Week 1 Project Report: Requirements Gathering & Business Scoping

Project: Techmentee 702 - College Placement Management Portal

Phase Focus: Understanding the "What" and "Why"

1. Executive Summary

The objective of this platform is to digitize and streamline the campus placement ecosystem. Currently, tracking student applications, recruiter interactions, and placement metrics is fragmented. This system will centralize operations into a single, role-based platform, providing real-time visibility to the Placement Cell (Admin), Students, and Recruiters.

2. Stakeholder & Role Definition

We have identified three distinct user personas, each requiring isolated environments and specific data access:

- **Students (The Applicants):** Need a frictionless way to build profiles, apply for opportunities, and track their application lifecycle.
- **Recruiters (The Demand):** Require an efficient pipeline to post jobs, filter verified student data, and update interview statuses.
- **Admin / TPO (The Overseers):** Act as the gatekeepers. They need absolute control over user verification, drive scheduling, and high-level reporting.

3. Core Functional Workflows

Based on the documentation, the system must support two primary operational workflows:

1. **The Off-Campus / Standard Job Workflow:** Continuous posting of jobs by recruiters $\rightarrow$ Admin approval $\rightarrow$ Student application $\rightarrow$ Pipeline tracking.
2. **The On-Campus Drive Workflow:** Event-based creation of placement drives by Admin $\rightarrow$ Bulk student registration $\rightarrow$ On-the-day attendance and mass status updates.

4. Defining the Analytical KPIs

To ensure we build the right database architecture in Week 2, we must define what we want to measure at the end of the project. The primary KPIs are:

- **Placement Efficiency:** % of eligible students placed (overall and by department).
- **Financial Metrics:** Highest, Average, and Median salary packages.
- **Engagement Metrics:** Number of companies visiting vs. offers rolled out.
- **Pipeline Drop-off:** At what stage (Aptitude, Technical, HR) are students failing?

Week 2 Project Report: System Architecture & Data Modeling

Project: Techmentee 702 - College Placement Management Portal

Phase Focus: Understanding the "How" (Database & Logic Design)

1. Entity-Relationship (ER) Foundation

To support the workflows defined in Week 1, we need a robust relational database model. Here are the core tables (entities) and their relationships:

- **Users Table:** The master authentication table (ID, Role, Email, Password Hash, Active Status).
- **Student_Profiles Table:** Linked 1:1 with Users. Stores CGPA, Dept, Skills array, and Resume link.
- **Recruiter_Profiles Table:** Linked 1:1 with Users. Stores Company Name, Industry, and Verification Status.
- **Jobs Table:** Linked 1:N with Recruiter_Profiles. Stores Role, Criteria, and Salary.
- **Job_Applications Table:** The critical junction table mapping Students to Jobs.
- **Placement_Drives Table:** Managed by Admins, mapping multiple Jobs and Recruiters to a specific event date.

2. Application State Management

A key technical challenge is tracking where a student is in the hiring process. We will implement a strict "State Machine" for the Application_Status column in the database:

- **State 0:** Applied
- **State 1:** Under Review (Viewed by Recruiter)
- **State 2:** Shortlisted_for_Interview
- **State 3:** Interview_Scheduled
- **State 4:** Offered OR Rejected
- *Rule:* The system must timestamp every state change to calculate metrics like "Time-to-Hire."

3. Security & Access Control Mapping

Data privacy is paramount, especially regarding student academic records and offer letters.

- **Row-Level Security (RLS):** A student can only query the Job_Applications table where Student_ID = their_own_id.
- **Admin Gatekeeping:** Recruiters cannot post jobs directly to the live feed. Their POST requests write to a staging status (Pending_Approval), which must be flipped to Published by an Admin.
- **File Storage:** Resumes and Offer Letters will be stored in secure cloud buckets (e.g., AWS S3), with the database merely storing the encrypted URL paths.

Moving Forward

With the business requirements (Week 1) and the relational database model (Week 2) clearly defined, the data is structured perfectly to feed into the analytical pipelines and dashboards we designed in Week 3.