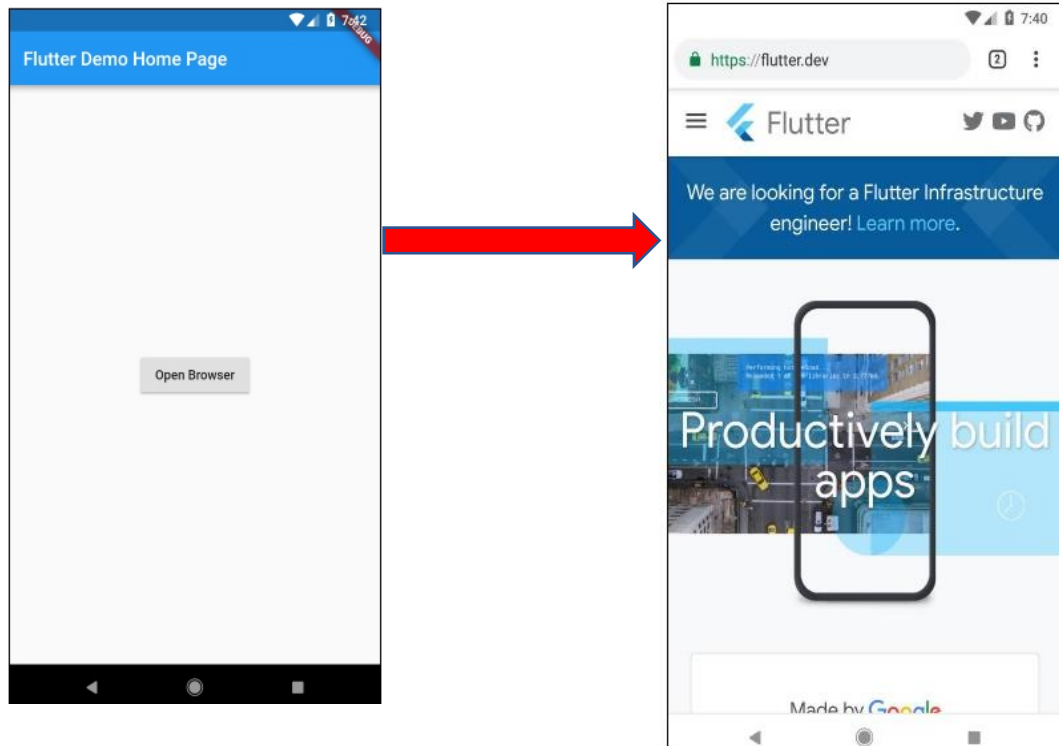


## BÀI 1 :



### MainActivity.java

```
package com.tutorialspoint.flutterapp.flutter_browser_app;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import io.flutter.app.FlutterActivity;
import io.flutter.plugin.common.MethodCall;
import io.flutter.plugin.common.MethodChannel.Result;
import io.flutter.plugins.GeneratedPluginRegistrant;

public class MainActivity extends FlutterActivity {
    private static final String CHANNEL = "flutterapp.tutorialspoint.com/browser";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        GeneratedPluginRegistrant.registerWith(this);
        new MethodChannel(getFlutterView(), CHANNEL).setMethodCallHandler(
            new MethodCallHandler() {
```

```

        @Override
        public void onMethodCall(MethodCall call, Result result) {
            String url = call.argument("url");
            if (call.method.equals("openBrowser")) {
                openBrowser(call, result, url);
            } else {
                result.notImplemented();
            }
        }
    }
}
);
}
private void openBrowser(MethodCall call, Result result, String url) {
    Activity activity = this; if (activity == null) {
        result.error(
            "ACTIVITY_NOT_AVAILABLE", "Browser cannot be opened without foreground activity", null
        );
        return;
    }
    Intent intent = new Intent(Intent.ACTION_VIEW);
    intent.setData(Uri.parse(url));
    activity.startActivity(intent);
    result.success((Object) true);
}
}
}

```

## main.dart

```

import 'package:flutter/material.dart';
import 'dart:async';
import 'package:flutter/services.dart';

void main() => runApp(MyApp());
class MyApp extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            title: 'Flutter Demo',
            theme: ThemeData(
                primarySwatch: Colors.blue,
            ),
            home: MyHomePage(
                title: 'Flutter Demo Home Page'
            ),
        );
    }
}

```

```

class MyHomePage extends StatelessWidget {
  MyHomePage({Key key, this.title}) : super(key: key);
  final String title;
  static const platform = const MethodChannel('flutterapp.tutorialspoint.com/browser');
  Future<void> _openBrowser() async {
    try {
      final int result = await platform.invokeMethod('openBrowser', <String, String>{
        'url': "https://flutter.dev"
      });
    }
    on PlatformException catch (e) {
      // Unable to open the browser print(e);
    }
  }
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(this.title),
    ),
    body: Center(
      child: RaisedButton(
        child: Text('Open Browser'),
        onPressed: _openBrowser,
      ),
    ),
  );
}

```

BÀI 2 :

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

```
void main() => runApp(new MyApp());
```

```

class Order {
  DateTime _dt;
  String _description;
}

```

```
double _total;
```

```
Order(this._dt, this._description, this._total);
```

```
double get total => _total;
```

```
String get description => _description;
```

```
DateTime get dt => _dt;
```

```
}
```

```
class Customer {
```

```
String _name;
```

```
String _location;
```

```
List<Order> _orders;
```

```
Customer(this._name, this._location, this._orders);
```

```
List<Order> get orders => _orders;
```

```
String get location => _location;
```

```
String get name => _name;
```

```
}
```

```
class MyApp extends StatelessWidget {
```

```
// This widget is the root of your application.
```

```
@override
```

```
Widget build(BuildContext context) {
```

```
return new MaterialApp(
```

```
title: 'PageView Navigation',
```

```

        theme: new ThemeData(
            primarySwatch: Colors.blue,
        ),
        home: new MyHomePage(),
    );
}
}

```

```

class MyHomePage extends StatelessWidget {
    final PageController _pageController = PageController(initialPage: 0);
    final Duration _duration = Duration(seconds: 1);
    final Curve _curve = Curves.ease;

    final List<Customer> _customerList = [
        Customer("Bike Corp", "Atlanta", [
            Order(DateTime(2018, 11, 17), "Bicycle parts", 197.02),
            Order(DateTime(2018, 12, 1), "Bicycle parts", 107.45),
        ]),
        Customer("Trust Corp", "Atlanta", [
            Order(DateTime(2017, 1, 3), "Shredder parts", 97.02),
            Order(DateTime(2018, 3, 13), "Shredder blade", 7.45),
            Order(DateTime(2018, 5, 2), "Shredder blade", 7.45),
        ]),
        Customer("Jilly Boutique", "Birmingham", [
            Order(DateTime(2018, 1, 3), "Display unit", 97.01),
            Order(DateTime(2018, 3, 3), "Desk unit", 12.25),
            Order(DateTime(2018, 3, 21), "Clothes rack", 97.15),
        ]),
    ];
}

```

```
    ),  
  ];
```

```
MyHomePage({Key key}) : super(key: key);
```

```
Widget pageViewItemBuilder(BuildContext context, int index) {  
  if (index == 0) {  
    return createHomePage(context);  
  } else {  
    return createDetailPage(context, index);  
  }  
}
```

```
Widget createHomePage(BuildContext context) {  
  List<Widget> widgetList = [];  
  widgetList.add(Padding(  
    padding: EdgeInsets.all(20.0),  
    child: Text(  
      "Customer List",  
      style: TextStyle(fontSize: 30.0, fontWeight: FontWeight.bold),  
      textAlign: TextAlign.center,  
    )),  
  for (int i = 0, ii = _customerList.length; i < ii; i++) {  
    Customer customer = _customerList[i];  
    widgetList.add(createHomePageListItem(context, customer, i));  
  }  
  return ListView(children: widgetList);
```

```
}
```

```
ListTile createHomePageListItem(  
  BuildContext context, Customer customer, int index) {  
  return new ListTile(  
    title: Text(customer.name),  
    subtitle: Text(customer.location),  
    trailing: Icon(Icons.arrow_right),  
    onTap: () => _pageController.animateToPage(index + 1,  
      duration: _duration, curve: _curve));  
  }
```

```
Widget createDetailPage(BuildContext context, int index) {  
  Customer customer = _customerList[index - 1];  
  List<Widget> widgetList = List.from(customer.orders  
    .map((Order order) => createOrderListWidget(context, customer, order)));  
  widgetList.insert(  
    0,  
    Container(  
      child: Column(  
        children: <Widget>[  
          Text(  
            customer.name,  
            style: TextStyle(fontSize: 30.0, fontWeight: FontWeight.bold),  
          ),  
          Text(  
            customer.location,
```

```

        style: TextStyle(fontSize: 24.0, fontWeight: FontWeight.bold),
    ),
    Text(
        "${customer.orders.length} Orders",
        style: TextStyle(fontSize: 20.0, fontWeight: FontWeight.bold),
    ),
    Padding(padding: EdgeInsets.all(20.0)),
],
),
padding: EdgeInsets.all(20.0)));
return ListView(children: widgetList);
}

```

```

ListTile createOrderListWidget(
    BuildContext context, Customer customer, Order order) {
    return new ListTile(
        title: Text(order.description),
        subtitle: Text("${order.dt.month}/${order.dt.day}/${order.dt.year}: "
            "\${order.total}"));
}

```

```

@override
Widget build(BuildContext context) {
    return new Scaffold(
        appBar: new AppBar(title: new Text("PageView Navigation"), actions: [
            IconButton(
                icon: Icon(Icons.home),

```



```
        onPressed: () => _pageController.animateToPage(0,  
            duration: _duration, curve: _curve))  
    ),  
    body: new Center(  
        child: new PageView.builder(  
            controller: _pageController,  
            itemBuilder: pageViewItemBuilder,  
            itemCount: _customerList.length + 1)),  
    );  
}  
}
```