

IntimeTec VisionSoft Pvt. Ltd

PREPARED BY

Akshat Awasthi (2018BTECHCSE024)

FACULTY GUIDE

Prof. Devendra bhavsar

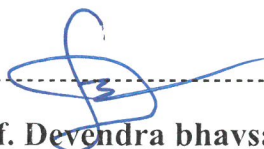


**Department of Computer Science &
Engineering Institute of Engineering and
Technology (IET) JK Lakshmipat University,
Jaipur**

July-2022

CERTIFICATE

This is to certify that the Practice School-2 project work entitled **IntimeTec Visionsoft Pvt. Ltd, Jaipur** submitted by **Akshat Awasthi (2018BTechCSE024)**, towards the partial fulfillment of the requirements for the degree of **Bachelor of Technology in Computer Science and Engineering** of JK Lakshmipat University Jaipur is the record of work carried out by them under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for the Practice School-II examination.



Prof. Devendra bhavsar

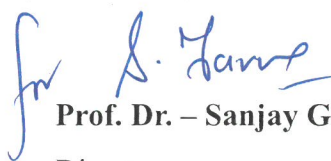
Associate Professor,
Department of Computer Science
Engineering
Institute of Engineering & Technology
(IET)
JK Lakshmipat University, Jaipur



Dr. Gireesh Kaushik

Ps-2 Co-ordinator,
Department of Computer Science
Engineering
Institute of Engineering & Technology
(IET)
JK Lakshmipat University, Jaipur

Date of Submission: 08-07-2022



Prof. Dr. – Sanjay Goel

Director
Institute of Engineering & Technology
JK Lakshmipat University Jaipur

ABSTRACT

The most fundamental software or any app development is to select which is the right front end framework or library to go with. The market has a wide variety due to the wide range of problems that developers face every day. For front end development, react.js is playing a vital role and creating new opportunities for developers to build new apps .The report tells about how react.js is helping in building those Application and what advantages it is having in building the front end. With more than 1,400 developers and over 92,000 sites making use of React.js to build their websites it wouldn't be an overstatement to call React the future of front-end development. After all, companies like Facebook, PayPal, Netflix utilize this JavaScript-based UI library for a major part of their application's front-end development for web as well as mobile. In this Report, the key features comprising this library is analyzed and how i have completed my training in the same technology in the organization.

CHAPTER 1: INTRODUCTION

1.1 About the organization



In Time Tec provides progressive software development services, enabling its clients to keep their brightest and most valuable talent focused on innovation. In Time Tec has a leadership team averaging 15 years in software/firmware R&D, and 20 years building onshore/offshore R&D teams. Every company has core and context work. As defined by author and speaker Geoffrey Moore, core work is your differentiator, your 'secret sauce'. Context work is non-core work that must get done. In Time Tec helps you focus on core work by taking on your context work. By using a blended model of onshore and offshore talent, we offer the following software development services:

It takes a special kind of person to stand with us on our platform of abundance. Our mission and our way of being drives everything we do. We want people to live their fullest potential while serving our clients and communities. It delivers a wide range of technical security services, from reviewing existing processes and documentation, and assessing the security of current environments, to the design of new secure IT systems to meet modern security requirements. It helps organizations meet their cyber compliance obligations by assisting in the development of risk compliance reviews governance activities, policies, and action plans

It can assist your organization to assess and improve your current cyber security resilience, through the use of mature security architecture, design, development, deployment, vulnerability testing, and management techniques. As cyber security threats evolve and new vulnerabilities are discovered, through an ongoing series of publications and presentations,

Intimetec Visionsoft Private Limited is a Private incorporated on 13 February 2009. It is classified as a non-govt company and is registered at the Registrar of Companies, Jaipur. Its authorized share capital is Rs. 50,000,000 and its paid-up capital is Rs. 37,730,720. It is involved in Software publishing, consultancy, and supply [Software publishing includes production, supply, and documentation of ready-made (non-customized) software, operating systems software, business & other applications software, and computer games software for all platforms. Consultancy includes providing the best solution in the form of custom software after analyzing the user's needs and problems. Custom software also includes made-to-order software based on orders from specific users. Also, included are writing of software of any kind following directives of the users; software maintenance, web-page.

CHAPTER 2: WORK FLOW

In this chapter I will tell you about the different parts on which I worked during my time at IntimeTec

2.1 Training programme for the assigned Technology

During the first week the company announced a list of trainees who were assigned to the different technologies and after the meetings with the mentors and managers and then the whole one week we were asked to study about GIT. After the meetings we were assigned a course in moodle platform where we have to study git and other courses were also assigned in the same platform.

2.2 JavaScript Training

After the successful completion of week one i was asked to study javaScript concepts in details and Assignment was there which i had to submit on time. one week time was allotted to study the concepts and complete the Assignment.

2.3 TypeScript Training

After the review of the week one assignment the next week started which was typeScript and two weeks were given to study the concepts deeply and assignment was to convert the same assignment into typescript.

2.4 The Deep Dive Session

My team mentor has taken an initiative that we have to prepare one topic from javascript one day one topic one person and we have to present that topic and anyone can ask any question in it.

2.5 React Js Training programme.

The React Js training programme comprises of different weeks in which topics were divided and weekly assignments were there according to the topics and in the final week there was a project submission which was to make a ecommerce website.

2.6 Deployment to the project

After the successful completion and review of the final project I was assigned to the project HPSC which is the HP Sales Central portal.

- JS String
- JS Date
- JS Math
- JS Number
- JS Boolean
- JavaScript DOM
- Document Object
- getElementById
- GetElementsByName()
- getElementsByTagName
- getElementsByTagName
- JS innerHTML property
- JS innerText property

• Properties of document object

- The properties of document object that can be accessed and modified by the document

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Calculator</title>
    <link href="Calculator with Js\style.css" rel="stylesheet">
    <script src="Calculator with JS\script.js" defer></script>
    <div class="calculator-grid">
      <div class="output">
        <div class="previous-operand"></div>
        <div class="current-operand"></div>
      </div>
      <button class="span-two">AC</button>
      <button>DEL</button>
      <button>+</button>
      <button>1</button>
      <button>2</button>
      <button>3</button>
      <button>*</button>
      <button>4</button>
      <button>5</button>
      <button>6</button>
      <button>+</button>
      <button>7</button>
      <button>8</button>
      <button>9</button>
      <button>-</button>
      <button>.</button>
      <button>0</button>
      <button class="span-two">=</button>
    </div>
  </head>
  <body>
  </body>
</html>

```

```

const numberButtons = document.querySelectorAll('[data-number]')
const operationButtons = document.querySelectorAll('[data-operation]')
const equalsButton = document.querySelector('[data-equals]')
const deleteButton = document.querySelector('[data-delete]')
const allClearButton = document.querySelector('[data-all-clear]')
const previousOperandTextElement = document.querySelector('[data-previous-operand]')
const currentOperandTextElement = document.querySelector('[data-current-operand]')

```

3.2 Week two The TypeScript

- **What is TypeScript?**
- TypeScript is an open-source pure object-oriented programming language. It is a strongly typed superset of JavaScript which compiles to plain JavaScript. It contains all elements of the JavaScript. It is a language designed for large-scale JavaScript application

- Null vs undefined
- TypeScript Variables
- Let vs Var
- TypeScript Operators
- Type Annotation
- Type Inference
- TypeScript Type Assertion
- TypeScript Arrays
- Typescript Tuples
- Typescript Union
- TypeScript String
- TypeScript Numbers
- TypeScript Decision Making
- TypeScript Switch Statement
- TypeScript Indefinite Loops
- TypeScript Definite Loop
- Typescript Enums
- Typescript forEach
- Typescript Map
- Typescript Set
- TypeScript Access Modifiers
- TypeScript Accessor

```

function px(x: number) { return (x * this.props.viewScale)+'px'; };

//== Evaluation algorithm ==//

function evaluate() {
  //-- Checking for valid input --//

  let errorMessage: string;

  //-- Checking for errors in entering numbers

  let numberError: boolean = false;

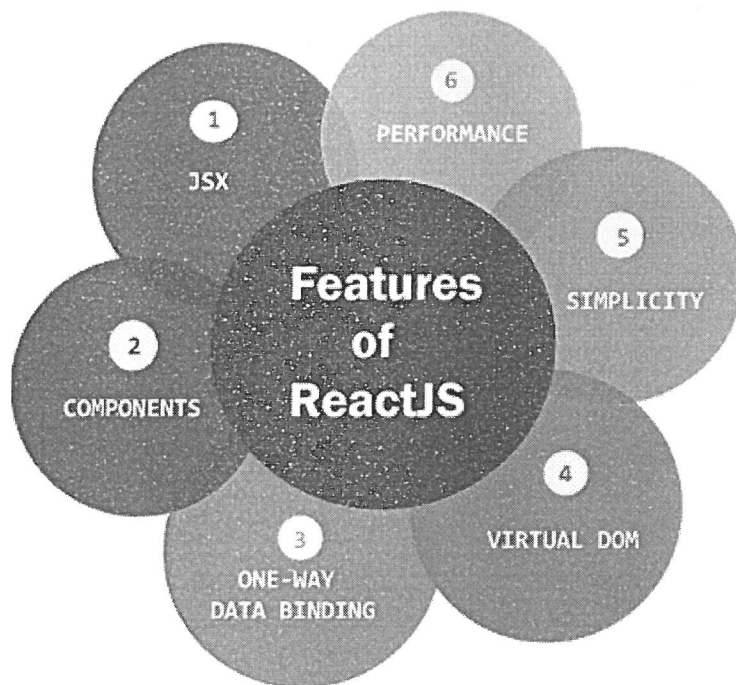
  // Combining numbers and decimal points
  let input: any[];
  input = this.state.in.reduce((prev: string[], curr: string, index: number) => {
    if( parseInt(curr) == curr
      && index !== 0
      && ( prev[prev.length-1] === "."
        || prev[prev.length-1] == parseFloat(prev[prev.length-1]) ) ) {
      prev[prev.length-1] += curr;
    }
    else if( curr === "."
      && index !== 0
      && prev[prev.length-1].indexOf(".") === -1
      && prev[prev.length-1] == parseFloat(prev[prev.length-1]) ) {
      prev[prev.length-1] += curr;
    }
    else prev.push(curr);
    return prev;
  }, []);
  numberError = input.reduce((prev: string, curr: string, index: number) => {
    if(prev) return prev;

    // Checks for extra decimal points
    if( curr === "." ) {

```

and then you need to focus on writing React code only. This tool sets up the development environment, provides an excellent developer experience, and optimizes the app for production.

React Features



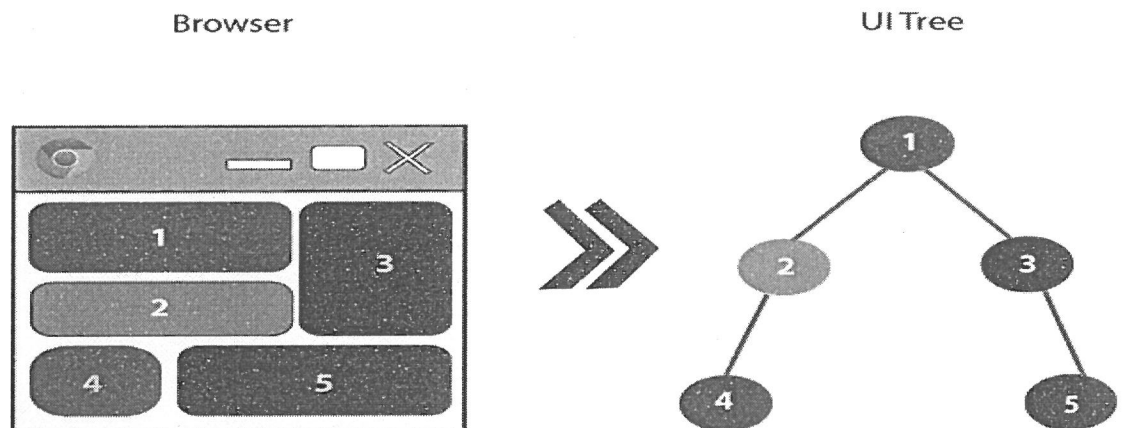
Currently, ReactJS is gaining quick popularity as the best JavaScript framework among web developers. It is playing an essential role in the front-end ecosystem. The important features of ReactJS are as following.

- JSX
- Components
- One-way Data Binding
- Virtual DOM
- Simplicity
- Performance

XHTML. The DOM exists entirely in memory. Due to this, when we create a component, we did not write directly to the DOM. Instead, we are writing virtual components that will turn into the DOM leading to smoother and faster performance.

React Components

- Earlier, the developers write more than thousands of lines of code for developing a single page application. These applications follow the traditional DOM structure, and making changes in them was a very challenging task. If any mistake found, it manually searches the entire application and update accordingly. The component-based approach was introduced to overcome an issue. In this approach, the entire application is divided into a small logical group of code, which is known as components.
- A Component is considered as the core building blocks of a React application. It makes the task of building UIs much easier. Each component exists in the same space, but they work independently from one another and merge all in a parent component, which will be the final UI of your application.
- Every React component have their own structure, methods as well as APIs. They can be reusable as per your need. For better understanding, consider the entire UI as a tree. Here, the root is the starting component, and each of the other pieces becomes branches, which are further divided into sub-branches



Functional Components

- In React, function components are a way to write components that only contain a render method and don't have their own state. They are simply JavaScript functions that may or may not receive data as parameters. We can create a function that takes props(properties) as input and returns what should be rendered. A valid functional

```

import React, { Component } from 'react';
class App extends React.Component {
  constructor() {
    super();
    this.state = {
      data: [
        {
          "name": "Abhishek"
        },
        {
          "name": "Saharsh"
        },
        {
          "name": "Ajay"
        }
      ]
    }
  }
  render() {
    return (
      <div>
        <StudentName/>
        <ul>
          {this.state.data.map((item) => <List data = {item} />)}
        </ul>
      </div>
    );
  }
}

class StudentName extends React.Component {
  render() {
    return (
      <div>
        <h1>Student Name Detail</h1>
      </div>
    );
  }
}

```

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

- **React Props**

- Props stand for "Properties." They are read-only components. It is an object which stores the value of attributes of a tag and work similar to the HTML attributes. It gives a way to pass data from one component to other components. It is similar to function arguments. Props are passed to the component in the same way as arguments passed in a function.
- Props are immutable so we cannot modify the props from inside the component. Inside the components, we can add attributes called props. These attributes are available in the component as `this.props` and can be used to render dynamic data in our render method.
- When you need immutable data in the component, you have to add props to `ReactDOM.render()` method in the `main.js` file of your ReactJS project and used it inside the component in which you need. It can be explained in the below example.

```
import React, { Component } from 'react';
class App extends React.Component {
  render() {
    return (
      <div>
        <h1> Welcome to { this.props.name } </h1>
        <p> <h4> JKLU </h4> </p>
      </div>
    );
  }
}
export default App;
```

Difference between State and Props

Props		State	
	Props are read-only.	State changes can be asynchronous.	
	Props are immutable.	State is mutable.	

```

import React, { Component } from 'react';

class App extends Component {
  constructor(props){
    super(props);
    this.state = {
      data: 'www.javatpoint.com'
    }
    this.handleClick = this.handleClick.bind(this);
  }
  handleClick(){
    console.log(this.props);
  }
  render() {
    return (
      <div className="App">
        <h2>React Constructor Example</h2>
        <input type="text" value={this.state.data} />
        <button onClick={this.handleClick}>Please Click</button>
      </div>
    );
  }
}
export default App;

```

React Component API

ReactJS component is a top-level API. It makes the code completely individual and reusable in the application. It includes various methods for:

- Creating elements

- Transforming elements

- Fragments

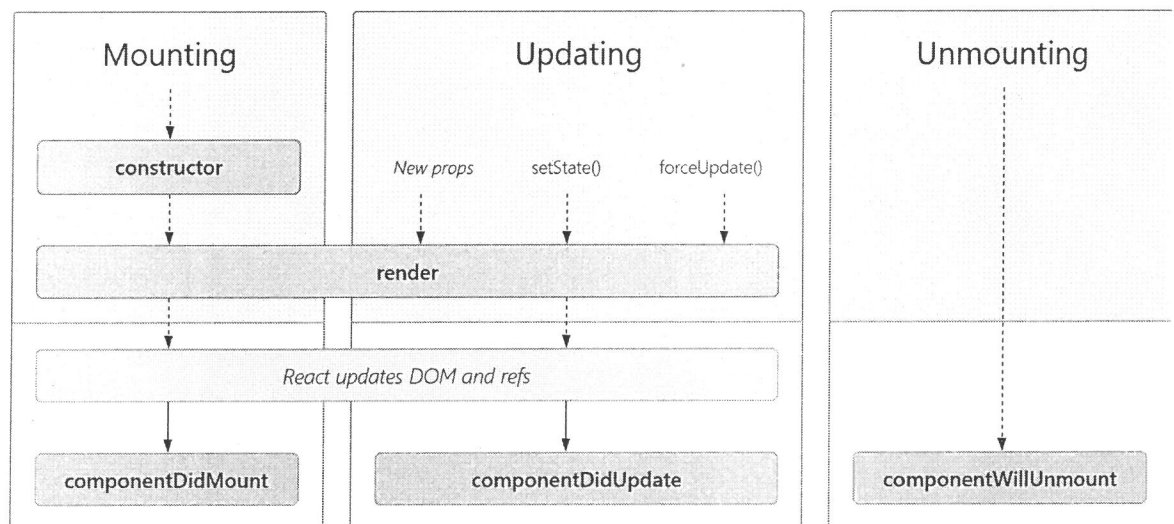
React Component Life-Cycle

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:

- Initial Phase
- Mounting Phase
- Updating Phase
- 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.



state and other React features without writing a class. Hooks are the functions which "hook into" React state and lifecycle features from function components. It does not work inside classes.

- Hooks are backward-compatible, which means it does not contain any breaking changes. Also, it does not replace your knowledge of React concepts.

When to use a Hooks

- If you write a function component, and then you want to add some state to it, previously you do this by converting it to a class. But, now you can do it by using a Hook inside the existing function component.

Rules of Hooks

- Hooks are similar to JavaScript functions, but you need to follow these two rules when using them. Hooks rule ensures that all the stateful logic in a component is visible in its source code. These rules are:

1. Only call Hooks at the top level

- Do not call Hooks inside loops, conditions, or nested functions. Hooks should always be used at the top level of the React functions. This rule ensures that Hooks are called in the same order each time a components renders.

2. Only call Hooks from React functions

- You cannot call Hooks from regular JavaScript functions. Instead, you can call Hooks from React function components. Hooks can also be called from custom Hooks.

Hooks State

- Hook state is the new way of declaring a state in React app. Hook uses useState() functional component for setting and retrieving state. Let us understand Hook state with the following example.

```

import React, { useState, useEffect } from 'react';

function CounterExample() {
  const [count, setCount] = useState(0);

  // Similar to componentDidMount and componentDidUpdate:
  useEffect(() => {
    // Update the document title using the browser API
    document.title = `You clicked ${count} times`;
  });

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}

export default CounterExample

```

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.

```
mkdir components
cd components
touch AddProduct.js Cart.js Login.js Productlist.js
```

In `AddProduct.js` add:

```
import React from "react";

export default function AddProduct() {
  return <>AddProduct</>
}
```

In `Cart.js` add:

```
import React from "react";

export default function Cart() {
  return <>Cart</>
}
```

In `Login.js` add:

```
import React from "react";

export default function Login() {
  return <>Login</>
}
```

```
value={{
  ...this.state,
  removeFromCart: this.removeFromCart,
  addToCart: this.addToCart,
  login: this.login,
  addProduct: this.addProduct,
  clearCart: this.clearCart,
  checkout: this.checkout
}}
```

```
<Router ref={this.routerRef}>
```

```
<div className="App">
```

```
<nav
```

```
  className="navbar container"
```

```
  role="navigation"
```

```
  aria-label="main navigation"
```

```
>
```

```
<div className="navbar-brand">
```

```
<b className="navbar-item is-size-4">ecommerce</b>
```

```
<label
```

```
  role="button"
```

```
  class="navbar-burger burger"
```

```
  aria-label="menu"
```

```
  aria-expanded="false"
```

```
  data-target="#navbarBasicExample"
```

```
import React, { Component } from "react";
import { Redirect } from "react-router-dom";
import withContext from "../withContext";

class Login extends Component {
  constructor(props) {
    super(props);
    this.state = {
      username: "",
      password: ""
    };
  }

  handleChange = e => this.setState({ [e.target.name]: e.target.value });

  login = (e) => {
    e.preventDefault();

    const { username, password } = this.state;
    if (!username || !password) {
      return this.setState({ error: "Fill all fields!" });
    }
    this.props.context.login(username, password)
      .then((loggedIn) => {
        if (!loggedIn) {

```

Login

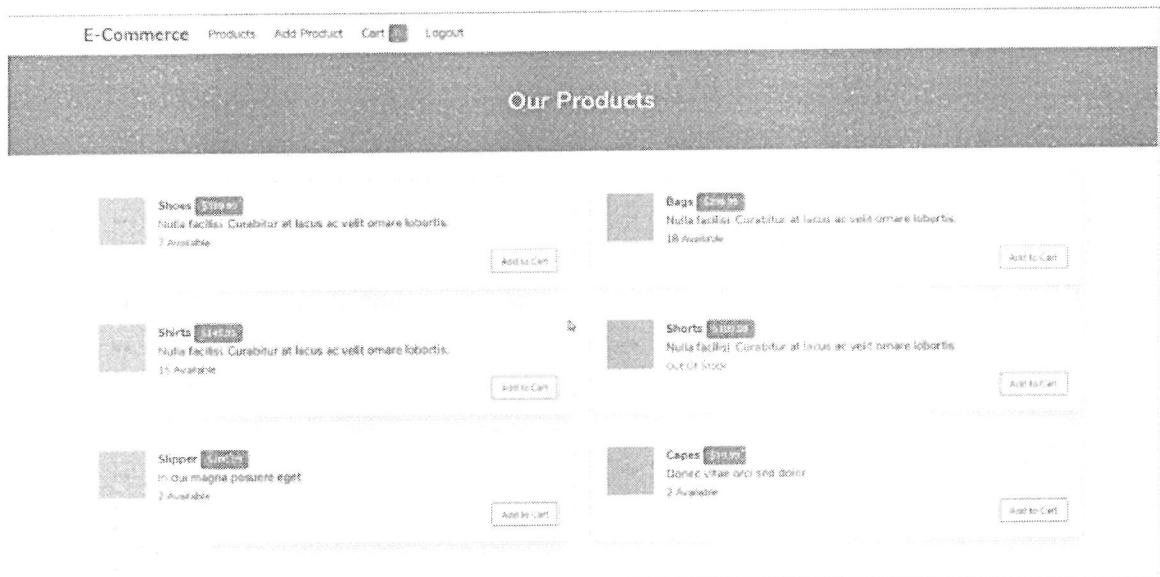
Email:

admin@example.com

Password:

Submit

Creating the Product Views



Adding a Product

```
if (name && price) {
  const id = Math.random().toString(36).substring(2) + Date.now().t

  await axios.post(
    'http://localhost:3001/products',
    { id, name, price, stock, shortDesc, description },
  )

  this.props.context.addProduct(
    {
      name,
      price,
      shortDesc,
      description,
      stock: stock || 0
    },
    () => this.setState(initState)
  );
  this.setState(
    { flash: { status: 'is-success', msg: 'Product created successf
  )};

} else {
  this.setState(
    { flash: { status: 'is-danger', msg: 'Please enter name and pri
  )};
}
```

Add Product

Product Name:

Cat

Price:

699

Available in Stock:

4

Short Description:

Goes Miaow

Description:

Very fluffy and needs constant attention, otherwise it will sharpen its claws on your sofa.

Submit

Adding Cart Management

```
async componentDidMount() {  
  let user = localStorage.getItem("user");  
  let cart = localStorage.getItem("cart");  
  
  const products = await axios.get('http://localhost:3001/products');  
  user = user ? JSON.parse(user) : null;  
  cart = cart ? JSON.parse(cart) : {};  
  
  this.setState({ user, products: products.data, cart });  
}
```

My Cart



Shoes 254: 90

Nulla facilisi. Curabitur at lacus ac velit ornare lobortis.
2 in cart



Shirts 254: 90

Nulla facilisi. Curabitur at lacus ac velit ornare lobortis.
2 in cart



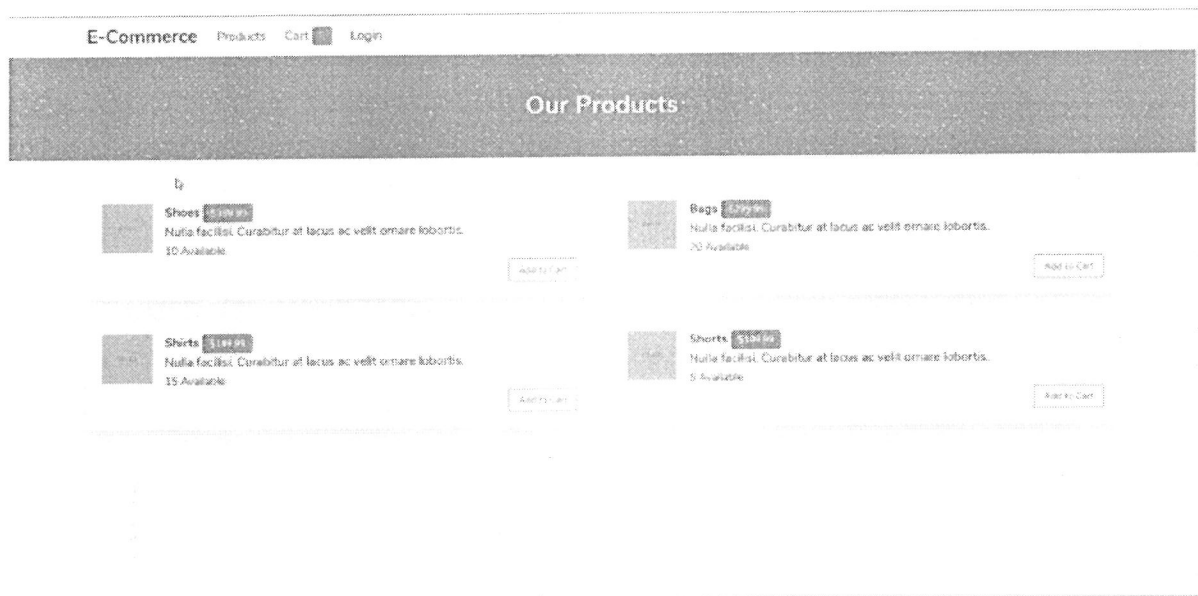
Bags 254: 90

Nulla facilisi. Curabitur at lacus ac velit ornare lobortis.
3 in cart



Clear cart

Checkout



CHAPTER 4: PROJECT DEPLOYMENT

4.1 Introduction

I am deployed to The project known as HPSC.

Which is HP Sales Central Portal.

4.2 What is HPSC?

- HP sales central portal is a portal where all the hp products are being sold.
- Each and every product of HP is being sold on this portal
- there is also one more product inside HPSC which is GIAA tool
- GIAA tool calculates the carbon quantity which is being used inside laptops and charger.

4.3 My Contribution in the project ?

- I have assigned the frontend part of the website using the ReactJs technology.

4.3 Tools and Technologies Used

CHAPTER 5: CONCLUSION

At IntimeTec Visionsoft Pvt. Ltd., the Practice School- II opportunity provided a lot of knowledge in Development technologies, DeveOps and DataOps. I can affirm that the professional experience and knowledge I acquired here would really assist my overall career. I have accomplished my internship by learning lots of new tools and technologies.

Working in a professional setting, such as IntimeTec Visionsoft Pvt. Ltd., has also taught me valuable skills such as teamwork and basic professional work ethics. The mentors here are extremely helpful and supportive.

Moving on to the project work. My contributions to the project work at the company were highly appreciated by my mentors and my seniors. Working on a live project improved my knowledge and skillsets. During this internship period.



Plot No. 1/2, Kanakpura Industrial Area, Kanakpura, Sirsi Road, Jaipur-302034
Site No.271, Sri Ganesha Complex, Hosur Main Road, BTM 1st Stage, Bangalore- 560068.

Date: 26th May 22

TO WHOMSOEVER IT MAY CONCERN

Employee Name: Akshat Awasthi

Employee ID: ITTV/EMP/2611

Address: Plot no. 1 / 2, Kanakpura Industrial Area, Kanakpura. Sirsi Road, Jaipur-302034.

This is to certify that **Mr. Akshat Awasthi** has pursued internship followed by employment with **In Time Tec Visionsoft Pvt. Ltd** from **1st Feb '22** at our **Jaipur** office located at **Plot no. 1/2, Kanakpura Industrial Area, Kanakpura. Sirsi Road, Jaipur-302034.**

During his Internship period, he was entitled to a stipend of Rs. 15,000/- per month, other details are valid as per the offer letter.

Regards,

Aayushi Jain

HR Associate

In Time Tec Visionsoft Pvt. Ltd.

Confidential

This is a Computer Generated Document. No Signature Required.