

Student Management System

This project is a simple Student Management System that allows for managing students, courses, assessments, grades, and instructors. The system interacts with a MySQL database to store and retrieve information.

Table of Contents

- Installation
- Usage
 - Students
 - Adding a Student
 - Enrolling a Student in a Course
 - Tracking Student Progress
 - Courses
 - Adding a Course
 - Updating a Course
 - Removing a Course
 - Searching for Courses
 - Sorting Courses
 - Assessments
 - Creating an Assessment
 - Grades
 - Inputting a Grade
 - Viewing Grades
 - Instructors
 - Adding an Instructor
 - Updating an Instructor
 - Removing an Instructor
 - Assigning an Instructor to a Course
- Database Schema
- Contributing
- License
- Dependencies
- Contact

Installation

To install and set up this project, follow these steps:

1. Clone the repository:

Bash

```
git clone https://github.com/yourusername/student-management-system.git
cd student-management-system
```

Use code [with caution](#).

content_copy

2. Set up the virtual environment:

It's recommended to use a virtual environment to isolate project dependencies. Here's how to set one up:

- **On Linux/macOS:**

Bash

```
python -m venv venv  
source venv/bin/activate
```

Use code [with caution](#).

content_copy

- **On Windows:**

Bash

```
python -m venv venv  
venv\Scripts\activate
```

Use code [with caution](#).

content_copy

3. Install the required dependencies:

The project relies on certain Python libraries. Install them using the following command:

Bash

```
pip install -r requirements.txt
```

Use code [with caution](#).

content_copy

4. Configure the database:

Ensure you have a MySQL database set up and running. This README assumes the following credentials for your database connection:

- Host: localhost
- User: root
- Password: shreejit (Replace with your actual password)
- Database: EduSchema (Replace with your desired database name)

Important: These credentials are currently used in the `create_connection` function within the `database.py` file. Update them with your own database details if necessary.

Usage

This section explains how to use the various functionalities offered by the Student Management System.

Students

Adding a Student

Use the `add_student` function to add a new student to the database. Here's an example:

Python

```
from student_management import add_student

add_student('John', 'Doe', 'john.doe@example.com', '123-456-7890', '2000-01-01')
```

Use code [with caution](#).

content_copy

This code snippet adds a student named John Doe with the provided email address, phone number, and date of birth.

Enrolling a Student in a Course

Use the `enroll_student` function to enroll an existing student in a course. Here's an example:

Python

```
from student_management import enroll_student

enroll_student(1, 101) # Where 1 is the student ID and 101 is the course ID
```

Use code [with caution](#).

content_copy

This code snippet enrolls the student with ID 1 in the course with ID 101. Remember to replace the IDs with the actual student and course IDs.

Tracking Student Progress

The `track_progress` function allows you to monitor a student's performance in a specific course. Here's how to use it:

Python

```
from student_management import track_progress

progress = track_progress(1, 101) # Where 1 is the student ID and 101 is the course ID
for assessment_name, score in progress:
    print(f"{assessment_name}: {score}")
```

Use code [with caution](#).

content_copy

This code retrieves the student's progress in course 101 (replace IDs as needed) and iterates through a list of assessment names and corresponding scores.

Courses (and so on)

The system offers functionalities for managing courses, assessments, grades, and instructors. Refer to the following sections for details on how to use them:

- **Courses**
 - Adding a Course
 - Updating a Course
 - Removing a Course
 - Searching for Courses
 - Sorting Courses
- **Assessments**
 - Creating an Assessment
- **Grades**
 - Inputting a Grade
 - Viewing Grades
- **Instructors**
 - Adding an Instructor
 - Updating an Instructor
 - Removing an Instructor