**Extra Feature**: Website API with Django-Rest-Framework

Uses: pip install djangorestframework, pip install markdown (Markdown support for the browsable API) and pip install django-filter (Filtering support).

Articles, comments and users have their own API app. This was done to have a cleaner file structure. Each API app has functionality for the app it is located inside. All views have a default IsAuthenticated permissions class. Some of the views have a custom IsOwnerOrReadOnly permission that restricts functionality based on if the current logged in user is the owner of the object. Some of the views have an overriding AllowAll permission class

**Article API**:

* List API view – list of all the articles in the app with some details. 127.0.0.1:8000/api/articles/ Has a get\_queryset function that lets you filter results. Example =127.0.0.1:8000/api/articles/?search=shini&q=hi&ordering=-user
* Detail API view – more details of an article 127.0.0.1:8000/api/articles/1/
* Delete article view – delete an article 127.0.0.1:8000/api/articles/1/delete/
* Update article view – update an article 127.0.0.1:8000/api/articles/1/edit/
* Create article view – create a new article 127.0.0.1:8000/api/articles/create/

**Comments API**:

* List API view – list of all the comment in the app with some details. 127.0.0.1:8000/api/comments/. Has a get\_queryset function that lets you filter results.
* Create API view – lets you create a comment. 127.0.0.1:8000/api/comments/create/
* Details API view – lets you view extra details of the comment and allows you to edit and delete if you have permissions. 127.0.0.1:8000/api/comments/1/

**User API**:

* Register API view – create a new user 127.0.0.1:8000/api/users/register/
* Login API view – login with a verified user 127.0.0.1:8000/api/users/login/
* List API view – list all the users with some information. 127.0.0.1:8000/api/users/

JSONWebTokenAuthentication functionality uses: pip install djangorestframework-jwt

Curl Tests: curl -X POST -d [email=belal@belal.com&password=belal](mailto:email=belal@belal.com&password=belal) <http://127.0.0.1:8000/api/auth/token/> Returned token : eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImJlbGFsQGJlbGFsLmNvbSIsImV4cCI6MTUxMzA4MTMwNCwiZW1haWwiOiJiZWxhbEBiZWxhbC5jb20iLCJ1c2VyX2lkIjoxNH0.hhl1u1DcF7p4tPoCKFs3FfywB7MyunB8OF-7t9y7mm4

curl http://127.0.0.1:8000/api/comments/ #tested with permissions AllowAll off. Did not compile

curl -H "Authorization: JWT eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImJlbGFsQGJlbGFsLmNvbSIsImV4cCI6MTUxMzA4MTMwNCwiZW1haWwiOiJiZWxhbEBiZWxhbC5jb20iLCJ1c2VyX2lkIjoxNH0.hhl1u1DcF7p4tPoCKFs3FfywB7MyunB8OF-7t9y7mm4" <http://127.0.0.1:8000/api/comments/> #complied

Current settings have the 'rest\_framework.authentication.SessionAuthentication' and 'rest\_framework.renderers.BrowsableAPIRenderer' turned on for testing purposes

Superuser: [shini@shini.com](mailto:shini@shini.com) password: blueblue

Open access link: <http://thenightlystormer-badal.apps.devcloud.eecs.qmul.ac.uk/>