|  |
| --- |
| 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

Details of how you converted from design to the actual realization of your project in terms of implementing the code.

Any choices that you made, and any modifications that you made to the design, in response to difficulties that you might have encountered while implementing the project.

A brief discussion of various modules and libraries that you used in implementing your project. In particular highlight the reasons for your choices briefly.

Include a few screen shots of your web application and hybrid mobile app in the report

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.

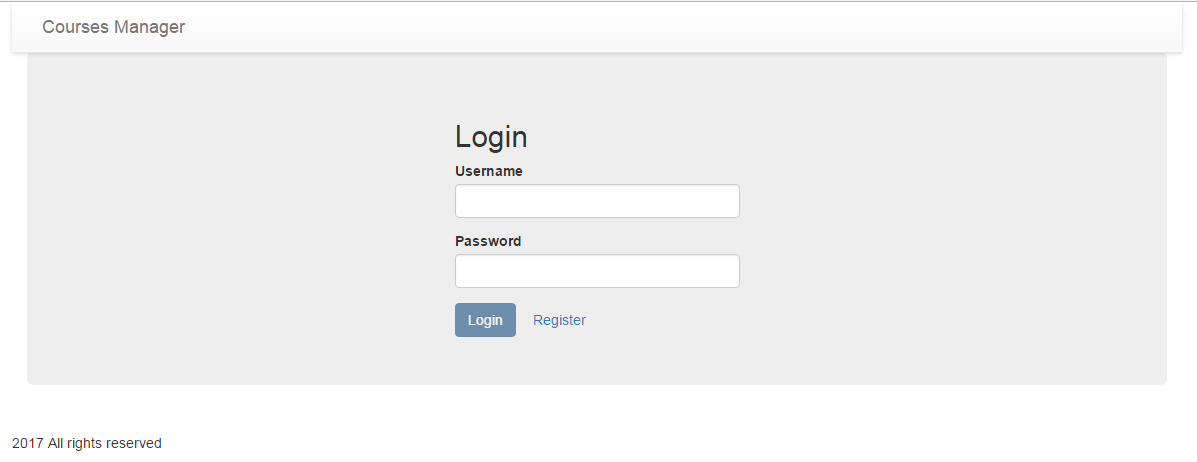
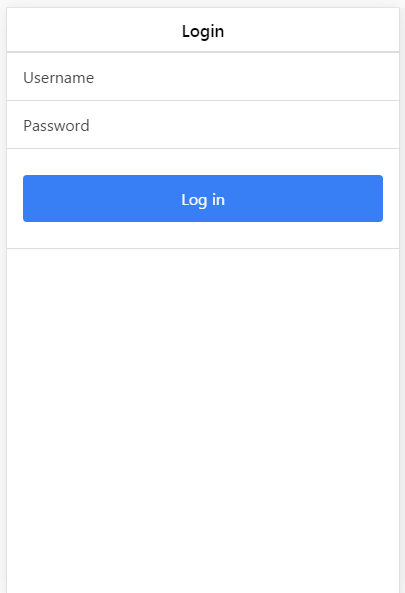
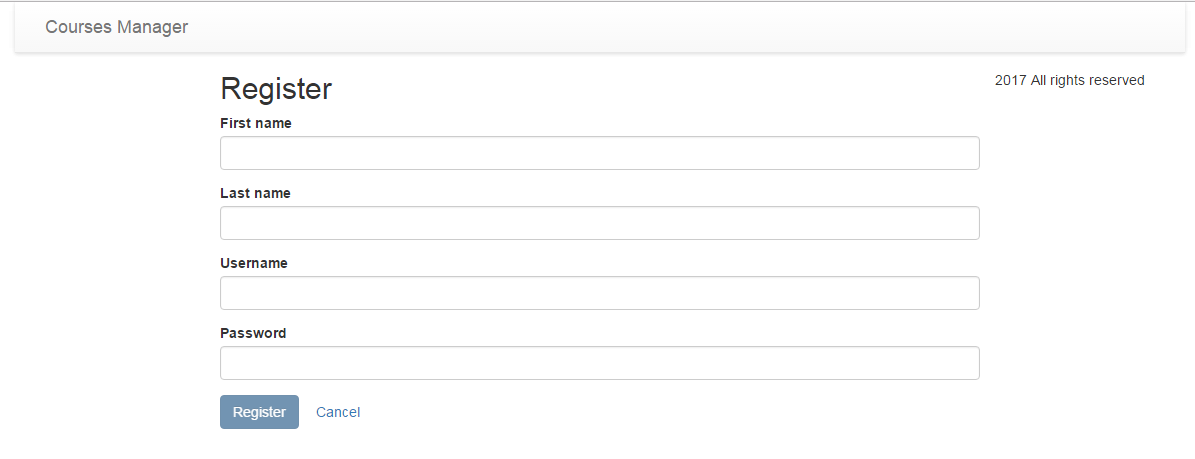
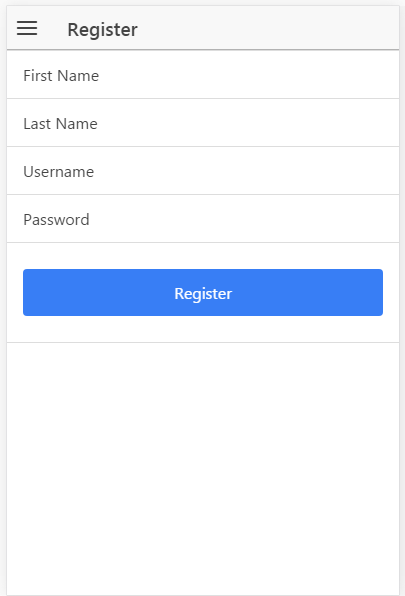
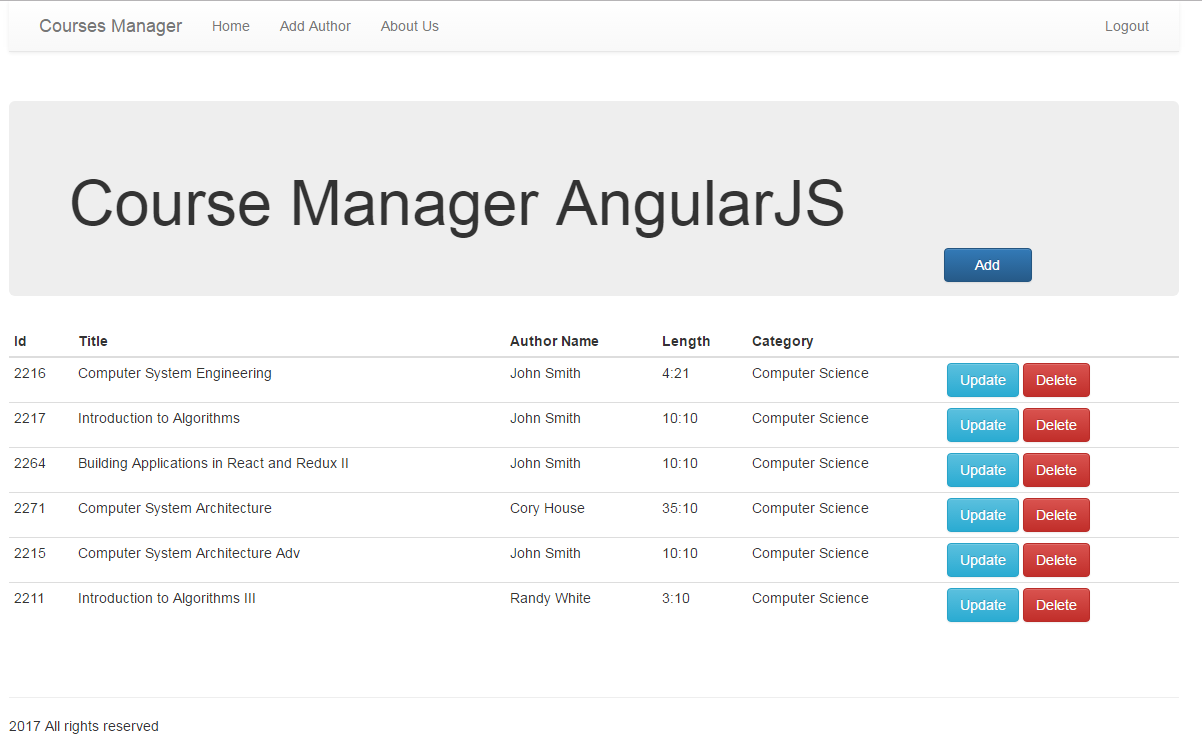
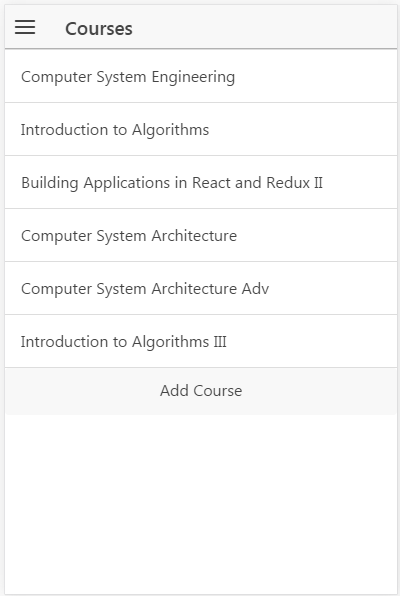
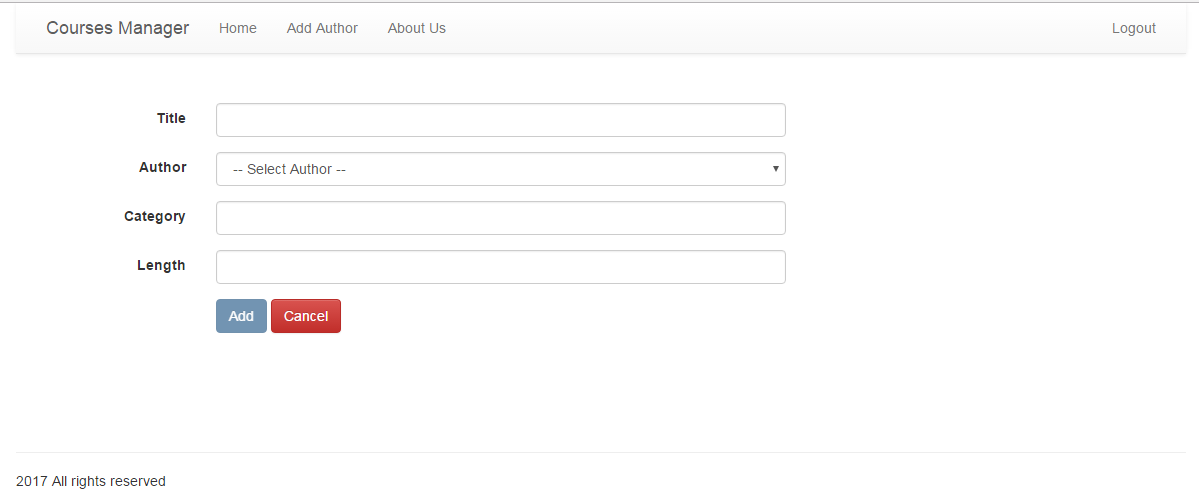
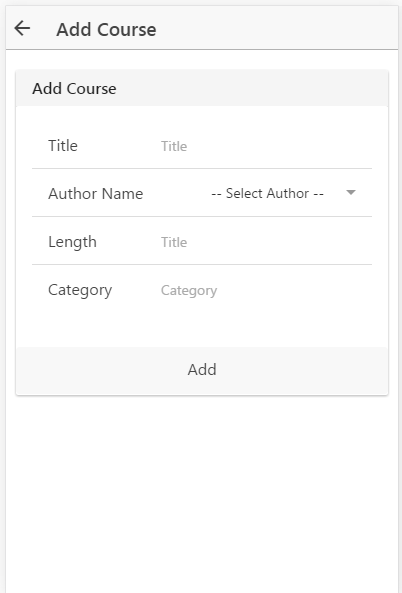
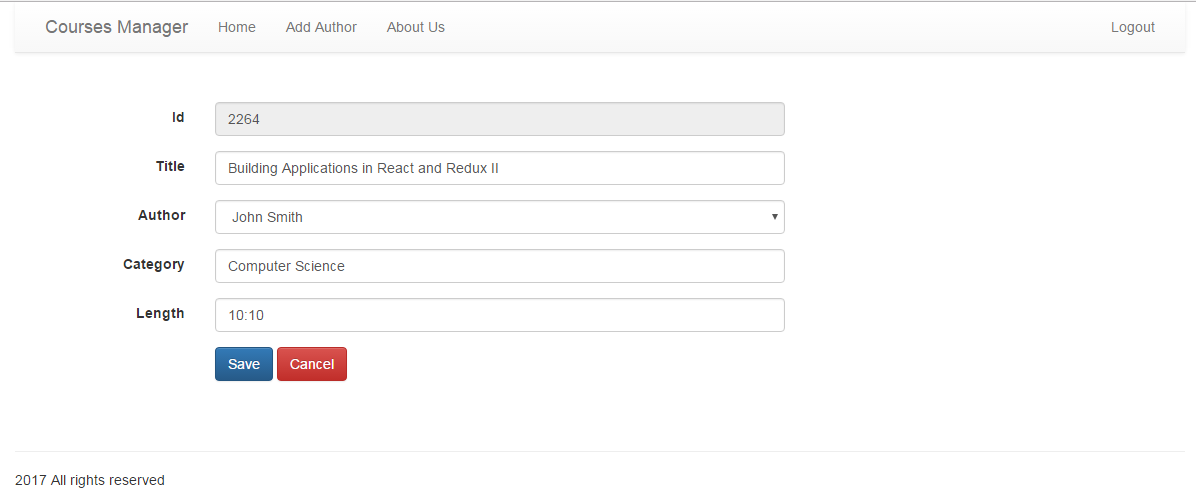
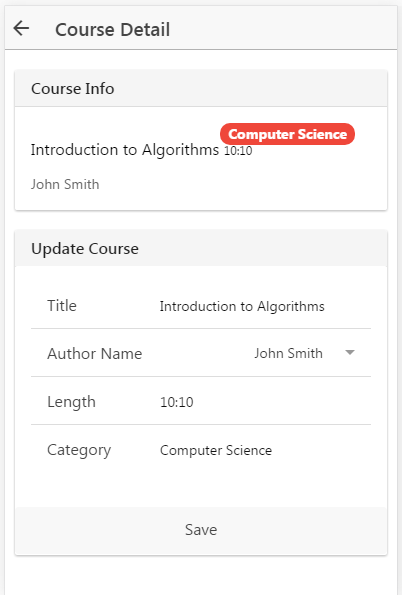
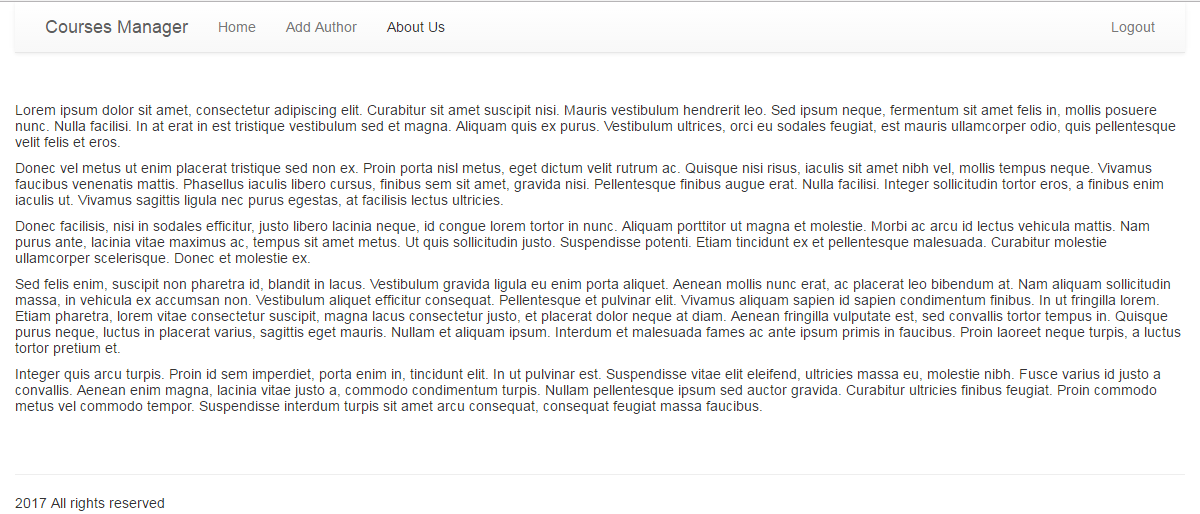
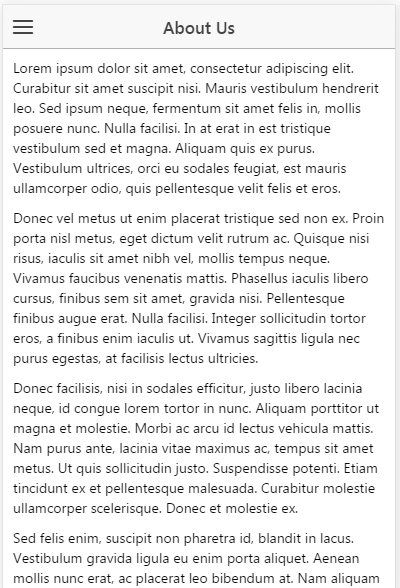
Initially, I only designed the different views and did the routing and templating so I knew I can navigate throughout the application. Then we applied the top-down approach so we developed a simple interface that could interact with the controllers. When the interaction between the views and the controllers 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 finish details that were lingering.

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.
* Morgan: to have a HTTP request logger middleware for nodeJS.
* Cookie-parser: a middleware to parse cookie headers.
* Body-parser: a middleware to parse incoming request’s bodies.
* Bootstrap: a library to simplify front end development
* 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 Ionic

Below are screenshots of the application

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

3. Conclusions

* Briefly state what results you obtained from your project.
* Discuss any features and shortcomings of the project.
* Discuss any choices that you might have made differently, in hindsight after completing the project.

In this project I want to put in practice all the knowledge acquired on the specialization, that is why I want to do a simple proof of concept that could help me assemble all the different pieces in one. Moreover, I would like to create a project that can be a CRUD boilerplate for other projects I have in mind. More specifically this project will help me create from scratch all the different layers of the program.

In this document, we have gone through out all the different layers of the design and architecture of the Course Manager. First, on how the REST API end points are going to look like. Second, how the projects’ file structure are organized, for both web and mobile projects, and know which design pattern we are going to use in the project, which is MVC. Third, we designed the database in a NoSQL environment and give more structure to the documents and collections. Finally, we discussed on how each API call is going to be and how the message will look like in terms of the request URL, Method and Payload.

4. References

Wireframe diagrams done with <https://www.gliffy.com>

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/>