CS236 Lab Assignment #5: Banner Web

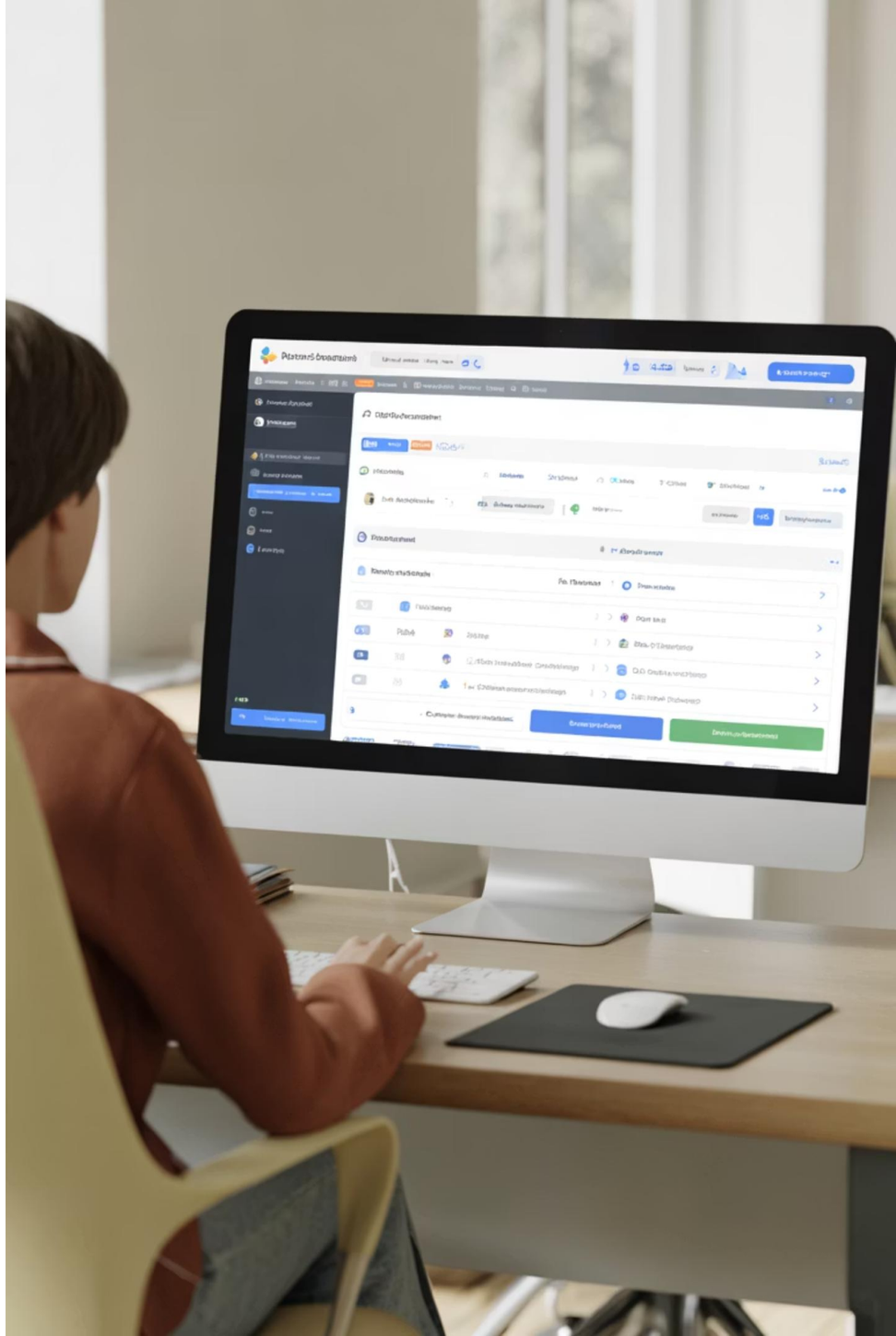
Project Documentation: Student Course Registration System

Chinyemba

Lleyton

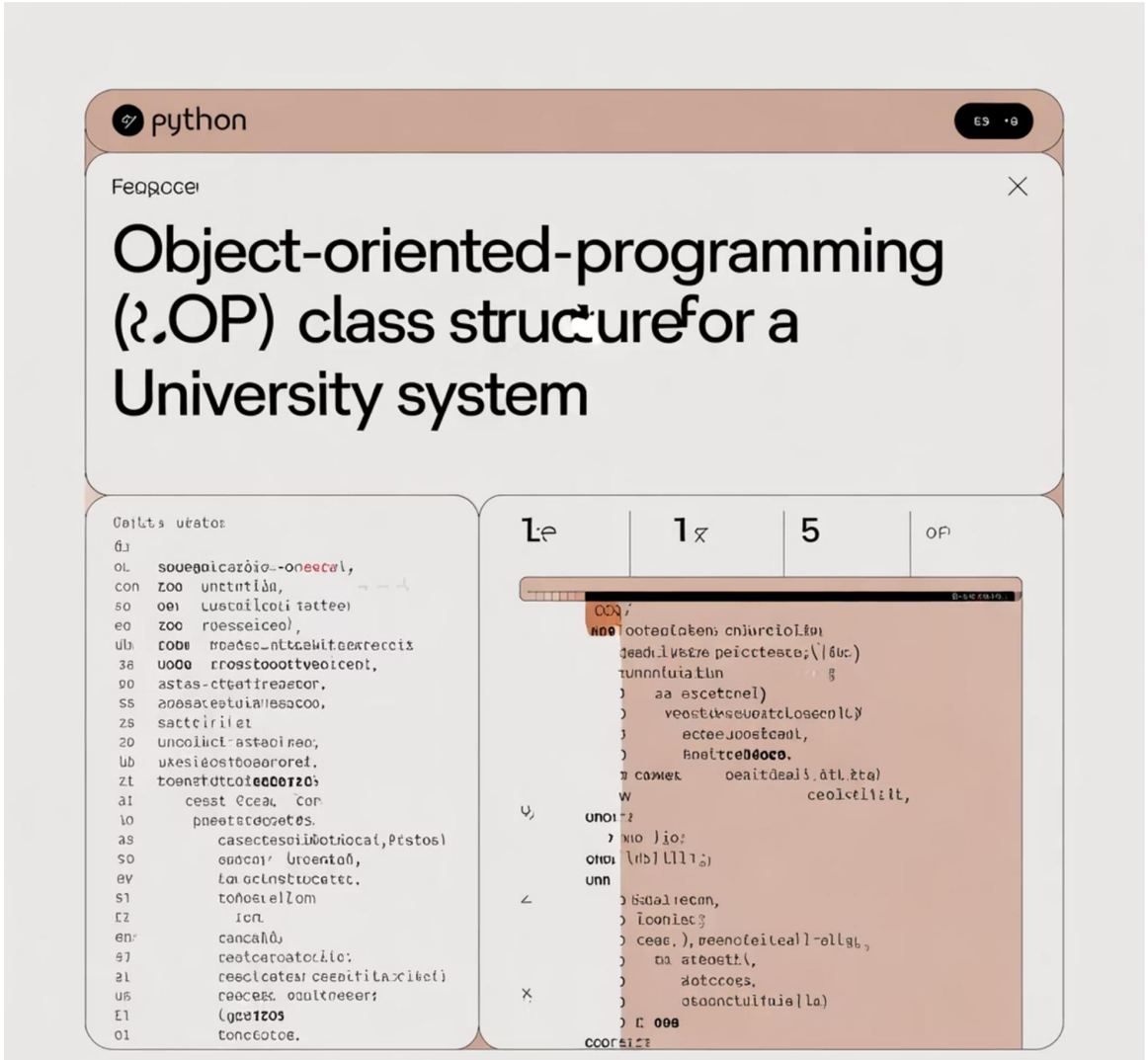
Gideon

Jonathan



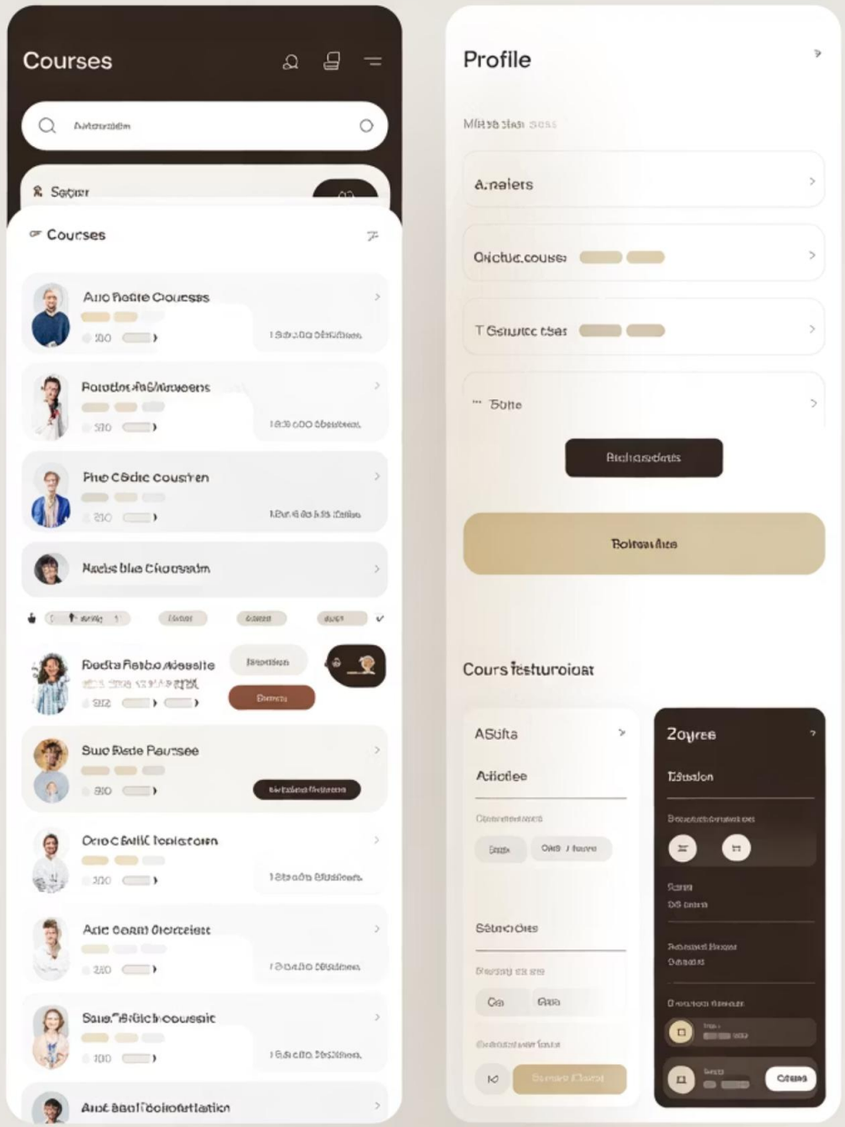
Project Overview: A Desktop Registration System

Banner Web is a university course registration simulator built using Python and the Tkinter library. It uses Object-Oriented Programming (OOP) principles to divide responsibilities cleanly across distinct classes.



Core Functionality

- ***Authenticated students can browse available courses.***
- ***Enroll or drop classes with validation.***
- ***View current schedule and print a formatted weekly timetable.***
- ***Designed for maintainability and future enhancements.***



Application Architecture: Two Primary Sections

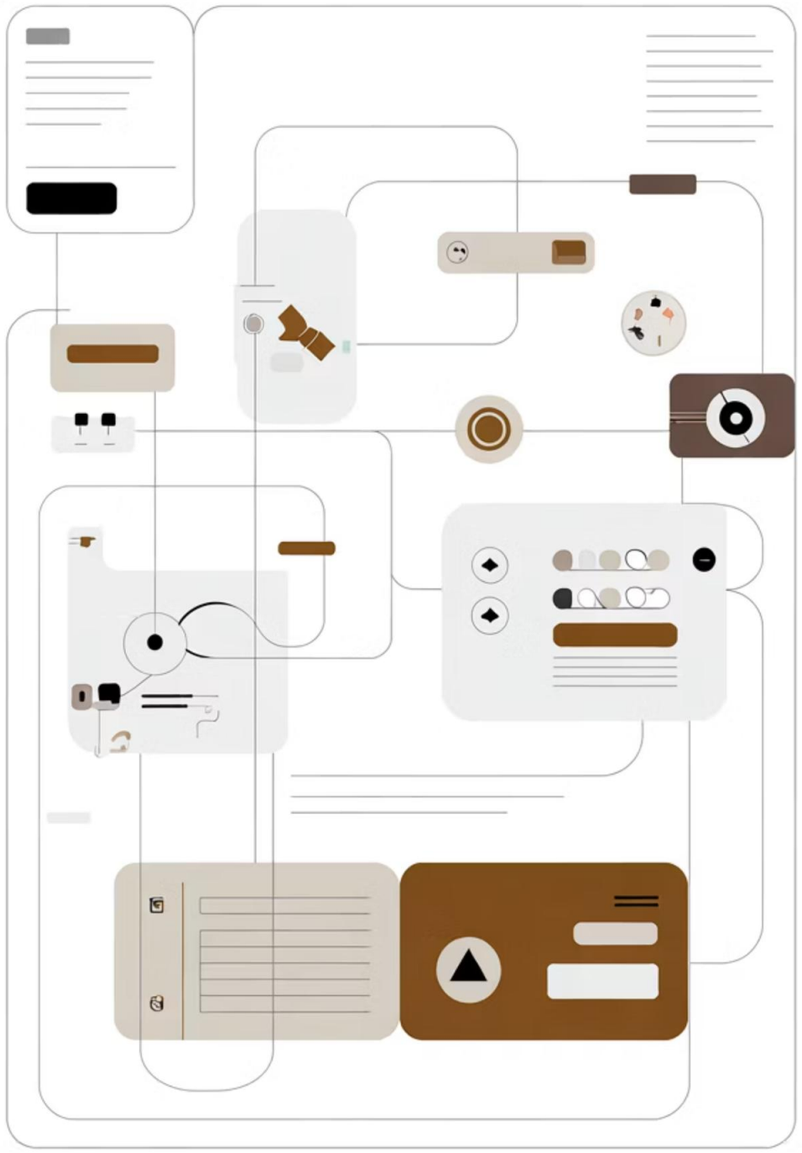
The application features a Graphical User Interface (GUI) divided into two interactive tabs for student management.

My Courses Tab

Displays currently enrolled classes as modern cards, including instructor, schedule, and location details. Allows students to drop a course and print their weekly schedule.

Browse Courses Tab

Lists all available courses with capacity and enrollment status. Supports one-click enrollment with validation for capacity and schedule conflicts.



Team Contributions: Core Enrollment Logic & GUI

Chinyemba: Core Logic & Course Model

- ***Implemented Course Class with capacity tracking (using a Set) and conflict detection utilities.***
- ***Built Enrollment System Class for business logic, CSV persistence, and schedule generation.***

Lleyton: Main Student GUI

- ***Created the modern, card-based Tkinter interface with two tabs.***
- ***Added UI polish: color scheme, status badges, enrollment limits, and live refresh after actions.***

Team Contributions: Data Models & Orchestration

Gideon: Student Model & Orchestration

Implemented the Student Class with a Set-backed collection for registered courses (O(1) operations). Wired the login flow to the main GUI and managed the application window lifecycle.

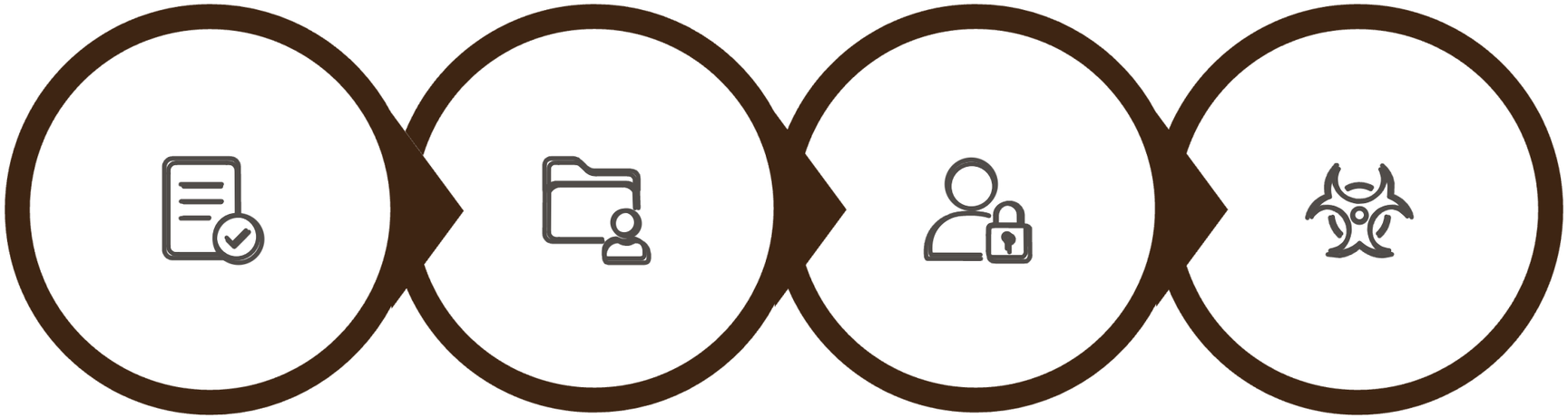
Jonathan: Authentication & UI

Built the Authentication System using SHA-256 hashing for secure credential management and CSV persistence. Developed the unified login/registration interface with form validation.

The modular design ensures efficient data handling and a smooth user experience from login to course enrollment.

Key Development Challenges

The team overcame several technical hurdles to ensure a robust and reliable system.



State Sync

Data Persistence

Auth Edge Cases

Conflict Detection

Addressing Complex Synchronization & Validation

State Synchronization

Ensuring both 'My Courses' and 'Browse Courses' tabs reflected the latest enrollment status required carefully placed refresh calls and clear data/presentation separation.

CSV Data Persistence

Reliable read/write across multiple CSVs (courses, students, enrollments) demanded strict header management and consistency between in-memory Sets and serialized formats.

Schedule Conflict Detection

Implementing practical conflict checks required robust parsing of schedule fields and guardrails for overlapping time ranges.

Capacity & Limits

Coordinating course capacity with per-student enrollment limits (max 6 courses) and providing user-friendly feedback in the GUI.

Conclusion: Successful Application of CS Concepts

The Banner Web project successfully delivered a functional, user-friendly desktop application demonstrating strong application of CS 236 concepts.

OOP Design

Modular architecture spanning authentication, business logic, and UI.

Efficient Data Structures

Use of Dictionaries and Sets for O(1) lookups and operations.

Data Persistence

File-based data storage using multiple CSV files.

Interactive GUI

Polished user interface developed with Tkinter.