REACT JS v16.0

Copyright © 2011 IGATE Corporation (a part of Capegemini Group). All rights reserved. No part of this publication shall be reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior written permission of IGATE Corporation (a part of Capegemini Group).

IGATE Corporation (a part of Capegemini Group) considers information included in this document to be Confidential and Proprietary.

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Revision No. | Author | **Summary of Changes** |
| March-2019 | 1.0 | Kathiresan.N | Lab book exercises created |

Table of Contents

[Table of Contents 3](#_Toc390968894)

[Getting Started 4](#_Toc390968895)

[Overview 4](#_Toc390968896)

[Setup Checklist for React dependancies & JEST dependancies 4](#_Toc390968897)

[Minimum System Requirements 4](#_Toc390968898)

[Creating the first React application: 4](#_Toc390968899)

[Lab 1. Creating application using webpack & Babel 5](#_Toc390968900)

[Lab 2. Creating a application demonstrating JSX 9](#_Toc390968901)

[Lab 3.1. Creating a application demonstrating State & Props 14](#_Toc390968902)

[Lab 3.2. Creating a application using nested Components 14](#_Toc390968902)

[Lab 3.3 Creating a application demonstrating List & Keys 14](#_Toc390968902)

[Lab 3.4 Creating a application using functional & class Components 14](#_Toc390968902)

[Lab 4.1 Create a React appllication rendering login page apply styles using inline styles 14](#_Toc390968902)

[Lab 4.2 Create a React appllication rendering login page apply styles using radium styles 14](#_Toc390968902)

[Lab 4.3 Creating a React application rendering register page ,apply styles using bootstrap 14](#_Toc390968902)

[Lab 4.4 Creating a React application rendering register page ,apply styles using reactstrap 14](#_Toc390968902)

[Lab 4.5 Creating a React application rendering register page ,apply styles using semanticUI 14](#_Toc390968902)

[Lab 5.1 Creating a React application & debug using google developer tool & react dev tool 14](#_Toc390968902)

[Lab 5.2 Creating a React application Handling errors using error boundaries 14](#_Toc390968902)

[Lab 6.0 Creating a React application demonstrating React Lifecycle 14](#_Toc390968902)

[Lab 7.1 Creating a React application and display content using Fragments 14](#_Toc390968902)

[Lab 7.2 Creating a React application demonstrating HOC and validation of PROPS 14](#_Toc390968902)

[Lab 7.3 Creating a React application for usage of context API 14](#_Toc390968902)

[Lab 7.4 Creating a React application Managing CRUD operations for Inventory using AXIOS 14](#_Toc390968902)

[Lab 8.0 Creating a React Router application convert 7.4 exercise (use switch,with router) 14](#_Toc390968902)

[Lab 9.1 Creating a React application for display a registration form and store in DB (validation to be done ) 14](#_Toc390968902)

[Appendices 15](#_Toc390968905)

[Appendix A: Class Diagrams 15](#_Toc390968906)

Getting Started

## Overview

This lab book is a guided tour for learning Basic Spring 4.0. It comprises solved examples and ‘To Do’ assignments. Follow the steps provided in the solved examples and work out the ‘To Do’ assignments given.

## Setup Checklist for React

Here is what is expected on your machine in order for the lab assignments to work.

## Minimum System Requirements

* Intel Pentium IV or higher
* Microsoft Windows (NT 4.0/XP/2K)
* Memory: 256MB of RAM (512 recommended)
* 500MB hard disk space
* Node JS v6.11.0 or 8 or 10
* Create-react-app 1.5.2(other plugins are pre-installed)
* VS Code

**Guidelines for all project below**

## Creating the first React application:

* Create a folder with your project name
* Ensure that node\_module is copied and pasted into your project folder
* Ensure you created a package.json file and src folder ,inside it

Lab 1: Creating application using webpack & Babel

|  |  |
| --- | --- |
| **Goals** | * Create a hello-world app using web pack and Babel |
| **Time** | 60 minutes |

* **Problem statement-1 : create a hello-world App**  using react, use the above guide lines
* **Problem statement-2.1 :**  **render,attribute expression and conditional rendering**

Create a react app with name **“React-Basic-App”. Follow the guidelines stated above and create the project.**

1. **Create a folder “React-Basic-App” and create packages.json and src folder in it**

Follow the PPT for the above demo React Application Using WebPack 4 & Babel.ppt

Lab 2: Creating a application demonstrating JSX

Lab 2.1 :

|  |  |
| --- | --- |
| **Goals** | * Create a basic react app and render data using JSX |
| **Time** | 30 minutes |

**Problem statement-2.1 :**  **Create a component and render using JSX**

Like previous demo create a App with name (ie folder) **“react-jsx”**

Ensure you have created package.json, copy the node\_modules and src folder is created

1. Create a file with name index.js and App.js

In App.js , display 2 text boxes using JSX in the browser

Lab 3: Props & State

Lab 3.1 :

|  |  |
| --- | --- |
| **Goals** | * Create a basic react app using React state & Props |
| **Time** | 20 minutes |

**Problem statement-:**  **Create a React application with name “react-create-props-movie-list”.Use props ,Store data in props and display the data in index.js , through render method.**

**Guidelines:**

1. **Create 2 js files with Name index.js, MovieList.js**
2. **Index.js is going to render the final output**

**Pass the ticket price and name of movie through the tag given in index.js ie MovieList tag.**

**Get those values from props store it in state and display it in MovieList.js render method.**

**(pass movie name as default and price as 80)**

1. **Display a button using MovieList.js with label “click to change”. On clicking this button it has to**

**Invoke a function called changeInfo() and change the value of Movie Name to ‘Avengers and price to**

**180**

Lab 3.2 :

|  |  |
| --- | --- |
| **Goals** | Creating a application using nested Components |
| **Time** | 10 minutes |

**Problem statement-3.2 :**  Create a React application with name “react-nested-components”.

Create Two Components Named  **“Fruits” and “FruitType”.**

**In FruitType.js**

**Return below content**

<h2>Fruits:</h2>

<ul>

<li>Apples</li>

<li>Blueberries</li>

<li>Strawberries</li>

<li>Bananas</li>

</ul>

**And render this using Fruit.js and tag should be rendered in browser using index.js only**

**Lab 3.3 : Creating a application demonstrating List & Keys**

|  |  |
| --- | --- |
| **Goals** | Creating a application demonstrating List & Keys |
| **Time** | 60 minutes |

**Problem statement-3.3 :**  **Create a React application with name “react-create-List”. Use lab 3.2 display the content displayed in FruitType using List and Keys. Where Keys should be auto numbered from 1.**

**Lab 3.4 : Creating a application using functional & class Components**

|  |  |
| --- | --- |
| **Goals** | Creating a application using functional & class Components |
| **Time** | 60 minutes |

**Problem statement-3.4 :**  **Create a React application with name “react-create-functional-component”. Use lab 2.1 & 3.1 by using**

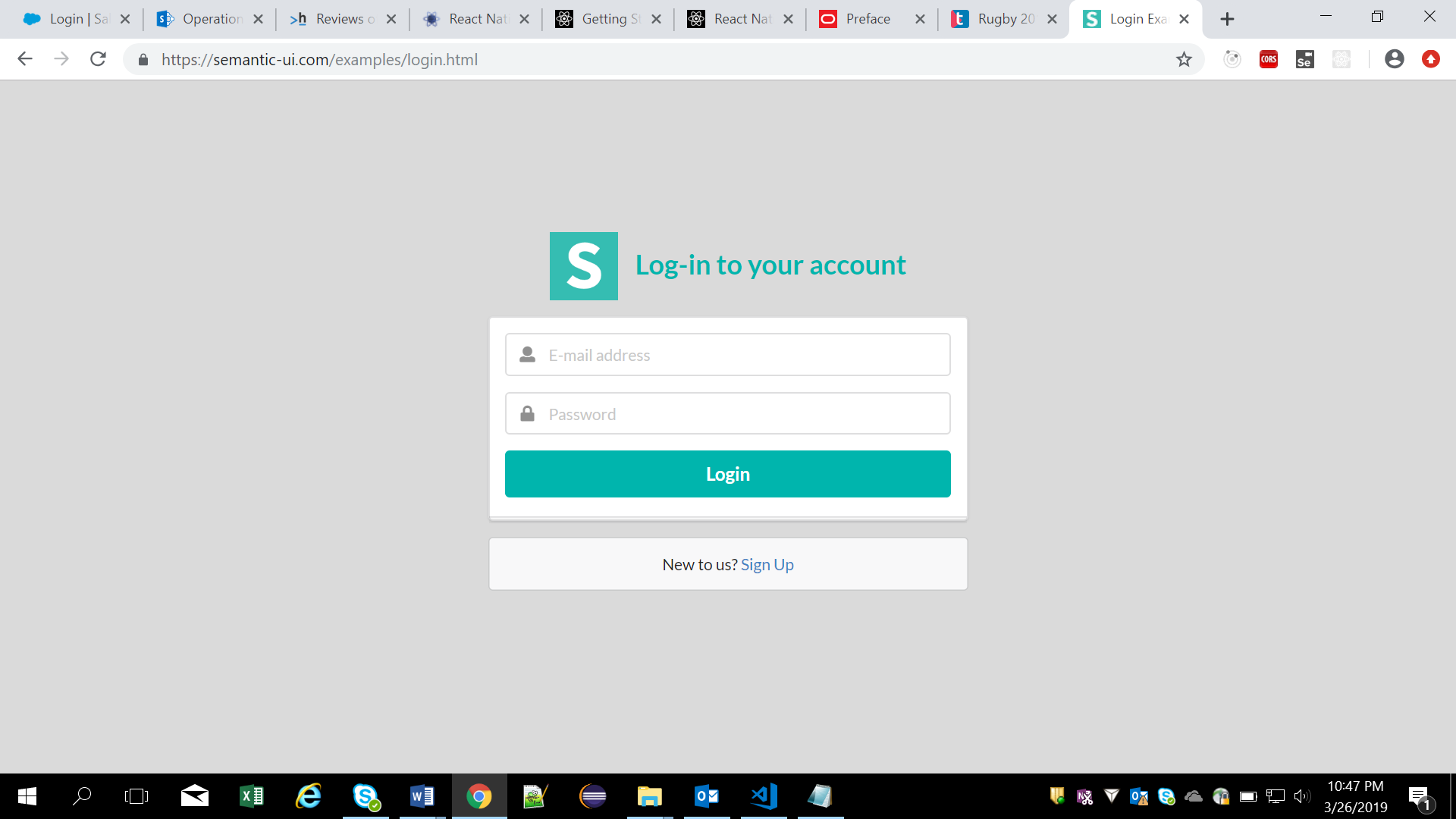
**Functional component representation and also Class Component.**

Lab 4: Styling

**Lab 4.1 : Create a React appllication rendering login page apply styles using inline styles**

|  |  |
| --- | --- |
| **Goals** | Create a React appllication rendering login page apply styles using inline styles |
| **Time** | 60 minutes |

**Problem statement-3.5 :**  **Create a React application with name “react-create-login”. Display login form as shown below**



**Lab 4.2 : Create a React application including 3.5 demo , add nav bar to it. On keeping mouse over the menus the color of the menus should change using radium styles**

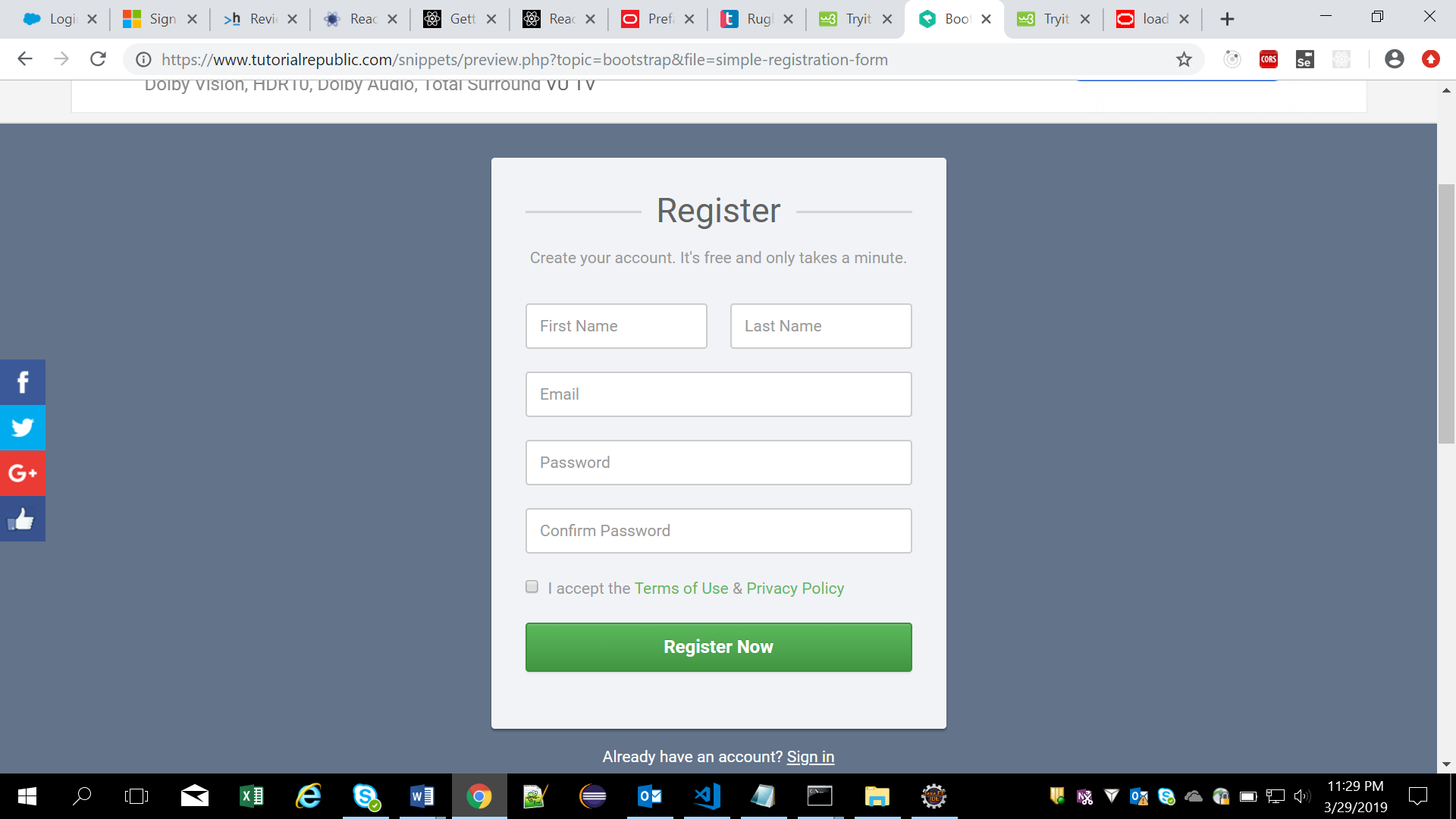
|  |  |
| --- | --- |
| **Goals** | Create a React appllication rendering login page apply styles using radium styles |
| **Time** | 60 minutes |

**Problem statement-3.6 :**  **Create a React application with name “react-create-login”. Display login form as shown below**

**Lab 4.3 : Creating a React application rendering register page ,apply styles using bootstrap**

|  |  |
| --- | --- |
| **Goals** | Creating a React application rendering register page ,apply styles using bootstrap |
| **Time** | 60 minutes |

**Problem statement-4.3 :**  **Create a React application with name “react-create-register”. Display register form as shown below**



Lab 4.4 :

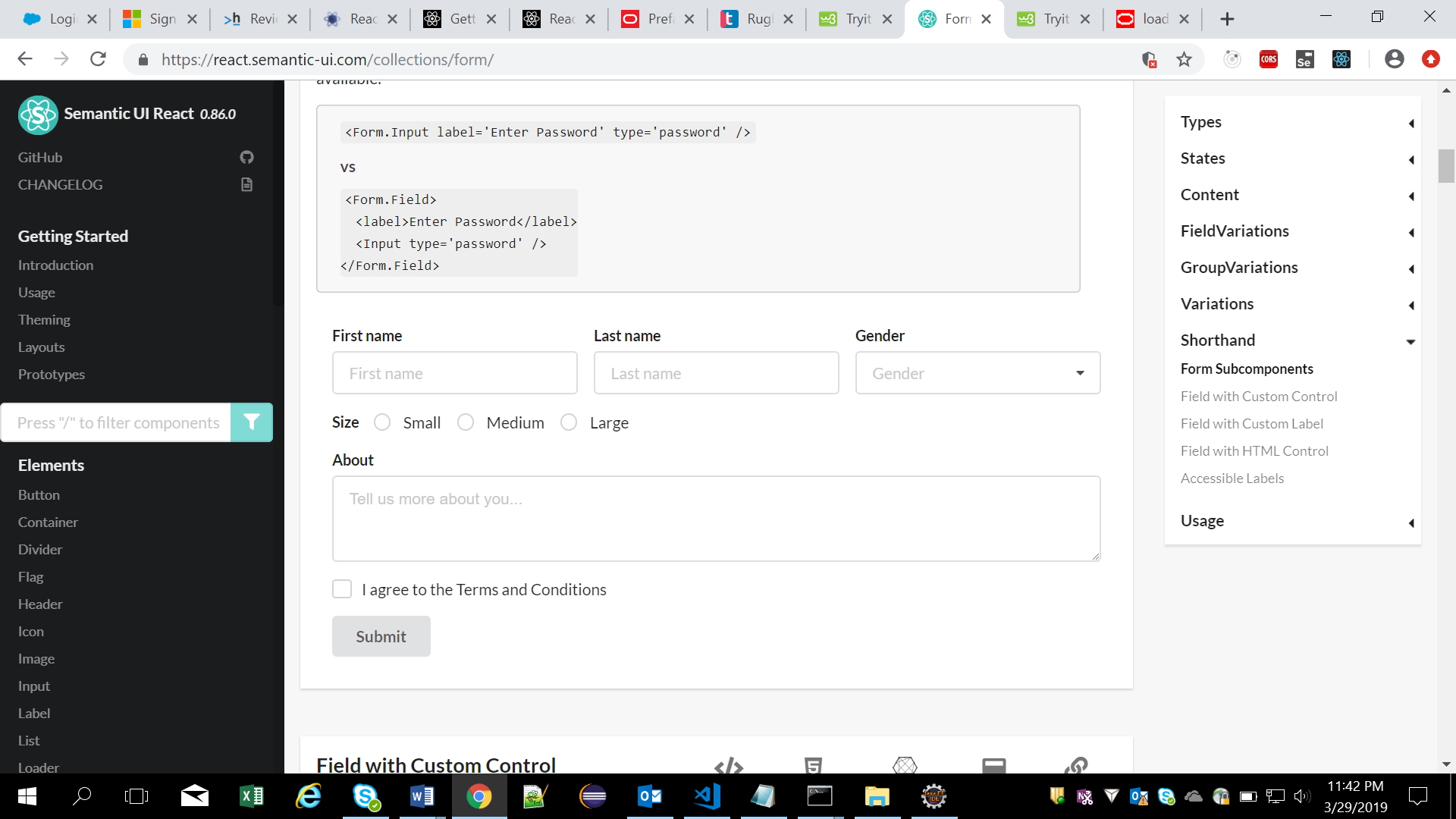
|  |  |
| --- | --- |
| **Goals** | Creating a React application rendering register page ,apply styles using reactstrap |
| **Time** | 60 minutes |

**Problem statement-4.4 :**  **Create a React application with name “react-create-register”. Display register form using reactstrap as shown 4.3 lab scenario**

Lab 4.5 :

|  |  |
| --- | --- |
| **Goals** | Creating a React application rendering register page ,apply styles using semt |
| **Time** | 60 minutes |

**Problem statement-4.5:**  **Create a React application with name “react-create-register-** **”. Display register form using semanticUI**



Lab 5: Debugging errors and error boundaries

Lab 5.1 :

|  |  |
| --- | --- |
| **Goals** | Choose any of the previous React application ,handle error using error boundaries |
| **Time** | 60 minutes |

**Problem statement-5.1 : Choose any of the previous React application ,handle error using error boundaries**

Lab 6: React Component Life Cycle

Lab 6.1 :

|  |  |
| --- | --- |
| **Goals** | Creating a React application demonstrating React Lifecycle |
| **Time** | 80 minutes |

**Problem statement-6.1 : Creating a React application demonstrating React Lifecycle**

Create a react application with two components which is nested.

1. Parent
2. Child ( nested inside Parent)

Create 2 variables in state named

Name : default

Price : 0.0

Print following messages using the life cycle methods

**componentWillMount – component Loaded & assign the values for the state variablles as**

**Name: TV**

**Price: 30000**

**Also print them in browser using component did mount**

**componentDidMount – Component Mounted**

**componentWillMount – component will mount**

**componentDidMount – component did mount & assign the values of state variables as**

**Name: Default**

**Price: 0.0**

Lab 6.1 :

|  |  |
| --- | --- |
| **Goals** | * Create a react Application state & lifecycle |
| **Time** | 90 minutes |

**Problem statement-6.2 :**  **Create a React application with name “react-create-state-clock”.**

**(use statefull component , necessary life cycle methods & props where ever needed)**

1. **Create a ClockTick component in a js and render it through index.js**
2. **The clock has to update for every second and display in browser using state**
3. **When Clock starts it tick the name has to be changed to CAPGEMINI from IGATE.**

Lab 7: Component in details

Lab 7.1 :

|  |  |
| --- | --- |
| **Goals** | Creating a React application and display content using Fragments |
| **Time** | 20 minutes |

**Problem statement-7.1 : Use Lab 3.2 and convert it into new demo, make use of fragments where ever its necessary.**

Lab 7.2 :

|  |  |
| --- | --- |
| **Goals** | Creating a React application demonstrating HOC and validation of PROPS |
| **Time** | 80 minutes |

**Problem statement-7.2 : Creating a React application demonstrating HOC and validation of PROPS**

Create a **EmployeeHOC** , Higher Order function which is used only to pass data to **MyComponent**.

This function takes **MyComponent**, enhances it with **EmpData which is a variable with data name and pwd, shown below**

var EmpData = {

name: ‘HOC’,

pwd: ‘hoc’

}

**Get this data and assign to variable of MyComponent through setState method** and returns the enhanced component that will be rendered on the screen.

**Validate the name and pwd in mycomponent using prop validation for**

1. **Value is mandatory**
2. **Name can be of any data type and pwd is of string data type**

Lab 7.3:

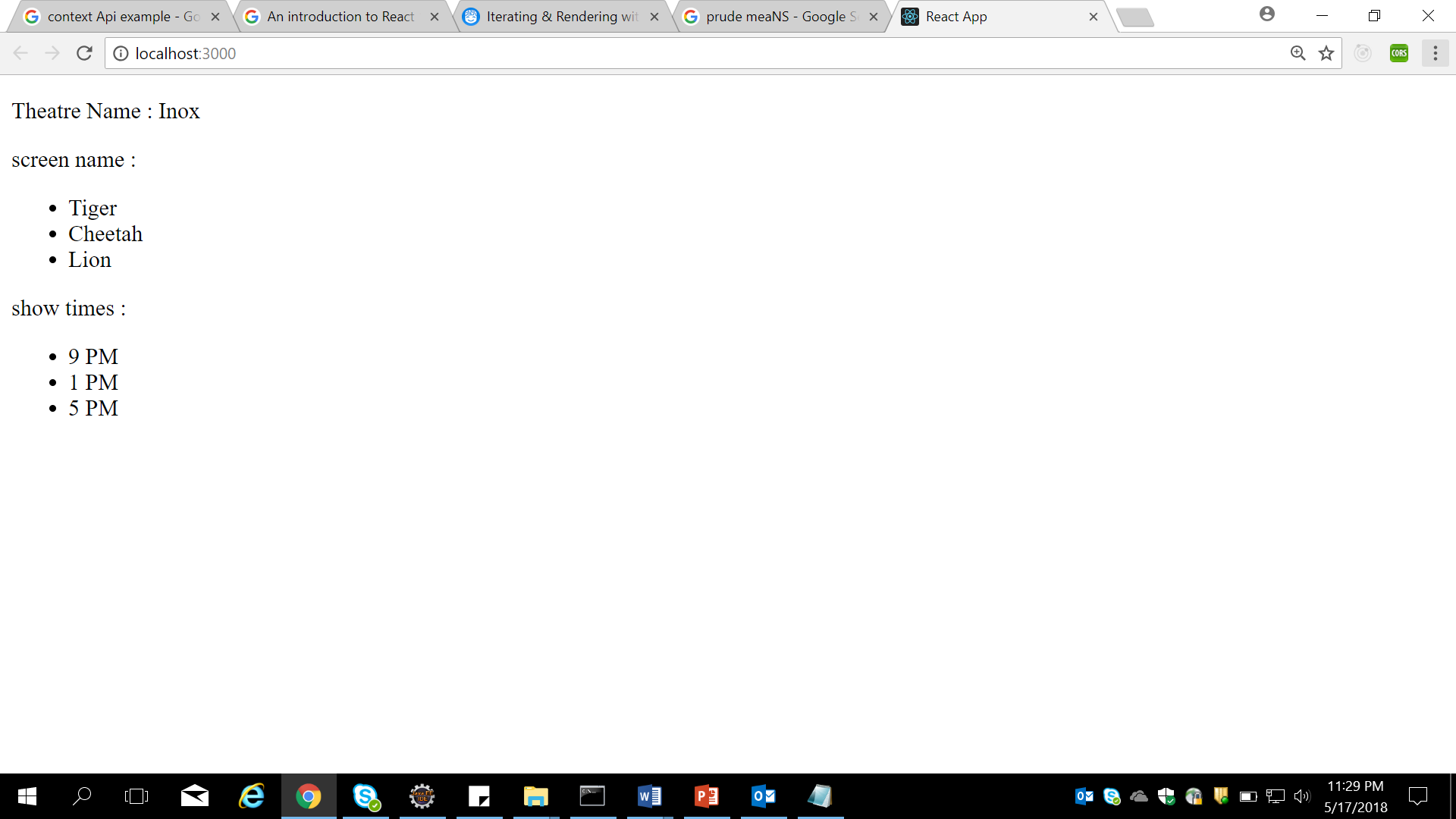
|  |  |
| --- | --- |
| **Goals** | * Create a basic react app using React Context API |
| **Time** | 60 minutes |

**Problem statement-7.3 :**  **Create a React application with name “react-create-context-mall”.Use Context Api and implement the below scenario**

Create 4 js files name as follows index , Mall, Theatre and Screen. Render the final output using index.js(using Mall in index)

Screen is child component of Theatre and Theatre is child component of Mall. Pass the values for variables in Screen through Context API

**Final output looks like :**



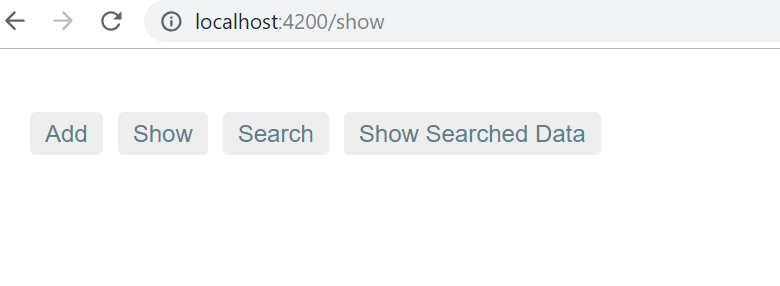
**Lab 7.4 :**

|  |  |
| --- | --- |
| Goals | Creating a React application Managing CRUD operations for Inventory using AXIOS & MongoDB |
| Time | 240 minutes |

Create an “Employee management System App” where in employee could be maintained. There is also a Search Data option allowing to search for employee.

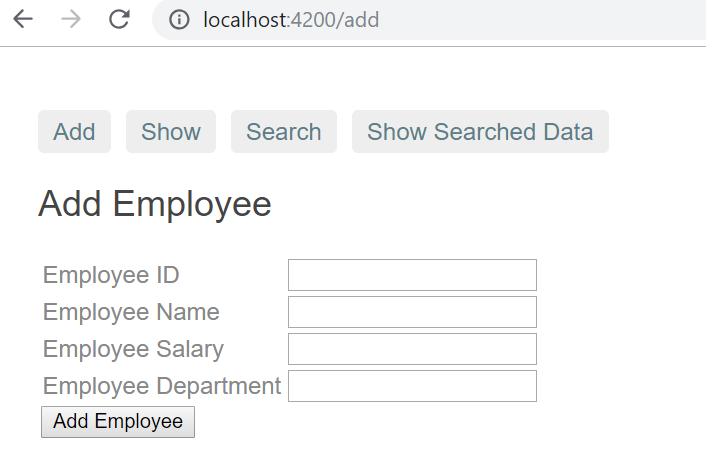
Implement CRUD operations using MongoDB .

Refer below screen shot:

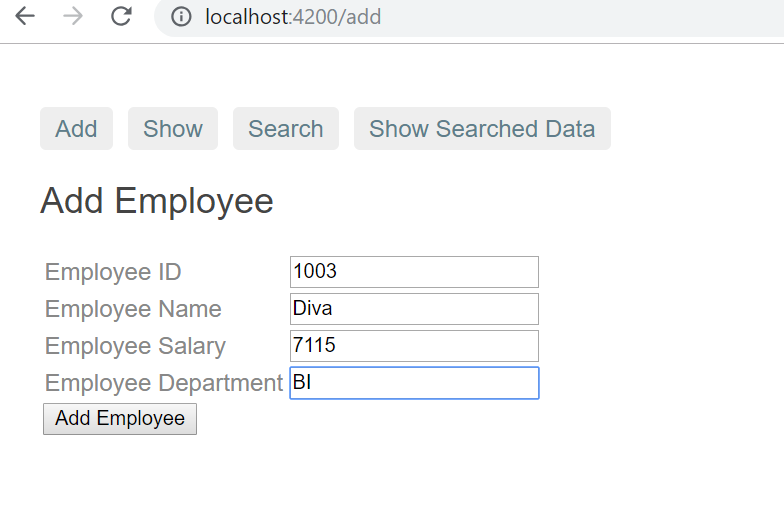


1. When the user clicks on “Add Employee” -> Allows a User to add a employee.

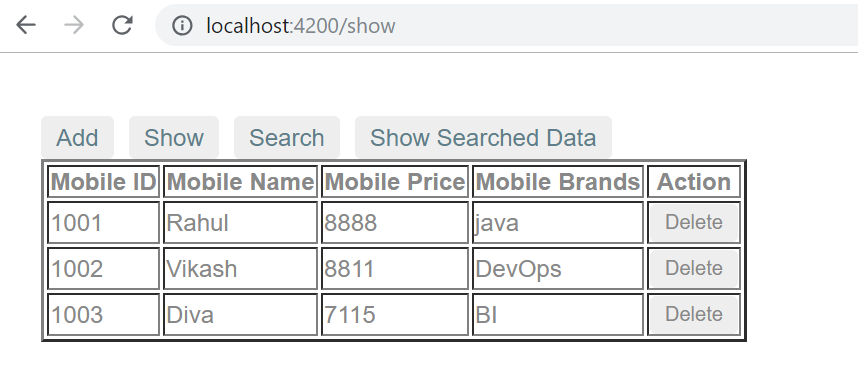
Refer below screen shot:



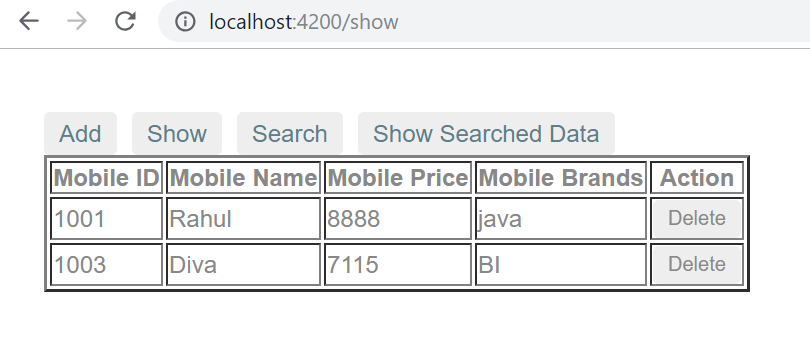
Enter Employee Id, Name, salary, Department & data will added in array .Click on show link & we will get all data Refer below screens:



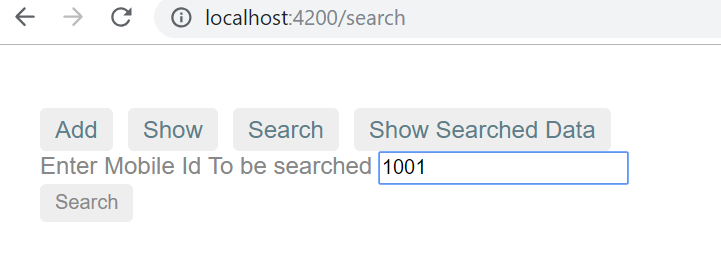
Click on Show



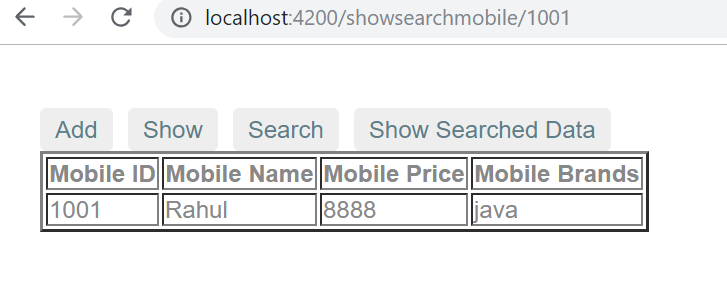
Clicking on Delete button…. i.e. 1002. That row will delete



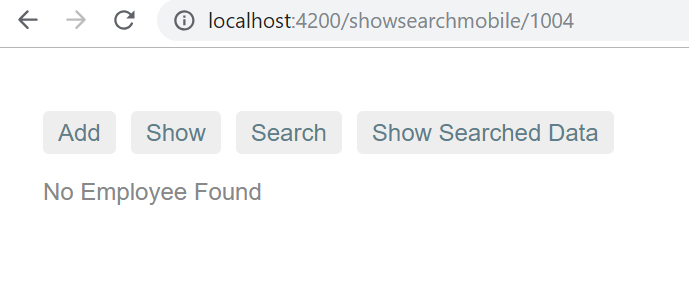
Search the employee by employee id



Data will come in Show Search Data



If no data there i.e. search for 1004



Lab 8: React Router

Lab 8.1 :

|  |  |
| --- | --- |
| **Goals** | Creating a React application using routers |
| **Time** | 20 minutes |

**Problem statement- 8.1 : use lab 7.4 and use router concept in it.**

Lab 8.2 :

|  |  |
| --- | --- |
| **Goals** | * Create a Gallery App using react Application using Reacter Router |
| **Time** | 120 minutes |

**Problem statement-8.2 :**  **Create a React application with name “react-create-route-gallery”.**

1. **Create a Application using router for a gallery of mobile application for minimum 3 category of mobiles**
2. **Create 1 component for atleast 3 category, like Samsung ,Mi and Nokia (add needed images in the src folder, minimum 4 )**
3. **Import images in to project and display it using image tag**
4. **Use Navigation.js for creating links to different pages**
5. **Use Main.js for writing the route details**
6. **Finally use App.js to add Navigation and Main then display through index.js**

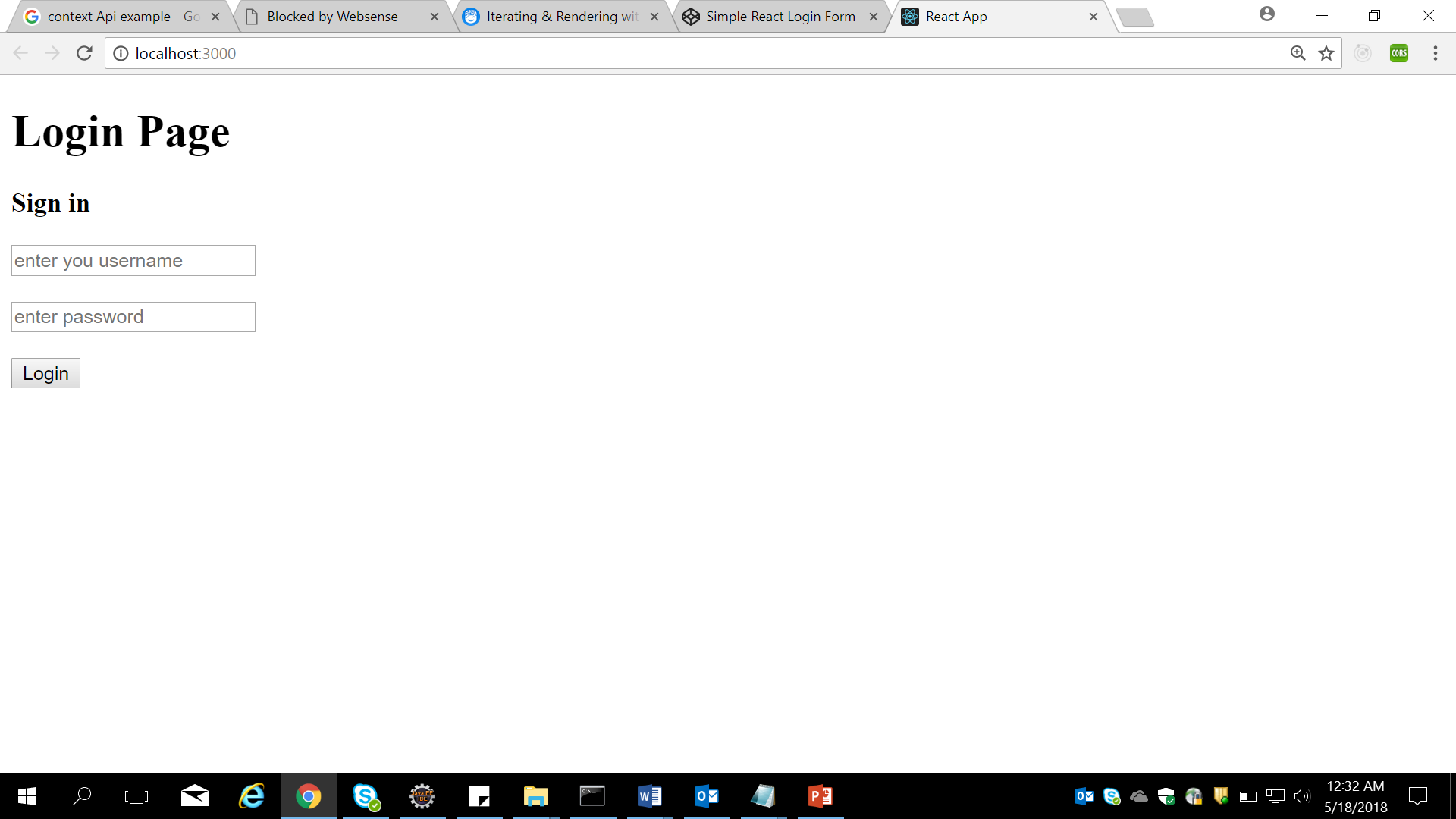
Lab 9: React Forms

Lab 9.1 :

|  |  |
| --- | --- |
| **Goals** | * Create a react LoginForm using React( nested components, nested Components, props, conditional rendering, inline styling, event handling , input refs) |
| **Time** | 120 minutes |

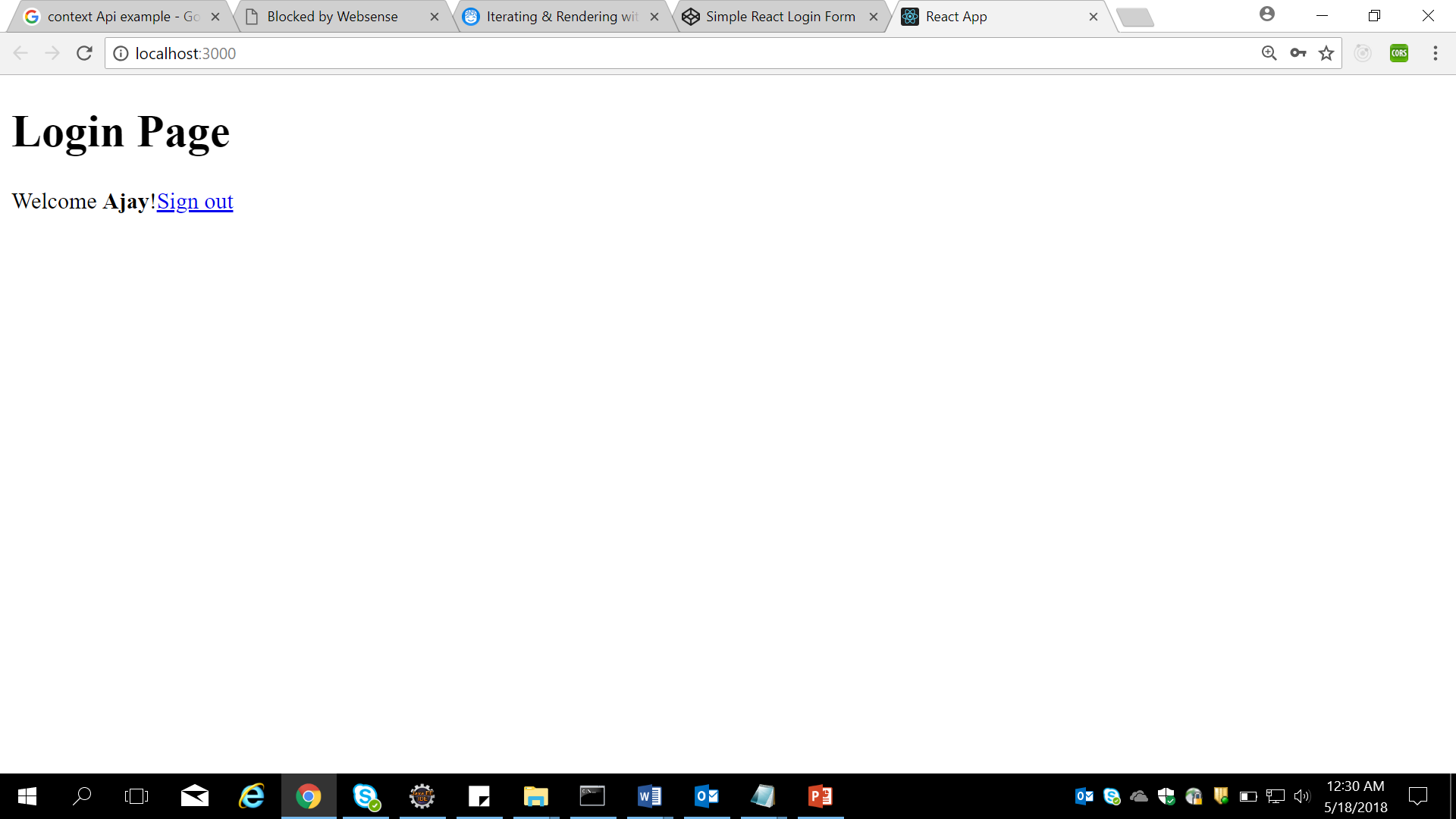
**Problem statement-9.1 :**  **Create a React application with name “react-create-login-form”.**

1. **On running it has to display a login form like below**



**After typing username and password,on submit,**

1. **Store the data of form in state**
2. **It has to render output like below screen**

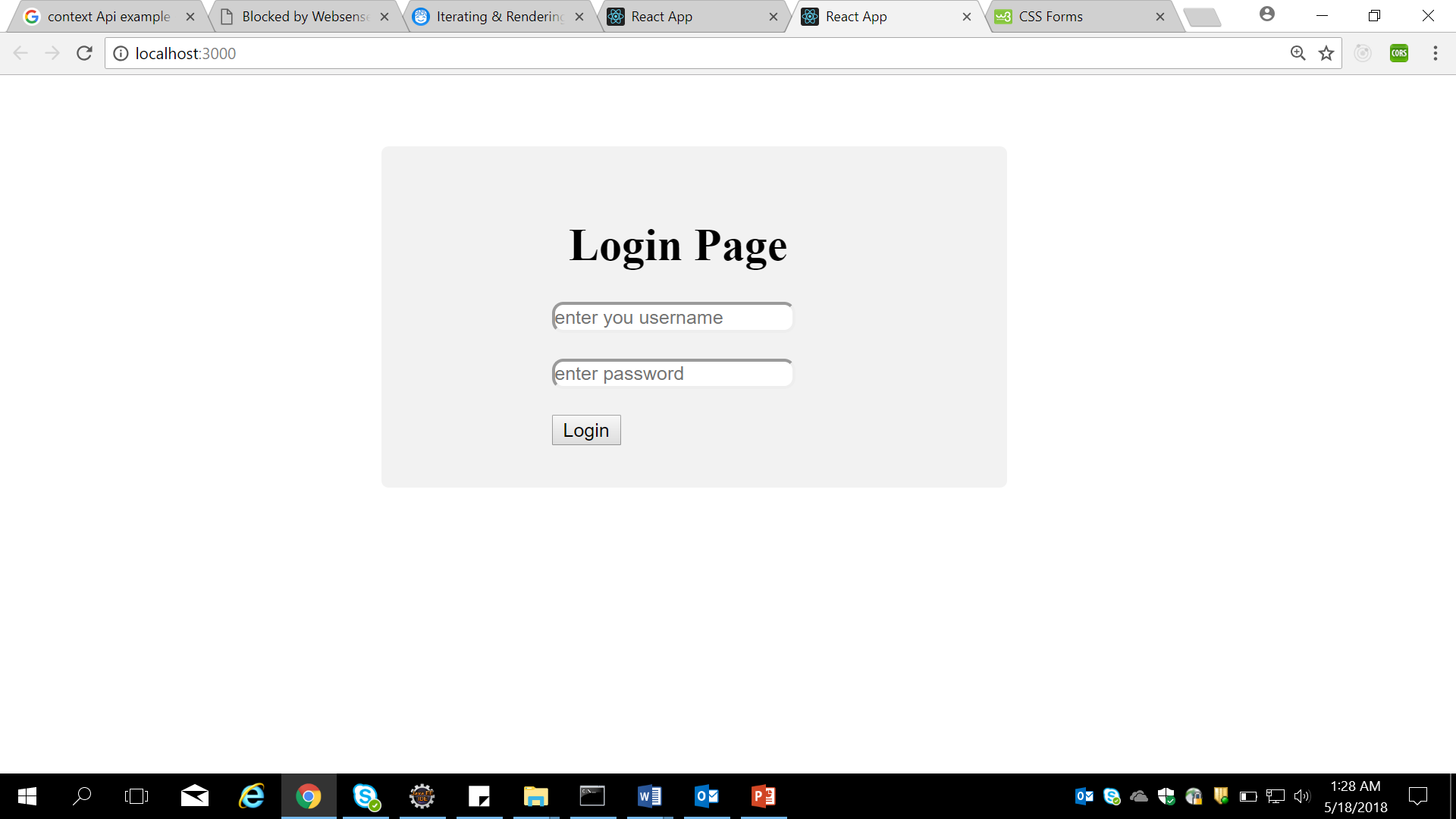


**If clicking signout it has to set values of name and password as null in state and render the sig in component**

**(note we are not using router for this, use conditional rendering)**

1. **Modify the look of the above form using css (asd given below)**

**Final Output of the above code should be:**



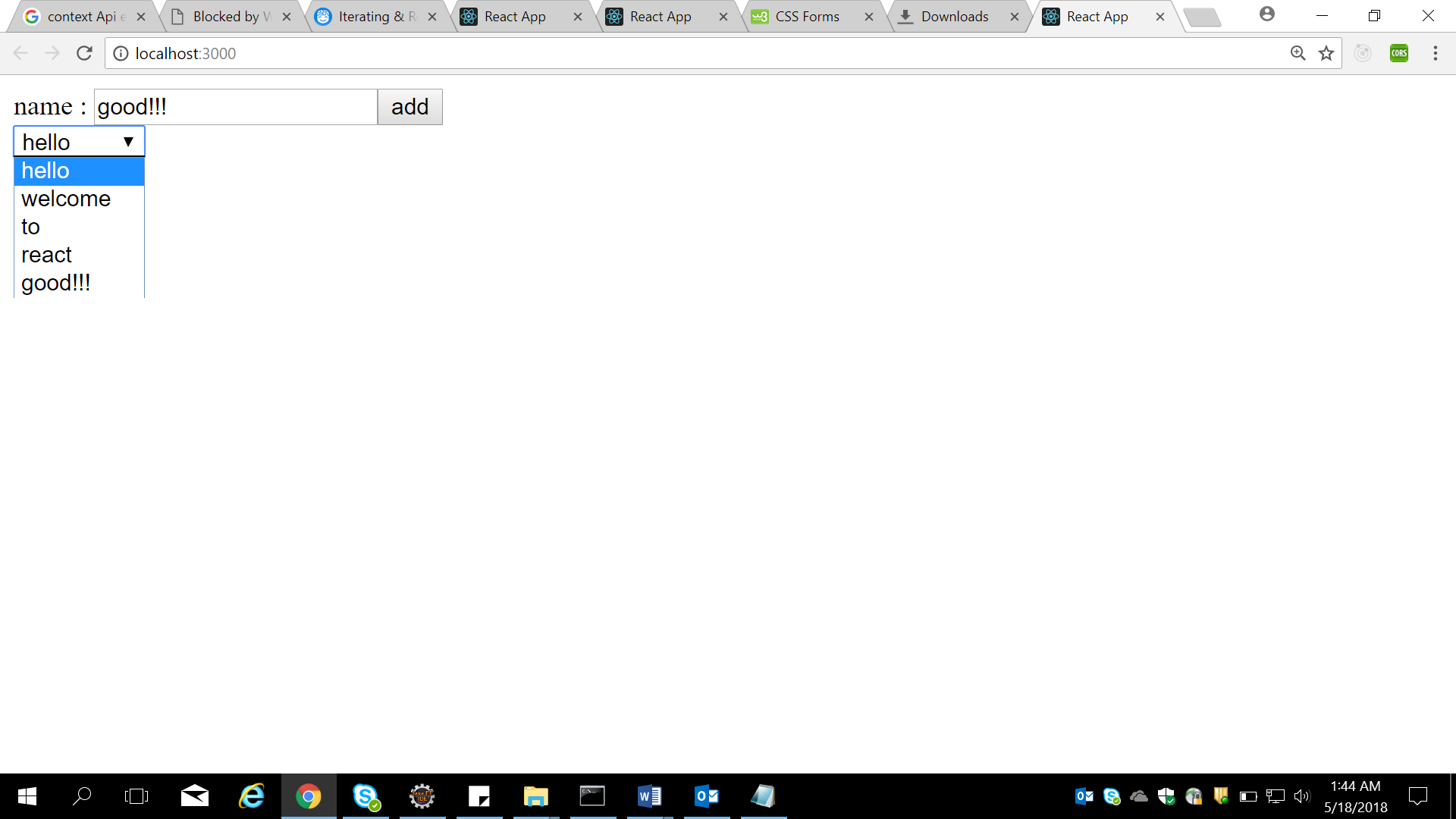
Lab 9.2 :

|  |  |
| --- | --- |
| **Goals** | * Create a react Application using React – dynamic keys to children |
| **Time** | 120 minutes |

**Problem statement-9.2 :**  **Create a React application with name “react-create-dynamic-keys”.**

1. **Create a text box and a button in a js file named FormEg.js**
2. **When user enters input and clicks the button , add the content typed to drop down box on same page**

**Final o/p should look like**



Lab 9.3 :

|  |  |
| --- | --- |
| **Goals** | Creating a React application for registration form |
| **Time** | 160 minutes |

**Problem statement- 9.3 : use lab 8.1 and use validation in add employee form.**