



UNIVERSITI MALAYSIA TERENGGANU  
FACULTY OF COMPUTER SCIENCE AND MATHEMATICS (FSKM)  
2024

CSM3114  
FRAMEWORK BASED MOBILE APPLICATION DEVELOPMENT

Prepared By:  
MUHAMMAD AFIQ BIN MOHAMAD  
(S63755)

Prepared For:  
DR. MOHAMAD NOR BIN HASSAN

**Project 2 – SchoolHub Final Report**  
**Bachelor of Computer Science (Mobile Computing) with Honours**  
**SEMESTER I 2023/2024**

<b>EXECUTIVE SUMMARY OF THE PROTOTYPE</b>	<b>3</b>
<b>Introduction</b>	<b>3</b>
<b>Key Features and Functionalities</b>	<b>3</b>
<b>Key Benefits</b>	<b>3</b>
<b>USE CASE</b>	<b>4</b>
<b>THE COMMON STRUCTURE OF TREE WIDGETS USED WHEN DESIGNING AND DEVELOPING THE APPLICATION</b>	<b>5</b>
<b>FLUTTER WIDGET AND FEATURES ADOPTED IN THE APPLICATION</b>	<b>7</b>
1. UI Widgets	7
2. State Management	7
3. Navigation and Routing	7
4. User Input and Forms	7
5. Third-Party Libraries and Plugins	8
6. Performance Optimization	8
7. Backend Stack	8
<b>SAMPLE OF INTERFACE WITH EXPLANATION</b>	<b>9</b>
<b>CONCLUSION</b>	<b>12</b>
<b>PROJECT GITHUB REPOSITORY</b>	<b>12</b>
<b>REFERENCES</b>	<b>13</b>
<b>Figure 1 Use Case Diagram</b>	<b>4</b>
<b>Figure 2 Login Screen Widget Tree</b>	<b>5</b>
<b>Figure 3 Dashboard Screen Widget Tree</b>	<b>5</b>
<b>Figure 4 StatsWidget Widget Tree</b>	<b>6</b>
<b>Figure 5 OptionMenu Widget Tree</b>	<b>6</b>
<b>Figure 6 Login and Dashboard Screen</b>	<b>9</b>
<b>Figure 7 Manage Student and Teacher Screen</b>	<b>9</b>
<b>Figure 8 Result Screen</b>	<b>10</b>
<b>Figure 9 Discipline Screen</b>	<b>11</b>

## **Executive Summary of the Prototype**

### **Introduction**

As part of its business strategy to develop mobile solutions that optimize remote access and connectivity, ABC Technologies is putting out a proposal for a school management application that would enable schools to digitalize many of their key administrative processes and give stakeholders real-time access to critical school information.

### **Key Features and Functionalities**

- The system allows for the input and updating of thorough student profiles.
- The application's dashboard gathers important performance metrics related to the school's operations in a passive manner, giving administrators a summary with illuminating graphical displays.
- The application passively aggregates records of disciplinary actions taken by the institution, including merit point deductions.

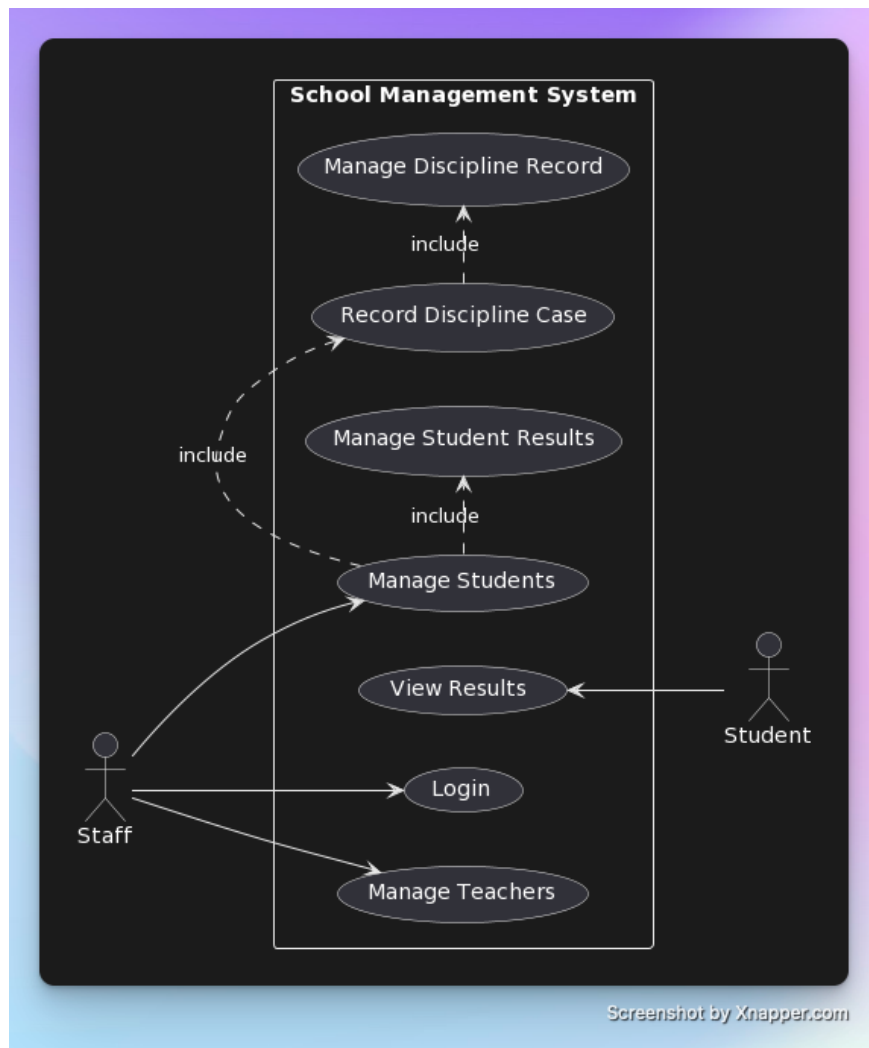
### **Key Benefits**

The proposed administrator-focused school management app aims to provide the following benefits:

- Digitization of key processes like admissions, assessments, discipline cases
- Instant access to integrated data on teacher profiles, student academics and discipline
- Better long-term monitoring through storage of academic and discipline trends
- Early flagging of achievement gaps based on data analytics
- Shaping of progress policies driven by insights from app data analytics

By granting administrators instant access to both overviews and granular detail on school operations, the app would enable better informed and data-driven decision making.

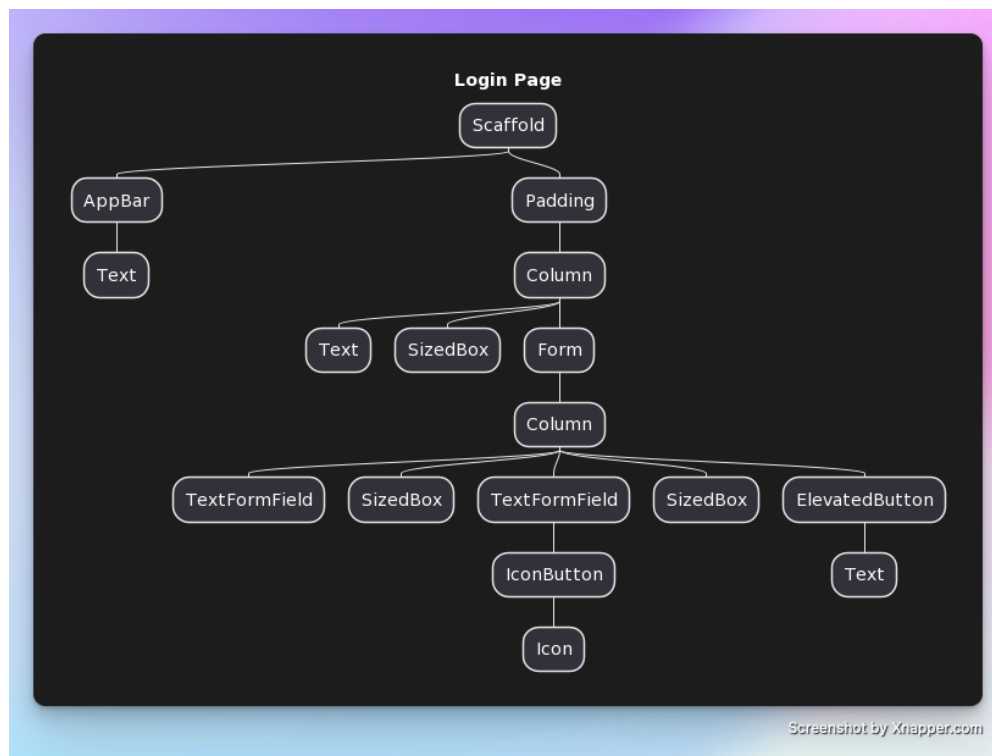
## Use Case



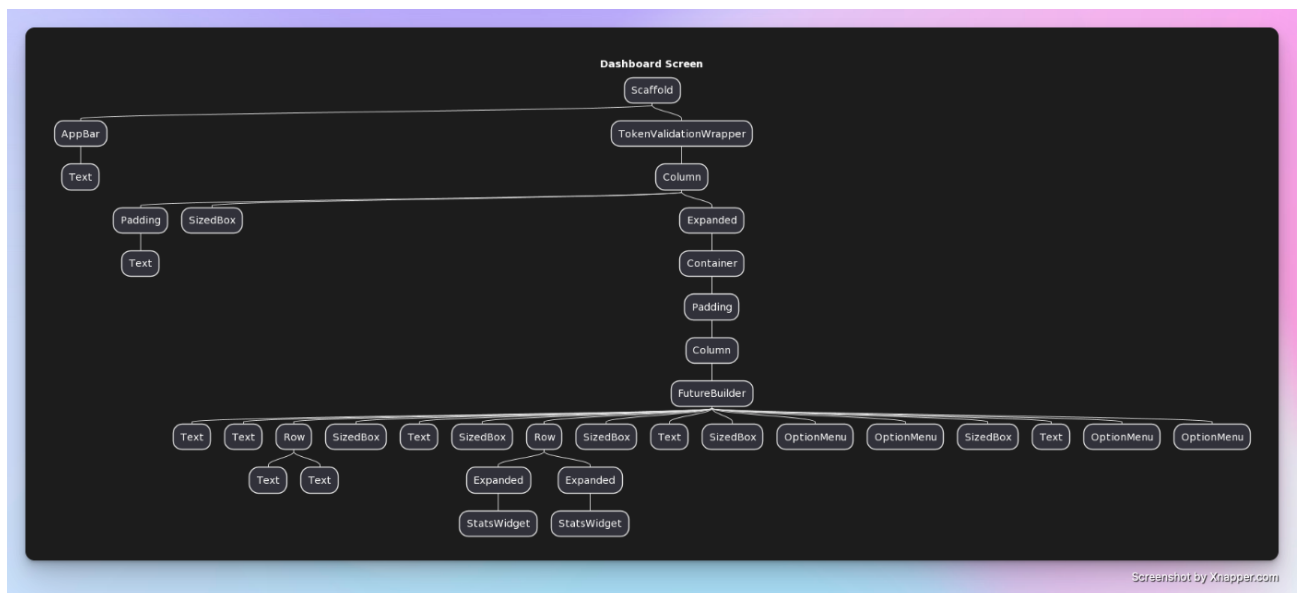
**Figure 1** Use Case Diagram

Staff and students are shown as the two user types of the School Management System in the provided use case diagram. To access management features, such as "Manage Teachers," "Manage Students," and "Manage Student Results," which regulate the creation and upkeep of teacher and student profiles and academic records, staff members must log in. Staff members can supervise disciplinary actions through the "Manage Discipline Record" use case, which goes into further detail in the "Record Discipline Case" use case. Students can review their academic performance after logging in by utilizing the "View Results" use case. The user interface and functionality of the system are briefly summarized in this diagram.

## The Common Structure of Tree Widgets Used When Designing and Developing the Application



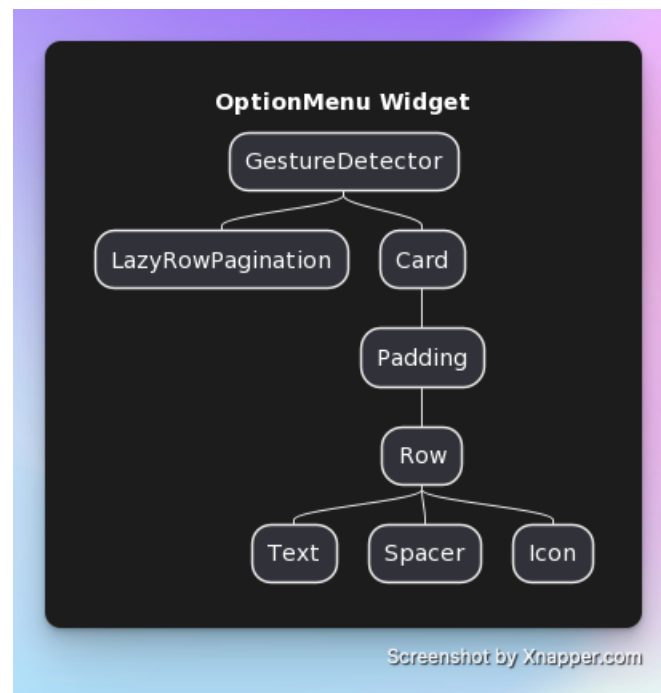
*Figure 2 Login Screen Widget Tree*



*Figure 3 Dashboard Screen Widget Tree*



**Figure 4** StatsWidget Widget Tree



**Figure 5** OptionMenu Widget Tree

The common widget tree used across the application features a "Scaffold" at the root, which includes an "AppBar" and a "Padding" widget on the body properties of "Scaffold". A "Text" widget in the "AppBar" creates a header layout that is uniform across screens. The "Padding" widget makes sure that every page's body has the same amount of space, which supports the application's overall logical visual structure.

## Flutter Widget and Features Adopted in the Application

### 1. UI Widgets:

- **Scaffold:** Forms the main framework of the application, providing a unified layout for the material design.
- **AppBar:** Acts as the application's top bar, providing actions, a title, and navigation controls. It facilitates easier user navigation and offers instant access to important features.
- **ListView:** Widely used to display discipline cases and results. Supports both vertical and horizontal orientations to accommodate various content types.
- **Card and ListTile:** These widgets are used to present data in an aesthetically pleasing and well-organized way. ListTiles are used within lists to display data in an organized and understandable manner, while Cards contain elements such as discipline records and student details.

### 2. State Management:

- **Provider:** For state management, Provider was chosen which makes use of MultiProvider to ensure effective UI updates and tidy dependency management. It preserves performance and streamlines state sharing throughout the application.
- **StatefulWidget:** Utilized for local state management in specific widgets where state changes are confined. For ease of use and efficiency, the `setState()` method is used in situations such as user interactions.

### 3. Navigation and Routing:

- **Navigator:** An essential part of controlling the app's navigation and screen transitions. It makes it possible for users to switch between various functionalities with ease and intuition.
- **Routes:** Carefully defined routes enable structured navigation and access to various app features, such as viewing student results and managing discipline cases.

### 4. User Input and Forms:

- **TextFormField and ElevatedButton:** Important components for creating forms that enable active user input and interaction. ElevatedButton offer a unique and convenient method of submitting forms.
- **Form Validation:** Implemented rigorously to maintain data integrity, ensuring that all input adheres to our validation rules before submission.

## 5. Third-Party Libraries and Plugins:

- **http:** Essential for handling network requests and API calls, enabling seamless communication with our backend services.
- **shared\_preferences:** Utilized for local storage of user preferences and essential data, enhancing the user experience by retaining key settings and information.

## 6. Performance Optimization:

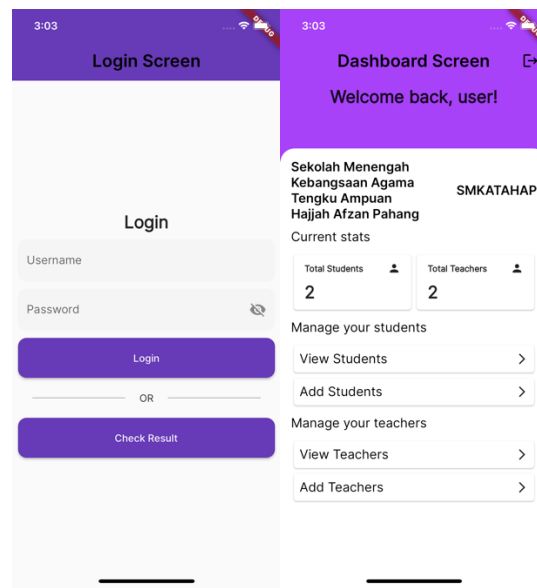
- **Lazy loading:** Used in table row rendering to maximize resource consumption and app performance. This greatly improves user experience, especially in data-intensive app sections.

## 7. Backend Stack:

- **BunJS:** Adopted Bun's JavaScript, a new and fast runtime and bundler, to support our TypeScript-based backend. It offers impressive performance gains and reduced overhead.
- **ElysiaJS:** The integration of Elysia's out-of-the-box authentication middleware strengthens our application's security posture. TypeScript support integrated into Elysia streamlines development and minimizes bugs, ensuring a dependable and secure backend for our application, and Elysia JS framework accelerates backend development by offering a strong structure for building scalable and efficient services.

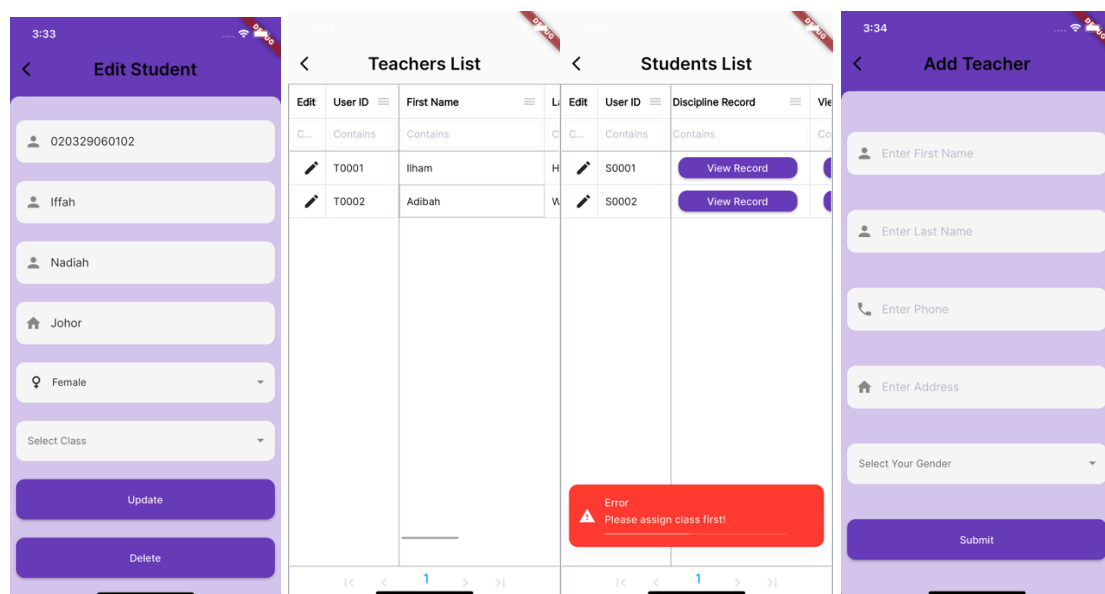


## Sample of Interface with Explanation



*Figure 6 Login and Dashboard Screen*

**Figure 6** depicted login screen, encountered post-onboarding, offers users the choice to sign in as Staff. This option opens a dashboard where important functions for staff members can be accessed and a series of statistics, personalized greetings, and the name of the school are presented in an organized manner.



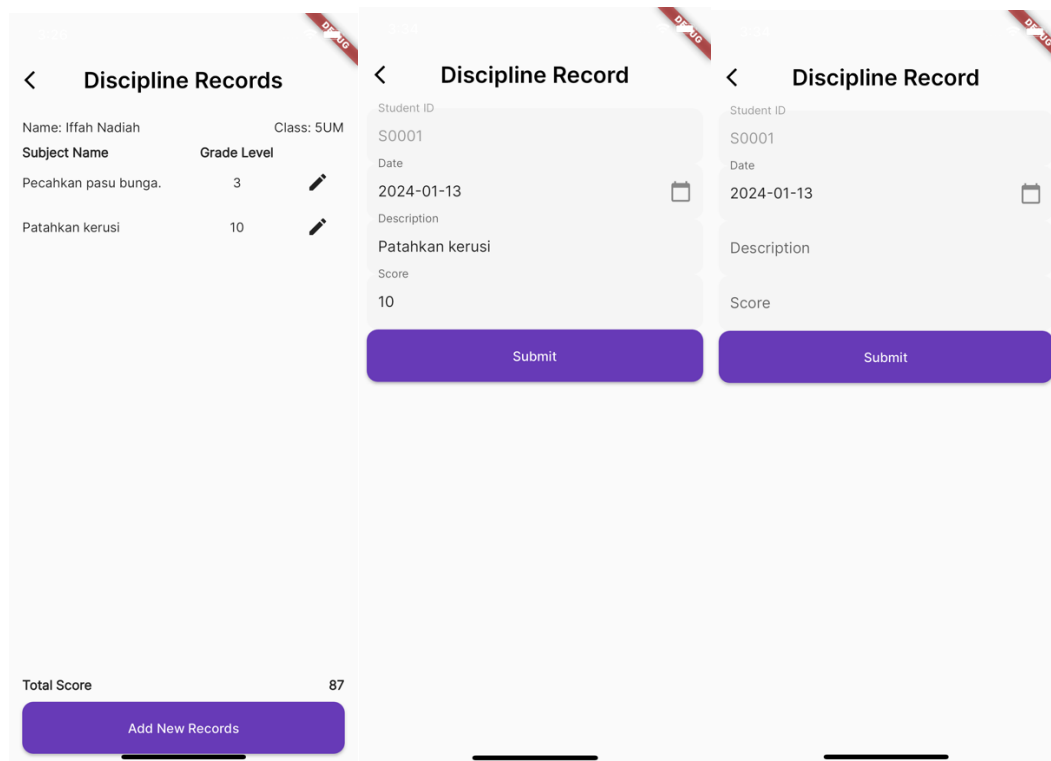
*Figure 7 Manage Student and Teacher Screen*

**Figure 7** shows that users are taken to specific pages where they can manage their teacher and student profiles. These are designed to present data in a paginated table format, which makes it easier for users to quickly and easily browse through big data sets. The table has been carefully constructed with interactive buttons that allow users to view individual records for each listed person, including academic results and discipline history. The layout makes sure that important data is easily accessible, which makes it easier to manage and supervise the important parties involved in the educational institution.

The image displays two versions of the 'View Result' screen. The left version is a form for entering or selecting data, while the right version shows the resulting data in a table format. Both screens include a back arrow, a title 'View Result', and a 'Clear' button. The table on the right has a header with 'Subject Name' and 'Grade Level'. The subjects listed are CSF3253 Intelligent System, CSM3114 Framework Mobile Dev, and CSF3253 Intelligent System. The grade levels are D, A, and B respectively. Each grade level has an edit icon (pencil). At the bottom of both screenshots are two buttons: 'Add Result' and 'Print'.

**Figure 8** Result Screen

**Figure 8** dedicated page with student results broken down by term. The ability to directly edit these results within the page itself gives staff members the ability to update and modify student performance records in real time. The page has an option that lets you print summary results reports in addition to editing features. This feature makes it easier to produce hard copy academic reports that can be distributed to parents and students for their perusal or used for administrative purposes. The design decision to combine editing and printing capabilities on one page was made with the intention of optimizing user engagement and system efficiency when it came to the academic reporting tools.



**Figure 9** Discipline Screen

**Figure 9** enables staff to view and manage student disciplinary actions directly. It simplifies the monitoring and management procedure in the classroom by showing the remaining merit points for every student and providing a concise summary of their disciplinary status.

## **Conclusion**

The SchoolHub prototype represents a significant first step toward my objective of developing cutting-edge mobile solutions to transform the way schools are run. Even though it's still in its infancy, this application has a lot of potential to develop into a sophisticated platform that helps schools adopt data-driven management practices automatically.

I'm eager to hone an intuitive UI/UX design after developing this prototype and broadening my Flutter skill set. In my opinion, SchoolHub should have more significant features, seamless integration with other educational systems, and predictive analytics.

The path ahead offers a thrilling chance to refine this concept into a polished product that gives educational institutions digital empowerment. I see a bright future for SchoolHub as a change agent that simplifies administrative tasks.

## **Project GitHub Repository**

Frontend: <https://github.com/apikmeister/S63755-Project-2>

Backend: [https://github.com/apikmeister/school\\_management\\_api](https://github.com/apikmeister/school_management_api)

## References

*kysely* | *Kysely*. (n.d.). <https://kysely.dev/>

*JSON to dart*. (n.d.). [https://javiercbk.github.io/json\\_to\\_dart/](https://javiercbk.github.io/json_to_dart/)

Cdleveille. (n.d.). *GitHub - cdleveille/discit-api: RESTful API for disc golf discs*. GitHub.  
<https://github.com/cdleveille/discit-api>

Proto Coders Point. (2023, February 10). *Flutter App Login with NodeJS API at backend & keep user logged in using JWT token | Part 8* [Video]. YouTube.  
<https://www.youtube.com/watch?v=qbB2aku6Otc>

Biessek, A. (2019). *Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2*.  
<https://openlibrary.telkomuniversity.ac.id/home/catalog/id/160493/slug/flutter-for-beginners-an-introductory-guide-to-building-cross-platform-mobile-applications-with-flutter-and-dart-2.html>

Biessek, A. (2019). *Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2*.  
<https://openlibrary.telkomuniversity.ac.id/home/catalog/id/160493/slug/flutter-for-beginners-an-introductory-guide-to-building-cross-platform-mobile-applications-with-flutter-and-dart-2.html>

*toastification* | *Flutter Package*. (n.d.). Dart Packages. <https://pub.dev/packages/toastification>

Lazebny, M. (2024, January 3). *Mastering error handling in Dart*. Michael Lazebny.  
<https://lazebny.io/mastering-error-handling/>

*dropdown\_button2* | *Flutter Package*. (n.d.). Dart Packages.  
[https://pub.dev/packages/dropdown\\_button2#7-using-dropdownbuttonformfield2-with-form](https://pub.dev/packages/dropdown_button2#7-using-dropdownbuttonformfield2-with-form)

*ChatGPT*. (n.d.). <https://chat.openai.com/share/3b907b2a-fe54-4f29-9933-8b41b3f37c17>

*Quick Start - ElysiaJS*. (n.d.). ElysiaJS. <https://elysiajs.com/quick-start>