**Design and Implementation of an E - Learning Platform using**

**Object-Oriented Programming Principles**

**Case Study Submitted to Professor Joseph C. Lorilla**

**College of Engineering and Information Technology**

**University of Southern Mindanao**

**OBJECT ORIENTED PROGRAMMING**

**BACHELOR OF SCIENCE IN COMPUTER SCIENCE**

**Lyca Juromay**

**Cloyd Villanueva**

**Khince Patrick Estrera**



**October 2024**

**System Overview**

The E-Learning Platform is a python program designed to manage users, courses, enrollments, assignments, and grades in order to support online learning. Platform administrators, instructors, and students are among the jobs that the system supports. The E-Learning Platform is an integrated software program created to support online learning by managing users, courses, enrollments, assignments, and grades in an organized manner. The platform is defined by a number of fundamental components and important features that improve the educational experience for both teachers and students.

**Essential Elements**

1. **User Roles:**

The platform supports multiple user roles, including Students, Instructors, and Platform Administrators. Each role has specific functionalities tailored to their responsibilities within the system.

1. **Course Management:**

Courses are central to the platform, with functionalities for creating, editing, deleting, and listing courses. Each course is linked to an instructor and contains essential details such as title and description.

1. **Enrollment System:**

The platform allows students to enroll in courses, facilitating their participation in various learning modules. Enrollment management includes adding and removing students from courses.

1. **Assignment Handling:**

Assignments can be created and managed within the courses, allowing instructors to assess student performance. Each assignment is associated with a specific course.

1. **Grading System:**

Instructors can assign grades to students for their completed assignments, enabling performance tracking and feedback.

1. **Lesson Management:**

Lessons can be created for each course, providing structured content delivery to students.

1. **Submission Tracking:**

Students can submit their assignments through the platform, which tracks submissions for grading purposes.

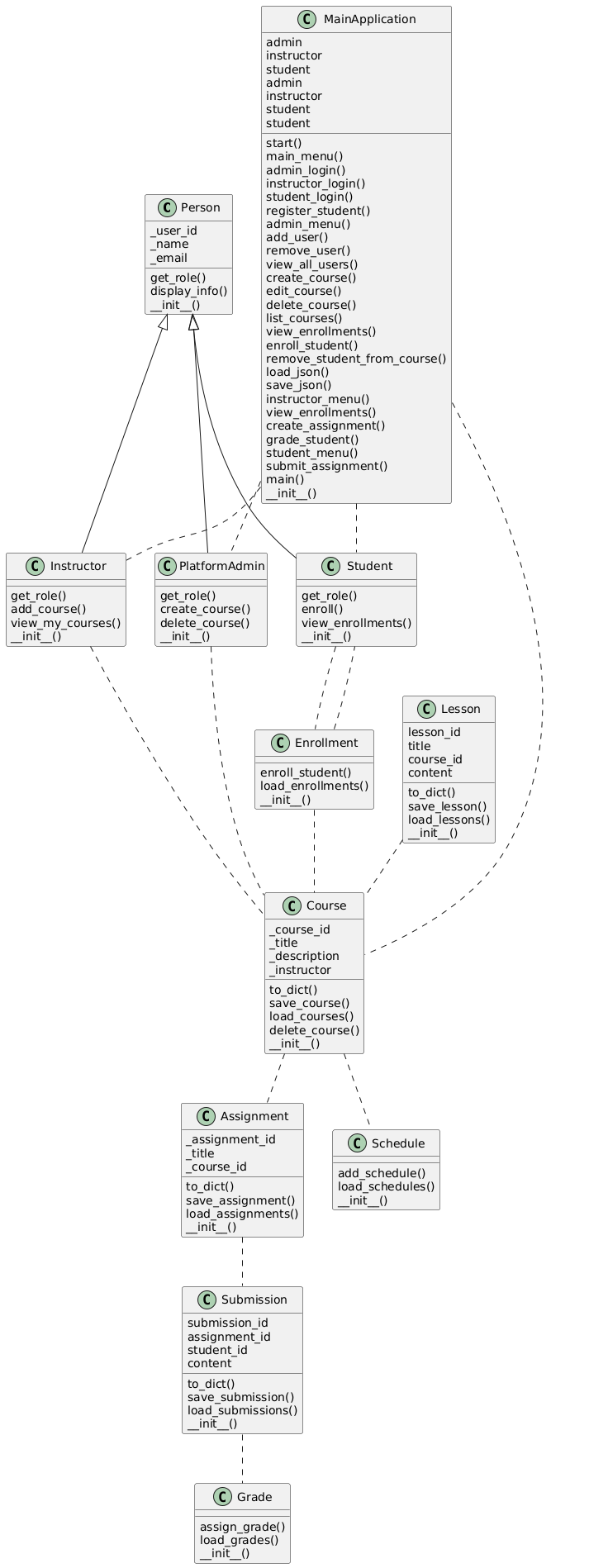
1. **Scheduling:**

The platform includes scheduling functionalities to manage class timings and ensure that students are informed about their course schedules.

**Key Functionalities**

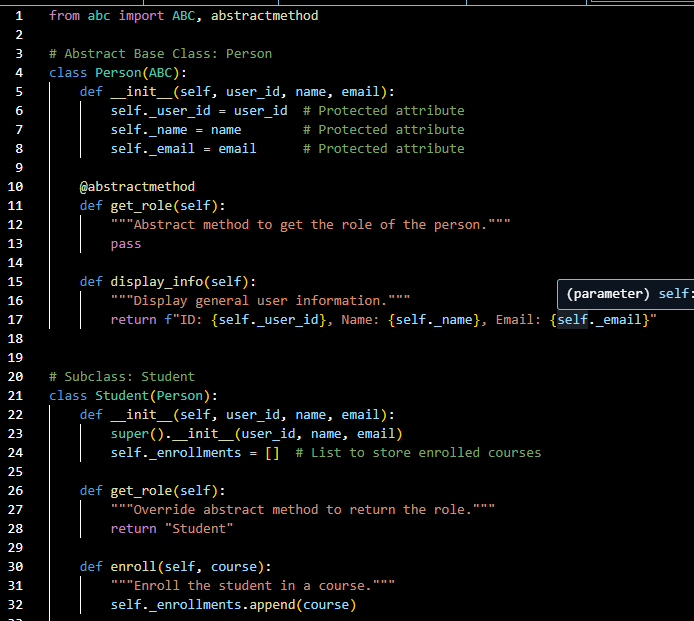
1. **User Authentication:**
   * The system includes robust login mechanisms for different user roles (admin, instructor, student), ensuring secure access to the platform.
2. **Dynamic Course Creation:**
   * Platform administrators can create new courses by specifying details such as title, description, and instructor assignment through a user-friendly interface.
3. **Comprehensive User Management:**
   * Administrators can manage users by adding or removing students and instructors from the system, as well as viewing all registered users.
4. **Interactive Menus:**
   * The application features interactive menus that guide users through various functionalities based on their roles, enhancing user experience.
5. **Data Persistence:**
   * The platform utilizes JSON files for data storage (e.g., users, courses, enrollments), allowing for persistent data management across sessions.
6. **Reporting Features:**
   * Instructors can view their courses and enrolled students easily, while administrators can generate reports on course enrollments and user activity.
7. **Error Handling:**
   * The system incorporates error handling mechanisms to manage invalid inputs or actions gracefully, providing feedback to users when needed.
8. **Extensibility:**
   * The modular design allows for future enhancements and additional features to be integrated into the platform without significant rework of existing code.

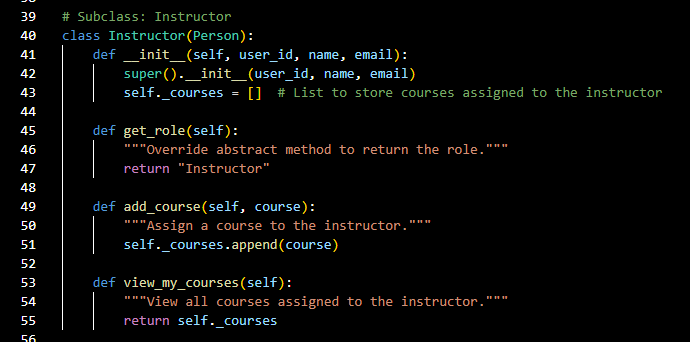
In summary, the E-Learning Platform is designed with a focus on usability and functionality, providing essential tools for managing online education effectively. Its structured approach ensures that all stakeholders—students, instructors, and administrators—can engage with the platform efficiently while maintaining a high standard of educational quality.

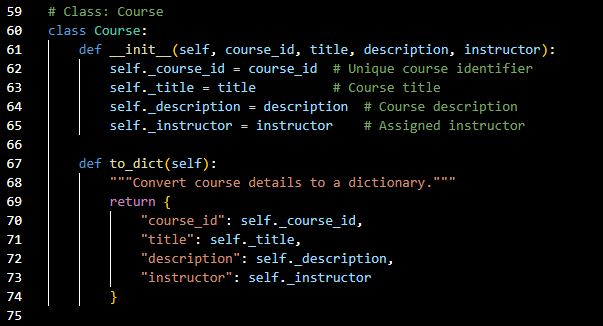
**Class Diagram**

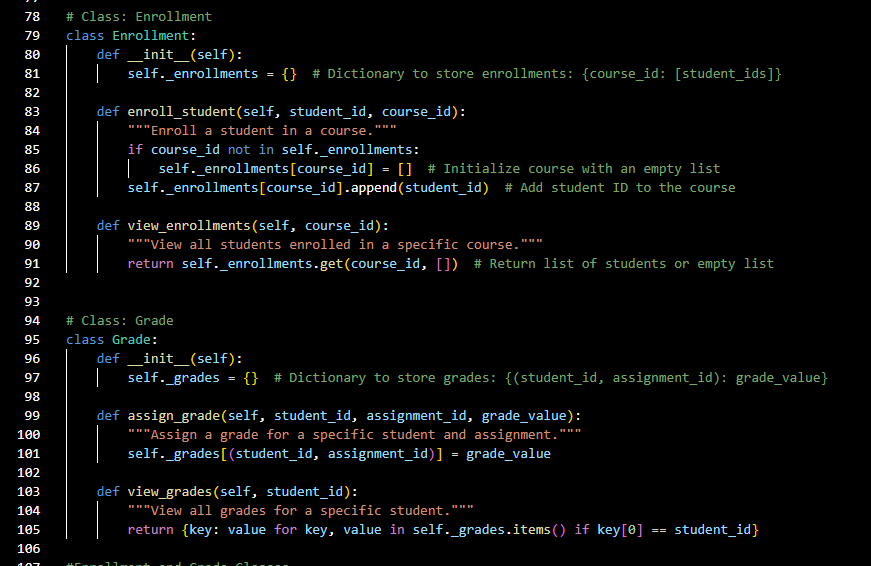
**Explanation of Classes and Methods for E-Learning Platform (\*INSERT PICS OF CODES SNIPPET\*)**

1. **Person (Abstract Class)**
   * **Attributes:**
     + **\_user\_id: Unique identifier for the user.**
     + **\_name: Name of the user.**
     + **\_email: Email address of the user.**
   * **Methods:**
     + **get\_role(): Abstract method that must be implemented in subclasses to return the user's role.**
     + **display\_info(): Returns formatted user details, including ID, name, and email.**
2. **Student (Subclass of Person)**
   * **Attributes:**
     + **\_enrollments: List that stores courses the student is enrolled in.**
   * **Methods:**
     + **get\_role(): Returns "Student" as the role of the user.**
     + **enroll(course): Adds a specified course to the student's list of enrollments.**
     + **view\_enrollments(): Returns a list of courses that the student is currently enrolled in.**
3. **Instructor (Subclass of Person)**
   * **Attributes:**
     + **\_courses: List that stores courses taught by the instructor.**
   * **Methods:**
     + **get\_role(): Returns "Instructor" as the role of the user.**
     + **add\_course(course): Adds a specified course to the instructor's list of courses.**
     + **view\_my\_courses(): Retrieves and displays all courses taught by the instructor.**
4. **PlatformAdmin (Subclass of Person)**
   * **Methods:**
     + **get\_role(): Returns "Platform Admin" as the role of the user.**
     + **create\_course(): Prompts for course details and creates a new course, saving it to a file.**
     + **delete\_course(course\_id): Deletes a specified course from the system.**
5. **Course**
   * **Attributes:**
     + **\_course\_id: Unique identifier for the course.**
     + **\_title: Title of the course.**
     + **\_description: Description of the course content.**
     + **\_instructor: Reference to the instructor object associated with this course.**
   * **Methods:**
     + **to\_dict(): Converts course attributes into a dictionary format for easy storage and retrieval.**
     + **save\_course(course\_data): Saves a new or updated course to a JSON file.**
     + **load\_courses(): Loads all courses from a JSON file into memory.**
     + **delete\_course(course\_id): Deletes a specific course based on its ID from storage.**
6. **Enrollment**
   * **Methods:**
     + **enroll\_student(student\_id, course\_id): Enrolls a student in a specified course and updates enrollment records in storage.**
     + **load\_enrollments(): Loads current enrollment data from a JSON file.**
7. **Assignment**
   * **Attributes:**
     + **\_assignment\_id: Unique identifier for an assignment.**
     + **\_title: Title of the assignment.**
     + **\_course\_id: Identifier for the associated course.**
   * **Methods:**
     + **to\_dict(): Converts assignment attributes into dictionary format for storage purposes.**
     + **save\_assignment(assignment\_data): Saves assignment data to a JSON file.**
     + **load\_assignments(): Loads assignments from storage.**
8. **Grade**
   * **Methods:**
     + **assign\_grade(student\_id, assignment\_id, grade\_value): Assigns a grade to a student for a specific assignment and updates grade records in storage.**
     + **load\_grades(): Loads existing grades from a JSON file.**
9. **Schedule**
   * **Methods:**
     + **add\_schedule(course\_id, timing): Adds scheduling information for a specified course and saves it to storage.**
     + **load\_schedules(): Loads scheduling data from storage.**
10. **Submission**
    * **Attributes:**
      + **submission\_id: Unique identifier for each submission instance.**
      + **assignment\_id: Identifier for the associated assignment.**
      + **student\_id: Identifier for the student making the submission.**
      + **content: The actual content submitted by the student for evaluation.**
    * **Methods:**
      + **to\_dict(): Converts submission attributes into dictionary format for easy handling and storage.**
      + **save\_submission(submission\_data): Saves submission data to a JSON file.**
      + **load\_submissions(): Loads submissions from storage.**
11. **Lesson**
    * **Attributes:**
      + **lesson\_id: Unique identifier for each lesson instance.**
      + **title: Title of the lesson content.**
      + **course\_id: Identifier linking to the associated course.**
      + **content: The actual lesson material provided to students.**
    * **Methods:**
      + **to\_dict(): Converts lesson attributes into dictionary format for easy handling and storage.**
      + **save\_lesson(lesson\_data): Saves lesson data to a JSON file.**
      + **load\_lessons(): Loads lessons from storage.**
12. **MainApplication**
    * **Attributes:**
      + **admin, instructor, and student attributes used to manage logged-in users within the application context.**
    * **Methods:**
      + **Various methods to handle user logins, display menus, manage administrative tasks, create or edit courses, and facilitate user interactions with the system.**

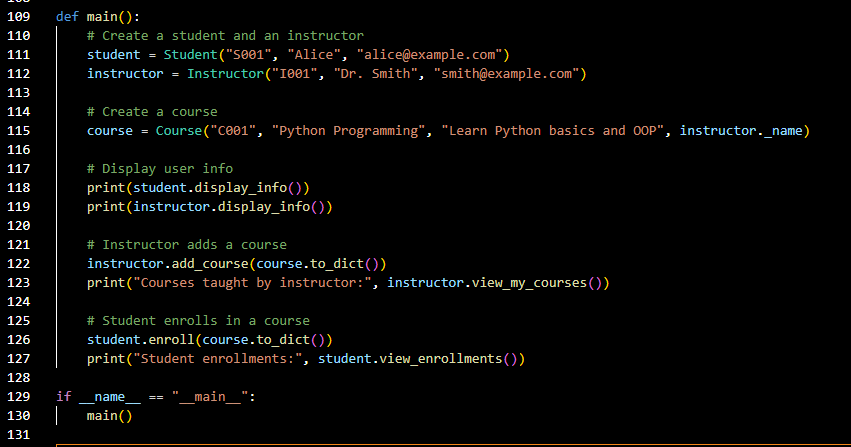
**Output: (\*SS OF OUTPUTS\*)  
  
CODE SNIPPETS   
  
Abstract Person Class and Student Subclass :**

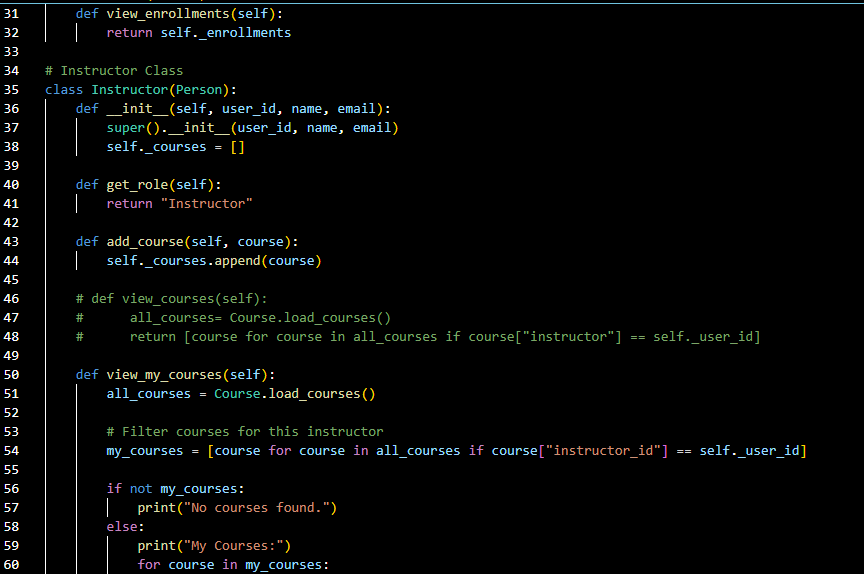
**Instructor Subclass :  
**

**Course Class :**

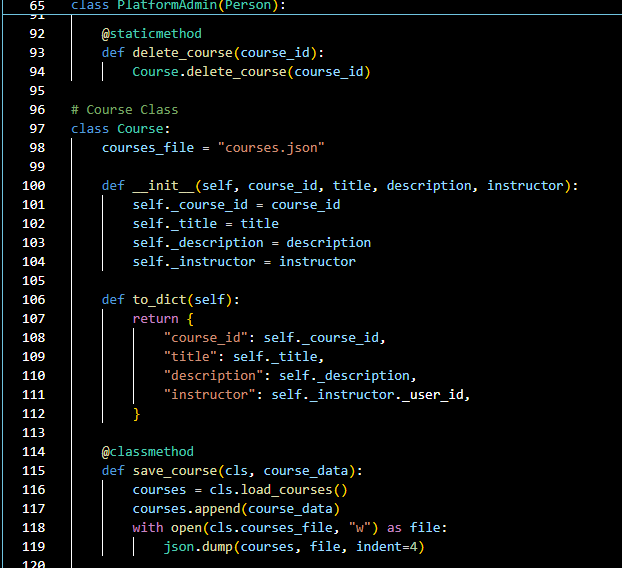
**Enrollment and Grade Classes :  
**

**Main Application Flow :**

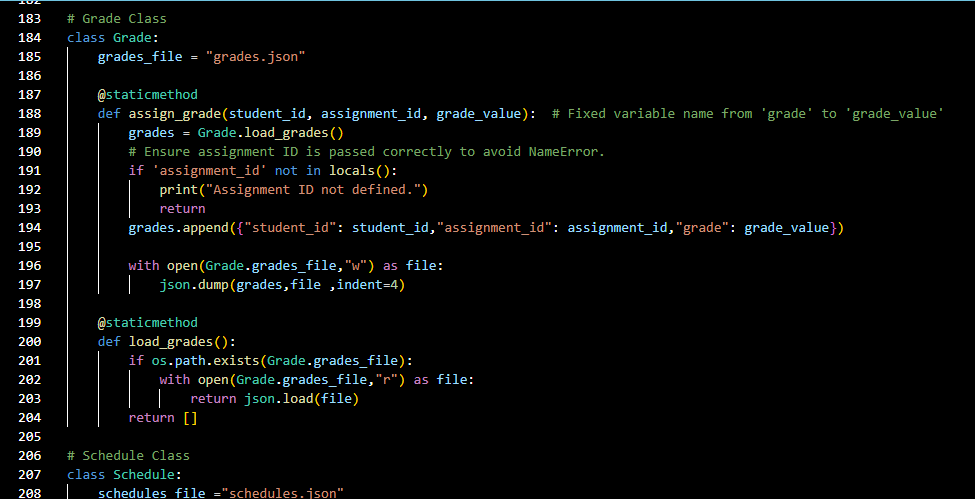
****

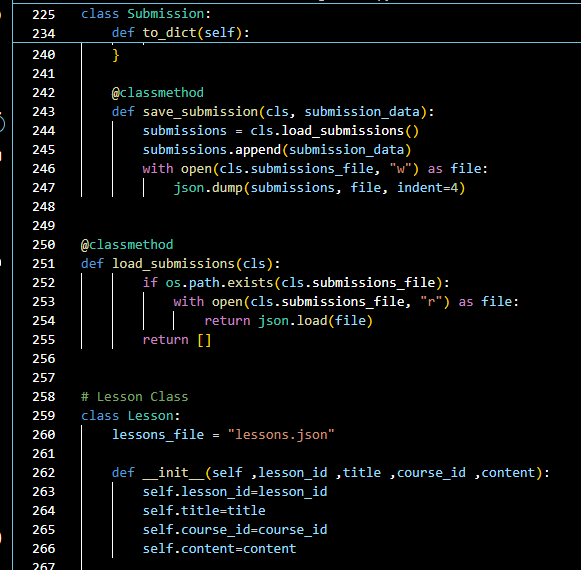
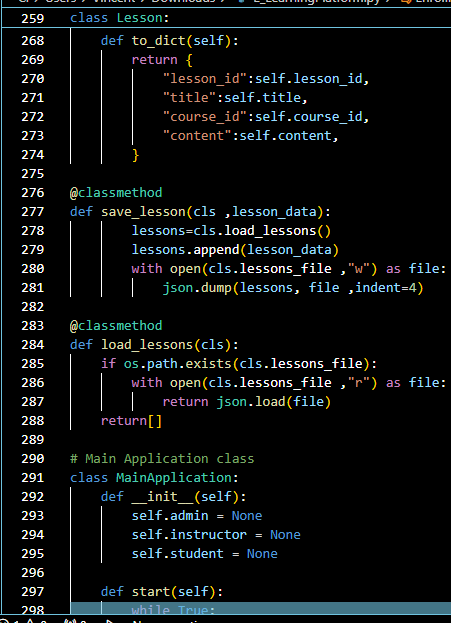
***E-LearningPlatform Code :  
***

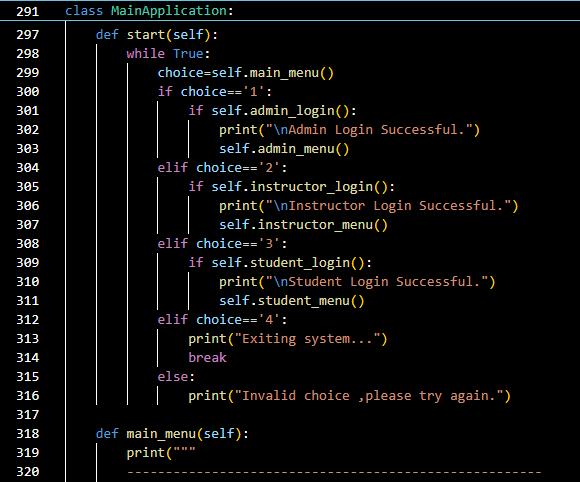
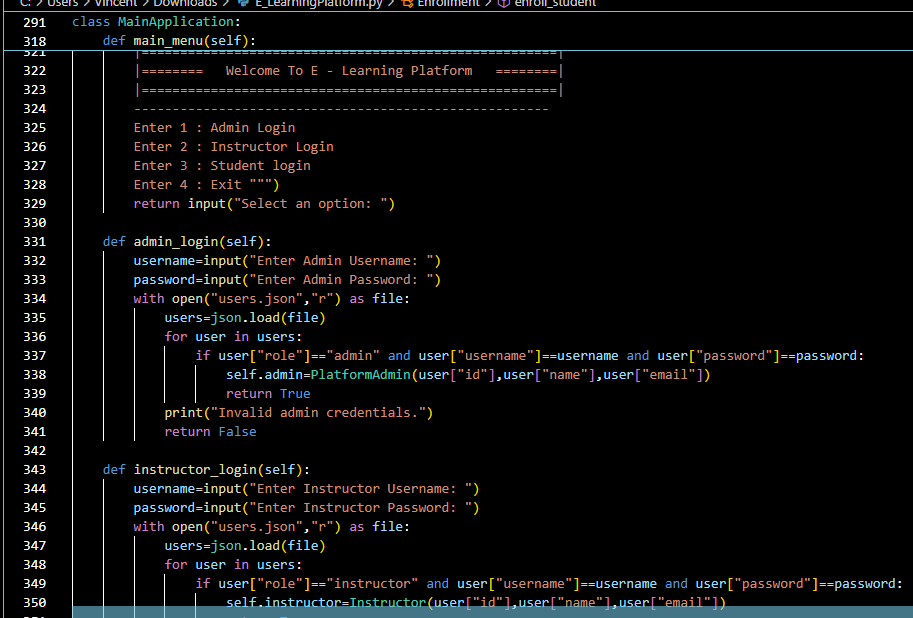
******

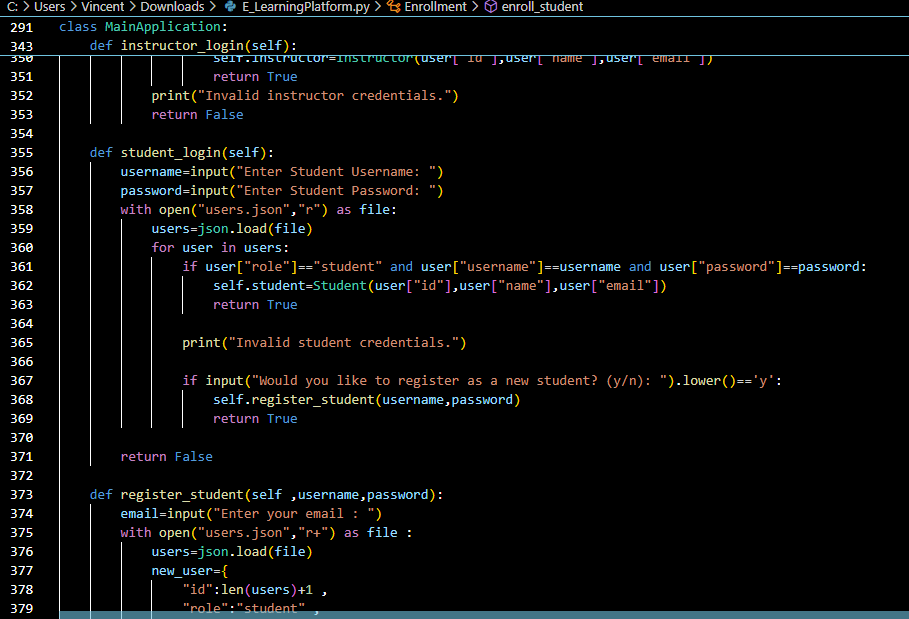
******

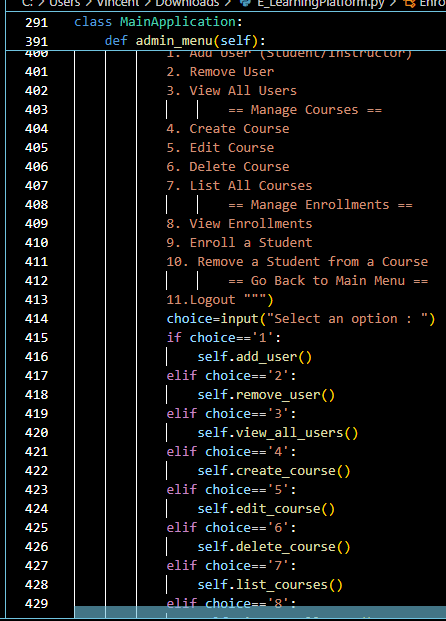
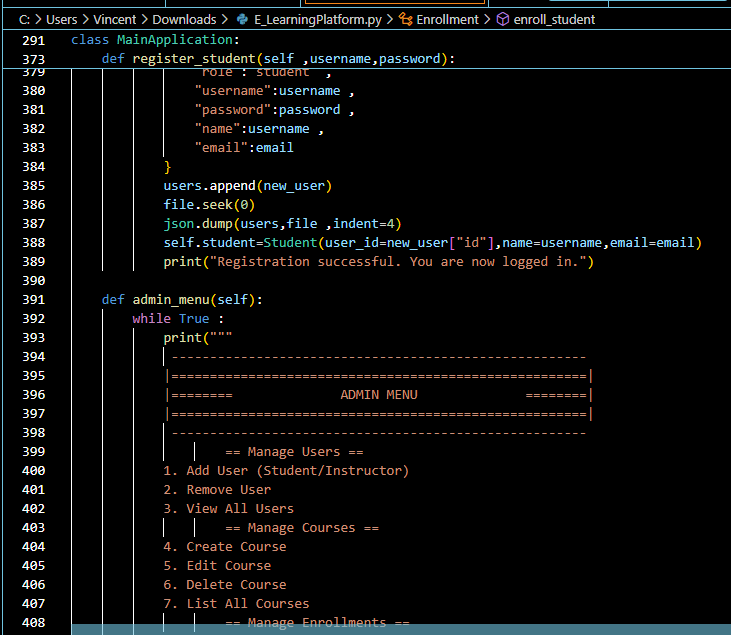
******

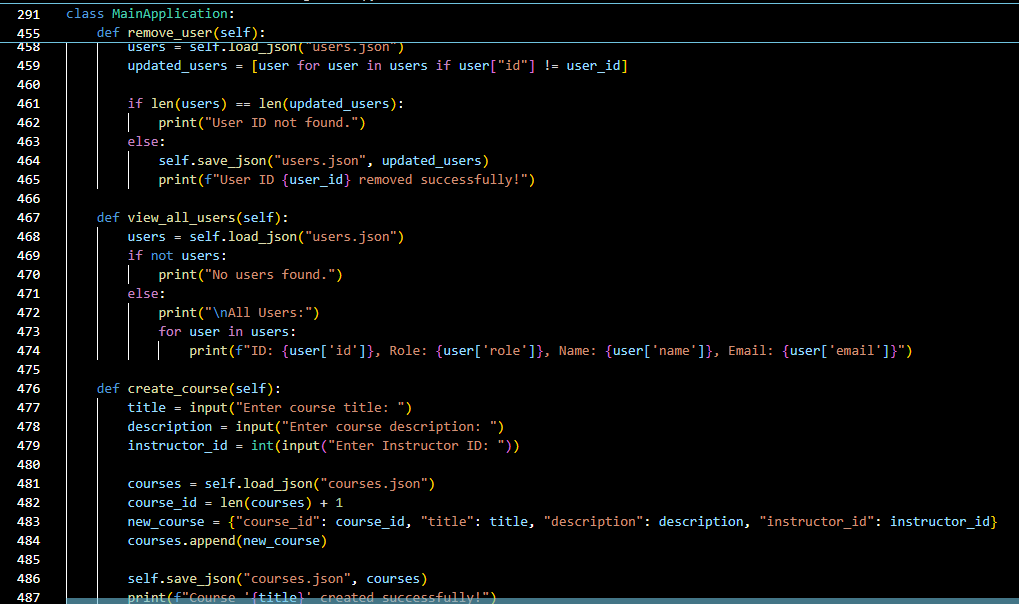
******

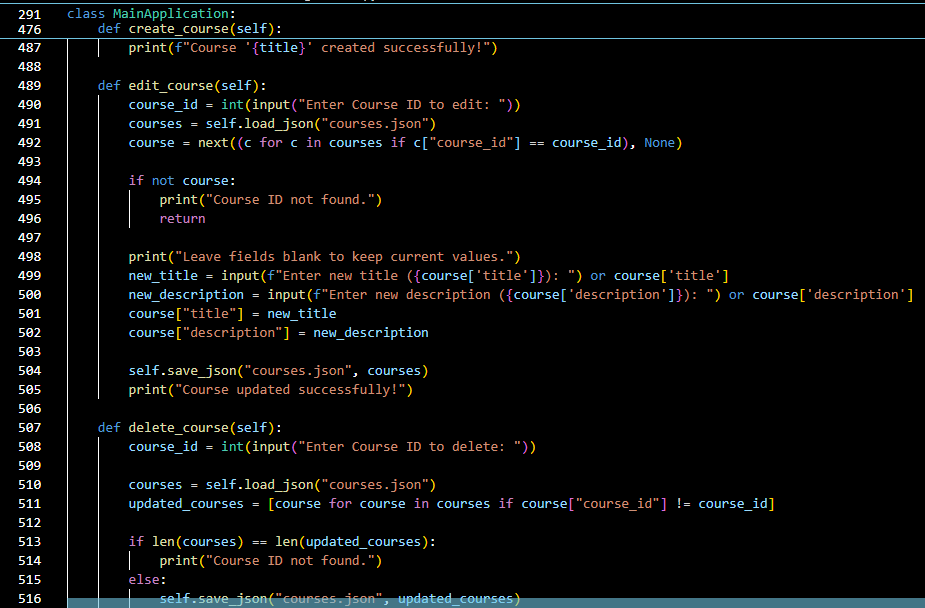
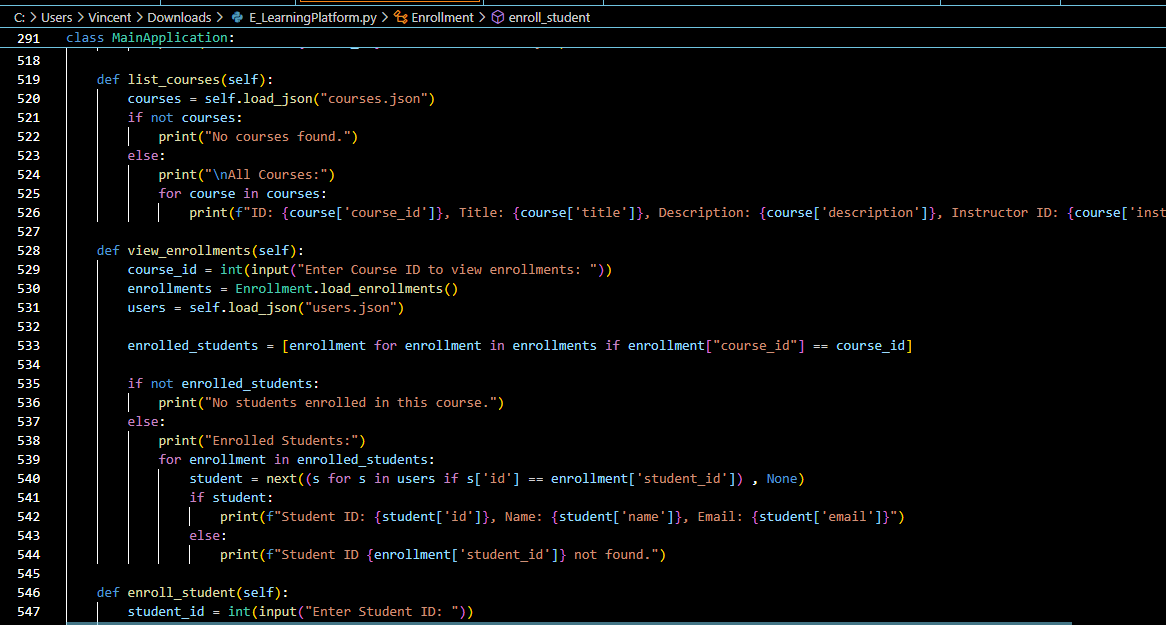
******

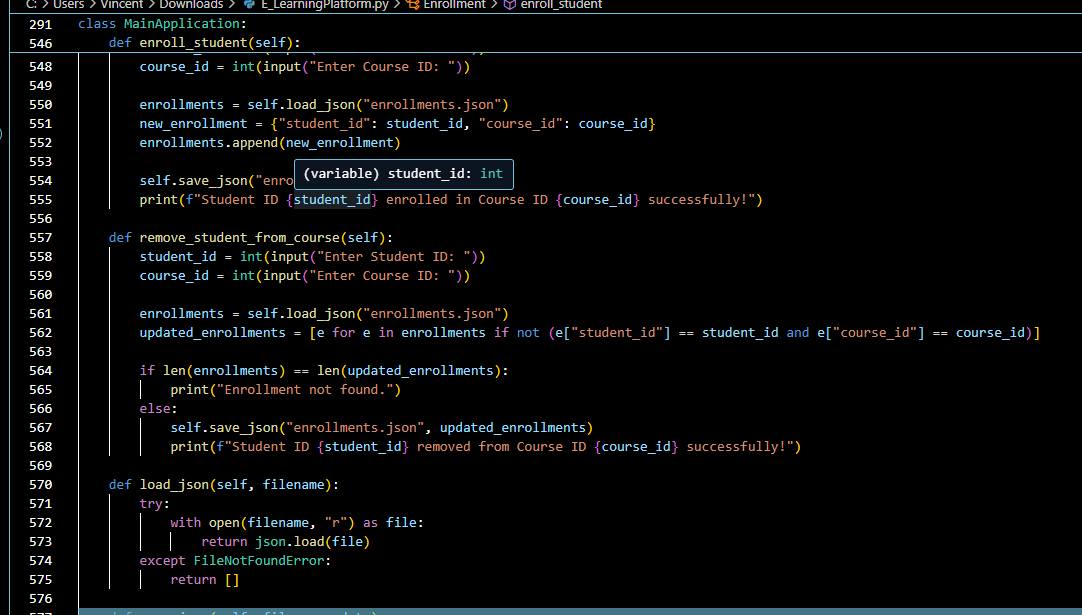
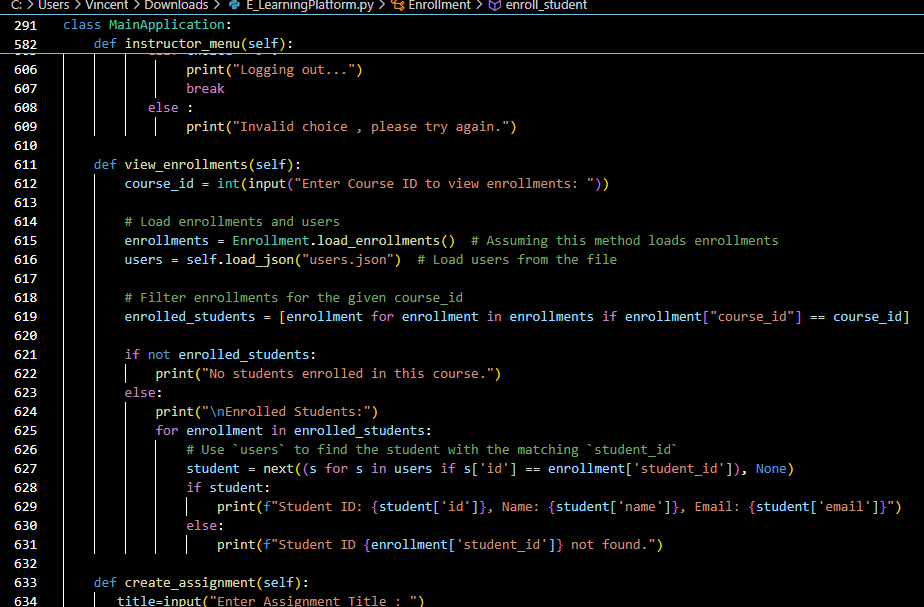
******

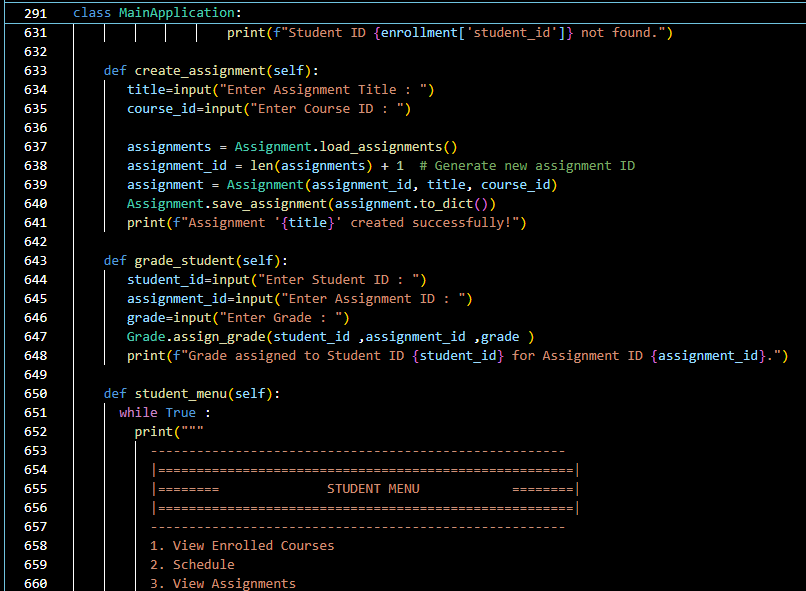
******

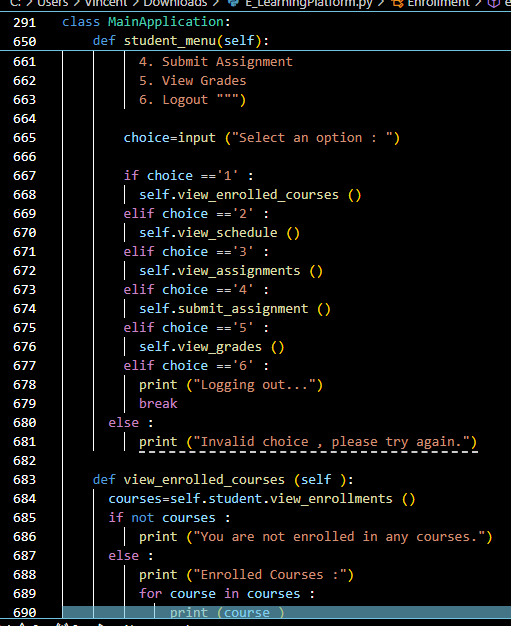
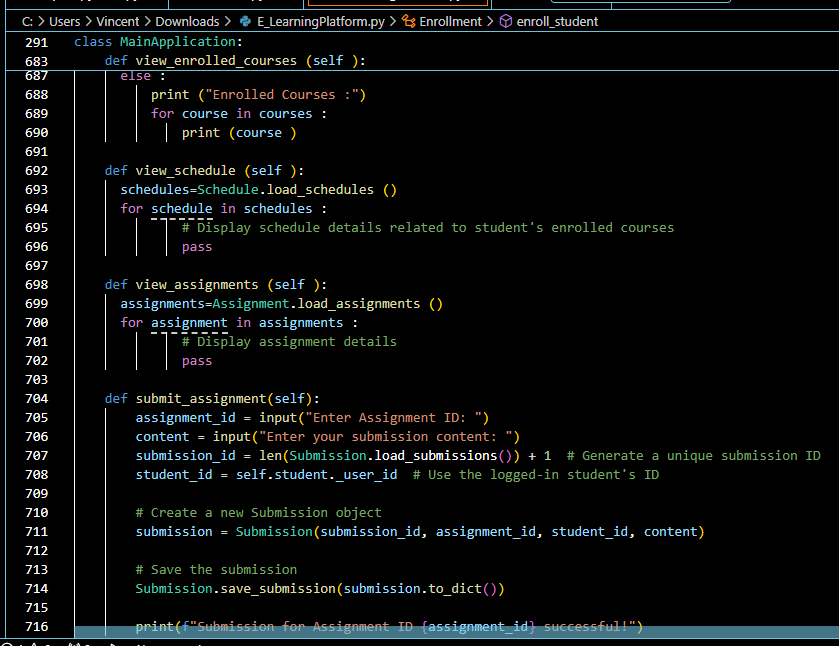
******

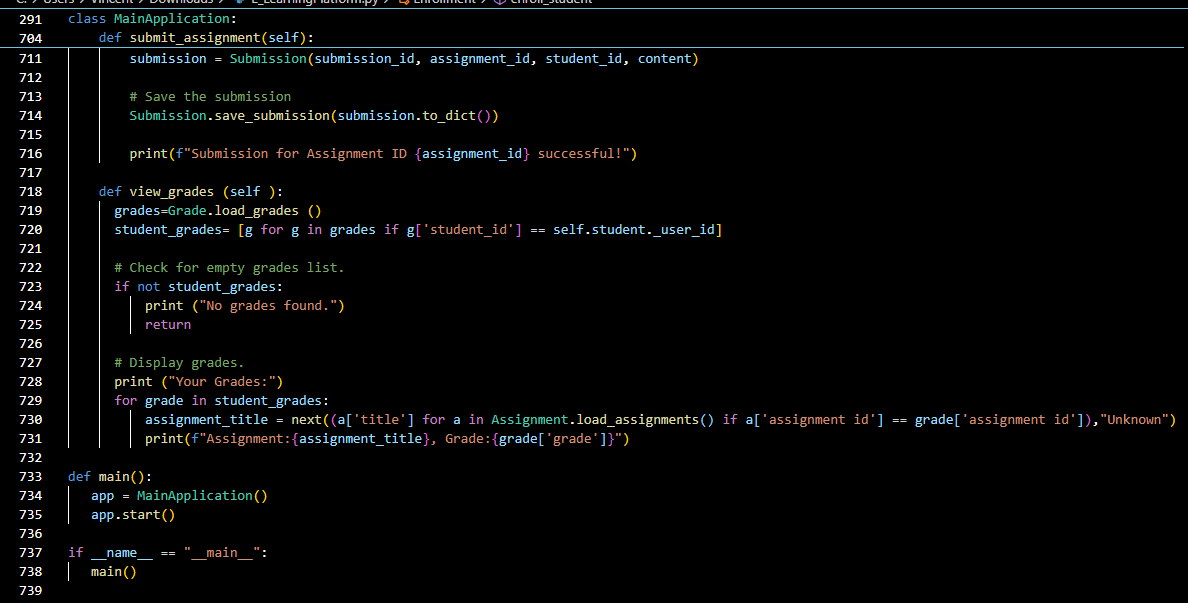
******

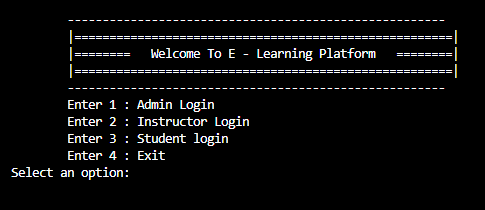
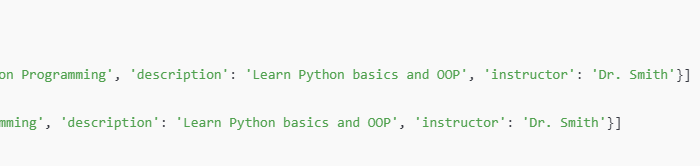
******

***zz***

******

******

******

******