

Atlas Universalis — Project Overview

This document provides a comprehensive inventory of all files in the Atlas Universalis project with descriptions of their purpose.

Last Updated: 2026-01-30

Project Repository: github.com/idiehl/site_repo

Production URL: atlasuniversalis.com

Project Summary

Atlas Universalis is a multi-application platform consisting of:

1. **Main Portfolio Site** (atlasuniversalis.com) - Company landing page showcasing projects and applications
 2. **QuickPRO** (quickpro.atlasuniversalis.com) - AI-powered job application management and resume generation platform (AtlasOps)
 3. **Browser Extension** - Chrome/Edge extension for capturing job postings
-

Technology Stack

Layer	Technology
Backend	Python 3.11+, FastAPI, SQLAlchemy (async)
Database	PostgreSQL
Migrations	Alembic
Queue	Redis + Celery
Frontend (Main)	Vue 3 + Vite + Tailwind CSS
Frontend (QuickPRO)	Vue 3 + Vite + Pinia + Vue Router
Reverse Proxy	Nginx
Hosting	DigitalOcean Droplet
CI/CD	GitHub Actions (SSH deploy)

Directory Structure

```
site_repo/
    alembic/                      # Database migrations
    atlasops/                     # QuickPRO backend application
    deploy/                        # Deployment configurations
    docs/                          # Documentation
    extension/                    # Browser extension (legacy location)
    frontend/                     # QuickPRO Vue frontend
    frontend-main/                # Main portfolio site Vue frontend
    internal/                     # Internal assets (wireframes, logo concepts)
    mcp/                           # Model Context Protocol server
    quickpro-extension/          # Browser extension (current)
    scripts/                       # Utility scripts
    templates/                    # Resume HTML templates
    app.py                         # FastAPI application entrypoint
    requirements.txt               # Python dependencies
```

File Inventory

Root Configuration Files

File	Purpose
app.py	FastAPI application entrypoint. Mounts AtlasOps API, serves static files, handles routing
requirements.txt	Python package dependencies
alembic.ini	Alembic migration configuration
.env.example	Environment variable template (secrets, API keys, database URL)
.gitignore	Git ignore patterns for node_modules, dist, pycache , .env
README.md	Project readme
package-lock.json	Root-level npm lock file

Backend: `atlasops/`

Core QuickPRO/AtlasOps backend application.

Configuration & Database

File	Purpose
atlasops/__init__.py	Package initialization
atlasops/config.py	Application configuration (env vars, settings)
atlasops/db.py	Database session management, SQLAlchemy async engine

API Layer: `atlasops/api/`

File	Purpose
api/__init__.py	API package init
api/deps.py	FastAPI dependencies (auth, database sessions)
api/v1/__init__.py	V1 API package init
api/v1/router.py	Main API router, aggregates all v1 endpoints
api/v1/auth.py	Authentication endpoints (login, register, OAuth)
api/v1/jobs.py	Job posting endpoints (ingest, list, detail, deep-dive)
api/v1/applications.py	Application tracking endpoints (status updates, follow-ups)
api/v1/profile.py	User profile endpoints (resume upload, profile enhancement)
api/v1/admin.py	Admin dashboard endpoints (user management, analytics)
api/v1/billing.py	Subscription/billing endpoints

Models: `atlasops/models/` SQLAlchemy ORM models for database tables.

File	Purpose
models/__init__.py	Model exports
models/user.py	User model (auth, profile, subscription)

File	Purpose
models/job.py	JobPosting model (scraped job data)
models/application.py	Application model (status tracking, events)
models/resume.py	GeneratedResume model (content, file paths)
models/analytics.py	Analytics/event tracking models

Schemas: `atlasops/schemas/` Pydantic models for request/response validation.

File	Purpose
schemas/__init__.py	Schema exports
schemas/user.py	User-related schemas (registration, profile, auth)
schemas/job.py	Job posting schemas (ingest request, job detail response)
schemas/application.py	Application schemas (status updates, events)
schemas/resume.py	Resume schemas (generation request, content structure)

Services: `atlasops/services/` Business logic layer.

File	Purpose
services/__init__.py	Service exports
services/llm_client.py	Unified LLM wrapper (OpenAI API calls, retries, logging)
services/scrapper.py	Job posting scraper (HTTP + Playwright fallback)
services/resume_generator.py	Resume generation service (template rendering, PDF)
services/resume_parser.py	Resume parsing service (extract structure from uploaded resume)
services/entitlements.py	Subscription/entitlement checking

Prompts: `atlasops/prompts/` Versioned LLM prompt templates.

File	Purpose
job_extraction_v1.md	Prompt for extracting structured job data from raw text
deep_dive_v1.md	Prompt for generating company deep-dive analysis
resume_optimizer_v1.md	Prompt for optimizing resume content for job match
resume_generator_v1.md	Prompt for generating tailored resume bullets
resume_parser_v1.md	Prompt for parsing uploaded resume into structured data
cover_letter_v1.md	Prompt for generating cover letters

Utilities: `atlasops/utils/`

File	Purpose
utils/__init__.py	Utility exports
utils/url_validator.py	URL validation, SSRF protection
utils/sanitize.py	HTML/text sanitization

Middleware: `atlasops/middleware/`

File	Purpose
middleware/__init__.py	Middleware exports
middleware/analytics.py	Request/response analytics tracking

Workers: atlasops/workers/

File	Purpose
workers/__init__.py	Worker exports
workers/tasks.py	Celery background tasks (scraping, resume generation)

Database Migrations: alembic/

File	Purpose
env.py	Alembic environment configuration
script.py.mako	Migration script template
versions/20260111_0001_initial_schema.py	Initial database schema
versions/20260115_0002_add_oauth_field.py	OAuth provider fields
versions/20260115_0003_enhance_user_profile.py	Enhanced user profile fields
versions/20260115_0010_add_cover_letter.py	Grouped newsletter table
versions/20260115_0011_add_admin_and_analytics.py	Analytics tables
versions/20260115_0012_add_subscription_field.py	Subscription provider fields
versions/20260116_0001_add_job_application_status.py	Application status enhancements

QuickPRO Frontend: frontend/

Vue 3 SPA for the QuickPRO application (served at quickpro.atlasuniversalis.com).

Configuration

File	Purpose
index.html	HTML entry point
package.json	npm dependencies and scripts
vite.config.js	Vite build configuration
tailwind.config.js	Tailwind CSS configuration
postcss.config.js	PostCSS configuration

Source: frontend/src/

File	Purpose
main.js	Vue app initialization
App.vue	Root Vue component
assets/main.css	Global styles, Tailwind imports

API Client: frontend/src/api/

File	Purpose
client.js	Axios HTTP client with auth interceptors

Router: frontend/src/router/

File	Purpose
index.js	Vue Router configuration, route definitions

State Stores: frontend/src/stores/

File	Purpose
auth.js	Pinia store for authentication state
jobs.js	Pinia store for job postings state
applications.js	Pinia store for applications state

Components: frontend/src/components/

File	Purpose
JobTable.vue	Data table displaying job postings
IngestModal.vue	Modal for URL ingestion
GenerateSection.vue	Resume generation interface
StatusBadge.vue	Application status badge component

Views: frontend/src/views/

File	Purpose
LoginView.vue	Login page
RegisterView.vue	Registration page
OAuthCallbackView.vue	OAuth callback handler
DashboardView.vue	Main user dashboard
JobDetailView.vue	Individual job posting detail
ApplicationsView.vue	Application tracking view
ProfileView.vue	User profile management
ExtensionView.vue	Browser extension info/download page
AdminDashboardView.vue	Admin overview dashboard
AdminAnalyticsView.vue	Admin analytics view
AdminSecurityView.vue	Admin security settings

Public Assets: frontend/public/

File	Purpose
quickpro-icon.svg	QuickPRO app icon
quickpro-extension.zip	Packaged browser extension
templates/*.svg	Resume template preview images

File	Purpose

Main Portfolio Site: `frontend-main/`

Vue 3 SPA for the main Atlas Universalis landing page (served at atlasuniversalis.com).

Configuration

File	Purpose
<code>index.html</code>	HTML entry point, favicon reference
<code>package.json</code>	npm dependencies
<code>vite.config.js</code>	Vite build configuration
<code>tailwind.config.js</code>	Tailwind CSS with custom color palette (atlas, night, gold)
<code>postcss.config.js</code>	PostCSS configuration

Source: `frontend-main/src/`

File	Purpose
<code>main.js</code>	Vue app initialization
<code>App.vue</code>	Root component
<code>assets/main.css</code>	Global styles, custom animations (fadeInUp, float)

Router: `frontend-main/src/router/`

File	Purpose
<code>index.js</code>	Route definitions (currently just home)

Components: `frontend-main/src/components/`

File	Purpose
<code>NavBar.vue</code>	Site navigation with mobile menu
<code>HeroSection.vue</code>	Homepage hero with logo, tagline, CTAs, scroll indicator
<code>AppsSection.vue</code>	Featured applications showcase
<code>ProjectsSection.vue</code>	Project portfolio section
<code>AboutSection.vue</code>	About/company information
<code>FooterSection.vue</code>	Site footer with contact info

Views: `frontend-main/src/views/`

File	Purpose
<code>HomeView.vue</code>	Homepage composition (assembles all sections)

Public Assets: `frontend-main/public/`

File	Purpose
<code>atlas-icon.svg</code>	Current logo - Orbital Hexagon design with A letterform
<code>atlas-icon.png</code>	Legacy PNG icon
<code>atlas-logo.png</code>	Legacy banner logo
<code>profile.png</code>	Profile photo
<code>quickpro-icon.svg</code>	QuickPRO icon
<code>logo-option-1.svg</code> through <code>logo-option-5.svg</code>	Logo design exploration
<code>logo-hex-1.svg</code> through <code>logo-hex-5.svg</code>	Hexagon logo variations
<code>logo-preview.html</code>	Logo comparison preview page

Browser Extension: `quickpro-extension/`

Chrome/Edge extension for capturing job postings from web pages.

File	Purpose
<code>manifest.json</code>	Extension manifest (permissions, scripts, icons)
<code>background.js</code>	Service worker for extension lifecycle
<code>content.js</code>	Content script injected into job posting pages
<code>content.css</code>	Styles for injected UI elements
<code>popup.html</code>	Extension popup UI
<code>popup.js</code>	Popup logic
<code>popup.css</code>	Popup styles
<code>generate_icons.py</code>	Script to generate icon sizes from SVG
<code>README.md</code>	Extension documentation
<code>icons/*.png</code>	Extension icons at various sizes
<code>img/*.png</code>	Store listing screenshots

Resume Templates: `templates/resumes/`

HTML templates for PDF resume generation.

File	Purpose
<code>basic.html</code>	Simple, ATS-friendly template
<code>classic.html</code>	Traditional professional template
<code>modern.html</code>	Contemporary design with accent colors
<code>creative.html</code>	Designer-style template
<code>executive.html</code>	Senior/leadership focused template
<code>simple.html</code>	Minimal, clean template
<code>technical.html</code>	Developer/engineering focused template

Deployment: deploy/

Nginx configurations and deployment utilities.

File	Purpose
nginx-main-site.conf	Nginx config for atlasuniversalis.com
nginx-quickpro.conf	Nginx config for quickpro subdomain
nginx-quickpro-fixed.conf	Updated QuickPRO nginx config
SETUP.md	Deployment setup documentation
clean_env.py	Environment cleanup utility
fix_env.py	Environment fix utility

GitHub Actions: .github/workflows/

File	Purpose
deploy.yml	CI/CD workflow: SSH deploy on push to master

Utility Scripts: scripts/

File	Purpose
setup_users.py	Initial user setup script
make_admin.py	Promote user to admin
create_test_user.py	Create test user for development
check_user.py	User verification utility
test_templates.py	Template testing utility
build_master_log.ps1	Windows script to build Master_Log.pdf
build_master_log.sh	Linux/Mac script to build Master_Log.pdf

Internal Assets: internal/

Non-deployed internal resources.

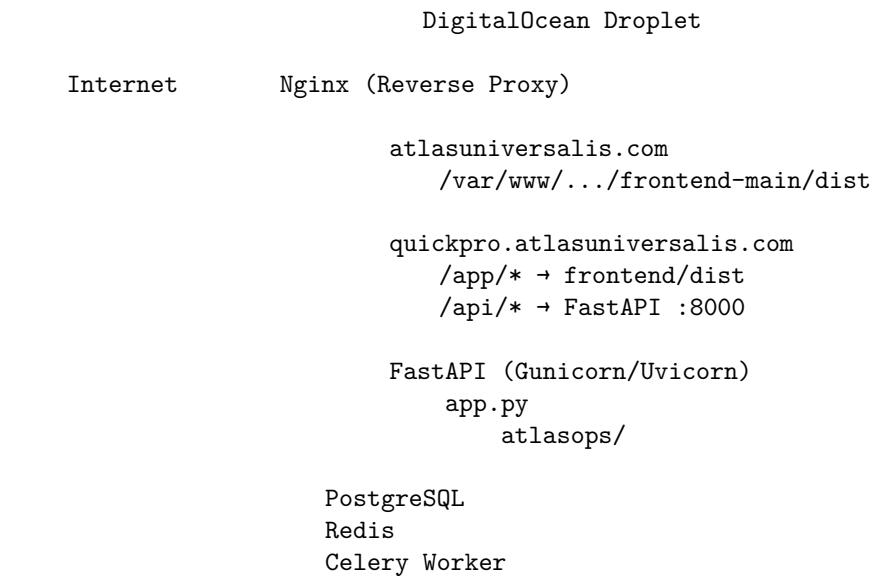
Directory	Contents
atlasops_wireframes/	UI wireframe mockups
Logo_Icon/	Logo design concepts and high-res assets
*.pdf	Project overview documents

MCP Server: mcp/

Model Context Protocol server for Cursor IDE integration.

File	Purpose
server.py	MCP server implementation
requirements.txt	Python dependencies
README.md	Setup instructions
cursor-mcp-config.example.json	Example Cursor configuration

Deployment Architecture



GitHub Actions deploys via SSH on push to master

Key Concepts Reference

Concept	Description
@concept:branding	Logo, colors, visual identity
@concept:homepage-hero	Main landing page hero section
@concept:email	Email configuration (Microsoft 365)
@concept:git-hygiene	Version control best practices
@concept:master-log	This documentation system