17. Feedback Survey App

Creating a Feedback Survey App using the MEAN stack (MongoDB, Express.js, Angular, Node.js) is a great project. It allows you to build a tool for creating and collecting feedback from users. Below is a high-level overview of how to build such an application, along with some code snippets to guide you:

Project Setup and Structure

Set up a new project folder and structure for your Feedback Survey App. Install the required Node.js packages and create a basic Angular application.

```
# Create a new Angular application

        ng new feedback-survey-app
```

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

Create the backend of your Feedback Survey 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/feedback-survey-app', {

  useNewUrlParser: true,

  useUnifiedTopology: true,

  useCreateIndex: true,
});

// Define Mongoose models for Survey and Response data
const Survey = mongoose.model('Survey', {
  title: String,
  questions: [String],
  active: Boolean,
});

const Response = mongoose.model('Response', {
  surveyId: mongoose.Schema.Types.ObjectId,
  answers: [String],
});

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

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

app.get('/api/surveys/:id', async (req, res) => {
  // Retrieve a specific survey and its questions
});

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

- **Frontend (Angular)**

Create the frontend of your Feedback Survey App using Angular. Design the user interface for creating and participating in surveys.

Design and UI

Design the user interface for your Feedback Survey App using Angular components, templates, and styles.

Survey Creation

Create components and forms for users to create surveys, including specifying the survey title and adding questions.

Survey Participation

Design components for users to participate in surveys by answering questions.

- **typescript**

```
// survey-participation.component.ts
import { Component } from '@angular/core';
import { SurveyService } from './survey.service';

@Component({
  selector: 'app-survey-participation',
  templateUrl: './survey-participation.component.html',
})
export class SurveyParticipationComponent {
  survey: any;
  answers: string[];

  constructor(private surveyService: SurveyService) {}
  loadSurvey(surveyId: string) {
    this.surveyService.getSurvey(surveyId).subscribe((data) => {
      this.survey = data;
      this.answers = new Array(data.questions.length).fill('');
    });
  }

  submitResponse() {
    this.surveyService.submitResponse(this.survey._id, this.answers);
  }
}
```

MongoDB

Create a MongoDB database to store survey and response 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 survey creation, participation, and response collection properly.

Building a Feedback Survey App is a practical and interactive project. You can expand it with additional features such as user authentication, survey result analysis, survey sharing, and result visualization.