13. To-Do List

Creating a To-Do List application using the MEAN stack (MongoDB, Express.js, Angular, Node.js) is a great project. Below is a high-level overview and some code snippets to guide you through the process:

Project Setup and Structure

Set up a new project folder and structure for your To-Do List app. Install the required Node.js packages and create a basic Angular application.

```
# Create a new Angular application

                ng new todo-list-app
```

- Backend (Node.js & Express.js)

Create the backend of your To-Do List app using Node.js and Express.js.

Installation of Packages

Install the necessary packages for Express.js, Mongoose (for MongoDB), and other dependencies.

```
npm install express mongoose cors
```

Setting up Express.js

Create your Express.js server, set up middleware, and handle routes.

- **javascript**

```
// server.js

const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');

const app = express();

// Middleware
app.use(express.json());
app.use(cors());

// Database connection
mongoose.connect('mongodb://localhost/todo-list-app', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
  useCreateIndex: true,
});
```

```
// Define Mongoose models for Task data
const Task = mongoose.model('Task', {
  title: String,
  priority: String,
  dueDate: Date,
  completed: Boolean,
});

// Routes for managing tasks
app.post('/api/tasks', async (req, res) => {
  // Create a new task
  // Save the task to the database
});

app.get('/api/tasks', async (req, res) => {
  // Retrieve a list of tasks
});

app.put('/api/tasks/:id', async (req, res) => {
  // Update a task
  // Save the updated task to the database
});

app.delete('/api/tasks/:id', async (req, res) => {
  // Delete a task
  // Remove the task from the database
});
```

- **Frontend (Angular)**

Create the frontend of your To-Do List app using Angular. Design the user interface for managing tasks, setting priorities, and specifying due dates.

Design and UI

Design the user interface for your To-Do List using Angular components, templates, and styles.

Task Management

Create components and forms for users to manage tasks, including specifying the task title, priority (e.g., high, medium, low), and due date.

- **typescript**

```
// task-management.component.ts
import { Component } from '@angular/core';
import { TaskService } from './task.service';

@Component({
  selector: 'app-task-management',
  templateUrl: './task-management.component.html',
})
export class TaskManagementComponent {
  title: string;
  priority: string;
  dueDate: Date;

  constructor(private taskService: TaskService) {}

  addTask() {
    this.taskService.addTask(this.title, this.priority, this.dueDate);
  }
}
```

MongoDB

Create a MongoDB database to store task data.

Putting It All Together

Integrate the frontend and backend by making API requests from Angular components to Node.js routes. Ensure that you handle task creation, updating, and deletion properly.

Building a To-Do List app is a practical and useful project. You can further improve it by adding features like task categories, reminders, notifications, and task sharing among users if needed.