Class Notes 10/11/2017

Zazil Toma M/W 7/2017 Cohort

* MVC
  + View

The user interface. Can be the page or the command line or other thing the user interacts with.

* + Controller

Processes the logic for the views. It’ll validate the data and sends to the database for example. The controller determines which view(s) will be displayed and creates instances of the objects created by the models. It builds or gathers the data to pass to the views using the models.

* + Models

The logic. Data / entry structure. What we want to store. So like modules that contain functions and constructors.

* Have separate folders for each – Views, Models and Controllers
* An entry line script when working with node. Single point of entry (like a controller) – i.e. app.js
* Don’t overarchitect a solution though.
* i.e. a credit card model can indicate all the things that a card would have so you can validate card info entries from any view using this model.
* Composition vs inheritance video is about the models
* Define the models by what they do and from that you can better define the properties they need
* Should be able to have very little code in the controllers file because all it’s doing is using the models and determining what is displayed in the views
* Activity – (My attempt)
  + Views: User View, Admin View
  + Models: getWeather, getUserSearches
  + Controller: directs the user to either get the user page or the admin page and retrieves the appropriate data for each
* Question – are constructors models? Or are they part of models? Part of model
* Use an obvious name for the entry point file. Otherwise they have to look into the files to know which one to use
* Common practice is index or app or server if it’s a server
* Solution:
  + Separate controllers for users and admins – AdminController.js and PublicController.js
  + Models:
    - WeatherLookupModel.js
    - App.js
* Start with controllers and pseudo code the actions
* Question – what is the difference between controller and app.js file?
* const is a variable that doesn’t change
* Always have error first callbacks. Have error as the first argument.
* let - The difference is scoping. var is scoped to the nearest function block and let is scoped to the nearest enclosing block, which can be smaller than a function block. Both are global if outside any block but window.[variable] will show undefined properties for let variables.
* ? and : mean if else?
* What is cb? Callback function
* Console.logs should be in the view files not controllers and models
* Inquirer is the view in this case
* A controller uses the model and updates the view
* Everything passes through the controller
* Analogy – controller of a tv
* The stuff inside the .then after the inquirer is the models stuff
* Even if only have one function in the controller, that’s ok. It’s setting up for scalability