Using Node and express

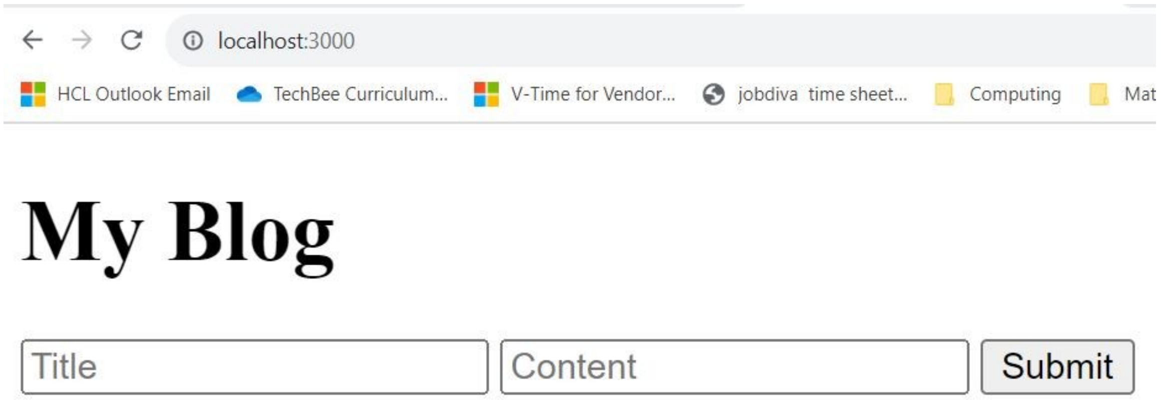
In this lab, you'll use the **ejb template engine with express** to dynamically update an HTML page using data returned from the express server.

Short story – you'll have *starter code* that you will complete.

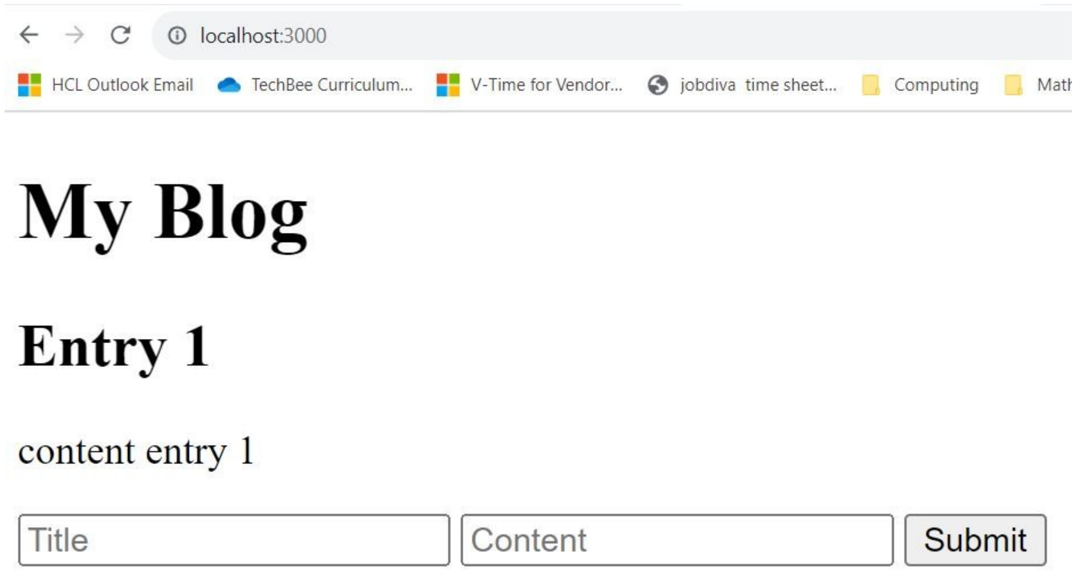
Longer story – you'll learn about some node packages used with express as you complete the lab.

First, let's see what the completed project renders:

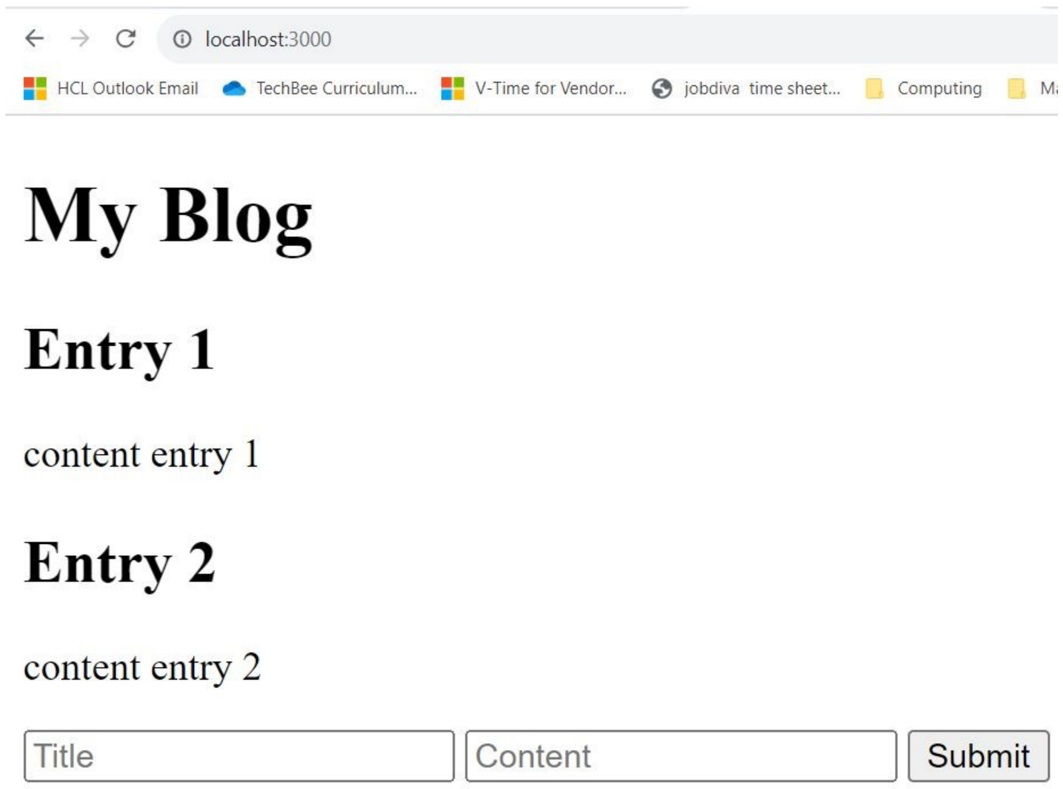
Start the express server, navigate to *localhost:3000* and press ENTER. You should see:



Enter *Entry 1* and *content entry 1* and click SUBMIT:



Enter *Entry 2* and *content entry 2* and click SUBMIT:



My Blog

Entry 1

content entry 1

Entry 2

content entry 2

You get the idea.

The ebj template engine

Template engines allow for mixing static (HTML tags) and dynamic content (expressions that resolve to previously computed data) into a single rendered page. Your labs dealt with using the DOM to create HTML elements or generating strings that contain HTML tags and appending/updating DIVs on a page to dynamically render content. We all know that's not how it's done in the real world.

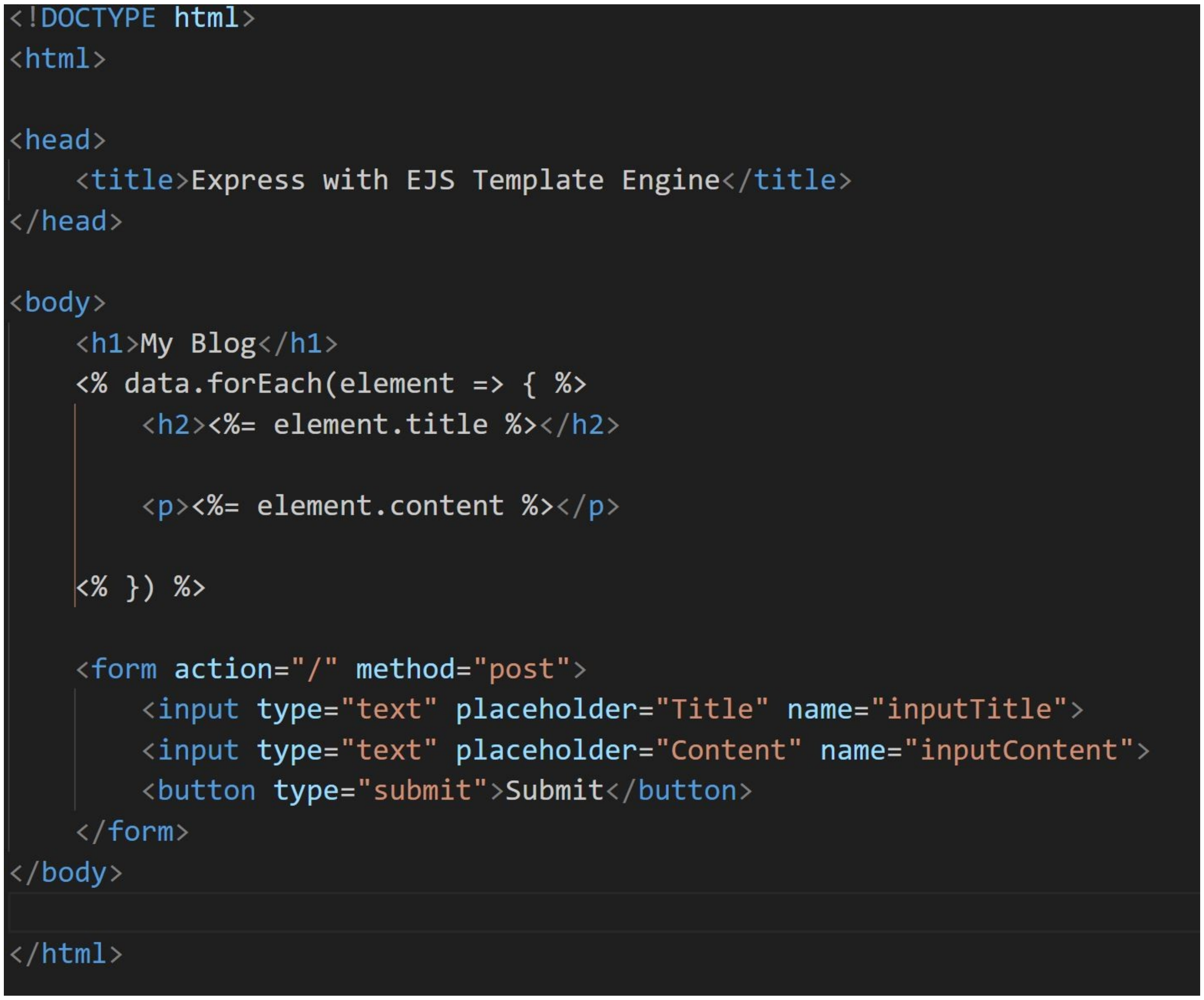
Express may use several template engines; the ejb engine is typical of them all. In short, the ejb engine combines static HTML with expressions. The engine *resolves the expressions* by substituting data from the server. Also, template engines have programming structures (loops, if/else statements) that are used to display collection data.

Let’s take a look at the ejb page you’ll use:

home.ejb

ejb requires that **you use the extension *ejb* AND you place your ejb page(s) in a folder named *views*.**

Here’s the page you’ll use, named **home.ejb**, stored in the **views** directory:



```
<!DOCTYPE html>
<html>

<head>
    <title>Express with EJS Template Engine</title>
</head>

<body>
    <h1>My Blog</h1>
    <% data.forEach(element => { %>
        <h2><%= element.title %></h2>

        <p><%= element.content %></p>

    <% }) %>

    <form action="/" method="post">
        <input type="text" placeholder="Title" name="inputTitle">
        <input type="text" placeholder="Content" name="inputContent">
        <button type="submit">Submit</button>
    </form>
</body>

</html>
```

The ejb statements that the engine uses to substitute text are shown in the **<% %>** brackets.

The item **data** in the *forEach* loop is data generated by the server. Each element of data has two properties, *title* and *content*. The server creates an array entry of an *object with these two properties*.

The server.js starter code

I’ll not show the starter code here – take a look at *server.js* in the starter code folder.

The code has copious comments to get you going.

All required code is present on one pr more slides we’ve seen in class.

Code required to get this lab up-and-running using the ejb engine and other modules is coded for you.

You’ll have to **npm install** a few modules. You’ll figure it out.