



COM5940: NEW MEDIA BUSINESS MODEL & INNOVATION ENTREPRENEURIAL MINDSET FOR DIGITAL INNOVATION

Bernard Suen
Center for Entrepreneurship
Chinese University of Hong Kong

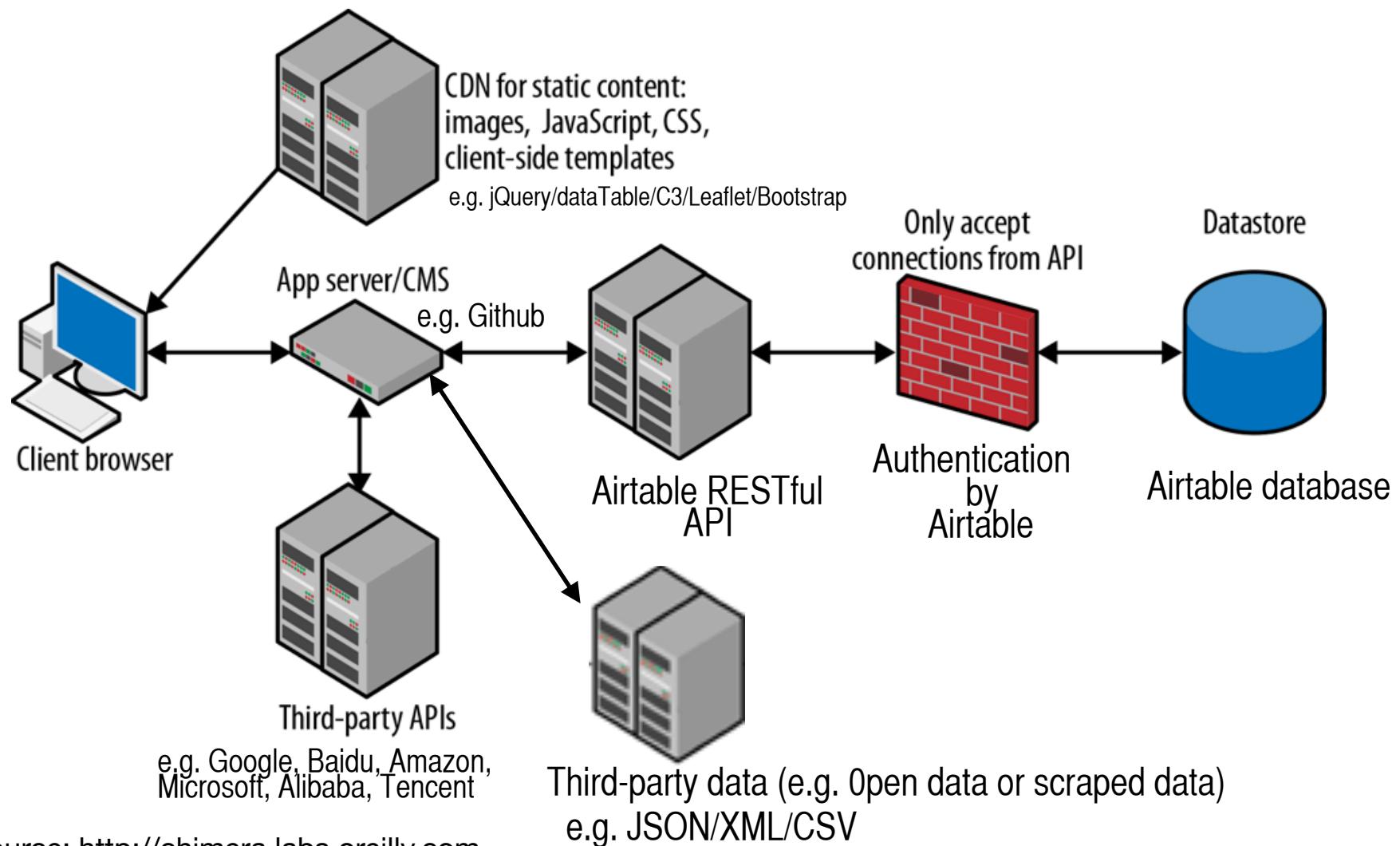


Center for
Entrepreneurship

Today's Agenda

- 1. Full-stack development re-visited**
- 2. What is Python Flask?**
- 3. Use Bootstrap codes in Flask Jinja 2 templates**
- 4. What is the Lean Approach?**
- 5. Customer Development: Startup vs. Corporation**

Quick Recap

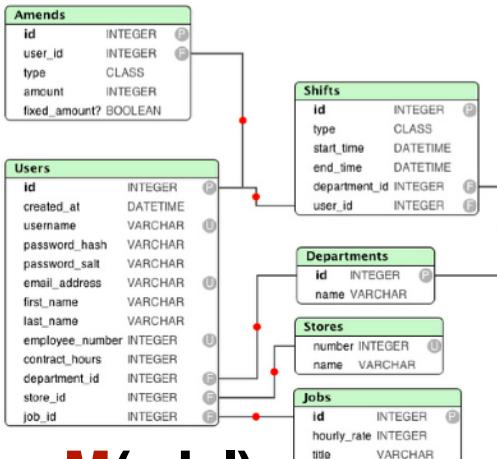


Source: <http://chimera.labs.oreilly.com>

Source: commons.wikimedia.org



Source: [Caius Durling \(Flickr\)](https://flic.kr)



M(odel)
(e.g. table, relations,
and business logic)

V(iew)
(e.g. Kanban, Calendar)

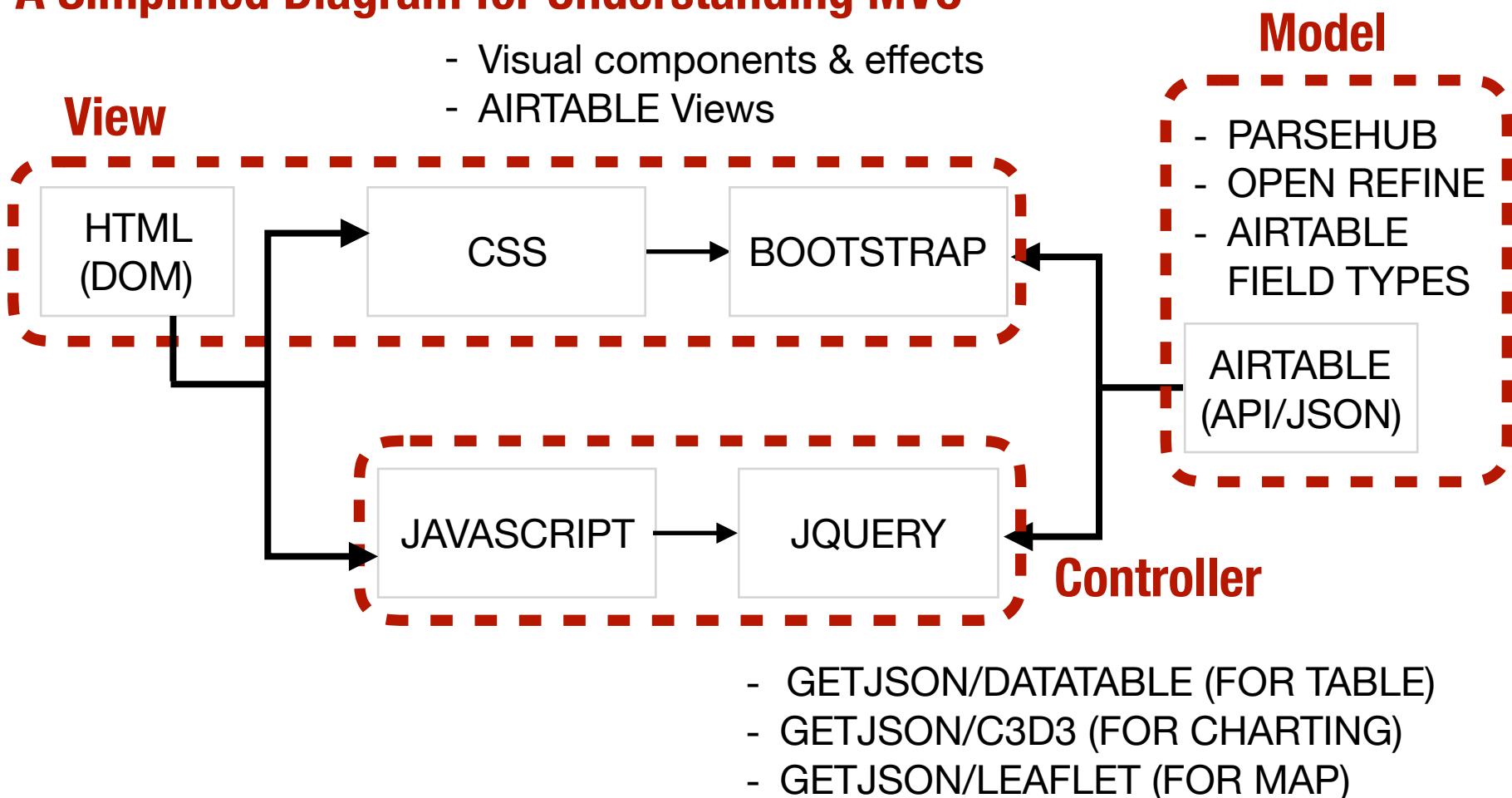
MVC MODEL

Source: pexels.com



C(ontroller)
(e.g. Sitemap, menu, navbar,
and dashboard)

A Simplified Diagram for Understanding MVC



From Front-end to Back-end

FRONT-END PROCESSING

HTML/CSS/JS

FRONT-END TECHNOLOGIES

jQuery/React/AngularJS/Flutter etc



DATA FLOW

BACK-END PROCESSING

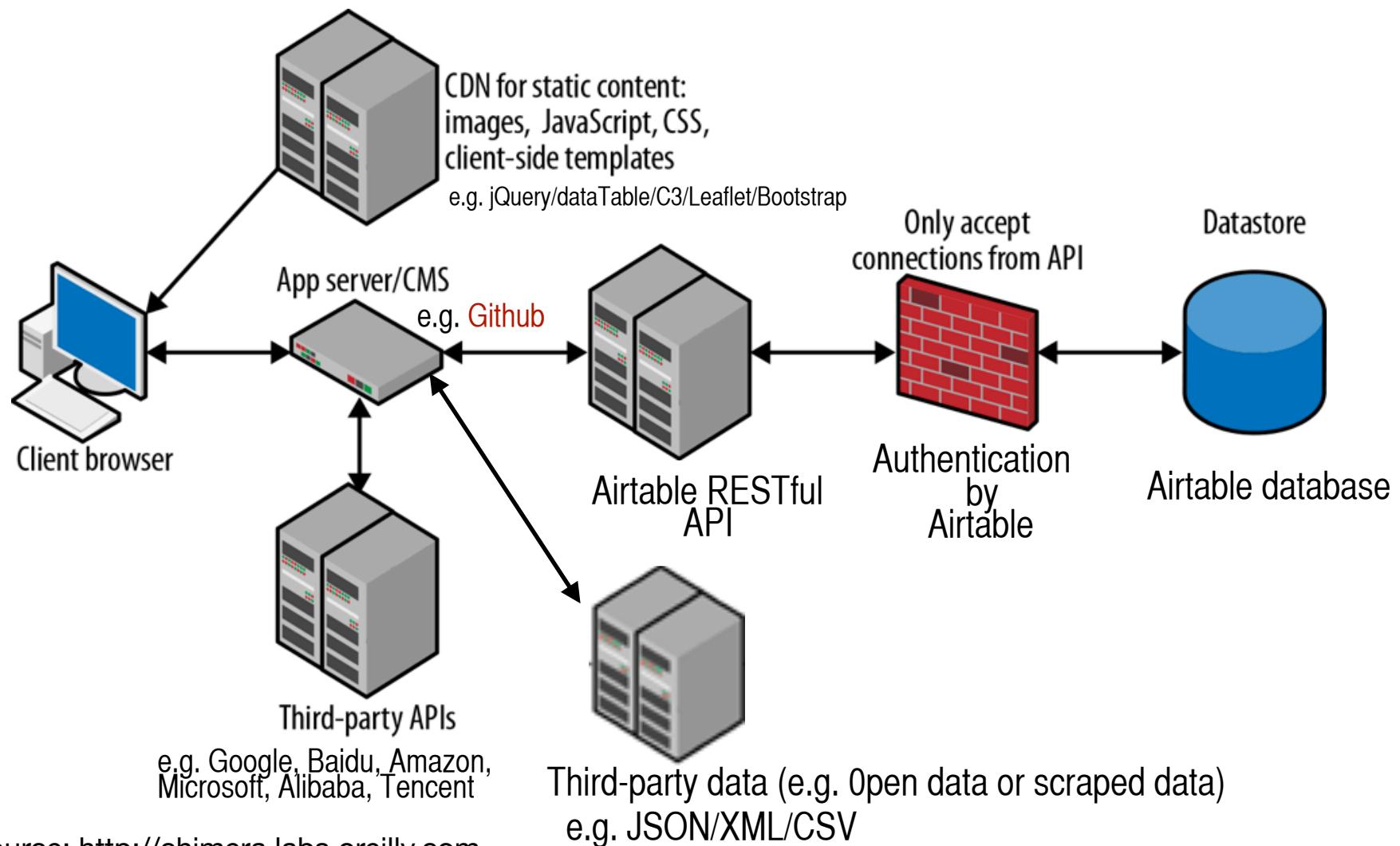
PHP, PYTHON, RUBY,
JAVA, NODE.JS/C# etc

WEB APP FRAMEWORKS

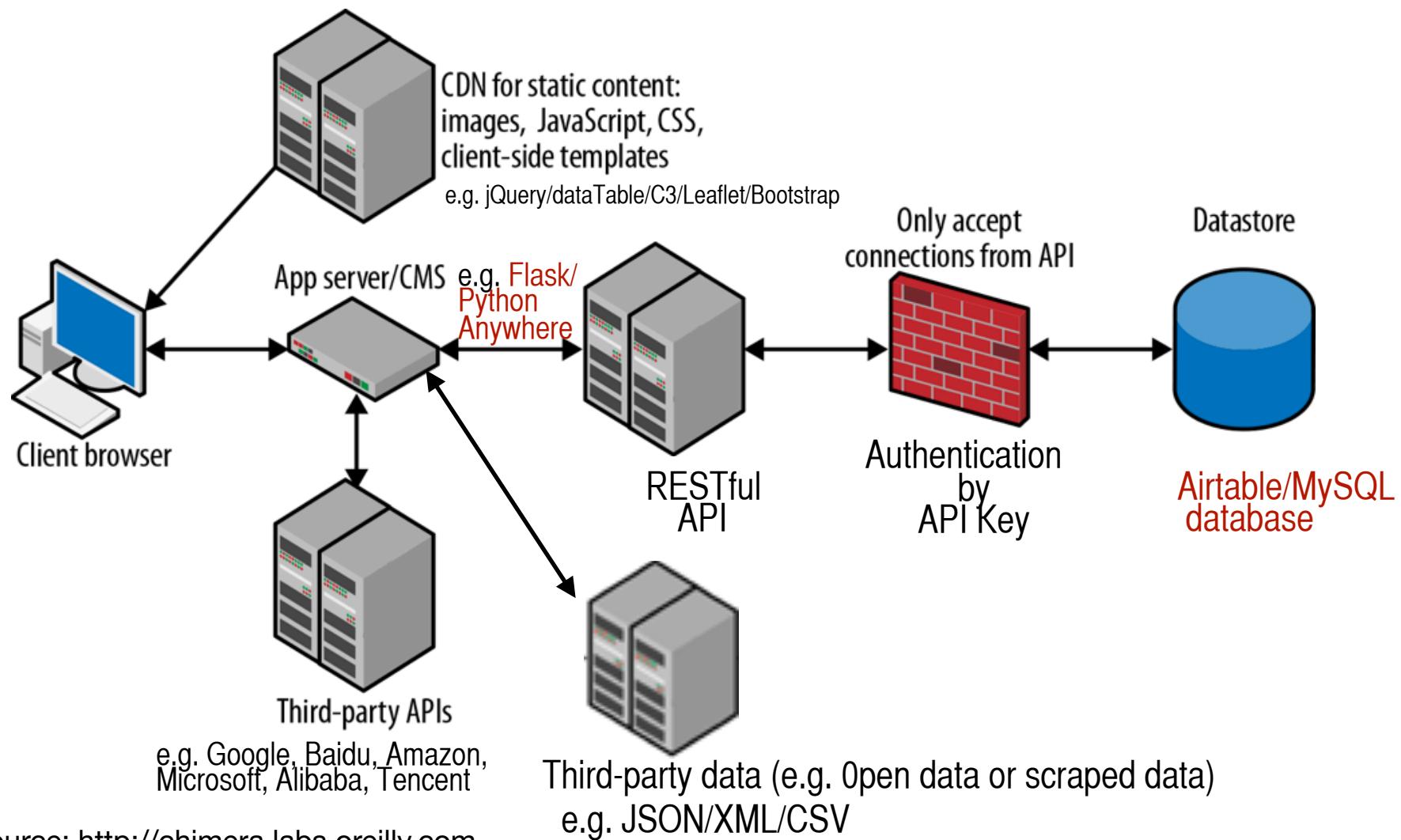
LARAVEL/CODEIGNITER,
DJANGO/FLASK, RUBY ON
RAILS, EXPRESS.JS,
ASP.NET MVC, and SPRING



Which functions should be
processed on the front-end and
which should be on the back-end?



Source: <http://chimera.labs.oreilly.com>

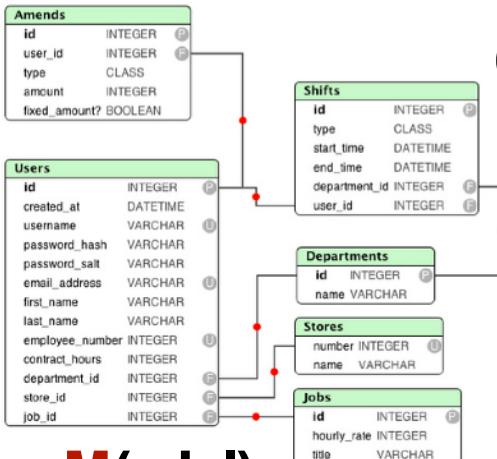


Source: <http://chimera.labs.oreilly.com>

Source: commons.wikimedia.org



Source: [Caius Durling \(Flickr\)](https://flic.kr)

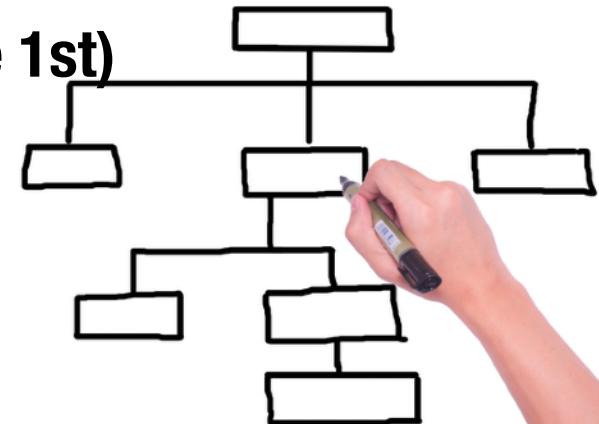


V(iew)
(e.g. Responsive/Mobile 1st)

MVC MODEL

M(odel)
(e.g. table, relations,
and business logic)

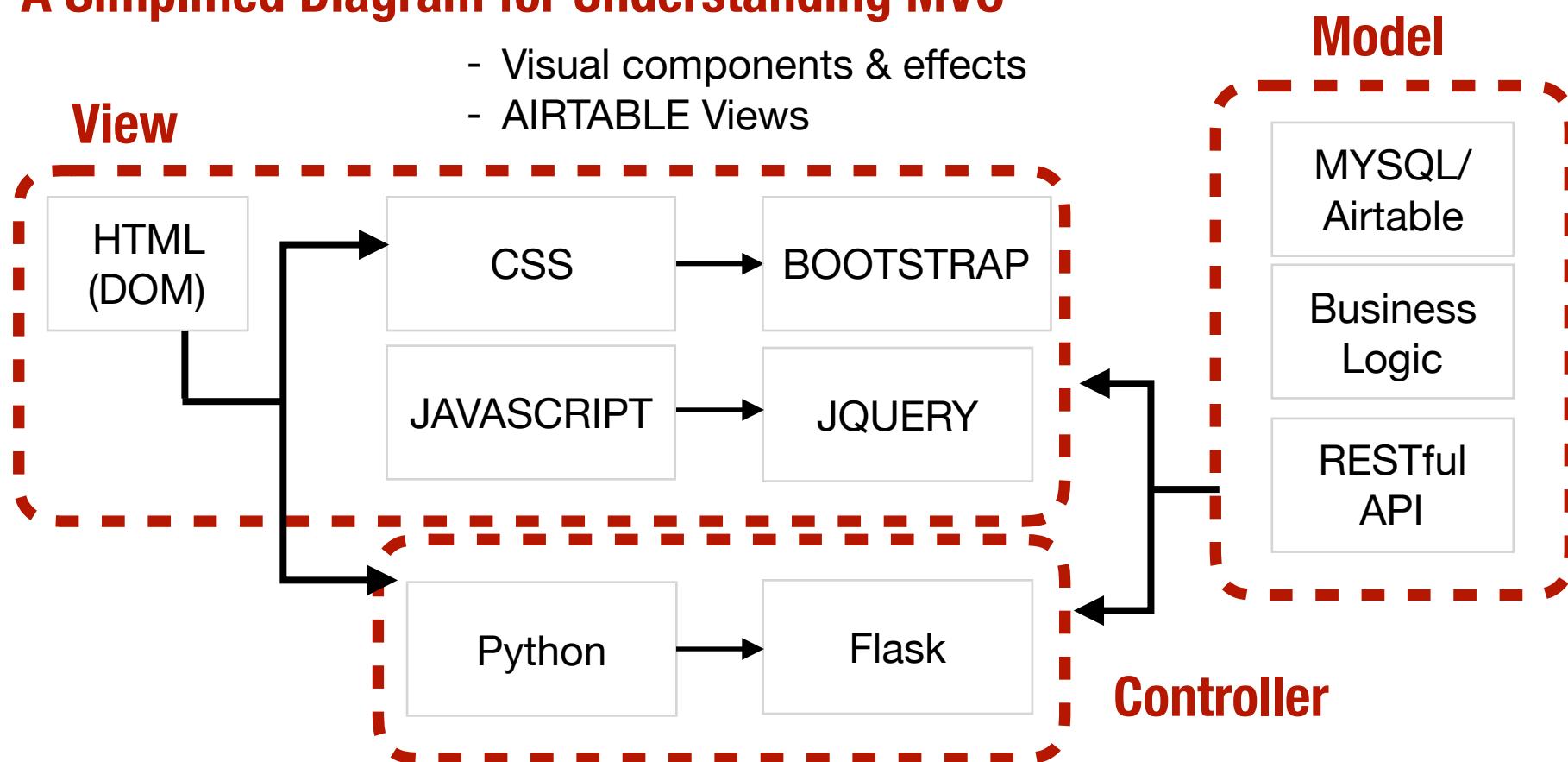
Source: pexels.com



C(ontroller)
(e.g. Flask routes and REST
API endpoints)

A Simplified Diagram for Understanding MVC

- Visual components & effects
- AIRTABLE Views



Full Stack Development (Front-end to Back-end)

STARTUP RESOURCES

WHAT PROGRAMMING LANGUAGE IS ALIBABA BUILT IN?

 KITTY GUPTA — JULY 4, 2018

The banner for the FreelancingGig Free E-mail Signature Generator. It features the FreelancingGig logo and the text "FREE E-MAIL SIGNATURE GENERATOR". Below this is a yellow button labeled "Try It Now" and a preview of the generated email signature.

TRENDING ARTICLES



How to Download and Install WhatsApp for Java Phones



How to Create a WhatsApp Account With a U.S. Number



11 Jokes Only Programmers Will Find Funny



5 Hilarious Programming Jokes for Programmers



Programmer Jokes of The Day





[Share](#)

[Tweet](#)

[Google](#)

[Pinterest](#)

[LinkedIn](#)

Alibaba is the largest business-to-business ecommerce platform in the world for small businesses. Alibaba Group started with Alibaba ecommerce website before starting out other ventures like AliExpress, Taobao, AliCloud, AliMusic and much more. The platform deals with importers and exporters of over 240 countries. To handle such high traffic, the technologies used in developing the platform needed to be modern and robust. It has not been developed with one programming languages. As a matter of fact, no website or web platform is developed around one programming language as such. Today, we are going to look in depth the programming languages used for one of the most used ecommerce platforms in the world.

Server Side Programming Languages -

Alibaba started its operation on 15th April 1999. At that time, it was totally based on the popular scripting language PHP. This does not come as a surprise as PHP was the go-to web development language at that time as far as the industrial level was concerned. There are many popular websites developed during that time which are based on PHP like Facebook for an example. But with the rise of the popularity of the platform, the company had to upgrade the technology and bring in more security and features to make the platform user-friendly. Therefore, at a later stage, JAVA was used as a server-side programming language for certain sections which plays a very important role for the website. The Spring framework has been used for implementing Java-based developments. But till date, PHP is the main server-side programming language.

For the server side, it relies on Tengine web server which is based on Nginx & Taoba. Taobao is a programming language developed by Alibaba Group, and it is based on PHP. The language is available in Chinese for the time being. It is mainly used for proxying contents that are static like CSS, static HTML pages, JS and likewise. This is the reason for its faster response time. Apart from these, it uses various software stacks, and one of them is CDN through Apache 2 for localized content. The operating system used on the web server is RedHat Linux. The servers are located in China and the United States.

Client Side Programming Languages -

Apart from the usual HTML, CSS, and JS, the website has constantly been upgrading its technologies and programming languages. For example, it has moved on to HTML5 and CSS3 after their release to be on the same page as the latest technologies and competitors. It heavily depends on JQuery for most of the unique features and functionalities offered to the users. For the JavaScript, different frameworks are used for robust development and making complicated things easy and organized.



f Share

t Tweet

Server Side Programming Languages -

Alibaba started its operation on 15th April 1999. At that time, it was totally based on the popular scripting language PHP. This does not come as a surprise as PHP was the go-to web development language at that time as far as the industrial level was concerned. There are many popular websites developed during that time which are based on PHP like Facebook for an example. But with the rise of the popularity of the platform, the company had to upgrade the technology and bring in more security and features to make the platform user-friendly. Therefore, at a later stage, JAVA was used as a server-side programming language for certain sections which plays a very important role for the website. The Spring framework has been used for implementing Java-based developments. But till date, PHP is the main server-side programming language.

Alibaba is the largest business-to-business ecommerce started with Alibaba ecommerce website before starting AliExpress and much more. The platform deals with high traffic, the technologies used in developing the website were developed with one programming language. As a result, there was no need for multiple programming languages as such. Today, we are one of the most used ecommerce platforms in the world.

Server Side Programming Languages -

Alibaba started its operation on 15th April 1999. At that time, it was totally based on the popular scripting language PHP. This does not come as a surprise as PHP was the go-to web development language at that time as far as the industrial level was concerned. There are many popular websites developed during that time which are based on PHP like Facebook for an example. But with the rise of the popularity of the platform, the company had to upgrade the technology and bring in more security and features to make the platform user-friendly. Therefore, at a later stage, JAVA was used as a server-side programming language for certain sections which plays a very important role for the website. The Spring framework has been used for implementing Java-based developments. But till date, PHP is the main server-side programming language.

For the server side, it relies on Tengine web server which is developed by Alibaba Group, and it is based on PHP. It is being used for proxying contents that are being served through Apache 2 for localized content. The operations are located in China and the United States.

Client Side Programming Languages -

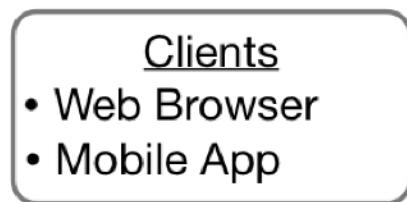
Apart from the usual HTML, CSS, and JS, the website also uses some client-side programming languages. For example, it has moved to React.js for the front-end page as the latest technologies and competitors. It has added more functionalities to offer to the users. For the JavaScript, it has used React.js which makes complicated things easy and organized.

For the server side, it relies on Tengine web server which is based on Nginx & Taobao. Taobao is a programming language developed by Alibaba Group, and it is based on PHP. The website also uses some client-side programming languages.

INFRASTRUCTURE FOR FULL STACK SOFTWARE DEVELOPMENT

云计算的三层架构

(Full Stack Developer
由全栈工程到全栈创业)



Software (软件)
e.g. Your app



Platform (平台)

e.g. Google App Engine/AWS Fargate/PythonAnywhere



Infrastructure(基建)

e.g. Amazon, Azure, Aliyun

Public(公共)/Private(私人)/Hybrid Clouds(混合)

API (e.g. RestFUL)



BACK-END TECHNOLOGY

后台

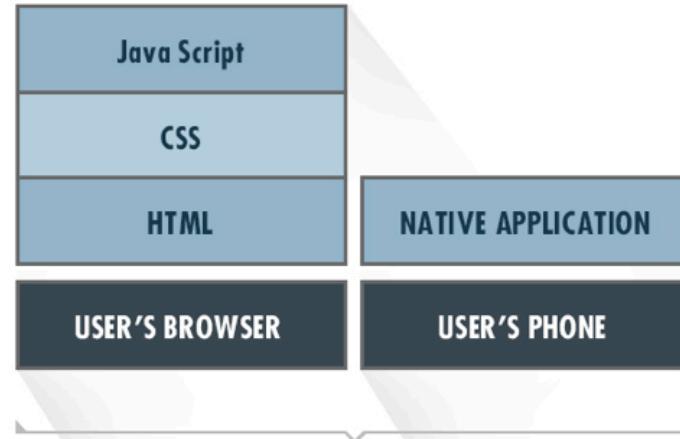
WHAT IS YOUR “CLOUD” AND “STACK” STRATEGY?

云架构及前端和后端的全栈策略

{ JSON }



THE INTERNET



FRONT-END TECHNOLOGY

前台



What
Job title, keywords, or company

Where

 [Advanced Job Search](#)

Lamp Stack Developer jobs

Sort by:
relevance - date

Salary Estimate

\$70,000+ (447)
\$85,500+ (356)
\$95,100+ (288)
\$105,000+ (206)
\$120,000+ (91)

Job Type

Full-time (502)
Contract (20)
Internship (13)
Temporary (8)
Part-time (5)
Commission (1)

Location

New York, NY (33)
Chicago, IL (19)
San Francisco, CA (16)
Los Angeles, CA (15)
Washington, DC (12)

[+ more »](#)

New! Join Indeed Prime - Get offers from great tech companies

Page 1 of 527 jobs

PHP Full Stack Developer

SaaS Start Up
Salt Lake City, UT

\$55,000 - \$80,000 a year

LAMP stack technologies. PHP Full Stack Developer. Mid-Level PHP Full Stack Developer. Why You Want This Job....

20 days ago [save job](#) more...

Full Stack Developer

DHI Telecom Group
Houston, TX

Full Stack Developer. 2+ LAMP experience. The Full Stack Developer considers the web application's intended function and works to integrate PHP code with...

1 day ago [save job](#) more...

LAMP Full Stack Developer (3+ years experience)

Career

Caxy Interactive
Chicago, IL

Responsibilities for a Full Stack Developer include... Our Full Stack Developers exhibit these competences... You may be perfect for this role if you have.....

Be the first to see new Lamp Stack Developer jobs

My email:

Also get an email with jobs recommended just for me

[Activate](#)

Full Stack Developer salaries in United States

\$113,616 per year ⓘ

Based on 33,198 salaries



Full Stack Developer salaries by company in United States

<https://www.indeed.com/q-Lamp-Stack-Developer-jobs.html>



What
Job title, keywords, or company

Where
City, state, or zip code

Find jobs Advanced Job Search

Tip: Enter your city or zip code in the "where" box to show results in your area.

Mean Stack jobs

My recent searches

Lamp Stack Developer

» clear searches

Sort by:

relevance - date

Salary Estimate

\$25,000+ (5525)

\$30,000+ (4729)

\$40,200+ (2851)

\$100,000+ (2096)

\$125,000+ (1076)

Job Type

Full-time (5638)

Part-time (239)

Contract (117)

Internship (44)

Temporary (31)

Commission (6)

Location

New York, NY (202)

San Francisco, CA (180)

Upload your resume - Let employers find you Page 1 of 5,806 jobs

Mean Stack Developer

Cognizant ★★★★★ 10,661 reviews

San Jose, CA +1 location

Familiarity with the whole web stack, including protocols and web server optimization techniques. Roles & Responsibilities:....

3 days ago [save job](#) more...

Full Stack Developer - MEAN stack

L&T Infotech ★★★★★ 989 reviews

New York, NY

Angular JS, Node JS, Mongo DB, TDD, MEAN stack, Redis/Memcache. UI/ stack developer with working knowledge of following technologies:....

18 days ago [save job](#) more...

Full Stack MEAN developer

Elevate Blue

Lake Tahoe, NV

We are looking for an AngularJS / MEAN stack developer to lead the evolution of one of our most promising and exciting portfolio companies....

21 days ago [save job](#) more...

Be the first to see new Mean Stack jobs

My email:

Also get an email with jobs recommended just for me

Activate

Full Stack Developer salaries in United States

\$113,616 per year ⓘ

Based on 33,198 salaries



Full Stack Developer salaries by company in United States

<https://www.indeed.com/q-Mean-Stack-Developer-jobs.html>



What
Job title, keywords, or company

Where
City, state, or zip code

 Advanced Job Search

React Node Full Stack Developer jobs

My recent searches

Mean Stack 67 new

Lamp Stack Developer 11 new

» clear searches

Sort by:

relevance - date

Salary Estimate

\$100,000+ (1269)

\$105,000+ (1180)

\$115,000+ (787)

\$120,000+ (607)

\$130,000+ (326)

Job Type

Full-time (1473)

Contract (68)

Internship (22)

Temporary (13)

Part-time (8)

Commission (7)

Location

New York, NY (108)

San Francisco, CA (104)

Seattle, WA (54)

Austin, TX (44)

Chicago, IL (41)

Tip: Enter your zip code in the "where" box to show results in your area.

New! Join Indeed Prime - Get offers from great tech companies

Page 1 of 1,530 jobs

Show: all jobs - 38 new jobs

Senior Full Stack Developer (React, Node) - CONTRACT (open t...)

RevUnit

Las Vegas, NV 89118 [+1 location](#)

Experience with React and Node. We are looking for a Lead Full Stack Developer to join us on a contract basis....

21 days ago [save job](#) more...

Full Stack Product Developer - CSS/JavaScript/React/Node

Carfax 80 reviews

Columbia, MO 65201

Join our elite IT team as a Full Stack Developer - JavaScript. As a Full Stack Developer, you will be a part of the Product Services team....

30+ days ago [save job](#) more...

Full Stack Engineer

Auth0

Remote

Our current stack focuses on Node, ElasticSearch, MongoDB, and React, but we like to focus on picking the right tools for the job....

30+ days ago [save job](#) more...

Be the first to see new React Node Full Stack Developer jobs

My email:

Also get an email with jobs recommended just for me

[Activate](#)

Full Stack Developer salaries in United States

\$113,628 per year

Based on 33,397 salaries



Full Stack Developer salaries by company in United States

<https://www.indeed.com/q-React-Node-Full-Stack-Developer-jobs.html>



https://www.youtube.com/watch?v=0rK5xs8_rnQ

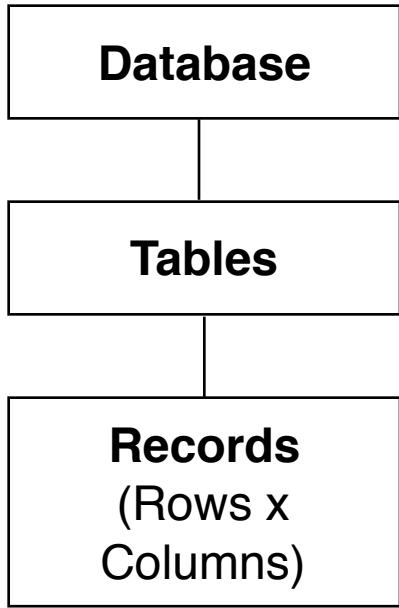
A Full-Stack Example of a Tech Firm

**But instead of using MongoDB, we
choose to use PostgreSQL
(Postgres) and Elasticsearch.**

In this semester, we'll focus on how to develop a full-stack solution using jQuery/Bootstrap/Phonegap on the front-end with a MVC framework using Python Flask on the back-end to access Airtable and MySQL databases. But we'll also touch upon other technologies just to broaden your sense of the internet industry.

First, starting with the database...

Data and Business/Application Logic Models



“A **database** is a collection of **tables** which relate to each other for storage, processing and queries **records** that are defined by rows and columns (**fields**).”



Product Blocks Templates Universe Pricing Sign in [Sign up](#)

Create, your way

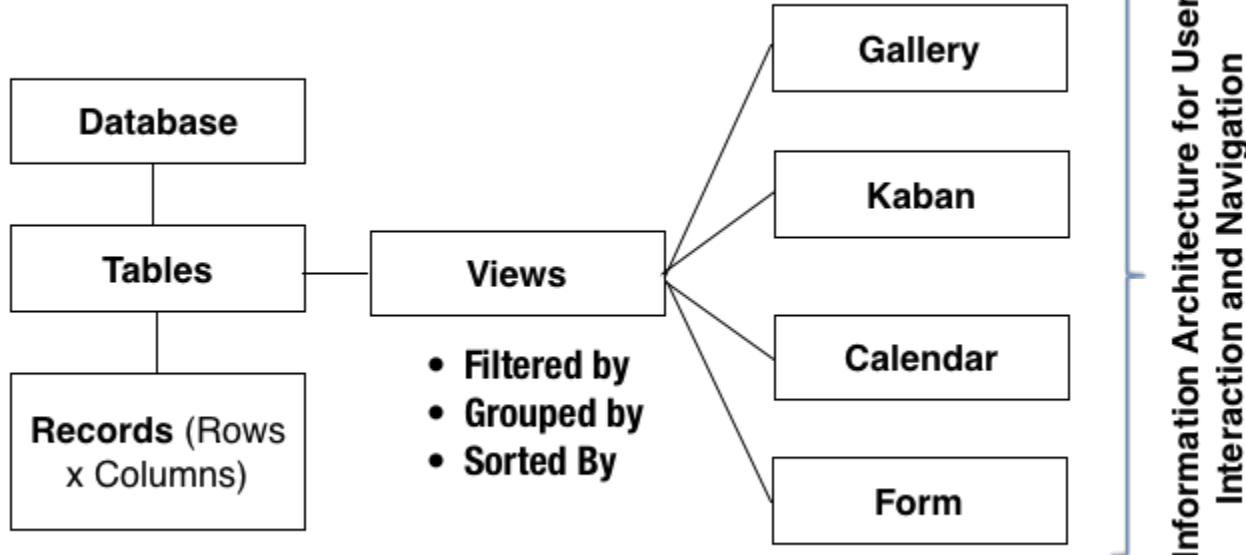
Part spreadsheet, part database, and entirely flexible, teams use Airtable to organize their work, their way.

Enter your email address

[Get started](#)

Already using Airtable? [Sign in](#)





Setting up MySQL on MAMP and Performing CRUD Operations Using SQL

phpMyAdmin

Recent | Favorites

Filter databases by name or

New

- acehub
- BucketList
- campus_prdev
- cellar
- ci
- com5940
- csdemo
- d734_e
- d739
- d744
- d744a
- d750a
- d751_a
- d753e
- d753f
- d753g
- d754_e

Server: localhost:3306

Databases SQL Status Users Export Import Settings Replication VariablesCharsets Engines

General Settings

Server connection collation: utf8mb4_unicode_ci

Appearance Settings

Language: English

Theme: Original

Font size: 82%

More settings

Database server

- Server: Localhost via UNIX socket
- Server type: MySQL
- Server version: 5.5.42 - Source distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

Web server

- Apache/2.2.29 (Unix) mod_wsgi/3.5 Python/2.7.10 PHP/5.6.10 mod_ssl/2.2.29 OpenSSL/1.0.2h DAV/2 mod_fastcgi/2.4.5 mod_perl/2.0.9 Perl/v5.22.0
- Database client version: libmysql - mysqldnd 5.0.11-dev - 20120503 - S: 3c686b6bbc30d36a3ec34fdd4b7b5b787fe5555 \$
- PHP extension: mysqli
- PHP version: 5.6.10

phpMyAdmin

- Version information: 4.4.10
- Documentation
- Wiki
- Official Homepage
- Contribute
- Get support
- List of changes

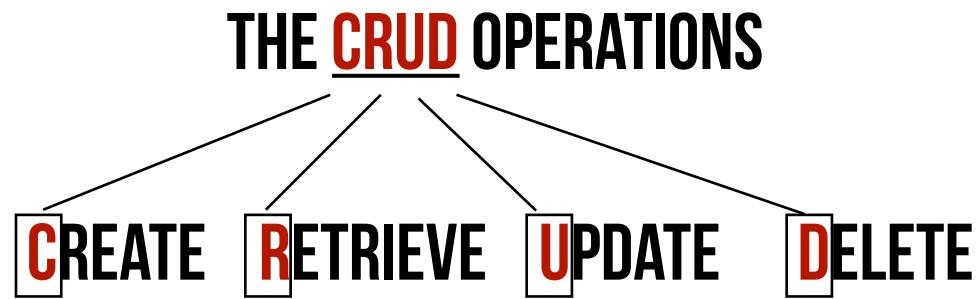


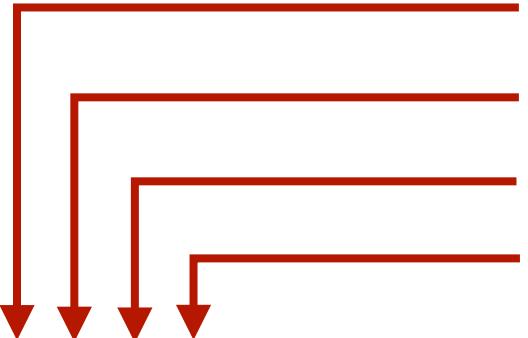
STRUCTURED QUERY LANGUAGE (SQL)

**DATA DEFINITION
LANGUAGE (DDL)**

**DATA MANIPULATION
LANGUAGE (DML)**

Data Manipulation Language (DML)

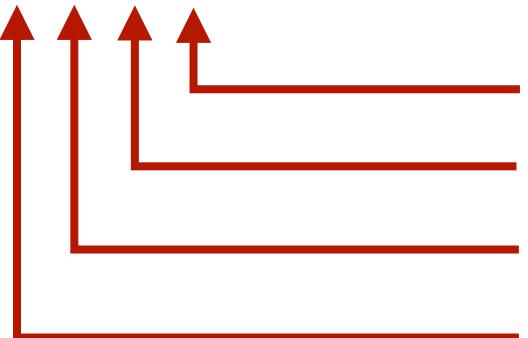




Create
Retrieve
Update
Delete

“where” filter can be appended to CRUD operations.

CRUD Operations in MySQL equivalent codes



delete from <table>
update <table> set <column> = ...
select * from <table>
insert into <table> values
(<columns>...)

GROUP BY Aggregate Functions in
MySQL:
COUNT, MAX, MIN, AVG, SUM, STD

**SELECT ProductCode, MAX(price), MIN(price),
AVG(price), STD(price), SUM(quantity) FROM
products GROUP BY ProductCode;**

Products

ProductCode	MAX(price)	MIN(price)	AVG(price)	STD(price)	SUM(quantity)
PEC	99999.99	0.47	14286.185714	34992.514069	19300
PEN	1.25	1.23	1.243333	0.009428	15000

Interacting with Airtable & MySQL in Jupyter Notebook

Server: localhost:3306 » Database: southwind » Table: products

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 9 (10 total, Query took 0.0002 seconds.)

```
SELECT * FROM `products`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP Code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

Products

	← ↑ →	↓	productID	productCode	name	quantity	price	supplierID				
<input type="checkbox"/>		Edit		Copy		Delete	1001	PEN	Pen Red	5000	1.23	501
<input type="checkbox"/>		Edit		Copy		Delete	1002	PEN	Pen Blue	8000	1.25	501
<input type="checkbox"/>		Edit		Copy		Delete	1003	PEN	Pen Black	2000	1.25	501
<input type="checkbox"/>		Edit		Copy		Delete	1004	PEC	Pencil 2B	10000	0.48	501
<input type="checkbox"/>		Edit		Copy		Delete	1005	PEC	Pencil 2H	8000	0.49	502
<input type="checkbox"/>		Edit		Copy		Delete	1006	PEC	Pencil HB	0	99999.99	501
<input type="checkbox"/>		Edit		Copy		Delete	2001	PEC	Pencil 3B	500	0.52	503
<input type="checkbox"/>		Edit		Copy		Delete	2002	PEC	Pencil 4B	200	0.62	501
<input type="checkbox"/>		Edit		Copy		Delete	2003	PEC	Pencil 5B	100	0.73	503
<input type="checkbox"/>		Edit		Copy		Delete	2004	PEC	Pencil 6B	500	0.47	502

Check All With selected: Edit Delete Export

Show all | Number of rows: 25 Filter rows: Search this table

Setup to Access MySQL

```
import pymysql.cursors
connection = pymysql.connect(host="localhost",
                             user="root",
                             passwd="root",
                             db="southwind",
                             charset='utf8mb4',
                             cursorclass=pymysql.cursors.DictCursor)
```

Loop through the data with the SQL “select” COMMAND

```
with connection.cursor() as cursor:  
    sql = "SELECT * FROM products"  
    cursor.execute(sql)  
    result = cursor.fetchall()  
    dataset = []  
    temp = {}  
    i = 0  
    for row in result:  
        temp[i] = row  
        dataset.append(temp[i])  
        i = i + 1  
    print('\n')  
    print(dataset)  
    print('\n')
```

southwind ▾

HELP ? 🔍 🔔



products + suppliers

SHARE 🔍 BLOCKS

Grid 2 Hide fields Filter Sort ...

	A productID	A productCode	A name	# quantity	\$ price	A supplierID	+
1	1001	PEN	Pen Red	5000.0	\$1.23	501	
2	1002	PEN	Pen Blue	8000.0	\$1.25	501	
3	1003	PEN	Pen Black	2000.0	\$1.25	501	
4	1004	PEC	Pencil 2B	10000.0	\$0.48	501	
5	1005	PEC	Pencil 2H	8000.0	\$0.49	502	
6	1006	PEC	Pencil HB	0.0	\$99,999.99	501	
7	2001	PEC	Pencil 3B	500.0	\$0.52	503	
8	2002	PEC	Pencil 4B	200.0	\$0.62	501	
9	2003	PEC	Pencil 5B	100.0	\$0.73	503	
10	2004	PEC	Pencil 6B	500.0	\$0.47	502	
+							
10 records				Max 10000.0	Sum \$100,007.03		

Airtable API for "southwind" ↗

curl | JavaScript | show API key

EXAMPLE USING BEARER TOKEN (RECOMMENDED)

```
$ curl https://api.airtable.com/v0/appVFlVS0NkWT41fM/api/docs#curl/authen... \n-H "Authorization: Bearer key"
```

EXAMPLE USING QUERY PARAMETER

```
$ curl https://api.airtable.com/v0/appVFlVS0NkWT41fM/api/docs#curl/authen... /products?api_key=key
```

INTRODUCTION

RATE LIMITS

AUTHENTICATION

PRODUCTS TABLE

Fields

List records

Retrieve a record

Create records

Update records

Delete records

SUPPLIERS TABLE

PRODUCT-TYPE TABLE

ERRORS

AUTHENTICATION

Airtable uses simple token-based authentication. To generate or manage your API key, visit your [account](#) page. **Your API key carries the same privileges as your user account, so be sure to keep it secret!**

You can authenticate to the API by providing your API key in the HTTP authorization bearer token header. Alternatively, a slightly lower-security approach is to provide your API key with the `api_key` query parameter.

All API requests must be authenticated and made over HTTPS.

PRODUCTS TABLE

Fields

Each record in the `products` table contains the following fields:

FIELD NAME	TYPE	DESCRIPTION
productID	Text	<code>string</code> A single line of text.

EXAMPLE VALUES

```
"1001"\n"1002"\n"1003"\n"1004"
```

Setup to Access Airtable

```
import requests

headers = {
    'Authorization': 'Bearer <API Key>',
}

params = (
    ('view', 'Main'),
)
```

Loop through the data returned by the Airtable REST API

```
r = requests.get('https://api.airtable.com/v0/<Base App Id>/products?api_key=<API  
Key>&sortField=_createdTime&sortDirection=desc', headers=headers, params=params)  
dict = r.json()  
dataset = []  
productID = []  
productCode = []  
name = []  
for i in dict['records']:  
    dict = i['fields']  
    dataset.append(dict)  
print(dataset)  
print('\n')  
for dicts in dataset:  
    productID.append(dicts.get('productID'))  
    productCode.append(dicts.get('productCode'))  
    name.append(dicts.get('name'))  
print(name)
```

Python for Web Development Using Flask

Running Flask in Local Mode Inside JupyterNotebook



```
In [ ]: 1 import sys  
2 !pip3 install Werkzeug
```

```
In [ ]: 1 import os  
2 from werkzeug.wsgi import SharedDataMiddleware  
3  
4 from flask import Flask  
5  
6 app = Flask(__name__)  
7  
8 @app.route('/')  
9 def hello_world():  
10     return 'Hello from Flask!'  
11  
12 if __name__ == '__main__':  
13     from werkzeug.serving import run_simple  
14     run_simple('localhost', 9002, app)
```

import os
from werkzeug.wsgi import SharedDataMiddleware

from flask import Flask

app = Flask(__name__)

@app.route('/)
def hello_world():
 return 'Hello from Flask!'

if __name__ == '__main__':
 from werkzeug.serving import run_simple
 run_simple('localhost', 9002, app)

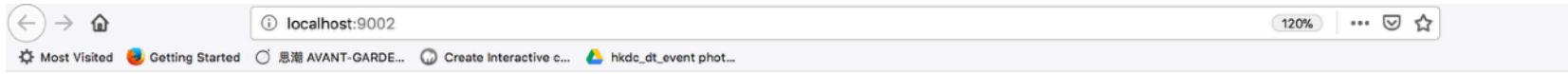
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In []: 1 `import sys`
2 `!pip3 install Werkzeug`

In [*]: 1 `import os`
2 `from werkzeug.wsgi import SharedDataMiddleware`
3
4 `from flask import Flask`
5
6 `app = Flask(__name__)`
7
8 `@app.route('/')`
9 `def hello_world():`
10 `return 'Hello from Flask!'`
11
12 `if __name__ == '__main__':`
13 `from werkzeug.serving import run_simple`
14 `run_simple('localhost', 9002, app)`

* Running on <http://localhost:9002/> (Press CTRL+C to quit)

In []: 1



Hello from Flask!



Bootstrap

Build responsive, mobile-first projects on the web with the world's most popular front-end component library.

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

[Get started](#)[Download](#)

Currently v4.3.1

Power-up the lifetime value of your app users with rewarded video ads via Carbon

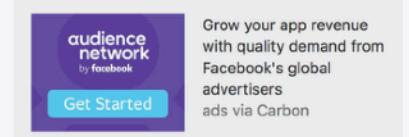
[Get Started](#)



Examples

Quickly get a project started with any of our examples ranging from using parts of the framework to custom components and layouts.

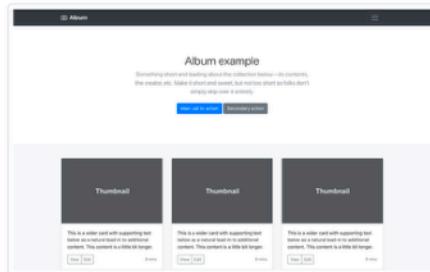
[Download source code](#)



Grow your app revenue with quality demand from Facebook's global advertisers ads via Carbon

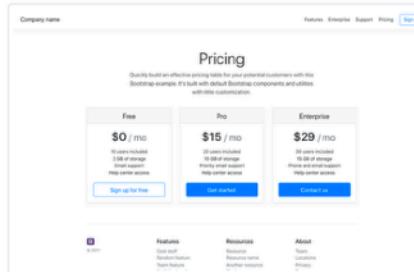
Custom components

Brand new components and templates to help folks quickly get started with Bootstrap and demonstrate best practices for adding onto the framework.



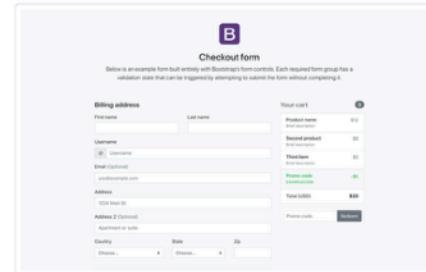
Album

Simple one-page template for photo galleries, portfolios, and more.



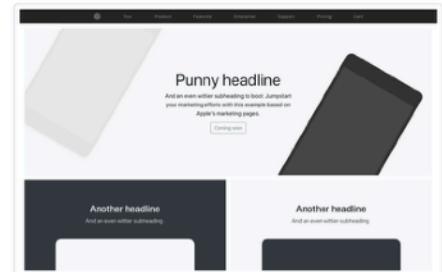
Pricing

Example pricing page built with Cards and featuring a custom header and footer.



Checkout

Custom checkout form showing our form components and their validation features.



Product

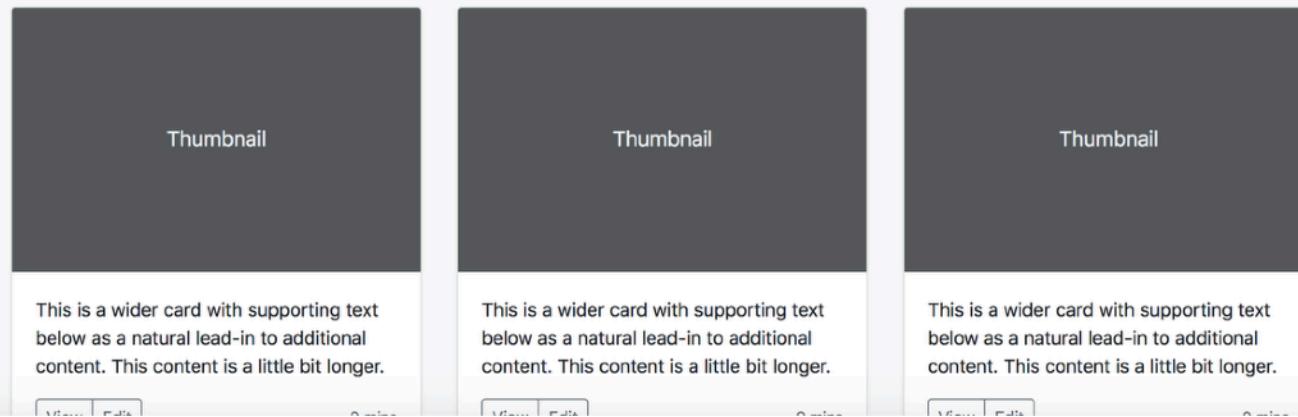
Lean product-focused marketing page with extensive grid and image work.

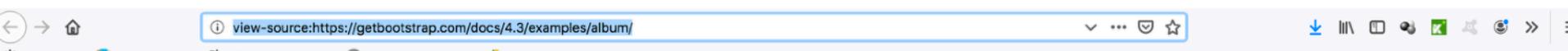
Album example

Something short and leading about the collection below—its contents, the creator, etc. Make it short and sweet, but not too short so folks don't simply skip over it entirely.

Main call to action

Secondary action





```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
6     <meta name="description" content="">
7     <meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap contributors">
8     <meta name="generator" content="Jekyll v3.8.5">
9     <title>Album example · Bootstrap</title>
10
11    <link rel="canonical" href="https://getbootstrap.com/docs/4.3/examples/album/">
12
13    <!-- Bootstrap core CSS -->
14    <link href="/docs/4.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MzW1T" crossorigin="anonymous">
15
16
17    <style>
18      .bd-placeholder-img {
19        font-size: 1.125rem;
20        text-anchor: middle;
21        -webkit-user-select: none;
22        -moz-user-select: none;
23        -ms-user-select: none;
24        user-select: none;
25      }
26
27      @media (min-width: 768px) {
28        .bd-placeholder-img-lg {
29          font-size: 3.5rem;
30        }
31      }
32    </style>
33    <!-- Custom styles for this template -->
34    <link href="album.css" rel="stylesheet">
35  </head>
36  <body>
37    <header>
38      <div class="collapse bg-dark" id="navbarHeader">
39        <div class="container">
40          <div class="row">
41            <div class="col-sm-8 col-md-7 py-4">
42              <h4 class="text-white">About</h4>
43              <p class="text-muted">Add some information about the album below, the author, or any other background context. Make it a few sentences long so folks can pick up some informative tidbits. Then, l...
44            </div>
45            <div class="col-sm-4 offset-md-1 py-4">
46              <h4 class="text-white">Contact</h4>
47              <ul class="list-unstyled">
48                <li><a href="#" class="text-white">Follow on Twitter</a></li>
49                <li><a href="#" class="text-white">Like on Facebook</a></li>
50                <li><a href="#" class="text-white">Email me</a></li>
51              </ul>
52            </div>
53          </div>
54        </div>
55      </div>
56      <div class="navbar navbar-dark bg-dark shadow-sm">
57        <div class="container d-flex justify-content-between">
58          <div href="#" class="navbar-brand d-flex align-items-center">
59            <img alt="Bootstrap logo" data-bbox="10 10 30 30" style="vertical-align: middle;" />
60            <strong>Album</strong>
61          </div>
62          <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarHeader" aria-controls="navbarHeader" aria-expanded="false" aria-label="Toggle navigation">
63            <span class="navbar-toggler-icon"></span>
64          </button>
65        </div>
66      </div>
67    </header>
68    <div class="container py-4">
69      <div class="row justify-content-between">
70        <div class="col-6">
71          <h2>Album Example</h2>
72          <img alt="Placeholder image for the album" data-bbox="10 10 30 30" style="width: 100%; height: auto; border-radius: 10px; margin-bottom: 10px;" />
73          <h3>Album Description</h3>
74          <p>This is a detailed description of the album, including its title, artist, release date, and a brief summary of its content. It's designed to be informative and engaging for potential listeners.</p>
75        </div>
76        <div class="col-6">
77          <h2>Song List</h2>
78          <table border="1" style="width: 100%; border-collapse: collapse; text-align: center; margin-bottom: 10px;">
79            <thead>
80              <tr>
81                <th>Song Name</th>
82                <th>Length</th>
83                <th>Album Version</th>
84              </tr>
85            </thead>
86            <tbody>
87              <tr>
88                <td>Song 1</td>
89                <td>3:45</td>
90                <td>Standard Version</td>
91              </tr>
92              <tr>
93                <td>Song 2</td>
94                <td>4:20</td>
95                <td>Deluxe Version</td>
96              </tr>
97              <tr>
98                <td>Song 3</td>
99                <td>2:59</td>
100               <td>Digital Version</td>
101             </tr>
102           </tbody>
103         </table>
104         <ul style="list-style-type: none; padding-left: 0; margin: 0; margin-top: 10px;">
105           <li><a href="#">Song 1</a></li>
106           <li><a href="#">Song 2</a></li>
107           <li><a href="#">Song 3</a></li>
108         </ul>
109       </div>
110     </div>
111   </div>
112   <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+9jwhV3D2O+RDuO8eRTSRQ8tWnwP" crossorigin="anonymous"></script>
113   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UOJZtCnqP1KdMk8A9q0JzjE3d7SNG0Kwq4P6bWVnJNz55IaVU9B6Iw" crossorigin="anonymous"></script>
114   <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQq/S8eECDifZVcW3FwKzZJg8hJ" crossorigin="anonymous"></script>
115 
```

```
1 import os
2 from werkzeug.wsgi import SharedDataMiddleware
3
4 from flask import Flask, render_template
5
6 app = Flask(__name__)
7
8 # The following line is only needed for running insdie Jupyter Notebook
9 root_path = os.path.sep.join(app.instance_path.split(os.path.sep)[:-1])
10
11 @app.route("/")
12 def home():
13     return render_template('myhome.html')
14
15 if __name__ == '__main__':
16     from werkzeug.serving import run_simple
17     app.wsgi_app = SharedDataMiddleware(app.wsgi_app,
18         '/static': root_path+'/flask_apps/static',
19         '/templates': root_path+'/flask_apps/templates'
20     )
21     run_simple('localhost', 9003, app)
```

```
import os
from werkzeug.wsgi import SharedDataMiddleware

from flask import Flask, render_template

app = Flask(__name__)

# The following line is only needed for running insdie Jupyter Notebook
root_path = os.path.sep.join(app.instance_path.split(os.path.sep)[:-1])

@app.route("/")
def home():

    return render_template('myhome.html')

if __name__ == '__main__':
    from werkzeug.serving import run_simple
    app.wsgi_app = SharedDataMiddleware(app.wsgi_app, {
        '/static': root_path+'/flask_apps/static',
        '/templates': root_path+'/flask_apps/templates'
    })
    run_simple('localhost', 9003, app)
```

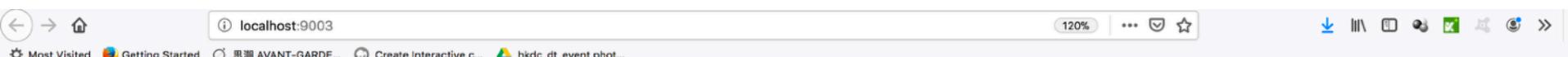
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [*]:

```
1 import os
2 from werkzeug.wsgi import SharedDataMiddleware
3
4 from flask import Flask, render_template
5
6 app = Flask(__name__)
7
8 # The following line is only needed for running inside Jupyter Notebook
9 root_path = os.path.sep.join(app.instance_path.split(os.path.sep)[:-1])
10
11 @app.route("/")
12 def home():
13     return render_template('myhome.html')
14
15 if __name__ == '__main__':
16     from werkzeug.serving import run_simple
17     app.wsgi_app = SharedDataMiddleware(app.wsgi_app, {
18         '/static': root_path+'/flask_apps/static',
19         '/templates': root_path+'/flask_apps/templates'
20     })
21     run_simple('localhost', 9003, app)
```

* Running on <http://localhost:9003/> (Press CTRL+C to quit)

In []: 1



About

Add some information about the album below, the author, or any other background context. Make it a few sentences long so folks can pick up some informative tidbits. Then, link them off to some social networking sites or contact information.

Contact

- [Follow on Twitter](#)
- [Like on Facebook](#)
- [Email me](#)

[Album](#)

Album example

Something short and leading about the collection below—its contents, the creator, etc. Make it short and sweet, but not too short so folks don't simply skip over it entirely.

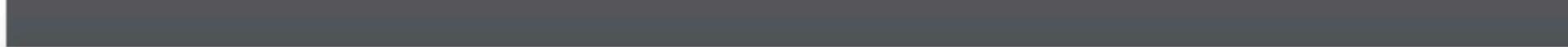
[Main call to action](#) [Secondary action](#)



Thumbnail

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

[View](#) [Edit](#)
9 mins



A screenshot of a web browser window titled "Album". The address bar shows "localhost:9003". The page content includes a title "Album example", a descriptive text block, and three dark gray rectangular placeholders labeled "Thumbnail". At the bottom, there are two buttons: a blue one labeled "Main call to action" and a gray one labeled "Secondary action". The browser interface includes standard navigation buttons (back, forward, home), a search bar, and a menu icon.

Album example

Something short and leading about the collection below—its contents, the creator, etc. Make it short and sweet, but not too short so folks don't simply skip over it entirely.

Main call to action

Secondary action

Thumbnail

Thumbnail

Thumbnail

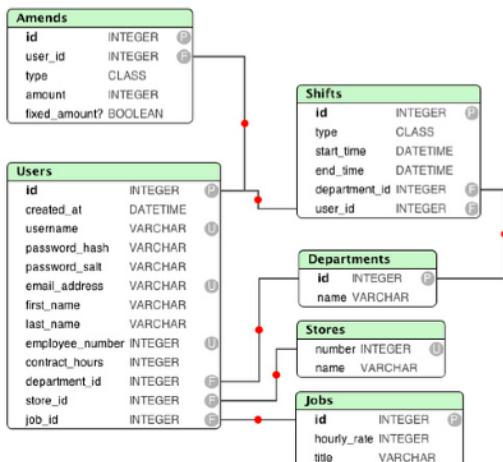
- ▶ **Web template development in Python: Introducing Flask Jinja2 (i.e. render_template library)**
- ▶ **Retrieving information from an Airtable base into a Python Flask program through the Airtable API**

MVC Framework Using the Python Flask

Source: commons.wikimedia.org



Source: [Caius Durling \(Flickr\)](https://flic.kr)



M(odel)
(e.g. Python/SQL functions)

V(iew) (Flask templates)

Source: pexels.com



**Routing control to
pre-defined functions**

C(ontroller)
(e.g. Routes)

A Flask Demonstration



- Home
- Dashboard
- List Users
- Add User
- Logout
- Login

Front-End

This website is built on Bootstrap with data visualization support provided by [jQuery dataTable](#), [C3js](#) and [Leaflet](#).

To try out the demo, please use "[demo](#)" for username and "[password](#)" for password to login.

Airtable Data Source

All the data used in the demonstration can be found in this Airtable database. The chart receives its data from the "Roll-up" table and the map from the "Venues" table. You can try add, update, or delete users in the "Users" table using the list and add users options on the menu. The student id and password created can be used to login to this demo site.

The dataset was originally derived from a tutorial by d3-media on learning Refine for data cleaning and transformation. The original link is [here](#).

Product							Roll-up	Venues	Users
	Grid view	Hide fields	Filter	Group	Sort	...			
	company	product_code	Roll-up	product_no	geocode2	json			
14	Phillips	radio	radio	56	Delfzijlstraat 54, ar...	http://maps.googleapis.com/maps/api/geocode/json?address=Delfzijlstraat+54,+Armenia+1701+NL&sensor=false			
15	Phillips	tv	tv	67	Delfzijlstraat 55, ar...	http://maps.googleapis.com/maps/api/geocode/json?address=Delfzijlstraat+55,+Armenia+1701+NL&sensor=false			
16	Phillips	tv	tv	21	Delfzijlstraat 56, ar...	http://maps.googleapis.com/maps/api/geocode/json?address=Delfzijlstraat+56,+Armenia+1701+NL&sensor=false			

<https://python30.kilo.ws/>

URL Routing in Flask

```
from flask import Flask, render_template
@app.route('/')
def main():
    :
    return render_template('index.html')

@app.route('/about_us')
def about_us():
    :
    return render_template('aboutus.html')
```

Python Code (control flow and
data model)



HTML Templates (view)

```
<!DOCTYPE html>
<head>
<meta charset="utf8">
<title>Main Page</title> </head>
<body>
<h1>This is the main page</h1>
    :
    :
</body> </html>
```

index.html

```
<!DOCTYPE html>
<head>
<meta charset="utf8">
<title>About Us</title> </head>
<body>
<h1>This is the about us page</h1>
    :
    :
</body> </html>
```

aboutus.html

**Template Tags
(i.e. comments, expressions,
statements)**

- **{#... comment#}** : template tag for comments
- **{{... expression...}}** : template tag for expressions
- **{%...statement...%}** : template tag for statements

Passing Values to Template Variables

```
from flask import Flask, render_template  
app = Flask(__name__)  
  
@app.route("/")  
def index():  
    user = {"name": "Bernard"}  
    return render_template("result.html", user=user, title="Home Page")
```

code variable and value

template file

template variables

Passing values to template variables.

template variables

```
44-
45   <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarHeader" aria-controls="navbarHeader" aria-expanded="false"
46     <span class="navbar-toggler-icon"></span>
47   </button>
48 </div>
49 </div>
50 </header>

51-
52 <main role="main">
53   <section class="jumbotron text-center">
54     <div class="container">
55       <!--h1 class="jumbotron-heading">Album example</h1-->
56       <h1 class="jumbotron-heading">{{user.name}}'s {{title}}</h1>
57       <p class="lead text-muted">Something short and leading about the collection below its contents, the creator, etc. Make it short and sweet, but n
58       <p>
59         <a href="https://getbootstrap.com/docs/4.0/examples/album/#" class="btn btn-primary my-2">Main call to action</a>
60         <a href="https://getbootstrap.com/docs/4.0/examples/album/#" class="btn btn-secondary my-2">Secondary action</a>
61       </p>
62     </div>
63   </section>
64
65   <div class="album py-5 bg-light">
66     <div class="container">
67
68       <div class="row">
69         <div class="col-md-4">
70           <div class="card mb-4 box-shadow">
71             
73               <p class="card-text">This is a wider card with supporting text below as a natural lead-in to additional content. This content is a littl
74               <div class="d-flex justify-content-between align-items-center">
75                 <div class="btn-group">
76                   <button type="button" class="btn btn-sm btn-outline-secondary">View</button>
77                   <button type="button" class="btn btn-sm btn-outline-secondary">Edit</button>
78                 </div>
79                 <small class="text-muted">9 mins</small>
80               </div>
81             </div>
82           </div>
83         </div>
84       </div>
```

Bernard's Home Page

Something short and leading about the collection below—its contents, the creator, etc. Make it short and sweet, but not too short so folks don't simply skip over it entirely.

Main call to action Secondary action

Thumbnail

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

[View](#) [Edit](#) 9 mins

Thumbnail

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

[View](#) [Edit](#) 9 mins

Thumbnail

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

[View](#) [Edit](#) 9 mins

Thumbnail

Thumbnail

Thumbnail

The Lean Startup Approach

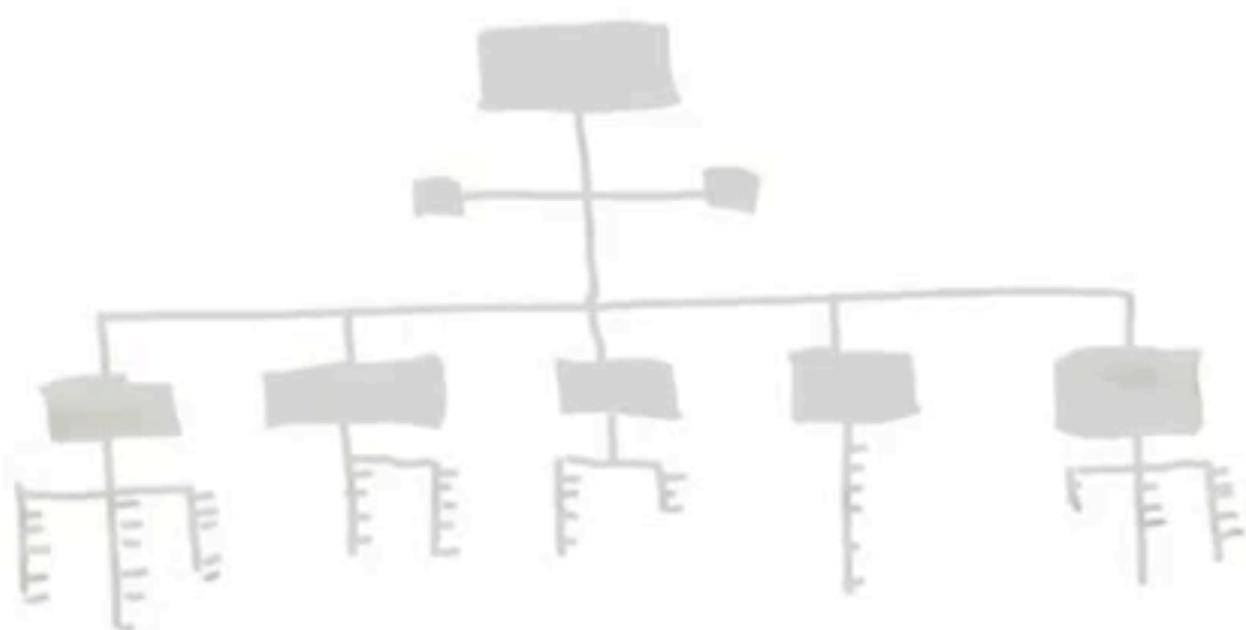
Lean StartUp 精益创业



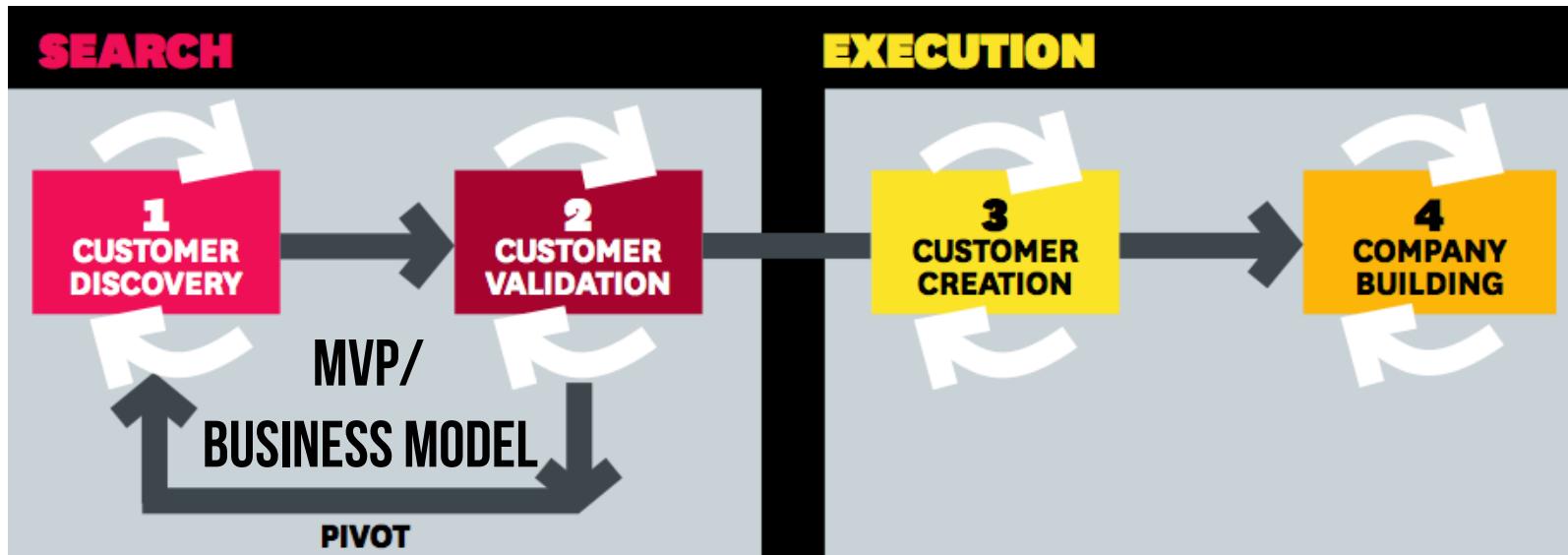
THE LEAN APPROACH: THE MINIMUM VIABLE PRODUCT

What Distinguishes a Startup from a Corporation?



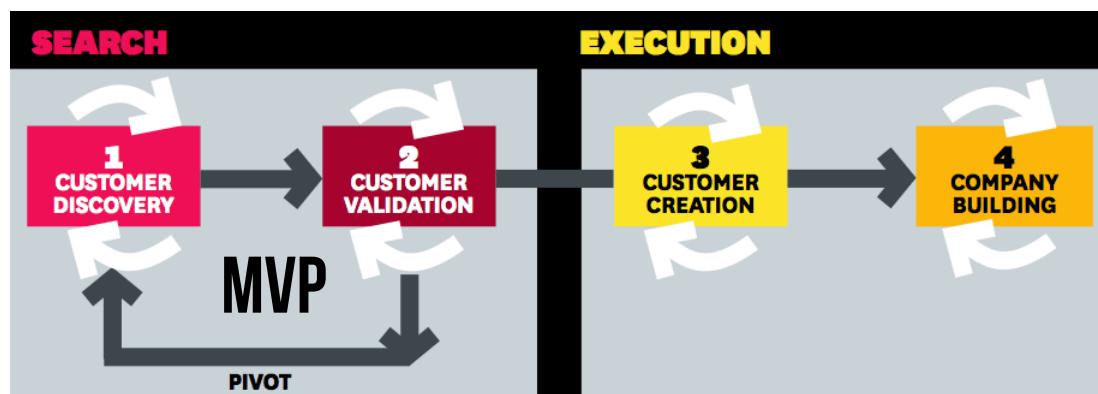
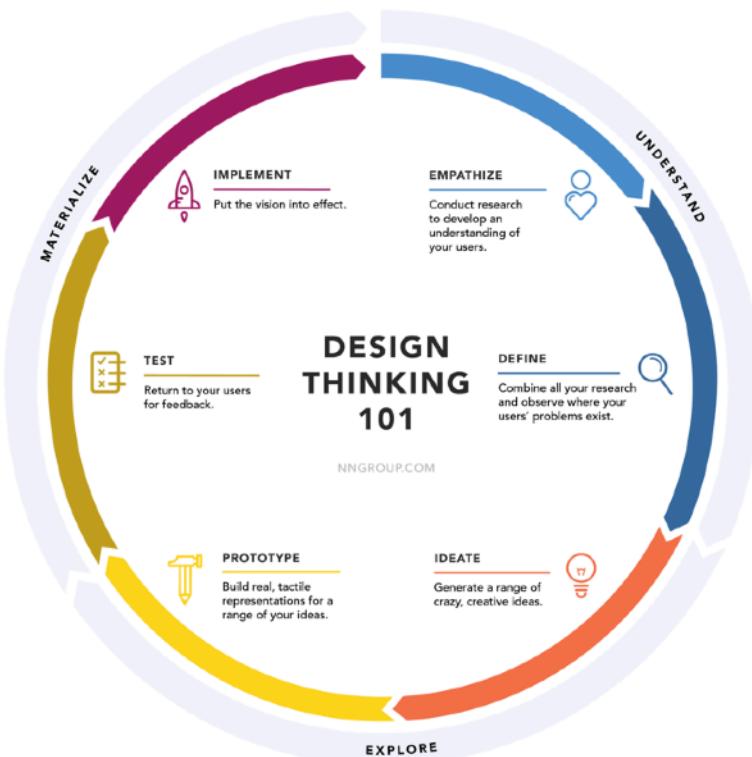


BUILD-MEASURE-LEARN CYCLE IN SEARCH OF PRODUCT-MARKET FIT



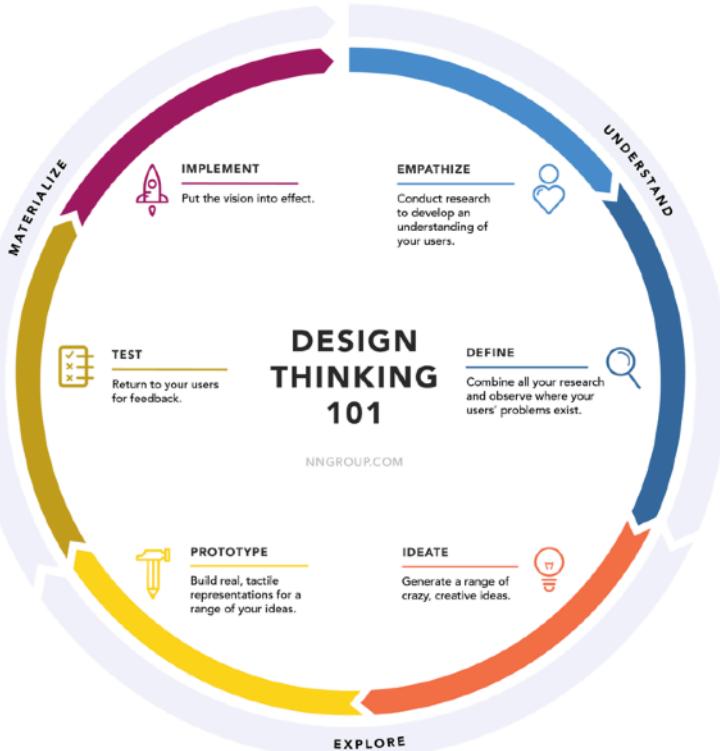
Source: “Why Lean Start-up Changes Everything” by Steve Blank, Harvard Business Review May 2013

Can Design Thinking and the Lean Method Work Together?



Source: “Why Lean Start-up Changes Everything” by Steve Blank, Harvard Business Review May 2013

Source: Sarah Gibbons,
Design Thinking 101
Nielsen Norman Group



USE DESIGN THINKING TO SUPPLEMENT MVP ITERATION



Source: “Why Lean Start-up Changes Everything” by Steve Blank, Harvard Business Review May 2013

Source: Sarah Gibbons,
Design Thinking 101
Nielsen Norman Group

Assignment #1

1. Create a MySQL database using phpMyAdmin and set up tables with data derived from your final com5961 project
2. Create a Jupyter notebook with Python codes demonstrating connection to the MySQL database and Airtable (through API)
3. Retrieve data from the MySQL tables and Airtable API to be displayed as output in the notebook.
4. Export the notebook as .ipynb Jupyter Notebook document and upload to Github.
5. Export the MySQL database as .sql file and upload to Github.
6. Setup Github with a folder called com5940 with 5 sub-folders, namely assignment 1 to assignment 5. Put the .ipynb and .sql files into the assignment#1 folder.

THANK YOU FOR YOUR TIME!