CSN – 254

SOFTWARE ENGINEERING

Quizziz

Final Report

April 17, 2022

Department of Computer Science and Engineering

Indian Institute of Technology

Roorkee

GROUP 15:-

Sai Bhargav	21114067	n_sbhargav@cs.iitr.ac.in
Yashwanth Krishna	21114093	s_yknaik@cs.iitr.ac.in
Suresh Kumar	21114066	n_skumar@cs.iitr.ac.in
Shashi Praneeth	21114089	s_spraneeth@cs.iitr.ac.in
Shyam Sundar	21114102	t_ssssundar@cs.iitr.ac.in
K P V Anil	21114048	k_pvanil@cs.iitr.ac.in

Contribution:

Sai Bhargav	-- Authentication (login, sign up)
Yashwanth Krishna	-- View Performance, Results
Suresh Kumar	-- Create quiz, Integration
Shashi Praneeth	-- Profile, UI Design
Shyam Sundar	--Attempt Quiz, Integration
K P V Anil	-- Timer

Github repository:

https://github.com/Bhargav4037/QuizApp

Problem Proposal:

- The primary obje0ctive of our "Quizziz" is to provide users with a platform to test their knowledge on a particular subject matter using different formats of questions to make users think differently and to reflect the results/grades of the users in a systematic manner. The quiz application can be used for educational purposes, recruitment of clubs, etc...

- Using our software would greatly minimize the time taken in the correction because rather than individually correcting each student's sheet, the author could save the answer keys of each question so that as soon as the quiz is submitted by the user, they get their responses and grades.

- Apart from the time reduction, it would also save a lot of the authors energy, making it easier for him/her.]

Technology Used :

- Flutter framework .
- DART language is used .
- Cloud Firestore for database.
- Firebase Authentication for login.

Application :

- A major problem that occurs to the paper setter/Exam author in today's world is conducting the exam for a large number of students and correcting all the answer sheets and keeping track of the results of the huge number of students involved.
- Instead of wasting their Precious time and energy correcting sheets of a large number of students, which could be used productively. This greatly reduces the efficiency of the organization. For example, if there are a lot number of students attempting the quiz/Exam, then the Examiner will take a lot of time to correct and reflect each student's report. It's really a difficult task for examiners to do all these. This time could be much more fruitfully utilized.
- The overall efficiency of any organization would be greatly improved if there was a faster method to take care of this Examination process. This is where our software, Quizz steps in. Quizz is a modern platform to conduct quizzes and allow users(students) to participate in online model.

Innovation:

- User-Friendly interface for better usage of resources. Quiz provides a database to maintain and review the progress of students, both student and teacher can access the results. 'Quizziz' only allow people of the same organization/ A particular class of people, who are allowed by the teacher.
- Another feature in 'Quizziz' is, users can check the progress of their previous quizzes .
- Security: It allows only the users with a valid quiz code which was provided by the teacher who created the quiz.

Description Of Implementation

Development Process

Agile SDLC model is a combination of iterative and incremental process models with a focus on process adaptability and customer satisfaction by rapid delivery of working software products. Our project needs a lot of iterative processes to be followed, Agile software methodology would be best suited. Since our project needs a lot of functionalities to be implemented and regular interaction with customers is important. So, at once we can't decide the entire scope of our project and the complete customer requirements can't be known beforehand. So, we decided to once release the project with only some of the functionality implemented and then taking customer feedback into input and allowing further changes to our features and then a better and upgraded version of our project is released.

Outline

Phase 1 | 5th March - 22nd March:

- Gather the resources and explore the tech stack required for the project.
- Feasibility Study report of the application will be prepared and shared.
- Prepare the Requirement Analysis document that will describe all the requirements of the application.

Phase 2 | 23rd March - 10thApril:

- Designing each pages and connecting them accordingly to each other.
- Setting up the data base, backend architecture of the project.
- Set up communication links between the frontend and backend part of the application.

Phase 3 | 10th April - 20th April:

- Testing and cleanup of the code.
- Deployment
- Brainstorm about future prospects of the application

Implementation Detail

On opening the app first page visible to you will be asking to enter the app as either teacher/student .

1.Authencation

Authentication is similar for student and teacher.

On selecting student/ teacher option, it will be directed to login page. If you already registered as a user you can login with your corresponding credentials (mail and password). Else you can select sign up by entering your email and setting a password.

We used Firebase authentication for this task, using the command

'*FirebaseAuth.instance.signInWithEmailAndPassword*'

Every time a user signs up , user needs to enter some details regarding them (Name , Enrollment Number, Email Id , Class). This data is stored to authentication, Cloud Firetore database in firebase , creating a new user in a collection named 'Teachers' for teacher and 'User' for student . And when a user tries to login , we check for validness of the entered email and password in the authentication database and intimates the user . If all are perfect home page of the respected student /teacher.

2. Database

There are two databases available in Firebase.

- Realtime database
- Cloud Firestore

We have used cloud Firestore for our app, because:

- **Realtime database** stores data as one large JSON tree which makes it very easy to store simple data but Complex or hierarchical data is harder to organise at scale. Where as, **Cloud Firestore** on

the other hand stores data in the form of documents which are organised in collections. In this, storing of simple data is similar to JSON and it is stored in documents.

- In **Realtime database**, there are basic write and transaction operations. Write data works as an individual operation and Transactions require a completion callback in the native SDKs. Where as **Cloud Firestore** allows atomic write and transaction operations. It batches operations and completes them automatically. Also, transactions automatically repeat themselves until they're completed.

- In our Code we initialized our database by the command,
  'FirebaseFirestore.instance'
  By this command we create an instance of the database.

3. Create quiz by Teacher

On successful login of teacher, they are directed to homepage where there will be an option to create a quiz for the students and a button on top right corner to view their profile.
On clicking 'Create Quiz' button, you need to enter basic information about quiz - Quiz Title, Quiz Code, Quiz Time and Class.
After entering quiz the information a page will be appear which consists of text fields for question, options and correct option (only in caps eg:-A),
There are two buttons in every new question page, one is for next question and the next is for complete creating.
On pressing complete create quiz information and the questions are stored to Database in a collection named 'quizavailable'.

4. View & Edit Profile

There will be a button at top right corner in the Homepage of both teacher and student for viewing profile of the user. On clicking it information provided by the user at the time of sign up is displayed in order. This info is brought from the database of the respected user using stream and snapshots.

Stream – it represents the flow of data in database

Snapshot – it represents the data in a particular document .

And there is also an option for editing this info. The user can update the info in case of any requirement, the data in database is also updated.

5. Attempt Quiz

The Homepage of Student has a button 'Attempt Quiz' which on clicking shows all the available quizzes in a new page. Student can only take a particular quiz by clicking on it and entering the corresponding Quiz Code. The quiz code is provided by teacher at the time of creation of quiz.

After entering the correct code, a basic preview of quiz appear which consists of blured images of questions and options and a button elevated in the center with title "***Attempt***"
On clicking the button questions will appear in order and the selected options will be saved
Automatically when we move to next question using the button present at the bottom '***Next***'.

Finally submit button will appear at the very last question to end the quiz.

6.Timer

While Creating Quiz Teacher has a option to set Time for Quiz (Maximum Time student can attempt Quiz) We can set max time as much as we want there is no limit to it in minutes format.

When a student starts attempting quiz timer will automatically appear on the appbar indicating the time left to complete the quiz. If the time is over, the Quiz automatically gets submitted and a page containing the Score and button to view the Results will appear. Last ten seconds of quiz time would be red color so that the student who has attempting quiz get know the quiz is about end in mere seconds of 10. Here timer Is for whole quiz not for each question.

6. Result

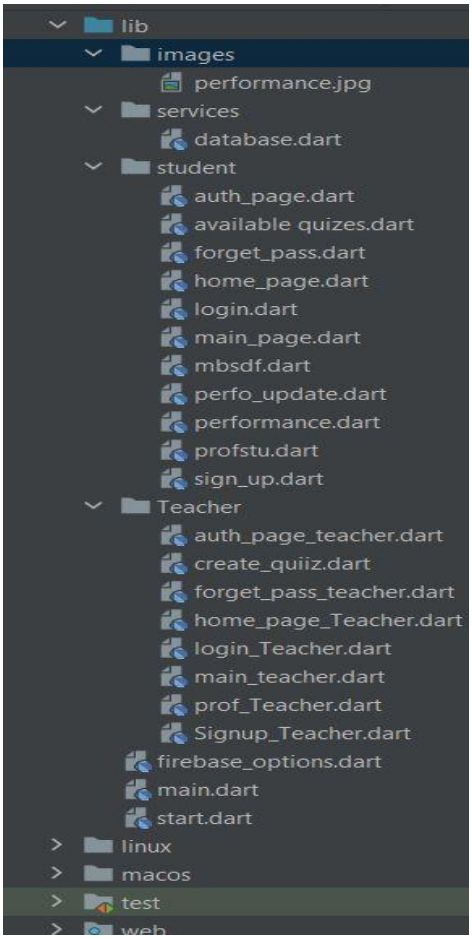
After completion of the time (or) after submitting the quiz, you will get the result of your quiz by clicking the button “Result”. It consists of every question indicating the attempted and the correct answers using colors – “yellow, Red and green”.

After reviewing all the answers, you will be navigated to the main page off the respected student.

7. View Performance

There will be button in the main page of the student to view their past quiz performances. On clicking this you will be navigated to a page which consists of all the past quizzes you have attempted. In each card you view the quiz title, score, no of un attempted questions, no of correct answers, and no of wrong answers .

8. File Hierarchy



References

1. **https://docs.flutter.dev/**
2. **https://firebase.google.com/docs/database/flutter/read-and-write**
3. **https://www.youtube.com/watch?v=n1PM9XcYD5s - CRUD operations**
4. **https://www.youtube.com/watch?v=1ukSR1GRtMU&list=PL4cUxeGkcC9jLYyp2Aoh6hcWuxFDX6PBJ**