Documentation for Flat Rock Technology TEST TASK Famous Quote Quiz

Used technologies:

Programming languages - C#, JavaScript

Database – MSSQL

Back end - ASP.NET Core, Entity Framework Core, Web API, Identity

Front end – React

Part 1 (FE).

The Login component provides a user interface for logging into the application. It includes form inputs for email and password, and handles form submission by making an API request to authenticate the user.

The Register component provides a user interface for registering a new user account. It includes form inputs for the user's first name, last name, email, and password. The component handles form submission by making an API request to register the user and displays any error messages that occur during the process.

The MainPage component is the primary component for a quiz application that allows users to select the type of questions (binary or multiple choice) and take a quiz accordingly. The application consists of two main tabs: "Quiz" and "Settings".

The MultipleChoiceQuestionPage component is designed to display a multiple-choice quiz question where users can select an author from a list of options to identify who said a given quote. The component fetches the quiz data from an API and manages the quiz state, including the user's answers and whether the result should be displayed.

The BinaryChoiceQuestion component is a React component designed to display a binary choice quiz question. Users can answer whether a randomly selected author said a given quote. The component fetches the quiz data from an API, manages the quiz state, displays the quiz results, and allows navigation to the next question.

Dependencies React:

A JavaScript library for building user interfaces.

axios: A promise-based HTTP client for the browser and Node.js.

React-Bootstrap: A library that provides Bootstrap components as React components.

Part 2 (BE).

Overview

The QuizQuestionService class in the FlatRockProject.Application.Services.Quiz namespace is a service class that handles operations related to quiz questions. It includes methods for creating quiz questions, adding authors and quotes, and retrieving binary and multiple-choice questions from the database. The service interacts with the database using Entity Framework Core.

Dependencies

FlatRockProject.Application.Models: Contains data models used in the application.

FlatRockProject.Application.Services.Abstract: Defines abstract service classes.

FlatRockProject.Infrastructure.Entities: Contains entity classes that map to database tables.

FlatRockProject.Infrastructure.Persistence: Contains the database context class (FRPDbContext).

Microsoft.EntityFrameworkCore: Provides Entity Framework Core features.

Optional: Provides utility methods for handling optional values.

Methods

CreateQuizAsync

Creates a new quiz with the provided quote and questions.

AddAuthor

Adds a new author to the database.

AddQuote

Adds a new quote to the database associated with a specific author.

AddBinaryQuestion

Adds a new binary choice question to the database.

AddMultipleChoiceQuestion

Adds a new multiple-choice question to the database.

GetLastBinaryChoiceQuestionAsync

Retrieves the last binary choice question added to the database.

GetLastMultipleChoiceQuestionAsync

Retrieves the last multiple-choice question added to the database.

GetBinaryChoiceQuestionAsync

Retrieves a specific binary choice question by its ID.

GetMultipleChoiceQuizQuestionAsync

Retrieves a specific multiple-choice question by its ID.

GetBinaryChoiceQuestionsAsync

Retrieves a list of binary choice questions.\

GetMultipleChoiceQuizQuestionsAsync

Retrieves a list of multiple-choice questions.

The UsersService class in the FlatRockProject.Application.Services.User namespace is responsible for user-related operations such as user login and registration. It interacts with the ASP.NET Core Identity framework for user management and authentication.

Dependencies

Microsoft.AspNetCore.Identity: Provides user management and authentication functionality.

FlatRockProject.Infrastructure.Entities.User: Represents the user entity.

FlatRockProject.Application.Models: Contains data transfer objects (DTOs) used by the service.

FlatRockProject.Infrastructure.Config: Contains configurations related to the application.

Authenticates a user based on the provided credentials and generates a JWT token for the authenticated user.

How to start Project:

Back end – please change connectionString in appsetting.json with yours and set API as your start project.

Front end – I have react project which is different solution and all you need is to write this command in cmd “**npm start”**