Project Setup for Backend

Set up:

npm init -y: install package.json

npm i express: node_modules, package-lock.json

create backend folder

create server.js (in backend folder): root file

set up git (pdf file in GitHub)

create .gitignore (outside backend): node_modules + .env

commit 'Initial Commit'

Create Express server:

```
const express = require('express');
const app = express();

const port = process.env.PORT || 5000;
app.listen(port, () => {
    console.log(`Server Running at ${port}`);
```

Create ‘start’ in scripts with nodemon:

npm i nodemon -g (or -D for each project)

```
"main": "server.js",
"scripts": {
"start": "node backend/server.js",
"backend": "nodemon backend/server.js"}
```

Check: npm start, or npm run backend

Setup Insomia for API testing:

```
// Routes
   app.get('/', (req, res) => {
   res.send('Home Page');}
```

Create a folder in Insomia:

GET → http://localhost:5000/ → Send

Setup MongoDB:

login in MongoDB

free option *(remember to change the name ‘Cluster’)*

username and password

IP address: ‘Allow to access anywhere’ in Network Access click on ‘Connect’ in DataBase → Connect mongoDB using native drivers → Copy Link

npm i dotenv → (outside backend folder) create .env file →

MONGO_URL=<Link> *(remember to change the <password>, before ‘?’ type ‘<name of project>’)*

Example:

MONGO_URI=mongodb+srv://taskmanagerapp:Duyen123456@taskmanagerapp.r9pv6re.mongodb.net/Task_Manager_App?retryWrites=true&w=majority

Connect to MongoDB:

npm i mongoose

inside backend folder, create config folder → connectDB.js

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


const connectDB = async () => {
  try {
    // mongodb connection string
    const connect = await mongoose.connect(process.env.MONGO_URI);


    console.log(`MongoDB Connected`);
  } catch (err) {
    console.log(err);
```

```
        process.exit(1);
    }
};

module.exports = connectDB;
```

In server.js:

```
const dotenv = require('dotenv').config();
const connectDB = require('./config/connectDB');
```

(Remember there are some changes, copy and paste here, check: mongo → server running)

```
const startServer = async () => {
    try {
        await connectDB();
        app.listen(port, () => {
            console.log(`Server Running at ${port}`);
        });
    } catch (error) {
        console.log(error);
    }
}

startServer();
```

2nd method to connect MongoDB: (Do NOT need to create config folder)

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


mongoose
    .connect(process.env.MONGO_URI)
    .then(() => {
        app.listen(port, () => {
            console.log(`Server Running at ${port}`);
    })
    .catch((err) => console.log(error));
```

Create task model and schema:

inside backend folder, create models folder → taskModel.js:

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


const taskSchema = new mongoose.Schema(
  {
    name: {
        type: String,
        required: [true, 'Please enter task name']
    },
    completed: {
```

```
                type: Boolean,

                required: true,

                default: false

            }

        },

        {

            timestamps: true

        }

    );


    const Task = mongoose.model('Task', taskSchema);


    module.exports = Task;
```

Create routes and test in Insomia:

in backend folder, create controllers folder → taskController.js

```
    const Task = require('../models/taskModel');


    // Create a task

    const createTask = async (req, res) => {

        try {

            const task = await Task.create(req.body);
            res.status(200).json(task);
        } catch (error) {
            res.status(500).json({
                message: error
            });
        }
    };

    // Get and read all tasks
    const getTasks = async (req, res) => {
        try {
            const tasks = await Task.find();
            res.status(200).json(tasks);
        } catch (error) {
            res.status(500).json({
                message: error
            });
        }
    };

    // Get a single task
    const getTask = async (req, res) => {
        try {
            const { id } = req.params;
            const tasks = await Task.findById(id);
            if (!tasks) {
                return res.status(404).json({
                    message: 'No task found with that ID'
                });
            }
            res.status(200).json(tasks);
        } catch (error) {
            res.status(500).json({
                message: error
            });
        }
    };

    // Delete a single task
    const deleteTask = async (req, res) => {
        try {
            const { id } = req.params;
            const tasks = await Task.findByIdAndDelete(id);
            if (!tasks) {
                return res.status(404).json({
                    message: 'No task found with that ID'
                });
            }
            res.status(200).json(tasks);
        } catch (error) {
            res.status(500).json({
                message: error
            });
        }
    };

    // Update a single task
    const updateTask = async (req, res) => {
        try {
            const { id } = req.params;
            const tasks = await Task.findByIdAndUpdate(
                { _id: id }, req.body, { new: true, runValidators: true }
            );
            if (!tasks) {
                return res.status(404).json({
                    message: 'No task found with that ID'
                });
            }
            res.status(200).json(tasks);
        } catch (error) {
            res.status(500).json({
                message: error
            });
        }
    };

    module.exports = {
        createTask,
        getTasks,
        getTask,
        deleteTask,
        updateTask
    };
```

In backend folder, create routes folder → taskRoute.js

```
const express = require('express');
const { createTask, getTasks, getTask, deleteTask, updateTask} =
require('../controllers/taskController');
const Task = require('../models/taskModel');
const router = express.Router();
```

```
router.route('/').post(createTask).get(getTasks);

router.route('/:id').get(getTask).delete(deleteTask).put(updateTask);



module.exports = router;
```

In server.js:

```
//Middleware:

app.use(express.json());

// Access to form in Insomia

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


// Added to complete

const taskRoutes = require('./routes/taskRoute');

app.use("/api/tasks", taskRoutes);
```