REQUIREMENTS ANALYSIS

BANNARI AMMAN INSTITUTE OF TECHNOLOGY

AUTHOR

NISHA M

DATE PREPARED

23/04/2024 -01/04/2024

PROJECT ID 02

SEAT NUM 364

PROJECT TITLE

TRAINING PERFORMANCE TRACKER

DOCUMENT AUTHORIZATION MEMORANDUM

TRAINING PERFORMANCE TRACKER

Project Manager

NAME	SIGNATURE	DATE
NISHA M		

Director

NAME	SIGNATURE	DATE
Mr. Ranjith		

TABLE OF CONTENTS

INTRODUCTION

Purpose

- ✓ The proposed P**erformance tracking application** allows students to create profiles and access personalized dashboards tracking academic performance, attendance, skills, projects, programming languages, and achievements. Students update details via a separate form page.
- ✓ Faculty can verify the skills of students under their supervision. Administrators have oversight, accessing all student data to monitor overall institutional performance.
- ✓ Parents gain visibility into their children's dashboards, staying informed about academic progress and accomplishments.
- ✓ The multi-user platform empowers students with self-tracking while enabling faculty verification, administrative monitoring, and parental engagement. It fosters collaboration among stakeholders for streamlined student support and success.

Operating Environment

COMPONENTS	STACK	DESCRIPTION
Backend	Node.js and Express.js	Building the server-side application and handling API requests
Frontend	React	Building the user interface and interacting with the backend API.
Database	MongoDB	A NoSQL document-based database, for storing and retrieving data.
API	OpenAPI	OpenAPI is a language-agnostic specification for describing RESTful APIs

Process flow

TASK NAME	START DATE	END DATE	DURATION (days)	STATUS
Requirement gathering	23-04-2024	23-04-2024	1	
Planning and documentation	24-04-2024	28-04-2024	5	
Entity flow	29-04-2024	30-04-2024	2	
Wireframes	01-05-2024	04-05-2024	4	
Design and prototyping	05-05-2024	15-05-2024	10	
Database design	16-05-2024	25-05-2024	10	
Backend development	26-05-2024	05-06-2024	9	
Testing				
Deployment				

DESCRIPTION

Here are some additional things that can be included in the performance tracking application:

1. **Feedback and Evaluation System**

   Allow faculty to provide detailed feedback and evaluations on student projects, assignments, and overall performance.

2. **Communication and Discussion Forums**

implement communication channels (e.g., messaging, forums) for students to interact with faculty, peers, and mentors.

3. **Goal Setting and Tracking**

Allow students to set academic and personal goals (e.g., target GPA, skill acquisition, project completion).

4. **Peer Evaluation and Collaboration**
5. **Career Readiness and Job Placements**

User Classes and Characteristics

USER CLASS	CHARACTERISTICS
Students	**Basic Information:** Name, Student ID, College/University, Department, Year/Semester **Academic Performance:** Overall GPA/Percentage, Grades for each course, Semester-wise performance **Attendance:** Attendance record for each course, Overall attendance percentage **Skills:** Technical skills (Programming languages, Frameworks, Tools), Soft skills (Communication, Leadership, etc.) **Projects:** Project titles, Descriptions, Technologies used, Roles, and Responsibilities **Programming Languages:** List of programming languages known, Proficiency level for each language **Achievements:** Awards, Winning Certifications, Publications, Competitions, and Extracurricular activities

Faculty	□ **Basic Information:** Name, Faculty ID, Department, Courses taught □ **Student Supervision:** List of students under their supervision □ **Skill Verification:** Ability to verify and approve the skills claimed by students under their supervision □ **Performance Monitoring:** Access to academic performance and attendance records of supervised students □ **Feedback and Evaluation**: Provide feedback and evaluation for student projects, assignments, and overall performance
Administrators	□ **Basic Information:** Name, Administrator ID □ **Student Management:** Access to all student profiles, academic records, and performance data □ **Faculty Management:** Access to faculty profiles and their assigned students □ **Analytics and Reports:** Generate reports and analytics on overall student performance, attendance, and skill distribution □ **System Administration:** Manage user accounts, access rights, and system settings
Parents	□ **Basic Information:** Name, Parent ID, Child's (Student) Information □ **Child's Performance:** Access to their child's academic performance, attendance, skills, projects, and achievements □ **Progress Tracking:** Monitor their child's progress over time □ **Communication**: Ability to communicate with faculty or administrators regarding their child's performance, if necessary

REQUIREMENTS

Include all of the requirements you collected from stakeholders.

Here are the key requirements for the performance tracking application in a concise form:

1. User Management (registration, authentication, roles)
2. Student Dashboard (performance, attendance, skills, projects, achievements)
3. Student Data Entry Form
4. Faculty Portal (student supervision, skill verification, feedback)
5. Administrator Portal (comprehensive data access, reporting, configuration)
6. Parent Portal (view child's performance)
7. Notifications and Alerts
8. Data Security and Privacy
9. Integration and Interoperability
10. Scalability and Performance
11. User Experience and Accessibility
12. Reporting and Analytics
13. Maintenance and Support

Performance Requirements

- Handle lots of users without slowing down
- Fast response times for dashboards and reports
- Real-time data updates
- Ability to grow and scale as needed

Security Requirements

- Access control based on user roles
- Encryption of data in transit and at rest
- Secure authentication like multi-factor or single sign-on
- Regular security checks and testing
- Follow data privacy regulations

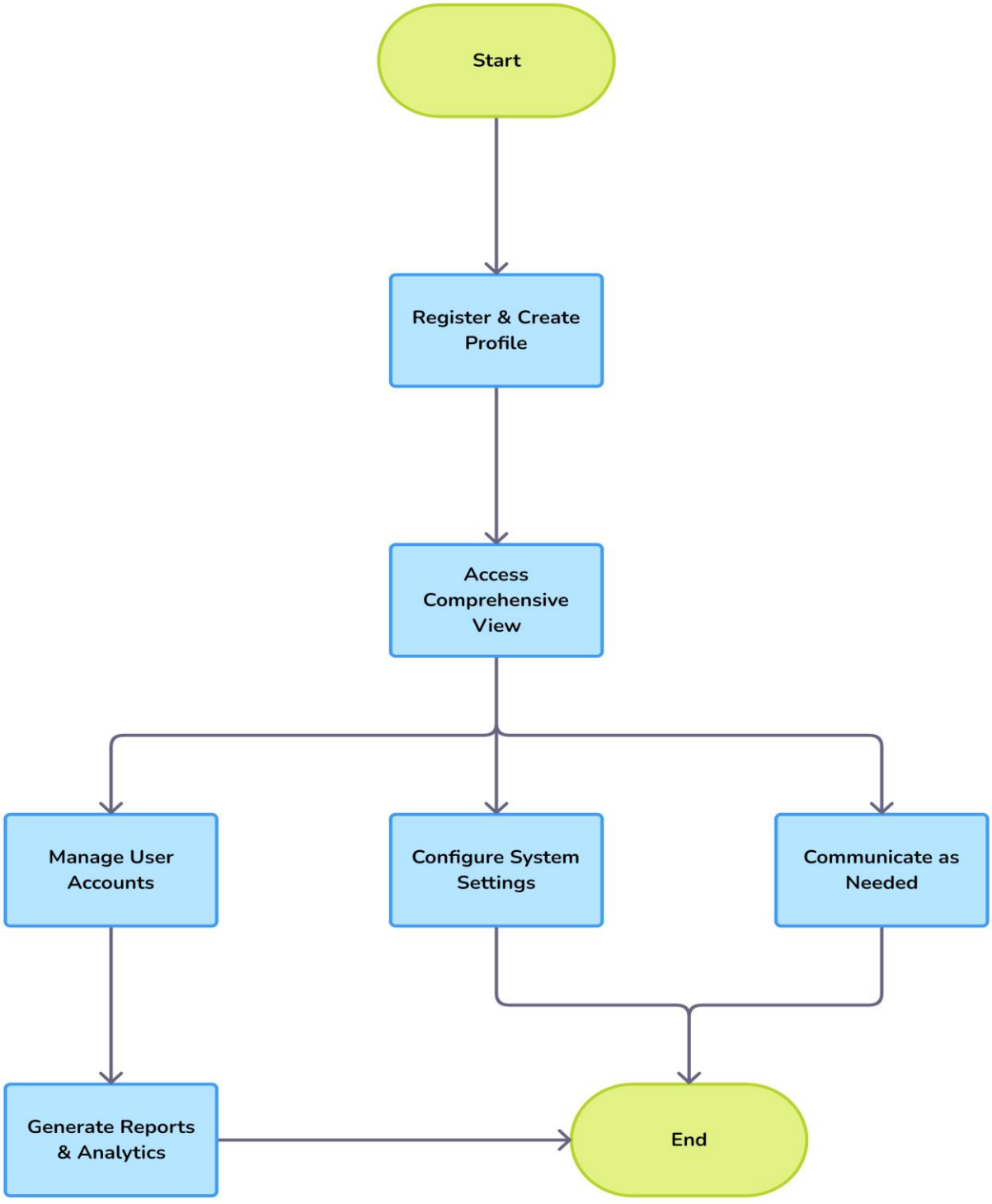
Usability Requirements

- Easy to use interfaces for all users
- Support multiple languages
- User guides and training
- Consistent and appealing design
- Customizable dashboards and reports

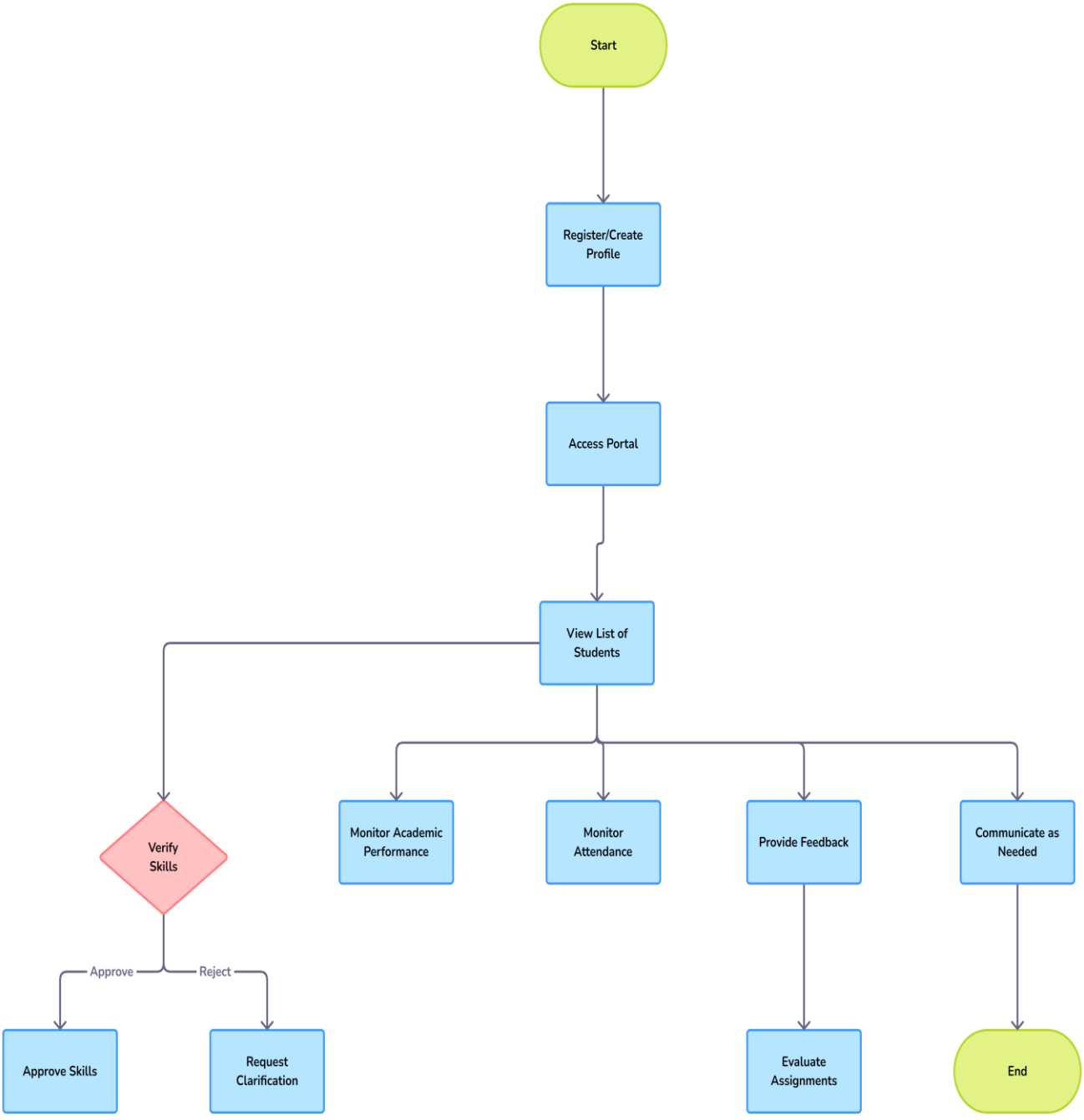
Other Requirements

- Integrate with existing student systems
- Data backup and recovery
- Logging and error handling
- Flexible architecture for future changes
- Automated testing and deployment
- Follow best coding practices

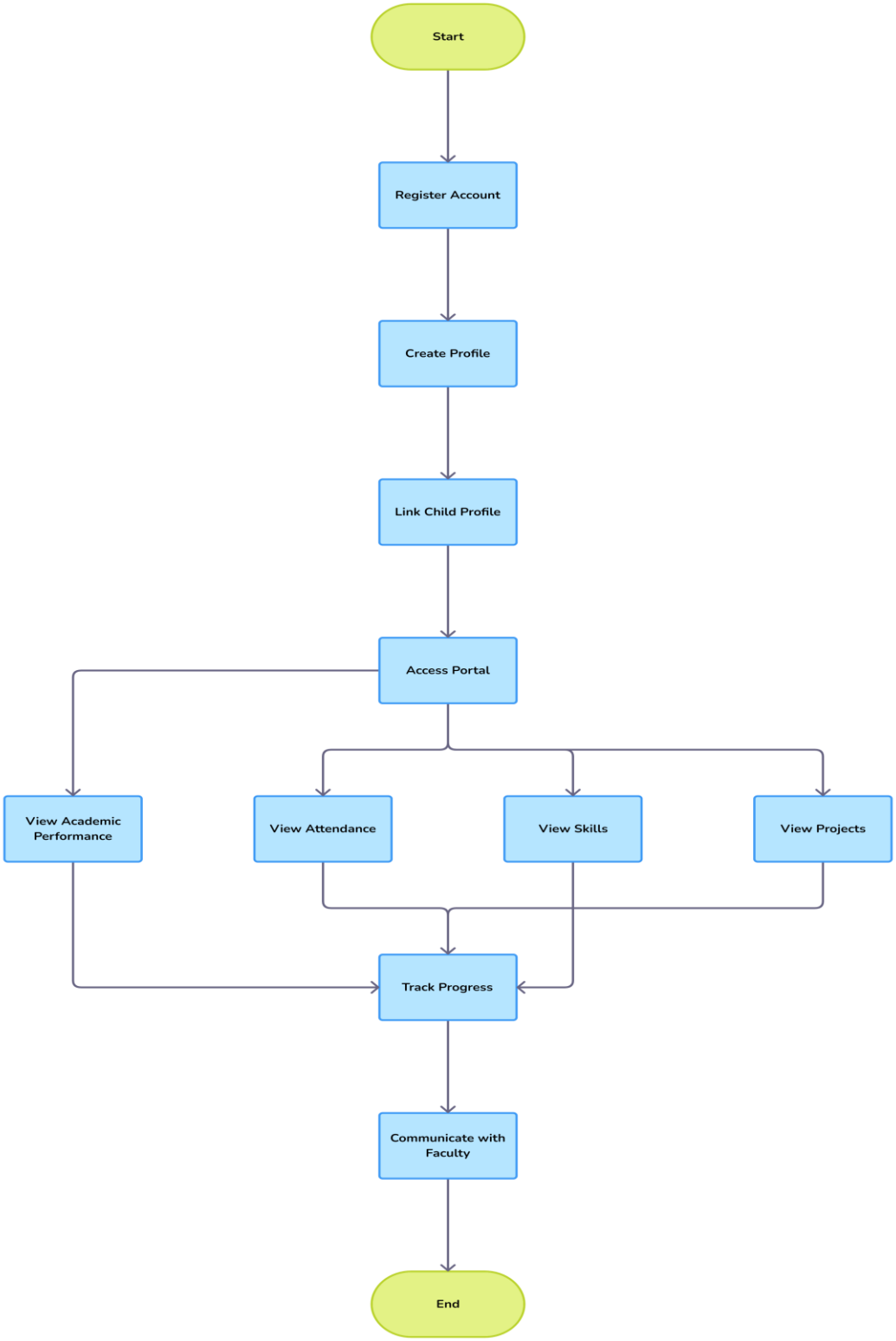
ENTITY A: ADMINISTRATOR



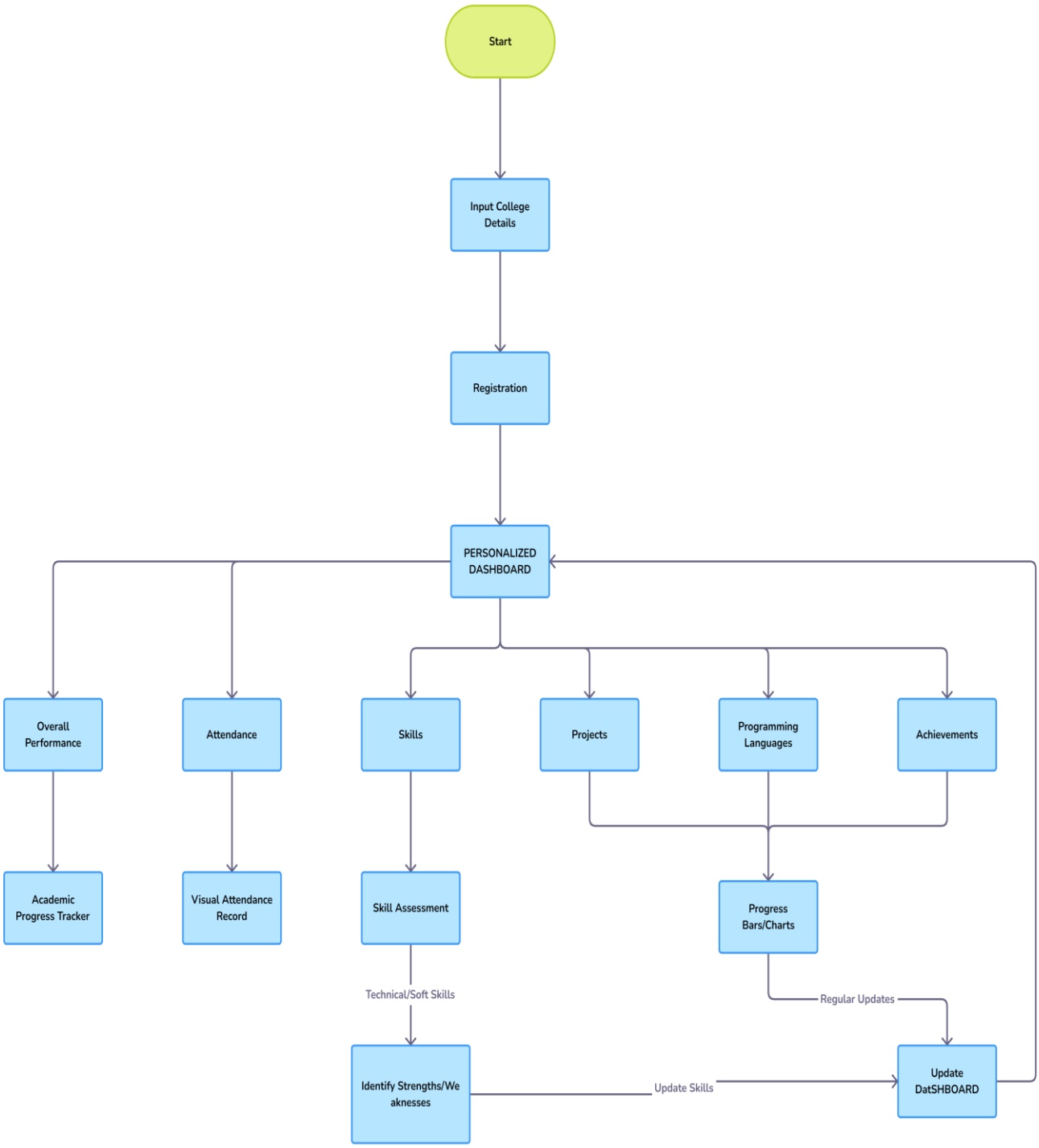
ENTITY B: FACULTY



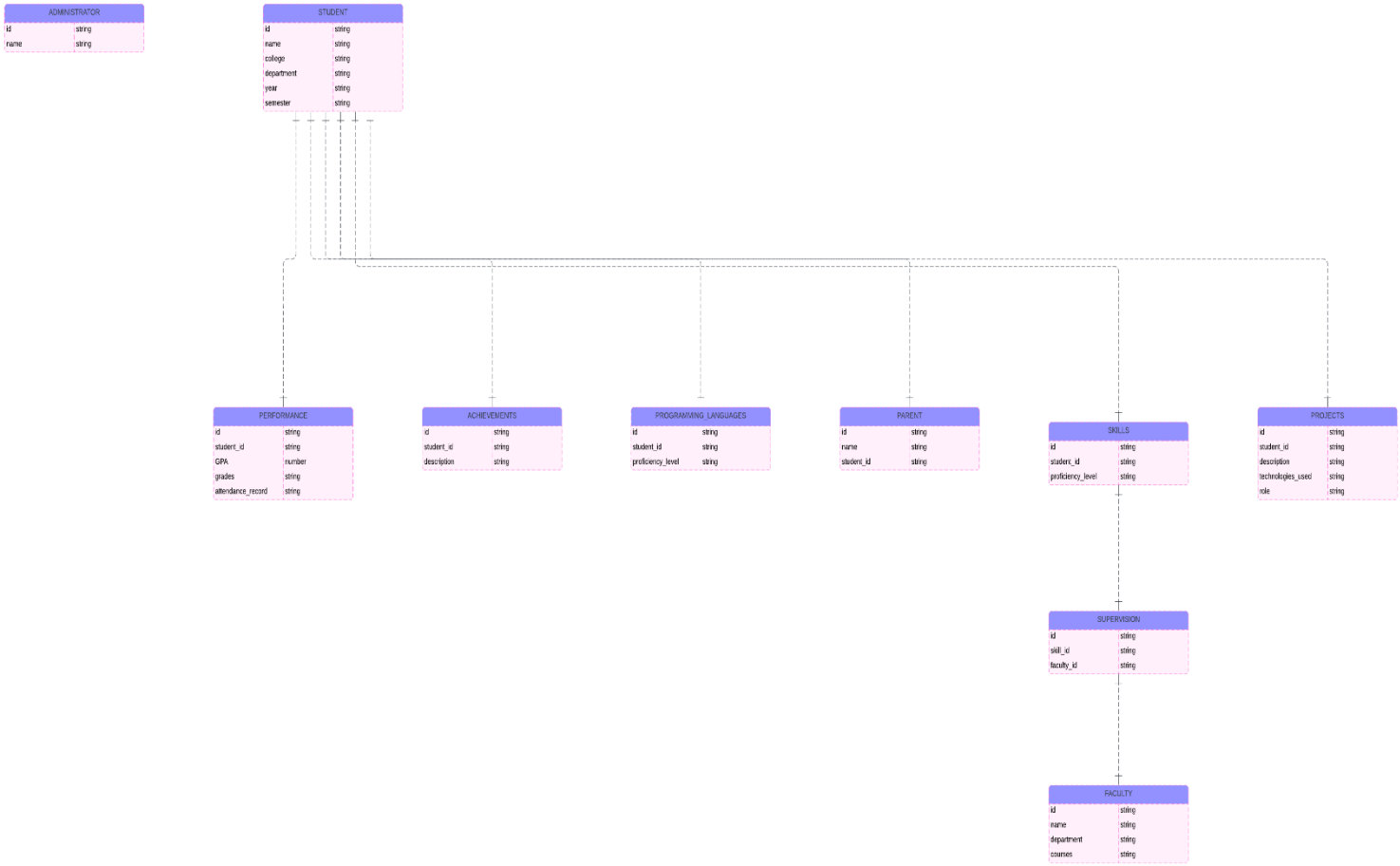
ENTITY C: PARENTS



ENTITY D: STUDENT



ER DIAGRAM


ADMINISTRATOR
id string
name string
STUDENT
id string
name string
college string
department string
year string
semester string
PERFORMANCE
id string
student_id string
GPA number
grades string
attendance_record string
ACHIEVEMENTS
id string
student_id string
description string
PROGRAMMING_LANGUAGES
id string
student_id string
proficiency_level string
PARENT
id string
name string
student_id string
SKILLS
id string
student_id string
proficiency_level string
PROJECTS
id string
student_id string
description string
technologies_used string
role string
SUPERVISION
id string
skill_id string
faculty_id string
FACULTY
id string
name string
department string
courses string