Here’s a **complete guide** to building a simple **frontend and backend project using Ubuntu**. This example project is a **Task Manager** app built using:

* **Frontend:** React.js
* **Backend:** Node.js + Express
* **Database:** MongoDB
* **Deployment (optional):** Docker or native Ubuntu services

**🧱 Project Structure**

pgsql

CopyEdit

task-manager/

├── backend/

│ ├── controllers/

│ ├── models/

│ ├── routes/

│ ├── .env

│ ├── server.js

├── frontend/

│ ├── public/

│ ├── src/

│ ├── package.json

**✅ Step-by-Step Setup Guide on Ubuntu**

**1. 🔧 Prerequisites**

Install the required software:

bash

CopyEdit

sudo apt update

sudo apt install -y nodejs npm mongodb

Install MongoDB and start it:

bash

CopyEdit

sudo systemctl start mongodb

sudo systemctl enable mongodb

**🖥️ Backend Setup (Node.js + Express)**

**2. Create Backend Folder**

bash

CopyEdit

mkdir -p task-manager/backend && cd task-manager/backend

npm init -y

npm install express mongoose dotenv cors

**3. Backend Code**

**server.js**

js

CopyEdit

const express = require('express');

const mongoose = require('mongoose');

const cors = require('cors');

require('dotenv').config();

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

const app = express();

app.use(cors());

app.use(express.json());

app.use('/api/tasks', taskRoutes);

mongoose.connect(process.env.MONGO\_URI, {

useNewUrlParser: true, useUnifiedTopology: true

}).then(() => {

app.listen(5000, () => console.log('Server running on port 5000'));

}).catch(err => console.error(err));

**.env**

ini

CopyEdit

MONGO\_URI=mongodb://127.0.0.1:27017/taskdb

**models/Task.js**

js

CopyEdit

const mongoose = require('mongoose');

const TaskSchema = new mongoose.Schema({

title: { type: String, required: true },

completed: { type: Boolean, default: false }

});

module.exports = mongoose.model('Task', TaskSchema);

**routes/taskRoutes.js**

js

CopyEdit

const express = require('express');

const router = express.Router();

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

// Create task

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

const task = new Task(req.body);

try {

const savedTask = await task.save();

res.json(savedTask);

} catch (err) {

res.status(400).json({ error: err.message });

}

});

// Get tasks

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

const tasks = await Task.find();

res.json(tasks);

});

// Update task

router.put('/:id', async (req, res) => {

try {

const updated = await Task.findByIdAndUpdate(req.params.id, req.body, { new: true });

res.json(updated);

} catch (err) {

res.status(400).json({ error: err.message });

}

});

// Delete task

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

try {

await Task.findByIdAndDelete(req.params.id);

res.json({ message: 'Task deleted' });

} catch (err) {

res.status(400).json({ error: err.message });

}

});

module.exports = router;

Run the backend:

bash

CopyEdit

node server.js

**🌐 Frontend Setup (React)**

**4. Create React App**

Open a new terminal:

bash

CopyEdit

cd ~/task-manager

npx create-react-app frontend

cd frontend

npm install axios

**5. React Code**

**src/App.js**

js

CopyEdit

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

import axios from 'axios';

function App() {

const [tasks, setTasks] = useState([]);

const [title, setTitle] = useState('');

const fetchTasks = async () => {

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

setTasks(res.data);

};

const addTask = async () => {

if (title.trim() === '') return;

await axios.post('http://localhost:5000/api/tasks', { title });

setTitle('');

fetchTasks();

};

const toggleTask = async (id, completed) => {

await axios.put(`http://localhost:5000/api/tasks/${id}`, { completed: !completed });

fetchTasks();

};

const deleteTask = async (id) => {

await axios.delete(`http://localhost:5000/api/tasks/${id}`);

fetchTasks();

};

useEffect(() => {

fetchTasks();

}, []);

return (

<div style={{ padding: 30 }}>

<h2>Task Manager</h2>

<input

placeholder="Enter task"

value={title}

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

/>

<button onClick={addTask}>Add</button>

<ul>

{tasks.map(task => (

<li key={task.\_id}>

<span

onClick={() => toggleTask(task.\_id, task.completed)}

style={{ textDecoration: task.completed ? 'line-through' : 'none', cursor: 'pointer' }}

>

{task.title}

</span>

<button onClick={() => deleteTask(task.\_id)} style={{ marginLeft: 10 }}>❌</button>

</li>

))}

</ul>

</div>

);

}

export default App;

Run the frontend:

bash

CopyEdit

npm start