# ZEWAIL CITY OF SCIENCE AND TECHNOLOGY

University of Science and Technology
CSAI 203: Introduction to Software Engineering

# Phase 4 Progress Report

*Core Functionality Prototype (50% Completion)*

**Submitted By: Team 01**

- **Omar Amgad Mohamed** - 202400515 (Team Lead)
- **Mohamed Sherif Farouk** - 202401051
- **Osama Mohsen Abdelaziz** - 202401039
- **Mohamed Islam Elsayed** - 202401058

**Submission Date:** December 7, 2025
**Instructor:** Dr. Mohamed Sami Rakha

## 1. Executive Summary

As of December 7, 2025, Team 01 has successfully implemented the **Core Functionality Prototype** for the "Academia Connect" platform. We have achieved the primary "Cycle of Life" defined in our scope: Users can register and login with specific roles (Industry vs. Student), Industry partners can successfully post projects to the database, and Students/Faculty can view these projects in a real-time dashboard.

The application is built on a **Monolithic MVC Architecture** using **Python Flask** and **Microsoft SQL Server**, strictly adhering to the design constraints defined in the SDD (Software Design Document).

## 2. Feature Implementation Status

The following table maps our current progress against the requirements defined in our **Software Requirements Specification (SRS)** dated Nov 6, 2025.

| SRS Section | Feature Name | Status | Implementation Details |
|---|---|---|---|
| **3.1.1** | User Registration | ☑ **Completed** | Users can sign up as "Industry" or "Student". Data is stored in the `Users` table via `UserDAO`. Passwords are hashed using SHA-256. |
| **3.1.2** | User Login & Auth | ☑ **Completed** | Role-Based Access Control (RBAC) is functional. The system detects the user role and redirects to the appropriate Dashboard view. Session management is active. |
| **3.2.1** | Project Submission | ☑ **Completed** | Industry users have a dedicated form to post projects. Data is validated and inserted into the `Projects` table via `ProjectDAO`. |
| **3.3.1** | Browse Projects | ☑ **Completed** | Students and Faculty can view a dynamic list of open projects. The system performs a SQL `JOIN` to display the Company Name alongside project details. |

| SRS Section | Feature Name | Status | Implementation Details |
|---|---|---|---|
| **3.5.1** | User Interface | 🔄 **In Progress** | The UI is built using raw HTML/CSS (no frameworks as per SRS). Basic layout and responsiveness are complete. Visual polish and mobile optimization will continue in Phase 5. |
| **3.2.2** | Apply to Project | ⏳ **Pending** | The "Apply" button exists in the UI but is currently disabled. The `Applications` table logic will be implemented in Phase 5. |
| **3.4.1** | Admin Dashboard | ⏳ **Pending** | Administrative features (approving users/projects) are scheduled for Phase 5. |

# 3. Technical Architecture Achievements

To ensure code quality and maintainability (as per Course Project Grading Rubric), we have implemented the following architectural patterns:

- **MVC Pattern:** The codebase is strictly separated into `app/models` (SQL logic), `app/controllers` (Route logic), and `app/templates` (HTML).
- **Singleton Pattern:** Implemented in `app/db.py` to manage a single, efficient database connection per request lifecycle.
- **DAO Pattern:** All SQL queries are isolated in `UserDAO` and `ProjectDAO` classes, preventing SQL code from leaking into the controller logic.
- **Factory Pattern:** The application is initialized using a `create_app()` factory function in `app/__init__.py`.

# 4. DevOps & Version Control

We have successfully established a professional development workflow:

- **CI/CD:** We implemented **GitHub Actions** (`.github/workflows/main.yml`) to automatically compile and syntax-check our Python code upon every push to the main branch.
- **Branch Protection:** We utilized a strict branching strategy where `main` is protected. All features were developed on separate branches (e.g., `islam/models`, `osama/controllers`) and merged via Pull Requests.
- **Deployment Scripts:** We created automated setup scripts (`deployment/setup_env.bat`) and database schema scripts (`database_schema.sql`) to ensure reproducibility across team members' machines.

# 5. Plan for Phase 5 (Final Delivery)

Our focus for the final submission will be:

1. **Completing the Application Logic:** Allowing Students to apply for projects and Industry to accept/reject applications.
2. **Testing:** Implementing the required automated unit tests using `unittest` and integration tests.
3. **Documentation:** Finalizing the User Manual and Technical Guide.

---

**Team Signatures:**

*Omar Amgad (Team Lead)* | *Mohamed Islam* | *Osama Mohsen* | *Mohamed Sherif*