■ Job Automation Platform

Product Requirements Document

For Development with Google Gemini + Claude Code

Version 1.0 MVP | February 02, 2026

■ Table of Contents

■ Document Overview

Product Name: JobAutomate

Version: 1.0 MVP

Target Launch: 6 weeks from start

Development Tools:

• AI Provider: Google Gemini (Antigravity)

• Development Assistant: Claude Code

• Primary Tech: Puppeteer, Chrome Extension, Node.js, React

■ Product Vision

One-Sentence Description:

"A Chrome extension and web platform that scrapes job listings automatically and helps users apply to jobs 10x faster using AI-powered form filling."

Problem We're Solving:

• Job hunting is time-consuming (applying takes 10-20 minutes per job)

• Job boards are fragmented (LinkedIn, Indeed, Glassdoor, etc.)

• Same questions asked repeatedly

• Hard to track applications across multiple platforms

Solution:

• Automatically collect jobs from multiple sources

• Display them in one unified dashboard

• Auto-fill applications using AI-generated, personalized answers

• Track all applications in one place

■ Target Users

Primary Users:

• Job Seekers (actively looking for work)

• Career Changers (transitioning industries)

• Recent Graduates (applying to many positions)

• Passive Job Seekers (exploring opportunities)

User Personas:

Persona 1: Sarah	Persona 2: Mike	Persona 3: Lisa
Recent Graduate	Career Changer	Passive Seeker
Age: 24	Age: 32	Age: 28
Looking for entry-level Software Engineer roles	Looking for mid-level Product Manager roles	Looking for better opportunities
Pain: Exhausted by repetitive forms	Pain: Unsure how to position himself	Pain: No time to browse daily
Goal: Apply faster, track better	Goal: AI help with custom questions	Goal: Get notified of best matches

■■ System Architecture

Component Overview:

• **Component 1: Puppeteer Scraper** - Runs on laptop/server, scrapes LinkedIn, Indeed, Glassdoor

• **Component 2: Chrome Extension** - Dual purpose: (A) Extracts job data in Puppeteer, (B) Auto-fills applications in user browser

• **Component 3: Backend API** - Node.js + Express, receives scraped jobs, handles AI requests, manages user data

• **Component 4: Database** - PostgreSQL, stores jobs, users, applications

• **Component 5: Web App** - React/Next.js, user dashboard, job browsing, application tracking

■ Technical Specifications

Tech Stack:

Category	Technology	Purpose
Frontend	React + Next.js 14	Modern UI framework
	Tailwind CSS	Styling
	Vercel	Hosting (free tier)
Backend	Node.js 18+	Runtime
	Express.js	API framework
	REST	API style
Database	SQLite → PostgreSQL	Data storage
	Prisma/Drizzle ORM	Database ORM
Extension	Manifest V3	Chrome extension standard
	Content Scripts	Page interaction
Automation	Puppeteer	Browser automation
	Node-cron	Scheduling
AI Provider	Google Gemini API	Answer generation

■ Component 1: Puppeteer Scraper

Purpose: Automatically collect jobs from multiple platforms

Job Source Support:

• LinkedIn (priority 1)

• Indeed (priority 2)

• Glassdoor (priority 3)

Data to Collect:

• Job title, company name, location

• Salary range (if available)

• Job type (Full-time, Part-time, Contract)

• Posted date

• Full job description

• Requirements and benefits

• Apply URL

• Easy Apply status

Scheduling:

• Run every 6 hours (configurable)

• Stagger scraping between platforms

• Scrape 100-500 jobs per run

■ Component 2: Chrome Extension

Purpose:

• In Puppeteer: Extract job data from pages

• In User Browser: Auto-fill job applications

Key Features:

• **Feature 1: Job Data Extraction** - Detect platform, use CSS selectors, extract job info, send to API

• **Feature 2: Form Auto-Fill** - Detect application forms, identify fields, fill basic info from resume

• **Feature 3: AI Answer Generation** - Detect custom questions, generate contextual answers using AI

• **Feature 4: User Interface** - Popup for resume upload, settings, application tracking

Platform-Specific Selectors:

The extension uses platform-specific CSS selectors to extract data from LinkedIn, Indeed, and Glassdoor job pages.

■ Component 3: Backend API

Purpose: Handle data, AI requests, user management

Key Endpoints:

Endpoint	Method	Purpose
/api/jobs	GET	List all jobs with filters
/api/jobs/:id	GET	Get job details
/api/jobs	POST	Add scraped job
/api/ai/generate	POST	Generate AI answer
/api/auth/register	POST	User signup
/api/auth/login	POST	User login
/api/applications	GET	List user applications
/api/applications	POST	Track new application

Gemini Integration:

Uses Google Gemini API (gemini-pro model) for generating contextual, personalized answers to job application questions.

■ Component 4: Database Schema

Key Tables:

• **jobs** - Stores all scraped job listings with title, company, location, salary, description, apply URL, etc.

• **users** - User accounts with email, resume, preferences, subscription tier

• **applications** - Tracks which users applied to which jobs, with status and notes

• **ai_usage** - Monitors AI API usage for billing and optimization

■ Component 5: Web Application

Key Pages:

• **Landing Page (/):** Hero section, feature highlights, how it works, pricing, CTA

• **Dashboard (/dashboard):** Search jobs, apply filters, view job cards, quick apply

• **Job Detail (/jobs/:id):** Full description, requirements, similar jobs, apply button

• **Applications (/applications):** Track status, filter, search, add notes

• **Profile (/profile):** Resume management, preferences, API usage, subscription

■ Security & Privacy

Data Protection:

• Encrypt user resumes at rest

• HTTPS only for all API calls

• Hash passwords with bcrypt (12 rounds)

• Sanitize all user inputs

• Rate limiting on all endpoints

Privacy Policy:

• Users own their data

• Can export data anytime

• Can delete account and all data

• No selling of user data

• Transparent about AI usage

■ Pricing & Monetization

Tier	Price	Features
Free	$0/month	10 AI answers/month, 100 applications tracked, basic filtering
Premium	$9.99/month	Unlimited AI answers, unlimited tracking, advanced filtering, multiple resumes
Pro	$19.99/month	All Premium + AI job matching, automated alerts, interview prep, priority support

■ Success Metrics

MVP Success Criteria:

• 50 active users within first month

• 500+ applications tracked

• 80% user satisfaction (surveys)

• 5+ testimonials from beta users

• 10% conversion from free to paid

Key Performance Indicators:

• Daily Active Users (DAU)

• Applications per user per week

• AI answer acceptance rate

• Time saved per application

• Conversion rate (free → paid)

• Average revenue per user (ARPU)

■ Development Timeline

Week	Phase	Deliverables
1	Chrome Extension	Extension structure, job page detection, data extraction
2	Puppeteer Scraper	Set up Puppeteer, install extension, scraping script
3	Backend + Database	API endpoints, database setup, AI integration
4	Web App	Job listing page, filters, authentication
5	Integration	Connect components, end-to-end testing
6	Testing & Launch	Beta testing, bug fixes, public launch

■ Deployment Strategy

Phase 1: Local Development (Week 1-5)

• Everything runs on your laptop

• SQLite database

• Puppeteer runs manually

• Web app runs locally

Phase 2: Staging (Week 6)

• Deploy web app to Vercel

• Keep backend + Puppeteer on laptop

• Invite 10 beta testers

Phase 3: Production (Week 7+)

• Move Puppeteer to VPS (Hetzner $5/month)

• Backend API on same VPS

• PostgreSQL on VPS or managed service

• Chrome Extension published to Web Store

• Web app on Vercel

■ Cost Breakdown

Phase	Monthly Cost	Details
Development (Week 1-5)	$0-$10	Laptop, free tiers, light AI testing
Staging (Week 6)	$5-$15	Vercel free, AI API usage
Production (Week 7+)	$5-$50	VPS $5, Database $0-$25, AI API $0-$20
At Scale (1000+ users)	$300-$500	Better VPS, managed DB, higher AI usage

■ MVP Feature Priority

Must Have (P0):

■ Scrape jobs from LinkedIn

■ Store jobs in database

■ Display jobs on web app

■ Basic filtering (title, location)

■ Chrome extension detects application forms

■ Auto-fill basic info (name, email, phone)

■ AI generates answers for custom questions

■ User can upload resume

■ Track applications

Should Have (P1):

■ Scrape from Indeed and Glassdoor

■ Advanced filtering (salary, remote)

■ User authentication

■ Application status updates

■ Email notifications

Nice to Have (P2):

■ Multiple resume profiles

■ AI job matching scores

■ Company research

■ Application analytics

■ Dark mode

■ Next Steps

Immediate Actions:

1. Set up development environment - Install Node.js 18+, PostgreSQL/SQLite, create project folders

2. Get API keys - Google Gemini API key (required)

3. Start with Chrome Extension - This is the core differentiator and easiest to test

4. Use Claude Code - Provide this PRD, ask to build extension first, iterate together

Recommended First Prompt for Claude Code:

```
"Please help me build the Chrome Extension as specified in this PRD. Let's start with the
manifest.json and content script that detects LinkedIn job pages and extracts job data."
```

■ Resources for Development

Documentation Links:

• Puppeteer: https://pptr.dev/

• Chrome Extension: https://developer.chrome.com/docs/extensions/mv3/

• Next.js: https://nextjs.org/docs

• Prisma: https://www.prisma.io/docs

• Gemini API: https://ai.google.dev/docs

Environment Variables Required:

• GEMINI_API_KEY - Google Gemini API key

• DATABASE_URL - PostgreSQL connection string

• JWT_SECRET - For user authentication

• PORT - Server port (default 3001)

■ Ready to Build!

This PRD provides everything needed to start development with Google Gemini and Claude Code.

Timeline: 6 weeks to working MVP

Cost: $5-15 for MVP, $5-50/month after launch

Good luck! ■