Backend Developer Exercise

A URL shortener redirects short URLs to long URLs and keeps track of the number of redirects for each URL. For example, A short URL “https://short.url/jdj23d” can redirect to the long URL https://ravkavonline.co.il/he/faq#ravkav-online

URL shorteners are mostly used in the following scenarios:

1. Using shorter texts in constrained platforms such as SMS messages and Tweets
2. Keeping track of the numbers of clicks. It’s a common practice to use URL shorteners to keep track of clicks in campaigns and other types of promotions

For this exercise you'll implement a simple URL shortener in Django

Instructions

- Implement the exercise using Django and avoid additional dependencies
- Use either SQLite or PostgreSQL as a database backend
- The project should include at least a model and a view (models.py & views.py)
- A **short URL is unique**, and different short URLs can reference the same full URL. Pay attention to how you generate the short URL. What can go wrong? How do you plan to handle it?
- **Keep a hit counter** for every short URL. Pay attention to how you increment the counter.
- **Include unit tests**: The project should include tests for creating and redirecting a short URL and a test for non-existing short URL. If you think additional tests are necessary include them as well.
- **Include meaningful comments** in appropriate places to help us understand your thinking process.
- No need to implement any type of authentication. For the purpose of this exercise you can allow anonymous access to all.
- To create a new short URL provide the following API:

```
$ curl -X POST "http://localhost:8000/create" \
       -H "Content-Type: application/json" \
       -d '{"url": "https://ravkavonline.co.il"}'

http://localhost:8000/s/slKj289
```

In this example, `http://localhost:8000/s/slKj289` will redirect to the full URL

⚠️ **Please be critical of code you find online**
There are several projects posted to popular sites with examples of URL shorteners implemented using Django. Most of them are incorrect in many ways.

Submitting the Exercise

Submit the exercise as git repo or a ZIP containing all the necessary files. Please include a requirements.txt file listing the dependencies and their versions, and make sure to not include the virtual environment directory, if used.

No need to deploy this to a server, it will be tested locally.

You can take all the time you need to complete the exercise. However, your time is valuable so we designed the exercise to not take more than a couple of hours