

itleFlutter Project Exam
Mini CV & Video Search App

Mohamed Aassou
Supervisor: Prof. Ayad

January 1, 2026

Abstract

This document is a short technical report (*rappo*t) for a Flutter application that combines a mini CV/portfolio UI with API-driven features: a video search experience and a GitHub repositories list. The goal is to demonstrate UI composition, navigation, simple animations, and REST API integration in Flutter.

Contents

1 Project Overview	3
1.1 Objective	3
1.2 Main Screens	3
2 Technology Stack	3
3 Architecture and Code Organization	3
4 API Integration	4
4.1 Video Search API	4
4.2 GitHub Repositories API	4
5 UI and Interaction Details	4
5.1 Transitions and Animations	4
5.2 Video Details and Playback	4
6 Build and Run	4
6.1 Run the Flutter app	4
6.2 Build this report (PDF)	4
7 Assets	5
8 Screenshots	5
9 Limitations and Improvements	7
10 References	7

1 Project Overview

1.1 Objective

The application showcases:

- A portfolio-style home screen (mini CV).
- A video search feature backed by a public video API.
- A GitHub repositories list powered by the GitHub REST API.
- A simple profile/settings UI.

1.2 Main Screens

The app uses a bottom navigation bar with four tabs:

1. **Home**: profile image + skills + projects.
2. **Video Search**: query input, results list, and navigation to details.
3. **GitHub Repos**: fetch and display public repositories for a user.
4. **Profile**: tabbed interface (About / Settings / Links).

2 Technology Stack

- **Flutter/Dart**: UI framework and language.
- **HTTP**: REST calls via the `http` package.
- **Video playback**: `video_player`.
- **Animations**: `animations` (Material motion) + custom staggered transitions.
- **Navigation**: `curved_navigation_bar` for bottom navigation.

3 Architecture and Code Organization

The codebase is organized by responsibility:

- `lib/models/`: data models (e.g. GitHub repo and video objects).
- `lib/services/`: thin API clients for HTTP calls.
- `lib/*.dart`: screens/pages and UI widgets.
- `assets/`: images and fonts.

This project primarily uses `FutureBuilder` for async UI updates (no additional state management framework).

4 API Integration

4.1 Video Search API

The current implementation calls Pixabay's video API endpoint:

```
https://pixabay.com/api/videos
```

The API key and base URL are configured in `lib/env.dart`.

4.2 GitHub Repositories API

The GitHub repositories tab requests public repositories via:

```
https://api.github.com/users/mohammedassou/repos
```

It parses the response into a `GitHubRepo` model and renders a list with basic repo metadata (stars, forks, language, and last update date when available).

5 UI and Interaction Details

5.1 Transitions and Animations

The app uses:

- Staggered fade/scale for list items (helper widget in `lib/widgets/staggered_fade_scale.dart`)
- Fade-through transitions when opening a video item (Material motion `OpenContainer`).

5.2 Video Details and Playback

The video details page initializes a network video controller, starts playback, and provides a tap-to-play/pause overlay.

6 Build and Run

6.1 Run the Flutter app

From the project root:

```
flutter pub get  
flutter run
```

6.2 Build this report (PDF)

From the `report/` directory:

```
pdflatex rapport.tex
```

Run `pdflatex` twice if you want the table of contents to update.

7 Assets

A profile image is included as an app asset:

```
assets/mohamed.png
```

If you want to include it in this report (optional), you can uncomment the figure below.

8 Screenshots

The following screenshots are stored in the `demo/` folder and document the main screens of the application.

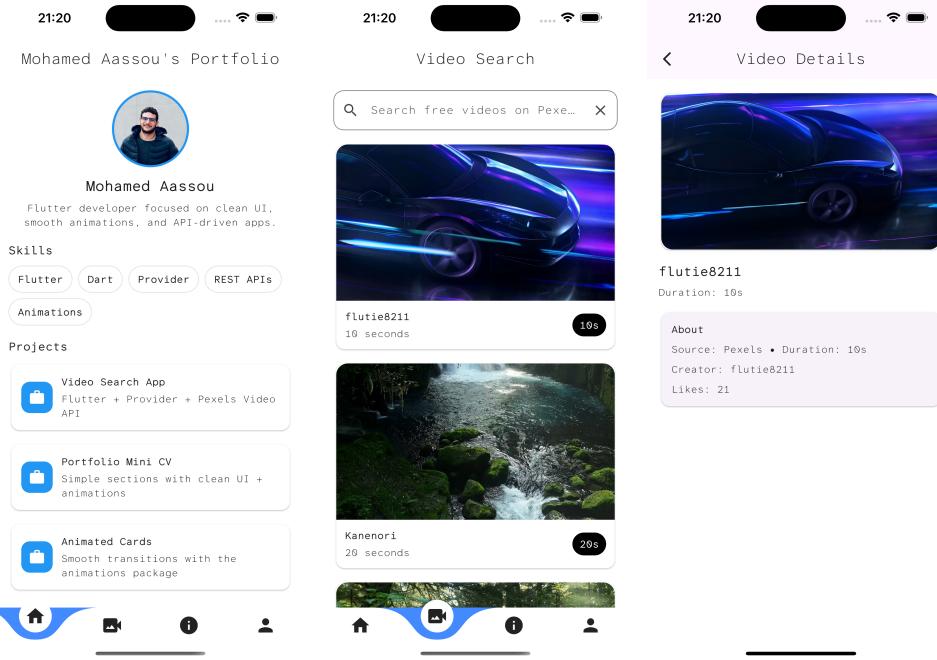


Figure 1: Home (Mini CV), Video Search, and Video Details.

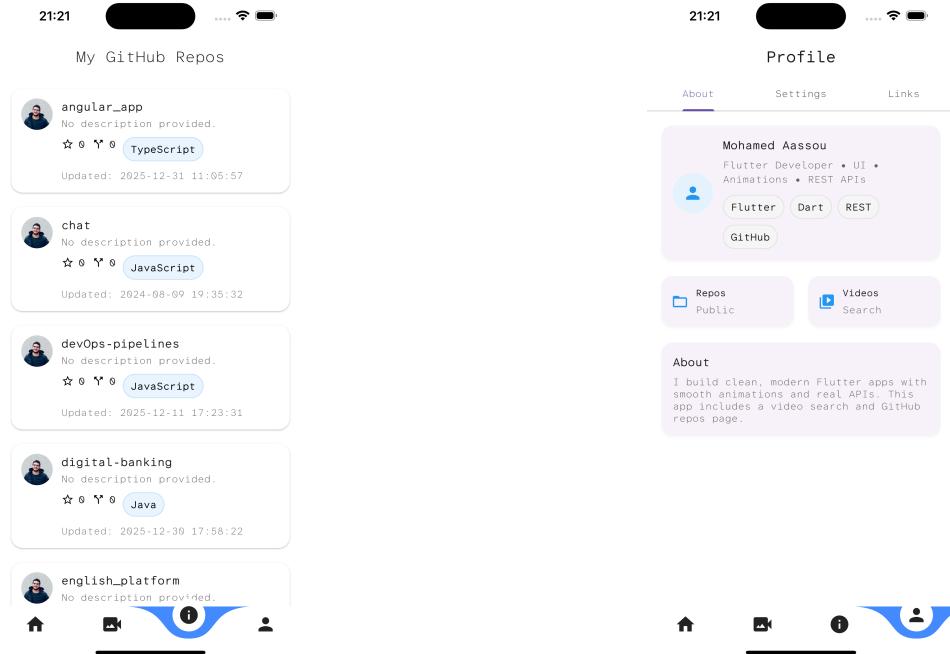


Figure 2: GitHub Repos and Profile tabs.

9 Limitations and Improvements

- The UI mentions “Pexels” in a few places, while the current API endpoint is Pixabay.
- Error handling is functional (retry + message), but could be improved with richer empty/error states.
- A dedicated state management solution (Provider/Riverpod/Bloc) could be introduced if the app grows.

10 References

- Flutter documentation: <https://docs.flutter.dev/>
- GitHub REST API: <https://docs.github.com/en/rest>
- Pixabay API (Videos): <https://pixabay.com/api/docs/>