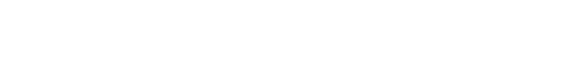
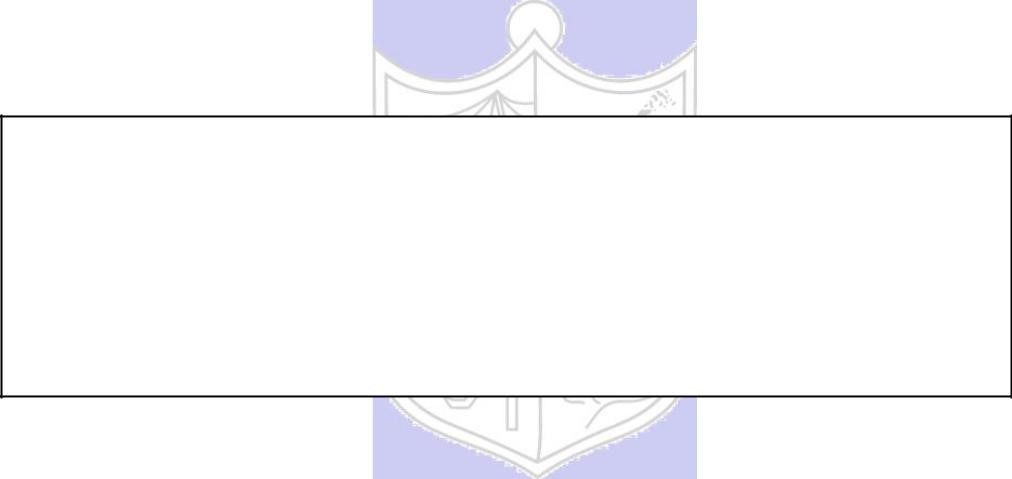
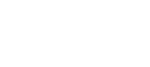
KJSCE/IT/SYBTech/SEM IV/WP



**Experiment No. 0**



**8**



**Title**



:



Design a web page using



React



JS



.



**Batch:B2** **Roll No:1914078 Experiment No:8**

**Aim:** To design a web page using React JS.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Resources needed:** Notepad, any Web Browser and Internet.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Theory:**

React (also known as React.js or ReactJS) is an open-source, front end, JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of singlepage or mobile applications. However, React is only concerned with state management and rendering that state to the DOM(Document Object Model), so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality. ReactJS is JavaScript library used for building reusable UI components.

**Features of React**

* **JSX** − JSX is JavaScript syntax extension. It isn't necessary to use JSX in React development, but it is recommended.
* **Components** − React is all about components. You need to think of everything as a component. This will help you maintain the code when working on larger scale projects.
* **Unidirectional data flow and Flux** − React implements one-way data flow which makes it easy to reason about your app. Flux is a pattern that helps keeping your data unidirectional.
* **License** − React is licensed under the Facebook Inc. Documentation is licensed under CC BY 4.0.

**ReactJS - Environment Setup**

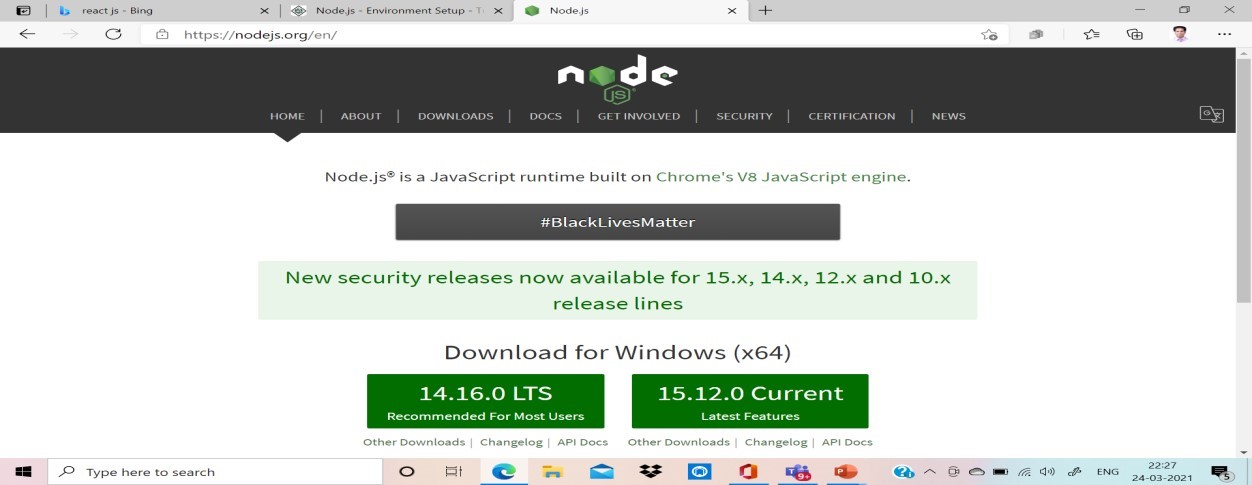
* 1. First you need to install NodeJS
  2. Second install ReactJS

**Install NodeJS:**

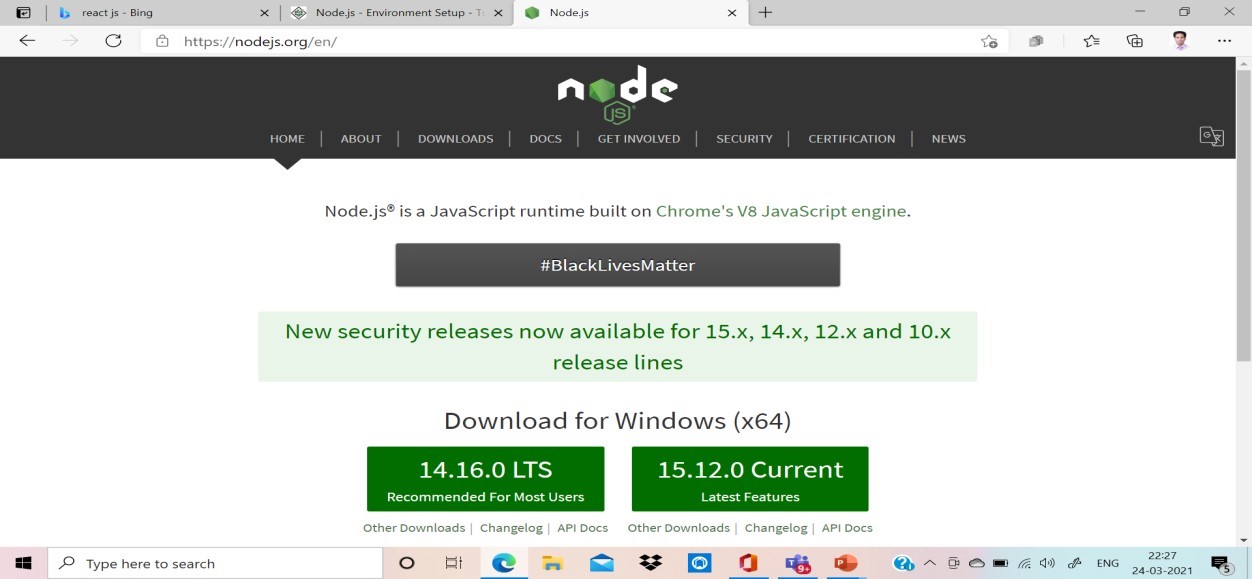
**Step 1:**

Visit the website www.Nodejs.org/en/. For installation on Windows ,you use the MSI file and follow the prompts to install the Node.js. By default, the installer uses the Node.js distribution in C:\Program Files\nodejs. The installer should set the C:\Program Files\nodejs\bin directory in window's PATH environment variable. Restart any open command prompts for the change to take effect. The source code written in source file is simply javascript. The Node.js interpreter will be used to interpret and execute your javascript code. Node.js distribution comes as a binary installable for SunOS , Linux, Mac OS X, and Windows operating systems with the 32-bit (386) and 64-bit (amd64) x86 processor architectures. Next step will guide to install Node.js binary distribution on windows OS.

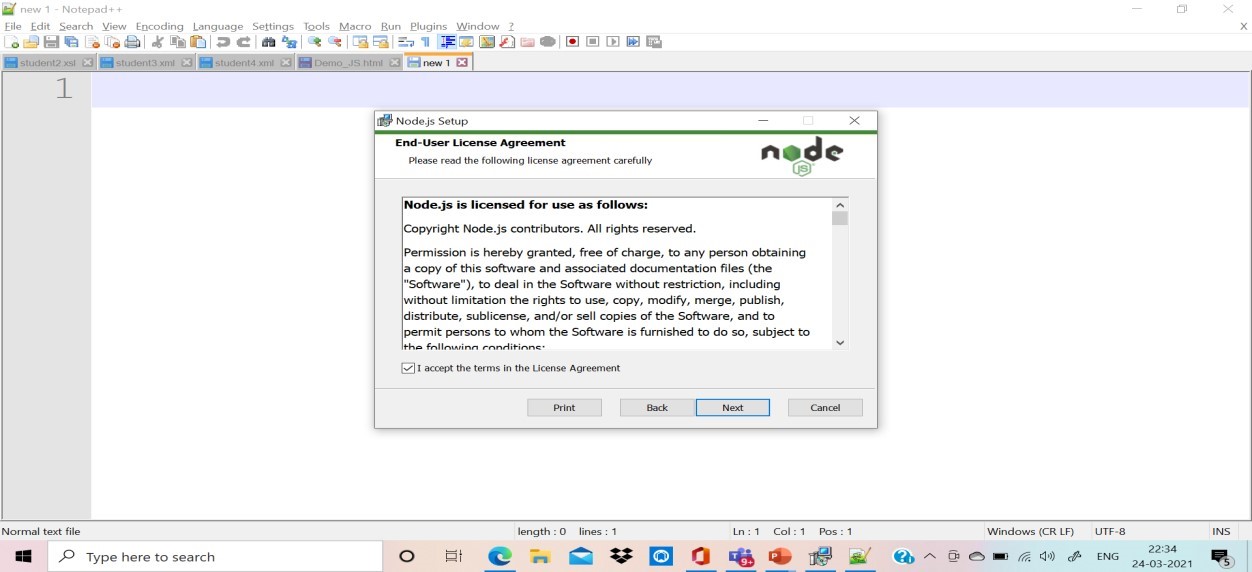
Step 2:



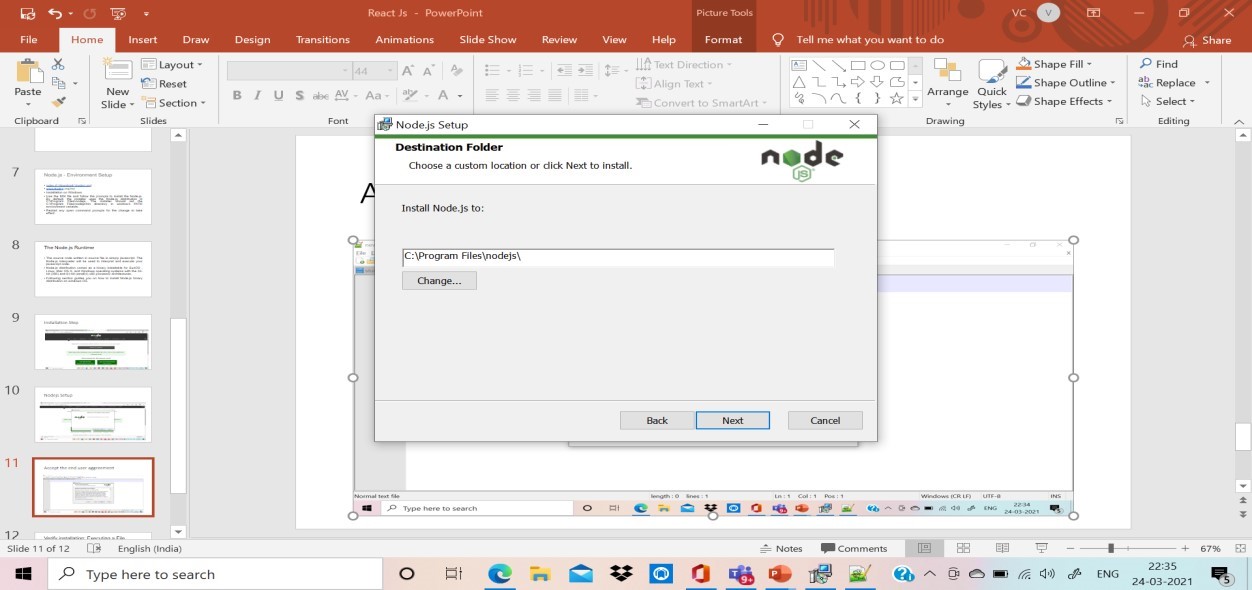
Step3:



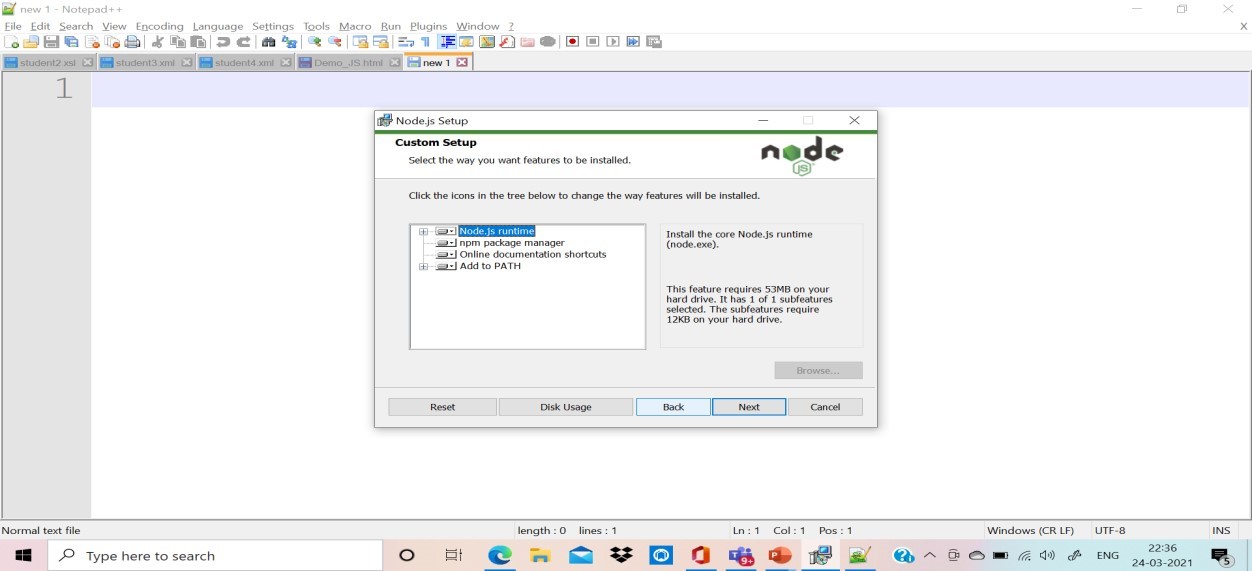
Step4: Accept the Agreement



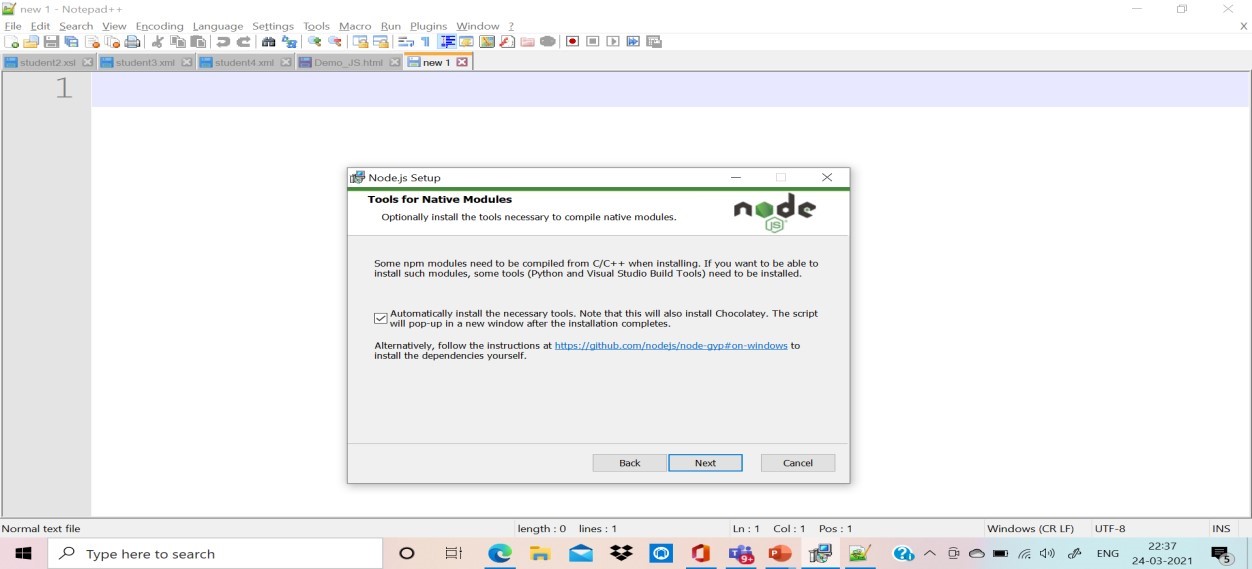
Step5: Choose Destination Folder



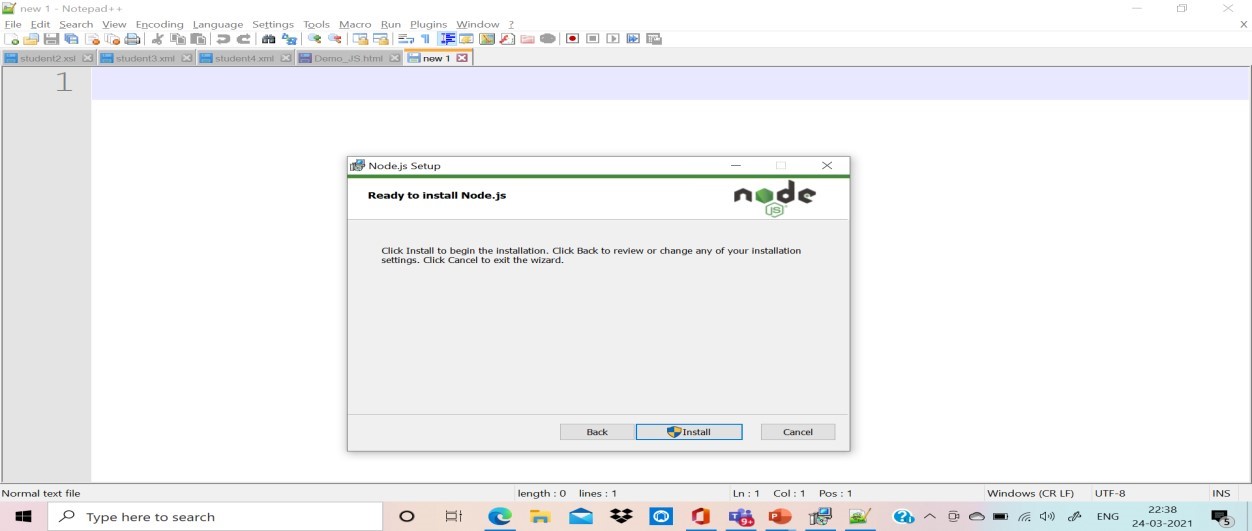
Step6: Custom Installation setup



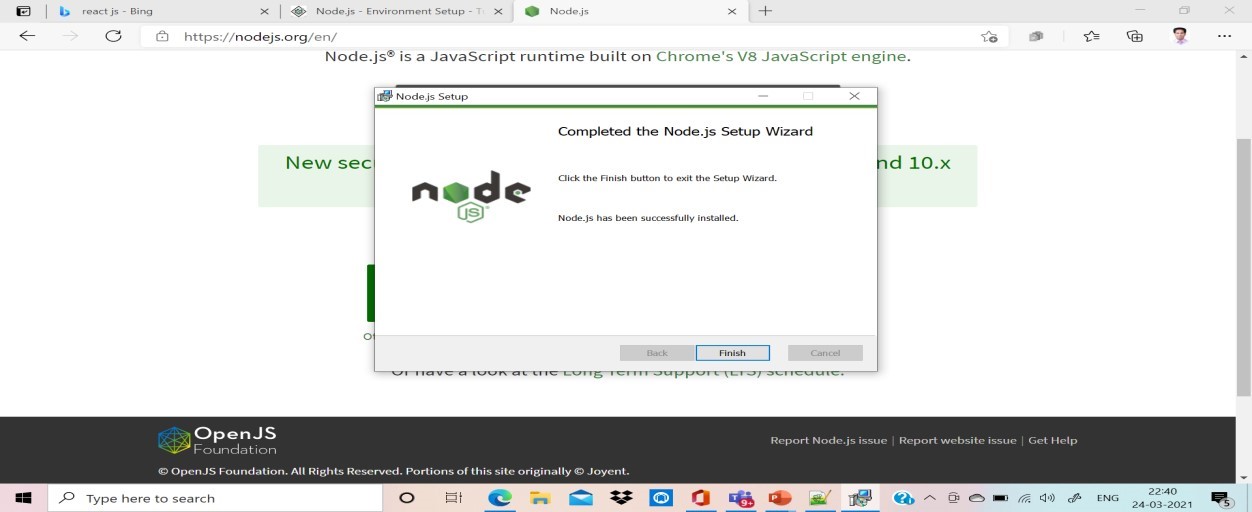
Step 7: Additional Tools Setup



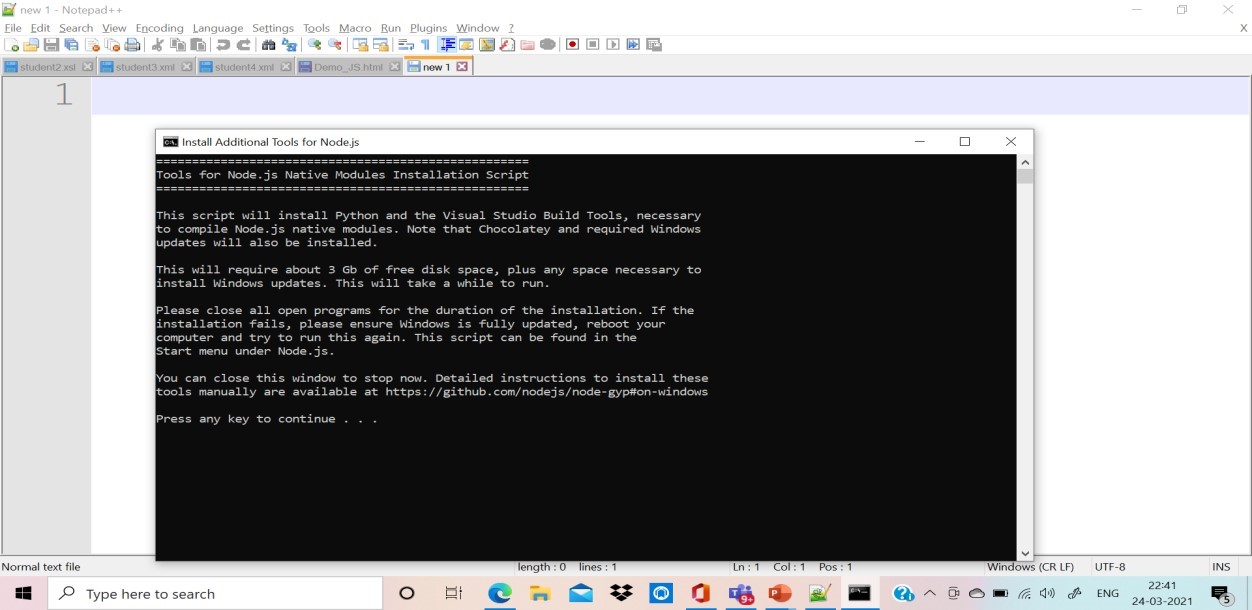
Step8: Ready to install Nodejs



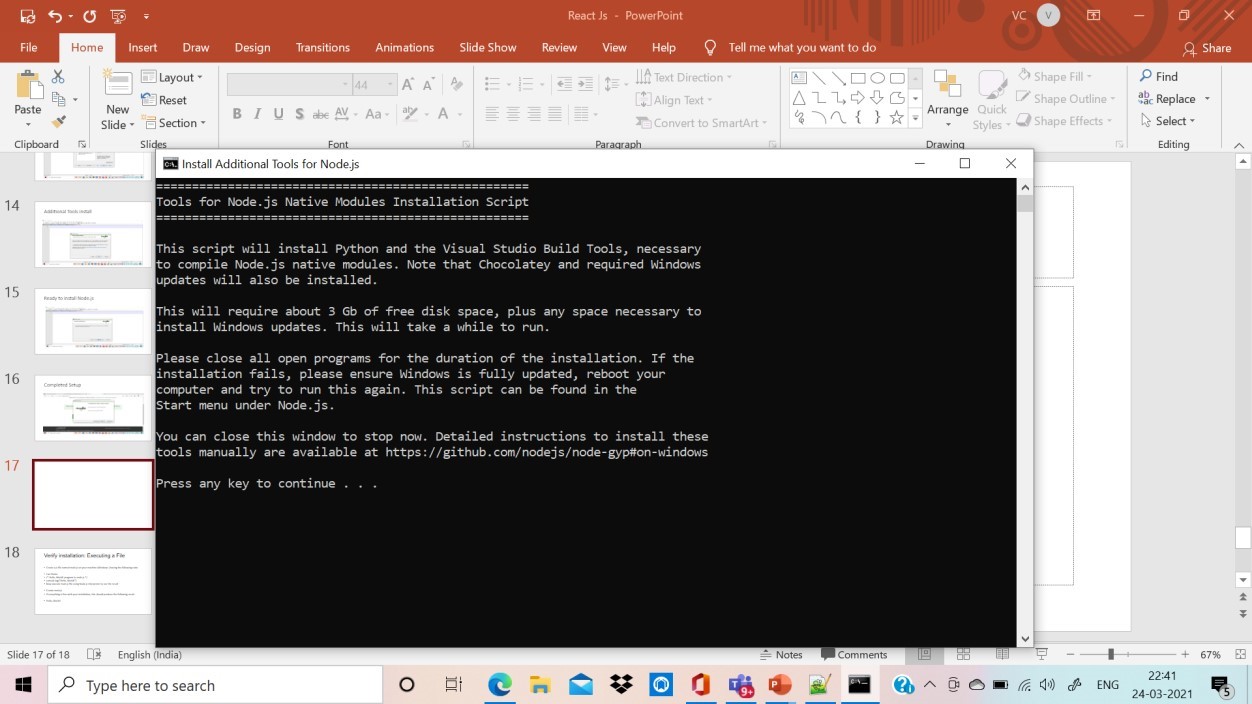
Step9: Completed Setup



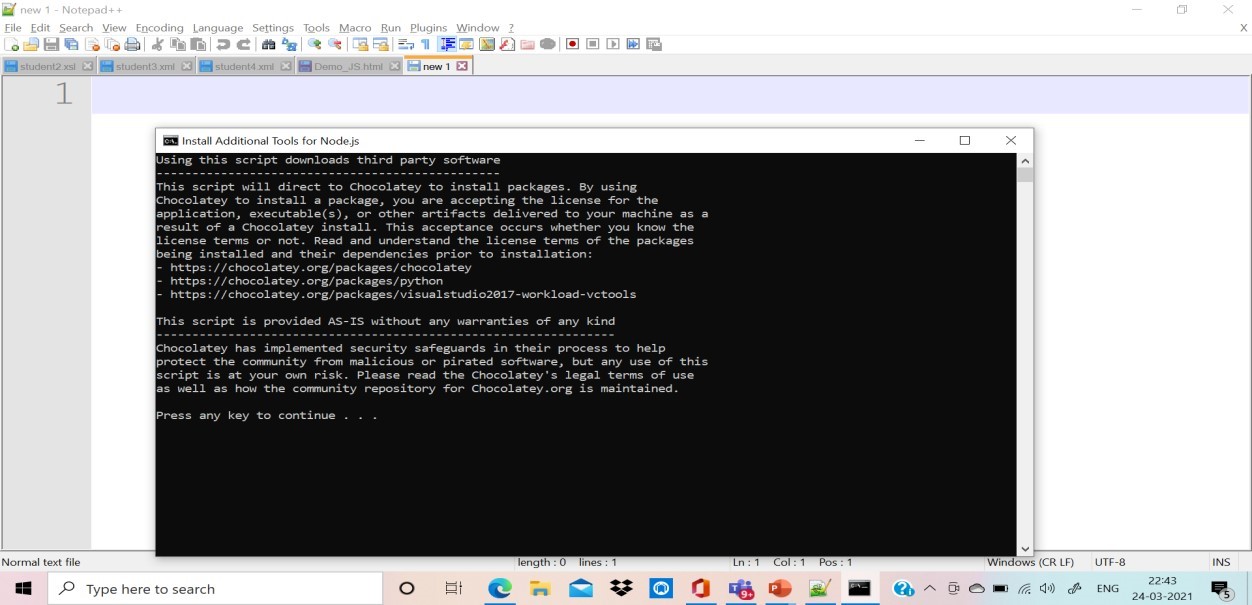
Step 10: Ready to start



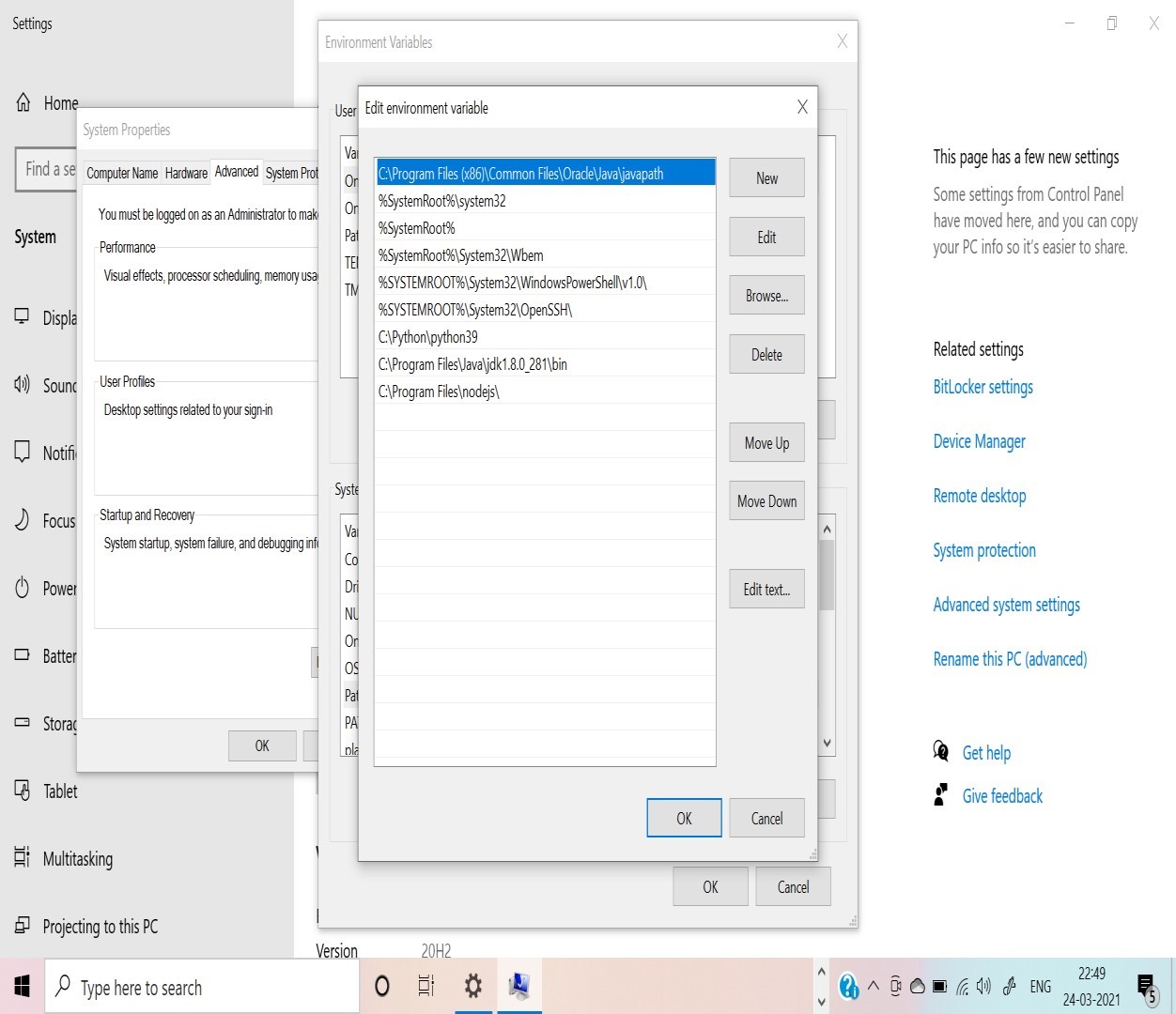
**Step11: Install Native Module**



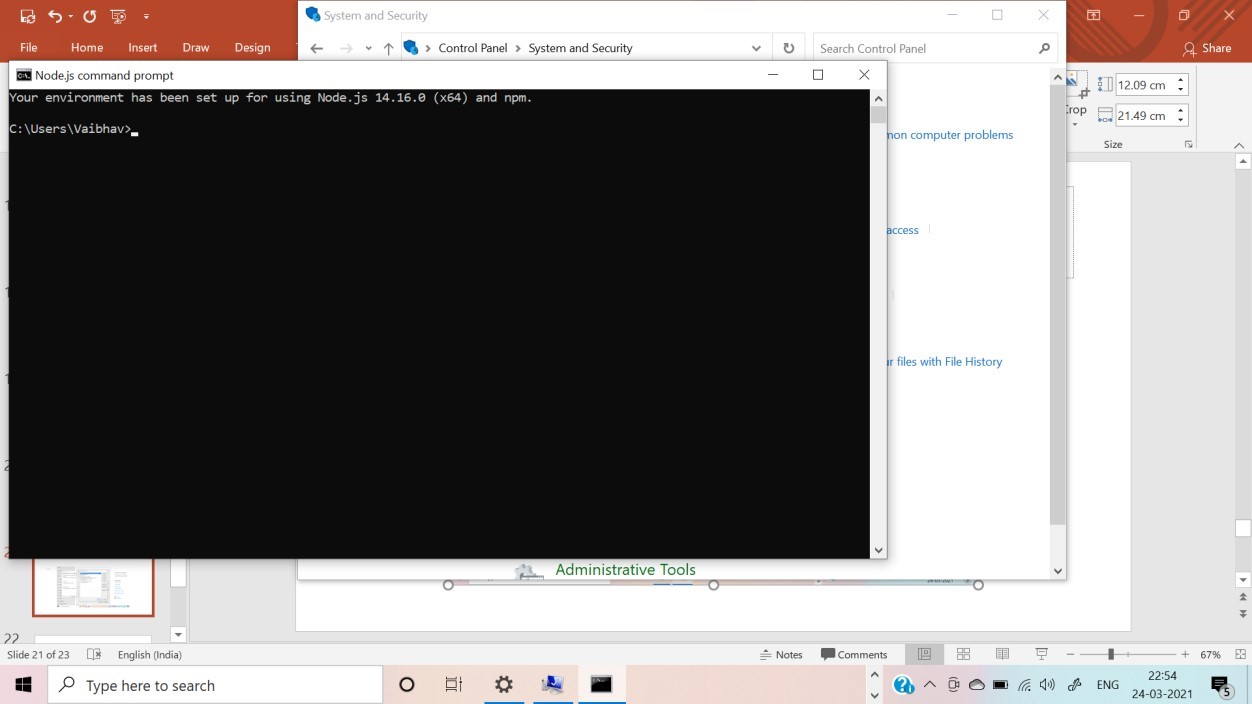
**Step 12: Additional Installation**



**Step 13: Verify the setup**



**Step 14: Setup Message on Command Prompt:** For this message go start menu and click the button you will find command prompt menu available with cmd .



**Step 15: Check the version**

* **C:\>node -v**
* **v14.16.0**

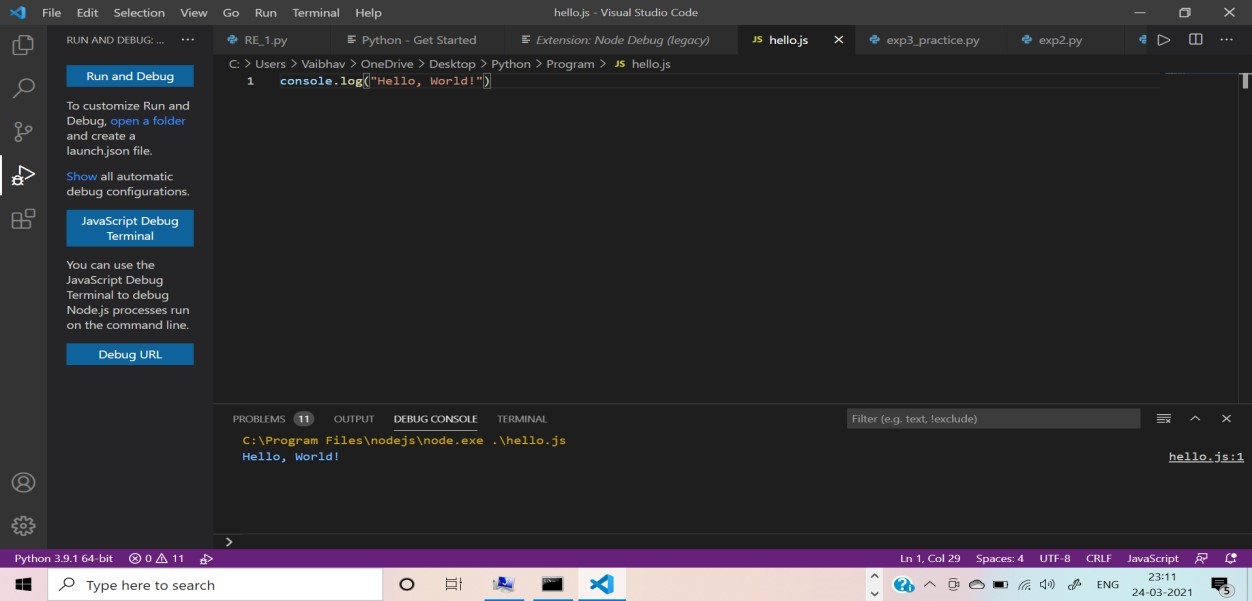
**Step 16: Type the Hello World and execute**

Type the command and install gyrescipt :

C:\>npm install -g typescript

* Create a js file named main.js on your machine (Windows ) having the following code.
* Live Demo
* /\* Hello, World! program in node.js \*/
* console.log("Hello, World!")
* Now execute main.js file using Node.js interpreter to see the result −
* $ node main.js
* If everything is fine with your installation, this should produce the following result −
* Hello, World!

Step 17: Install VS Code: Editor Tool



**Install ReactJS:**

There are 3 ways to install ReactJS :

* 1. webpack
  2. Babel
  3. create-react-app

Webpack:

**Webpack** is a module bundler (manages and loads independent modules). It takes dependent modules and compiles them to a single (file) bundle. You can use this bundle while developing apps using command line or, by configuring it using webpack.config file.

**Steps:**

1. Since we are using webpack to generate bundler install webpack, webpack-dev-server and webpack-cli.
2. C:\Users\username\Desktop\reactApp>npm install webpack --save
3. C:\Users\**username\Desktop\reactApp>npm install webpack-dev-server** --save
4. C:\Users\username\Desktop\reactApp>npm install webpack-cli --save
5. Or, you can install all of them in single command as −
6. C:\Users\username\Desktop\reactApp>npm install webpack webpack-dev-server -save

Babel:

Babel is a JavaScript compiler and transpiler. It is used to convert one source code to other. Using this you will be able to use the new ES6 features in your code where, babel converts it into plain old ES5 which can be run on all browsers.

1. Install babel, and its plugins babel-core, babel-loader, babel-preset-env, babel-presetreact and, html-webpack-plugin
2. C:\Users\username\Desktop\reactApp>npm install babel-core --save-dev
3. C:\Users\username\Desktop\reactApp>npm install babel-loader --save-dev
4. C:\Users\username\Desktop\reactApp>npm install babel-preset-env --save-dev
5. C:\Users\username\Desktop\reactApp>npm install babel-preset-react --save-dev
6. C:\Users\username\Desktop\reactApp>npm install html-webpack-plugin --save-dev
7. Or, you can install all of them in single command as −
8. C:\Users\username\Desktop\reactApp>npm install babel-core babel-loader babelpreset-env 9) babel-preset-react html-webpack-plugin --save-dev

**Working with ReactJS**

In this experiment , we are going to use this steps of create react app

Steps are as follows:

1. Create a folder with name reactApp on the desktop to install all the required files, using the mkdir command.

C:\Users\username\Desktop>mkdir reactApp

1. Change the directory:

C:\Users\username\Desktop>cd reactApp

C:\Users\vaibhav>cd C:\Users\vaibhav\reactapp\

1. Install ReactJS:

C:\Users\vaibhav\reactapp>npx create-react-app my-app

This will create a folder named my-app on the desktop and installs all the required files in it.

**npm** (node package manager) is the dependency/package manager you get out of the box when you install Node.js. It provides a way for developers to install packages both globally and locally

**npx:** The npx stands for Node Package Execute and it comes with the npm, when you installed npm above 5.2.0 version then automatically npx will installed. It is an npm package runner that can execute any package that you want from the npm registry without even installing that package. The npx is useful during a single time use package.

If you have installed npm below 5.2.0 then npx is not installed in your system.

4) Delete all source files

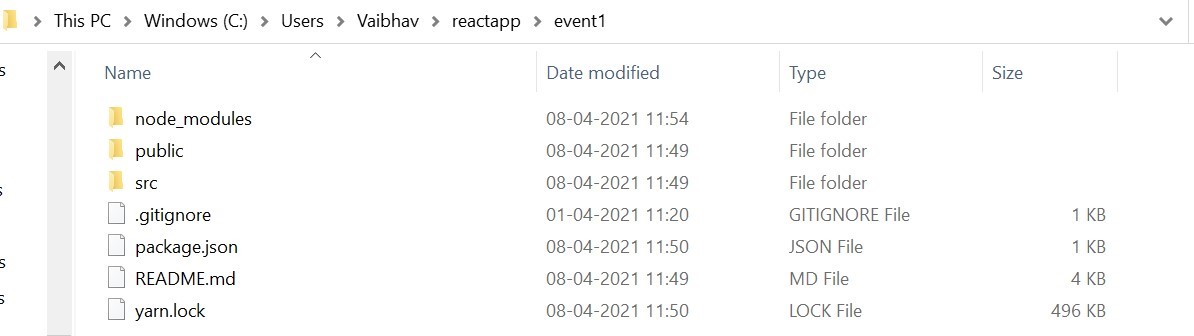
* Browse through the src folder in the generated my-app folder and remove all the files in it as shown below −
* C:\Users\Desktop>cd my-app/src
* C:\Users\Desktop\my-app\src>del \*
* C:\Users\Desktop\my-app\src\\*, Are you sure (Y/N)? y

5) Add files with names index.css and index.js in the src folder as −

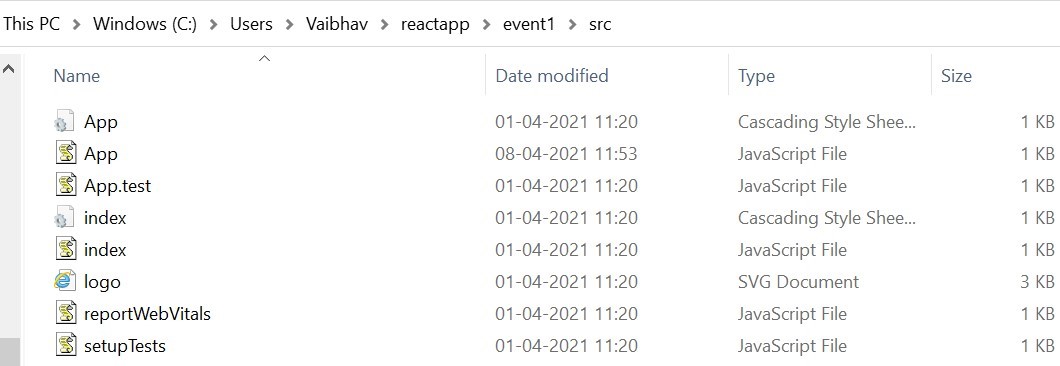
* C:\Users\Desktop\my-app\src>type nul > index.css
* C:\Users\Tutorialspoint\Desktop\my-app\src>type nul > index.js

6)Skip step 4 and 5 and only delete the file called App.js

7) Locate your code into these folder as event1 as shown below:



1. Open the src folder as shown below:



1. Choose the App.js file and indite code which is given below.

import React, {Component} from 'react'; class App extends React.Component { constructor(props) { super(props); this.state = { companyName: ''

};

}

changeText(event) { this.setState({ companyName: event.target.value

});

} render() { return (

<div>

<h2>Simple Event Example</h2>

<label htmlFor="name">Enter company name: </label>

<input type="text" id="companyName" onChange={this.changeText.bind(this)}/>

<h4>You entered: { this.state.companyName }</h4>

</div>

);

}

} export default App;

**10 .Execute the React Code:**

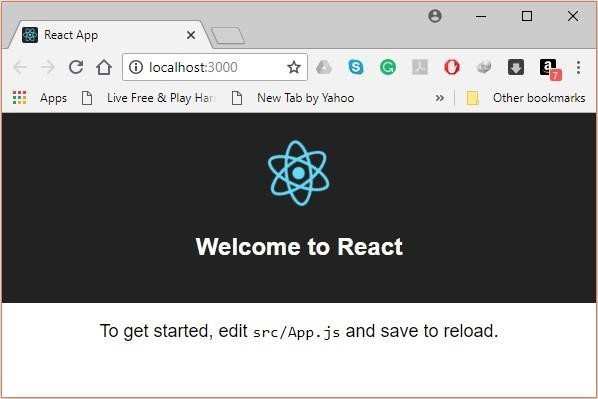
To run the react code you have to type the command on Vs Code Terminal as it is:

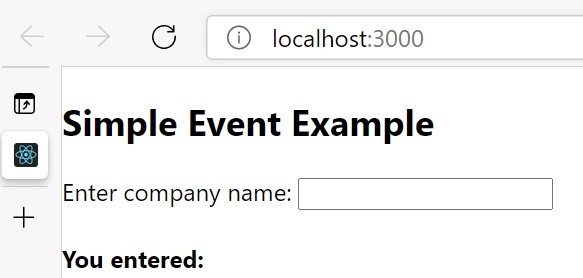
* npm start
* yarn start(if you install yarn utility)

To do this on, terminal of VS CODE,

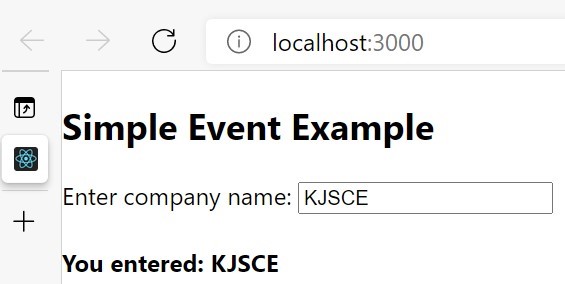
C:\Users\Vaibhav\reactapp\myevent\npm start.

By default port number 3000 will start on web browser with the code execution.





11. After running the event the outlook:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Activities:**

To design a web page using React JS on your theme to manipulate the DOM elements.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Results: (Document printout as per the format discussed by the faculty *t*)**

Display of the designed webpage along with the code.

**Code:-**

# Index.js:-

import React from 'react'; import ReactDOM from 'react-dom'; import './App.css'; import './bootstrap.min.css' class MyForm extends React.Component { constructor(props) { super(props); this.state = { fname: '', lname: '', phone: null, guest: 'S', accept: '', checked: false, fnameErr: '', lnameErr: '', guestErr: '', phoneErr: '', acceptErr: ''

};

}

myChangeHandler = (event) => { let nam = event.target.name; let val = event.target.value; let checked = event.target.checked;

let err = '';

if (nam === "fname") { if (val == "") { err = <strong>\*First Name is required</strong>;

}

this.setState({fnameErr: err});

}

if (nam === "lname") { if (val == "") {

err = <strong>\*Last Name is required</strong>;

}

this.setState({lnameErr: err});

}

if (nam === "guest") { if (val == "S") { err = <strong>\*Please choose appropriate option</strong>;

}

this.setState({guestErr: err});

}

if (nam === "phone") { if (val == "") { err = <strong>\*Phone Number is required</strong>;

} else if (val !="" && !Number(val)) { err = <strong>\*Your phone number must be a number</strong>;

} else if (val > 9999999999 || val < 7000000000) { err = <strong>\*Enter Valid Phone Number</strong>;

}

this.setState({phoneErr: err});

}

if (nam === "accept") { this.setState({[nam]: checked}); if (checked == false) { err = <strong>\*Please accept the terms and conditions</strong>;

}

this.setState({acceptErr: err});

}

this.setState({[nam]: val});

}

mySubmitHandler = (event) => { event.preventDefault(); let fname = this.state.fname;

let lname = this.state.lname; let phone = this.state.phone; let guest = this.state.guest;

let accept = this.state.accept; let err = '';

if (fname == "") { err = <strong>\*First Name is required</strong>; this.setState({fnameErr: err});

}

if (lname == "") { err = <strong>\*Last Name is required</strong>; this.setState({lnameErr: err});

}

if (guest == "S") { err = <strong>\*Please choose appropriate option</strong>; this.setState({guestErr: err});

}

if (phone == null) { err = <strong>\*Phone Number is required</strong>; this.setState({phoneErr: err}); } else if (!Number(phone)) { err = <strong>\*Your Phone Number must be a number</strong>; this.setState({phoneErr: err});

} else if (phone > 9999999999 || phone < 7000000000) { err = <strong>\*Enter Valid Phone Number</strong>; this.setState({phoneErr: err});

}

if (accept == false) { err = <strong>\*Please accept the terms and conditions</strong>; this.setState({acceptErr: err});

}

if (this.state.fnameErr == '' && this.state.lnameErr == '' && this.state.guestErr == '' &&

this.state.phoneErr == '' && this.state.acceptErr == '') { alert('Your Table is reserved');

}

}

reset = () => { let err = ''; this.setState({fnameErr: err}); this.setState({lnameErr: err}); this.setState({guestErr: err}); this.setState({phoneErr: err}); this.setState({acceptErr: err});

}

render() { return (

<div className="con d-flex flex-column justify-content-center align-items-center">

<div className="main py-5">

<center>

<h4 className="pb-

3" id="heading">Welcome {this.state.fname} {this.state.lname}!!</h4>

</center>

<div className="d-flex">

<form id="myForm" onSubmit={this.mySubmitHandler}>

<img src="https://www.salondesvinscharnay.fr/img/slider\_01.png" /><br /><br />

<div className="mb-3 d-flex">

<div className="me-3 width">

<label className="form-label" for="fname">First name:</label>

<input className="form-

control" type="text" id="fname" name="fname" onChange={this.myChangeHandler} />

</div>

<div className="width">

<label className="form-label" for="lname">Last name:</label>

<input className="form-

control" type="text" id="lname" name="lname" onChange={this.myChangeHandler} />

</div>

</div>

<div className="mb-3 d-flex">

<div className="me-3 width">

{this.state.fnameErr}

</div>

<div className="width">

{this.state.lnameErr}

</div>

</div>

<div className="mb-3">

<label className="form-label">Number of Guests:</label>

<select className="form-

select" name="guest" value={this.state.guest} onChange={this.myChangeHandler}>

<option value="S">Select</option>

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

<option value="4">5</option>

<option value="4">6</option>

<option value="4">7</option>

<option value="4">8</option>

<option value="9">9</option>

<option value="10+">10 or More</option>

</select>

{this.state.guestErr}

</div>

<div className="d-flex mb-3">

<div className="me-3 width">

<label className="form-label" for="start">Booking Date:</label> <input className="form-control" type="date" id="start" name="tripstart" value="2020-02-18" min="2020-02-18" max="2020-06-31" />

</div>

<div className="width">

<label className="form-label">Booking Time:</label>

<input className="form-

control" type="time" id="time" name="time" value="12:00" autocomplete="off" />

</div>

</div>

<div className="mb-3">

<label className="form-label" for="phone">Phone Number:</label>

<input className="form-

control" type="text" id="phone" name="phone" onChange={this.myChangeHandler} />

{this.state.phoneErr}

</div>

<div className="mb-3">

<input className="form-check-input me-

1" type="checkbox" name="accept" onChange={this.myChangeHandler} checked={this.state .check} />

<label className="form-checklabel" for="accept">I accept the terms and conditions.</label>

<br />{this.state.acceptErr}

</div>

<input className="btn btn-primary me-

3" type="submit" id="submitbtn" value="Sign Up" />

<input className="btn btn-

danger" type="reset" id="resetbtn" value="Reset" onClick={this.reset}/>

</form>

</div>

</div>

</div>

);

}

}

ReactDOM.render(<MyForm />, document.getElementById('root'));

# Css :-

.main { border: 2px solid black; padding-left: 80px; padding-right: 80px; border-radius: 30px;

} .con { height: 150vh;

} h2{ color: rgb(133, 12, 82);

} .width { width: 50%;

} img { height: 200px; width: 400px; padding-left: 20px;

}

# Index.html:-

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>React App</title>

</head>

<body>

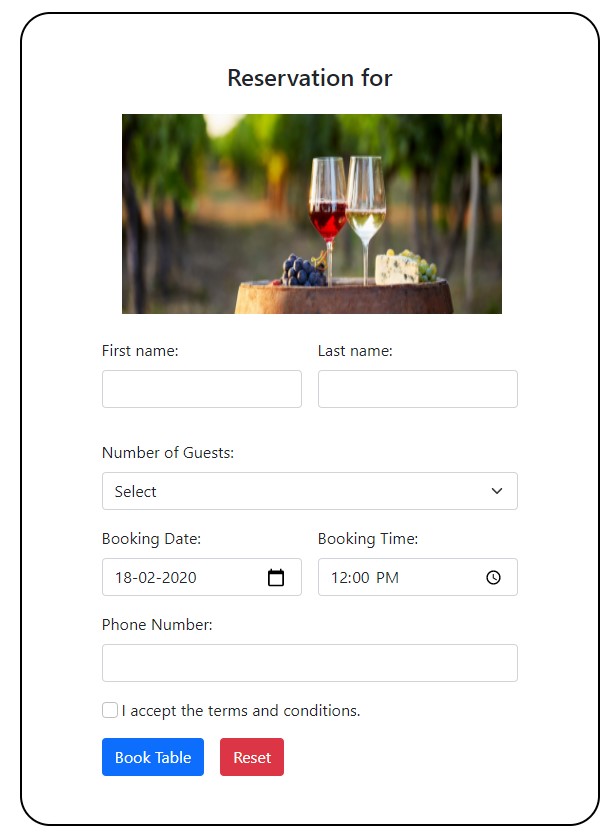
<div id="root"></div>

</body>

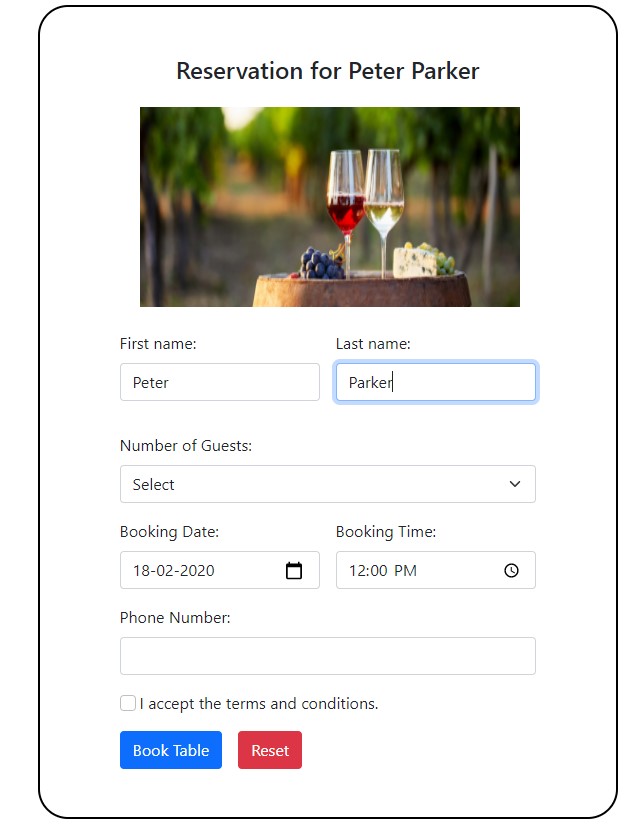
</html>

# Output:-

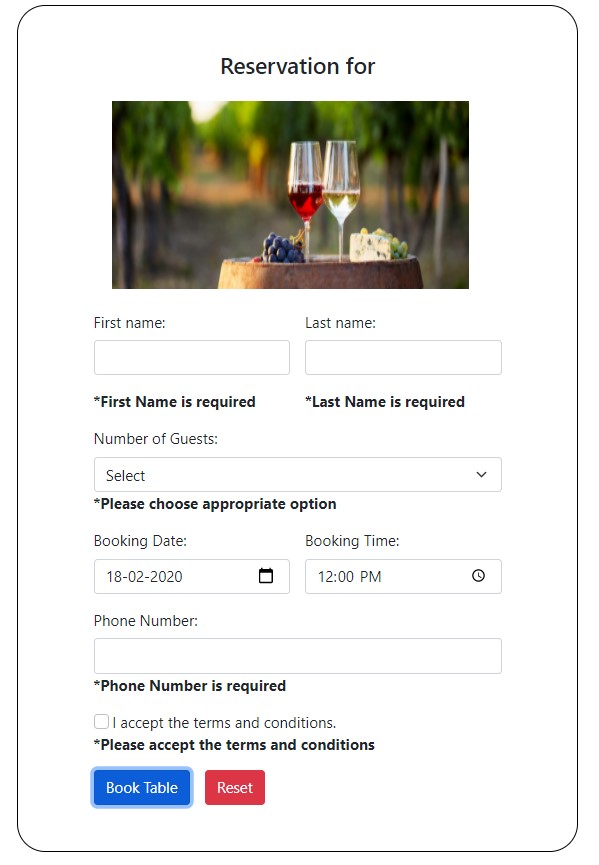
1.After the page loads, this is how the form looks :



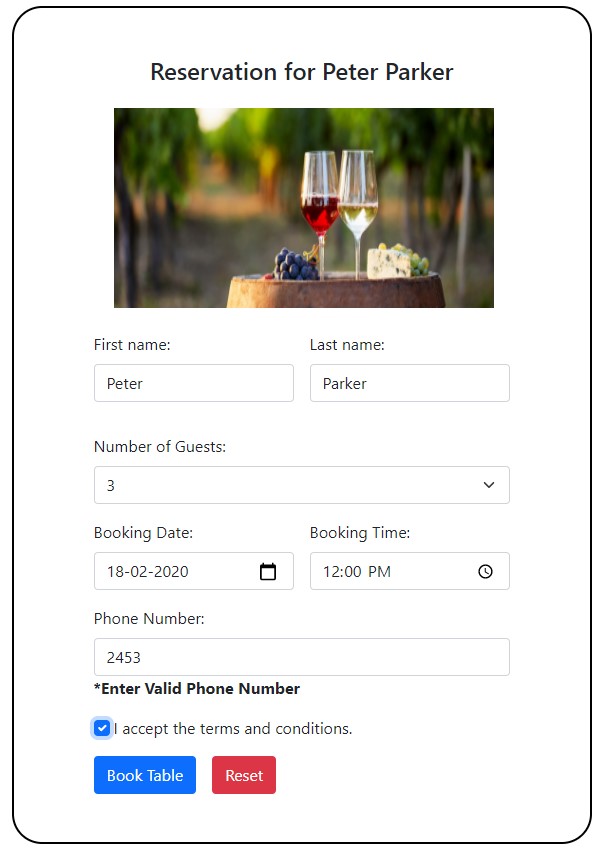
2.As soon as you enter the name, it appears after the word “Reservation for”:-



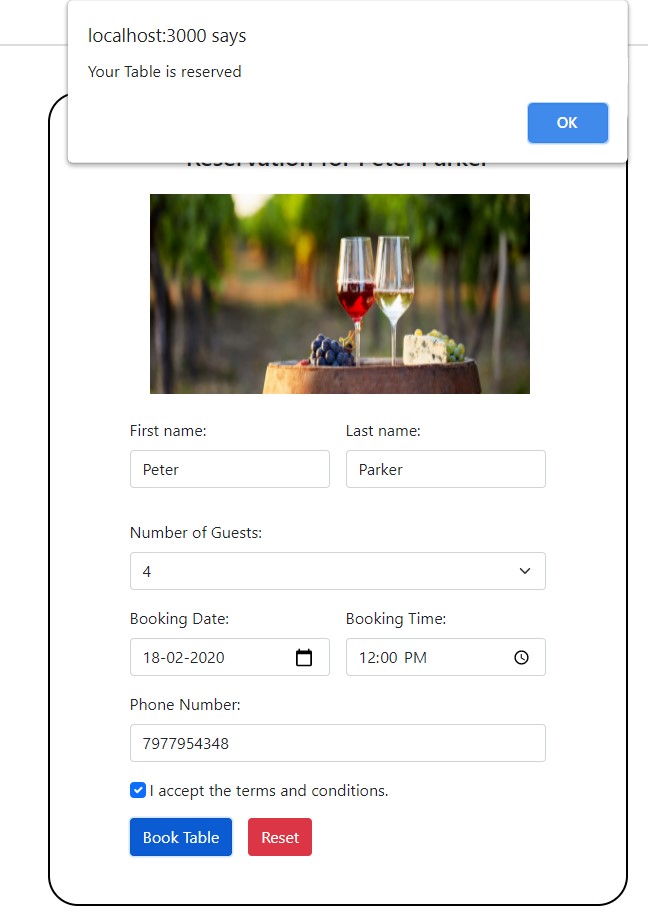
3.The following validations have been added:



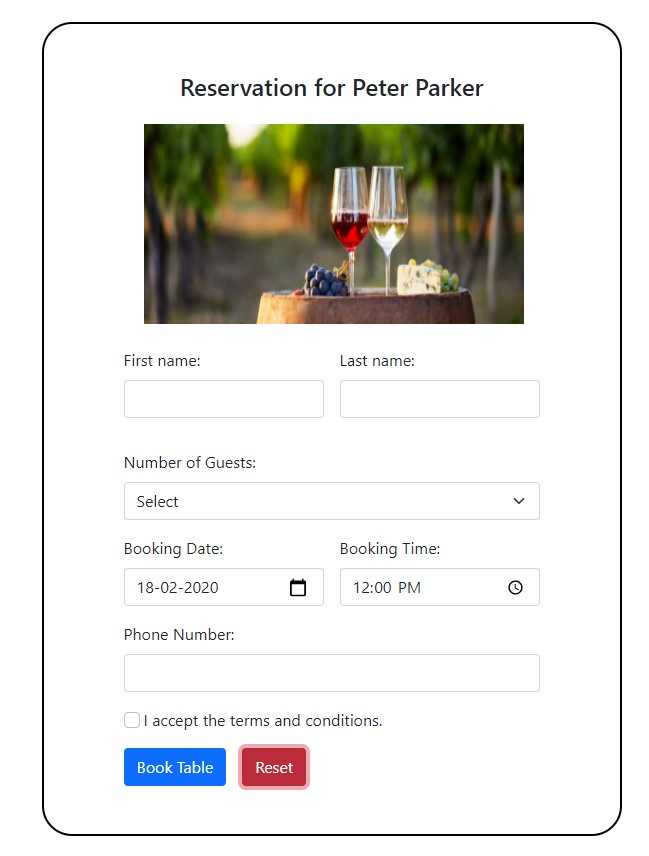
4.If you enter an invalid phonenumber (ie. One having more/less than 10 digits ,starts with wrong digit) then an error is thrown:-



5.Once you enter all the details correctly and submit the form, an alert is thrown:



6. Finally on clicking reset ,the form resets itself :-



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Questions**

**1. What are the differences between Angular JS and ReactJS with suitable example?**

**ANS.**

|  |  |  |  |
| --- | --- | --- | --- |
| **React** | | **Angular** | |
| React is a JavaScript library, and it is much older compared with Angular. | | Angular is a complete framework. | |
| React js can be packaged with other programming libraries. | | Angular is a complete solution in itself. | |
|  | |  | |
| It is based on Virtual DOM. |  | Based on MVC (Model View Controller) |  |
|  |  |
| When it comes to community support React doesn't offer much. | | It has a viable and dependable community support system | |
| React takes longer to set up. But, it is really fast for delivering projects and building apps. | | Angular is easy to set up but may lead to an increase in coding time which also results in delayed project deliveries. | |
| It gives you the freedom to choose the tools, architecture, and libraries, for developing an app. | | It offers a limited amount of freedom and flexibility. | |
| React language uses one-way data binding, which means that the Ul elements can't be changed without updating the corresponding model state. | | Angular, on the other hand, uses the two-way data binding method. It helps you to ensure that the model state automatically changes when any change is made. | |
| Some companies using ReactJS- Facebook, Uber Technologies, Instagram, Netflix, Pinterest, etc. | | Some companies using AngularJS- Wepay,  Beam, Auto Trader, Mesh, Streamline Social, etc | |

**2.What are the different components of ReactJS?**

**ANS.**

1. **Functional Components:**

By writing a javascript function, we can create a functional component in React Apps. To make React app efficient, we use functional component only when we are sure that our component does not require to interact with any other components. Functional components do not require data from other components.

1. **Class Components**

The class components are similar to the functional component but has some additional features that makes class component a little more complex than the functional components.We can pass data from one class component to other class component.

1. **Higher-Order Components**

Higher order components, or known under the abbreviation HOCs are the component which takes a component as input and returns the component as output but with extended functionalities.

1. **Dumb Components**

A Dumb Component can very easily be defined as a stateless functional component. A stateless component is much more efficient than a stateful one, because it doesn't require as many computer resources to render (memory, CPU, and GPU in terms of graphic-intensive apps).

**5] Smart Components**

A Smart Component is any class component that manages its own state. Smart Components are stateful components and works similar to class components. When working with Babel or ES6style React, we’ve come to know this as any class-like object that extends Component.This includes either React.Component or in our case Other.Component.

1. **Presentational Components**

The presentation component is often called as stateless functional component that takes props and renders UI. A stateless functional components are plain JavaScript functions that do not have states. The components that receive state from the higher-order component will function as presentational components. State gets passed to them and they conditionally render UI based on it. They do not bother with the management of state. Presentational Component mainly concerned with how things look.

1. **Container components**

Container component is a class component that provides the data and behavior to presentational or other container components. A container does data fetching and then renders its corresponding sub-component. This component mainly concerned with how things work. Container components call flux actions and provides these as callbackss to the presentational component.

**3.What is Virtual DOM? How virtual DOM Works? What is the purpose of render of react DOM?**

**ANS.** **DOM** stands for “Document Object Model”. The DOM in simple words represents the UI of your application. Everytime there is a change in the state of your application UI, the DOM gets updated to represent that change.

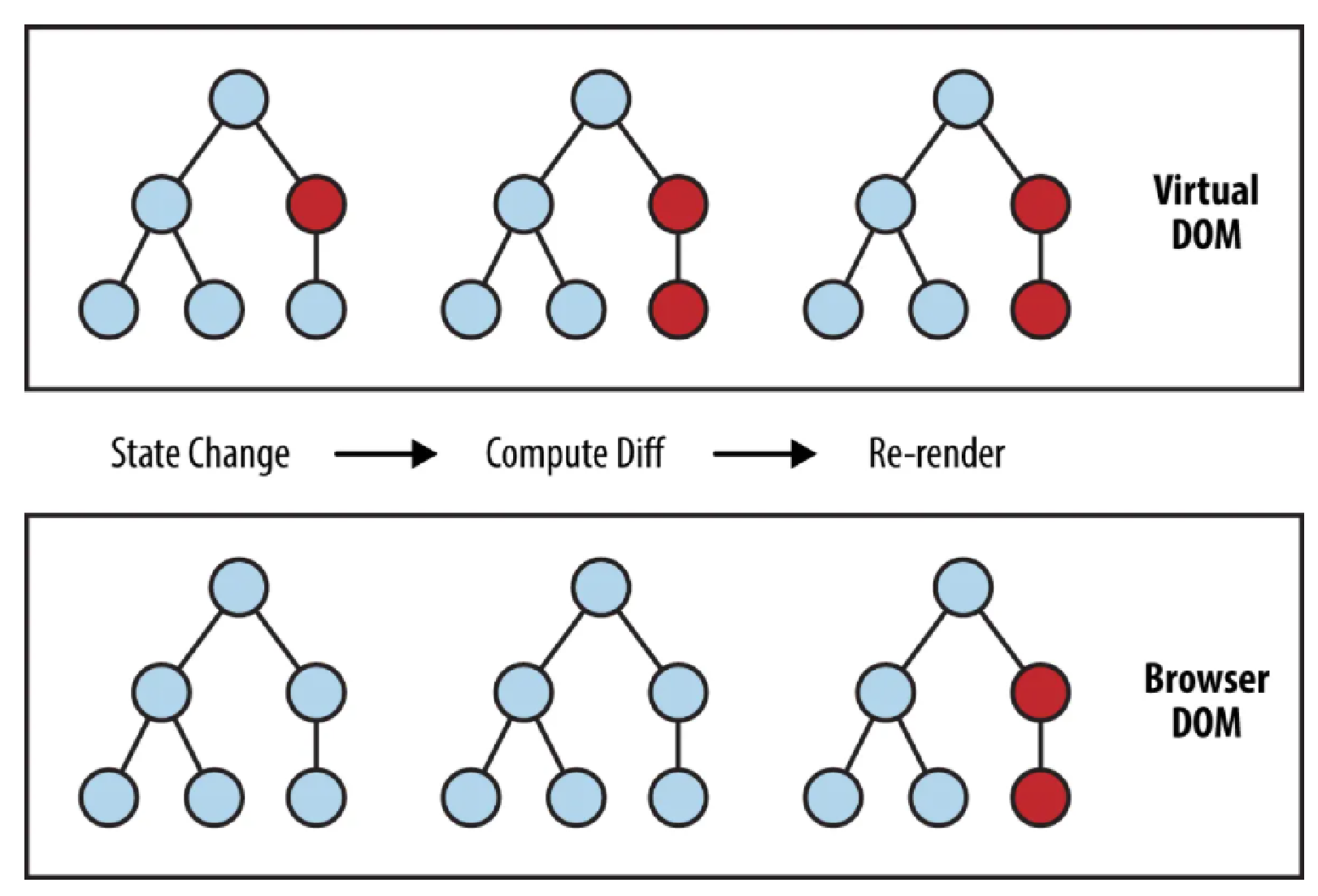
**The virtual DOM** is only a virtual representation of the DOM. Everytime the state of our application changes, the virtual DOM gets updated instead of the real DOM.

**How virtual DOM Works?Heres how:**

When new elements are added to the UI, a virtual DOM, which is represented as a tree is created. Each element is a node on this tree. If the state of any of these elements changes, a new virtual DOM tree is created. This tree is then compared or “diffed” with the previous virtual DOM tree.

Once this is done, the virtual DOM calculates the best possible method to make these changes to the real DOM. This ensures that there are minimal operations on the real DOM. Hence, reducing the performance cost of updating the real DOM.

The image below shows the virtual DOM tree and the diffing process.



The red circles represent the nodes that have changed. These nodes represent the UI elements that have had their state changed. The difference between the previous version of the virtual DOM tree and the current virtual DOM tree is then calculated. The whole parent subtree then gets re-rendered to give the updated UI. This updated tree is then batch updated to the real

DOM.

**Purpose of render of react DOM:**

*render()* function is the point of entry where the tree of React elements are created. When a *state* or *prop* within the component is updated, the *render()* will return a different tree of React elements. If you use *setState()* within the component, React immediately detects the state change and rerenders the component.

React then figures out how to efficiently update the UI to match the most recent tree changes.

This is when React updates its virtual DOM first and updates only the object that have changed in the real

DOM.

**4.What is the difference between traditional client server request response architecture and angular JS client server request response architecture. ANS.**

**Traditional client server request response architecture**

Most web applications are developed by separating its main functions into layers, or tiers. This allows you to easily replace and upgrade each layer independently. This architectural pattern is called Multi- or Three-Tier Architecture.

**Presentation layer**

The presentation layer is accessible to users via a browser and consists of user interface components and UI process components that support interaction with the system. It’s developed using three core technologies: HTML, CSS, and JavaScript. While HTML is the code that determines what your website will contain, CSS controls how it will look. JavaScript and its frameworks make your website interactive – responsive to a user’s actions. Developers use JavaScript frameworks such as Angular and React to make the content on the page dynamic.

**Business layer**

This layer, also called Business Logic or Domain Logic or Application Layer, accepts user requests from the browser, processes them, and determines the routes through which the data will be accessed. The workflows by which the data and requests travel through the back end are encoded in a business layer. For example, if your application is a hotel booking website, business logic will be responsible for the sequence of events a traveler will go through when booking a room.

**Persistence layer**

Also called the storage or data access layer, the persistance layer is a centralized location that receives all data calls and provides access to the persistent storage of an application. The persistence layer is closely connected to the business layer, so the logic knows which database to talk to and the data retrieving process is more optimized.

The data storage infrastructure includes a server and a Database Management System, to communicate with the database itself, applications, and user interfaces to obtain data and parse it.

**Angular JS client server request response architecture:**

AngularJS is built upon the MVC design pattern. The principles behind the MVC architecture are very well incorporated in AngularJS. One might have known MVC to be a robust architecture for many server-side languages. AngularJS amalgamated the MVC pattern on the client-side as well.

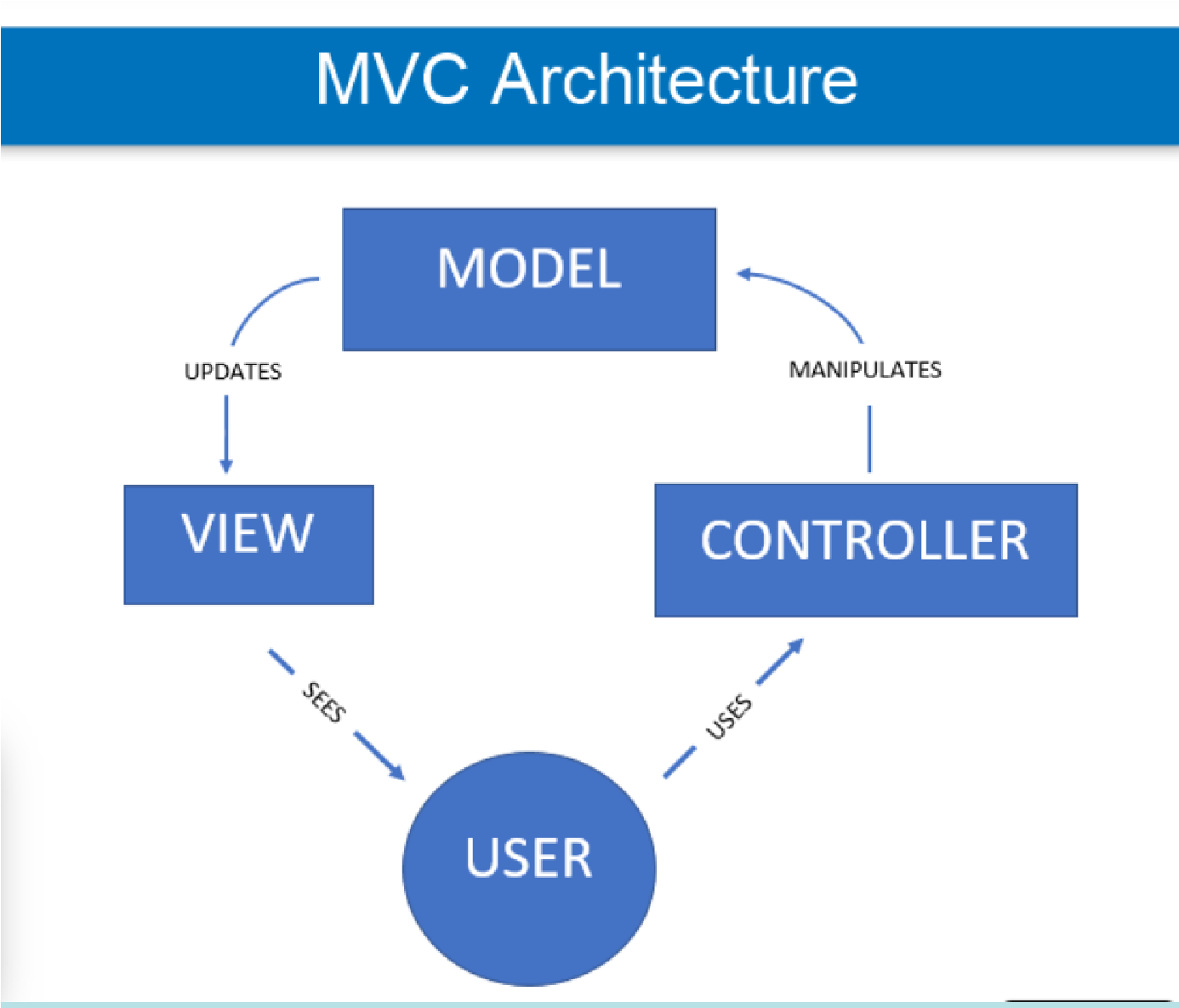
**MVC – Model, View, Controller**

MVC pattern stands for Model View Controller pattern.

* **Model:** It is the lowest level of the architecture. It is responsible for holding and maintaining the application data. The data is maintained throughout the lifecycle of the page and sometimes even between pages. The model updates itself based on the instructions received from the controller.
* **View:** A view is the front-face of the application. It is the presentation layer of the architecture responsible for displaying the data to the user. It holds the complete code for the UI pages in any browser-compatible language, usually HTML. The View is triggered by its respective controller. A view sends requests to its controller based on user interaction with the application. The controller then regenerates the view based on the response received from the server.
* **Controller:** A controller is the processing brain behind the view and the model. It takes the decisions to generate, regenerate or destroy the view and the model. All the business operations

and code logics are written inside the controller. (Some developers prefer writing the business logic in the Model itself). The controller is also responsible for sending requests to the server and receiving a response. It then updates the View and Model based on the response. In short, the controller controls everything.

The MVC architecture can be graphically represented through the below image.



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Outcomes:**

CO3: Implement web application using React JS, Angular JS, Json and CBOR

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Conclusion:**

**(Conclusion to be based on objectives and outcomes achieved)**

In this experiment, we learned to install ReactJS and design a web page using React JS on our theme by manipulating the DOM elements.

**Grade: AA/AB/BB/BC/CC/CD/DD/FF**

**Signature of faculty in-charge with date**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**References: Books/ Journals/ Websites:**

1. React – A JavaScript library for building user interfaces (reactjs.org)
2. "React - A JavaScript library for building user interfaces"*. React*. Retrieved 7 April 2018*.*
3. *Krill, Paul (May 15, 2014).* "React: Making faster, smoother UIs for data-driven Web apps"*.* InfoWorld*.*
4. *Hemel, Zef (June 3, 2013).* "Facebook's React JavaScript User Interfaces Library Receives Mixed Reviews"*. InfoQ.*
5. *Dawson, Chris (July 25, 2014).* "JavaScript's History and How it Led To ReactJS"*. The New Stack.*
6. *Dere, Mohan (2018-02-19).* "How to integrate create-react-app with all the libraries you need to make a great app"*. freeCodeCamp*. Retrieved 2018-06-14*.*
7. React Tutorial (w3schools.com)
8. ReactJS - Overview - Tutorialspoint
9. ReactJS Tutorials - GeeksforGeeks