# QUIZ APPLICATION

**What will our program do? (Objectives)**

- Helps us to train our brain in mental mathematics.
- User sees a number of questions that he/she needs to answer correctly in 30 seconds.

**Why quiz app?**

Training the brain to do fast computations helps improve mental ability, especially in small children. The app aims to bring out the positive in technology under the hands of small children where they can benefit from the app. The app also generates an interest to test oneself to achieve higher scores.

**There are 2 parts to the project:**

- Frontend (Acadview Link)
- Application Flow Logic

(a) Open link. Find activity_main.xml and activity_screen_splash.xml XML code.
(b) Call activity_main.xml and activity_screen_splash.xml from MainActivity.java and SplashScreen.java respectively to set the layout(if you made and empty activity this code would automatically written there).
(c) Run that and Test.
(d) You can add features on your own thinking. But for hint we are providing you a step wise instruction.

**(I) Activities**

1) Splash Screen (first screen)
2) Main Activity (showing the quiz)
3) Score Activity (shows previous scores)

**Workflow**

1) When app starts it shows Splash Screen.
2) After Splash Screen, Main Activity consisting of title as “Brain Trainer” and a question with 4 options to choose from.
3) Clicking on any of the options starts a timer of 30 seconds, marking the start of the quiz.
4) Questions are seen at the top-center position of the screen and the user needs to select one of the 4 options provided.
5) On selecting an option, the score gets updated depending on whether the user answered correctly or not..
6) When the quiz is over, user is shown his/her score and given an option to try again.

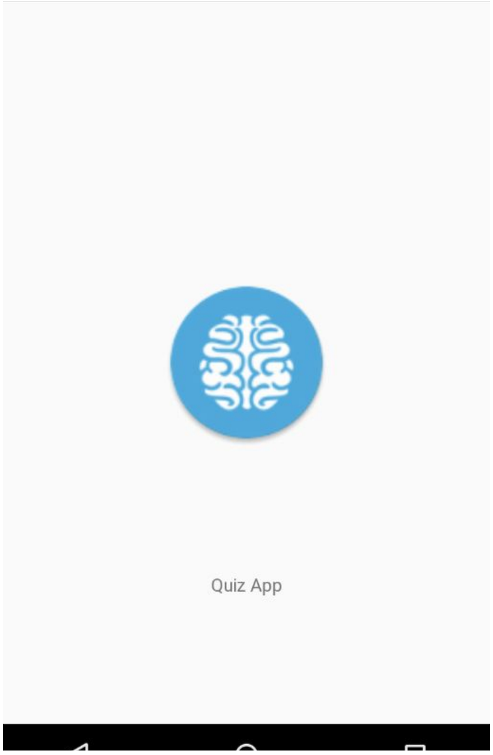
**Tasks**

Broadly the project should have the following features

1) Splash Screen
2) Main Activity (includes the app logic)

### 1.) Splash Screen

Layout of Splash Screen will be provided, which will contain an image and some text

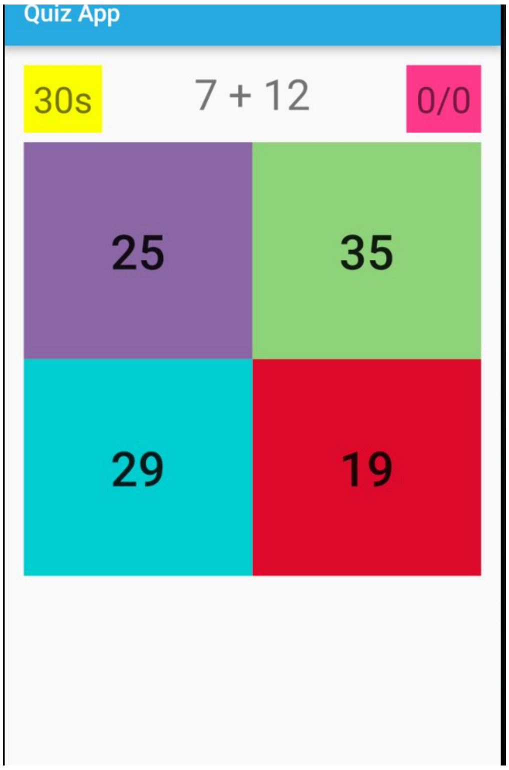


- Create a SplashScreen Activity and setContentView as the Splash Screen Layout (activity_screen_splash.xml) in the onCreate function.
- Create a new timer and set a time delay of 3 seconds which means 3000 milliseconds.
- Inside the finish callback of the timer create a function and set an intent which takes the flow of app from splash screen to main activity.

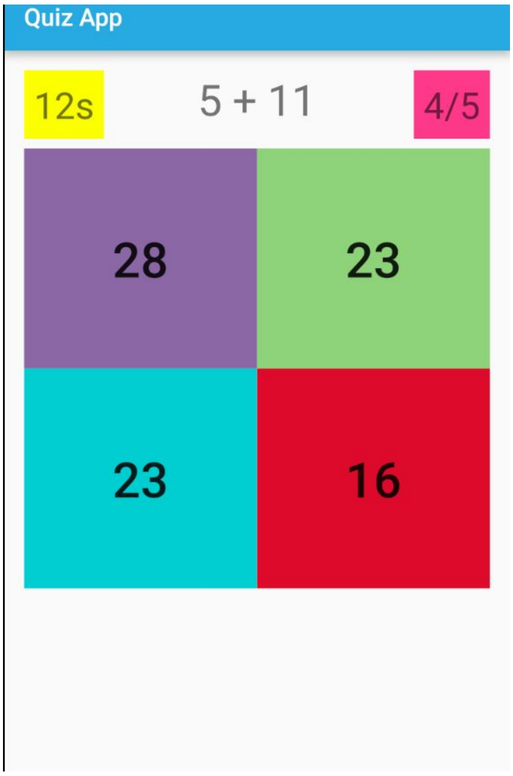
Basically the Splash screen stays for three seconds and then the function in the timer finish callback fires an intent which takes the flow to main activity.

### 2.) Main Activity

Main activity consists of a title named “Brain Trainer” and 4 buttons as the options to choose from.



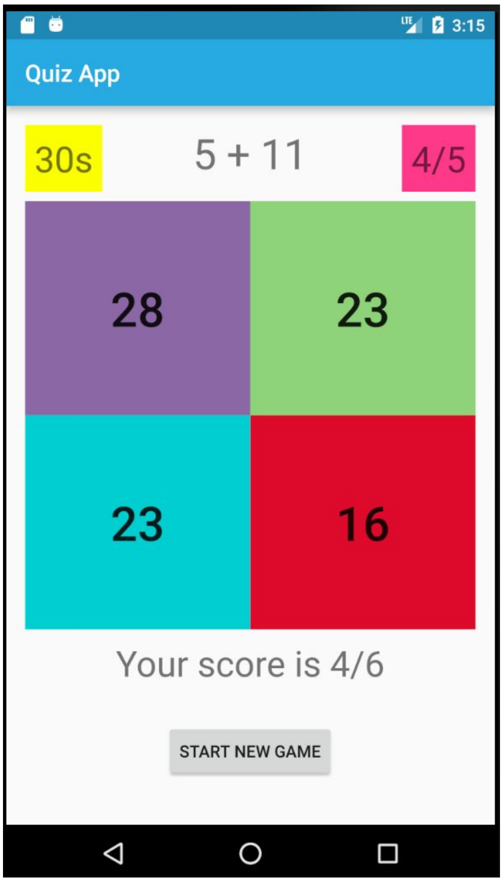
- setContentView to main activity layout (mainActivity.xml) in onCreate function.
- When any option (button) is clicked you can start the quiz, start the timer and update the score depending on whether the user chose the correct answer or not.
- Add a button at the bottom saying view scores which redirects the user to ScoreActivity.
- ScoreActivity should contain a list of previous scores of the user.



**NOTE:** Your app may not exactly look like the one in shown in the screenshots. Use these screenshots just for reference and try building your own.

- Create 3 Textviews - Score Textview, Question Textview and Timer Textview.
- Create 4 buttons as 4 options.
- Generate Random 3 numbers and place them as 3 wrong options and generate one correct number as the answer to the question and place it in the options.
- When user clicks on an option, Generate 2 random numbers to add and repeat the step above.
- Also, start a timer the moment user chooses the option for the first question.
- On the end of the timer, Display the final score and also display a button with option to restart.
- The restart button should set score to 0 and reset the timer also.

  Note: You can set the timer for as long as you want. Here it has been set for 30 seconds.



Your score is 4/6

**Main Activity after the timer is over.**

### 4.) ScoreActivity

It should display a listview of all the previous SCORES in the Database. Here SCORES is the score shown in the Score colum of the database.

**DATABASE STRUCTURE :**

Table Name - **Scores**

| **Correct Answers** **(int)** | **Incorrect Answers** **(int)** | **Score** **(varchar)** | **Percentage** **(decimal)** | **Total Questions Answered** **(int)** |
|---|---|---|---|---|

**HERE Percentage is the percentage of correct answers and is evaluated as (correct/total)*100.**