

Turnitin Originality Report

Processed on: 24-Jul-2023 04:38 CDT

ID: 2136010339

Word Count: 1778

Submitted: 1

Code Plagiarism By GuturuLakshmi Neeraja

Similarity Index

20%

Similarity by Source

Internet Sources: 16%
Publications: 2%
Student Papers: 17%

5% match (Internet from 23-Aug-2022)

https://www.researchgate.net/publication/357096997_Entwicklung_eines_Konzepts_fur_eine_App_zur_Uberwindung_des_Medienbruchs_beim

3% match (Internet from 19-Dec-2022)

<http://repositorio.ug.edu.ec/bitstream/redug/55995/1/Salabarria%20Mera%20Xavier%20Arturo%20.pdf>

3% match (Internet from 12-Dec-2022)

<https://forum.djangooproject.com/t/creating-a-new-table-with-request-data/1956>

3% match (student papers from 29-Nov-2021)

[Submitted to University of Greenwich on 2021-11-29](#)

2% match (Internet from 17-May-2023)

<https://blog.openreplay.com/create-a-contacts-app-with-react-and-supabase/>

2% match (student papers from 04-May-2017)

[Submitted to Birla Institute of Technology and Science Pilani on 2017-05-04](#)

2% match (student papers from 09-May-2022)

[Submitted to University of Hertfordshire on 2022-05-09](#)

```

Home Page //importing modules import React, { useState, useEffect } from 'react' import './Main.css' import { Button } from
'primereact/button'; import { Dialog } from 'primereact/dialog'; import { Paginator } from 'primereact/paginator'; import
axios from 'axios'; import Popup from 'reactjs-popup'; import { useNavigate } from 'react-router-dom'; //functions
declaration var baseUrl = 'http://localhost:8000/Task/task-data/' export const Main = () => { const [visible, setVisible] =
useState(false) const [data, setData] = useState([]) const [name, setName] = useState('') const [isError, setIsError] =
useState('') const [assignedDate, setAssignedDate] = useState('') const [dueDate, setDueDate] = useState('') const [level,
setLevel] = useState('') const [description, setDescription] = useState('') const [input, setInput] = useState('') const [week,
setWeek] = useState('') const [form, setForm] = useState('') const [openEdit, setOpenEdit] = useState(false) const
[openDialog, setOpenDialog] = useState(false) const [idata, setIdata] = useState('') const navigate = useNavigate() async
function getTaskData() { try { await axios.get(baseUrl) .then((response) => { setData(response.data)
console.log(response.data) }) } catch (error) { console.log(error.message) setIsError(error.message) } } useEffect(() => {
getTaskData(); }, []) const Addhandler = () => { try { axios.post(`${baseUrl}`, { task_name: name, task_assigned_date:
assignedDate, task_due_date: dueDate, priority_level: level, description: description }) .then((response) => {
console.log(response.data) // setData(response.data) getTaskData(); }) SetOpenDialog(false) } catch (error) {
console.log(error.message) setIsError(error.message) } } const deleteHandler = (id) => { try { var result =
window.confirm("Are you sure ..? \n Do You Want to delte this task..?"); if (result) { axios.delete(`${baseUrl}${id}`)
.then(() => { getTaskData(); }) alert('task deleted') } } catch (error) { console.log(error) setIsError(error.message) } }
const searchHandler = (value) => { try { axios.get(`${baseUrl}filter/${value}`) .then((response) => {
console.log(response.data) // fetchHandler(week) setData(response.data) // getTaskData() }) } catch (error) {
console.log(error) setIsError(error.message) } } const editHandler = (id) => { console.log(idata) try {
axios.put(`${baseUrl}${id}`, { task_name: idata.task_name, task_assigned_date: idata.task_assigned_date,
task_due_date: idata.task_due_date, priority_level: idata.priority_level, description: idata.description, status: idata.status, })
.then((response) => { console.log(response.data) // setData(response.data) getTaskData(); }) SetOpenEdit(false); } catch
(error) { } } const getForm = (id) => { console.log(id, '*****') console.log("clicked...!") setVisible(true)
SetOpenEdit(true) try { axios.get(`${baseUrl}${id}`) .then((response) => { console.log(response.data) setIdata(
response.data) }) } catch (error) { } } const taskData = (task) => { const tdata = { task_name: task.task_name,
task_assigned_date: task.task_assigned_date, task_due_date: task.task_due_date, priority_level: task.priority_level,
description: task.description } navigate("/task",{state : tdata}); } const fetchHandler = (fetch, filter) => { try { if (filter
=== "") { console.log(fetch) axios.get(`${baseUrl}fetch/${fetch}`) .then((response) => { console.log(response.data) //
searchHandler(input) setData(response.data) // getTaskData() }) } else if (fetch === "") {
axios.get(`${baseUrl}filter/${filter}`) .then((response) => { console.log(response.data) // fetchHandler(week) setData(
response.data) // getTaskData() }) } else { console.log(fetch, filter) axios.get(`${baseUrl}full-search/${fetch}/${filter}`)
.then((response) => { console.log(response.data) // searchHandler(input) setData(response.data) // getTaskData() }) }
catch (error) { console.log(error) setIsError(error.message) } } const addfooterContent = (

```

SetOpenDialog(false)) className="p-button-text" />

); const editfooterContent = (

SetOpenEdit(false)) className="p-button-text" /> editHandler(idata.id)) autoFocus />

); //Rendering to the web page return (<>