

## 16. Stock Portfolio Tracker

Creating a Stock Portfolio Tracker using the MEAN stack (MongoDB, Express.js, Angular, Node.js) is a valuable project that can help you monitor and analyze your stock investments. Below is a high-level overview of how to build such a tool, along with some code snippets to guide you:

### Project Setup and Structure

Set up a new project folder and structure for your Stock Portfolio Tracker. Install the required Node.js packages and create a basic Angular application.

#### # Create a new Angular application

```
ng new stock-portfolio-tracker
```

### - Backend (Node.js & Express.js)

Create the backend of your Stock Portfolio Tracker 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/stock-portfolio-tracker', {  
  useNewUrlParser: true,  
  useUnifiedTopology: true,  
  useCreateIndex: true,
```

```
});
```

```
// Define Mongoose models for User and Portfolio data
```

```
const User = mongoose.model('User', {  
  username: String,  
  password: String, // Use hashing for security  
  portfolios: [{ type: mongoose.Schema.Types.ObjectId, ref: 'Portfolio'  
}],  
});
```

```
const Portfolio = mongoose.model('Portfolio', {  
  name: String,  
  stocks: [{ symbol: String, shares: Number }],  
});
```

```
// Routes for user registration, login, and portfolio management
```

```
app.post('/api/register', async (req, res) => {  
  // Register a new user  
  // Store hashed password in the database  
});
```

```
app.post('/api/login', async (req, res) => {
```

```
// Authenticate user and generate a JWT token

});

app.post('/api/portfolios', async (req, res) => {

// Create a new portfolio for the user

// Save the portfolio to the database

});

app.get('/api/portfolios', async (req, res) => {

// Retrieve a list of user's portfolios

});

// Create similar routes for adding, updating, and deleting stocks in a portfolio
```

## **- Frontend (Angular)**

Create the frontend of your Stock Portfolio Tracker using Angular. Design the user interface for managing portfolios, adding stocks, and analyzing investments.

## Design and UI

Design the user interface for your Stock Portfolio Tracker using Angular components, templates, and styles.

## Portfolio Management

Create components and forms for users to manage portfolios, including specifying the portfolio name and adding stocks.

## Stock Analysis

Design components for analyzing stock investments, including tracking performance, calculating returns, and displaying relevant statistics.

## User Authentication

Implement user registration and login functionality.

- **typescript**

```
// portfolio-management.component.ts
```

```
import { Component } from '@angular/core';
```

```
import { PortfolioService } from './portfolio.service';
```

```
@Component({
```

```
  selector: 'app-portfolio-management',
```

```
  templateUrl: './portfolio-management.component.html',
```

```
  })  
  
  export class PortfolioManagementComponent {  
  
    portfolioName: string;  
  
    constructor(private portfolioService: PortfolioService) {}  
  
    createPortfolio() {  
      this.portfolioService.createPortfolio(this.portfolioName);  
    }  
  }  
}
```

## MongoDB

Create a MongoDB database to store user profiles, portfolio data, and stock information.

## Stock Data

You may need to use an external API or service to fetch real-time or historical stock data. Services like Alpha Vantage or Yahoo Finance can be integrated to retrieve stock information.

## Putting It All Together

Integrate the frontend and backend by making API requests from Angular components to Node.js routes. Ensure that you handle

portfolio creation, stock management, and investment analysis properly.

Building a Stock Portfolio Tracker is a practical and informative project. You can expand it with additional features such as stock alerts, financial news integration, and portfolio sharing among users.