\*\*\*\*\*TOPIC FOR PRESENTATION\*\*\*\*\*\*\*\*  
Presentation slides [here](https://docs.google.com/presentation/d/1VCdNsyfoYUPn6ig9ndzRf7af7DnpvFiTSbBPZK-eqgk/edit?usp=sharing) or shared to your rmail.

**DJANGO**

* MVC core architecture (**M**odel **V**iew **C**ontroller)
* Files and data models are written in Python which is used throughout the framework
* Security features to prevent clickjacking, cross-site scripting, and SQL injection  
  Clickjacking - where a malicious site wraps itself in frame, Cross Site Scripting (XSS) is where a user is able to inject malicious client side scripts into the browser of other users, SQL injection is an attack where a user can execute SQL code to delete or leak a database
* Used to manage Instagram, Youtube, Dropbox, Spotify, etc.
* Scalable
* Django is unique as it has three types of files that handle website requests also known as a model-template-view architectural pattern, picture here → <https://www.tutorialspoint.com/django/django_overview.htm>. The developer provides the model, view, and template and links it to a URL that Django will use to present to a user.
  + URLs (py) → Instead of processing URLs with one big function, Django processes URL redirects and sorts them into different viewing functions, which is more maintainable.
  + Models (py) → Receive viewing data and compile it into a comprehensible structure. Allow for database management and query.
  + View (py) → Views in django are request functions. They receive requests and return responses. The requests are received from the URLs and the responses are processed through models
  + Template (html) → Templates are just what it is… templates. They have the structure of the website and output whatever is programmed to them

