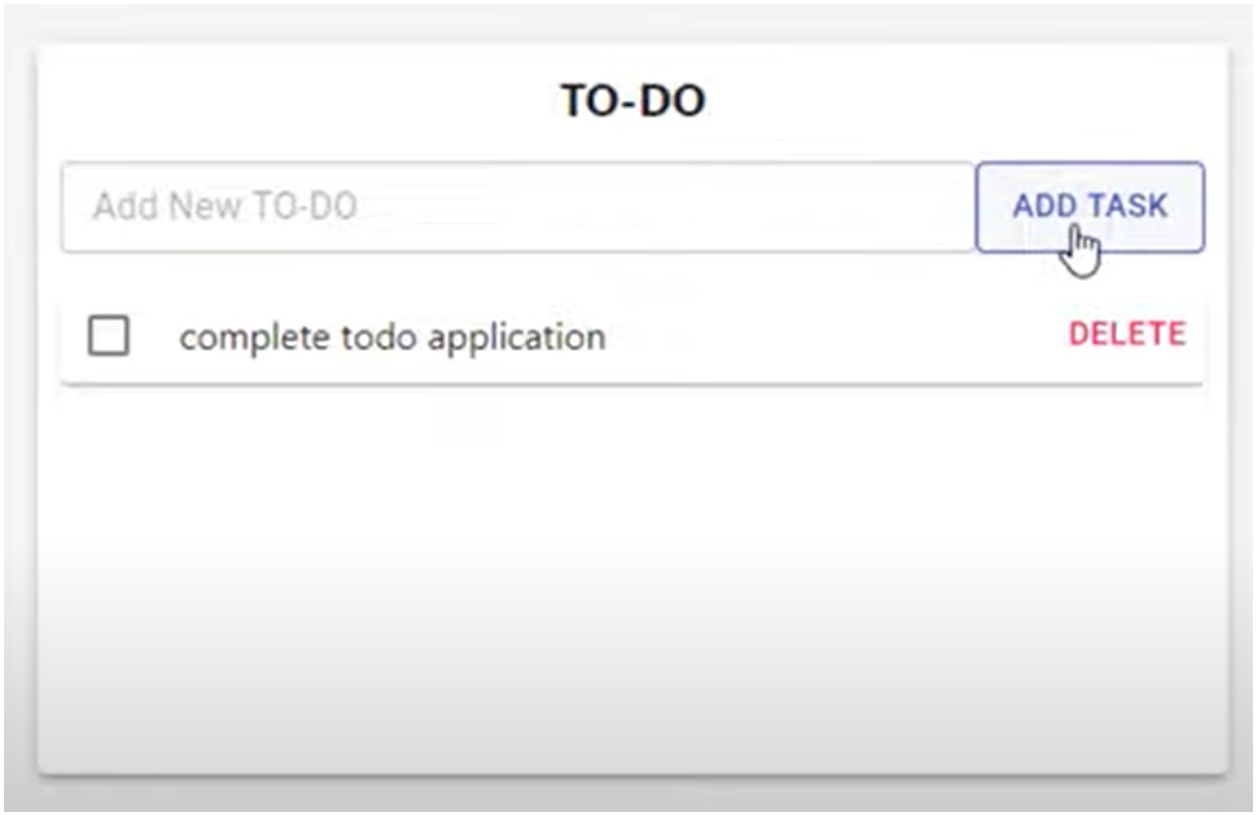
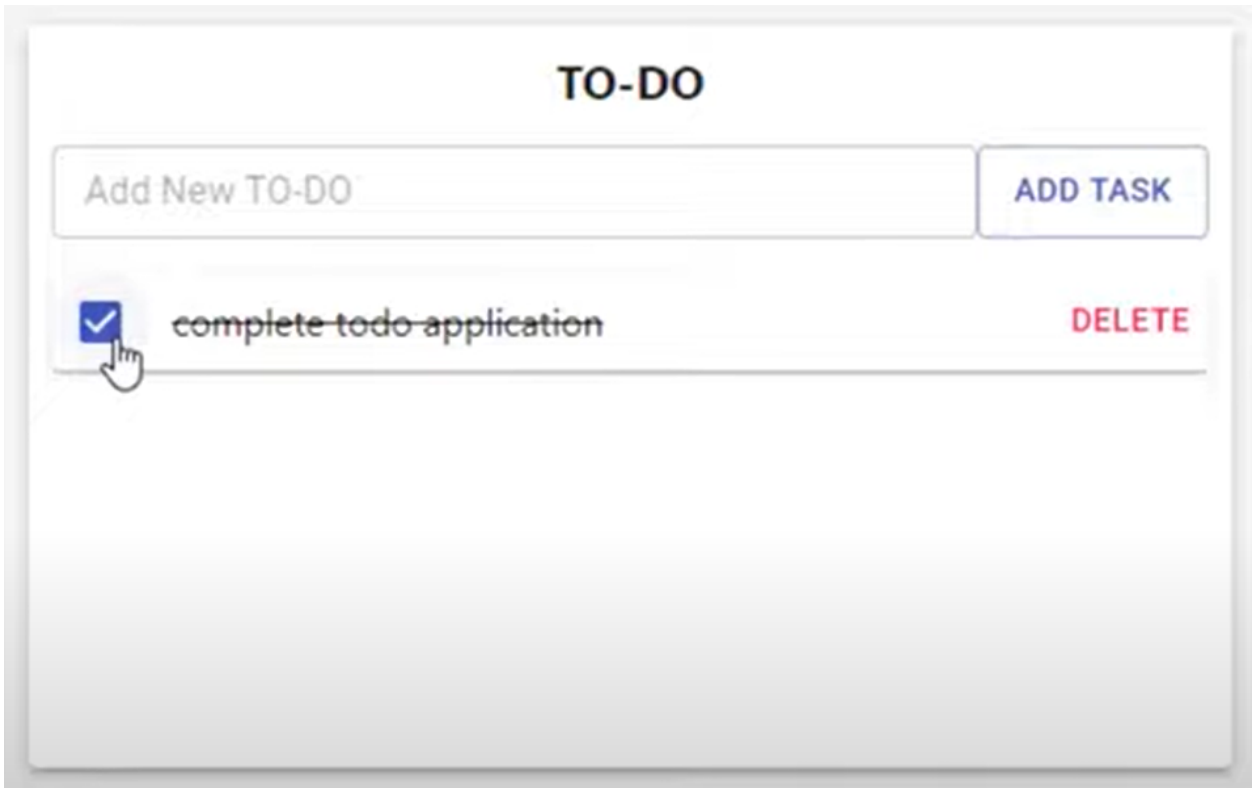

TO-DO
Add New TO-DO
ADD TASK
complete todo application
DELETE


TO-DO
Add New TO-DO
ADD TASK
complete todo application
DELETE

Tasks.js

```
import { Component } from "react";
import {
  addTask,
  getTasks,
  updateTask,
  deleteTask,
} from "./services/taskServices";

class Tasks extends Component {
  state = { tasks: [], currentTask: "" };

  async componentDidMount() {
    try {
      const { data } = await getTasks();
      this.setState({ tasks: data });
    } catch (error) {
      console.log(error);
    }
  }

  handleChange = ({ currentTarget: input }) => {
    this.setState({ currentTask: input.value });
  };

  handleSubmit = async (e) => {
    e.preventDefault();
    const originalTasks = this.state.tasks;
    try {
      const { data } = await addTask({ task: this.state.currentTask });
      const tasks = originalTasks;
      tasks.push(data);
      this.setState({ tasks, currentTask: "" });
    } catch (error) {
      console.log(error);
    }
  };

  handleUpdate = async (currentTask) => {
    const originalTasks = this.state.tasks;
    try {
      const tasks = [...originalTasks];
      const index = tasks.findIndex((task) => task._id === currentTask);
      tasks[index] = { ...tasks[index] };
```

```
            tasks[index].completed = !tasks[index].completed;
            this.setState({ tasks });
            await updateTask(currentTask, {
                completed: tasks[index].completed,
            });
        } catch (error) {
            this.setState({ tasks: originalTasks });
            console.log(error);
        }
    };

    handleDelete = async (currentTask) => {
        const originalTasks = this.state.tasks;
        try {
            const tasks = originalTasks.filter(
                (task) => task._id !== currentTask
            );
            this.setState({ tasks });
            await deleteTask(currentTask);
        } catch (error) {
            this.setState({ tasks: originalTasks });
            console.log(error);
        }
    };
}

export default Tasks;
```

TaskServices.js

```
import axios from "axios";
const apiUrl = "http://localhost:8080/api/tasks";

export function getTasks() {
   return axios.get(apiUrl);
}

export function addTask(task) {
   return axios.post(apiUrl, task);
}

export function updateTask(id, task) {
   return axios.put(apiUrl + "/" + id, task);
}

export function deleteTask(id) {
   return axios.delete(apiUrl + "/" + id);
}
```

tasks.js

```
const Task = require("../models/task");
const express = require("express");
const router = express.Router();

router.post("/", async (req, res) => {
    try {
        const task = await new Task(req.body).save();
        res.send(task);
    } catch (error) {
        res.send(error);
    }
});

router.get("/", async (req, res) => {
    try {
        const tasks = await Task.find();
        res.send(tasks);
    } catch (error) {
        res.send(error);
    }
});

router.put("/:id", async (req, res) => {
    try {
        const task = await Task.findOneAndUpdate(
            { _id: req.params.id },
            req.body
        );
        res.send(task);
    } catch (error) {
        res.send(error);
    }
});

router.delete("/:id", async (req, res) => {
    try {
        const task = await Task.findByIdAndDelete(req.params.id);
        res.send(task);
    } catch (error) {
        res.send(error);
    } });

module.exports = router;
```

task.js

```
const mongoose = require("mongoose");
const Schema = mongoose.Schema;

const taskSchema = new Schema({
   task: {
      type: String,
      required: true,
   },
   completed: {
      type: Boolean,
      default: false,
   },
});

module.exports = mongoose.model("task", taskSchema);
```

db.js

```
const mongoose = require("mongoose");

module.exports = async () => {
   try {
      const connectionParams = {
         useNewUrlParser: true,
         useCreateIndex: true,
         useUnifiedTopology: true,
      };
      await mongoose.connect(
         "mongodb://localhost/todo-app",
         connectionParams
      );
      console.log("Connected to database.");
   } catch (error) {
      console.log("Could not connect to database.", error);
   }
};
```