

JobBot Content Empire

Complete Replit Development Prompt for Multi-Agent Job Automation & Viral Content System

Python • FastAPI • AI Agents • Automation • Analytics

Project Overview

System Purpose

Build a comprehensive job application automation system with integrated viral content creation for Remy (LoveLogicAI founder) targeting restaurant management, cannabis industry, high-end sales, and tech consulting positions across the Brentwood to Walnut Creek Bay Area corridor.

Key Features

- Multi-industry job automation
- AI-powered resume generation
- Viral content creation
- Social media automation
- Performance analytics

Multi-Agent System Architecture

Job Hunting Agents

- RestaurantManagerBot
- CannabisLeadershipBot
- HighEndSalesBot
- TechConsultantBot

Content Creation Agents

- ViralStrategistAgent
- ContentGeneratorAgent
- SocialMediaManagerAgent
- AnalyticsTrackerAgent

Support Systems

- MasterOrchestrator
- ConfigurationManager
- DatabaseManager
- APIIntegrationHub

Technical Implementation Stack

Backend Technologies

```
# Core Framework
- Python 3.11+
- FastAPI (async web framework)
- Pydantic (data validation)
- SQLAlchemy (ORM)
- Alembic (migrations)

# AI & Automation
- OpenAI API (GPT-4 for content)
- LangChain (agent orchestration)
- LangGraph (workflow management)
- Selenium (web automation)
- BeautifulSoup4 (web scraping)

# Task Management
- Celery (background tasks)
- Redis (message broker)
- APScheduler (job scheduling)

# Database
- PostgreSQL (primary data)
- Redis (caching/sessions)
```

Frontend & Integration

```
# Frontend
- React 18 + TypeScript
- Tailwind CSS (styling)
- Chart.js (analytics)
- Socket.io (real-time updates)

# Social Media APIs
- TikTok for Developers API
- Instagram Basic Display API
- LinkedIn Marketing API
- Twitter API v2
- YouTube Data API v3

# Job Platforms
- Indeed API
- LinkedIn Jobs API
- Google Jobs API
- Custom scrapers for niche sites

# File Storage
- AWS S3 (resumes/content)
- Local storage (development)
```

Complete Replit Development Prompt

PROMPT FOR REPLIT AI:

Create a complete job application automation system with viral content creation called "JobBot Content Empire" with the following specifications:

CORE FUNCTIONALITY:

- Multi-agent job hunting system targeting 4 industries: restaurant management, cannabis retail, high-end sales, and tech consulting
- Automated resume and cover letter generation using AI
- Web scraping and application submission automation
- Viral content creation for TikTok, Instagram, LinkedIn, YouTube
- Social media posting automation
- Real-time analytics and performance tracking

TECHNICAL ARCHITECTURE:

- Backend: Python FastAPI with async/await patterns
- Database: PostgreSQL with SQLAlchemy ORM
- Task Queue: Celery with Redis broker
- Frontend: React with TypeScript and Tailwind CSS
- AI Integration: OpenAI API, LangChain agents
- Web Automation: Selenium WebDriver

USER PROFILE CONFIGURATION:

Name: Jeremy "Remy" Morgan-Jones Sr.
Company: LoveLogicAI LLC (founder)
Experience: 5+ years restaurant (all positions), 3 years cannabis (Abundant Delivery), 2 years high-end sales (\$100K in 100 days achievement)
Location: Brentwood to Walnut Creek corridor, Bay Area, CA
Target: Management and senior positions with experience-based positioning

AGENT SPECIFICATIONS:

- RestaurantManagementBot:** Target chains like Cheesecake Factory, Olive Garden for management roles
- CannabisLeadershipBot:** Focus on dispensary manager positions at Cookies, Harborside, The Apothecarium
- HighEndSalesBot:** Target Tesla, BMW, Williams Sonoma leveraging \$100K achievement
- TechConsultantBot:** Remote AI/ML consulting highlighting LoveLogicAI founder status
- ViralContentAgent:** Create TikTok/Instagram content documenting the automation process

Project File Structure

```
jobbot-content-empire/
├── backend/
│   ├── app/
│   │   ├── __init__.py
│   │   ├── main.py           # FastAPI application
│   │   ├── config.py        # Configuration settings
│   │   ├── database.py      # Database connection
│   │   └── models/
│   │       ├── __init__.py
│   │       ├── user.py      # User/profile models
│   │       ├── job.py       # Job and application models
│   │       ├── content.py   # Social media content models
│   │       └── analytics.py # Analytics and tracking models
│   ├── agents/
│   │   ├── __init__.py
│   │   ├── base_agent.py   # Base agent class
│   │   └── job_hunters/
│   │       ├── restaurant_bot.py
│   │       ├── cannabis_bot.py
│   │       ├── sales_bot.py
│   │       └── tech_bot.py
│   ├── content_creators/
│   │   ├── viral_strategist.py
│   │   ├── content_generator.py
│   │   ├── social_media_manager.py
│   │   ├── analytics_tracker.py
│   │   └── orchestrator.py # Master orchestrator
│   ├── services/
│   │   ├── __init__.py
│   │   ├── job_scraper.py  # Web scraping service
│   │   ├── resume_generator.py # AI resume creation
│   │   ├── social_media.py # Social media API integration
│   │   ├── email_service.py # Email notifications
│   │   └── file_manager.py  # File upload/storage
│   ├── api/
│   │   ├── __init__.py
│   │   ├── auth.py         # Authentication endpoints
│   │   ├── jobs.py         # Job related endpoints
│   │   ├── content.py      # Content management endpoints
│   │   ├── analytics.py    # Analytics endpoints
│   │   └── automation.py   # Automation control endpoints
│   ├── utils/
│   │   ├── __init__.py
│   │   ├── logger.py       # Logging configuration
│   │   ├── helpers.py      # Utility functions
│   │   └── validators.py   # Data validation
│   └── tasks/
│       ├── __init__.py
│       ├── job_hunting.py  # Background job hunting tasks
│       ├── content_creation.py # Content generation tasks
│       └── social_posting.py # Social media posting tasks
├── requirements.txt
├── alembic.ini             # Database migration config
├── migrations/            # Database migrations
├── frontend/
│   ├── public/
│   │   ├── index.html
│   │   └── manifest.json
│   └── src/
│       ├── components/
│       │   ├── Dashboard.tsx # Main dashboard
│       │   ├── JobsPanel.tsx # Job applications view
│       │   ├── ContentPanel.tsx # Content management
│       │   ├── AnalyticsPanel.tsx # Analytics dashboard
│       │   └── SettingsPanel.tsx # Configuration
│       ├── services/
│       │   ├── api.ts        # API client
│       │   ├── websocket.ts  # Real-time updates
│       │   ├── auth.ts       # Authentication
│       ├── hooks/
│       │   ├── useJobs.ts    # Job data hooks
│       │   ├── useContent.ts # Content hooks
│       │   ├── useAnalytics.ts # Analytics hooks
│       ├── utils/
│       │   ├── constants.ts  # App constants
│       │   └── helpers.ts     # Utility functions
│       ├── App.tsx
│       ├── index.tsx
│       ├── package.json
│       └── tailwind.config.js
├── config/
│   ├── jobbot-config.yaml # Main configuration
│   ├── target_companies.yaml # Company targets
│   ├── content_templates.yaml # Content templates
│   ├── docker-compose.yml # Development environment
│   ├── Dockerfile         # Production container
│   ├── env.example        # Environment variables template
│   ├── .gitignore
│   ├── README.md
│   └── pyproject.toml     # Python project configuration
```

Implementation Roadmap

- Core Infrastructure Setup**
Initialize FastAPI backend, PostgreSQL database, React frontend, and basic authentication system.
- User Profile & Configuration**
Build YAML-based configuration system for user profile, job preferences, and industry targeting.
- Job Hunting Agents**
Implement the four specialized job hunting bots with web scraping and application automation.
- AI Resume & Cover Letter Generation**
Integrate OpenAI API for dynamic resume and cover letter creation based on job requirements.
- Content Creation System**
Build viral content generation agents for TikTok, Instagram, LinkedIn, and YouTube.
- Social Media Automation**
Integrate social media APIs for automated posting and engagement tracking.
- Analytics & Monitoring**
Implement comprehensive analytics for job applications, content performance, and ROI tracking.
- Production Deployment**
Deploy to production with proper scaling, monitoring, and backup systems.

Required API Integrations

Job Platforms

LinkedIn Jobs API
Professional networking and job search

Indeed API
Large job aggregation platform

Google Jobs API
Google's job search integration

AngelList API
Startup and tech job opportunities

Social Media Platforms

TikTok for Developers
Video content automation

Instagram Basic Display API
Photo/video content posting

LinkedIn Marketing API
Professional content publishing

YouTube Data API v3
Video upload and management

Database Schema Design

```
# Core Models

class User(Base):
    id: UUID
    email: str
    name: str
    profile_data: JSON
    created_at: datetime
    updated_at: datetime

class JobApplication(Base):
    id: UUID
    user_id: UUID
    company_name: str
    position_title: str
    industry: str # restaurant, cannabis, sales, tech
    application_date: datetime
    status: str # pending, applied, responded, interviewed, offered, rejected
    application_url: str
    resume_version: str
    cover_letter_content: TEXT
    response_data: JSON

class Company(Base):
    id: UUID
    name: str
    industry: str
    location: str
    website: str
    career_page_url: str
    contact_info: JSON
    last_scraped: datetime

class ContentPost(Base):
    id: UUID
    user_id: UUID
    platform: str # tiktok, instagram, linkedin, youtube
    content_type: str # video, image, text
    title: str
    caption: TEXT
    hashtags: str
    media_url: str
    scheduled_time: datetime
    posted_time: datetime
    engagement_metrics: JSON

class Analytics(Base):
    id: UUID
    user_id: UUID
    date: date
    metric_type: str # job_applications, content_engagement, interviews
    metric_value: float
    metadata: JSON
```

Configuration System

User Profile YAML

```
personal_info:
  name: "Jeremy Morgan-Jones Sr."
  email: "remy@lovelogicai.com"
  phone: "1-xxx-xxx-xxxx"
  linkedin: "linkedin.com/in/remy"

location_preferences:
  primary_location: "Brentwood to Walnut Creek, CA"
  search_radius: 45
  remote_preference: "hybrid_preferred"

job_criteria:
  tier_1_roles:
    - "AI Engineer"
    - "ML Engineer"
    salary_range: [200000, 200000]
  tier_2_roles:
    - "Dispensary Manager"
    - "Restaurant Manager"
    salary_range: [50000, 80000]

experience:
  restaurant:
    years: 5
    positions: ["Server", "Bartender", "Cook"]
  cannabis:
    years: 3
    company: "Abundant Delivery"
  sales:
    years: 2
    achievement: "$100K in 100 days"
  entrepreneurship:
    current: "LoveLogicAI LLC Founder"
```

Automation Settings

```
automation_settings:
  daily_limits:
    tier_1: 5
    tier_2: 15
    tier_3: 35

  content_schedule:
    tiktok:
      posts_per_day: 3
      best_times: ["9:00", "15:00", "21:00"]
    linkedin:
      posts_per_day: 1
      best_times: ["10:00"]
    instagram:
      posts_per_day: 2
      best_times: ["12:00", "18:00"]

  platforms:
    linkedin:
      easy_apply: true
      connection_requests: true
      indeed:
        auto_apply: true
        daily_limit: 25
        company_direct:
          enabled: true
          follow_up_days: 7
```

Replit Deployment Guide

1. Initialize Replit Project

```
replit init jobbot-content-empire --template python
cd jobbot-content-empire
```

2. Install Dependencies

```
pip install fastapi uvicorn sqlalchemy psycopg2-binary
pip install celery redis python-multipart
pip install openai langchain selenium beautifulsoup4
pip install pydantic-settings python-jose passlib
```

3. Environment Variables

```
OPENAI_API_KEY=your_openai_key
DATABASE_URL=postgresql://user:pass@localhost/jobbot
REDIS_URL=redis://localhost:6379
SECRET_KEY=your_secret_key
LINKEDIN_CLIENT_ID=your_linkedin_id
INSTAGRAM_ACCESS_TOKEN=your_instagram_token
```

4. Run Application

```
# Start backend
uvicorn app.main:app --host 0.0.0.0 --port 8000 --reload

# Start Celery worker (separate terminal)
celery -A app.tasks.worker --loglevel=info

# Start frontend (separate terminal)
cd frontend && npm start
```

Success Metrics & KPIs

Job Application Metrics

- Applications per day: 50+
- Response rate: 25%+
- Interview rate: 15%+
- Offer rate: 5%+
- Time per application: <2min

Content Performance

- TikTok views: 10K+ per video
- LinkedIn engagement: 5%+
- Instagram reach: 5K+ per post
- Content creation: 15 posts/day
- Viral moments: 1 per week

Business Impact

- LoveLogicAI leads: 10+ per week
- Speaking opportunities: 2+ per month
- Media mentions: 5+ per month
- Follower growth: 1K+ per week
- Revenue increase: 50%+