Zewail City of Science, Technology and Innovation

University of Science and Technology

School of Computational Sciences and Artificial Intelligence

CSAI 203 - Fall 2025

EduTrack

Final Delivery & Documentation

Team Number:29

Team Members:3

● Habiba Mohammed-202402213

● Ahmed Mahmoud-202401491

● Mohamed Essam-202401171

Representative Contact: s-ahmed.abda@zewailcity.edu.eg

Date:21/12/2025

Introduction:

EduTrack, The Web-Based Learning Management System, Offers Support for Students/Teachers and Administrators: Course Management, Assignment Management, Lecture Notes, Attendance Tracking, Chat, Notifications, etc. Will Describe Methodology of Testing, Usage of System, and Technical Architecture.

1.Testing Methodology:

1.1 Manual Testing

Manual testing was conducted to validate core functional and non-functional requirements.
Test scenarios included login, course creation, enrollment, assignment submission, grading, and notifications.

- Login / Logout
- Create Course
- Join Course
- Upload Lecture
- Submit Assignment
- Admin actions

1.2Automated Testing

Automated unit testing was implemented using Pytest.
The tests cover authentication, course creation, enrollment, and validation scenarios.

- Authentication tests
- Course creation tests
- Course enrollment tests
- Invalid input handling

A total of 5+ unit tests were implemented to validate major system functionalities.

2. Non-Functional Requirements Testing

Usability
3 users tested basic operations (login, view courses).
All users completed tasks within 5 minutes.

Performance
Core pages loaded under 2 seconds during manual testing.

Security
Passwords are stored using hashing techniques.
Invalid login attempts are rejected.

Reliability
The system was tested for continuous usage for over one hour without crashes.

Maintainability
The system follows MVC architecture with modular structure.

Storage Constraint
File uploads larger than 10MB are rejected.

3. User Documentation

🔸 3.1 Installation & Running

1. Install Python
2. Create virtual environment
3. Install dependencies using requirements.txt
4. Run the Flask application
5. Access the system via browser

   The system can also be run using Docker containers.

🔸 3.2 How to Use#

Student

- Register / Login
- Join Course
- View Lectures
- Submit Assignments
- View Grades
- View Notifications

Teacher

- Create Course
- Upload Lectures
- Create Assignments
- Grade Submissions
- Record Attendance

Admin

- Manage Users
- Delete Users
- Delete Courses
- Export Reports

Technical Documentation

4. Technical Documentation

◆ 4.1 Architecture

The project follows the Model–View–Controller (MVC) architecture.

Model

- Database tables
- Entities (User, Course, Assignment...)

View

- HTML templates (Jinja2)
- Bootstrap-based UI

Controller

- Flask Blueprints
- Handles routing and requests

◆ 4.2 Technologies Used

- Flask
- SQL Server
- PyODBC
- Pytest
- Docker
- GitHub Actions

◆ 4.3 CI/CD

GitHub Actions pipeline runs automated tests on each push and builds Docker images.