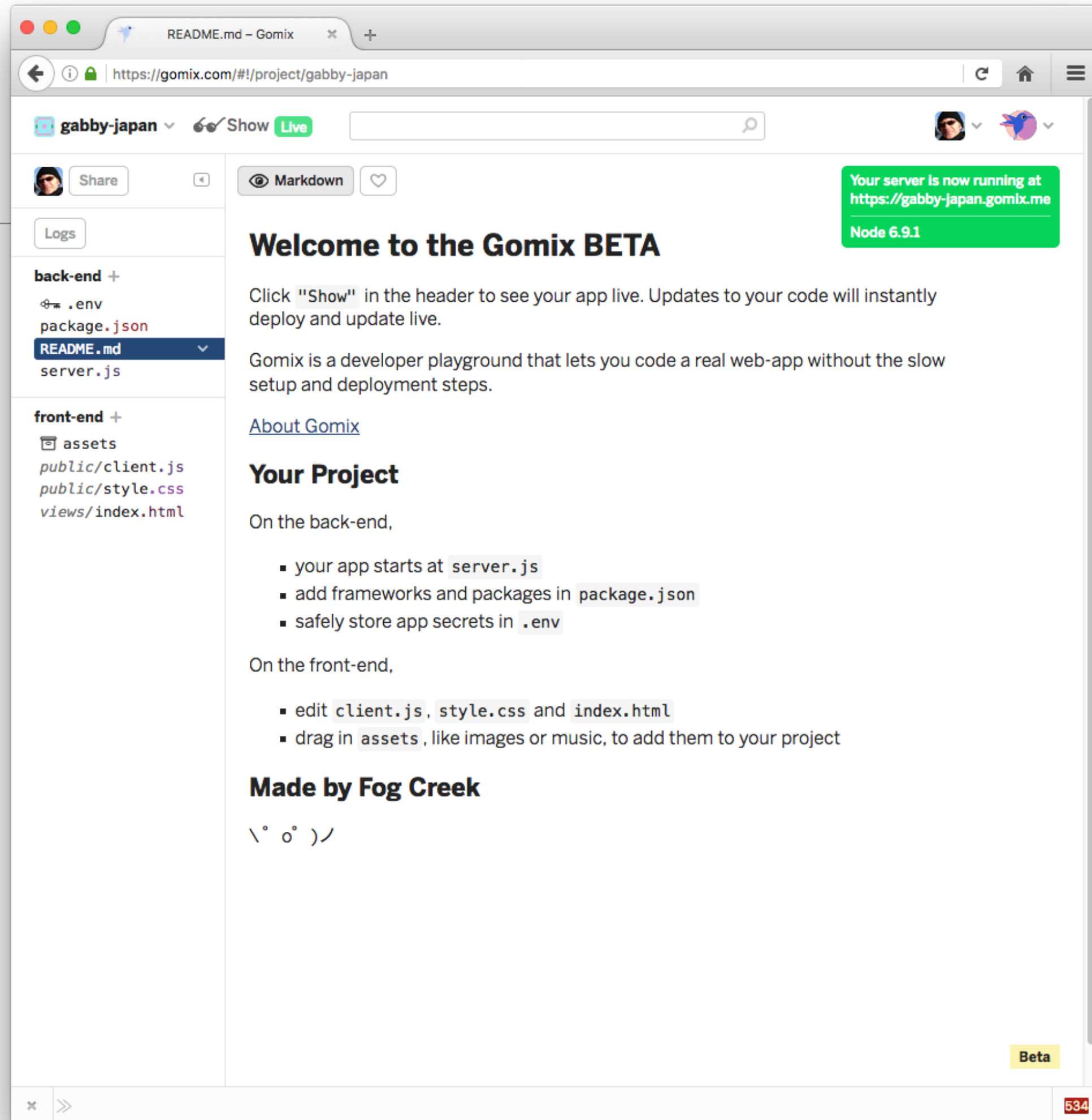


Gomix Tour

Prerequisite tools on your Workstation

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

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



Project name
(automatically
generated)

Link to running
app (to share)

Files in the
project

Current File
(editable)

Link to your
Profile

Link to
Community,
resources, options

The screenshot shows the Gomix web interface for a project named 'gabby-japan'. The browser address bar shows 'https://gomix.com/#!/project/gabby-japan'. The interface includes a header with the project name, a 'Show' button, and a 'Live' status indicator. A green notification box in the top right corner states 'Your server is now running at https://gabby-japan.gomix.me Node 6.9.1'. The left sidebar displays a file explorer with a tree structure: 'back-end +' containing '.env', 'package.json', 'README.md' (selected), and 'server.js'; and 'front-end +' containing 'assets', 'public/client.js', 'public/style.css', and 'views/index.html'. The main content area features a 'Welcome to the Gomix BETA' message, instructions to click 'Show' to view the live app, and a list of project files. Below this, there are sections for 'Your Project' with instructions for back-end and front-end development, and a 'Made by Fog Creek' section with a logo. The bottom right corner of the interface has a 'Beta' label and a page number '534'.

Project name (automatically generated)

Link to running app (to share)

Files in the project

Current File (editable)

Link to your Profile

Link to Community, resources, options

Project name: gabby-japan

Show Live

Share

Logs

back-end +

- .env
- package.json
- README.md (selected)
- server.js

front-end +

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

Welcome to the Gomix BETA

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

Gomix is a developer playground that lets you code a real web-app without the slow setup and deployment steps.

About Gomix

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

On the front-end,

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

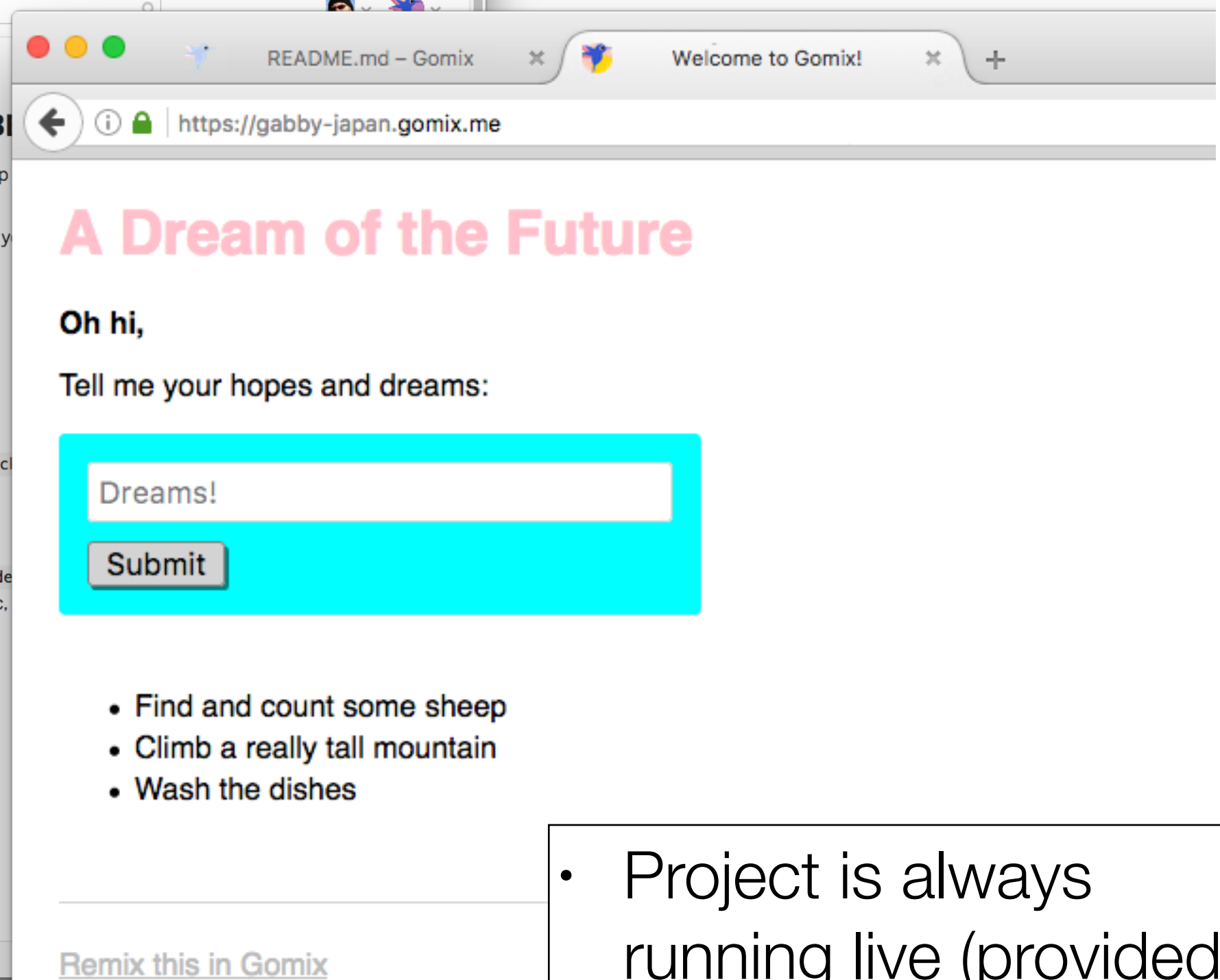
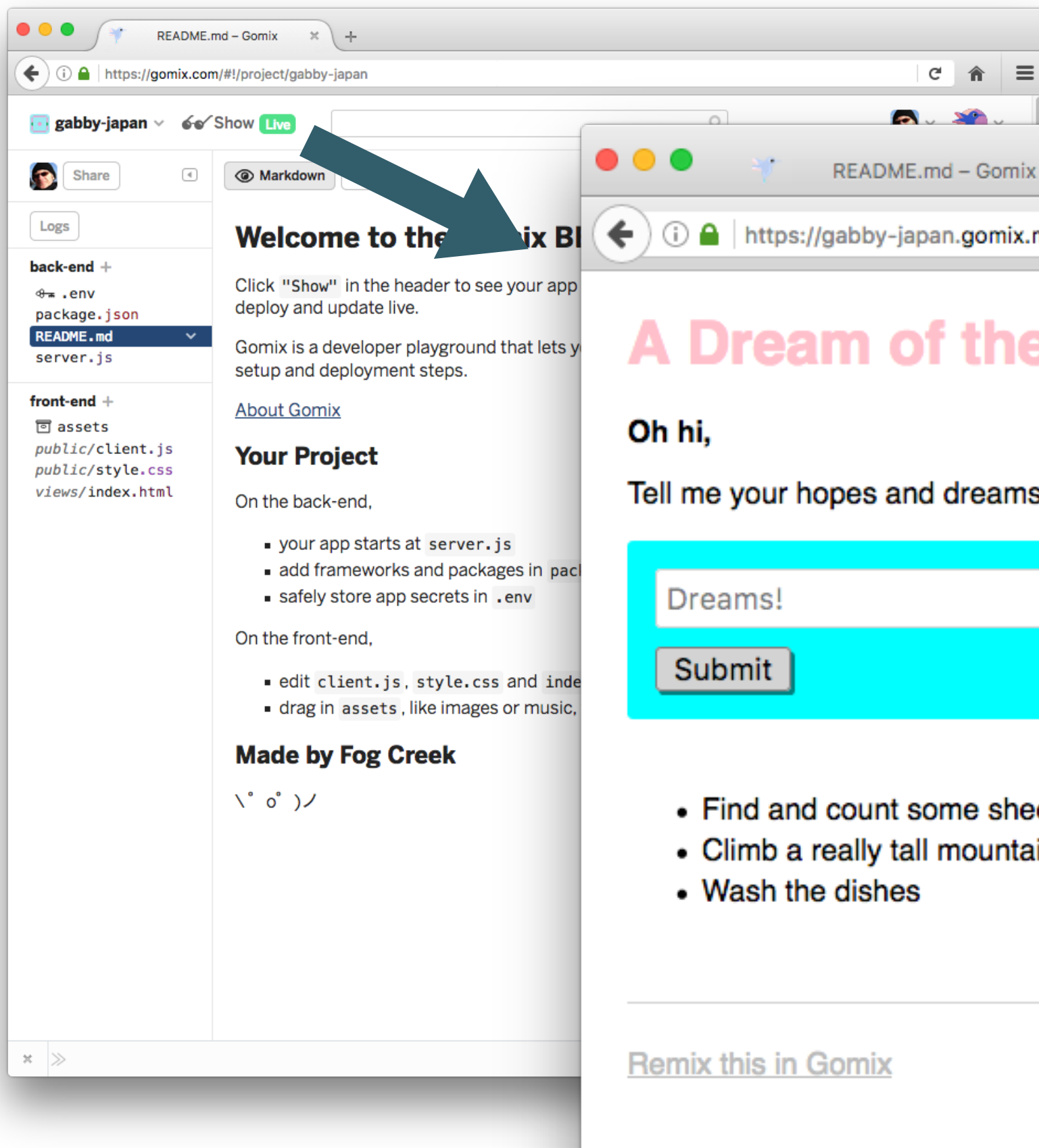
Made by Fog Creek

\° o°)/

Your server is now running at https://gabby-japan.gomix.me Node 6.9.1

Beta

534



- Project is always running live (provided there are source errors)

Project Structure

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

back-end +

🔑 .env

package.json

README.md

server.js

front-end +

📁 assets


public/client.js

public/style.css

views/index.html

Front End

front-end +

 **assets**

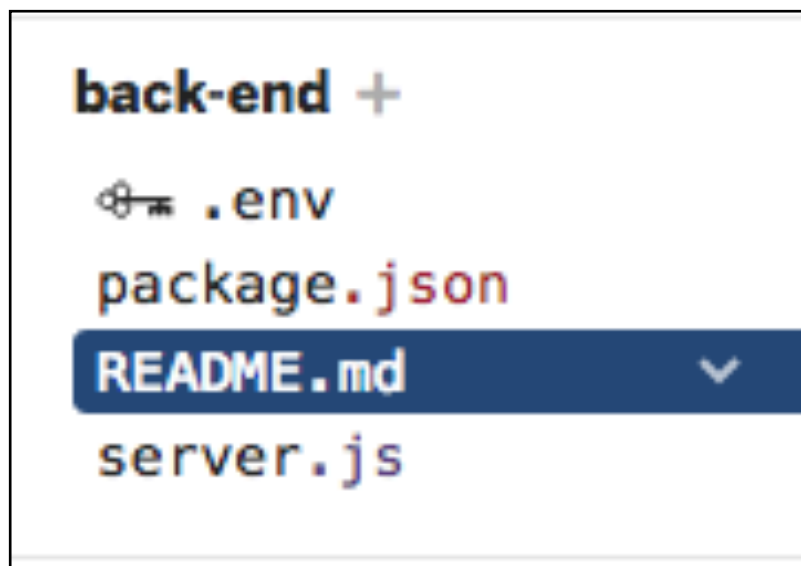
public/client.js

public/style.css

views/index.html

- Comparable to the web site you have been developing to date:
 - 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



- A new departure over web site dev 1.
- 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 gomix - but other variants are planned.

The Starter App

The screenshot shows the Gomix web editor interface. The browser address bar displays `https://gomix.com/#!/project/actually-specialist`. The page title is `actually-specialist`. The left sidebar contains a file explorer with two sections: **back-end +** (containing `.env`, `package.json`, `README.md`, and `server.js`) and **front-end +** (containing `assets`, `public/client.js`, `public/style.css`, and `views/index.html`, which is currently selected). The main editor area displays the HTML code for `views/index.html`, which is a static file. The code includes comments, a DOCTYPE declaration, and HTML elements for a title, meta tags, a header with a h1, a main section with a form and a list, and a footer with a link to the Gomix website. The code is as follows:

```
1 <!-- This is a static file -->
2 <!-- served from your routes in server.js -->
3
4 <!-- You might want to try something fancier: -->
5 <!-- html/nunjucks docs: http://mozilla.github.io/nunjucks/ -->
6 <!-- jade: http://jade-lang.com/ -->
7 <!-- haml: http://haml.info/tutorial.html -->
8 <!-- hbs(handlebars): http://handlebarsjs.com/expressions.html -->
9
10 <!DOCTYPE html>
11 <html>
12 <head>
13   <title>Welcome to Gomix!</title>
14   <meta name="description" content="A cool thing made with Gomix">
15   <link id="favicon" rel="icon" href="https://gomix.com/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   <link rel="stylesheet" href="/style.css">
20 </head>
21 <body>
22 <header>
23   <h1>
24     A Dream of the Future
25   </h1>
26 </header>
27
28 <main>
29   <p class="bold">Oh hi,</p>
30   <p>Tell me your hopes and dreams:</p>
31   <form>
32     <input type="text" maxlength="100" placeholder="Dreams!">
33     <button type="submit">Submit</button>
34   </form>
35   <section class="dreams">
36     <ul id="dreams">
37     </ul>
38   </section>
39 </main>
40
41 <footer>
42   <a href="https://gomix.com">
43     Remix this in Gomix
44   </a>
45 </footer>
46
47 <!-- Your web-app is https, so your scripts need to be too -->
48 <script src="https://code.jquery.com/jquery-2.2.1.min.js"
49   integrity="sha256-gvQgAFzTH6trSrAwOHi1Po9Xc96QxSZ3feW6kem+000="
50   crossorigin="anonymous"></script>
51 <script src="/client.js"></script>
52
53 </body>
54 </html>
55
```

A **Beta** badge is visible in the bottom right corner of the editor.

The Starter App

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](#)

index.html – Gomix

https://gomix.com/#!/project/actually-specialist

actually-specialist Show Live

Share

Logs

back-end +

- .env
- package.json
- README.md
- server.js

front-end +

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

```
1 <!-- This is a static file -->
2 <!-- served from your routes in server.js -->
3
4 <!-- You might want to try something fancier: -->
5 <!-- html/nunjucks docs: http://mozilla.github.io/nunjucks/ -->
6 <!-- jade: http://jade-lang.com/ -->
7 <!-- haml: http://haml.info/tutorial.html -->
8 <!-- hbs(handlebars): http://handlebarsjs.com/expressions.html -->
9
10 <!DOCTYPE html>
11 <html>
12 <head>
13   <title>Welcome to Gomix!</title>
14   <meta name="description" content="A cool thing made with Gomix">
15   <link id="favicon" rel="icon" href="https://gomix.com/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   <link rel="stylesheet" href="/style.css">
20 </head>
21 <body>
22   <header>
23     <h1>
24       A Dream of the Future
25     </h1>
26   </header>
27
28   <main>
29     <p class="bold">Oh hi,</p>
30     <p>Tell me your hopes and dreams:</p>
31     <form>
32       <input type="text" maxlength="100" placeholder="Dreams!">
33       <button type="submit">Submit</button>
34     </form>
35     <section class="dreams">
36       <ul id="dreams">
37
38       </ul>
39     </section>
40   </main>
41
42   <footer>
43     <a href="https://gomix.com">
44       Remix this in Gomix
45     </a>
46   </footer>
47
48   <!-- Your web-app is https, so your scripts need to be too -->
49   <script src="https://code.jquery.com/jquery-2.2.1.min.js"
50     integrity="sha256-gvQgAFzTH6trSrAWoH1iPo9Xc96QxSZ3few6kem+000="
51     crossorigin="anonymous"></script>
52   <script src="/client.js"></script>
53
54 </body>
55 </html>
```

Beta

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

```
<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="Your dream" />
      <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

```
// 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);
});

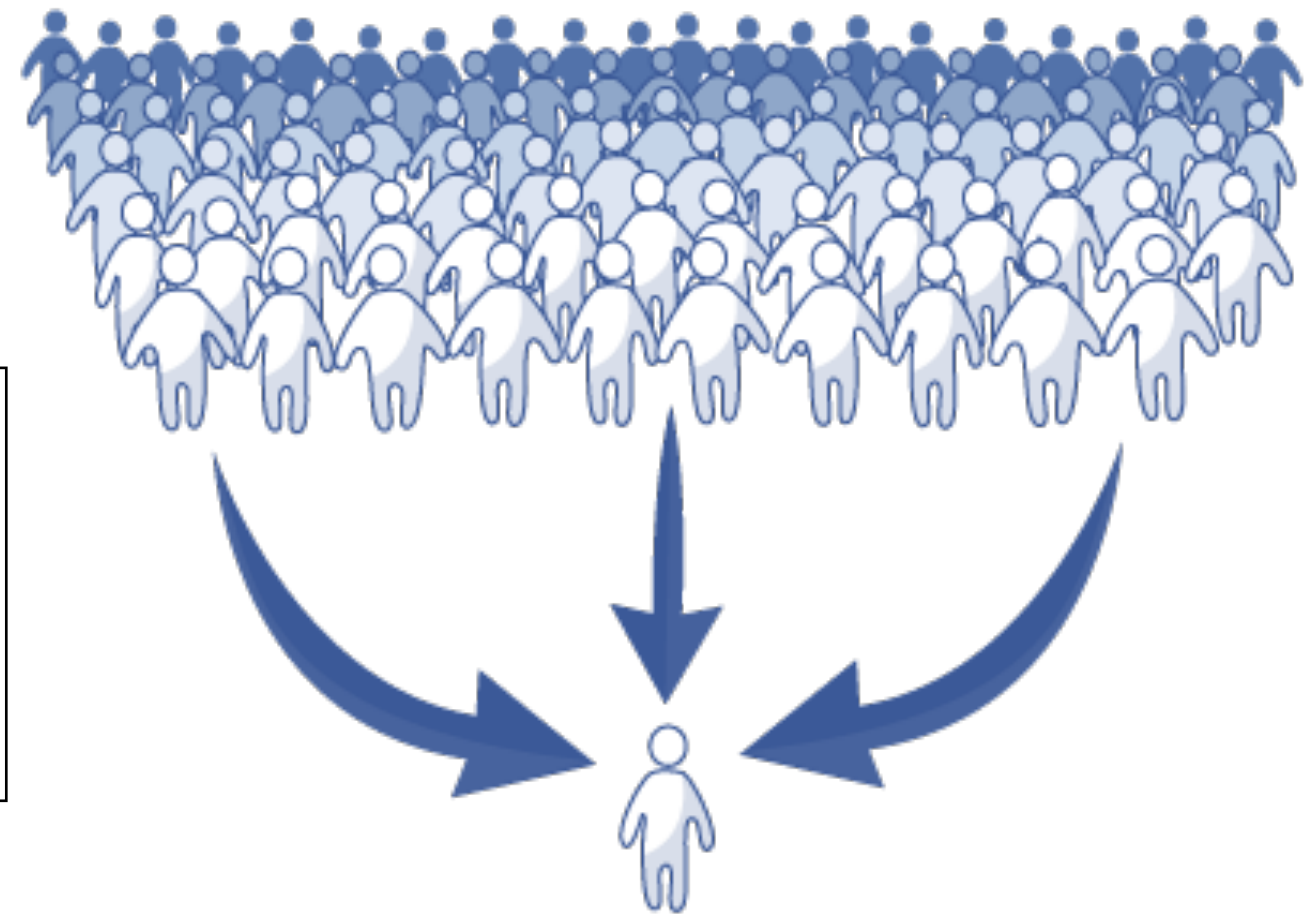
// could also use the POST body instead of query string: http://expressjs.com/en/api.html#req.body
~ app.post("/dreams", function (request, response) {
  dreams.push(request.query.dream);
  response.sendStatus(200);
});

// 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 for this Course

- Web App Development 1
 - Basic Javascript knowledge
 - Back end development in Javascript
- Front end javascript development is deferred for a future course

```
// 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'));

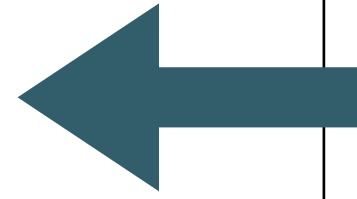
// http://expressjs.com/en/starter/basic-routing.html
app.get("/", function (request, response) {
  response.sendFile(__dirname + '/views/index.html');
});

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);
});
```

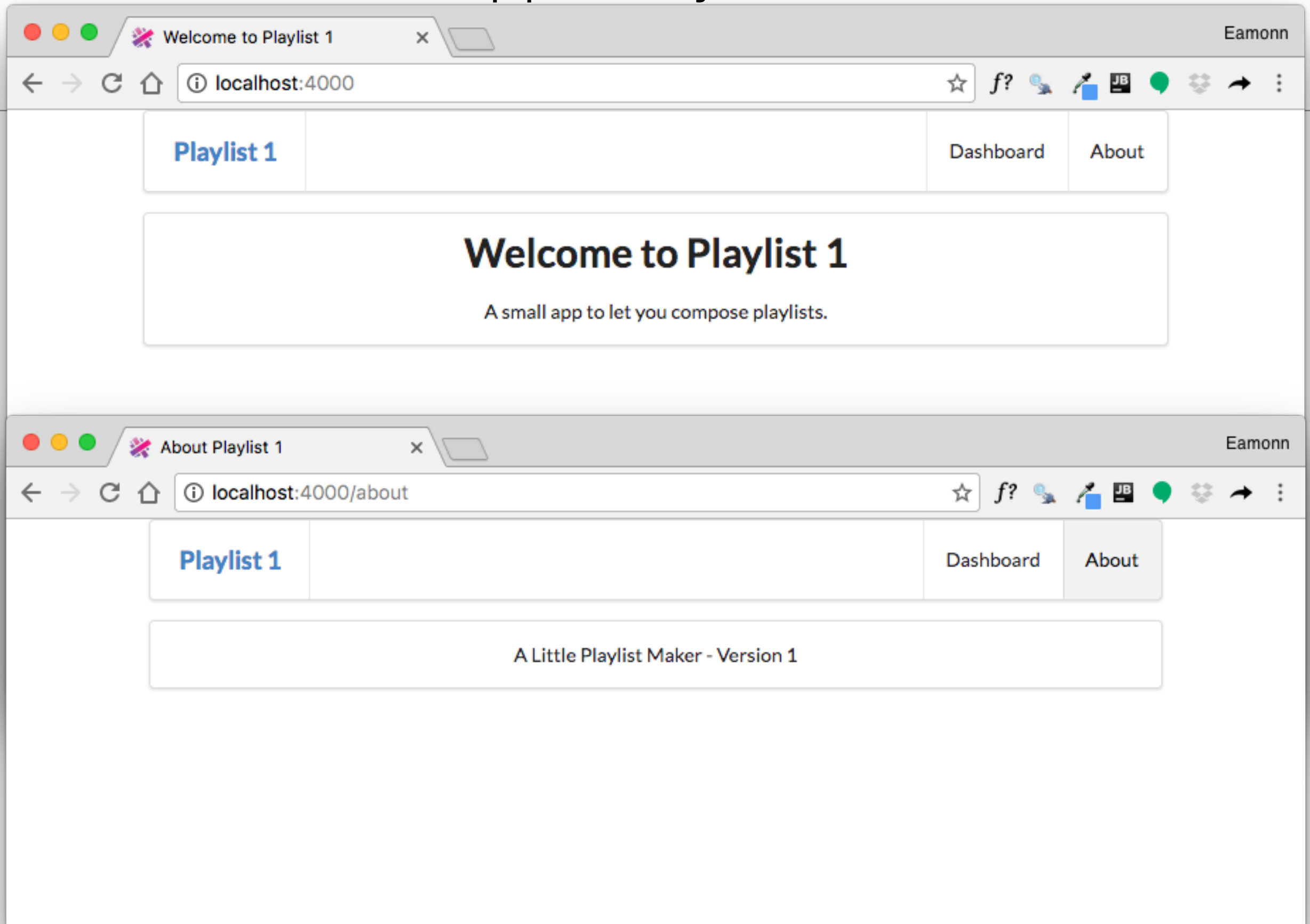


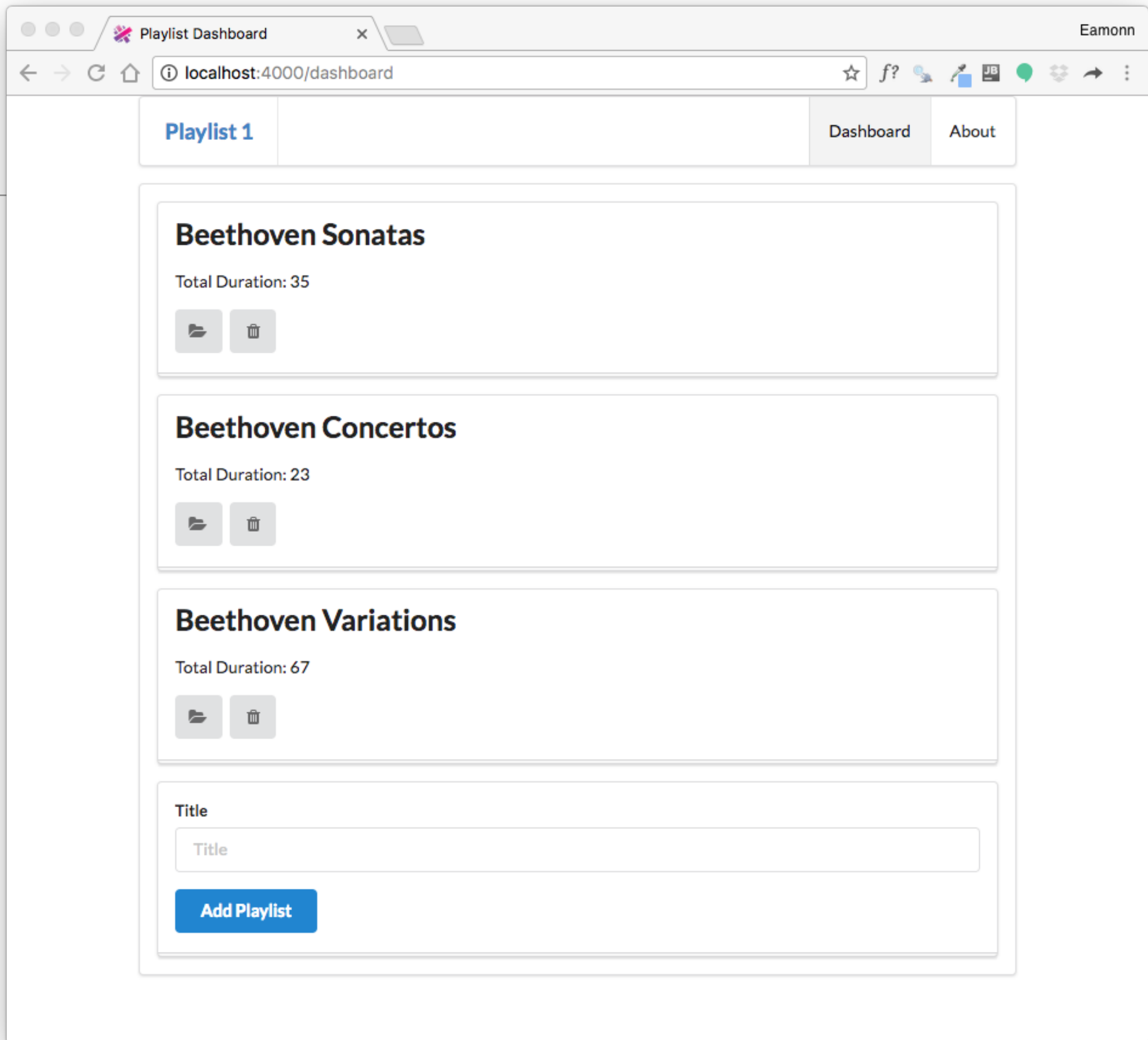
We will learn what all of this means.

- + how to build a fully featured web app including:
 - templating (like semester 1)
 - forms to submit information
 - How store information in models
 - create user accounts, and tie account to a each user

All of this requires
intermediate level
Javascript skills

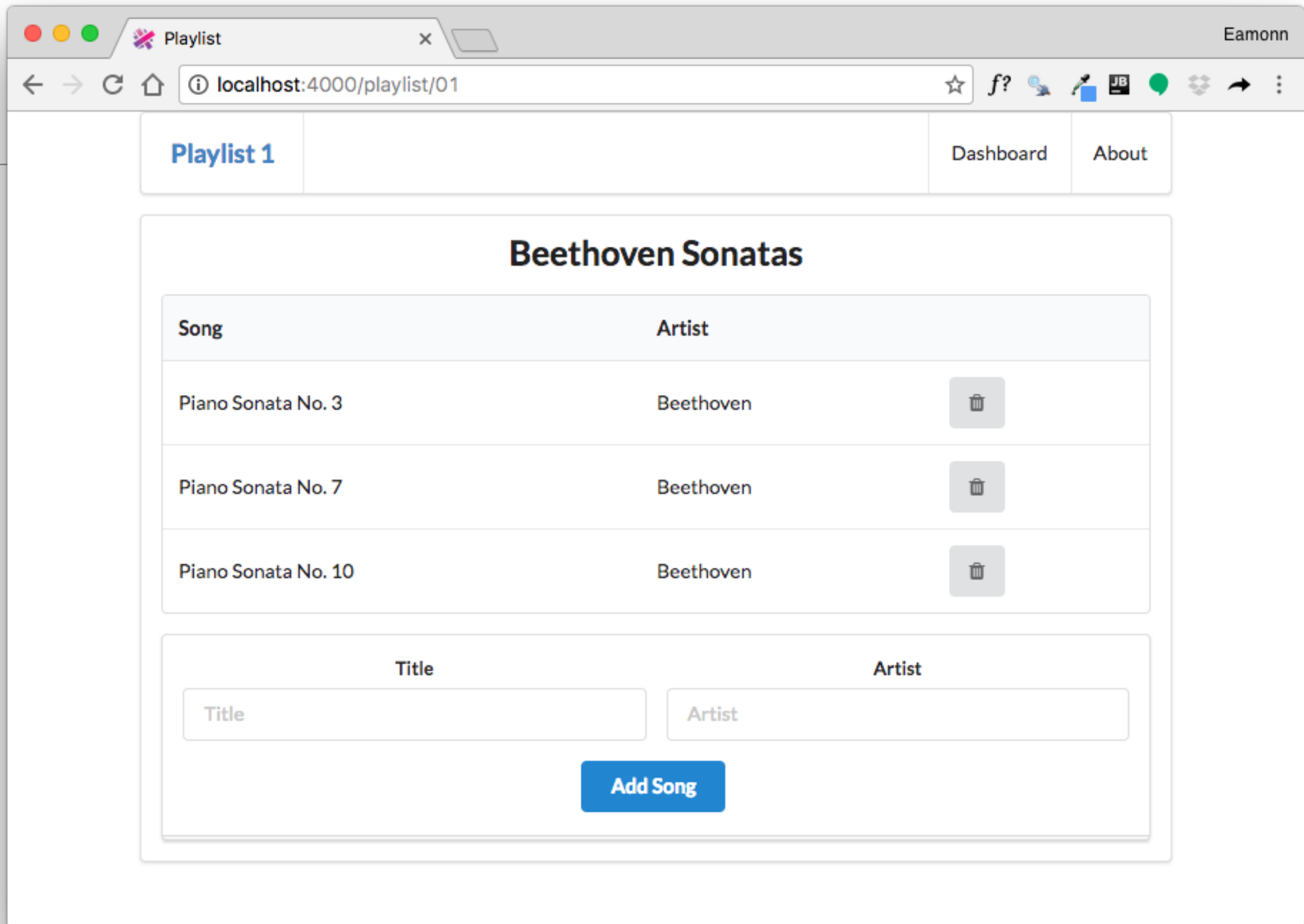
A tour of our first app - Playlist





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.






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

Title

Artist

Add Song