

MVVC (Model-View-ViewController) হলো একটি সফটওয়্যার আর্কিটেকচার প্যাটার্ন যা Flutter অ্যাপে কোড ম্যানেজমেন্ট সহজ ও ক্লিন রাখতে ব্যবহৃত হয়। এটি একটি Flutter অ্যাপে কিভাবে আলাদা অংশ কাজ করবে তা নির্ধারণ করে। বাংলায় MVVC প্যাটার্ন নিয়ে আলোচনা করলে তা এইভাবে ব্যাখ্যা করা যায়:

MVVC-এর মূল উপাদান

1. Model

- ডেটা বা ব্যবসায়িক লজিক সংরক্ষণ করে।
- এটি ডেটা ক্লাস বা API থেকে আসা তথ্য প্রক্রিয়াজাত করে।

2. View

- UI অংশ যা ব্যবহারকারীর সাথে ইন্টারঅ্যাকশন করে।
- **Widget**-এর মাধ্যমে ডিজাইন তৈরি হয়।

3. ViewModel

- View এবং Model-এর মধ্যে ব্রিজ হিসেবে কাজ করে।
- এটি ডেটা প্রসেস করে UI-তে পাঠায়।
- **State Management Tools** (যেমন Provider, Riverpod, অথবা Flutter Bloc) ব্যবহৃত হয়।

4. Controller

- View এবং ViewModel-এর মধ্যে কমিউনিকেশন পরিচালনা করে।
- Event বা Command পরিচালনা করে।

Flutter-এ MVVC উদাহরণ

Step 1: Model তৈরি

dart

Copy code

```
class Animal {  
  
    final String name;  
  
    final String image;  
  
    Animal({required this.name, required this.image});  
  
}
```

Step 2: ViewModel তৈরি

dart

Copy code

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

import 'animal_model.dart';

class AnimalViewModel extends ChangeNotifier {

  List<Animal> _animals = [

    Animal(name: "Cat", image: "cat.png"),

    Animal(name: "Dog", image: "dog.png"),

  ];

  List<Animal> get animals => _animals;

  void addAnimal(Animal animal) {

    _animals.add(animal);

    notifyListeners();

  }

}
```

Step 3: View তৈরি

dart

Copy code

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

import 'package:provider/provider.dart';

import 'animal_viewmodel.dart';
```

```
class AnimalListView extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    final animalViewModel =
Provider.of<AnimalViewModel>(context);

    return Scaffold(

      appBar: AppBar(title: Text("Animals")),

      body: ListView.builder(

        itemCount: animalViewModel.animals.length,

        itemBuilder: (context, index) {

          final animal = animalViewModel.animals[index];

          return ListTile(

            title: Text(animal.name),

            leading: Image.asset(animal.image),

          );

        },

      ),

      floatingActionButton: FloatingActionButton(

        onPressed: () {

          animalViewModel.addAnimal(

            Animal(name: "New Animal", image:
"new_animal.png"));

        },

        child: Icon(Icons.add),
```

```
        ),  
      );  
    }  
  }  
}
```

Step 4: Main.dart ফাইল

dart

Copy code

```
import 'package:flutter/material.dart';  
  
import 'package:provider/provider.dart';  
  
import 'animal_viewmodel.dart';  
  
import 'animal_list_view.dart';  
  
  
void main() {  
  runApp(  
    MultiProvider(  
      providers: [  
        ChangeNotifierProvider(create: (_) =>  
AnimalViewModel()),  
      ],  
      child: MyApp(),  
    ),  
  );  
}
```

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      home: AnimalListView(),  
    );  
  }  
}
```

MVVC প্যাটার্নের সুবিধা

1. কোড ক্লিন এবং মেইনটেইনেবল।
2. আলাদা আলাদা লেয়ারে দায়িত্ব ভাগ হওয়ার ফলে ডিবাগিং সহজ।
3. **View** এবং **Business Logic** আলাদা করা যায়।

এটি ছোট ও মাঝারি Flutter অ্যাপের জন্য খুবই কার্যকর।