Database-Backed Websites

No screens

Prof. Lydia Chilton
COMS 4170
26 February 2020
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)
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"
    }
]
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

```javascript
profiles = [
{
  "name": "maddy",
  "image": "/maddy.png",
  "likes": "1000",
  "dislikes": 0,
},
{
  "name": "julia",
  "image": "/julia.png",
  "likes": "1000",
  "dislikes": 0,
},
]
```
YouTube: Database of videos
Facebook: Database of **posts**
GCalendar: Database of **events**
Google Maps: A database of locations
New York Times: Database of news articles
Craigslist: Database of sales posts

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

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: 6494768599 posted: 12 days ago updated: 14 minute ago email to friend best of
We need to have another computer **store and serve** the data. That server is running a Python application called **Flask**.
This website has no database. Why not?

User Interface Design
COMS 4170 · Spring 2018

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.
Interacting with Data

CRUD: Create, Read, Update, and Delete data
Gmail is a database of emails

<table>
<thead>
<tr>
<th>ID</th>
<th>From</th>
<th>To</th>
<th>Subject</th>
<th>Body</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Baldwin</td>
<td>Jeffrey</td>
<td>Blogloving’</td>
<td>Not that I know of...</td>
<td>158904993585835</td>
</tr>
<tr>
<td>2</td>
<td>Ann Feldman</td>
<td>Jeffrey</td>
<td>Fwd: Classroom visit?</td>
<td>Ann Feldman District Technology</td>
<td>158904993585923</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...
How do we store multiple users’ data?

<table>
<thead>
<tr>
<th>ID</th>
<th>From</th>
<th>To</th>
<th>Subject</th>
<th>Body</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Baldwin</td>
<td>Jeffrey</td>
<td>Blogloving’</td>
<td>Not that I know of...</td>
<td>158904993585835</td>
</tr>
<tr>
<td>2</td>
<td>Ann Feldman</td>
<td>Jeffrey</td>
<td>Fwd: Classroom visit?</td>
<td>Ann Feldman District Technology</td>
<td>158904993585923</td>
</tr>
<tr>
<td>3</td>
<td>Lydia Chilton</td>
<td>Lee Bollinger</td>
<td>Bowling tuesday</td>
<td>Hey dude! Donnie and I were thinking of going bowling...</td>
<td>158904993443772</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CRUD operations on data: 
*Create, Read, Update, and Delete Data*

<table>
<thead>
<tr>
<th>ID</th>
<th>From</th>
<th>To</th>
<th>Subject</th>
<th>Body</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Baldwin</td>
<td>Jeffrey</td>
<td>Blogloving’</td>
<td>Not that I know of…</td>
<td>158904993585835</td>
</tr>
<tr>
<td>2</td>
<td>Ann Feldman</td>
<td>Jeffrey</td>
<td>Fwd: Classroom visit?</td>
<td>Ann Feldman District Technology</td>
<td>158904993585923</td>
</tr>
<tr>
<td>3</td>
<td>Lydia Chilton</td>
<td>Lee Bollinger</td>
<td>Bowling tuesday</td>
<td>Hey dude! Donnie and I were thinking of going bowling…</td>
<td>158904993443772</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>ID</th>
<th>creator</th>
<th>text</th>
<th>Likes</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Charles Morton</td>
<td>Charles Age 41: Personal..</td>
<td>210</td>
<td>15890499</td>
</tr>
</tbody>
</table>
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 autoload 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

<table>
<thead>
<tr>
<th>Creator</th>
<th>Start date</th>
<th>End Date</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chilton</td>
<td>Feb 28, 7pm</td>
<td>Feb 28, 8pm</td>
<td>Dinner with Lee</td>
</tr>
<tr>
<td>Chilton</td>
<td>Feb 14, 4pm</td>
<td>Feb 17, 12pm</td>
<td>National Academy of Sciences event</td>
</tr>
</tbody>
</table>
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**

**Read**

**Update**

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

<table>
<thead>
<tr>
<th>Title</th>
<th>Plot summary</th>
<th>poster</th>
<th>year</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Big Lebowski</td>
<td>“The dude” Lebowski, mistaken...</td>
<td>Lebowski.jpg</td>
<td>1998</td>
</tr>
<tr>
<td>The Big Short</td>
<td>In 2006-7 a group of investors...</td>
<td>Big_short.jpg</td>
<td>2015</td>
</tr>
<tr>
<td>The Big Chill</td>
<td>A group of 7 former roommates</td>
<td>Big_chill.jpg</td>
<td>1983</td>
</tr>
</tbody>
</table>
What goes in the template?
What goes in the template?

### Title

The title area is where you would input the title of the movie or TV show.

- **IMDb**
  - Find Movies, TV shows, Celebrities and more...
  - FULL CAST AND CREW
  - TRIVIA
  - USER REVIEWS
  - IMDbPro
  - MORE
  - SHARE

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

---

- **Description**
- **Stills and Posters**
- **Genre**
- **Director**
- **Cast**
- **Release Date**
- **Runtime**

---

- **Reviews**
- **Critics' Ratings**
- **Audience Ratings**
- **Box Office
collections**

---

- **Trivia**
- **Behind the Scenes**
- **Interviews**
- **Deleted Scenes**

---

- **Videos**
- **Trailers**
- **Music Videos**
- **Shorts and Vlogs**

---

- **Community**
  - Discussion
  - FanTheories
  - FanArt

---

- **Watchlist**
- **List of Similar Movies**
- **Most Viewed Movies**
- **Most Popular Titles**

---

- **Explore**
  - IMDb Originals
  - IMDb Kids
  - IMDb TV
  - IMDb Originals

---

- **Settings**
- **Notifications**
- **Account**
- **Help & Feedback**
What goes in the template?
What goes in the template?

[TITLE]

[Score]
What goes in the template?
What goes in the template?

- **TITLE**
- **Score**
- Poster Image
What goes in the template?

**TITLE**

**Score**

Poster Image

???????
What goes in the template?

**TITLE**

**Score**

**Poster Image**

**Trailer Video**
IMDB uses a template and dynamically fills the template from a database query.
Anyone remember the other categories?
Anyone remember the other categories?
Anyone remember the other categories?
IMDB Template
Templates are standardizations. What can go wrong?
Templates are standardizations. What can go wrong?
Templates are standardizations. What can go wrong?
Standardization is hard.
Expect to **iterate** on your templates.
Implementing Templates in Flask
How to render a template with data

```python
from flask import Flask
from flask import render_template

app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello World'

@app.route('/hello/<name>/')
def hello(name=None):
    if name == '__main__':
        return render_template('hello.html', name=name)
    return 'Hello {{name}}!'

if __name__ == '__main__':
    app.run()
```

```html
<html>
<head></head>
<body>
<p>Hello {{name}}!</p>
</body>
</html>
```
Homework 6 & 7: Building a Search Application
HW6:
Search Application Part 1: Functionality
(due Wed. 3/4)

• Search Data
• Create Data
• View Data
• Update Data
• Delete Data

• Submit:
  • code
  • short write up
HW7: Search Application Part 2: **Usability** (due Wed. 3/11)

- Information hierarchy
- Font, color, gestalt
- Feedback
- Accessibility
- Widgets
- More Ajax

- Submit:
  - code
  - short write up
  - **video**
Pick a dataset and put 30 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 soreness 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 their fantasy baseball draft.

US Elections in 2018

- A political candidate who wants to see how safe the incumbent in their district based on the results in the 2018 election.
- Any civically minded citizen or citizen who wants to be more well-read about our politics.

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

- What data to store, what interactions to enable.
HW7 is the midterm in this class.

- We can’t assess mastery of the material in a 2 hour written test, so we create an assignment that cover all the material in the first half
- You are still allowed to ask for help in the normal ways.
  - (Even masters of this material ask for help)
- The same late policies apply. No more than 5 days after the deadline, even with a note.
- No office hours or Piazza help after Friday 3/13 5pm.
Summary
The main goal of many websites is to interact with data.
CRUD: Operations for interacting with a database

Create

Read

Update

Delete
From a back-end perspective, database-backed websites are very similar. But they serve different data and different information needs.
<table>
<thead>
<tr>
<th>Title</th>
<th>Plot summary</th>
<th>poster</th>
<th>year</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Big Lebowski</td>
<td>“The dude” Lebowski, mistaken...</td>
<td>Lebowski.jpg</td>
<td>1998</td>
</tr>
<tr>
<td>The Big Short</td>
<td>In 2006-7 a group of investors...</td>
<td>Big_short.jpg</td>
<td>2015</td>
</tr>
<tr>
<td>The Big Chill</td>
<td>A group of 7 former roommates</td>
<td>Big_chill.jpg</td>
<td>1983</td>
</tr>
</tbody>
</table>
How to render a template with data

```python
from flask import Flask
from flask import render_template

app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello World'

@app.route('/hello/<name>/')
def hello(name=None):
    if name == '__main__':
        return render_template('hello.html', name=name)
    return render_template('hello.html', name=name)

if __name__ == '__main__':
    app.run()
```

```
<html>
<head></head>
<body>
<p>Hello {{name}}!</p>
</body>
</html>
```
Fill out participation now!
HW 6 is out.