OUTLINE

This document outlines the development of a Django web application with integrated OTP authentication. The purpose was to enhance security in a Django application by implementing OTP functionality using the django-otp package.

Task-

```
# Django Web App + OTP

Install Python3 in case it is not installed on your machine. Type python or python3 in your terminal of choice. You should see the Python interpreter
show up. Next, we will want environment isolation so this project can be separate from the rest of your machine. You can do one of the following:

```
- Install Anaconda / Miniconda. Once installed
    - conda create --name forageenv python=3.9
    - conda activate forageenv
- Use venv
    - python3 -m venv forageenv
    - source forageenv/bin/activate
- Use another tool you are comfortable with
    - use it as your normally would
```

Let's get started. First unzip the mysite.zip file. For a detailed discussion, please follow [this]
(https://docs.djangoproject.com/en/3.2/intro/tutorial01/).
```
cd mysite

# install required Python modules
pip3 install -r requirements.txt

# synchronizes the database to the existing configuration and creates it if it does not exist
python3 manage.py migrate

# creates the site admin user
python3 manage.py createsuperuser
# enter username, email, password of your choice

# runs the web application
python3 manage.py runserver
```

Follow the Django OTP installation docs https://django-otp-official.readthedocs.io/en/stable/overview.html#installation and modify `settings.py`. We
will be using the `otp_totp` plugin. Hint: you will need to modify two variables.
```
# stop the running webserver with Ctrl-C
# modify settings.py
python3 manage.py migrate
python3 manage.py runserver
```

Modify urls.py. Simply add the following after the existing import statements.
```
from django_otp.admin import OTPAdminSite
admin.site.__class__ = OTPAdminSite
```

Navigate to localhost:8000/admin/ and add a device. Then click on QRcode and copy the link so answer the final quiz question. Replace the `secret` value
with `<SECRET>`.
```

Finished product-
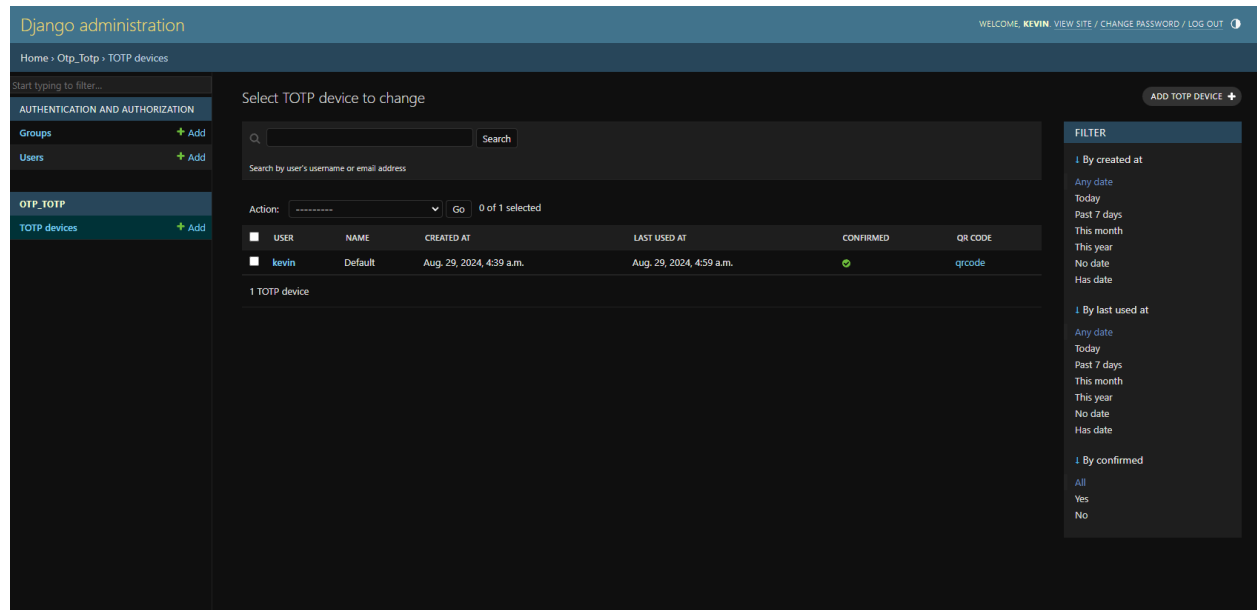
Conclusion -

The Django web application now includes OTP authentication, which enhances security. The integration was successfully tested, and the admin interface supports managing OTP devices.