**Project name:** Quiz application

**Created by:** Nacafli Aygun, Huseynzade Aydan, Ahmedova Amina

**GitHub:**

1. **Lib/pages/home\_page.dart**

The HomePage widget is where users land first in the app. It shows a list of available quizzes and lets users start a quiz by navigating to a new page.

* **Stateful Widget**: The page reacts to changes in quiz data. It fetches quiz details asynchronously when the page is loaded, using a function called QuizDataLoader.loadQuizData().
* **Displaying Quizzes**:
  + While the data is loading, users see a loading spinner.
  + If there’s an error, an error message is shown.
  + If no quizzes are available, it displays a friendly "No quizzes available" message.
* **Quiz Cards**:
  + Each quiz is displayed in a card that shows its title, a brief description, and the number of questions.
  + The card also has a "Start Quiz" button that users can click to dive into the quiz.
* **Navigation**: When users tap "Start Quiz", they are taken to a new page to begin answering questions.
* **Design**: The quiz cards are visually appealing with borders, shadows, and buttons that make it easy for users to interact with the app.

1. **Lib/pages/quiz\_page.dart**

The QuizPage widget is where users take the quiz. It displays one question at a time, allows users to select an option, and navigate through the quiz. When the quiz is complete, it redirects to the results page.

#### Key Features:

* **Stateful Widget**: The page uses a StatefulWidget to handle user interaction and track their answers as they progress through the quiz.
* **Question Navigation**:
  + The quiz progresses one question at a time.
  + A "Next Question" button moves to the next question or shows a "Finish Quiz" button when the quiz ends.
* **Answer Selection**:
  + Users can select an answer option for each question.
  + If no answer is selected, a snack bar will prompt the user to choose an option.
* **Progress Tracker**:
  + A linear progress bar shows how far the user has progressed through the quiz.
  + The current question number and total number of questions are displayed.
* **Option Display**:
  + Each question's options are displayed as tappable items. The selected option is highlighted.
* **Results Navigation**:
  + After completing all questions, the app navigates to the ResultPage to show the user's results.

#### Visuals:

* **Question Images**: If available, the question’s image is displayed above the options.
* **Option Styling**: Selected options are visually highlighted with a checkmark.

1. **Lib/pages/result\_page.dart**

The ResultPage is displayed after the user finishes a quiz. It shows their score, feedback, and options to restart the quiz or go back home.

#### Key Features:

* **Score Calculation**:
  + The page calculates the number of correct answers and the percentage score.
  + Feedback is displayed based on the score, with messages like "Excellent!" or "Keep practicing!" depending on the performance.
* **Confetti Animation**:
  + If the score is above 75%, confetti animates to celebrate the user's success.
* **UI Elements**:
  + **Score Circle**: Shows the user's score percentage and the number of correct answers.
  + **Feedback Message**: Displays a motivational message based on the user's performance.
  + **Quiz Summary**: Shows the total correct answers out of the total questions.
* **Actions**:
  + Users can restart the quiz or return to the home page.
  + If there are wrong answers, users can view them by clicking the "View Wrong Answers" button.
* **Design**:
  + A colorful and engaging design with confetti, score circle, and action buttons that improve the user experience.

1. **Lib/pages/wrong\_answers\_page.dart**

The WrongAnswersPage is displayed when users want to review the answers they got wrong during the quiz. It shows each incorrect question, the user's answer, and the correct answer for comparison.

#### Key Features:

* **Wrong Answers List**:
  + The page loops through the user's answers and compares them with the correct answers.
  + For each question that was answered incorrectly, it stores the question, the user's answer, and the correct answer in a list.
* **UI Elements**:
  + Each wrong answer is displayed in a card format with the question, the user's incorrect answer (highlighted in red), and the correct answer (highlighted in green).
* **Design**:
  + Simple layout with padding and cards for better readability.
  + Clear distinction between the user's answer and the correct answer using color coding (red for wrong, green for correct).

1. **Lib/models/quiz\_model.dart**

This code defines three models for a quiz app:

1. **Option**: Represents an answer option for a question, holding the text of the option.
2. **Question**: Represents a question, including the question text, options, correct answer index, and an optional image URL.
3. **Quiz**: Represents a quiz, including the title, description, and a list of questions.

Each model has a fromJson method to convert data from JSON into objects.

1. **Lib/sevices/quiz\_data\_loader.dart**

This code defines a QuizDataLoader class that loads quiz data from a local JSON file:

**loadQuizData**: This static method reads the quiz\_data.json file from the app's assets, decodes it from JSON, and converts it into a list of Quiz objects using the fromJson method.

The rootBundle.loadString method is used to load the JSON file, and json.decode is used to parse the content.

1. **Lib/main.dart**

This is the main entry point of the quiz app:

* **main()**: Sets the status bar to transparent and runs the QuizApp.
* **QuizApp**: Defines the app with a custom theme (colors, fonts, buttons) and sets the HomePage as the starting screen.
* **Theme**: Custom pink, blue, and green colors with rounded buttons and cards for a playful, pastel aesthetic.