Iteration 3 Report
EECE 2560: Fundamentals of Engineering Algorithms
Student Registration System

Rafi Bachir
Department of Electrical and Computer Engineering
Northeastern University
bachir.r@northeastern.edu

November 5, 2025

Important: Each student must submit individually, even though the project was completed collaboratively. This report should be prepared in **Overleaf** and exported as a **PDF** before submission.

GitHub Link: `https://github.com/leoleocantyousee/Computing_Fund_Final_Project`

Contents

1 Summary of Team Progress and Development Updates

Throughout Iteration 3, our group concentrated on transforming the Student Registration System into a functioning web-based application. The main objective for this stage was to link the Python backend to an HTML interface so that users could interact with the system visually rather than through the command line. I worked on integrating the HTML layout into the Python code, ensuring that all user actions—such as signing up, logging in, and checking out or returning books—were correctly processed by the backend.

User authentication and data persistence were major milestones for this iteration. User data was managed through Python dictionaries and saved to text files, allowing information to be reloaded at runtime. Testing involved creating several user accounts, validating login credentials, and confirming that actions like book checkouts required an active login session.

We also completed the book management feature, which stores each book as a dictionary inside a list. The system now tracks whether a book is available, who has borrowed it, and when it is due. Testing confirmed that the system properly updates availability and user records after each transaction. Looking ahead, Kyle is leading the effort to move this data into a structured SQL database for stronger reliability and security in Iteration 4.

2 Implemented Core Features

During this phase, several essential components of the system were developed or refined.

User Authentication

Purpose: Provide secure access for users to register, log in, and log out. **Implementation:** Usernames and passwords were stored in text files and loaded into dictionaries for quick verification. I implemented Python functions to handle registration validation and login authentication. **Testing:** Multiple login attempts using valid and invalid credentials confirmed that the feature worked correctly and blocked unauthorized access.

Book Management

Purpose: Track which books are checked out, available, or returned. **Implementation:** Each book is represented as a dictionary with information such as title, borrower, and status. Functions like `addBook()`, `checkoutBook()`, and `returnBook()` were created to handle the corresponding logic. **Testing:** Several manual tests were conducted to ensure availability updates and user records remained accurate after checkouts and returns.

HTML Integration

Purpose: Replace the command-line interface with a user-friendly website. **Implementation:** I combined the HTML front end with the Python backend, allowing the web interface to display real-time information and respond to user actions. **Testing:** The site was opened in a browser and checked to ensure that each function—login, registration, and checkout—operated as expected and displayed confirmation messages correctly.

3 Challenges and Resolutions

- **Challenge 1:** Synchronizing the HTML front end with dynamically changing data from Python. **Resolution:** Adjusted the code structure so the backend could regenerate the HTML page using helper functions like `generateHTMLPage()`. This allowed live updates after user interactions without breaking the interface.
- **Challenge 2:** Ensuring that file updates remained consistent when multiple operations were performed in quick succession. **Resolution:** Added dedicated save functions that write user and book data immediately after every action, eliminating temporary mismatches between memory and storage files. Verification was done by checking file contents before and after execution.

4 Leadership Rotation and Team Contributions

Leadership Summary

Week/Span	Leader	Responsibilities	Key Outcomes
Week 1	Leonardo Fernandes	Designed and formatted the HTML interface	Developed main layo
Week 2	Rafi Bachir	Integrated Python backend with web interface	Enabled website to l
Week 3	Kyle Preger	Planned SQL database design	Started implementin

Individual Contributions

Team Member	Contributions (Technical / Documentation)	Hours
Leonardo Fernandes	Developed HTML templates and interface styling	5 hrs
Rafi Bachir	Merged HTML with Python backend, implemented authentication and feature testing	7 hrs
Kyle Preger	Assisted with HTML integration and drafted SQL database structure	6 hrs

Statement by the Individual Submitter

I, **Rafi Bachir**, confirm that the information above accurately represents my contributions during Iteration 3.