

AI-Powered Resume Screening System

Sprint Goals for This Week

Date:- 19/03/2025

AI Development

- Upload a resume file (PDF, Word, or Image) and extract text from it.
- Parse the extracted resume text into key entities such as name, email, phone number, skills, experience, and education using NER techniques (like spaCy , Gemini NER), Pyreparser, etc.
- Test with sample resumes to validate parsing accuracy and handling of different formats.

Front-end Development

- Login and sign-in functionality page.
- User-specific pages.

Back-End Development

- Set up User Authentication for secure access.
- Integrate cloud to store Job Descriptions (JD), Role Clarity Documents (RC), and Resumes, saving their URLs in the database.
- Develop an API to upload resumes to the system.
- Show data from the resume uploaded.

Individual Deliveries

Sudipto Das

- **User Login Page**
 - Login page for recruiters and hiring administrators.
- **User Home Page**
 - Recruiter Home Page with options:
 - Handling bulk uploads and managing the UI after the upload.
 - Upload resumes for specific jobs
 - Manual upload of JD and Role Clarity Document.
 - View candidates after resume uploads (for demo purposes only).
 - Upload Role Clarity Document and JD
 - Manual upload of JD and Role Clarity Document.
 - View candidates after resume uploads (for demo purposes only).
- **Integrating with Backend APIs**

Prashant Sahu

- **Recruiter Registration Page**

- Recruiters can request HR admin for registration.
- **Hiring Admin Home Page**
 - Can track recruiters and their respective job openings.
 - Can view recruiter registration requests and approve them.
- **Responsive Design**
 - Making the web app mobile-friendly.
 - UI improvements and styling.
- **Integrating with Backend APIs**

Nidhi Dhinoja

- **Database Implementation**
 - Implement database tables using Node.js and Sequelize.
 - Set up associations and run migrations.
 - Resolve foreign key constraint issues.
 - Add database to a centralized server for shared access.
 - Ensuring all the data is stored correctly or not if not then debugging the issues.

Nainesh Vidja

- **User Registration API**
 - Develop an endpoint to allow users to create accounts.
 - Implement validation for user inputs.
 - Apply password hashing using bcrypt.
- **User Login API**
 - Develop an endpoint for users to log in with email and password.

Rohit

- **Resume Upload Endpoint**
 - Develop and test endpoints for uploading resumes in different formats (PDF, DOCX, IMG, etc.).
 - Ensure the system correctly processes resumes from various formats.
- **Text Extraction from Resumes**
 - Implement functionality to extract text from resumes in multiple formats.
 - Handle errors and edge cases such as corrupted or unreadable files.
- **Library and API Experimentation**
 - Test different libraries and APIs (e.g., PyPDF2, PDFMiner, docx, pytesseract) for text extraction.
 - Compare the performance and accuracy of different libraries.
 - Optimize implementation based on results.

Sanjana

- **Resume Data Parsing**

- Parse extracted text to identify key information (e.g., name, contact details, skills, work experience).
- Develop logic to structure parsed data into a standardized format (e.g., JSON, database entries).
- **Library and API Experimentation**
 - Test various libraries, pre-trained models and APIs (e.g., Spacy, NLTK, StanfordNLP, Hugging Face Model Hub, custom regex, Gemini API, etc.) for parsing structured data.
 - Evaluate how well each library handles different resume formats.
 - Optimize parsing logic based on results.
- **Data Validation and Testing**
 - Ensure parsed data is accurate and consistent.
 - Test with different resume formats to ensure robustness.

Niket Patadiya

- **Authentication Middleware**
 - Implement middleware to protect authentication-required routes.
 - Generate and verify JWT tokens for secure access.
- **Basic API Testing**
 - Perform API testing for authentication endpoints.
 - Validate security and error-handling mechanisms.

Vinay Danidhariya

- **Resume and Document Upload**
 - Implement API routes to upload resumes and role clarity documents.
 - Store uploaded files in an S3 bucket.
 - Maintain database entries for uploaded documents.

Rishi Shori

- **GitHub Branching Setup**
 - Set up GitHub repository branching strategies.
- **AWS System Architecture**
 - Design and configure AWS system architecture S3 bucket for project deployment.
- **Create a database server**
 - Create a centralised server for the database.

Complete Features at End of Sprint

- ATS system with recruiter and hiring admin login.
- Uploading role clarity documents for job postings.
- Uploading resumes for job applications.
- Display all the parsed candidate resume information (not integrated with portal)