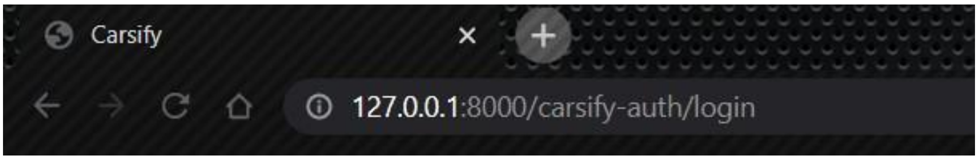
Django-Custom-Authentication

A complete custom user authentication with email verification.

Created a custom authentication module by overriding the **AbstractBaseUser** and **BaseUserManager** model, that runs on the base URL of **<base>/carsify-auth/**

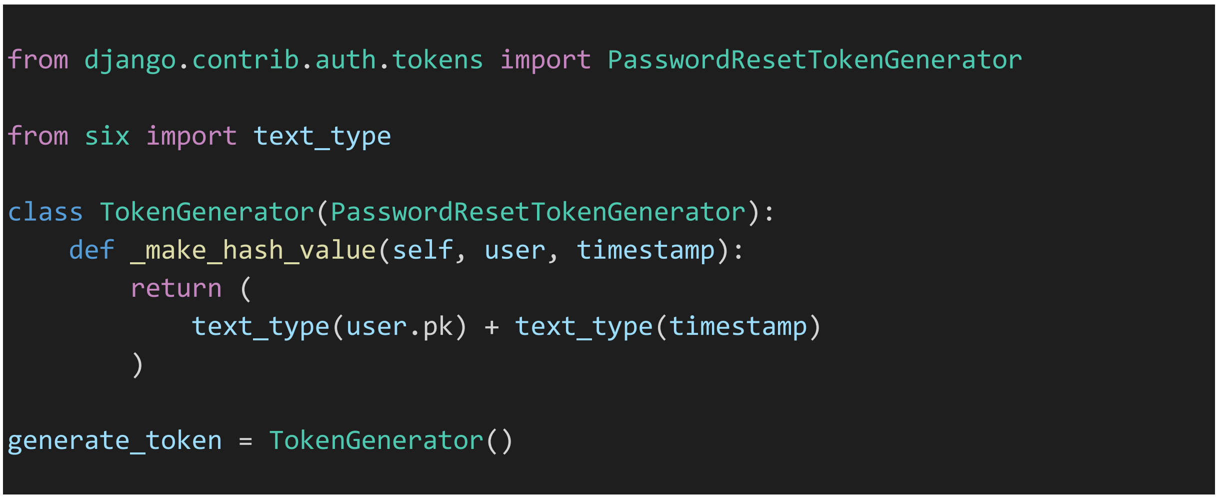


127.0.0.1:8000/carsify-auth/login

Email Verification:

Provided email verification during signing up in which the user receives an email containing the activation link, which is generated by an unique Id and token generator.

The **TokenGenerator** class is developed by overriding the **PasswordResetTokenGenerator** ,



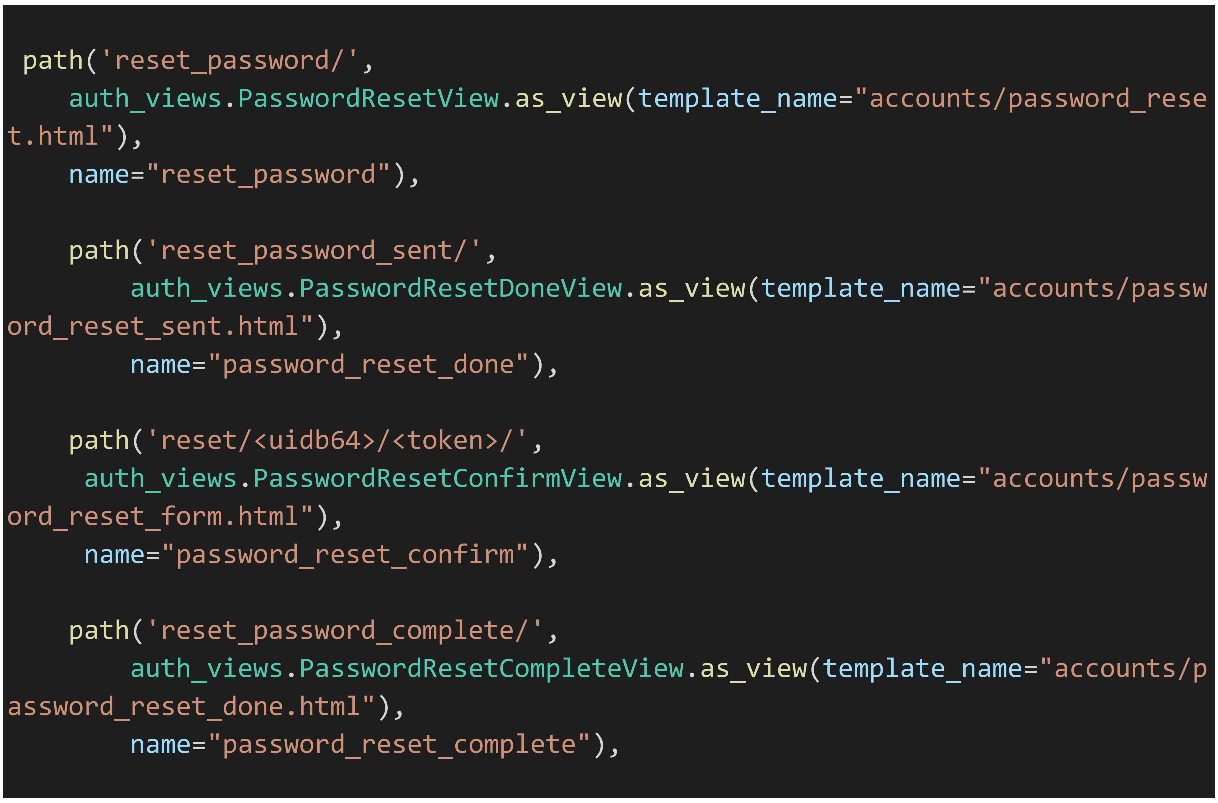
```
from django.contrib.auth.tokens import PasswordResetTokenGenerator

from six import text_type

class TokenGenerator(PasswordResetTokenGenerator):
    def _make_hash_value(self, user, timestamp):
        return (
            text_type(user.pk) + text_type(timestamp)
        )

generate_token = TokenGenerator()
```

install six by the command: **pip install six**

Forget Password:

Provided Forgot password feature, in which the user receives a password reset link in their email to reset the password.

Used Django built-in authentication views for resetting password and provided custom templates in the urls.py:



```
 path('reset_password/',
    auth_views.PasswordResetView.as_view(template_name="accounts/password_reset.html"),
    name="reset_password"),

    path('reset_password_sent/',
        auth_views.PasswordResetDoneView.as_view(template_name="accounts/password_reset_sent.html"),
        name="password_reset_done"),

    path('reset/<uidb64>/<token>/',
     auth_views.PasswordResetConfirmView.as_view(template_name="accounts/password_reset_form.html"),
     name="password_reset_confirm"),

    path('reset_password_complete/',
        auth_views.PasswordResetCompleteView.as_view(template_name="accounts/password_reset_done.html"),
        name="password_reset_complete"),
```

Profile View:

The logged in users have access to view their profile page and logout their session. The redirect url for logging out is the base url.

Provided redirect next url in case the user views their profile without logging in, in that case the user will be redirected to the login page with next argument in the url.

This Django project contains two apps, ‘accounts’ for the whole authentication and authorizations purposes which can be managed in the admin site and ‘home’ for the user’s profile, dashboard and the basic home of the website.

Project Links:

Website: http://vimal11.pythonanywhere.com/

Github: https://github.com/vimal-11/django-custom-authentication