## Planning Your Mid-Term

Or: Why you're **not** ready to code yet



## Pick a project

- 1. Wiki Map
- 2. Card Game
- 3. Decision Maker
- 4. Smart TODO List
- 5. Resource Wall
- 6. Schoodle
- 7. Food Pick-Up Ordering



### Where do I start?



# DATA



### **Everything starts with data**

What is the data you can access or acquire?

 Can you correlate that data to something else?

How can you pivot that data to add value?



# FEATURES VS. BENEFITS



UNIVERSAL VALUES

# Look good

Feel good

Save time

Save money



# USER STORIES

\* - You're not ready to code.



## What is a User Story?

```
"As a _____,
I want to _____
Because ___."
```



## **Another variant on a User Story**

```
"As a _____,
I should be able to _____."
Because ___."
```



### **User Scenarios**

"Given

When \_\_\_\_

Then \_\_\_\_.'



### User should be able to save story

Traditional User Story

Scenario

Additional info

As a user
I want to save a story I'm reading
Because I found it useful

Given that I'm reading a story
When I tap the *icon* to save a story
Then save it to my 'Saved Stories'

iver in items in

### User should be able to save story

As a user I want to save a story I'm reading Because I found it useful

Given that I'm reading a story
When I tap the icon to save a story
Then save it to my 'Saved Stories'
'And' And alter the *icon* to indicate success

iver in items in

## FEATURE COMPLETE

WAIT, WHAT IS AN MVP?





# Decide on your A-HA moment.





# If you aren't going to demo it... DON'T BUILD IT

FAQ / About pages

Signup pages

<controversy>Login pages</controversy>



## Write a login route for yourself \*

```
app.get('/login/:id', (req, res) => {
  req.session.user_id = req.params.id;
  res.redirect('/');
});
```

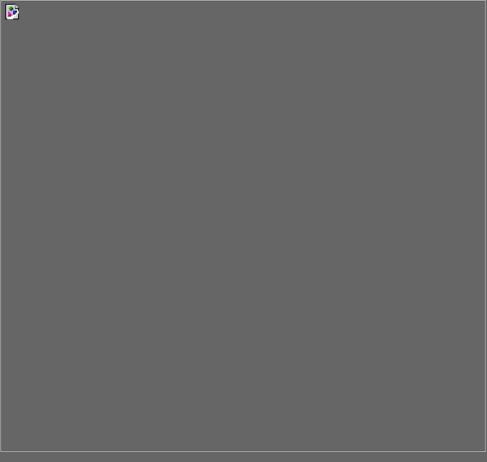
\* - NEVER do this in production



# **ERD**

\* - You're still not ready to code.





# TABLES are NOUNS



#### User should be able to save story

As a user I want to save a story I'm reading Because I found it useful

Given that I'm reading a story
When I tap the icon to save a story
Then save it to my 'Saved Stories'
'And' And alter the *icon* to indicate success

iver in items in

# TECH CHOICES

## DEPLOYMENT

\* - You're still not ready to code.



### **Two Options**

http://localhost:3000

Deploying to HEROKU



# SERVER FRAMEWORK

\* - You're still not ready to code.



## For your Mid-terms, this one is easy:



express



## Don't forget about





## ROUTES

\* - You're still not ready to code.



### You have to plan your routes

Points to consider:

GET vs. POST vs. PUT/PATCH vs. DELETE

Params in URLs?

RESTful conventions

If tables are nouns, routes are the verbs.



### User should be able to save story

As a user I want to save a story I'm reading Because I found it useful

Given that I'm reading a story
When I tap the icon to save a story
Then save it to my 'Saved Stories'
'And' And alter the *icon* to indicate success

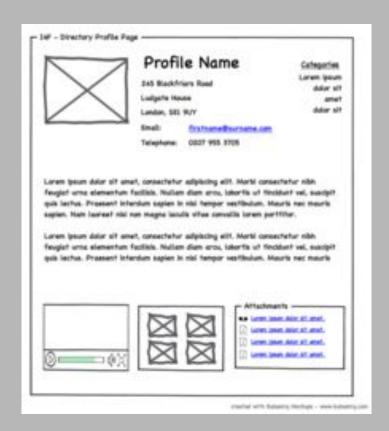
iver in items in

# WIREFRAME

\* - You're still not ready to code.



### What is this wireframe of which you speak?



Wireframes are a way to describe layout, define data hierarchy, and demonstrate interactions.



### **IMAGE**

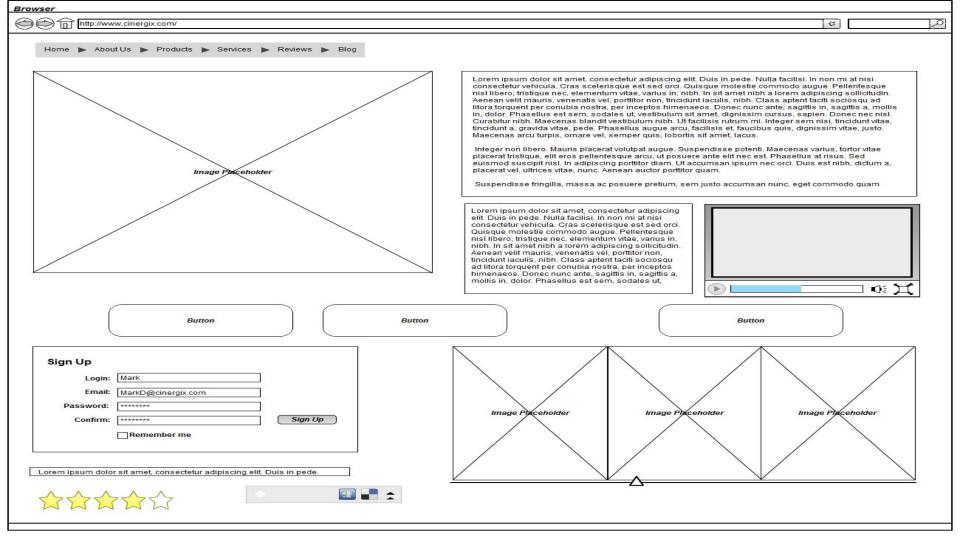
Yep, makes sense.

#### Title goes here

**Notifications** 

Totally.

WTF?



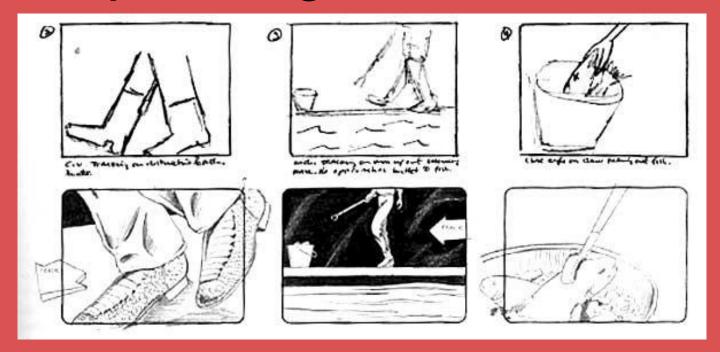
### Storyboarding

Storyboarding is the art of telling a story in a chronological format. Deciding what the scenes are, how they progress from one point in time to another, is important to be able to tell the story.

This applies to the web as well.



#### Storyboarding in a comic / movie





#### Wireframing for the web is the same

You want to set up a timeline of your user workflow. Figure out what the ideal path is for users to take when going through the features you're building.

Guess where you can get that path from?



#### User should be able to save story

As a user I want to save a story I'm reading Because I found it useful

Given that I'm reading a story
When I tap the icon to save a story
Then save it to my 'Saved Stories'
'And' And alter the *icon* to indicate success

iver in items in

# DESIGN

\* - You're still not ready to code.



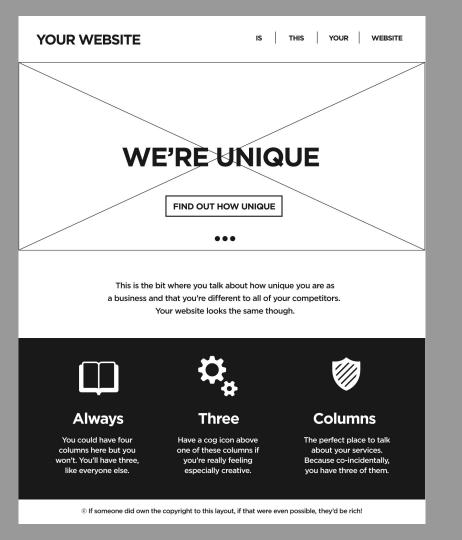
#### Design matters.

#### HOWEVER

You are developers, not designers!

Use resources like Dribbble, FolioFocus, and Smashing.





Every.
Website.
Ever.



#### A WISE Man once said:



"Doing Development WITHOUT a Design is LIKE DOING ORIGAMI BLINDFOLDED.

Yes, you will get FOLDED Paper, it will not be a pretty crane."

### **UI FRAMEWORK**

\* - You're still not ready to code.



#### Grids make building easier

They	allow	you to	place
content	easily	on the	page.

But which one do I use?



#### **Bootstrap**

Groundwork

Skeleton

Black Hole

Bulma

Min

Bourbon

+ Neat

**PureCSS** 

**Foundation** 



### Which one do I use?



#### Nice and easy!

# **B** Bootstrap

OR





# GIT

\* - You're still not ready to code.



### Remember talking about

this on Day 1?



#### Use good git practices!!

- 1. Clone
- 2. Branch
- 3. Code! \*
- 4. Checkout master
- 5. Pull\*
- 6. Merge\*
- 7. Push
- 8. Repeat from Step 2

\* - TEST!



#### Please...

### Don't code on master!



# SCAFFOLDING

\* - You're almost ready to code.





#### CHECKLIST FOR SCAFFOLDING

- Set up GitHub repository, add collaborators
- Download / Install all resources
- Use CDN links whenever possible
- Make sure your app loads, after installing things
- Push app to GitHub, so team can clone
- Decide on team responsibilities



## **DIVIDING TASKS**

\* - You're practically ready to code.



#### Tasks can be divided according to:

- Front-end
- Back-end
- APIs
- JS
- Database
- Feature

- Paired
- Migrations
- Responsive
- Deployment
- Whatever works



# READY TO CODE!





#### Tips for Front-end Scaffolding

- Start with static pages first. If the CSS doesn't work here, making it dynamic won't help.
- Decide on class and ID nomenclature. This is a team process!
- Use placeholders like fillmurray.com and Ipsum generators (Bacon Ipsum, Hipster Ipsum, etc)
- Try your layouts with different quantities of data. Don't always put in "good" data, either.



#### Tips for Back-end Scaffolding

- Start with your data objects, always.
- Make sure it works in the REPL before trying to make it work in the app.
- Code atomically! Use helper methods. Refactor. Modularize.



### COMMUNICATION



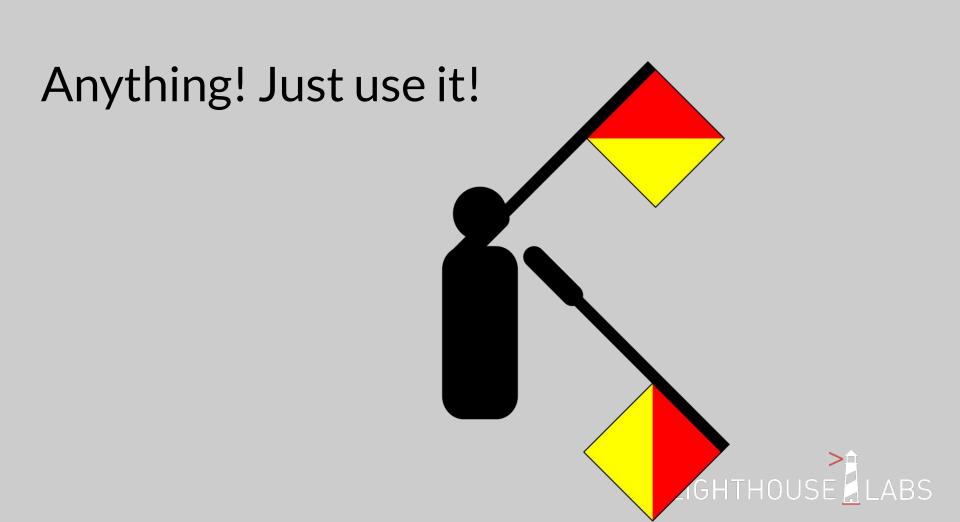
#### There are many communication tools

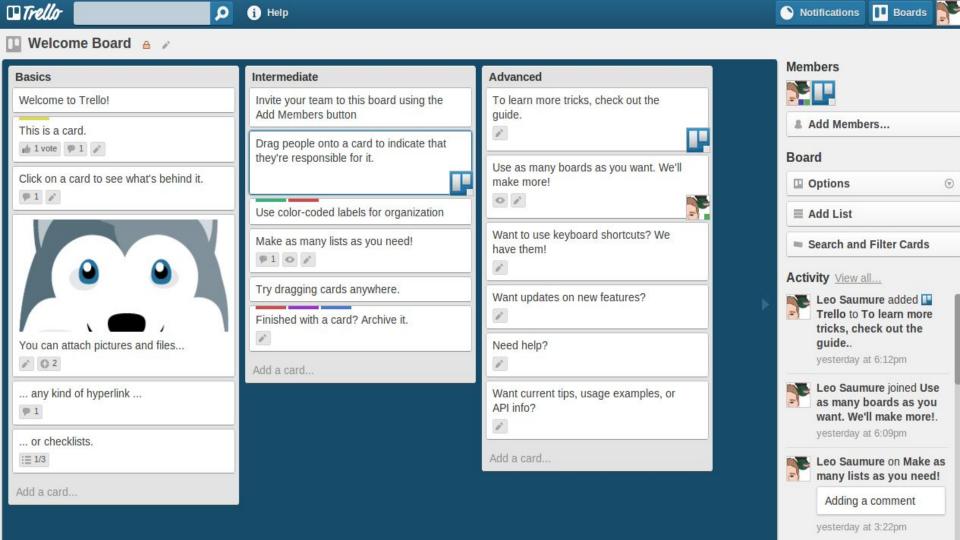


Pick one!











### Thank you.

