1. How to call api with ajax in javascript?

Ans: <!DOCTYPE html>

<html>

<head>

<title>Sandbox</title>

<meta charset="UTF-8" />

<link rel="stylesheet" href="src/styles.css" />

<script>

// var url = "helloworld";

function Load() {

const res = new XMLHttpRequest();

res.open("GET", "https://jsonplaceholder.typicode.com/todos/1");

res.onload = function () {

// document.getElementById("text").innerHTML = this.responseText;

var result = JSON.parse(this.responseText);

document.getElementById("tag").innerHTML = result.title;

};

res.send();

// console.log(this.ponseText);

}

</script>

</head>

<body>

<!-- <h1>HTML/CSS/JS Scratch Template</h1> -->

<div id="app">

<h1 id="text"></h1>

<p id="tag"></p>

</div>

<button type="submit" onclick="Load()">change</button>

<!-- <script src="./src/index.js"></script> -->

</body>

</html>

2.how to create when user enter one input field data it will auto fill remaining inputs with javascript and html?

Ans: body>

EmpId <input type="number" id="empid" onkeyup="autoFill()"/><br/>

Name <input type="text" id="name" /><br/>

Email <input type="email" id="email" /><br/>

Fullname <input type="text" id="fname" /><br/>

Nickname <input type="text" id="Nname" /><br/>

<!-- <p id="res"></p> -->

<script>

var data = [

{

"0": 1234,

"1": "Dinesh",

"2": "dinesh@gmail.com",

"3": "DineshV",

"4": "D"

},

{

"0": 5678,

"1": "Siva",

"2": "siva@gmail.com",

"3": "SivaV",

"4": "S"

}

];

function autoFill(){

var result = document.getElementById("empid").value;

var resultarr = [document.getElementById("name"), document.getElementById("email"), document.getElementById("fname"), document.getElementById("Nname")]

for(let i in data){

if(data[i][0]==result){

for(let j=0;j<resultarr.length;j++){

resultarr[j].value = data[i][j+1];

}

}

}

}

</script>

</body>

how to create if I have one text when user clicks that text it will show some of checkboxes using html , css , javascript.

<!DOCTYPE html>

<html>

<head>

<title>Employee Data Form</title>

<!-- Add Bootstrap CSS link -->

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<!-- https://code.jquery.com/jquery-3.5.1.slim.min.js -->

<style>

body {

background-color: #f7f7f7;

}

.container {

background-color: #fff;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

border-radius: 5px;

padding: 20px;

}

.card {

border: none;

margin-bottom: 20px;

}

.card-header {

background-color: #007bff;

color: #fff;

}

.btn {

margin-bottom: 10px;

}

.form-check-label {

font-weight: 400;

}

.form-control {

border-radius: 5px;

}

#employeeId {

margin-bottom: 10px;

}

#submitForm {

background-color: #007bff;

color: #fff;

border: none;

border-radius: 5px;

transition: background-color 0.3s ease-in-out;

}

#submitForm:hover {

background-color: red;

}

.img-fluid {

max-width: 100%;

height: auto;

}

.select-action {

display: none;

}

.additional-checkboxes {

display: none;

}

.hiddenCheckbox {

display: none;

}

.btn-select-more:hover {

color: #0056b3;

}

</style>

</head>

<body>

<div class="container mt-5">

<h1 class="text-center">Employee Data Form</h1>

<div class="row">

<div class="col-md-6">

<div class="card">

<div class="card-header">Our Works</div>

<div class="card-body">

<div class="form-group">

<select class="form-control" id="actionDropdown">

<option value="" disabled selected>Select For Employee Info</option>

<option value="create">Employee Related info</option>

</select>

</div>

<div class="select-action">

<button class="btn btn-primary btn-block" id="button1">Create New Employee</button>

<button class="btn btn-success btn-block" id="button2">Update Existing Employee</button>

</div>

</div>

</div>

</div>

<div class="col-md-6">

<div class="card">

<div class="card-header">Report Submission</div>

<div class="card-body">

<form id="employeeForm">

<div class="form-group">

<label for="employeeId">Employee ID:</label>

<input type="text" id="employeeId" name="employeeId" class="form-control"

pattern="IN[0-9]{9}" title="Enter proper Employee ID">

</div>

<div>

<label>Select Options:</label>

<div class="form-check">

<input type="checkbox" class="form-check-input" name="options[]" value="option1"

id="selectallCheckbox">

<label class="form-check-label">Select All</label>

</div>

</div>

<div class="d-flex justify-content-between">

<div class="form-group">

<div class="form-check">

<input type="checkbox" class="form-check-input" name="options[]"

value="option1">

<label class="form-check-label">Employee Name</label>

</div>

<div class="form-check">

<input type="checkbox" class="form-check-input" name="options[]"

value="option2">

<label class="form-check-label">Date of Joining</label>

</div>

<div class="form-check">

<input type="checkbox" class="form-check-input" name="options[]"

value="option3">

<label class="form-check-label">Date of Relieving</label>

</div>

<div class="form-check">

<input type="checkbox" class="form-check-input" name="options[]"

value="option4">

<label class="form-check-label">Reporting Manager</label>

</div>

<!-- Added code -->

<div class="hiddenCheckbox">

<div class="form-check">

<input type="checkbox" class="form-check-input" name="options[]"

value="option1">

<label class="form-check-label">Employee Name</label>

</div>

<div class="form-check">

<input type="checkbox" class="form-check-input" name="options[]"

value="option2">

<label class="form-check-label">Date of Joining</label>

</div>

<div class="form-check">

<input type="checkbox" class="form-check-input" name="options[]"

value="option3">

<label class="form-check-label">Date of Relieving</label>

</div>

<div class="form-check">

<input type="checkbox" class="form-check-input" name="options[]"

value="option4">

<label class="form-check-label">Reporting Manager</label>

</div>

</div>

</div>

<div class="form-group hiddenCheckbox">

<div class="form-check">

<input type="checkbox" class="form-check-input" name="options[]"

value="option1">

<label class="form-check-label">Employee Name</label>

</div>

<div class="form-check">

<input type="checkbox" class="form-check-input" name="options[]"

value="option2">

<label class="form-check-label">Date of Joining</label>

</div>

<div class="form-check">

<input type="checkbox" class="form-check-input" name="options[]"

value="option3">

<label class="form-check-label">Date of Relieving</label>

</div>

<div class="form-check">

<input type="checkbox" class="form-check-input" name="options[]"

value="option4">

<label class="form-check-label">Reporting Manager</label>

</div>

<div class="form-check">

<input type="checkbox" class="form-check-input" name="options[]"

value="option1">

<label class="form-check-label">Employee Name</label>

</div>

<div class="form-check">

<input type="checkbox" class="form-check-input" name="options[]"

value="option2">

<label class="form-check-label">Date of Joining</label>

</div>

<div class="form-check">

<input type="checkbox" class="form-check-input" name="options[]"

value="option3">

<label class="form-check-label">Date of Relieving</label>

</div>

<div class="form-check">

<input type="checkbox" class="form-check-input" name="options[]"

value="option4">

<label class="form-check-label">Reporting Manager</label>

</div>

</div>

<!-- Added code -->

</div>

<div class="select-more text-right">

<button class="btn btn-select-more" id="addMoreCheckboxes">Add More

Checkboxes</button>

</div>

<button type="button" class="btn btn-primary btn-block" id="submitForm">Submit</button>

</form>

</div>

</div>

</div>

</div>

<div class="row mt-5">

<div class="col-md-6 offset-md-6">

<img src="your-image.jpg" alt="Your Image" class="img-fluid">

</div>

</div>

</div>

<!-- Add Bootstrap JS and jQuery scripts at the end of the body -->

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

<script>

$(document).ready(function () {

// Show/hide the action buttons and "Select More" option based on dropdown selection

$("#actionDropdown").change(function () {

const selectedAction = $(this).val();

if (selectedAction === "create") {

$(".select-action").show();

$(".select-more").hide();

} else {

$(".select-action").hide();

$(".select-more").show();

}

});

// JavaScript code for handling form submission

$("#submitForm").click(function () {

const action = $("#button1").hasClass("active") ? "Create New Employee" : "Update Existing Employee";

const employeeId = $("#employeeId").val();

const options = $("input[name='options[]']:checked").map(function () {

return this.value;

}).get();

// Construct URL or perform desired action based on the selected data

const url = `/your-java-endpoint?action=${action}&employeeId=${employeeId}&options=${options.join(",")}`;

// Send data to the Java endpoint or handle as needed

console.log("Sending data to Java endpoint:", url);

});

// Toggle active class on button click

$("#button1, #button2").click(function () {

$(this).toggleClass("active");

$(this).siblings().removeClass("active");

});

// "Add More Checkboxes" button functionality

$("#addMoreCheckboxes").click(function (e) {

e.preventDefault();

$(".hiddenCheckbox").attr("style", "display:block");

$("#addMoreCheckboxes").hide();

});

// Select all checkbox functionality

$("#selectallCheckbox").change(function () {

$(".hiddenCheckbox").attr("style", "display:block");

$("#addMoreCheckboxes").hide();

$(".form-check-input").attr("checked", true)

});

});

</script>

</body>

</html>

Create html validations in only html using patterns:

import React, { useState } from "react";

import axios from "axios";

const CreateEmployee = () => {

const [employeeData, setEmployeeData] = useState({

id: 1,

employeeId: "",

currentDesignation: "",

businessUnit: "",

reportingManager: "",

officialEmail: "",

joiningDate: "",

releavingDate: "",

reasonForLeaving: "",

employeeStatus: "",

totalExperience: 0,

currentCTC: 0,

currentWorkingTechnologies: "",

additionalTechnology1: "",

additionalTechnology2: "",

additionalTechnology3: "",

additionalTechnology4: "",

certification1: "",

certification2: "",

certification3: "",

certification4: "",

reference1Name: "",

reference1Email: "",

reference2Name: "",

reference2Email: "",

reference3Name: "",

reference3Email: "",

personalDetails: {

id: 1,

employeeId: "0",

firstName: "",

lastName: "",

dateOfBirth: "",

personalEmail: "",

contactNumber1: "",

contactNumber2: "",

compliance: false

},

compliance: {

id: 1,

employeeId: "0",

uanNumber: "",

panNumber: "",

aadharNumber: "",

passportNumber: "",

bankname: "",

bankAccountNumber: "",

ifscCode: ""

},

experiences: [

{

id: 1,

employeeId: "0",

professionalEmployee: false,

companyName: "",

companyLocation: "",

designation: "",

ctc: 0,

hikePercentage: 0,

joiningDate: "",

releavingDate: "",

reasonforreleaving: "",

totalexperience: ""

}

]

});

const handleChange = (e) => {

const { name, value } = e.target;

setEmployeeData({

...employeeData,

[name]: value

});

};

const handleExperienceChange = (e, index) => {

const { name, value } = e.target;

const updatedExperiences = [...employeeData.experiences];

updatedExperiences[index] = {

...updatedExperiences[index],

[name]: value

};

setEmployeeData({

...employeeData,

experiences: updatedExperiences

});

};

const addExperience = () => {

setEmployeeData({

...employeeData,

experiences: [

...employeeData.experiences,

{

id: employeeData.experiences.length + 1,

employeeId: "0",

professionalEmployee: false,

companyName: "",

companyLocation: "",

designation: "",

ctc: 0,

hikePercentage: 0,

joiningDate: "",

releavingDate: "",

reasonforreleaving: "",

totalexperience: ""

}

]

});

};

const removeLastExperience = () => {

const updatedExperiences = [...employeeData.experiences];

updatedExperiences.pop(); *// Remove the last experience*

setEmployeeData({

...employeeData,

experiences: updatedExperiences

});

};

const handleSubmit = async (e) => {

e.preventDefault();

try {

const response = await axios.post(

"http://localhost:8080/api/employees",

employeeData

);

setEmployeeData(response);

*// Handle success, e.g., show a success message or redirect*

} catch (error) {

*// Handle error, e.g., show an error message*

console.error("Error creating employee:", error);

}

};

const handleComplianceChange = () => {

setEmployeeData({

...employeeData,

personalDetails: {

...employeeData.personalDetails,

compliance: !employeeData.personalDetails.compliance

}

});

};

return (

<div>

<h2>Create New Employee</h2>

<form onSubmit={handleSubmit}>

{*/\* {/ Employee Information /} \*/*}

<div>

<label>

Employee ID:

<input

type="text"

name="employeeId"

value={employeeData.employeeId}

onChange={handleChange}

maxLength={11}

minLength={6}

pattern="IN[0-9]+"

*// pattern="[0-9]{2}[0-9]{2}[0-9]{4}"*

title="Please enter Employee Id in correct format"

required

/>

</label>

</div>

<div>

<label>

Current Designation:

<input

type="text"

name="currentDesignation"

value={employeeData.currentDesignation}

onChange={handleChange}

pattern="[a-z]+"

title="please enter alphabets values"

required

/>

</label>

</div>

<div>

<label>

businessUnit:

<input

type="text"

name="businessUnit"

value={employeeData.businessUnit}

onChange={handleChange}

pattern="[A-z]\*[a-z]+"

title="please enter alphabets values"

required

/>

</label>

</div>

<div>

<label>

Reporting Manager :

<input

type="text"

name="reportingManager"

value={employeeData.reportingManager}

onChange={handleChange}

pattern="[a-z]+"

title="please enter alphabets values"

required

/>

</label>

</div>

<div>

<label>

Official Email :

<input

type="email"

name="officialEmail"

value={employeeData.officialEmail}

onChange={handleChange}

pattern="[a-z]+[.]\*[a-z]+[0-9]\*@[a-z]+.[a-z]+"

required

/>

</label>

</div>

<div>

<label>

joiningDate:

<input

type="date"

name="joiningDate"

value={employeeData.joiningDate}

onChange={handleChange}

/>

</label>

</div>

<div>

<label>

employeeStatus:

<input

type="text"

name="employeeStatus"

value={employeeData.employeeStatus}

onChange={handleChange}

pattern="[a-z]+"

title="please enter alphabets values"

required

/>

</label>

</div>

<div>

<label>

totalExperience:

<input

type="text"

name="totalExperience"

value={employeeData.totalExperience}

onChange={handleChange}

pattern="[0-9]+.[0-9]+"

title="please Enter in 9.0 format"

/>

</label>

</div>

<div>

<label>

currentCTC:

<input

type="text"

name="currentCTC"

value={employeeData.currentCTC}

onChange={handleChange}

pattern="[0-9]+.[0-9]+"

title="Please enter in 9.0 format"

required

/>

</label>

</div>

<div>

<label>

currentWorkingTechnologies:

<input

type="text"

name="currentWorkingTechnologies"

value={employeeData.currentWorkingTechnologies}

onChange={handleChange}

pattern="[a-z]+,"

title="please enter alphabets values"

required

/>

</label>

</div>

<div>

<label>

additionalTechnology1:

<input

type="text"

name="additionalTechnology1"

value={employeeData.additionalTechnology1}

onChange={handleChange}

pattern="[a-z]+"

title="please enter alphabets values"

required

/>

</label>

</div>

<div>

<label>

additionalTechnology2:

<input

type="text"

name="additionalTechnology2"

value={employeeData.additionalTechnology2}

onChange={handleChange}

pattern="[a-z]+"

title="please enter alphabets values"

required

/>

</label>

</div>

<div>

<label>

certification1:

<input

type="text"

name="certification1"

value={employeeData.certification1}

onChange={handleChange}

pattern="[a-z]+"

title="please enter alphabets values"

required

/>

</label>

</div>

<div>

<label>

certification2:

<input

type="text"

name="certification2"

value={employeeData.certification2}

onChange={handleChange}

pattern="[a-z]+"

title="please enter alphabets values"

required

/>

</label>

</div>

<div>

<label>

reference1Name:

<input

type="text"

name="reference1Name"

value={employeeData.reference1Name}

onChange={handleChange}

pattern="[a-z]+"

title="please enter alphabets values"

required

/>

</label>

<label>

reference1Email:

<input

type="email"

name="reference1Email"

value={employeeData.reference1Email}

onChange={handleChange}

pattern="[a-z]+.\*[a-z]+[0-9]\*@[a-z]+.[a-z]+"

required

/>

</label>

</div>

<div>

<label>

reference2Name:

<input

type="text"

name="reference2Name"

value={employeeData.reference2Name}

onChange={handleChange}

pattern="[a-z]+"

title="please enter alphabets values"

required

/>

</label>

<label>

reference2Email:

<input

type="email"

name="reference2Email"

value={employeeData.reference2Email}

onChange={handleChange}

pattern="[a-z]+.\*[a-z]+[0-9]\*@[a-z]+.[a-z]+"

required

/>

</label>

</div>

<div>

<label>

reference3Name:

<input

type="text"

name="reference3Name"

value={employeeData.reference3Name}

onChange={handleChange}

pattern="[a-z]+"

title="please enter alphabets values"

required

/>

</label>

<label>

reference3Email:

<input

type="email"

name="reference3Email"

value={employeeData.reference3Email}

onChange={handleChange}

pattern="[a-z]+.\*[a-z]+[0-9]\*@[a-z]+.[a-z]+"

required

/>

</label>

</div>

{*/\* {/ Personal Details /} \*/*}

<h3>Personal Details</h3>

<div>

<label>

First Name:

<input

type="text"

name="personalDetails.firstName"

value={employeeData.personalDetails.firstName}

onChange={(e) => {

setEmployeeData({

...employeeData,

personalDetails: {

...employeeData.personalDetails,

firstName: e.target.value

}

});

}}

pattern="[a-z]+"

title="please enter alphabets values"

required

/>

</label>

</div>

<div>

<label>

Last Name:

<input

type="text"

name="personalDetails.lastName"

value={employeeData.personalDetails.lastName}

onChange={(e) => {

setEmployeeData({

...employeeData,

personalDetails: {

...employeeData.personalDetails,

lastName: e.target.value

}

});

}}

pattern="[a-z]+"

title="please enter alphabets values"

required

/>

</label>

</div>

<div>

<label>

Date of Birth:

<input

type="date"

name="personalDetails.dateOfBirth"

value={employeeData.personalDetails.dateOfBirth}

onChange={(e) => {

setEmployeeData({

...employeeData,

personalDetails: {

...employeeData.personalDetails,

dateOfBirth: e.target.value

}

});

}}

/>

</label>

</div>

<div>

<label>

Personal Email:

<input

type="email"

name="personalDetails.personalEmail"

value={employeeData.personalDetails.personalEmail}

onChange={(e) => {

setEmployeeData({

...employeeData,

personalDetails: {

...employeeData.personalDetails,

personalEmail: e.target.value

}

});

}}

pattern="[a-z]+.\*[a-z]+[0-9]\*@[a-z]+.[a-z]+"

required

/>

</label>

</div>

<div>

<label>

Contact Number 1:

<input

type="text"

name="personalDetails.contactNumber1"

value={employeeData.personalDetails.contactNumber1}

onChange={(e) => {

setEmployeeData({

...employeeData,

personalDetails: {

...employeeData.personalDetails,

contactNumber1: e.target.value

}

});

}}

pattern="[0-9]{10}"

title="please enter numeric values"

required

/>

</label>

</div>

<div>

<label>

Contact Number 2:

<input

type="text"

name="personalDetails.contactNumber2"

value={employeeData.personalDetails.contactNumber2}

onChange={(e) => {

setEmployeeData({

...employeeData,

personalDetails: {

...employeeData.personalDetails,

contactNumber2: e.target.value

}

});

}}

pattern="[0-9]{10}"

title="please enter numeric values"

required

/>

</label>

</div>

<div>

<label>

Compliance:

<input

type="checkbox"

name="personalDetails.compliance"

checked={employeeData.personalDetails.compliance}

onChange={handleComplianceChange}

/>

True

</label>

</div>

{*/\* {/ Compliance Details /} \*/*}

{*/\* {/ Compliance Details /} \*/*}

{employeeData.personalDetails.compliance && (

<div>

<h3>Compliance Details</h3>

<div>

<label>

UAN Number:

<input

type="text"

name="compliance.uanNumber"

value={employeeData.compliance.uanNumber}

onChange={(e) => {

setEmployeeData({

...employeeData,

compliance: {

...employeeData.compliance,

uanNumber: e.target.value

}

});

}}

pattern="[0-9]{12}"

title="please enter numeric values"

required

/>

</label>

</div>

<div>

<label>

Aadhar Number:

<input

type="text"

name="compliance.aadharNumber"

value={employeeData.compliance.aadharNumber}

onChange={(e) => {

setEmployeeData({

...employeeData,

compliance: {

...employeeData.compliance,

aadharNumber: e.target.value

}

});

}}

pattern="[0-9]{12}"

title="please enter numeric values"

required

/>

</label>

</div>

<div>

<label>

PAN Number:

<input

type="text"

name="compliance.panNumber"

value={employeeData.compliance.panNumber}

onChange={(e) => {

setEmployeeData({

...employeeData,

compliance: {

...employeeData.compliance,

panNumber: e.target.value

}

});

}}

pattern="[A-Z]{5}[0-9]{4}[A-Z]{1}"

title="please enter in ABCDE1478F format"

required

/>

</label>

</div>

<div>

<label>

Passport Number:

<input

type="text"

name="compliance.passportNumber"

value={employeeData.compliance.passportNumber}

onChange={(e) => {

setEmployeeData({

...employeeData,

compliance: {

...employeeData.compliance,

passportNumber: e.target.value

}

});

}}

pattern="S[0-9]{7}"

title="please enter numeric values"

required

/>

</label>

</div>

<div>

<label>

Bank Name:

<input

type="text"

name="compliance.bankname"

value={employeeData.compliance.bankname}

onChange={(e) => {

setEmployeeData({

...employeeData,

compliance: {

...employeeData.compliance,

bankname: e.target.value

}

});

}}

pattern="[A-Z][a-z]+"

title="please enter alphabets values"

required

/>

</label>

</div>

<div>

<label>

Bank Account Number:

<input

type="text"

name="compliance.bankAccountNumber"

value={employeeData.compliance.bankAccountNumber}

onChange={(e) => {

setEmployeeData({

...employeeData,

compliance: {

...employeeData.compliance,

bankAccountNumber: e.target.value

}

});

}}

pattern="[0-9]{15}"

title="please enter numeric values"

required

/>

</label>

</div>

<div>

<label>

IFSC Code:

<input

type="text"

name="compliance.ifscCode"

value={employeeData.compliance.ifscCode}

onChange={(e) => {

setEmployeeData({

...employeeData,

compliance: {

...employeeData.compliance,

ifscCode: e.target.value

}

});

}}

pattern="[A-Z]{4}[0-9]{7}"

title="please enter numeric values"

required

/>

</label>

</div>

</div>

)}

{*/\* {/ Experiences /} \*/*}

<h3>Experiences</h3>

{employeeData.experiences.map((experience, index) => (

<div key={index}>

<div>

<label>

Company Name:

<input

type="text"

name={`companyName`}

value={experience.companyName}

onChange={(e) => handleExperienceChange(e, index)}

pattern="[a-z]+"

title="please enter alphabets values"

required

/>

</label>

</div>

<div>

<label>

companyLocation:

<input

type="text"

name={`companyLocation`}

value={experience.companyLocation}

onChange={(e) => handleExperienceChange(e, index)}

pattern="[a-z]+"

title="please enter alphabets values"

required

/>

</label>

</div>

<div>

<label>

designation:

<input

type="text"

name={`designation`}

value={experience.designation}

onChange={(e) => handleExperienceChange(e, index)}

pattern="[a-z]+"

title="please enter alphabets values"

required

/>

</label>

</div>

<div>

<label>

ctc:

<input

type="text"

name={`ctc`}

value={experience.ctc}

onChange={(e) => handleExperienceChange(e, index)}

pattern="[0-9]+.[0-9]+"

title="Please enter in 9.0 format"

required

/>

</label>

</div>

<div>

<label>

hikePercentage :

<input

type="text"

name={`hikePercentage`}

value={experience.hikePercentage}

onChange={(e) => handleExperienceChange(e, index)}

pattern="[0-9]+.[0-9]+"

title="Please enter in 9.0 format"

required

/>

</label>

</div>

<div>

<label>

ctc:

<input

type="date"

name={`joiningDate`}

value={experience.joiningDate}

onChange={(e) => handleExperienceChange(e, index)}

/>

</label>

</div>

<div>

<label>

releavingDate:

<input

type="date"

name={`releavingDate`}

value={experience.releavingDate}

onChange={(e) => handleExperienceChange(e, index)}

/>

</label>

</div>

<div>

<label>

reasonforreleaving:

<input

type="text"

name={`reasonforreleaving`}

value={experience.reasonforreleaving}

onChange={(e) => handleExperienceChange(e, index)}

pattern="[a-z]+"

title="please enter alphabets values"

required

/>

</label>

</div>

<div>

<label>

totalexperience:

<input

type="text"

name={`totalexperience`}

value={experience.totalexperience}

onChange={(e) => handleExperienceChange(e, index)}

pattern="[0-9]+.[0-9]+"

title="Please enter in 9.0 format"

required

/>

</label>

</div>

<button type="button" onClick={removeLastExperience}>

Remove Last Experience

</button>

<br></br>

<div>

............................................................................................

</div>

{*/\* {/ Add more fields for experiences here /} \*/*}

</div>

))}

<button type="button" onClick={addExperience}>

Add Experience

</button>

<button type="submit">Submit</button>

</form>

</div>

);

};

export default CreateEmployee;