**Day 5 Assignment - 10.01.2025**

**Full Stack Development Workshop-Organised by CSE Dr.SACOE.**

**Create a Todo app using react and express.**

**Source code:**

// Directory structure suggestion:

// - server

// - server.js

// - client

// - src

// - App.js

// - components

// - TodoList.js

// - TodoItem.js

// - AddTodo.js

// - index.js

// Server-side: Express (server/server.js)

const express = require('express');

const cors = require('cors');

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

const app = express();

const PORT = 5000;

app.use(cors());

app.use(bodyParser.json());

let todos = [

{ id: 1, text: 'Learn React', completed: false },

{ id: 2, text: 'Learn Express', completed: false }

];

// Get all todos

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

res.json(todos);

});

// Add a new todo

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

const newTodo = { id: Date.now(), text: req.body.text, completed: false };

todos.push(newTodo);

res.json(newTodo);

});

// Update a todo

app.put('/api/todos/:id', (req, res) => {

const { id } = req.params;

const { completed } = req.body;

const todo = todos.find(todo => todo.id === parseInt(id));

if (todo) {

todo.completed = completed;

res.json(todo);

} else {

res.status(404).send('Todo not found');

}

});

// Delete a todo

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

const { id } = req.params;

todos = todos.filter(todo => todo.id !== parseInt(id));

res.status(204).send();

});

app.listen(PORT, () => console.log(`Server running on http://localhost:${PORT}`));

// Client-side: React (client/src/App.js)

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

import './App.css';

import TodoList from './components/TodoList';

import AddTodo from './components/AddTodo';

function App() {

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

useEffect(() => {

fetch('http://localhost:5000/api/todos')

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

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

}, []);

const addTodo = text => {

fetch('http://localhost:5000/api/todos', {

method: 'POST',

headers: { 'Content-Type': 'application/json' },

body: JSON.stringify({ text })

})

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

.then(newTodo => setTodos([...todos, newTodo]));

};

const toggleTodo = id => {

const todo = todos.find(t => t.id === id);

fetch(`http://localhost:5000/api/todos/${id}`, {

method: 'PUT',

headers: { 'Content-Type': 'application/json' },

body: JSON.stringify({ completed: !todo.completed })

})

.then(() => {

setTodos(

todos.map(t =>

t.id === id ? { ...t, completed: !t.completed } : t

)

);

});

};

const deleteTodo = id => {

fetch(`http://localhost:5000/api/todos/${id}`, { method: 'DELETE' })

.then(() => setTodos(todos.filter(t => t.id !== id)));

};

return (

<div className="App">

<h1>Todo App</h1>

<AddTodo addTodo={addTodo} />

<TodoList todos={todos} toggleTodo={toggleTodo} deleteTodo={deleteTodo} />

</div>

);

}

export default App;

// Components (e.g., client/src/components/AddTodo.js)

import React, { useState } from 'react';

function AddTodo({ addTodo }) {

const [text, setText] = useState('');

const handleSubmit = e => {

e.preventDefault();

if (text) {

addTodo(text);

setText('');

}

};

return (

<form onSubmit={handleSubmit}>

<input

type="text"

value={text}

onChange={e => setText(e.target.value)}

placeholder="Add a new todo"

/>

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

</form>

);

}

export default AddTodo;

// Components (e.g., client/src/components/TodoList.js)

import React from 'react';

import TodoItem from './TodoItem';

function TodoList({ todos, toggleTodo, deleteTodo }) {

return (

<ul>

{todos.map(todo => (

<TodoItem

key={todo.id}

todo={todo}

toggleTodo={toggleTodo}

deleteTodo={deleteTodo}

/>

))}

</ul>

);

}

export default TodoList;

// Components (e.g., client/src/components/TodoItem.js)

import React from 'react';

function TodoItem({ todo, toggleTodo, deleteTodo }) {

return (

<li>

<span

onClick={() => toggleTodo(todo.id)}

style={{

textDecoration: todo.completed ? 'line-through' : 'none',

cursor: 'pointer'

}}

>

{todo.text}

</span>

<button onClick={() => deleteTodo(todo.id)}>Delete</button>

</li>

);

}

export default TodoItem;