**Project Name:** opportunityknocks.ai - Wizard's Toolkit

**Version:** 1.1

**Date:** August 9, 2025

**Author:** Software Entrepreneur Agent for Jim Younkin

**## 1. Overview & Goals**

**1.1. Problem Statement**

Aspiring entrepreneurs spend dozens of hours on unstructured, manual research to find and validate new business ideas. This process is inefficient and a significant barrier to starting a new venture. 1

**1.2. Proposed Solution**

An internal, command-line toolkit that automates the discovery and synthesis of business ideas from YouTube and academic papers. 2 This tool will empower a "human wizard" (the founder) to rapidly generate a high-quality

**"Opportunity Dossier"**—a comprehensive research package complete with source attribution—for early customers.

**1.3. Strategic Goals**

* **Market Validation:** To validate customer demand for the Opportunity Dossier service before investing in a full-fledged SaaS platform. 3
* **Enable Manual Fulfillment:** To equip the founder with a tool that makes the "Wizard of Oz" fulfillment process efficient and repeatable. 4
* **IP Development:** To create the core data processing and AI synthesis logic that will serve as the foundation for the future automated product. 5

**## 2. Scope**

**2.1. In Scope for MVP ✅**

* A Python script executed from the command line (CLI). 6
* A module to fetch and analyze YouTube video transcripts. 7
* A module to fetch and analyze academic papers from arXiv.org. 8
* Integration with a Large Language Model for text synthesis. 9
* Configuration managed via

.env and config.py files. 10

* Output of

**enriched, structured JSON data**, including source metadata, to local files. 11

* Basic error handling for common issues. 12

**2.2. Out of Scope for MVP ❌**

* Any customer-facing UI (No Vite/React frontend). 13
* A web server or API (No FastAPI). 14
* User accounts, authentication, or databases. 15
* Deployment to a cloud environment. 16

**## 3. User Persona & Use Case**

* **Persona:** "The Wizard" (the Founder, Jim). A technically proficient operator who needs to fulfill customer research orders quickly and to a high standard. 17
* **Key Use Case:**
  1. The Wizard receives a customer order with a specified topic. 18
  2. He runs the toolkit script, passing the topic as an argument. 19
  3. The script generates two enriched JSON output files containing synthesized ideas and all supporting source data. 20
  4. The Wizard uses these files as the primary building blocks to write the final, polished PDF **Opportunity Dossier**, complete with a Source & Evidence Appendix, for the customer.

**## 4. Functional Requirements**

**FR-1: Core Application**

1.1. The application

**must** be a Python script run from the command line. 21

1.2. The application must manage all dependencies via Poetry and a pyproject.toml file.

1.3. The application

**must** load sensitive API keys from a .env file. 22

1.4. The application

**must** provide a command-line interface (e.g., using argparse) that accepts --topic and optional --source arguments. 23

**FR-2: YouTube Analysis Module**

2.1. The module

**must** use a library like youtube-transcript-api to fetch transcripts. 24

2.2. The module

**must** find the Top N (configurable) relevant videos for a topic. 25

2.3. The module

**must** aggregate the text from all retrieved transcripts. 26

2.4. The module

**must** pass the aggregated text to an LLM for synthesis, handling chunking as needed. 27

2.5. The module

**must** save its output as a structured JSON file that includes the synthesized ideas and a list of all source evidence, following the structure below. 28

**FR-3: arXiv Analysis Module**

3.1. The module

**must** use the arxiv library to search for papers and pypdf to extract text. 29

3.2. The module

**must** download the PDFs of the Top N (configurable) relevant papers. 30

3.3. The module

**must** extract the full text content from each downloaded PDF. 31

3.4. The module

**must** implement a two-step AI synthesis process (summarize then synthesize). 32

3.5. The module

**must** save its output as a structured JSON file that includes the synthesized ideas and a list of all source evidence, following the structure below. 33

**FR-4: Output Data Structure**

The JSON output for both modules **must** follow this enriched structure:

JSON

{

"search\_topic": "AI for preventative healthcare",

"synthesized\_opportunities": [

{

"idea": "AI-powered personalized nutrition plans based on health data.",

"description": "A service that uses AI to create weekly meal plans...",

"supporting\_evidence\_indices": [0, 2]

}

],

"source\_evidence": [

{

"index": 0,

"source\_type": "YouTube",

"title": "The Future of AI in Nutrition",

"author": "HealthTech Talks",

"url": "A direct URL to the video.",

"key\_quote": "AI can now analyze... to recommend specific dietary changes..."

},

{

"index": 1,

"source\_type": "arXiv",

"title": "A Novel Approach to AI-Driven Dietary Planning",

"author": "J. Doe et al.",

"url": "A direct URL to the paper's abstract page.",

"key\_quote": "Our model shows a 30% improvement in adherence..."

}

]

}

**## 5. Non-Functional Requirements**

* **NFR-1 (Configurability):** Key parameters (number of sources, LLM model name) **must** be easily editable in a config.py file. 34
* **NFR-2 (Error Handling):** The script **must not** crash on common errors and should log warnings. 35
* **NFR-3 (Security):** The script **must not** contain any hardcoded secrets. 36

**## 6. Data Flow**

CLI Command (Topic) -> [Modules] -> Fetch Raw Content -> Extract Text -> Chunk & Synthesize -> **Enriched JSON File with Sources** -> Wizard uses JSON to write Final PDF "Opportunity Dossier" with Source Appendix

**## 7. Success Criteria & Timeline**

* **Success Criteria:** The project is complete when the Wizard can run a single command and receive two well-structured, **enriched JSON output files containing source metadata**, enabling the efficient creation of a customer-ready Opportunity Dossier. 37
* **Timeline:** This PRD is to be implemented within the 1-Week "First Dollar" Sprint. 38