

IEEE.ZSB.ROS

PROJECT

TEAM F

# TABLE OF CONTENTS:

- Team members
- Overview about project
- Roadmap
- challenges
- problems
- Solutions

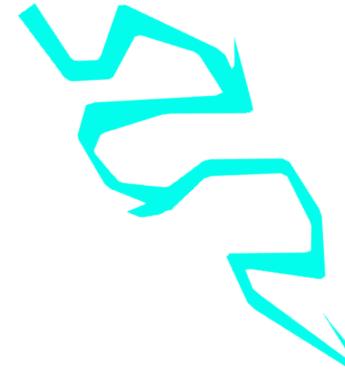


# TEAM MEMBERS:

- *Mohsen Muhammed*
- *Omar Elghitany*
- *Omar Ayman*
- *Rawida*

# OVERVIEW:

## University Management System



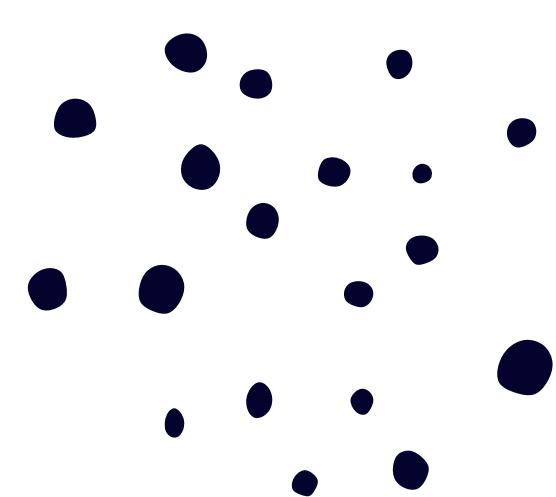
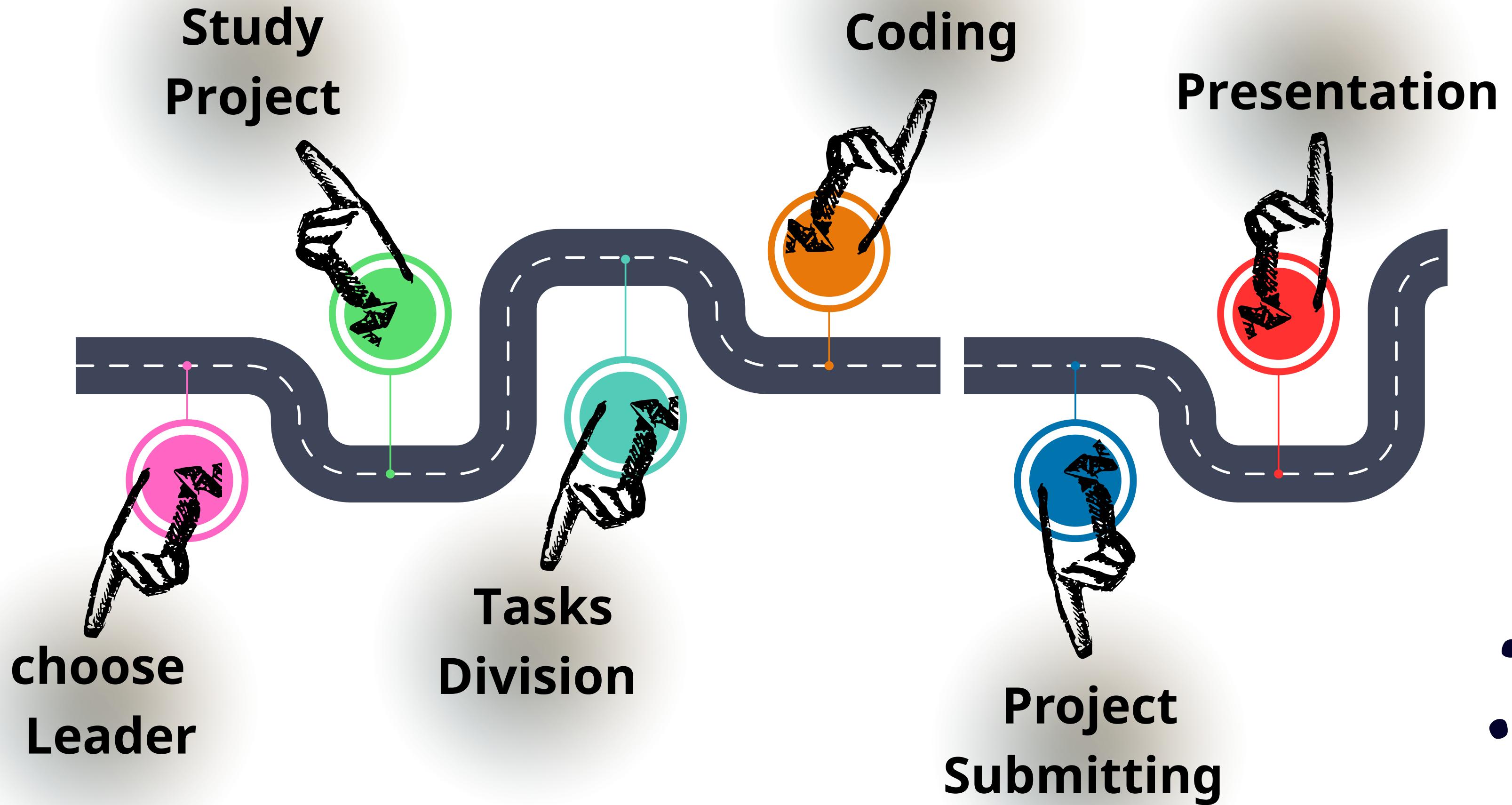
### “ Purpose

Manage student enrollments, courses, attendance, and grading.

### “ Key Features

- Student registration and course enrollment.
- Professor grading/attendance tools.
- Automated GPA calculation.
- College-specific courses.

# ROADMAP:



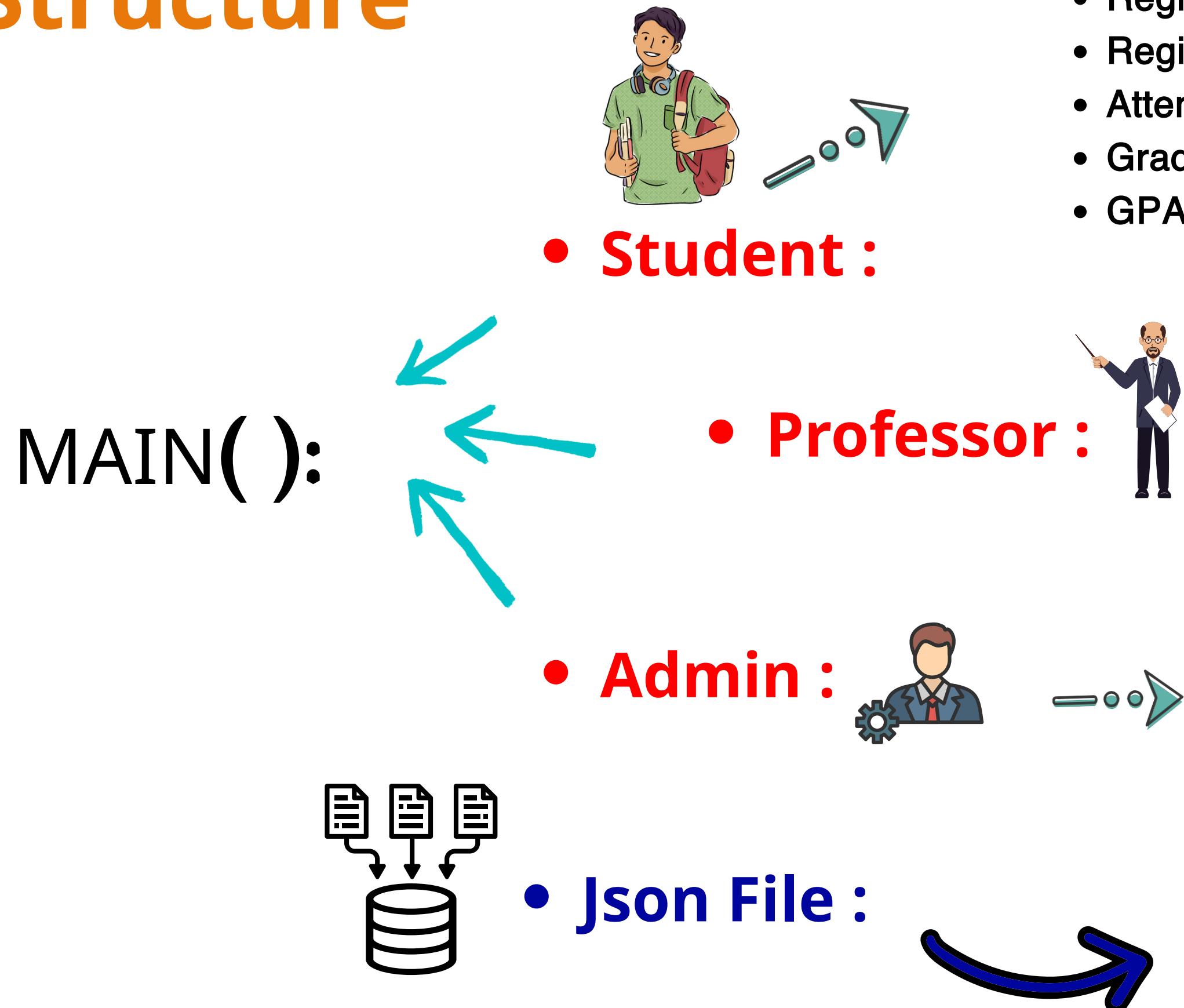
# CHALLENGES & PROBLEMS :

- Structure
- Interface
- Data Movement
- Main ()
- Solutions



# CHALLENGES :

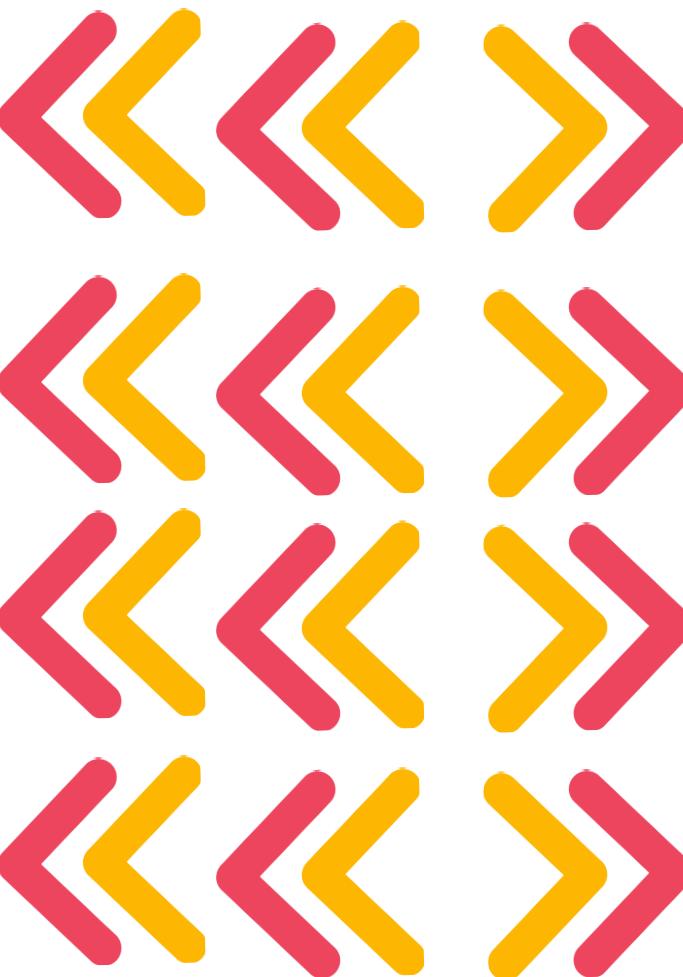
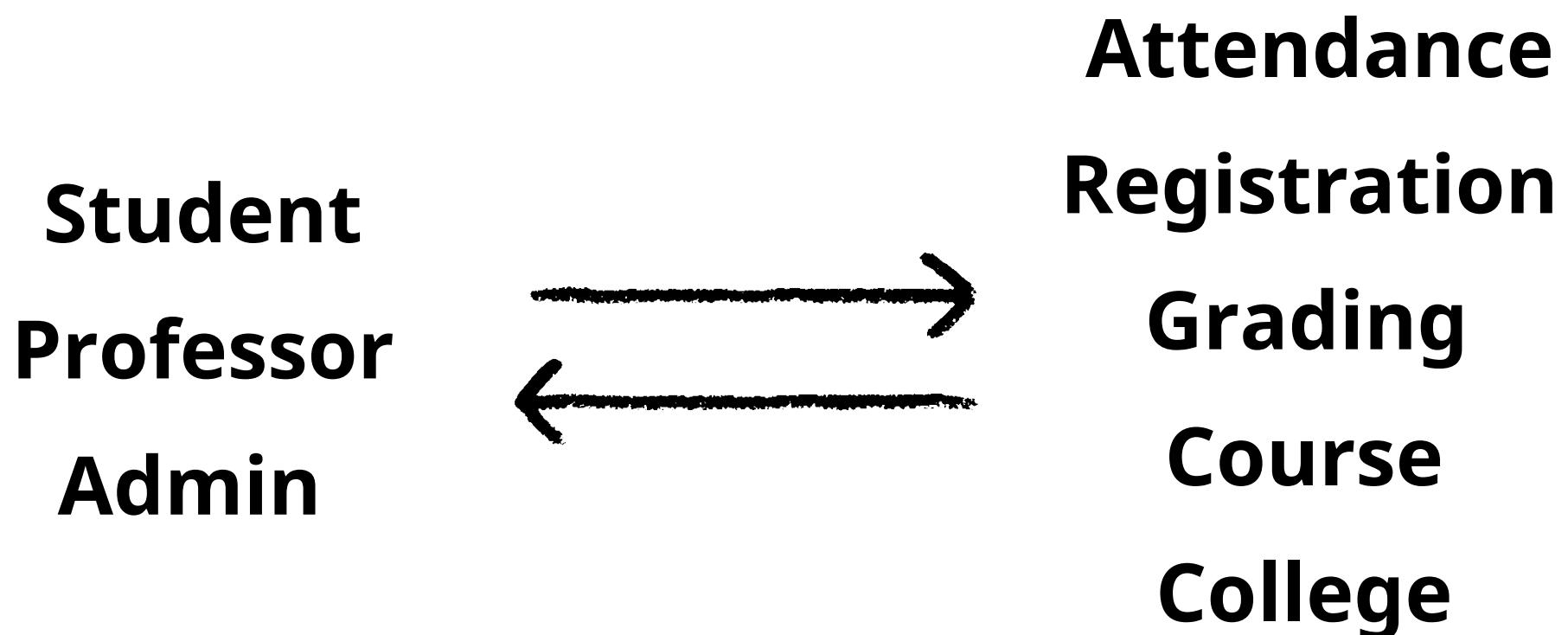
## Structure



- Register Student () :
- Register Courses () :
- Attendance () :
- Grades () :
- GPA Calculation () :
- Assign Grades () :
- Mark Attendance () :
- Manage Courses () :
  - Course Name
  - Credits
  - Department
- Assign To Courses () :
- Assign Professors To Courses () :
- Student () : *Add & Remove*
- Professors () : *Add & Remove*
- Courses () : *Add & Remove*
- Initial Data
- Call @start & Save @End
- Write & Read

# CHALLENGES :

## Data Movement



# CHALLENGES :

## Structure

### Main Functions:

**register\_student()**: Registers a new student and adds them to the system.

**register\_course\_for\_student()**: Registers courses for the student.

**mark\_attendance()**: Marks attendance for students in each course.

**assign\_grade()**: Assigns grades to students.

**calculate\_gpa()**: Calculates GPA based on registered courses and grades.

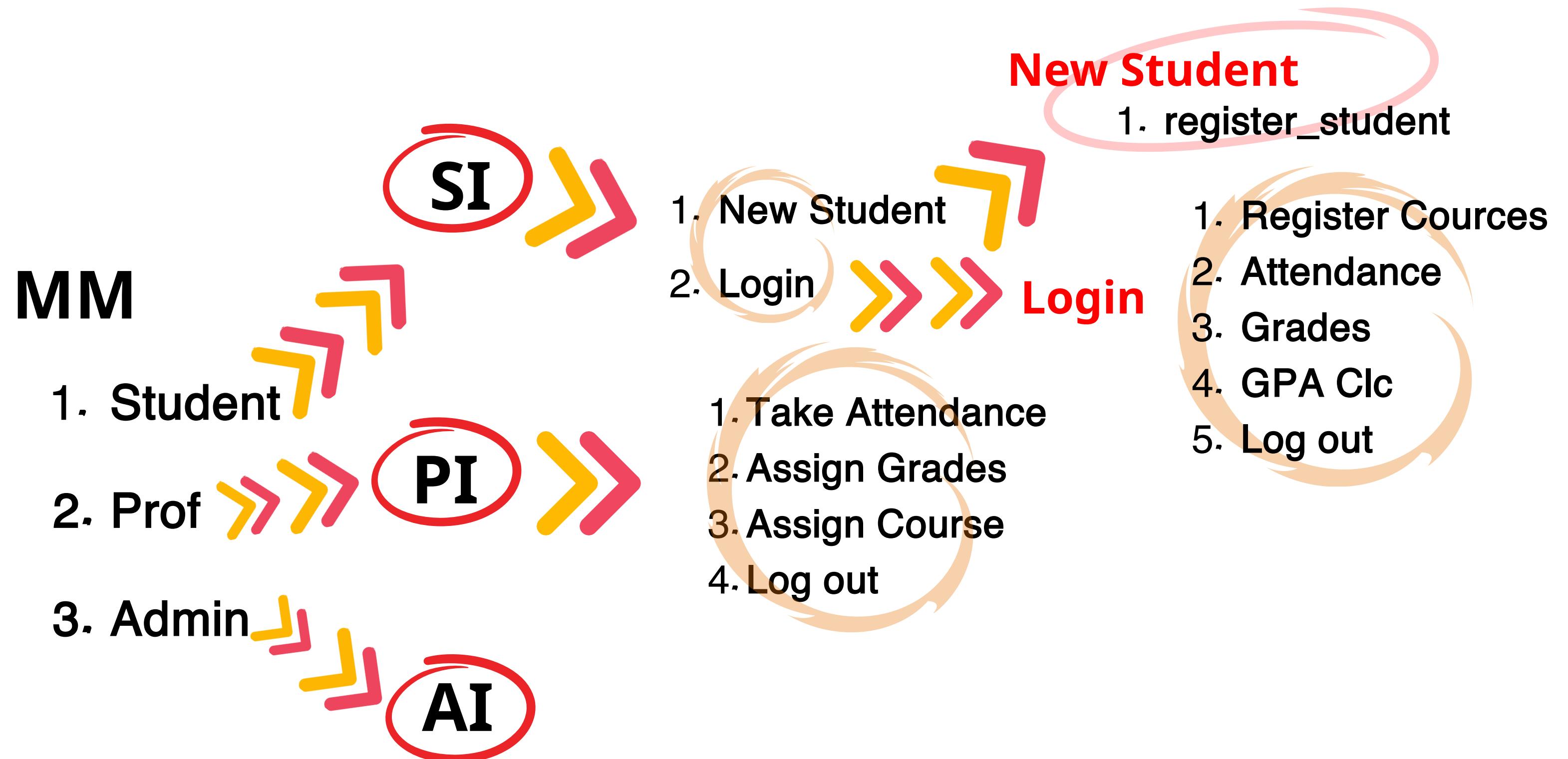


### Main code files:

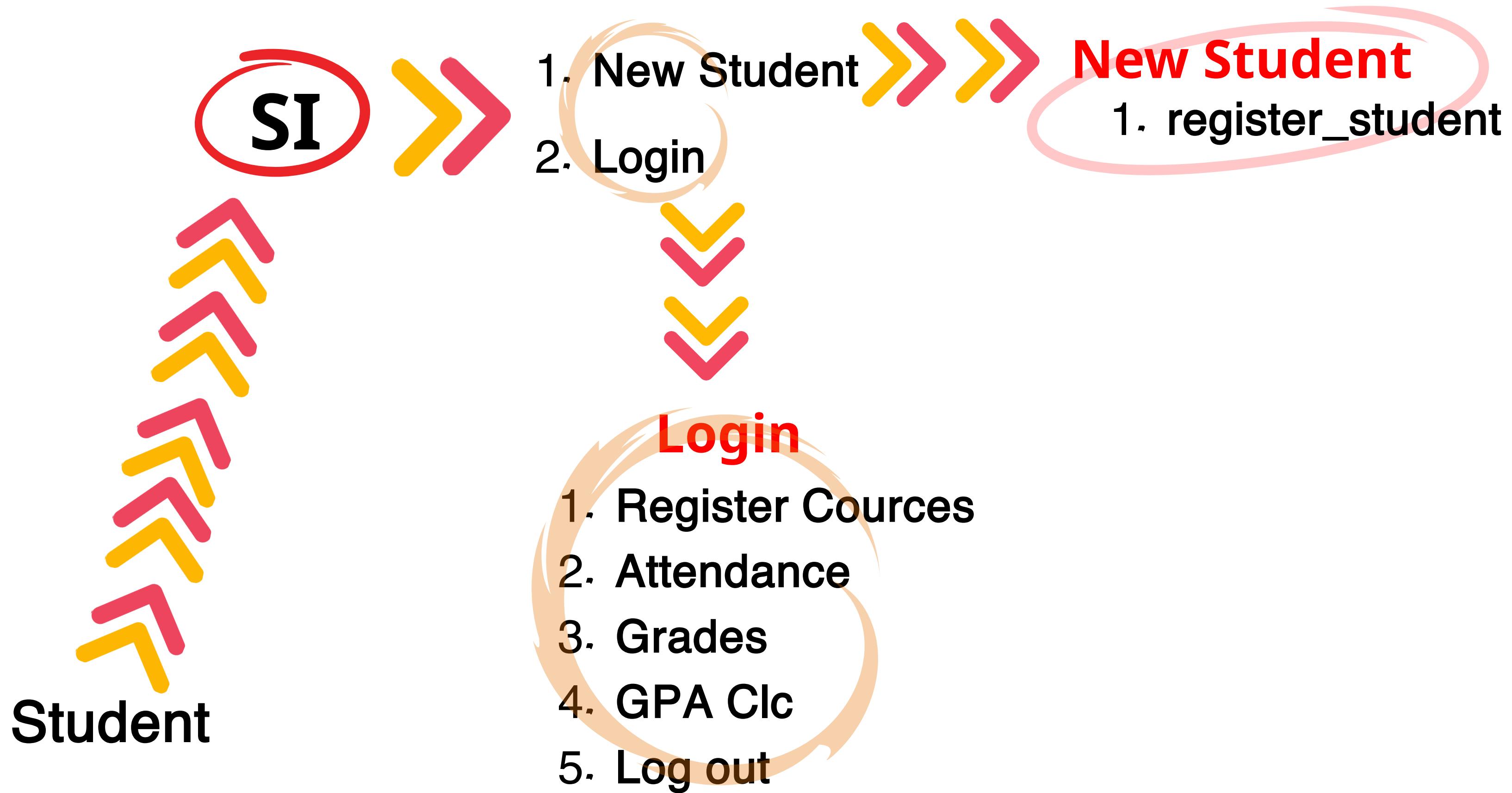
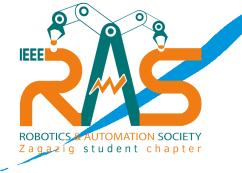
student.py	# Student <b>class</b>
professor.py	# Professor <b>class</b>
course.py	# Course <b>class</b>
college.py	# College <b>class</b>
registration.py	# Registration <b>functions</b>
attendance.py	# Attendance <b>functions</b>
grading.py	# Grading <b>functions</b>
data.json	# Persistent data storage
config.py	# Configuration settings

# CHALLENGES :

## Interface

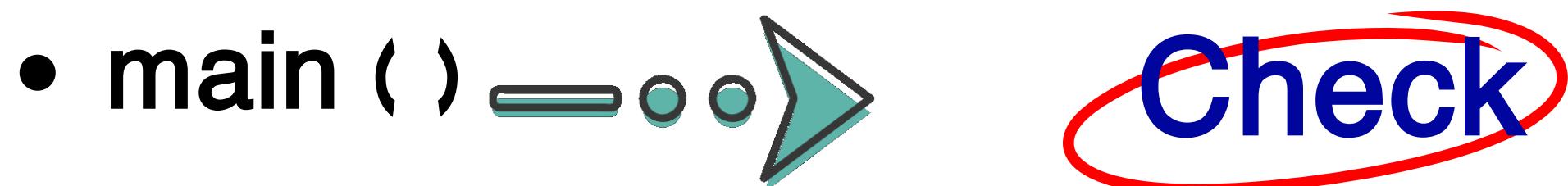
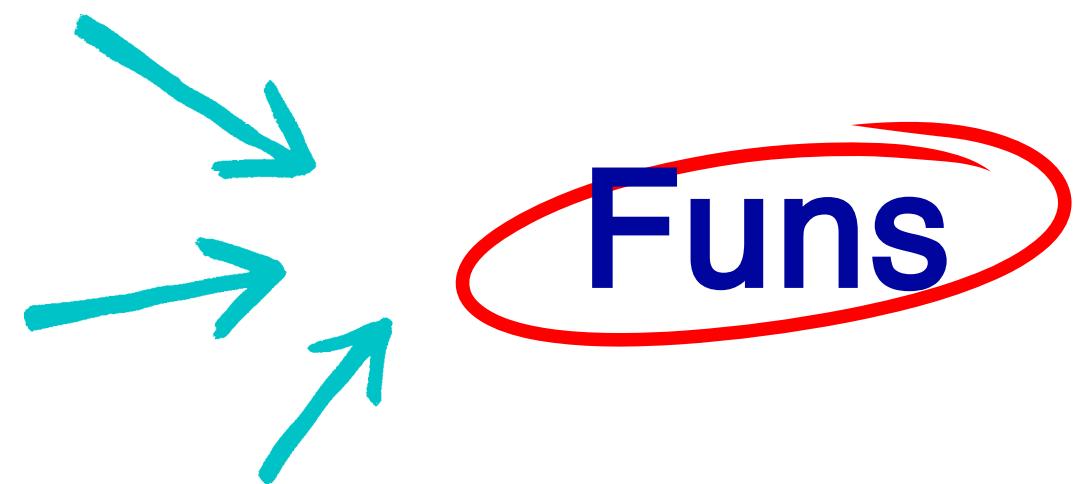


# STUDENT INTERFACE



# Functions & Classes

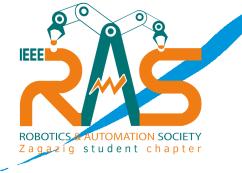
- `load_data()`
- `save_data()`
- `load_student()`
- `login()`



```
while True:
    print ("\n1. Register Courses\n2. Attendance\n3. Grading system\n4. Logout\n")
    choice = input(f"How Can I Help YOU: ")
    if choice == "1":
        register_course
        break
    elif choice == "2":
        # Attendance
        pass
    elif choice == "3":
        print("\n--- Grading System ---")
        grading_system = Grading()
        while True:
            print("\nMain Menu")
            print("1. Assign Grades")
            print("2. Calculate GPA")
            print("3. Exit")
            choice = input("Enter your choice (1-3): ")

            if choice == '1':
                grading_system.__assign_grades__()
                break
            elif choice == '2':
                grading_system.calculate_gpa()
                break
            elif choice == '3':
                print("Goodbye!")
                break
            else:
                print("Invalid choice. Try again.")
                continue
        break

    elif choice == "4":
        print(f"Logging out {student_logged.name}...\n")
        break
    else:
        print("Invalid choice. Try again.")
```



# PROBLEMS:

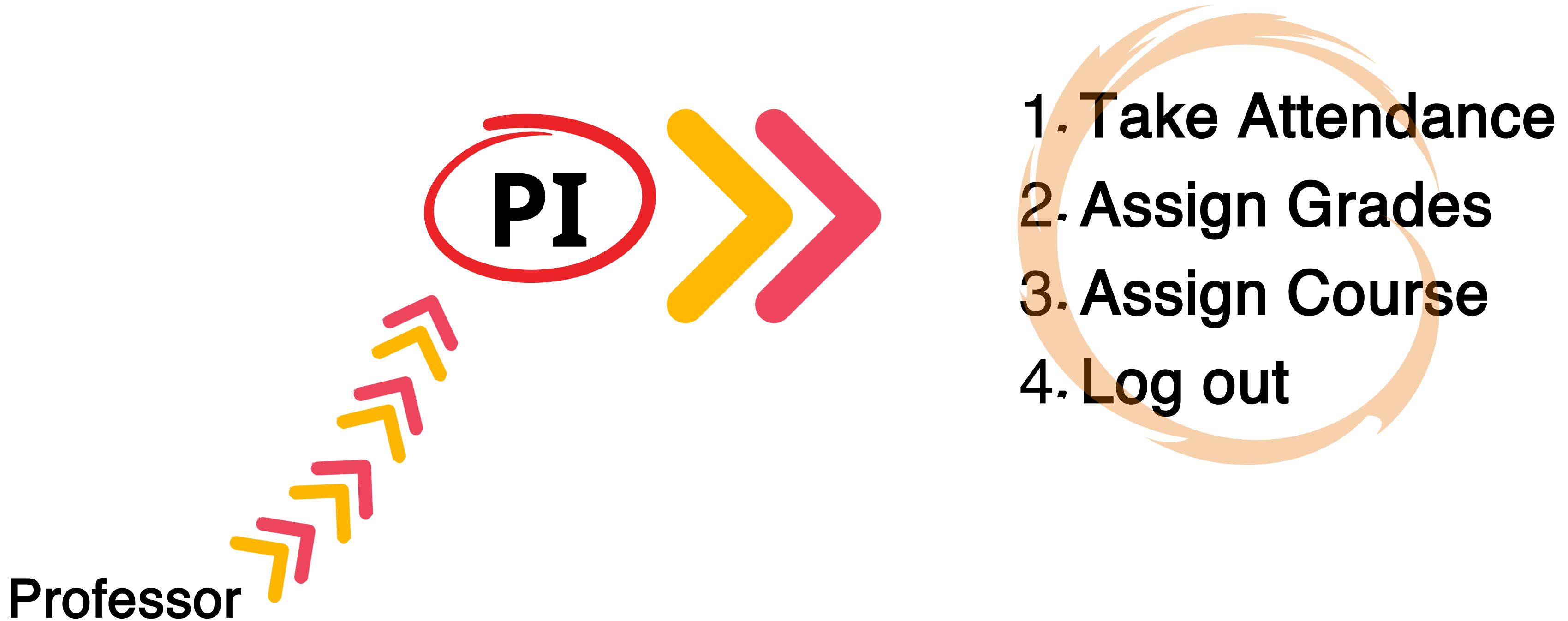
## Register Student , Course

## Attendance

## Grades

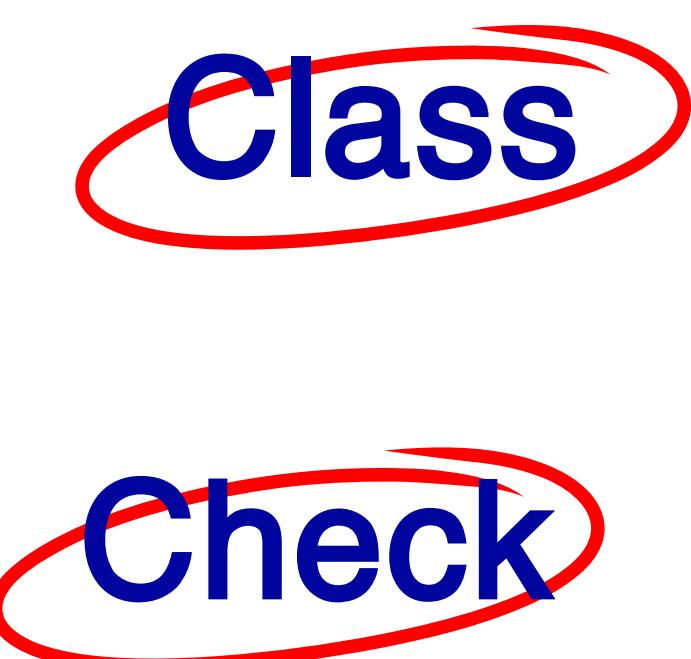
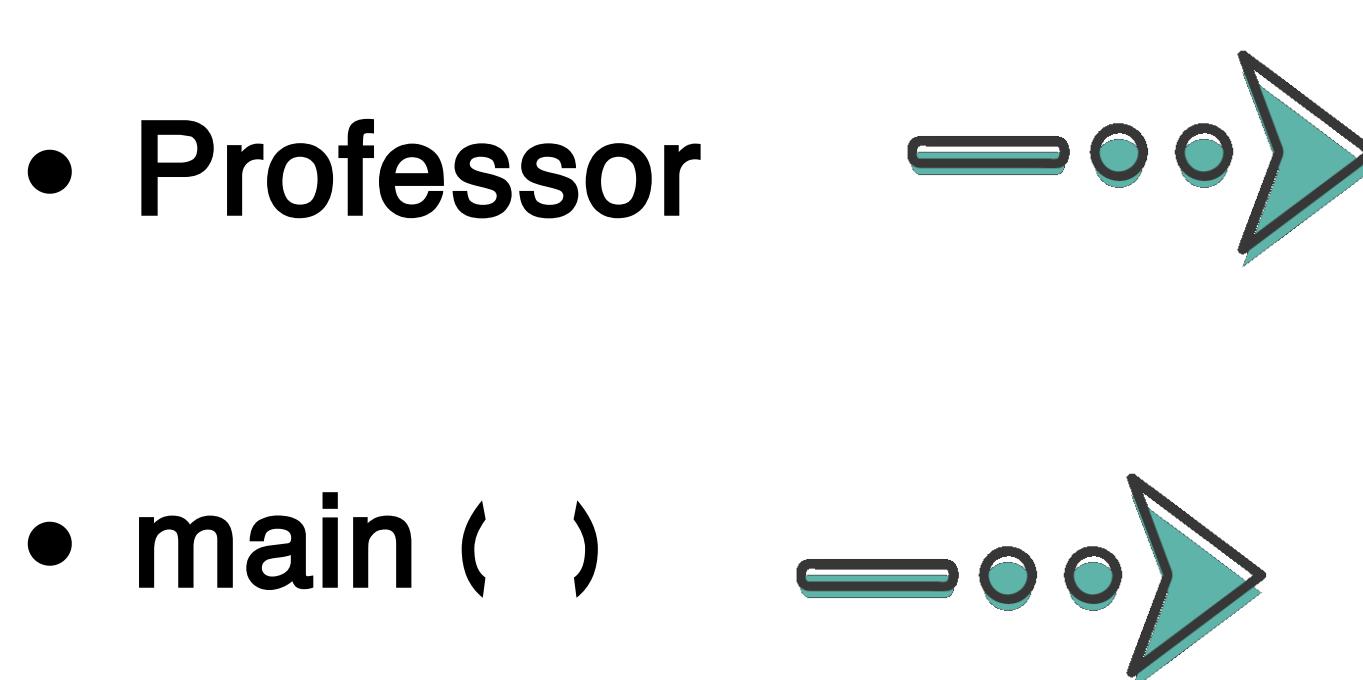
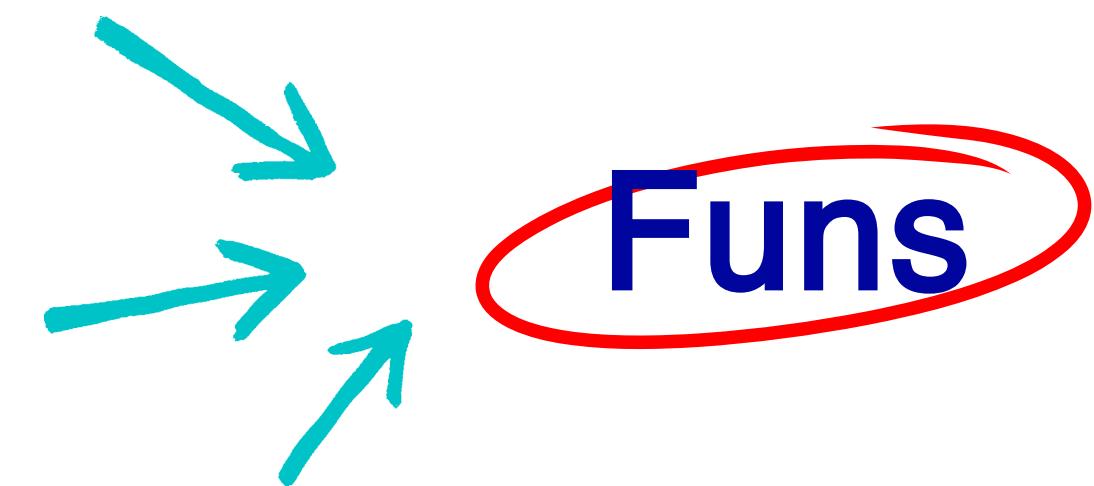
## Gpa

# PROF INTERFACE



# Functions & Classes

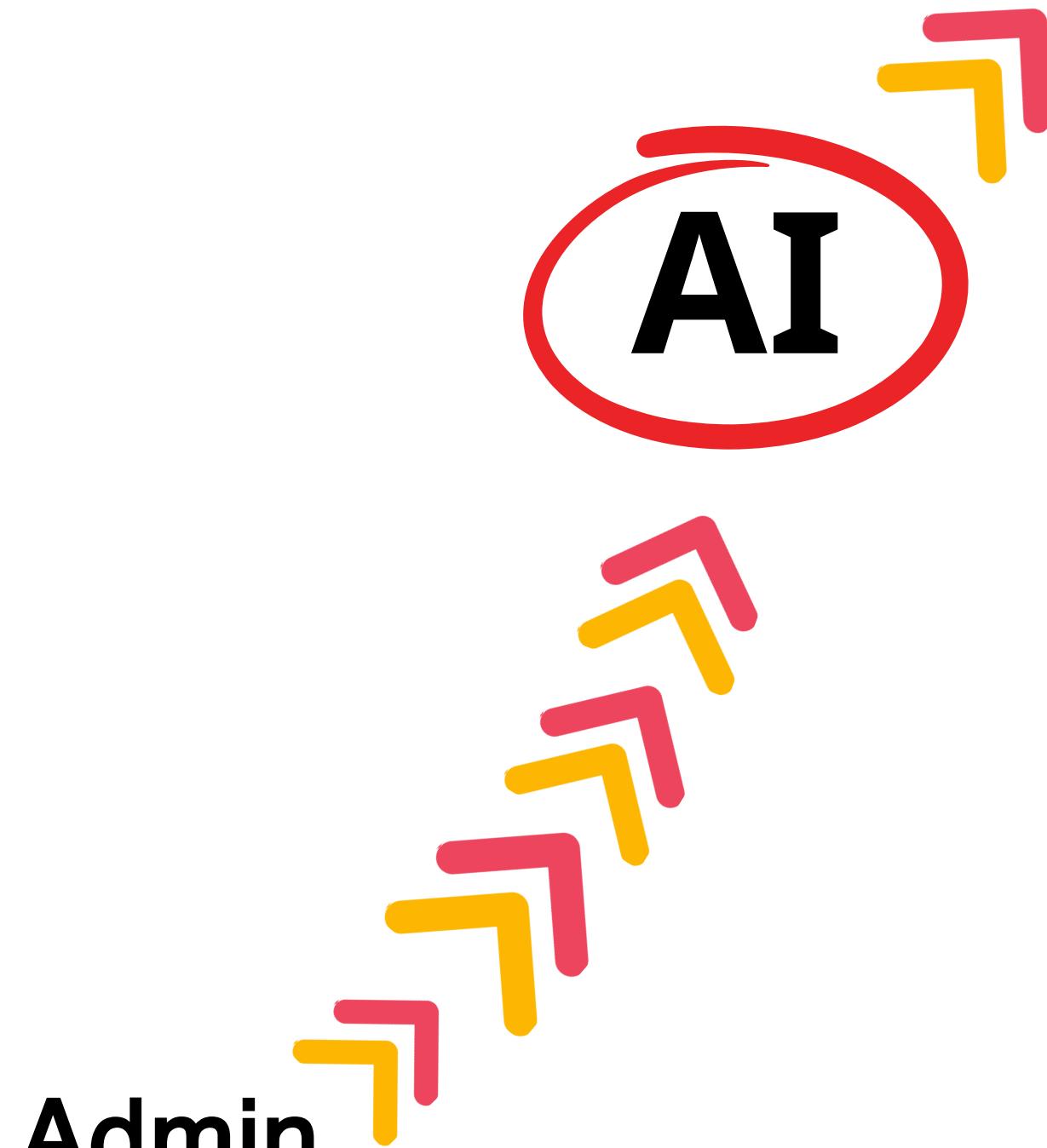
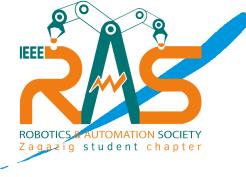
- `load_data()`
- `save_data()`
- `load_professors()`
- `login()`



```
elif choice == "2":\n\n    professors_list= load_professors()\n    for prof in professors_list:\n        print(prof.id,prof.name)\n    print("==== Professor Interface ===")\n    while True:\n        professor_logged = login(professors_list)\n        if professor_logged:\n            while True:\n                print("\nMain Menu")\n                print("1. Take Attendance")\n                print("2. Assign Grades")\n                print("3. Assign Course")\n                print("4. Logout")\n                choice = input("Enter your choice (1-4): ") \n                if choice == '1':\n                    professor_logged.__take_attendance__()\n                    break\n                elif choice == '2':\n                    professor_logged.assign_grades()\n                    break\n                elif choice == '3':\n                    professor_logged.__assign_course__()\n                    break\n                elif choice == '4':\n                    print(f"Logging out Dr. {professor_logged.name}...\n")\n                    break\n            else:\n                print("Invalid choice. Try again.")\n                continue
```



# ADMIN INTERFACE



1. Add Student
2. Add Course
3. Enroll Student in Course
4. Assign Grade
5. Record Attendance

# PROBLEMS:

**Attendance fun in student interface**

**Importing some funs in main**

**Admin Interface**

THANKS

THANKS

THANKS

THANKS