Milestone 3

Team 22

Software Engineering Project: GenATS

Name	Roll Number
Suriya Prakaash J L	21f1001124
Thumula Praneeth Rao	23f2005748
Srujan Umesh Patwardhan	22f3001954
Jeevika S	21f3001259
Jyoti Sharma	22f3001795
Saksham Sirohi	22f1001635
Devendra Kumar	21f2000575
Ashish Al Rashid	22f1001551

Task Distribution

Roles and Responsibilities:

Name	Responsibility
Suriya Prakaash J L	User stories, User interviews, Wireframing, Storyboard, Frontend development
Thumula Praneeth Rao	User stories, User interviews, Wireframing, Frontend development
Srujan Umesh Patwardhan	User stories, Documentation, Project planning and assigning tasks, Backend development
Jeevika S	User stories, User interviews, Backend diagrams, Backend development
Jyoti Sharma	User stories, Wireframing, Testing, Scheduling, AI development
Saksham Sirohi	Project Management, User stories, Designing high and low fidelity prototypes, Documentation, Scheduling
Devendra Kumar	Project Management, User stories, Storyboard, Wireframe Feedback, AI development
Ashish Al Rashid	User stories, Designing high and low fidelity prototypes, Frontend development, Backend (interface with frontend)

The above roles and responsibilities were assigned taking into account each team member's availability, capability, and unique strengths to ensure a balanced workload.

According to the project milestones, individual roles and responsibilities and the timeline and complexity of the project, we broke down the project into 6 sprints of 1 week each.

Project Schedule

Sprint Schedule:
- **Sprint 1 (15th Oct - 21st Oct)**
Objective: Identify primary, secondary and tertiary users of the application, conduct interviews and formulate user stories

  We successfully identified the primary, secondary and tertiary users of our HR Management application and formulated user stories based on user interviews. (Refer [milestone 1](#))

- **Sprint 2 (22nd Oct - 28th Oct)**
Objective: Develop low-fidelity prototypes of the system, a storyboard based on user requirements, and feedback on wireframes

  We have also achieved this objective by using Figma to design low fidelity wireframes for our application pages, developing a storyboard based on requirements and getting feedback from experts on our wireframes. (Refer [milestone 2](#))

- **Sprint 3 (29 Oct - 4 Nov)**
Objective: Plan the entire project, divide it into sprints, assign relevant user stories to sprints, assign tasks to relevant team members, design high fidelity prototypes, develop class and ER diagrams for backend development, develop most of the pages in the frontend.

  We have split the entire project into 6 sprints, and are going to use JIRA for project planning and task management. We have designed high fidelity prototypes of the system, developed class and ER diagrams for the system <<insert link>> or <<diagrams>>, and developed most of the pages in the frontend (elaborated below).

- **Sprint 4 (4 Nov - 11 Nov)**
Objective: Developing the API endpoints for the system, documenting the APIs in YAML format.

- **Sprint 5 (11 Nov - 18 Nov)**
Objective: Test cases and extensive testing and documentation of the application

- **Sprint 6 (18 Nov - 25 Nov)**
Objective: Finishing remaining work and final submission

Gantt Chart:

Detailed Project Gantt Chart (7-Day Interval Timeline)
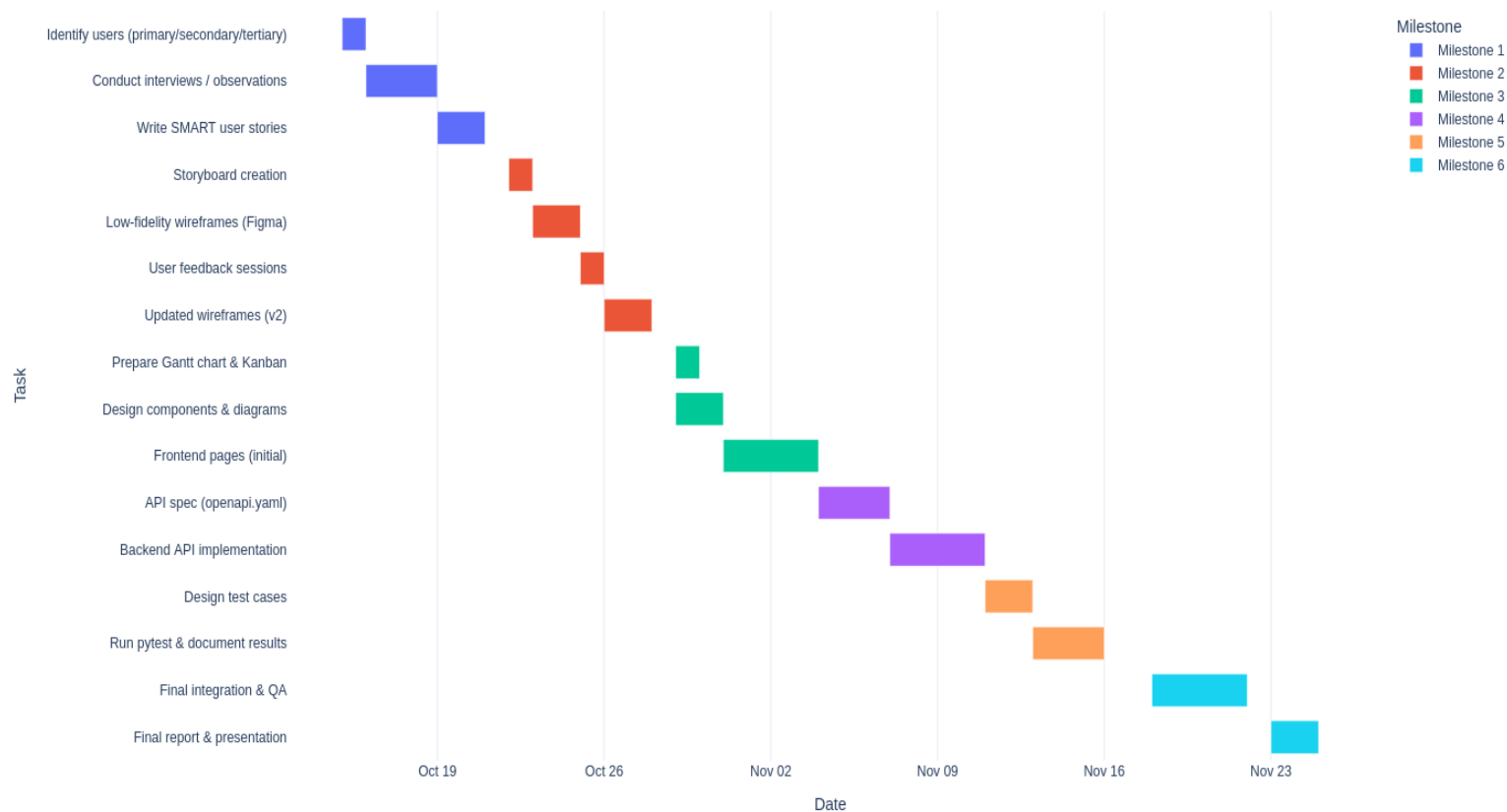
The above Gantt chart has been prepared keeping the project milestones in mind. We majorly tackle identifying users, requirements and formulating appropriate user stories in the first half of October month, and preparing storyboards, wireframes, and low fidelity prototypes in the second half of October.

From the last week of October to the first week of November, we majorly handle the frontend development part, along with proper scheduling and assigning of tasks. The backend development starts simultaneously from the first week of November till mid-November.

Testing and appropriate documentation will be handled from mid-November till November end, while preparing the final report and presentation and finishing all remaining tasks will also be wrapped up by November end.

Scrum Meetings

Below is a summary of the scrum meetings conducted for the project, formatted for clarity and alignment with standard software engineering documentation practices:

Scrum Meeting Schedule: We decided to have a meeting every Monday and Thursday

Meeting 1: 27/10/2025

Proposed to build the frontend using Vue 3. Requirement for responsive UI, reusable components, and role-based access. Agreed on role-based access for HR, Applicant, and Organization modules. Discussed the entire codebase structure and assigned roles for each person. Jira was used for creating sprints for the same and assigning work.

Meeting 2: 30/10/2025

Discussed about the current progress on the initial website design and potential improvements. Finalized DashboardLayout, NavBar, and Sidebar as shared components across roles. Confirmed completion of sprint-wise tasks. HR module prioritized first: HR Dashboard, Candidate Management, Job Creation. Organization module planned next: Org Profile, HR Management, Settings. Applicant module design to follow a cleaner, minimal theme.

Meeting 3: 3/11/2025

Assessed the current status of the project, HR Components were completed 90% with some improvements being needed. The organization part was 100% completed. Theme consistency corrections made; removed mismatched colors. The team decided that the applicant side has to be done on priority.

Project Scheduling Tools

Our team decided to use [JIRA](#), a popular project management software by Atlassian. This was due to the simple interface that JIRA provides, which will help us to effectively plan our project, assign tasks and monitor progress.

We divided our project into 6 sprints, which were aligned with the milestones as well as aligned with the appropriate user stories that we had formulated in Milestone 1.

JIRA Backlog view:

The backlog view gives an overview of items not currently in any sprint (backlog) and items assigned to particular sprints.

JIRA Board View:

The board view (Kanban style) lists all the user stories in the current sprint and helps everyone in the team manage and track the stories' status (To do, In progress, In review, Done).
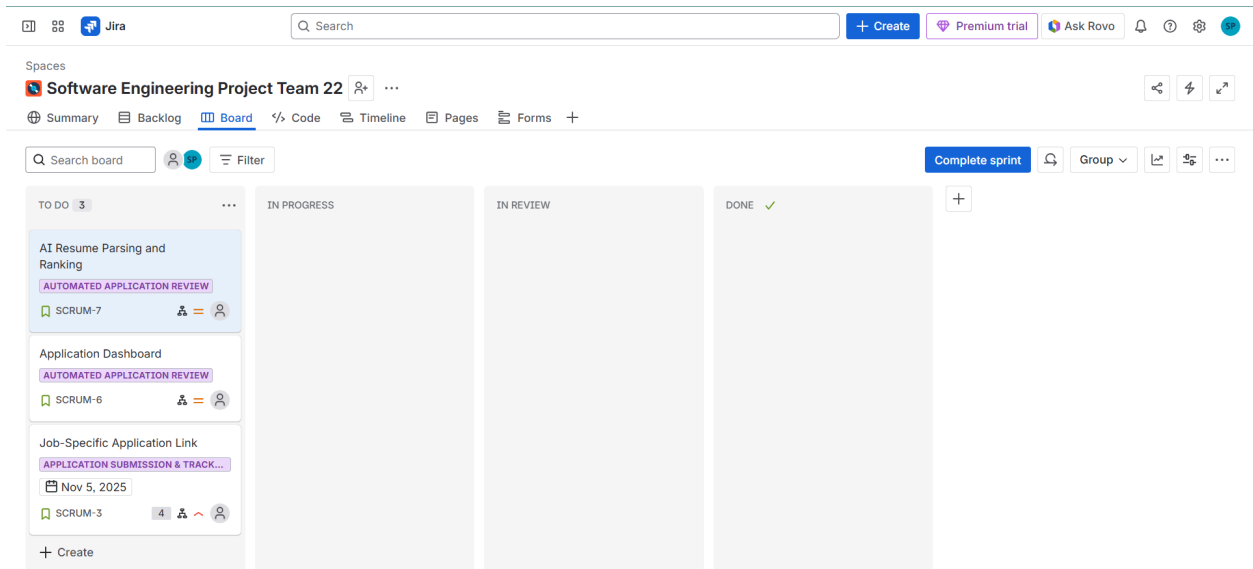
JIRA Timeline View:

An epic is a collection of user stories. The JIRA timeline view allows you to visualize the epics, stories and tasks for your project in a timeline view, based on your sprints.
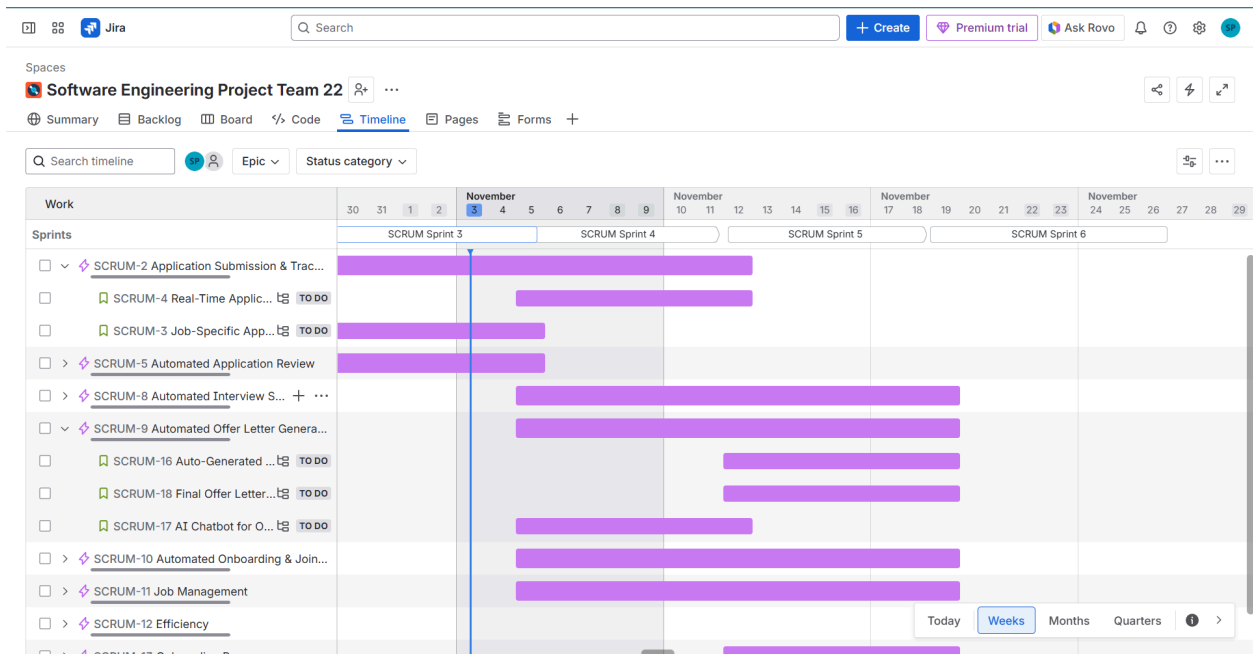
Tasks for a user story:

A user story is further divided into tasks, which can be assigned to different members of the team based on their roles.

Search   + Create   Premium trial   Ask Rovo

Epic - Change epic

Spaces / Software Engineering... / SCRUM-9 / SCRUM-17

AI Chatbot for Offer Letter Refinement

Description

As a recruiter,
I want AI-generated offer and regret letters using editable templates and contextual personalization,
so that candidate communication becomes error-free and automated within 24 hours of final decision, improving candidate satisfaction scores.

Subtasks

0% Done

Work	Priority	Assi...	Status
SCRUM-61 Frontend: Editable offer letter and templates	= M...	TR T..	TO DO ∨
SCRUM-62 Frontend: Page where list of previously used templates is listed	= M...	SL S..	TO DO ∨
SCRUM-63 Backend: Get relevant user details	= M...	SP S..	TO DO ∨
SCRUM-64 AI: Autofill information according to context and prompt	= M...	DK D..	TO DO ∨

Linked work items

Add linked work item

Confluence content ⓘ

To Do ∨   ⚡   ⁑ Improve Story

Details

Assignee	⊝ Unassigned
	Assign to me
Priority	= Medium
Parent	⬦ SCRUM-9 Automated Offer Letter Gen
Due date	None
Labels	None
Team	None
Start date	None
Sprint	SCRUM Sprint 4
Story point estimate	None
Development	∨
Reporter	SP Srujan Patwardhan

Automation ⚡ Rule executions

Frontend Designs

Sign Up

Welcome back!!

Please Sign In

Candidate   HR   Organization

Email address

candidate@example.com

Password

••••••••••   👁

☐ Remember me   I forgot my password

Sign In

🚀 Quick Login (Frontend Dev)
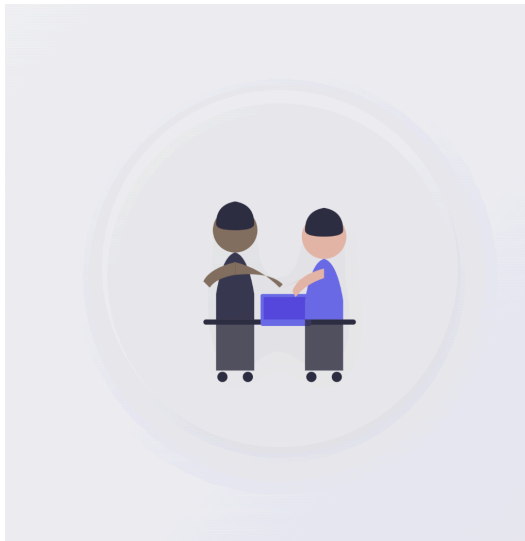
Candidate   HR   Admin

Sign-In Page - Login screen with tabs for Candidate/HR/Organization roles, offering email/password authentication and quick login options for development.
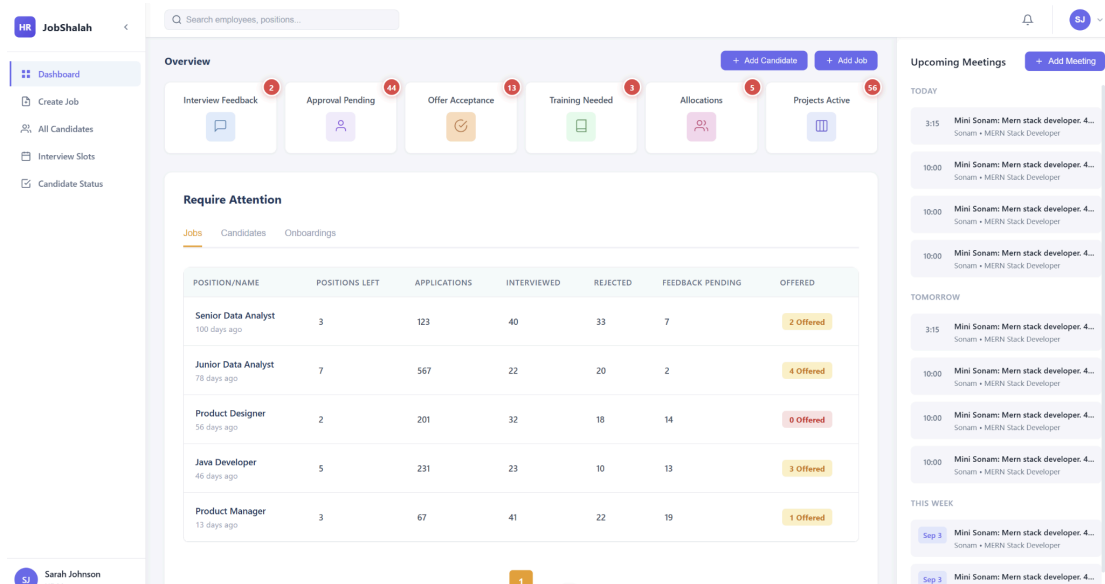
HR Overview Dashboard - High-level metrics showing interview feedback, approval pending, offer acceptance, training needs, active projects, allocations, and upcoming meetings schedule.

Job Opening Creation Form - Interface for HR to create new job postings with fields for title, level, salary, experience, skills, and job description upload options.

All Candidates List - Dashboard displaying 10 candidates with their names, roles, gender, and AI match scores, with filtering and pagination options.

Candidate Profile Detail - Comprehensive view of candidate Vishnu Divakaran showing contact info, status, interview schedule, uploaded documents, experience, and 75% potential fit score.
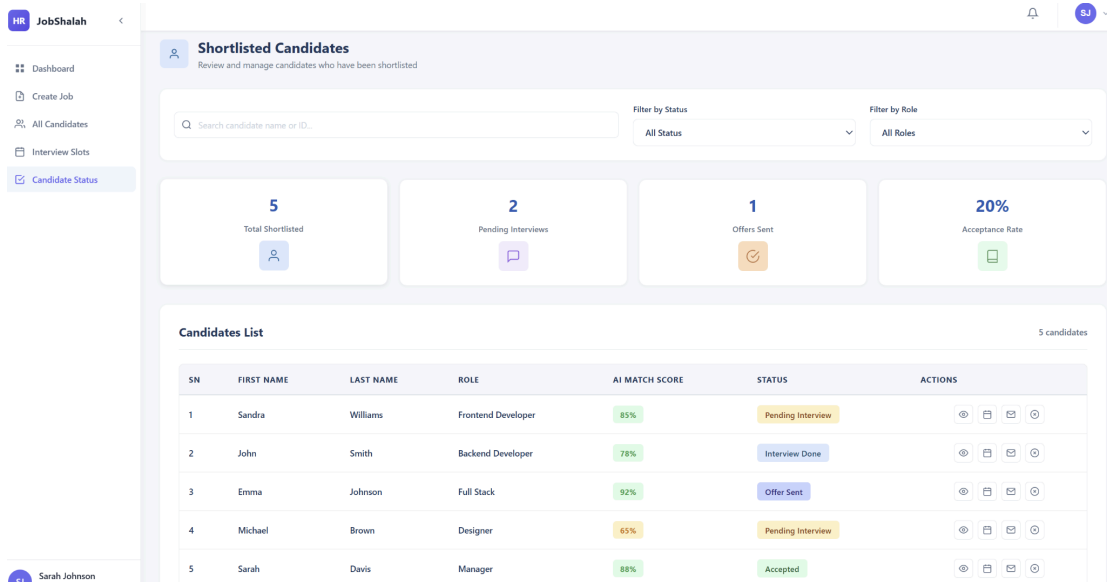
Shortlisted Candidates Dashboard - Status overview showing 5 shortlisted candidates with metrics including 2 pending interviews, 1 offer sent, and 20% acceptance rate
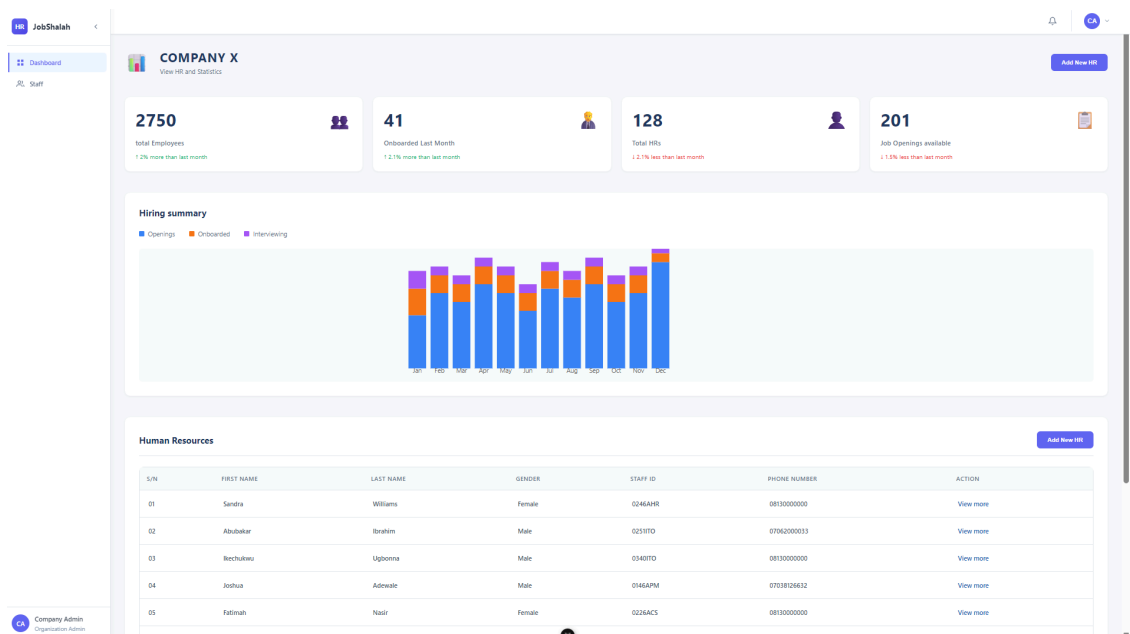
Organization Dashboard - High-level metrics showing hiring stats, all HR in the department, and all staff.

Staff Page For Organisation - List of all staff and a CRUD-enabled table from where the organisation can review each staff HR member's information.

Candidate Dashboard - High-level metrics showing recent applications, Application Status, Upcoming Interviews, and Quick Links.

Candidate's Profile - This page is dedicated to setting up the Candidate's own profile and storing information about themselves that they want to share with organizations.
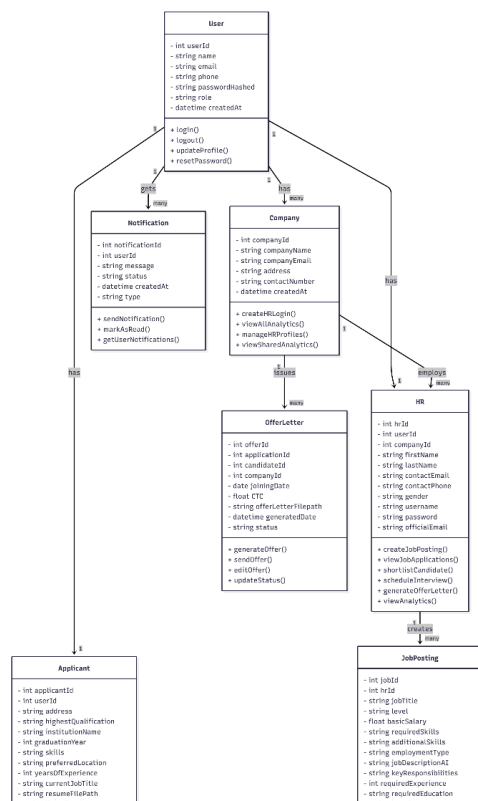

All Opportunities - Here, Candidates can get notified about all new Job Postings.
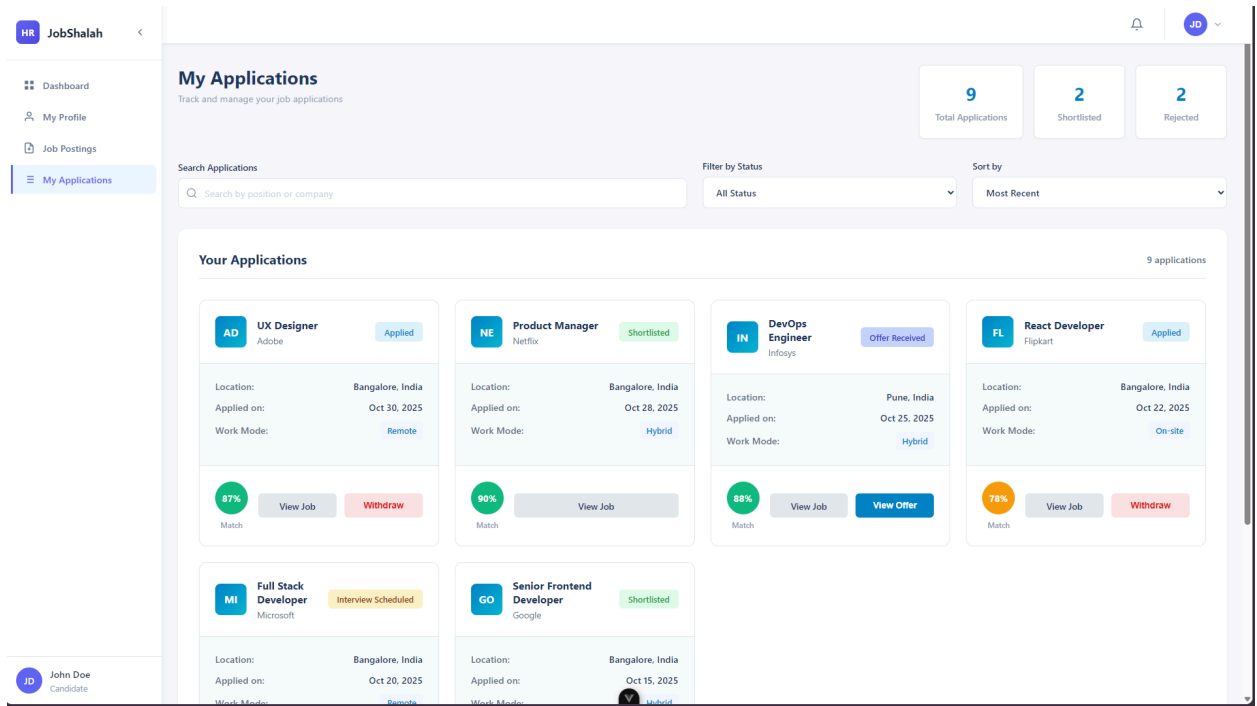
My Application - Here Candidates can review their all applications that they have applied for and their status.

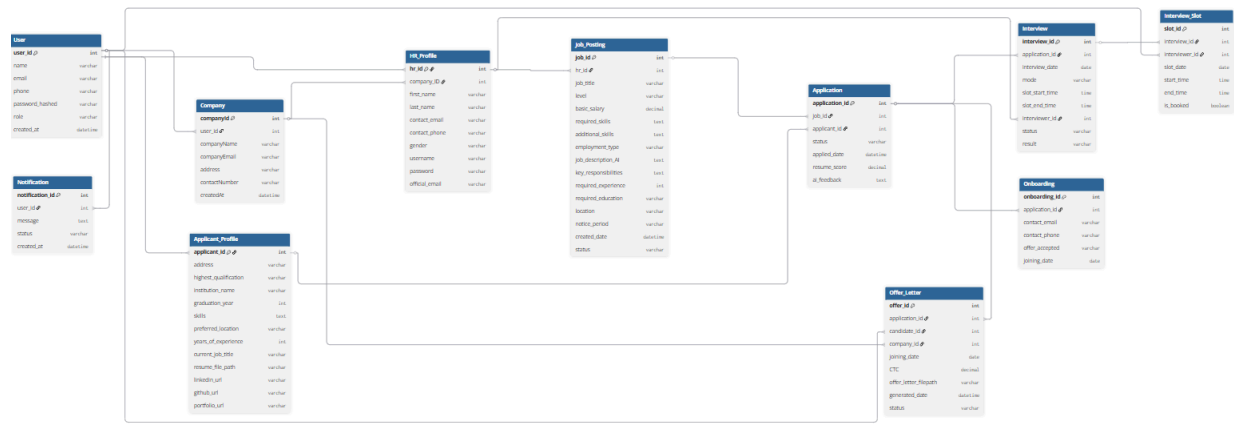# Class and ER diagrams

Class Diagram:

The class diagram shows the different components and classes present in the system along with the relationships between them.

The methods and attributes of each class along with their scopes have also been depicted.

For a better view, visit this link.

ER Diagram:

The system centralizes user information (`User` and `Notification`) and extends it via role-specific profiles for applicants (`Applicant_Profile`) and recruiters (`HR_Profile`). Recruiters associated with a `Company` can post jobs through the `Job_Posting` entity, which applicants respond to via `Application` records. Each application can progress through `Interview` and `Interview_Slot` scheduling, leading to an `Offer_Letter` upon successful evaluation, and ultimately `Onboarding` for finalized hires. The schema enforces data integrity through foreign keys that mirror real-world relationships, ensuring a seamless and traceable workflow across all recruitment stages.

For a better view, visit this [link](#).