Typetone engineering assignment

Create a webservice in python which can shorten urls like TinyURL and bit.ly.

Deliverables

- Source code
- Simple way to setup a virtual environment with the necessary python libraries
- Instructions to start the application
- Instructions to run unit test(s)

Tips

- Keep the code clean and simple.
- Make sure your unit test(s) is/are only testing your own code.
- **Important:** We will use the delivered assignment to see how you would build a scalable product. Whilst the service of this assignment is relatively small, we would like to see a structured approach as if it were to scale into something bigger.

Application requirements

- Written in Python
- Use FastAPI as the web framework of the application
- As datastore you could use PostgresSql
- We should be able to execute it without crashing on setup (surprise).
- The service should be exposing the following endpoints:

1. **POST /shorten**

   The **request body** should have the following content:

```
{
 "url": "https://www.example.com/",
 "shortcode": "abn123"
}
```

   When no shortcode is provided it should create a random shortcode for the provided url. The random shortcode has a length of 6 characters and should contain only alphanumeric characters or underscores.

   If a shortcode is given by the user these restrictions do not apply.

   Returns HTTP status 201 with the following body:

```
{
 "shortcode": "abn123",
 "update_id: <update_id>
}
```

   Where ***<update_id>*** contains a unique identifier to update the url for the given shortcode.

   Errors:

400	Url not present
409	Shortcode already in use
412	The provided shortcode/url is invalid

2. **POST /update/<update_id>**

   The **request body** should have the following content:

```
{
 "url": "https://www.example.com/"
}
```

   Sending this request will update the url for the shortcode related to this **<update_id>.**

   Returns HTTP status 201 with the following body:

```
{
 "shortcode": "abn123"
}
```

   Errors:

400	Url not present
401	The provided update ID does not exist
412	The provided url is invalid

3. **GET /<shortcode>**

   Returns http status 302 with the Location header containing the url

   Errors:

404	Shortcode not found

4. **GET /<shortcode>/stats**

   Returns http status 200 with the following body:

```
{
 "created": "2017-05-10T20:45:00.000Z",
 "lastRedirect": "2018-05-16T10:16:24.666Z",
 "redirectCount": 6
}
```

   <created> contains the creation datetime of the shortcode (in ISO8601)
   <lastRedirect> contains the datetime of the last usage of the shortcode (in ISO8601)
   <redirectCount> indicates the number of times the shortcode has been used

   Errors:

404	Shortcode not found