

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
$ npm install ejs -save
$ node 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.toString(),
  });
});

app.listen(3000);
```

index.js

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

view/index.ejs

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 shop page that shows what kinds of products are in the shop. The list of products should just be a hardcoded list in your index.js.

Then also make a product page that displays the details for a product (based on which product they clicked on the shop page) and provides a 'Add to Cart' button.

The 'Add to Cart' button should be part of a form that POSTs the product's name and price to "/addToCart" – you'll write the logic for this POST in the next quesiton.

To be clear, you should make an index.js controller, a shop.ejs template, and a product.ejs template (both of which receive data from the controller, where it's hard-coded). You are free to decide what exactly these pages should look like.

4. Because we don't have sessions yet we'll use a static / application wide object to be our shoppingcart. The problem with this is that all users have the same cart – we'll fix that problem with sessions on the next homework.

Write a handler method for /addToCart that receives the name and price and then puts them into the shoppingcart object using the name as the key. The value should be an object containing name, price, and quantity.

If the same product is added again its quantity and price shouold be updated in the cart.

The post should redirect to a shoppingcart page that displays the 'contents' of a shopping cart. Your template should receive the cart object, where each entry has a name, price and quantity.

In other words: update the index.js controller to make a POST to /addToCart and a GET for /cart and make a shoppingcart.ejs template that receives its data from the controller. You are free to decide what exactly the shopping cart page should look like.