|  |  |  |
| --- | --- | --- |
| **Chapter**  **No.** | **Title** | **Page**  **No.** |
| 1. | Database Schema | 2 |
| 2 | Virtual Environment | 3 |
| 3. | Hockey Project | 3 |
| 4. | App (Six A Side) | 4 |
| 5. | Models | 4 |
| 6. | Static and Templates | 5-6 |
| 7. | Views | 6-7 |
| 8. | Frontend | 8-9 |

**Schema for database for Indoor Hockey Association**

Following are settings for Database configuration in the setting.py file:

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.mysql',

        'NAME': 'dusty123$hockey',

        'USER': 'dusty123',

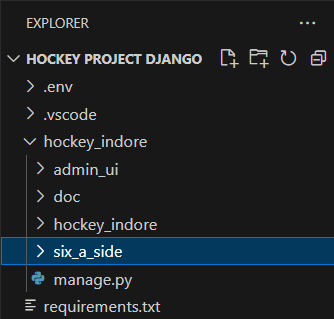
        'PASSWORD': 'dastgir@123',

        'HOST':'dusty123.mysql.pythonanywhere-services.com',

        'PORT':'3306',

    }

}

;

The Whole project is divided into TWO Major Directories: .env and Hockey\_indore

1. **.env** file consist of the required libraries for working of project. It is the virtual environment for the project and should be activated before using or debugging the apps inside the project.

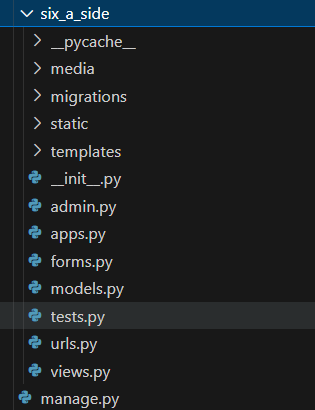
* Cd into project main route and enter .env/Scripts/activate to activate the environment.



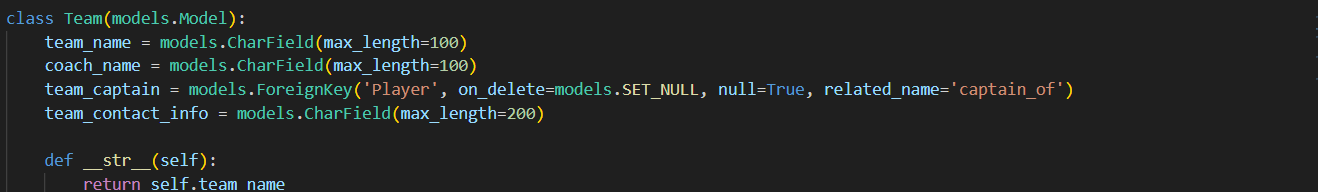
* In case .env file does not exist then create virtual environment using any virtual package you want for eg: pip install venv, python -m venv (environment name). write this in the command prompt. Then install the **requirements.txt** file. pip install -r reqirements.txt.

1. When successfully installed and activated cd into project’s directory cd **hockey\_indore** then use manage.py as a build in uitility for further operations.

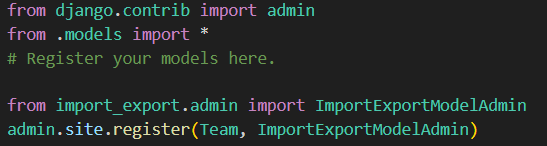




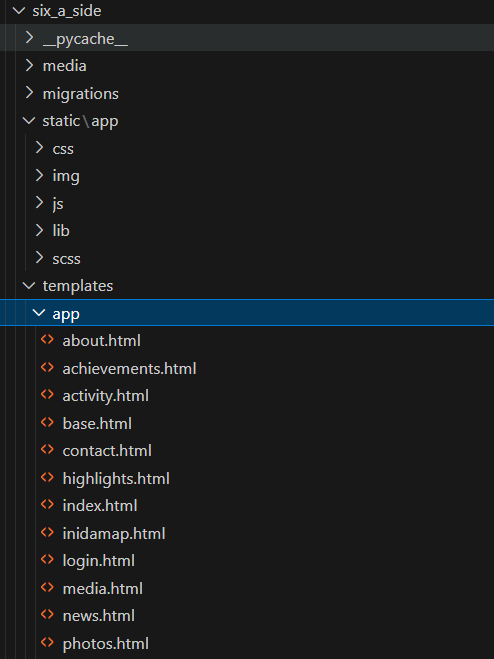
1. Models.py is the Orm for our project’s app Six\_a\_Side.
2. 



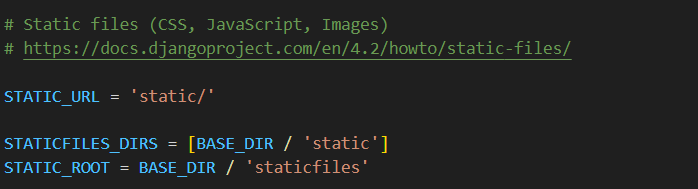
* To create a table just make a class and inherit it from models.Model class to make rows and constrains just create attribute to the class and use the build in Django.db.models’ attribute. Read Django’s documentations for detailed note: <https://docs.djangoproject.com/en/4.2/topics/db/models/>
* To create table in your database run these two commands in your command prompt: **python manage.py makemigrations, Python manage.py migrate.** To create all the tables in your database as defined by the classes in your models.py
* To register your model in the project write the following in admin.py file:



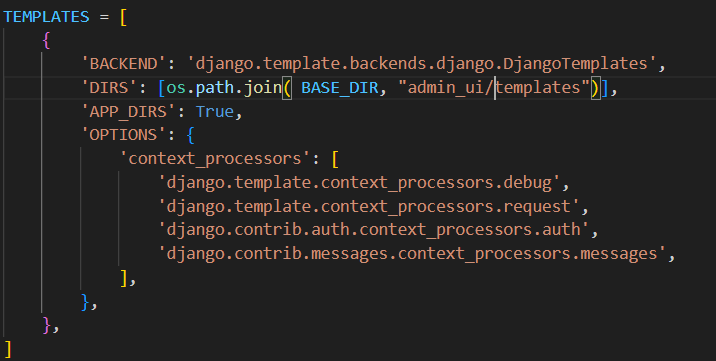
1. Static and Templates Files:



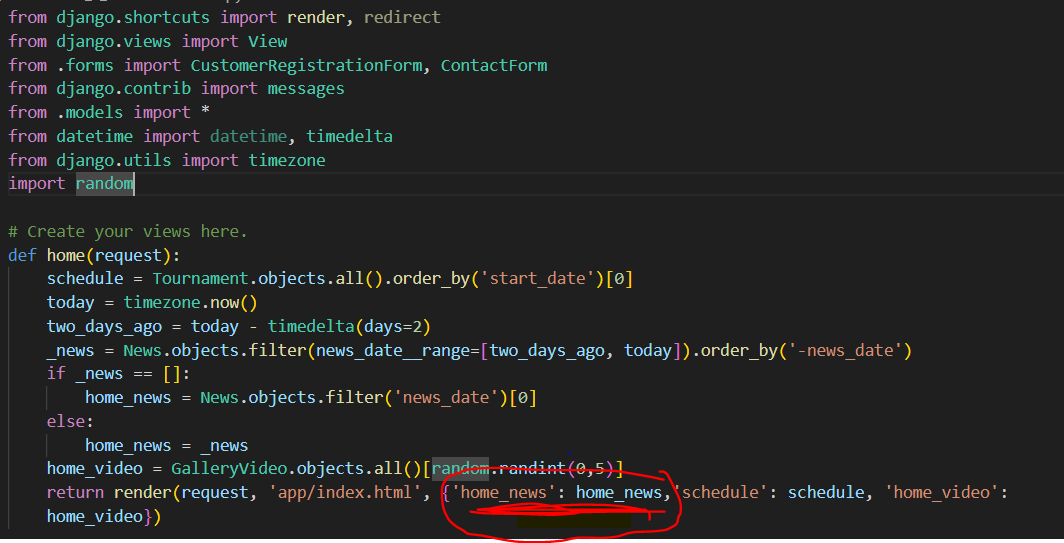
* Static files contain the CSS, images and JavaScript of the app. Make sure to provide the right settings for these files in the setting.py file:

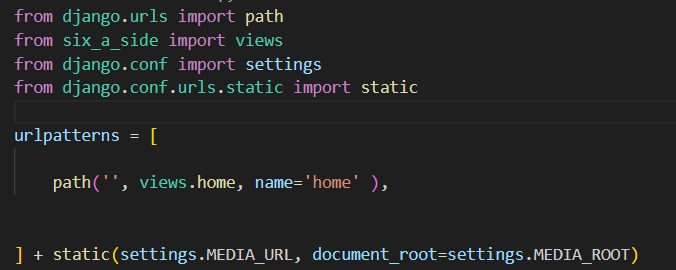


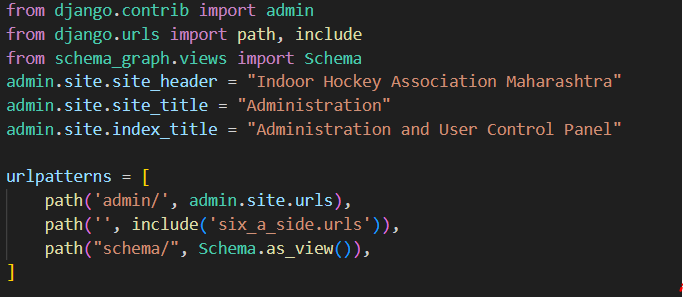
* For templates do the same in the settings.py file



1. Views.py file connects the models and templates file to make the site dynamic all the logic, routing and Authentication takes place here.



* Views.py is mainly used to create route to pass.
* The template and static files are then rendered simultaneously then to make the data dynamic a context is passed to the view function which is highlighted in red in the above code snippet.
* Make function within the function apply the necessary logic as per your convenience and then pass in the form a context.
* To register the route and urls make these adjustment in app and project’s url.py file
* 



This will create the rout made in views.py file.

* Now to register the app do the following in settings.py file:



This will make an app inside the Django projects and following the above process you can create multiple apps with a single project.

1. Now to run the app within production server run the following command and set debug=True in setting.py file.

Python manage.py runserver

1. **Frontend:** The UI are simple bootstrap templates mentioned previously.

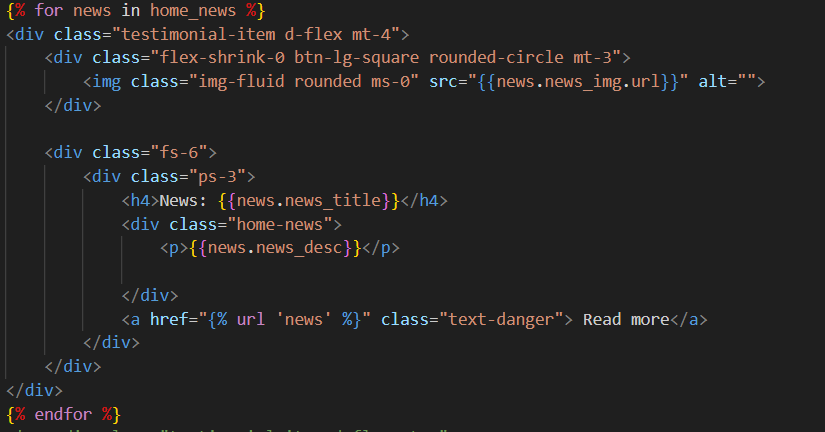
* **Base.html** is the base containing the header and footer and all the other templates inherit this template using Django template

****

* Make sure to load all the static files and pass it like {% static ‘path’ %} in an img or scr tag.

****

* Context can be utilised in the following way:

****

For additional details read official documentation of templates:

https://docs.djangoproject.com/en/4.2/topics/templates/