



Web Apps With Django

With Nouar Chaouki, software
engineer student at Ibn
Khalidoun Univ



What we gonna see today


WEB DEVELOPMENT

1. Introduction on web development.
2. How do we engage with it.
3. Framework vs Scratch code.
4. How to choose the right framework.

DJANGO FRAMEWORK

1. What is Django.
2. What are we building with django.
3. Starting a project.
4. Models.
5. Views, URLs and Templates.

Summary





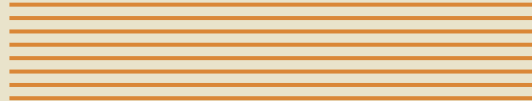
WHAT IS **WEB DEVELOPMENT**


INTRODUCTION

Web development is to build and maintain websites.

The Three Types of Web Developers :


1. The Front-End Developer (Web Designer)
2. The Back-End Developer
3. The Full-Stack Developer (The Superman)





Frontend development: A frontend developer is, in large part, responsible for what a user sees. The 3 main tools that frontend developer use are HTML, CSS, and JavaScript. Their primary focus should be to create a great user experience.

Backend development: A backend developer on the other hand works on what goes on behind the scenes. Backend developers work with a wide range of libraries, APIs, frameworks, etc. He is responsible for the implementation of database systems, generating backend functionality, and more.



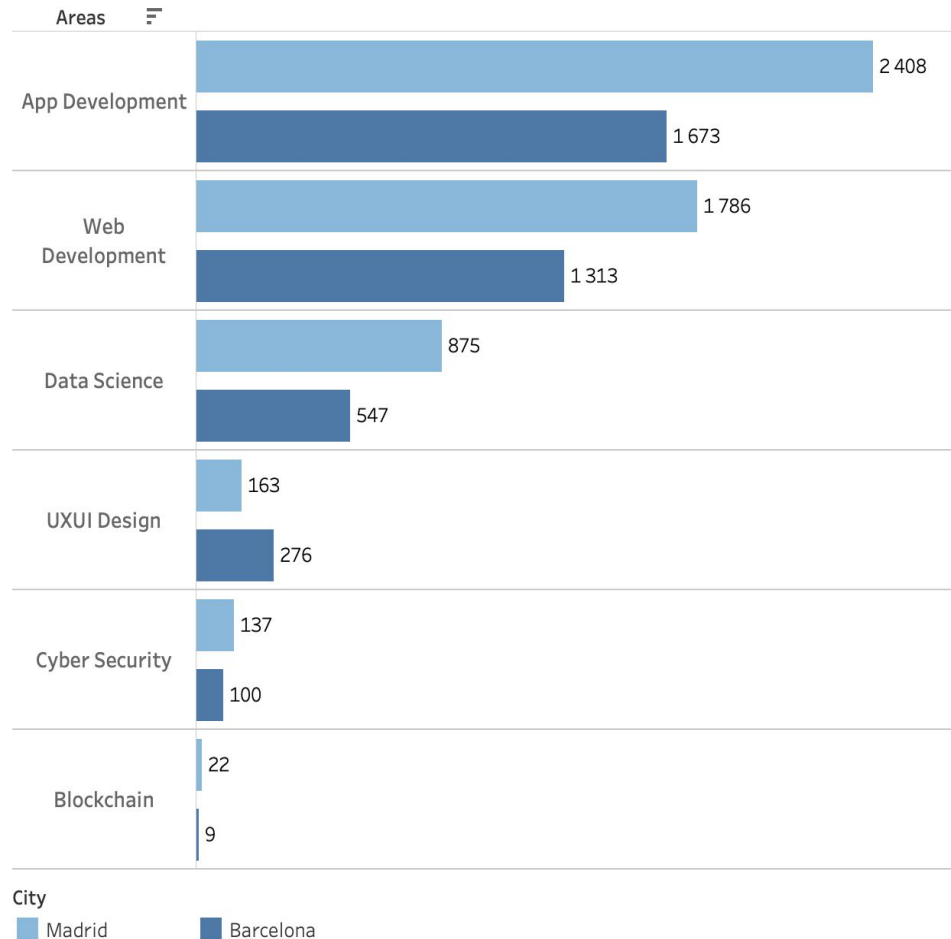
Frontend



Backend



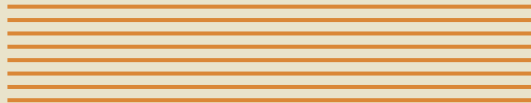
Job postings in 6 Emerging Tech areas



How do we engage with it

Most important think you have to decide which frameworks and libraries fit your needs when it comes to frontend and backend

Framework vs library :The key difference between a library and a framework is “Inversion of Control”. When you call a method from a library, you are in control. But with a framework, the control is inverted: the framework calls you.






Framework vs Scratch code

Often people argue about using framework or scratch coding.

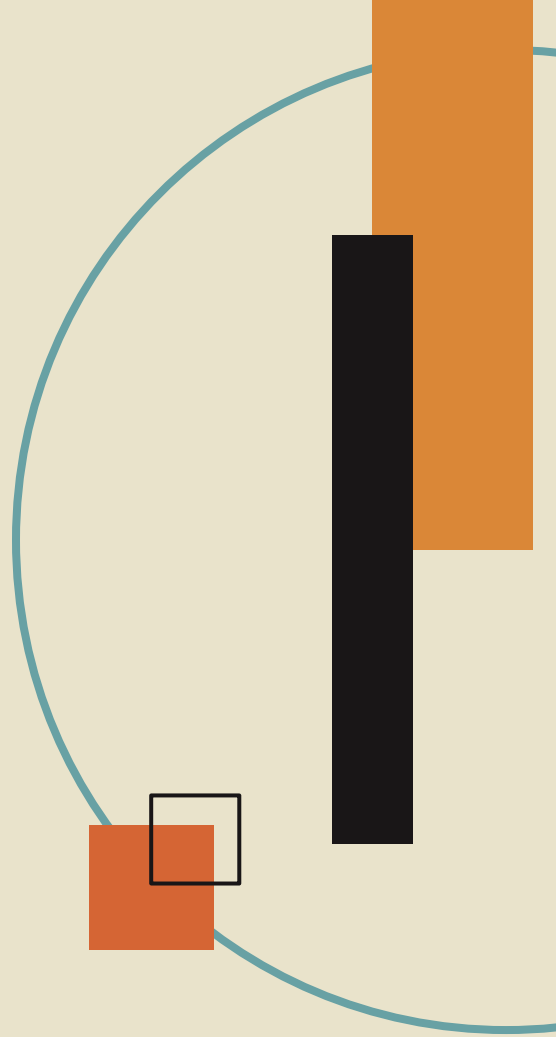
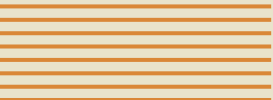
In my opinion frameworks are your option in 99% of the cases, here is why :

1. Save time... and time equal money.
2. Frameworks are already been tested and they are bug free.
3. Managed and maintained by someone else with ease.
4. Write less code.

Note : Scratch code makes you a better developer, It's a good way to learn the basics.



Most popular frameworks today





Top 10 Frameworks for Web Application Development



Asp.net

Ruby on Rails



Laravel



Vue



Django



Express



Spring




Play



Angular



Code Igniter



What to consider when selecting a backend framework?


Learning curve: if you're familiar with a particular programming language already, you may want to stick to a framework which utilizes that language for time's sake.

Documentation/community: if you ever get stuck you can search for the proper documentation or ask someone in the community for help.

Libraries: This allows developers to implement or configure a library in order to perform a certain task that would otherwise be time consuming or not possible.

Security: you may require certain security features which some backend frameworks perform better than others.

Scalability: If you're building an app for the masses, you'll want to ensure that your framework provides the ability to scale with you.





Any Questions

What is Django

Django is a high-level Python framework that is built with the idea of "batteries included". Meaning almost everything a developer would want is included out of the box. Django however, is built for larger applications. Therefore, if you're planning on building something small.

A few examples of large websites built on Django include: Mozilla, National Geographic, Pinterest.

Features

1. Highly customizable
2. No need to reinvent the wheel, encourages rapid development
3. Very scalable
4. Extensive community and documentation

What are we building with django

Start the project

1. Install python from python.org
2. Install Django : `~pip install django`
3. Create a folder and start the project : `~django-admin startproject project`
4. Go to the project folder and create app: `~python manage.py startapp projectApp`
5. Add the app name to `INSTALLED_APPS` in `settings.py`
6. Migrate the changes: `~python manage.py migrate`
7. Create a superuser(admin): `~python manage.py createsuperuser`
8. Run the server: `~python manage.py runserver`

Models

1. Create a model named Comment with ForeignKey relation with User
2. Migrate
3. Add to Admin: `~admin.site.register(Comment)`

Django relationships

1. ForeignKey (Comment)
2. OneToOneField (Profile Picture)
3. ManyToManyField (Group)

Views, URLs and Templates

Template: is where the HTML files goes.

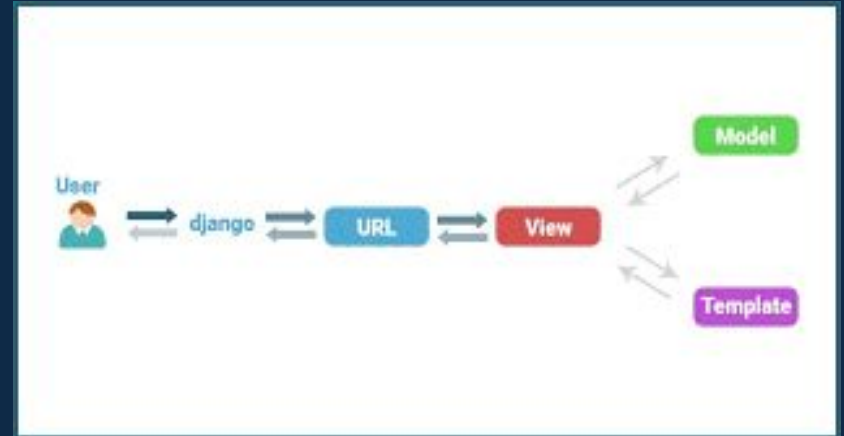
We create a new folder named templates and put the html files in it

View: define a function “comments” with request as Arg

And return `render(request, "comments.html", {})`

URL: connect a view with an URL.


`~path('comments/', views.comments, name='comments')`





Summary

There are a variety of backend frameworks to choose from, each with their own set of pros and cons. Determining which backend framework is the best is rather subjective as it depends on what you're trying to achieve.





Any Questions