Feedback on preliminary Web Tech

Pair: ny18442, ba18276

You didn't give a report about what you did. I hope you will provide one in the final submission. Some bullet points under each heading is enough (but it will be crucial for image work, for example, and I would like to know things like whether your server is going to be secure).

You have done some decent work in some areas, and your estimated grades look fine.

General feedback

The main weakness I have seen, which I haven't necessarily commented on individually, is security. Here are the basics.

Database security

This applies to you if you are using a served database, that's MariaDB, MySQL, PostgreSQL, MongoDB, ...

You are opening a database server port for your web server to connect to. That means a hacker can connect direct to your database, without going through your web server. There are tens of thousands of completely unprotected databases in use by companies out in industry for their critical data. I hope you aren't going to add to them (or connect to them, steal their data, and wipe them clean, even if they deserve it).

Step 1: Check if there is a way of restricting database server requests to `localhost` only. That's a quick temporary way of ensuring safety. There probably isn't a way to do that easily.

Step 2: Establish a username and password for database access. That means configuring the database so your data can't be accessed without them, and then using them in your web server when you connect. Since no human ever has to type the password, it can be long and random.

Step 3: A password is useless unless it is secret. That means the communication between your web server and your database server needs to be encrypted, otherwise your password will be readable by a hacker as it travels over the network. Your database may offer its own (probably weak and inadequate) encryption or (preferably) SSL-based encryption. You need to turn that on.

Step 4: You need to check security of your web server. If hackers can read the source of your server, they can see the password.

Maybe now you can see why I recommend using an embedded database system. Also, frameworks and tutorials almost never point out that what they are telling you to do is totally insecure. So maybe you can see why I don't recommend blindly following frameworks or tutorials.

Web server security

This mainly applies if you are using express, rather then starting from my server. Again, almost every tutorial describes how to construct a dangerously insecure server.

Step 1: If you are only interested in doing the simplest safe thing, don't start up your server with:

```
app.listen(3000);
```

Instead, start it up with:

```
app.listen(3000, 'localhost');
```

The documentation and tutorial support for this option is appalling, but you can find out about it if you try hard enough. It makes sure that the web server only responds to requests from a browser running on the same computer. Obviously, this doesn't work if you want to host your site somewhere.

Step 2: Check all the information in the server and database chapters of my notes, to guard against URL and SQL injection attacks. Beware that express tutorials recommend lots of extremely helpful and clever security components but, as far as I am aware, not one of them does simple URL validation properly, so they give you a false sense of security.