

Model Deployment

: إرسال البيانات عن طريق Form

- إنشاء مشروع وتسمية المشروع MyApp
- فتح Terminal ثم تنزيل Django عن طريق الأمر

```
python -m pip install django
```

- إنشاء مشروع MySite عن طريق كتابة الأمر

```
django-admin startproject MySite
```

- الدخول للمشروع عن طريق كتابة `cd MySite` ثم إنشاء تطبيق MyApp عن طريق كتابة الأمر

```
python manage.py startapp MyApp
```

- بداخل ملف `settings.py` نقوم بإضافة التطبيق كالتالي:

```
INSTALLED_APPS = ['MyApp']
```

- الدخول إلى مجلد MyApp ثم إنشاء مجلد `templates` ثم إنشاء ملف `index.html`

```
<body>

<form action="result">
  {% csrf_token %}
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>

</body>
```

- إنشاء ملف `result.html` بداخل مجلد `templates`.

```
<body>
```

```
{{ result }}
```

```
</body>
```

- بداخل مجلد MyApp نقوم بتعديل ملف `views.py` بالشكل التالي:

```
from django.shortcuts import render

# Create your views here.
def home(request):
    return render(request, 'index.html')

def result(request):
    fname = request.GET['fname']
    lname = request.GET['lname']
    return render(request, 'result.html', {'result': [fname, lname]})
```

- داخل مجلد MySite نقوم بتعديل ملف `urls.py` كالتالي:

```
from MyApp.views import home, result
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', home, name='home'),
    path('result/', result, name='result')
]
```

- نقوم بتشغيل الملف عن طريق الأمر:

```
python manage.py runserver
```

خطوات Deployment باستخدام Django:

- إنشاء مشروع وتسمية المشروع TitanicProject
- فتح Terminal ثم تنزيل Django عن طريق الأمر

```
python -m pip install django
```

- إنشاء مشروع TitanicSite عن طريق كتابة الأمر

```
django-admin startproject TitanicSite
```

- الدخول للمشروع عن طريق كتابة cd TitanicSite ثم إنشاء تطبيق TitanicApp عن طريق كتابة الأمر

```
python manage.py startapp TitanicApp
```

- بداخل ملف settings.py نقوم بإضافة التطبيق كالتالي:

```
INSTALLED_APPS = ['TitanicApp']
```

- لصق ملف ml_model وملف البيانات train.csv في مجلد المشروع TitanicSite - المجلد الخارجي (Level 1).
- تحميل المكتبات المطلوبة بداخل ملف ml_model.
- تشغيل ملف ml_model ليتم إنشاء ملفات.pkl.
- بداخل مجلد TitanicApp نقوم بلصق ملف views.py و مجلد templates.
- داخل مجلد TitanicSite نقوم بتعديل ملف urls.py كالتالي:

```
from TitanicApp.views import home, result
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', home, name='home'),
    path('result/', result, name='result')
]
```

- نقوم بتشغيل الملف عن طريق الأمر:

```
python manage.py runserver
```

خطوات Deployment باستخدام Flask:

- إنشاء مشروع وتسمية المشروع TitanicFlaskProject
- فتح Terminal ثم تنزيل Flask عن طريق الأمر

```
python -m pip install flask
```

- لصق الملفات التالية train.csv، ml_model.py، app.py ومجلد templates.
- تحميل المكتبات المطلوبة بداخل ملف ml_model.
- تشغيل ملف ml_model ليتم إنشاء ملفات.pkl.

- نقوم بتشغيل الملف عن طريق الأمر:

```
python -m flask run
```

مصادر أخرى للإطلاع:

- <https://satr.codes/courses/FymypsKhJJ/view>
- <https://www.jetbrains.com/pycharm/download/#section=mac>
- <https://www.youtube.com/watch?v=tJbkJDKcZ3w&t=17s>
- <https://towardsdatascience.com/how-to-deploy-machine-learning-models-601f8c13ff45>
- <https://machinelearningmastery.com/save-load-machine-learning-models-python-scikit-learn/>
- <https://towardsdatascience.com/why-turn-into-a-pickle-b45163007dac>
- <https://www.youtube.com/watch?v=6M3LzGmlAs>
- <https://www.youtube.com/watch?v=6M3LzGmlAso>