

# Backend Software Engineer Take Home Assignment

---

## Feature list

---

- **Address Search:** Users can easily search for specific addresses by typing in the search box. The website will display a list of matching records related to the entered address.
- **UNIX Timestamp Search:** Users can search for records opened before a specific UNIX timestamp by entering the timestamp in the search box. The website will show a list of matching records based on the search criteria.
- **Reset Functionality:** Users can reset the search criteria by simply clicking on the reset button. This will clear the search box and display all records.
- **Record Count:** The website provides a visual representation of the total number of records available. Users can quickly see the count to get an idea of the data volume.

## Technology Choices

---

I have chosen Django, PostgreSQL, and Redis for the following reasons:

- **Django:** Django is a high-level web framework that simplifies development, with built-in features like ORM and authentication. It accelerates development while ensuring code quality and maintainability.
- **PostgreSQL:** PostgreSQL is an open-source and reliable database management system. It offers advanced features, scalability, and data integrity for efficient storage and management of structured data.

- **Redis:** Redis is an in-memory data store that serves as a cache, providing fast data retrieval. It significantly improves response time, enhancing user experience by caching frequently accessed data.

## Design Decision

---

The choice of a 5-minute cache time strikes a balance between information freshness and user experience. With this duration, users can expect to find the information they are seeking within a reasonable timeframe. If the market data undergoes frequent changes, a shorter cache time may be necessary to ensure up-to-date information availability. By setting the cache time to 5 minutes, we optimize accessibility for a larger user base. Users can enjoy faster response times, enabling them to retrieve the desired information promptly and efficiently. This approach enhances the overall user experience, allowing individuals to interact with the service smoothly and access the data they need with minimal delay.

## Security

---

- **Input Filtering:** Implement input filtering to prevent the execution of malicious code or injection of harmful scripts through user input.
- **Error Handling:** Implement proper error handling and error messages to prevent the exposure of sensitive information. Avoid displaying detailed error messages to end users, as they can provide potential attackers with valuable information.
- **Source Code Security:** Storing sensitive information like passwords and tokens in a separate .env file enhances security by preventing accidental exposure.
- **CSRF tokens:** Make sure the request is made by same

website so that it prevents malicious websites from tricking users into submitting unauthorized requests on their behalf.