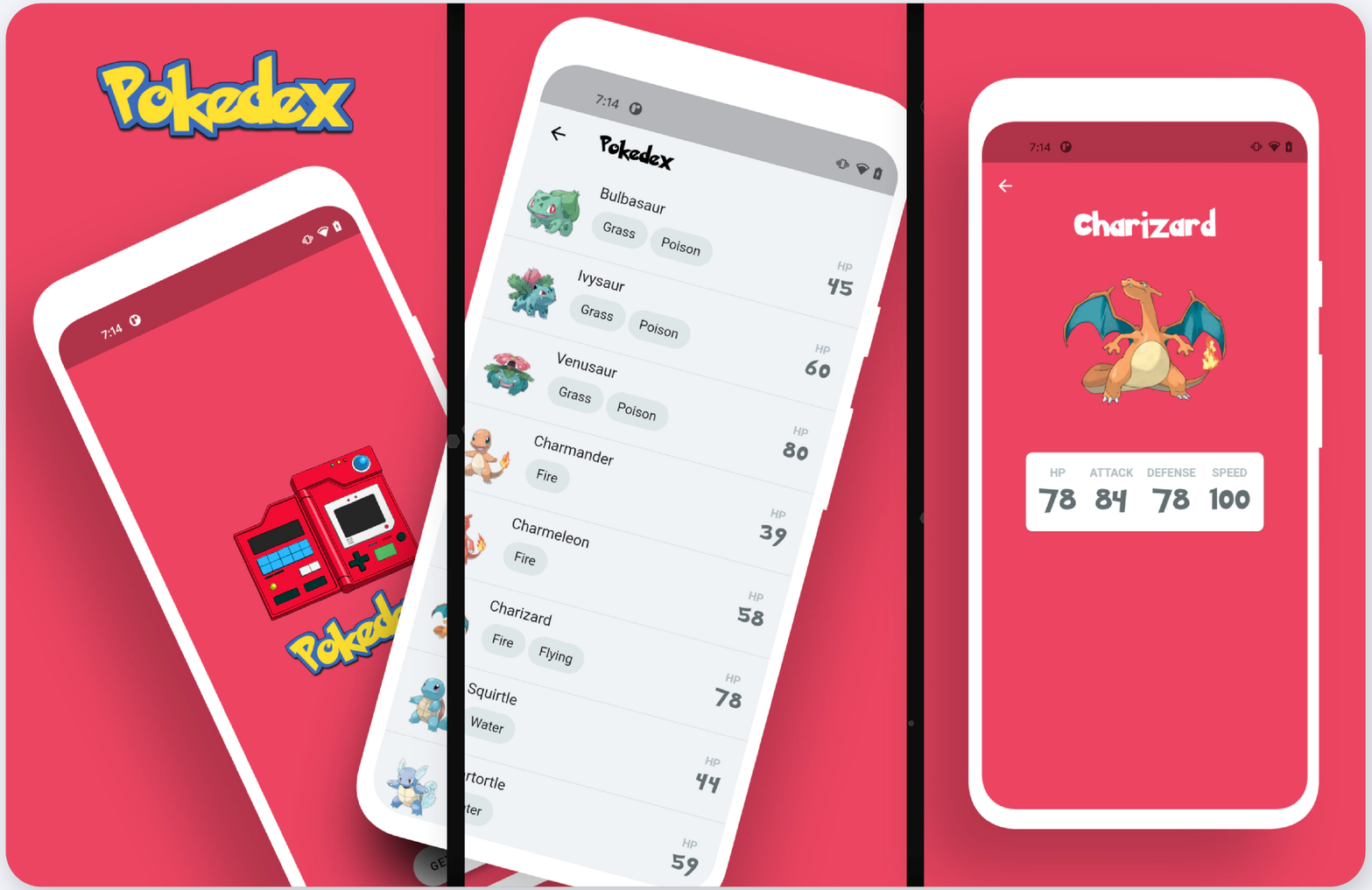
Mohamed Ibrahim

What will we be building?



https://pokedexweb.surge.sh

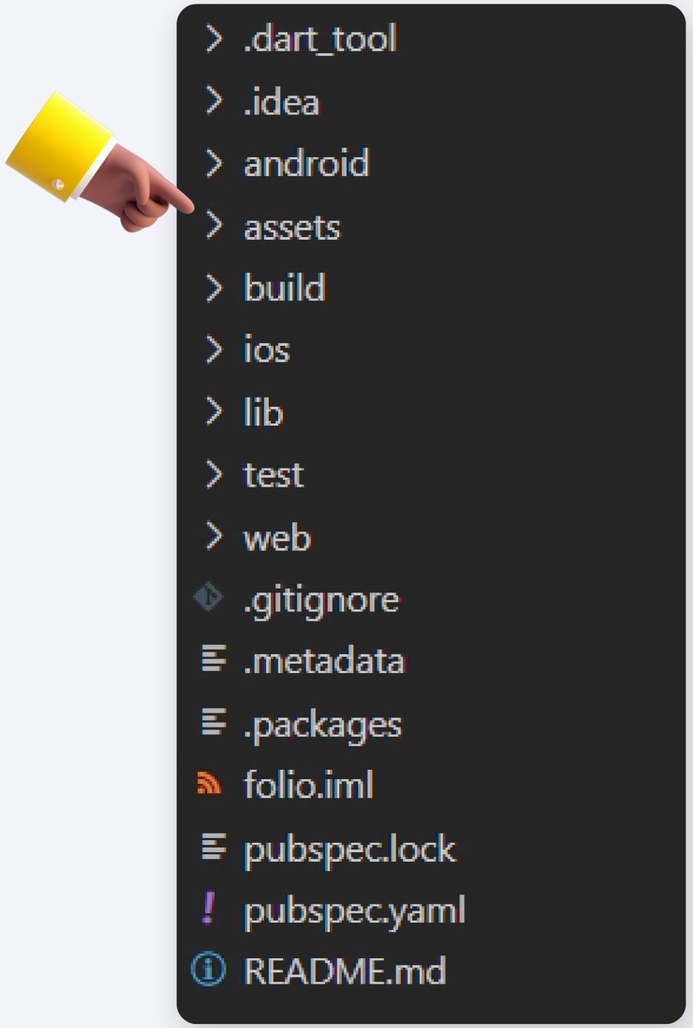
Create new project

```
$flutter create pokedex
$cd pokedex
$code . // open in vscode
```

Add project assets

Download the ***assets*** folder and add it to the root directory of the project

https://drive.google.com/drive/folders/11twWm8hmgVZH3GhdZQLkhBdLovEON2cu



Add the necessary dependencies

Step 1

Add the lint package to the dev_dependencies in *pubspec.yaml*

```
dev_dependencies:
  flutter_launcher_icons: ^0.9.0
  flutter_native_splash: ^1.1.8+4
  flutter_test:
    sdk: flutter
  lint: ^1.0.0
```

Packages:

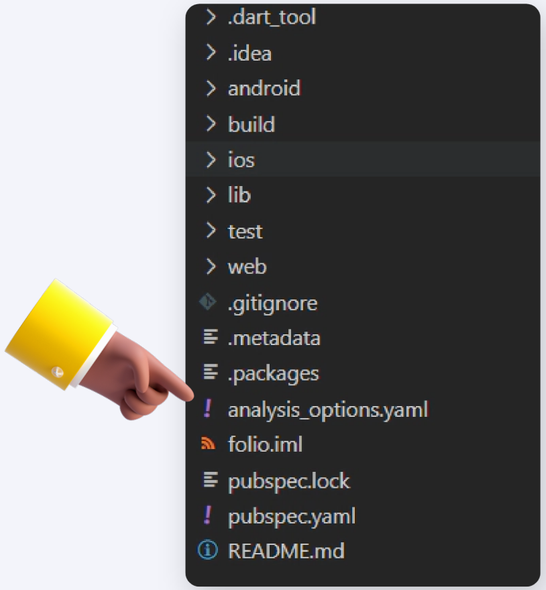
https://pub.dev/packages/lint

https://pub.dev/packages/flutter_launcher_icons

https://pub.dev/packages/flutter_native_splash

Step 2

Create a new file in the root directory and add this line




```
include: package:lint/analysis_options.yaml
```

Configure starter assets in *pubspec.yaml*

```yaml
flutter:
  uses-material-design: true
  # Add this
  assets:
    - assets/images/
    - assets/images/large/
    - assets/images/small/
  fonts:
    - family: Pokemon
      fonts:
        - asset: assets/fonts/Pokemon-Regular.ttf
```

Let's create a constants file and add the colors we want to use in our app

```
const kRedColor = Color(0×FFEC4561);
const kOffWhite = Color(0×FFF0F2F6);
const kDarkGrey = Color(0×FF636e72);
const kLightGrey = Color(0×FFb2bec3)
;
```

Starter code

```dart
import 'package:flutter/material.dart';

void main() {
  runApp(
    Pokedex(),
  );
}

class Pokedex extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(),
    );
  }
}
```

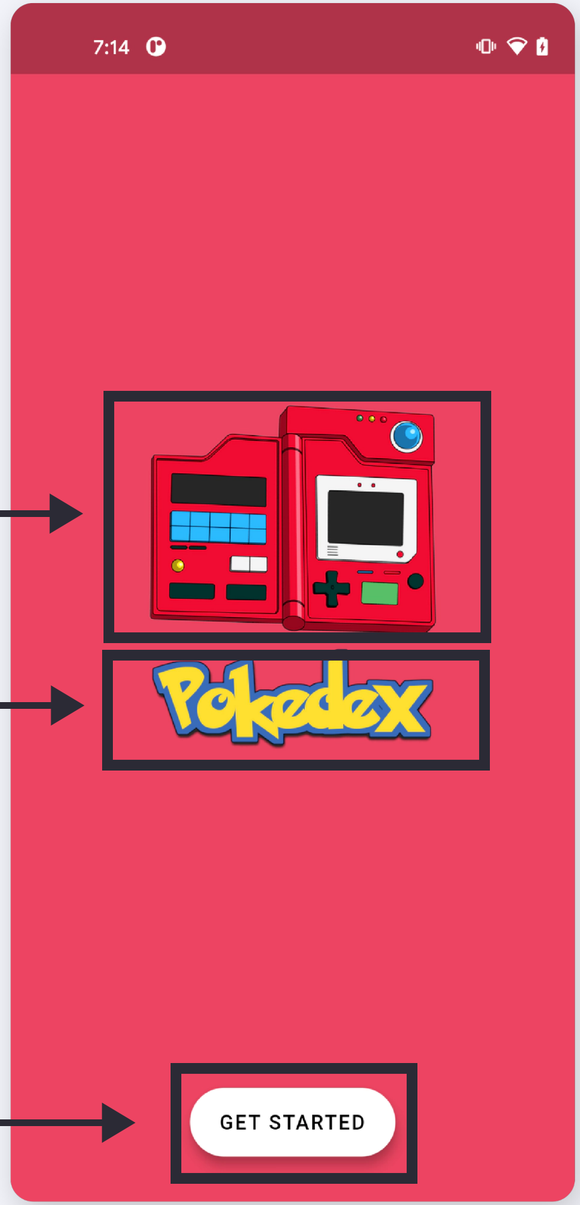
Replace *Scaffold* with a *LandingScreen* widget

```
class LandingScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Container();
  }
}
```

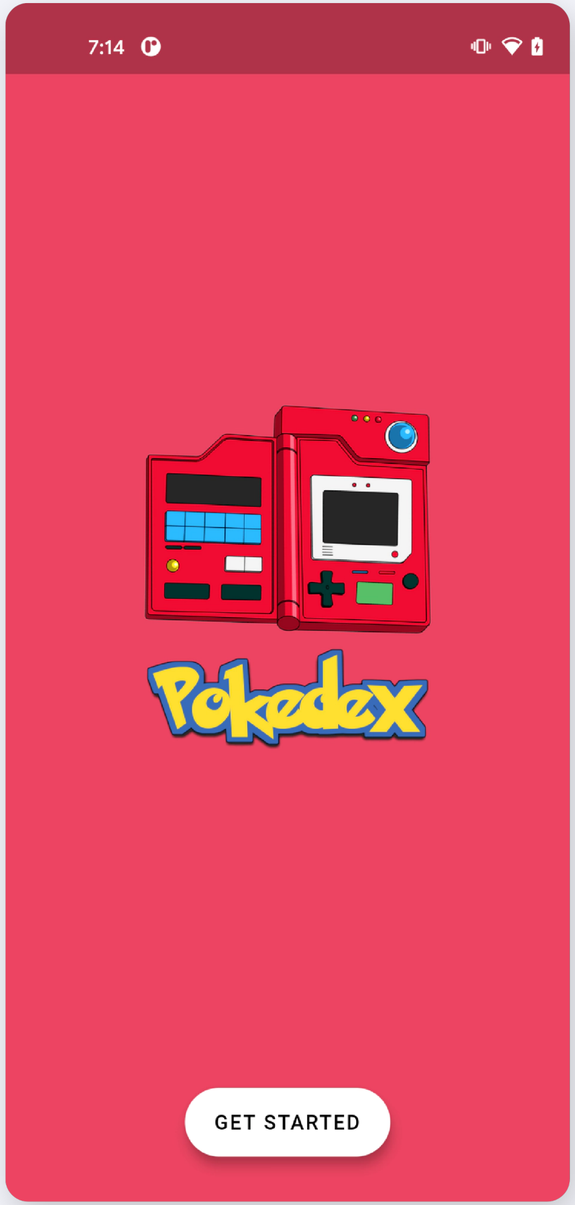
device.png

pokedex.png

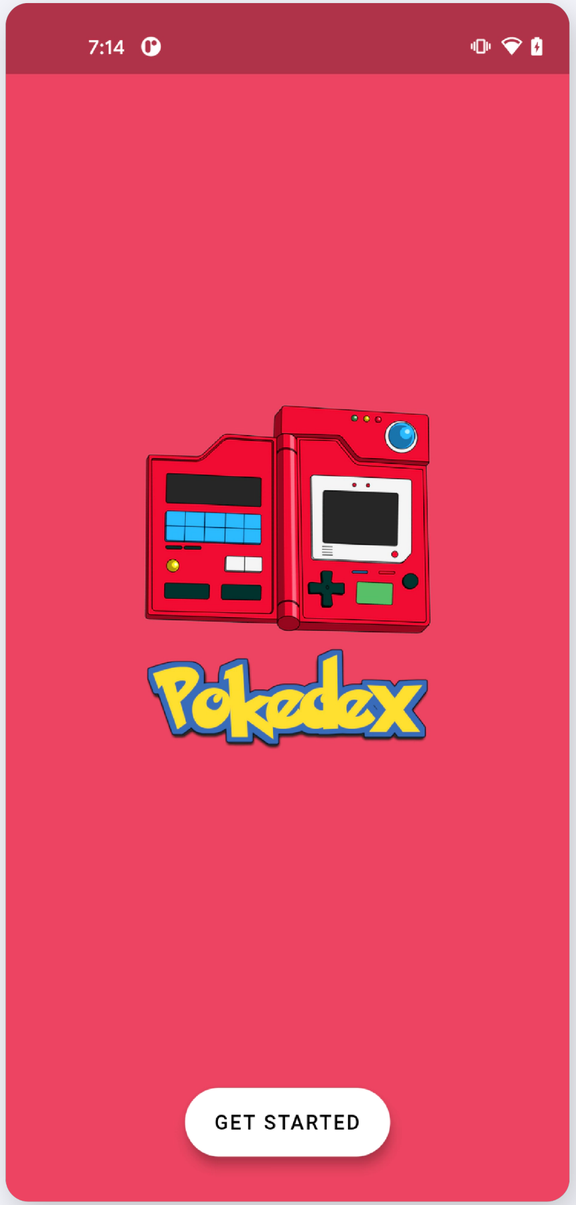
FloatingActionButton
.extended

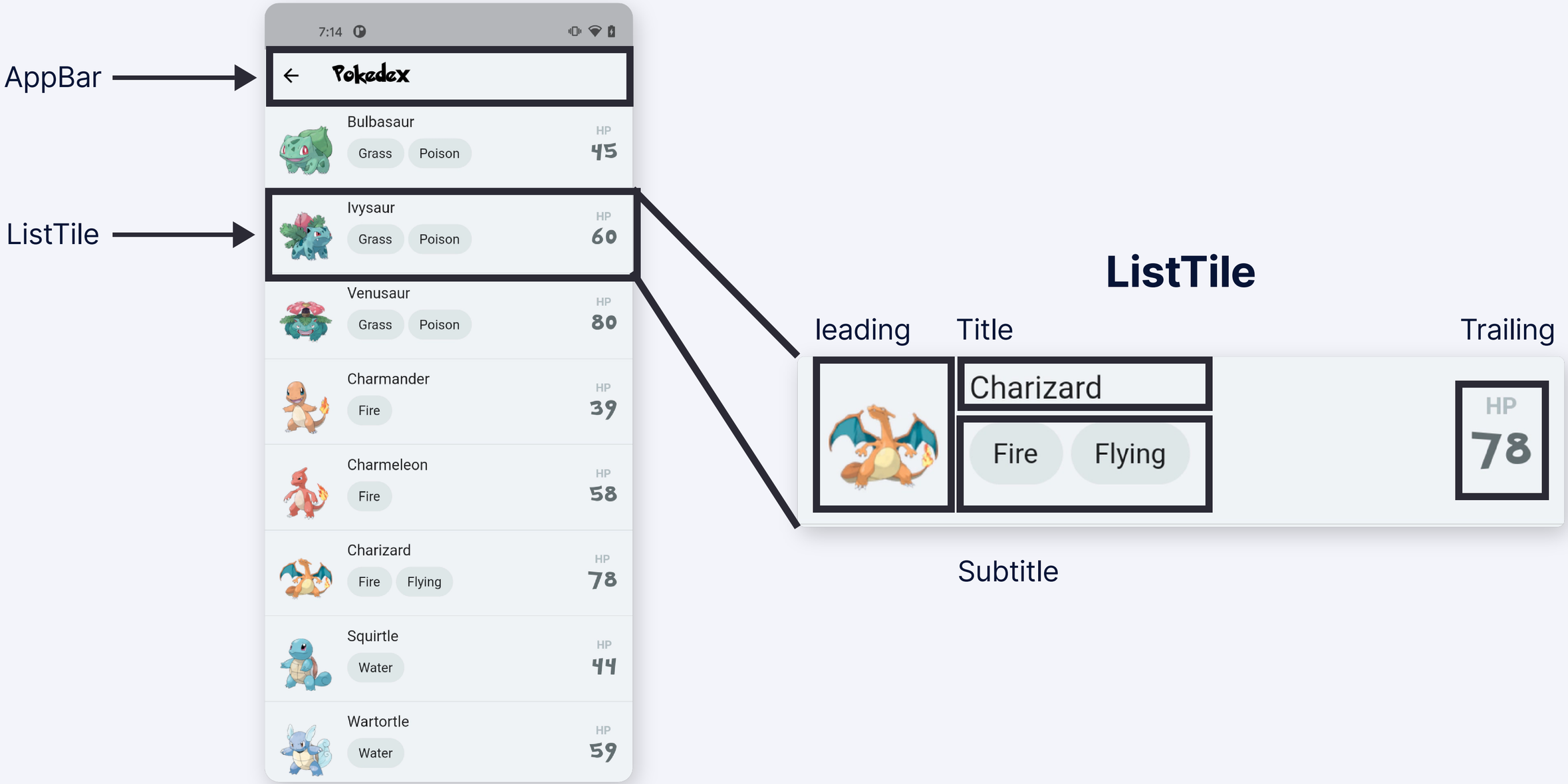


```dart
Scaffold(
    backgroundColor: kRedColor,
    body: SafeArea(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Image.asset(
            "assets/images/device.png",
            width: 200,
          ),
          const SizedBox(height: 8.0),
          Image.asset(
            "assets/images/pokedex.png",
            width: 200,
          ),
          SizedBox(
            height: MediaQuery.of(context).size.height * 0.1,
            width: double.infinity,
          ),
        ],
      ),
    ),
  );
```



```
floatingActionButton: FloatingActionButton.extended(
  onPressed: () {},
  backgroundColor: Colors.white,
  label: Text(
    'GET STARTED',
    style: TextStyle(
      color: Colors.black,
    ),
  ),
),
floatingActionButtonLocation: FloatingActionButtonLocation.centerFloat,
```




AppBar
ListTile
7:14
Pokedex
Bulbasaur
Grass
Poison
HP
45
Ivysaur
Grass
Poison
HP
60
Venusaur
Grass
Poison
HP
80
Charmander
Fire
HP
39
Charmeleon
Fire
HP
58
Charizard
Fire
Flying
HP
78
Squirtle
Water
HP
44
Wartortle
Water
HP
59
ListTile
leading
Title
Trailing
Charizard
Fire
Flying
HP
78
Subtitle

Create *HomeScreen* widget

```
class HomeScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(),
      body: SafeArea(
        child: Container(),
      ),
    );
  }
}
```

Navigate from *LandingScreen* to *HomeScreen*

```
floatingActionButton: FloatingActionButton.extended(
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) ⇒ HomeScreen(),
      ),
    );
  },
   ...
),
```

Built in types in dart

- Numbers (int, double)
- Strings (String)
- Booleans (bool)
- Lists (List, also known as *arrays*)
- Sets (Set)
- Maps (Map)
- The value null (Null)

...

https://dart.dev/guides/language/language-tour#built-in-types

Object Oriented Programming in Dart (OOP)

```dart
class Athlete {
  final String name;
  final String team;

  Athlete(this.name, this.team);

  void shoot() {
    print("shoot the ball");
  }
}

void main() {
  final mohamedSalah = Athlete("Mohamed Salah", "Liverpool");
  print(mohamedSalah.name);
  final kyrieIrving = Athlete("Kyrie Irving", "Brooklyn Nets");
  print(kyrieIrving.name);
}
```

Photo creds: Skysports, Boston globe

Object Oriented Programming in Dart (OOP)

```dart
class Pokemon {
 final String name;
 final int attack;

 Pokemon(
  this.name,
  this.attack,
 );
}

Pokemon charizard = Pokemon("Charizard", 23);
print("${charizard.name}, ${charizard.attack}");
```

https://dart.dev/guides/language/language-tour#classes

Create a data directory and add a *Pokemon* model

```dart
class Pokemon {
  final String id;
  final String name;
  final List<String> types;
  final int hitPoints;
  final int attack;
  final int defense;
  final int speed;

  Pokemon({
    required this.id,
    required this.name,
    required this.types,
    required this.hitPoints,
    required this.attack,
    required this.defense,
    required this.speed,
  });
}
```

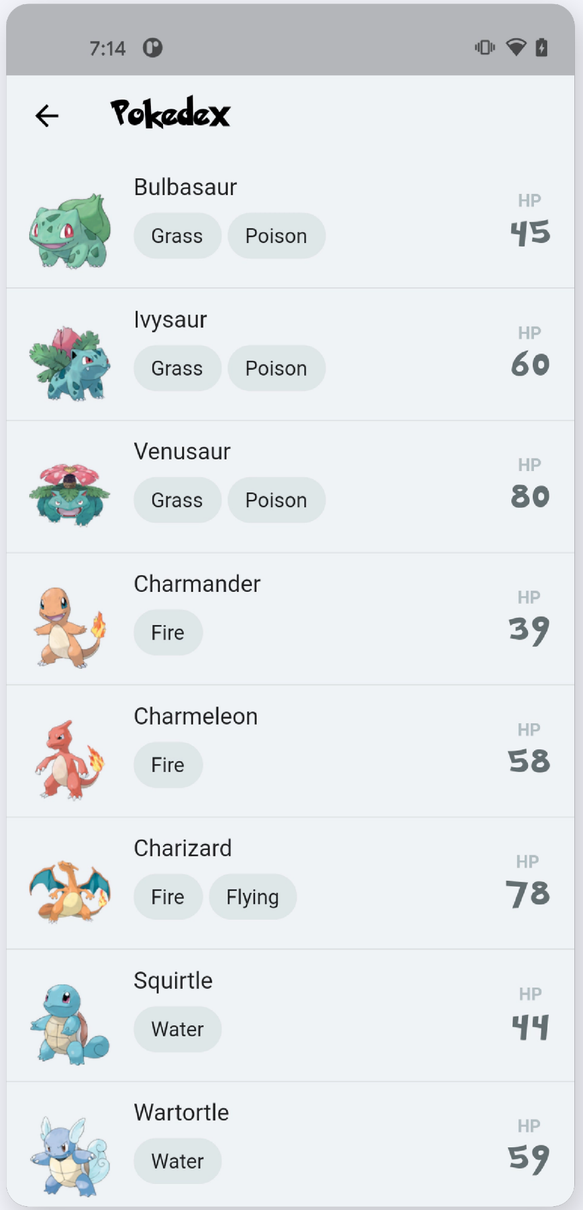
Add *Pokemon* data

Add *ListView.builder*

```
ListView.builder(
  itemCount: pokemons.length,
  itemBuilder: (context, index) {
    ... // Return widget
  },
);
```

Create *AppBar*

```
AppBar(
  title: Text(
    'Pokemon',
    style: TextStyle(
      fontFamily: 'Pokemon',
      color: Colors.black,
    ),
  ),
  backgroundColor: kOffWhite,
  elevation: 0,
  iconTheme: IconThemeData(
    color: Colors.black,
  ),
),
```



```dart
return ListTile(
  title: Text(pokemon.name),
  leading: Image.asset(
    "assets/images/small/${pokemon.id}.png",
  ),
  trailing: Column(
    children: [
      const Text(
        "HP",
        style: TextStyle(
          fontWeight: FontWeight.bold,
          color: kLightGrey,
        ),
      ),
      const SizedBox(height: 4.0),
      Text(
        pokemon.hitPoints.toString(),
        style: TextStyle(
          fontFamily: 'Pokemon',
          fontSize: 18,
          fontWeight: FontWeight.w400,
          color: kDarkGrey,
        ),
      ),
    ],
  ),
);
```

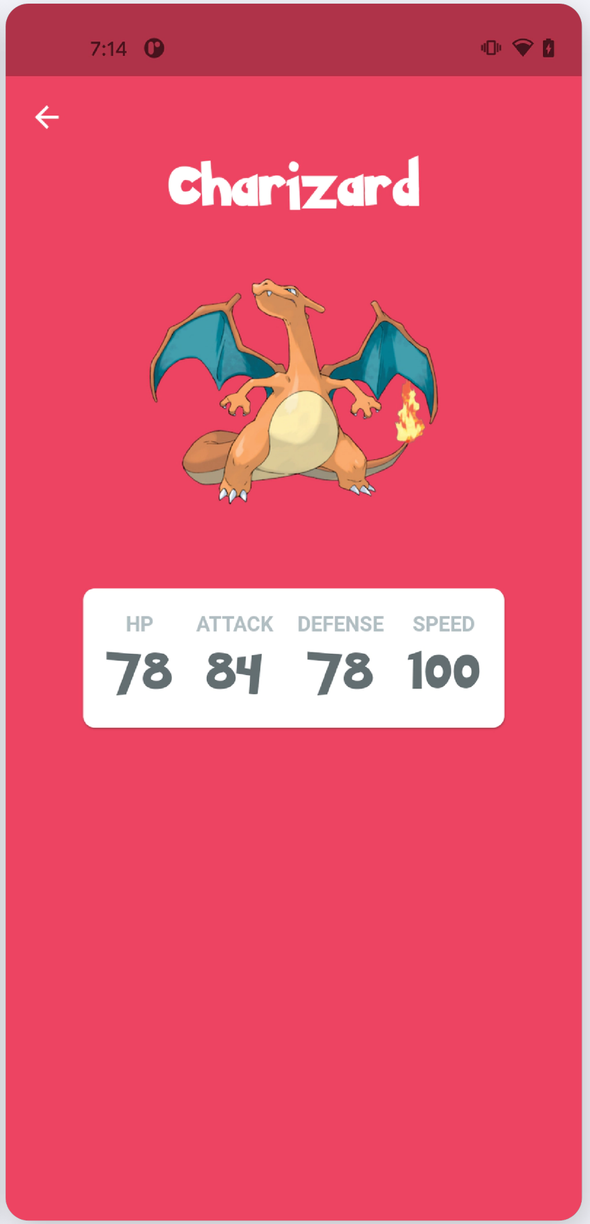
Setup *ListTileWidget*

Add types in the *Subtitle*

```
subtitle: Wrap(
  runSpacing: 4.0,
  spacing: 4.0,
  children: pokemon.types
      .map(
        (type) ⇒ Chip(
          label: Text(type),
          backgroundColor: const Color(0×FFdfe6e9),
        ),
      )
      .toList(),
),
```

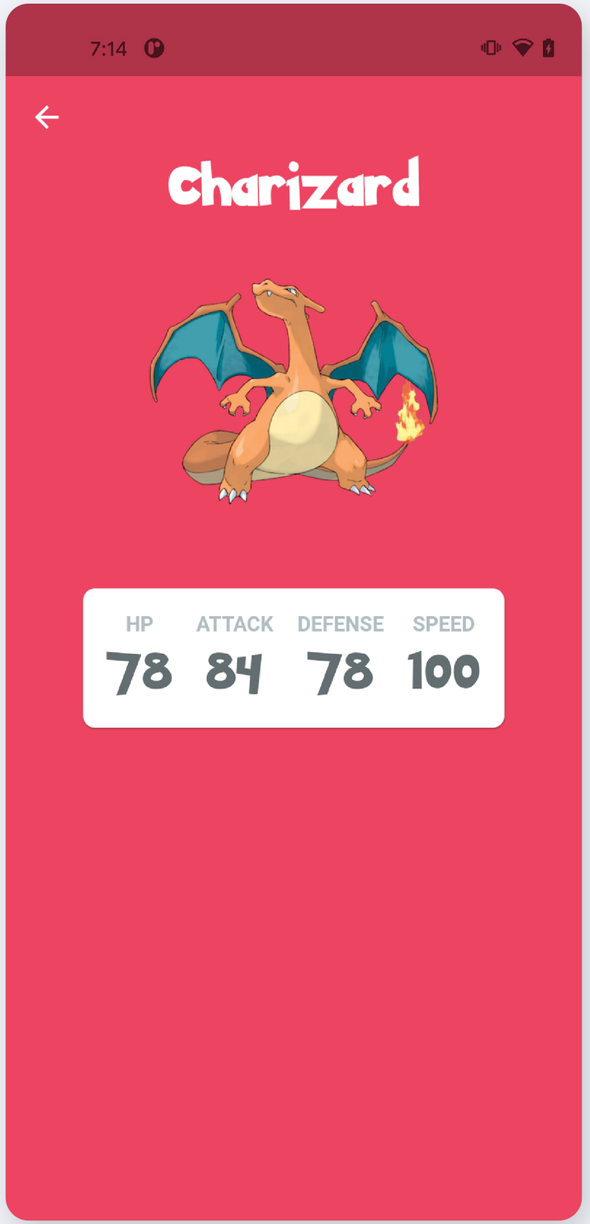
Create *DetailScreen* widget

```
final pokemon = pokemons[index];
return Scaffold(
   backgroundColor: kRedColor,
   appBar: AppBar(backgroundColor: Colors.transparent, elevation: 0),
   body: Column(
     children: [
       const SizedBox(width: double.infinity),
       Text(
         pokemon.name,
         style: const TextStyle(
             fontFamily: 'Pokemon',
             fontSize: 32,
             color: Colors.white,
             letterSpacing: 2),
       ),
       const SizedBox(height: 16.0),
       Image.asset("assets/images/large/${pokemon.id}.png"),
     ],
   ),
 );
```



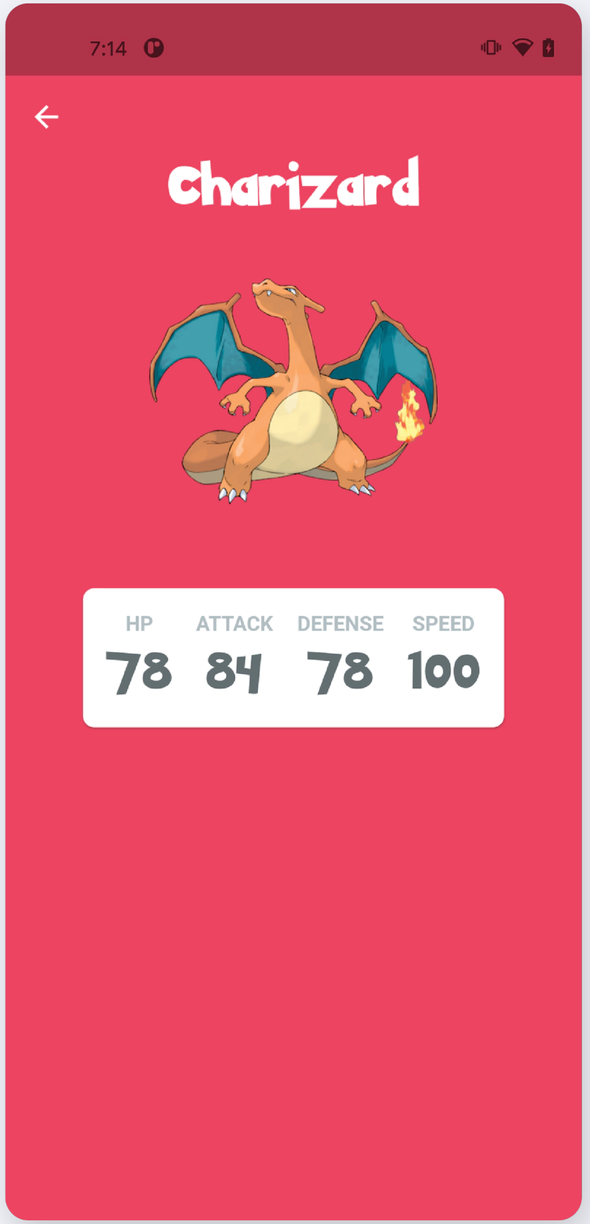
Add Stats

```
Wrap(children: [
  _renderStat(label: 'HP', value: pokemon.hitPoints.toString()),
  _renderStat(label: 'ATTACK', value: pokemon.attack.toString()),
  _renderStat(label: 'DEFENSE', value: pokemon.defense.toString()),
  _renderStat(label: 'SPEED', value: pokemon.speed.toString()),
]),
```



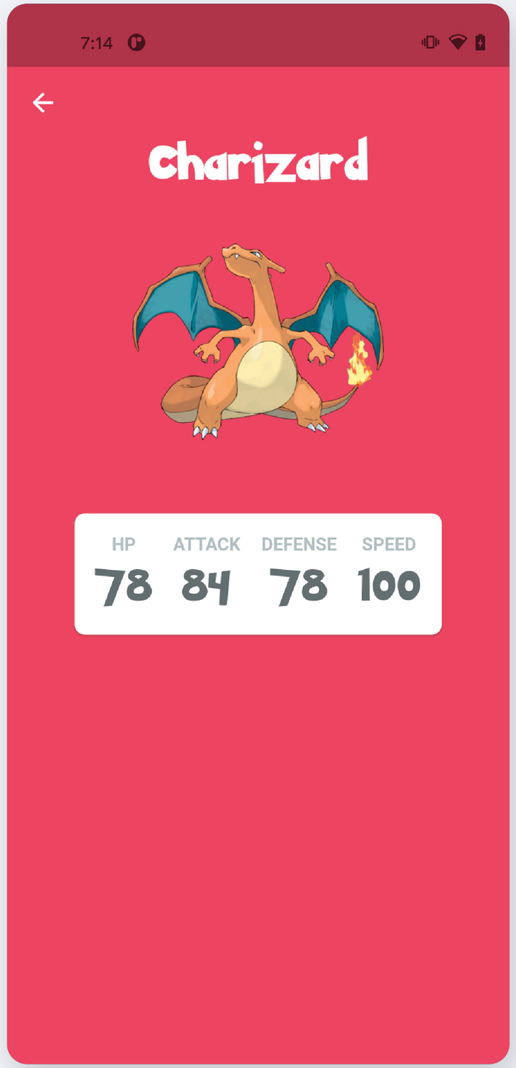
Write *_renderStat* method

```dart
Widget _renderStat({
  required String label,
  required String value,
}) {
  return Column(
    children: [
      Text(
        label,
        style: const TextStyle(
          color: Color(0×FFb2bec3),
          fontWeight: FontWeight.bold,
        ),
      ),
      const SizedBox(height: 8.0),
      Text(
        value.toString(),
        style: const TextStyle(
          color: Color(0×FF636e72),
          fontSize: 28,
          fontWeight: FontWeight.w400,
          letterSpacing: 4,
          fontFamily: 'Pokemon',
        ),
      ),
    ],
  );
}
```



Add *SizedBox* after image and Wrap Stats with a *Card* widget

```
const SizedBox(height: 16.0),
Card(
  shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(8.0),
  ),
  child: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Wrap(
        alignment: WrapAlignment.center,
        spacing: 16.0,
        runSpacing: 8.0,
   ...
```



Configure *launcher icon* and *splash screen* *in pubspec.yaml*

```yaml
flutter_icons:
  android: "launcher_icon"
  ios: true
  remove_alpha_ios: true
  image_path: "assets/images/logo.png"
  adaptive_icon_background: "assets/images/logo.png"
  adaptive_icon_foreground: "assets/images/logo.png"

flutter_native_splash:
  color: "#EC4561"
  image: "assets/images/logo.png"
  android: true
  ios: true
```

Generate launcher icons and splash screen

```
$flutter pub run flutter_launcher_icons:main

$flutter pub run flutter_native_splash:create
```

Deploy app

```
$surge
```

1. Sign in
2. Type domain and submit