Data Ingestion

- Create a Python script to:
    - Read approximately ~5,000 text files
    - Parse key-value pairs from each file
    - Load data into MongoDB
- Note: Files may have varying numbers of key-value pairs
    - Example listing files: https://drive.google.com/drive/folders/1GvQZaGYZ2HV8M0qM3T5NLx1i-LMTpBjH?usp=sharing
    - Example python script: https://drive.google.com/file/d/1pcwUHUjRmkLW5XBLIMC6sp7bLxVvS6DD/view?usp=sharing

Database

- Use MongoDB hosted (MongoDB Atlas - free tier https://mongodb.com/)
- No specific requirements for data model or collection structure
- Store all available key-value pairs for each listing

REST API Development

- Implement a REST API with the following features:
    - Endpoint for querying listings
    - Simple API key authentication (exact match with a secret key)
    - Query parameters to include:
        - Address
        - Description
        - City
        - State
        - Price
        - Type of listing (commercial, land, residential, etc.)
        - Lot size
    - JSON response should include all available data for matching listings
- Create a Postman configuration for querying the API

API Functionality

- Read-only operations (no create, update, or delete functions required)
- No required parameters
- Implement result paging:
    - Add 'page' and 'limit' query parameters
    - 'page': specifies the page number (default to 1 if not provided)
    - 'limit': specifies the number of results per page (default to 10 if not provided)
    - Include metadata in the response about total results and current page

Example API Endpoint: GET /listings?city=Bedford&price_max=750000

- Implement header-based API key authentication
- Example: Headers: { "Authorization": "Bearer YOUR_API_KEY_HERE" }

Example API Endpoint: GET /listings?city=Poultney&price_max=750000&page=1&limit=20

Example JSON Response Structure:

```
{ "metadata":
        { "total_results": 100,
        "page": 1,
        "limit": 20,
        "total_pages": 5 },
        "results": [ // Array of up to 20 listing objects ]
}
```

Additional Considerations

- No specific performance requirements (POC stage)
- No specific framework requirements (beyond recommendation of MongoDB Atlas Free tier)
- No formal API documentation required
- No data validation or cleaning required during parsing

Testing

- Develop basic tests to ensure:
  - Successful data ingestion from text files to MongoDB
  - API returns correct results for various query combinations
  - Authentication works as expected