## Django React

1) Create new folder **4_django_react** beside **3_react**.

```
mkdir 4_django_react
```

2) Check python version available 3.7.

```
py -0
```

3) Create new python virtual environment.

```
py -3.7 -m venv django2.2-venv
```

4) Activate virtual environment.

```
./django2.2-venv/Script/activate
```

5) Install django and check by pip list.

```
python -m pip install django==2.2
```

6) Create new project using django.

```
python -m django startproject hrms
```

7) Rename hrms project name as hrms-api.

```
cd hrms-api
```

8) Run server by manage.py file.

```
python manage.py runserver
```

9) Test in localhost:8000 in browser.

**The install worked successfully! Congratulations!**

## Application Programming Interface (API)

1) Create new django application.

```
python manage.py startapp api
```

2) Register new app in setting.py

3) Database tables migrate.

```
python manage.py migrate
```

4) Check **hrms-api/db.sqlite3** database.
Download SQL Query Browser (www.sqlitebrowser.org)

5) Create new admin user.

```
python manage.py createsuperuser
```

username: admin
email: admin@gmail.com
password: superuser

6) Login django administration at localhost:8000/admin in browser.

7) Create new employee table.

**hrms-api/api/models.py**

```
from django.db import models
# Create your models here.
class EmployeeModel(models.Model):
        name = models.CharField(max_length=20)
        phone = models.CharField(max_length=20)
        address = models.CharField(max_length=20)
```

8) makemigrations and migrate for new change.

```
python manage.py makemigrations api
python manage.py migrate api
```

9) Register for django administration.

```
from django.contrib import admin
from .models import EmployeeModel

# Register your models here.
admin.site.register(EmployeeModel)
```

## Django Restful Framework

1) Install django restful framework. (www.django-rest-framework.org)

```
python -m pip install djangorestframework==3.9.2
```

2) Register in setting.py

```
INSTALLED_APPS = [
    . . .
    'django.contrib.staticfiles',
    'rest_framework',
    'api',
]
```

3) Create new serializers.py

**hrms-api/api/serializers.py**

```
from rest_framework import serializers
from .models import EmployeeModel

class EmployeeSerializer(serializers.ModelSerializer):
        class Meta:
                model = EmployeeModel
                fields = ['id', 'name', 'phone', 'address']
```

4) Edit views.py.

**hrms-api/api/views.py**

```
from django.shortcuts import render

# Create your views here.
from rest_framework import viewsets
from .models import EmployeeModel
from .serializers import EmployeeSerializer

class EmployeeViewSet(viewsets.ModelViewSet):
        serializer_class = EmployeeSerializer
        queryset = EmployeeModel.objects.all()
```

5) Creat new urls.py

**hrms-api/api/urls.py**

```
from rest_framework import routers
from django.urls import path, include
from .views import EmployeeViewSet

router = routers.DefaultRouter()
router.register('employees', EmployeeViewSet)

urlpatterns = [
        path('', include(router.urls))
]
```

6) Edit root urls.py

**hrms-api/hrms/urls.py**

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/', include('api.urls'))
]
```

7) Run **localhost:8000/api/employees** in browser.

## Test API Method

### POST (Create new employee)

```
HTTP 201 Created
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 2,
    "name": "Mg Mg",
    "phone": "09787897878",
    "address": "Mandalay"
}
```

### GET (Read employee)

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "name": "Kyaw Kyaw",
        "phone": "09383838",
        "address": "Yangon"
    }
]
```

### PUT (Update employee)

Change url => **http://localhost:8000/api/employees/2**

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 2,
    "name": "Maung Maung",
    "phone": "09787897878",
    "address": "Mandalay"
}
```

**DELETE (Delete employee)**

Change url => **http://localhost:8000/api/employees/2**

```
HTTP 204 No Content
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
```

**Are you sure you want to delete this Employee Instance?**

## Auth Token

1) Register in setting.py

```
INSTALLED_APPS = [
    . . .
    'rest_framework',
    'rest_framework.authtoken',
    'api',
]
```

2) Migrate auth token table.

```
Python manage.py migrate
```

3) Create token for admin user at django administration.

**5bfc020cdc3bbe1f3e399fe2c5727c6c7e85c28a**

## Postman API Platform

Download postman (**www.postman.com**)

1) Edit root urls.py

**hrms-api/hrms/urls.py**

```
from django.contrib import admin
from django.urls import path, include
from rest_framework.authtoken.views import obtain_auth_token

urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/', include('api.urls')),
    path('auth/', obtain_auth_token)
]
```

2) In postman api platform.

POST => **localhost:8000/auth/**.

**Body Form Data**

username = admin
password = superuser

**Return Result**

```
{"token":"5bfc020cdc3bbe1f3e399fe2c5727c6c7e85c28a"}
```

## Permission IsAuthenticated

1) Edit setting.py

```
. . .
WSGI_APPLICATION = 'hrms.wsgi.application'

REST_FRAMEWORK = {
    'DEFAULT_PERMISSION_CLASSES': (
        'rest_framework.permissions.IsAuthenticated',
    )
}
. . .
```

2) Edit views.py

```
from django.shortcuts import render

# Create your views here.
from rest_framework import viewsets
from .models import EmployeeModel
from .serializers import EmployeeSerializer
from rest_framework.authentication import TokenAuthentication

class EmployeeViewSet(viewsets.ModelViewSet):
            serializer_class = EmployeeSerializer
            queryset = EmployeeModel.objects.all()
            authentication_classes = (TokenAuthentication,)
```

3) In postman api platform.

GET => **localhost:8000/employees**.

**Body Form Data**

username = admin
password = superuser

**Return Result**

```
{
  "detail": "Authentication credentials were not provided."
}
```

4) Include headers.

GET => **localhost:8000/employees**.

**Body Form Data**

username = admin
password = superuser

**Headers**

Authorization = Token 5bfc020cdc3bbe1f3e399fe2c5727c6c7e85c28a

**Return Result**


```
[
   {
      "id": 1,
      "name": "Kyaw Kyaw",
      "phone": "09383838",
      "address": "Yangon"
   }
]
```