Project Plan for Lead Generation Using BabyAGI and Apache Nutch

July 7, 2025

Project Overview

This project aims to generate a CSV file of leads for marketing AI-powered software testing services (functional, performance, and load testing) using BabyAGI and Apache Nutch on a Windows 11 localhost. The tasks are atomized into small, specific actions to ensure modularity and scalability. Apache Nutch handles all web scraping tasks, replacing external APIs. The system avoids references to paid services or social platforms, ensuring compliance with data privacy laws (e.g., GDPR, CCPA).

1 Objectives

- Generate a CSV file with leads, including company name, contact name, email, industry, company size, interest level, and notes.
- Use BabyAGI to manage a task queue and Apache Nutch for web scraping.
- Ensure tasks are small, specific, and iterative for flexibility and error handling.

2 System Setup

- **BabyAGI**: Configured with a task queue, execution agent, and result storage. Objective: "Generate a CSV file of leads for AI-powered software testing services."
- **Apache Nutch**: Installed at `C:\nutch` on Windows 11, configured for web crawling with a custom user agent (`LeadGenBot`).
- **Environment**: Windows 11 localhost, Python 3.9+, asyncio for asynchronous task execution, and logging for debugging.

3 Data Schema

The CSV file (`leads.csv`) includes:

- **Company Name**: Name of the company.
- **Contact Name**: Name of the decision-maker.
- **Email**: Contact email address.
- **Industry**: Companys industry (e.g., Fintech, Healthcare).
- **Company Size**: Number of employees (e.g., 50-200).
- **Interest Level**: High, Medium, or Low based on AI adoption signals.

- **Notes**: Additional context (e.g., "Interested in AI-powered testing").

4 Atomized Task Breakdown

Each task is a small, specific function executed by BabyAGI, with Apache Nutch handling data collection:

1. **Identify Target Industries**: Research industries with high software testing needs (e.g., Fintech, Healthcare, E-commerce, Gaming, SaaS).
2. **Search for Companies in Industry**: Use Nutch to crawl Google for companies in the specified industry.
3. **Collect Company Details**: Crawl Google to extract company size and other details.
4. **Identify Decision-Maker**: Crawl company websites or Google for executive team members (e.g., CTO, QA Manager).
5. **Find Contact Email**: Crawl Google or company websites for decision-maker email addresses.
6. **Qualify Lead**: Analyze crawled data for AI adoption signals to assign interest level (High, Medium, Low).
7. **Compile Lead Data**: Aggregate data into a structured format for CSV.
8. **Write to CSV**: Append lead data to `leads.csv` with proper formatting.

5 Implementation Details

- **Python Script**: `lead_generator.py` uses BabyAGIs task queue and asyncio for asynchronous execution. Each task is a coroutine with error handling and retries (max 2).
- **Nutch Integration**: Nutch crawls Google with targeted queries (e.g., `industry + software companies`). Output is parsed for companies, contacts, emails, and sizes using regex.
- **Error Handling**: Tasks retry on failure (e.g., Nutch crawl errors). Failed tasks are logged for review.
- **Logging**: Comprehensive logging for task execution, errors, and results.

6 Execution Flow

1. Initialize `leads.csv` with headers.
2. Add initial task: "Identify target industries."
3. BabyAGI processes tasks from the queue, generating subtasks (e.g., company search after industries).
4. Nutch crawls Google or company websites for each task, with results parsed and stored.
5. Tasks continue until `MAX_TASKS` (150) is reached or queue is empty.

7 Output

The output is `leads.csv`, incrementally updated as leads are compiled. Each row follows the schema, with mock data replaced by Nutch-crawled data in production.

8 Testing and Validation

- Test each function independently (e.g., Nutch crawl, regex parsing).
- Validate email formats and company data accuracy.
- Monitor task queue for bottlenecks or infinite loops.

9 Scalability and Enhancements

- **Rate Limiting**: Limit Nutch crawls to avoid IP bans (e.g., sleep between tasks).
- **Prioritization**: Prioritize high-interest industries (e.g., Fintech) in the task queue.
- **Compliance**: Ensure crawled data complies with GDPR/CCPA (e.g., public data only).
- **Enhancements**: Improve regex for better parsing, add task prioritization based on lead quality.

10 Assumptions and Constraints

- **Assumptions**: BabyAGI and Apache Nutch are installed and configured on Windows 11. Nutch can access Google and company websites.
- **Constraints**: Limited to public web data. Mock data used for testing; production requires robust Nutch parsing.
- **Ethical Considerations**: Avoid private data scraping; ensure compliance with data privacy laws.

11 Next Steps

- Integrate advanced Nutch parsing for richer data (e.g., company revenue).
- Enhance lead qualification with keyword analysis for AI adoption.
- Implement task cleanup to remove temporary Nutch files.
- Test with real web data to validate scalability.