Project 2

Start with Docker

Manual Build

Download the code

```
$ git clone https://github.com/app-generator/django-datta-able.git
$ cd django-datta-able
```

☞ Set Up for Unix, MacOS

Install modules via VENV

```
$ virtualenv env
$ source env/bin/activate
$ pip3 install -r requirements.txt
```

Set Up Database

```
$ python manage.py makemigrations
$ python manage.py migrate
```

Generate API

```
$ python manage.py generate-api -f
```

Start the APP

```
$ python manage.py createsuperuser # create the admin
$ python manage.py runserver        # start the project
```

At this point, the app runs at http://127.0.0.1:8000/.

☞ Set Up for Windows

> Install modules via VENV (windows)

```
$ virtualenv env
$ .\env\Scripts\activate
$ pip3 install -r requirements.txt
```

> Set Up Database

```
$ python manage.py makemigrations
$ python manage.py migrate
```

> Start the APP

```
$ python manage.py createsuperuser # create the admin
$ python manage.py runserver       # start the project
```

At this point, the app runs at http://127.0.0.1:8000/.

Codebase Structure

The project is coded using a simple and intuitive structure presented below:

```
< PROJECT ROOT >
   |
   |-- core/
   |    |-- settings.py                # Project Configuration
   |    |-- urls.py                     # Project Routing
   |
   |-- home/
   |    |-- views.py                    # APP Views
   |    |-- urls.py                     # APP Routing
```

```
    |      |-- models.py                  # APP Models
    |      |-- tests.py                    # Tests
    |      |-- templates/                  # Theme Customisation
    |          |-- pages                   #
    |              |-- custom-index.py     # Custom Dashboard
    |
    |-- requirements.txt                   # Project Dependencies
    |
    |-- env.sample                         # ENV Configuration (default
 values)
    |-- manage.py                          # Start the app - Django
 default start script
    |
    |--
 ***********************************************************************
```

How to Customize

When a template file is loaded in the controller, Django scans all template directories starting from the ones defined by the user, and returns the first match or an error in case the template is not found.
The theme used to style this starter provides the following files:

```
# This exists in ENV: LIB/admin_datta
< UI_LIBRARY_ROOT >
   |
   |-- templates/                     # Root Templates Folder
   |     |
   |     |-- accounts/
   |     |    |-- auth-signin.html    # Sign IN Page
   |     |    |-- auth-signup.html    # Sign UP Page
   |     |    |
   |     |-- includes/
   |     |    |-- footer.html         # Footer component
   |     |    |-- sidebar.html        # Sidebar component
   |     |    |-- navigation.html     # Navigation Bar
   |     |    |-- scripts.html        # Scripts Component
   |     |    |
   |     |-- layouts/
   |     |    |-- base.html           # Masterpage
   |     |    |-- base-auth.html      # Masterpage for Auth Pages
   |     |    |
   |     |-- pages/
   |          |-- index.html          # Dashboard Page
   |          |-- profile.html        # Profile Page
   |          |-- *.html              # All other pages
   |
   |--
 ***********************************************************************
```

Sending requests

Since the website has a login, you can add a user under Authentication, this can be done manually from the application. However, to send request in python,
you will need two cookies: csrftoken and sessionid. After loggin with your credentials, you can open the developer console (F12) and copy this two cookies from the
Storage tab. A example_request.py script is provided, you can retrieve this .

```
headers = {
    'Cookie': 'csrftoken=nonj00WFbEN8EFpvvVLeMyWBlRpZM7xJ;
sessionid=05lc07de5r8viashy813v6tixw3g57py'
}
```