



# **Student Progress Management System (CLI Edition)**



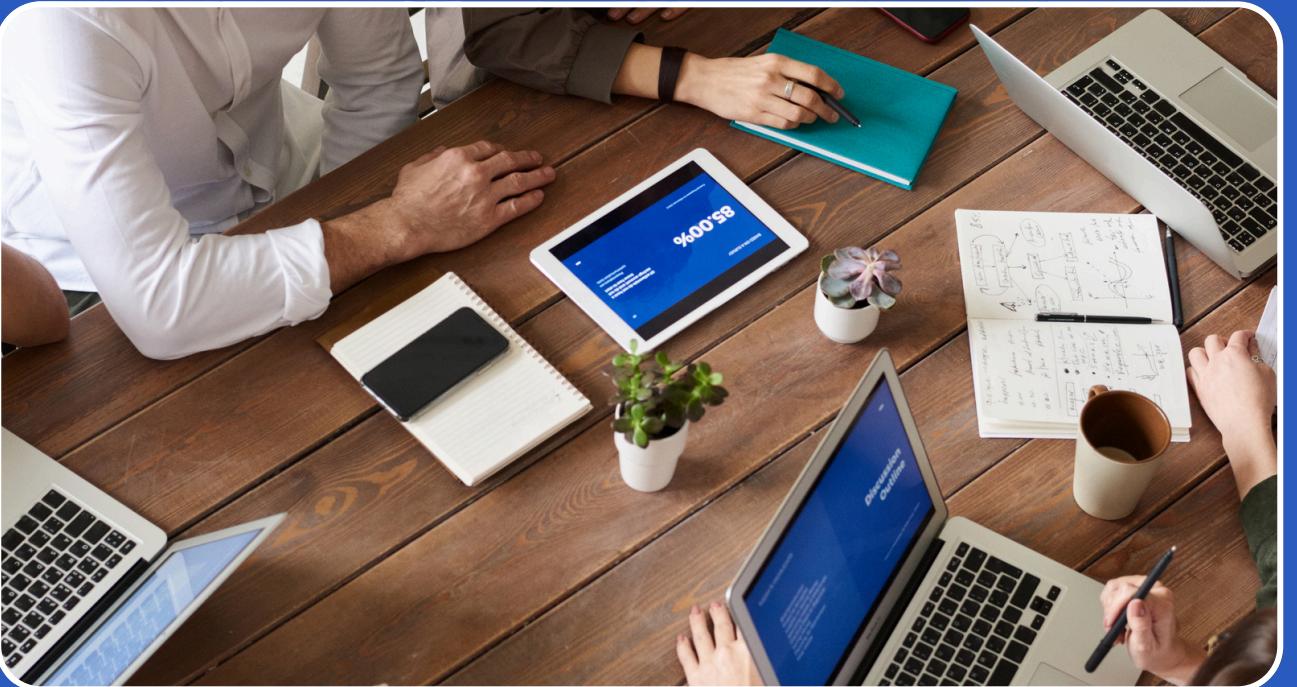
# Problem Statement



In many educational institutions, students often lack access to a centralized, personalized system for tracking their academic performance, financial standing, and disciplinary records. Existing systems are either paper-based, fragmented across departments, or too generalized to cater to individual student needs. This gap makes it difficult for learners to monitor their own academic journey in real time, identify areas needing improvement, and take ownership of their progress.

# SOLUTION

Our mission is to address key gaps in monitoring student progress, ensuring early intervention where academic, financial, or disciplinary challenges arise. Our solution empowers instructors and administrators with a streamlined way to assess and support learners, while students gain awareness of their standing.



# Student Progress Management System

## LMS



 **What it is:** A command-line learning management system built for role-based access and modular structure.



 **What it does:** Guides users through tailored menus and securely interacts with a MySQL database.

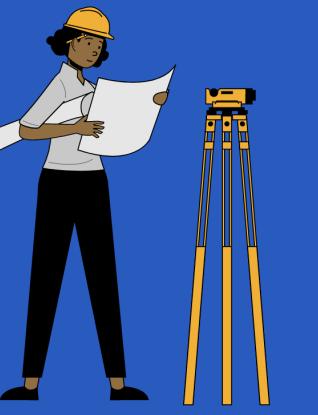


 **Why it matters:** Built for clarity, collaboration, and scalable growth in real-world dev projects.

## User Journey Diagram



# Prototype Description



## Application User Flow

This diagram illustrates the primary user journey. The application starts, authenticates the user, and directs them to a specific menu based on their role. This structured flow ensures users only access the features relevant to them.

**Start Application**  
(main.py)



**User Login & Role Selection**

Authentication Gate

**Student Menu**  
(menus.py)

**Instructor Menu**  
(menus.py)

**Admin Menu**  
(menus.py)

```
PS C:\Users\HP\OneDrive\Desktop\LMS> cd ..\..\LMS  
C:\Users\HP\OneDrive\Desktop\LMS>
```

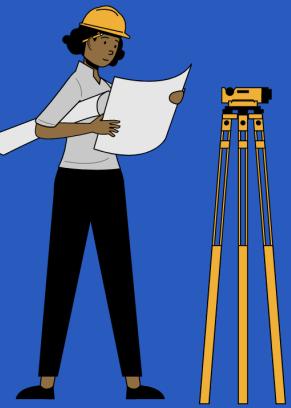
```
=====  
Hello, welcome to Group 30's LMS  
=====
```

**Authentication:**

1. Login
2. Sign up
3. Exit LMS

Enter choice (1, 2, 3):

# Login



# Signup



## User Role Capabilities

Student	Instructor	Admin
<p>Students have access to features focused on their personal academic journey, from viewing course materials and assignments to checking their overall standing.</p> <ul style="list-style-type: none"><li>• View Courses</li><li>• Check Academic Progress</li><li>• Check Financial Standing</li><li>• Disciplinary Standing</li><li>• View Assignments</li></ul>	<p>Instructors are equipped with tools to manage their courses and students effectively, including assignment creation and progress tracking.</p> <ul style="list-style-type: none"><li>• See Courses Facilitated</li><li>• Create Assignments</li><li>• Check Student Records</li><li>• Update Academic Progress</li></ul>	<p>Admins hold the keys to the entire system, managing users, courses, and system-wide statuses to ensure the LMS runs smoothly.</p> <ul style="list-style-type: none"><li>• Manage People</li><li>• Assign Instructor</li><li>• View Signup Requests</li><li>• Update Disciplinary</li><li>• Update Financial</li><li>• Add Course</li></ul>

Welcome to our login page.

Enter your email: p.obeng@alustudent.co  
Enter your password: peniel@123  
Login successful. Welcome student!

What would you like to do?:  
1. View Courses  
2. Check Academic Progress  
3. Check Financial Standing  
4. Disciplinary Standing  
5. View Assignments  
6. Exit  
Enter your choice (1-6):

Welcome to our login page.

Enter your email: kenia@alueducation.co  
Enter your password: kenia  
Login successful. Welcome instructor!

What would you like to do?:  
1. View Courses You Facilitate  
2. Create Assignments  
3. Check Student Records  
4. Update Academic Progress  
5. Exit  
Enter your choice (1-5):

Welcome to our login page.

Enter your email: test@aluadmin.com  
Enter your password: admin123  
Login successful. Welcome admin!

What would you like to do?:  
1. Manage People  
2. Assign Instructor  
3. View Signup Requests  
4. Update Disciplinary  
5. Update Financial  
6. Add course  
7. Exit  
Enter your choice (1-7):

Hello, welcome to Group 30's LMS

=====

**Authentication:**

1. Login
2. Sign up
3. Exit LMS

Enter choice (1, 2, 3): 2

Enter your full names: New Student

Enter your email: n.student@alustudent.com

Enter your password: new

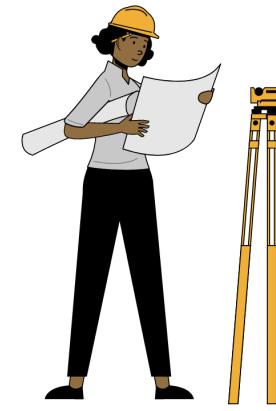
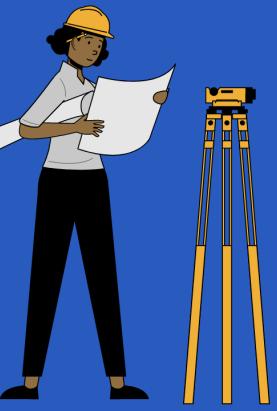
Signup request submitted. Awaiting admin approval.

pending\_signups LIMIT 100

id	full_name	* email	* password_hash	* role
	New Student	n.student@alustudent.com	\$2b\$12\$d3V9fx22GhG3qMl	student

**brypt**

# Prototype Description



View Courses

```
Enter your choice (1-6): 1
Courses enrolled by p.obeng@alustudent.com:
Courses:
- Learn the basics of programming using Python. (Course ID: Introduction to Programming)
- An in-depth look at data structures and algorithms. (Course ID: Data Structures and Algorithms)
- Build dynamic websites using HTML, CSS, and JavaScript. (Course ID: Web Development)
- Bse (Course ID: Python programming)
- Programming (Course ID: BSE)
- msq1 with python (Course ID: Data base)
- Lower level programming (Course ID: C programming)
- All about web infrastructures (Course ID: Web Infrastructure)
- Self Leadership And Team Dynamics (Course ID: SLTD)
```

View Assignments

```
2. Check Academic Progress
3. Check Financial Standing
4. Disciplinary Standing
5. View Assignments
6. Exit
Enter your choice (1-6): 5
Emacs | fa | Weight: 12.0% | Grade: 20.0%
Python_classes | sa | Weight: 30.0% | Grade: 25.0%
```

Student Status

```
Enter the student's full name: Peniel Nissi Obeng
Current status for Peniel Nissi Obeng is:
Academic: good
Financial: Good
Disciplinary: Suspended
```

Admin Approval

```
Admin Approval Menu
-----
1. View Pending Requests
2. Approve Signup by Email
3. Reject Signup by Email
4. Exit
Choose an option: 1
Pending Sign-Up Requests:
1. New Student - n.student@alustudent.com - Requested on 2025-07-27 21:53:33

Admin Approval Menu
-----
1. View Pending Requests
2. Approve Signup by Email
3. Reject Signup by Email
4. Exit
Choose an option: 1
```

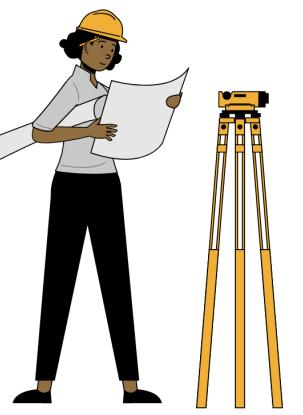
```
=====MANAGE LMS USERS=====
1. See all users
2. Remove People
3. Exit
Enter your choice (1-3): 1
--- List of Users ---
Name: Test Admin -- Email: test@aluadmin.com -- Role: admins
Name: Peniel Nissi Obeng -- Email: p.obeng@alustudent.com -- Role: students
Name: Natinael Boda Borana -- Email: n.borana@alustudent.com -- Role: students
Name: Natinael Admin -- Email: natinael@aluadmin.com -- Role: admins
Name: Kenia Umutoni -- Email: kenia@alueducation.com -- Role: instructors
```

Manage People

Add a new course

```
6. Add course
7. Exit
Enter your choice (1-7): 6
Enter course title or 'done' to finish: OOP
Enter course description: Object oriented programming
Course 'OOP' added.
Enter course title or 'done' to finish: done
Course entry complete.
```

# Important Features



- » Role Based Menus
  - Regular Expressions (re)
  - Pattern Matching
- » Secure Authentication
  - Admin approval for sign-ups
  - Login tied to user role
- » Enrollment and Data Integrity
  - Password Encryption(bcrypt)
  - Database Hygiene
  - Error handling

## Snippets

```
# Email pattern matching
pattern_student = r"^[a-zA-Z0-9._%+-]+@alustudent\.com$"
pattern_education = r"^[a-zA-Z0-9._%+-]+@alueducation\.com$"
pattern_admin = r"^[a-zA-Z0-9._%+-]+@aluadmin\.com$"
```

```
    menus.Student_menu(email)
    break
elif re.match(pattern_education, email) and stored_role == "instructor":
    menus.Instructor_menu(email)
    break
elif re.match(pattern_admin, email) and stored_role == "admin":
```

```
Choose an option: 1
Pending Sign-Up Requests:
1. New Student - n.student@alustudent.com - Requested on 2025-07-27 21:53:33

Admin Approval Menu
-----
1. View Pending Requests
2. Approve Signup by Email
3. Reject Signup by Email
4. Exit
```

```
# Password verification helper
def check_password(password: str, hashed: str) -> bool:
    return bcrypt.checkpw(password.encode('utf-8'), hashed.encode('utf-8'))
```

# CHALLENGES



Collaboration Across Modules

Database Connectivity Management

Version Control Conflicts

Difficulty In command-line  
Interaction





# Uniqueness

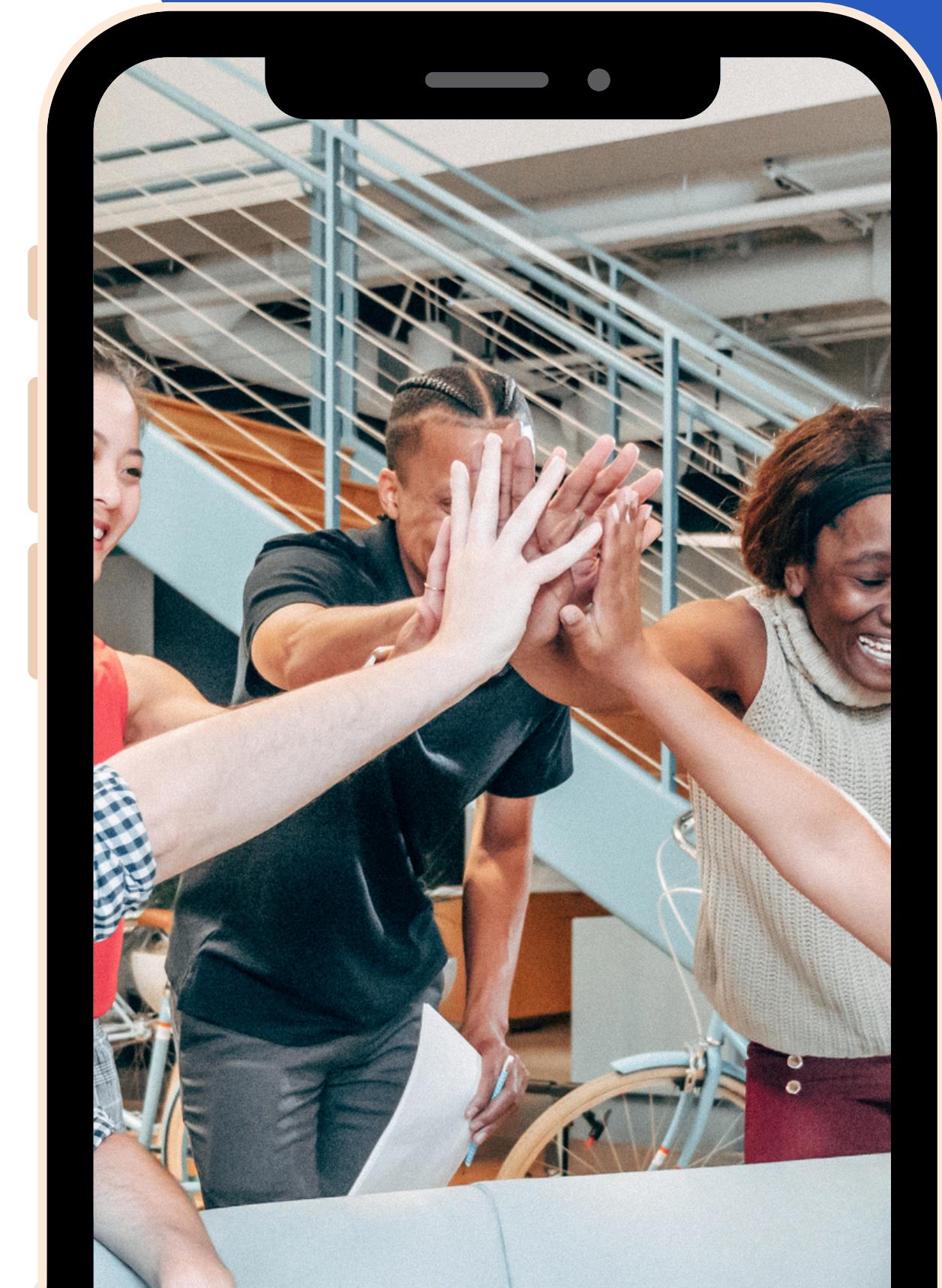
## What makes Student Progress Unique?

- ✓ Holistic student overview – students view academic, financial, and disciplinary standing all in one place
- ✓ Accessibility – no need to go to offices
- ✓ Role based access
- ✓ Offline friendly

# CONCLUSION

## Goals and Objectives:

- Build a modular LMS in Python.
- Enable students, instructors, and admins to interact with the system according to their specific responsibilities.
- Ensure real-time academic tracking by allowing students to instantly view their grades, disciplinary records, and financial standing through an interactive terminal-based interface.



# Thank You!



Group members:

- Peniel Nissi Obeng
- Natinael Boda Borana
- Esther KIRABO
- Kenia Umutoni