**NAME : MANGESH A. GHADWAJE**

**ROLL NO:24**

**BATCH : B2**

**COURSE: AWT PRACTICAL**

**Assginment No. 6**

**Problem Statment : Build a To-do Application with React and MongoDB /Firebase.**

**CODE:**

**Server File**

**1 index.js**

const express = require("express");

const mongoose = require("mongoose");

const cors = require("cors");

const TodoModel = require("./Models/Todo");

const app = express();

app.use(cors());

app.use(express.json());

mongoose.connect("mongodb://127.0.0.1:27017/test");

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

  TodoModel.find()

    .then((result) => res.json(result))

    .catch((err) => res.json(err));

});

app.put("/update/:id", (req, res) => {

  const {id} = req.params;

  console.log(id);

  TodoModel.findByIdAndUpdate({\_id: id}, {done: true})

  .then(result => res.json(result))

  .catch(err => res.json(err))

})

app.delete('/delete/:id', (req, res) => {

  const {id} = req.params;

  TodoModel.findByIdAndDelete({\_id: id})

  .then(result => res.json(result))

  .catch(err => res.json(err))

})

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

  const task = req.body.task;

  TodoModel.create({

    task: task,

  })

    .then((result) => res.json(result))

    .catch((err) => res.json(err));

});

app.listen(3001, () => {

  console.log("server is running");

});

2. Todo.js

const mongoose = require('mongoose');

const TodoSchema = new mongoose.Schema({

    task:String,

    done: {

        type: Boolean,

        default: false

    }

})

const TodoModel = mongoose.model('Todos', TodoSchema)

module.exports = TodoModel

**React file**

Create.jsx

import React, { useState } from 'react'

import axios from 'axios'

const Create = () => {

  const [task, setTask] = useState();

  const handleAdd = () => {

        axios.post('http://localhost:3001/add', {task:task})

        .then(result => {location.reload()})

        .catch(err => console.log(err))

    }

  return (

        <div className='create\_form'>

            <input type="text" id='' placeholder='Enter Task' onChange={(e) =>setTask(e.target.value)}/>

            <button type='button' onClick={handleAdd} >Add</button>

        </div>

  )

}

1. export default Create

**2. Home.jsx**

import React, { useState, useEffect } from "react";

import Create from "./Create";

import axios from "axios";

const Home = () => {

  const [todos, setTodos] = useState([]);

  useEffect(() => {

    axios

      .get("http://localhost:3001/get")

      .then((result) => setTodos(result.data))

      .catch((err) => console.log(err));

  }, []);

  const handleEdit = (id) => {

    axios

      .put("http://localhost:3001/update/" + id)

      .then((result) => {

        location.reload()

      })

      .catch((err) => console.log(err));

  };

  const handleDelete = (id) => {

    axios

      .delete("http://localhost:3001/delete/" + id)

      .then((result) => {

        location.reload()

      })

      .catch((err) => console.log(err));

      }

  return (

    <div className="home">

      <h2> To Do List <i class="uil uil-clipboard-notes"></i></h2>

      <Create />

      {todos.length === 0 ? (

        <div>

          <h2>No Record</h2>

        </div>

      ) : (

        todos.map((todo, index) => (

          // Corrected the arrow function to return the JSX element

          <div key={index} className="task">

            <div className="checkbox" onClick={() => handleEdit(todo.\_id)}>

              {todo.done ? <i class="uil uil-check-circle"></i>

              :

              <i class="uil uil-circle"></i>

              }

              <p className={todo.done ? "line\_throungh" :"" }>

              {todo.task}

              </p>

            </div>

            <i class="uil uil-trash-alt" onClick={() =>handleDelete(todo.\_id)}></i>

          </div>

        ))

      )}

    </div>

  );

};

export default Home;

**3. app.css**

.home {

  display: flex;

  flex-direction: column;

  align-items: center;

}

h2{

  font-size: 30px;

  font-weight: bold;

}

.create\_form input {

  width: 300px;

  font-size: 15px;

  padding: 10px;

  border: none;

  border-bottom: 2px solid;

  outline: none;

  width: 200px;

}

.uil-check-circle{

  color: green;

}

.create\_form button {

  padding: 10px;

  font-size: 15px;

  border: none;

  width: 100px;

  margin-left: 5px;

  border-radius: 20px;

  background-color: rgb(250, 42, 42);

  color: white;

  cursor: pointer;

}

.task {

  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);

  margin-top: 10px;

  border: solid 2px #000;

  border-radius: 50px;

  display: flex;

  justify-content: space-between;

  width: 100%;

}

.task:hover{

  background: gainsboro;

}

p {

  font-size: 20px;

  display: inline-block;

}

i {

  cursor: pointer;

  margin: 10px;

  font-size: 30px;

}

.line\_throungh {

  text-decoration: line-through;

}