Training Report React.js

BTech (Kurukshetra University)

Studocu is not sponsored or endorsed by any college or university
Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

INDUSTRIAL TRAINING REPORT

Infowiz Software Solutions

Submitted in partial fulfillment of the

Requirements for the award of

Degree of Bachelor of Technology

In

Computer Science and Engineering

Submitted By :

Shiya Verma

1219303

7th Semester

SUBMITTED TO:

Er. Simran

Computer Science and Engineering Department

Seth Jai Parkash Mukand Lal Institute of Engineering and Technology,

Radaur – 135133 (Yamuna Nagar)

This document is available free of charge on studocu

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

DECLARATION

I hereby declare that the Industrial Training Report on Web Development (HTML,CSS,JS,REACT.JS) is an authentic record of my own work as requirements of Industrial Training during the period from 10 july,2022 to 30 September,2022 for the award of degree of B.Tech. (in Computer Science and Engineering), at Seth Jai Parkash Mukand Lal Institute of Engineering & Technology.

Shiya
(1219303)

Date:

Certified that the above statement made by the student is correct to the best of our knowledge and belief.

Head of Department

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

Certificate

INFOWIZ®
A SOFTWARE SOLUTION

CII Member of Confederation of Indian Industry | ISO 9001 CERTIFIED | 11 YEAR OF EXCELLENCE

H.O.: SCO 118 - 120, Sector 34 - A, **CHANDIGARH**
B.O.: First Floor, Crown Tower, 100 Ft. Road, **BATHINDA**
Web.: www.infowiz.co.in E-mail : info@infowiz.co.in

Certificate of Training

Certificate

No. Infowiz/6w2022/1893

This is certified that Mr./Ms. Shiya S/D/o. Sh. Ram Karan

of IMIT, Radaur has successfully undergone Training Course React. Js

From 15 July 2022 to 30 Aug 2022 During the tenure of the above course, we found him/her a hardworking & innovative individual.

We wish him/her a very bright and prosperous future.

Technical Head

Managing Director

Chandigarh : 0171 4567888, 9023400888, 96460 00952 Bathinda : 0164 5007088, 90235 00888, 90236 00888

This document is available free of charge on studocu

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

ACKNOWLEDGEMENT

First and foremost, I wish to express my sincere thanks and gratitude to my esteemed Mentor –Mr. Laxman‖ who has contributed so much for successful completion of my Industrial Training by his thoughtful reviews and valuable guidance.

Next I would like to tender my sincere thanks to –Dr. Gaurav Sharma‖ (Head of CSE Department) for his co-operation and encouragement.

Shiya
(1219303)

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

WEB DEVELOPMENT

Website is a collection of related web pages, including multimedia content, typically identified with a common domain name, and published on at least one web server. A website may be accessible via a public Internet Protocol (IP) network, such as the Internet or a private local area network (LAN), by referencing a uniform resource locator (URL) that identifies the site.

Websites can have many functions and can be used in various fashions, a website can be a personal website, a commercial website for a company, a government website or a non-profit organization website. Websites are typically dedicated to a particular topic or purpose, ranging from entertainment and social networking to providing news and education. All publicly accessible website for its employees, are typically a part of an intranet.

Web development is a broad term for the work involved in developing a website for the internet (World Wide Web) or an intranet (a private network). Web development can range from simplest static single page of plain text to the most complex web- based internet application (or just _web apps‘) electronic business, and social network services. A more comprehensive list of tasks to which web development commonly refers, may include web engineering, web server and network security configuration, and e-commerce development. Among web professionals, ‒web development‖ usually refers to the main non-design aspects of building web sites: writing markup and coding…

This document is available free of charge on studocu

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

Several Aspect of Web Developing

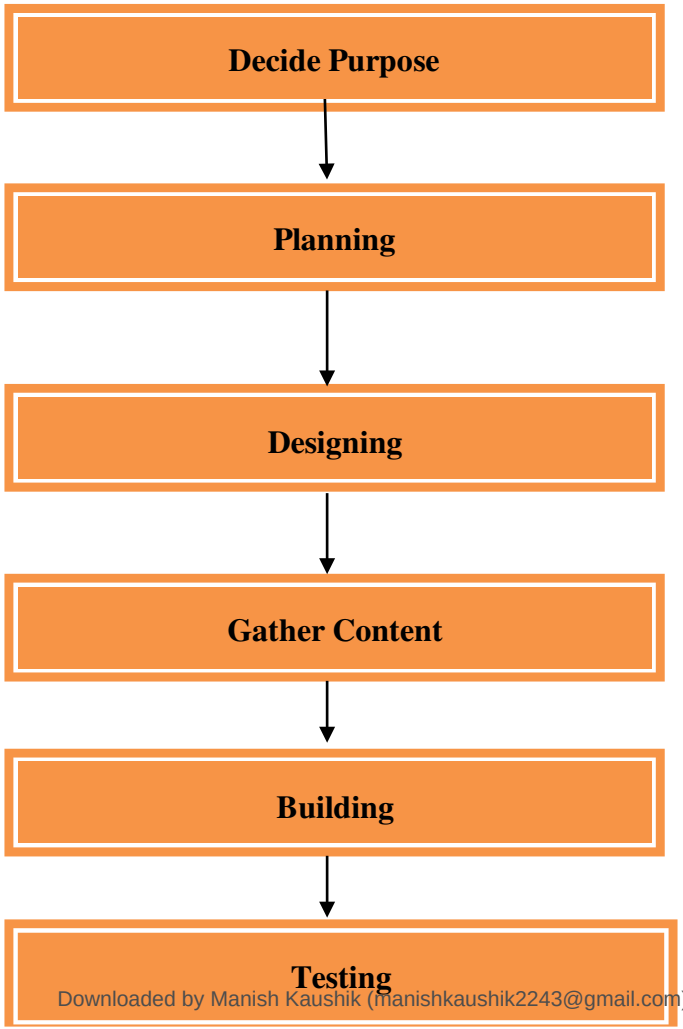
Before developing a web site once should aspects in mind like.

- What to put on the web site?
- Who will host it?
- How to make it interaction?
- How to secure code frequently?
- Will the web site design well in different browsers?
- Will the navigation menus be easy to use?

- Will the web site loads quickly?
- How easily will the site pages print?

Process

These are the steps considered while developing a webpage:



Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

HARDWARE AND SOFTWARE REQUIREMENT:

Hardwares Required:

TABLE 1-HARDWARES REQUIRED

Number	Description
1.	Pentium 4, Window XP/ Window 7
2.	256 MB RAM

Softwares Required:

TABLE 2-SOFTWARES REQUIRED

Number	Description
1.	Windows 7,10,11
2.	Node.js
3.	React.js
4.	HTML/CSS/JavaScript

This document is available free of charge on studocu

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

TOOLS

Introduction

The Translate and Edit application had been planned to consist of two parts- front-end and back-end development. The front-en is the part of the web that you can see and interact with (e.g. Client-side programming). While front-end code interacts with the user in real time, the back-end interacts with the server to return user ready results. The front-end is a combination of HTML, CSS and JavaScript coding. By using JavaScript, modifications of the design of a web page can be made immediately, however only temporary and visible only by the user.

Normally the user would not have rights to modify web content dynamically on the server side. Logically, administrators are the ones who deal with back-end modification of databases for example, as they often contain sensitive data which should not be available to see or modify by the general public. These front-end and back-end tools includes languages like HTML, CSS , JavaScript, PHP, MYSQL etc. we will discuss only HTML, CSS and JavaScript languages in brief as given below.

Features

- Web Pages Assests, Resources and Network Information
- Profting and Auditing

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

HTML

Introduction:

HTML (Hyper Text Mark-Up Language) is what is known as a –Mark-Up Language‖ whose role is to prepare written documents using formatting tags. The tags indicate how the document is presented and how it links to other documents.

The **World Wide Web** (WWW for short), or simply the Web, is the worldwide network formed by all the documents (called –**web pages"**) which are connected to one another by hyperlinks.

Web pages are usually **organized** around a main page, which acts as a hub for browsing other pages with hyperlinks. This group of web pages joined by hyperlinks and centered around a main pages is called a **website.**

The Web is a vast living archive composed of a myriad of web sites, giving people access to web pages that may contain formatted text, images, sounds, video, etc.

What is the Web?

The Web is composed of web pages stored on the web servers, which are machines that are constantly connected to the internet and which provide the pages that users request. Every web page, and more generally any online resources, such as images, video, music and animation, is associated with a unique address called a URL. The key element for viewing web pages is the **browser,** a software program which sends requests to web servers, then processes the resulting data and displays the information as intended, based on instructions in the HTML page.

The most commonly used browsers on the Internet include:

- Mozilla Firefox,
- Microsoft Internet Explorer,
- Netscape Navigator,
- Safari,
- Opera

Versions of HTML

HTML was designed by Tim Berners-Lee, at the time a researcher at CERN (Chinese Ecosystem Research Network), beginning in 1989. He officially announced the creation of the web on Usenet in August 1991. However, it wasn‘t until 1993 that HTML was considered advanced enough to call it a language (HTML was then symbolically christened HTM 1.0)

RFC 1866, dated November 1995, represented the first official version of HTML, called HTML 2.0. After the brief appearance of HTML 3.0, which was never officially released, HTML 3.2

This document is available free of charge on studocu

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

became the official standard on January 14, 1997. The most significant changes to HTML 3.2 were the standardization of tables, as well as many features relating to the presentation of web pages.

On December 18, 1997, HTML 4.0 was released. Version 4.0 of HTML was notable for standardizing style sheets and frames. HTML version 4.01, which came out on December 24, 1999, made several minor modifications to HTML 4.0.

Example-

<HTML>

<HEAD>

</HEAD>

<BODY>

<H5>THIS IS AN EXAPLE</H5>

</BODY>

</HTML>

Basic HTML structure

HTML is composed of *elements*. These elements structure the webpage and define its content.

A *tag* and the *content* between it is called an HTML element.

Let‘s quickly review each part of the element :

- HTML element (or simply, element) — a unit of content in an HTML document formed by HTML tags and the text or media it contains.
- HTML Tag — the element name, surrounded by an opening (<) and closing (>) angle bracket.
- Opening Tag — the first HTML tag used to start an HTML element. The tag type is surrounded by opening and closing angle brackets.
- Content — The information (text or other elements) contained between the opening and closing tags of an HTML element.
- Closing tag — the second HTML tag used to end an HTML element. Closing tags have a forward slash (/) inside of them, directly after the left angle bracket.

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

Our first html code :-

```
<html>
        <head>
                <title> </title>
        </head>
        <body>
        </body>
</html>
```

Explanation:

HTML Tag

Anything between <html> and </html> will be considered HTML code. Without these tags, it's possible that browsers could incorrectly interpret your HTML code and present HTML content in unexpected ways.

Head & Title Tag

Let's also give the browser some information about the page. We can do this by adding a <head> element.

The <head> element will contain information about the page that isn't displayed directly on the actual web page i,e title of the page

The browser displays the title of the page because the title can be specified directly inside of the <head> element, by using a <title> element.

Body Tag

Before we can add content, we have to add a *body* to the HTML file. Once the file has a body, many different types of content can be added within the body, like text, images, buttons, and much more.

<body> </body>

This document is available free of charge on studocu

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

CSS

What is CSS?

- CSS stands for Cascading Style Sheets.
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files.
- CSS describes how HTML elements should be displayed.
- CSS Saves a Lot of Work! The style definitions are normally saved in external .css files.
- With an external stylesheet file, we can change the look of an entire website by changing just one file!
- CSS can be either external or internal.

CSS Syntax:

A CSS rule-set consists of a selector and a declaration block:

CSS selector: The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

A CSS declaration always ends with a semicolon, and declaration bolocks are surrounded by curly braces.

The **External CSS** can be declared in the required HTML page as:

<link rel = ‖ stylesheet ‖ href = ‖ CSS_file_name.css ‖>

The external CSS file is saved by using the .css extension, whereas the internal CSS is saved in corresponding HTML file using the **<style>** tag. Using External CSS is much better than using Internal. **Here are a few reasons this is better.**

- Easier Maintenance
- Reduced File Size
- Reduced Bandwidth
- Improved Flexibility

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

The selectors that can be used to select the HTML part are-

- Id selector
- Class selector

Id Selector:

The id selector uses the id attribute of an HTML element to select a specific element. The id of an element should be unique within a page, so the id selector is used to select one unique element! To select an element with a specific id, write a hash (#) character, followed by id of the element. The style rule below will be applied to the HTML element with id=‖ para1‖:

Example-

Suppose the HTML content is as follow,

<h1 id=‖para1‖>content </h1>

Then Id will be declared as

para1{

text - align: center;

color : blue;

font - family: jokerman;

}

The class Selector:

The class selector selects elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the name of the

Class.

Example-

. para1

{

text - align: center;

This document is available free of charge on studocu

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

color : blue;

font-family : jokerman;

}

CSS Comments:

Comments are used to explain the code, and may help when you edit the source code at a later date. Comments are ignored by browsers. A CSS comment starts with /* and ends with /*. Comments can also span multiple lines.

Example-

.para1{

Text-align: center;

Color; blue;

Font-family: jokerman; /*this is the single line comment */

}

In the example above, all HTML elements with class=para1 will be blue and center – aligned.

CSS Style:

- Background properties
- Border properties
- Padding
- Margin
- Color
- Font properties
- Text properties, link properties/ navigation bar properties

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

JAVASCRIPT

What is JavaScript?

JavaScript is an object-based scripting language that is lightweight and cross-platform. JavaScript is not compiled by translated. The JavaScript translator (embedded in browser) is responsible to translate the JavaScript code.

It is mainly used for

- Client- side validation
- Dynamic drop- down menus.
- Displaying data and time.
- Displaying popup windows and dialog boxes(like alert dialog box, confirm dialog box and prompt dialog box).
- Displaying clocks etc.

Example of JavaScript-

<h2>Welcome to JavaScript document</h2>

<script>

document.write(— hello JavaScript by JavaScript‖);

</script>

Here, <script> tag is used to initialize the script and document.write() is a function used to write. Like CSS, JavaScript also can be placed in:

- Between the body tag of html
- In .js file (external javascript)
- Between the head tag of html

JavaScript Example: code between the body tag-

In the given example, we have displayed the dynamic content using JavaScript. Let‖s see the simple example of JavaScript that displays alert dialog box.

<script type=‖text/javascript‖>

alert(—hello JavaScript‖);

This document is available free of charge on studocu

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

</script>

JavaScript Example: code in .JS file-

- **Message.js file**

```
Function msg()
{
  alert(―Hello JavaScript‖);
}
```

- **Index.html**

```
<head>
<script type=‖text/javascript‖ src=‖message.js‖></script>
</head>
<body>
<p> Welcome to JavaScript</p>
<form>
<input type=‖buttom‖ value=‖click‖ onclick=‖msg()‖/>
</form>
</body>
```

We can create external JavaScript file and embed it in many html page.
It provides **code re usability** because single JavaScript file can be used in several html pages. An external JavaScript file must be saved by .js extension. It is reconnected to embed all JavaScript files into a single file. It increases the speed of the webpage.

Between the head tag of html

In the example given below, we are having a function msg() which is called. To create a function, we use function name with keyword **function.** For function call, we need to have an event.

Example-

```
<head>
<script type=‖text/javascript‖>
function msg()
{
  alert(―Hello JavaScript‖);
```

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

```
  }
  </script>
</head>
<body>
<p> Welcome to JavaScript</p>
<form>
<input type="button" value="click" onclick="msg()"/>
</form>
</body>
```

How to Change Content of HTML using a JavaScript?

One of many JavaScript HTML methods is **getElementById()**

This example uses the method to "find" an HTML element (with id="demo") and change the element content (**innerHTML**) to "Hello JavaScript".

Example-

```
document.getElementById("demo").innerHTML = "Hello JavaScript";
document.getElementById("demo").style.fontSize="25px";

<html>
<head>
<script>
Function myFunction()
{
    Document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
```

This document is available free of charge on studocu

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

</head>

<body>

<h1> My web Page</h1>

 A Paragraph

<button type=‖button‖ onclick=‖myFunction()‖>Try it </button>

</body>

</html>

Comments in JavaScript:

The **JavaScript comments** are meaningful way to deliver message. It is used to add information about the code, warning or suggestions so that end user can easily interpret the code. The JavaScript comment is ignored by the JavaScript engine i.e. embedded in in the browser.

Advantages of JavaScript comments:

There are mainly two advantages of JavaScript comments

- **To make code easy to understand:** It can be used to elaborate the code so that end user can easily understand the code.
- **To make code easy to understand:** It can be used to elaborate the code so that end user can easily understand the code.
- **To avoid the unnecessary code:** It can be used to avoid the code being executed. Sometimes, we add the code to perform some action. But after sometimes, there may be need to disable the code. In such case, it is better to use comments.

Example-

<script type=‖text/javascript‖>

Function msg()

{

Alert(–Hello Javatpoint‖); **/*this is a comment*/**

}

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

<script>

JavaScript Variable:

A **JavaScript variable** is simply a name of storage location. There are two types of variable in JavaScript: local variable and global variable. There are some rules while declaring a JavaScript variable (also known as identifiers).

- Name must start with a letter (a to z or A to Z), underscore (_), or dollar($) sign.
- After first letter we can digits (0 to 9), for example value 1.

JavaScript variables are case sensitive, for example x and X are different variables.

JavaScript Form Validation:

It is important to validate the form submitted by the user because it can have inappropriate values. So validation is must.

The JavaScript provides you the facility the validate the form on the client side so processing will be fast than server- side validation. So, most of the web developers prefer JavaScript form validation.

Through JavaScript, we can validate name, password, email, date, mobile number etc. fields .

Example-

<script>

function validateform()

{

var name = document.myform.name.value;

var password = document.myform.password.value;

if (name== null || name== — ‖)

{

alert(— Name can‘t be blank‖);

return false;

}

This document is available free of charge on studocu

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

```
else if ( password.length<6)

{

alert(— Password must be at least 6 characters long.‖);

return false;

}

}

</script>

<body>

<form name= –myform‖ method= –post‖ action= –abc.jsp‖ onsubmit= –return validateform()‖ >

Name: <input type= –text‖ name= –name‖><br/>

Password: <input type= –password‖ name= –password‖><br/>

<input type= –submit‖ value= –register‖>

</form>

</body>
```

In this example, we are going to validate the name and password. The name can‘t be empty and password can‘t be less than 6 characters long. Here, we are validating the form on form submit. The user will not be forwarded to the next page until given values are correct.

JavaScript Functions:

JavaScript functions are used to perform operations. We can call JavaScript function many times to reuse the code.

Advantages of JavaScript Function:

There are mainly two advantages of JavaScript functions.

- Code reusability
- Less coding

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

JavaScript Function Syntax

The syntax of declaring function is given below.

```
function functionName( [arg1, arg2,….argN])
{
//code to be executed
}
```

JavaScript Functions can have 0 or more arguments.

Example-

```
<script>
function msg()
{
alert( –hello! This is message‖);
}
</script>
<input type= –button‖ onclick = –msg()‖ value= –call function‖/>
```

Output of the above example:

Hello! This is message

This document is available free of charge on studocu

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

React.js

React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called ―components‖.

We‘ll get to the funny XML-like tags soon. We use components to tell React what we want to see on the screen. When our data changes, React will efficiently update and re-render our components.

The render method returns a *description* of what you want to see on the screen. React takes the description and displays the result. In particular, render returns a **React element**, which is a lightweight description of what to render. Most React developers use a special syntax called ―JSX‖ which makes these structures easier to write. The syntax is transformed at build time to React.createElement('div'). The example above is equivalent to:

Passing Data Through Props

To get our feet wet, let‘s try passing some data from our Board component to our Square component.

We strongly recommend typing code by hand as you‘re working through the tutorial and not using copy/paste. This will help you develop muscle memory and a stronger understanding.
In Board‘s renderSquare method, change the code to pass a prop called value to the Square:

```
class Board extends React.Component {
  renderSquare(i) {
    return <Square value={i} />;  }
}
```

Change Square‘s render method to show that value by replacing {/* TODO */} with {this.props.value}:

```
class Square extends React.Component {
  render() {
    return (
      <button className="square">
        {this.props.value}      </button>
    );
  }
}
```

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

Function Components

We'll now change the Square to be a **function component**.
In React, **function components** are a simpler way to write components that only contain a render method and don't have their own state. Instead of defining a class which extends React.Component, we can write a function that takes props as input and returns what should be rendered. Function components are less tedious to write than classes, and many components can be expressed this way.

Replace the Square class with this function:

```
function Square(props) {
  return (
    <button className="square" onClick={props.onClick}>
      {props.value}
    </button>
  );
}
```

We have changed this.props to props both times it appears.

React State

The state is an updatable structure that is used to contain data or information about the component. The state in a component can change over time. The change in state over time can happen as a response to user action or system event. A component with the state is known as stateful components. It is the heart of the react component which determines the behavior of the component and how it will render. They are also responsible for making a component dynamic and interactive.

A state must be kept as simple as possible. It can be set by using the **setState()** method and calling setState() method triggers UI updates. A state represents the component's local state or information. It can only be accessed or modified inside the component or by the component directly. To set an initial state before any interaction occurs, we need to use the **getInitialState()** method.

This document is available free of charge on studocu

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

React Component Life-Cycle

In ReactJS, every component creation process involves various lifecycle methods. These lifecycle methods are termed as component's lifecycle. These lifecycle methods are not very complicated and called at various points during a component's life. The lifecycle of the component is divided into **four phases**. They are:

1. Initial Phase
2. Mounting Phase
3. Updating Phase
4. Unmounting Phase

Each phase contains some lifecycle methods that are specific to the particular phase. Let us discuss each of these phases one by one.

1. Initial Phase

It is the **birth** phase of the lifecycle of a ReactJS component. Here, the component starts its journey on a way to the DOM. In this phase, a component contains the default Props and initial State. These default properties are done in the constructor of a component. The initial phase only occurs once and consists of the following methods.

- **getDefaultProps()**
  It is used to specify the default value of this.props. It is invoked before the creation of the component or any props from the parent is passed into it.
- **getInitialState()**
  It is used to specify the default value of this.state. It is invoked before the creation of the component.

2. Mounting Phase

In this phase, the instance of a component is created and inserted into the DOM. It consists of the following methods

- **componentWillMount()**
  This is invoked immediately before a component gets rendered into the DOM. In the case, when you call **setState()** inside this method, the component will not **re-render**.

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

- **componentDidMount()**
  This is invoked immediately after a component gets rendered and placed on the DOM. Now, you can do any DOM querying operations.
- **render()**
  This method is defined in each and every component. It is responsible for returning a single root **HTML node** element. If you don't want to render anything, you can return a **null** or **false** value.

3. Updating Phase

It is the next phase of the lifecycle of a react component. Here, we get new **Props** and change **State**. This phase also allows to handle user interaction and provide communication with the components hierarchy. The main aim of this phase is to ensure that the component is displaying the latest version of itself. Unlike the Birth or Death phase, this phase repeats again and again. This phase consists of the following methods.

- **componentWillRecieveProps()**
  It is invoked when a component receives new props. If you want to update the state in response to prop changes, you should compare this.props and nextProps to perform state transition by using **this.setState()** method.
- **shouldComponentUpdate()**
  It is invoked when a component decides any changes/updation to the DOM. It allows you to control the component's behavior of updating itself. If this method returns true, the component will update. Otherwise, the component will skip the updating.
- **componentWillUpdate()**
  It is invoked just before the component updating occurs. Here, you can't change the component state by invoking **this.setState()** method. It will not be called, if **shouldComponentUpdate()** returns false.
- **render()**
  It is invoked to examine **this.props** and **this.state** and return one of the following types: React elements, Arrays and fragments, Booleans or null, String and Number. If shouldComponentUpdate() returns false, the code inside render() will be invoked again to ensure that the component displays itself properly.
- **componentDidUpdate()**
  It is invoked immediately after the component updating occurs. In this method, you can

This document is available free of charge on studocu

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

put any code inside this which you want to execute once the updating occurs. This method is not invoked for the initial render.

4. Unmounting Phase

It is the final phase of the react component lifecycle. It is called when a component instance is **destroyed** and **unmounted** from the DOM. This phase contains only one method and is given below.

- **componentWillUnmount()**
  This method is invoked immediately before a component is destroyed and unmounted permanently. It performs any necessary **cleanup** related task such as invalidating timers, event listener, canceling network requests, or cleaning up DOM elements. If a component instance is unmounted, you cannot mount it again.

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

React Router

Routing is a process in which a user is directed to different pages based on their action or request. ReactJS Router is mainly used for developing Single Page Web Applications. React Router is used to define multiple routes in the application. When a user types a specific URL into the browser, and if this URL path matches any 'route' inside the router file, the user will be redirected to that particular route.

React Router is a standard library system built on top of the React and used to create routing in the React application using React Router Package. It provides the synchronous URL on the browser with data that will be displayed on the web page. It maintains the standard structure and behavior of the application and mainly used for developing single page web applications.

Need of React Router

React Router plays an important role to display multiple views in a single page application. Without React Router, it is not possible to display multiple views in React applications. Most of the social media websites like Facebook, Instagram uses React Router for rendering multiple views.

React Router Installation

React contains three different packages for routing. These are:

1. **react-router:** It provides the core routing components and functions for the React Router applications.
2. **react-router-native:** It is used for mobile applications.
3. **react-router-dom:** It is used for web applications design.

It is not possible to install react-router directly in your application. To use react routing, first, you need to install react-router-dom modules in your application. The below command is used to install react router dom.

1. $ npm install react-router-dom --save

This document is available free of charge on studocu

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

React Redux

Redux is an open-source JavaScript library used to manage application state. React uses Redux for building the user interface. It was first introduced by **Dan Abramov** and **Andrew Clark** in **2015**.

React Redux is the official React binding for Redux. It allows React components to read data from a Redux Store, and dispatch **Actions** to the **Store** to update data. Redux helps apps to scale by providing a sensible way to manage state through a unidirectional data flow model. React Redux is conceptually simple. It subscribes to the Redux store, checks to see if the data which your component wants have changed, and re-renders your component.

Redux was inspired by Flux. Redux studied the Flux architecture and omitted unnecessary complexity.

- Redux does not have Dispatcher concept.
- Redux has an only Store whereas Flux has many Stores.
- The Action objects will be received and handled directly by Store.

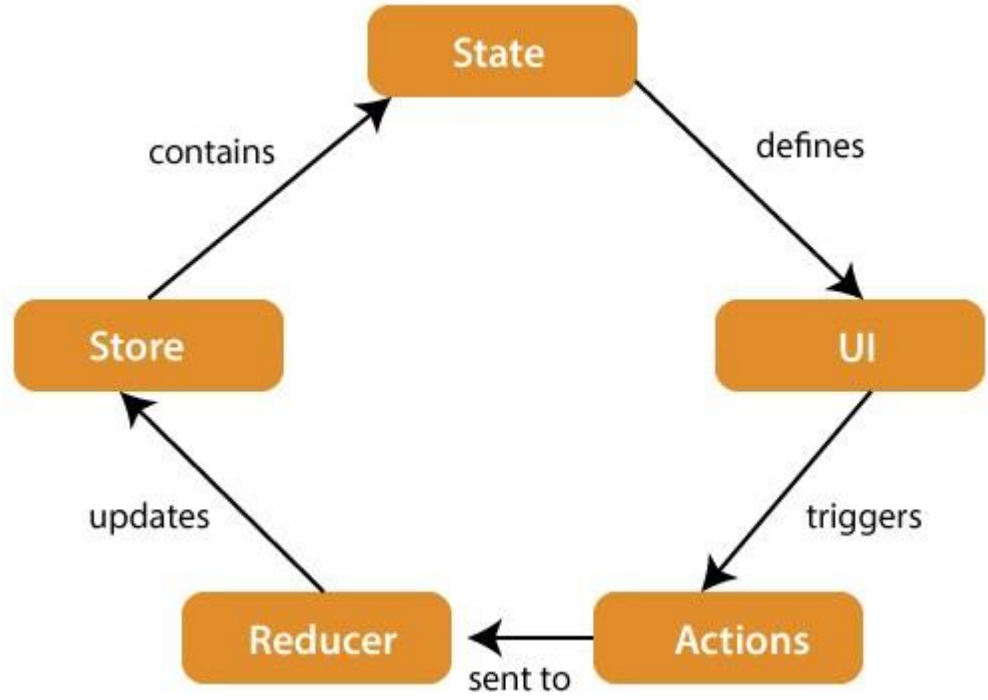
Why use React Redux?

The main reason to use React Redux are:

- ensure that your React components behave as expected.
- It encourages good 'React' architecture.
- It implements many performance optimizations internally, which allows to components re-render only when it actually needs.

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

Redux Architecture



The components of Redux architecture are explained below.

STORE: A Store is a place where the entire state of your application lists. It manages the status of the application and has a dispatch(action) function. It is like a brain responsible for all moving parts in Redux.

ACTION: Action is sent or dispatched from the view which are payloads that can be read by Reducers. It is a pure object created to store the information of the user's event. It includes information such as type of action, time of occurrence, location of occurrence, its coordinates, and which state it aims to change.

REDUCER: Reducer read the payloads from the actions and then updates the store via the state accordingly. It is a pure function to return a new state from the initial state.

Redux Installation

Requirements: React Redux requires React 16.8.3 or later version.

To use React Redux with React application, you need to install the below command

$ npm install redux react-redux --save

This document is available free of charge on studocu

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)

Conclusion:

In a nutshell. This internship has been an excellent and rewarding experience. I can conclude that there have been a lot I‘ve learnt from my work at the training and research centre. Needless to say, the technical aspects of the work I‘ve done are not flawless and could be improved provided enough time.

As someone with no prior experience in JavaScript whatsoever I believe my time spent in training and discovering new languages was well worth it and contributed to finding an acceptable solution to an important aspect of web design and development. Two main things that I‘ve learned the importance of time – management skills and self- motivation.

Working with web development language has increased my interest in them, hence prompting me to transfer to the Web Design and Development course at my college.

Downloaded by Manish Kaushik (manishkaushik2243@gmail.com)