Title: ResuMatch

Problem Statement- AI-Powered Resume Screening & Job Matching

Team Name- Hack the present

TEAM INNOVATORS

Members	Student ID
Akshay Kumar Tapraniya	202311006
Arpit Jain	202311016
Gokul Faleja	202311032
Kavya Sanghvi	202311042
Saransh Naik	202311077

OBJECTIVE AND SOLUTION OVERVIEW

Objective:
Create an intelligent resume screening platform that automates candidate-job matching, reduces bias, and provides actionable insights for recruiters.

Solution Overview:
ResuMatch uses advanced algorithms to analyze resumes, extract key skills and experience, and match candidates to job requirements with precision. The platform streamlines the entire recruitment workflow from job posting to candidate selection.

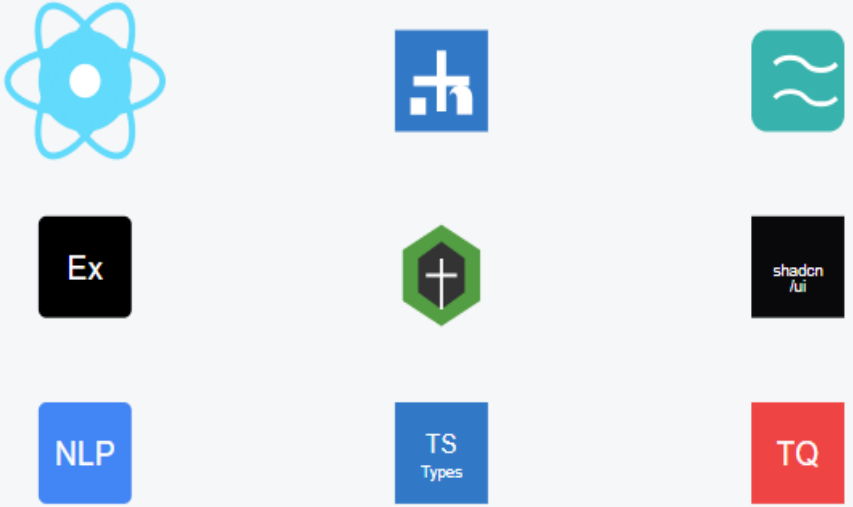
KEY FEATURES & TECHNOLOGIES

KEY FEATURES

- Intelligent resume parsing and skill extraction
- AI-powered matching algorithm
- Interactive analytics dashboard
- Bias-reduction mechanisms
- Customizable job requirement weighting

Technologies Used:

- Frontend: React, TypeScript, TailwindCSS, shadcn/ui
- Backend: Express.js, Node.js
- NLP: Natural language processing for text analysis
- Data Management: In-memory storage with TypeScript types
- State Management: TanStack Query



SOLUTIONS IMPLEMENTED

Technical Solutions:

- Created a schema-first architecture ensuring type safety between frontend and backend
- Implemented custom resume parsing that extracts skills, education, and experience
- Built a sophisticated matching algorithm that scores candidates against job requirements
- Developed an intuitive UI with real-time feedback on candidate-job compatibility
- Added export functionality and filtering capabilities for recruiter workflow

Business value:

- Reduces screening time by up to 75%
- Increases quality of candidate shortlisting
- Provides data-driven insights for hiring decisions
- Minimizes unconscious bias in candidate evaluation

CHALLENGES AND LEARNINGS

Challenges Faced:

•Accurately extracting relevant skills from unstructured resume text

•Ensuring type consistency between frontend forms and backend data storage

•Creating an algorithm that balances technical matching with soft skills

•Building an intuitive UI that simplifies complex matching data

Key Learnings:

•The importance of unified type definitions between frontend and backend

•Value of user-centric design in complex technical applications

•Benefits of iterative development and testing with real resume data

•Balancing automation with human oversight in recruitment processes