How Servers and Frameworks Work

-Websites send HTML to your personal browser

-How does it know what to send? Your browser requests it via HTTP requests

HTTP Protocol

-based on a request-response model

-communication is always initiated by your browser

-**HTTP Methods:**

-HTTP GET: requests data from the server, does not change web application at all

-HTTP POST: carry data entered by the user to the server, like registering a user

-Response Codes: 200- everything is fine, 204- POST may send this saying “everything sent fine but I don’t have anything to show you”

-Most basic HTTP server:

import socket

HOST = ''

PORT = 80

listen\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

listen\_socket.bind((HOST, PORT))

listen\_socket.listen(1)

connection, address = listen\_socket.accept()

request = connection.recv(1024)

connection.sendall("""HTTP/1.1 200 OK

Content-type: text/html

<html>

<body>

<h1>Hello, World!</h1>

</body>

</html>""")

connection.close()

**MVC Pattern:**

- “Model View Controller”

-Django uses this design pattern, except controllers are called “views” and views are “templates”

-a way of logically separating different responsibilities of the application

-*Models* represent data tables such as a table of users from the database

-*Controllers* contain business logic of application and operate on models

-*Views* are given information in order to dynamically generate the HTML version of the page

**Routing in Django:**

-The process of mapping a requested URL to the code responsible for generating the HTML

-We need to map each URL to a view function, for example when ([www.foo.com/bar](http://www.foo.com/bar)) is requested, the function “handle\_bar()” generates the response

-However, if the URL also contains data, then we need to map each URL to a view AND get info

-Django does this by mapping URL regular expressions to view functions that take parameters

-for example, you may say URLs that match “^/users/(?P<id>\d+)/$” call the “display\_user(id)” function

[Regular Expressions/RegEx info](https://medium.com/factory-mind/regex-tutorial-a-simple-cheatsheet-by-examples-649dc1c3f285)

**Routing in Flask:**

-Flask has a different approach to routing that uses a “route()” function

-This Flask code is the same as the regex above:

**@app.route('/users/<id:int>/')**

**def** display\_user(id):

*# ...*

-basically a simplified regex to map URLs to arguments

Once we have the URL mapped, we need to dynamically generate HTML.

-Both Django and Flask do this through HTML templating

-It is similar to str.format() function, with the desired output written with placeholders for dynamic values (like “How is %f doing?”, Emmett)

**Database Interaction:**

-Django includes an ORM (Object Relational Mapper)

-This maps Python classes to database tables and abstracts away the differences between different database engines

-It also gives Django the ability to create a CRUD (Create Read Update Delete) application.

This paper mainly discussed Django and Flask, two popular web frameworks for web applications written in Python. ReactJS is a popular framework for Javascript, and it is what we will be using to connect our code to the web server. ReactJS, like Django, uses the Model View Controller design pattern.