HEIZEN

Product Requirements Document (PRD)

Project: Workforce Management Platform

1. Overview

1.1. Project Purpose

To build a workforce management platform that connects organizations with candidates through a streamlined job posting and application process. The platform enables organizations to publish job openings and manage candidate applications efficiently.

1.2. Project Goal

Deliver a functional MVP that allows:

- Admin to onboard organizations
- Organizations to create and publish job openings
- Candidates to discover and apply for jobs
- Organizations to review and filter applications

1.3. Target Audience

- **Primary:** Organizations seeking to hire talent
- **Secondary:** Job seekers/candidates looking for opportunities
- **Tertiary:** Platform administrators managing organization onboarding

2. Features & Scope

2.1. Week 1: UX Prototype & Design Finalization (3 days)

Objective: Create interactive prototypes to finalize user flows, fields, and overall UX with Sarah before development begins.

User Stories:

- **U1:** As Sarah (stakeholder), I want to review clickable prototypes so that I can validate the user experience before development.
- **U2:** As a product team, we want to confirm all required fields and data

structures so that development proceeds without ambiguity.
- **U3:** As a designer, I want to establish the complete user journey so that all edge cases are considered.

Technical Deliverables:

- High-fidelity clickable prototypes (Figma/similar)
- Documented user flows for all personas (Admin, Org, Candidate)
- Finalized field requirements and data schema
- Approved design system and component library alignment with shadcn

2.2. Week 2: Core Development (7 working days)

Objective: Build and deploy the complete workforce management platform with all core features.

User Stories:

Admin Portal:

- **U4:** As an admin, I want to invite organizations by entering their email and organization name so that they can access the platform.
- **U5:** As an admin, I want a dashboard to view all registered organizations so that I can manage the platform.

Authentication System:

- **U6:** As a user (Org/Candidate), I want to sign up/login using email + OTP so that I can securely access my account without passwords.
- **U7:** As a user, I want to receive OTP via email so that I can verify my identity.

Organization Features:

- **U8:** As an organization, I want to log in with email + OTP so that I can access my dashboard.
- **U9:** As an organization, I want to create a job opening from predefined templates so that I can quickly post positions.
- **U10:** As an organization, I want to publish job openings so that candidates can discover them.
- **U11:** As an organization, I want to view all candidates who applied to my job postings so that I can review applications.
- **U12:** As an organization, I want to apply basic filters to candidate lists so that I can find relevant applicants quickly.

Candidate Features:

- **U13:** As a candidate, I want to browse published job openings so that I can find opportunities.
- **U14:** As a candidate, I want to sign up/login with email + OTP so that I can apply for jobs.
- **U15:** As a candidate, I want to fill in my information and create a profile so that organizations can review my application.
- **U16:** As a candidate, I want to apply directly to job openings so that I can submit my candidacy.
- **U17:** As a candidate, I can go through the dynamic list of questions that can determine my documents requirements that will be option during onboarding

Technical Deliverables:

- Fully functional frontend (Next.js + ShadCN UI)
- RESTful API backend (Nest.js)
- PostgreSQL database with proper schema and relationships
- Email + OTP authentication system
- Admin panel for organization invitations
- Organization dashboard (job creation, publishing, candidate management)
- Candidate portal (job browsing, profile creation, application submission)
- Basic filtering functionality for candidate lists
- State management implementation (Zustand)
- Data fetching and caching (TanStack Query)
- Deployed application (staging environment)

3. Technical Architecture

Component	Technology	Purpose
Frontend Framework	Next.js 14+ (App Router)	Server-side rendering, routing, and UI framework
Backend Framework	Nest.js	RESTful API, business logic, and server-side operations
UI Components	shadcn/ui	Pre-built, accessible component library

Database	PostgreSQL	Relational data storage for users, orgs, jobs, applications
State Management	Zustand	Client-side global state management
Data Fetching	TanStack Query	Server state management, caching, and data synchronization
Authentication	Email + OTP	Passwordless authentication system
Email Service	(TBD - e.g., SendGrid, AWS SES)	OTP delivery and notifications
Hosting	(TBD - e.g., Vercel, AWS)	Application deployment

---

4. Timelines

Week	Focus	Key Deliverables
Week 1 (Days 1-3)	UX Design & Prototyping	• Interactive prototypes • Finalized user flows • Confirmed field requirements • Design approval from Sarah

Week 2 (Days 4-10)	Full-Stack Development	• Authentication system • Admin portal • Org dashboard • Candidate portal • Job posting & application features • Basic filtering • Deployed MVP

Total Duration: 10 working days (2 weeks)

5. Assumptions & Dependencies

Assumptions:

- Sarah will be available for prototype review sessions within Week 1
- Job opening templates will be provided/defined during prototype phase
- Email service provider account is available for OTP delivery
- Hosting infrastructure is ready or can be set up quickly
- No complex approval workflows required in MVP
- Basic filtering refers to simple field-based filters (e.g., date applied, status)

Dependencies:

- Sarah's timely feedback on prototypes (Week 1)
- Access to email service provider API keys
- Database hosting credentials
- Domain name and SSL certificates (if applicable)
- Design assets and brand guidelines (if any)

6. Out of Scope

The following features are **NOT** included in this 2-week MVP:

- Advanced candidate matching algorithms
- Interview scheduling functionality
- Video/document upload capabilities
- Multi-stage hiring workflows
- Email notifications beyond OTP
- Candidate messaging system
- Analytics and reporting dashboards
- Mobile applications (native)
- Payment processing
- Background check integrations
- Resume parsing/AI features
- Custom job template creation (only hardcoded templates)
- Role-based access control within organizations (multi-user orgs)
- Advanced search functionality
- Social media login options
- Candidate profile editing after creation

7. Requirements from Client

Functional Requirements:

- Email + OTP must be the only authentication method
- Job templates must be hardcoded (no dynamic template creation)
- Candidates must create a complete profile during application
- Organizations must see all applicants for their jobs
- Basic filtering capability for candidate lists

Non-Functional Requirements:

- Responsive design (desktop and mobile)
- Secure authentication flow
- Fast page load times (<3 seconds)
- Clean, intuitive UI aligned with shadcn design system

Prototype Phase Requirements:

- 3 days allocated for UX design and client confirmation
- Sarah must approve designs before development begins

- All user flows and fields must be documented

8. Project Pricing

Rate: $1,000 per sprint (one week)

Duration: 2 weeks (2 sprints)

- Week 1: Prototype & Design Finalization
- Week 2: Full-Stack Development

Total Project Cost: $2,000