|  |
| --- |
| Full Stack Web Development Specialization Capstone Project |
| Final Report |
| Course Manager |

|  |
| --- |
| Francisco Villar  5/14/2017 |

1. Introduction

While developing an app, you go though many different tutorial videos in order to get the skills you need. That’s why I want to create Courses Manager, an app that tracks all the different courses/videos that a user sees to avoid watching a video multiple times if I don’t want to.

In Course Manager the user will be able to login and register to the website. Also, the user will be able to View, Add, Edit and Delete courses from their profile. Finally, and more important, the user will be able to access the about us page.

2. Design and Implementation

Transforming the design to code was easy, since we spend so much time on the design phase; it covers for most of what was implemented later.

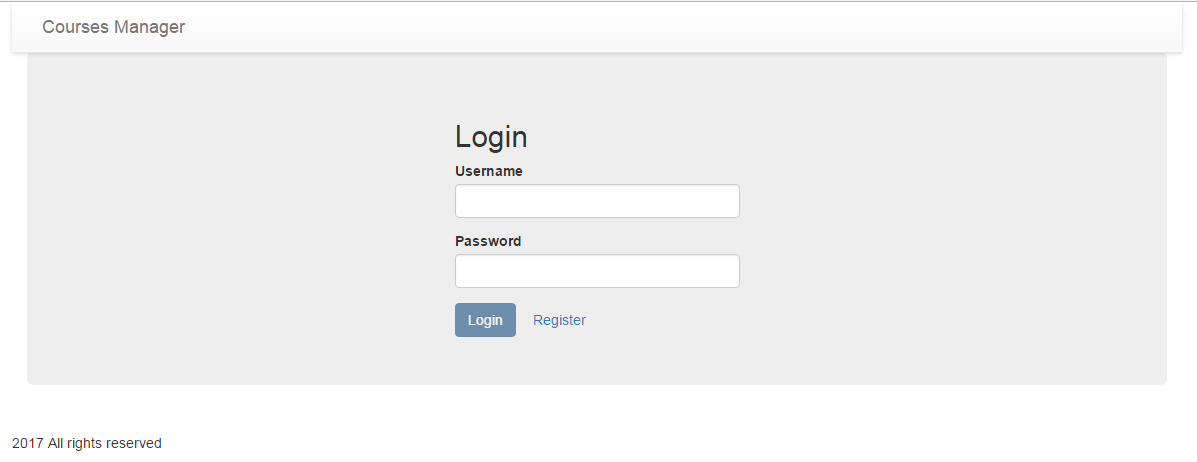
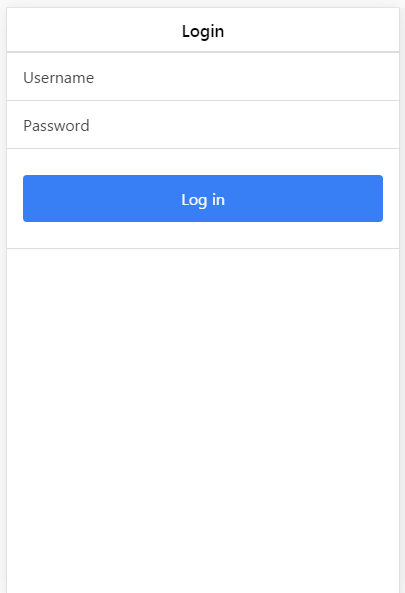
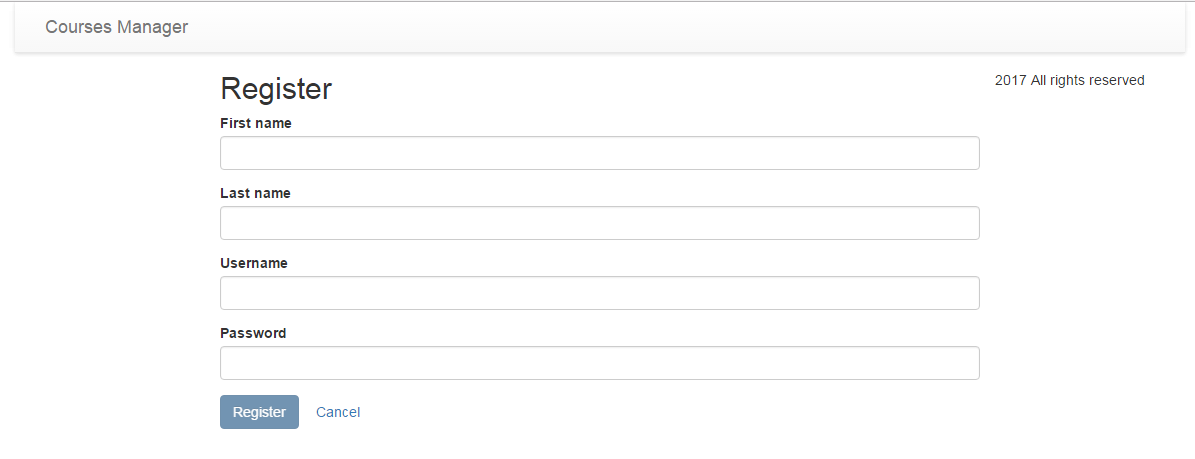
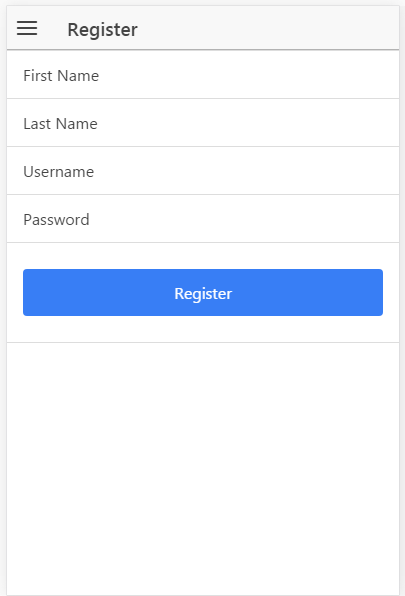
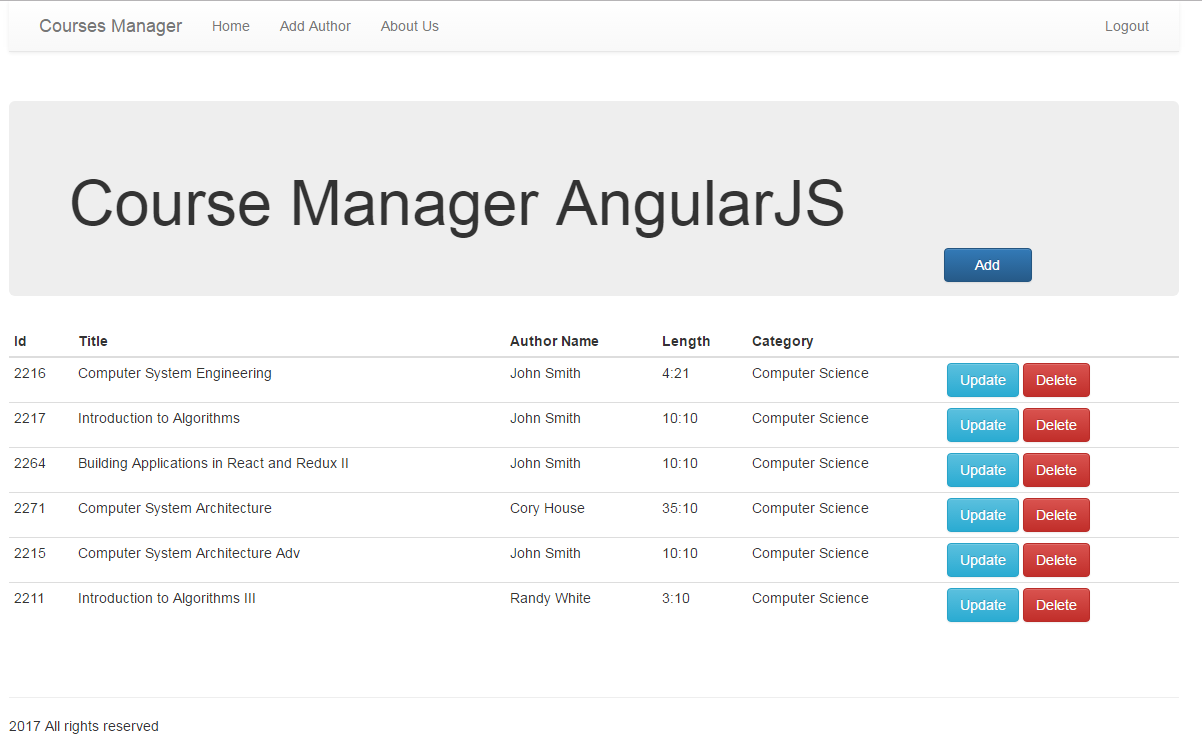
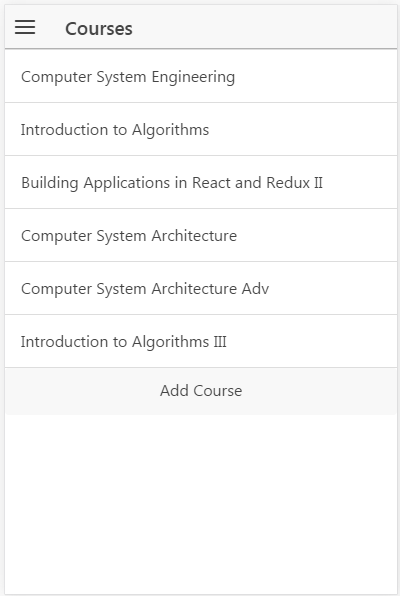
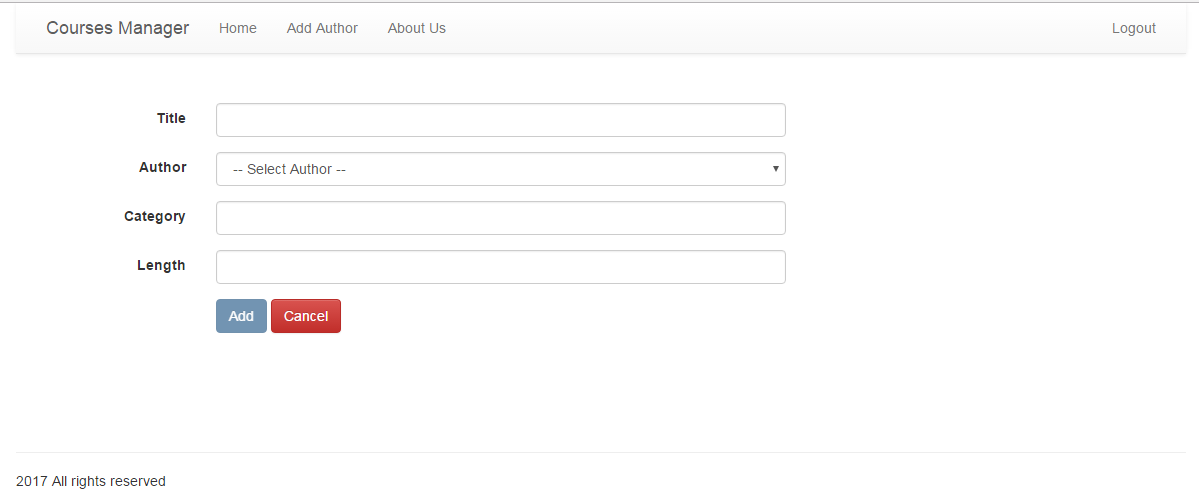
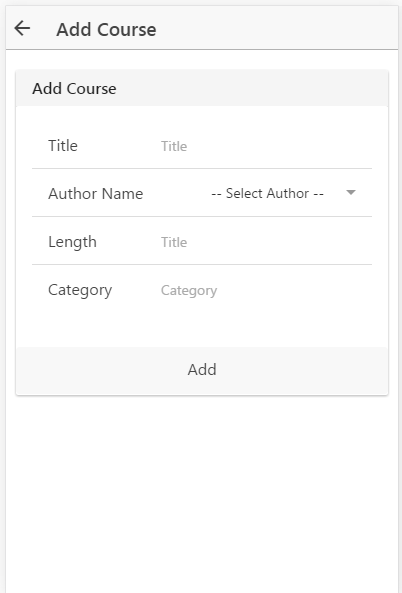
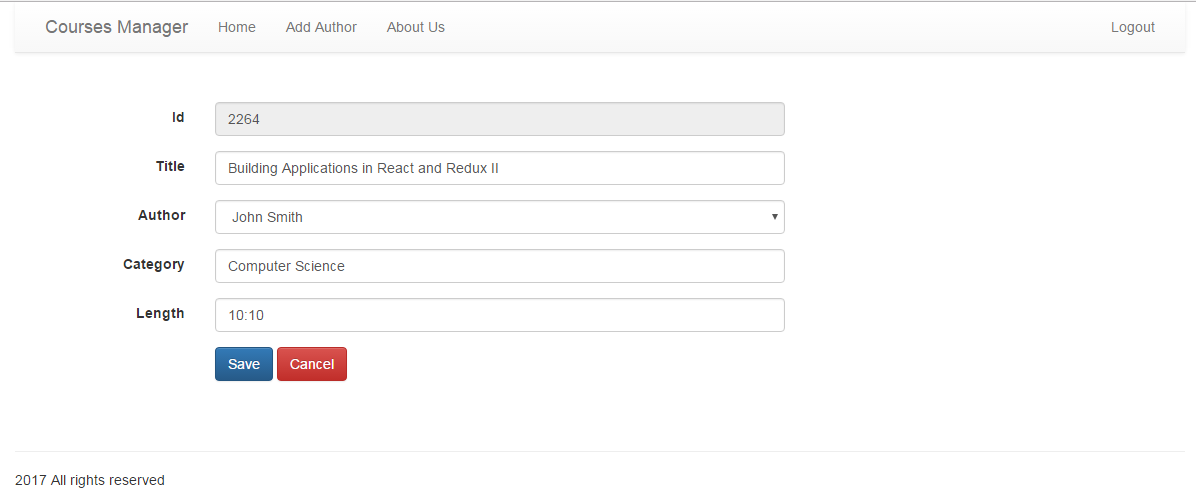
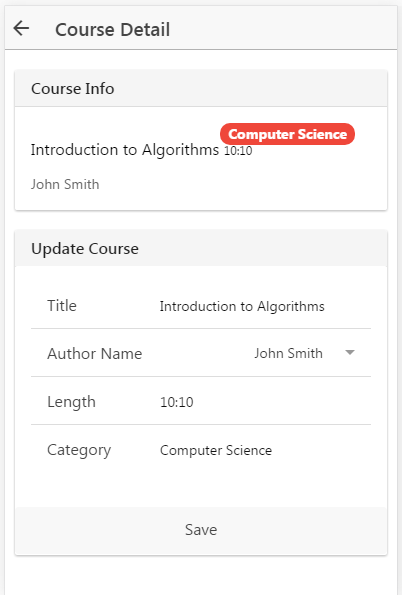
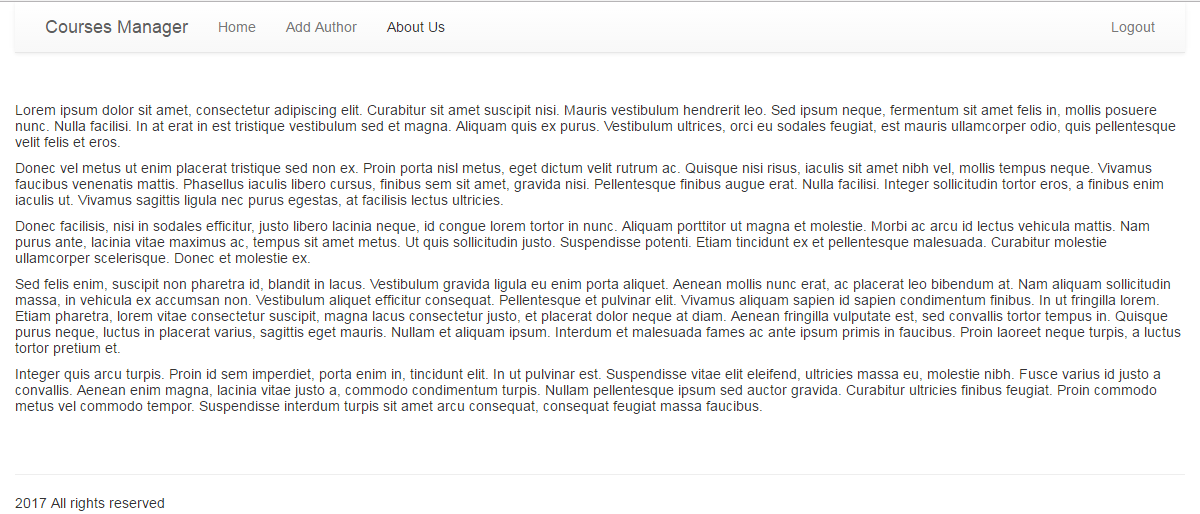
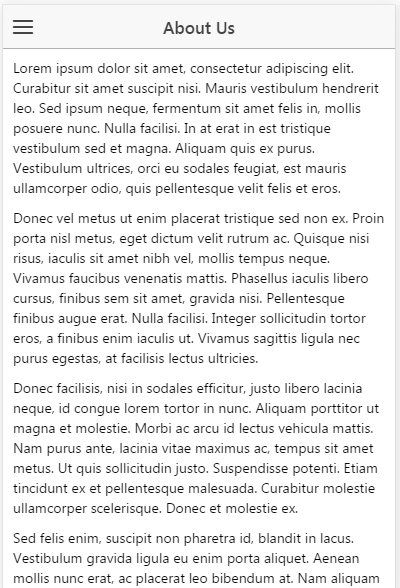
Initially, I only designed the different views and did the routing and templating so I knew I can navigate throughout the application. Then we applied the top-down approach so we developed a simple interface that could interact with the controllers. When the interaction between the views and the controllers was ready, then I focused on the service and the back-end interaction. Finally, it was just matter of make the UI more user friendly and finishing details that were lingering around.

On the other hand, not every little piece of the requirements was covered in my design. We specifically had one gap in order to show the authors. Since we wanted to show the authors in a drop-down menu, instead of a free text box, then we needed to include a section for adding new authors to the system, this will be shared across users.

In terms of the modules and libraries used are as follows:

* Express: to generate the base nodeJS structure and to serve as the backend of the project. Super easy way to start the backend skeleton of your app.
* MongoJS: a nodejs driver for mongodb. This library makes the use to connect the backend with the DB really easy. I used this instead of Mongoose because the underlying structure was super simple and did not need to create complex queries.
* Open: to facilitate development I used open, a library that will open the browser when the backend is build and ready to be use. If not, every time you want to start the server, and then you need to manually go to the browser and run the app.
* Path: to also facilitate development and get a copy of the nodeJS registry path. This is part of the Express boilerplate.
* Morgan: to have a HTTP request logger middleware for nodeJS. This is part of the Express boilerplate.
* Cookie-parser: a middleware to parse cookie headers. This is part of the Express boilerplate.
* Body-parser: a middleware to parse incoming request’s bodies.
* Bootstrap: a library to simplify front end development. It helps a lot because it will be matter of just inserting the corresponding CSS class name in you HTML tag.
* Angular: this will be the main module for the frontend in the web project; in here you can make use of the Angular framework.
* Angular-resource: this library is the main piece that interacts with a RESTful API via using $resource provided by ngResource. There are other classes that we can use for this like $httpProvider. I choose this because it is more suitable because of our RESTful API.
* Angular-ui-router: this is the routing module for AngularJS. I used this instead of ngRouter because ui-router provides more features, so basically is ngRouter with Steroids. With ui-router I have more control and it will more scalable for future iterations.
* CordovaPlugins: I used a variety of Cordova plugins which are needed for the Ionic development. I especially used cordova-plugin-whitelist, cordova-plugin-console, cordova-plugin-statusbar, cordova-plugin-device, cordova-plugin-splashscreen, ionic-plugin-keyboard.
* Ionic: I used this the same way I used Angular for the web project, but in the mobile project. There are other choices like react-native, but since we were learning Ionic I choose this framework.

Below are screenshots of the application in its current state for both the web and mobile projects.

* Login  
   
* Register  
    
  
* Courses  
    
  
* Add new course  
    
  
* Update a Course  
    
  
* About Us  
    
  

3. Conclusions

In this project I put in practice all the knowledge acquired on the specialization, that is why I wanted to do a simple proof of concept that could help me assemble all the different pieces in one consolidated project. Moreover, I created a project that can be a CRUD boilerplate for future projects I have in mind. More specifically this project helped me create from scratch all the different layers of the MEAN stack.

I personally believe the project was successfully implemented since it provides all the proposed features from the initial documents. Of course not everything was as I expected and planed. I profoundly believe taking your time to design the app pays off, but there are always gaps that we need to fill on the implementation. For me the major gap I got to fill, as I mentioned before, was the inexistence of a feature that adds and maintain authors. I came up with a patch to create a form that adds an author, but that is not how I envision this feature for a phase 2. I think that the authors needs to have their own section, similar to the courses where you actually see and maintain the authors, this could be a feature for an Admin, inexistent in this first phase.

In general, I am happy with the final product since helped me to go through out every detail and it was successfully implemented. Hopefully there is a phase 2 where I can re-iterate over the Course Manager, where there still is room for improvement.

4. References

Angular $http vs service vs ngResource (Xavi Montero) <http://stackoverflow.com/questions/17667455/angular-http-vs-service-vs-ngresource>

Mongoose vs mongojs on Node.js (Luis Elizondo): <http://www.luiselizondo.net/2012-08-11/mongoose-vs-mongojs-on-node-js/>

What is the difference between angular-route and angular-ui-router? (Wilgert) <http://stackoverflow.com/questions/21023763/what-is-the-difference-between-angular-route-and-angular-ui-router>

npmJS <https://www.npmjs.com/>