A Project Report On…



**College Management System**

Developed By…

***Ayushi Mehta***

Submitted To..

*Geetanjali College of Computer Science and Commerce (BBA)*

Saurashtra University Rajkot

***2025***

***Prof.***

1

### Acknowledgement

I would like to express my deepest gratitude to all those who contributed to the successful completion of this project.

First and foremost, I would like to thank all the faculties, for their invaluable guidance, support, and encouragement throughout the development of this college management system. Their expertise and insights were crucial in shaping the project and ensuring its success.

I would also like to extend my sincere thanks to the **Geetanjali College of Computer Science and Commerce (BBA)** for providing the resources and environment needed to complete this project. The faculty members of the Computer Science Department were always available for consultation and provided me with the necessary academic guidance.

A special mention goes to my friends and classmates for their constant encouragement and for being there to discuss ideas, troubleshoot issues, and share knowledge. Their support was instrumental in overcoming the challenges faced during the project.

Thank you all for your contributions to this work. I am truly grateful for your support.

**Ayushi Mehta**

The **College Management System** is an essential tool for modern educational institutions, designed to streamline and manage various administrative tasks effectively. This project, developed using Python, aims to simplify the complex processes involved in managing student data, faculty information, course details, attendance, and other related administrative functions.

In today’s fast-paced world, educational institutions need efficient systems to handle the increasing amount of data generated daily. Traditional methods of managing college operations are not only time-consuming but also prone to errors. Recognizing this challenge, I embarked on developing a comprehensive solution that would address these issues by automating and integrating various administrative tasks.

This project has been a significant learning experience, allowing me to explore the practical applications of Python programming in real-world scenarios. It also provided an opportunity to enhance my understanding of database management, user interface design, and system integration, all of which are critical components in the development of robust software solutions.

The College Management System project is designed with a user-friendly interface, making it accessible for users with varying levels of technical expertise. It ensures data accuracy, reduces redundancy, and improves overall efficiency in managing college operations.

I hope that this project not only serves as a useful tool for educational institutions but also as a stepping stone for future developments in the field of educational software. The experience and knowledge gained during this project have been invaluable, and I look forward to applying these skills in future endeavors.

**Ayushi Mehta**

****

|  |  |  |
| --- | --- | --- |
| ***Sr. No.*** | ***Name*** | ***Page No.*** |
| 1. | Project Profile | 5 |
| 2. | Project Requirement | 6 |
| 3. | Technology Requirement | 7 |
| 4. | ER Diagram | 8 |
| 5. | SDLC | 9 |
| 6. | DFD | 10 |
| 7. | Data Dictionary | 12 |
| 8. | Screenshots | 16 |
| 9. | Future Requirements of Project | 24 |
| 10. | Limitations of Project | 25 |
| 11. | Webliography | 26 |

# Objectives

To develop a comprehensive College Management System using Python that streamlines and automates various administrative processes, including student enrollment, course management, attendance, and faculty information.



# Key Features

Key features for a College Management System project in Python includes adding students, faculties and courses. It also allows admin to modify the records of students and faculties. It also allows admin to modify course details. It allows admins to take students’ and faculties’ attendance.

**Project:** College Management System

**Technology:** Python



**Database:** MySQL

#### Functionality:

**User Authentication:**

* Login for administrators.

#### Student Management:

* Register new students.
* Modify existing student records (e.g., personal information, academic details, course enrollment).

#### Faculty Management:

* Register new faculty members.
* Modify existing faculty records (e.g., personal information, teaching assignments, qualifications, etc.).

#### Course Management:

* Create and manage course information (e.g., course name, description, prerequisites).
* Modify existing course details.

**Student Attendance:**

* Attendance of student is taken by course.

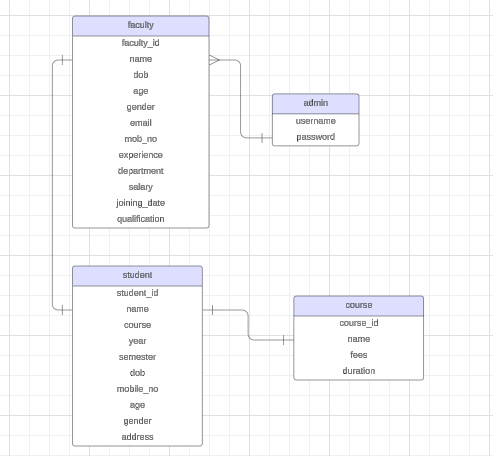
**Faculty Attendance:**

* Attendance of faculties can be taken.

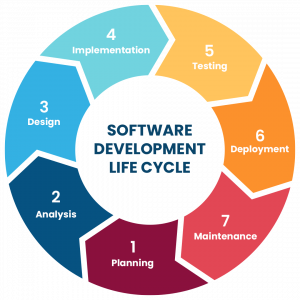


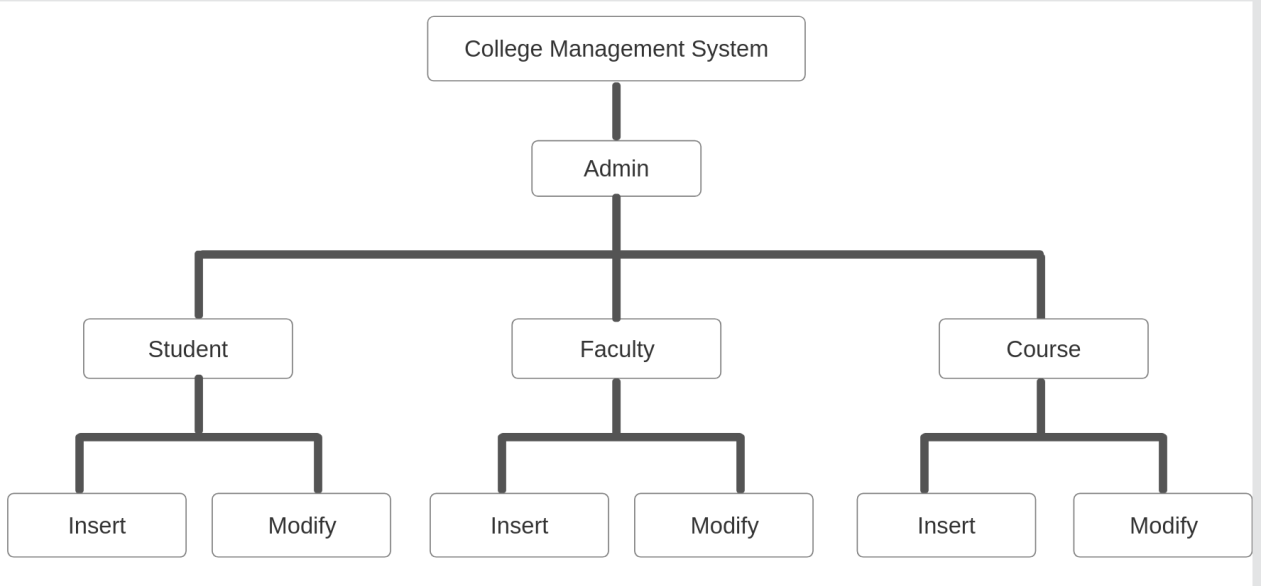
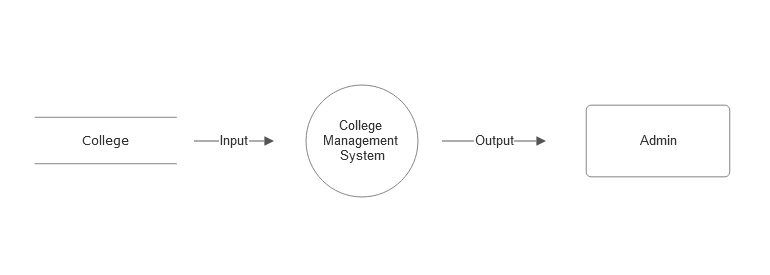
|  |  |
| --- | --- |
| **Operating System** | Windows 8 or higher |
| **Development Environment** | VS Code |
| **Database** | MySQL |
| **Python Version** | >= 3.0 |
| **Language** | Python |

[ER Diagram]



[SDLC]





**Database used:** MySQL

**Database name:** college\_mgt

**List of Tables:**

1. admin
2. student
3. faculty
4. course
5. stud\_attendance
6. fac\_attendance

***Data Dictionary***



**admin**

**student**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1. | id | INT(11) | PRIMARY KEY  Not Null | ID |
| 2. | username | VARCHAR(50) | Not Null | Username |
| 3. | password | VARCHAR(50) | Not Null | Password |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1. | stud\_id | INT(11) | PRIMARY KEY  Not Null | Student ID |
| 2. | name | VARCHAR(50) | Not Null | Name |
| 3. | course | VARCHAR(50) | Not Null | Course |
| 4. | year | VARCHAR(50) | Not Null | Year |
| 5. | semester | VARCHAR(50) | Not Null | Semester |
| 6. | dob | VARCHAR(MAX) | Not Null | Date of Birth |
| 7. | mobile\_no | VARCHAR(10) | Not Null | Mobile Number |
| 8. | age | VARCHAR(50) | Not Null | Age |
| 9. | gender | VARCHAR(1) | Not Null | Gender |
| 10. | address | VARCHAR(MAX) | Not Null | Address |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1. | faculty\_id | INT(11) | PRIMARY KEY  Not Null | Faculty ID |
| 2. | name | VARCHAR(50) | Not Null | Name |
| 3. | dob | VARCHAR(MAX) | Not Null | Date of Birth |
| 4. | age | VARCHAR(50) | Not Null | Age |
| 5. | gender | VARCHAR(1) | Not Null | Gender |
| 6. | email | VARCHAR(100) | Not Null | Email |
| 7. | mob\_no | VARCHAR(10) | Not Null | Mobile Number |
| 8. | experience | VARCHAR(50) | Not Null | Experience |
| 9. | department | VARCHAR(50) | Not Null | Department |
| 10. | salary | VARCHAR(10) | Not Null | Salary |
| 11. | joining\_date | VARCHAR(MAX) | Not Null | Joining Date |
| 12. | qualification | VARCHAR(MAX) | Not Null | Qualification |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1. | course\_id | INT(11) | PRIMARY KEY  Not Null | Course ID |
| 2. | name | VARCHAR(50) | Not Null | Course Name |
| 3. | fees | VARCHAR(10) | Not Null | Fees |
| 4. | duration | VARCHAR(50) | Not Null | Duration |

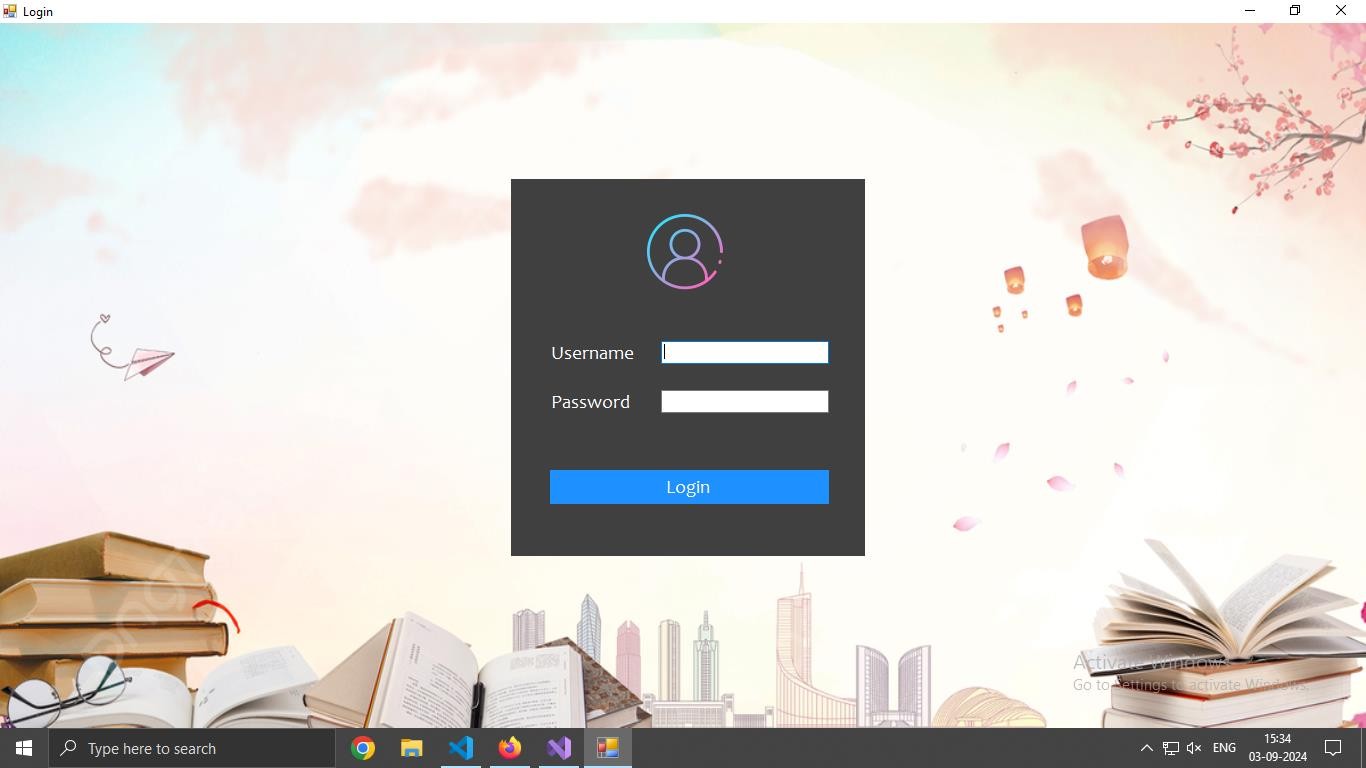


**stud\_attendance**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1. | id | INT(11) | PRIMARY KEY  Not Null | ID |
| 2. | student\_id | INT(11) | FOREIGN KEY  Not Null | Student ID |
| 3. | date | DATE | Not Null | Date |
| 4. | status | ENUM(‘Present’, ‘Absent’, ‘Leave’) | Not Null | Status |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1. | id | INT(11) | PRIMARY KEY  Not Null | ID |
| 2. | faculty\_id | INT(11) | FOREIGN KEY  Not Null | Faculty ID |
| 3. | date | DATE | Not Null | Date |
| 4. | status | ENUM(‘Present’, ‘Absent’, ‘Leave’) | Not Null | Status |

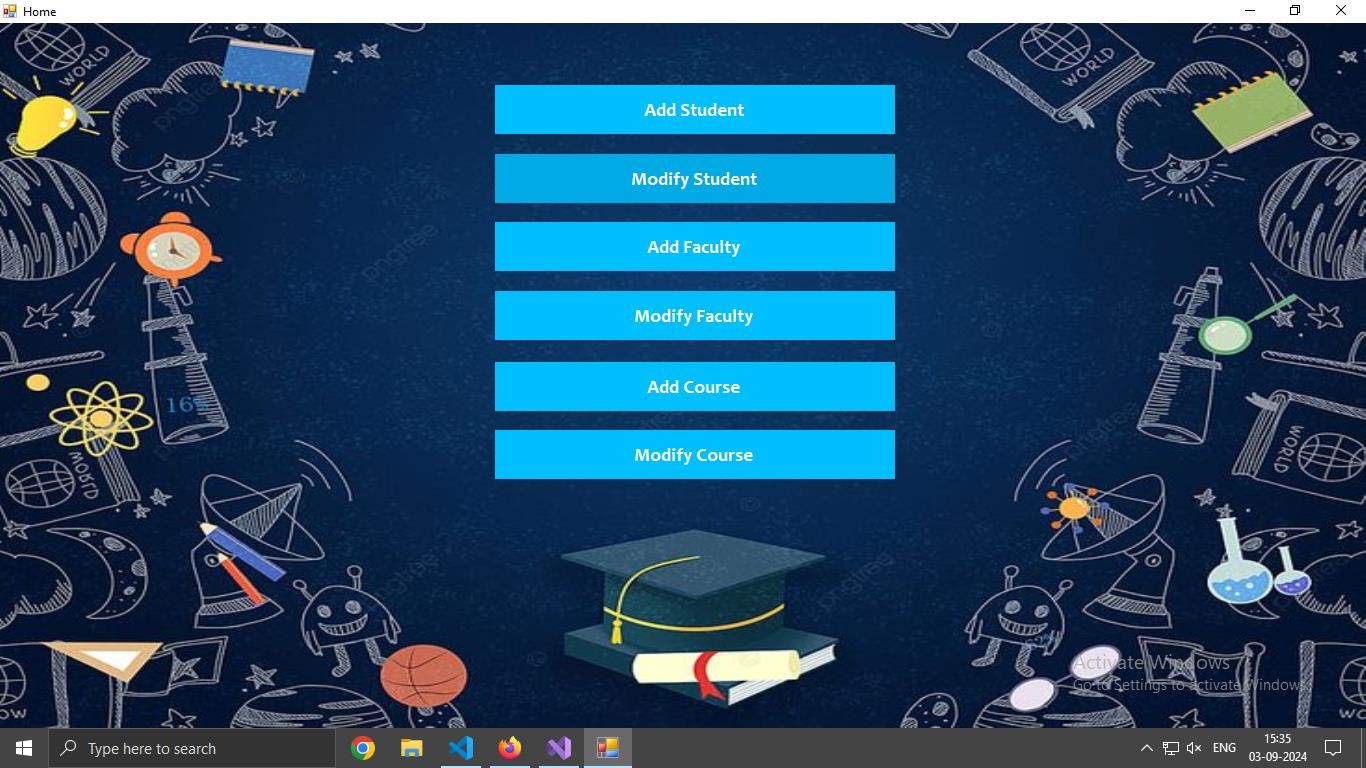




 Admin can login through this page.



***Screen Shots***



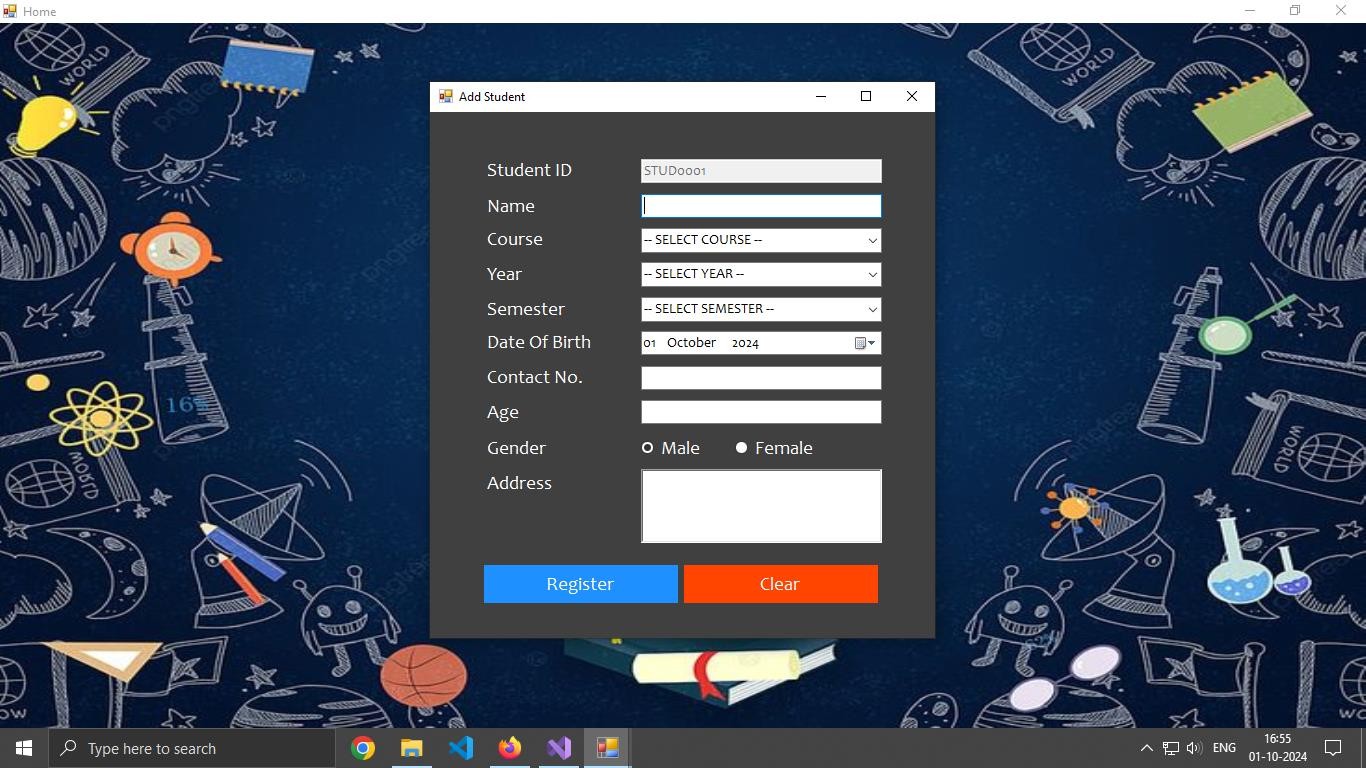
After login admin will be redirected to this page.



Admin have options to add students, faculties and courses. Admin can also modify student, faculty and course records.

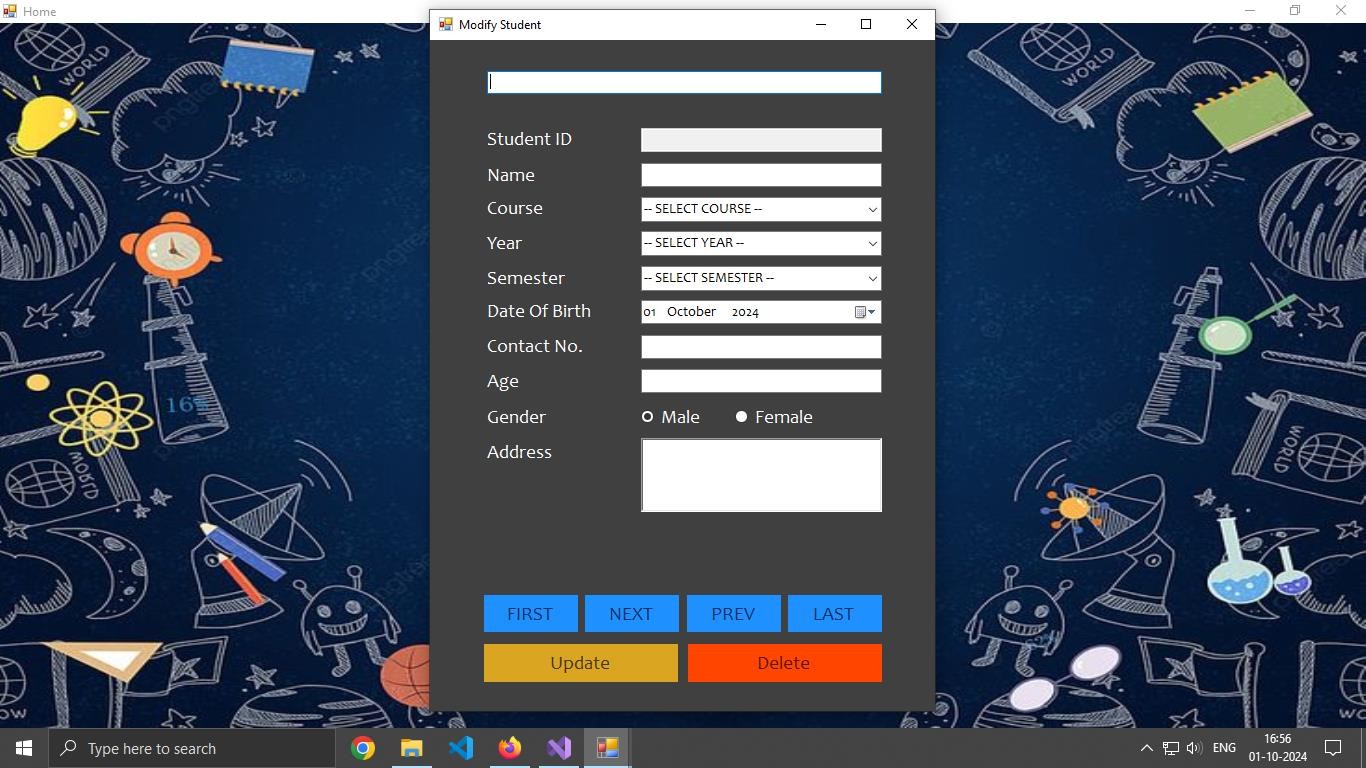


***Screen Shots***



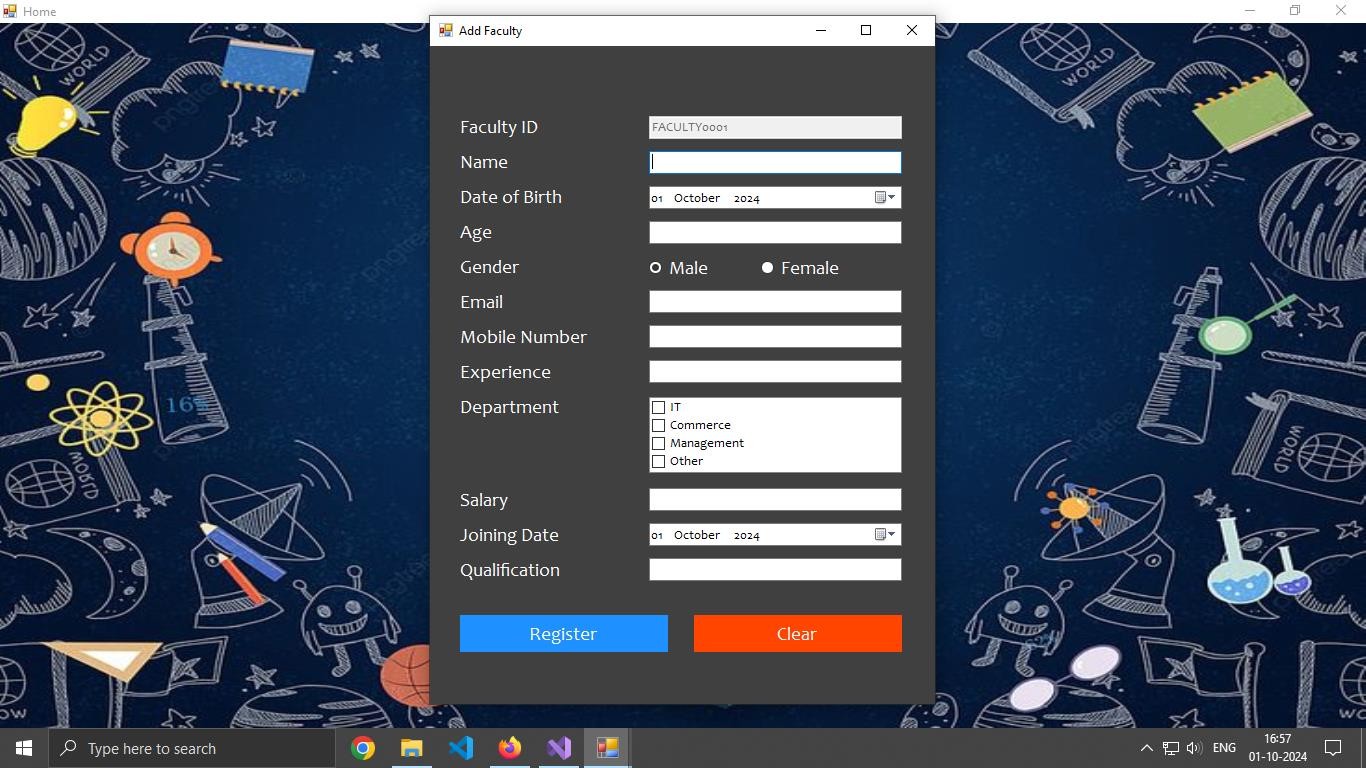
 Admin can add student using Add Student Page.

## Modify Student

****

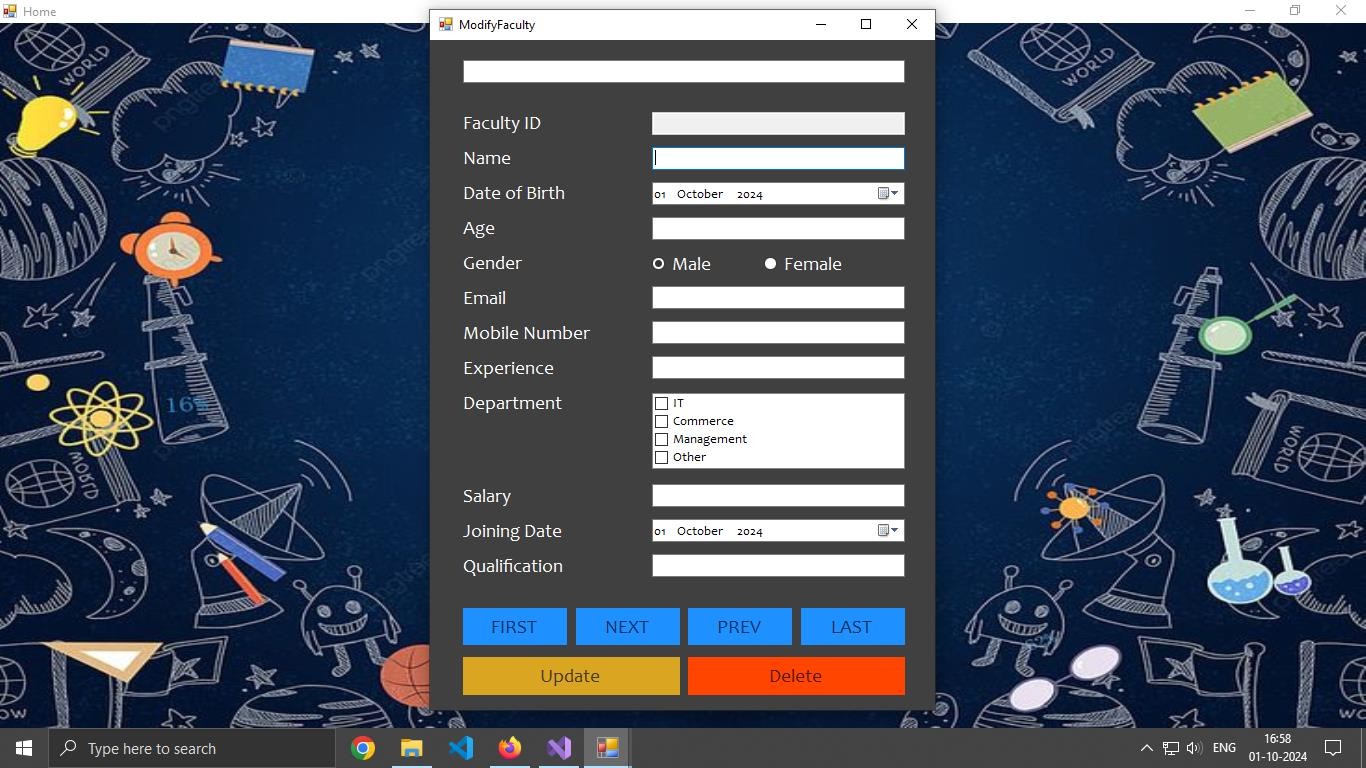
 Admin can modify student records using Modify Student Page.

## Add Faculty

****

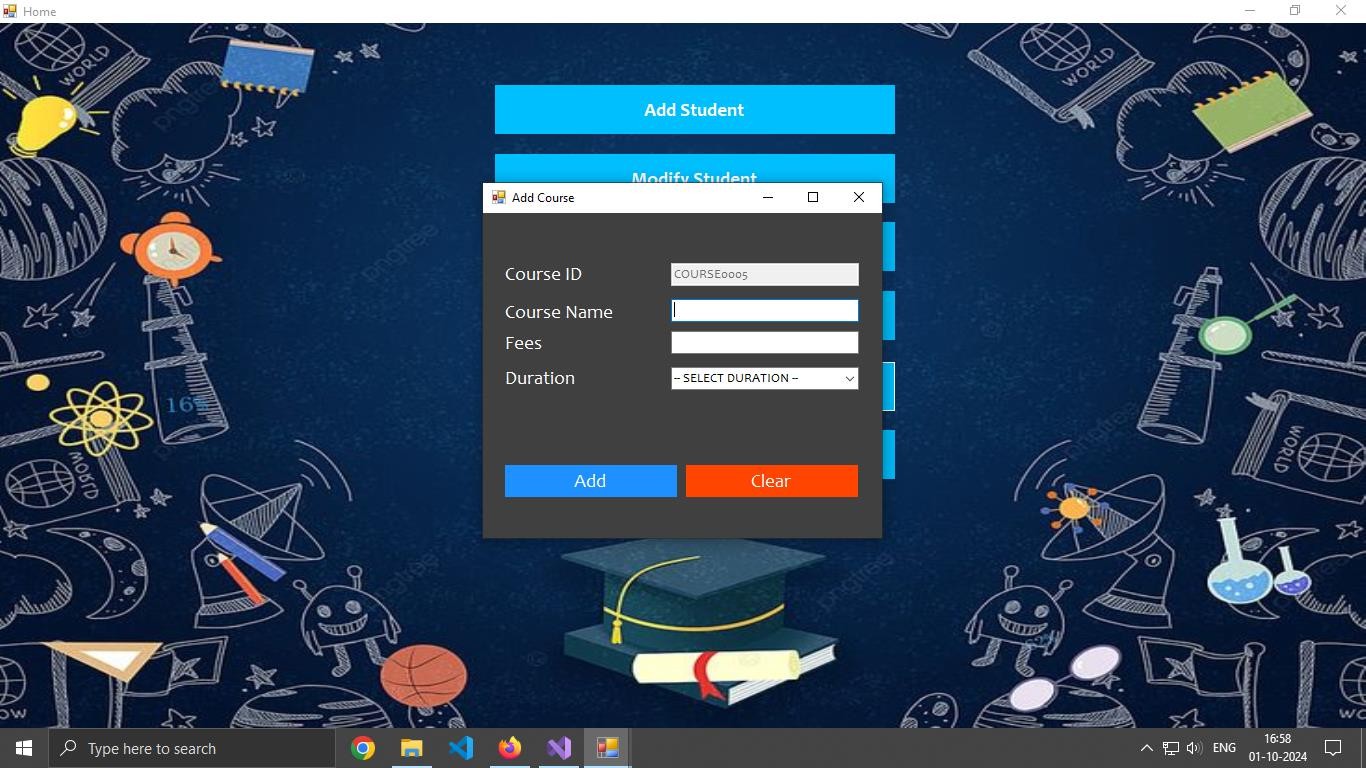
 Admin can add student using Add Faculty Page.

## Modify Faculty Records

****

 Admin can modify faculty records using this page.

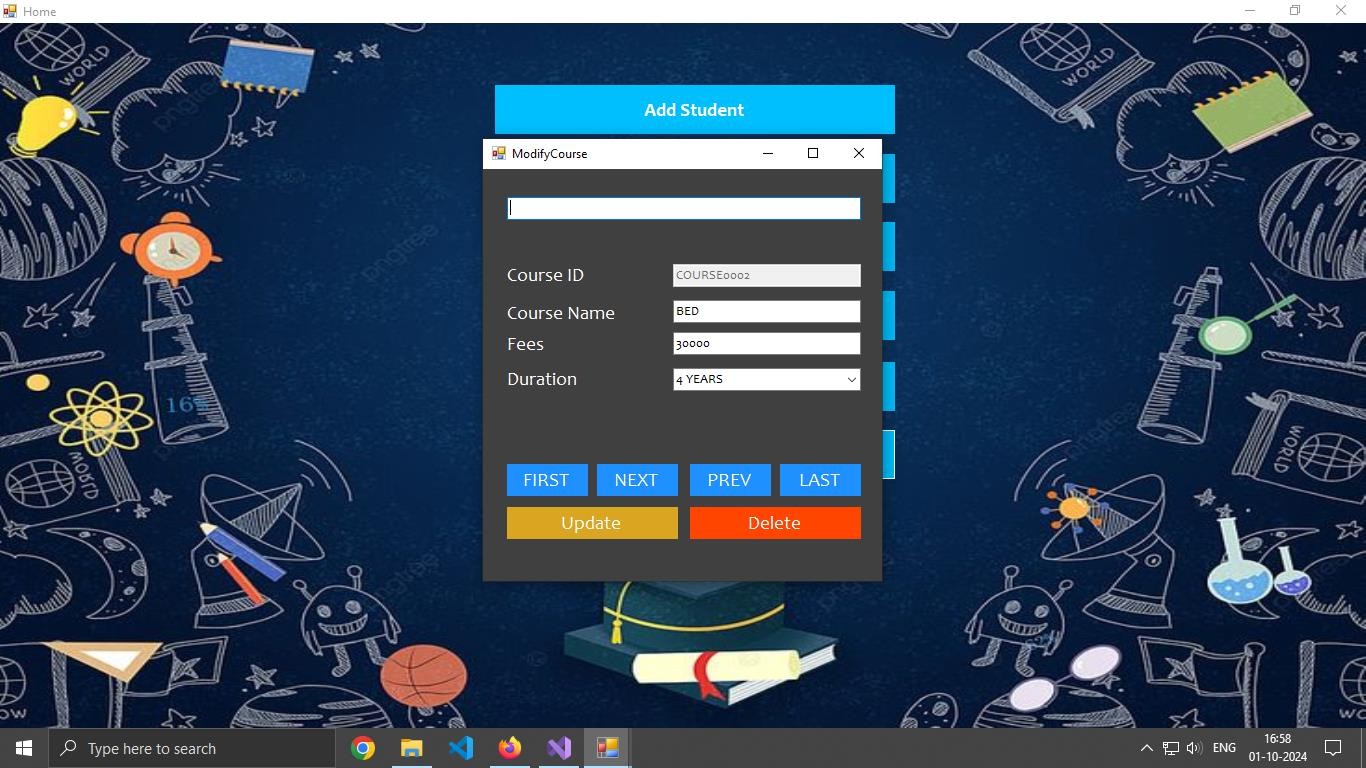
## Add Course

****

 Admin can add student using Add Course Page.

## Modify Course

***Screen Shots***

****

 Admin can modify faculty records using Modify Course Page.



Syllabus of courses

Generate fees report

 Python version must be greater than or 3.0



There is only the admin side. No user side.

#### Google

Source: [https://google.com](https://www.google.com/)



**Stackoverflow**

Source: [https://stackoverflow.com](https://stackoverflow.com/)

**Github**

Source: [https://github.com](https://github.com/)