




Glitch Tour

Glitch Tour  



Building Blocks

PLAY VIDEO

A look at at the components
of a glitch project. Also
types of project will we
build?

Prerequisite tools on your Workstation

none!

(apart from a browser + a github account)

First screen is the “source” for a running, live web project

The screenshot shows a web browser window with the URL `https://glitch.com/edit/#!/common-enquiry?path=README.md:1:0`. The page title is "README.md - common-enquir". The browser's address bar and various extension icons are visible at the top. Below the browser, the Glitch editor interface is shown. On the left, there is a sidebar with a "Share" button and a "New File" dropdown. A file tree lists several files: `assets`, `public/client.js`, `public/style.css`, `views/index.html`, `.env`, `README.md` (which is selected and highlighted in blue), `package.json`, and `server.js`. The main area of the editor is titled "Markdown" and contains the following text:

Welcome to Glitch

Click `Show` in the header to see your app live. Updates to your code will instantly deploy and update live.

Glitch is the friendly community where you'll build the app of your dreams. Glitch lets you instantly create, remix, edit, and host an app, bot or site, and you can invite collaborators or helpers to simultaneously edit code with you.

Find out more [about Glitch](#).

Your Project

On the front-end,

- edit `public/client.js`, `public/style.css` and `views/index.html`
- drag in `assets`, like images or music, to add them to your project

On the back-end,

- your app starts at `server.js`
- add frameworks and packages in `package.json`
- safely store app secrets in `.env` (nobody can see this but you and people you invite)

Made by [Glitch](#)

\ ° o°)/

At the bottom left of the editor, there are navigation arrows and a "Tools" button.

Project name
(automatically
generated)

Link to running
app (to share)

Files in the
project

Current File
(editable)

The screenshot shows a web browser window with the URL `https://glitch.com/edit/#!/common-enquiry?path=README.md:1:0`. The browser's address bar shows the project name `common-enquiry` and the file name `README.md`. The interface includes a header with a `Show` button, a `Share` button, and a search bar. A left sidebar lists project files: `assets`, `public/client.js`, `public/style.css`, `views/index.html`, `.env`, `README.md` (highlighted), `package.json`, and `server.js`. The main content area displays a `Markdown` editor with the following text:

Welcome to Glitch

Click `Show` in the header to see your app live. Updates to your code will instantly deploy and update live.

Glitch is the friendly community where you'll build the app of your dreams. Glitch lets you instantly create, remix, edit, and host an app, bot or site, and you can invite collaborators or helpers to simultaneously edit code with you.

Find out more [about Glitch](#).

Your Project

On the front-end,

- edit `public/client.js`, `public/style.css` and `views/index.html`
- drag in `assets`, like images or music, to add them to your project

On the back-end,

- your app starts at `server.js`
- add frameworks and packages in `package.json`
- safely store app secrets in `.env` (nobody can see this but you and people you invite)

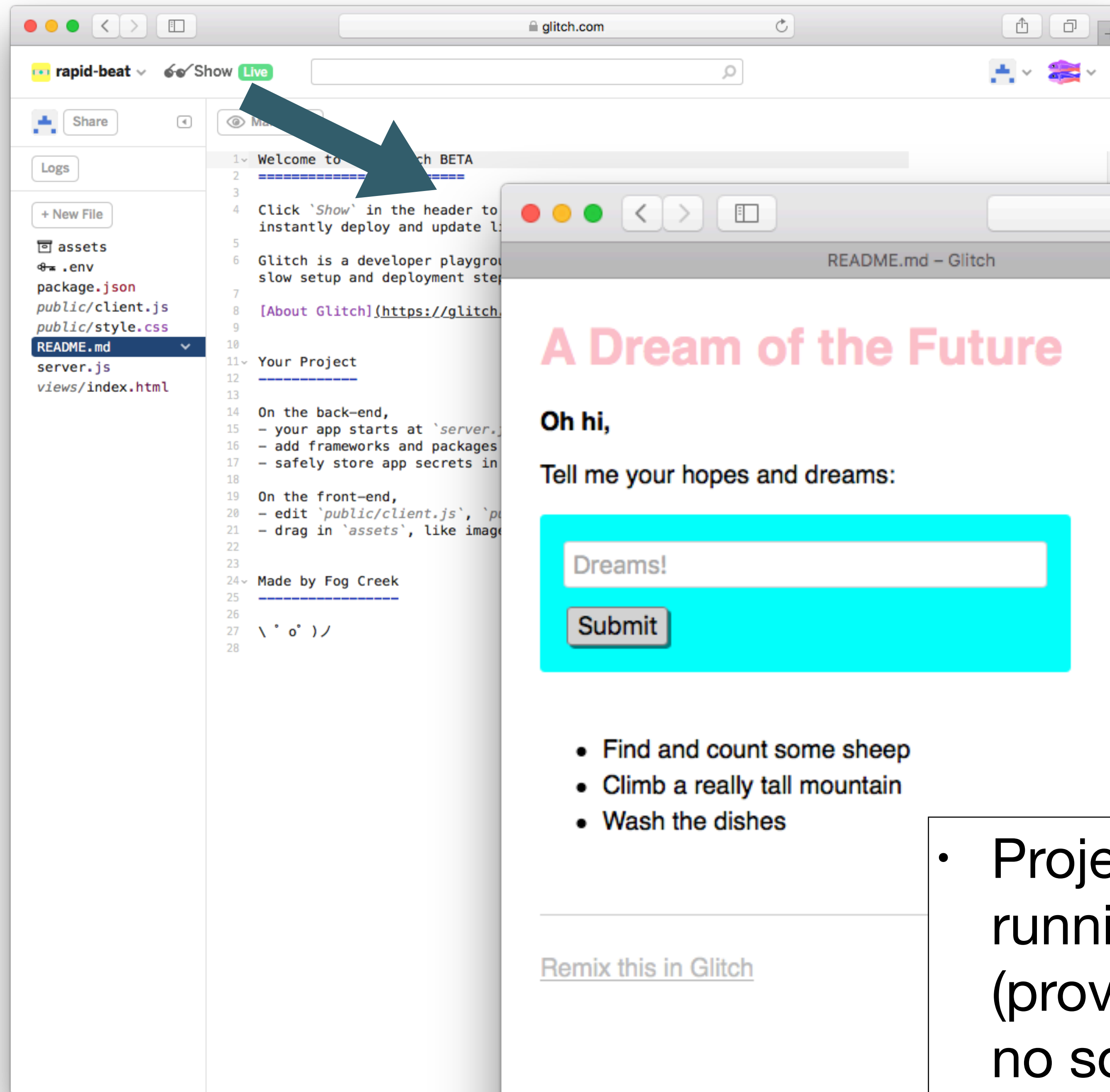
Made by [Glitch](#)

\\ ° °) /

At the bottom left, there is a `Tools` button. On the right side of the interface, there are two profile icons: a user profile and a community icon.

Link to your
Profile

Link to Community,
resources, options



A Dream of the Future

Oh hi,

Tell me your hopes and dreams:

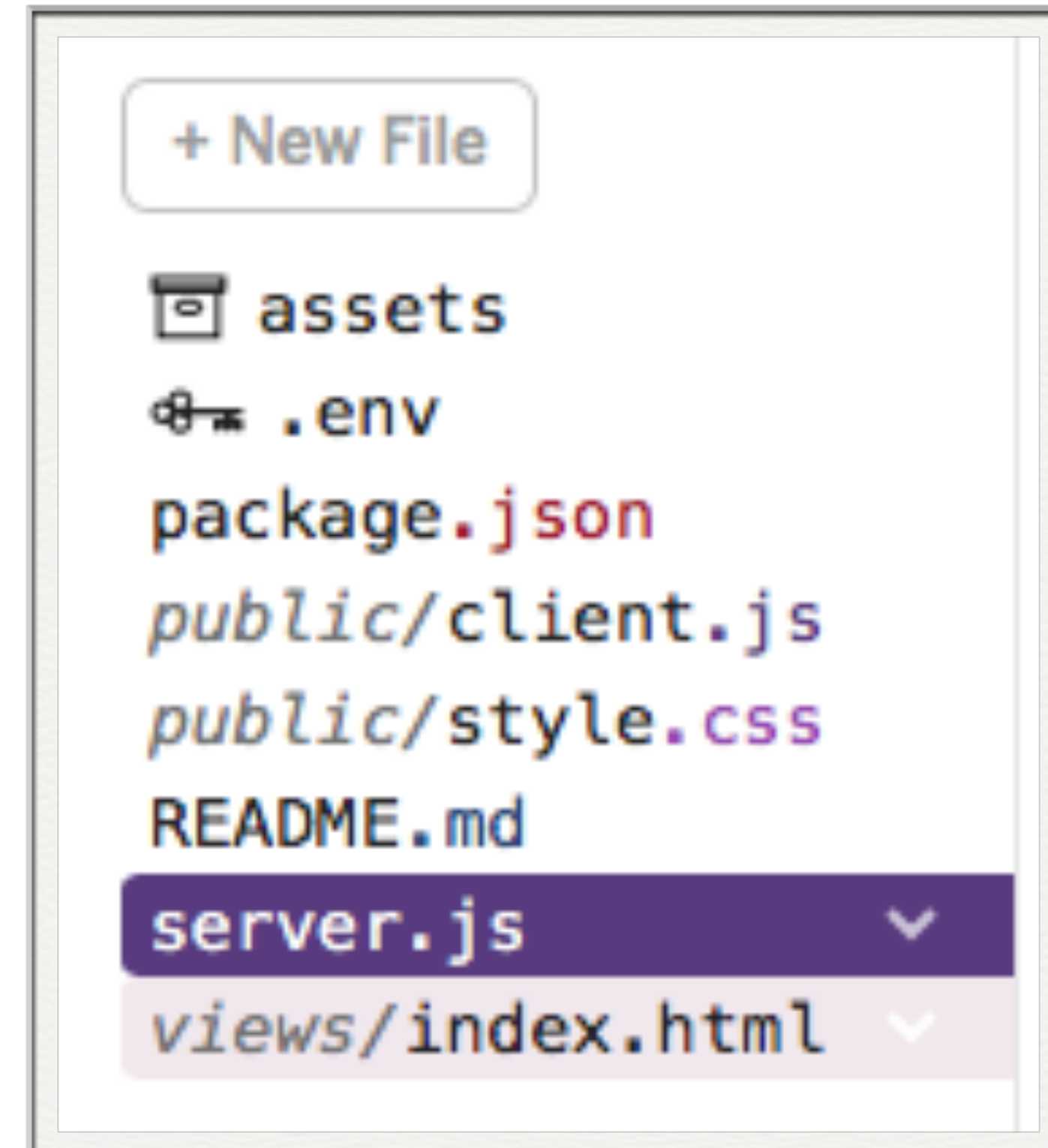
- Find and count some sheep
- Climb a really tall mountain
- Wash the dishes

[Remix this in Glitch](#)

- Project is always running live (provided there are no source errors)

Project Structure

- Glitch projects not just web sites!
- They are fully featured web apps - with full server-side resources

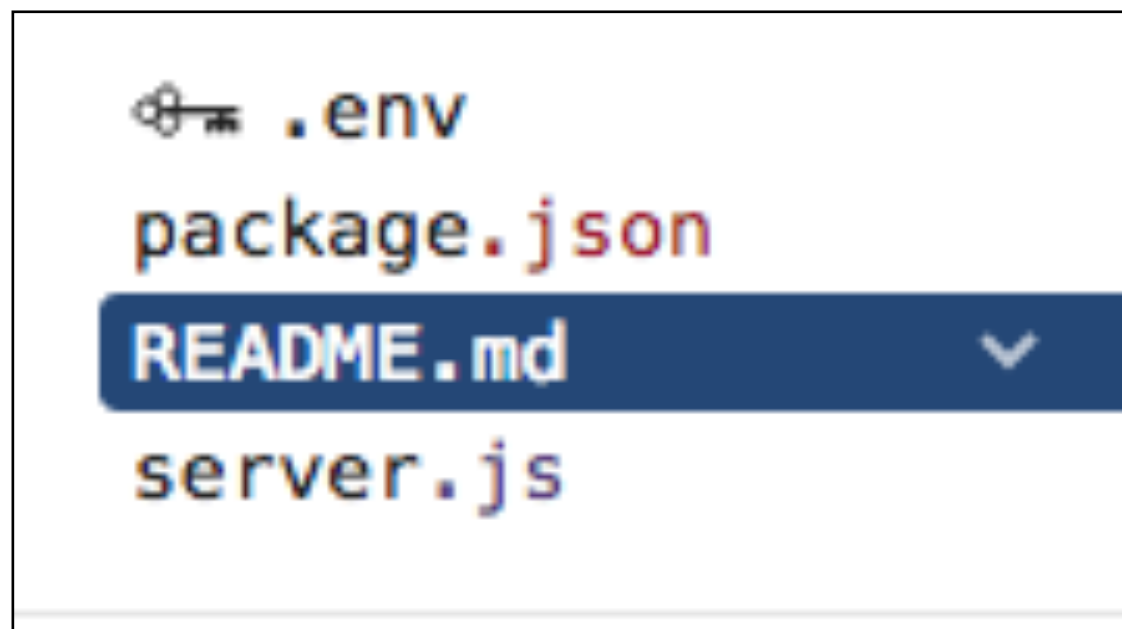


Front End

```
assets  
public/client.js  
public/style.css  
views/index.html
```

- Comparable to a static web site:
- html files + stylesheets + images
- Templating also possible.
- Also, access to the server side is implicit.
- This means you can build apps that have behaviour + state (much more on this later)

Back end



- An application - written in javascript - and hosted in the cloud.
- Application built in Javascript using a technology called node.js

The Starter App

The screenshot displays a web browser window with a code editor. The browser's address bar shows the URL `https://glitch.com/edit/#!/common-enquiry?path=views/index.html:1:0`. The editor interface includes a file explorer on the left with the following files: `assets`, `public/client.js`, `public/style.css`, `views/index.html` (selected), `.env`, `README.md`, `package.json`, and `server.js`. The main editor area shows the HTML code for `views/index.html`, which includes a header with a favicon, meta tags for charset, UA compatibility, and viewport, a link to a stylesheet, and a script for client-side JavaScript. The body contains a header with the text "A Dream of the Future", a main section with a bolded "Oh hi," and a form for submitting dreams. The footer includes a link to Glitch and a Glitch button. The code is as follows:

```
15 <link id="favicon" rel="icon" href="https://glitch.com/edit/favicon-app.ico" type="image/x-icon">
16 <meta charset="utf-8">
17 <meta http-equiv="X-UA-Compatible" content="IE=edge">
18 <meta name="viewport" content="width=device-width, initial-scale=1">
19
20 <!-- import the webpage's stylesheet -->
21 <link rel="stylesheet" href="/style.css">
22
23 <!-- import the webpage's client-side javascript file -->
24 <script src="/client.js" defer></script>
25 </head>
26 <body>
27 <header>
28 <h1>
29   A Dream of the Future
30 </h1>
31 </header>
32
33 <main>
34 <p class="bold">Oh hi,</p>
35
36 <p>Tell me your hopes and dreams:</p>
37
38 <form>
39   <input name="dream" type="text" maxlength="100" placeholder="Dreams!" aria-labelledby="submit-dream">
40   <button type="submit" id="submit-dream">Submit Dream</button>
41 </form>
42
43 <section class="dreams">
44   <ul id="dreams"></ul>
45 </section>
46 </main>
47
48 <footer>
49   Made with <a href="https://glitch.com">Glitch</a>!
50 </footer>
51
52 <!-- include the Glitch button to show what the webpage is about and
53      to make it easier for folks to view source and remix -->
54 <div class="glitchButton" style="position:fixed;top:20px;right:20px;"></div>
55 <script src="https://button.glitch.me/button.js"></script>
56
57 </body>
58 </html>
```

The Starter App

The image shows a web browser window with a Glitch editor. The browser tab is titled "index.html - common-enquiry". The address bar shows the URL "https://glitch.com/edit/#!/common-enquiry?path=views/index.html:1:0". The editor interface includes a file explorer on the left with "assets" and "public/client.js", a "Share" button, and a "New File" button. The main editor area shows the source code for "views/index.html".

The rendered application in the foreground has the following content:

A Dream of the Future

Oh hi,
Tell me your hopes and dreams:

- Find and count some sheep
- Climb a really tall mountain
- Wash the dishes

[Remix this in Gornix](#)

```
<link id="favIcon" rel="icon" href="https://glitch.com/edit/favIcon-app.ico" type="image/x-icon">
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- import the webpage's stylesheet -->
<link rel="stylesheet" href="/style.css">
<!-- import the webpage's client-side javascript file -->
<script src="/script.js" defer></script>
<h1>A Dream of the Future</h1>
<p>Oh hi,
Tell me your hopes and dreams:</p>
<input type="text" value="Dreams!" />
<input type="button" value="Submit" />
<ul>
<li>• Find and count some sheep</li>
<li>• Climb a really tall mountain</li>
<li>• Wash the dishes</li>
</ul>
<p><a href="https://glitch.com">Glitch</a>!</p>
<!-- A button to show what the webpage is about and
for folks to view source and remix -->
<div id="button" style="position:fixed;top:20px;right:20px;"></div>
<script src="https://glitch.com/button.js"></script>
```

A Dream of the Future

Oh hi,

Tell me your hopes and dreams:

- Find and count some sheep
- Climb a really tall mountain
- Wash the dishes

[Remix this in Gomix](#)

```
<body>
  <header>
    <h1>
      A Dream of the Future
    </h1>
  </header>

  <main>
    <p class="bold">Oh hi,</p>
    <p>Tell me your hopes and dreams:</p>
    <form>
      <input type="text" maxlength="100" placeholder="Dreams!">
      <button type="submit">Submit</button>
    </form>
    <section class="dreams">
      <ul id="dreams">
      </ul>
    </section>
  </main>

  <footer>
    <a href="https://gomix.com">
      Remix this in Gomix
    </a>
  </footer>
```

html

client side javascript

```
<body>
  <header>
    <h1>
      A Dream of the Future
    </h1>
  </header>

  <main>
    <p class="bold">Oh hi,</p>
    <p>Tell me your hopes and dreams:</p>
    <form>
      <input type="text" maxlength="100" placeholder="
      <button type="submit">Submit</button>
    </form>
    <section class="dreams">
      <ul id="dreams">
      </ul>
    </section>
  </main>

  <footer>
    <a href="https://gomix.com">
      Remix this in Gomix
    </a>
  </footer>
```

```
// client-side js
// run by the browser each time your view template is loaded

// by default, you've got jQuery,
// add other scripts at the bottom of index.html

$(function() {
  console.log('hello world :o');

  $.get('/dreams', function(dreams) {
    dreams.forEach(function(dream) {
      $('<li></li>').text(dream).appendTo('ul#dreams');
    });
  });

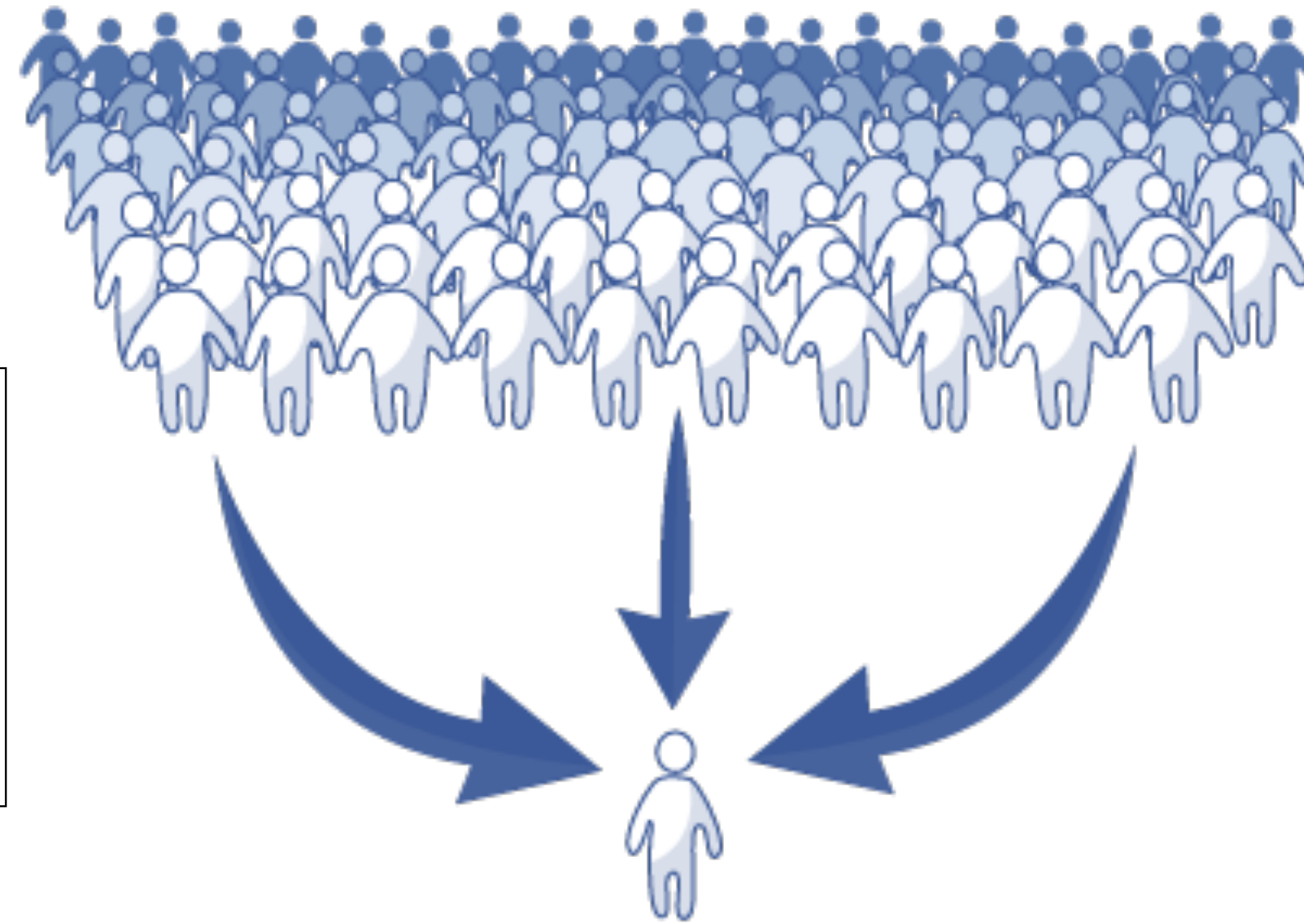
  $('form').submit(function(event) {
    event.preventDefault();
    dream = $('input').val();
    $.post('/dreams?' + $.param({dream: dream}), function() {
      $('<li></li>').text(dream).appendTo('ul#dreams');
      $('input').val('');
      $('input').focus();
    });
  });
});
```

server side javascript

```
1 // server.js
2 // where your node app starts
3
4 // init project
5 const express = require('express');
6 const app = express();
7
8 // we've started you off with Express,
9 // but feel free to use whatever libs or frameworks you'd like through `package.json`.
10
11 // http://expressjs.com/en/starter/static-files.html
12 app.use(express.static('public'));
13
14 // http://expressjs.com/en/starter/basic-routing.html
15 app.get('/', function(request, response) {
16   response.sendFile(__dirname + '/views/index.html');
17 });
18
19 // listen for requests :)
20 const listener = app.listen(process.env.PORT, function() {
21   console.log('Your app is listening on port ' + listener.address().port);
22 });
23
```

Client side javascript runs
in each users browser

```
$('#form').submit(function(event) {  
  event.preventDefault();  
  dream = $('#input').val();  
  $.post('/dreams?' + $.param({dream: dream}), function() {  
    $('<li></li>').text(dream).appendTo('ul#dreams');  
    $('#input').val('');  
    $('#input').focus();  
  });  
});
```



```
// could also use the POST body instead of query string: http://expressjs.com/en/api.html#req.body  
app.post("/dreams", function (request, response) {  
  dreams.push(request.query.dream);  
  response.sendStatus(200);  
});
```

A node runs the server side
javascript. All browsers
connected to this node

Skills for this Course

- Assumptions:
 - Foundation Knowledge in HTML + CSS
 - Working knowledge of Semantic UI CSS Framework
- Major focus of this course:
 - Javascript Programming
 - Node.js Web Application Development
- Glitch is the platform
- Front end javascript development will **not** be covered.

```
// server.js
// where your node app starts

// init project
var express = require('express');
var app = express();

// we've started you off with Express,
// but feel free to use whatever libs or frameworks you'd like through `package.json`.

// http://expressjs.com/en/starter/static-files.html
app.use(express.static('public'));

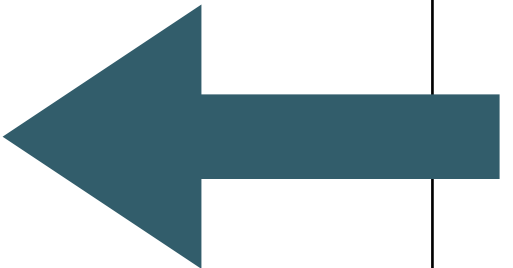
// http://expressjs.com/en/starter/basic-routing.html
app.get("/", function (request, response) {
  response.sendFile(__dirname + '/views/index.html');
});

app.get("/dreams", function (request, response) {
  response.send(dreams);
});

// could also use the POST body instead of query string: http://expressjs.com/en/api.html#req.body
app.post("/dreams", function (request, response) {
  dreams.push(request.query.dream);
  response.sendStatus(200);
});

// Simple in-memory store for now
var dreams = [
  "Find and count some sheep",
  "Climb a really tall mountain",
  "Wash the dishes"
];

// listen for requests :)
var listener = app.listen(process.env.PORT, function () {
  console.log('Your app is listening on port ' + listener.address().port);
});
```

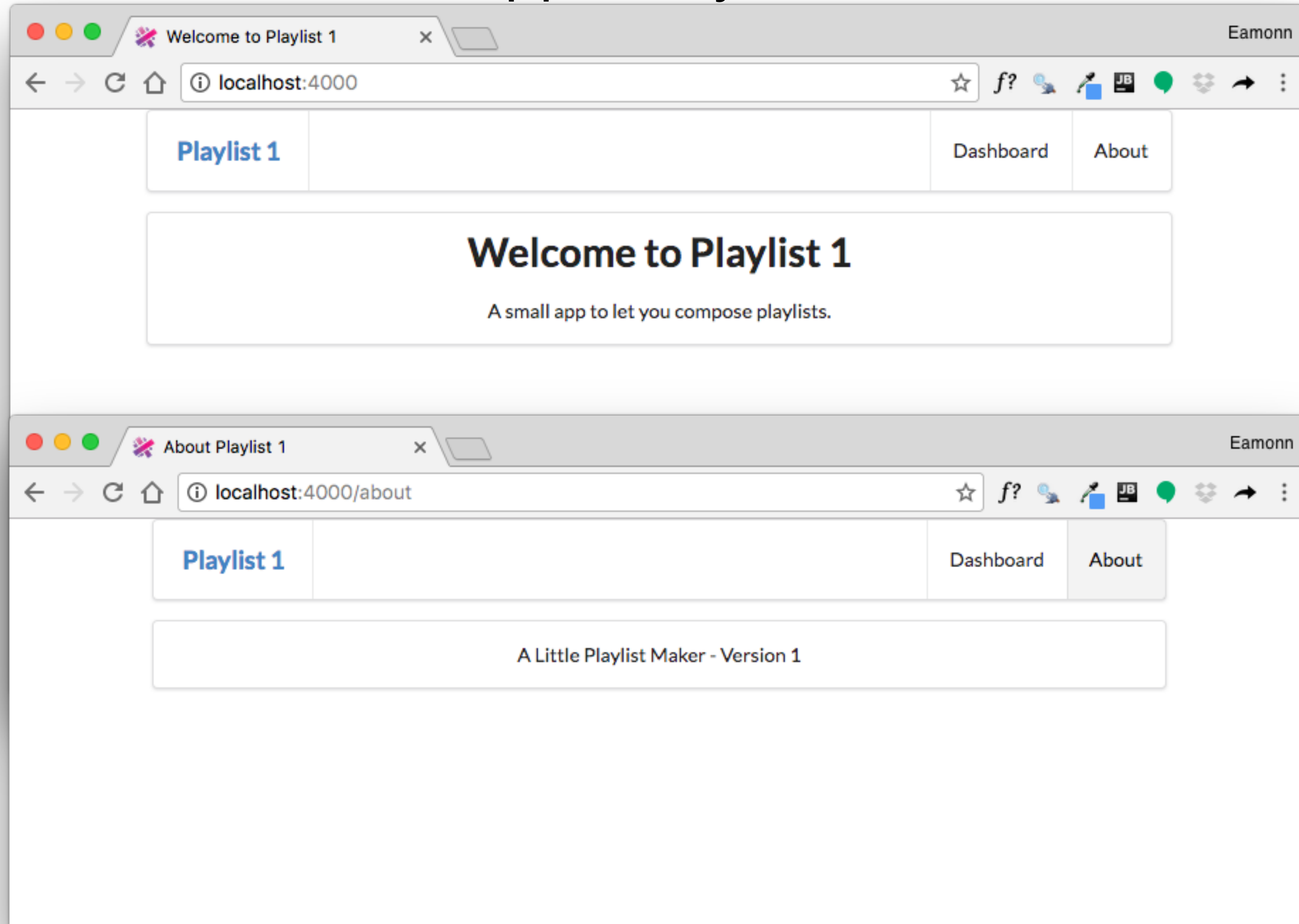


We will learn what all of this means.

- + how to build a fully featured web app including:
 - templating
 - forms to submit information
 - How store information in models
 - create user accounts, and tie account to a each user

All of this requires beginner/intermediate level Javascript skills

A tour of our first app - Playlist





Playlist Dashboard x Eamonn

localhost:4000/dashboard

Playlist 1 Dashboard About



Beethoven Sonatas

Total Duration: 35



Beethoven Concertos

Total Duration: 23

Beethoven Variations

Total Duration: 67




Title

[Add Playlist](#)

Browser window: Playlist | localhost:4000/playlist/01 | Eamonn

Navigation: Playlist 1 | Dashboard | About

Beethoven Sonatas

Song	Artist	
Piano Sonata No. 3	Beethoven	
Piano Sonata No. 7	Beethoven	
Piano Sonata No. 10	Beethoven	

Title	Artist
<input type="text" value="Title"/>	<input type="text" value="Artist"/>

Playlist Labs

- We will do Four playlist labs in the next few sessions
 - Playlist 1: simple rendering of static playlist
 - Playlist 2: render multiple playlists, ability to delete playlists
 - Playlist 3: ability to create playlists. Store playlists long term.
 - Playlist 4: ability to support different users in the same application
- These labs will be interleaved with Javascript Introductory labs, which will gradually introduce you to the language