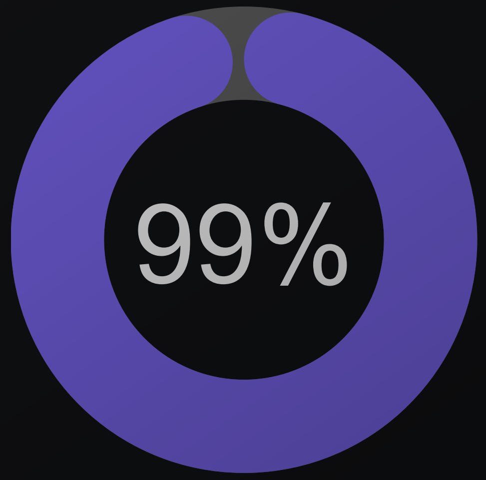
Empower Instructors and Administrators

Offer comprehensive tools for assignment management, analytics, re-grading, and system monitoring to enable effective course oversight and decision-making.s

02 Tracking Progress

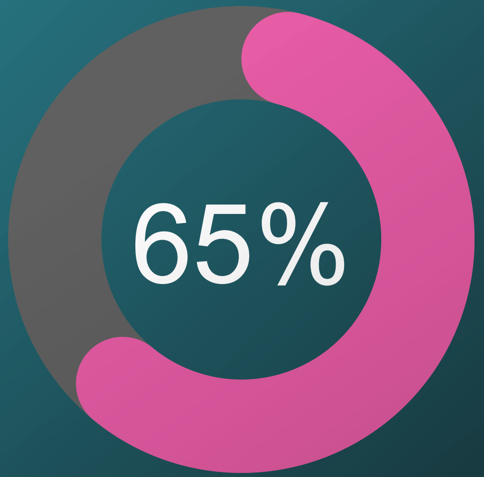
Tracking progress in ACCL is a core capability that supports continuous learning and informed instruction. The system records submission history, test outcomes, scores, and repeated failure patterns over time, allowing students to clearly observe their improvement and understand how their skills evolve across assignments. For instructors, progress tracking enables early identification of learning gaps, monitoring of individual and class-level performance trends, and evaluation of assignment effectiveness. By presenting this information through structured histories and analytics, ACCL promotes accountability, motivates students, and supports timely, data-driven educational interventions.

reliability



System monitoring and audit logging provide continuous visibility into platform operations and user actions. These mechanisms record grading activities, access events, and system status to ensure reliability, accountability, and trust among students, instructors, and administrators.

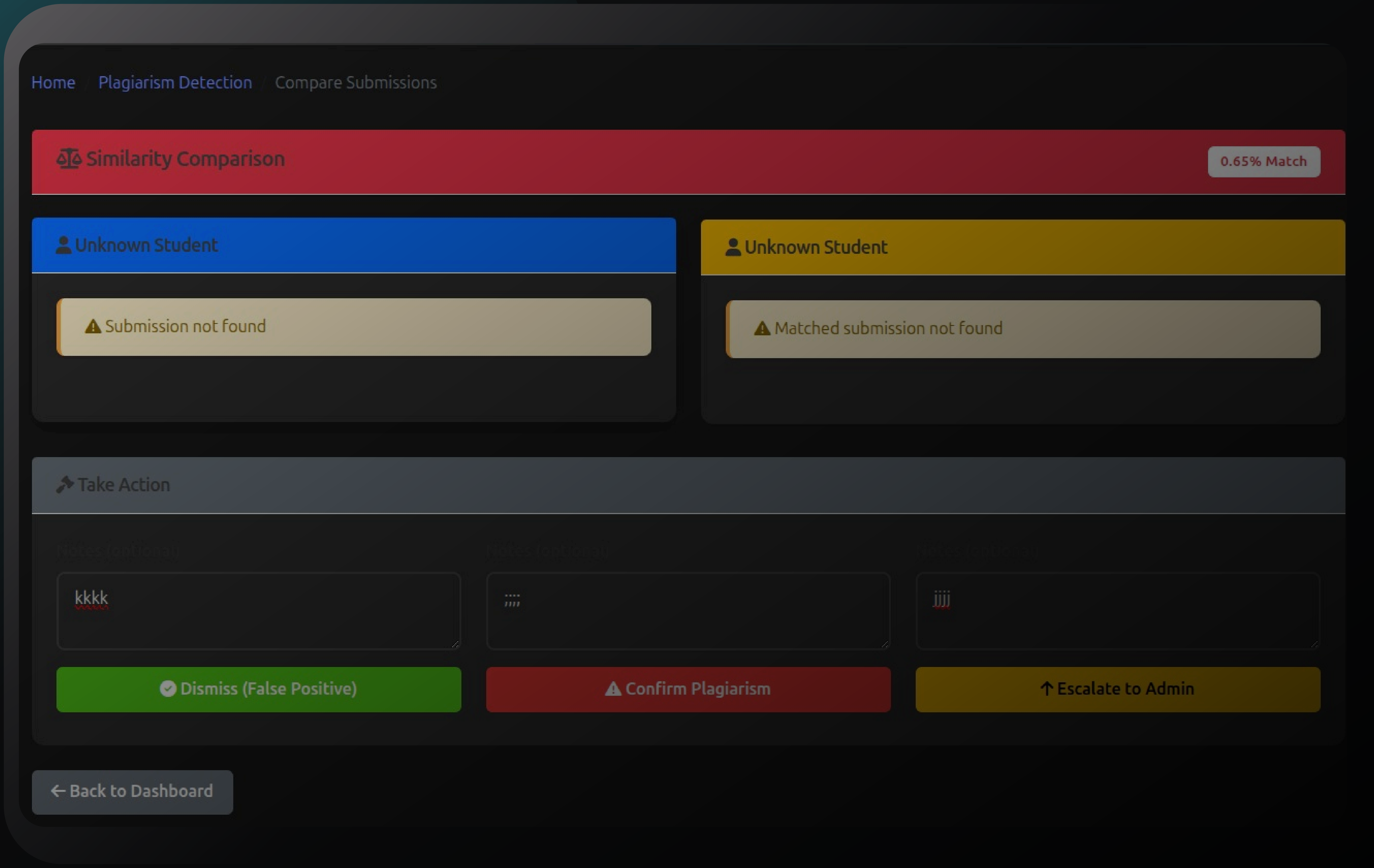
Similarity



Similarity analysis is a systematic process used to compare student submissions and identify overlapping structures or patterns. It presents instructors with clear indicators that support academic integrity while allowing human judgment in final decisions. This approach ensures fairness without relying on automated accusations.

Analysis

- Similarity analysis in ACCL is designed to support academic integrity while maintaining fairness and transparency.
- The system compares student submissions for the same assignment using code normalization, token-based methods, and embedding-based similarity techniques.
- It detects overlapping structures, logic patterns, and reused code segments across submissions.
- Instead of applying automatic penalties, the system provides similarity scores and highlighted matching sections for instructor review.
- Contextual metadata helps instructors distinguish between common boilerplate code and potential plagiarism.
- All similarity results and instructor review actions are recorded in audit logs to ensure accountability and trust.

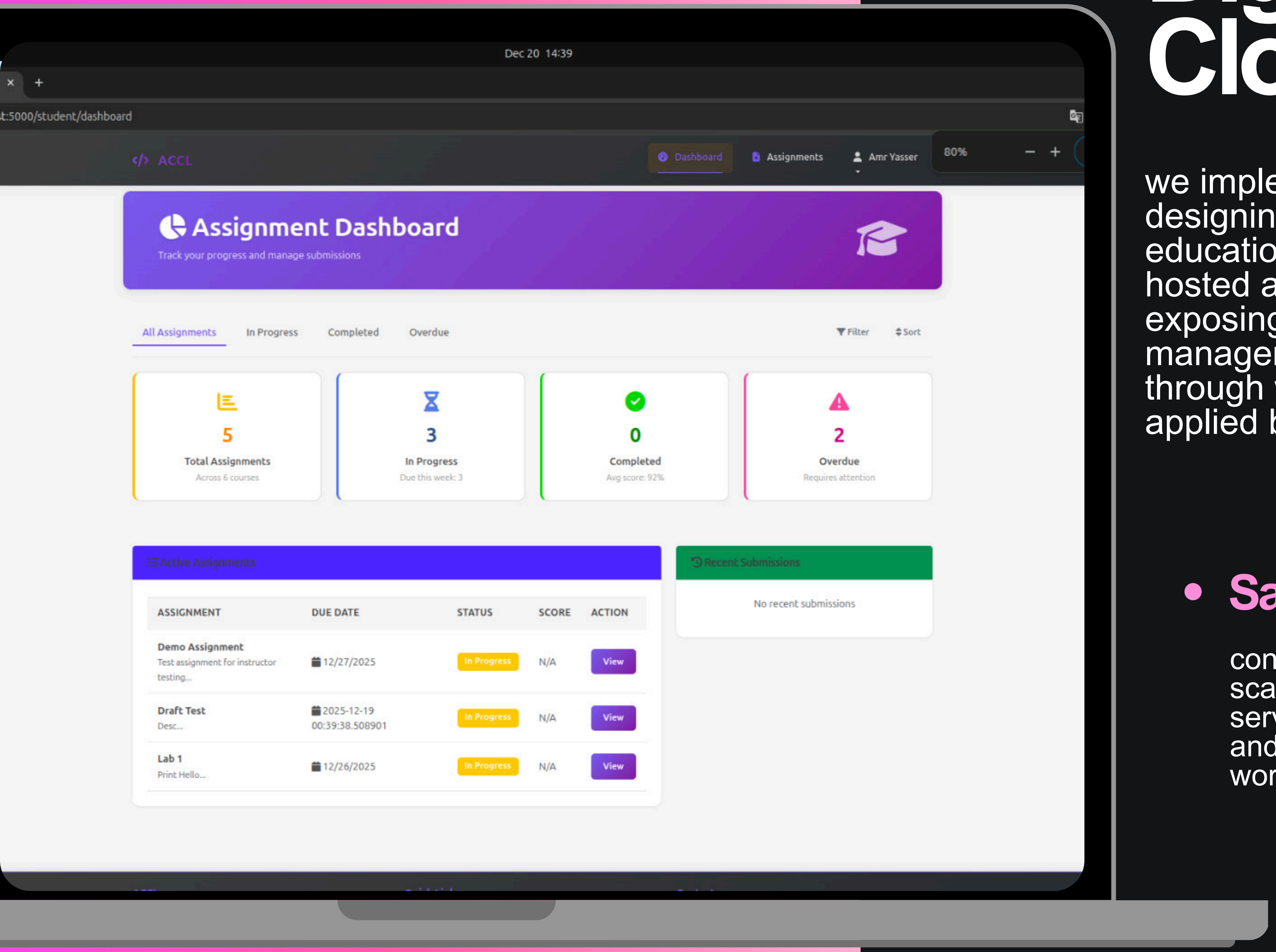


03 Architectural Style

The architecture and components of ACCL are designed for modularity and reliability. The system uses an MVC-based Flask application supported by background workers and secure sandbox containers for code execution. Additional components, such as the database, grading services, and similarity analysis engine, work together to deliver automated evaluation and feedback efficiently.

Key components

- Monolithic, layered architecture with internal MVC/MVT using Flask.
- Flask web application handles routing, views, and business logic separation.
- SQLAlchemy with SQLite for data storage; Redis for caching and job queues.
- Background workers (RQ/Celery) execute grading and asynchronous tasks.
- Docker containers provide isolated and secure code execution.
- External AI and embedding services integrated through service wrappers.
- Audit logs and metrics support monitoring and debugging.
- Architecture enables rapid development and easy deployment.
- Backend: Python, Flask, SQLAlchemy, Redis, RQ/Celery, Docker.
- Frontend: Jinja2, Bootstrap 5, Vanilla JavaScript with AJAX.
- Data: SQLite database and local file storage.
- AI: External APIs for hints and similarity scoring.
- Tools: GitHub, pytest, flake8/black, Docker Compose, GitHub Actions.



Digital Service Cloud platform

we implemented the ACCL system as a Digital Service Cloud platform by designing it as a centralized, web-based service that delivers multiple educational functionalities through a unified architecture. The platform is hosted as a single Flask application that acts as the core service layer, exposing digital services such as authentication, assignment management, automated grading, AI-assisted feedback, and analytics through well-defined controllers and APIs. Cloud-style principles were applied by using asynchronous background workers for grading,

- ## Sand Box

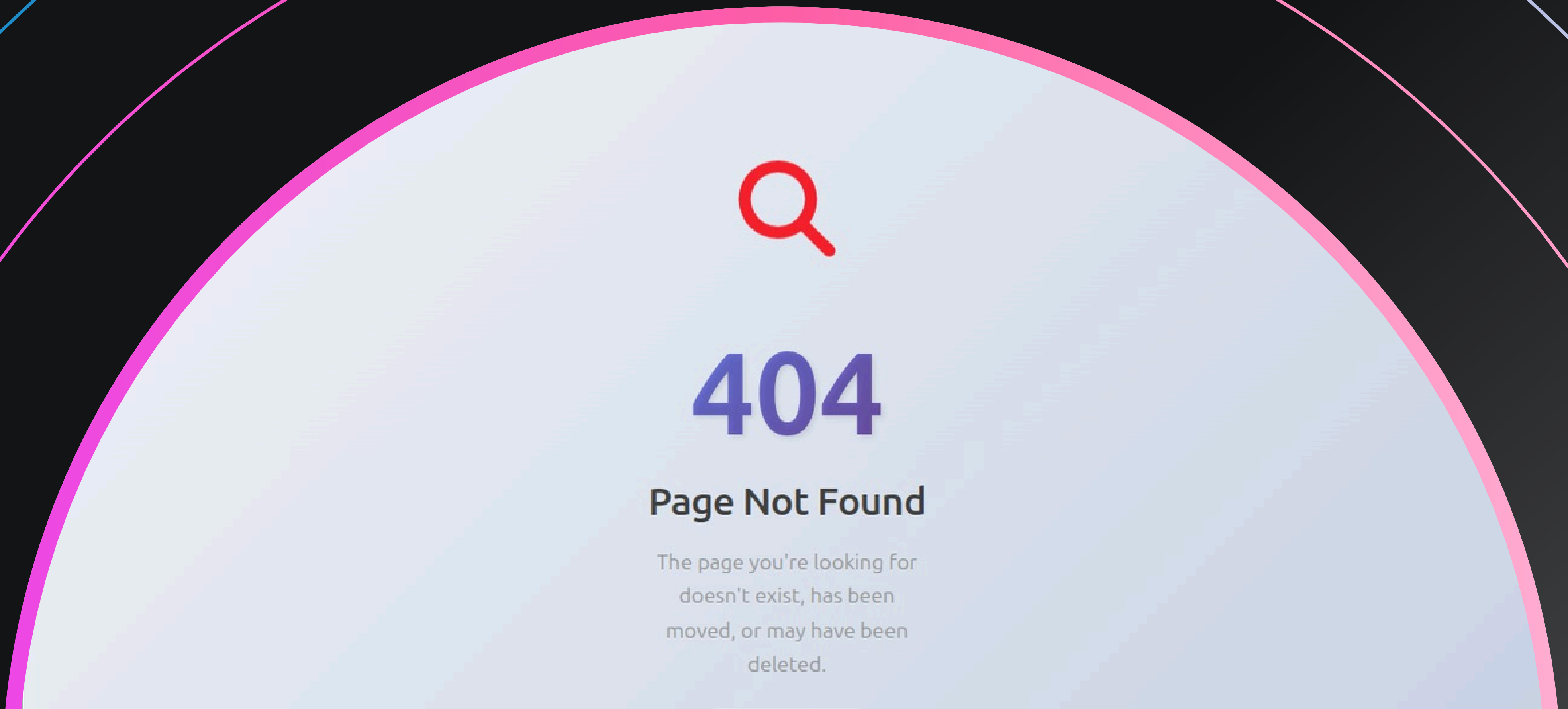
  containerized sandbox execution for scalability and isolation, and shared services such as caching, queues, and databases to manage state and workloads efficiently.

- ## Backend

  This approach allows ACCL to function as an on-demand digital service platform where users access learning, evaluation, and insight services through a browser, while computation-heavy tasks are handled transparently in the backend, reflecting modern cloud-based service design.

Resolving technical problems

Resolving technical problems in the ACCL project is handled through a combination of architectural design choices and operational tools. The system isolates complex and error-prone processes, such as code execution and grading, into background workers and sandboxed containers, preventing failures from affecting the main application. Detailed logging, audit trails, and captured execution outputs enable developers and instructors to quickly identify the source of errors. Additionally, modular components and clear separation of responsibilities make it easier to debug, update, and fix issues without disrupting other parts of the system, ensuring reliability and maintainability.



Virus and malware help

Malware and virus risks in ACCL are mitigated by enforcing strong execution and access controls across the platform. User-submitted code is never run directly on the main server; instead, it is processed in tightly controlled sandbox containers that are destroyed after execution. The system limits CPU, memory, and execution time to prevent abuse, while disabling network access and restricting file operations. These measures ensure that even if malicious code is submitted, its impact remains contained, protecting the platform, stored data, and other users from security threats.

1 **Support consultations**

Virus and malware protection in the ACCL project is addressed through preventive design rather than reactive cleanup.

2 **Local run**

All student code is executed inside isolated, containerized sandbox environments with no network access, restricted filesystem permissions, and strict resource limits, which prevents malicious code from affecting the host system or other users.

3 **Blocking outside connections**

The platform does not allow arbitrary downloads or external connections, reducing exposure to malware vectors. In addition, execution logs and audit records help detect suspicious behavior early, ensuring that potentially harmful activity is contained, monitored, and safely handled without compromising system integrity or user data.

Quote

“The goal of education is not to increase the amount of knowledge but to create the possibilities for a child to invent and discover.”
— Jean Piaget

Contact

s-amr.anwar@zewailcity.edu.eg

Representative ID: 202301043

Zewail City

Thank You