W3D6 Homework

There are 4 assignments, although the last two should be done in the same project. I therefore recommend making a **w3d6** directory, and then making q1, q2, and q3_4 subdirectories.

The code below demonstrates the workings of a basic **index.js** file that calls an **index.ejs** template. To run it place the index.js file inside the **q1** directory, then create a subdirectory called **view** inside q1 and put the index.ejs file into that.

Next execute the following commands to install and then run everything inside your q1 directory:

```
$ npm init
$ npm install express -save
$ nmp install ejs -save
$ node index.js
```

index.js

```
const express = require('express');
const path = require('path');
const app = express();

app.set('view engine', 'ejs');
app.set('views', path.join(__dirname, "view"));

app.get('/', (req, res) => {
  const date = new Date();
  res.render("index", {
    time: date.toTimeString(),
  });
});

app.listen(3000);
```

view/index.ejs

```
<!DOCTYPE html>
<html lang='en'>
  <head>
    <meta charset="utf-8">
    <title>Current time</title>
  </head>
  <body>
    This page was loaded at <%= time %>
  </body>
</html>
```

Assignments:

1. Update the example code to link to ‘day.css’ between 6am and 6pm, and to ‘night.css’ between 6pm and 6am. See the W3D5 homework for a description of what should be in these css files.
2. Update q2 from W3D5 to use templates (both for the form and for the result output). Feel free to copy your code from w3d5/q2 to get started, but remember to npm init, npm install express, and npm install ejs in your w3d6/q2 directory first.
3. In your q3_4 project make a product page that can display different products based on what the controller passes. Your template should receive parameters from your controller for the product’s name, price, description, and id.

   There should also be a ‘Add to Cart’ button on the page, which is part of a form that POSTs the product id to “/addToCart”. You don’t have to implement anything yet for this addToCart url (no controller or view).

   To be clear, should make both an index.js controller and a product.ejs template (which receives its data from the controller, where it’s hard-coded). You are free to decide what exactly the product page should look like.
4. Add to your q3_4 project a shoppingcart page that displays the ‘contents’ of a shopping cart. Your template should receive an array of products, where each product has a name, price and quantity.

   Again, you need to make a index.js controller and shoppingcart.ejs template that receives its data from the controller (where it’s hard-coded). You are free to decide what exactly the shopping cart page should look like.

An ideal solution for question 3 and 4 uses common parts (for instance a header or footer) that are used in both the product page and the shopping cart page (using the ejs include / layout syntax).