# A
# PROJECT REPORT
# ON
## "Student Management System"

Project submitted in partial fulfilment of requirements
For the degree of
B.Sc. in Computer Science under CBCS
of
Computer Science Department of Maharaja Manindra Chandra College
of
**University of Calcutta**

By


| | | |
|---|---|---|
| Arnab Manna | Roll No.: 193211-21-0030 | Reg. No: 211-1111-0353-19 |
| Kunal Pal | Roll No.: 193211-21-0027 | Reg. No: 211-1111-0347-19 |
| Niketan Pal | Roll No.: 193211-21-0042 | Reg. No: 211-1114-0335-19 |
| Akash Yadav | Roll No.: 193211-21-0037 | Reg. No: 211-1111-0362-19 |


under the supervision
of

Monali Poddar
Assistant Professor and Head
Department of Computer Science
Maharaja Manindra Chandra College
University of Calcutta
Kolkata, West Bengal, India

2022

## Certificate from the Supervisor

This is to certify that the work embodied in this thesis entitled **"Student Management System"** has been satisfactorily completed by

| | | |
|---|---|---|
| Arnab Manna | Roll No.: 193211-21-0030 | Reg. No: 211-1111-0353-19 |
| Kunal Pal | Roll No.: 193211-21-0027 | Reg. No: 211-1111-0347-19 |
| Niketan Pal | Roll No.: 193211-21-0042 | Reg. No: 211-1114-0335-19 |
| Akash Yadav | Roll No.: 193211-21-0037 | Reg. No: 211-1111-0362-19 |

It is bona-fide piece of work carried out under my supervision and guidance at Maharaja Manindra Chandra College, Kolkata for partial fulfilment of the requirements for the awarding of the B.Sc. in computer science degree of the Department of Computer Science, University of Calcutta, during the academic year 2021-22.

_____

Monali Poddar,

Assistant Professor and Head,

Maharaja Manindra Chandra College,

University of Calcutta.

_____

Dr. Manturam Samanta,

Principal,

Maharaja Manindra Chandra College,

University of Calcutta.

## ACKNOWLEDGEMENT

We are pleased to express our gratitude and regard towards our Project Guide **Srimati Monali Poddar**, Assistant Professor and Head, Department of Computer Science, Maharaja Manindra Chandra College, University of Calcutta, without whose valuable guidance, inspiration and attention towards us, pursuing our project would have been impossible.

We are grateful towards all the members of **Maharaja Manindra Chandra College, University of Calcutta** for cooperating with us in all possible ways and providing us with a good working environment throughout the duration of work.

Last but not the least, we express our regards towards our friends and family for bearing with us and for being a source of constant motivation during the entire term of the work.

## Table of Contents:

## Abstract:

Student Information Management System can be used by education institutes to maintain the records of students easily. Achieving this objective is difficult using a manual system as the information is scattered, can be redundant and collecting relevant information may be very time consuming. All these problems are solved using this project.

**Objectives:**

- Maintenance of student records
- Searching student records

**Users Views:**

- Administrator
- User View

**Platform Operating Systems:**

- Microsoft Windows

**Technologies Used:**

**Application Front:**

- Python (Front End)
- MySQL (Back End)

**Web Front:**

- HTML / CSS / JavaScript
- Java Servlet Technology

**Software Requirements:**

- Python 3.10
- MySQL Workbench
- Apache Tomcat v10
- Microsoft Visual Studio Code
- Microsoft Windows

**Hardware Requirements:**

- Intel Pentium IV processor or equivalent or higher / With 512 MB Ram or Higher
- 20 GB HDD or Higher
- Active Internet Connection

## Introduction:

**Purpose:**

The objective of Student Management System is to allow the administrator of any organization to edit and find out the personal details of a student. It'll also facilitate keeping all the records of students, such as their id, name, mailing address, phone number, DOB etc. So, all the information about a student will be available in a few seconds. Overall, it'll make Student Information Management an easier job for the administrator. The main purpose of this SRS document is to illustrate the requirements of the project Student Management System and is intended to help any organization to maintain and manage its student's personal data.

**Scope:**

Without a Student Management System, managing and maintaining the details of the student is a tedious job for any organization. Student Information system will store all the details of the students including their background information, educational qualifications, personal details and all the information related to their resume.

**Login module:**

Login module will help in authentication of administrator accounts. Admins who have valid login id and password can only login into the system. Students can login to view his/her details and Pay their fees.

**Search module:**

Suppose there are hundreds of students and from this we have to search a particular student and we know the name of the student. In manual system it is a tedious task though we know the name of the student, but using this module we can easily search the student by specifying the name of the student in the search criteria. Thus, this module will help the administrator in searching the student with various criteria easily.

**Product Perspective:**

Student Management System is intended to be a stand-alone product. The system will also have an administrator who has full-fledged rights with regards to performing all actions related to control and management of the system. It also has a web front which is for displaying the details to students.

**Product Functions:**

An Administrator can login into the system and perform any of the available operations,

- Can edit student information to the database.
- Can make search for a specific student.
- Can access all the details of the student

A Student can login into the system and perform any of the available operations,

- Can check their details
- Pay their fees

**Constraints:**

- Every user must be comfortable using computer.
- All operations are in English so user must have basic knowledge of English.

**User Classes and Characteristics:**

There are mainly two kinds of users for the product. The users include:

- Administrator
- Student

## Background:

The student management system project proposal states the solution and the problems faced by attendance management. Student Management System (SMS) is a solution tool that is designed to track, maintain and manage all the data of the students. And let students check their details.

**Problem Statement:**

The common issue faced by student management in educational institutions is the data loss, Schools and Universities are the foundation of knowledge and an educational body on which students rely upon. Therefore, they need to maintain a proper database of its students to keep all the updated records and easily share information with students.

**Solution:**

The main purpose of the Student Management System is to track profiles, courses, logins, tests, and fees. It keeps track of all profile, student, fee, and profile-related information. Student management systems make it easier for professors to retrieve and sort information, which makes their jobs easier.

**Scope:**

This system is designed to be fully user-friendly and efficient. Its tasks could range from enrolling new students to managing fees and results, as well as everything else needed to keep the school's administrative division running properly

**Benefits of Using Student Management System:**

- Admission Data Management
- Tracking Student's Fees
- Well-Organized Management
- Reduction of Labour, Paper and Workload
- Improved Data Management

## Methodology:

**Problem Formulation:**

The topic of our project is “Student Management System”. The admin can log in by entering the provided official log in credentials. If the credentials do not match the system won’t let the user in. After successfully logging in the admin can perform all the database management tasks with the tools provided in the system. The user can manage “Courses”, ” Fees”,” Results”,” Students Details” with the individual tabs. Where as the students can log in to the website. They can check their details and pay their fees.

**Proposed Methodology:**

The approach to develop and deploy the Student Management System is employing micro-service architecture. The micro-service architecture is implemented using python which allows us to break up our app into smaller parts that communicate with each other also a rapid application development platform.

The Website Is created using all the usual web-tools like HTML, CSS, JavaScript. And The Back-End is done using Java (JSP, JDBC).

**Gathering Requirements:**

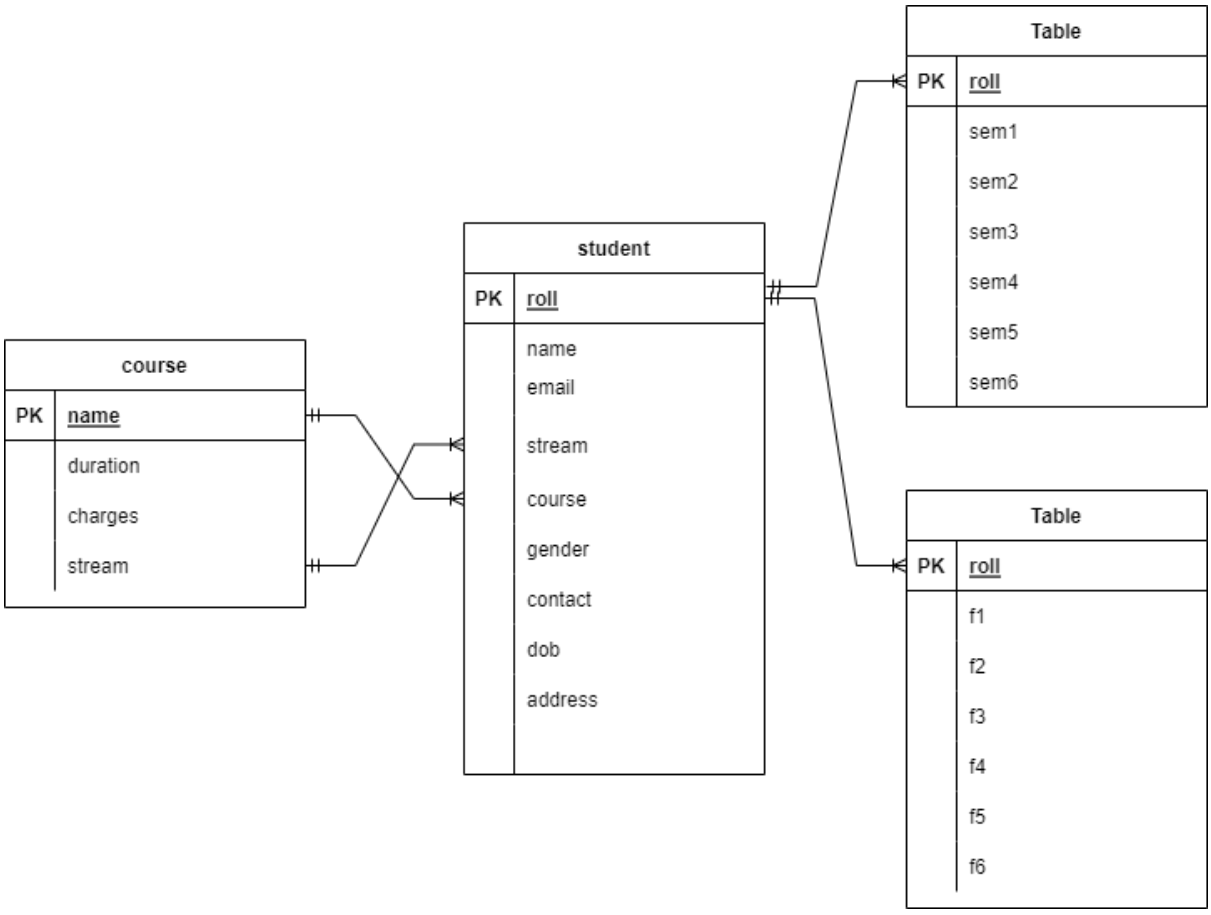
Before taking up any projects, the requirements must be collected and verified for the feasibility. The project can continue if the requirements are feasible. In this phase all the requirements necessary to develop and implement the project are collected and are conveyed to the developer and designer of the project. In this project, whose final product will be a (.py) application, the requirements are categorized into four categories such as, Student Management Service, Course Management Service, Fees Management Service, Results Management Service. The website is done separately then added with the system to make it whole.

**Design:**

Once the requirements are collected and analysed, they must be given a proper structure. In this phase the architecture of the project will be designed based on the requirements collected in the previous phase.

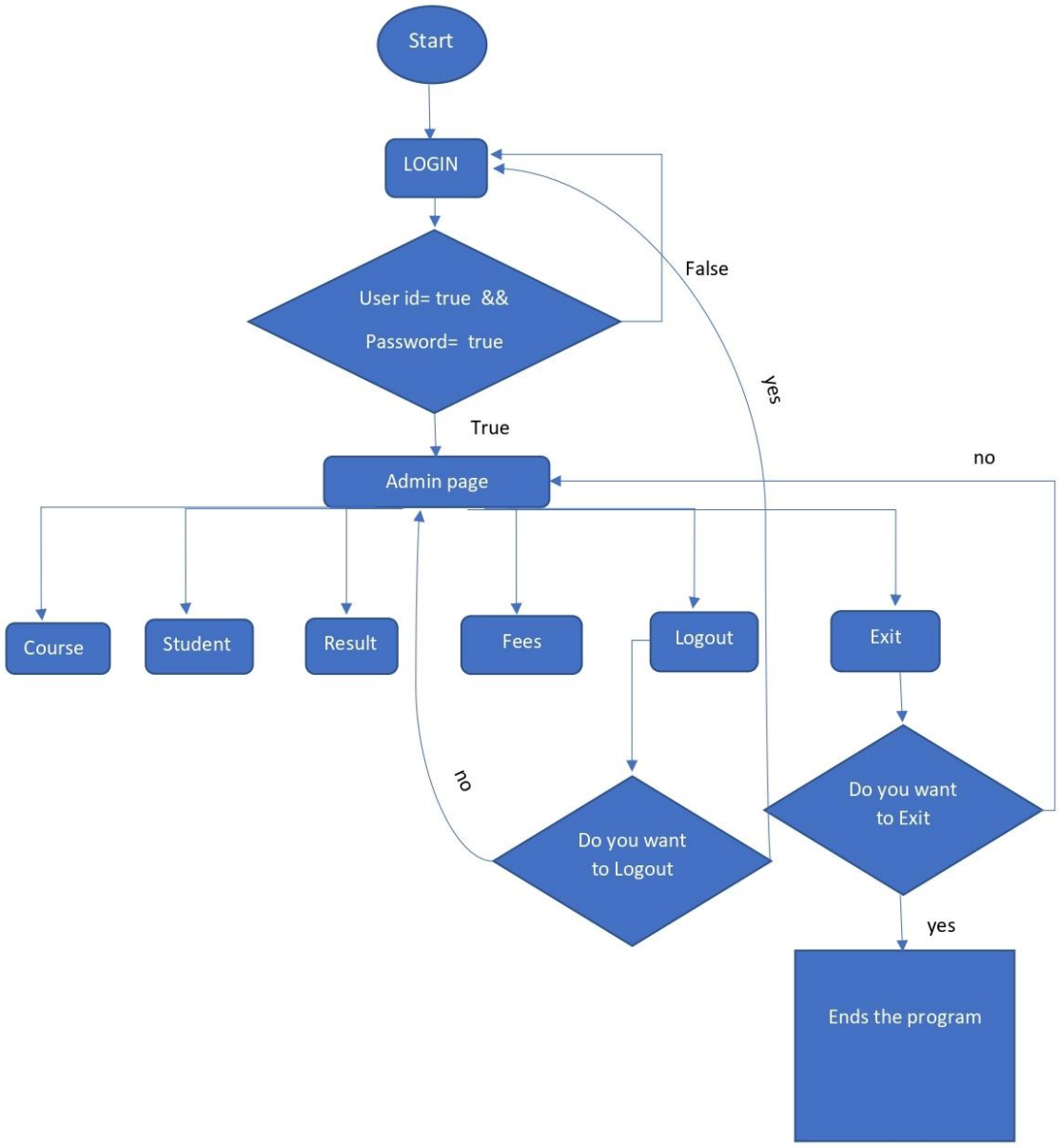
In this phase many architectural diagrams such as ER diagram, DFD (data flow diagram), Use case diagram, etc. are designed. The ER diagram defines the relationship between the entities and how they are inter-dependent.

**ER-Diagram of Student Management System:**

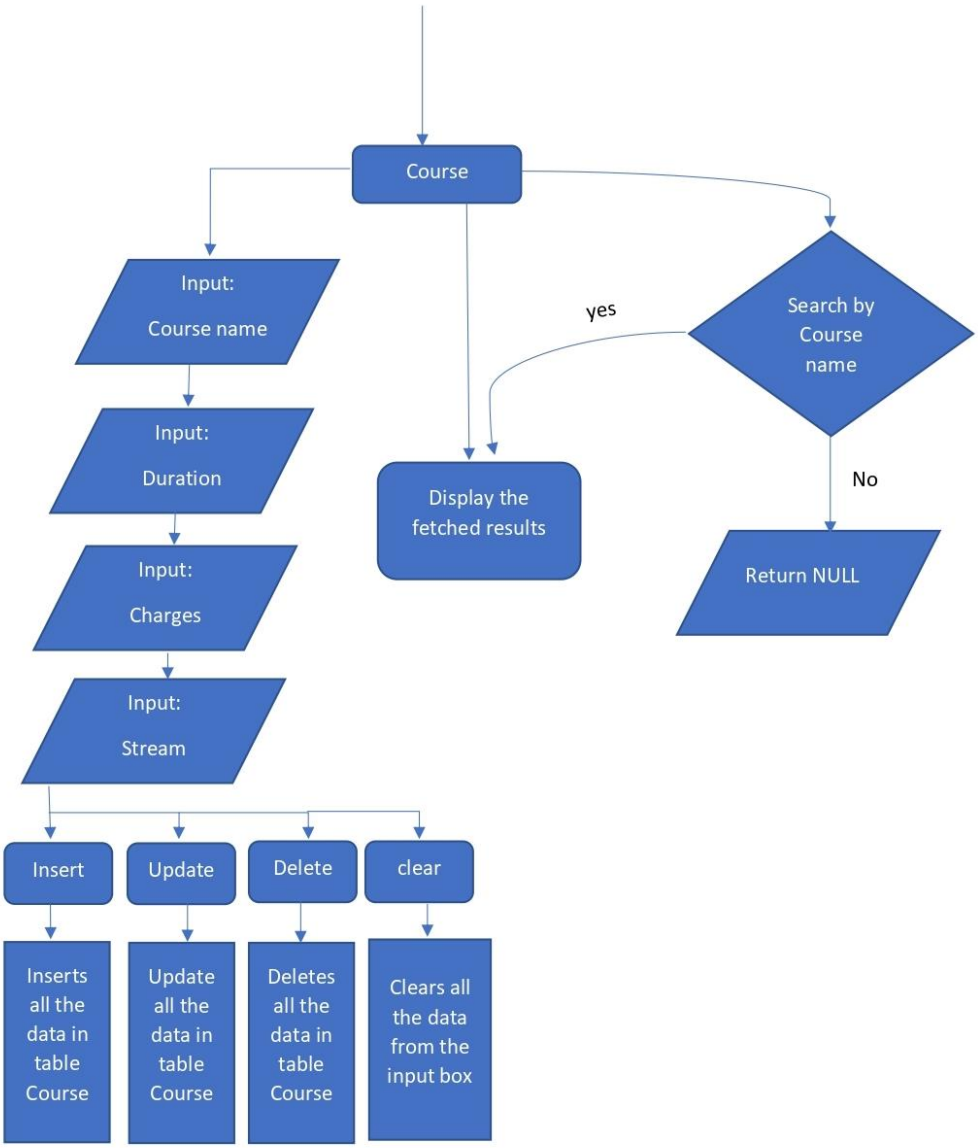


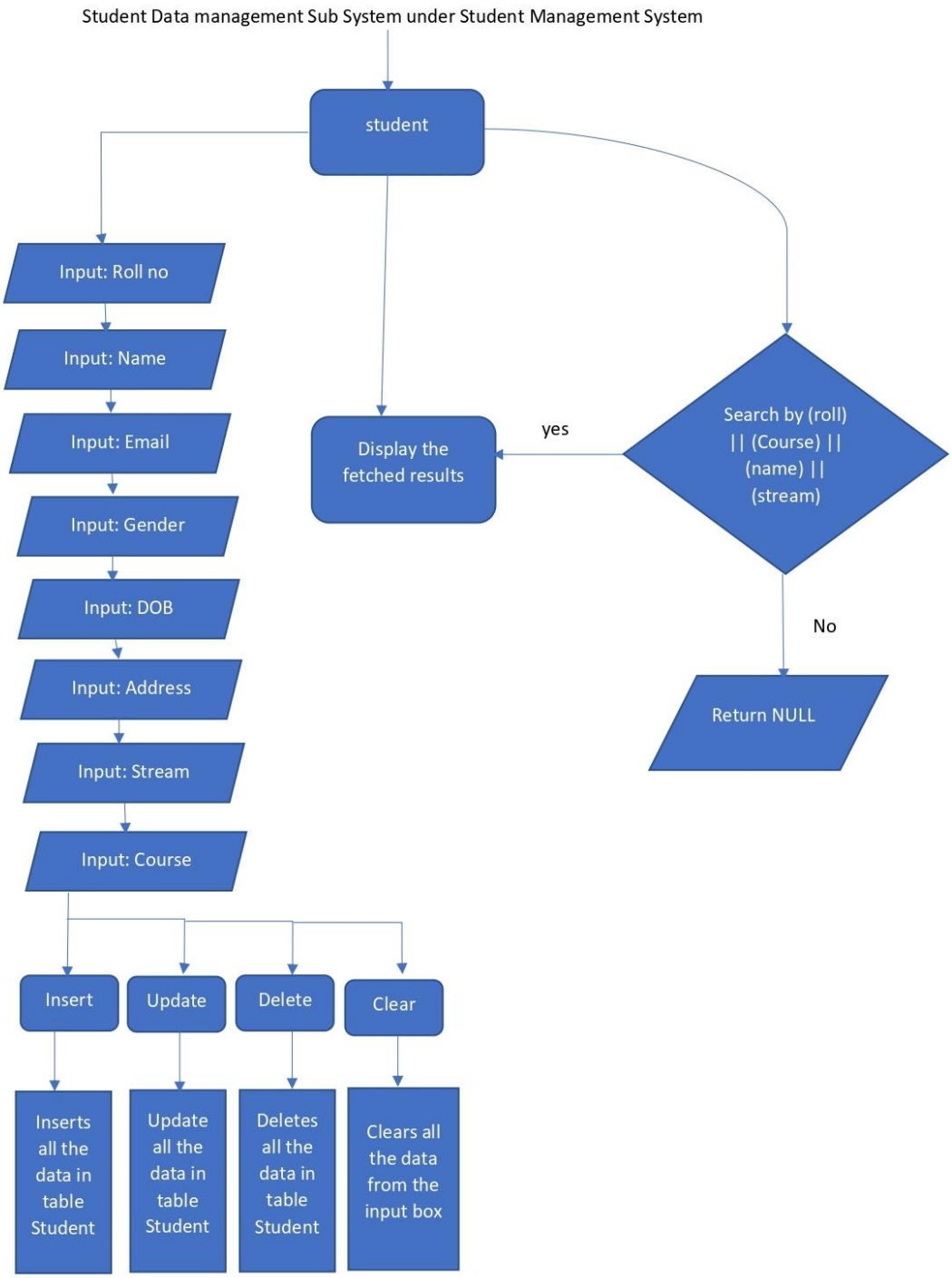
## Flow Chart:

Flow Chart of Student Management System

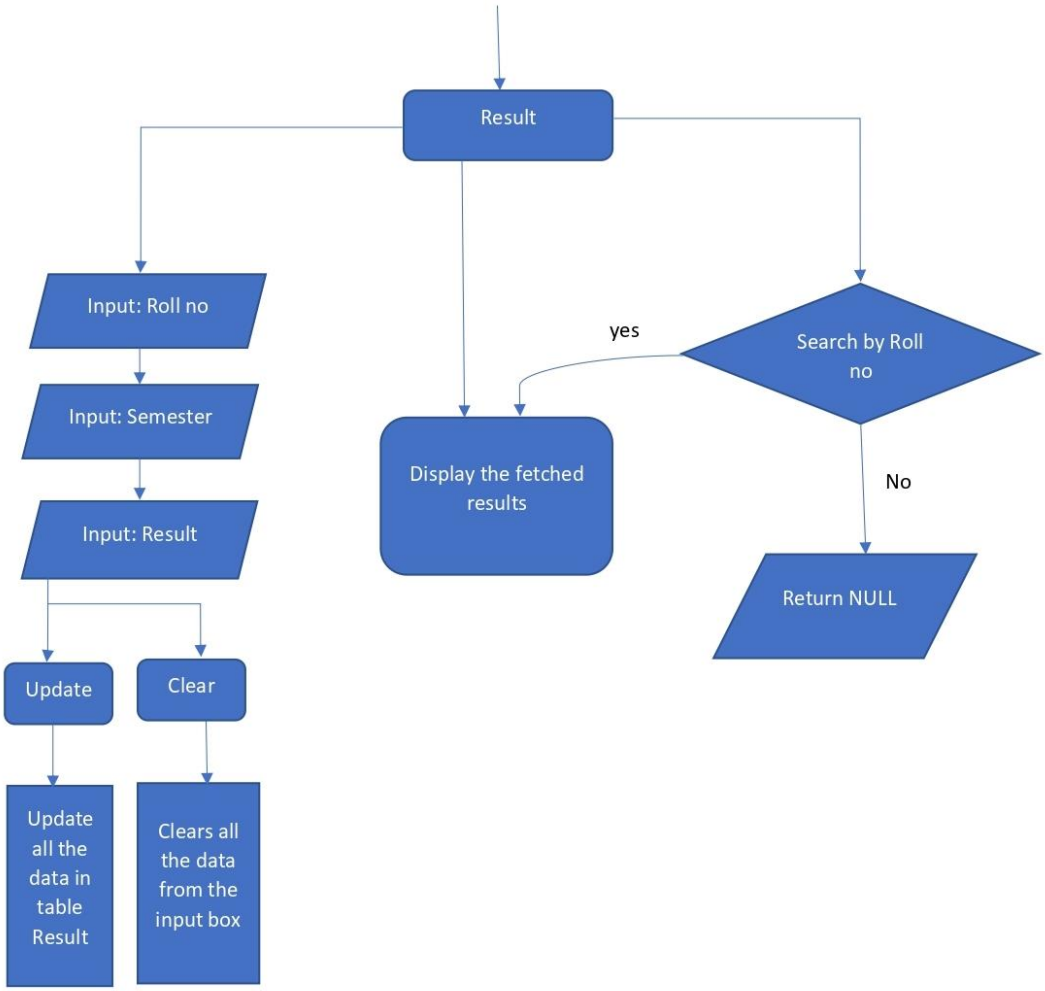


Course Management Sub System in Student Management System

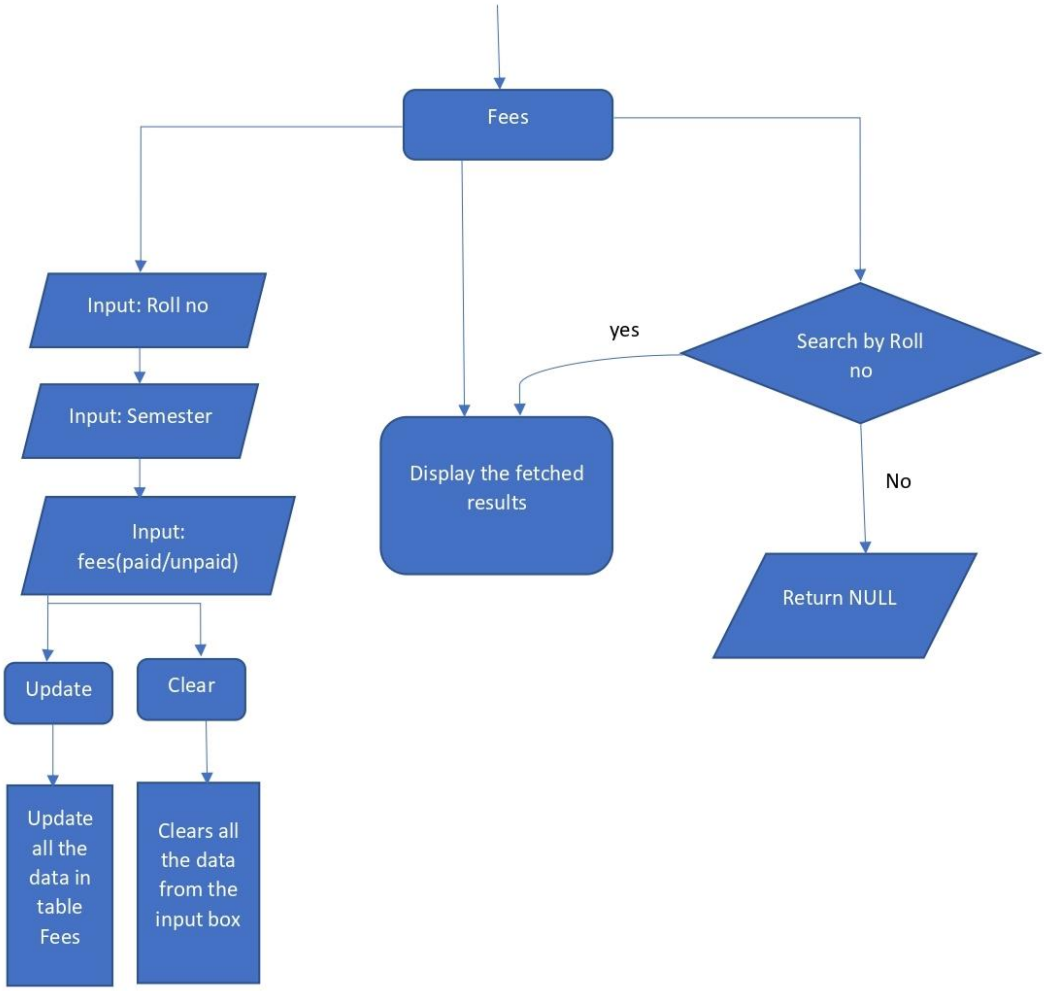



Student Data management Sub System under Student Management System
student
Input: Roll no
Input: Name
Input: Email
Input: Gender
Input: DOB
Input: Address
Input: Stream
Input: Course
Insert
Update
Delete
Clear
Inserts all the data in table Student
Update all the data in table Student
Deletes all the data in table Student
Clears all the data from the input box
Display the fetched results
yes
Search by (roll) || (Course) || (name) || (stream)
No
Return NULL

Result Management Sub System under Student Management System.



Fees Management Sub System under Student Management System.



**Testing:**

Any project before exposed to user must be tested to ensure that it behaves as expected. In this project, the application is tested by giving various types of input to check whether they are being validated or not and whether the application is behaving as expected or not.

**Maintenance:**

Once the application is tested and deployed, it must be maintained to satisfy the various constraints such as availability, reliability, etc. The newer versions of the applications can be developed depending on the success or feedback of the users.

## Implementation:

## Application Front:

**Python (.py) Files:**

We start by creating python files of the student management system. We created different programs and interlinked them to create the whole system. In those python files we implemented as well as connected the database. We created UI to perform all the tasks for the student management system.

We created a setup.py file which creates all the tables and installs all the libraries required to run the software.

**Used Python (.py) Libraries:**

**Tkinter:**

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit

**OS:**

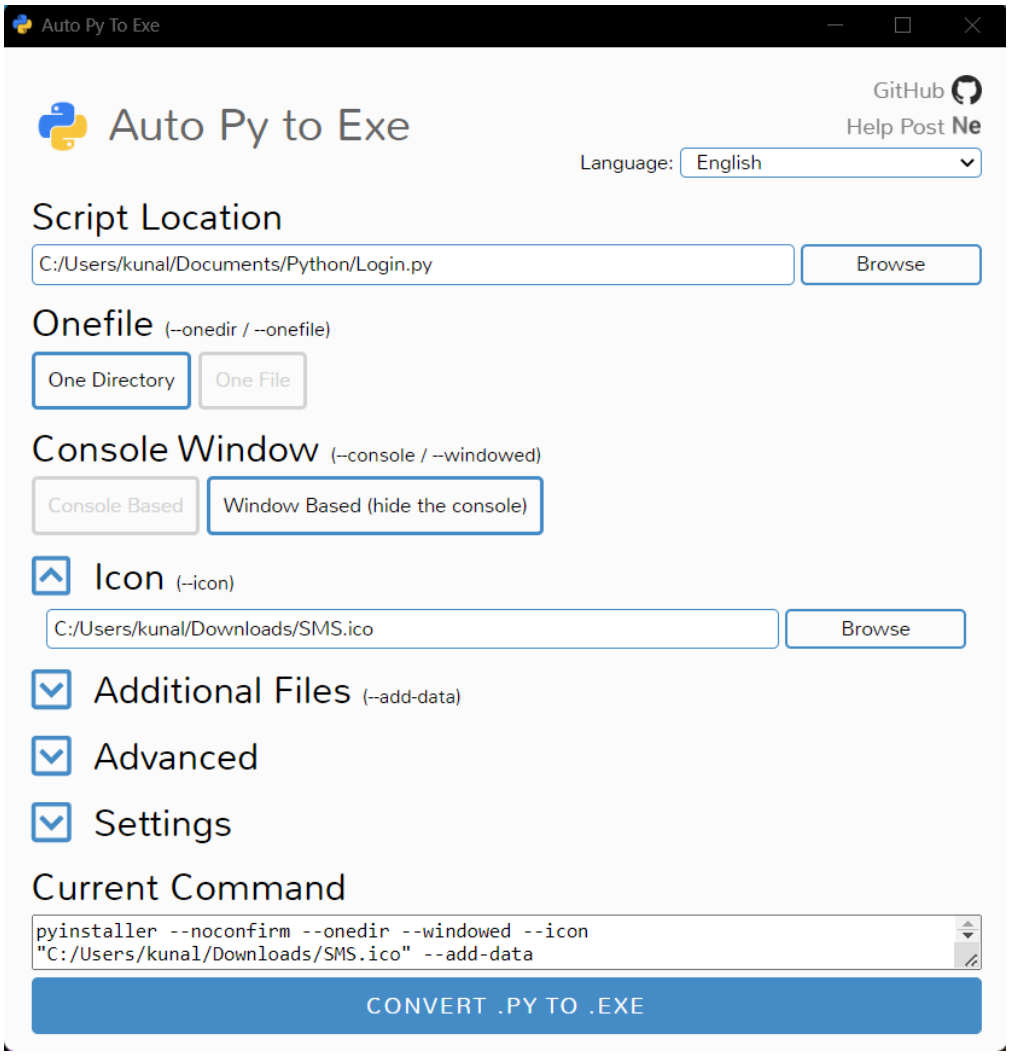
The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system-dependent functionality. The "os" and "os.path "modules include many functions to interact with the file system.

**Pillow:**

Pillow is a Python Imaging Library (PIL), which adds support for opening, manipulating, and saving images. The current version identifies and reads a large number of formats. Write support is intentionally restricted to the most commonly used interchange and presentation formats.

**Converting the Python (.py) Files to Executable (.exe) Format:**

We used **"auto-py-to-exe"** to convert the .py files to .exe format.

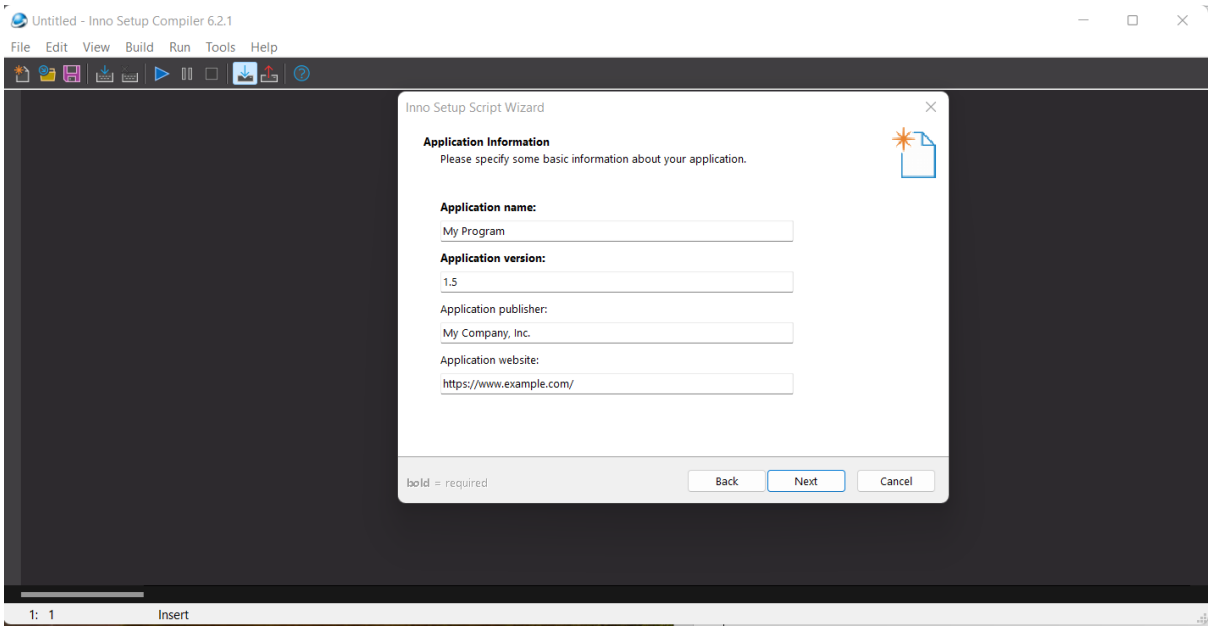



The command generated by auto-py-to-exe is:

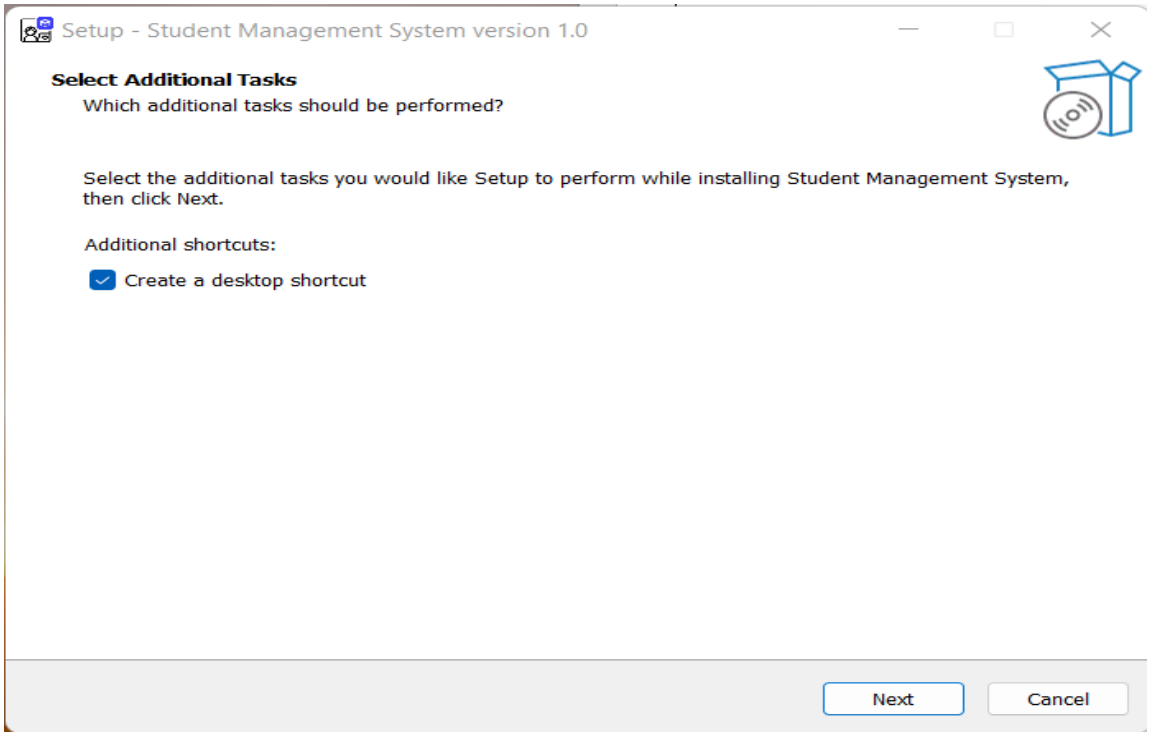
**pyinstaller --noconfirm --onedir --windowed --icon "C:/Users/kunal/Downloads/icons8-student-64.ico" --add-data "C:/Users/kunal/Documents/Python/course.py;." --add-data "C:/Users/kunal/Documents/Python/fee.py;." --add-data "C:/Users/kunal/Documents/Python/main.py;." --add-data "C:/Users/kunal/Documents/Python/result.py;." --add-data "C:/Users/kunal/Documents/Python/stdnt.py;." "C:/Users/kunal/Documents/Python/Login.py"**

**Converting Executable (.exe) File To an installer:**

After creating the .exe file we used the **“Inno Setup Compiler”** to create an install wizard.



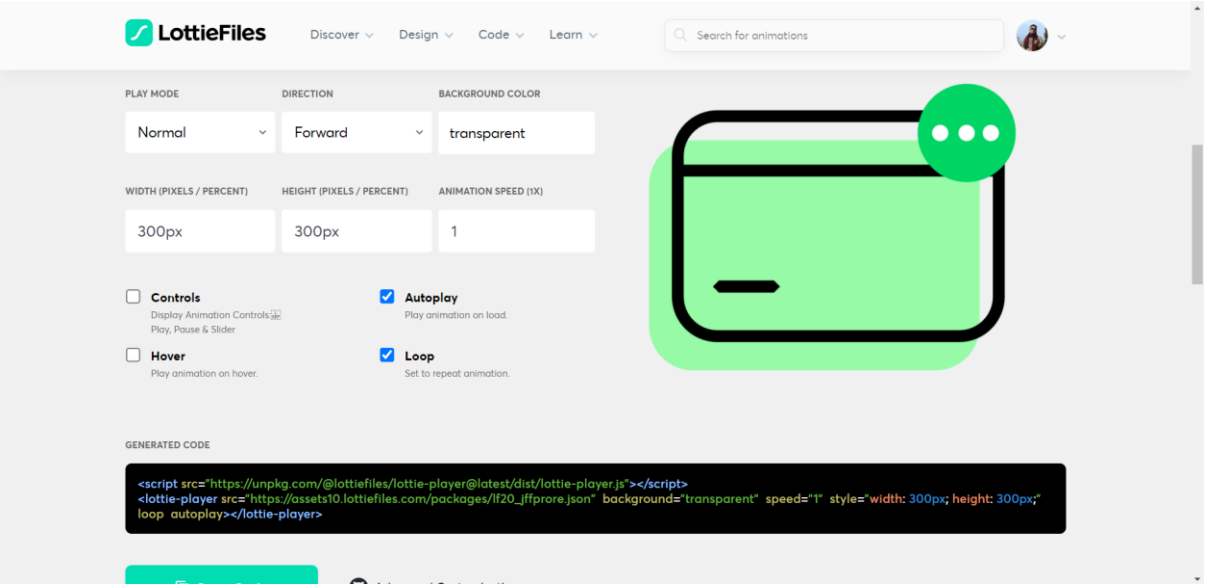

**Generated Setup Wizard:**



## Website Front:

The website is done using html, CSS, JavaScript. The home page layout is a simple, it has two animated CSS buttons. The first button leads us to the log in page and the second button lead us to the pay fees page. After log in to the system the user can see their details in the system. The pay fees page consists of a few animated Lottie files, clicking the directed button opens a quick response code through which the user can pay directly to the collage/school.

In this website we used a lot of animated Lottie files which are implemented directly via the direct embed feature present in the **"lottiefiles.com"** website. This allows us to get beautiful animations into any web page, just by pasting in a snippet of code.



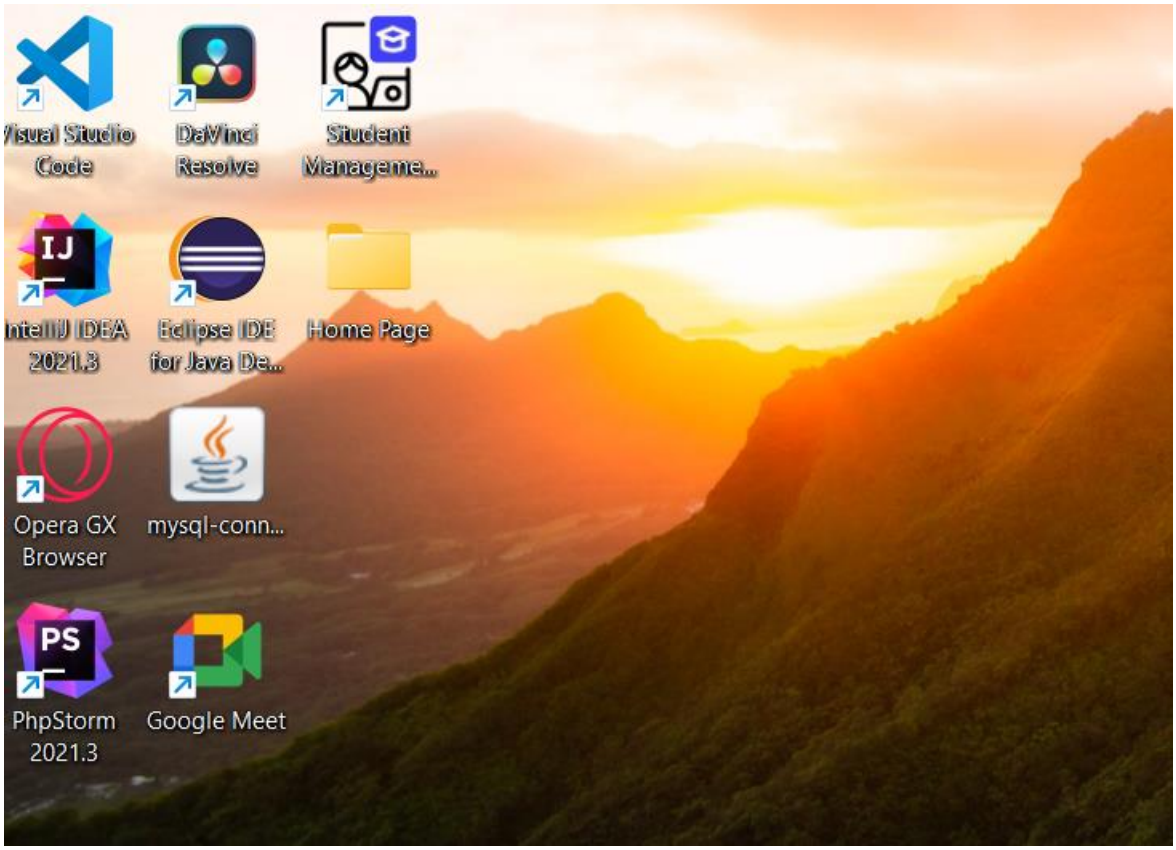

The kind of code snippet generated by Lottie files:

**<script src="https://unpkg.com/@lottiefiles/lottie-player@latest/dist/lottie-player.js"></script>**
**<lottie-player src="https://assets10.lottiefiles.com/packages/lf20_jffprore.json" background="transparent" speed="1" style="width: 300px; height: 300px;" loop autoplay></lottie-player>**

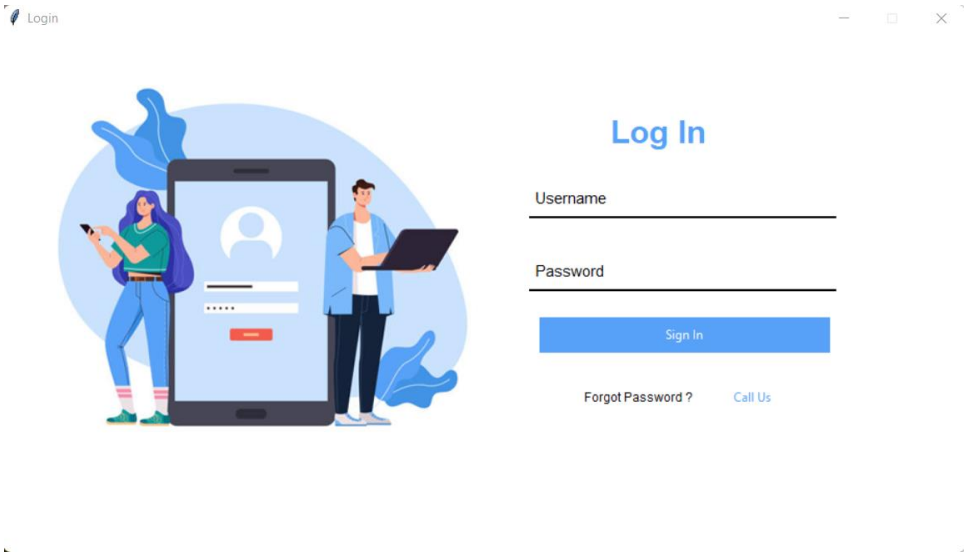
## Result & Discussion:

### Application-Front:

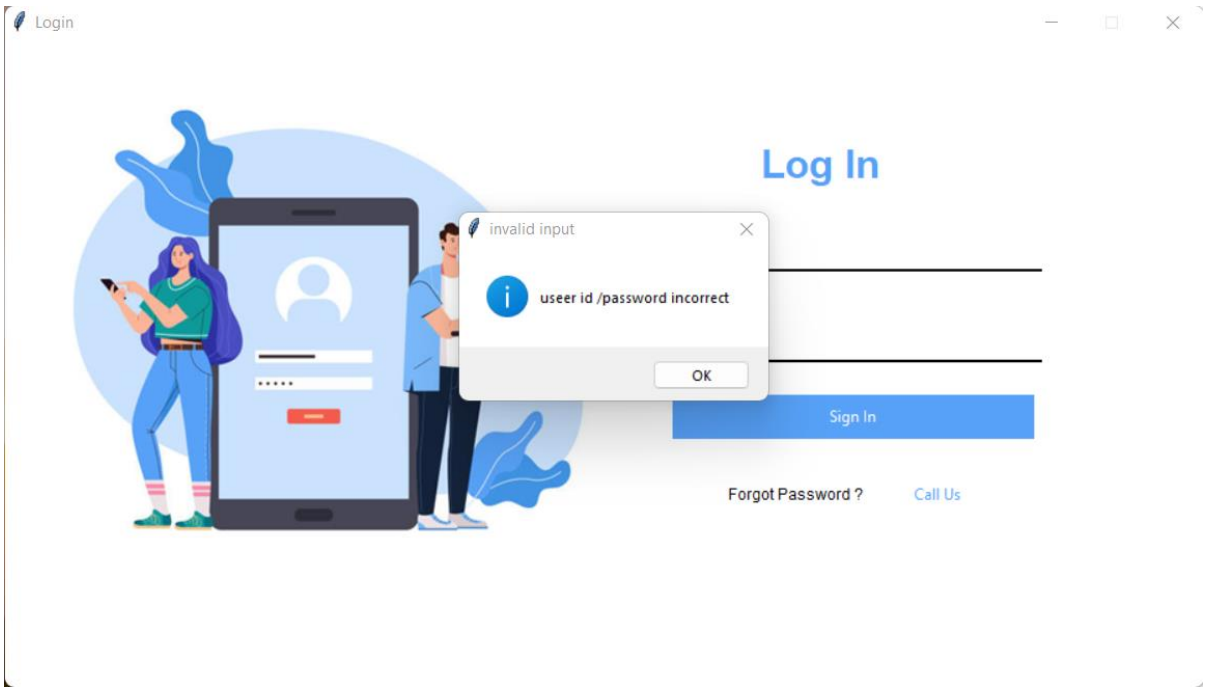
We start the software by double clicking the “Student Management System” icon on the desktop after installing the software.



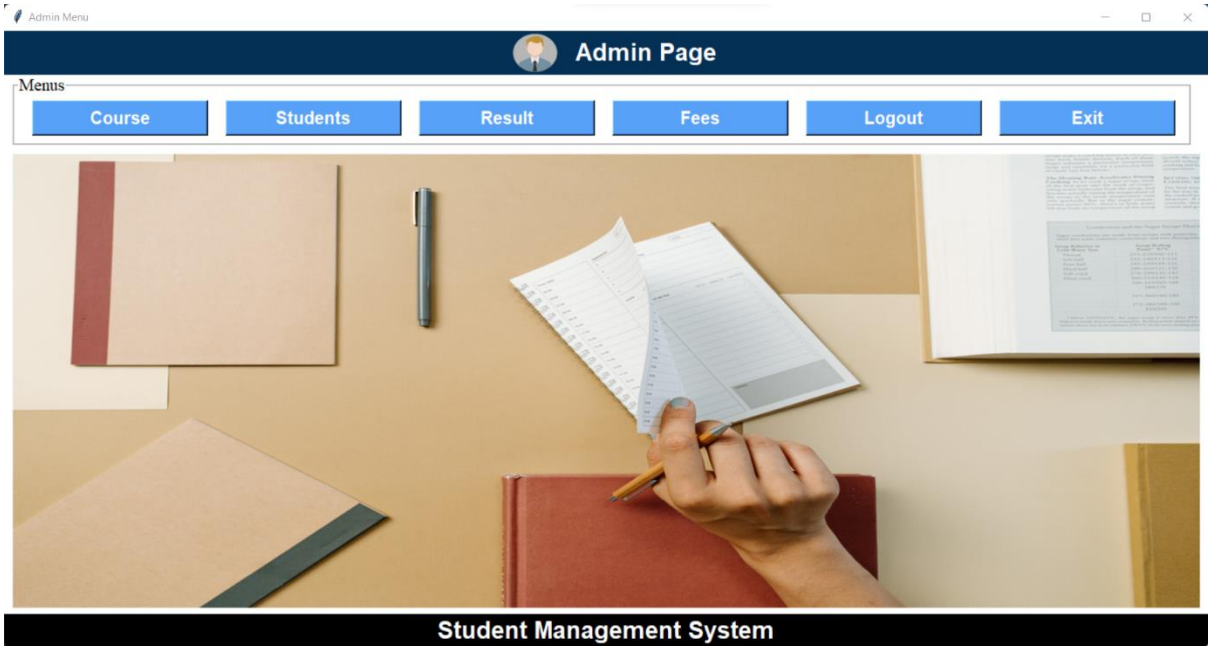

Now we are presented with a log in window,



If wrong id / password is given then a pop-up will occur.

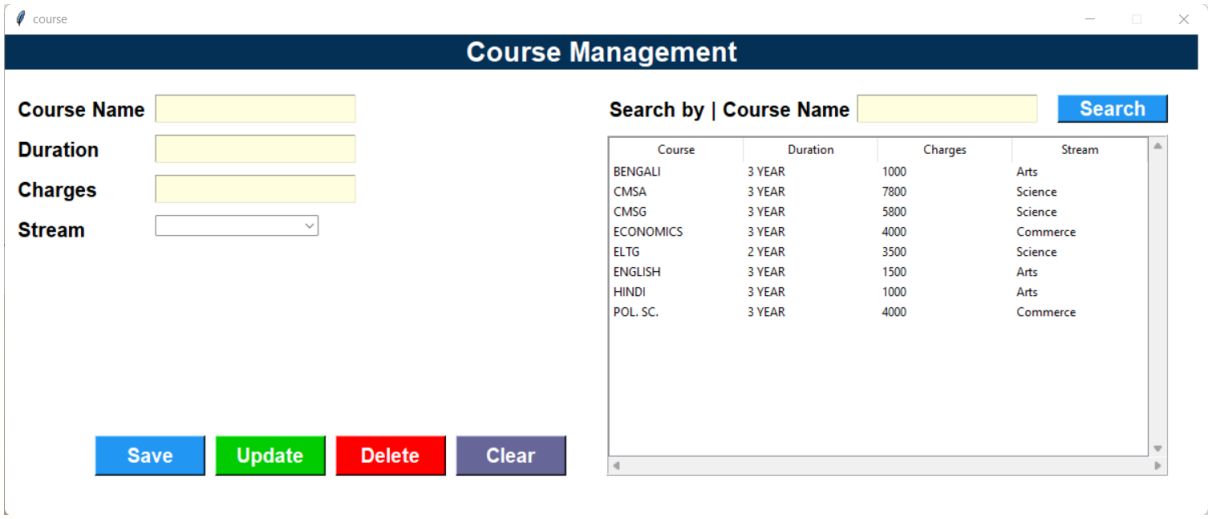



After successful log in, admin page will appear.

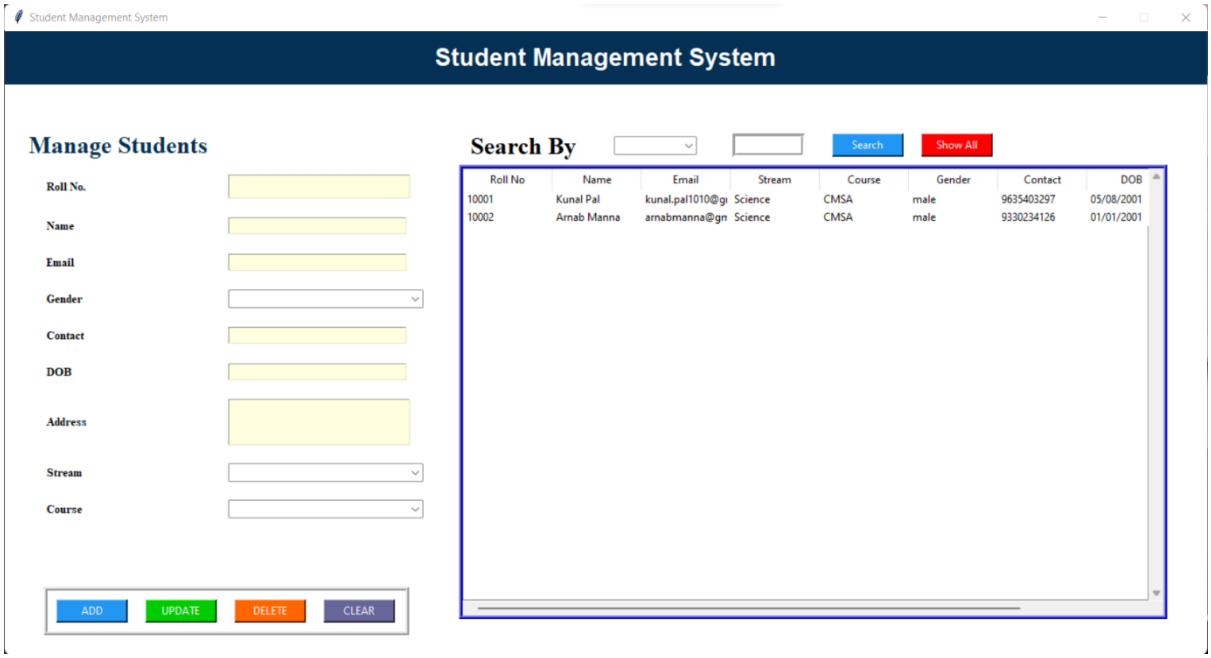


There are four operation tab present, these are Course, Students, Result, Fees.

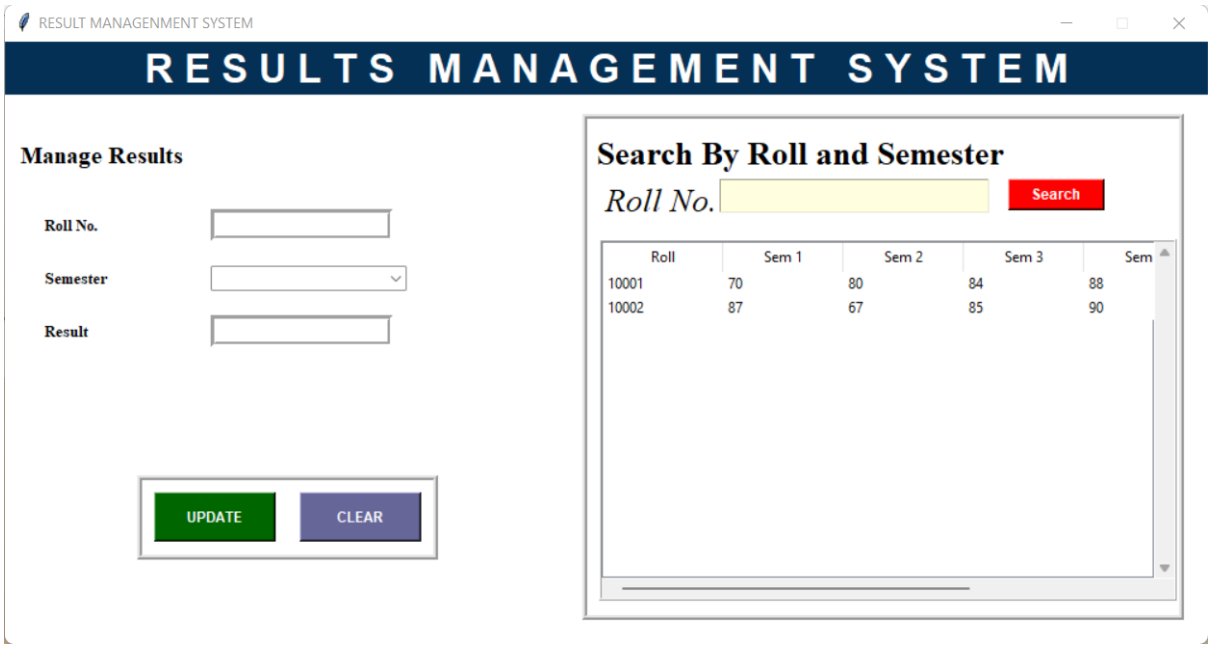
Course Tab,



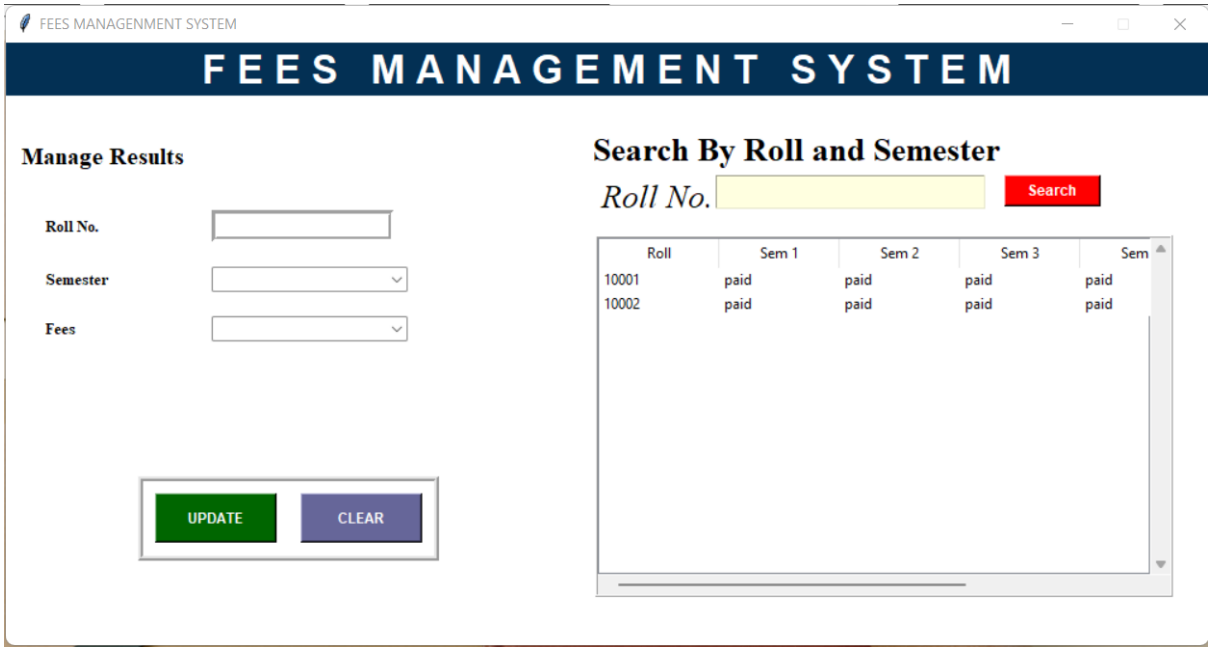

Students Tab,



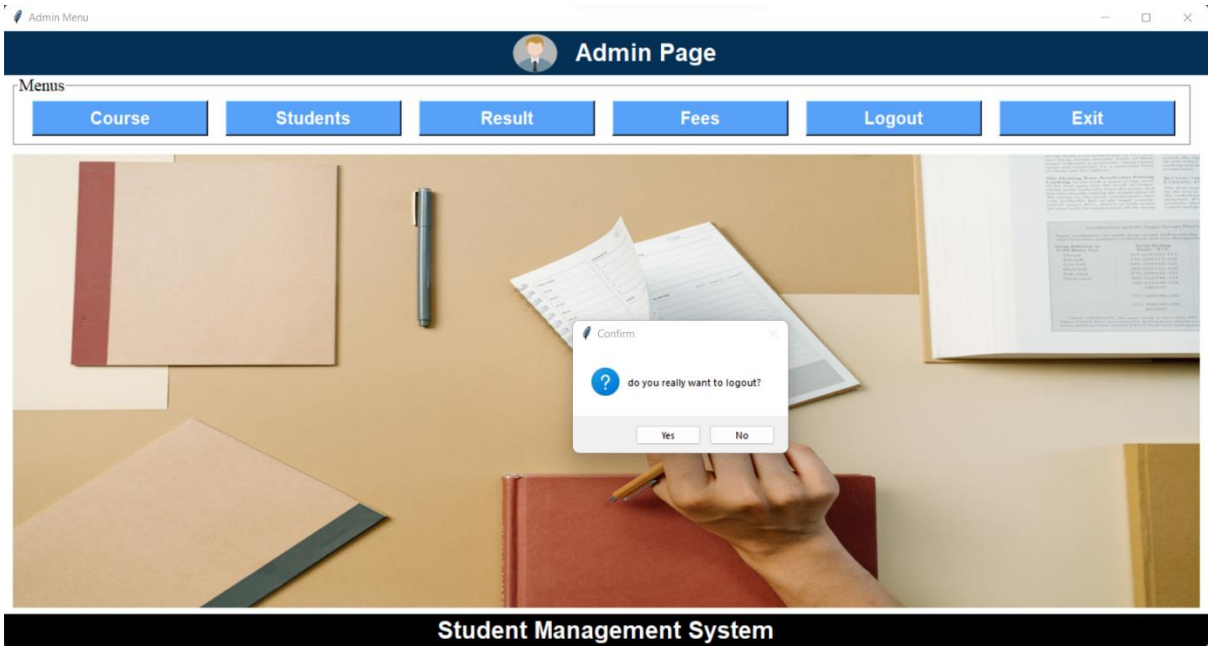
Result Tab,



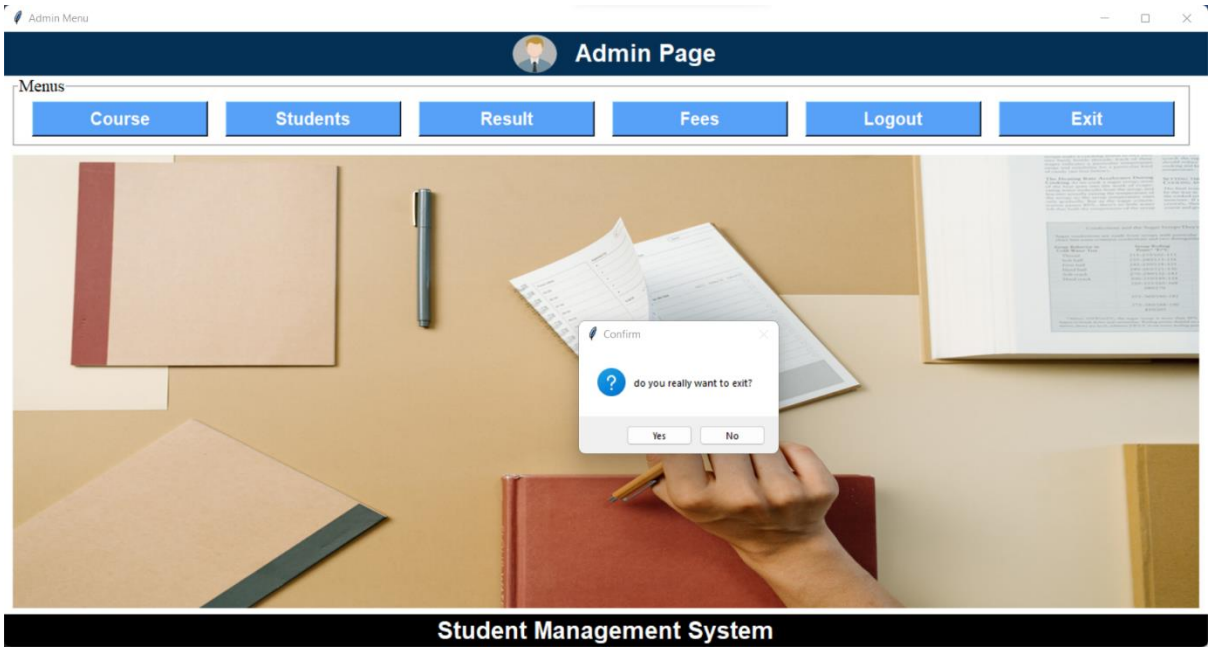

Fees Tab,



There is a logout button that prompts a window, when the user press yes to that prompt, it will take the user back to the login window.

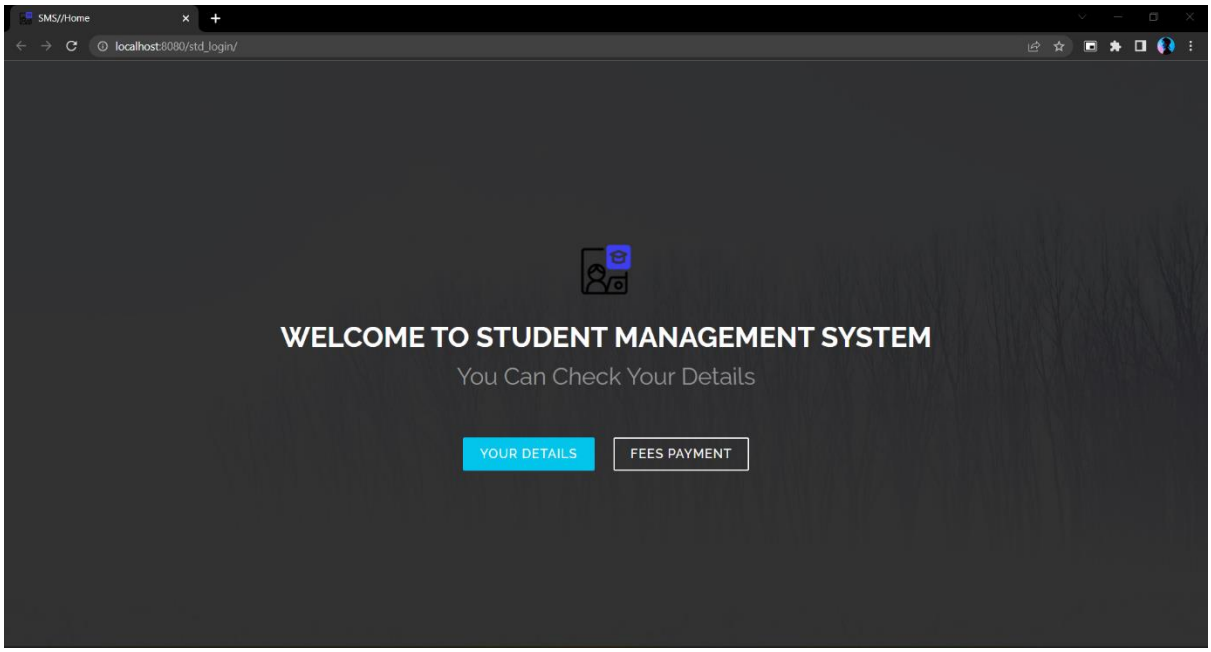



Beside the logout button there is an exit button which closes the software.

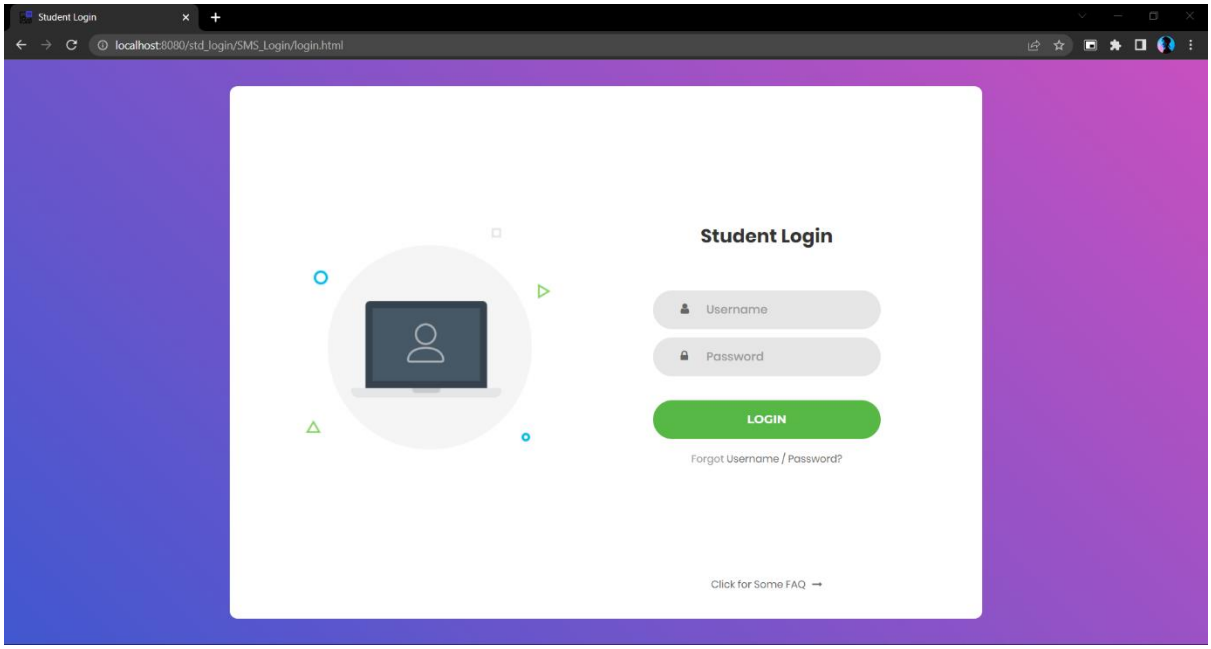


**Web-Front:**

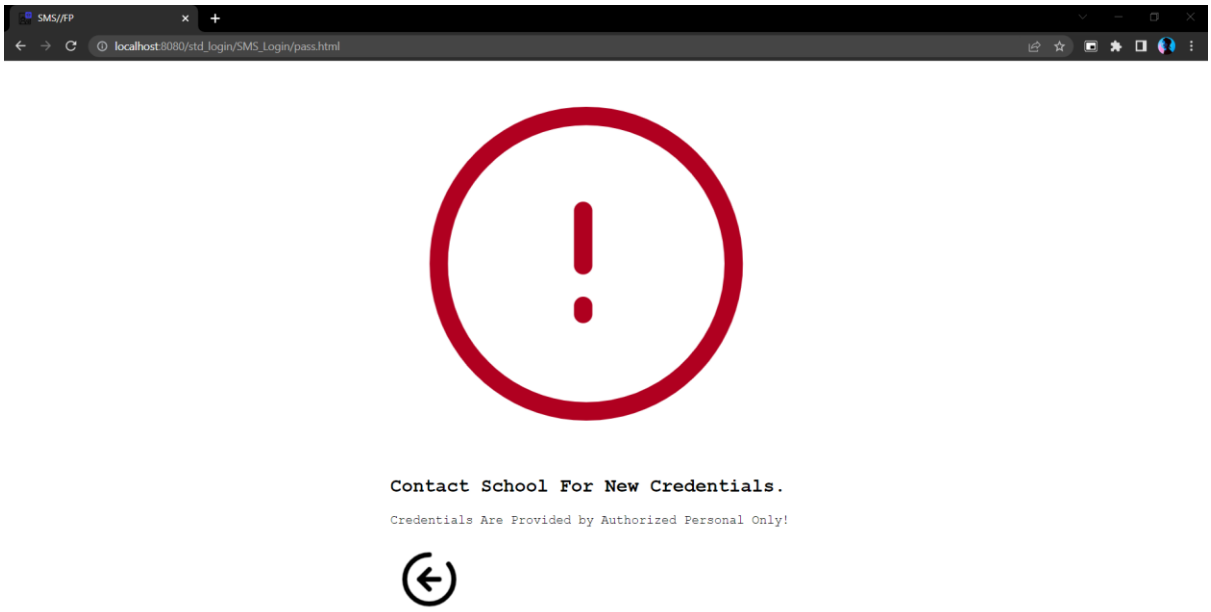
Home page of the website,



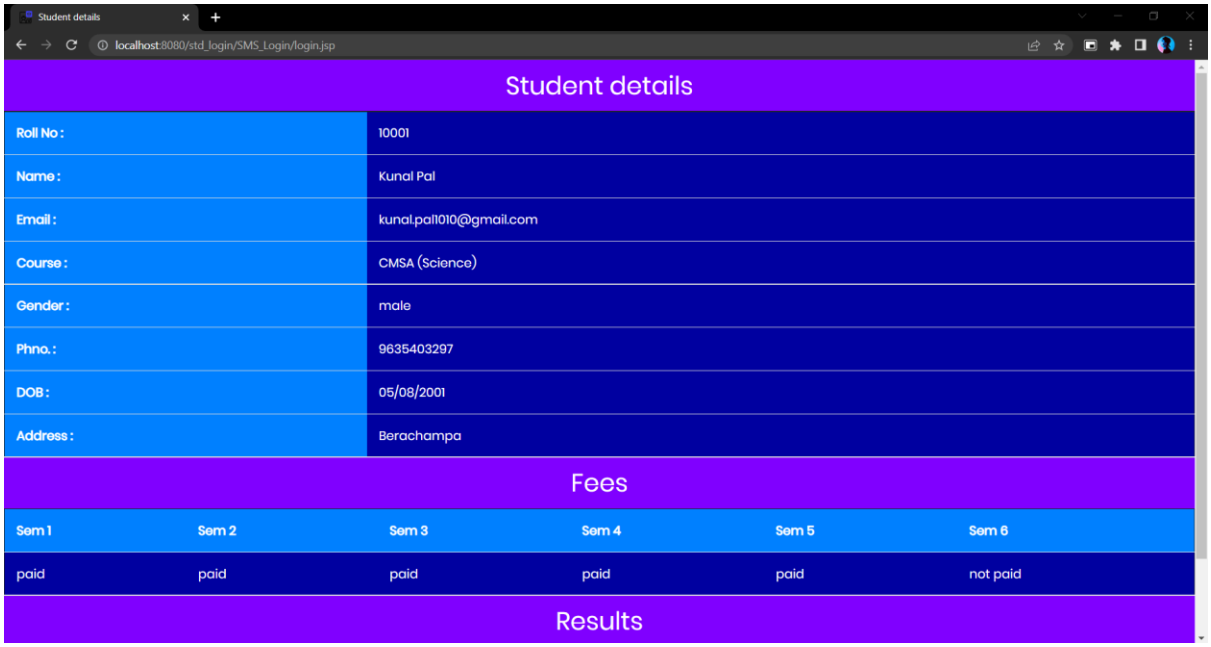

Login page of the website,



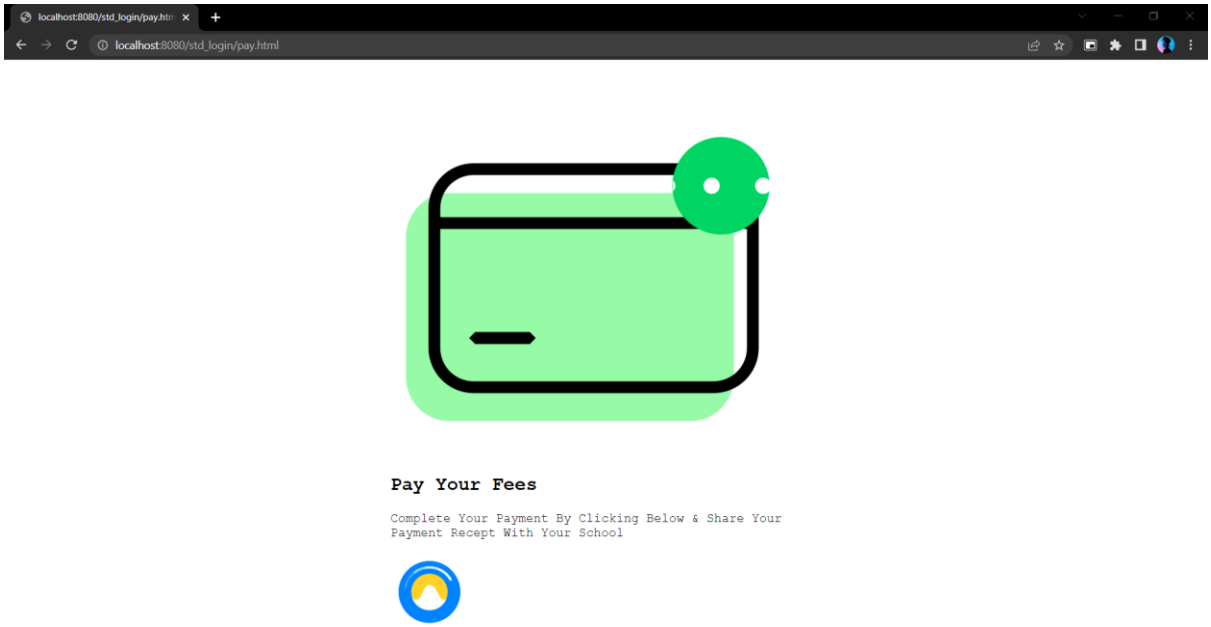
Forgot Password Page,




After Login It shows the students details,

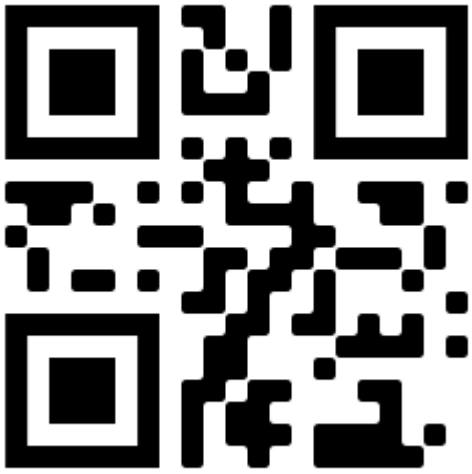


Pay fees page,




---

After Clicking the icon, it shows a QR code,



## The problem we faced through this work and how we solved them:

In python tkinter the UI design is a hassle, there are limited no. of UI design tools which resulted the UI to look old school. Apart from that there were no major problems in the coding front.

The second major problem we faced was during the conversion phase of the .py to .exe format. The problem was that some **dynamic link library (.dll)** were missing in the converted exe file directory. This happened due to the windows defender removed those files because it saw those libraries as unidentified software fragments. The solution was simple we added an exception rule for those particular libraries, which was identifying the system that those files are actually a part of our software.

The last problem we faced during our application front development which was the creation of the installer wizard file (setup file). There are not a lot of functional tools available for the creation of the exe installer file. Most of the good tools were paid tools with complex user interface. The fix was just simple as we found a free software called **Inno Setup Compiler**.

We didn't face much problem during the web design phase of the project. The one and only problem we faced was the java server on which the web server is running sometimes didn't update the code so the codes remain same as the previous time so we have to restart the server multiple time to see the changes which turned the work efficiency very bad, but we took our time and completed the work within the time frame.

## CONTRIBUTION IN THE PROJECT:

Student information management system lead to a better organization structure since the information management of the students is well structured and also lead to better as well as efficient utilization of resources.

Student Information Management System can be used by education institutes to maintain the records of students easily. Achieving this objective is difficult using a manual system as the information is scattered, can be redundant and collecting relevant information may be very time consuming. All these problems are solved using this project.

Our project Student Management System was developed by all four of us. We, a team of four persons took a step by step approach in order to reach our goal. We applied the knowledge we gained during our study at **Maharaja Manindra Chandra College, university of Calcutta**. and developed this project **"STUDENT MANAGEMENT SYSTEM"**.

## Conclusion:

The results obtained from the experiments and testing ensures that the proposed method is efficient and user-friendly. As compared to existing methods of managing the academic institutions, this project which yields centralized software makes the work administration and management easier and provides detailed information about the topic of user's interest in just one mouse click.

## Future Scope of The Project:

- The Student Management System (SMS) can be enhanced to include some other functionality like online sick leave approve, attendance management.
- Talent management of students based on their performance evaluation can be added.
- Social networking can also be added wherein students can interact with each other.
- Online class functionality can be added.
- Can evolve as an online institution.
- Functionality of chat and messages can be added.
- Online exam functionality can be added.
- Online resume builder functionality can also be added.

## Reference:

**Books:**


- Modern Tkinter for Busy Python Developers (by Mark Roseman)
- Tkinter GUI Programming by Example (by David Love)
- HTML & CSS: Design and Build Web Sites (by Jon Duckett)
- You Don't Know JS: Scope & Closures (by Kyle Simpson)


**Websites:**


- https://www.google.co.in/
- https://en.wikipedia.org/
- https://www.javatpoint.com/python-tkinter
- https://lottiefiles.com/
- https://www.geeksforgeeks.org/python-programming-language/
- https://jrsoftware.org/isdl.php
- https://animate.style/