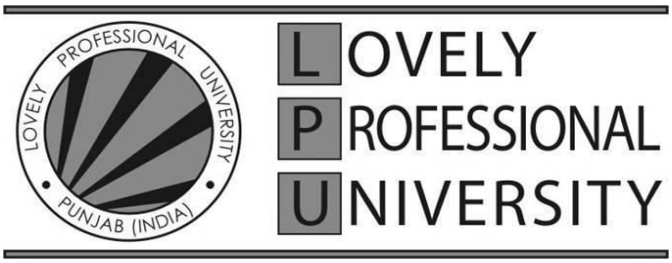


SIX WEEKS SUMMER TRAINING REPORT

on

REACT.js

Submitted by

Nabin Kumar Adhikari

Registration No. : 11617707

Programme Name: B.Tech.(Computer Science and Engineering)

Under the Guidance of

Simplilearn

School of Computer Science & Engineering
Lovely Professional University, Phagwara

(June-July, 2017)

DECLARATION

I hereby declare that I have completed my six weeks summer training at React.js from 2018/05/27 to 2018/06/16 under the guidance of Simplilearn. I declare that I have worked with full dedication during these six weeks of training and my learning outcomes fulfill the requirements of training for the award of degree of Completion Of Training Program on React.js, Lovely Professional University, Phagwara.

Name of Student : Nabin Kumar Adhikari

Registration no. : 11617707

Date: _____

Acknowledgements

The summer training opportunity I had with Simplilearn, which is an MOOCS platform proved to be a great chance for learning and an aspiring experience.This acknowledgement addresses the wonderful opportunity provided to the students by the university with the united efforts of various online course study platforms like Simplilearn,the one I selected to do my study on.The university significantly put the summer break on a great usage with the students being guided to utilize their time to learn and progress.Student life is great for learning new things and prospering own knowledge.The university provided a needed push for the students to employ their time in learning and prospering.

Simplilearn is one of the top online study platforms with certification.It has vast course contents. The course I choose to study was React.js, which is indeed very interesting and has great possibilities in the field of web content development. Simplilearn provides a dedicated study portal for the users called as LMS. It also provides the users with various features to aid the users for their convenient learning.This has definitely helped me a lot to study in a managed way.

Learning from only one path is also not sufficient.So, I would also like to mention various sites that helped me to study more thoroughly. Definitely, Google comes at the top, along with which various sites like Stackoverflow, FreeCodeCamp, Quora provided various informations when I needed and answers when I had confusions. I did my coding practice mostly on the coding environments provided by Codepen and Scrimba. These coding environments were great to practice and also to just play around with. The react web page itself had various tutorials and documentations on react. Those were also very helpful with the learning process.

Certificate Of Completion of Training Program

Table of Contents

INTRODUCTION

The online world is growing faster than ever. More than half of the population on earth has access to internet. With internet one can reach out to countless online resources, services and facilities available on the internet. It has a huge range of interlinked hypertext documents and applications of World Wide Web. Although World Wide Web is only a part of what internet offers, it is so important that the world wide web is often used interchangeably with internet. Web is a collection of various web pages linked through URLs (Uniform Resource Locators) and hyperlinks. These web pages are developed through various coding and scripting languages. One of the most important language used in almost all web pages available today is Javascript. Javascript is an core technology used to develop web pages. Javascript enables developers to build interactive web pages by enabling developers to control the behaviour of the web page and thus is very widely used. It has various libraries which are pre written javascripts which makes the development of javascript based applications or other web technologies much easier. Among the various javascript libraries, React is a graphical user interface related library.

Developed by Jordan Walke, a software engineer at Facebook, React is described to be a declarative, efficient and flexible javascript library used to build user interfaces. Maintained and supported by Facebook and community of developers, React was first deployed in news feed of Facebook in 2011.It was later also used in Instagram in 2012 and open sourced at JSconf US in May 2013. React is used in some of the major websites like Facebook, Netflix, Instagram, Yahoo Mail, Dropbox, WhatsApp and many more either partially or completely due to the flexibility, simplicity, efficiency, speed and many features React offers. It also has a great reusability feature due to the component based design it has adopted. React is very useful for creating especially single page interactive web apps which deals with real time exchange and updation of data. Often other javascript libraries are not easy to be used for real time data management but react does it with great simplicity.Today, React is very popular, so popular that React is dominant on the other javascript front end libraries in terms of usability and popularity. Hence, today it's almost like a

industry standard to use React on front-end development as it is already adopted by many top players in the internet industry.

This report mainly focuses on React, its features, usability, practicality,implementation and future prospects I learned from my time on studying React especially on Simplilearn since I did my summer training on it. Besides Simplilearn, I used many other websites to learn and write this report. We will discuss React more technically in the next section, Technology Description.Then I will discuss my reason for choosing this technology for my summer training. I learned a lot of things during this summer training which will be discussed in the Learning Outcomes section of this report. After that comes the gantt chart for my training period followed by the bibliography section with all the sources of my studies and information gathering.

Technology Description

React (also known as React.js or ReactJS) is a graphical user interface (GUI) developing javascript library. React is component oriented which allows to create rich and composed components. It then helps a developer to develop a responsive UI by efficiently rendering the components of a react application to update the data among the components whenever it changes. This makes the user experience enriched and extends functionality of the app. React is used mostly to create single page apps with great styling and data management features.

React is simple to use in any web pages. We write our code in javascript or ECMAScript. React uses a concept called JSX (Javascript XML), which is an syntax extension for XML (Extensible

Markup Language) to write code in javascript, which is then pre processed to XML. We can also use React without JSX, but JSX makes React more elegant and easy to use which in turn delivers a high performing application with ease. We can use Babel script or ECMAScript for this. The code we write is then compiled and run. For this, react sets up a virtual DOM (Document Object Model) which is an representation of the documents of HTML or XML in nodes and objects.DOM is an programming interface with object oriented representation of the web page which enables us to access and manipulate the web components with the use of scripting languages like javascript. Virtual DOM is a lightweight abstraction of the actual DOM, which can be updated without having to update the actual DOM of a web page. While the actual DOM may be inefficient during the updation due to the fact that it re-renders all the components even if only one is updated, virtual DOM gains its speed and efficiency as it does not have to re-render all the components but the only one that has been updated. This is done by using a process called reconciliation, that compares the components after the updates to know the updated components which can then be rendered without having to render the whole DOM. This makes React development very easy, fast and efficient even if we have a large number of components and contents in our react script. After the render process in the virtual DOM we can then view our web page in browsers.

Component and Its Rendering in React

React is highly component oriented. We can build or create many components in react to perform various functions or even compose our single component in such a way that it can perform all the functions without loss in any performance, instead making the component more rich and robust. The whole react application also acts as an component in itself, that has various child components which in turn can have their own child components. Components in react lets us look at individual parts of the UI design as an isolated piece which can also be reused and be changed or edited in isolation. Conceptually, the components in react are like functions or methods in various programming and coding languages. A component in react can be created as we create a javascript function as :

```
function Hello (props)
{
      return <h1>Hello World, {props.name}</h1>;
 }
```

Here, a javascript function named ‘Hello’ is created. It passes a single argument called props (properties in react) named ‘name’ and returns a element. These types of components are called functional as they are javascript functions.
Using ES6 we can define a component as we create a class like :

```
class Hello extends React.Component
 {
       render()
       {  return <h1>Hello World, {this.props.name} </h1>; }
  }
```

Here, a ES6 class named ‘Hello’ is created as component in react when it extends the component class from the react core library. It also passes a single argument as ‘name’. The render function returns a react element.

React elements can also be represented as user-defined components :

```
const element = <Hello name=”world” />;
```

Here, the constant named ‘element’ has value ‘world’. This user defined component can be rendered by using ReactDOM like any other component but it requires some component (‘Hello’ in this case) to accept it as a prop i.e it needs to be passed as argument in a component.

A component needs to be rendered in react. Every component has a render function which returns the elements of react. ReactDOM is a rendering library for react which renders elements of react as :

ReactDOM.render (< element />, target);

Here, 'element' may be the component to be rendered and 'target' can be a element like a division in the main or index HTML file.

A HTML document with minimal setup for react can be created as :

```
<doctype HTML!>
    <html>
      <head>
       <script src="https://unpkg.com/react@15/dist/react.min.js"></script>
       <script src="https://unpkg.com/react-dom@15/dist/react-dom.min.js">
       </script>
       <script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js">
       </script>   </head>
      <body>
         <div id="target">  </div>
          <script type="text/babel">                    //we can also import an
                //the react script here                 //javascript script here
          </script>                                     //with react code
       </body>
      </html>
```

Here, a HTML file has imported React, ReactDOM, Babel scripts respectively in the head section of the document. In the body section, we have a division named 'target' which will be the target for ReactDOM to return the rendered elements. Then, we have the script tags, to write the react code within them, with type 'babel'. Instead of doing this, we can just import a javascript file where all of our react code will be. Following is the example of importing a javascript file in a HTML doc :

```
<script src="path_to_file.js"> </script>
```

Here, we have imported a javascript file having our react code. The path to the file is written in the double quotation of 'src'. In the javascript file we have to import 'React' from the 'react' library and 'ReactDOM' from the 'react-dom' library, the later if only if we intend to render our react elements by ReactDOM. A javascript file importing these is shown below :

```
import React from 'react';
import ReactDOM from 'react-dom';
//your react code here
```

This way you would have a complete running react component in your browsers. As mentioned earlier, a react component can have child components too. A example is shown having a child component below :

```
class Hello extends React.Component
{
    render()
    {
        return( <div>
                    <World/>
                    <Play/>
                </div> );
    }
}
```

Here, in the component 'Hello' we have child components 'World' and 'Play'. These components can have their own piece of code completely different from one another. We create these child components like any other component but we do not have to render these child components using ReactDOM. When we render the parent component, the child component is also rendered with it.

That means we can have hundreds of components as childrens of a parent component, but we only have to write the code to render the parent component with ReactDOM.

Props and States in React

Earlier we saw that components can pass an argument called 'Props' i.e. Properties in react. Props in react are the objects having some value. Props cannot be modified by the components. They can be used in data handling purposes.When we extend the Component class while creating any component, we also inherit the props. We can set a value to any prop and use it as :

```
class Parent extends React.Component
{
    const hello = "hello world message" ;   //creating constant 'hello'
    render ()
     {
        return  ( <Child  message={hello}/>   //creating prop message
                );
     }
}
class Child extends React.Component
{
     render ()
     {
        return ( <h1> {this.props.message} </h1>   //accessing prop
                );
     }
}
```

Here, a constant name is 'hello' with value 'hello world message' is created. While returning the rendered element in 'Parent' class we pass the prop 'message' to the class 'Child', we give the

value of the constant ‘hello’ to the prop ‘message’. In this way, we can pass any prop giving them any value. The passed prop is accessed in the ‘Child’ class through ‘this.props.message’, where message is the prop name.

We can pass just any type of value with the help of props. That is why props are very useful in the data management in the react application for a static application i.e. value of a prop remains fixed during the lifetime of a component. But, the web applications of today are dynamic and the data in them can constantly change and in need of constant update. That’s where the concept of states kicks in. State have similar qualities like props but their value can be changed or updated by the components in real time. It is one of the reasons, react applications are dynamic and can change over time. States are also the objects that hold some value very alike props, but value of states can be changed through a component. The use of a state to pass a value is shown below :

```
class Parent extends React.Component
{
    constructor ( )
    {   super () ;
        this.state = {
                        message : “Hello Humans!”
                    };
    }
    render () {
                return ( <h1> {this.state.message} </h1> )
            }
}
```

Here, a class named ‘Parent’ is created with its constructor inside which we have a ‘super’ component too. This ‘super’ component must be everytime in a constructor. Then inside ‘this.state’, we have a state named ‘message’ with value ‘Hello Humans!’. In this way, we can create a state inside a constructor of a component. This state can either be passed to some child

component as we pass props or be accessed to use in the same component. We have accessed a state and its value by the keyword 'this.state.message', message being the name of the state we created in the above example.

State, however can be changed. So, to change a state we can use a keyword 'this.setState'. The example to a changing state is given below :

```
class Parent extends React.Component
{
    constructor ( )
    {   super () ;
        this.state = { message : "I am A Robot !" };
    }
    this.updater = this.updater.bind (this);
    updater(){
            this.setState ( { message : "I am a Human !"} ) ;
        }

    render () {
            return  ( <div>
                <h1> {this.state.message} </h1>
                <button onClick={this.updater}>Click</button>
                    </div> );
        }
}
```

Here, we have a simple demonstration of the change in the state 'message'. When we click the button created, it triggers 'updater', which has the code 'this.setState'. It changes the state 'message' when we click the button.

The state of an object can change during any time of the duration of execution of a react application. When the state of an object changes, react only re-renders the changed objects

effectively. This is done with the help of virtual ReactDOM, which re-renders the react application after the change and then injects the changes in the actual DOM, needing only the changed elements to be re-rendered. This is one of the biggest features of react.This not only saves our time and effort, but also provides our application with fluidity and a very fast performance. This is only a simple example of what changing states can do. We can use this when we have input from user in real time, change in server side, changing components in a intervals and so on. This state change in react is what helps a react application to be dynamic, adaptive, responsive and simply have a great user experience in the react applications.

Event Handling and Routing in React

Event Handling explores the concept of the state and props we discussed above. Events can be triggered through some actions like button clicking, navigation, rollovers and so on. For an event to be triggered and responded, we need some classes to control what exactly triggers the event, what happens after the event is triggered and to manage the data during the event. The piece of code we have written earlier for changing states, also defines event handling. The click on the button was a event which was handled by a function 'updater'. The function 'updater' then changed the state of the object 'message'. Here, we had binded our event trigger with a button clicking and had called a function to manipulate the state of an object. This is a good example on how to trigger and handle events in react applications.

React is usually used for single page applications. To navigate in these application requires the concept of routing. Routing is handled in react with the Routing library which can be imported through the script sourced to :

https://unpkg.com/ui-router-react@0.5.0/_bundles/ui-router-react.js

We also have to import various components for routing from the 'ui-router-react' library like Router, route, IndexRoute, Link e.t.c. We can enable the navigation buttons like a browser in any react applications. We can also have the feature of 'history' in our react applications. For this also,

we have to import 'history' script and library separately from core react library. For using routing, we can use Router tag in the ReactDOM render function as :

```
ReactDOM.render (  <Router history = {hashHistory} >
                        <Route path="path_to_navigate"> </Route>
                        </Router> );
```

Here, have also enabled the history feature through 'hashHistory' object. The path to navigate is written under 'path' in 'Route' tag. Here, we can also render our homepage from where we can route out to other pages. For this we have to add 'component' keyword alongside path, with the component name value which acts as our homepage. We can then add links or routes to other pages or components from the one we rendered our routing tags :

```
< Link to = "path_to_component" >  < /Link >
```

We can also add more than one links separated by ' | '. This will appear as links in the browser which can be used to navigate throughout our react application.

Styling in React

Styling in react is pretty easy. We can even style in Bootstrap or any other styling library in react if we import them. React library for styling also has some great styling features with component level styling. The styling approach of react is 'in-line' and independent of the css style sheets. This means that every react component can be styled separately. For this, we create a object within the render function of the component which has the component's styling properties. Then, within the return function of the component we can apply the styling as :

```
return ( < div style = {object_with_styling_properties} >
            <h1> This heading is Styled with the above object </h1>
        < /div > );
```

This approach is pretty neat and easy when we need to apply styles to independent components. But, when the whole react application needs any styling change or re-styling, this approach

becomes a bit tedious. For this, we can import Bootstrap in our react application. Using bootstrap for overall application styling is popular among react users.

There are some of the basic concepts in react which helps to start developing in react.React is very powerful and flexible. It's being used in various websites and web application for its effectiveness in user interface design and development. Some of the examples include Facebook, Instagram, Netflix, Whatsapp, Google Photos and Drive, Whatsapp, Yahoomail and many more. React is not only used in web development but also in mobile application development. The huge trend going on in the mobile development world is to make lightweight, fast application with less load time and less memory consumption. The answer to this is hybrid applications, the applications which utilize the web technology to make all of this possible. React specializes in these types of application development through React Native, which is a library for application design for both android and ios. React native is a separate library from the core react. Due to this, react is not only heavily used in webs but also in mobile app development.

Reason for choosing this Technology

The world is going through a digital revolution. The internet is a critical part of this revolution. The explosion of internet availability has made it an decisive technology in the world today. It is so effective that a average person today, spends almost 20 hours a week in internet. The most crucial part of the internet is the web technology where people spend most of their time. I was always interested in the web technology and its development. The development of web is always creative and analytical. While presenting an web page, its front end makes the first impression on the visitors. While there are many languages out there used in front end development, React is one of the most recent and powerful language with increasing popularity.

Before choosing this technology, I did my research on react. During this, I learnt more about react and its features. The idea that each and every element in a web page can be manipulated, changed and updated in isolated manner specially took my interest. React does this by adopting the component based design. The feature of building fluent, attractive, fast, dynamic and responsive user interface is also quite attractive. React, is said to be the future of web development. Even the big players in the internet market like facebook, instagram, netflix, whatsapp web and so on are using react either completely or partially to develop their front end. React has become like an industry standard today. For a technology to have this much importance, it must have something special in store. React definitely does not disappoint. It has in store some pretty awesome features and properties. From its lightning fast performance to its flexibility and effectiveness react is certainly a gamechanger in the front end development. Interested in trying out new things also played a part in choosing this technology, as I had not studied react before. All these factors and information on react were enough to centralize my attention and interest on this course.

Learning Outcomes

I have learnt many things from this summer training and the technology, react. I will discuss some major learning outcomes from react then discuss the future possibilities and usability of react for me.

The first thing to learn is about react itself. It is a javascript library for front end development of a web application. It is mostly used to develop user interfaces which are beautiful, fast, adaptive, dynamic and responsive. We can either import some scripts to run react directly in a browser. To run react application in local machines, we need to install and configure NPM, which is a package manager for javascript programming language. JSX can be used to write code in javascript which

is later processed to XML. We also have alternatives to JSX like Babel which is a javascript compiler and configurable transpiler. The code written in react needs to be rendered to be represented or viewed in a browser. The rendering in react is done through its own virtual DOM, instead of an actual DOM, called as ReactDOM. This provides react with more speed, ease, efficiency and flexibility.

React utilizes its own virtual DOM, ReactDOM more efficiently due to the fact that it has adopted the component based development design. Since every element in a react app can be modified and manipulated using the concept of components, react is component based. The heart of every react application is its components. We can manipulate and update every element in a web application in isolated manner through the use of components, as we can create every single elements in a different component in react. This is where magic of the virtual DOM shines. The ability of ReactDOM to re-render only the changed or updated components in the actual DOM is one of the main features of react. It makes the development process of react applications very easy and fast. ReactDOM needs to be imported separately because it's separate from the react core library. This is because React can be used to develop more than just web apps. The arrival of react-native, react-canvas, react-art e.t.c. has made react able to render in more than just browser development environment. For example, react native enables us to build native mobile applications on IOS or Android. The core react though can be used in every development environment react supports.

React also has the concept props and states. The props extends the data management feature of react. The data can be passed and managed through the use of props. This gives us ability to create a static web page with data management features. However, the value of props cannot be changed during the lifetime of a react component. Here, the feature of states provide an edge. The value of state can be modified and manipulated even in the runtime of a react application. This gives us ability to create a dynamic application with adaptive and responsive user interface.

Any dynamic website requires to be beautiful and the more it can provide immersive user experience the better. React does this with its styling ability. The styling in react is at component

level with ‘in-line’ styling method. This means that react provides us the ability to style every single component separately. We can save our styling properties in a react object and apply that to style a component. This type of styling is reusable and can be used to style even the minute details easily. However, during redesign or re-styling of a whole application modifying each and every styling object of each component can be very tedious. For this, react allows developers to use the javascript libraries like bootstrap for styling. Bootstrap is especially popular among react developers for its great features and compatibility with react development environment. Apart from bootstrap, we can also use other javascript libraries for styling our react application.

React is specially used to develop single page applications. This single page application need to be easy to browse and navigate. For this , react has a Routing library. This library needs to be imported and then can be used in react applications. Routing provides us features like link navigating, history, navigate forward or backward e.t.c directly in our application. This makes even a single page application more interactive and navigable easily.

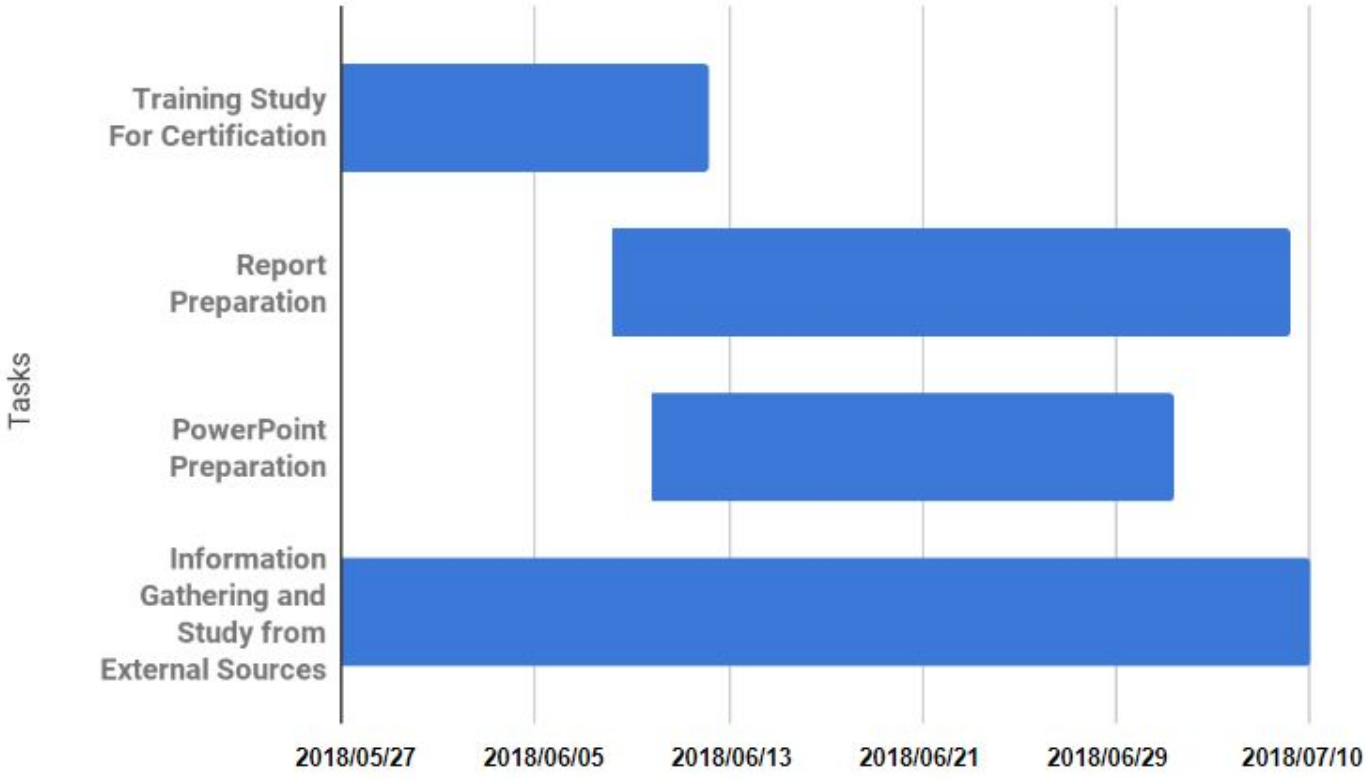
These are some of the concepts I learned from my summer training in react. These learning outcomes can be very useful in the coming years, specially in web development area. Industry demands for web developers who can code in react is increasing rapidly. So react is definitely going to be a plus point in my career. Apart from web development, mobile applications of react is also surging upwards. Nowadays, the mobile applications are moving from heavy applications to simpler, lighter and fast web like applications i.e. web apps. Many internet facilities today, offer web apps in android or ios along with their regular application. Though, react can be used to build both types of applications, web apps are its speciality. Mobile development world is always searching for a way to build light-weight application with better performance in short development and deployment time. React effectively provides this without any compromise but advantages like efficiency, less memory usage, less load time, support for third party plugins, smoother experience and many more. So, react is very useful in mobile application development area. This is very advantageous to me as I am an enthusiast in android application development.

Apart from this, I also sharpened my time management skills since we had to regularly log in to study and carry out the studies required for my training completion and report preparation. This type of study in this summer definitely helped to expand my knowledge base on overall web technology. Before this, I looked at online courses as not a concrete way of study. But, after this experience in Simplilearn, I have changed my view towards online learning. Online learning can be very effective and a easy way to learn anything in a short amount of time. The resources available today in internet is very vast and easily accessible, this makes online learning very fast, extensive, self paced and also cost effective. Talking about internet resources, I also learned to browse the internet in search of specific informations or resources more better. As I had to search for react informations.

I also came across some awesome online development environments like Scrimba and Codepen. These were definitely helpful during the learning process of react. Trying out new things and experimenting with the code became much easier and faster. Even just playing around in these development environments was fun and always gave me something new to learn. These development environments are certainly going to be helpful in the future too.

Gantt Chart

Following is the gantt chart for my summer training studies and project report preparation :



Bibliography

Following are the sources that were used to study and collect information on react :

https://www.simplilearn.com , React.js Course Study Materials

https://reactjs.org/docs.html

https://reactjs.org/tutorial/tutorial.html

https://medium.com/@hidace/understanding-reacts-virtual-dom-vs-the-real-dom-68ae29039951

http://reactkungfu.com/

https://facebook.github.io/react-native/docs/state

https://github.com/facebook/react

https://en.wikipedia.org/wiki/React_(JavaScript_library)

https://scotch.io/tutorials/learning-react-getting-started-and-concepts

https://www.google.com

https://scrimba.com

https://codepen.io