

# Technical Prototypes

Prof. Lydia Chilton  
COMS 4170  
13 April 2022

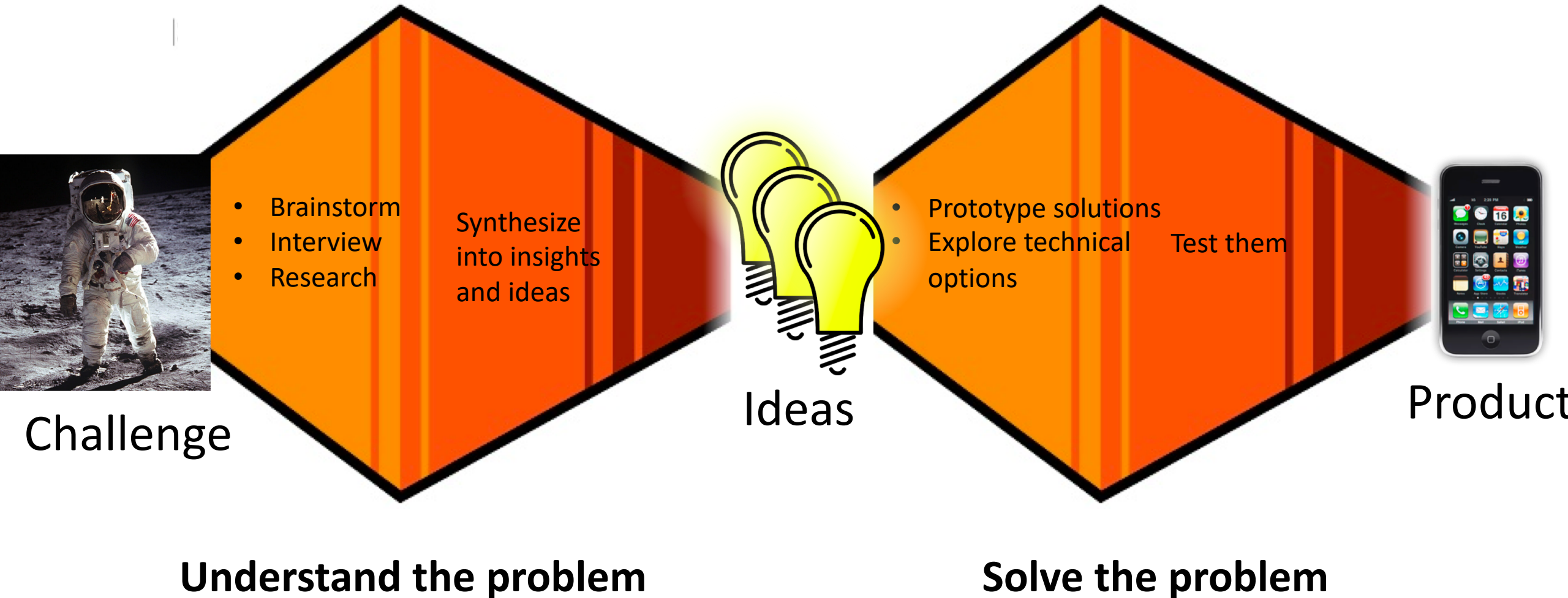
Raise your hand or type in zoom



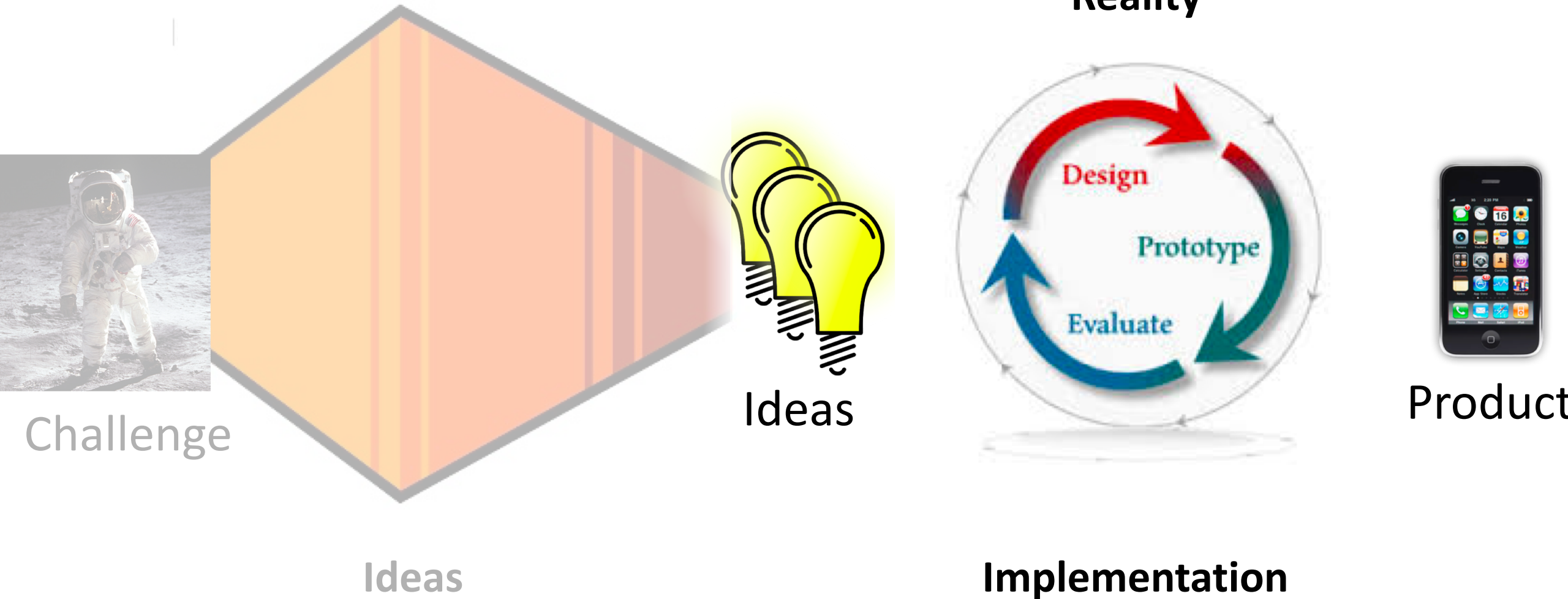
# Design Challenge:

- Design an interface to help a user **learn** an introductory topic **interactively**
- And help them assess themselves with a **quiz**.

# The Double Diamond Process



# Implementation is iterative.



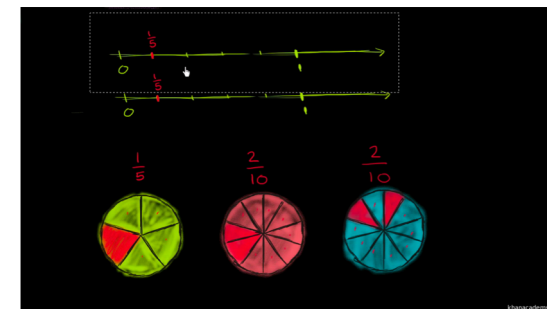
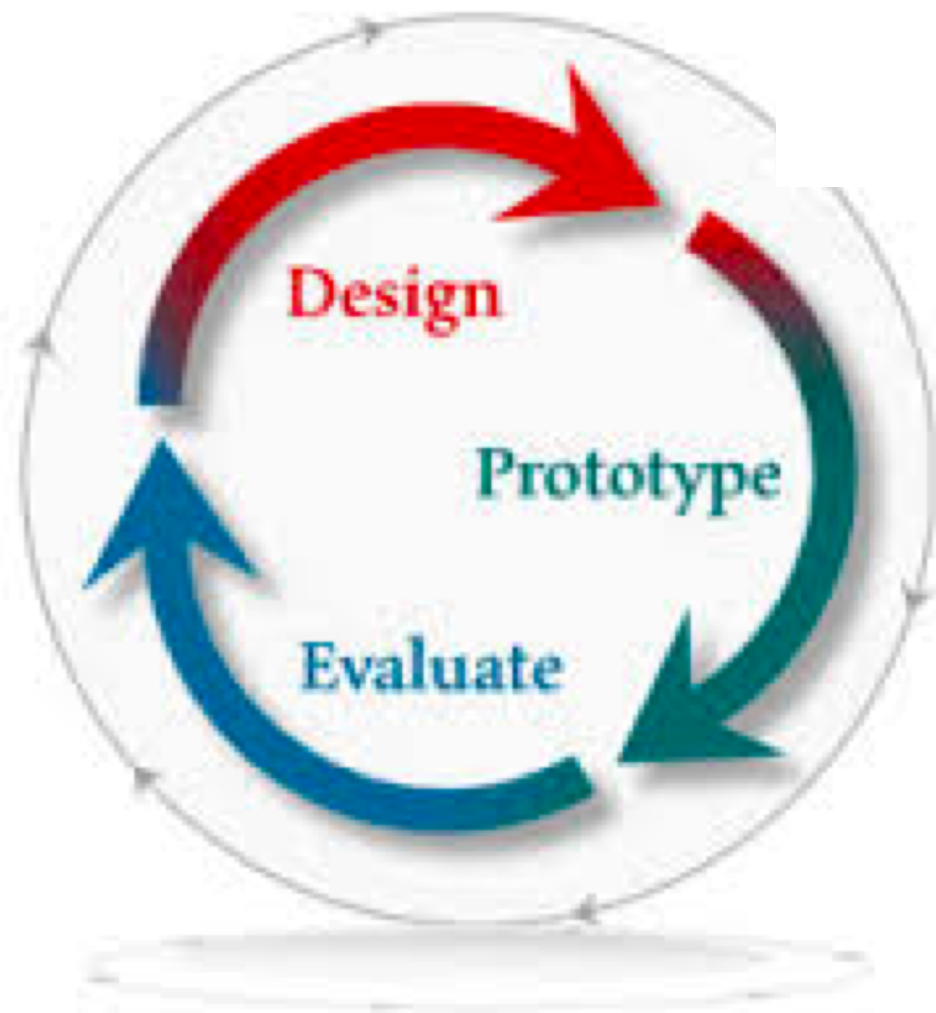
The **user** is at the center of the process



# Translating an idea into a prototype is HARD.

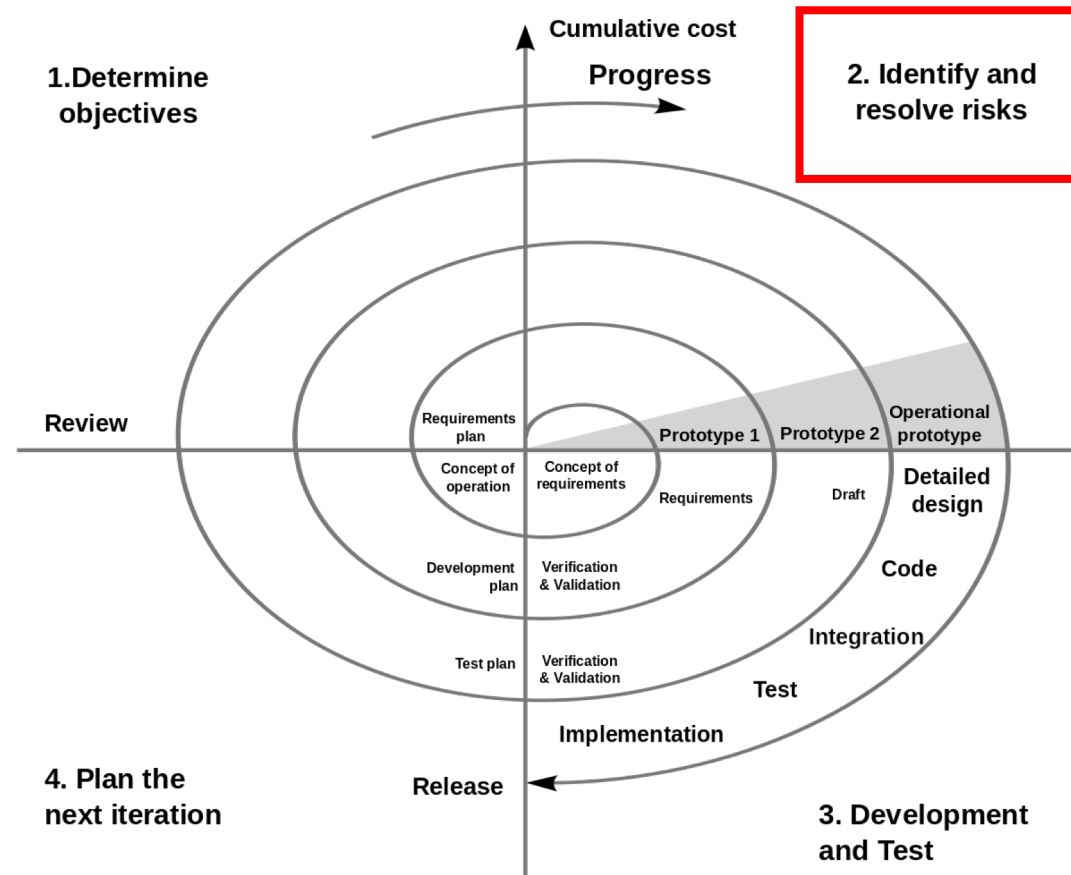


Teach fractions by working through problems slowly.



Low-Fi Prototype

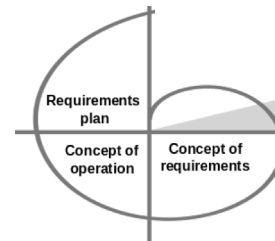
# Iterative Design is good because it minimizes risk



# The first iteration should be as low-fidelity as possible

**1. Determine objectives**

**2. Identify and resolve risks**



**4. Plan the next iteration**

**3. Development and Test**



# Low- and Mid-Fi Prototypes in Google Slides

Low-Fi Prototype 1

File Edit View Tools Help Last edit was made on April 22 by Kingsley Neequaye

Present Share

View only

1 Low-Fi Prototype 1

2 YOU AND THE POLICE

3

4

5


6

7

HOME POLICE AT YOUR DOOR PULLED OVER STOPPED IN PUBLIC ARRESTED TEST

## Police are at your door

Next section



**Your Rights**

- To remain silent
  - This means xyz
- To see a warrant
  - This means you do not have to let them in until you see a warrant

[What are warrants?](#)

**Tips to reduce risk to yourself**

DO

- Ask the officer if you can see a warrant and ID
  - Ask the officer to slip the warrant under the door or hold it up to the window so you can read it
- Write down everything officers do if you are searched

DO NOT

- Invite the officer into your house without seeing ID and warrant
- Speak to the officers about anything

Indoor Fitness Prototype

File Edit View Tools Help Last edit was made on April 11 by William Hall

Present Share

View only

1

2

3




4

5

Workout Indoors Exercises Routine FAQs QUIZ

## Exercises

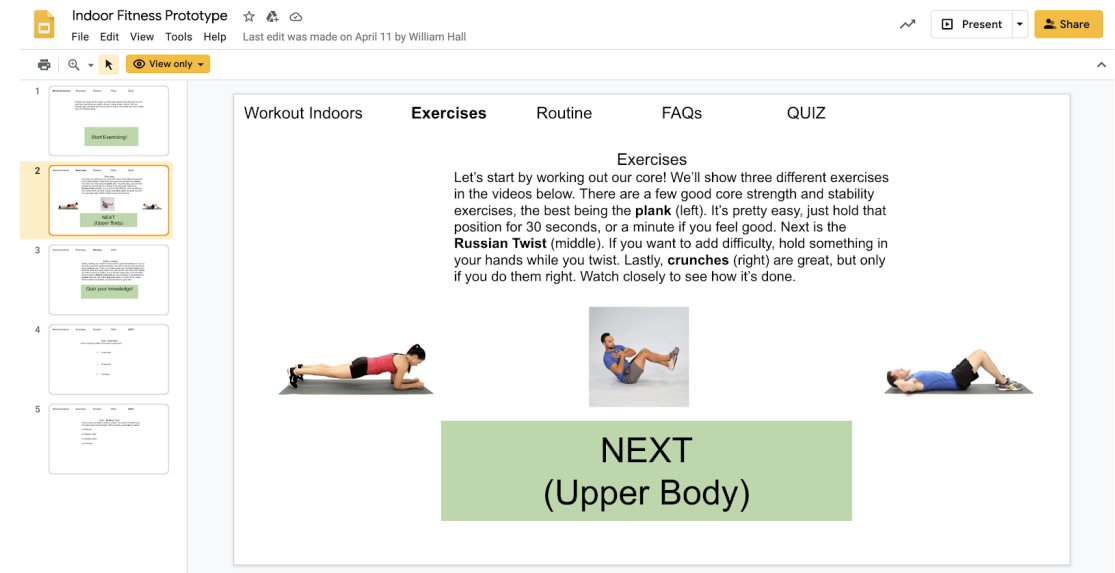
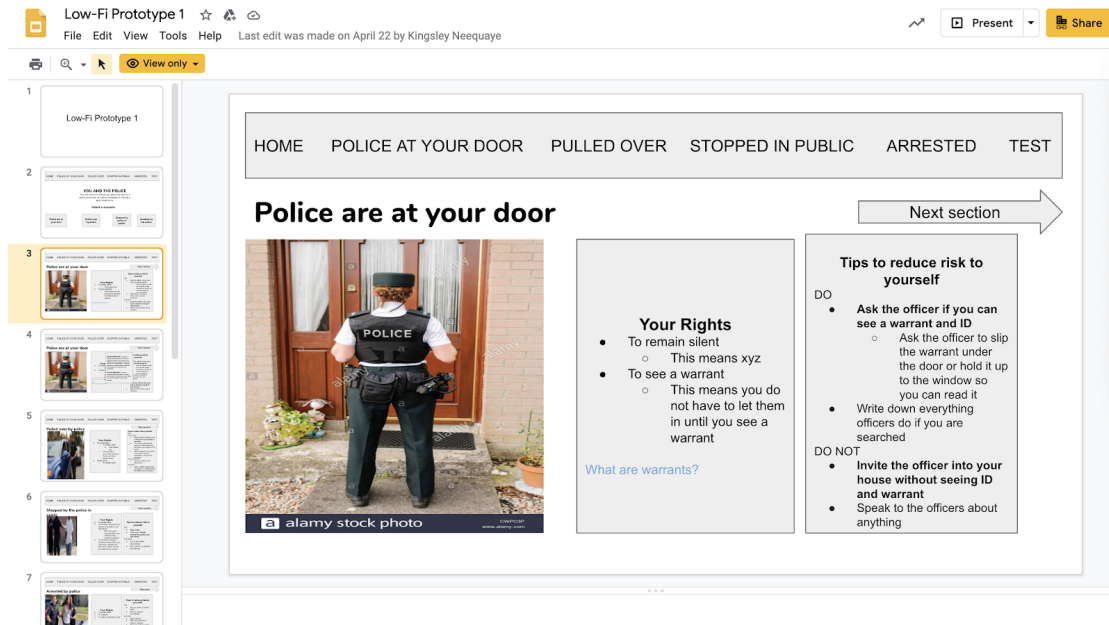
Let's start by working out our core! We'll show three different exercises in the videos below. There are a few good core strength and stability exercises, the best being the **plank** (left). It's pretty easy, just hold that position for 30 seconds, or a minute if you feel good. Next is the **Russian Twist** (middle). If you want to add difficulty, hold something in your hands while you twist. Lastly, **crunches** (right) are great, but only if you do them right. Watch closely to see how it's done.



**NEXT**  
(Upper Body)

With feedback from TA's and user tests

# HW10: Prototypes in Google Slides



With TA feedback and feedback from user tests

Now you're ready to start  
implementing

But we're going to do it iteratively

# HW11: Technical Prototype

**General Goal:** Figure out what's the biggest technical risk, and create a prototype to test it.

**Your HW 11 Goal:** Program and end-to-end experience that allows the user to get through the application.

(no graphic design, little interactivity)

What counts as a technical risk?

Anything new.

Anything that has never been done before.

Anything **you** have never been done before.

# We're going to watch a video prototype

- List everything that's risky (anything new)



# What tech need to be prototyped?

Drones following you!  
And projecting



Menu selection on the ground



Menu selection on the hand



Multi user interaction



Menu selection on the wall



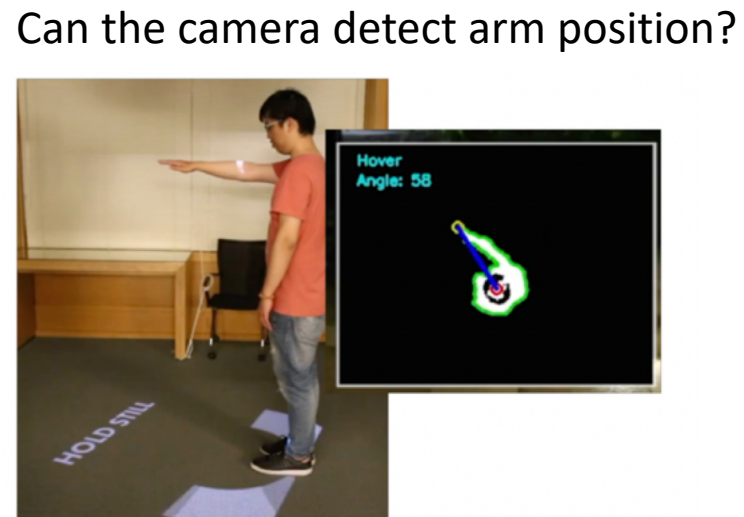
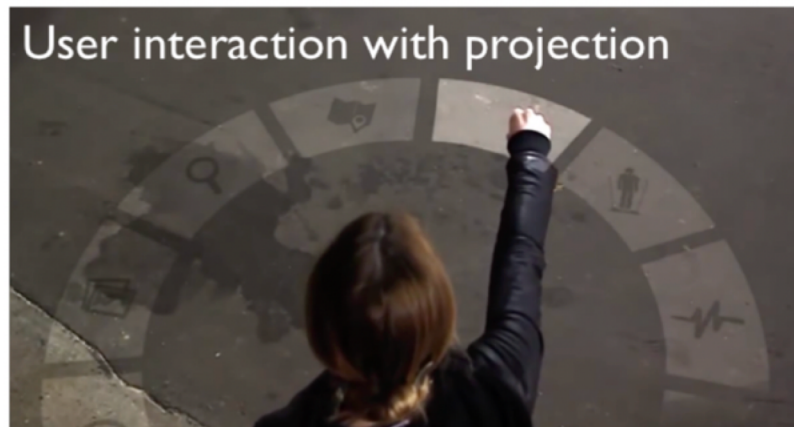


Now we're going to watch a video of the things they prototyped

- Which risks did they prototype?
  - How?
- Which risks did they not prototype?



# What did they prototype? How?



During the outdoor test,  
what new risk did they discover?



How could they have mitigated this risk?



They probably should have tested projecting in flight outdoors before implementing the menu.

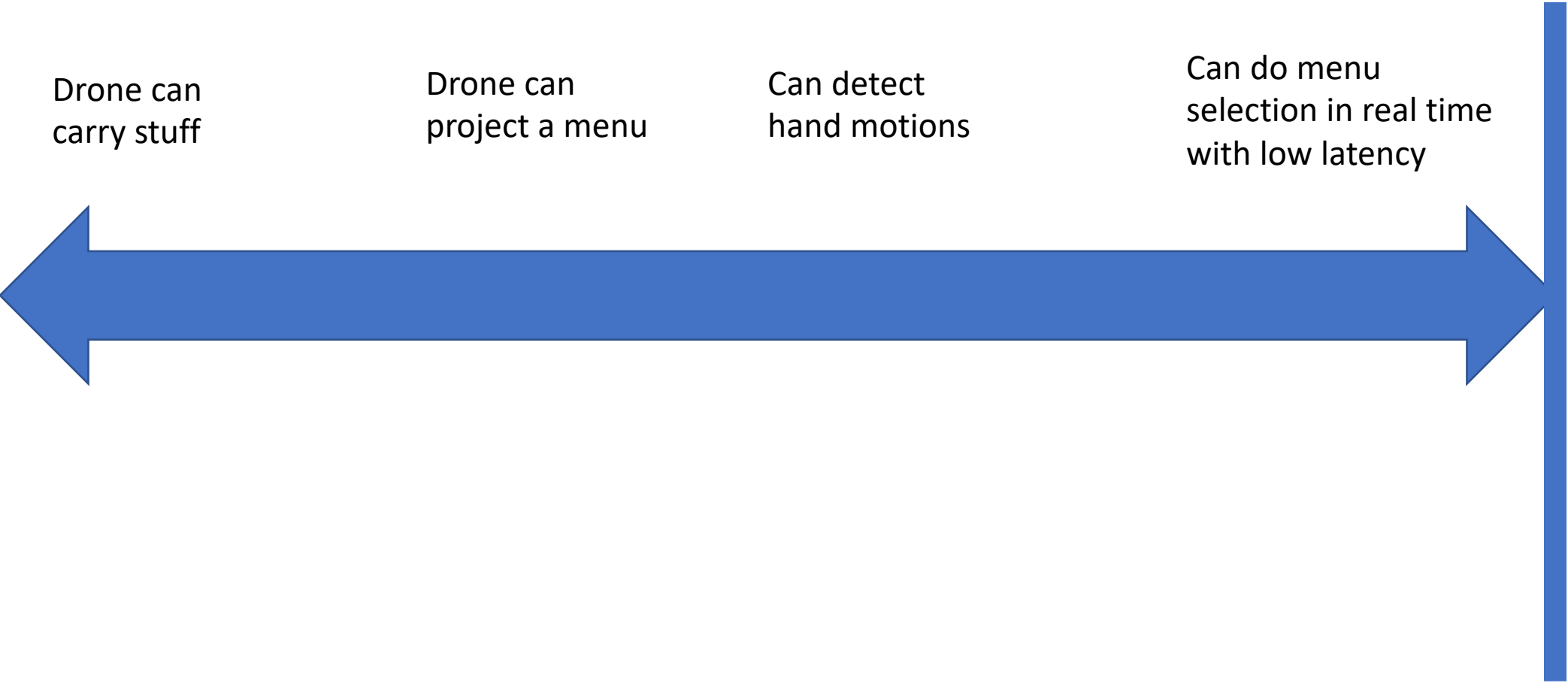
# Everything in the end-to-end process is a risk

Drone can  
carry stuff

Drone can  
project a menu

Can detect  
hand motions

Can do menu  
selection in real time  
with low latency



But not outdoors

# HW 11: Technical Prototypes

Your biggest risk is having an end-to-end working system.

Start lesson

Do lesson

Next lesson logic

Start quiz

Quiz Logic

Quiz Score





# 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 /10 28,073 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 | 4,222 (↑ 509) Popularity

**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 /10 609,031 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 | 515 (↓ 32) Popularity

**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 /10 274,582 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

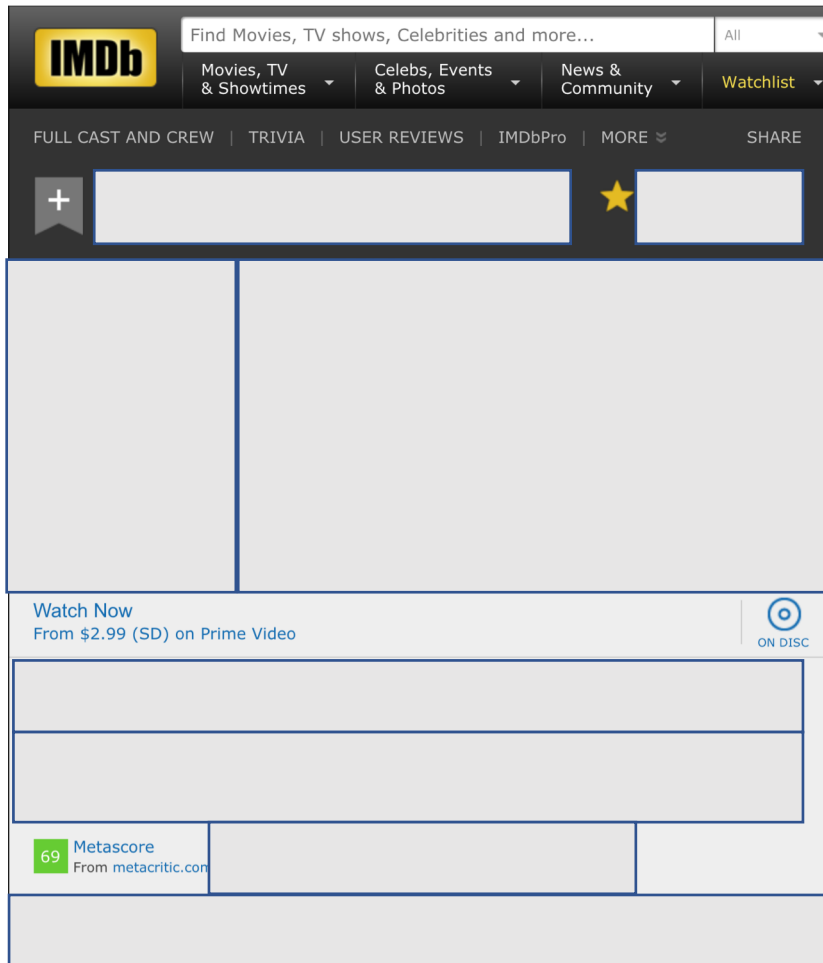
In 2006-7 a group of investors bet against the US mortgage market. In their research they discover how flawed and corrupt the market is.

**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 | 434 (★ 47) Popularity

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

# 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



# IMDB Template

The image shows a screenshot of the IMDb website template. At the top, there is a search bar with the text "Find Movies, TV shows, Celebrities and more..." and a dropdown menu set to "All". Below the search bar are navigation tabs: "Movies, TV & Showtimes", "Celebs, Events & Photos", "News & Community", and "Watchlist". A secondary navigation bar includes "FULL CAST AND CREW", "TRIVIA", "USER REVIEWS", "IMDbPro", "MORE", and "SHARE".

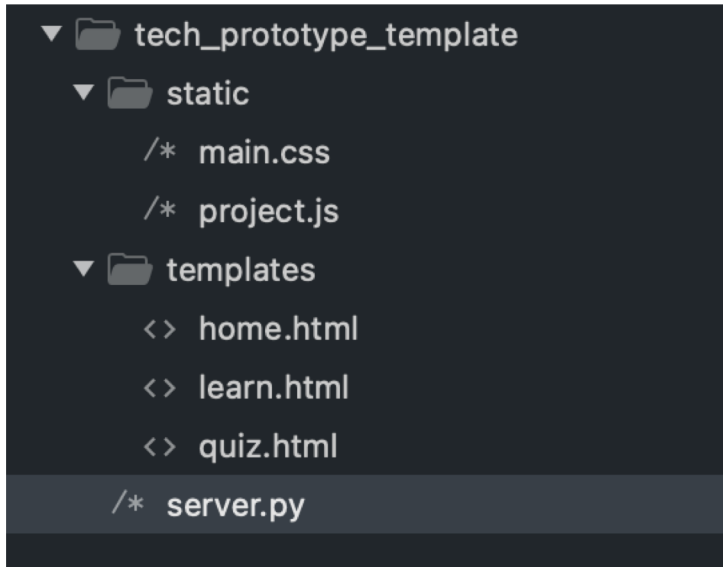
The main content area features a header with a plus sign icon, a text input field containing the placeholder `{{movie.title}}`, a star icon, and another text input field containing the placeholder `{{movie.star}}`. Below this header is a large rectangular area divided into two columns. The left column contains the HTML code ``.

Below the large area is a "Watch Now" section with the text "From \$2.99 (SD) on Prime Video" and an "ON DISC" icon. This section is followed by a series of stacked rectangular boxes containing the following placeholders: `{{movie.plot_summary}}`, `{{movie.main_creatives}}`, `{{movie.reviews}}`, and `{{movie.awards}}`.

# Why are static pages so bad?

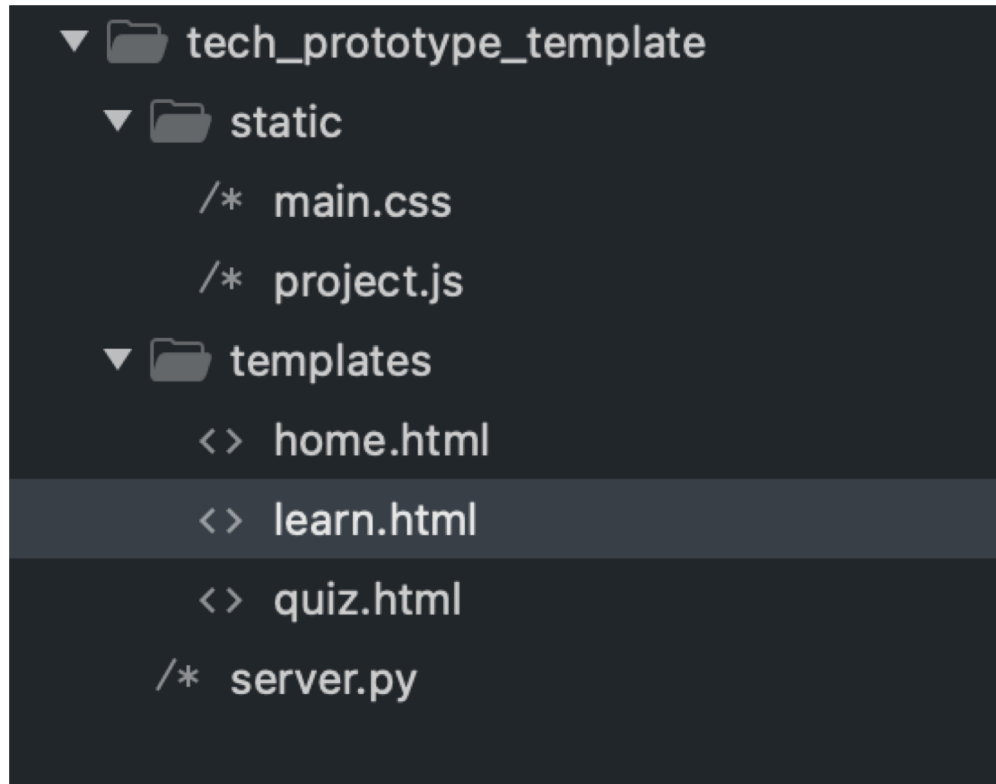
- They are very hard to change.

# Represent your data in JSON (server.py)



```
1 from flask import Flask
2 from flask import render_template
3 from flask import Response, request, jsonify
4 app = Flask(__name__)
5
6
7 lessons = {
8     "1":{
9         "lesson_id": "1",
10        "title": "Say Hello in Serbian",
11        "video": "https://media4.giphy.com/media/f4IitLXgceg0BzXUDe/200w.gif?cid=
12        "text": "Zdravo",
13        "next_lesson":"2"
14    },
15    },
16    "2":{
17        "lesson_id": "2",
18        "title": "Say Goodbye in Serbian",
19        "video": "https://c.tenor.com/6zCJ2mbdXg4AAAAC/ciao.gif",
20        "text": "Ciao",
21        "next_lesson":"end"
22    }
23 }
24
25 quiz_questions = {
26     "1":{
27         "quiz_id": "1",
28         "english": "hello",
29         "serbian": "zdravo",
30         "next_question":"2"
31     },
32     "2":{
33         "quiz_id": "2",
34         "english": "goodbye",
35         "serbian": "ciao",
36         "next_question":"end"
37     }
38 }
39
```

# Render learning and quizzing templates with correct data



```
1 from flask import Flask
2 from flask import render_template
3 from flask import Response, request, jsonify
4 app = Flask(__name__)
5
6
7 lessons = {
8     "1":{
9         "lesson_id": "1",
10        "title": "Say Hello in Serbian",
11        "video": "https://media4.giphy.com/media/f4IitLXgceg0BzXUDe/200w.gif?cid=82a1493bcb6xt",
12        "text": "Zdravo",
13        "next_lesson":"2"
14    },
15    },
16    "2":{
17        "lesson_id": "2",
18        "title": "Say Goodbye in Serbian",
19        "video": "https://c.tenor.com/6zCJ2mbdXg4AAAAC/ciao.gif",
20        "text": "Ciao",
21        "next_lesson":"end"
22    }
23 }
24
25
41 @app.route('/')
42 def home():
43     return render_template('home.html')
44
45 @app.route('/learn/<lesson_id>')
46 def learn(lesson_id):
47     lesson = lessons[lesson_id]
48     return render_template('learn.html', lesson = lesson)
49
50 @app.route('/quiz/<quiz_id>')
51 def quiz(quiz_id):
52     question = quiz_questions[quiz_id]
53     return render_template('quiz.html', question = question)
54
55 if __name__ == '__main__':
56     app.run(debug = True)
```

# On each page, show the correct data.

Learn.html template



Say Hello in Serbian

Serbian: Zdravo

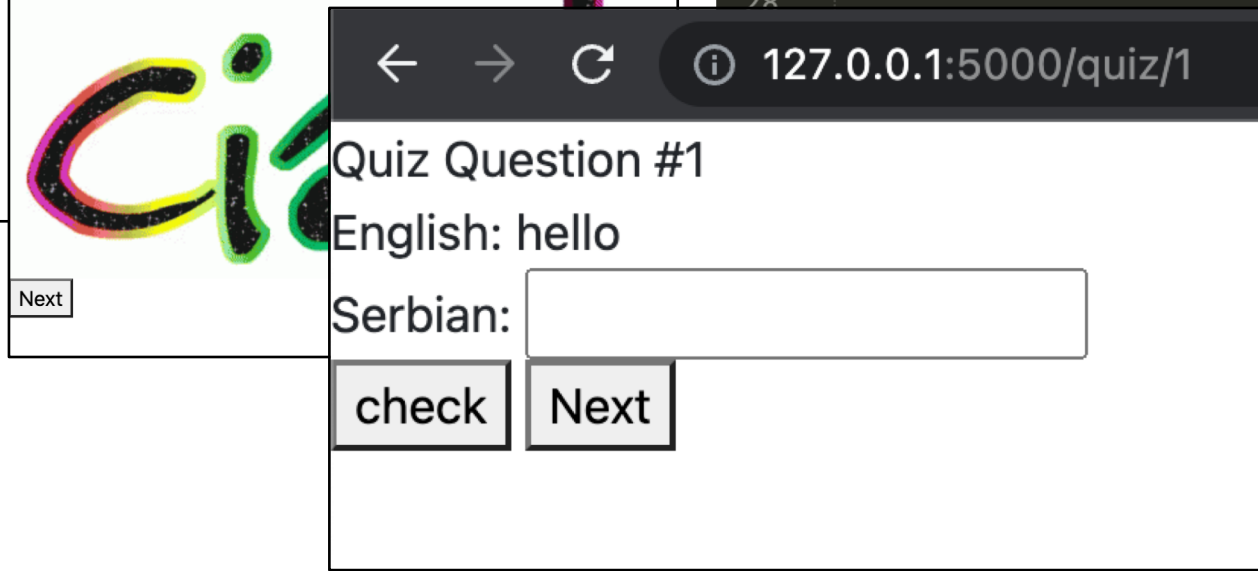
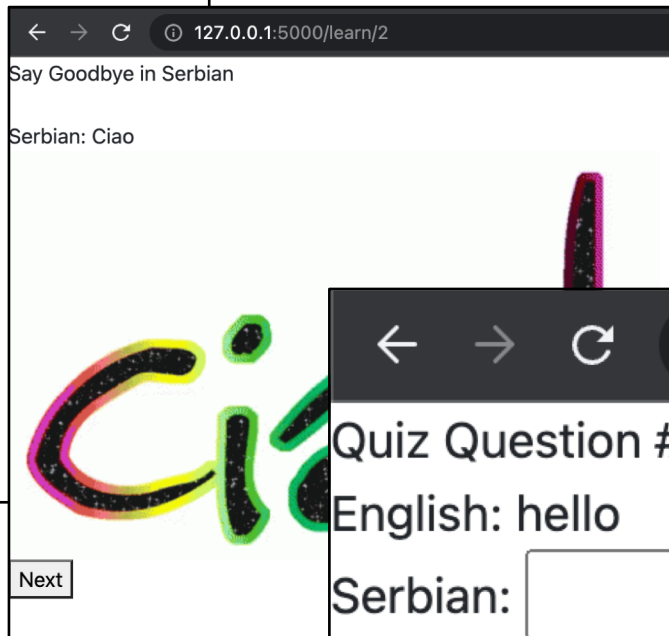
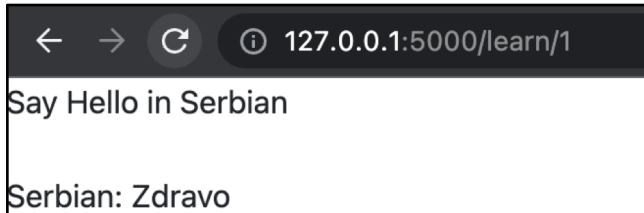


Next

```
15 <body>
16 <script type="text/javascript">
17   let lesson = {{lesson|tojson}}
18
19   $( document ).ready(function() {
20     $("#next_button").click(function(){
21       if("{{lesson.next_lesson}}" == "end"){
22         window.location.href = "/quiz/1"
23       }else{
24         window.location.href = "/learn/{{lesson.next_lesson}}"
25       }
26     })
27   })
28
29 </script>
30
31 {{lesson.title}}<br>
32 <br>
33 Serbian: {{lesson.text}}<br>
34 <br>
35
36 <button id="next_button"> Next </button>
37
38
39 </body>
```

# On each page, have logic that advances to the next state.

Learn.html template



```
15 <body>
16 <script type="text/javascript">
17   let lesson = {{lesson|tojson}}
18
19   $( document ).ready(function() {
20     $("#next_button").click(function(){
21       if("{{lesson.next_lesson}}" == "end"){
22         window.location.href = "/quiz/1"
23       }else{
24         window.location.href = "/learn/{{lesson.next_lesson}}"
25       }
26     })
27   })
28
```

```
<br>
o}}"><br>
"> Next </button>
```



# Why is representing your application in data (rather than in HTML) so good?

## Easy to change **content**

```
6
7 lessons = {
8   "1":{
9     "lesson_id": "1",
10    "title": "Say Hello in Serbian",
11    "video": "https://media4.giphy.com/
12    "text": "Zdravo",
13    "next_lesson":"2"
14
```

## Easy to change **logic**

```
6
7 lessons = {
8   "1":{
9     "lesson_id": "1",
10    "title": "Say Hello in Serbian",
11    "video": "https://media4.giphy.com/
12    "text": "Zdravo",
13    "next_lesson":"2"
14
```

```
16 <script type="text/javascript">
17   let lesson = {{lesson|tojson}}
18
19   $( document ).ready(function() {
20     $("#next_button").click(function(){
21       if("{{lesson.next_lesson}}" == "end"){
22         window.location.href = "/quiz/1"
23       }else{
24         window.location.href = "/learn/{{lesson.next_lesson}}"
25       }
26     })
27   })
28
29 </script>
30
```

# Your biggest risk is having an end-to-end working system.

Start lesson

Do lesson

Next lesson logic

Start quiz

Quiz Logic

Quiz Score



Unless you have some huge risky interactive feature!  
If do you, contact your TA at once, and we'll modify HW11 for you.

# Divide up work with roles

- **Learning portion** roles:

- Architecting the data
- Implementing the UI
- Testing that you can click through the app. (this cannot be done by the learning UI implementer)

- **Quiz portion** roles:

- Architecting the data
- Implementing the UI
- Testing that you can click through the app. (this cannot be done by the quiz UI implementer)

If this does not work out for your group, contact your TA immediately.