

# Rapport de Développement de l'Application Mobile de Suivi de Matches

## Introduction

Cette application mobile a pour but de permettre à ses utilisateurs de suivre en temps réel les matchs sur lesquels ils ont parié. Elle offre une interface utilisateur intuitive qui permet de visualiser les matchs, leur état (en cours, terminé, ou non commencé), ainsi que des détails comme les scores et les commentaires. Les utilisateurs peuvent également suivre les paris qu'ils ont effectués. Cette application a été développée avec **Flutter**, une technologie multiplateforme, et un backend en **PHP** avec une base de données **MySQL**.

## Technologie utilisée : Flutter

Flutter est un framework open-source créé par Google, utilisé pour le développement d'applications natives sur Android, iOS, et le web à partir d'un seul code source. Ce choix technologique nous a permis d'avoir une seule base de code tout en ciblant plusieurs plateformes.

### Avantages de Flutter :

- **Multiplateforme** : Un seul code pour Android, iOS, et même le web.
- **Performances natives** : Les applications Flutter sont compilées directement en code machine, offrant ainsi une excellente performance.
- **Widgets** : Flutter utilise des widgets pour créer des interfaces utilisateur (UI) dynamiques et réactives, ce qui est essentiel pour une application avec des mises à jour en temps réel.
- **Hot Reload** : Flutter permet de visualiser instantanément les changements dans le code sans redémarrer complètement l'application, ce qui accélère considérablement le développement.

## Conception de la Base de Données

La base de données utilisée pour cette application a été conçue avec **XAMPP** pour gérer le serveur local et **PhpMyAdmin** pour la gestion des bases de données MySQL. Cette architecture backend assure que les informations des matchs sont bien structurées et disponibles pour l'application mobile via une API.

### Détails de la table match :

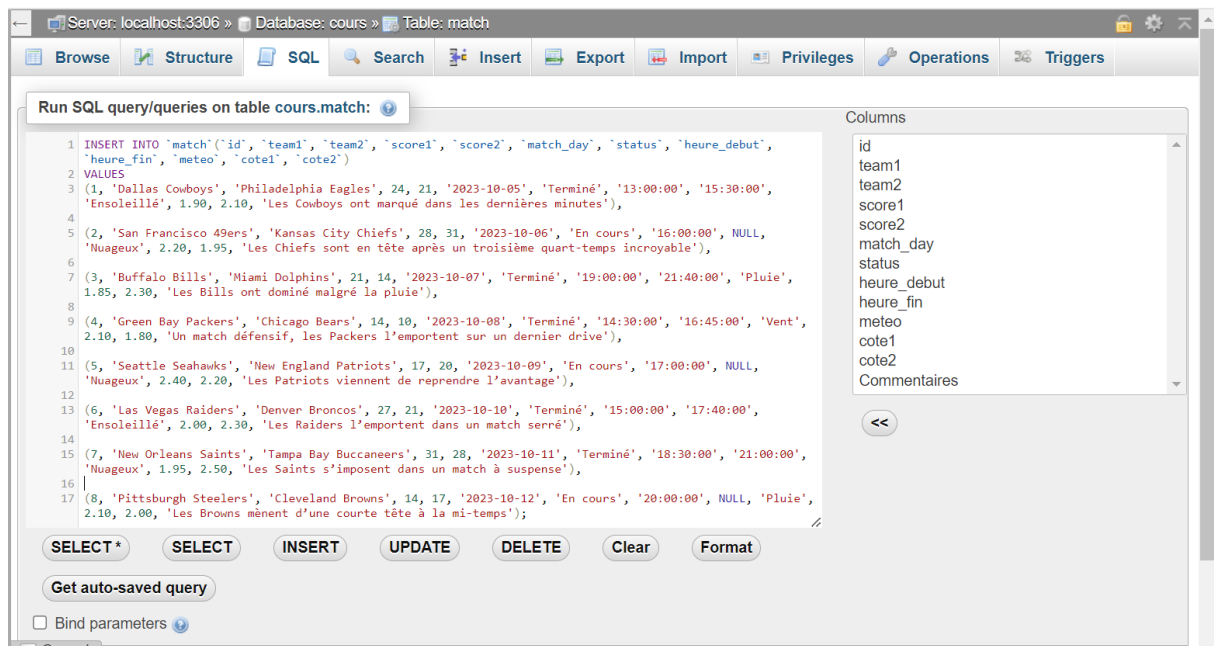
The screenshot shows the phpMyAdmin interface for a database named 'cours'. The 'match' table is selected, and its structure and data are displayed. The table has 12 columns: id, team1, team2, score1, score2, match\_day, status, heure\_debut, heure\_fin, meteo, cote1, cote2, and Commentaires. The data is as follows:

id	team1	team2	score1	score2	match_day	status	heure_debut	heure_fin	meteo	cote1	cote2	Commentaires
1	Dallas Cowboys	Philadelphia Eagles	24	21	2023-10-05	Terminé	13:00:00	15:30:00	Ensoleillé	1.9	2.1	Les Cowboys ont marqué dans les dernières minutes
2	San Francisco 49ers	Kansas City Chiefs	28	31	2023-10-06	En cours	16:00:00	NULL	Nuageux	2.2	1.95	Les Chiefs sont en tête après un troisième quart-t.
3	Buffalo Bills	Miami Dolphins	21	14	2023-10-07	Terminé	19:00:00	21:40:00	Pluie	1.85	2.3	Les Bills ont dominé malgré la pluie
4	Green Bay Packers	Chicago Bears	14	10	2023-10-08	Terminé	14:30:00	16:45:00	Vent	2.1	1.8	Un match défensif, les Packers l'emportent sur un...
5	Seattle Seahawks	New England Patriots	17	20	2023-10-09	En cours	17:00:00	NULL	Nuageux	2.4	2.2	Les Patriots viennent de reprendre l'avantage
6	Las Vegas Raiders	Denver Broncos	27	21	2023-10-10	Terminé	15:00:00	17:40:00	Ensoleillé	2	2.3	Les Raiders l'emportent dans un match serré
7	New Orleans Saints	Tampa Bay Buccaneers	31	28	2023-10-11	Terminé	18:30:00	21:00:00	Nuageux	1.95	2.5	Les Saints s'imposent dans un match à suspense
8	Pittsburgh Steelers	Cleveland Browns	14	17	2023-10-12	En cours	20:00:00	NULL	Pluie	2.1	2	Les Browns mènent d'une courte tête à la mi-temps

Nous avons créé une table match qui contient les informations suivantes :

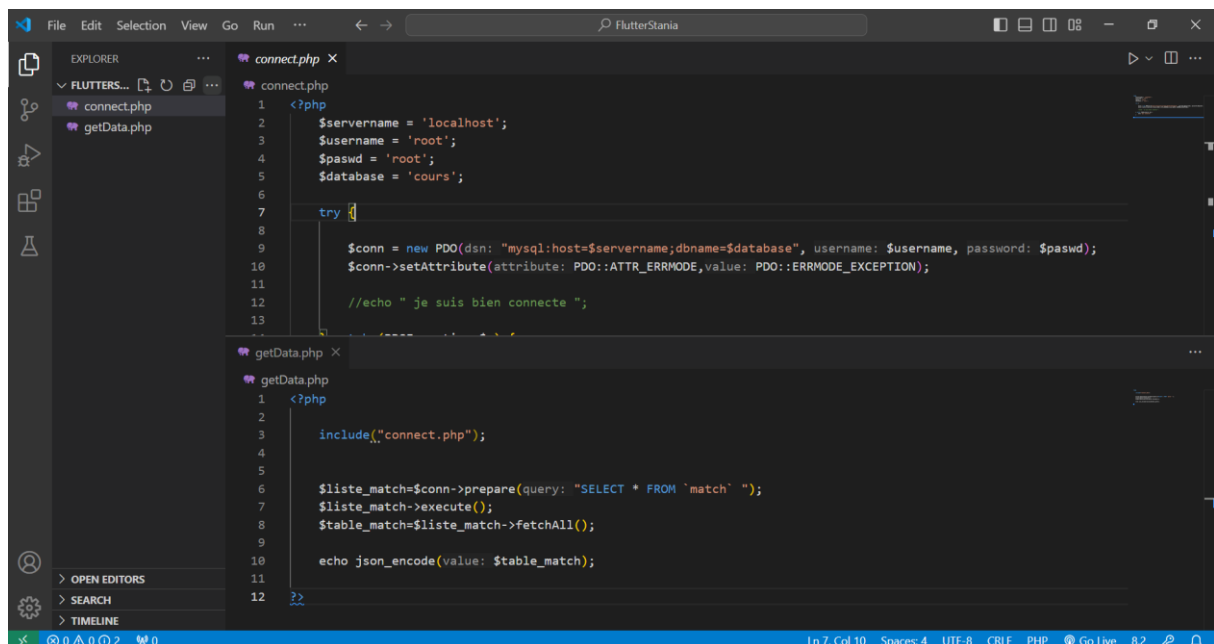
- id (INT) : Identifiant unique du match.
- team1 (VARCHAR) : Nom de la première équipe.
- team2 (VARCHAR) : Nom de la deuxième équipe.
- score1 (INT) : Score de la première équipe (si le match est terminé ou en cours).
- score2 (INT) : Score de la deuxième équipe.
- match\_day (DATE) : Date du match.
- status (ENUM) : Statut du match (par exemple, "en cours", "terminé", "pas commencé").
- heure\_debut (TIME) : Heure de début du match.
- heure\_fin (TIME) : Heure de fin prévue du match.
- meteo (VARCHAR) : Condition météorologique pendant le match.
- cote1 (FLOAT) : Cote de l'équipe 1 pour les paris.
- cote2 (FLOAT) : Cote de l'équipe 2.
- Commentaires (TEXT) : Commentaires actualisés sur le déroulement du match.

Cette table stocke toutes les informations nécessaires pour afficher la liste des matchs ainsi que les détails d'un match en particulier.



## API pour la récupération des données

Pour permettre à l'application de récupérer les données, une API a été développée en PHP. Cette API repose sur un script nommé **getData()**, qui interroge la table match et renvoie les informations sous format **JSON**, ce qui est directement utilisable par l'application Flutter. Voici un extrait du code PHP de l'API :



Cette API permet de récupérer toutes les informations sur les matchs et de les afficher dans l'application.

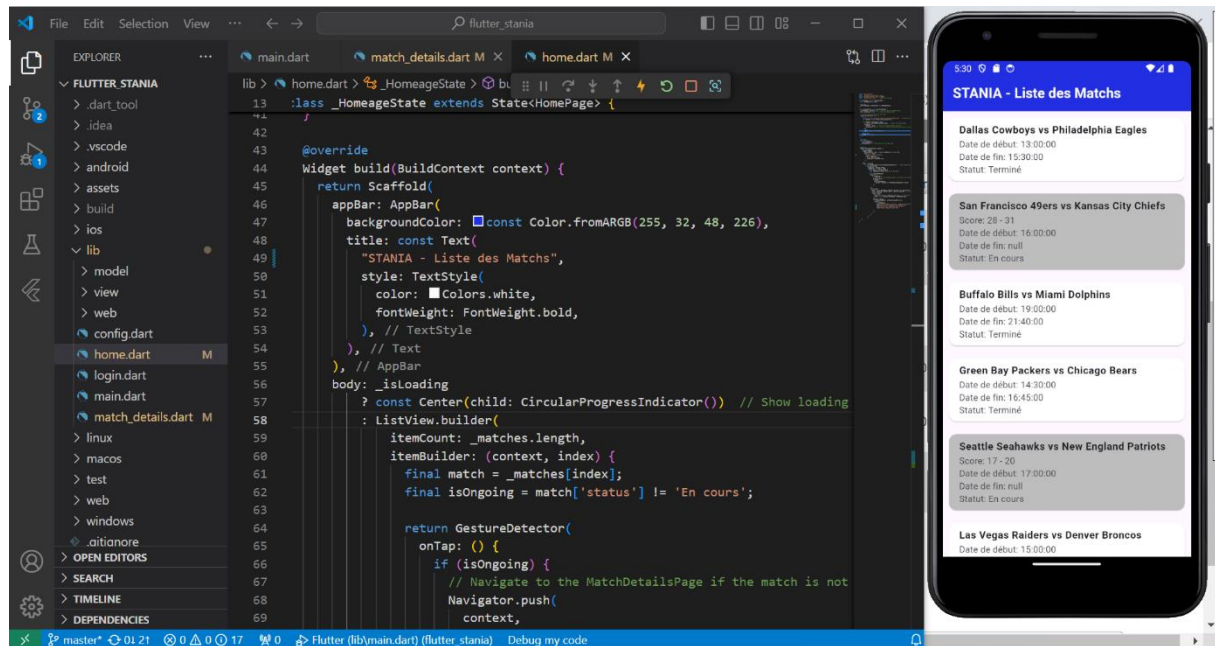
## Développement avec Flutter sur Visual Studio Code

Le développement de l'application a été réalisé en utilisant **Visual Studio Code**. Deux pages principales ont été créées :

## 1. Page d'accueil (home.dart)

Cette page affiche la liste des matchs récupérés via l'API. Les matchs sont affichés avec leur état visuel (grisé ou non selon qu'ils sont en cours ou terminés). La liste est interactive, et l'utilisateur peut cliquer sur un match pour en visualiser les détails.

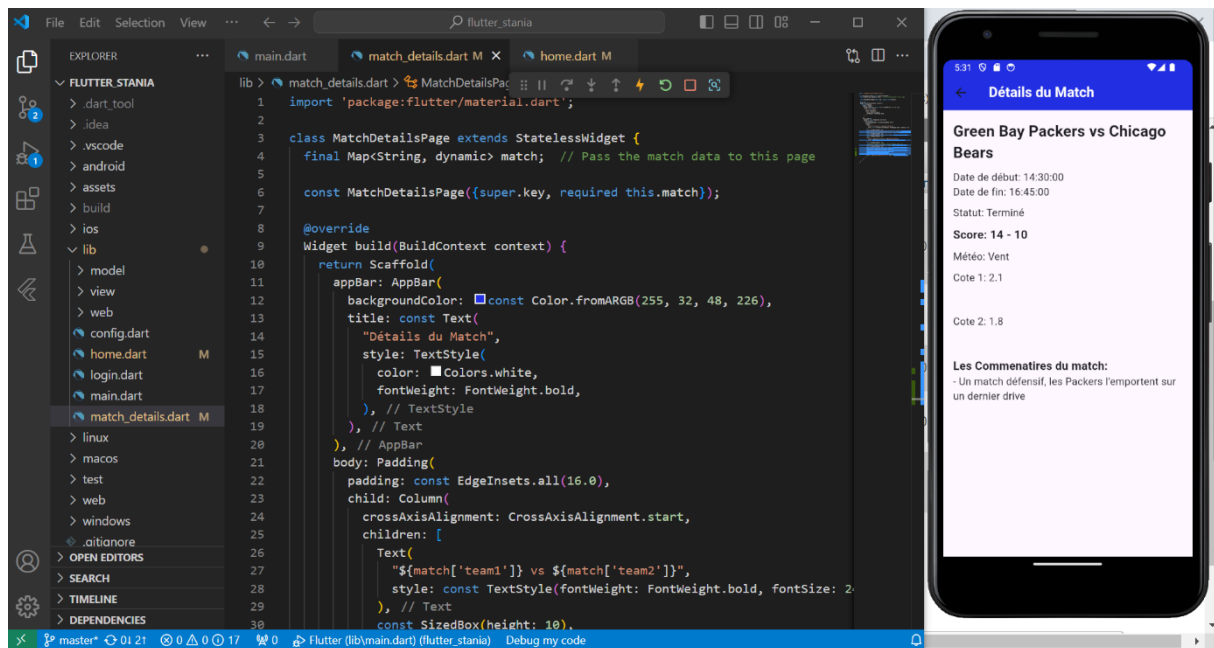
Extrait du code Flutter de home.dart :



## 2. Page des détails d'un match (details\_match.dart)

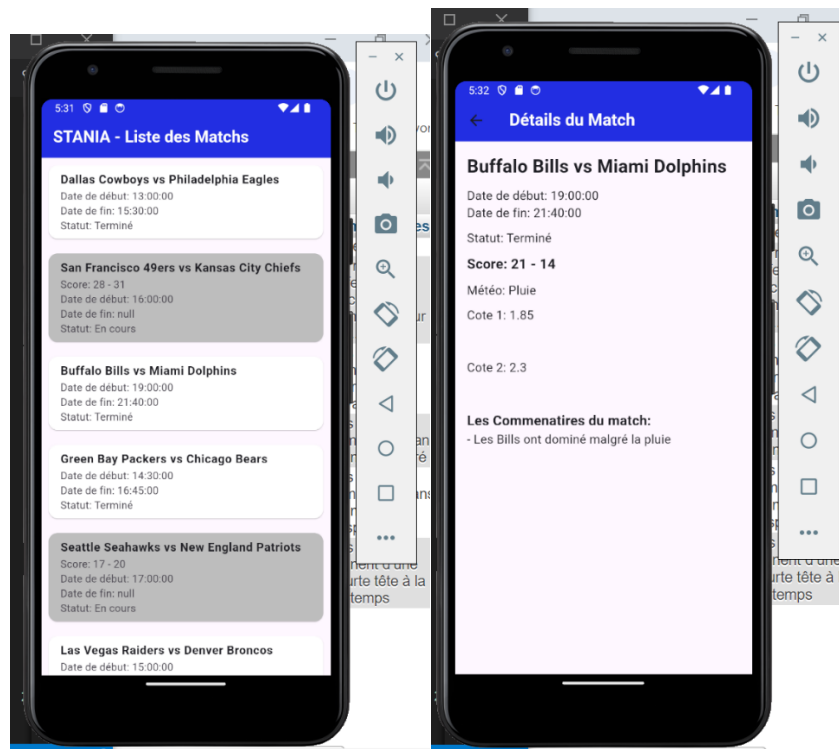
Lorsque l'utilisateur clique sur un match en cours, il est redirigé vers cette page où il peut voir des informations plus détaillées, telles que les équipes, les scores actuels, les commentaires du match, ainsi que les cotes pour les paris.

Extrait du code Flutter de details\_match.dart :



## Environnement d'exécution et de test

L'application a été testée et exécutée dans une **machine virtuelle (VM)** utilisant un émulateur Android **Pixel**. Ce choix a permis de simuler un environnement Android réel et d'effectuer des tests et du débogage directement depuis **Visual Studio Code**.



## Conclusion

Le développement de cette application mobile a permis de mettre en œuvre une interface utilisateur réactive et intuitive, grâce à **Flutter**. L'application interagit efficacement avec une base

de données MySQL à l'aide d'une API en **PHP**, permettant de récupérer et d'afficher des informations sur les matchs en temps réel. Avec la possibilité de suivre les scores et les commentaires en direct, l'application offre une expérience fluide et informative à ses utilisateurs.