Nortal Home Assignment

To be honest the home assignment had a lot of terms I was not familiar with. I had never created an API and didn’t really know what RESTful APIs are. The second problem was that I had to work part-time while studying the subject and creating the API.

For the first days I spent my time studying the subject and planning what technologies to use.  
I decided to work with Python, because it is the programming language that I’m most familiar with and I had limited time. I was considering whether to use Flask or Django as the framework. In the end I decided to use Django, because I believe handling it will be more beneficial in the future.

Django had a great documentation and plenty of tutorials to start from. After getting the grasp of Django I started to apply my new knowledge onto my own API. I decided to build an API that allowed the users to create accounts and leave feedback on other users’ accounts.

At first I created the base API and a web interface which allowed the user to sign up and leave feedback for other users. After that I began building the endpoints to make it RESTful. This was the hardest part as the whole RESTful concept was new to me. At the same time I also started to learn the basics of Postman to test the REST functionality. For SQL database I decided to stick with SQLite, which is the default SQL database system in Django. I am aware that SQLite is too lightweight to use in any bigger project, but due to the time limit I did not have time to change to PostgreSQL. The API uses session authentication.

After three days of hard work, sweat and googling I finally had all the functionality working. All in all it was a really pleasant journey. I love learning new things and this new knowledge will help me in the future for sure.

How to:

1. Install python 3 and Pip.
2. Run “pip install django” on CMD to install Django.
3. Run “pip install djangorestframework” on CMD to install Django Rest Framework.
4. Run “pip install drf\_yasg” on CMD to install Yet another Swagger generator.
5. In project folder run “python manage.py runserver” in CMD to start the server.
6. Enter the provided url in your browser of choice. <http://127.0.0.1:8000/>. The web interface can be used here.
7. I’ve included a premade admin account “admin” with password “admin”.
8. <http://127.0.0.1:8000/api/> has all the endpoints listed as drf\_yasg generated Swagger documentation. The POST endpoints require the parameters in json format.
   * **/delete\_user/**  
     Requires “id” of the user you want to delete. Requires you to be logged in as an admin.
   * **/feedbacks/**  
     Returns all feedbacks.
   * **/feedbacks/{id}**  
     Returns a feedback.
   * **/login/**  
     Log in as user. Requires “username” and “password”.
   * **/logout/**  
     Log out from user.
   * **/new\_feedback/**  
     Requires “body” and “owner” (owner as id). Requires you to be logged in.
   * **/signup/**  
     Requires “username”, “password1” and “password2”. Creates new user.
   * **/users/**  
     Returns all users and their ids.
   * **/users/{id}/**  
     Returns a user.
   * **/users/{id}/feedbacks/**  
     Returns all feedbacks of a user.

Example:

<http://127.0.0.1:8000/signup/>  
**POST {“username”: “TestUser”, “password1”: “password”, “password2”: “password"}**  
Creates user “TestUser” with password “password” and automatically logs in.

<http://127.0.0.1:8000/users/>  
**GET**Returns a json list of all users and their IDs.

<http://127.0.0.1:8000/new_feedback/>  
**POST {“body”: “Test feedback”, “owner”: 3}**Creates new feedback “Test feedback” for user with ID 3. Requires you to be logged in. Feedback author is automatically set as the logged in user.