Midterm

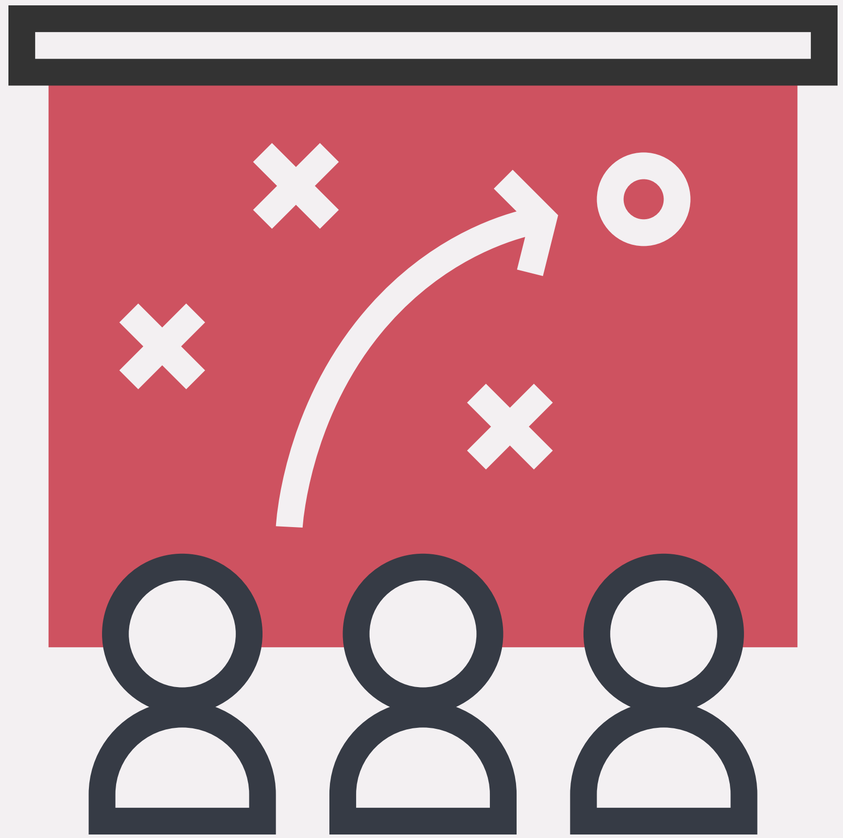
Midterm
Midterm

W4D3

Pick a Project

Wiki Map
Card Game
Decision Maker
Smart TODO List
Resource Wall
Schoodle
Food Pick-Up
Ordering

Planning



User Stories

📖 User Stories

As a... **Role**

I want to... **Goal**

Because... **Benefit**

📖 Scenarios

Given… **Context**

When… **Action**

Then… **Result**

Title

User should be able to save a story

User Story

As a user

I want to save a story I’m reading

Because I found it useful

Scenario

Given that I’m reading a story

When I tap the bookmark icon to save a story

Then save it to my ‘Saved Stories’

Features

MVP

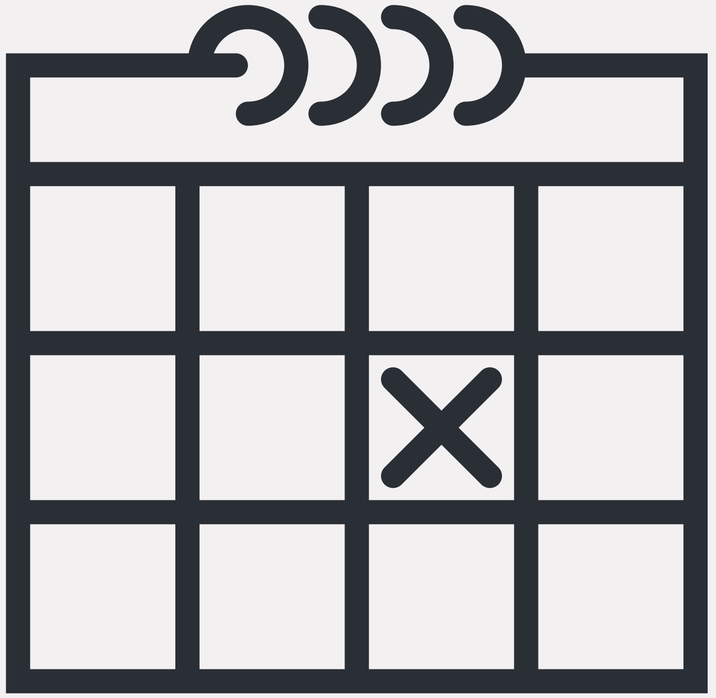
MINIMUM

VIABLE

PRODUCT

Wouldn't it be cool...

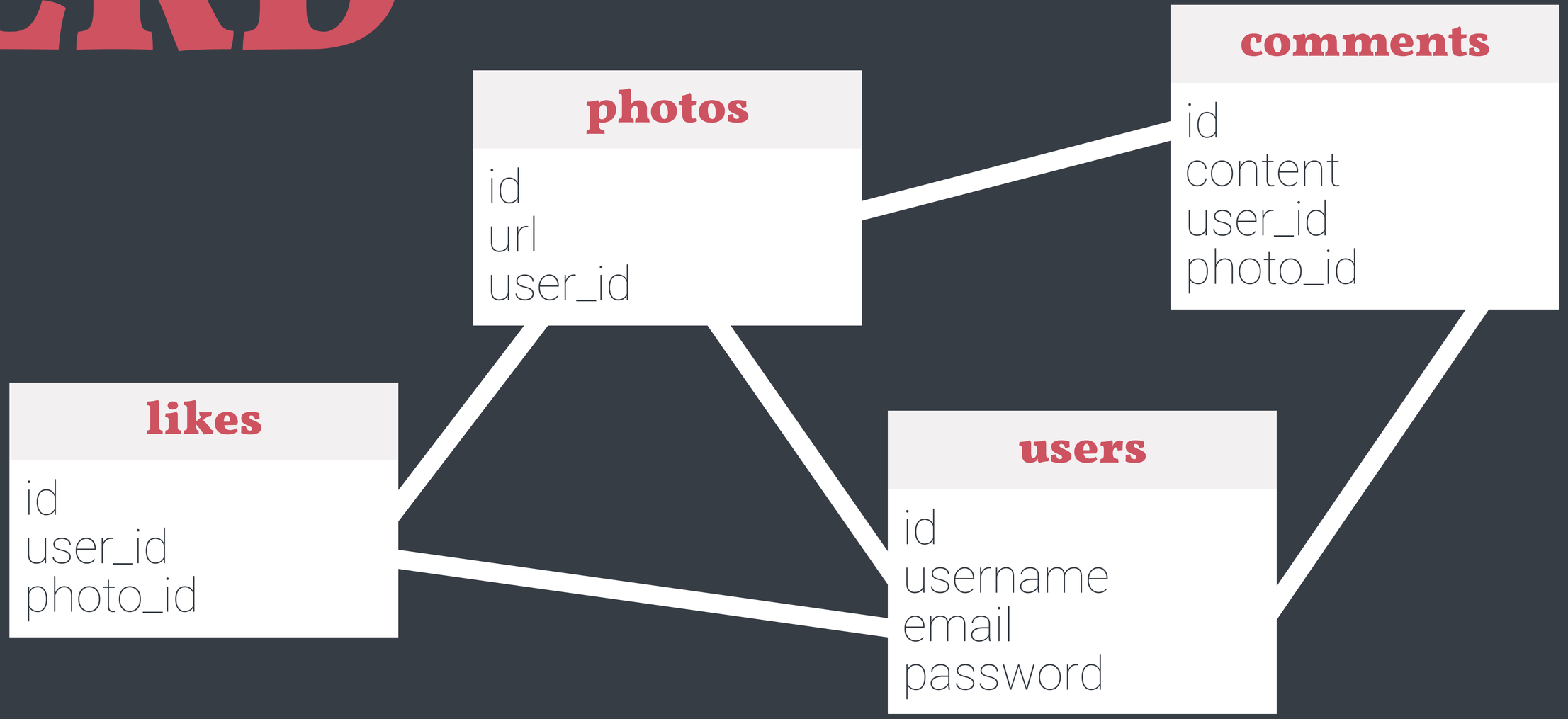
if we had something to demo?



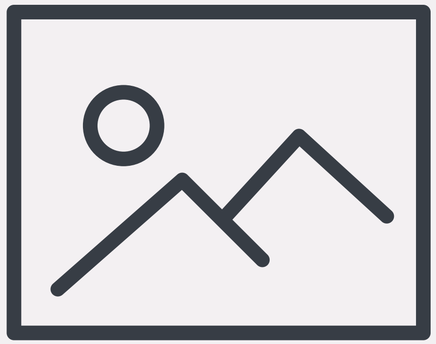
You don't need forms to login.

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

ERD


photos
id
url
user_id
comments
id
content
user_id
photo_id
likes
id
user_id
photo_id
users
id
username
email
password

Resource



A **photo**.

/Routes

Browse/List/Index **GET /photos**

Create/Add **POST /photos**

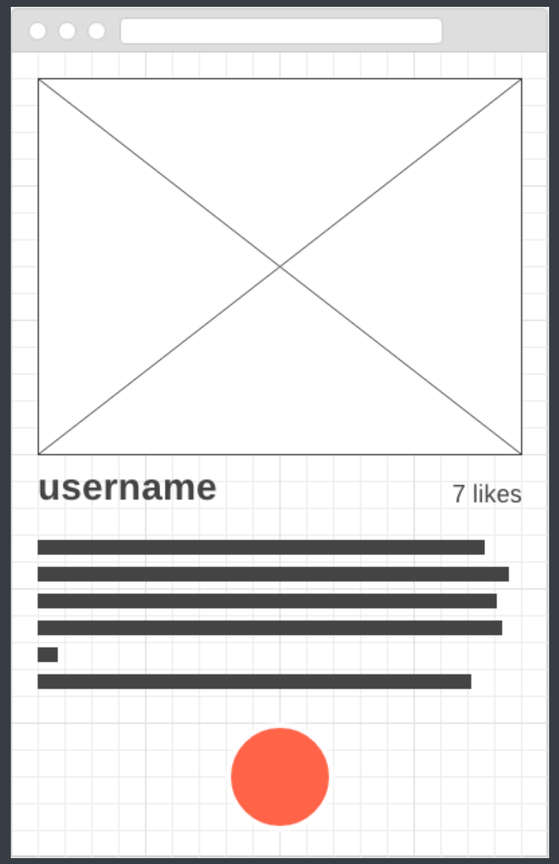
Read/Show **GET /photos/:id**

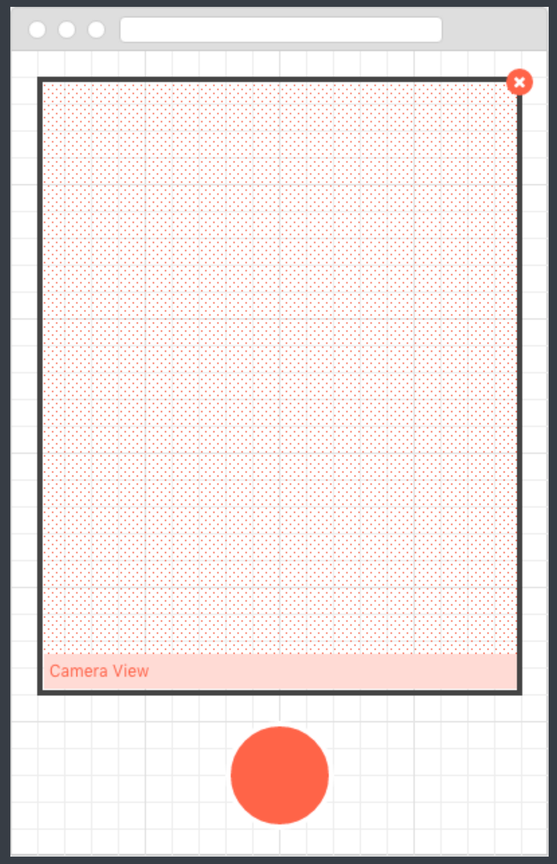
Update/Edit **PUT /photos/:id**

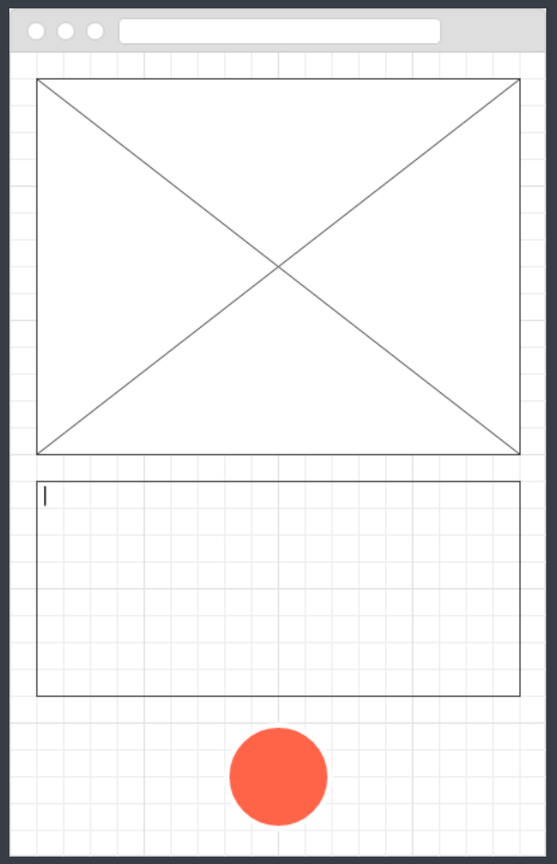
Remove/Destroy **DELETE /photos/:id**

Wireframes

As a user, I want to post a photo, because I like to share my experiences with friends.







Design
is important.

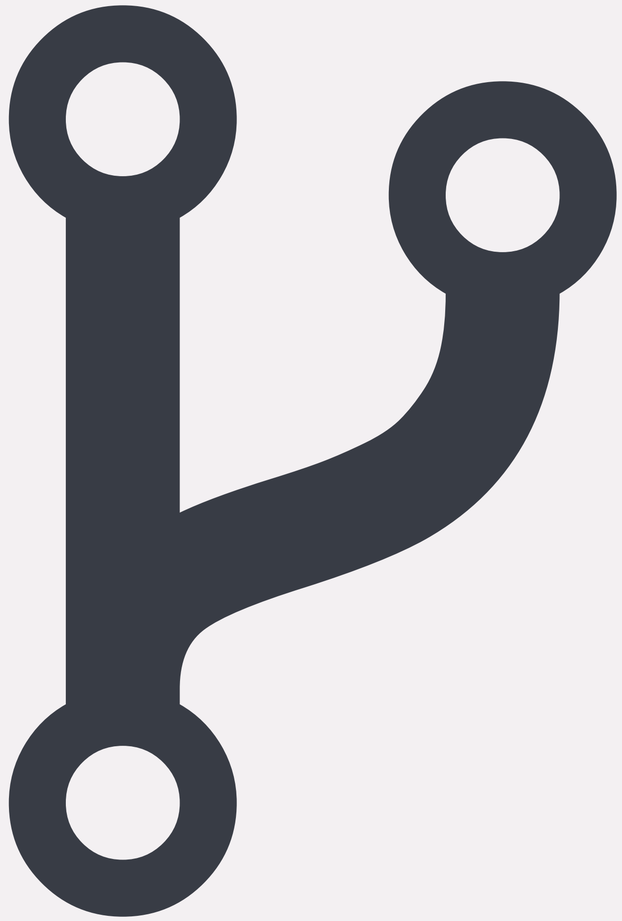
https://**fonts**.google.com/

Colors

CSS

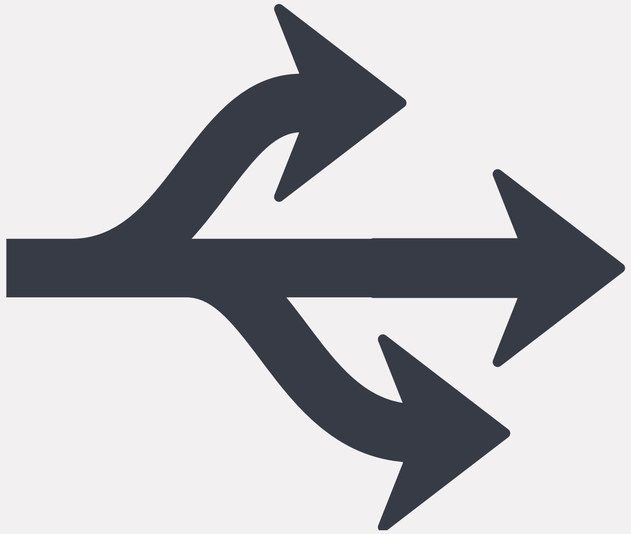
Skeleton
Bulma
Bourbon + Neat
Foundation
Bootstrap
Groundwork
Semantic UI

git

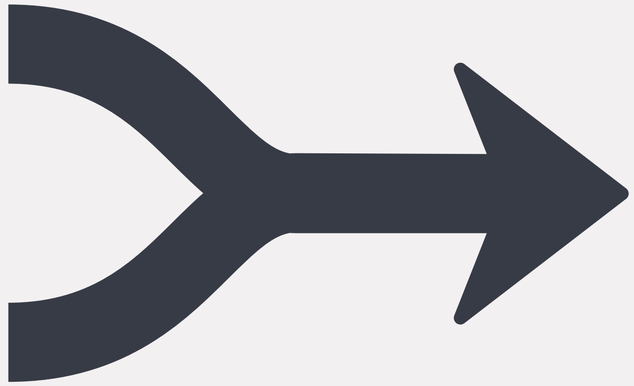


Clone then

1. Branch
2. Code & **Test**
3. Checkout master
4. Pull & **Test**
5. Merge & **Test**
6. Push
7. Branch & Repeat



Branch



Merge

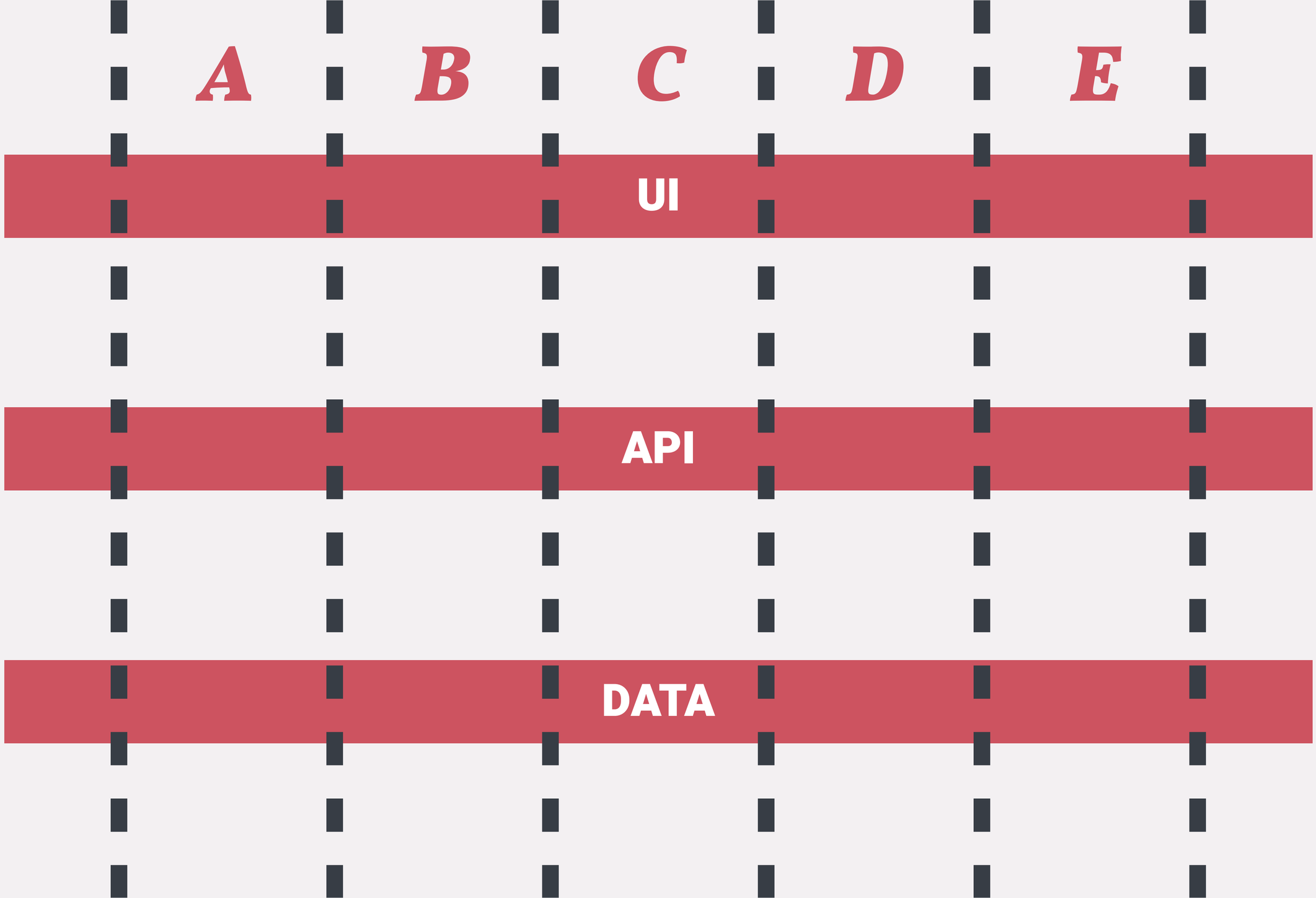
Project Setup

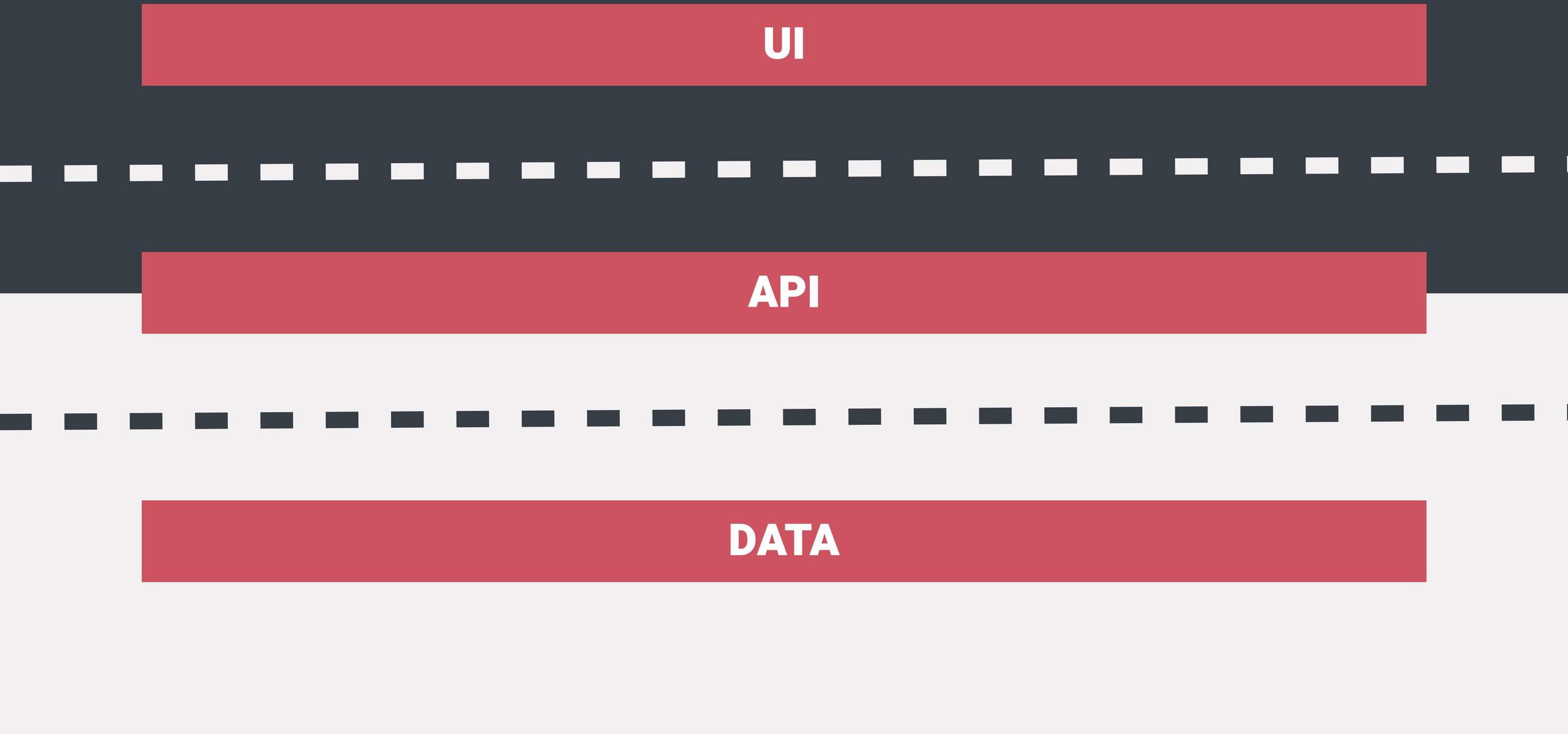
Together

Setup GitHub repository with collaborators
Clone github.com/lighthouse-labs/node-skeleton
Follow instructions in node-skeleton README
Make sure your app loads, check for console output
Commit and push changes to GitHub
Decide on team responsibilities

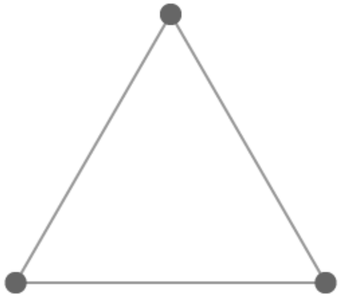
Dividing

Tasks

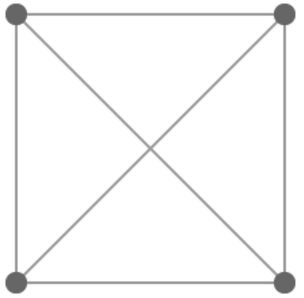

A
B
C
D
E
UI
API
DATA


UI
API
DATA

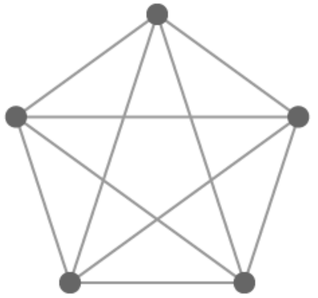
Communication
Communication
Communication
Communication
Communication
Communication
Communication
Communication
Communication
Communication
Communication
Communication
Communication
Communication



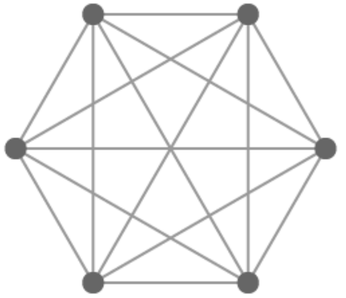
3 people, 3 lines



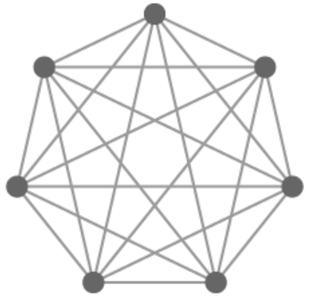
4 people, 6 lines



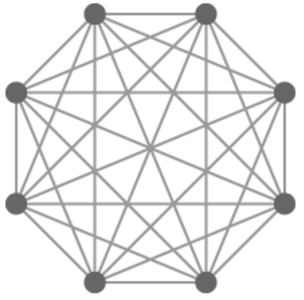
5 people, 10 lines



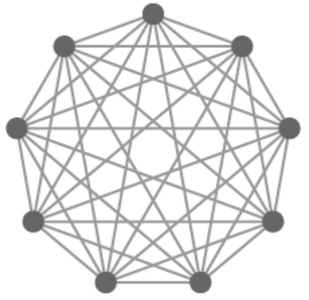
6 people, 15 lines



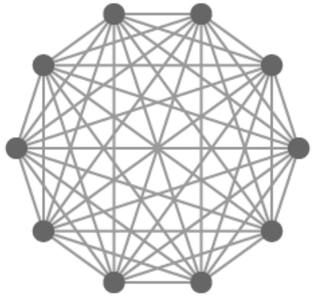
7 people, 21 lines



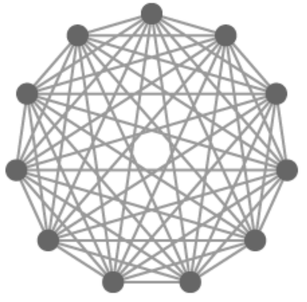
8 people, 28 lines



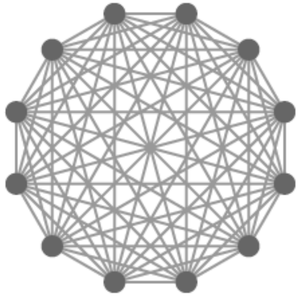
9 people, 36 lines



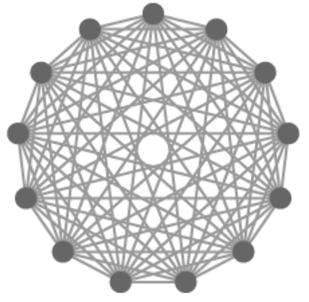
10 people, 45 lines



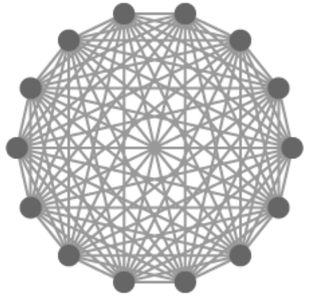
11 people, 55 lines



12 people, 66 lines



13 people, 78 lines



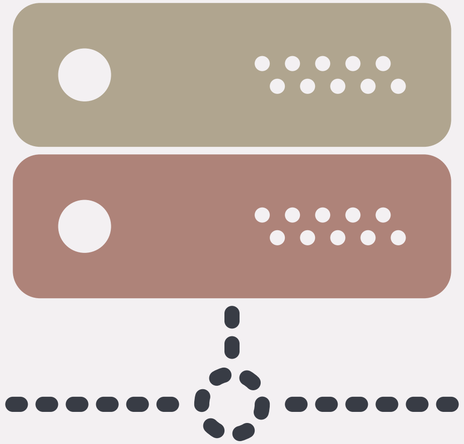
14 people, 91 lines

Use

Trello

Now we code.

On the Client

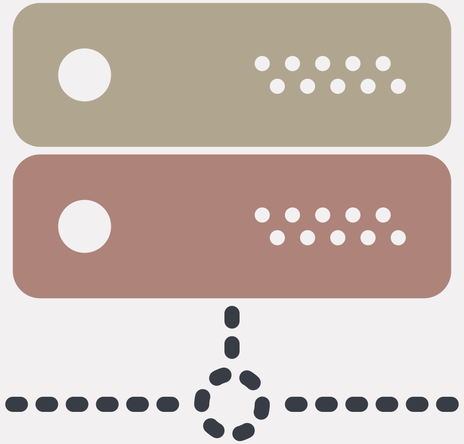


Start with a static page.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="description" content="<fill this in>">
    <meta name="author" content="<fill this in>">

    <style>
    </style>
  </head>
  <body>
  </body>
</html>
```

On the Server



```
SELECT * FROM photos;
SELECT * FROM photos WHERE user_id = $1;
SELECT * FROM photos WHERE id = $1;
SELECT * FROM users;
INSERT INTO photos (url, user_id) VALUES ($1, $2);
DELETE FROM photos WHERE id = $1;
```

Start with your queries.

Match them to routes.

```
GET /photos
GET /users/:id/photos
GET /photos/:id
GET /users
POST /photos
DELETE /photos/:id
```

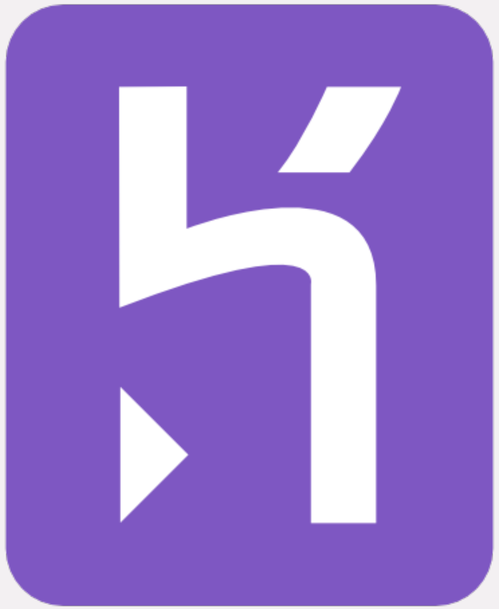
```
app.get('/users/:id/photos', (req, res) => {
  knex(‘photos’)
  .where('user_id', req.params.id)
  .then(photos => {
    res.json(photos);
  });
});
```

Provide the data through the interface.

You could pass the data to a template.

```
app.get('/users/:id/photos', (req, res) => {
  knex(‘photos’)
  .where('user_id', req.params.id)
  .then(photos => {
    res.render(‘photos_index’, { photos });
  });
});
```

You don't have to deploy.



http://127.0.0.1:8080/
to demo.

Questions?