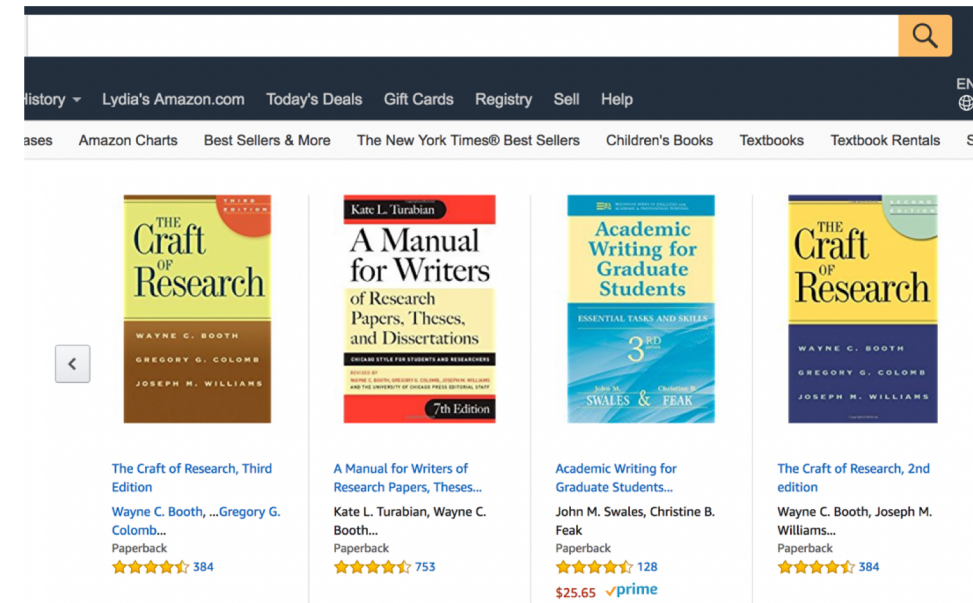
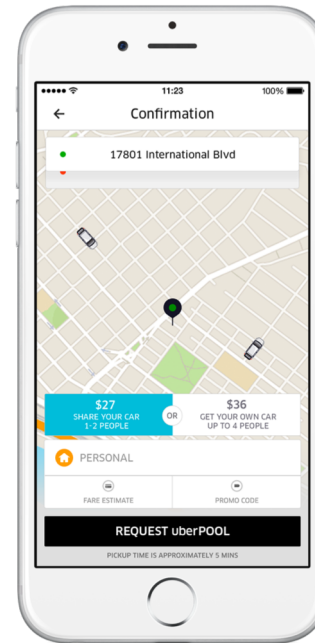
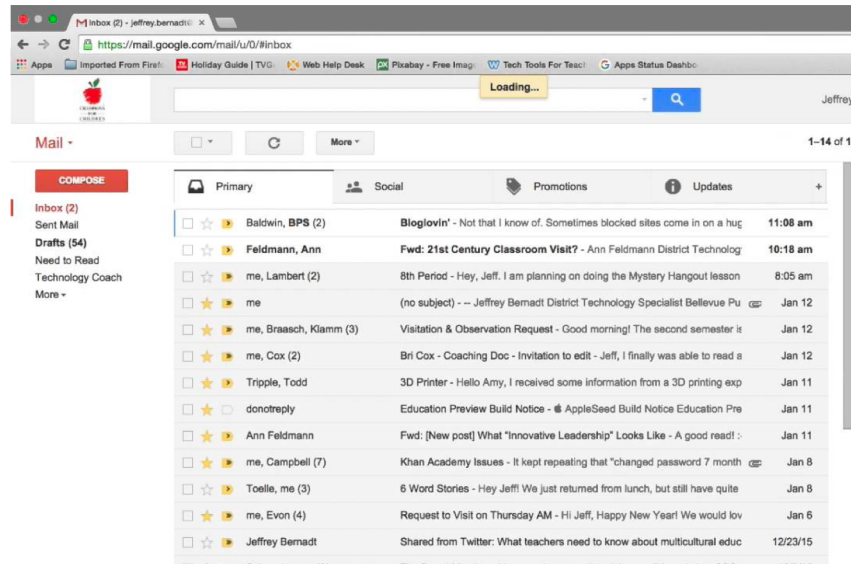


# Database-Backed Websites

Prof. Lydia Chilton  
COMS 4170  
21 February 2024

# The main goal of many websites is to interact with data.



We need servers to store the data.  
Clients request data from servers (and display it)

```
people — Python · Python server.py — 80x24
Last login: Sun Feb 16 09:18:51 on ttys001
Lydias-MacBook-Pro:people lydiachilton$ python server.py
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 723-907-492
127.0.0.1 - - [19/Feb/2020 07:07:16] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2020 07:07:17] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [19/Feb/2020 07:07:46] "GET /people HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2020 07:07:46] "GET /static/people.js HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2020 07:07:56] "POST /add_name HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2020 07:08:01] "GET /people HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2020 07:09:25] "POST /add_name HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2020 07:09:28] "GET /people HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2020 07:26:36] "GET /people HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2020 07:26:38] "GET /people HTTP/1.1" 200 -
^CLydias-MacBook-Pro:people lydiachilton$ python server.py
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 723-907-492
127.0.0.1 - - [19/Feb/2020 07:26:46] "GET /people HTTP/1.1" 200 -
█
```

```
server.py
1 from flask
2 from flask
3 from flask
4 app = Flask
5
6
7 current_id
8 data = [
9     {
10         "id": 1,
11         "name": "michael scott"
12     },
13     {
14         "id": 2,
15         "name": "jim halpert"
16     },
17 ]
```

**Server:  
store the data**

127.0.0.1:5000/people

Reload this page

Hello people:

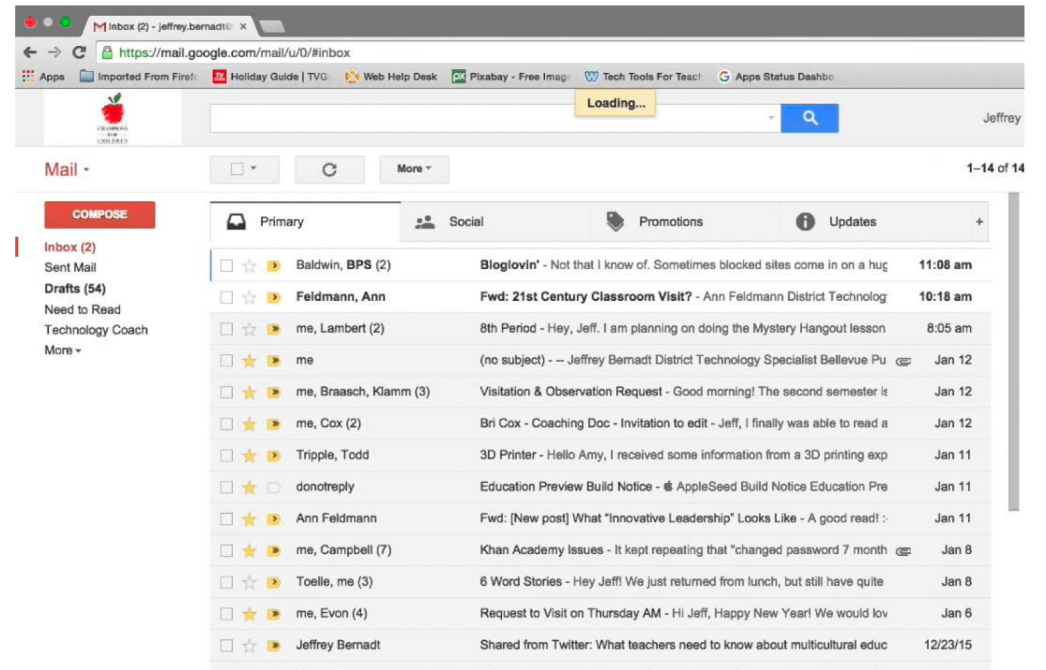
name:  Submit

michael scott  
jim halpert

**Client:  
requests data from server**

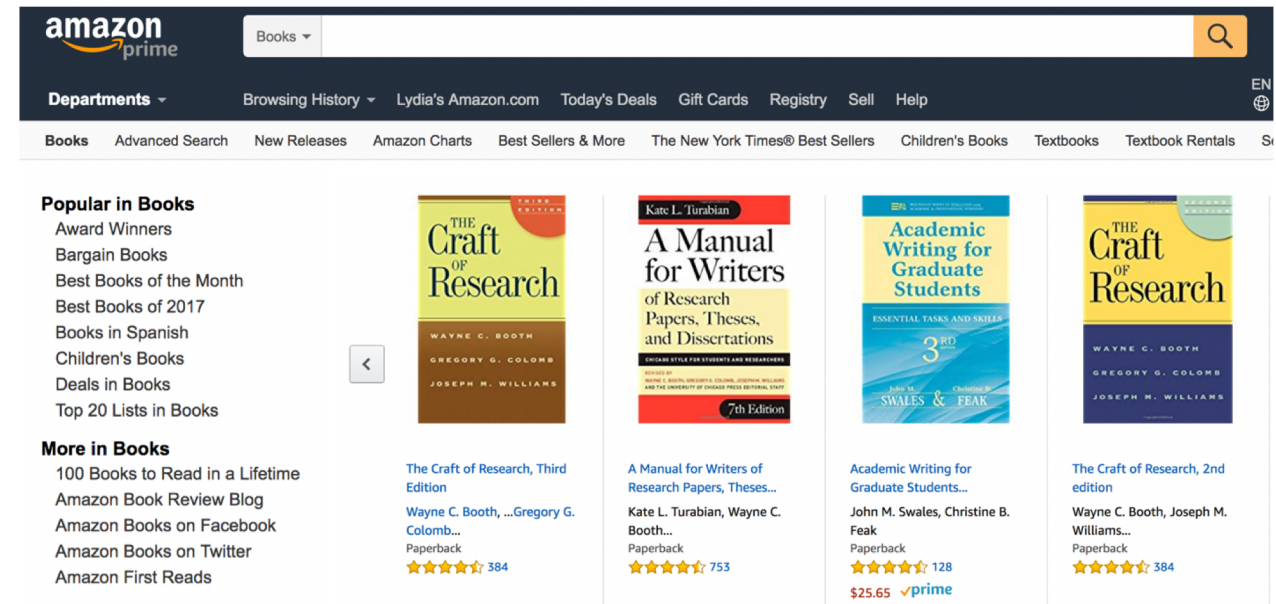
# Gmail is a database of emails

```
emails = [  
  {  
    "from": "bollinger",  
    "to": "chilton",  
    "subject": "4170 is awesome!"  
  },  
  {  
    "from": "obama",  
    "to": "chilton",  
    "subject": "belated medal of freedom"  
  }  
]
```



# Amazon is a database of books

```
products = [  
  {  
    "title": "Ivy League Web Design",  
    "author": "chilton",  
    "stars": "5"  
  },  
  {  
    "title": "JavaScript and You",  
    "author": "chilton",  
    "stars": "6"  
  },  
]
```



The screenshot shows the Amazon Books homepage. At the top, there is a search bar with the word "Books" and a magnifying glass icon. Below the search bar, there are navigation links for "Departments", "Browsing History", "Lydia's Amazon.com", "Today's Deals", "Gift Cards", "Registry", "Sell", and "Help". A secondary navigation bar includes "Books", "Advanced Search", "New Releases", "Amazon Charts", "Best Sellers & More", "The New York Times® Best Sellers", "Children's Books", "Textbooks", and "Textbook Rentals".

The main content area is divided into two sections: "Popular in Books" and "More in Books".

**Popular in Books**

- Award Winners
- Bargain Books
- Best Books of the Month
- Best Books of 2017
- Books in Spanish
- Children's Books
- Deals in Books
- Top 20 Lists in Books

**More in Books**

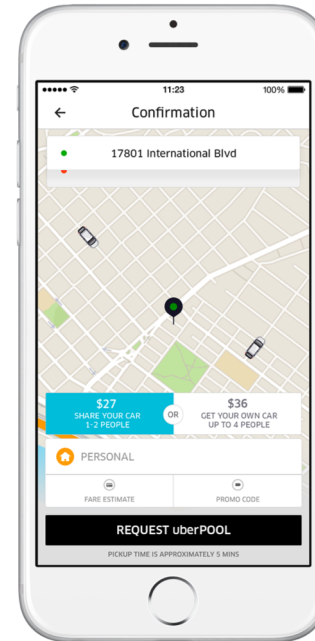
- 100 Books to Read in a Lifetime
- Amazon Book Review Blog
- Amazon Books on Facebook
- Amazon Books on Twitter
- Amazon First Reads

Four book covers are displayed in a row:

- The Craft of Research, Third Edition** by Wayne C. Booth, Gregory G. Colomb, and Joseph M. Williams. It has a 4.5-star rating and 384 reviews.
- A Manual for Writers of Research Papers, Theses, and Dissertations, 7th Edition** by Kate L. Turabian and Wayne C. Booth. It has a 4.5-star rating and 753 reviews.
- Academic Writing for Graduate Students, 3rd Edition** by John M. Swales and Christine B. Feak. It has a 4.5-star rating and 128 reviews, and is available for \$25.65 with Prime.
- The Craft of Research, 2nd edition** by Wayne C. Booth and Joseph M. Williams. It has a 4.5-star rating and 384 reviews.

# Uber is a database of drivers.

```
cars = [  
  {  
    "location": "116 and broadway",  
    "driver": "kenny",  
    "car type": "uber XL"  
  },  
  {  
    "location": "times square",  
    "driver": "jen",  
    "car type": "normal"  
  },  
]
```



# Tinder is a database of profiles

```
profiles = [  
  {  
    "name": "maddy",  
    "image": "./maddy.png",  
    "likes": "1000",  
    "dislikes": 0,  
  },  
  {  
    "name": "julia",  
    "image": "./julia.png",  
    "likes": "1000",  
    "dislikes": 0,  
  },  
]
```



# YouTube: Database of videos



windows 95 jennifer aniston matthew perry



Windows 95 Video Guide with Matthew Perry & Jennifer Aniston

216,782 views

2K 48 SHARE



**Favorite Videos**  
Published on Dec 12, 2016

SUBSCRIBE 201K

Watch all about the show Friends in a single playlist - <https://www.youtube.com/playlist?list...>  
Subscribe and watch scenes from all seasons of Friends on my channel - <https://www.youtube.com/channel/UCA8V...>

SHOW MORE

Up next

AUTOPLAY



**Friends - The One that Goes Behind the Scenes**  
Favorite Videos  
1.8M views



**Would I Lie to You? - S11 E08 (NEW WILTY 19th January)**  
David Mitchell  
Recommended for you  
New



**Actor Replacements That Totally Ruined The Show**  
Looper  
3M views



**Inside Matthew Perry's Tragic Real Life Story**  
Nicki Swift  
952K views



**Friends - ET On VH1: Behind The Friends Finale**  
Favorite Videos  
1M views



**REACT S7 · E19 TEENS REACT TO WINDOWS 95**  
FBE  
12M views



**Friends - Ross and Monica's Cousin**  
Favorite Videos  
3.6M views



# Facebook: Database of posts

facebook



**PictureCorrect Photography Tips**

What is your Photography-Related New Year's Resolution??

15 minutes ago · Like · Comment

Adel Gomaa and Richard Schneider like this.



**Sadi Molloy** get my business looking more professional: car signed: tick, making up portrait folders: tick, stickers & cd covers: still to do... its getting there though.

11 minutes ago · Flag



**PictureCorrect Photography Tips** My resolution for 2011 is to capture more landscape photos outside the U.S. and add more countries to my portfolio! What is yours?

-Richard

11 minutes ago · Flag



**James Donnelly** Larger concerts to cover and improve my portrait editing style/skill

10 minutes ago · Flag



**Jose Antonio Rubido** To push my photpography foward full throttle

9 minutes ago · Flag



**Deacon Jonathan Jerome Johnson** To learn more and enhance the quality of my shots!

5 minutes ago · Flag



**Latu Peti Vimahi Manulua-hafoka** learn learn learn!!!

4 minutes ago · Flag



**Tina Taylor Hammer** Get my new business moving and have it pay for that lens and other equipment that I want (and a little for me too).

2 minutes ago · Flag

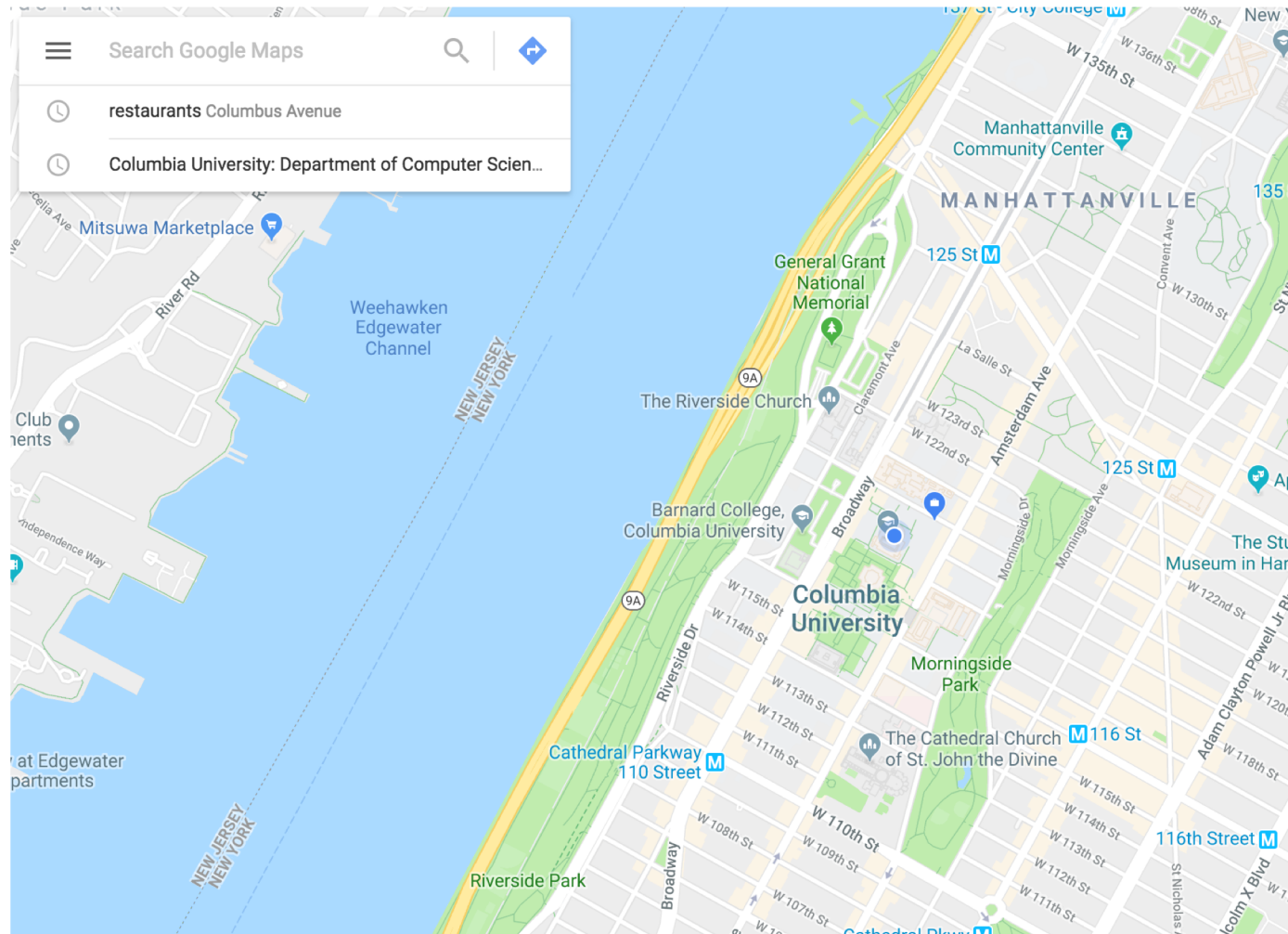


**Stefanie Bakewell** get a website up so that i can get clients full steam ahead

about a minute ago · Flag



# Google Maps: A database of locations



# New York Times: Database of news articles

SECTIONS SEARCH

hmslydia

ENGLISH 中文 (CHINESE) ESPAÑOL

## The New York Times

Friday, January 19, 2018 Today's Paper Video 34°F S. & P. 500 +0.19%

World U.S. Politics N.Y. Business Opinion Tech Science Health Sports Arts Style Food Travel Magazine T Magazine Real Estate ALL

### Trump in Shutdown Talks With Schumer

#### White House Invites Democratic Leader to Negotiate a Deal

By THE NEW YORK TIMES  
49 minutes ago

- The government will shut down at 12:01 a.m. Saturday if lawmakers fail to get an agreement on some kind of spending bill.
- With Democrats balking, President Trump canceled plans to travel to his Florida resort and contacted Senator Chuck Schumer, the Democratic leader.
- **Timeline: How Congress Got to the Brink of a Shutdown**
- **Right and Left React to Prospect of Shutdown**



“We are not going to desert these young people,” he said, referring to DACA beneficiaries.

Senator Bernie Sanders of Vt.

#### How the Debate in Congress Breaks Down

By ALICIA PARLAPIANO and WILSON ANDREWS  
Lawmakers have until Friday at midnight to pass a short-term spending bill to avoid a government shutdown.

#### Senate to Vote With Shutdown at Stake

By THOMAS KAPLAN and SHERYL GAY STOLBERG 44 minutes ago  
Senate Democrats appear ready to block legislation to keep the government funded past midnight, gambling that Mr. Trump will offer concessions on immigration and other issues.

574 Comments

#### The Chaos President vs. His Iron-Fisted Chief of Staff

By MAGGIE HABERMAN and JULIE

VIDEO INTERVIEW

#### Opinion

EDITORIAL

#### Here's Another Fine Mess They've Gotten Us Into

Once again, Trump and Congress cannot perform a basic task, preventing a government shutdown.



- **Krugman: Doomsday Politics**
- **The Breakdown in Trust That Could Lead to a Shutdown**
- **Brooks: It's Time to Talk About the Power of Touch**
- **Goldberg: Is This the Collusion We Were Waiting For?**
- **Kristof: Swallowed by the Sea**
- **Leonhardt: A Warning for Democrats**
- **Egan: The Mad King**
- **The Shadow Safety Net**

FRANK BRUNI

#### Donald Trump's Radical Honesty

In his own wild way, this president is as truthful and transparent as they come.



#### Clubbable, but in the Worst Way

By MICHAEL GOLDFARB

Donald Trump is a man of his class — the nouveau-riche, country-club class.



- **How the Pro-Life Movement Has Promoted Liberal Values**
- **ICE Detained My Husband for Being an Activist**
- **Follow us on Twitter »**

TIMES INSIDER »

‘I, Figure Skating Reporter’: Full Circle With Tonya Harding

THE CROSSWORD »

Play Today's Puzzle



33m The Justice Department wants to ask if you are a U.S. citizen in the 2020 census. But public distrust of the government could lead Hispanic people to avoid being counted.






5h Can software predict crime? A study suggests that human amateurs are about as accurate as a popular software that many judges use to




# Craigslist: Database of random crap for sale

CL new york > brooklyn > for sale > furniture - by owner

reply   prohibited  Posted 12 days ago [◀ prev](#) [▲](#) [next ▶](#)


☆ **Gorgeous new 6ft. tall throne chair - \$1199 (Brooklyn)** 



condition: new

Gorgeous new 6ft. tall throne chair for \$1199obo. I will deliver it to you!

- do NOT contact me with unsolicited services or offers

post id: 6490476659    posted: 12 days ago    updated: 34 minutes ago    [email to friend](#)    [♥ best of](#) 

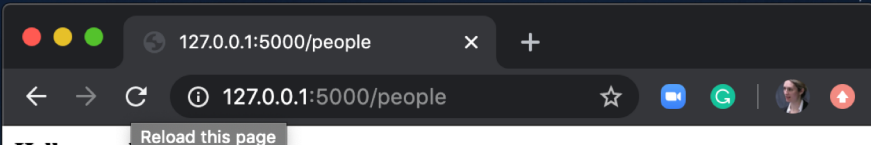
- [safety tips](#)
- [prohibited items](#)
- [product recalls](#)
- [avoiding scams](#)

We need to have another computer store and serve the data.  
That server is running a Python application called Flask.

```
people — Python · Python server.py — 80x24
Last login: Sun Feb 16 09:18:51 on ttys001
Lydias-MacBook-Pro:people lydiachilton$ python server.py
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 723-907-492
127.0.0.1 - - [19/Feb/2020 07:07:16] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2020 07:07:17] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [19/Feb/2020 07:07:46] "GET /people HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2020 07:07:46] "GET /static/people.js HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2020 07:07:56] "POST /add_name HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2020 07:08:01] "GET /people HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2020 07:09:25] "POST /add_name HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2020 07:09:28] "GET /people HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2020 07:26:36] "GET /people HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2020 07:26:38] "GET /people HTTP/1.1" 200 -
^CLydias-MacBook-Pro:people lydiachilton$ python server.py
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 723-907-492
127.0.0.1 - - [19/Feb/2020 07:26:46] "GET /people HTTP/1.1" 200 -
█
```

```
server.py
server.py
1 from flask
2 from flask
3 from flask
4 app = Flas
5
6
7 current_id
8 data = [
9
10     {
11         "id": 1,
12         "name": "michael scott"
13     },
14     {
15         "id": 2,
16         "name": "jim halpert"
17     },
18 ]
```

**Server:  
keeps the data**



127.0.0.1:5000/people

Reload this page

**Hello people:**

name:

michael scott  
jim halpert

**Client:  
gets data from server  
(and displays it to all users)**

# This website has no database. Why not?

 Columbia University

## User Interface Design

COMS 4170 · Spring 2018

[Home](#) [Syllabus](#) [Assignments](#) ▾

[Piazza](#)

**Part 1** Build websites that suit the needs and abilities of users.

**Part 2** When the needs and abilities of users are uncertain, design systems by learning from iteration and experimentation.

### INSTRUCTOR

[Prof. Lydia Chilton](#)

OH: Tuesday 3-4 pm, CEPSR 612

Please contact staff through Piazza only

### TAS

Tessa Hurr

Eleanor Murguia

Lucille Sui

### WEEKLY SCHEDULE

Lecture

Mon, Wed 4:10–5:25pm, 413 Kent Hall

# But most websites do have databases. They allow you to both see and interact with data.

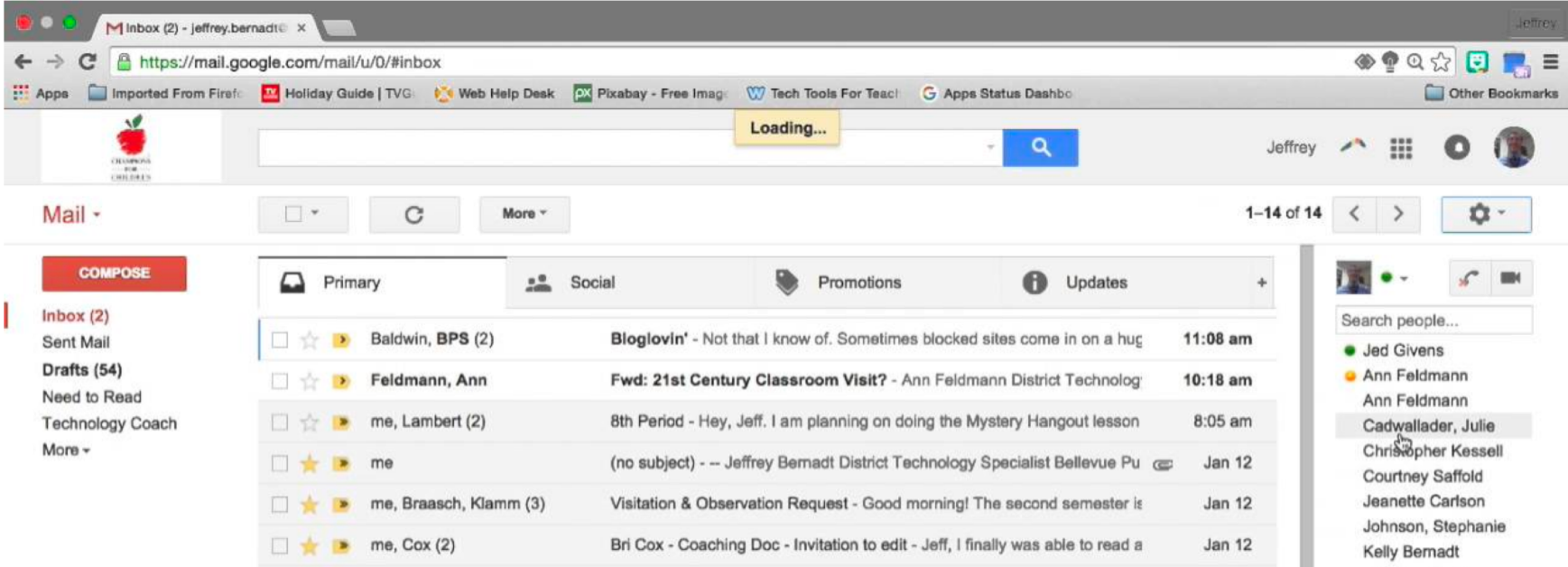
The image is a collage of three web browser screenshots. The top-left screenshot shows the Amazon.com homepage with a search bar, navigation menu, and various book recommendations. The top-right screenshot shows a Gmail inbox with a list of emails, including one from 'Baldwin, BPS' and another from 'Jeffrey Bernadt'. The bottom screenshot shows a Facebook post from 'PictureCorrect Photography Tips' with a blue 'facebook' watermark overlaid on it. The post content includes a question about New Year's resolutions and several user comments.



# Interacting with Data

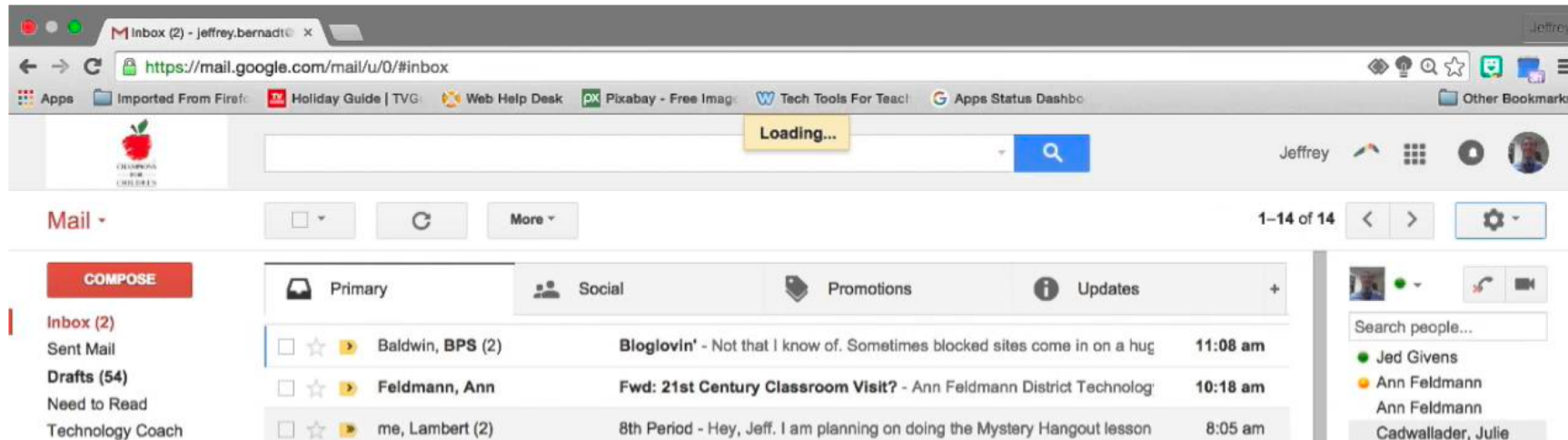
CRUD: Create, Read, Update, and Delete data

# Gmail is a database of emails



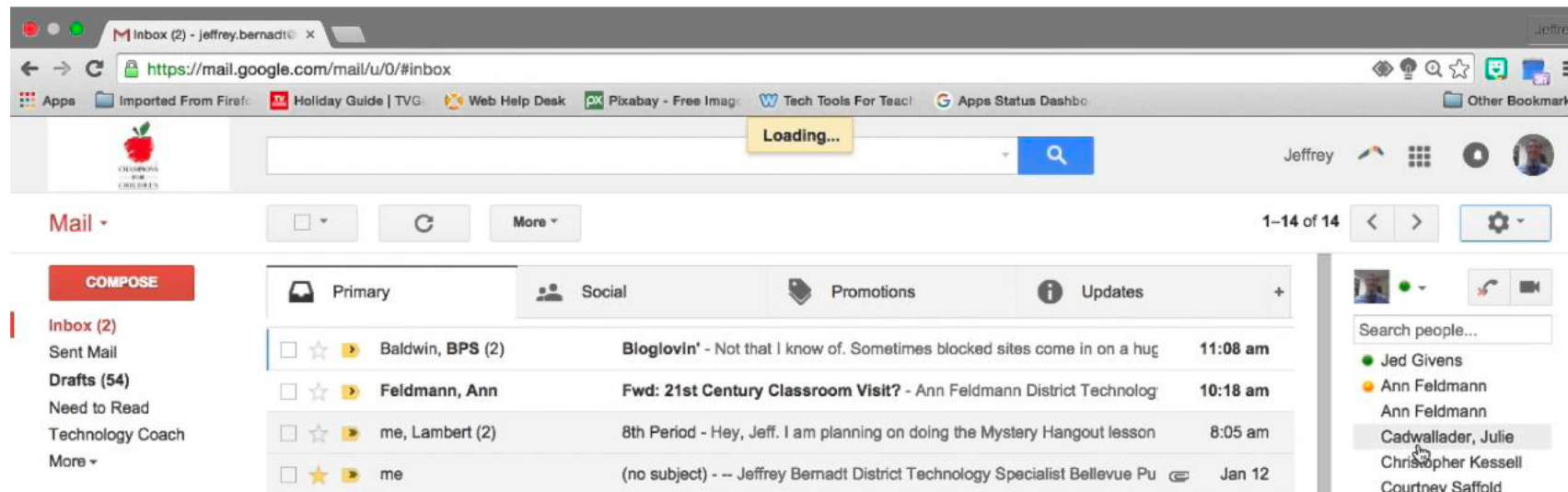
ID	From	To	Subject	Body	Time
1	Baldwin	Jeffrey	Bloglovin'	Not that I know of...	158904993585835
2	Ann Feldman	Jeffrey	Fwd: Classroom visit?	Ann Feldman District Technology	158904993585923
...					

# All emails are stored in one big database. How do we only show Jeff's email to Jeff?



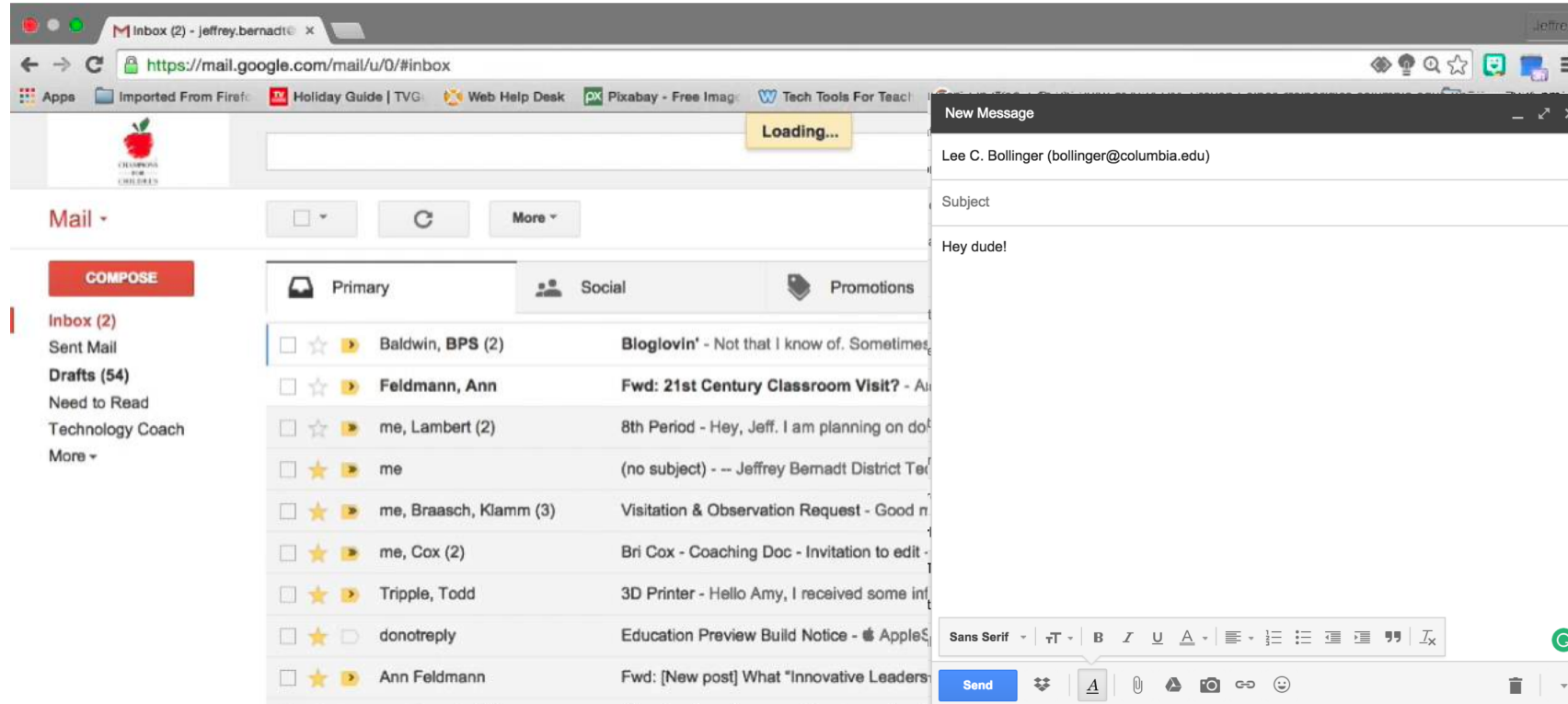
ID	From	To	Subject	Body	Time
1	Baldwin	Jeffrey	Blogloving'	Not that I know of...	158904993585835
2	Ann Feldman	Jeffrey	Fwd: Classroom visit?	Ann Feldman District Technology	158904993585923
3	Lydia Chilton	Lee Bollinger	Bowling tuesday	Hey dude! Donnie and I were thinking of going bowling...	158904993443772
...					

# CRUD operations on data: Create, Read, Update, and Delete Data



ID	From	To	Subject	Body	Time
1	Baldwin	Jeffrey	Blogloving'	Not that I know of...	158904993585835
2	Ann Feldman	Jeffrey	Fwd: Classroom visit?	Ann Feldman District Technology	158904993585923
3	Lydia Chilton	Lee Bollinger	Bowling tuesday	Hey dude! Donnie and I were thinking of going bowling...	158904993443772

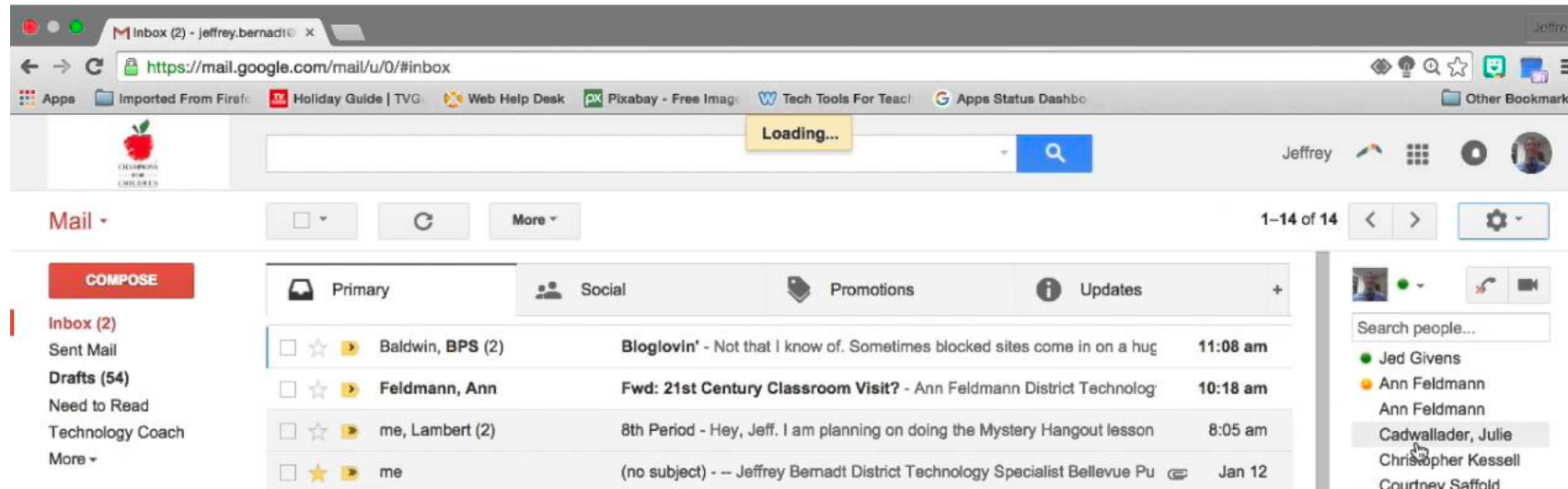
# CREATE: How do users create data in GMail?



1. Compose a new email and send it
2. Reply to an email

# READ: How do users read data in GMail?

How do they see different portions of the database?

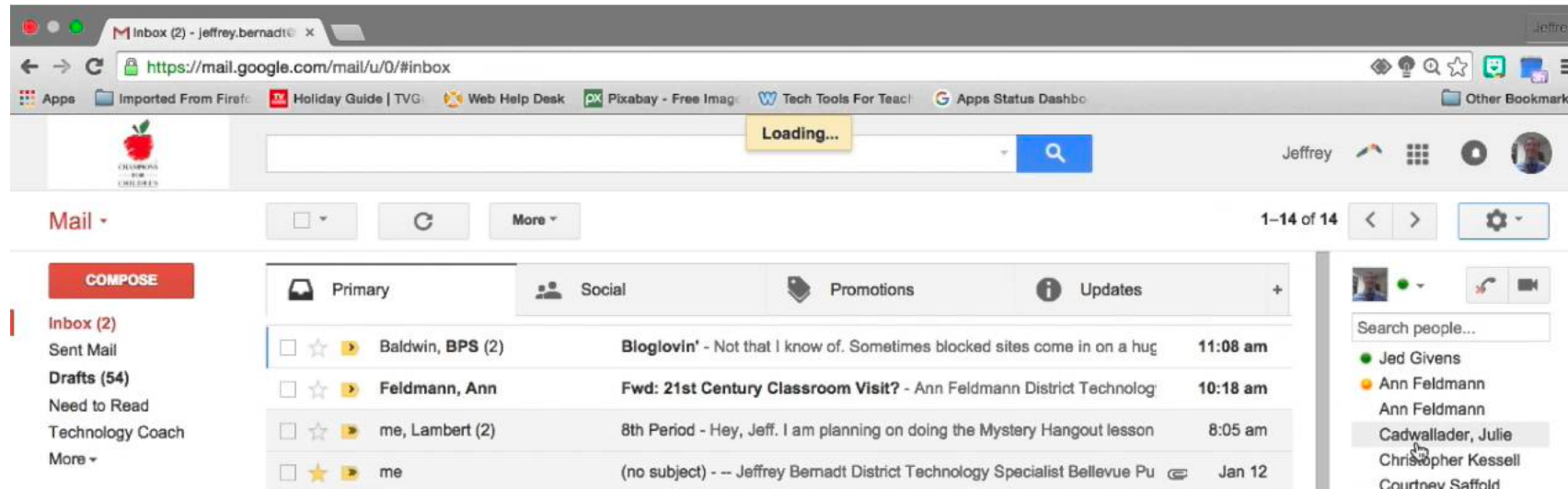


1. Load the page

2. Search inbox

# UPDATE: How do users update data in GMail?

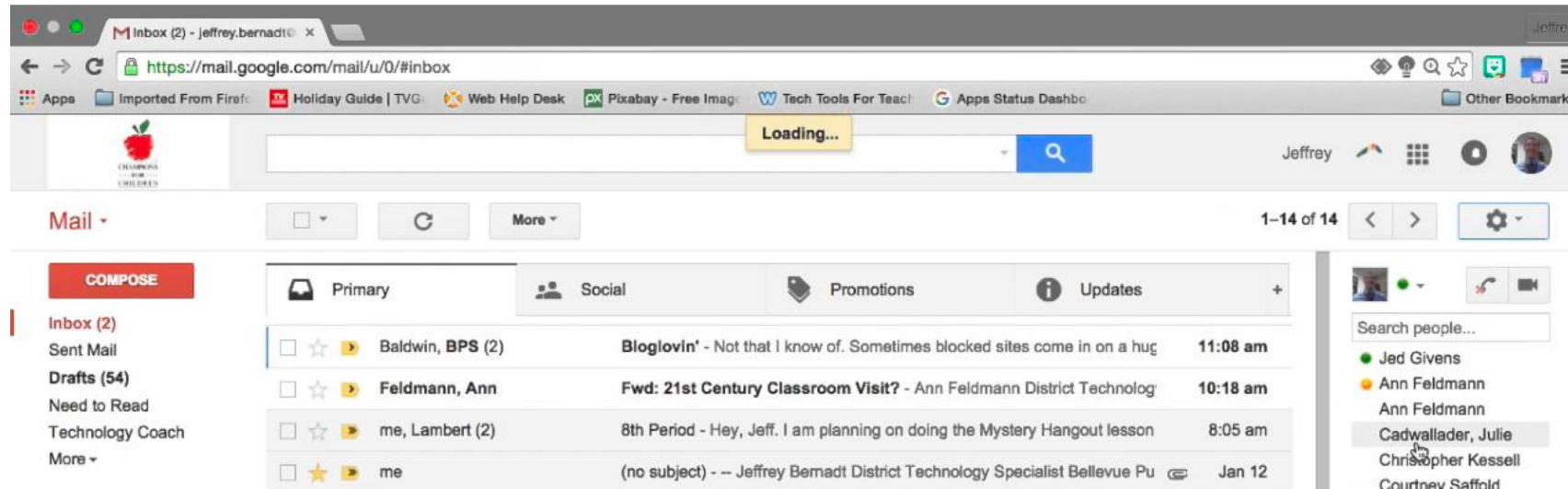
What do they update? And how do they do it in the interface?



1. Read an email (it gets marked as “read”)
2. Star and email
3. Add/remove a label
4. Reply to an email ( the original email gets update to point to the reply)

# DELETE: How do users delete data in GMail?

First, what objects can be deleted in GMail?



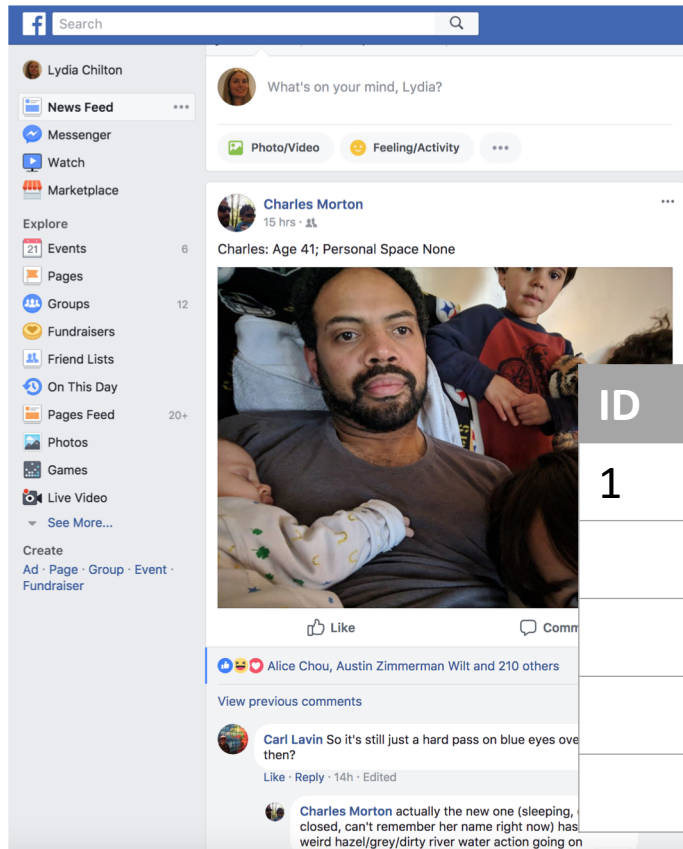
1. Discard a draft

2. Leave spam untouched for 30 days.

Deleting an email doesn't actually delete it – other users may still have it.  
But it does UPDATE the email and mark it as deleted, so it doesn't show up for you.

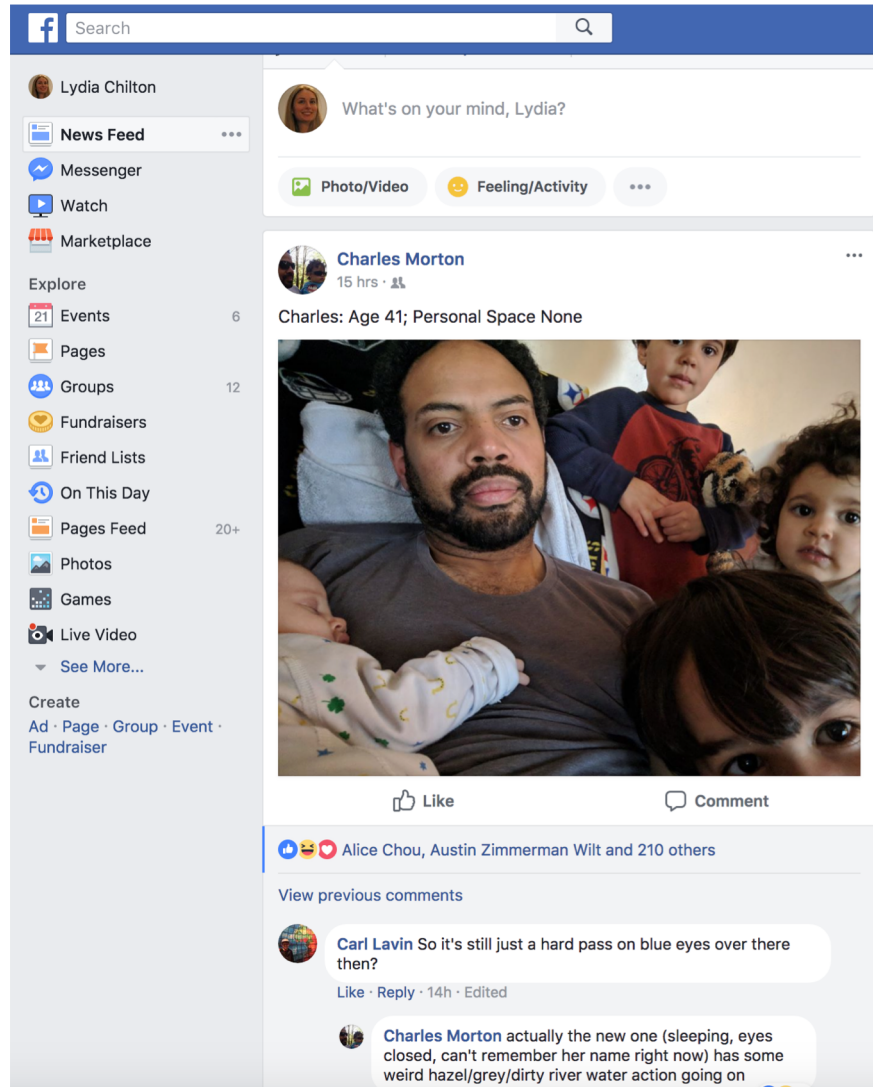


# Facebook is a database of posts.



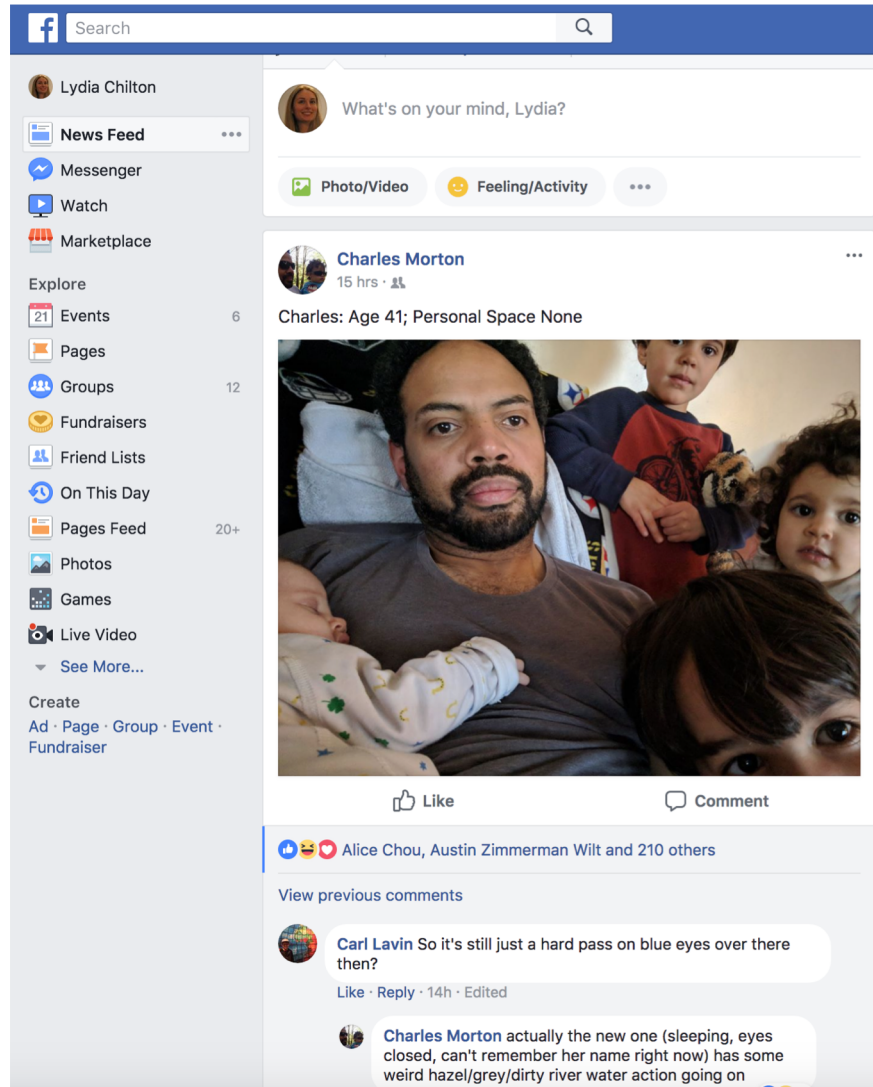
ID	creator	text	Likes	Time
1	Charles Morton	Charles Age 41: Personal..	210	15890499

# CREATE: How do users create data in Facebook?



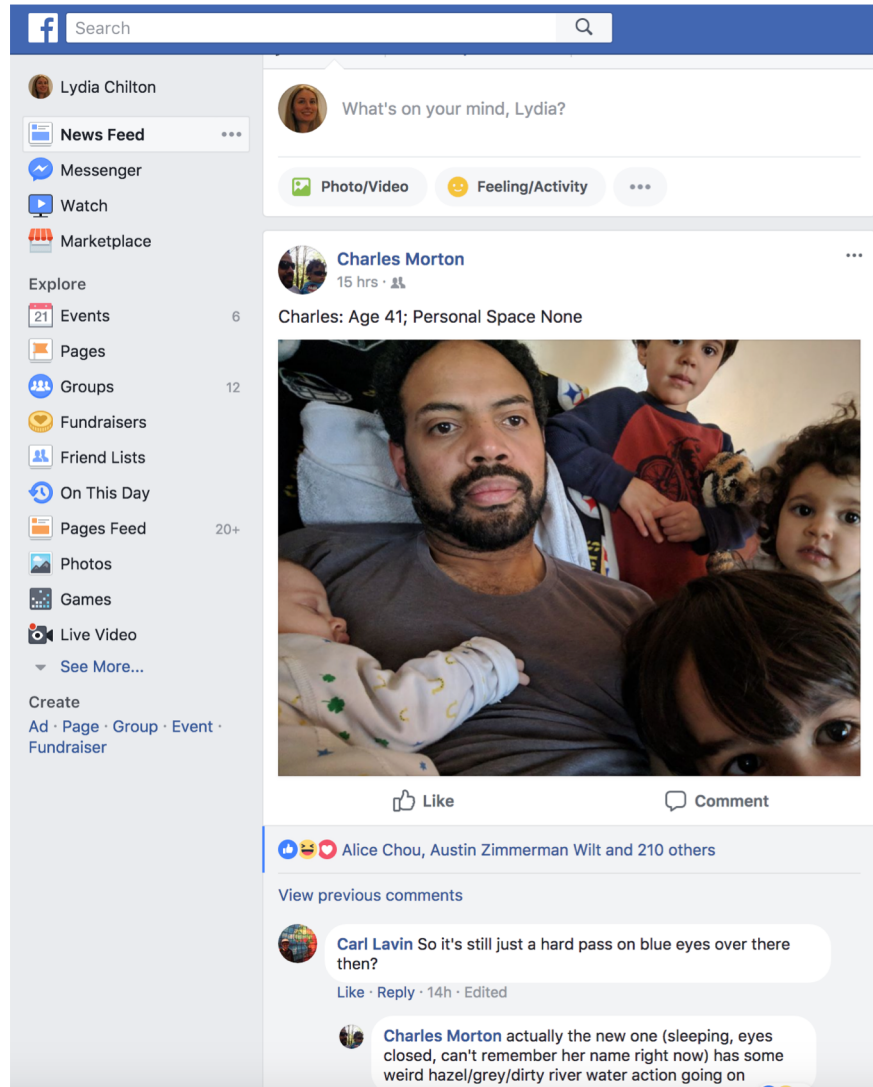
1. Create Post
2. Create Reply

# READ: How do users read data in Facebook?



1. Load the page
2. Search for posts
3. Scroll –and it will autoloading posts

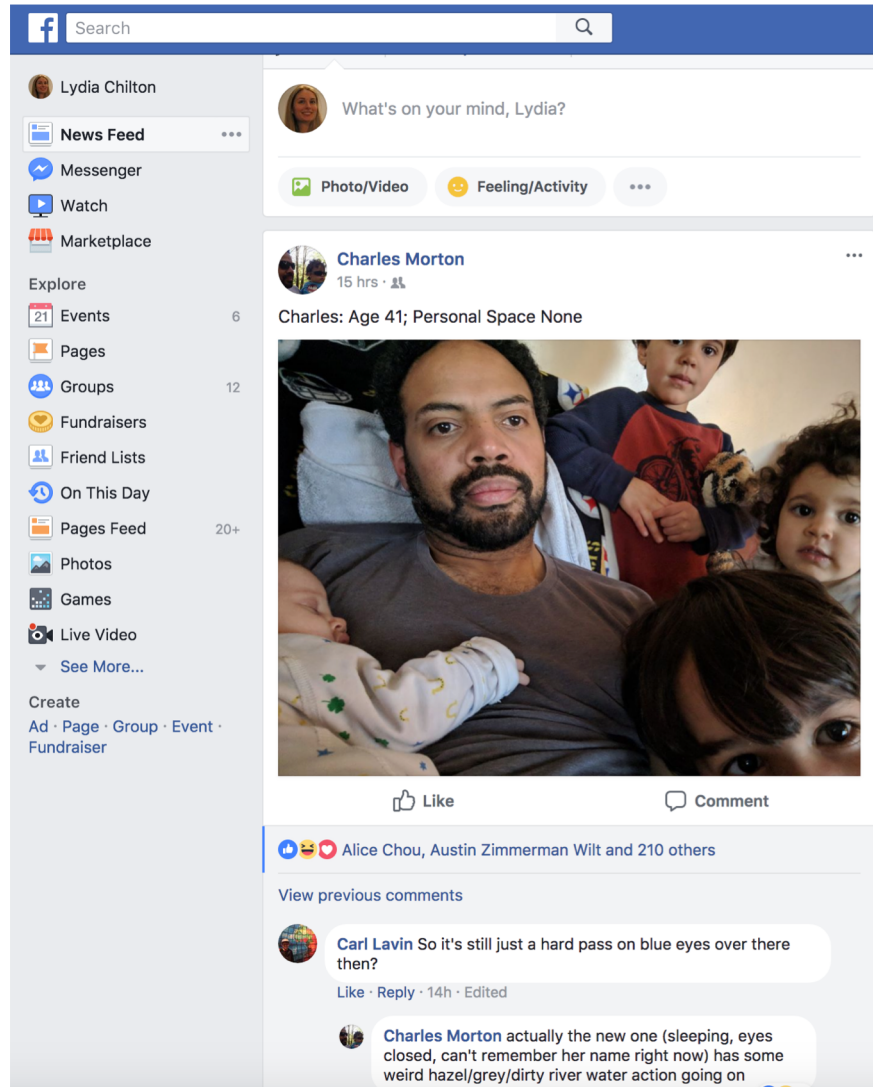
# UPDATE: How do users update data in Facebook?



## 1. Like

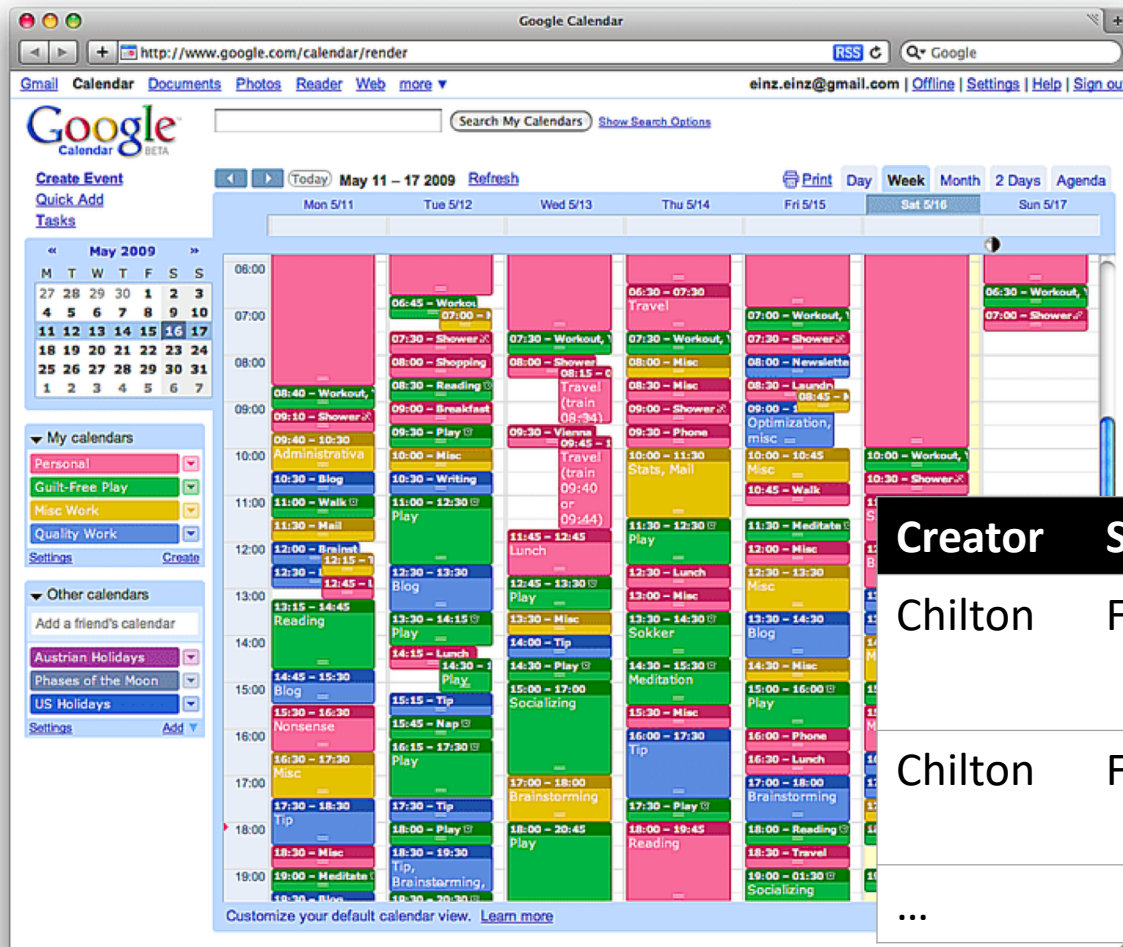


# DELETE: How do users delete data in Facebook?



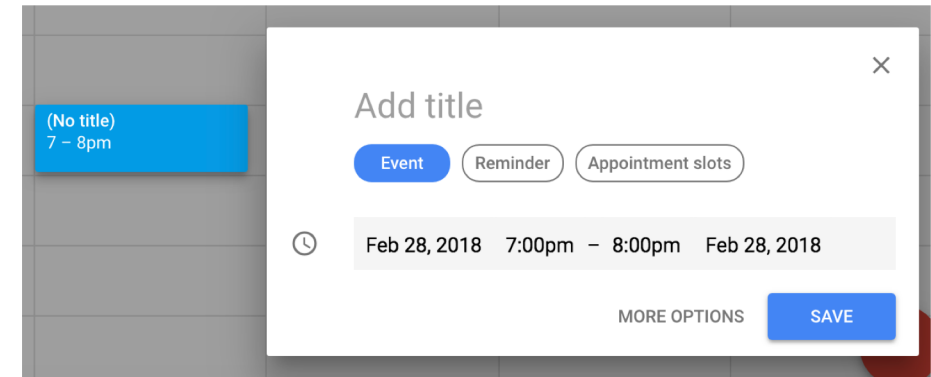
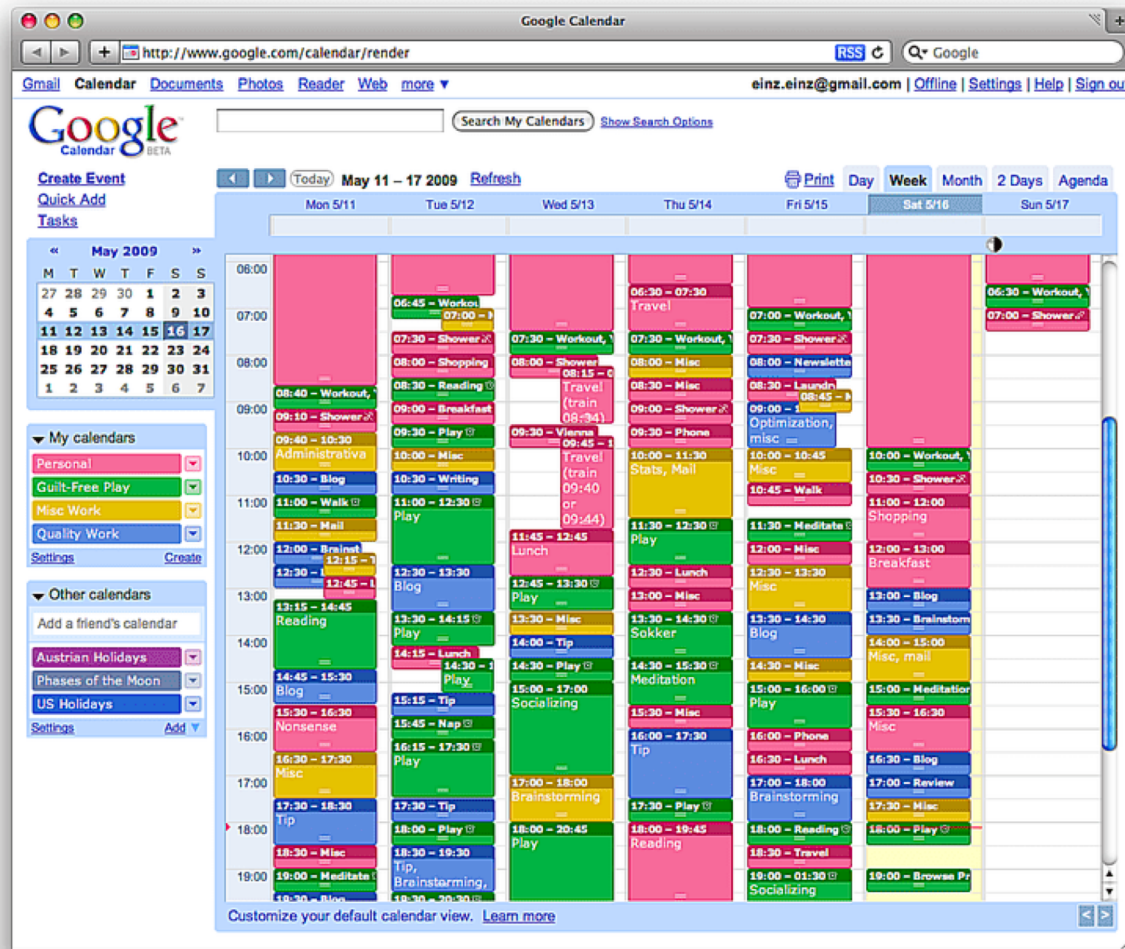
1. Delete the post?

# Google Calendar is a database of events



Creator	Start date	End Date	Title
Chilton	Feb 28, 7pm	Feb 28, 8pm	Dinner with Lee
Chilton	Feb 14, 4pm	Feb 17, 12pm	National Academy of Sciences event
...			

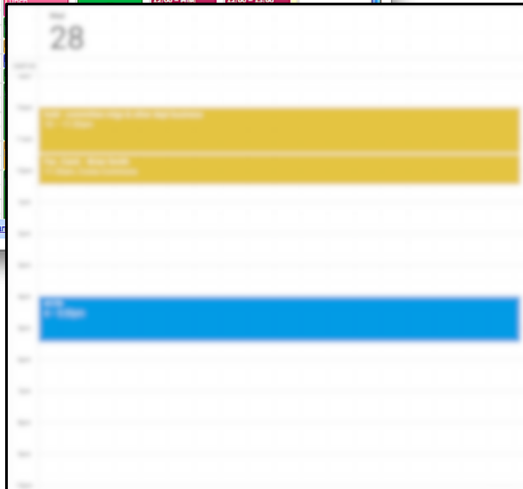
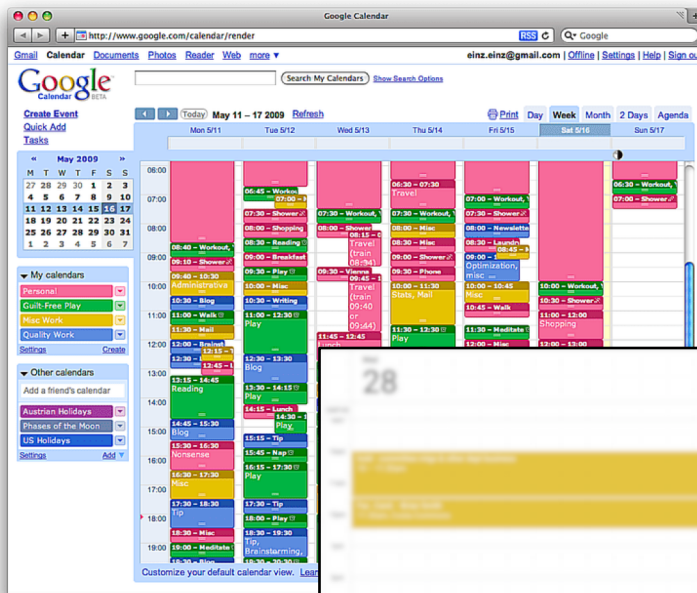
# CREATE: How do users create data in Calendar?



# READ: How do users read data in Calendar?

How do they see different portions of the database?

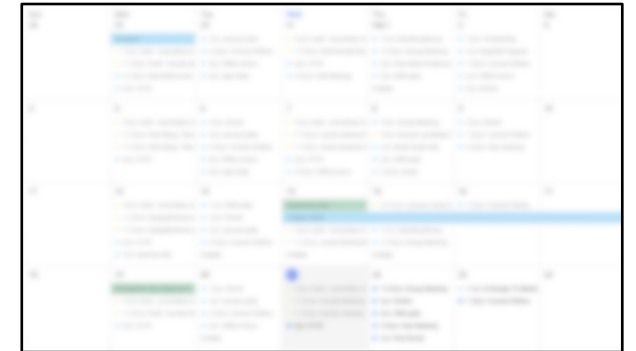
1. Load the page
2. Choose different views



“Day” view



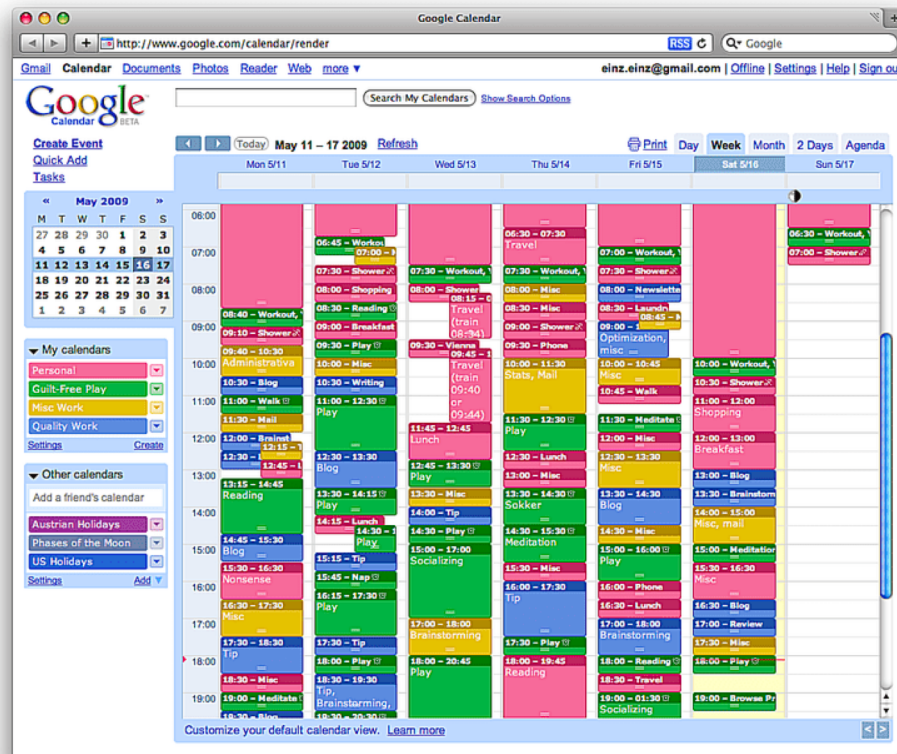
“Week” view



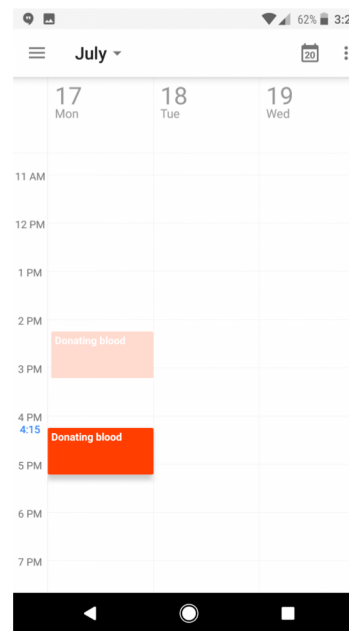
“Month” view



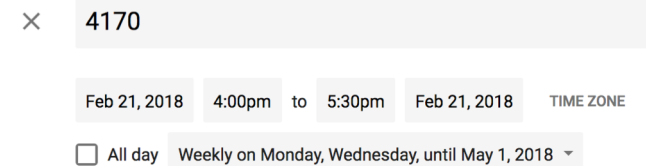
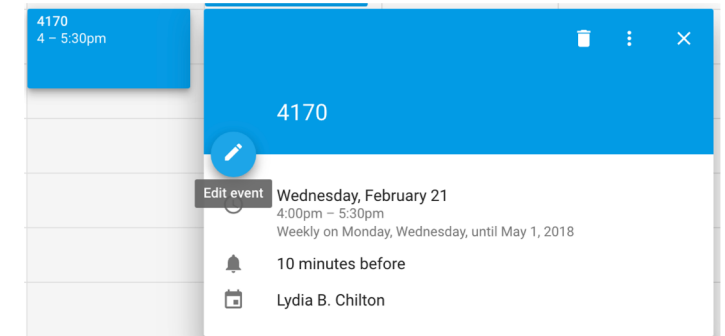
# UPDATE: How do users update data in Calendar?



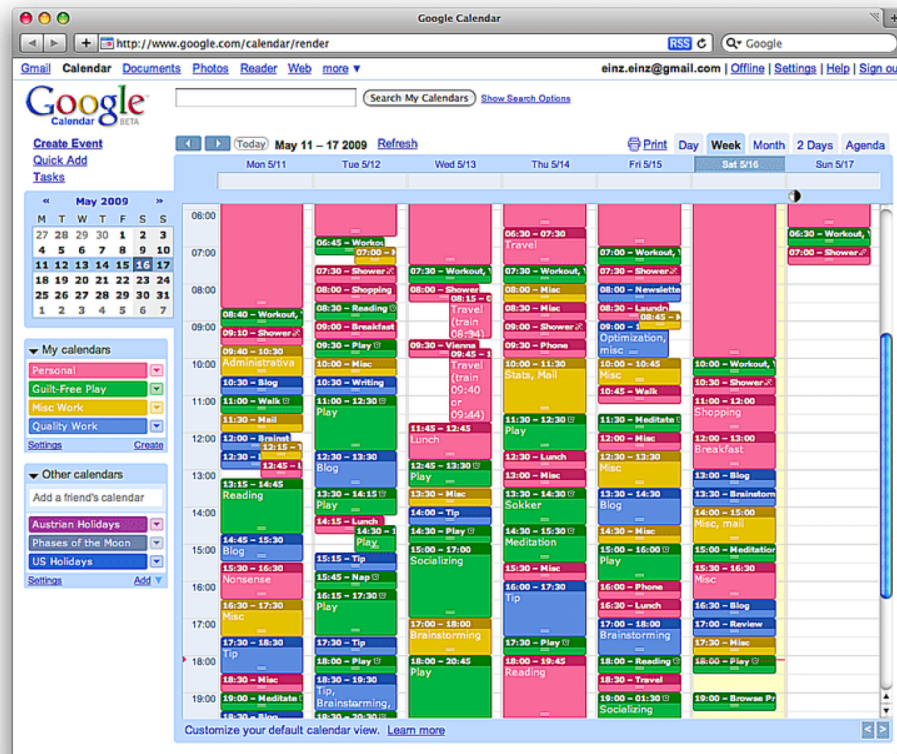
## Drag and drop



## Select an event and edit a form



# DELETE: How do users delete data in Calendar?



Selecting an event and clicking delete



# CRUD: Operations for interacting with a database

## Create


**New Message**

To  |

From Lydia Chilton <chilton@cs.columbia.edu> ▾

Subject

Hey dude!

 What's on your mind, Lydia?

**Add title**

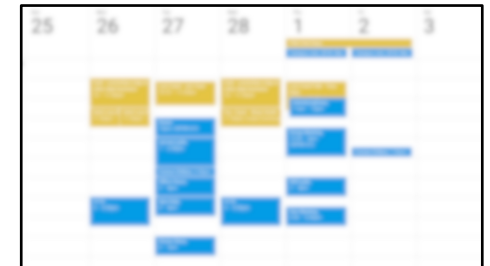
(No title)  
7 - 8pm

Feb 28, 2018 7:00pm - 8:00pm Feb 28, 2018

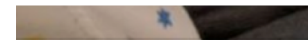
MORE OPTIONS

## Read

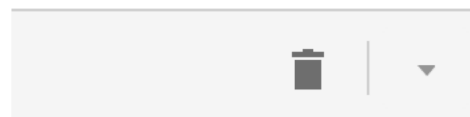
Baldwin, BPS (2)	Bioglovin' - Not that I know of. Sometimes bi
Feldmann, Ann	Fwd: 21st Century Classroom Visit? - Ann
me, Lambert (2)	8th Period - Hey, Jeff. I am planning on doing
me	(no subject) - -- Jeffrey Bernadt District Tech



## Update

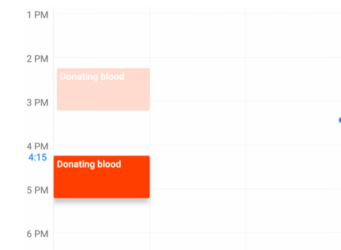


## Delete



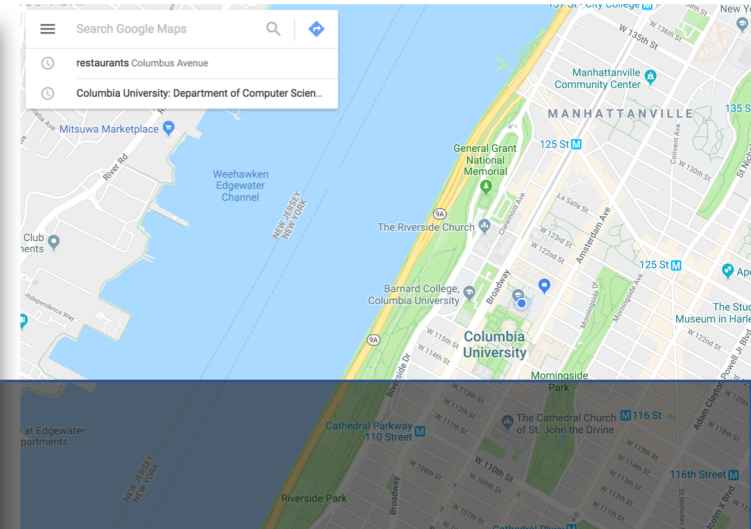
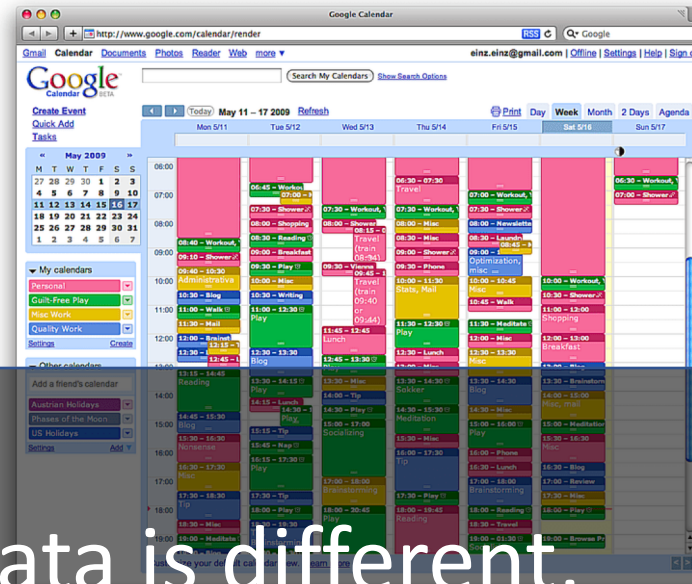
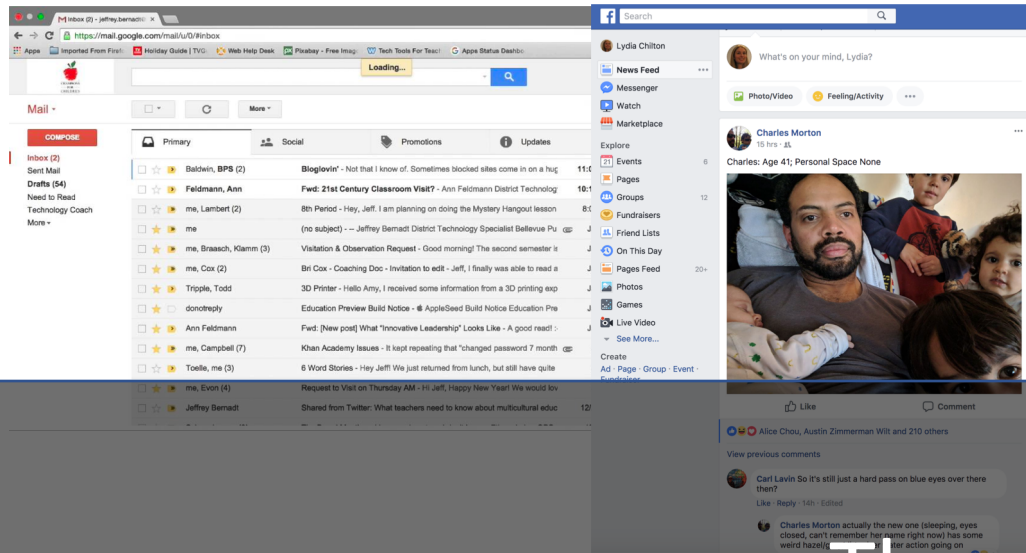
**Delete Post?**

Are you sure you want to delete it?



4170

# From a back-end perspective Facebook, Gmail are very similar. What's different?



The data is different,  
And the information needs they serve are different.

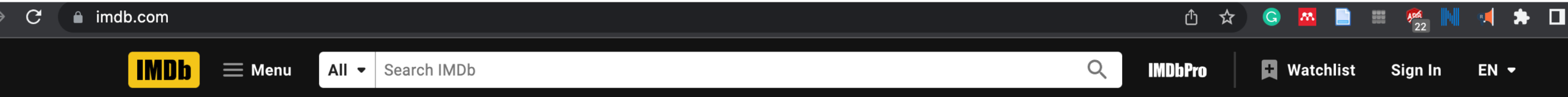
You can drive a lot of human behavior  
with one database update.



# Rendering Templates

Dynamically Generating Webpages from Database Content

# IMDb is a database of movies. (Obviously)



## Featured today




New and Upcoming Superhero Movies and Series

The Most Anticipated Prequels, Sequels, and Spin-Offs

## What to watch

[Get more recommendations >](#)

# What goal are most people trying to achieve on IMDB?



IMDb Find Movies, TV shows, Celebrities and more... All

Movies, TV & Showtimes | Celebs, Events & Photos | News & Community | Watchlist

FULL CAST AND CREW | TRIVIA | USER REVIEWS | IMDbPro | MORE | SHARE

**The Big Lebowski** (1998) ★ 8.2/10 (609,031) ☆ Rate This

R | 1h 57min | Comedy, Crime | 6 March 1998 (USA)

JEFF BRIDGES JOHN GOODMAN  
JULIANNE MOORE STEVE BUSCEMI JOHN TURTURRO

0:31 Trailer 9 VIDEOS 141 IMAGES

Watch Now  
From \$2.99 (SD) on Prime Video ON DISC

"The Dude" Lebowski, mistaken for a millionaire Lebowski, seeks restitution for his ruined rug and enlists his bowling buddies to help get it.

**Directors:** [Joel Coen](#), [Ethan Coen](#) (uncredited)  
**Writers:** [Ethan Coen](#), [Joel Coen](#)  
**Stars:** [Jeff Bridges](#), [John Goodman](#), [Julianne Moore](#) | [See full cast & crew »](#)

69 Metascore From [metacritic.com](#) | Reviews 1,075 user | 257 critic | Popularity 515 (▲ 32)

**Top Rated Movies #167** | 4 wins & 17 nominations. [See more awards »](#)

Look up movie information:

- Who's that actor?
- What's its rating?
- What year did it come out?
- What else did that person direct?



# On IMDB, are there 100,000 static HTML pages sitting around?

IMDb Find Movies, TV shows, Celebrities and more... All

Movies, TV & Showtimes | Celebs, Events & Photos | News & Community | Watchlist

FULL CAST AND CREW | TRIVIA | USER REVIEWS | IMDbPro | MORE | SHARE

**The Big Chill (1983)** ★ 7.2 <sup>10</sup>/<sub>28,073</sub> ☆ Rate This

R | 1h 45min | Comedy, Drama | 30 September 1983 (USA)



2:02 | Trailer | 1 VIDEO | 76 IMAGES

Watch Now With Prime Video

WATCH NOW ON DISC

A group of seven former college friends gather for a weekend reunion at a South Carolina winter house after the funeral of one of their friends.

**Director:** Lawrence Kasdan  
**Writers:** Lawrence Kasdan, Barbara Benedek  
**Stars:** Tom Berenger, Glenn Close, Jeff Goldblum | See full cast & crew »

62 Metascore From metacritic.com | 186 user | 61 critic | Popularity 4,222 (▲ 509)

**Nominated for 3 Oscars.** Another 3 wins & 5 nominations. See more awards »



IMDb Find Movies, TV shows, Celebrities and more... All

Movies, TV & Showtimes | Celebs, Events & Photos | News & Community | Watchlist

FULL CAST AND CREW | TRIVIA | USER REVIEWS | IMDbPro | MORE | SHARE

**The Big Lebowski (1998)** ★ 8.2 <sup>10</sup>/<sub>609,031</sub> ☆ Rate This

R | 1h 57min | Comedy, Crime | 6 March 1998 (USA)



0:31 | Trailer | 9 VIDEOS | 141 IMAGES

Watch Now From \$2.99 (SD) on Prime Video

ON DISC

"The Dude" Lebowski, mistaken for a millionaire Lebowski, seeks restitution for his ruined rug and enlists his bowling buddies to help get it.

**Directors:** Joel Coen, Ethan Coen (uncredited)  
**Writers:** Ethan Coen, Joel Coen  
**Stars:** Jeff Bridges, John Goodman, Julianne Moore | See full cast & crew »

69 Metascore From metacritic.com | 1,075 user | 257 critic | Popularity 515 (▲ 32)

**Top Rated Movies #167** | 4 wins & 17 nominations. See more awards »

IMDb Find Movies, TV shows, Celebrities and more... All

Movies, TV & Showtimes | Celebs, Events & Photos | News & Community | Watchlist

FULL CAST AND CREW | TRIVIA | USER REVIEWS | IMDbPro | MORE | SHARE

**The Big Short (2015)** ★ 7.8 <sup>10</sup>/<sub>274,582</sub> ☆ Rate This

R | 2h 10min | Biography, Comedy, Drama | 23 December 2015 (USA)



2:05 | Trailer | 39 VIDEOS | 128 IMAGES

Watch Now From \$2.99 (SD) on Prime Video

ON DISC

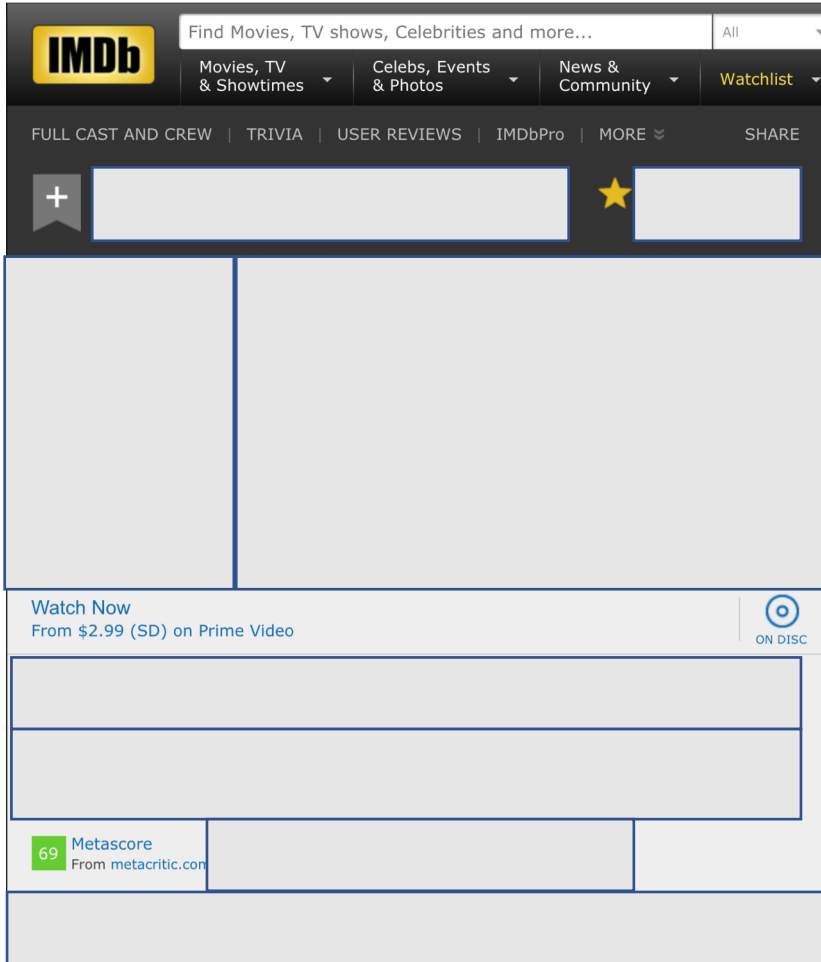
In 2006-7 a group of investors bet against the US mortgage market. In their research they discover how flawed and corrupt the market is.

**Director:** Adam McKay  
**Writers:** Charles Randolph (screenplay), Adam McKay (screenplay) | 1 more credit »  
**Stars:** Christian Bale, Steve Carell, Ryan Gosling | See full cast & crew »

81 Metascore From metacritic.com | 428 user | 454 critic | Popularity 434 (▲ 47)

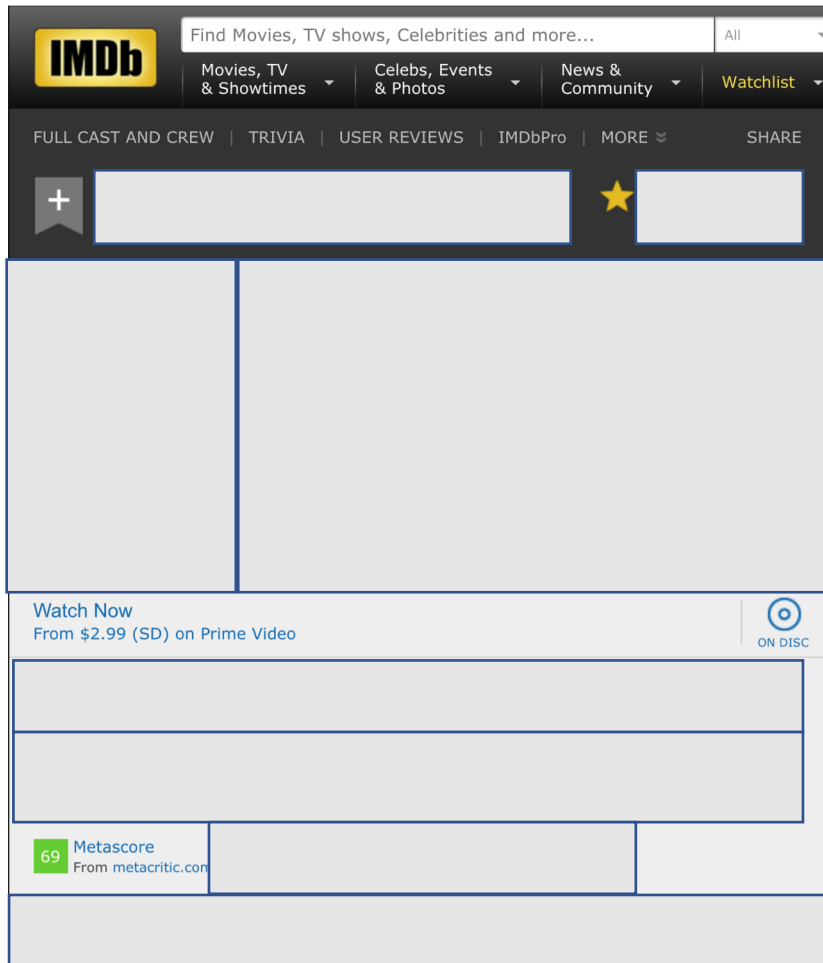
**Won 1 Oscar.** Another 37 wins & 79 nominations. See more awards »

# IMDB uses a template and a database...



Title	Plot summary	poster	year
The Big Lebowski	"The dude" Lebowski, mistaken...	Lebowski.jpg	1998
The Big Short	In 2006-7 a group of investors...	Big_short.jpg	2015
The Big Chill	A group of 7 former roommates	Big_chill.jpg	1983

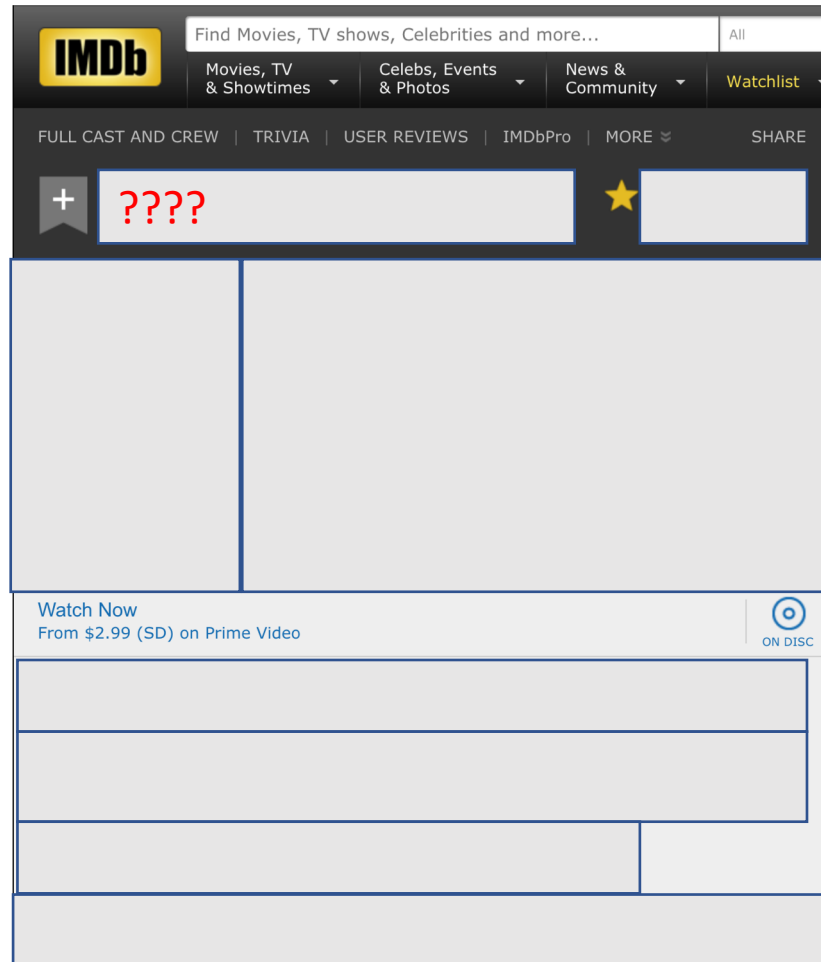
# IMDb uses a template and a database... And fills in the data dynamically on pageload



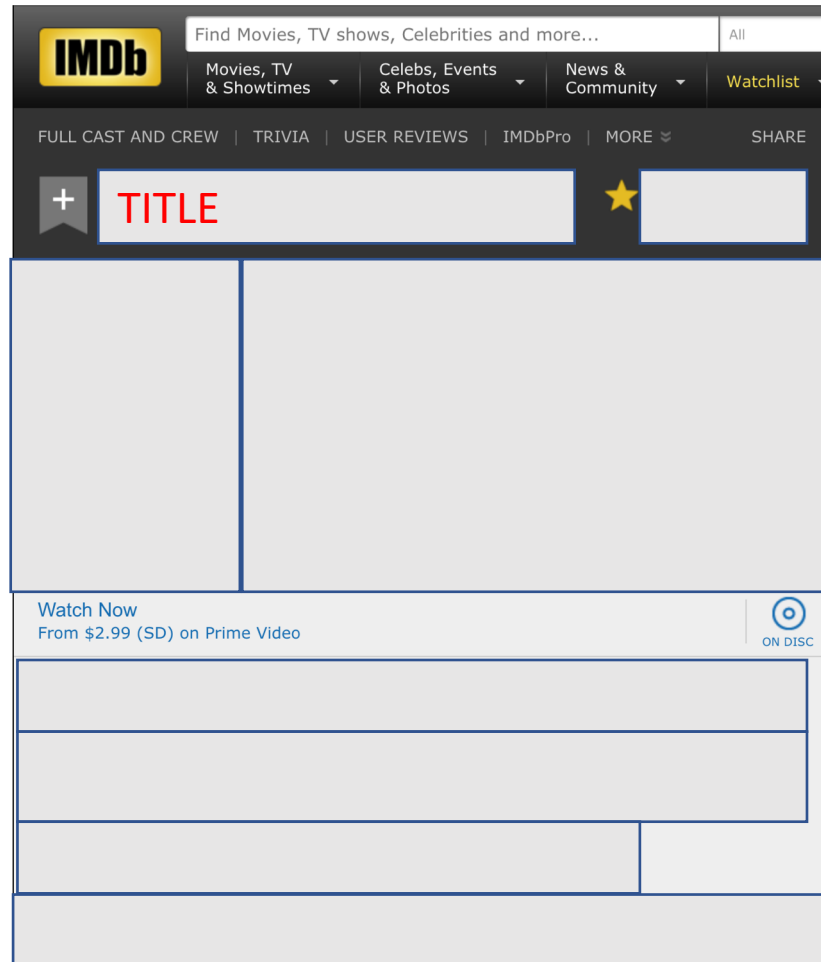
Title	Plot summary	poster	year
The Big Lebowski	"The dude" Lebowski, mistaken...	Lebowski.jpg	1998
The Big Short	In 2006-7 a group of investors...	Big_short.jpg	2015
The Big Chill	A group of 7 former roommates	Big_chill.jpg	1983



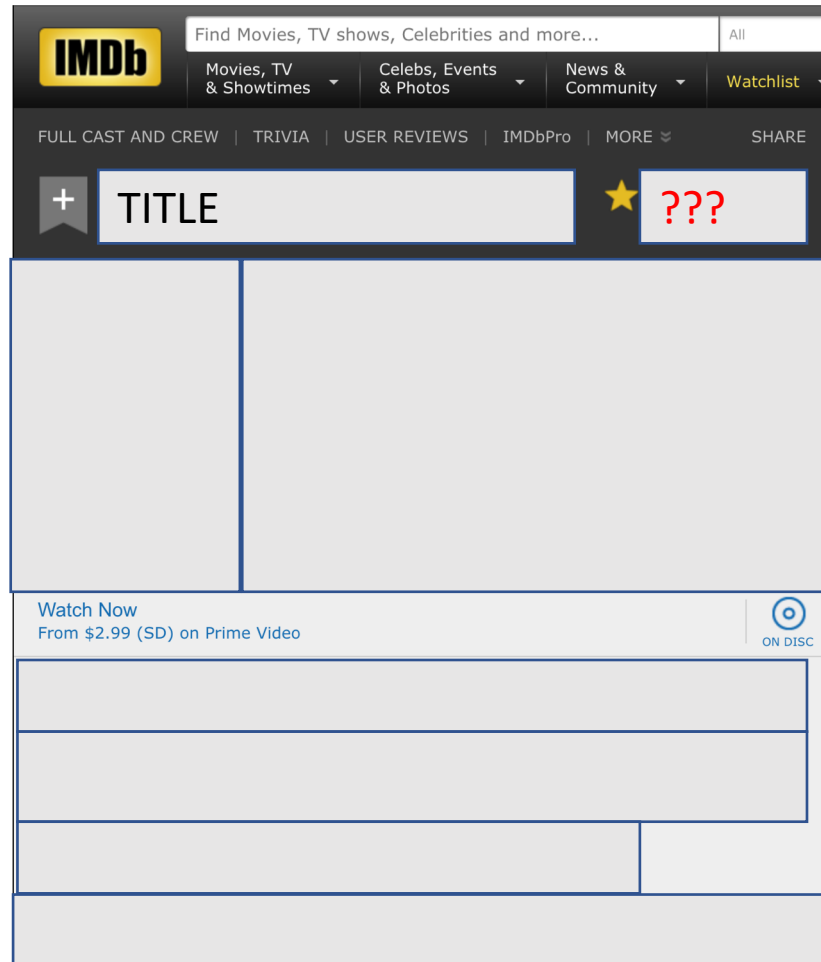
# What goes in the template?



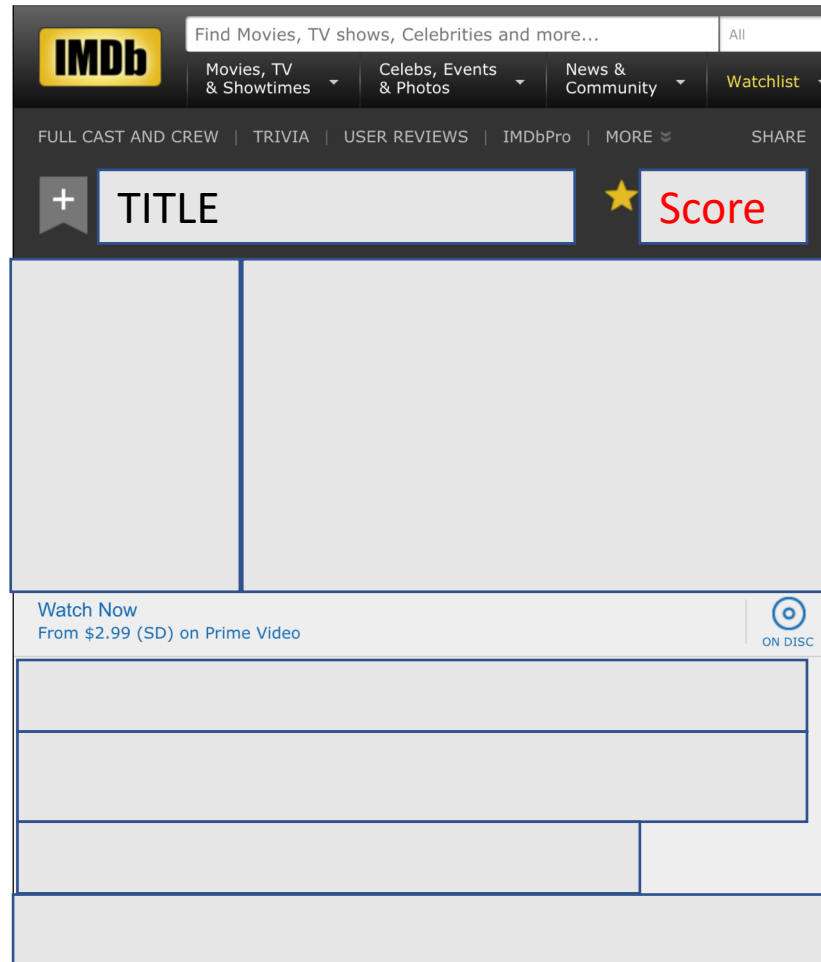
# What goes in the template?



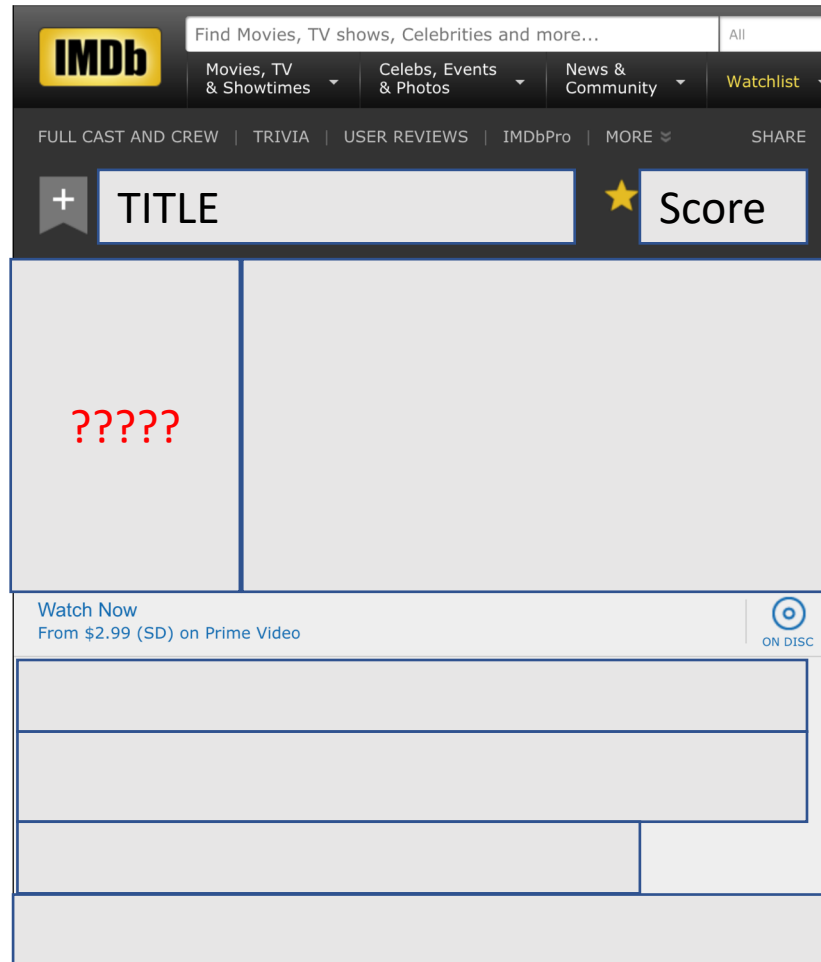
# What goes in the template?



# What goes in the template?



# What goes in the template?

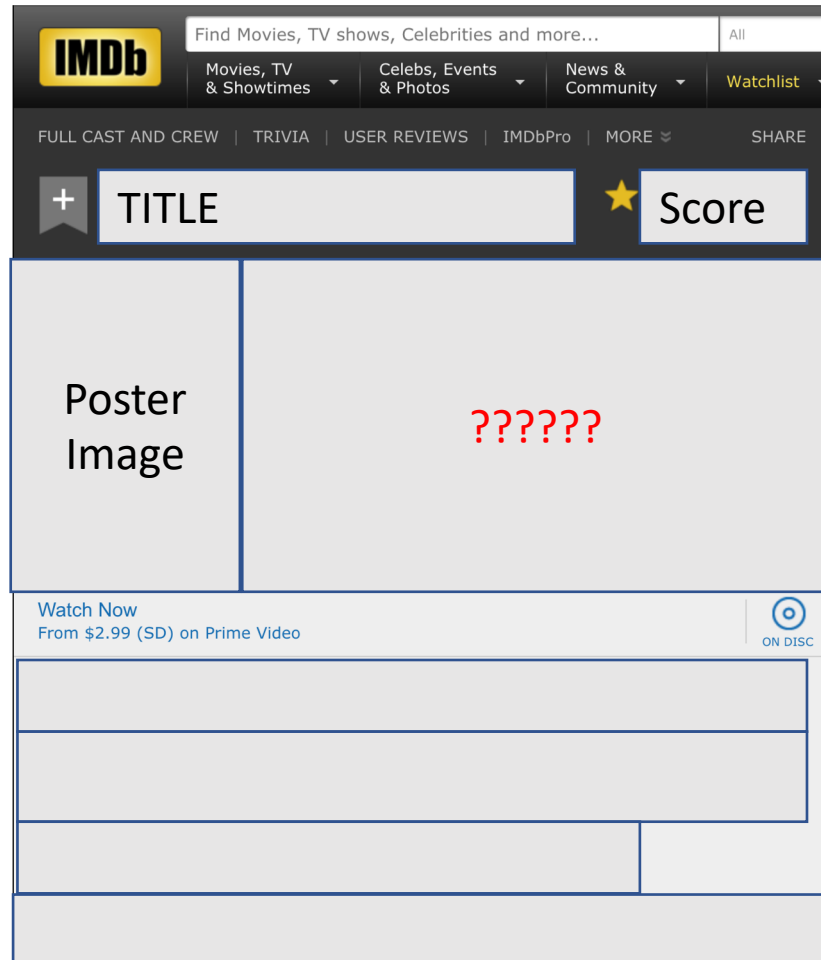




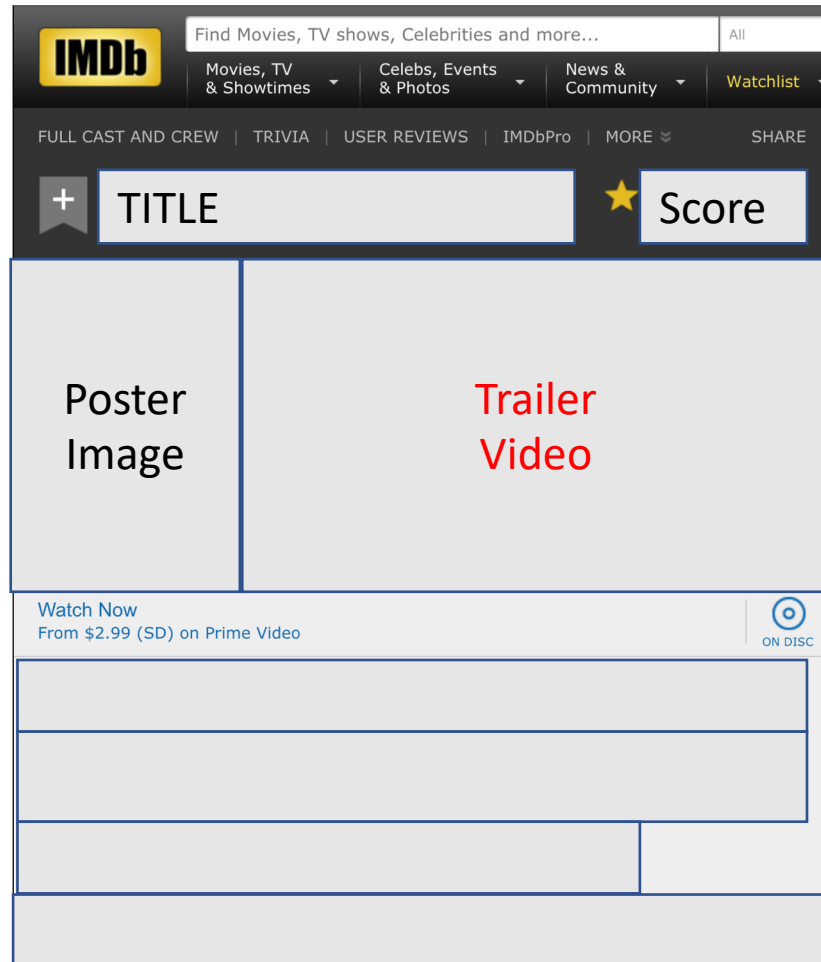
# What goes in the template?



# What goes in the template?



# What goes in the template?



IMDB uses a template and dynamically fills the template from a database query

The screenshot displays the IMDb page for the movie "The Big Lebowski" (1998). At the top, there is a search bar and navigation tabs for "Movies, TV & Showtimes", "Celebs, Events & Photos", "News & Community", and "Watchlist". Below the navigation, there are links for "FULL CAST AND CREW", "TRIVIA", "USER REVIEWS", "IMDbPro", and "MORE". The movie title "The Big Lebowski (1998)" is prominently displayed, along with its rating of 8.2/10 and a "Rate This" button. The movie's genre is listed as "Comedy, Crime" and its release date as "6 March 1998 (USA)". A video player is embedded, showing a scene from the movie with a play button and a "Trailer" label. Below the video player, there is a "Watch Now" section with a price of "From \$2.99 (SD) on Prime Video" and a "ON DISC" button. The bottom of the page features several empty rectangular boxes, which are likely placeholders for a template structure.

# Anyone remember the other categories?

IMDb Find Movies, TV shows, Celebrities and more... All

Movies, TV & Showtimes | Celebs, Events & Photos | News & Community | Watchlist

FULL CAST AND CREW | TRIVIA | USER REVIEWS | IMDbPro | MORE | SHARE

**The Big Lebowski (1998)** ★ 8.2<sup>10</sup>  
609,031 Rate This

R | 1h 57min | Comedy, Crime | 6 March 1998 (USA)

**BIG LEBOWSKI**  
FROM THE CREATORS OF "FARGO"

JEFF BRIDGES JOHN GOODMAN  
JULIANNE MOORE STEVE BUSCEMI JOHN TURTURRO

0:31 Trailer 9 VIDEOS 141 IMAGES

Watch Now  
From \$2.99 (SD) on Prime Video

ON DISC

[Empty rectangular boxes representing missing content or redactions]

# Anyone remember the other categories?

IMDb Find Movies, TV shows, Celebrities and more... All

Movies, TV & Showtimes | Celebs, Events & Photos | News & Community | Watchlist

FULL CAST AND CREW | TRIVIA | USER REVIEWS | IMDbPro | MORE | SHARE

**The Big Lebowski (1998)** ★ 8.2<sup>10</sup>  
609,031 Rate This

R | 1h 57min | Comedy, Crime | 6 March 1998 (USA)

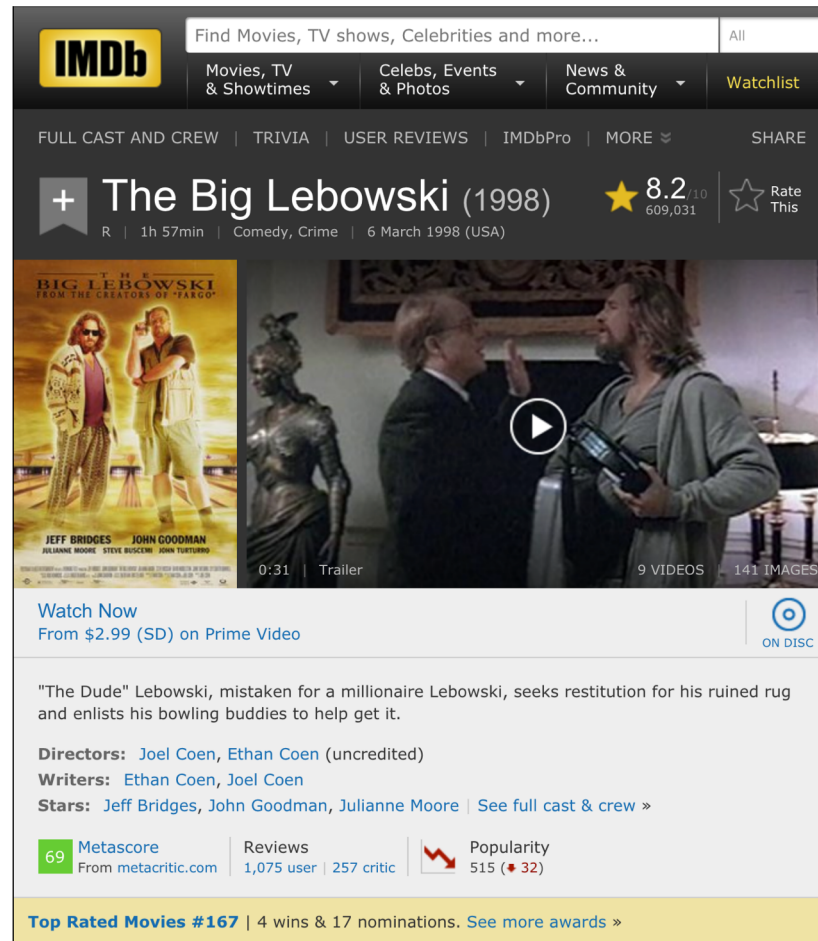
**Plot summary**

**Main creatives**

**Reviews**

**Awards**

# Anyone remember the other categories?



IMDb Find Movies, TV shows, Celebrities and more... All

Movies, TV & Showtimes | Celebs, Events & Photos | News & Community | Watchlist

FULL CAST AND CREW | TRIVIA | USER REVIEWS | IMDbPro | MORE | SHARE

**The Big Lebowski (1998)** ★ 8.2<sup>10</sup><sub>609,031</sub> ☆ Rate This

R | 1h 57min | Comedy, Crime | 6 March 1998 (USA)

**Watch Now**  
From \$2.99 (SD) on Prime Video

"The Dude" Lebowski, mistaken for a millionaire Lebowski, seeks restitution for his ruined rug and enlists his bowling buddies to help get it.

**Directors:** [Joel Coen](#), [Ethan Coen](#) (uncredited)  
**Writers:** [Ethan Coen](#), [Joel Coen](#)  
**Stars:** [Jeff Bridges](#), [John Goodman](#), [Julianne Moore](#) | [See full cast & crew »](#)

69 Metascore From [metacritic.com](#) | 1,075 user | 257 critic | Popularity 515 (▲ 32)

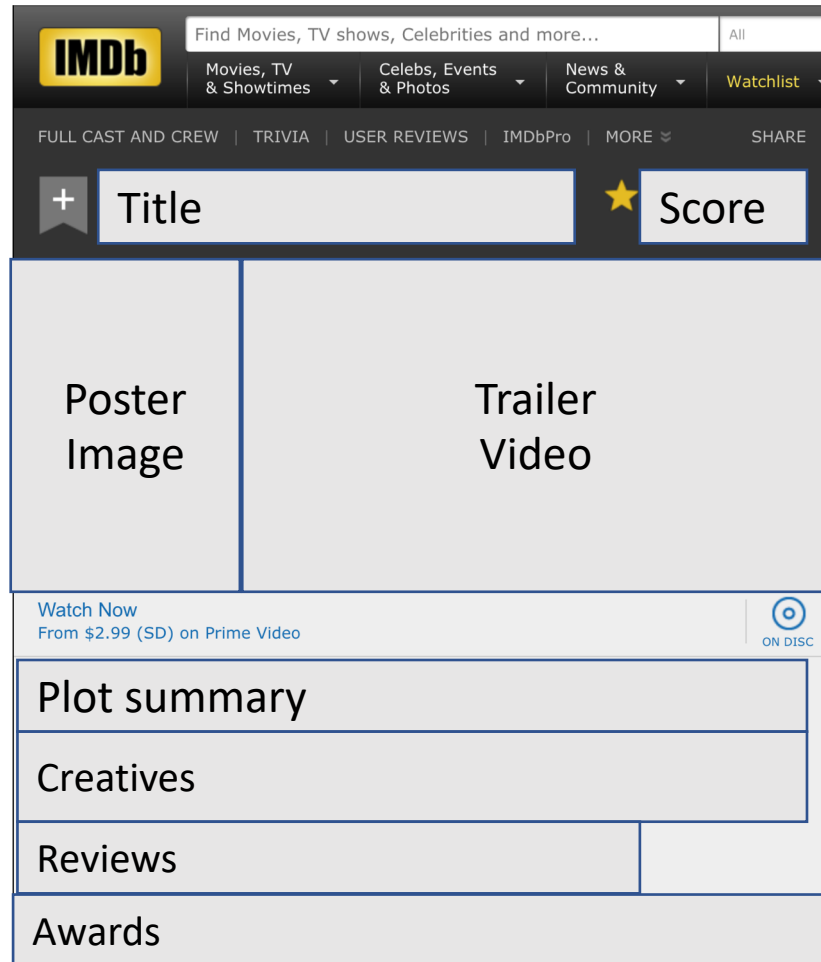
**Top Rated Movies #167** | 4 wins & 17 nominations. [See more awards »](#)

# IMDB Template

The image shows a screenshot of the IMDb website interface with various HTML placeholders for a movie page. The top navigation bar includes the IMDb logo, a search bar, and menu items like 'Movies, TV & Showtimes', 'Celebs, Events & Photos', 'News & Community', and 'Watchlist'. Below the navigation bar, there are links for 'FULL CAST AND CREW', 'TRIVIA', 'USER REVIEWS', 'IMDbPro', 'MORE', and 'SHARE'. The main content area features a title placeholder `{{movie.title}}` with a plus icon and a star icon next to a star placeholder `{{movie.star}}`. Below this, there are two large boxes: one for a poster placeholder ``. A 'Watch Now' section indicates availability on Prime Video from \$2.99 (SD) and includes an 'ON DISC' icon. The bottom section contains several more placeholders: `{{movie.plot_summary}}`, `{{movie.main_creatives}}`, `{{movie.reviews}}`, and `{{movie.awards}}`.



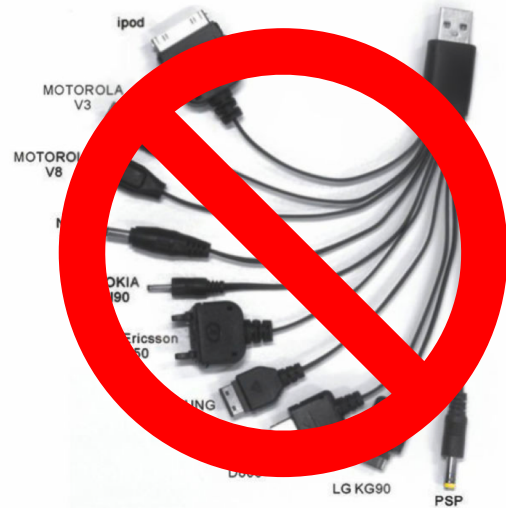
# Templates are standardizations. What can go wrong?



# Pros and cons of standardization

## Pro: Simpler.

Frees you from worrying about millions of options



## Con:

Standards might not fit all the options.



# How should template display the creatives?

The screenshot shows the IMDb page for the movie 'The Big Short' (2015). The page includes a search bar at the top, navigation tabs for 'Movies, TV & Showtimes', 'Celebs, Events & Photos', 'News & Community', and 'Watchlist'. Below the navigation, there are links for 'FULL CAST AND CREW', 'TRIVIA', 'USER REVIEWS', 'IMDbPro', 'MORE', and 'SHARE'. The movie title 'The Big Short (2015)' is displayed with a star rating placeholder and a plus sign icon. Below the title, the genre 'Biography, Comedy, Drama' and release date '23 December 2015 (USA)' are shown. The main content area is mostly obscured by a large grey template box. At the bottom, the following text is visible: **Director:** Adam McKay; **Writers:** Charles Randolph (screenplay), Adam McKay (screenplay) | 1 more credit »; **Stars:** Christian Bale, Steve Carell, Ryan Gosling | See full cast & crew ».

The screenshot shows the IMDb page for the movie 'The Big Lebowski' (1998). The page layout is identical to the first screenshot, including the search bar, navigation tabs, and movie title 'The Big Lebowski (1998)'. The genre 'Comedy, Crime' and release date '6 March 1998 (USA)' are shown. The main content area is obscured by a large grey template box. A red text placeholder '?????????' is visible in the middle of the page, and the start of the 'Stars' list is visible at the bottom: **Stars:** Jeff Bridges, John Goodman, Julianne Moore | See full cast & crew ».

You can have multiple directors.  
Directors needs to be a list.

The screenshot shows the IMDb page for the movie 'The Big Short' (2015). The page includes a search bar at the top, navigation tabs for 'Movies, TV & Showtimes', 'Celebs, Events & Photos', 'News & Community', and 'Watchlist'. Below the navigation is a horizontal menu with 'FULL CAST AND CREW', 'TRIVIA', 'USER REVIEWS', 'IMDbPro', 'MORE', and 'SHARE'. The movie title 'The Big Short (2015)' is displayed with a star rating and a placeholder box. Below the title, the genre 'Biography, Comedy, Drama' and release date '23 December 2015 (USA)' are shown. The main content area is mostly obscured by a large grey placeholder. At the bottom, the 'Director' is listed as Adam McKay, and 'Writers' are listed as Charles Randolph (screenplay) and Adam McKay (screenplay). 'Stars' listed include Christian Bale, Steve Carell, and Ryan Gosling.

The screenshot shows the IMDb page for the movie 'The Big Lebowski' (1998). The page includes a search bar at the top, navigation tabs for 'Movies, TV & Showtimes', 'Celebs, Events & Photos', 'News & Community', and 'Watchlist'. Below the navigation is a horizontal menu with 'FULL CAST AND CREW', 'TRIVIA', 'USER REVIEWS', 'IMDbPro', 'MORE', and 'SHARE'. The movie title 'The Big Lebowski (1998)' is displayed with a star rating and a placeholder box. Below the title, the genre 'Comedy, Crime' and release date '6 March 1998 (USA)' are shown. The main content area is mostly obscured by a large grey placeholder. A snippet of text is visible: 'and enlists his bowling buddies to help get it.' Below this, the 'Directors' are listed as Joel Coen and Ethan Coen (uncredited). A large grey placeholder box is present below the director information.

# Standardization is hard. Expect to **iterate** on your templates

IMDb Find Movies, TV shows, Celebrities and more... All

Movies, TV & Showtimes | Celebs, Events & Photos | News & Community | Watchlist

FULL CAST AND CREW | TRIVIA | USER REVIEWS | IMDbPro | MORE | SHARE

**The Big Chill (1983)** ★ 7.2 <sup>10</sup>/<sub>28,073</sub> ☆ Rate This

R | 1h 45min | Comedy, Drama | 30 September 1983 (USA)



02:02 | Trailer | 1 VIDEO | 76 IMAGES

Watch Now With Prime Video

WATCH NOW ON DISC

A group of seven former college friends gather for a weekend reunion at a South Carolina winter house after the funeral of one of their friends.

**Director:** [Lawrence Kasdan](#)  
**Writers:** [Lawrence Kasdan](#), [Barbara Benedek](#)  
**Stars:** [Tom Berenger](#), [Glenn Close](#), [Jeff Goldblum](#) | See full cast & crew »

62 Metascore From metacritic.com | 186 user | 61 critic | Popularity 4,222 (▲ 509)

Nominated for 3 Oscars. Another 3 wins & 5 nominations. See more awards »



IMDb Find Movies, TV shows, Celebrities and more... All

Movies, TV & Showtimes | Celebs, Events & Photos | News & Community | Watchlist

FULL CAST AND CREW | TRIVIA | USER REVIEWS | IMDbPro | MORE | SHARE

**The Big Lebowski (1998)** ★ 8.2 <sup>10</sup>/<sub>609,031</sub> ☆ Rate This

R | 1h 57min | Comedy, Crime | 6 March 1998 (USA)



0:31 | Trailer | 9 VIDEOS | 141 IMAGES

Watch Now From \$2.99 (SD) on Prime Video

ON DISC

"The Dude" Lebowski, mistaken for a millionaire Lebowski, seeks restitution for his ruined rug and enlists his bowling buddies to help get it.

**Directors:** [Joel Coen](#), [Ethan Coen](#) (uncredited)  
**Writers:** [Ethan Coen](#), [Joel Coen](#)  
**Stars:** [Jeff Bridges](#), [John Goodman](#), [Julianne Moore](#) | See full cast & crew »

69 Metascore From metacritic.com | 1,075 user | 257 critic | Popularity 515 (▲ 32)

Top Rated Movies #167 | 4 wins & 17 nominations. See more awards »

IMDb Find Movies, TV shows, Celebrities and more... All

Movies, TV & Showtimes | Celebs, Events & Photos | News & Community | Watchlist

FULL CAST AND CREW | TRIVIA | USER REVIEWS | IMDbPro | MORE | SHARE

**The Big Short (2015)** ★ 7.8 <sup>10</sup>/<sub>274,582</sub> ☆ Rate This

R | 2h 10min | Biography, Comedy, Drama | 23 December 2015 (USA)



2:05 | Trailer | 39 VIDEOS | 128 IMAGES

Watch Now From \$2.99 (SD) on Prime Video

ON DISC

In 2006-7 a group of investors bet against the US mortgage market. In their research they discover how flawed and corrupt the market is.

**Director:** [Adam McKay](#)  
**Writers:** [Charles Randolph](#) (screenplay), [Adam McKay](#) (screenplay) | 1 more credit »  
**Stars:** [Christian Bale](#), [Steve Carell](#), [Ryan Gosling](#) | See full cast & crew »

81 Metascore From metacritic.com | 428 user | 454 critic | Popularity 434 (▲ 47)

Won 1 Oscar. Another 37 wins & 79 nominations. See more awards »

# Implementing Templates in Flask

# How to render a template with data

```
server.py x hello.html x
1 from flask import Flask
2 from flask import render_template
3 app = Flask(__name__)
4
5 @app.route('/')
6 def hello_world():
7     return 'Hello World'
8
9 @app.route('/hello/<name>')
10 def hello(name=None):
11     return render_template('hello.html', name=name)
12
13 if __name__ == '__main__':
14     app.run()
15
16
```

```
server.py x hello.html x
1 <html>
2 <head></head>
3 <body>
4
5 <b>Hello {{name}}!</b>
6 </body>
7 </html>
8
9
```

```
127.0.0.1:5000/hello/dave x +
127.0.0.1:5000/hello/dave
Hello dave!
```

# You can also send objects to templates

```
from flask import Flask
from flask import render_template
from flask import Response, request, jsonify
app = Flask(__name__)

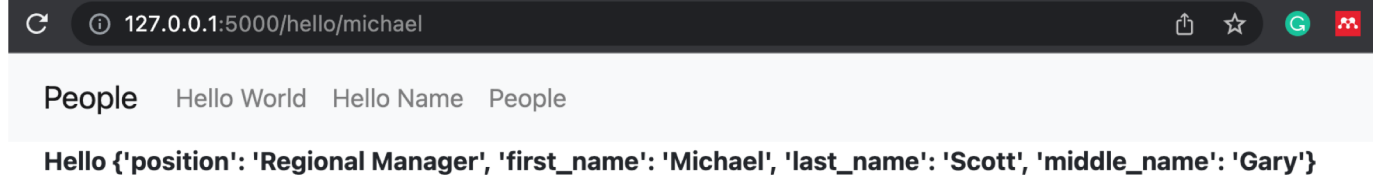
employees = {
    "michael":{
        "first_name": "Michael",
        "middle_name": "Gary",
        "last_name":"Scott",
        "position": "Regional Manager",
    },
    "dwight":{
        "first_name": "Dwight",
        "middle_name": "Kurt",
        "last_name":"Schrute",
        "position": "Assistant to the Regional Manager"
    }
}

# ROUTES
@app.route('/hello/<name>')
def hello_name(name=None):
    global employees

    employee = employees[name]

    return render_template('hello_name.html', employee=employee)
```

```
1 {% extends "layout.html" %}
2
3 {% block content %}
4
5 <b>Hello {{employee}} </b>!
6
7 {% endblock %}
8
9
```



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/hello/michael'. The page content includes a navigation menu with links for 'People', 'Hello World', 'Hello Name', and 'People'. Below the menu, the rendered output is displayed: 'Hello {'position': 'Regional Manager', 'first\_name': 'Michael', 'last\_name': 'Scott', 'middle\_name': 'Gary'}'.



# And access data within the object

```
from flask import Flask
from flask import render_template
from flask import Response, request, jsonify
app = Flask(__name__)

employees = {
    "michael":{
        "first_name": "Michael",
        "middle_name": "Gary",
        "last_name":"Scott",
        "position": "Regional Manager",
    },
    "dwight":{
        "first_name": "Dwight",
        "middle_name": "Kurt",
        "last_name":"Schrute",
        "position": "Assistant to the Regional Manager"
    }
}

# ROUTES
@app.route('/hello/<name>')
def hello_name(name=None):
    global employees

    employee = employees[name]

    return render_template('hello_name.html', employee=employee)
```

```
1  {% extends "layout.html" %}
2
3  {% block content %}
4
5  <b>Hello {{employee.first_name}} </b>!
6
7  {% endblock %}
8
9
```

127.0.0.1:5000/hello/michael

People Hello World Hello Name People

**Hello Michael !**

# And access it multiple times

```
from flask import Flask
from flask import render_template
from flask import Response, request, jsonify
app = Flask(__name__)

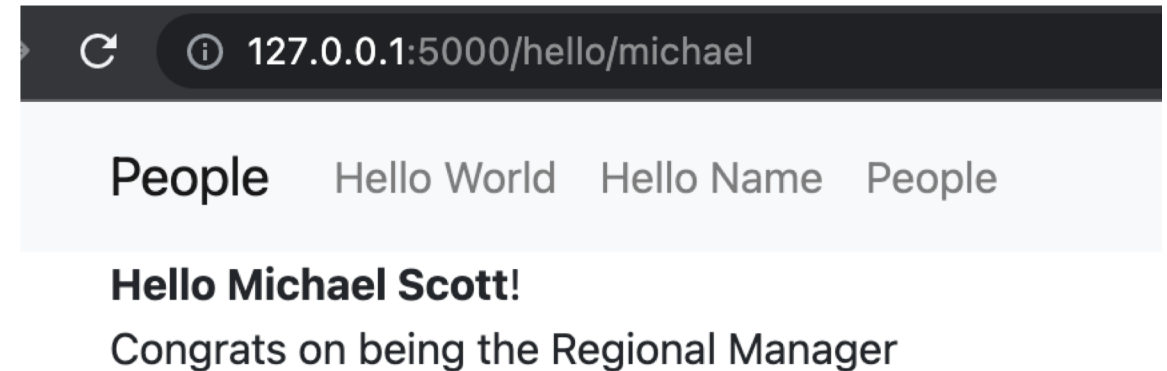
employees = {
    "michael":{
        "first_name": "Michael",
        "middle_name": "Gary",
        "last_name":"Scott",
        "position": "Regional Manager",
    },
    "dwight":{
        "first_name": "Dwight",
        "middle_name": "Kurt",
        "last_name":"Schrute",
        "position": "Assistant to the Regional Manager"
    }
}

# ROUTES
@app.route('/hello/<name>')
def hello_name(name=None):
    global employees

    employee = employees[name]

    return render_template('hello_name.html', employee=employee)
```

```
1  {% extends "layout.html" %}
2
3  {% block content %}
4
5  <b>Hello {{employee.first_name}} {{employee.last_name}}</b>!
6  <div>
7      Congrats on being the {{employee.position}}
8  </div>
9
10 {% endblock %}
11
```



# You can even loop over data.

## How can we show all Michael's awards?

```
from flask import Flask
from flask import render_template
from flask import Response, request, jsonify
app = Flask(__name__)

employees = {
    "michael":{
        "first_name": "Michael",
        "middle_name": "Gary",
        "last_name":"Scott",
        "position": "Regional Manager",
        "awards": [
            "Dundee 2001: Best Manager",
            "Dundee 2002: Best Manager",
            "Dundee 2003: Best Manager",
            "Dundee 2004: Best Manager",
        ],
    },
    "dwight":{
        "first_name": "Dwight",
        "middle_name": "Kurt",
        "last_name":"Schrute",
        "position": "Assistant to the Regional Manager"
    }
}

# ROUTES
@app.route('/hello/<name>')
def hello_name(name=None):
    global employees

    employee = employees[name]

    return render_template('hello_name.html', employee=employee)
```

```
1  {% extends "layout.html" %}
2
3  {% block content %}
4
5  <b>Hello {{employee.first_name}} {{employee.last_name}}</b>!
6  <div>
7      Congrats on being the {{employee.position}}
8  </div>
9  <br>
10
11 <div>
12     Awards:
13     <ul>
14         {% for award in employee.awards %}
15             <li>{{award}}</li>
16         {% endfor %}
17     </ul>
18 </div>
19
20 {% endblock %}
21
22
```

People Hello World Hello Name People

**Hello Michael Scott!**

Congrats on being the Regional Manager

Awards:

- Dundee 2001: Best Manager
- Dundee 2002: Best Manager
- Dundee 2003: Best Manager
- Dundee 2004: Best Manager

# Templates are a great way to show all the information of a database element

People Hello World Hello Name People

**Hello Michael Scott!**

Congrats on being the Regional Manager

Awards:

- Dundee 2001: Best Manager
- Dundee 2002: Best Manager
- Dundee 2003: Best Manager
- Dundee 2004: Best Manager

# Implementing Search on the Server

# What if we wanted to return all employees with any type of manager role. How do we search?

```
from flask import Flask
from flask import render_template
from flask import Response, request, jsonify
app = Flask(__name__)

employees = {
    "michael":{
        "first_name": "Michael",
        "middle_name": "Gary",
        "last_name":"Scott",
        "position": "Regional Manager",
    },
    "dwight":{
        "first_name": "Dwight",
        "middle_name": "Kurt",
        "last_name":"Schrute",
        "position": "Assistant to the Regional Manager"
    },
}

# ROUTES
@app.route('/hello/<name>')
def hello_name(name=None):
    global employees

    employee = employees[name]

    return render_template('hello_name.html', employee=employee)
```

1. Give the route a search term rather than a name. (like "manager" instead of "Michael")
2. Iterate over all the employees and find the matching ones.
3. Return an array of results (not just one result)
4. You could display the array of results using a template, but you could also just dynamically create the list in JavaScript. Both are fine.

# Homework 6 & 7: Building a CRUD Application

# HW6: Search Application Functionality (The R of CRUD)

- **Search** Data
- **View** Data
- Create Data
- Update Data
- Delete Data



# Pick a dataset and put 10 items in it by hand.

**NBA All-Stars**



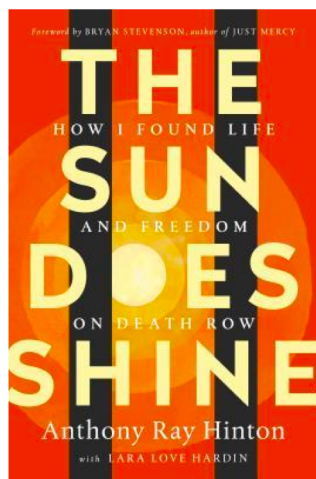
**Affordable make-up**



**Academy Award Winning Films**



**Book I want to read**



**Independent Coffee Shops in NYC**



**Fauvist Paintings**



# Have a user need in mind. Who needs this data? Why?

## **Family-owned** Ice cream parlors in the US



A college student is graduating and wants to road trip from NYC to Florida. Along the way, they want to find top-rated family-owned ice cream parlors so they can write about it for their travel blog.

# More specific needs are easier to design for.

Yoga poses



A person who has exercised or is looking to make a part of their body more flexible.

Top players for the 2019 baseball season



Someone who wants to prepare for his or her fantasy baseball draft

US Elections in 2018



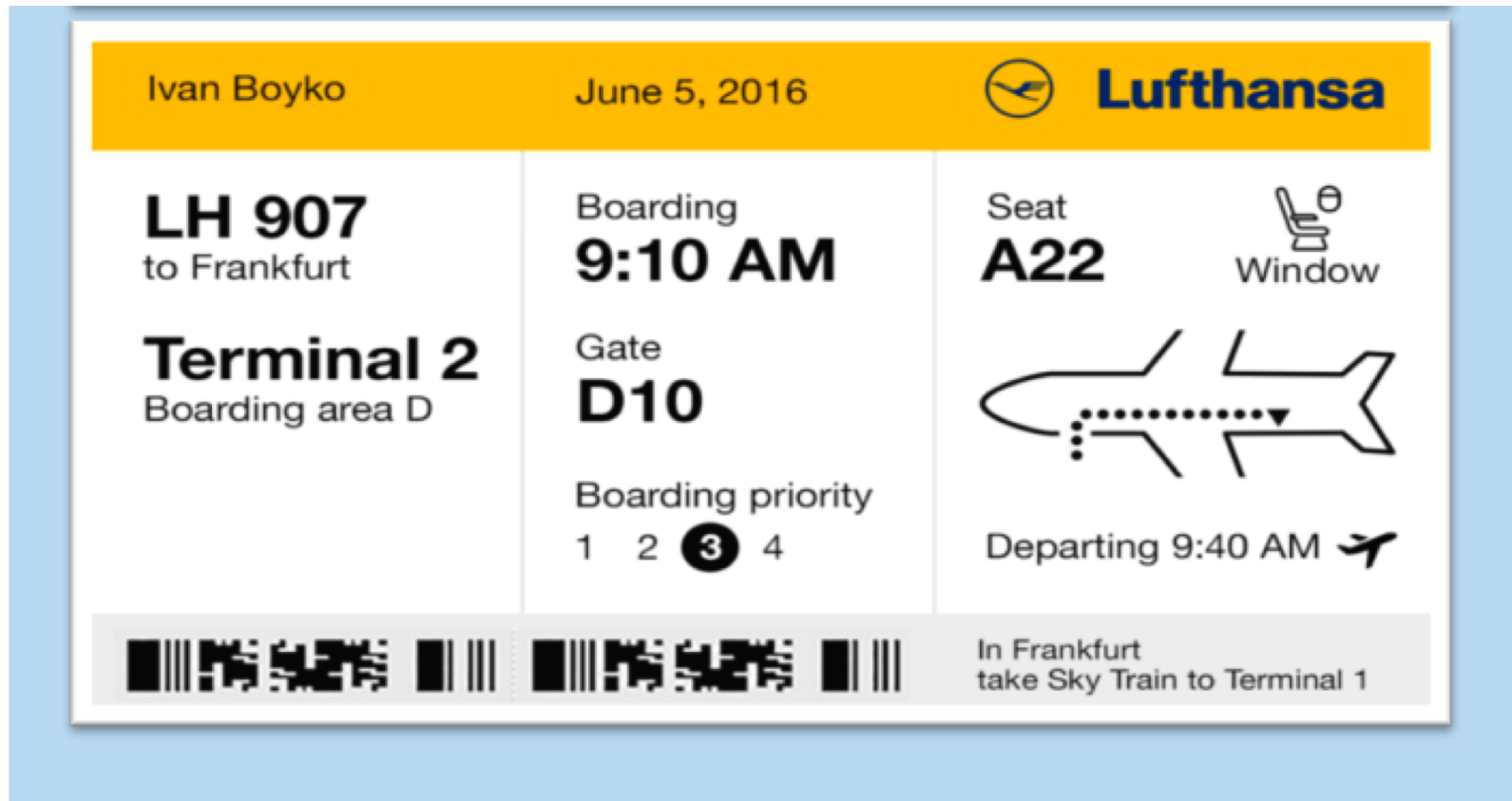
1. Any civically minded citizen or citizen who wants to be more well-read about our politics

2. A political candidate who wants to see how she or he will perform in their district based on the results in the 2018 election.

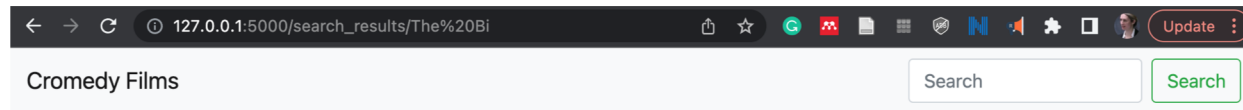
It is easier to make design decisions when the goal is clear.

What data to store, what interactions to enable.

Remember, to do good design, we need to know what the users goal is.



# HW6: Enable searching over data and viewing individual data elements (like IMDB).



## Search Results for "The Bi"

The Big Lebowski  
The Big Short



## The Big Lebowski (1998)



Score: 8.1 / 10

When 'The Dude' Lebowski is mistaken for a millionaire Lebowski, two thugs urinate on his rug to coerce him into paying a debt he knows nothing about. While attempting to gain recompense for the ruined rug from his wealthy counterpart, he accepts a one-time job with high pay-off. He enlists the help of his bowling buddy, Walter, a gun-toting Jewish-convert with anger issues. Deception leads to more trouble, and it soon seems that everyone from porn empire tycoons to nihilists want something from The Dude.

Director(s):

- Joel Coen
- Ethan Coen

Stars:

- Jeff Bridges
- John Goodman
- Julianne Moore

Genres:

- Comedy
- crime
- buddy

It doesn't have to be pretty,  
Just functional

I implemented this by starting with the “people” example and adding to it.

- When starting with another code base, be really sure to do iterative programming!
- Don't just delete stuff that you think you might need. It can create errors.

# Weird thing in Flask to watch out for...



## TypeError

`TypeError: Undefined is not JSON serializable`

### Traceback (most recent call last)

```
File "/Users/lydiachilton/Library/Python/2.7/lib/python/site-packages/flask/app.py", line 1836, in __call__
    return self.wsgi_app(environ, start_response)
File "/Users/lydiachilton/Library/Python/2.7/lib/python/site-packages/flask/app.py", line 1820, in wsgi_app
    response = self.make_response(self.handle_exception(e))
File "/Users/lydiachilton/Library/Python/2.7/lib/python/site-packages/flask/app.py", line 1403, in handle_exception
    reraise(exc_type, exc_value, tb)
File "/Users/lydiachilton/Library/Python/2.7/lib/python/site-packages/flask/app.py", line 1817, in wsgi_app
    response = self.full_dispatch_request()
File "/Users/lydiachilton/Library/Python/2.7/lib/python/site-packages/flask/app.py", line 1477, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/Users/lydiachilton/Library/Python/2.7/lib/python/site-packages/flask/app.py", line 1381, in handle_user_exception
    reraise(exc_type, exc_value, tb)
```

```
1  {% extends "layout.html" %}
2
3  {% block content %}
4
5  <script>
6      let data = {{data|tojson}}
7      console.log(data)
8  </script>
9
10
11 <b>Hello {{employee.first_name}} {{employee.last_name}}</b>!
12 <div>
13     Congrats on being the {{employee.position}}
14 </div>
15 <br>
16
```

Let's comment it out, and try again



The image is a composite meme. On the left, a browser window shows a JavaScript error: "TypeError: ..." with a "Traceback" section listing several "File "/User..." entries and code snippets like "return", "respon", "rerais", "respon", "rv = S", and "rerais". In the center is a meme image of a man with a confused expression, with the text "YOU MEAN THAT COMMENTS" at the top and "CAN GENERATE BUGS?" at the bottom. On the right, a code editor shows a snippet of code with a commented-out script: 

```
{% extends "layout.html" %}  
{% block content %}  
<script>  
    // let item = {{employees|tojson}}  
    // console.log(item)  
</script>
```

WHY????????



# Even when commented out, Flask will still replace all the `{{}}` syntax in the template with data (or die trying).



## TypeError

TypeError: Undefined is not JSON serializable

### Traceback (most recent call last)

```
File "/Users/lydiachilton/Library/Python/2.7/lib/python/site-packages/flask/app.py", line 1836, in __call__
    return self.wsgi_app(environ, start_response)
File "/Users/lydiachilton/Library/Python/2.7/lib/python/site-packages/flask/app.py", line 1820, in wsgi_app
    response = self.make_response(self.handle_exception(e))
File "/Users/lydiachilton/Library/Python/2.7/lib/python/site-packages/flask/app.py", line 1403, in handle_exception
    reraise(exc_type, exc_value, tb)
File "/Users/lydiachilton/Library/Python/2.7/lib/python/site-packages/flask/app.py", line 1817, in wsgi_app
    response = self.full_dispatch_request()
File "/Users/lydiachilton/Library/Python/2.7/lib/python/site-packages/flask/app.py", line 1477, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/Users/lydiachilton/Library/Python/2.7/lib/python/site-packages/flask/app.py", line 1381, in handle_user_exception
    reraise(exc_type, exc_value, tb)
File "/Users/lydiachilton/Library/Python/2.7/lib/python/site-packages/flask/app.py", line 1475, in full_dispatch_request
    rv = self.dispatch_request()
File "/Users/lydiachilton/Library/Python/2.7/lib/python/site-packages/flask/app.py", line 1461, in dispatch_request
```

```
1  {% extends "layout.html" %}
2
3  {% block content %}
4
5  <script>
6      // let item = {{employees|tojson}}
7      // console.log(item)
8
9  </script>
```

# Hint: How did I store my employees?

```
1 from flask import Flask
2 from flask import render_template
3 from flask import Response, request, jsonify
4 app = Flask(__name__)
5
6 employees = {
7     "michael":{
8         "first_name": "Michael",
9         "middle_name": "Gary",
10        "last_name":"Scott",
11        "position": "Regional Manager",
12    },
13    "dwight":{
14        "first_name": "Dwight",
15        "middle_name": "Kurt",
16        "last_name":"Schrute",
17        "position": "Assistant to the Regional Manager"
18    },
19 }
20
21 # ROUTES
22 @app.route('/people')
23 def people():
24     return render_template('people.html', data=employees)
25
26
27 @app.route('/hello/<name>')
28 def hello_name(name=None):
29     global employees
30
31     employee = employees[name]
32
33     return render_template('hello_name.html', employee=employee)
34
35
36
```

As a dictionary with the name as the key. That's not generally good.

- Definitely store your data as a dictionary.
  - Why? It makes it easy to retrieve items that way.
- Do not use the name/title as the key.
  - Why? Two items (people, movies, etc) have the same name. Remakes in film. Three ultra famous people named "Michael Jordan."

# Homework 6: Search Application Functionality

## Warm up: What data will you pick?

You must pick a dataset that someone might want to search. Your dataset needs to have at least 10 items in it. We suggest you create them by hand, like the list of local business were in the log\_sales application. Typically, 10 items are not enough to need search functionality for, but we are going to implement it anyway.

Each data item needs to have multiple fields. At the minimum, this must include:

- An id
- A short title or name for the item (movie title, restaurant name, etc)
- A link to media (image, video, or gif).
  - For large media files, you must provide an external link (like a link to a youtube video). Do NOT download the file and submit it with the assignment because if you did that, it will take your grader forever to download all the assignments.
- A text paragraph of explanation (At least 4 sentences)
- Some sort of numerical data (a year, a price, a rating, etc.)
- A list of some kind of data (such as a list of reviews for the movie, list of popular dishes at the restaurant, a list of similar restaurants nearby, etc)

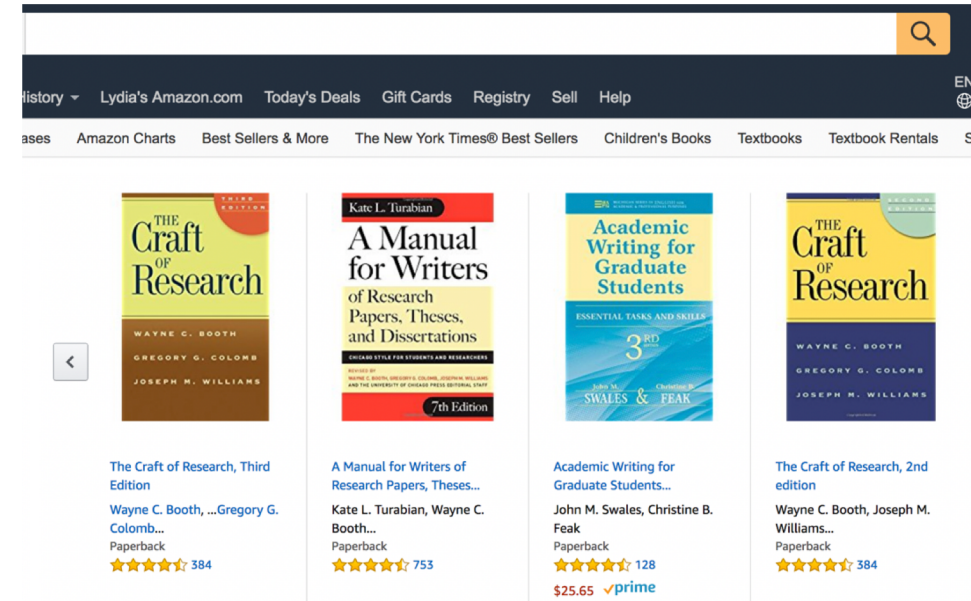
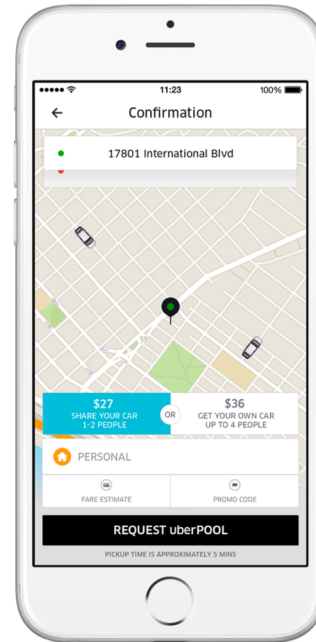
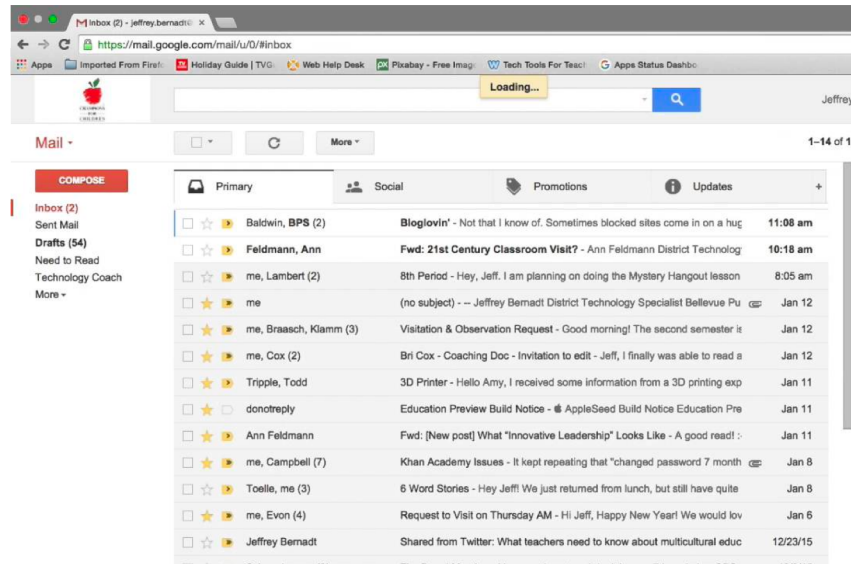
## Main: Code AND a video

### What to submit:

1. Write a short sentence describing your participation on Monday (unless instructed otherwise by your section TA).
2. Hw6\_UNI.zip: A Flask project containing:
  - server.py
  - templates/ (and the HTML templates you need)
  - static/ (and any static files you need)
3. A link to a YouTube video showing off the functionality of your site.

# Summary

# The main goal of many websites is to interact with data.



# CRUD: Operations for interacting with a database

## Create


**New Message**

To

From Lydia Chilton <chilton@cs.columbia.edu> ▾

Subject

Hey dude!

 What's on your mind, Lydia?

**Add title**

(No title)  
7 - 8pm

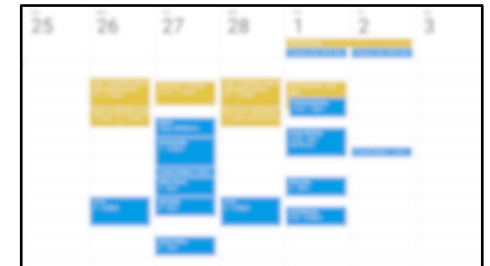
Event Reminder Appointment slots

Feb 28, 2018 7:00pm - 8:00pm Feb 28, 2018

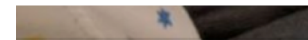
MORE OPTIONS SAVE

## Read

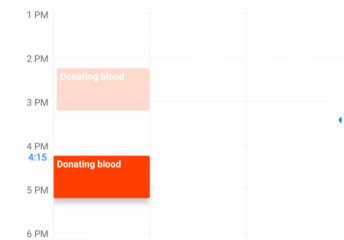
Baldwin, BPS (2)	Bioglovin' - Not that I know of. Sometimes bi
Feldmann, Ann	Fwd: 21st Century Classroom Visit? - Ann
me, Lambert (2)	8th Period - Hey, Jeff. I am planning on doing
me	(no subject) - -- Jeffrey Bernadt District Techr



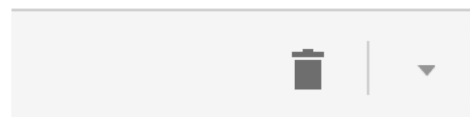
## Update



Like



## Delete



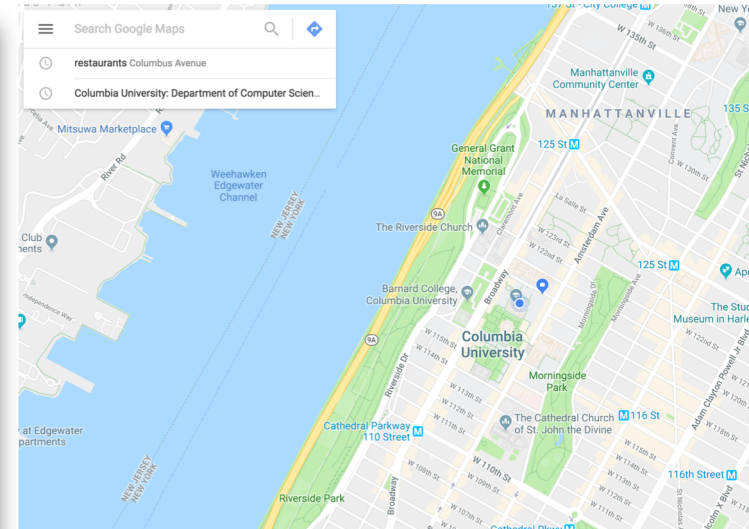
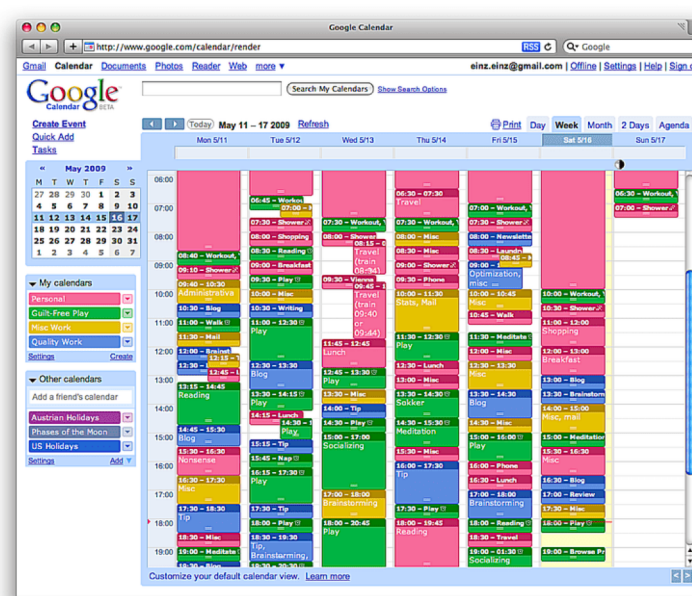
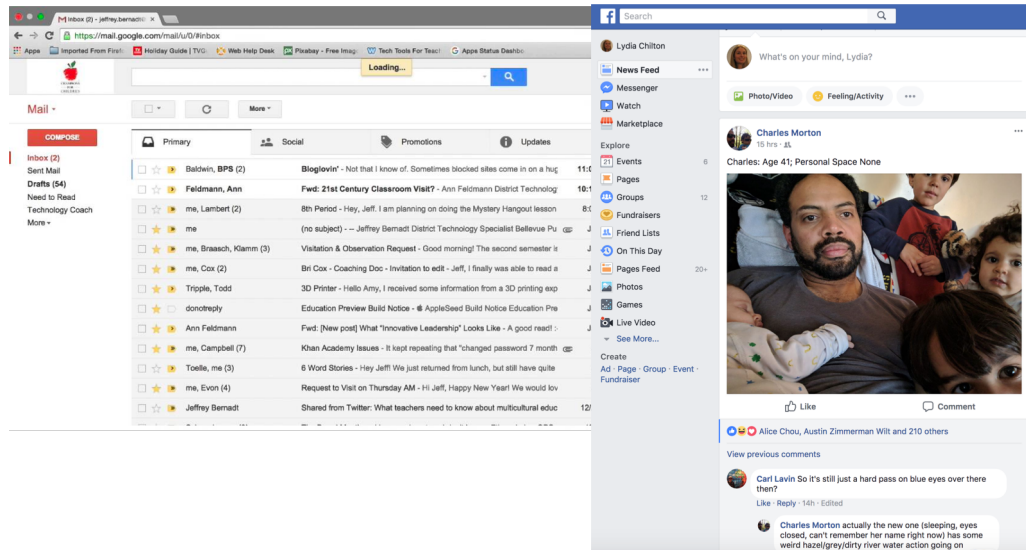
**Delete Post?**

Are you sure you want to delete it?

Delete Cancel

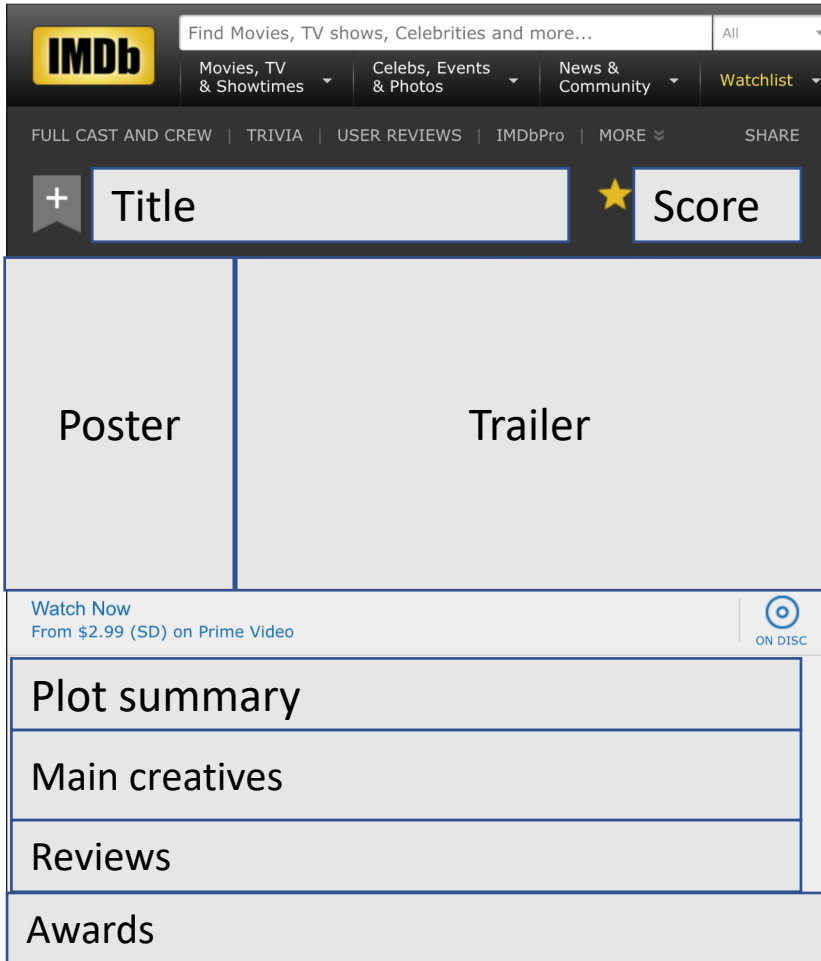


# From a back-end perspective Database-backed websites are very similar.



## But they serve different data and different information needs

# Use templates to dynamically generate pages from data in the database



Title	Plot summary	poster	year
The Big Lebowski	"The dude" Lebowski, mistaken...	Lebowski.jpg	1998
The Big Short	In 2006-7 a group of investors...	Big_short.jpg	2015
The Big Chill	A group of 7 former roommates	Big_chill.jpg	1983

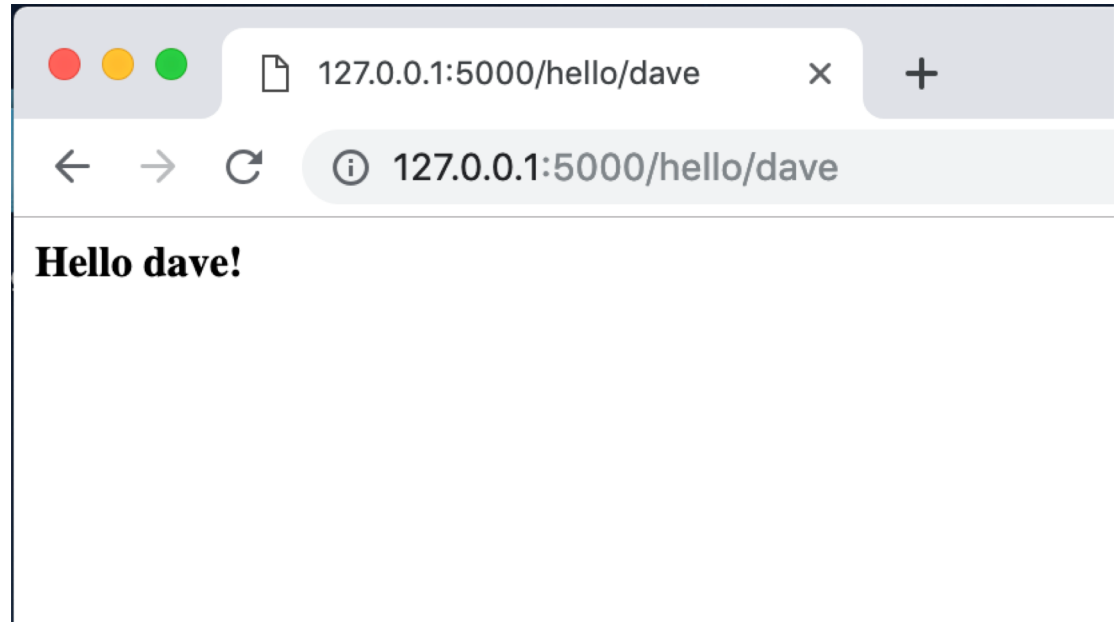




# How to render a template with data

```
server.py x hello.html x
1 from flask import Flask
2 from flask import render_template
3 app = Flask(__name__)
4
5 @app.route('/')
6 def hello_world():
7     return 'Hello World'
8
9 @app.route('/hello/<name>')
10 def hello(name=None):
11     return render_template('hello.html', name=name)
12
13 if __name__ == '__main__':
14     app.run()
15
16
```

```
server.py x hello.html x
1 <html>
2 <head></head>
3 <body>
4
5 <b>Hello {{name}}!</b>
6 </body>
7 </html>
8
9
```



# Homework 6: Search Application Functionality

## Warm up: What data will you pick?

You must pick a dataset that someone might want to search. Your dataset needs to have at least 10 items in it. We suggest you create them by hand, like the list of local business were in the log\_sales application. Typically, 10 items are not enough to need search functionality for, but we are going to implement it anyway.

Each data item needs to have multiple fields. At the minimum, this must include:

- An id
- A short title or name for the item (movie title, restaurant name, etc)
- A link to media (image, video, or gif).
  - For large media files, you must provide an external link (like a link to a youtube video). Do NOT download the file and submit it with the assignment because if you did that, it will take your grader forever to download all the assignments.
- A text paragraph of explanation (At least 4 sentences)
- Some sort of numerical data (a year, a price, a rating, etc.)
- A list of some kind of data (such as a list of reviews for the movie, list of popular dishes at the restaurant, a list of similar restaurants nearby, etc)

## Main: Code AND a video

### What to submit:

1. Write a short sentence describing your participation on Monday (unless instructed otherwise by your section TA).
2. Hw6\_UNI.zip: A Flask project containing:
  - server.py
  - templates/ (and the HTML templates you need)
  - static/ (and any static files you need)
3. A link to a YouTube video showing off the functionality of your site.