30. Task Reminder Bot

Creating a Task Reminder Bot for Slack or Telegram using the MEAN stack (MongoDB, Express.js, Angular, Node.js) is an interesting and practical project. Below is a high-level overview of how to build such a bot, along with some code snippets to guide you:

Project Setup and Structure

Set up a new project folder and structure for your Task Reminder Bot. Install the required Node.js packages and create a basic Angular application.

```
# Create a new Angular application

ng new reminder-bot-app
```

- Backend (Node.js & Express.js)

Create the backend of your Task Reminder Bot 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 axios
```

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 axios = require('axios');

const app = express();

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

// Database connection
mongoose.connect('mongodb://localhost/reminder-bot-app', {
  useNewUrlParser: true,
```

```
  useUnifiedTopology: true,
  useCreateIndex: true,
});

// Define Mongoose models for Reminder and User data
const Reminder = mongoose.model('Reminder', {
  userId: mongoose.Schema.Types.ObjectId,
  text: String,
  dueDate: Date,
  // Add more reminder-related fields as needed
});

const User = mongoose.model('User', {
  username: String,
  // Add more user-related fields as needed
});

// Routes for managing reminders and user profiles
app.post('/api/register', async (req, res) => {
  // Register a new user
  // Store user information in the database
```

```
});

app.post('/api/login', async (req, res) => {
  // Authenticate user and generate a JWT token
});

app.post('/api/reminders', async (req, res) => {
  // Create a new reminder
  // Save the reminder to the database
});

// Create routes for viewing and managing reminders
```

- ## Frontend (Angular)

Create the frontend of your Task Reminder Bot using Angular. Design the user interface for setting reminders, viewing reminders, and user accounts.

Design and UI

Design the user interface for your Task Reminder Bot using Angular components, templates, and styles.

Reminder Management

Create components and forms for users to set reminders, view their reminders, and manage them.

User Authentication

Implement user registration and login functionality.

- **typescript**

```
// reminder.component.ts
import { Component } from '@angular/core';
import { ReminderService } from './reminder.service';

@Component({
  selector: 'app-reminder',
  templateUrl: './reminder.component.html',
})
export class ReminderComponent {
  text: string;
  dueDate: Date;

  constructor(private reminderService: ReminderService) {}
```

```
  createReminder() {
    this.reminderService.createReminder(this.text, this.dueDate);
  }
}
```

MongoDB

Create a MongoDB database to store user profiles and reminders.

Integration with Slack or Telegram

Integrate your bot with the Slack or Telegram API to send and receive messages. You can use libraries or SDKs provided by these platforms.

Putting It All Together

Integrate the frontend and backend by making API requests from Angular components to Node.js routes. Ensure that you handle reminder creation, user authentication, and reminder management properly.

Building a Task Reminder Bot is a practical project that can be extended with features like recurring reminders, snooze options, notification preferences, and integration with other messaging platforms for a more versatile reminder bot.