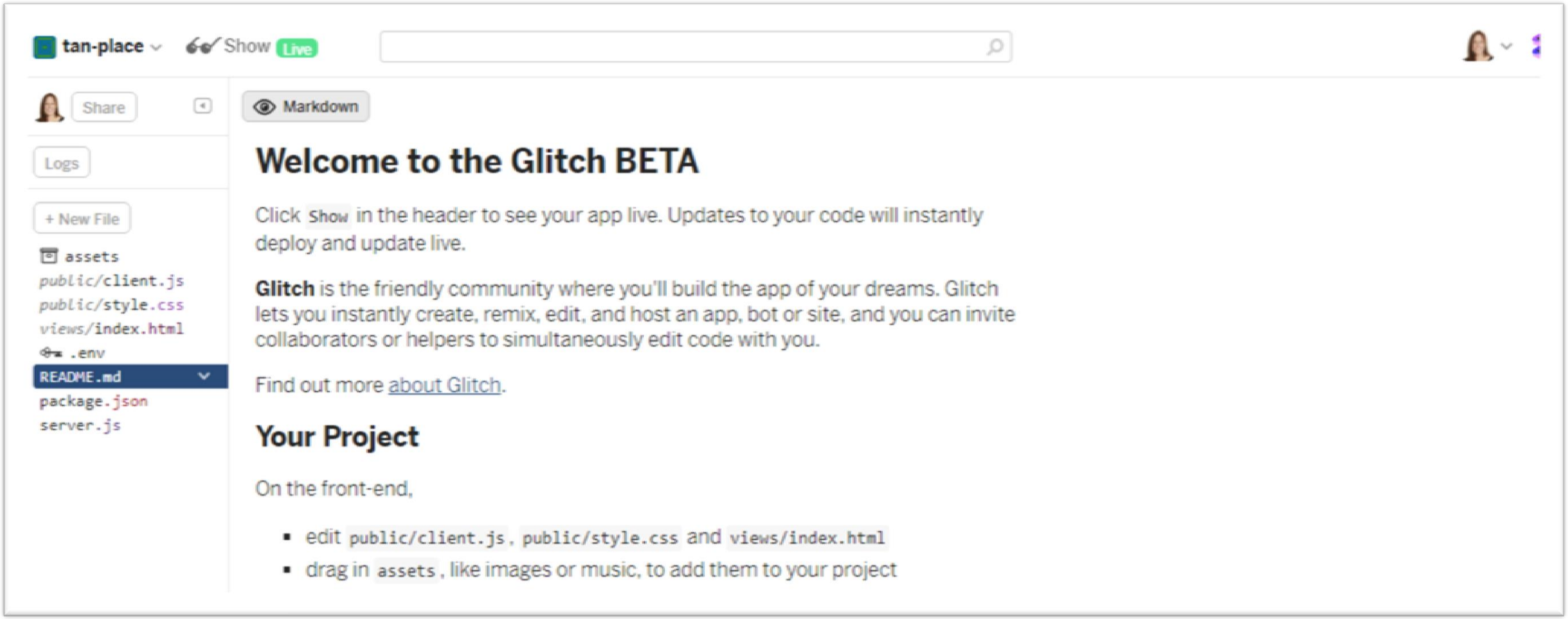
Glitch Tour

Prerequisite tools on your Workstation

- none!
- (apart from a browser + a github account)



Welcome to the Glitch BETA

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.

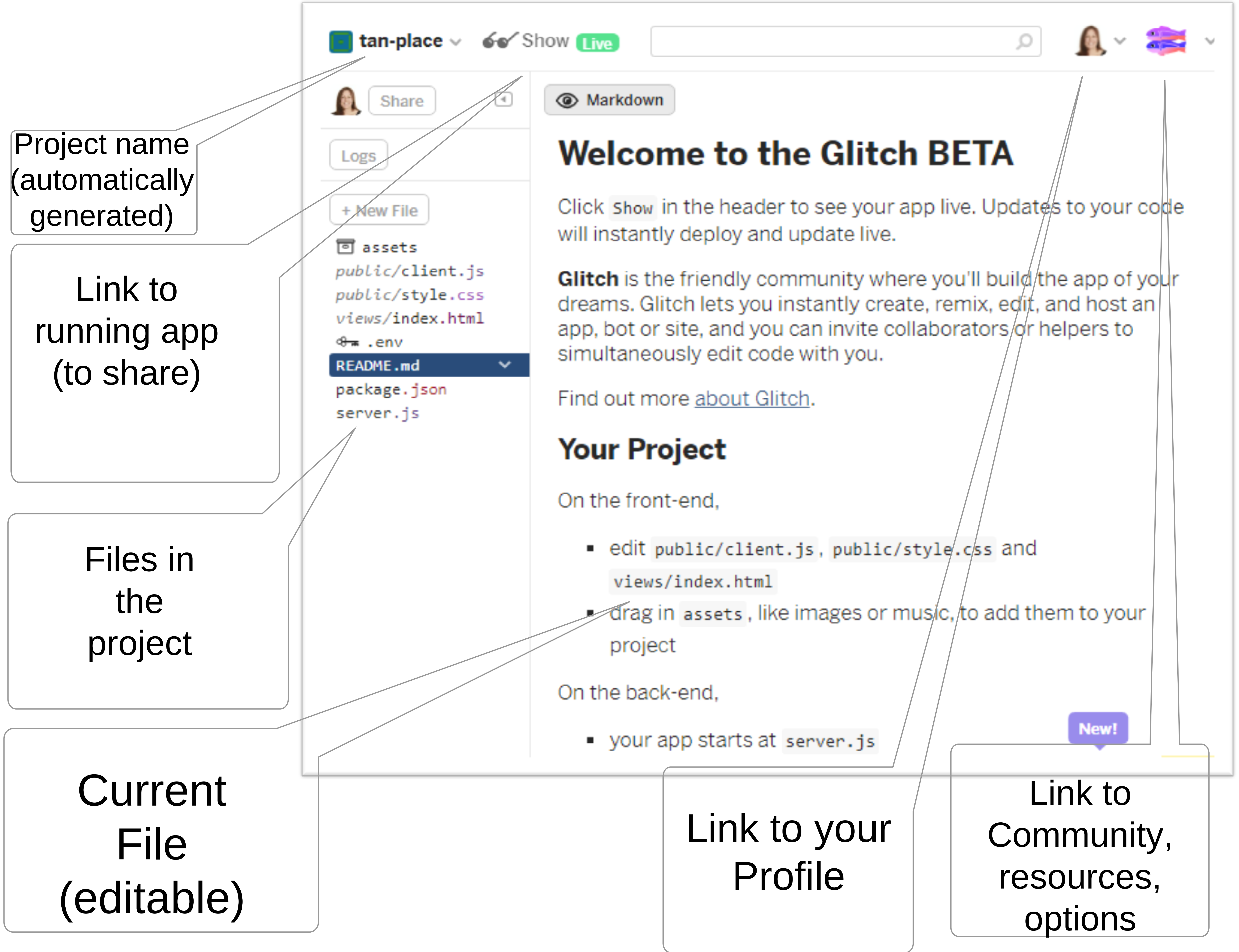
Find out more about Glitch.

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

- First screen is the "source" for a running, live web project



Welcome to the Glitch BETA

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.

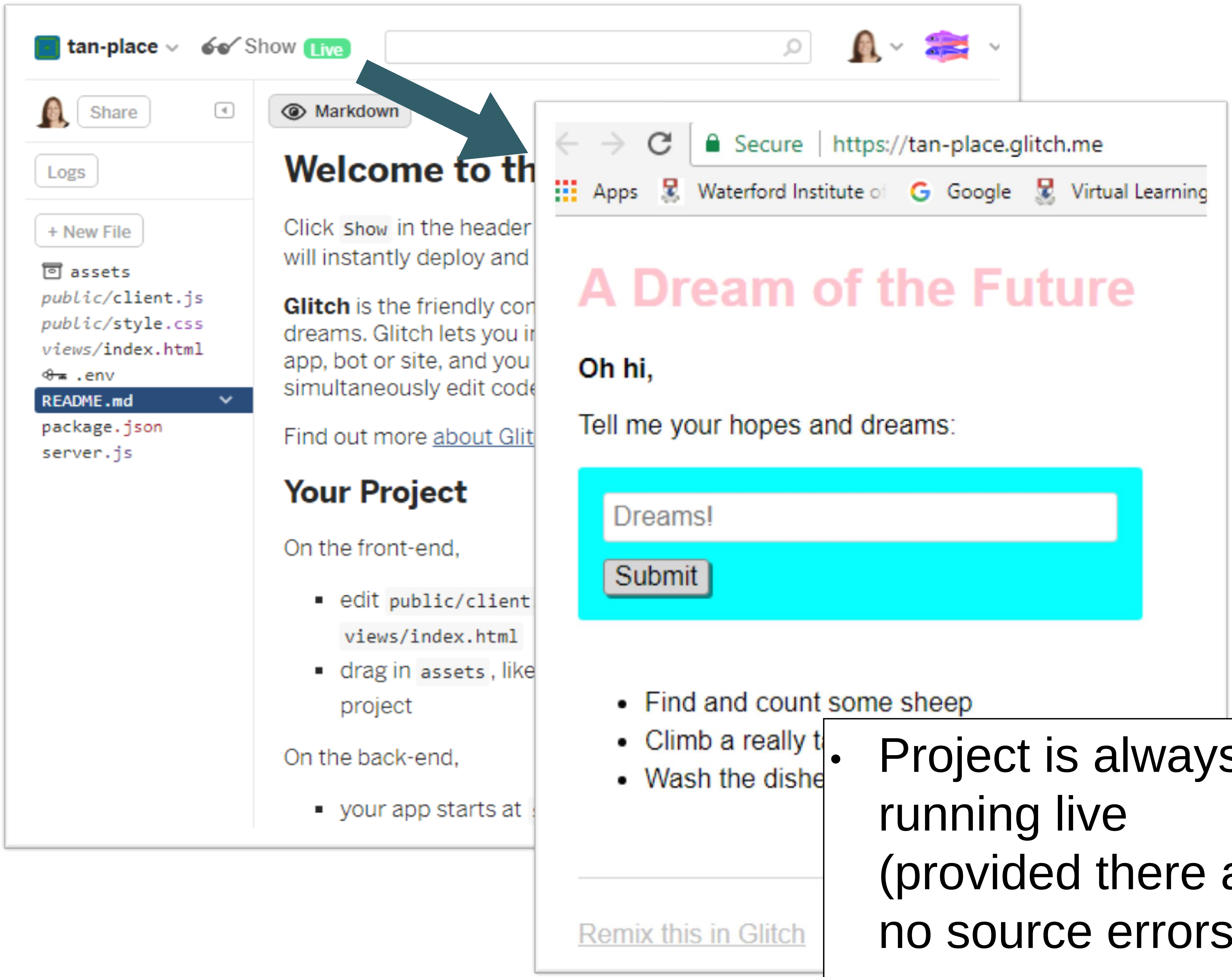
Your Project

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`



tan-place Show Live

Share Logs + New File

assets
public/client.js
public/style.css
views/index.html
.env
README.md
package.json
server.js

Markdown

Welcome to th

Click `show` in the header will instantly deploy and

Glitch is the friendly con dreams. Glitch lets you in app, bot or site, and you simultaneously edit code

Find out more about Glit

Your Project

On the front-end,

- edit `public/client` `views/index.html`
- drag in `assets`, like project

On the back-end,

- your app starts at

Secure | https://tan-place.glitch.me

Apps Waterford Institute o Google Virtual Learning

A Dream of the Future

Oh hi,

Tell me your hopes and dreams:

Dreams!

Submit

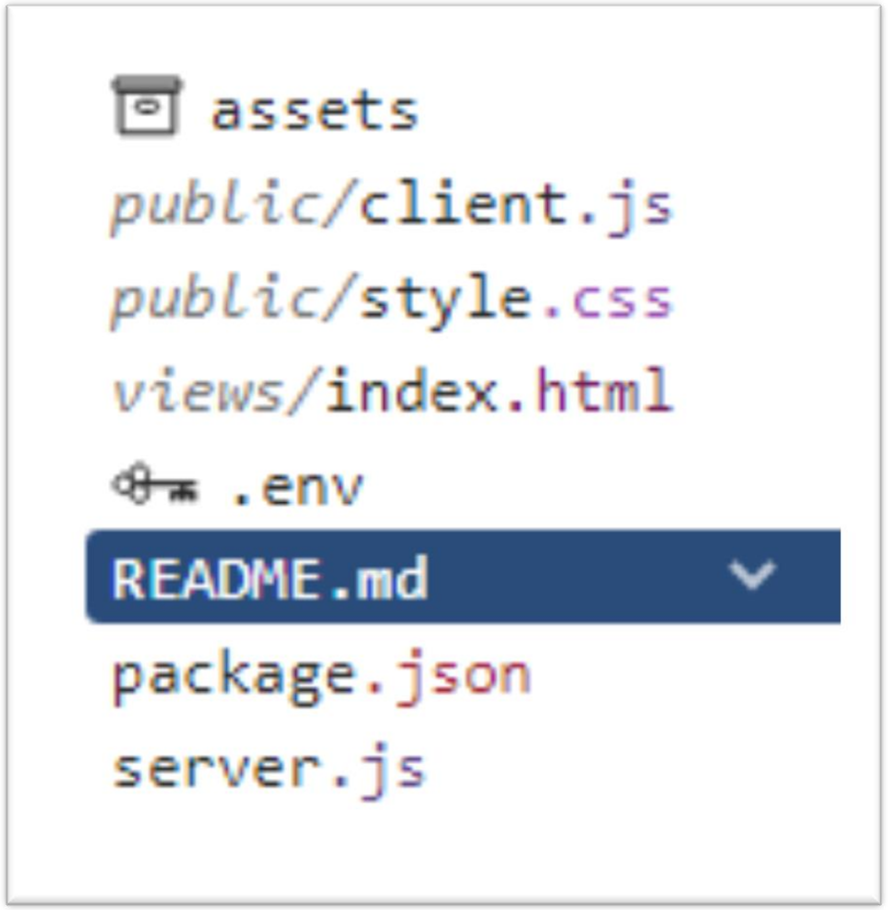
- Find and count some sheep
- Climb a really t
- Wash the dishe

Remix this in Glitch

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

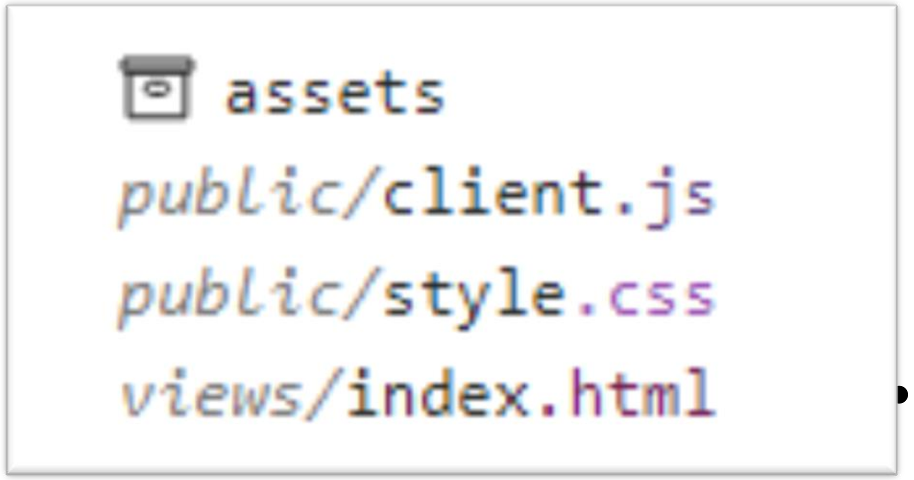
Project Structure

- Glitch projects not just web sites!
- They are web apps, divided into:
    - Front-end files
    - Back-end files



```
assets
public/client.js
public/style.css
views/index.html
.env
README.md
package.json
server.js
```

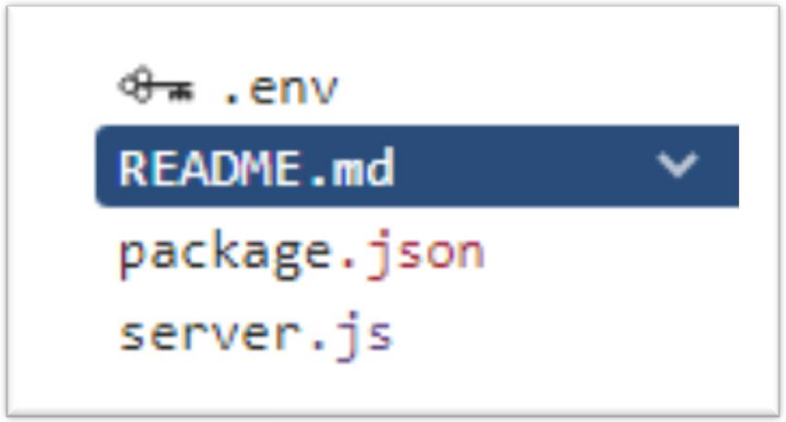
Front End



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

- Comparable to the web site you developed in previous module(s).
  - 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)

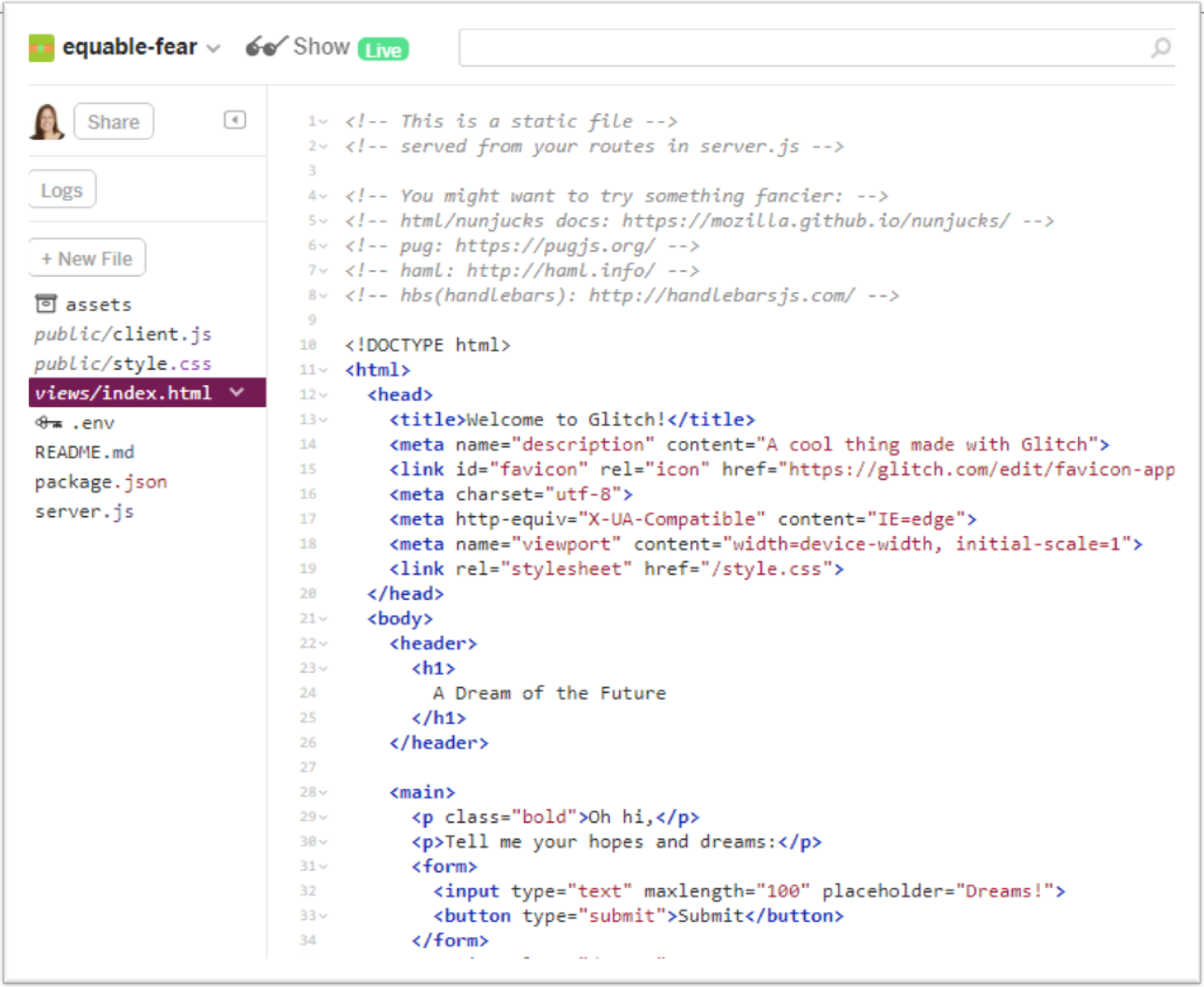
Back end



.env
README.md
package.json
server.js

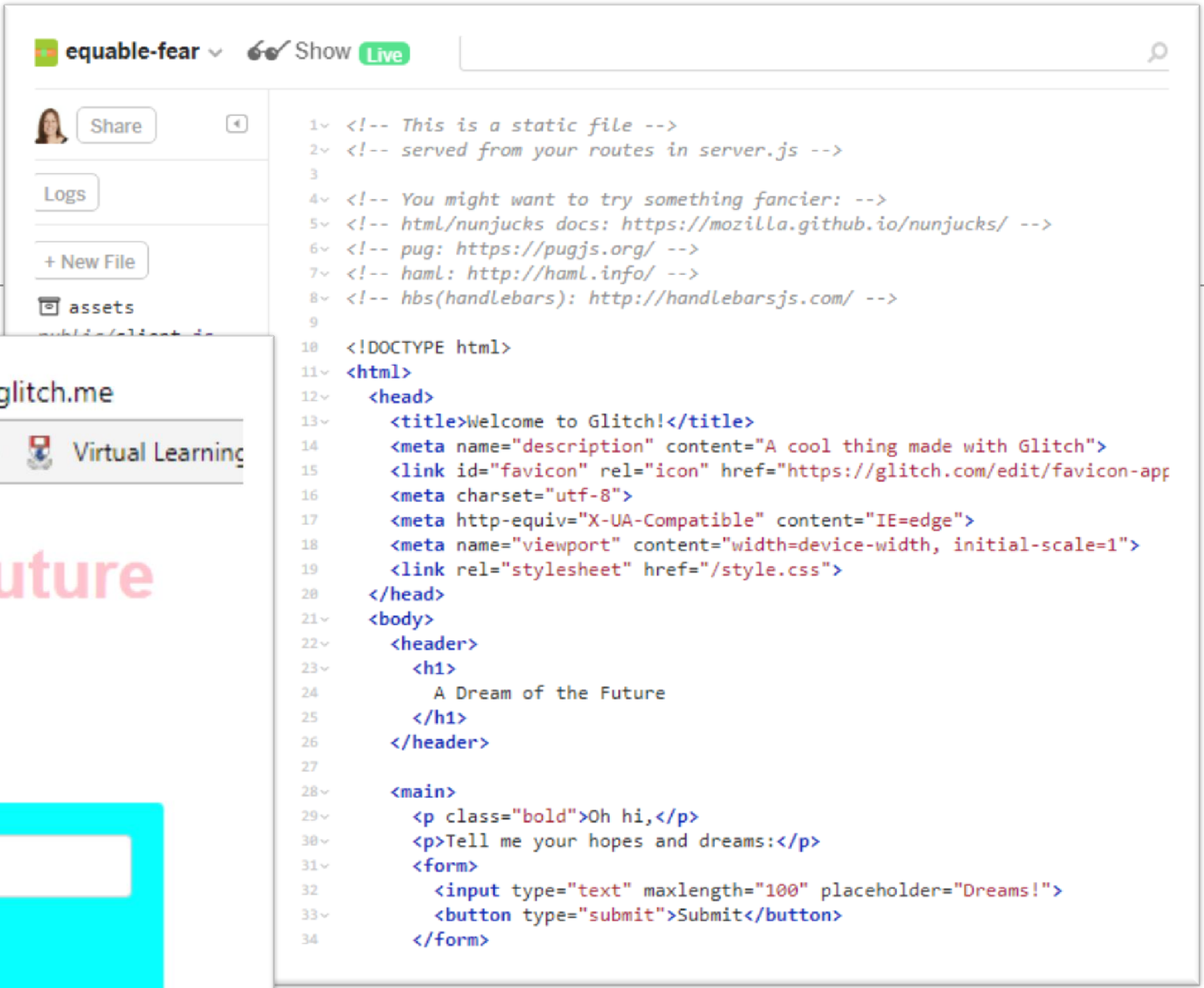
- An application - written in javascript - and hosted in the cloud.
- Many types of application supported.
- We will focus on Javascript applications written using node.js
- This is the default toolkit for Glitch.

The Starter App

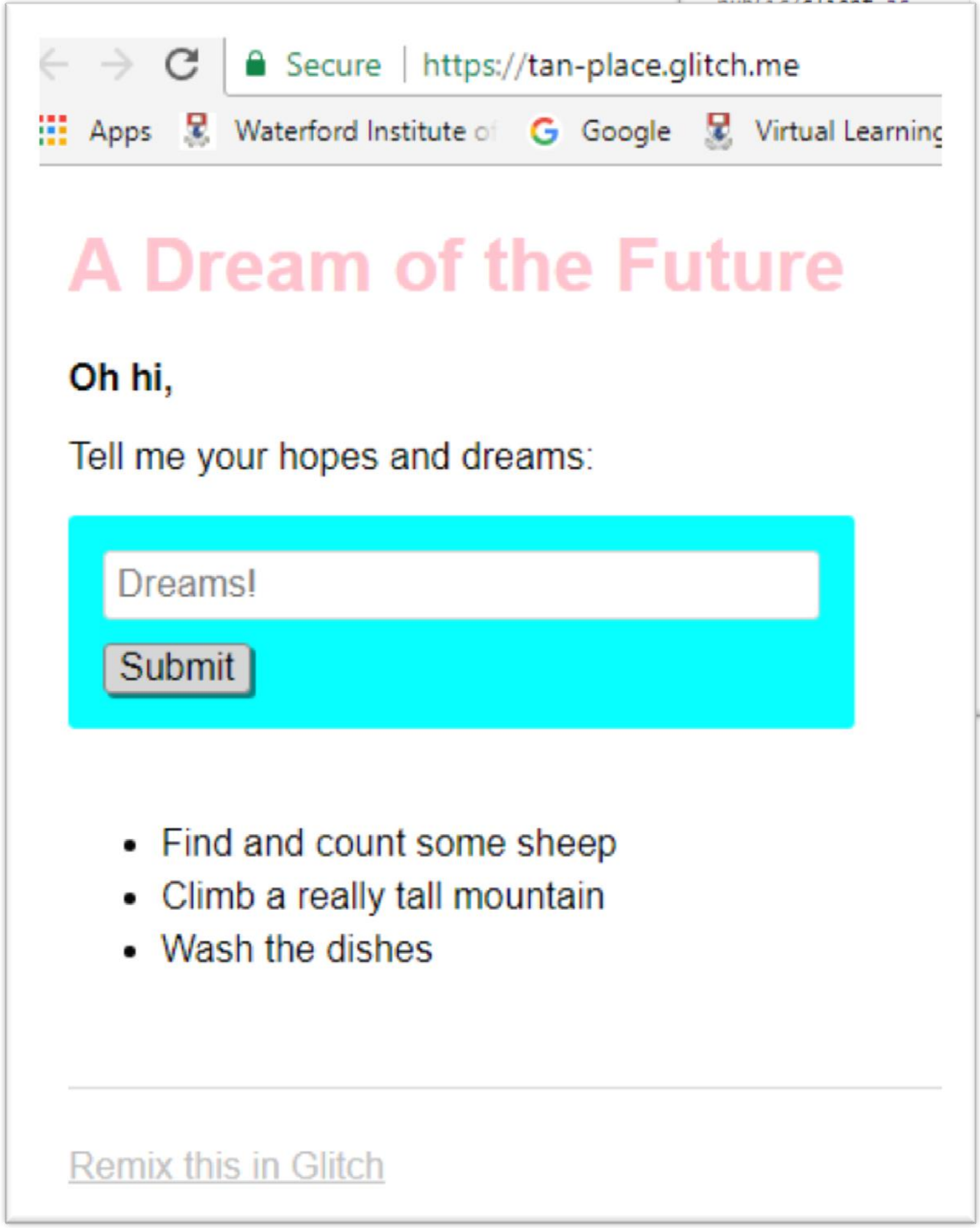


```html
<!-- This is a static file -->
<!-- served from your routes in server.js -->

<!-- You might want to try something fancier: -->
<!-- html/nunjucks docs: https://mozilla.github.io/nunjucks/ -->
<!-- pug: https://pugjs.org/ -->
<!-- haml: http://haml.info/ -->
<!-- hbs(handlebars): http://handlebarsjs.com/ -->

<!DOCTYPE html>
<html>
  <head>
    <title>Welcome to Glitch!</title>
    <meta name="description" content="A cool thing made with Glitch">
    <link id="favicon" rel="icon" href="https://glitch.com/edit/favicon-app
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="/style.css">
  </head>
  <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>
```

The Starter App



```html
<!-- This is a static file -->
<!-- served from your routes in server.js -->

<!-- You might want to try something fancier: -->
<!-- html/nunjucks docs: https://mozilla.github.io/nunjucks/ -->
<!-- pug: https://pugjs.org/ -->
<!-- haml: http://haml.info/ -->
<!-- hbs(handlebars): http://handlebarsjs.com/ -->

<!DOCTYPE html>
<html>
  <head>
    <title>Welcome to Glitch!</title>
    <meta name="description" content="A cool thing made with Glitch">
    <link id="favicon" rel="icon" href="https://glitch.com/edit/favicon-app
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="/style.css">
  </head>
  <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>
```



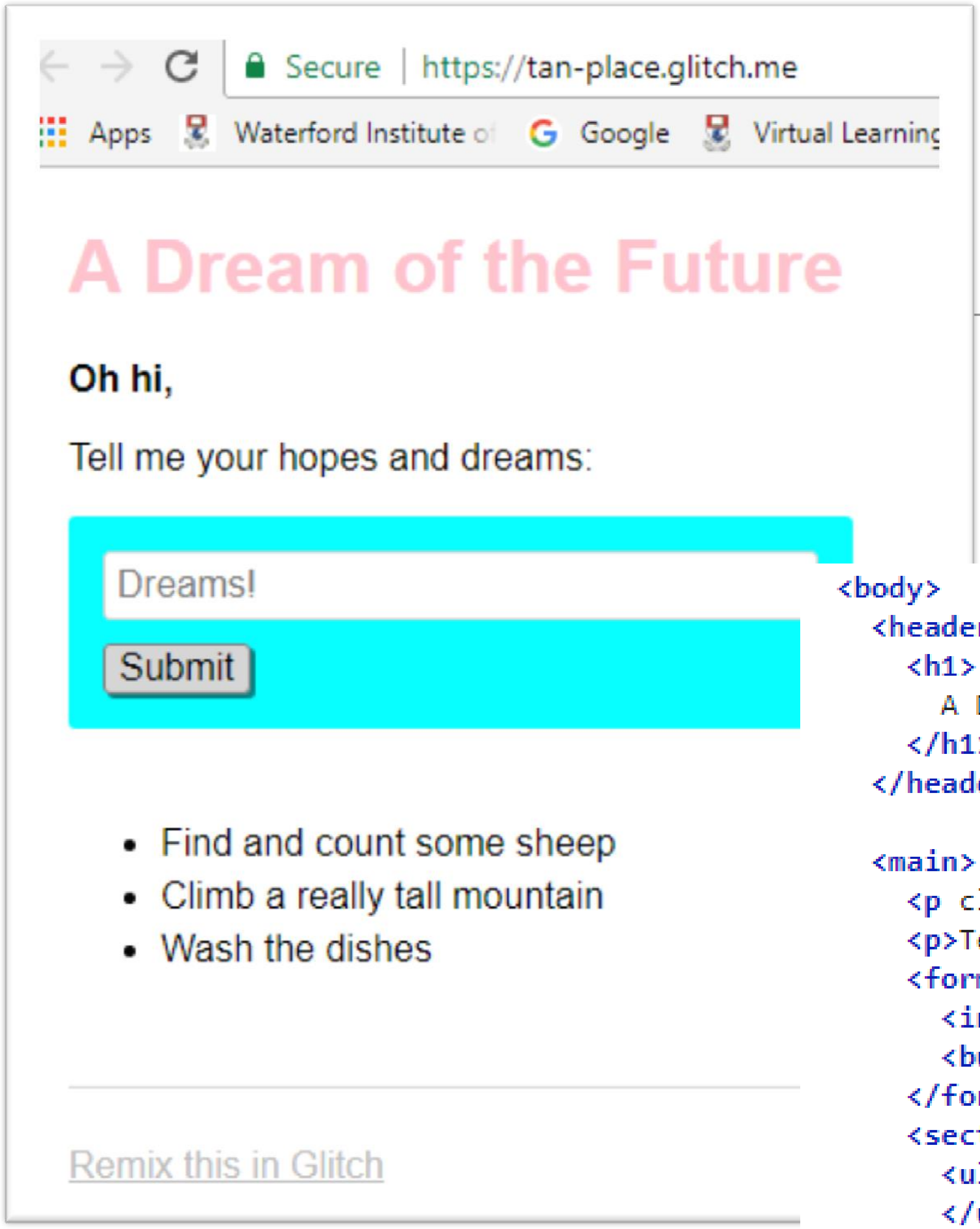
A Dream of the Future

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


Secure | https://tan-place.glitch.me

Apps Waterford Institute o Google Virtual Learning

A Dream of the Future

Oh hi,

Tell me your hopes and dreams:

Dreams!

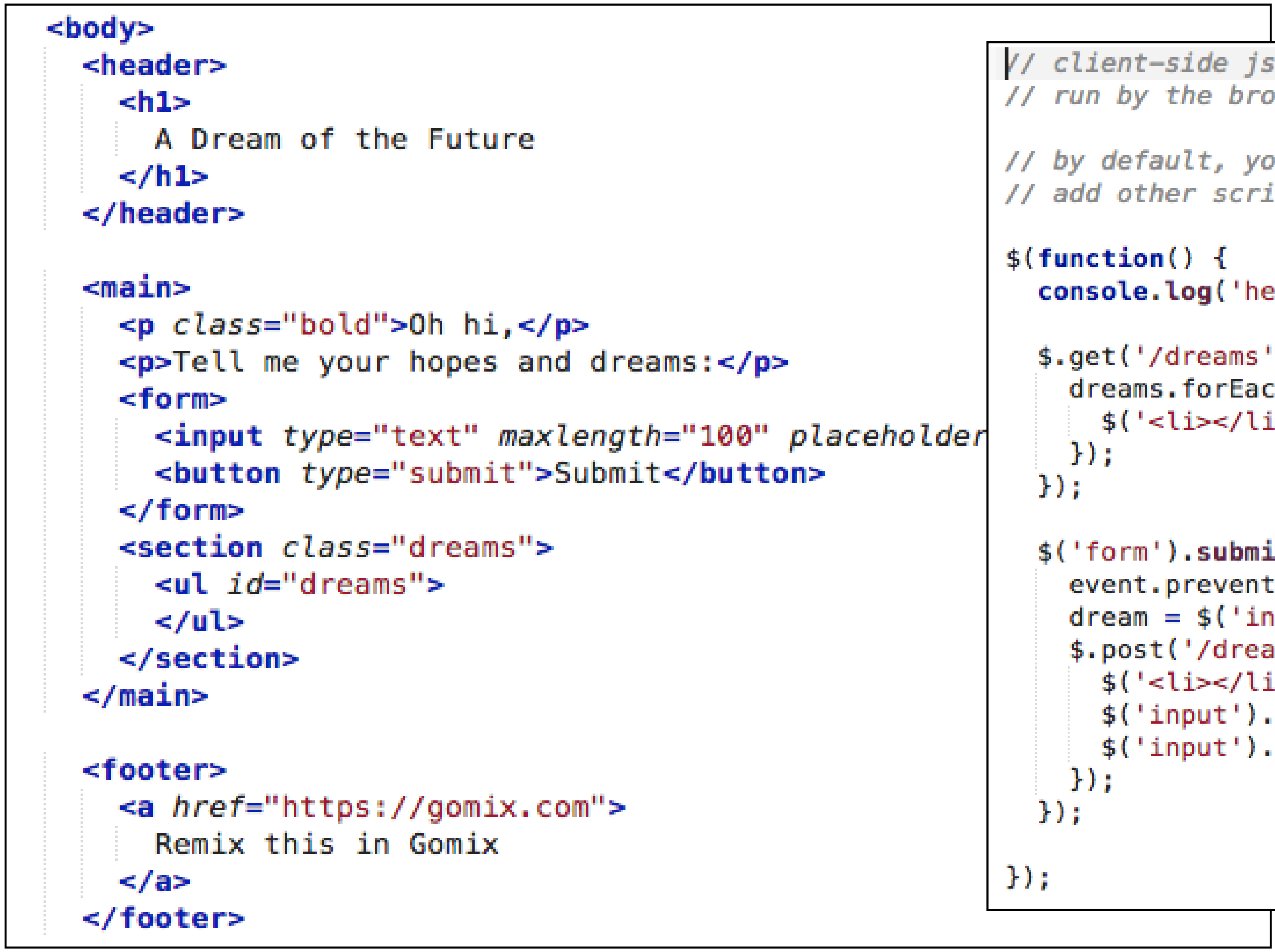
Submit

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

Remix this in Glitch

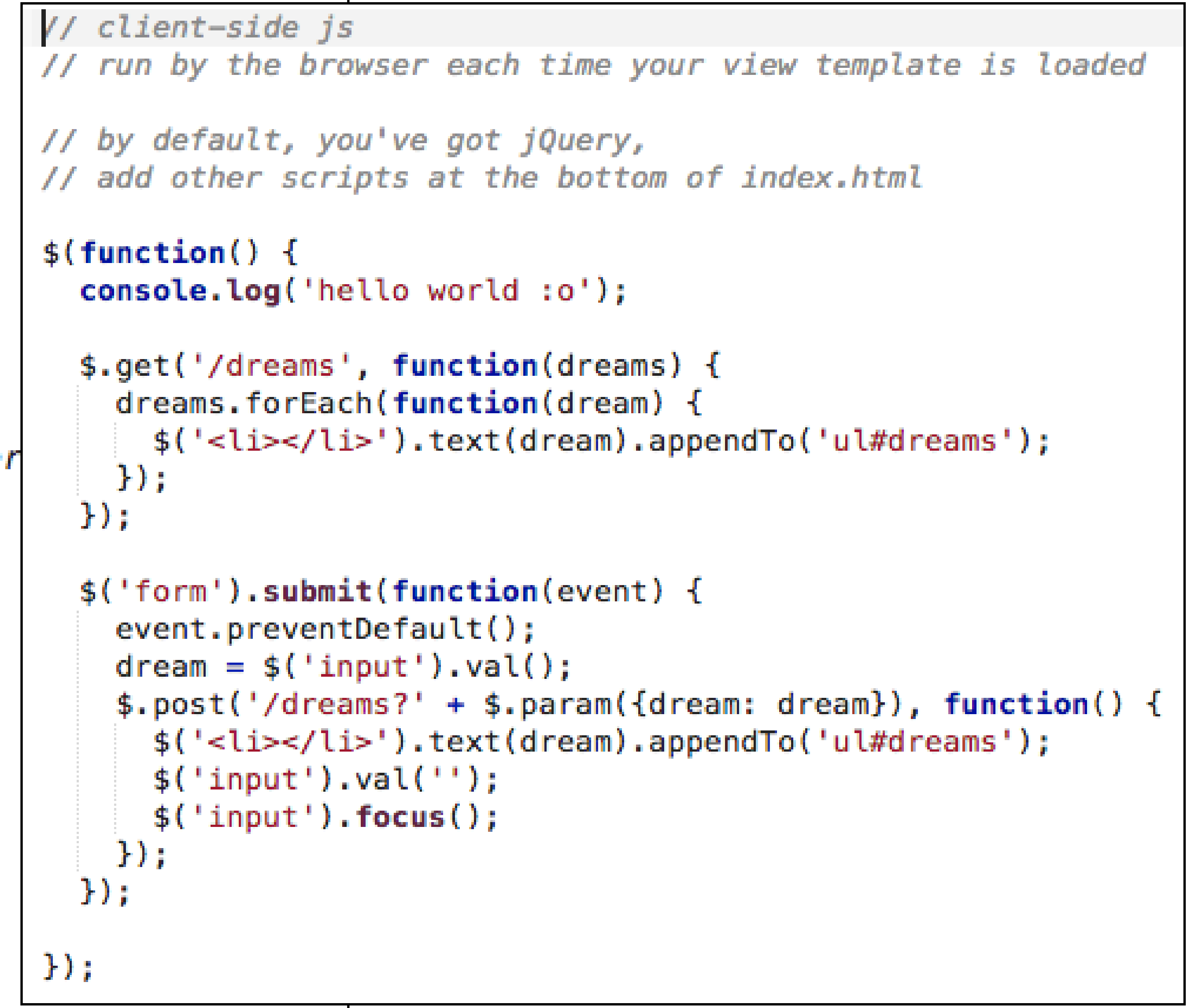
```
<body>
  <header>
    <h1>
      A Dream of the Future
    </h1>
  </header>

  <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://glitch.com">
      Remix this in Glitch
    </a>
  </footer>
```

html



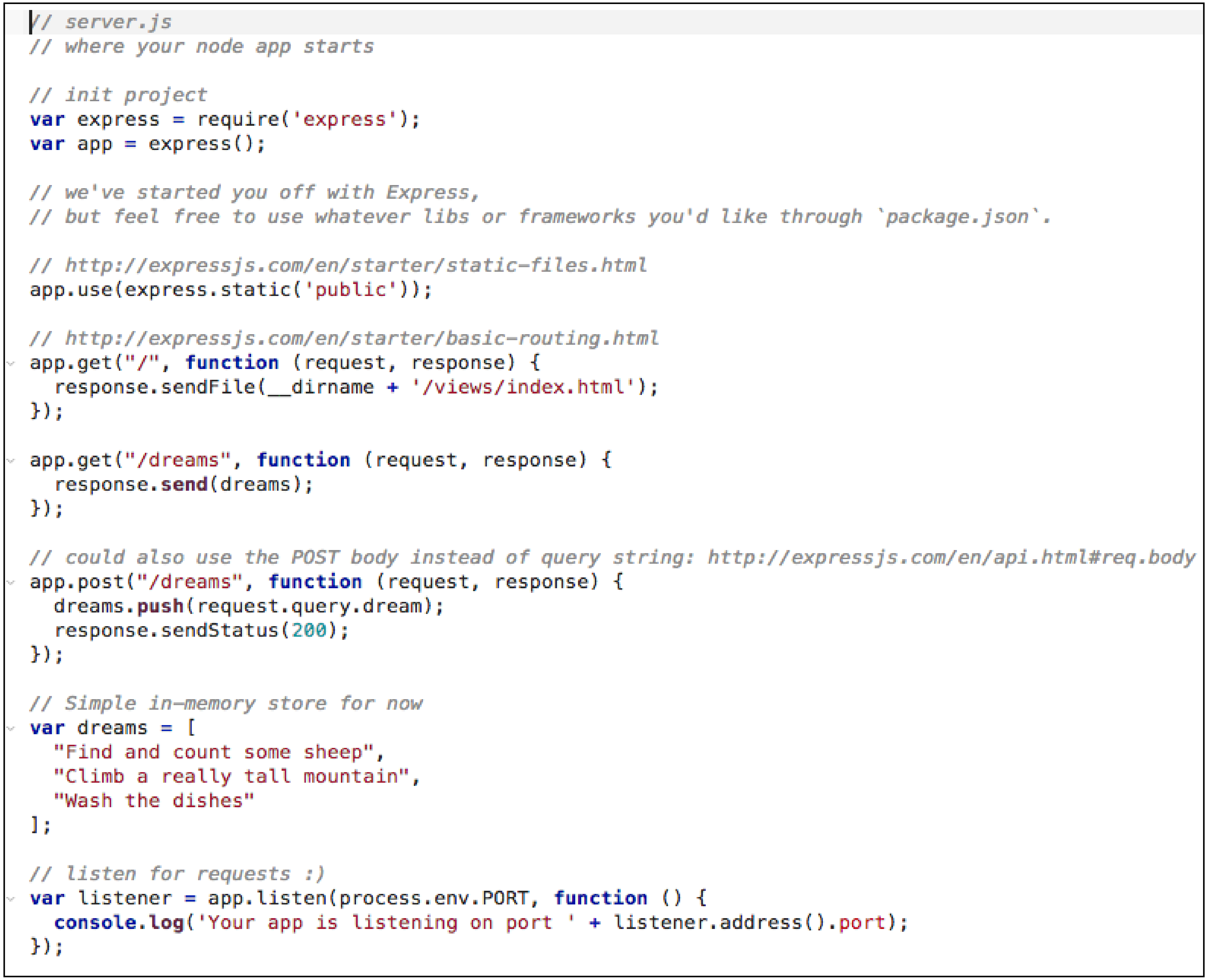
```html
<body>
  <header>
    <h1>
      A Dream of the Future
    </h1>
  </header>

  <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 javascript



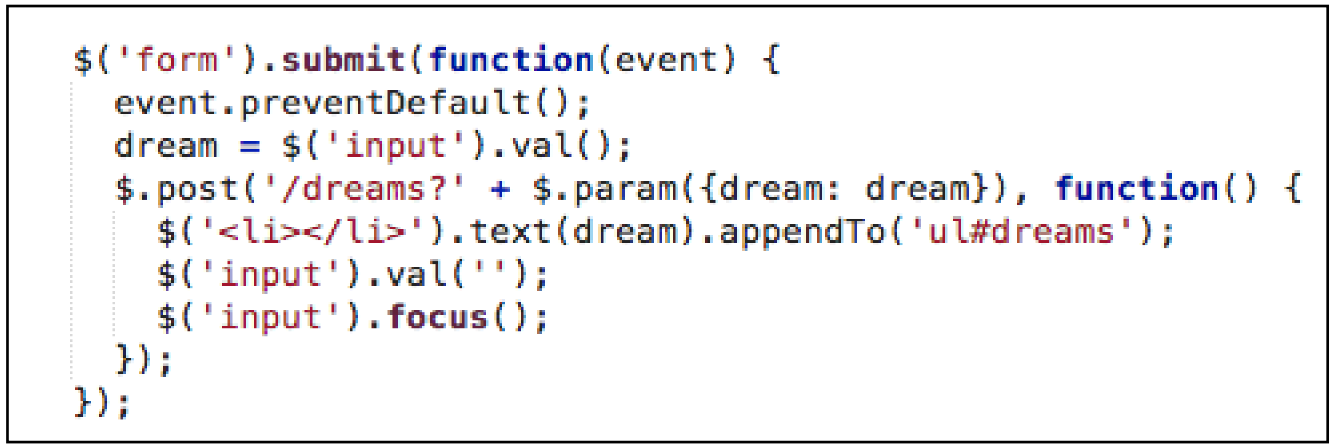
```javascript
// client-side js
// run by the browser each time your view template is loaded

// 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

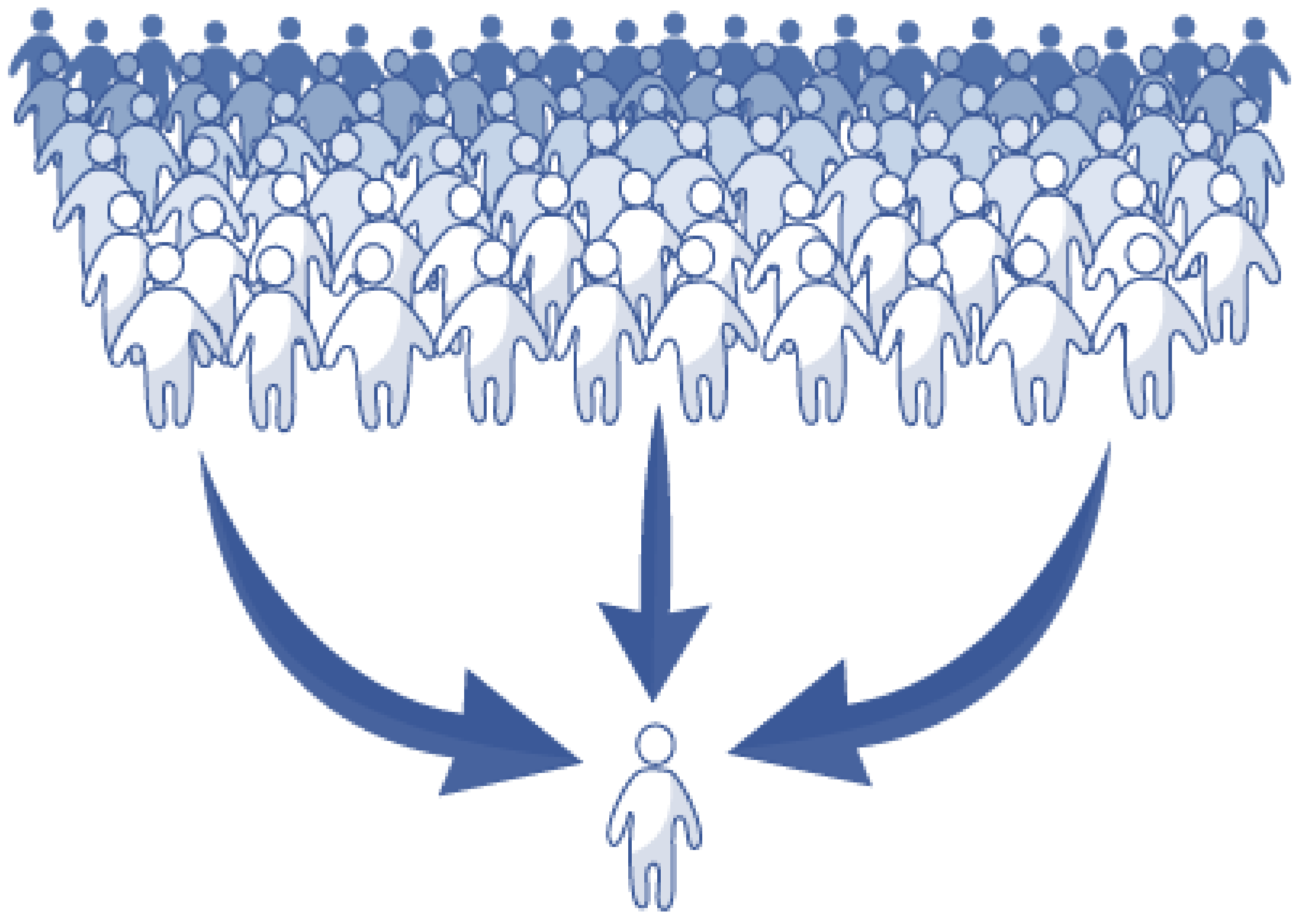


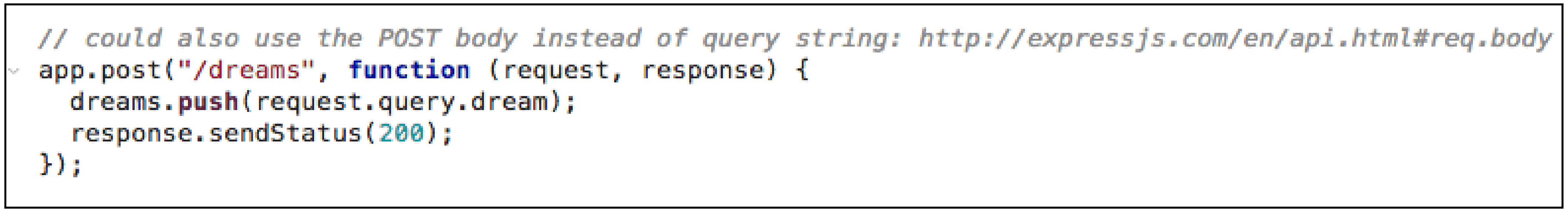
```
// server.js
// where your node app starts

// 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);
});
```

- 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();
  });
});
```



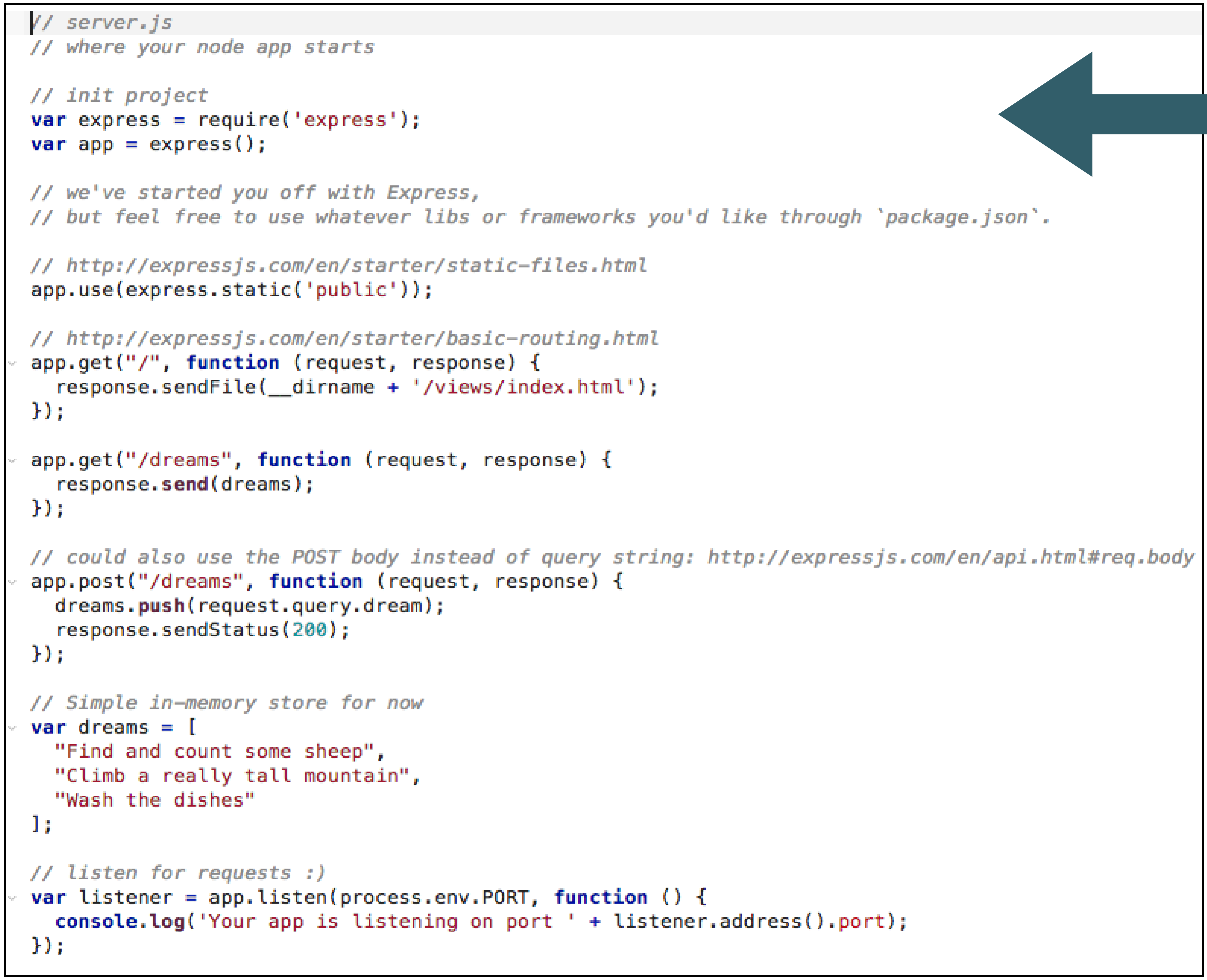


```
// could also use the POST body instead of query string: http://expressjs.com/en/api.html#req.body
app.post("/dreams", function (request, response) {
  dreams.push(request.query.dream);
  response.sendStatus(200);
});
```

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

Skills developed in this Module

- Web App Development 1
  - Basic Javascript knowledge
  - Back end development in Javascript
- Front end javascript development is delivered in a different module

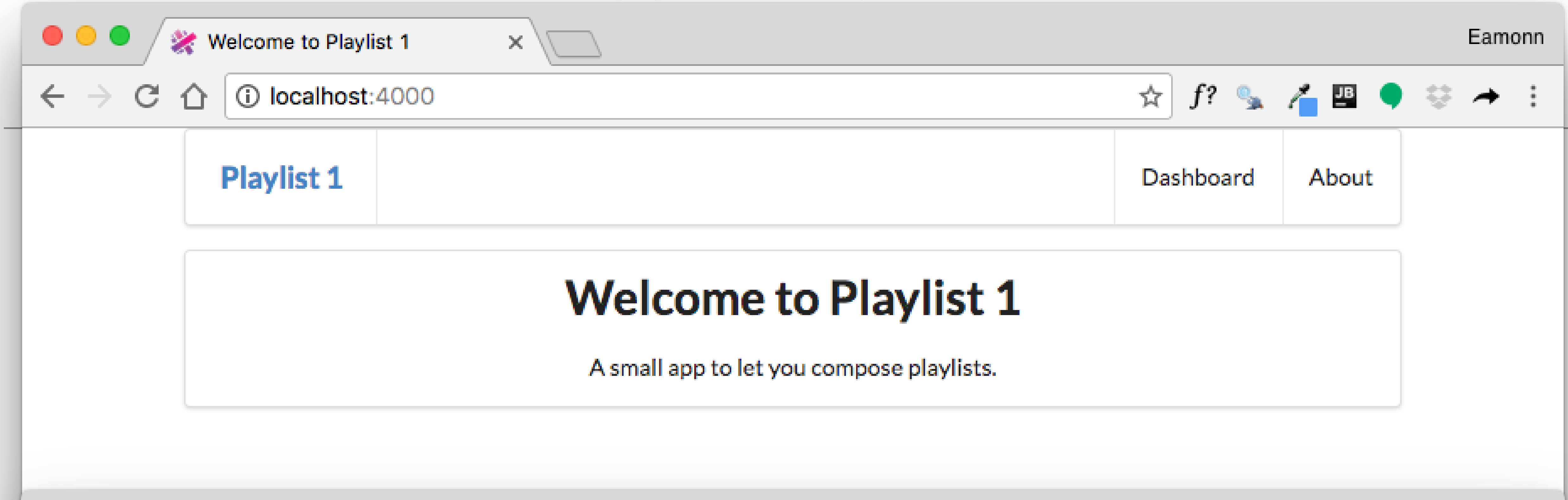


```
// server.js
// where your node app starts

// 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 to store data in models
  - create user accounts, and tie account to a each user

A tour of our first app - Playlist



Welcome to Playlist 1

localhost:4000

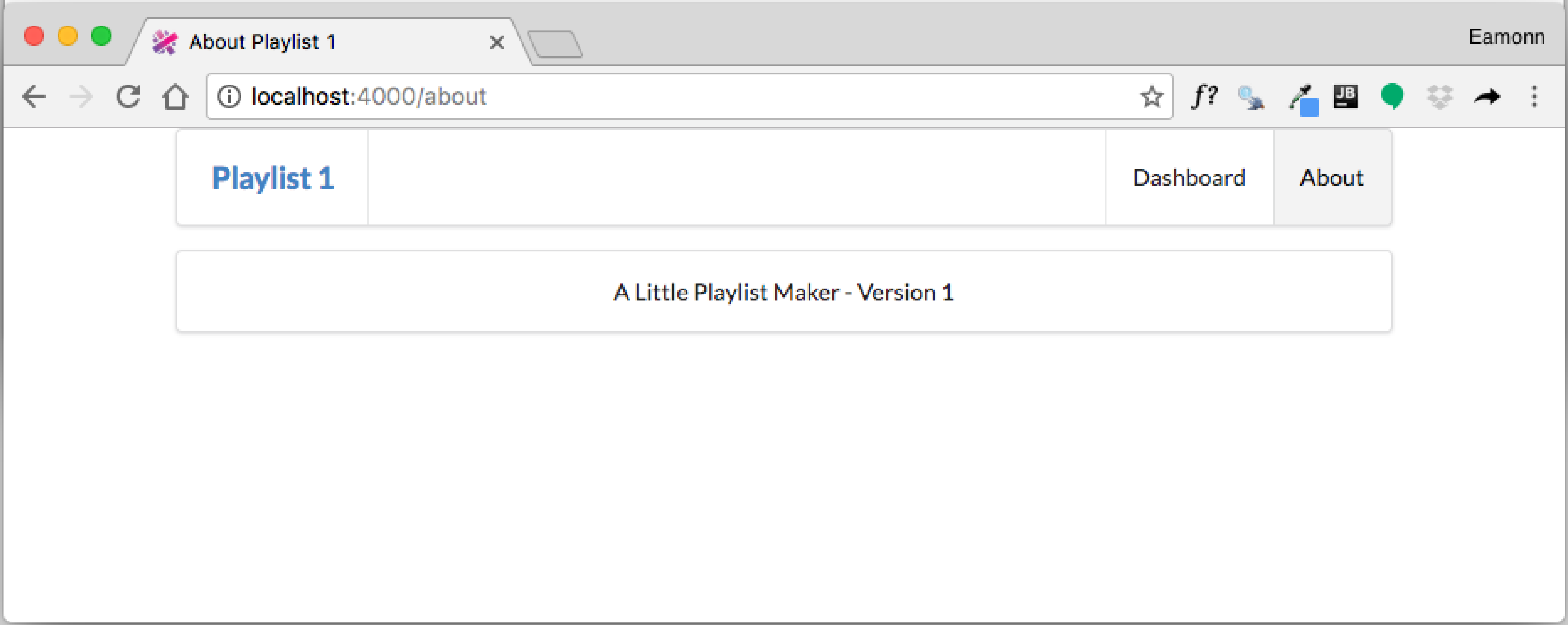
Playlist 1

Dashboard

About

Welcome to Playlist 1

A small app to let you compose playlists.



About Playlist 1

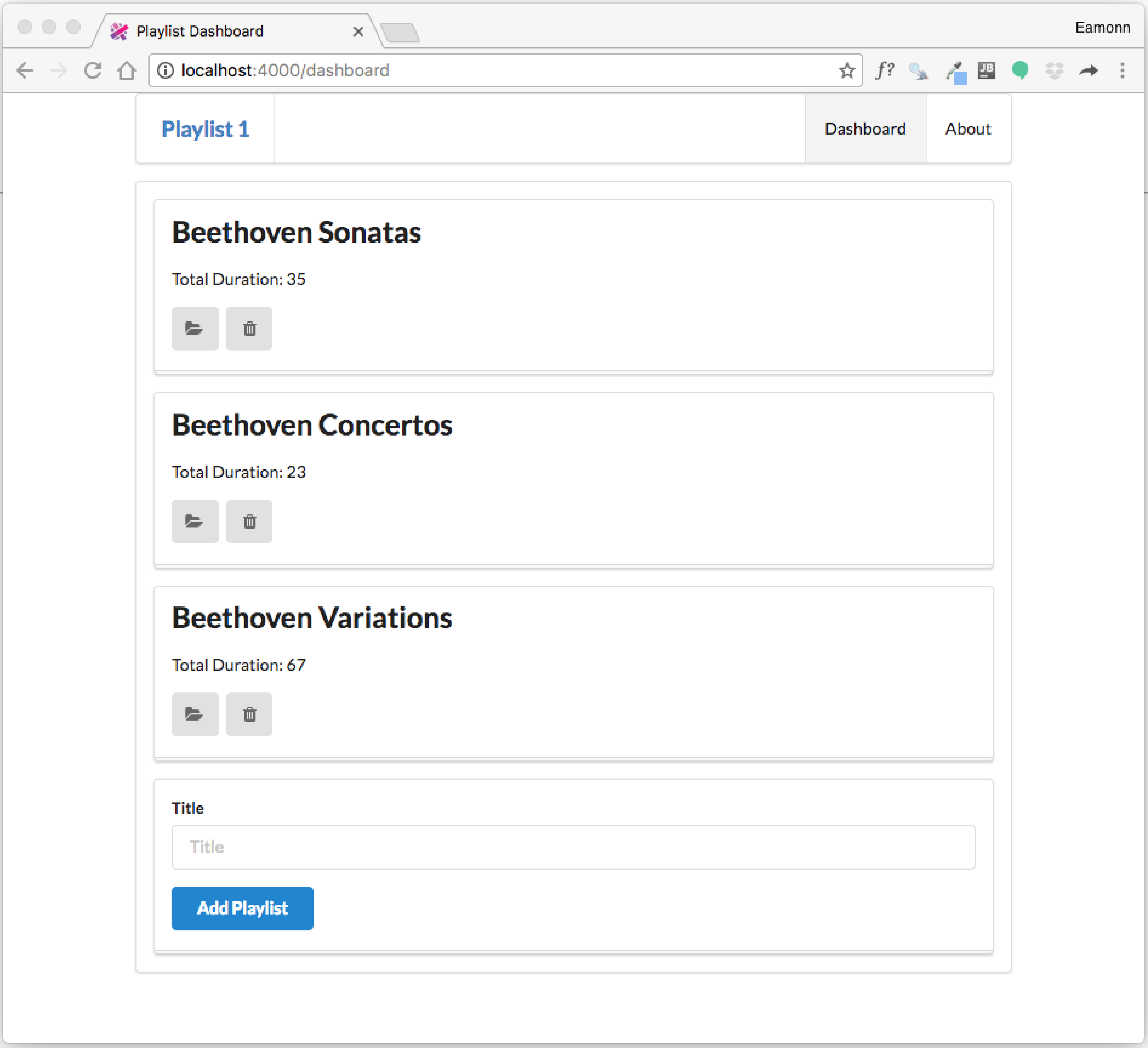
localhost:4000/about

Playlist 1

Dashboard

About

A Little Playlist Maker - Version 1



Beethoven Sonatas

Total Duration: 35

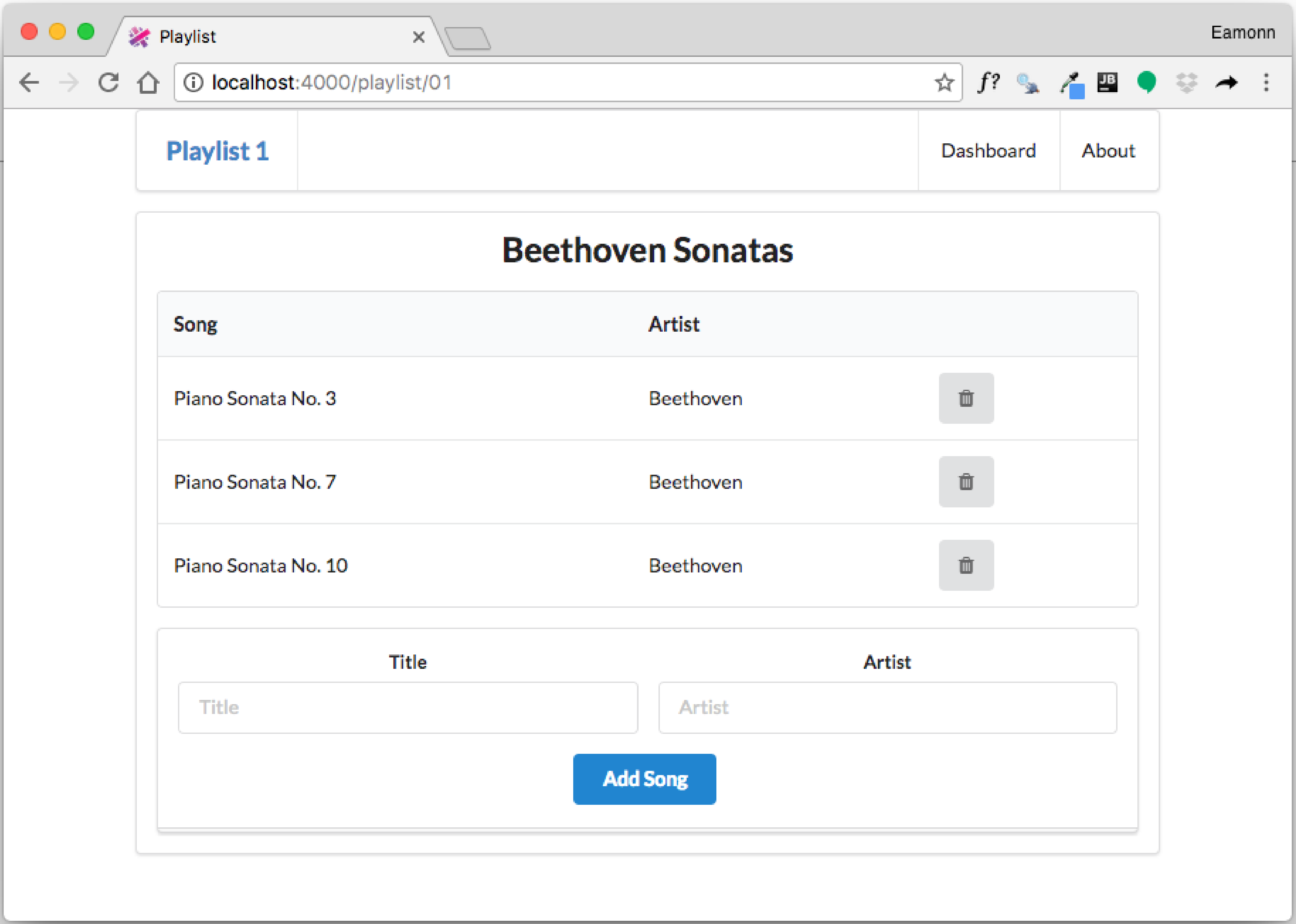
Beethoven Concertos

Total Duration: 23

Beethoven Variations

Total Duration: 67

Title



Beethoven Sonatas

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

Title	Artist
Title	Artist

Playlist Labs

- We will do three playlist labs
  - 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.
- These labs will be interleaved with Javascript Introductory labs, which will gradually introduce you to the language