Software Requirements Specification

AI-Powered Resume Screening System

Date: March 13, 2025

Table of Contents

1. Introduction

1.1 Purpose

This document outlines the comprehensive requirements for an AI-Powered Resume Screening System designed to streamline the hiring process through automated resume parsing, candidate screening, and data-driven decision-making support.

1.2 Scope

The system delivers the following core functionalities:.

- Resume collection through multiple channels (manual upload and email integration crawling).
- AI-driven resume parsing.
- candidate screening according to job requirement.
- Automated notifications for recruiters.
- Integration with Hiring Bull for pushing Selected Candidates.

2. System Users and Stakeholders

2.1 User Roles

2.1.1 HR Administrators

- Complete system access with administrative privileges.
- Access to reporting and analytics dashboards.

2.1.2 Recruiters

- Ability to upload individual or multiple resumes.
- Access to candidate match scores and insights.
- Authority to approve or reject candidates.

2.2 Key Stakeholders

- Milin Desai
- Manish Anjara
- Archie Bhatt

3. Functional Requirements

3.1 Resume Upload & Parsing

3.1.1 Upload Capabilities

- Support for both single and multiple resume uploads in multiple formats (PDF, WORD FILES, Images).
- Automated resume collection through email inbox crawling (The process of scanning and retrieving resume from an email inbox).

3.1.2 Parsing Resume

- Extraction of critical candidate information including:
    - Personal details (name, contact information)
    - Profile Summary
    - Educational background
    - Professional experience
    - Technical and soft skills

3.2 AI-Driven Screening & Decision-Making

3.2.1 Job Requirement Analysis

- Retrieve the Role Clarity Document and Job Description (JD).
- Map job requirements against candidate profiles to assess suitability.
- Provide job fitment scores based on alignment with job criteria.

3.2.2 AI Screening Capabilities

- Matching between candidate profiles and job requirements.
- Ranking of candidates with mentioning of skill gaps.
- candidate insights dashboard.

3.2.3 AI-Based Decision Support

- The system will calculate a match percentage between candidate profiles and job descriptions, providing solid,
- Explanation behind the match to support recruitment decisions.

3.3 Role-Based Access Control

- Granular user role management ensures appropriate data access.

- Department-specific views for targeted recruiting efforts.

4. Non-Functional Requirements

4.1 Security

- Full compliance with relevant data protection regulations.
- Robust authentication protocols to ensure secure access.

4.2 Usability

- An intuitive and responsive design optimized for desktop use.

5. Technical Architecture

5.1 System Components

- **Frontend:** React.js
- **Backend:** Node.js (Express) and FastAPI
- **Database:** PostgreSQL
- **AI/NLP:** spaCy and NER for advanced parsing and screening
- **Communication:** SendGrid Email integration
- **File Storage:** Cloud-based secure document management
- **API Gateway:** Promact Career site and hiring bull integration.

5.2 Core Database Entities

- **Parsed Resumes**: Stores parsed resume data, including structured candidate information and processing status.
- **Roles**: Defines roles within the system with associated permissions and descriptions.
- **Users**: Contains user details like name, email, password, and role assignments.
- **Applications**: Tracks candidate applications, including match scores, status, and associated documents.
- **Candidates**: Stores candidate personal details, skills, experience, education, and parsed resume information.
- **Jobs**: Contains job descriptions, requirements, and the associated recruiter or HR user.
- **Screening Results**: Stores the results of candidate screenings, including match scores and missing skills.
- **Departments**: Defines organizational departments with names, descriptions, and related job listings.

- **Feedback**: Tracks feedback messages from departments regarding candidate applications.
- **Education**: Stores educational details of candidates, including degree titles and statuses.
- **Skills**: Contains structured skill data extracted from candidate resumes.
- **Email Configuration**: Stores email configuration details, including SMTP settings and credentials.
- **Permissions**: Defines system permissions for access control and functionality.
- **Roles-Permission**: Manages associations between roles and their respective permissions.

6. AI Model Specification

6.1 Resume Parsing Model

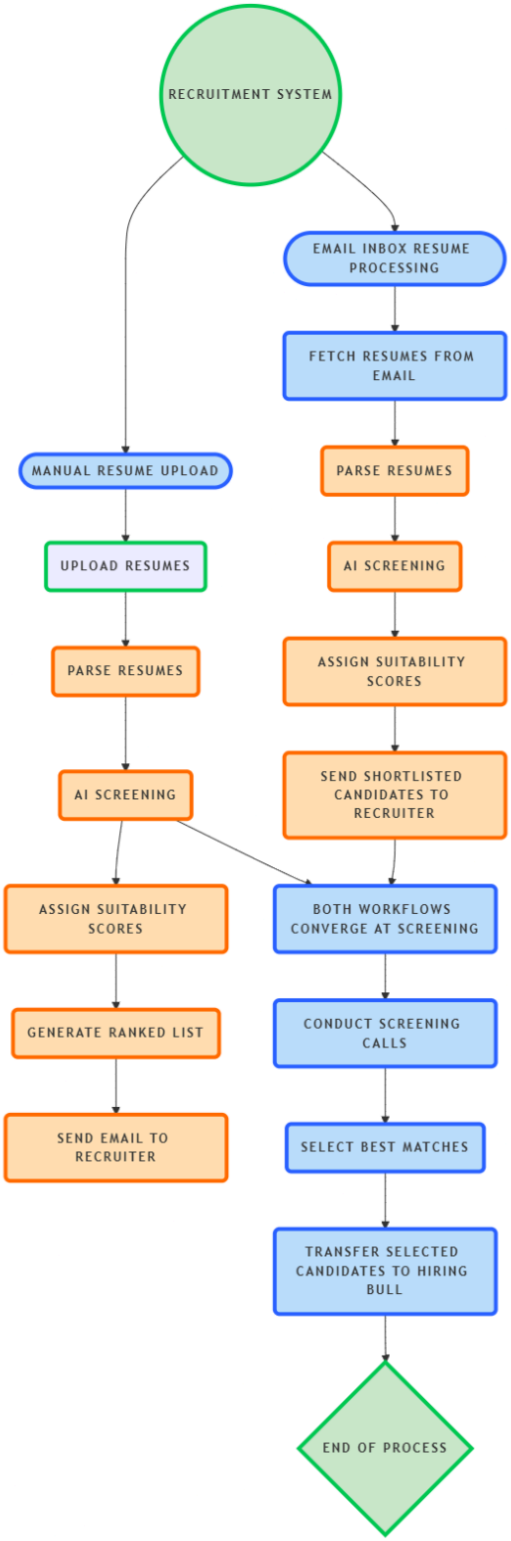
- NLP and Named Entity Recognition (NER) algorithms.

6.2 AI Screening Model

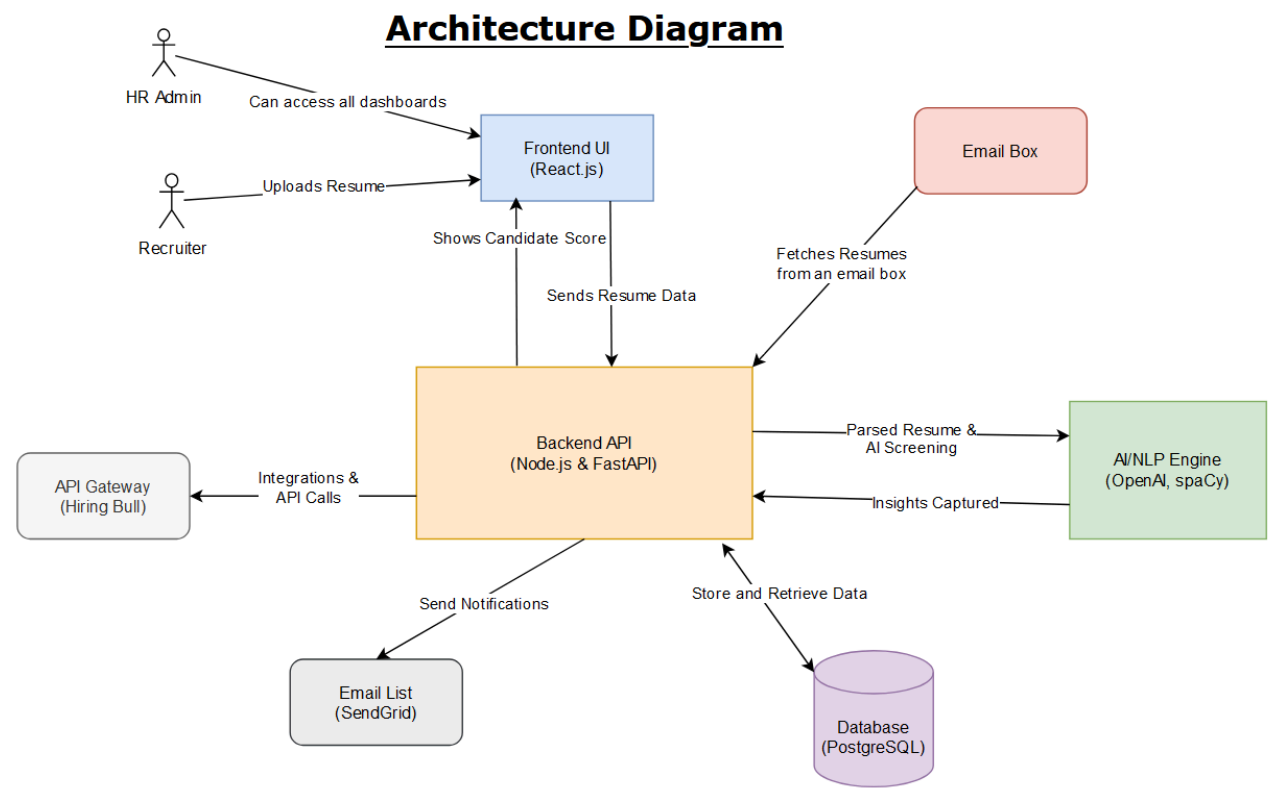
- Sophisticated NLP methods for candidate evaluation.

7. System Workflow

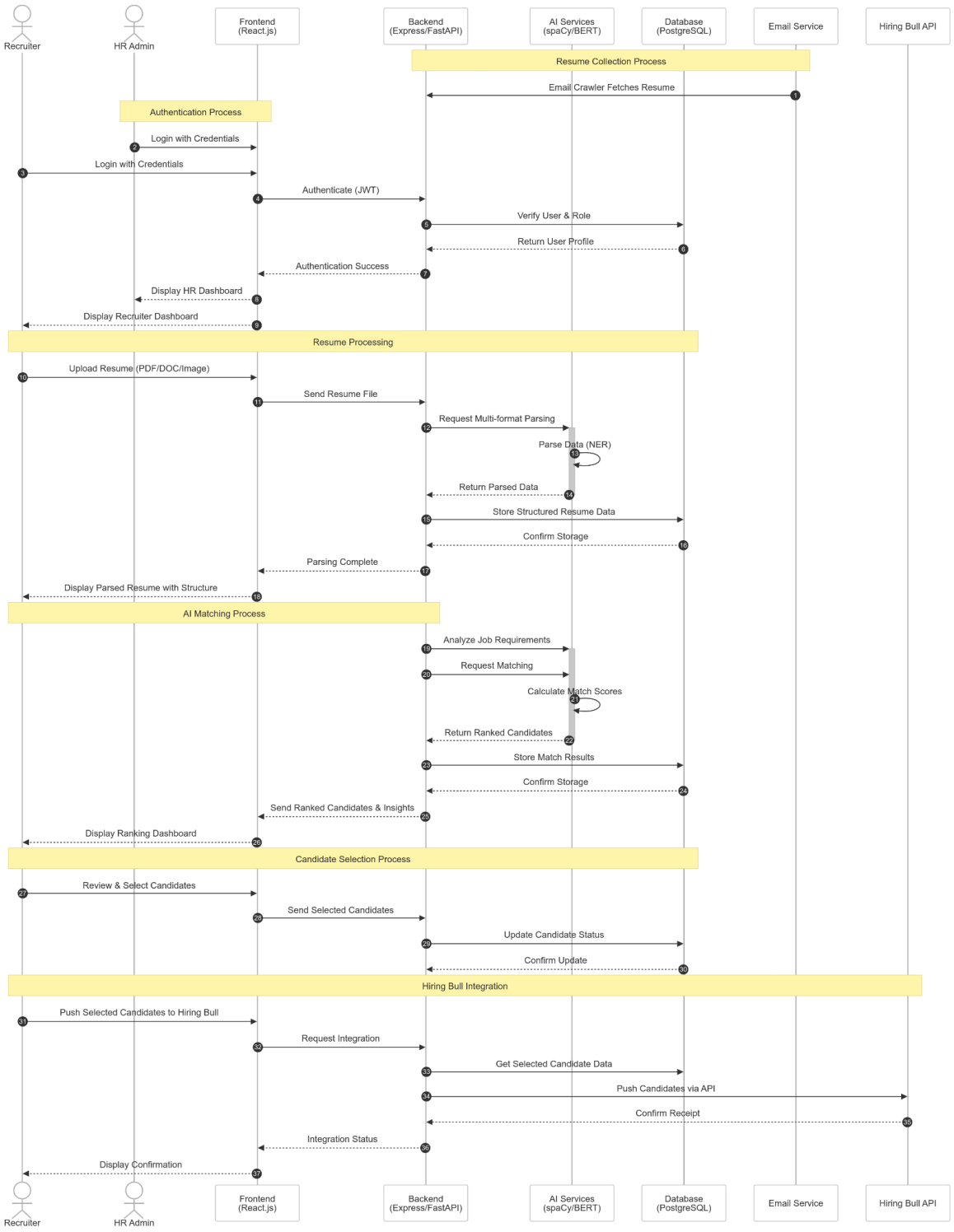
7.1 Flow Chart



7.2 Architecture Diagram

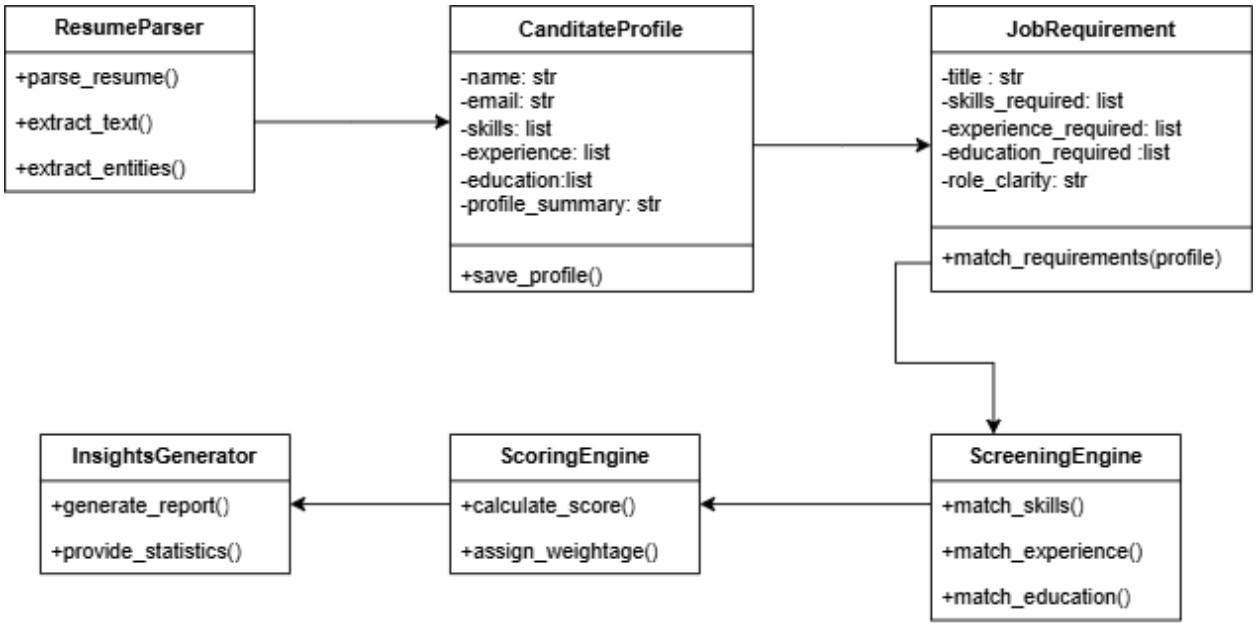


7.3 Sequence Diagram :

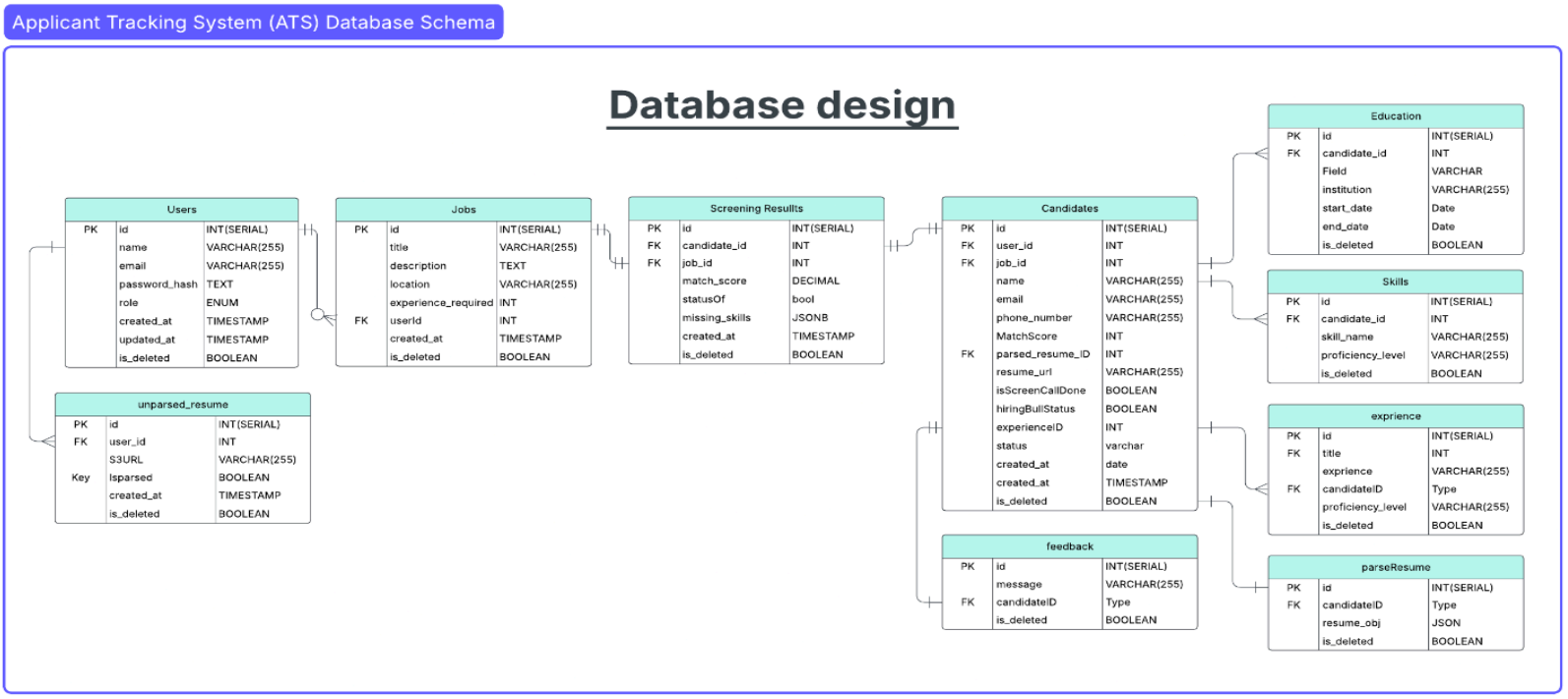


[Link of Diagram]

7.4 Class Diagram



7.5 ER Diagram :



[Link of ERD]

8. Process Workflow

8.1 Resume Acquisition

- Supports resume collection via upload or email.
- Extracts structured data automatically.
- Creates candidate profiles automatically.

8.2 Screening Process

- Analyzes job descriptions.
- Matches and ranks candidates using AI.
- Provides AI-driven initial screening with recommendations.

8.3 Recruitment Pipeline

- HR reviews AI-recommended candidates.
- Collects feedback systematically for improvement.

8.4 Notifications

- Sends automated emails to recruiters.

8.5 Candidate Selection

- Forwards selected candidates to Hiring Bull.
- Stores rejected candidates for future opportunities.

9. User Stories

9.1 HR Administrator

- "As an HR Administrator, I want to see system statistics to assess recruitment performance."
- "As an HR Administrator, I need to filter candidates by recruiter to track team performance."
- "As an HR Administrator, I want to manage user roles and permissions to ensure proper access control."
- "As an HR Administrator, I want to view candidate feedback.

9.2 Recruiter

- "As a Recruiter, I want to upload multiple resumes at once to speed up candidate processing."
- "As a Recruiter, I want to mark candidates whose screening is complete to have a clear view of their status."
- "As a Recruiter, I need to set screening criteria to find the best candidates."
- "As a Recruiter, I want to generate recruitment reports to support hiring decisions."
- "As a Recruiter, I want to track candidate status in the hiring process to stay updated."
- "As a Recruiter, I need to receive automated alerts when a candidate progresses to the next stage."
- "As a Recruiter, I want to leave comments and feedback on candidates for better collaboration."
- "As a Recruiter, I need to store rejected candidates in a talent pool for future opportunities."

10. Implementation Phases

10.1 MVP (Minimum Viable Product) – Core Functionalities

- Back-End:
    - User Management and role-based access control for HR Administrator and recruiters
    - Storing of Basic Details of Resume in PgSQL with candidate details.
    - Storing Role Clarity Document with Job Details
- Front-End:
    - Login Page.
    - Admin Dashboard.
    - Recruiter Dashboard.
    - View Candidate Details.
- Resume Upload & Parsing:
    - Fetch resume in multiple formats (PDF, Word Doc, Image files).
    - Basic data parsing from resumes personal details, work experience, education, skills, and summary.
    - Ensure the system can handle different resume formats (PDF, Word, Images) and reliably parse data from them.
    - Parsed data should be sent to the backend for storage in the database.

11. Success Criteria

11.1 Business Metrics

- Reduce time-to-hire through intelligent automation
- Increase recruiter productivity with streamlined workflows
- Enhance candidate experience through responsive communication and structured processes

11.2 Technical Metrics

- Achieve decent resume parsing accuracy
- Ensure seamless integration with existing hiring bull system

12. Definitions and Acronyms

- **SRS:** Software Requirements Specification

- **NLP:** Natural Language Processing
- **ML:** Machine Learning
- **UI:** User Interface
- **API:** Application Programming Interface

13. Project Team

Leadership

- **Stakeholders:** Archie Bhatt, Milin Desai, Manish Anjara
- **Project Manager:** Firoja Parveen
- **Project Lead:** Asha Patel
- **DevOps Lead:** Vivek Kadam/Saquib Mansuri

Development Team

- **Software Engineers:**
    - Nainesh Vidja
    - Nidhi Dhinoja
    - Niket Patadiya
    - Prashant Sahu
    - Sudipto Das
    - Vinay Danidhariya
- **AI/ML Developers:**
    - Sanjana Shaw
    - Rohit Ezhuthachan
- **DevOps Engineer:**
    - Rishi Shori