Week 3 Project Report: Data Analytics & Pipeline Architecture

Project: Techmentee 702 - College Placement Management Portal

Role: Lead Data Analyst

1. Data Strategy & Entity Mapping

To generate the required reports (Placement %, Packages, Selection rates), we first need to extract data from the portal's transactional database (likely SQL-based) and map it into an analytical schema (Star Schema).

We will consolidate the raw data into the following core tables:

- **Dim_Student:** Student ID, Dept, 10th/12th/Degree scores, CGPA, Standardized Skills.
- **Dim_Company:** Recruiter ID, Company Name, Industry.
- **Dim_Job:** Job ID, Role, Location, Salary Package, Required Skills.
- **Fact_Applications:** The core transactional table tracking Student ID, Job ID, Application Date, Interview Status, and Final Outcome (Offered/Rejected).
- **Fact_Drives:** Tracking Drive ID, Company ID, Date, Total Registrations, and Attendance.

2. Data Preprocessing & Quality Assurance (QA)

User-inputted data is notoriously messy. Before this data reaches the Admin Dashboard, we must implement automated preprocessing scripts (e.g., using Python/Pandas or SQL stored procedures) to handle the following:

- **Academic Score Normalization:** Students might enter academic scores differently (e.g., CGPA out of 10 vs. percentages out of 100). We will implement a standardization rule (e.g., `Percentage = CGPA * 9.5` or whatever the specific university standard is) to create a uniform `Normalized_Score` column for filtering.
- **Skill Standardization (Entity Resolution):** Recruiters might ask for "ReactJS" while students input "React.js" or "React". We will implement string-matching or a predefined taxonomy to group these skills so search filters work accurately.
- **Handling Nulls/Missing Data:**

    - *Unplaced Students:* Salary fields will be NULL. We must ensure these are excluded from "Average Package" calculations rather than treated as $0, which would heavily skew the metrics.
    - *Incomplete Profiles:* Flag profiles missing resumes or 10th/12th marks to trigger automated "Profile Incomplete" notifications.
- **Salary Formatting:** Convert all salary inputs (LPA, CTC, monthly stipends) into a standardized annual numeric format (e.g., 5,50,000) to enable accurate "Highest/Avg Package Stats" calculations.

3. Feature Engineering

To provide advanced insights beyond basic counts, we will engineer new data points (features) from the raw data:

- **Placement_Status_Binary:** A simple 1 (Placed) or 0 (Unplaced) derived from the Offer Letter upload status. This makes calculating "Placement Percentage by Department" computationally lightweight.
- **Application_to_Offer_Ratio:** (Total Offers / Total Applications). This helps the Admin identify students who are applying to many jobs but failing interviews (indicating a need for interview prep).
- **Skill_Gap_Index:** A calculated metric comparing a student's listed skills against the skills most frequently requested by recruiters in Dim_Job. This can inform the college about what modern courses to teach.

4. Technical Analytical Workflow (The Pipeline)

Here is the step-by-step flow of how data moves from a user clicking "Apply" to the Admin seeing a chart on their dashboard.

Step 1: Data Ingestion (Transactional)

- Students, Recruiters, and Admins interact with the portal.
- Data is recorded in real-time in the primary transactional database (OLTP) following the workflows outlined in Modules 6 & 7.

Step 2: Extraction & Transformation (ETL Process)

- **Schedule:** A daily batch job runs every night at 12:00 AM.
- **Action:** It pulls new registrations, application status changes, and drive attendance.
- **Processing:** Applies the preprocessing rules (normalizing CGPAs, standardizing salaries, handling nulls) and engineers our new features.

Step 3: Data Loading (Analytical Warehouse)

- The cleaned data is loaded into the analytical tables (Fact_Applications, Dim_Student, etc.).

Step 4: Reporting & Visualization (Module 8)

- The Admin Dashboard queries these clean tables to generate real-time metrics.
- **Outputs generated:**
    - *Bar Charts:* Company-wise Student Selection.
    - *KPI Cards:* Highest Package, Average Package, Total Placed %.
    - *Data Tables:* Auto-updating "Unplaced Student List" for follow-up.

Lead Analyst Note:

This structure ensures that by the time data hits the Admin's "Reports & Analytics" module, it is clean, accurate, and mathematically sound. It also prevents the live application from slowing down, as heavy calculations are done on the analytical side, not the transactional side.