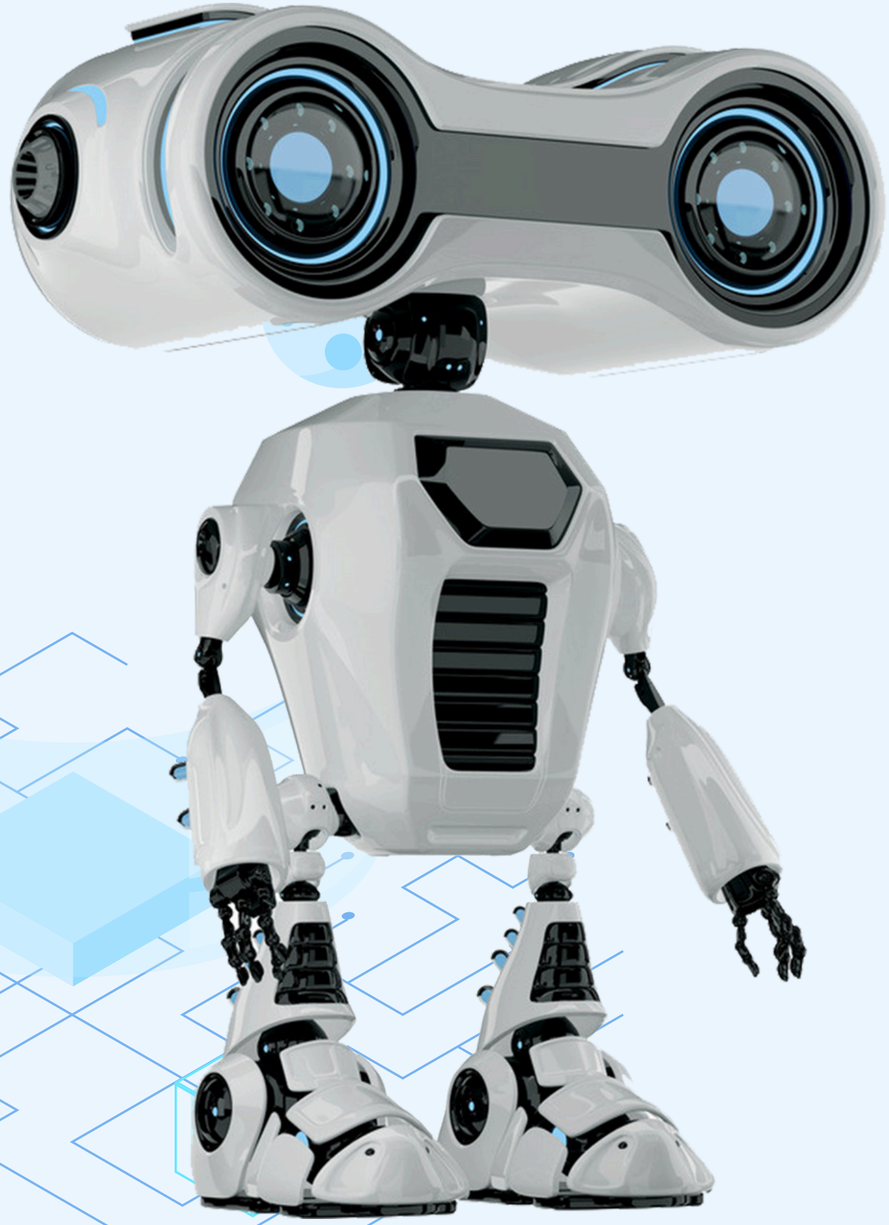


STUDENT MANAGEMENT SYSTEM

-CLASS TRACK

MADE BY:- AASHISH SHAH , UMESH POUDEL, AASHIKA NEUPANE,TILASHA MOKTAN

About Us

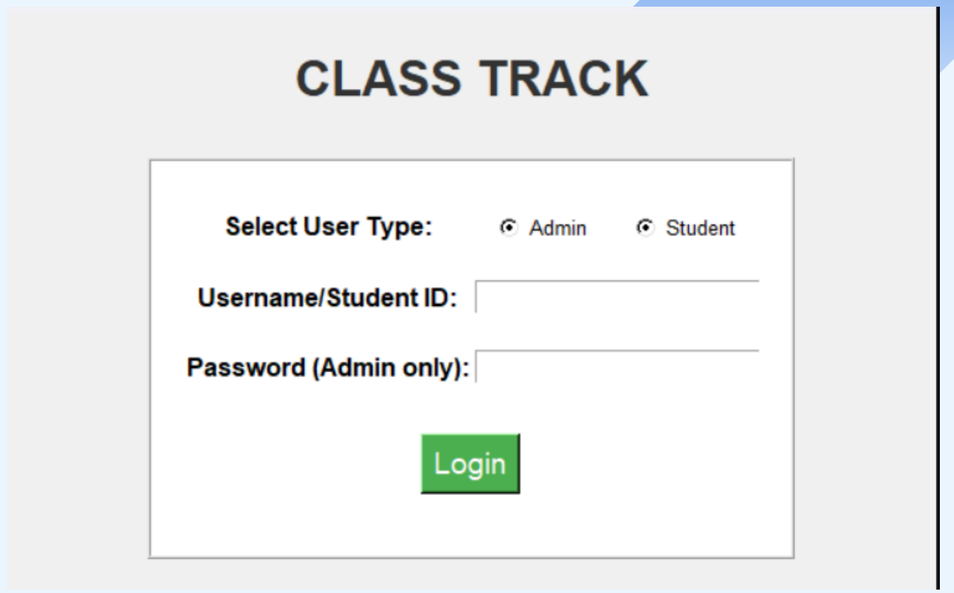
WHAT IS THE CLASS TRACK(STUDENT MANAGEMENT SYSTEM)?

- A SOFTWARE APPLICATION DEVELOPED USING PYTHON'S TKINTER LIBRARY FOR MANAGING STUDENT RECORDS.
- IT ALLOWS ADMINS TO ADD, VIEW, DELETE STUDENTS, AND MANAGE ATTENDANCE AND NOTICE.
- STUDENTS CAN LOG IN TO VIEW THEIR ATTENDANCE AND NOTICE.

WHY PYTHON AND TKINTER?

- PYTHON IS WIDELY USED FOR RAPID DEVELOPMENT.
- TKINTER PROVIDES A SIMPLE WAY TO CREATE GRAPHICAL USER INTERFACES.

WORKFLOW OVERVIEW



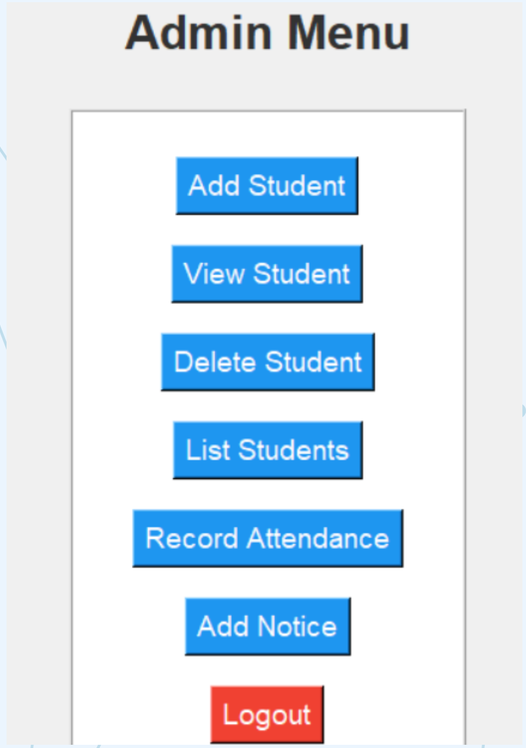
Login System

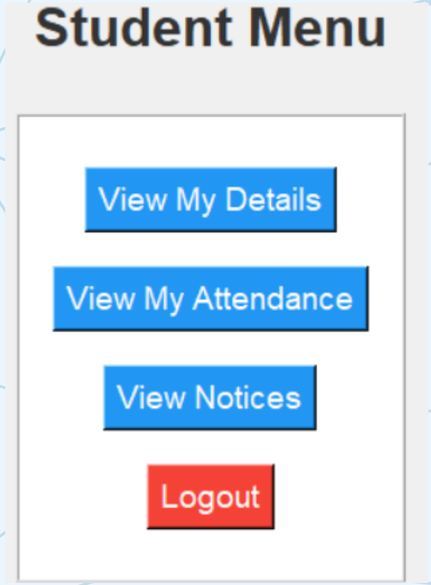
Admin Login:

Secure access to manage student data.

Student Login:

Access limited to viewing personal data, attendance, and results.





Admin Capabilities:

- Add Student:
  - Form to enter and store new student details.
- View Student:
  - Retrieve and display a student's full profile including attendance and notice.
- Delete Student:
  - Remove a student record from the system.
- Record Attendance:
  - Input and save attendance status for students.
- Record Notice:
  - Enter and review academic notice for students.

Student Capabilities:

- View Details:
  - Access personal details like name, age, and major.
- View Attendance:
  - Check attendance records.
- View Notice:
  - Review academic notice.

TIME FOR A QUICK DEMO OF THE CODE

CODE OVERVIEW

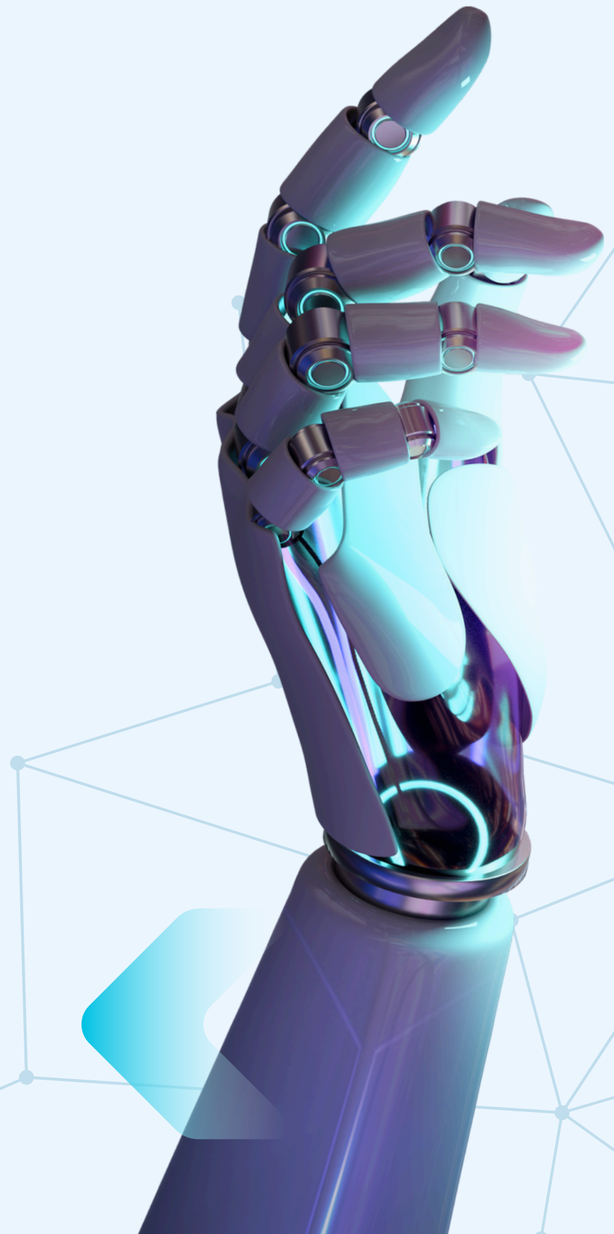
Class: Student

Attributes:

- student_id: (String) A unique identifier for the student.
- name: (String) The student's name.
- age: (Integer) The student's age.
- major: (String) The student's major or field of study.
- attendance: (List of tuples) A list to store attendance records, where each record is a tuple containing the date and attendance status (e.g., ("2024-08-24", "Present")).

Methods:

- __init__(self, student_id, name, age, major): Constructor to initialize the student's attributes.
- add_attendance(self, date, status): Adds an attendance record to the student's attendance list.
- view_attendance(self): Returns the student's attendance records.



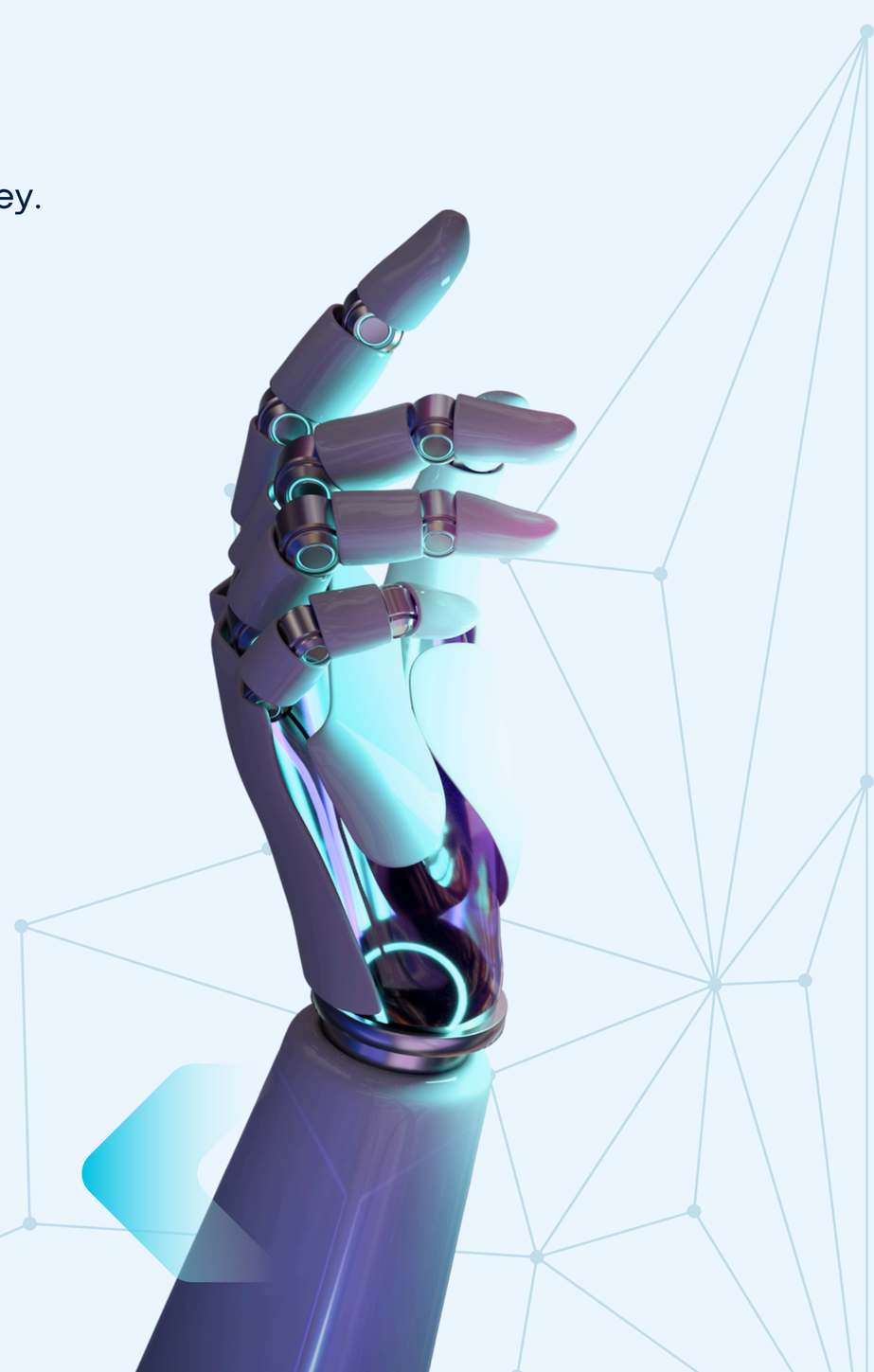
Class: StudentManagementSystem

Attributes:

- root: (Tkinter Tk instance) The main window for the application.
- students: (Dictionary) A dictionary storing Student objects, with student_id as the key.
- current_user: (String or None) Tracks the currently logged-in user (student ID for students, "admin" for the admin).
- notices: (List) A list storing notices (strings) posted by the admin.

Methods:

1. Initialization and Setup:
    - __init__(self, root): Initializes the system, sets up the main window, and loads the login screen.
    - clear_screen(self): Clears the current GUI elements from the window.
2. Login and Menu Screens:
    - login_screen(self): Displays the login screen where users can input their credentials.
    - login(self): Handles the login logic, verifying the credentials and loading the appropriate menu (admin or student).
    - admin_menu(self): Displays the admin's main menu with options like adding students, recording attendance, and managing notices.
    - student_menu(self): Displays the student's main menu with options to view personal details, attendance, and notices.



Admin Functionality:

- add_student_screen(self): Displays the form for adding a new student.
- add_student(self): Handles the logic for creating a new Student object and adding it to the students dictionary.
- view_student_screen(self): Displays the form to input a student ID and view their details.
- view_student(self): Retrieves and displays the details of the student with the specified ID.
- delete_student_screen(self): Displays the form to input a student ID for deletion.
- delete_student(self): Handles the logic for removing a student from the students dictionary based on the entered student ID.
- list_students_screen(self): Displays a list of all students currently in the system.
- record_attendance_screen(self): Displays the form to input a student ID and record attendance.
- record_attendance(self): Adds an attendance record to the specified student's attendance list.
- add_notice_screen(self): Displays the form to input and add a notice.
- add_notice(self): Handles the logic for adding a notice to the notices list.

Student Functionality:

- view_my_details(self): Displays the details of the currently logged-in student.
- view_my_attendance(self): Displays the attendance records of the currently logged-in student.
- view_notices(self): Displays all notices added by the admin.

Utility:

- logout(self): Logs out the current user and returns to the login screen.

Execution

- The code is executed by creating an instance of the Tk class (from Tkinter) and passing it to the StudentManagementSystem class, which manages the entire application flow.

TIME FOR A QUICK LOOK OF THE CODE

CONCLUSION

- Benefits of the System:
  - Efficiency: Simplifies the management of student records.
  - Accessibility: Easy-to-use interface for both admins and students.
  - Scalability: Can be expanded with more features as needed.
- Future Enhancements:
  - Integration with Databases: Storing data in a relational database for better management.
  - User Authentication: Adding more robust security features.
  - Reports and Analytics: Generating detailed reports on attendance and results.
- Final Thoughts:
  - The Student Management System is a practical tool for educational institutions, streamlining the process of managing and accessing student data.

THANK YOU!

Get in Touch With Us

https://github.com/Aashish-Shah-dev

https://github.com/Ankur-1104

https://github.com/Tilasha123

https://github.com/aash-cyber