Zoomprop Senior Python Engineer take home assessment.

Instructions:

- Please document your thought process and any assumptions you make.
- Provide your code in a well-organized repository (e.g., GitHub) with a README file explaining how to run your solution.

Problem Statement

You are tasked with building a mini-application that processes and analyzes real estate data. The application should be able to:

1. **Ingest Real Estate Data:**
    - Create a script to ingest real estate property data from a CSV file into a SQLite database.
    - The CSV file will contain columns like `PropertyID`, `Address`, `City`, `State`, `ZipCode`, `Price`, `Bedrooms`, `Bathrooms`, `SquareFeet`, and `DateListed`.
2. **Data Processing and Analysis:**
    - Implement a function to calculate basic statistics from the data, such as:
        - Average property price
        - Median property price
        - Average price per square foot
        - Total number of properties listed
    - Implement a function to filter properties based on criteria such as price range, number of bedrooms, number of bathrooms, and city.
3. **API Endpoint:**
    - Develop a RESTful API using FastApi with the following endpoints:
        - `GET /properties` - Retrieve a list of properties with optional query parameters for filtering (price range, bedrooms, bathrooms, city).
        - `GET /properties/statistics` - Retrieve the basic statistics calculated from the data.
4. **Data Visualization:**
    - Create a simple visualization using a library like Matplotlib or Plotly to display:
        - Distribution of property prices
        - Distribution of properties by number of bedrooms

Requirements

1. **Code Quality:**
    - Write clean, maintainable, and well-documented code.
    - Follow Python best practices and PEP 8 guidelines.
2. **Testing:**
    - Write unit tests for your data processing and API functions.
    - Ensure that your code is testable and include instructions on how to run the tests.

3. **Documentation:**
    - Provide a README file that includes:
        - An overview of your solution.
        - Instructions on how to set up and run the application.
        - Example requests for the API endpoints.
        - Explanation of your data visualization.

Bonus Points

1. **Advanced Data Processing:**
    - Implement additional data processing features such as detecting outliers in property prices.
    - Provide insights or trends based on the historical data provided in the `DateListed` column.
2. **Security:**
    - Implement basic security measures for your API endpoints, such as input validation and sanitization.
3. **Deployment:**
    - Deploy your application to a cloud platform (e.g., Heroku, AWS) and provide the live URL in your README file.

Deliverables

- Link to the GitHub repository containing your solution.
- Detailed README file with setup and usage instructions.
- Any additional documentation or notes explaining your approach and any assumptions made.

We look forward to reviewing your submission!