Code Strategy Document for LinkedIn Crawler and Analysis

Code Organization

The codebase follows an object-oriented approach to improve modularity, reusability, and clarity.

```
project-root/
│
├── src
│   ├── crawler
│   │   ├── linkedin_scraper.py
│   │   ├── queue_manager.py
│   │   ├── database_handler.py
│   │
│   │
│   └── analysis/
│       └── analysis_notebook.ipynb
│
├── Dockerfile
├── README.md
```

Key Components

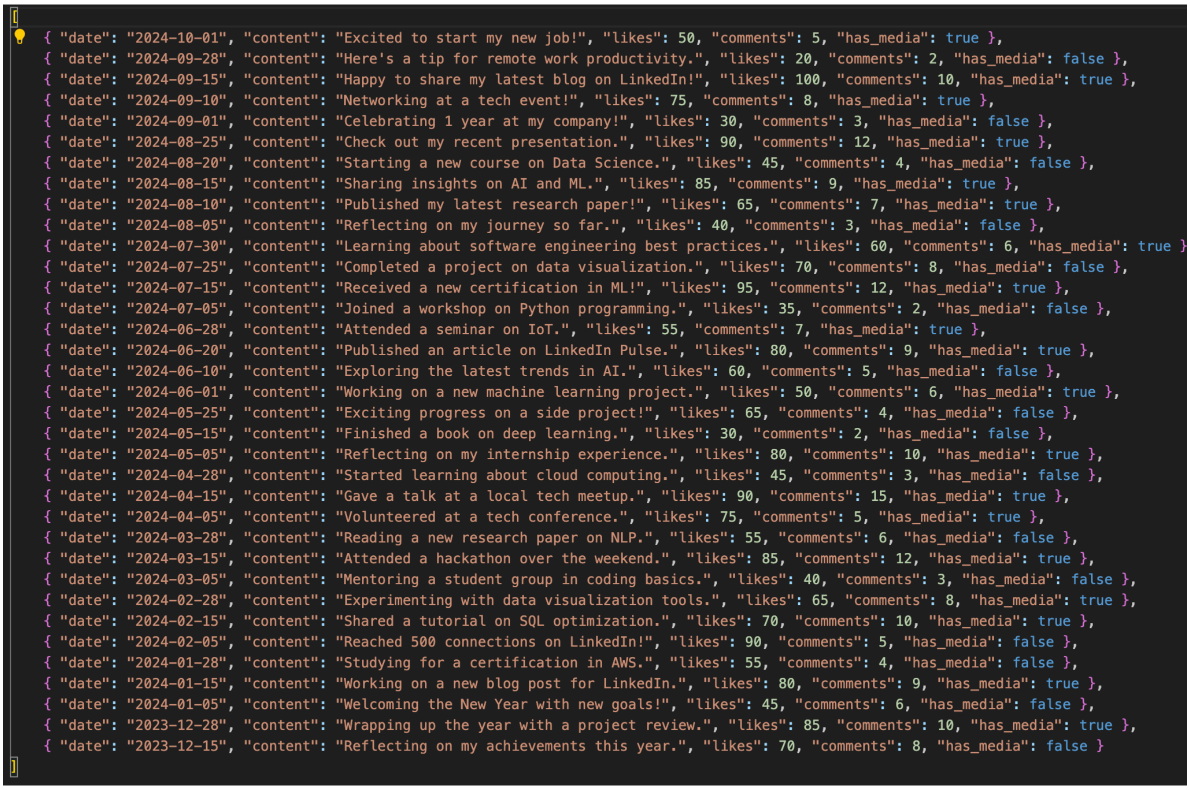
1. **LinkedInScraper** (`linkedin_scraper.py`): Manages scraping of profiles and posts, session handling, and data extraction.
2. **QueueManager** (`queue_manager.py`): Handles URL queueing and concurrent processing with Redis/Kafka.
3. **DatabaseHandler** (`database_handler.py`): Manages storing and retrieving data.
4. **Analysis Notebook**: Jupyter notebook for data analysis (post frequency, average likes/comments, etc.).

Setup and Deployment

1. **Setup**: Use `Poetry` for dependencies. Start Redis/Kafka with `docker-compose up`.
2. **Run Scraper**: Run `main.py` to initiate scraping.
3. **Analyze Data**: Open `analysis_notebook.ipynb` in Jupyter to run analytical tasks.

Database:

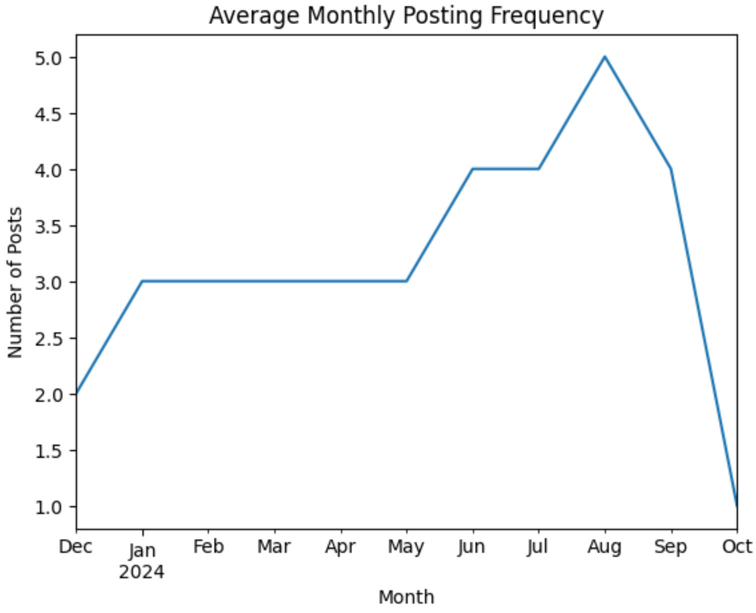
Date	content	likes	comments	Has_media



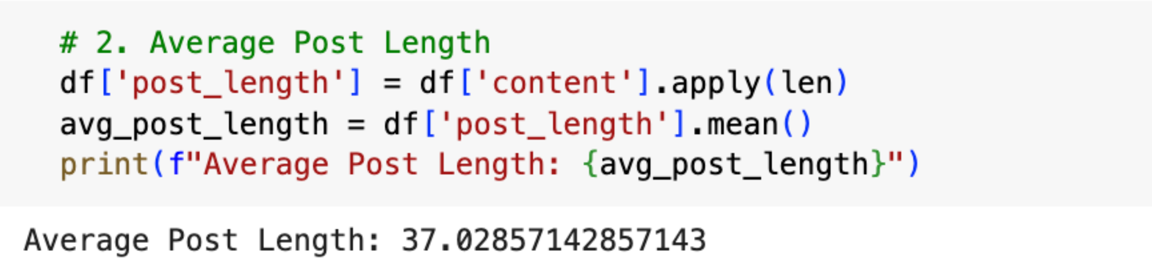
```
[
  { "date": "2024-10-01", "content": "Excited to start my new job!", "likes": 50, "comments": 5, "has_media": true },
  { "date": "2024-09-28", "content": "Here's a tip for remote work productivity.", "likes": 20, "comments": 2, "has_media": false },
  { "date": "2024-09-15", "content": "Happy to share my latest blog on LinkedIn!", "likes": 100, "comments": 10, "has_media": true },
  { "date": "2024-09-10", "content": "Networking at a tech event!", "likes": 75, "comments": 8, "has_media": true },
  { "date": "2024-09-01", "content": "Celebrating 1 year at my company!", "likes": 30, "comments": 3, "has_media": false },
  { "date": "2024-08-25", "content": "Check out my recent presentation.", "likes": 90, "comments": 12, "has_media": true },
  { "date": "2024-08-20", "content": "Starting a new course on Data Science.", "likes": 45, "comments": 4, "has_media": false },
  { "date": "2024-08-15", "content": "Sharing insights on AI and ML.", "likes": 85, "comments": 9, "has_media": true },
  { "date": "2024-08-10", "content": "Published my latest research paper!", "likes": 65, "comments": 7, "has_media": true },
  { "date": "2024-08-05", "content": "Reflecting on my journey so far.", "likes": 40, "comments": 3, "has_media": false },
  { "date": "2024-07-30", "content": "Learning about software engineering best practices.", "likes": 60, "comments": 6, "has_media": true }
  { "date": "2024-07-25", "content": "Completed a project on data visualization.", "likes": 70, "comments": 8, "has_media": false },
  { "date": "2024-07-15", "content": "Received a new certification in ML!", "likes": 95, "comments": 12, "has_media": true },
  { "date": "2024-07-05", "content": "Joined a workshop on Python programming.", "likes": 35, "comments": 2, "has_media": false },
  { "date": "2024-06-28", "content": "Attended a seminar on IoT.", "likes": 55, "comments": 7, "has_media": true },
  { "date": "2024-06-20", "content": "Published an article on LinkedIn Pulse.", "likes": 80, "comments": 9, "has_media": true },
  { "date": "2024-06-10", "content": "Exploring the latest trends in AI.", "likes": 60, "comments": 5, "has_media": false },
  { "date": "2024-06-01", "content": "Working on a new machine learning project.", "likes": 50, "comments": 6, "has_media": true },
  { "date": "2024-05-25", "content": "Exciting progress on a side project!", "likes": 65, "comments": 4, "has_media": false },
  { "date": "2024-05-15", "content": "Finished a book on deep learning.", "likes": 30, "comments": 2, "has_media": false },
  { "date": "2024-05-05", "content": "Reflecting on my internship experience.", "likes": 80, "comments": 10, "has_media": true },
  { "date": "2024-04-28", "content": "Started learning about cloud computing.", "likes": 45, "comments": 3, "has_media": false },
  { "date": "2024-04-15", "content": "Gave a talk at a local tech meetup.", "likes": 90, "comments": 15, "has_media": true },
  { "date": "2024-04-05", "content": "Volunteered at a tech conference.", "likes": 75, "comments": 5, "has_media": true },
  { "date": "2024-03-28", "content": "Reading a new research paper on NLP.", "likes": 55, "comments": 6, "has_media": false },
  { "date": "2024-03-15", "content": "Attended a hackathon over the weekend.", "likes": 85, "comments": 12, "has_media": true },
  { "date": "2024-03-05", "content": "Mentoring a student group in coding basics.", "likes": 40, "comments": 3, "has_media": false },
  { "date": "2024-02-28", "content": "Experimenting with data visualization tools.", "likes": 65, "comments": 8, "has_media": true },
  { "date": "2024-02-15", "content": "Shared a tutorial on SQL optimization.", "likes": 70, "comments": 10, "has_media": true },
  { "date": "2024-02-05", "content": "Reached 500 connections on LinkedIn!", "likes": 90, "comments": 5, "has_media": false },
  { "date": "2024-01-28", "content": "Studying for a certification in AWS.", "likes": 55, "comments": 4, "has_media": false },
  { "date": "2024-01-15", "content": "Working on a new blog post for LinkedIn.", "likes": 80, "comments": 9, "has_media": true },
  { "date": "2024-01-05", "content": "Welcoming the New Year with new goals!", "likes": 45, "comments": 6, "has_media": false },
  { "date": "2023-12-28", "content": "Wrapping up the year with a project review.", "likes": 85, "comments": 10, "has_media": true },
  { "date": "2023-12-15", "content": "Reflecting on my achievements this year.", "likes": 70, "comments": 8, "has_media": false }
]
```

Metrics

1. **Average Monthly Posting Frequency**:
   - Counts posts per month and averages them.



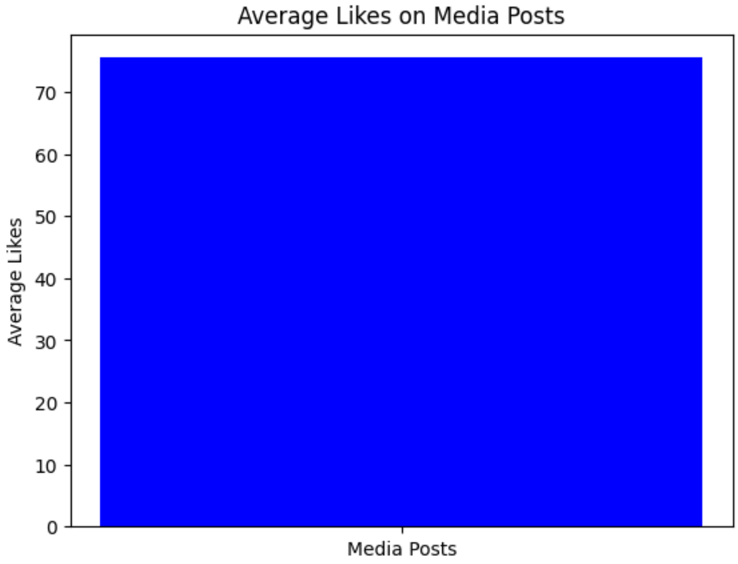
2. **Average Post Length**:
   - Calculates the average word count for each post.

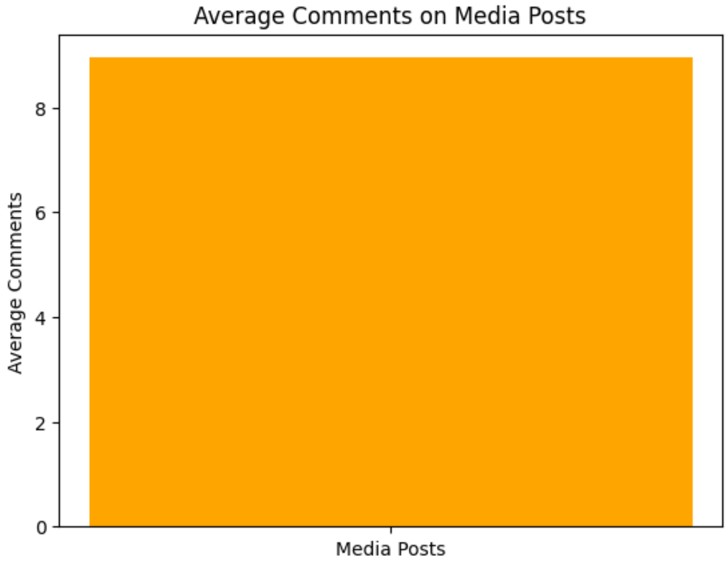


```
# 2. Average Post Length
df['post_length'] = df['content'].apply(len)
avg_post_length = df['post_length'].mean()
print(f"Average Post Length: {avg_post_length}")
```

```
Average Post Length: 37.02857142857143
```

3. **Average Likes and Comments on Media Posts**:
   - Filters posts containing media and computes average likes and comments, visualized with bar graphs.





4. **Engagement Rate**:
   - Calculates total engagement (likes + comments) per post as a percentage of all posts.

```python
# 5. Engagement Rate (Likes + Comments per Post)
df['engagement'] = df['likes'] + df['comments']
avg_engagement = df['engagement'].mean()
print(f"Average Engagement Rate per Post: {avg_engagement}")
```

```
Average Engagement Rate per Post: 70.51428571428572
```

```python
analysis_summary = {
    "Average Monthly Posting Frequency": monthly_frequency.mean(),
    "Average Post Length": avg_post_length,
    "Average Likes on Media Posts": media_likes_avg,
    "Average Comments on Media Posts": media_comments_avg,
    "Average Engagement Rate per Post": avg_engagement,
}
```

```python
# Write summary to file
with open("analysis_summary.txt", "w") as f:
    for key, value in analysis_summary.items():
        f.write(f"{key}: {value}\n")
print("\nAnalysis summary saved to analysis_summary.txt")
```

```

Analysis summary saved to analysis_summary.txt
```