UNIVERSITI MALAYSIA TERENGGANU

BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING)
WITH HONOURS

SEM I 2023 / 2024

CSM 3114: MOBILE DEVELOPMENT FRAMEWORK (K1)

Assignment 2: School Management Application

PREPARED BY:

MUHAMMAD NAJMUDDIN BIN AZLI (S62728)

LECTURER:

DR MOHAMAD NOR HASSAN

Table of Contents

1. Executive summary

LearnWeave Application emerges as a revolutionary school management tool, expertly designed to streamline and optimize the complex dynamics of educational institutions. At its core, LearnWeave is an intuitive, user-friendly platform that simplifies administrative tasks and enhances communication between teachers, students, and parents. Key features include secure login portals, seamless addition and management of teacher and student profiles, and an efficient system for recording and monitoring academic performance and disciplinary cases. The app's design focuses on creating a cohesive, interconnected educational ecosystem, akin to a loom weaving together the various threads of school management into a harmonious fabric.

The heart of LearnWeave lies in its innovative approach to handling student data and academic tracking. The app provides a robust framework for teachers to input and access student results, offering a transparent and accessible view of academic progress. This feature not only aids in the tracking of individual student performance but also enables teachers and parents to collaboratively identify and address academic challenges. Additionally, LearnWeave’s discipline module offers a discreet and effective way to record and manage behavioral incidents, ensuring a holistic approach to student development and wellbeing.

Beyond functionality, LearnWeave is designed with the future of education in mind. It recognizes the evolving landscape of school administration and the increasing need for digital solutions that are both comprehensive and adaptable. With its user-centric design, secure data handling, and versatile features, LearnWeave stands out as not just an application but a vital partner in the journey of educational excellence. It empowers schools to not only manage their day-to-day operations with greater ease and efficiency but also to weave together the various strands of academic life into a cohesive and thriving educational tapestry.

2. Use Case

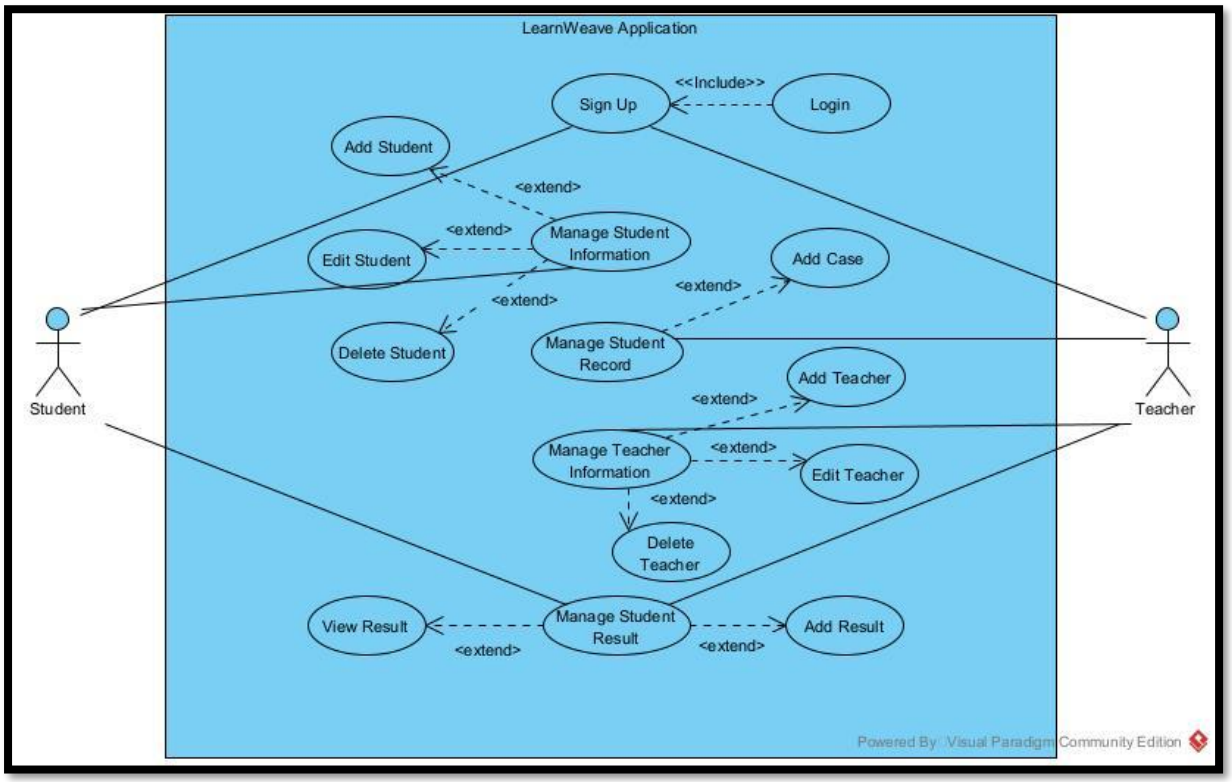


Figure 2.1: LearnWeave Use Case

3. Common structure of tree widget

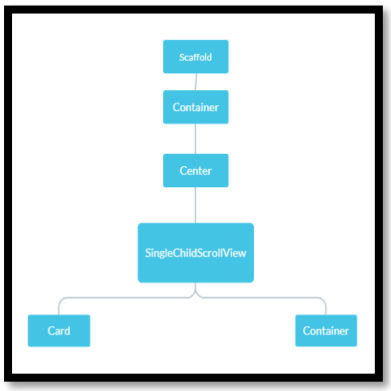


Figure 3.1: main.dart

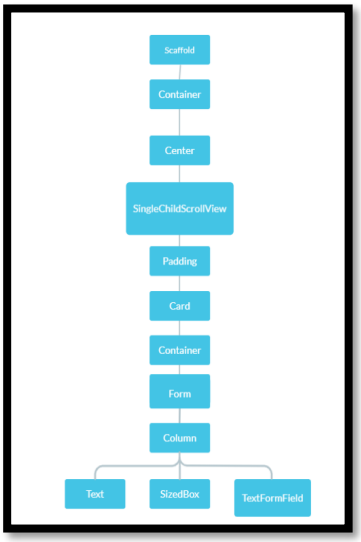


Figure 3.2 sign up.dart

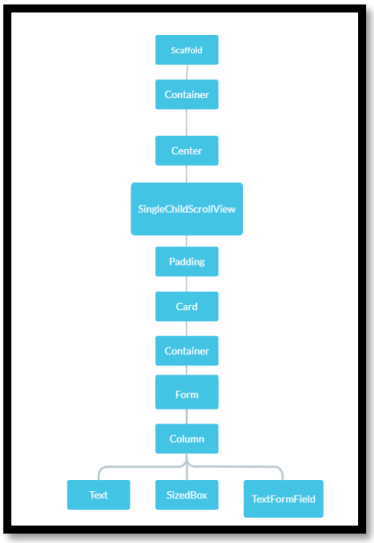


Figure 3.3: login.dart

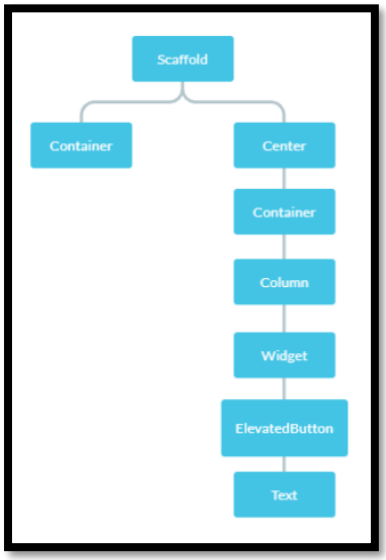


Figure 3.4: homepage.dart

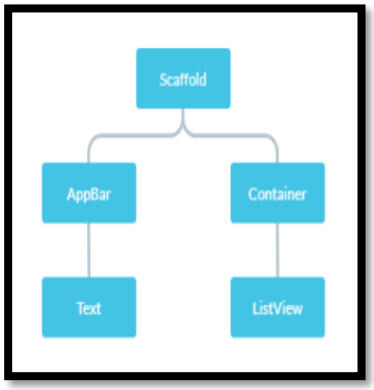


Figure 3.5: second_page.dart

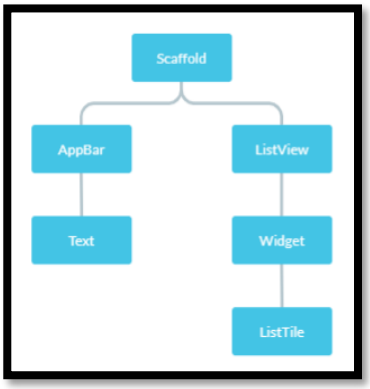


Figure 3.6: third_page.dart

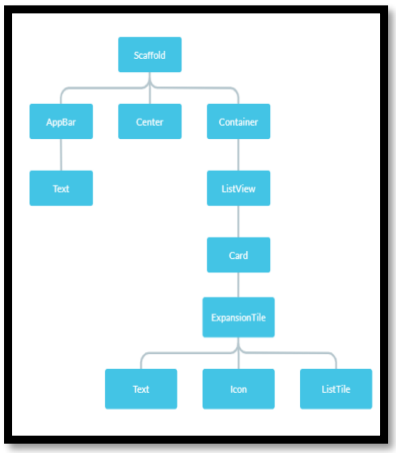


Figure 3.7: teacher_data.dart

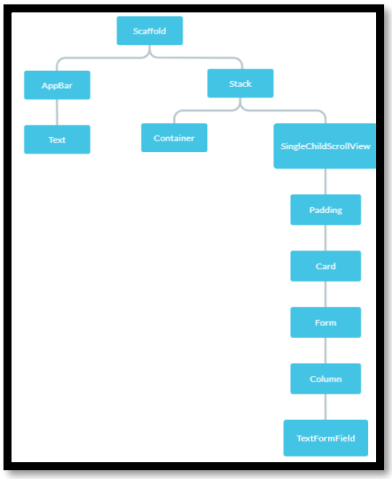


Figure 3.8 add_teacher.dart

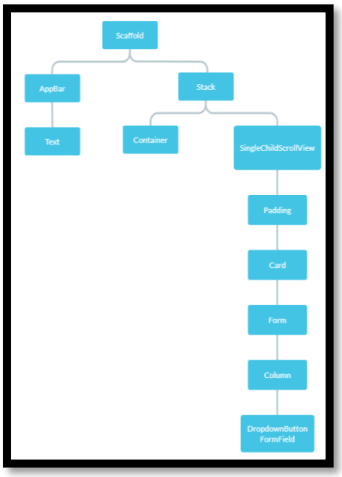


Figure 3.9 edit_teacher.dart

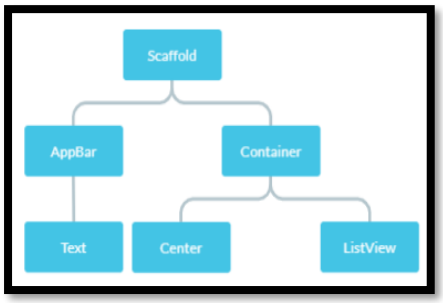


Figure 3.10: student_result.dart

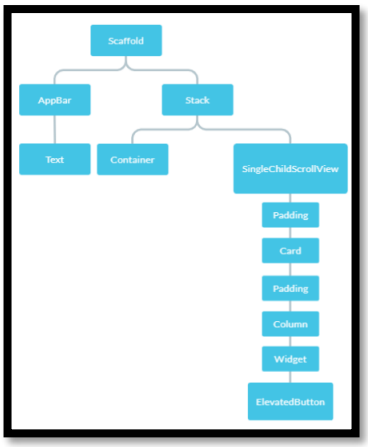


Figure 3.11: marks.dart

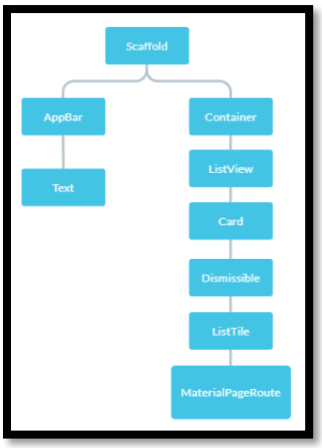


Figure 3.12: record_discipline.dart

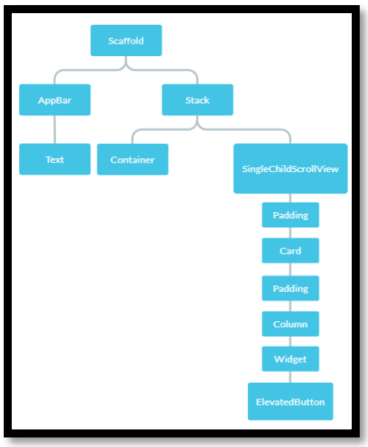


Figure 3.13: add_case.dart

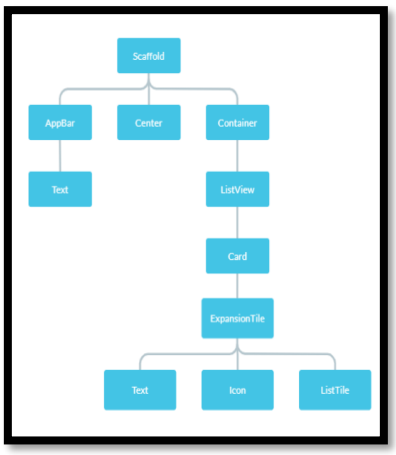


Figure 3.14: student_data.dart

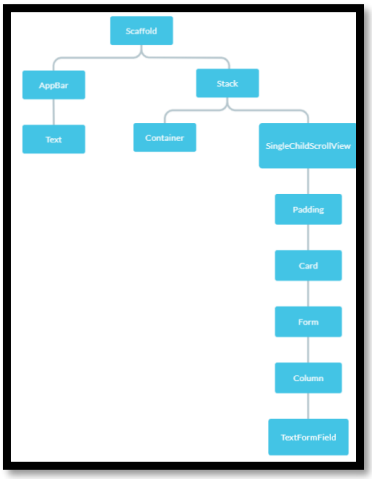


Figure 3.15: add_student.dart

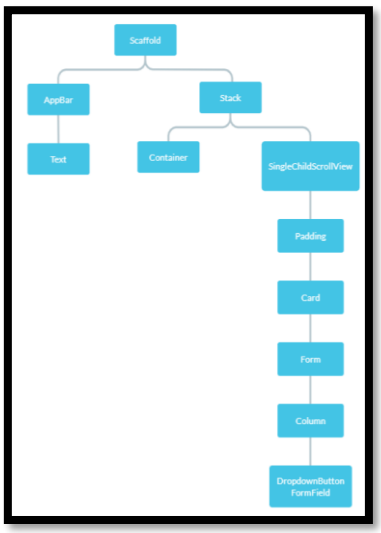


Figure 3.16: edit_student.dart

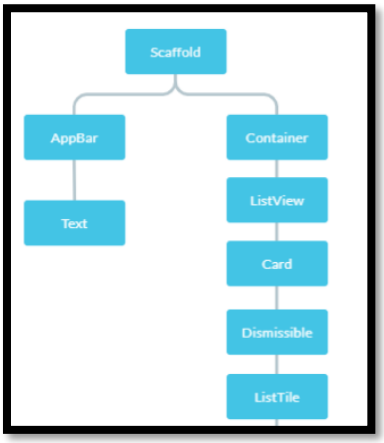


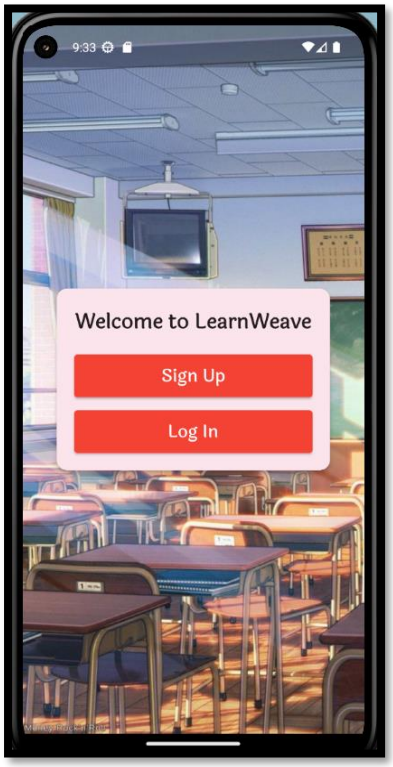
Figure 3.17: view_result.dart

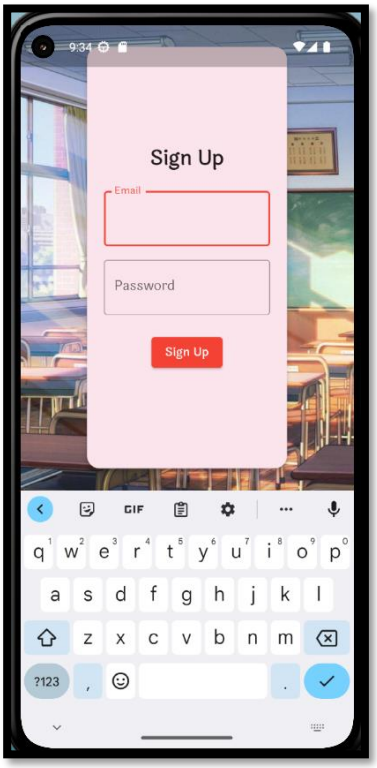
4. Flutter widget and features in the application

1. **MaterialApp**: The MaterialApp is the root widget that configures the top-level MaterialApp-specific features, such as the theme and routes.
2. **Scaffold**: The Scaffold widget represents the basic material design structure and provides a framework to implement the basic visual structure of the application.
3. **AppBar**: The AppBar widget represents the app bar at the top of the screen, typically containing the application title and action icons.
4. **ListView.builder**: The ListView.builder widget is used to create a scrollable list of widgets. It efficiently builds items on demand.
5. **Card**: The Card widget is used to create a material design card, grouping related information in a visually appealing way.
6. **ListTile**: The ListTile widget is used in conjunction with ListView to represent a single fixed-height row. It is often used inside a Card or ExpansionTile.
7. **ExpansionTile**: The ExpansionTile widget creates a Material Design expansion tile that expands to reveal additional content when tapped.
8. **Icon**: The Icon widget displays an icon representing common actions or subjects. Different icons are used for visual representation.
9. **CircularProgressIndicator**: The CircularProgressIndicator widget is used to indicate that the application is busy with a task. It is commonly displayed when data is being fetched.
10. **SnackBar**: The SnackBar widget is used to display a temporary message at the bottom of the screen. It is often used for showing feedback or notifications.
11. **FutureBuilder**: The FutureBuilder widget is used to create a widget that builds itself based on the latest snapshot of interaction with a Future.
12. **DropdownButtonFormField**: The DropdownButtonFormField widget is a button that, when pressed, displays a dropdown menu of items. It's used to select a value from a predefined list.
13. **ElevatedButton**: The ElevatedButton widget is a material design raised button. It triggers an action when pressed.
14. **TextFormField**: The TextFormField widget is a text input field that allows the user to enter data. It's often used within a form.
15. **Form**: The Form widget is used to create a form that contains multiple fields. It helps in validating and saving user input.
16. **LinearGradient**: The LinearGradient widget is used to create a linear gradient, providing a range of colors that transition in a linear manner.
17. **Dismissible**: The Dismissible widget allows the user to dismiss items in a list. It's often used for actions like deleting an item.

18. **PageRouteBuilder**: The PageRouteBuilder widget is used to define custom page transitions when navigating between screens.
19. **Container**: The Container widget is a box model that can contain other widgets. It's often used for styling and positioning.
20. **Stack**: The Stack widget is used to overlay widgets on top of each other. It allows for complex layouts.
21. **GradientAppBar**: Custom widget or implementation that creates an AppBar with a gradient background.
22. **Text**: The Text widget displays a string of text. It's used to show information and labels.
23. **FloatingActionButton**: The FloatingActionButton is a circular button that typically triggers the primary action in an application.
24. **DropDownMenu**: Although the exact widget is not provided, a dropdown menu is created using the DropdownButtonFormField widget.
25. **PageRoute**: Used in navigation to define the route for transitioning between screens.
26. **Gradient**: LinearGradient is a gradient used for background styling, providing a smooth transition between colors.

5. UI with explaination





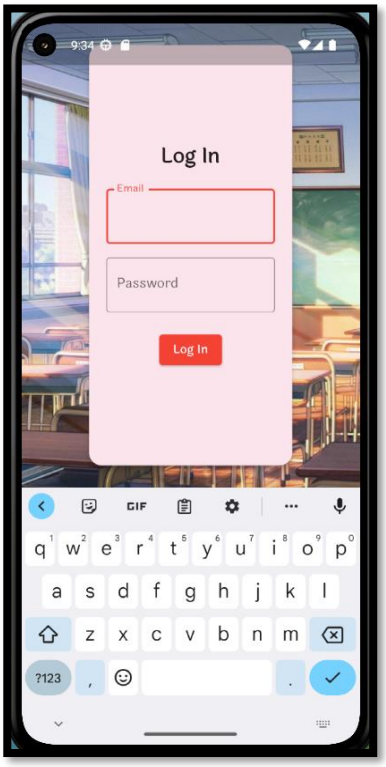
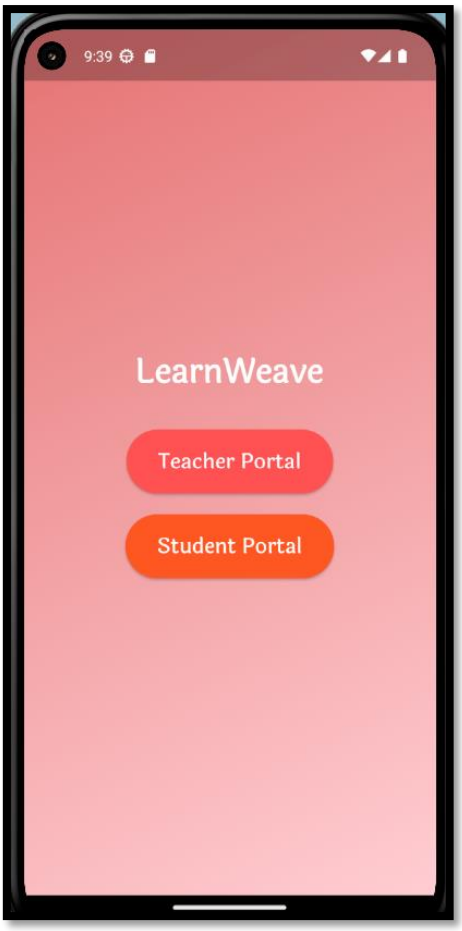


Figure 5.1: main.dart *Figure 5.2: sign_up.dart* *Figure 5.3: login.dart*

The first three pages, which include the first page, sign-up page, and login page, are shown in Figure 5.1, 5.2, and 5.3. The user is led by these pages to insert an email and password to enter the homepage.



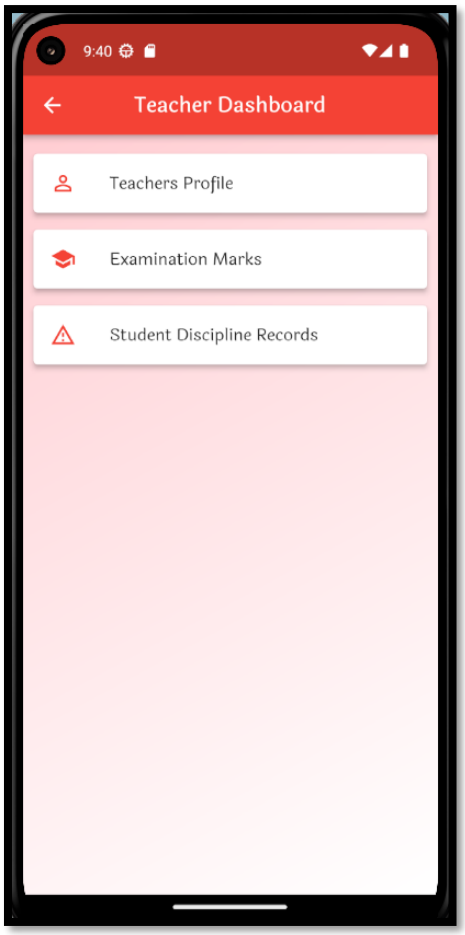
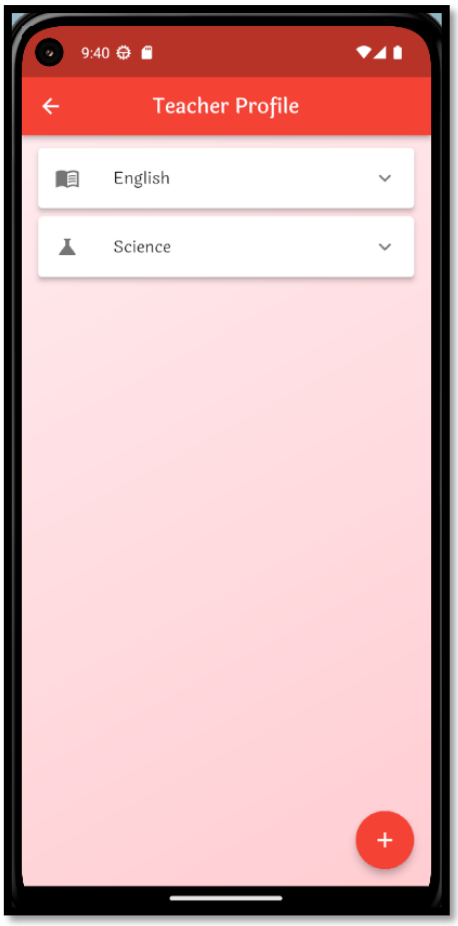


Figure 5.4: homepage.dart

Figure 5.5: second_page.dart

The homepage and the second page, which demonstrate the functionalities of the application, are shown in the figure above. Different functions of each page are provided by the teacher portal and student portal.



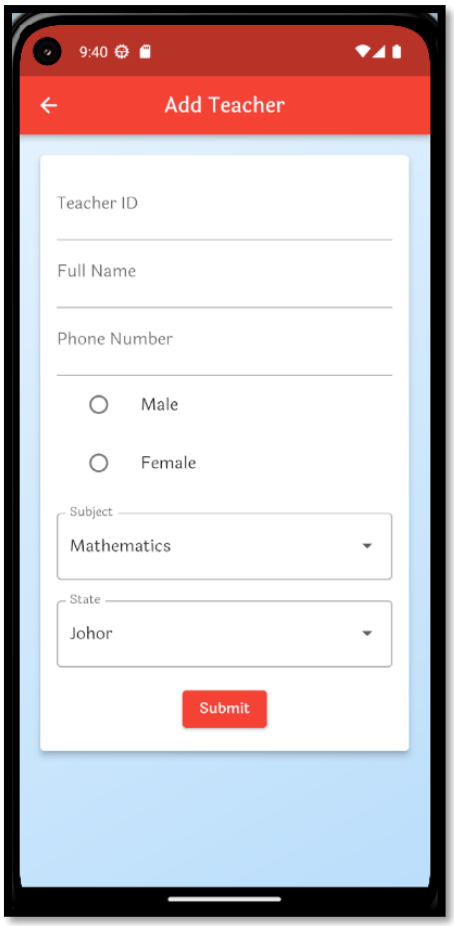


Figure 5.6: teacher_data.dart

Figure 5.7: add_teacher.dart

The form on these pages allows users to input personal teacher data, which is then inserted into the database. After the data is submitted by the user, the details can be viewed in the expansion tile widget.

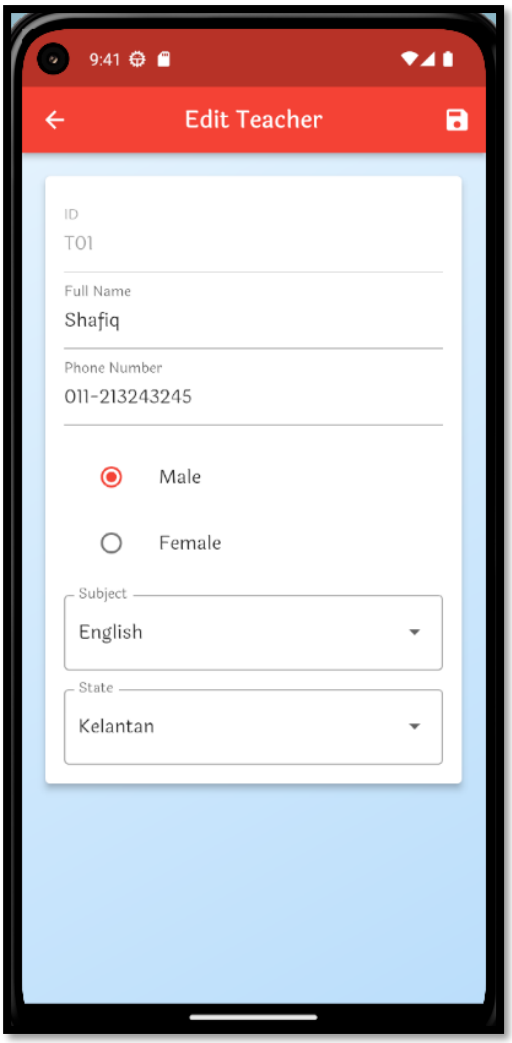


Figure 5.8: edit_teacher.dart

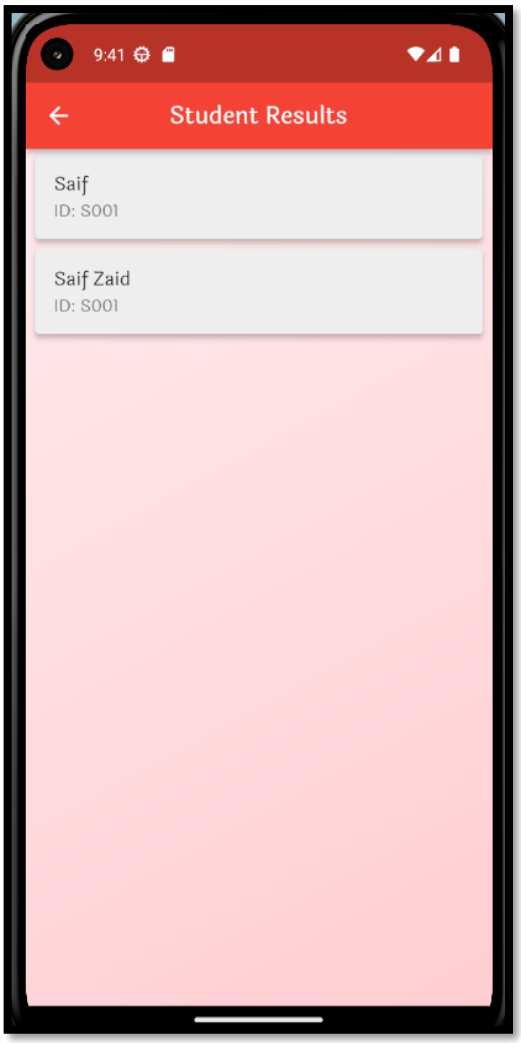


Figure 5.9: student_result.dart

On these pages, users are allowed to edit the data and update the list accordingly. Marks can also be given to the student by the teacher when the student details are clicked.

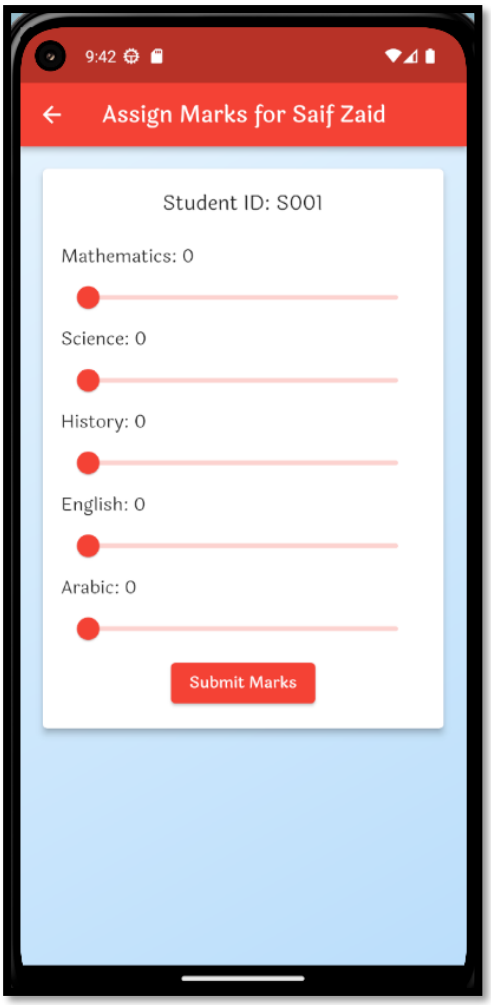


Figure 5.10: marks.dart

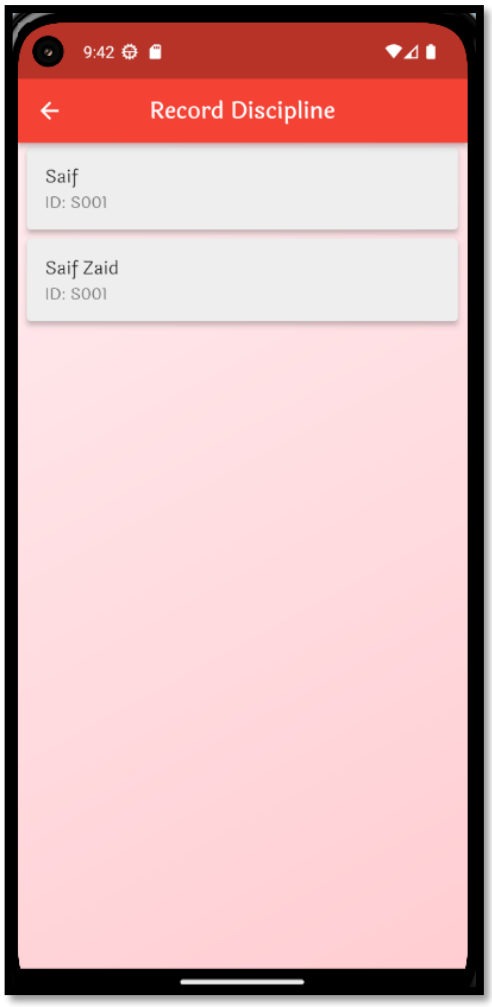


Figure 5.11: record_discipline.dart

These pages allow teachers to give marks just by slides the marks of each subject and submit the marks. Teachers also can add record discipline for students who doesn't follow the rules by selecting the student data display.

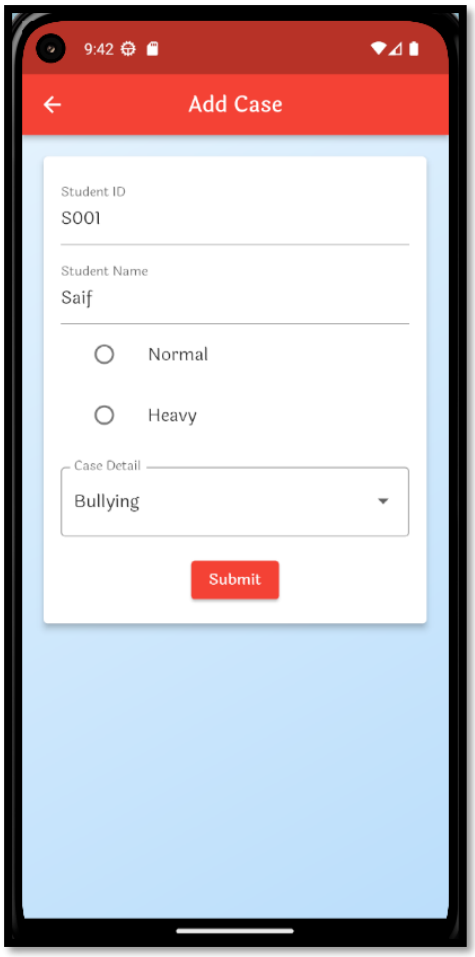


Figure 5.12: add_case.dart

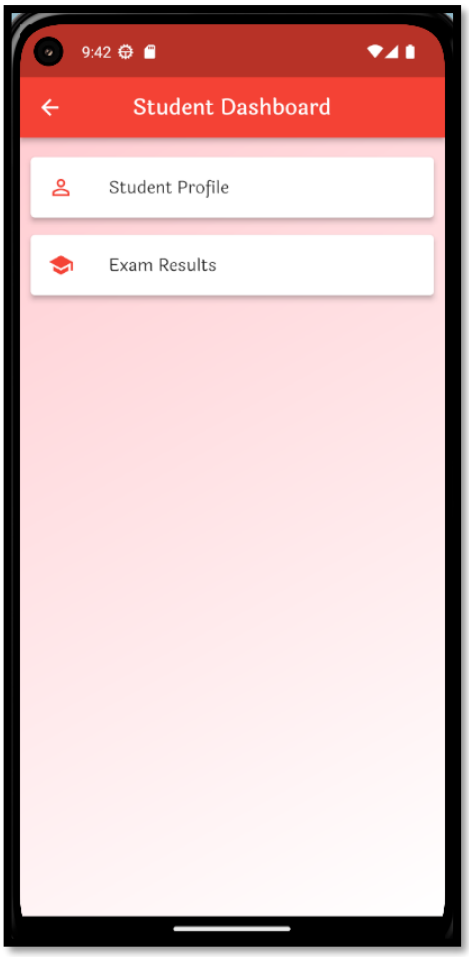


Figure 5.13: third_page.dart

The add case page are the form that teacher needs to fill before submitting the record to the database. Meanwhile, for student function in this application are adding their information and view their results.

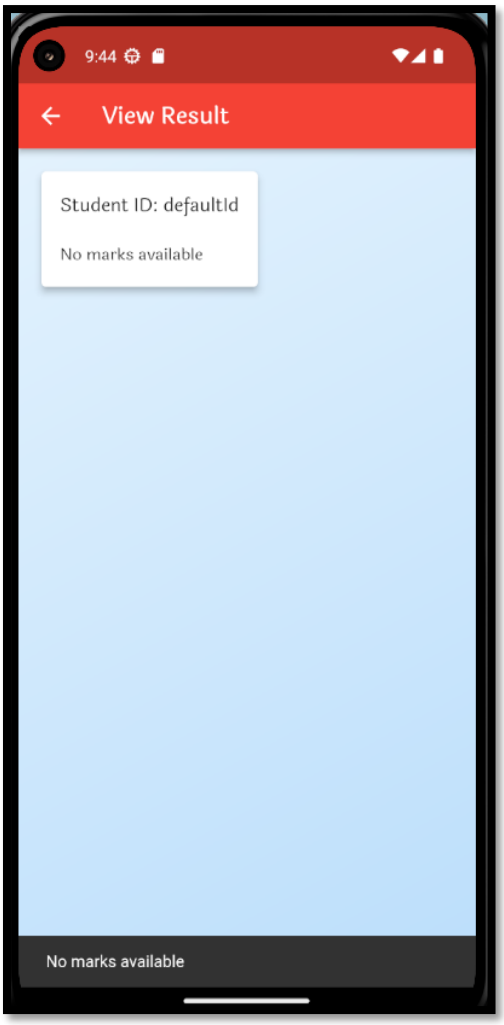


Figure 5.14: student_data.dart

Figure 5.15: view_result.dart

These pages allow students to do the same function as teachers which are add student data, edit and update to the database. Last function shows here are student can view their results that have been marks by the teachers.

6. Conclusion

In conclusion, the LearnWeave Application emerges as a revolutionary force in the realm of school management, seamlessly integrating innovative features to simplify administrative tasks, enhance communication, and foster a cohesive educational environment. By embracing a user-friendly design, secure data handling, and a versatile array of functionalities, LearnWeave not only addresses current challenges but also positions itself as an adaptive solution for the evolving landscape of education. Beyond its technical prowess, the application symbolizes a commitment to empowering schools in their pursuit of academic excellence. It weaves together the diverse aspects of school management, offering a holistic platform that not only efficiently manages day-to-day operations but also contributes to the flourishing educational journey of students, teachers, and parents alike. LearnWeave stands as a testament to the transformative power of technology in shaping the future of education.

7. References

1. https://api.flutter.dev/flutter/material/ExpansionTile-class.html

2. https://api.flutter.dev/flutter/widgets/Dismissible-class.html

Link GitHub:

https://github.com/najmuddin02/Assignment-2_S62728