MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

Software Engineering Lab

Campus Placement Management System (CPMS)

Date: 28-08-2025

Team Members:

Section CSE – A – Batch 1

- Krishna Agarwal – Reg. No: 230905028 – Roll No: 6
- Devadathan N R – Reg. No: 230905010 – Roll No: 4
- Ritanshu Kumar – Reg. No: 230905216 – Roll No: 26

Lifecycle Model:

We have chosen the **Incremental Development Model** for developing the Campus Placement Management System (CPMS) because:

1. **Modular Structure** – The system is divided into multiple independent modules (User Management, Drive Management, Eligibility, Results, Analytics) which can be developed and delivered in increments.

2. **Early Availability of Core Features** – Essential functionalities like student registration, login, and job postings can be implemented first, allowing users to benefit from the system at an early stage.

3. **User Feedback & Requirement Flexibility** – Since students, companies, and placement officers are stakeholders, their evolving needs can be incorporated in later increments based on feedback.

4. **Reduced Risk** – Early delivery of working modules helps identify technical and functional issues quickly, reducing overall project risk.

5. **Improved Testing and Reliability** – Each increment can be tested thoroughly before integration, ensuring better system stability and reliability.

6. **Support for Enhancements** – Optional and advanced features like dashboards, analytics, and email notifications can be added in future increments without disturbing existing functionality.

Software Requirements Specification (SRS)

1) Introduction

Purpose/Goal:
The purpose of CPMS is to automate and manage the entire campus placement process. It provides a centralized web-based platform to streamline registration, job postings, eligibility verification, applications, interviews, and offer management.

Project Scope:
The system digitizes the end-to-end placement workflow. Functionalities include student registration and profile management, company job postings with eligibility criteria, real-time application workflow, result publishing, and placement analytics.

Environmental Characteristics:
The system will be a web-based application running on institutional servers or cloud. It must handle high concurrent usage during peak placement season and integrate with institutional databases for student verification.

2) Overall Description of Organization of SRS Document

Product Perspective:
CPMS is an enterprise web application with three roles: Student, Placement Officer (Admin), and Company Recruiter.

Product Features:

- Role-based authentication.

- Student profile & resume management.

- Job drive creation and eligibility filtering.
- Application submission and withdrawal.
- Shortlisting, interview, and offer management.
- Automated notifications and dashboards.

User Classes:

- Students (~2000 users per cycle).
- Placement Officers/Admin (~10).
- Recruiters (~50 per cycle).

Operating Environment:

- OS: Windows/Linux server.
- Backend: PostgreSQL/MySQL with Node.js or Java Spring Boot.
- Frontend: React/Angular.
- Browser compatibility: Chrome, Firefox, Edge.
- Mobile responsive.

Design & Environmental Constraints:

- Secure role-based access control.
- Must integrate with institutional LDAP/SMTP.
- GDPR-compliant for data privacy.

User Documentation:
A user manual will be provided for students, admins, and recruiters describing registration, login, application workflows, and report generation.

3) Functional Requirements

The functionalities are categorized by user class:

3.1 Student

- **R.1.1 Register/Login** – Students register via institutional email and authenticate with password.
- **R.1.2 Manage Profile** – Update personal details, resume, and skills.
- **R.1.3 Apply for Drives** – View eligible drives and apply.
- **R.1.4 Withdraw Application** – Withdraw before deadline.
- **R.1.5 Track Application Status** – View real-time updates (Applied, Shortlisted, Rejected).
- **R.1.6 Accept/Reject Offers** – Manage offers and mark final placement status.

3.2 Placement Officer (Admin)

- **R.2.1 Manage Drives** – Create job postings, define eligibility criteria.
- **R.2.2 Validate Eligibility** – System auto-checks criteria; admin can override if required.
- **R.2.3 Publish Results** – Upload shortlists and final results.
- **R.2.4 Generate Reports** – Statistics on placement progress, drive participation, and selection ratios.
- **R.2.5 Broadcast Announcements** – Send notices to students.

3.3 Company Recruiter

- **R.3.1 Create Job Postings** – Post openings with criteria (CGPA, backlog).
- **R.3.2 View Applicants** – See list of eligible candidates.
- **R.3.3 Upload Shortlists** – Submit shortlisted candidates and interview schedules.
- **R.3.4 Upload Offers** – Add final results (selected/rejected).

4) External Interface Requirements

User Interfaces:

- Responsive web-based GUI with dashboards for students, admins, and recruiters.

Hardware Interfaces:

- Standard institutional desktops/laptops, mobile devices.

Software Interfaces:

- Institutional database (for academic data).
- Email/SMTP service for notifications.

Communication Interfaces:

- HTTPS protocol for secure communication.

5) Other Non-Functional Requirements

Performance Requirements:

- Must support 2000+ concurrent logins.
- Response time ≤ 3 seconds for major operations.

Security Requirements:

- All data transfer over HTTPS.
- Passwords stored with SHA-256 + salt.
- Audit logs maintained for 1 year.

Usability Requirements:

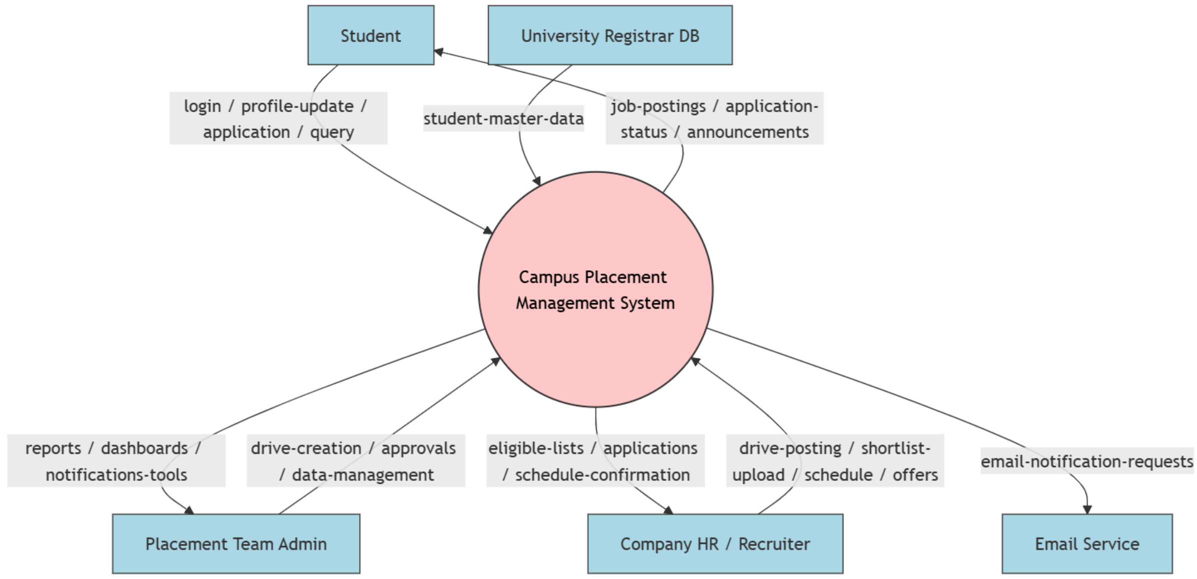
- Mobile-friendly, intuitive UI.
- Clear and actionable error messages.

Maintainability & Portability:

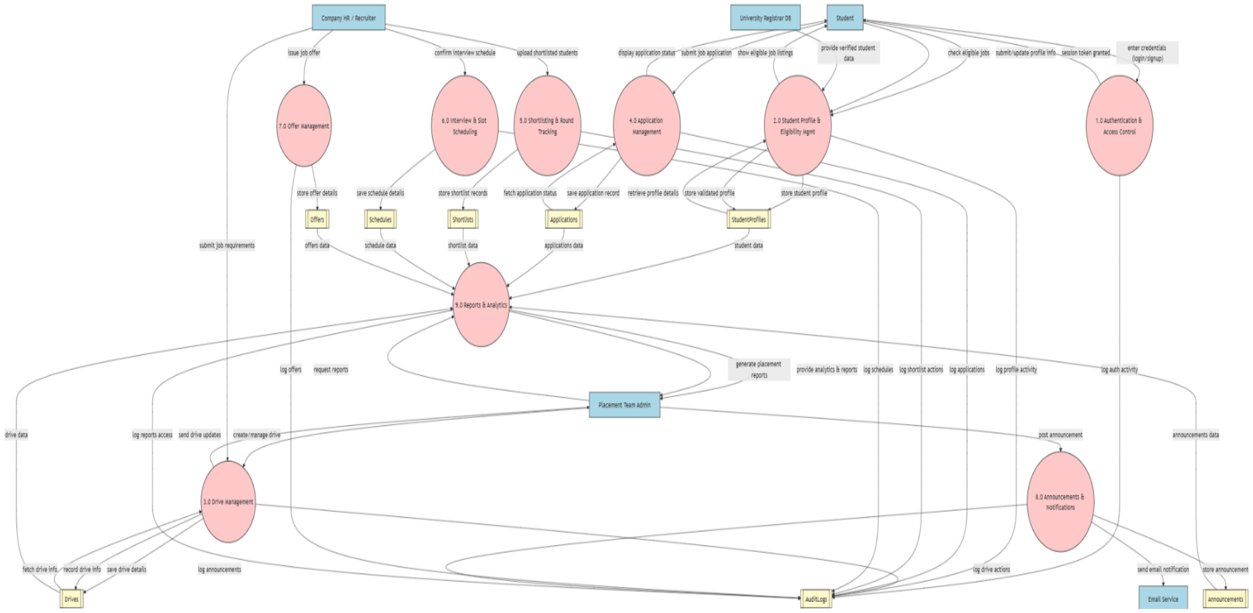
- Modular architecture for independent upgrades.
- Deployable on both on-premise and cloud.

Data Flow Diagram:

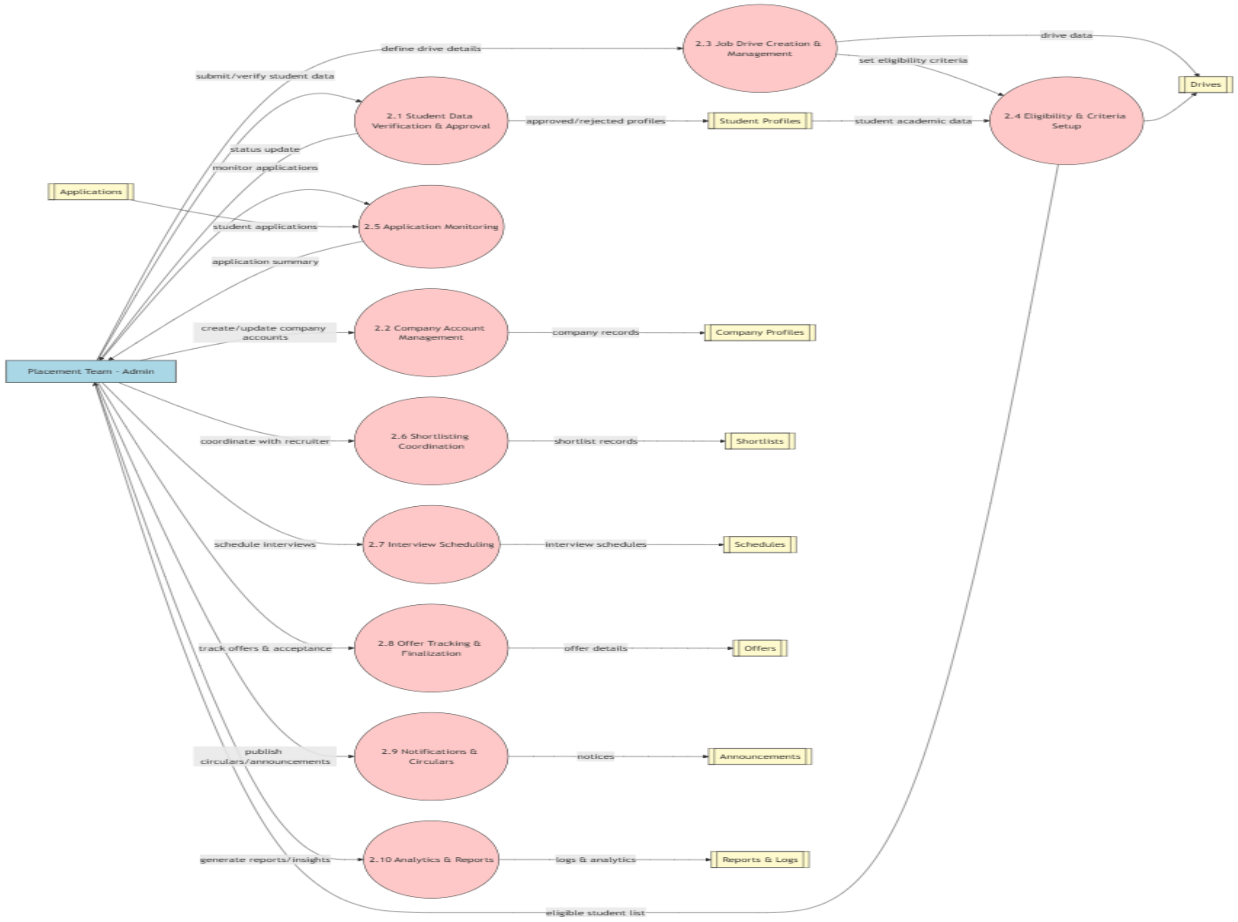
Level 0 DFD


Student
University Registrar DB
login / profile-update / application / query
student-master-data
job-postings / application-status / announcements
Campus Placement Management System
reports / dashboards / notifications-tools
drive-creation / approvals / data-management
eligible-lists / applications / schedule-confirmation
drive-posting / shortlist-upload / schedule / offers
email-notification-requests
Placement Team Admin
Company HR / Recruiter
Email Service

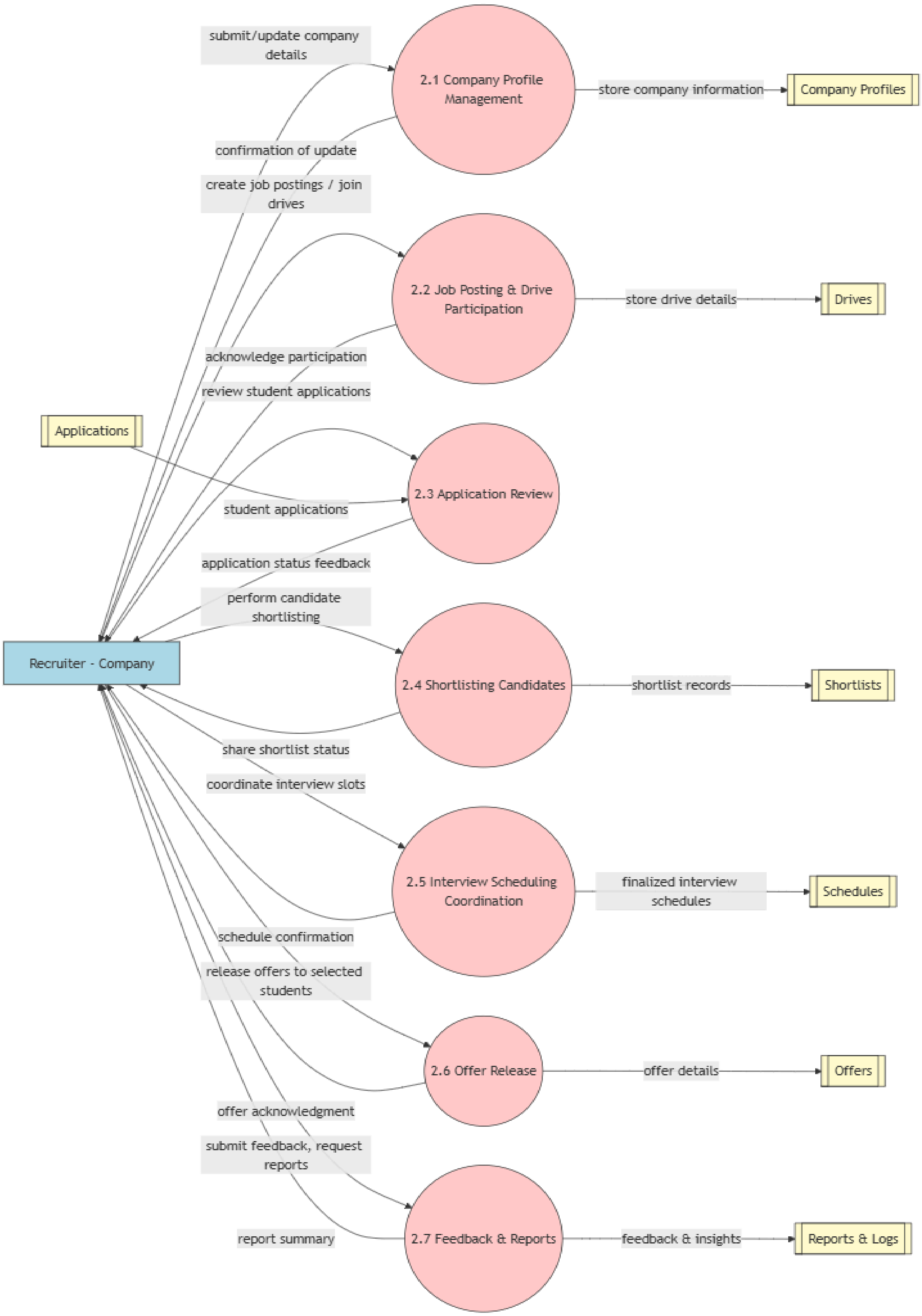
Level 1 DFD



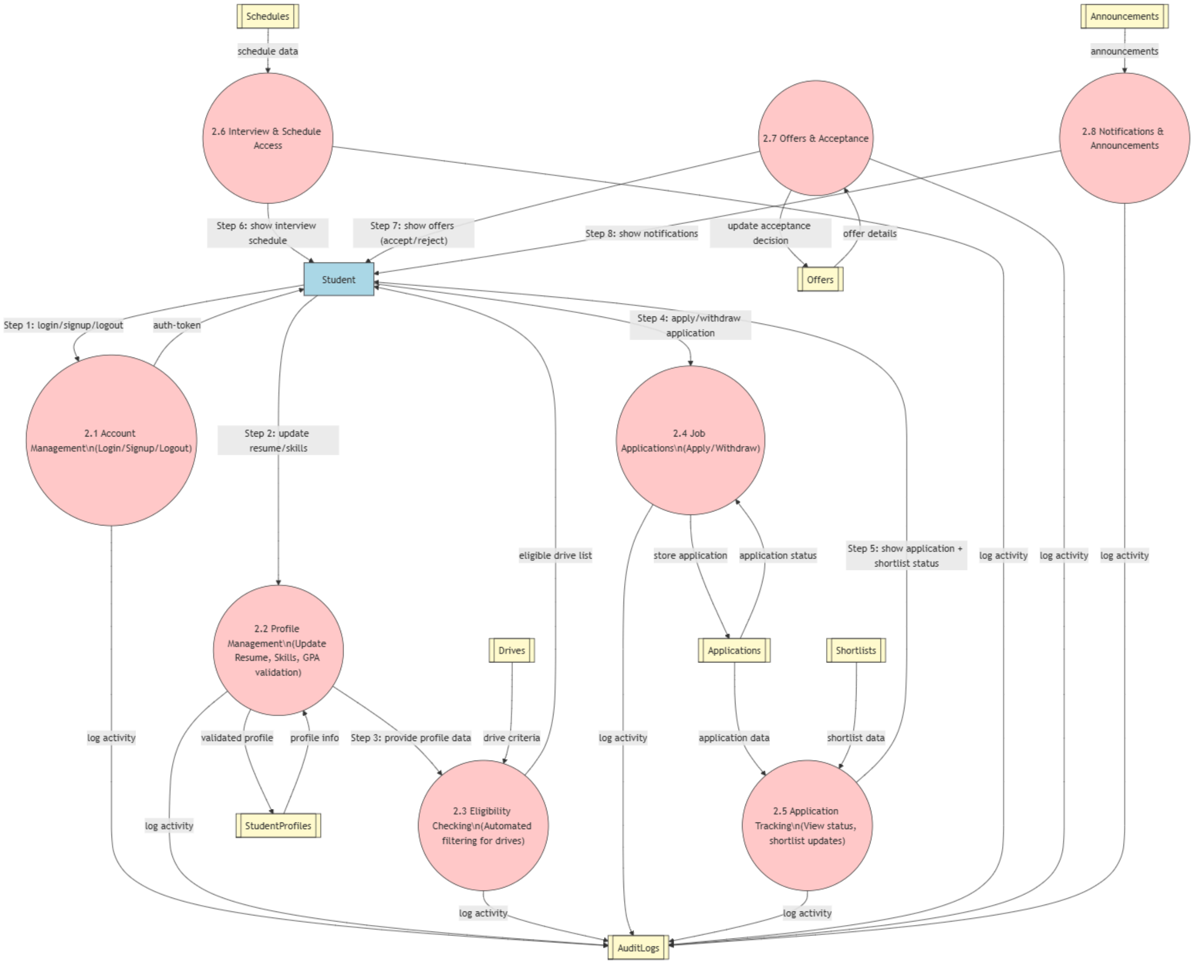
Level 2 DFD/Admin



Level 2 DFD/Recruiter


submit/update company details
2.1 Company Profile Management
store company information
Company Profiles
confirmation of update
create job postings / join drives
2.2 Job Posting & Drive Participation
store drive details
Drives
acknowledge participation
review student applications
Applications
2.3 Application Review
student applications
application status feedback
perform candidate shortlisting
Recruiter - Company
2.4 Shortlisting Candidates
shortlist records
Shortlists
share shortlist status
coordinate interview slots
2.5 Interview Scheduling Coordination
finalized interview schedules
Schedules
schedule confirmation
release offers to selected students
2.6 Offer Release
offer details
Offers
offer acknowledgment
submit feedback, request reports
report summary
2.7 Feedback & Reports
feedback & insights
Reports & Logs

Level 2 DFD/Student


Schedules
schedule data
Announcements
announcements
2.6 Interview & Schedule Access
2.7 Offers & Acceptance
2.8 Notifications & Announcements
Step 6: show interview schedule
Step 7: show offers (accept/reject)
Step 8: show notifications
update acceptance decision
offer details
Offers
Student
Step 1: login/signup/logout
auth-token
Step 4: apply/withdraw application
2.1 Account Management\n(Login/Signup/Logout)
Step 2: update resume/skills
2.4 Job Applications\n(Apply/Withdraw)
eligible drive list
store application
application status
Step 5: show application + shortlist status
log activity
log activity
log activity
2.2 Profile Management\n(Update Resume, Skills, GPA validation)
Drives
Applications
Shortlists
log activity
validated profile
profile info
Step 3: provide profile data
drive criteria
log activity
application data
shortlist data
log activity
StudentProfiles
2.3 Eligibility Checking\n(Automated filtering for drives)
2.5 Application Tracking\n(View status, shortlist updates)
log activity
log activity
AuditLogs

Data Dictionary:

Data Flows

- login/signup/logout → credentials {roll-no/email, password} + session-token.
- auth-token → secure token for session validation.
- profile-update → student-id + resume + skills + GPA + certifications + contact-info.
- validated-profile → student-id + verified-academic-record + resume-status.
- student-master-data → student-id + name + roll-no + course + GPA + backlogs + academic-history.
- job-postings / drive-info → drive-id + company-id + role + eligibility-criteria + last-date + schedule.
- eligible-listings → student-id + drive-ids (matching criteria).
- apply-job / withdraw → student-id + drive-id + application-status-change.

- application-status → application-id + status {submitted / shortlisted / rejected / in-process}.
- drive-creation → drive-id + company-id + role + criteria + schedule-info.
- drive-updates → drive-id + company-id + status + timeline.
- shortlist-upload → {application-id, student-id}* + shortlist-status.
- shortlist-record → shortlist-id + drive-id + student-ids.
- schedule-confirmation → schedule-id + date + slot + student-list.
- schedule-info → schedule-id + student-id + interview-slot.
- offer-release → offer-id + drive-id + student-id + role + CTC + joining-date.
- offer-details → offer-id + student-response {accepted/rejected/pending}.
- announcement / notification → announcement-id + content + recipients.
- email-notification-request → message-id + recipient-list + subject + body.
- reports / analytics → aggregated placement data (drives, applications, shortlists, offers).
- feedback-report → recruiter-id + student-performance + hiring-feedback.
- status-update (Admin ↔ Student/Recruiter) → profile/drive/application/offer decision.

Data Stores

- StudentProfiles (D1) → student-id + name + contact + academic-record + resume + skills + certifications + eligibility-status.
- CompanyProfiles (D2) → company-id + company-name + recruiter-info + login-credentials + job-history.
- Drives (D3) → drive-id + company-id + role + eligibility-criteria + schedule + status.
- Applications (D4) → application-id + student-id + drive-id + submission-date + status.
- Shortlists (D5) → shortlist-id + drive-id + student-id + shortlist-status.
- Schedules (D6) → schedule-id + drive-id + student-list + interview-panel + slots.
- Offers (D7) → offer-id + student-id + drive-id + role + package + joining-date + acceptance-status.
- Announcements (D8) → announcement-id + content + type (drive / general / placement-update) + recipients + date.
- Reports & Analytics (D9) → report-id + generation-date + filters + placement-metrics + visualizations.
- AuditLogs (D10) → event-id + user-id + action + timestamp + IP + module + result {success/failure}.