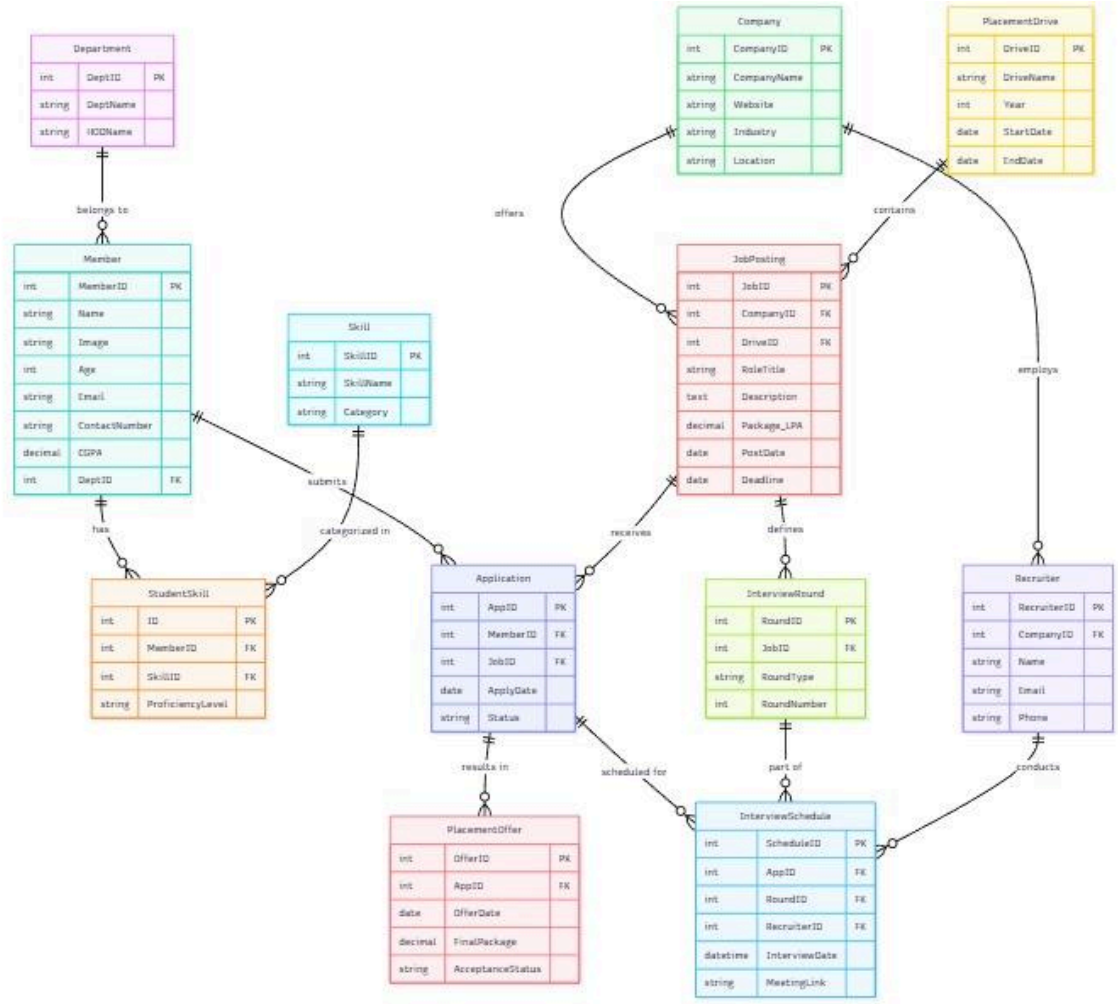
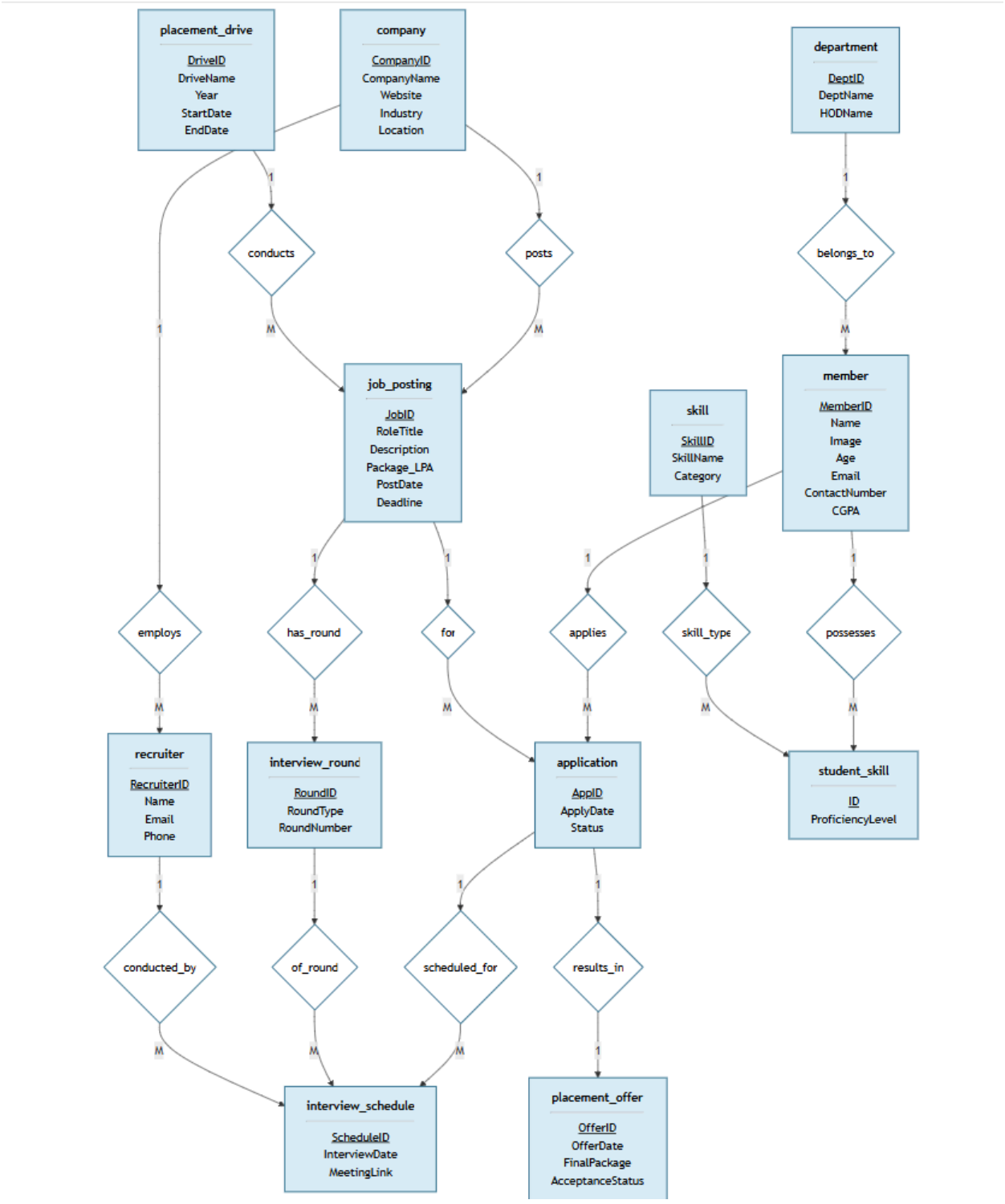
DATABASES
CAREER TRACK – PLACEMENT MANAGEMENT SYSTEM
MODULE-B



ER diagram


placement_drive
DriveID
DriveName
Year
StartDate
EndDate
company
CompanyID
CompanyName
Website
Industry
Location
department
DeptID
DeptName
HODName
conducts
posts
belongs_to
job_posting
JobID
RoleTitle
Description
Package_LPA
PostDate
Deadline
skill
SkillID
SkillName
Category
member
MemberID
Name
Image
Age
Email
ContactNumber
CGPA
employs
has_round
for
applies
skill_type
possesses
recruiter
RecruiterID
Name
Email
Phone
interview_round
RoundID
RoundType
RoundNumber
application
AppID
ApplyDate
Status
student_skill
ID
ProficiencyLevel
conducted_by
of_round
scheduled_for
results_in
interview_schedule
ScheduleID
InterviewDate
MeetingLink
placement_offer
OfferID
OfferDate
FinalPackage
AcceptanceStatus

1.System Overview

The **CareerTrack – Placement Management System** is a database-driven software solution developed to automate and manage the college placement process efficiently.

In many institutions, placement activities are handled manually using spreadsheets, paper records, and emails. This often leads to:

- Data duplication
- Inconsistent records
- Difficulty in tracking applications
- Delays in communication
- Lack of centralized data

The CareerTrack system provides a structured and centralized platform where:

- Student details are stored and managed
- Companies can register and post job roles
- Students can apply for jobs
- Placement officers can monitor the complete process
- Final placements are recorded and tracked

This system ensures transparency, efficiency, accuracy, and better coordination between students, companies, and placement officers.

2. Objectives

The primary objectives of the CareerTrack system are:

1. To create a centralized database for placement management.
2. To automate the job application process.
3. To track student eligibility and placement status.
4. To manage company and job role details.
5. To ensure data consistency and integrity.
6. To reduce manual errors in placement records.
7. To provide real-time tracking of applications and selections.

3. Functional Description

The system performs the following major functions:

Student Registration and Management

- Add new student records
- Store academic details (Branch, CGPA, Graduation Year)
- Update student information
- View student placement status

Company Registration and Management

- Add company details
- Store company contact information
- Track companies visiting campus
- View company hiring history

Job Posting and Management

- Add job roles offered by companies
- Define eligibility criteria (minimum CGPA)
- Mention job package and job type
- Link jobs to companies

Application Processing

- Students apply for eligible jobs
- Store application date
- Track application status (Applied / Selected / Rejected)

Placement Recording

- Record final selected students
- Store company and role details
- Record placement date
- Ensure only one final placement per student

4. System Architecture Explanation

The system follows a **Three-Tier Architecture**:

1. Presentation Layer (User Interface)

- Used by Students, Companies, and Placement Officers.
- Provides forms for:
    - ➢ Registration
    - ➢ Login
    - ➢ Applying for jobs
    - ➢ Viewing status
- Can be developed using HTML/CSS/Java (or any frontend tool).

2. Application Layer (Business Logic)

- Handles:
    - ➢ Eligibility checking
    - ➢ Application processing
    - ➢ Validation of input data
    - ➢ Updating placement results

- Ensures business rules are followed.

Example:

- Students cannot apply if CGPA < required CGPA.
- Students cannot get multiple final placements.

3. Database Layer (Data Storage)

- Stores all system data in relational tables
- Maintains relationships using primary and foreign keys.
- Ensures data integrity.

This layered architecture improves:

- Security
- Scalability
- Maintainability
- Performance

5. Roles and Role-Based Operations

The system supports different user roles with specific permissions.

Student Role

Operations:

- View job postings
- Check eligibility
- Apply for jobs
- View application status
- View final placement result

Students cannot:

- Modify other students' data
- Add companies
- Delete job roles

Company Role

Operations:

- Post job roles
- Define eligibility criteria
- View applicant list
- Select students
- Confirm placements

Companies cannot:

- Edit student academic records
- Delete other company data

Placement Officer / Admin Role

Operations:

- Add and update student records
- Add and manage company details
- Add job postings
- Monitor applications
- Update placement status
- Generate reports
- Maintain database consistency

6. Use Case Explanation

Below are the major use cases explained clearly:

Use Case 1: Student Applies for a Job

Actors: Student

Steps:

1. Students log into the system.
2. The system displays available job roles.
3. Students select a job.
4. System checks eligibility (CGPA requirement).
5. If eligible → application is submitted.
6. Application status is stored as "Applied".

Use Case 2: Company Selects Students

Actors: Company

Steps:

1. Company logs in.
2. Views list of applicants for a job.
3. Selects eligible students.
4. System updates application status as "Selected".
5. Placement records are created.

Use Case 3: Admin Adds New Job

Actors: Placement Officer

Steps:

1. Admin logs in.
2. Selects "Add Job".
3. Enters job details and eligibility criteria.
4. The job is linked to a company.
5. The system stores job records in a database.

7. Data Flow Explanation

The system follows this data flow:

Step 1: Data Input

- Students enter profile data.
- Admin enters company and job data.
- Companies update selection results.

Step 2: Processing

- System validates:
    - ➢ CGPA eligibility
    - ➢ Duplicate applications
    - ➢ Foreign key relationships
    - ➢ Application status is updated.

Step 3: Storage

- Data is stored in relational tables:
    - ➢ Student
    - ➢ Company
    - ➢ Job
    - ➢ Application
    - ➢ Placement

Step 4: Output

- Application status displayed
- Placement reports generated
- Student placement summary provided

8. Database Explanation

The system uses a relational database with the following tables:

Student Table

Primary Key: Student_ID
Stores student academic and personal details.

Company Table

Primary Key: Company_ID
Stores company information.

Job Table

Primary Key: Job_ID
Foreign Key: Company_ID

Relationship:
One Company → Many Jobs (1:M)

Application Table

Primary Key: Application_ID
Foreign Keys:

- Student_ID
- Job_ID

Resolves Many-to-Many relationship between Student and Job.

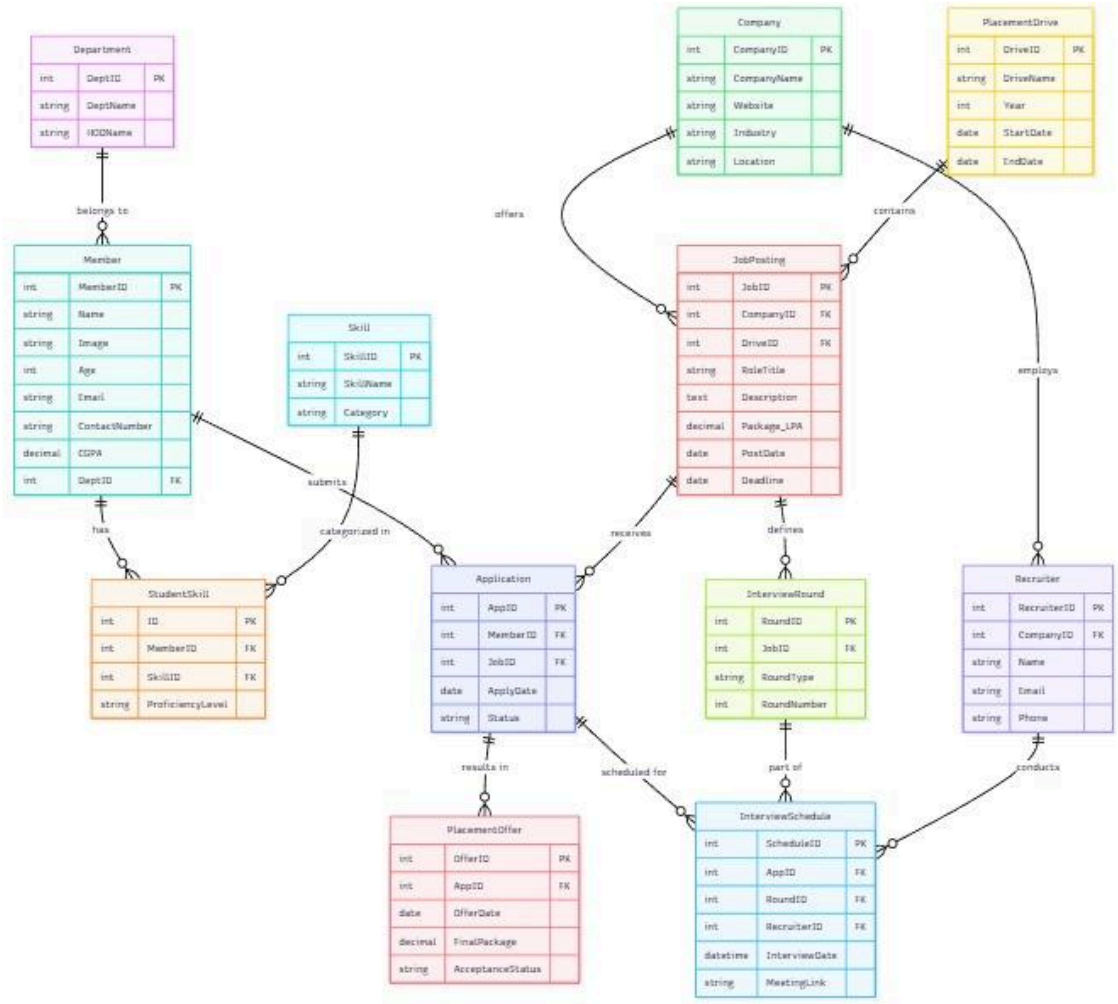
Placement Table

Primary Key: Placement_ID
Foreign Keys:

- Student_ID
- Company_ID
- Job_ID

Ensures:

- One student → One final placement
- One company → Many placements



9. Data Integrity

The system maintains data integrity using:

1. Primary Keys

- Uniquely identify each record.
- Prevent duplicate entries.

2. Foreign Keys

- Maintain relationships between tables.
- Prevent invalid references.

Example:
A job cannot exist without a valid company.

3. NOT NULL Constraints

- Important fields cannot be empty.
- Ensures completeness of data.

4. Unique Constraints

- Prevent duplicate email IDs.

5. Referential Integrity

- Deleting a company is restricted if jobs exist.
- Ensures database consistency.

6. Normalization

The database follows 1NF, 2NF, and 3NF.
This reduces redundancy and ensures logical data storage.

10. How the Database Supports the System

The database is the backbone of the CareerTrack system.

It supports the system by:

1. Storing all structured placement data.
2. Maintaining relationships between students, companies, and jobs.
3. Enforcing eligibility and constraints.
4. Providing accurate placement tracking.
5. Supporting report generation.
6. Preventing data duplication.
7. Ensuring secure and consistent data access.

Without a properly designed relational database

- Data inconsistency would occur.
- Duplicate placements could happen.
- Application tracking would be difficult.

Thus, the database design directly supports system functionality, integrity, and performance.

11. Conclusion

The CareerTrack – Placement Management System provides a structured, efficient, and scalable solution for managing college placement activities.

The system:

- Automates application tracking
- Maintains strong entity relationships
- Ensures data integrity through constraints
- Supports role-based access
- Follows proper database normalization
- Uses layered architecture for scalability

The relational database design ensures reliability, accuracy, and transparency in placement management, making it suitable for real-world implementation in educational institutions.

Team Members and Contributions

CareerTrack – Placement Management System

The project was completed collaboratively by five team members. Each member contributed significantly to different phases of the system design, database implementation, and documentation.

1. Bhavitha Somireddy -24110350

- Worked on Module B – Conceptual Design.
- Contributed to analysing entities, attributes, and relationships based on project requirements.
- Assisted in defining primary keys, foreign keys, and relationship types.
- Helped in preparing explanations for schema alignment and relationship justification.

2. Killada Eswara Valli – 24110165

- Worked on Module B – Conceptual Design.
- Focused on converting the conceptual model into a proper ER structure.
- Ensured correct representation of cardinality (1:1, 1:M, M:N).
- Verified consistency between the conceptual design and implemented database schema.

3. Garv Singhal - 24110119

- Worked on Module B – ER Diagram (Module 2)
- Designed the ER diagram using standard notation.
- Clearly marked Primary Keys and Foreign Keys.
- Ensured the ER diagram correctly reflects the SQL tables from Module A.

4. Pramith Joy - 23110152

- Worked on Module A – Database Implementation.
- Created SQL tables with proper Primary Keys and Foreign Keys.
- Applied NOT NULL and logical constraints.
- Ensured referential integrity during update and delete operations.
- Inserted meaningful real-life sample data.

5. Divyansh Saini - 23110101

- Worked on Module A – SQL Dump and Testing.
- Generated and finalized the SQL dump file for submission.
- Performed data validation and query testing.
- Verified that all Module A constraints and minimum requirements were satisfied.
- Assisted in debugging and final project verification.