**To do Application**

**Receive name, description values in the two textboxes and display using the expression.**

import logo from './logo.svg';

import './App.css';

import { useState } from 'react';

function App() {

  const [name, setName] = useState('');

  const [desc, setDesc] = useState('');

  return (

    <div className="App">

      <h2> To do Application </h2>

      <input type="text" placeholder= 'name' onChange={(e)=>setName(e.target.value)}/>

      <input type="text" placeholder= 'desc.' onChange={(e)=>setDesc(e.target.value)}/>

      <h2>{name}</h2>

      <h2>{desc}</h2>

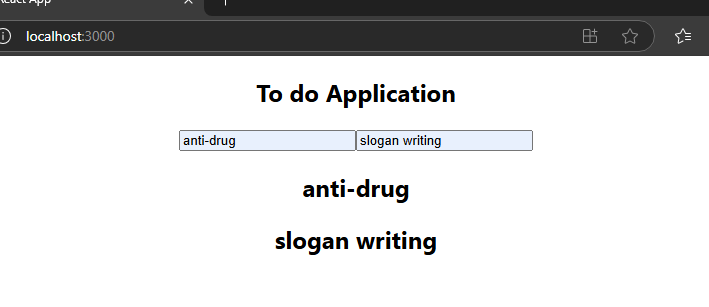
    </div>

  );

}

export default App;

Output:



**Receive name, description values in the two textboxes and send them to the server.**

**App.js**

import logo from './logo.svg';

import './App.css';

import { useState } from 'react';

import axios from 'axios';

function App() {

  const [name, setName] = useState('');

  const [desc, setDesc] = useState('');

  const handleInsert = async ()=>{

    const params = {name, desc};

    const result = await axios.post('http://localhost:5000/', params);

  }

  return (

    <div className="App">

      <h2> To do Application </h2>

      <input type="text" placeholder= 'name' onChange={(e)=>setName(e.target.value)}/>

      <input type="text" placeholder= 'desc.' onChange={(e)=>setDesc(e.target.value)}/>

      <button type='button' onClick = {handleInsert}> insert </button>

      <h2>{name}</h2>

      <h2>{desc}</h2>

    </div>

  );

}

export default App;

**TodoServer.js**

const express = require('express');

const app=express();

const cors = require('cors');

app.use(cors());

const bodyParser = require('body-parser');

app.use(bodyParser.json());

app.use(express.urlencoded({extended:true}));

app.listen(5000, ()=>{

    console.log('Server is waiting for the client.');

});

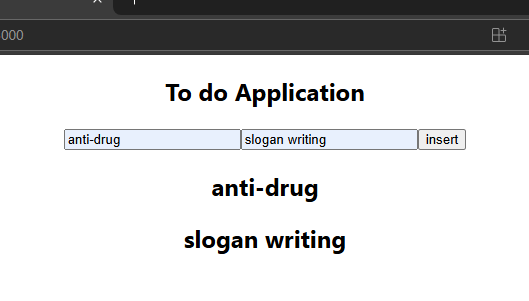
app.post('/', (req,res)=>{

    console.log('name is: ', req.body.name);

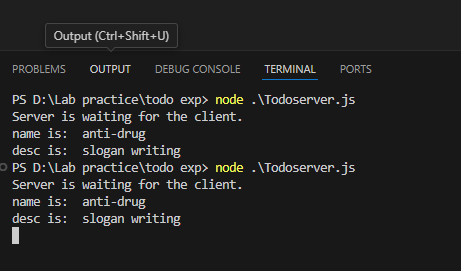
    console.log('desc is: ', req.body.desc);

});

**Output:**

****

**Server program’s terminal output line:**

****

**Receive name, description values in the two textboxes and send them to the server.**

**From the server, update the data to the MongoDB.**

**TodoServer.js**

const express = require('express');

const app=express();

const cors = require('cors');

app.use(cors());

const bodyParser = require('body-parser');

app.use(bodyParser.json());

app.use(express.urlencoded({extended:true}));

app.listen(5000, ()=>{

    console.log('Server is waiting for the client.');

});

const {MongoClient} = require('mongodb');

const client = new MongoClient('mongodb://localhost:27017');

app.post('/', (req,res)=>{

    console.log('name is: ', req.body.name);

    console.log('desc is: ', req.body.desc);

    const db =  client.db('dbtodo');

    const col = db.collection('items');

    const result = col.insertOne({'name': req.body.name, 'description': req.body.desc});

});

**Create a jsx file.**

**Give a button.**

**When the user clicks it, call a method to fetch data from the server.**

**Then display them using the map() method.**

**TodoServer.js**

const express = require('express');

const app=express();

const cors = require('cors');

app.use(cors());

const bodyParser = require('body-parser');

app.use(bodyParser.json());

app.use(express.urlencoded({extended:true}));

app.listen(5000, ()=>{

    console.log('Server is waiting for the client.');

});

const {MongoClient} = require('mongodb');

const client = new MongoClient('mongodb://localhost:27017');

app.post('/', async (req,res)=>{

    console.log('name is: ', req.body.name);

    console.log('desc is: ', req.body.desc);

    const db =  client.db('dbtodo');

    const col = db.collection('items');

    const result = await col.insertOne({'name': req.body.name, 'description': req.body.desc});

    res.end();

});

/\* app.get('/', async function (req,res){

    const output = await fn\_return\_data();

    console.log('result is: ', output);

    res.json(output);

    res.end();

});

const fn\_return\_data = async ()=>{

    const db =  client.db('dbtodo');

    const col = db.collection('items');

    const result = await col.find();

    return result.toArray();

} \*/

app.get('/', async (req, res)=>{

    const db = client.db('dbtodo');

    const result = await db.collection('items').find().toArray();

    res.json(result);

    res.end();

});

App.js

import logo from './logo.svg';

import './App.css';

import Receive\_data from './Receive\_data';

import { Route, BrowserRouter as Router, Routes } from 'react-router-dom';

import Insert from './Insert';

function App() {

  return(

  <Router>

    <Routes>

      <Route path='/' element={<Insert />}/>

      <Route path='/Receive\_data' element={<Receive\_data />}/>

    </Routes>

  </Router>

  );

}

export default App;

Insert.jsx

import logo from './logo.svg';

import './App.css';

import { useState } from 'react';

import axios from 'axios';

function Insert() {

  console.log('inside insert');

  const [name, setName] = useState('');

  const [desc, setDesc] = useState('');

  const handleInsert = async ()=>{

    const params = {name, desc};

    const result = await axios.post('http://localhost:5000/', params);

  }

  return (

    <div className="App">

      <h2> To do Application </h2>

      {console.log('just to check')}

      <input type="text" placeholder= 'name' onChange={(e)=>setName(e.target.value)}/>

      <input type="text" placeholder= 'desc.' onChange={(e)=>setDesc(e.target.value)}/>

      <button type='button' onClick = {handleInsert}> insert </button>

      <h2>{name}</h2>

      <h2>{desc}</h2>

    </div>

  );

}

export default Insert;

Receive\_data.jsx:

import axios from 'axios';

import React, { useState } from 'react';

function Receive\_data(){

    const [output, setResult] = useState([]);

    const handleClick = async ()=>{

        const res = await axios.get('http://localhost:5000');

        setResult(res.data);

    }

    return(

        <div>

            <button onClick={handleClick}> Display data </button>

            {

                output.map(item=><li>{item.name}: {item.description}</li>)

            }

        </div>

    );

}

export default Receive\_data;