

# مکتب شریف

بوت کمپ 116

پروژه گروهی

ساخت فریم ورک اختصاصی

## فهرست مطالب

1. دانلود و اجرای فایل های مورد نیاز برای frontend پروژه ..... 1
2. استفاده از jinja2 ..... 1
3. تعریف مدل ها ..... 1
4. ساختار پروژه ..... 2
5. احراز هویت با Session ..... 3

سلام خدمت کاروند های عزیز. امیدوارم که تا اینجای کار تونسته باشید مطالب مطرح شده در طول دوره را دنبال کنید. پروژه ی حاضر یکی از مهمترین پروژه هایی هست که در تا اینجا مطرح میشه چون قراره زیر ساخت توسعه ی backend رو شکل بده همچنین کار کردن با فایل های jinja و html و css هم توی این پروژه لحاظ شده. امیدوارم با تلاش و پشتکارتون بتونید این پروژه رو به نحو احسن انجام بدید.

## 1. دانلود و اجرای فایل های مورد نیاز برای frontend پروژه

در این مرحله ابتدا ویدیوی که در این [لینک](#) وجود دارد را با دقت تماشا کنید و مطالب مطرح شده در کلاس فرانت را مرور کنید. بعد از تسلط روی مفاهیم مورد استفاده در ویدیو، کدهای مورد نیاز برای پروژه را از [لینک](#) گیت هاب دانلود کنید. صفحات مورد نیاز برای پروژه به شرح زیر است:

- Login
- Register
- Posts
- Post

در نهایت باید فایل ها را به گونه ای کنار هم قرار دهید که در مرور گر سیستم شما این صفحات قابل نمایش باشد.

## 2. استفاده از jinja2

در این مرحله از پروژه نیاز است تا صفحات html را تبدیل به template کنید. تفاوت این دو در واقع در این است که در template می توانید مقادیر تعریف شده در پایتون را جایگذاری کنید. برای آشنایی با jinja2 template engine به این [لینک](#) مراجعه کنید. صفحات مورد نیاز در مرحله ی قبل را ایجاد کنید و در پوشه ای به نام template ذخیره کنید. در مراحل بعدی از این صفحات استفاده خواهد شد.

## 3. تعریف مدل ها

در این مرحله نیاز است تا با استفاده از sql alchemy مدل های مورد نیاز برنامه را تعریف کنید. برای آشنایی با این کتابخانه به این [لینک](#) مراجعه کنید. مدل های مورد نیاز برای این پروژه به صورت زیر است:

- User
  - Id(int)
  - Username(string)
  - Password(string)
  - Last\_login(date time)

- Post •
  - Author (foreign key) ○
  - Create\_al(date time) ○
  - Title(string) ○
  - Body(string) ○
- Session •
  - Session\_id(string) ○
  - Session\_data(string) ○
  - Expire\_date(date) ○

#### 4. ساختار پروژه

برای ساخت پروژه ساختاری شبیه به ساختار زیر را خواهیم داشت:

```
framework/
├── app/
│   ├── __init__.py
│   ├── models.py
│   ├── views.py
│   ├── templates/
│   │   └── index.html
│   └── urls.py
├── core/
│   ├── __init__.py
│   ├── http.py # Manages requests and responses
│   ├── routing.py # URL routing
├── main.py # Entry point
└── requirements.txt
```

در این ساختار باید سه عنصر اساسی MVT را پیاده سازی کنیم.

- Model برای ذخیره سازی داده ها
- View برای پیاده سازی منطق برنامه
- Template برای نمایش دادن داده ها در قالب مناسب

همچنین علاوه بر اجزای اصلی فوق، نیاز به یک router داریم تا request ها را به view مناسب متصل کند. برای این کار از فایل url استفاده می کنیم.

ساختار کلی به این شکل است که ابتدا هر request با توجه به url ای که داخل آن است با یکی از الگوهایی که در فایل urls.py قرار دارد تطبیق داده میشود. هر الگو داخل فایل url با یکی از view های موجود در فایل views.py متصل شده است.

هر view داخل فایل views.py یک تابع است که که اطلاعات request را دریافت می کند و بعد از محاسبات لازم یک template را رندر می کند و جواب را برمی گرداند.

در هر مورد که لازم باشد می توانید از اطلاعات موجود در مدل ها استفاده کنید تا بتوانید پاسخ مناسب را تولید کنید.

## 5. احراز هویت با Session

در مورد سیستم احراز هویت با استفاده از session تحقیق کنید (این [لینک](#)) و قسمتی که مرتبط به این بخش است پیاده سازی کنید. توضیحات بیشتر در جلسه ی تدریس این هفته ارائه می شود.

در فایل های پیوست یک نمونه ی ساده برای این پروژه نوشته شده که کامل نیست و باید توسط گروهتون تکمیل بشه. با الگو گرفتن از این ساختار، پروژه ی مطرح شده رو پیش ببرید. دقت کنید که قسمت زیادی از این پروژه نیاز به تحقیق و مطالعه داره و وقت گذاشتن روی این مفاهیم میتونه ادامه ی کار شما رو برای جنگو به شدت تسهیل کنه پس ترس به دلتون راه ندید و با قدرت پروژه رو پیش ببرید.

نوذری پاک