**Requirements Document for an Educational/Training Management System**

**1. Introduction**

This document outlines the functional and data requirements for a system based on the provided Entity-Relationship Diagram (ERD). The system aims to manage various aspects of an educational or training institution, including students, instructors, courses, tracks, exams, and graduations, as well as job placement.

**2. System Scope**

The system will encompass the management of:

* Student information and enrollment.
* Instructor details and their assigned courses/tracks.
* Course and track definitions.
* Exam creation, administration, and grading.
* Graduation records and certificate issuance.
* Job placement details for graduates.
* Master data for various fields (e.g., country, university, field).

**3. Data Requirements (Entities and Attributes)**

This section details the required data entities and their respective attributes as derived from the ERD.

**3.1. Master Data Entities**

* **Master (Lookup for various fields)**
  + M\_ID (Primary Key)
  + M\_Title
  + Country
  + University
  + Field
* **Dependents**
  + Dep\_ID (Primary Key)
  + Gender
  + Age
  + Name
* **Instructors**
  + Inst\_ID (Primary Key)
  + City
  + Zip-Code
  + Street
  + Phone
  + Email
  + Fname (First Name)
  + Lname (Last Name)
  + Age
  + BirthDate
  + HiringDate
  + Salary
* **Tracks**
  + Track\_ID (Primary Key)
  + Track\_Name
  + Track\_Duration
* **Courses**
  + Course\_ID (Primary Key)
  + Course\_Name
  + C\_Status (Course Status)
  + C\_Duration (Course Duration)
* **Topics**
  + Topic\_ID (Primary Key)
  + Topic\_Name
* **Exams**
  + Exam\_ID (Primary Key
  + Exam name
  + duration
  + Exam\_Date
  + Remark
  + Level
* **Questions**
  + Question\_ID (Primary Key)
  + Question\_Txt
  + Question\_Type
  + Question\_Level
  + Correct\_Answer
  + Answer choice (many)
* **Certificates**
  + Cert\_ID (Primary Key)
  + Cert\_Name
  + Acquired\_From (Where the certificate was acquired from)
  + Provider name
* **Students**
  + Stud\_ID (Primary Key)
  + Address(Zip-Code-City- Street)
  + Phone
  + Email
  + Fname (First Name)
  + Lname (Last Name)
  + Age
  + Gender
  + BirthDate
  + LinkedIn\_URL
* **Intake**
  + Intake\_ID (Primary Key)
  + Intake\_Name
  + StartDate
  + EndDate
* **Round**
  + Round\_ID (Primary Key)
  + Round\_Name
  + StartDate
  + EndDate
* **Graduates**
  + Grad\_ID (Primary Key)
  + Git\_URL
  + LinkedIn\_URL
  + grad\_Name (Graduate's name)
* **Company**
  + C\_ID (Primary Key)
  + C\_Name
  + C\_City
  + C\_type
* **Faculty**
  + F\_code
  + F\_name
  + F\_City
* **Freelance**
  + F\_ID (Primary Key)
  + COST
  + Duration

**4. Data Relationships (Functional Requirements)**

This section describes the relationships between entities, implying functional requirements for how data will be linked and managed. The cardinality (e.g., 1:M, M:N) indicates the nature of the relationship.

**4.1. Instructor and Management Relationships**

* **Instructor manages Instructor (Manager)**: An instructor can supervise multiple instructors, and an instructor can be supervised by one instructor (1:M recursive relationship).
  + *Requirement:* The system must allow assigning a manager to an instructor and retrieving all instructors managed by a specific instructor.
* **Instructor Teaches Track**: An instructor can teach multiple tracks, and a track can be taught by multiple instructors (M:M).
  + *Requirement:* The system must allow assigning instructors to tracks and vice versa.
* **Instructor Teaches Course**: An instructor can teach multiple courses, and a course can be taught by multiple instructors (1:M).
  + *Requirement:* The system must allow assigning instructors to courses and vice versa.
* **Instructor has Dependents**: An instructor can have multiple dependents, and a dependent belongs to one instructor (1:M).
  + *Requirement:* The system must allow adding and managing dependent information for each instructor.

**4.2. Course and Track Management**

* **Track Contains Courses**: A track contains multiple courses, and a course belongs to one track (M:M).
  + *Requirement:* The system must allow defining tracks and assigning courses to them.
* **Course Has Topics**: A course has multiple topics, and a topic belongs to one course (1:M).
  + *Requirement:* The system must allow defining topics for each course.

**4.3. Exam Management**

* **Questions generate Exams**: questions generatemultiple An exam, and a question can be part of multiple exams (M:M).
  + *Requirement:* The system must allow associating questions with exams and retrieving questions for a specific exam.
* **course include Exams**: A course can include multiple Exams, and an Exam belongs to one course(1:M).
  + *Requirement:* The system must allow defining multiple answers for a question, marking the correct one, and storing individual answer grades.
* **Students answers questions in an exam**: (1:M:M).
  + *Requirement:* The system must allow creating exams from questions.

**4.4. Student Management**

* **Students Enrolled-in Round**: A student can be enrolled in one round, and a round can have multiple students (1:M).
  + *Requirement:* The system must allow enrolling students into specific intake rounds.
* **Students Takes Exams**: A student can take multiple exams, and an exam can be taken by multiple students (M:M). This relationship also has an attribute: Exam\_Percentage.
  + *Requirement:* The system must record which exams a student has taken and store the percentage achieved for each exam.
* **Students Gets Certificates**: A student can get multiple certificates, and a certificate can be obtained by multiple students (M:M).
  + *Requirement:* The system must record which certificates a student has obtained.
* **Students Enrolled-in Intake**: A student can be enrolled in one intake, and an intake can have multiple students (1:M).
  + *Requirement:* The system must allow enrolling students into specific intakes.

**4.5. Graduation and Placement Management**

* **Graduates Master**: A graduate is linked to multiple master data entries (e.g., country, university, field), and a master entry can be linked to multiple graduates (M:M).
  + *Requirement:* The system must allow associating graduates with relevant master data such as their country, university, and field of study.
* **Graduates Works-at Company**: A graduate can work at one company, and a company can have multiple graduates (1:M).
  + *Requirement:* The system must record the company where a graduate is employed.
* **Company Has Job\_Titles**: A company can have multiple job titles, and a job title can be offered by multiple companies (M:M).
  + *Requirement:* The system must allow defining job titles associated with companies.
* **Graduates Has Job\_Titles**: A graduate can have one job title, and a job title can be held by multiple graduates (1:M).
  + *Requirement:* The system must record the specific job title held by a graduate.
* **STDENT Take Freelance**: A graduate can undertake multiple freelance cases, and a freelance case can be taken by one graduate (1:M).
  + *Requirement:* The system must allow recording freelance projects undertaken by graduates.

**5. User Roles (Implicit)**

Based on the ERD, potential user roles and their associated functionalities include:

* **Administrator/System Manager:** Full access to manage all entities, including creating/updating instructors, students, courses, tracks, exams, and master data.
* **Instructor:** Manage courses/tracks they teach, create/manage exams for their courses, view student performance.
* **Student:** View enrolled courses/tracks, take exams, view grades, manage personal information.
* **Placement Officer:** Manage graduate information, record job placements, manage company and job title data.

**6. Non-Functional Requirements (Initial Considerations)**

* **Performance:** The system should be responsive for common operations (e.g., retrieving student records, generating exam results).
* **Security:** Data must be protected against unauthorized access. Role-based access control (RBAC) should be implemented.
* **Scalability:** The system should be able to handle a growing number of students, instructors, courses, and exams.
* **Usability:** The user interface should be intuitive and easy to navigate for all user roles.
* **Data Integrity:** Mechanisms must be in place to ensure the accuracy and consistency of data (e.g., foreign key constraints).
* **Backup and Recovery:** Regular data backups and a recovery plan are essential.