EMumba Pvt. Ltd.

Training Manual

A road to glory of Javascript Development

Ahmed Waleed
01/10/2018

eMumba Training Manual

Targeted Technologies: React and Node JS

We recognize the awesomeness of Javascript. Apart from it being the universal language for web we are huge fan of ReactJS on client side and Node JS environment for server side.

“Why Javascript stack?”

Javascript is a sane and automatic choice for client applications today such as web, mobile and even desktop applications. To make things even more interesting JS has become a sane choice on server as well till you have a solid reason to why not and even at that point, i would suggest to doubt the viability of that reason.

If you want to look more into the “Why’s?”. Mr Google says “Hi!” to you and please keep asking Google till you believe that Javascript enables you to communicate with supernatural and extraterrestrial forces.

Still not satisfied? Please doubt your research skills and as a punishment you have permission to disturb people around you on your own insurance and find out why they think it's great!

Preface

You are a next level programmer, problem solver and algorithm designer. We acknowledge and respect that and it's why you are here. However, learning the tools to communicate and express awesomeness inside you, should be plain, simple and fast.

The purpose of this manual is not to give you difficult coding problems or prepare you for a hackathon but to get you started with the technologies we work on and make sure you know how to change color of that button[Humour intended :)]. So, if you find the content / exercises / problems simple then please know this we don't doubt your IQ and intelligence. You will get chance to show your kung fu skills on real life projects and problems so save your breath.

However, if you have had enough hands-on practice of any topic beforehand then talk with your mentor and move on to advanced stuff. You don’t have to be a “Lakeer ka faqeer” by following everything written here but when you say you know some topic we assume your understanding lies above 8/10.

This document will tell what to learn, from where to learn and the practice assignments to do after you learn. The effort will be your own, what this document cannot do is, sit and code in your seat.

P.S. Motive of this manual is not to make you obey but to help you learn and if you think you know another effective procedure or a missing resource then feel free to reach out at (ahmed.waleed@emumba.com). We appreciate feedback and suggestions!

Coding Prerequisites:

How do you become a genius programmer?

We would love to say a few words on this subject however there is nothing we can say that could come close to this masterpiece video from two Googleans. It's a must watch and please do share it with your programmer friends as well!

Maintaining Standards of Writing Code:

1. Check out these coding guidelines, everything in it mayn't be relevant for you at the start, but that's okay you can study it as it becomes relevant through the manual
2. Before any formal review, you should review your own code per the guidelines above and believe me, you will find lots of improvements plus this will make your mentor happy

How to share code with your Mentor?

It's been long since Git won the Version Control systems war and Github became the preferred choice for open source repositories and code.

If you have no idea what Git and Github, see this and this or if you want more depth to it or need to solidify your understanding, you can watch this video series.

1. Create a new repository for assignment or task
2. Push the code
3. Share the link to your mentor

Guidelines for Code Improvements

1. Add a new file named as Code Review (Markdown format preferred) and add the feedback / improvements suggested by mentor
2. Create a new branch with appropriate name(it's a test as well :)) and push the code to the remote branch on Github

Editor Wars

In case, you are not aware of any modern text editors for WEB technologies (Yes! we see you vim users!), let us name a few and you can choose whatever sounds cool to you because they are all quite comparable.

The recommendation, however, is to choose an electron based editor so you can tweak it the way you want using technologies you work in i.e. HTML/CSS or JS.

- **Recommended:** Microsoft always impresses with their tools for developers and this time its Electron based VS Code which is surprisingly open source, free to use and with lot of extensions. Plus i believe VS code can be declared as eMumba's official Editor because everyone i know uses it.
- Atom is an Electron based open source Editor from GitHub with a lot of users and extensions. It's a sound choice if you hate microsoft for some reason.
- Then, there is the Sublime Text which revolutionized the modern code editors. It is built on python so it is faster than the above two, with of course a lot of extensions. Downside to it is that it is not open source although it is free to use if you are okay with it bugging you sometimes to pay for it.
- There are many others but the above three are the trending ones on tech scene and Cross Platform as well. ***Best editor*** is the one you are comfortable working in! You will spend major part of your day looking at your editor so make sure you like living in it!

If you are curious to know, what editor is author using? Author has experienced all of the above editors and a lot others including Vim and Emacs. However, for web technologies author is currently sticking to open source awesomeness from Microsoft, VS Code with many extensions installed.

Week 1, let's get started!

Goals of the Week:

1. Reintroduction to HTML and CSS if required
2. Modern CSS
3. Reintroduction to Javascript if required
4. Getting hang of the basic concepts of Javascript
5. Learning the syntax sugar of the language along with understanding the ES6 / newer developments on top of basic JS

Day 1 and Day 2: Coming Back to HTML and CSS 101

This is where the web started so this is where we start. There is no shame in taking the 101 courses as long as you get better however what would make you look bad is if you lack in these basic concepts at some point after this training.

Register your account on Freecodecamp which is an interactive learning platform.

Suggested Topics / Learning Pathways to cover from the freecodecamp curriculum:

1. Basic HTML and HTML5
2. Basic CSS
3. CSS FlexBox
4. CSS Grid

You can skip through the sections or cover selectives depending on your experience. If you are out of touch skipping through may hurt you while working on the upcoming Tasks or real life projects. Plus somewhere down the road you can complete other sections and earn the certificate. It has good reputation internationally.

Additional Resources:

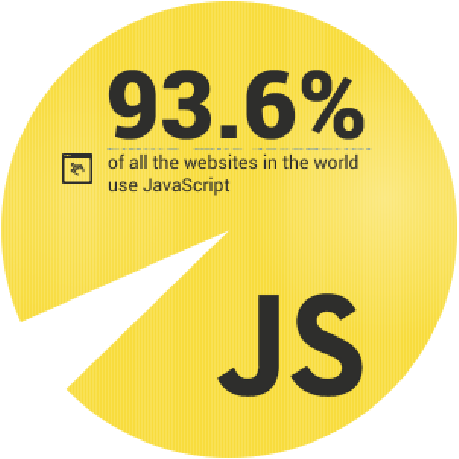
You are not supposed to remember every tag or property. And often you will find yourself searching for a specific tag or styling property on Google so it helps saving time and avoiding extra keystrokes if you have a reference cheat sheet on your PC.

- HTML Cheat Sheet
- CSS Cheat Sheet
- Need a filler text? Visit Lorem Ipsum.

Day 3 Responsive Design Task Due

Ask your mentor to share Clipmine assignment with you. Your task is to design the Responsive webpage using the techniques you learned above.

Day 4 and 5: Itching to Code? You better be!



Get ready for a complete hands-on programming day. You will learn the JavaScript by doing instead of going through boring books or tutorials. You will be working through the Freecodecamp exercises interactively again.

Topics to cover from Javascript Algorithms and Data structures certification portion of curriculum:

1. Basic Javascript
2. ES6
3. Debugging
4. Basic Data Structures
5. Basic Algorithm scripting

What this will do is help you get started with syntax while without being hard on your cognitive memory.

Task Due:

Do you like to read books? Regardless of your answer, you will be creating a mini library management system by only using HTML, CSS, and JS. Your challenge is not only to create the application but also to showcase the concepts you learned above:

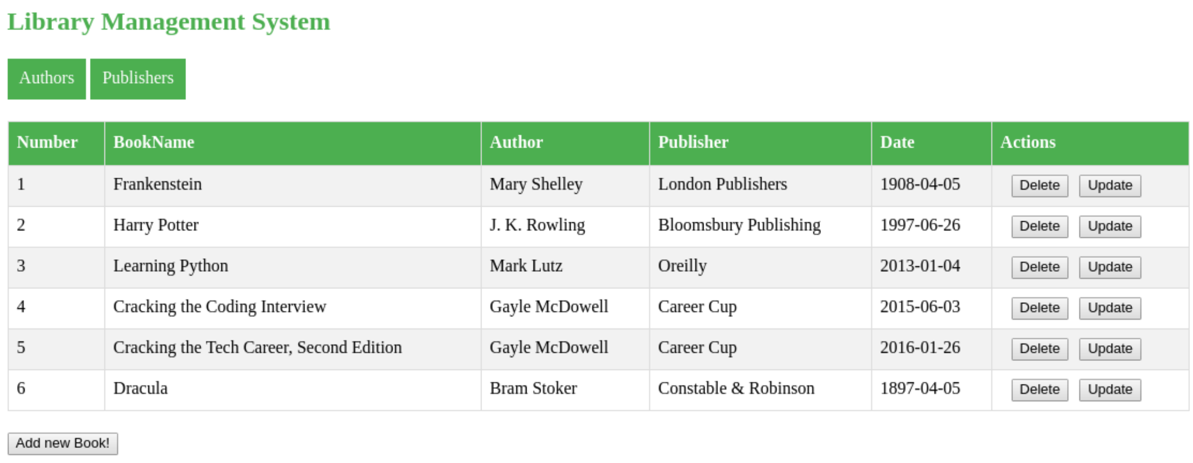
Books have the following properties:

1. Book Name
2. Author
3. Publisher
4. Date published

Requirements:

- List all the books.
- Books can be updated or deleted.
- List all the Authors on a different page along with a *number of books* written by them.
- List Publishers on a different page along with a *number of books* they published.
- Both Authors and Publishers can be deleted which in turn would also delete corresponding books.
- All the data should be persisted in local storage.
- Validation checks especially on duplicate entries while adding or updating.
- Usage of JQuery or AJAX is strictly prohibited.
- We are expecting you to write quality code!

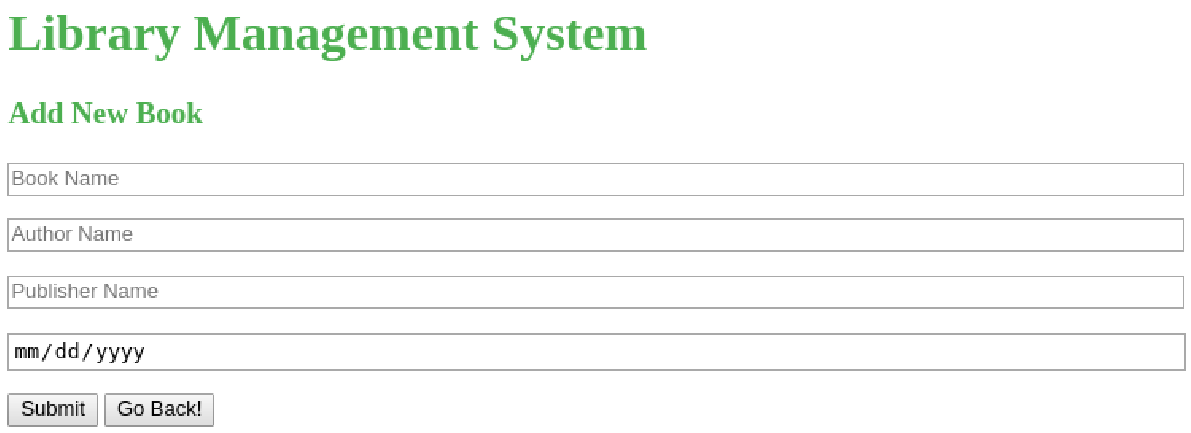
For more information have a look at screenshots below. You are not required to follow the visuals or styling. You can be as creative as you want.



Library Management System

Number	BookName	Author	Publisher	Date	Actions
1	Frankenstein	Mary Shelley	London Publishers	1908-04-05	Delete Update
2	Harry Potter	J. K. Rowling	Bloomsbury Publishing	1997-06-26	Delete Update
3	Learning Python	Mark Lutz	Oreilly	2013-01-04	Delete Update
4	Cracking the Coding Interview	Gayle McDowell	Career Cup	2015-06-03	Delete Update
5	Cracking the Tech Career, Second Edition	Gayle McDowell	Career Cup	2016-01-26	Delete Update
6	Dracula	Bram Stoker	Constable & Robinson	1897-04-05	Delete Update

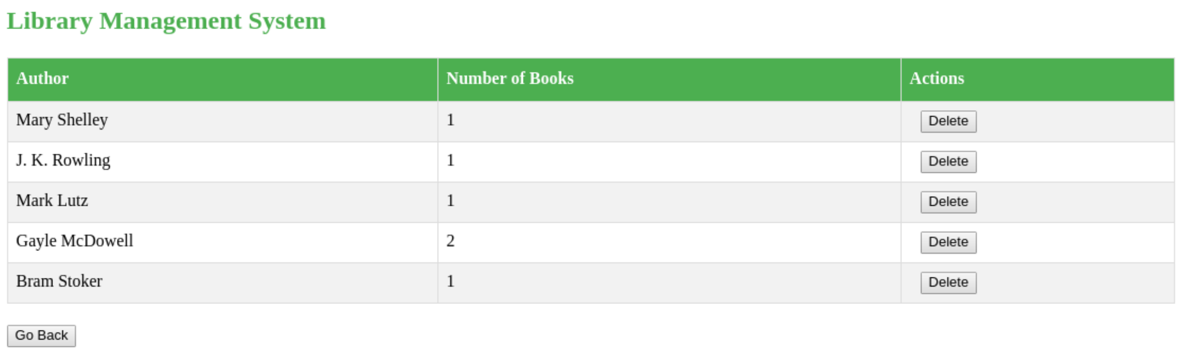
Figure 5 Index Page



Library Management System

Add New Book

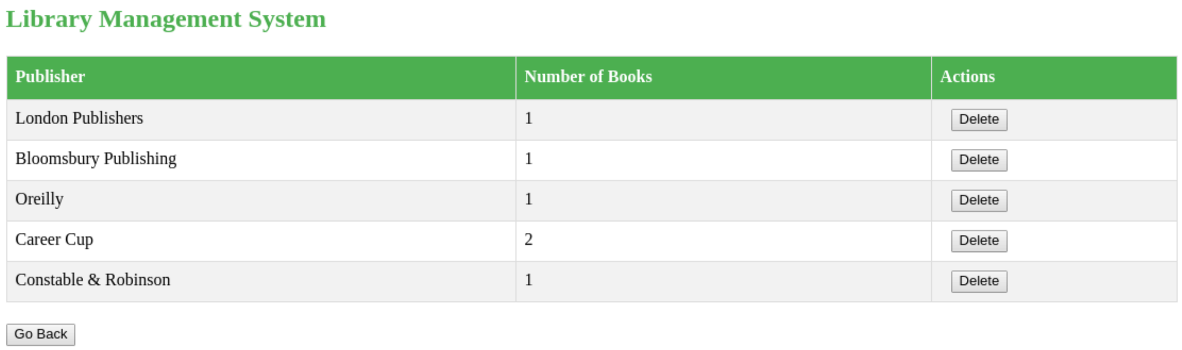
Figure 6 Add Book



Library Management System

Author	Number of Books	Actions
Mary Shelley	1	Delete
J. K. Rowling	1	Delete
Mark Lutz	1	Delete
Gayle McDowell	2	Delete
Bram Stoker	1	Delete

Figure 7 Authors Page



Library Management System

Publisher	Number of Books	Actions
London Publishers	1	Delete
Bloomsbury Publishing	1	Delete
Oreilly	1	Delete
Career Cup	2	Delete
Constable & Robinson	1	Delete

Figure 8 Publishers Page

Code Review:

Once done with the assignment, reach out your mentor and do a code review session for the tasks you did. Make improvements per the guidelines and update the code in repository.

Week 2, Hope you had a nice warm up!

Goals of the Week:

Learning the JS you are actually going to use is a lot different than what we just learned its because of how language keeps evolving over the time although the basics you learned remain the same and you can fallback to them any time but that's not fashionable.

You will be programming in multiple paradigms and using different semantics which can be overwhelming at the start. Our goal is to raise your bar to the point where you are excited to explore more instead of being overwhelmed.

1. Learn Functional Programming paradigm
2. Code practice in Functional style
3. Learn ReactJS

Day 1: Writing pure code is more important than eating pure

Don't worry if you don't understand the heading yet, you will vouch on this by the end of the day. If not don't say we didn't warn:

Again, we will be utilizing the Freecodecamp to start with Functional Programming. You need to complete the **Functional Programming section.**

Task Due:

1. Task 1: Telephone number validator

2. Task 2: Caesars Cipher

Getting the correct output via Functional Programming practices you learned above is the goal.

Code Review:

Once done with the assignment, reach out your mentor and do a code review session for the tasks you just did. Make improvements per the guidelines and update the code in repository.

Day 2: ReactJS, What the world is mad about!

ReactJS has revolutionized the way web client applications are developed and created a whole new ecosystem of tools, frameworks and transpilers/compilers on top of Javascript. We start by learning the basics of react via react section on freeCodeCamp.

Goal of the day:

Understand basics of react via completing the react section above.

Day 3: React JS, Getting started on local environment!

You have been writing Code in the browser till now but how do you start in the local environment? Follow these instructions from official react docs to get started in the local dev environment.

If you feel you still need to solidify your react concepts, going through the rest of tutorial will only help plus you will be making a cool game via react :).

Goal of the day:

1. Start with react on local environment
2. Learn how to think in React
3. Learn how to handle the Asynchronous stuff

How to think in component based frameworks such as React?

These are some highly recommended documents from React JS documentation which will help you in understanding and working in react better:

1. Lifting state up
2. Composition vs Inheritance
3. Thinking in React

Handling the Asynchronous nature of Javascript:

1. The first step is to understand the problem very well, this guide from pluralsight does a fine job of laying out the problems that come with Async nature and presenting you with basic solutions.
2. To understand Promises better and how they work, see this video.
3. Trust me watching this guy on Monday morning does make you feel better :). For some people error handling is the most boring thing but for our guy its not. We really think error handling

ability differentiates between real programmers and script kiddos. Watch Error Handling with Promises.

4. How to fetch data using fetch API. This is the recommended way of doing API calls.

Day 4 and 5: Let’s see what you got!

Task Due:

Details and submission guidelines

Done with the Task?

It's time to reiterate the lessons learned in “Thinking in React” lesson and make sure that you did it way the it should be done.

Code Review:

Once done with the assignment, reach out your mentor and do a code review session for the tasks you just did. Make improvements per the guidelines and update the code in repository.

Week 3, let's get started!

Goals of the Week:

1. State management and why its an important and hard problem to solve
2. How Redux fits in the puzzle
3. Learning Redux
4. Static Type Systems in JS
5. Utility Libraries

Day 1: State Management and Redux

Key to solving any problem is in understanding the problem to its very core. Previous Assignment you did should have given you some context of what is the problem of State Management. If you aren't able to figure out the problem yet, no worries, let's explore the problem and possible solutions.

1. State Management and solutions
2. How does Redux work?
3. Read how to use it and the advanced concept of Redux Thunk via official documentation. It's very well written and you will know that once you read it.

Task Due:

You will have to implement state management via Redux in the assignment you did using React before this. To start working on the task, create a new branch in your Github repo and push it to the Github. This is to ensure we can do comparisons and preserve your previous work.

New Requirement: Add a spinner / loading animation where appropriate.

Day 2: Code...

Code Review:

Once done with the assignment, reach out your mentor and do a code review session for the tasks you just did. Make improvements per the guidelines and update the code in repository.

Day 3: Types and Utility Library

1. Learning basics of Static Types and why use them in an artificial way? This 4 part series will make sure that you are properly introduced.
2. To get started with Flow types, consult this link. For any advanced learning, i would suggest to dig in to official documentation. Type system is a world on its own
3. How to integrate Types in React Components? See official documentation.

Task Due:

Lets identify the flow of data in the Application you worked on previously and add types to your application..

4. Introduction and how to think in Ramda(Functional Utility Library). This 3 part series of articles will help you learn it.

Task Due:

Replace the data manipulation functions you are using in your assignment with corresponding functions from Ramda.

5. Ramda is also a world on its own, you don't need to remember every function name however its documentation is a very handy reference to have opened as a pinned tab in your browser

Day 4 and 5:

Code Review:

Once done with the assignment, reach out your mentor and do a code review session for the tasks you just did. Make improvements per the guidelines and update the code in repository.

Slack Time, What to do?

Any Technical Debt you want to clear? Any topic you think is left unclear or any thing made you intrigued you can dig deeper or perhaps you want to polish your table tennis skills? This is the time to do that!

However you need to report your mentor on how you are going to spend the slack time.

Week 4, Evaluation…