HyStudy

Flutter,Django,React

FLUTTER

Tutorial 1

SCAFFOLD

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

void main()=> runApp(MaterialApp(
home: Scaffold(
appBar: AppBar(
title: Text("Twinkle App"),
centerTitle: true,
),
body:Center(
child: Text("Hello Money"),
),
floatingActionButton: FloatingActionButton(
child:Text("Click"),
onPressed: (){},
),
),
```

```
));
```

COLOURS AND FONTS

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

void main()=> runApp(MaterialApp(
home: Scaffold(
appBar: AppBar(
title: Text("Twinkle App"),
centerTitle: true,
backgroundColor: Colors.red[600],
),
body:Center(
child: Text("Hello Money Power",
style:TextStyle(
fontSize: 20,
fontWeight: FontWeight.bold,
letterSpacing: 2,
```

```
color: Colors.grey,
fontFamily: "IndieFlower",

),
),
),
floatingActionButton: FloatingActionButton(
child:Text("Click"),
onPressed: (){},
backgroundColor: Colors.red[600],
),
backgroundColor: Color.fromARGB(255, 33, 30, 30)
),

));
```

STATELESS WIDGET AND HOT RELOAD

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

void main()=> runApp(MaterialApp(
home:Home()

));

class Home extends StatelessWidget {

@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text("Twinkle App"),
centerTitle: true,
backgroundColor: Colors.red[600],
),
body:Center(
child: Text("Hello Money",
style:TextStyle(
```

```
fontSize: 20,
fontWeight: FontWeight.bold,
letterSpacing: 2,
color: Colors.grey,
fontFamily: "IndieFlower",

),
),
),
floatingActionButton: FloatingActionButton(
child:Text("Click"),
onPressed: (){},
backgroundColor: Colors.red[600],
),
backgroundColor: Color.fromARGB(255, 33, 30, 30)
);
}
}
```

IMAGES AND ASSETS

Online Images

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

void main()=> runApp(MaterialApp(
home:Home()

));

class Home extends StatelessWidget {

@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text("Twinkle App"),
```

```
centerTitle: true,
backgroundColor: Colors.red[600],
),
body:Center(
child: Image(
image: NetworkImage("https://cdn.pixabay.com/photo/2015/04/23/22
/00/tree-736885__480.jpg"),)
),
floatingActionButton: FloatingActionButton(
child:Text("Click"),
onPressed: (){},
backgroundColor: Colors.red[600],
),

);
}
}
```

Offline Images

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

void main()=> runApp(MaterialApp(
home:Home()

));

class Home extends StatelessWidget {

@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text("Twinkle App"),
centerTitle: true,
backgroundColor: Colors.red[600],
),
body:Center(
child:Image(
image: AssetImage("assets/1.jpg"),
```

```
),
),
floatingActionButton: FloatingActionButton(
child:Text("Click"),
onPressed: (){},
backgroundColor: Colors.red[600],
),

);
}
}
```

SHORTCUT

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

void main()=> runApp(MaterialApp(
home:Home()
```

```
));

class Home extends StatelessWidget {

@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text("Twinkle App"),
centerTitle: true,
backgroundColor: Colors.red[600],
),
body:Center(
child:Image.asset("assets/1.jpg"),
//child:Image.network("https://imagelocation")
),
floatingActionButton: FloatingActionButton(
child:Text("Click"),
onPressed: (){},
backgroundColor: Colors.red[600],
```

```
),

);
}
}
```

BUTTONS AND ICONS

ICONS

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

void main()=> runApp(MaterialApp(
home:Home()

));

class Home extends StatelessWidget {
```

```
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text("Twinkle App"),
centerTitle: true,
backgroundColor: Colors.red[600],
),
body:Center(
child:Icon(
Icons.airport_shuttle,
color: Colors.lightBlue,
size: 50,
),
),
floatingActionButton: FloatingActionButton(
child:Text("Click"),
onPressed: (){},
backgroundColor: Colors.red[600],
```

```
),

);
}
}
```

BUTTON

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

void main()=> runApp(MaterialApp(
home:Home()

));

class Home extends StatelessWidget {

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

```
return Scaffold(
appBar: AppBar(
title: Text("Twinkle App"),
centerTitle: true,
backgroundColor: Colors.red[600],
),
body:Center(
child:ElevatedButton.icon(onPressed: (){}, icon: Icon(Icons.mail), label:
Text("mail me",style: TextStyle(color:Colors.black )), style:
ElevatedButton.styleFrom(
backgroundColor: Colors.amber
),)

),
floatingActionButton: FloatingActionButton(
child:Text("Click"),
onPressed: (){},
backgroundColor: Colors.red[600],
),
backgroundColor: Colors.black87,
```

```
);
}
}
```

ICONBUTTON

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

void main()=> runApp(MaterialApp(
home:Home()

));

class Home extends StatelessWidget {

@override
Widget build(BuildContext context) {
return Scaffold(
```

```
appBar: AppBar(
title: Text("Twinkle App"),
centerTitle: true,
backgroundColor: Colors.red[600],
),
body:Center(
child:IconButton(
onPressed: (){},
icon: Icon(Icons.alternate_email),
color: Colors.amber,
),

),
floatingActionButton: FloatingActionButton(
child:Text("Click"),
onPressed: (){},
backgroundColor: Colors.red[600],
),
backgroundColor: Colors.black87,
);
```

```
}
}
```

CONTAINERS AND PADDING

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

void main() => runApp(MaterialApp(home: Home()));

class Home extends StatelessWidget {
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text("Twinkle App"),
centerTitle: true,
backgroundColor: Colors.red[600],
),
```

```
body: Container(
color: Colors.grey[400],
child: Text("Hello"),
padding: EdgeInsets.symmetric(horizontal: 30, vertical: 10),
margin: EdgeInsets.all(30),
),
floatingActionButton: FloatingActionButton(
child: Text("Click"),
onPressed: () {},
backgroundColor: Colors.red[600],
),
backgroundColor: Colors.black87,
);
}
}
```

PADDING WIDGET

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

```
void main() => runApp(MaterialApp(home: Home()));

class Home extends StatelessWidget {
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text("Twinkle App"),
centerTitle: true,
backgroundColor: Colors.red[600],
),
body: Padding(
child: Text("Hello"),
padding: EdgeInsets.symmetric(horizontal: 30, vertical: 10),
),
floatingActionButton: FloatingActionButton(
child: Text("Click"),
onPressed: () {},
backgroundColor: Colors.red[600],
```

```
),
backgroundColor: Colors.black87,
);
}
}
```

ROWS

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

void main() => runApp(MaterialApp(home: Home()));

class Home extends StatelessWidget {
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text("Twinkle App"),
```

```
centerTitle: true,
backgroundColor: Colors.red[600],
),
body: Row(
mainAxisAlignment: MainAxisAlignment.spaceBetween,
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Text("Hello World"),
ElevatedButton(onPressed: (){},
style: ElevatedButton.styleFrom(
backgroundColor: Colors.amber,
),
child: Text("Click Me"),),
Container(
color:Colors.cyan,
padding: EdgeInsets.all(30),
child: Text("Inside Container"),
),
],
),
```

```
floatingActionButton: FloatingActionButton(
child: Text("Click"),
onPressed: () {},
backgroundColor: Colors.red[600],
),
);
}
}
```

COLUMNS

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

void main() => runApp(MaterialApp(home: Home()));

class Home extends StatelessWidget {
@override
Widget build(BuildContext context) {
```

```
return Scaffold(
appBar: AppBar(
title: Text("Twinkle App"),
centerTitle: true,
backgroundColor: Colors.red[600],
),
body: Column(
mainAxisAlignment: MainAxisAlignment.spaceBetween,
crossAxisAlignment: CrossAxisAlignment.stretch,
children: [
Container(
padding: EdgeInsets.all(20),
color: Colors.black,
child:Image.asset("assets/1.jpg")
),
Container(
padding: EdgeInsets.all(20),
color: Colors.yellow,
child: Text("Middle"),
),
```

```
Container(
padding: EdgeInsets.all(20),
color: Colors.green,
child:Image.asset("assets/2.jpg")
),
],
),
floatingActionButton: FloatingActionButton(
child: Text("Click"),
onPressed: () {},
backgroundColor: Colors.red[600],
),
);
}
}
```

COLUMNS AND ROWS TOGETHER

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

void main() => runApp(MaterialApp(home: Home()));

class Home extends StatelessWidget {
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text("Twinkle App"),
centerTitle: true,
backgroundColor: Colors.red[600],
),
body: Column(
mainAxisAlignment: MainAxisAlignment.spaceBetween,
crossAxisAlignment: CrossAxisAlignment.stretch,
children: [
Row(
mainAxisAlignment: MainAxisAlignment.spaceEvenly,
crossAxisAlignment: CrossAxisAlignment.center,
```

```
children: [
Text("data"),
Text("HEYY"),
Text("HELLO"),
],
),
Container(
padding: EdgeInsets.all(20),
color: Colors.black,
child:Image.asset("assets/1.jpg")
),
Container(
padding: EdgeInsets.all(20),
color: Colors.yellow,
child: Text("Middle"),
),
Container(
padding: EdgeInsets.all(20),
color: Colors.green,
child:Image.asset("assets/2.jpg")
```

```
),
],
),
floatingActionButton: FloatingActionButton(
child: Text("Click"),
onPressed: () {},
backgroundColor: Colors.red[600],
),
);
}
}
```

EXPANDED WIDGETS & FLEX

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

void main() => runApp(MaterialApp(home: Home()));

class Home extends StatelessWidget {
```

```
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text("Twinkle App"),
centerTitle: true,
backgroundColor: Colors.red[600],
),
body: Row(
children: [
Expanded(
flex: 3,
child: Container(
padding: EdgeInsets.all(20),
color: Colors.red,
child: Text("One")),
),
Expanded(
flex: 2,
child: Container(
```

```
padding: EdgeInsets.all(20),
color: Colors.yellow,
child: Text("Two")),
),
Expanded(
flex: 1,
child: Container(
padding: EdgeInsets.all(20),
color: Colors.green,
child: Text("three")),
),
],
),
floatingActionButton: FloatingActionButton(
child: Text("Click"),
onPressed: () {},
backgroundColor: Colors.red[600],
),
);
}
```

```
}
```

Flutter Ninja ID Practice

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

void main(){
runApp(MaterialApp(
home: NinjaCard(),
),);
}

class NinjaCard extends StatelessWidget {

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[900],
```

```
appBar: AppBar(title: Text("Ninja ID Card"),
centerTitle: true,
backgroundColor: Colors.grey[850],
elevation: 0.0,
),
body:Padding(
padding: EdgeInsets.fromLTRB(30, 40, 30, 0),
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Center(
child: CircleAvatar(
backgroundImage: AssetImage("assets/1.jpg"),
radius: 60,
),
),
Divider(
height: 90,
color: Colors.grey[600],
),
```

```
Text(
'NAME',
style: TextStyle(
color: Colors.grey,
letterSpacing: 2,
),
),
SizedBox(height: 10,),
Text(
'Solomon Danso',
style: TextStyle(
color: Colors.amberAccent[200],
letterSpacing: 2,
fontSize: 20,
fontWeight: FontWeight.bold
),
),
SizedBox(height: 30,),

Text(
```

```
'Current Ninja Level',
style: TextStyle(
color: Colors.grey,
letterSpacing: 2,
),
),
SizedBox(height: 10,),
Text(
'8',
style: TextStyle(
color: Colors.amberAccent[200],
letterSpacing: 2,
fontSize: 20,
fontWeight: FontWeight.bold
),
),
SizedBox(height: 30,),
Row(children: [
Icon(Icons.email,
color:Colors.grey[400]),
```

```
SizedBox(width: 10,),
Text("solomondanso217@gmail.com",
style:TextStyle(
color: Colors.grey[400],
fontSize: 20,
letterSpacing: 2,
),
)
],),

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

STATEFUL WIDGETS

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

void main() {
runApp(
MaterialApp(
home: NinjaCard(),
),
);
}

class NinjaCard extends StatefulWidget {
@override
State<NinjaCard> createState() => _NinjaCardState();
}

class _NinjaCardState extends State<NinjaCard> {
int ninjaLevel = 0;
```

```
@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[900],
appBar: AppBar(
title: Text("Ninja ID Card"),
centerTitle: true,
backgroundColor: Colors.grey[850],
elevation: 0.0,
),
body: Padding(
padding: EdgeInsets.fromLTRB(30, 40, 30, 0),
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Center(
child: CircleAvatar(
backgroundImage: AssetImage("assets/1.jpg"),
radius: 60,
```

```
),
),
Divider(
height: 90,
color: Colors.grey[600],
),
Text(
'NAME',
style: TextStyle(
color: Colors.grey,
letterSpacing: 2,
),
),
SizedBox(
height: 10,
),
Text(
'Solomon Danso',
style: TextStyle(
color: Colors.amberAccent[200],
```

```
letterSpacing: 2,
fontSize: 20,
fontWeight: FontWeight.bold),
),
SizedBox(
height: 30,
),
Text(
'Current Ninja Level',
style: TextStyle(
color: Colors.grey,
letterSpacing: 2,
),
),
SizedBox(
height: 10,
),
Text(
'$ninjaLevel',
style: TextStyle(
```

```
color: Colors.amberAccent[200],
letterSpacing: 2,
fontSize: 20,
fontWeight: FontWeight.bold),
),
SizedBox(
height: 30,
),
Row(
children: [
Icon(Icons.email, color: Colors.grey[400]),
SizedBox(
width: 10,
),
Text(
"solomondanso217@gmail.com",
style: TextStyle(
color: Colors.grey[400],
fontSize: 20,
letterSpacing: 2,
```

```
),
)
],
),
],
),
),
floatingActionButton: FloatingActionButton(
onPressed: () {
setState(() {
ninjaLevel += 1;
});
},
child: Icon(Icons.man),
backgroundColor: Colors.grey[800],
),
);
}
}
```

LIST OF DATA

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

void main() => runApp(MaterialApp(
home: QuoteList(),
));

class QuoteList extends StatefulWidget {
const QuoteList({super.key});

@override
State<QuoteList> createState() => _QuoteListState();
}

class _QuoteListState extends State<QuoteList> {
List<String> quotes = [
"What will be, will be ",
"Until i win, i will never quit",
```

```
"Sometimes we win, sometimes we learn"
];

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[200],
appBar: AppBar(
title: Text("Awesome Quote"),
centerTitle: true,
backgroundColor: Colors.redAccent,
),
body: Column(
children: quotes.map((quote) {
return Text(quote);
}).toList(),
//Shorthand
//children:quotes.map((e)=>Text(e)).toList(),
));
}
```

```
}
```

CUSTOME CLASSES

quote class

```
class Quote {
String text;
String author;
Quote({ required this.text, required this.author});
}
```

quote usage

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

```
void main() => runApp(MaterialApp(
home: QuoteList(),
));

class QuoteList extends StatefulWidget {
const QuoteList({super.key});

@override
State<QuoteList> createState() => _QuoteListState();
}

class _QuoteListState extends State<QuoteList> {
List<Quote> quotes = [
Quote(text: "Until I win i will never quit", author: "Hydrogen"),
Quote(text: "Sometime we win, sometimes we learn", author:
"Champion"),
Quote(text: "The Stronger the storm, the stronger the strength", author:
"Riyard Kimpling")
];
```

```
@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[200],
appBar: AppBar(
title: Text("Awesome Quote"),
centerTitle: true,
backgroundColor: Colors.redAccent,
),
body: Column(
children: quotes.map((quote) {
return Text("${quote.text} => ${quote.author}");
}).toList(),
//Shorthand
//children:quotes.map((e)=>Text(e)).toList(),
));
}
}
```

CARDS

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

void main() => runApp(MaterialApp(
home: QuoteList(),
));

class QuoteList extends StatefulWidget {
const QuoteList({super.key});

@override
State<QuoteList> createState() => _QuoteListState();
}

class _QuoteListState extends State<QuoteList> {
List<Quote> quotes = [
Quote(text: "Until I win i will never quit", author: "Hydrogen"),
```

```
Quote(text: "Sometime we win, sometimes we learn", author:
"Champion"),
Quote(
text: "The Stronger the storm, the stronger the strength",
author: "Riyard Kimpling")
];
Widget quoteTemplate(quote) {
return Card(
margin: EdgeInsets.fromLTRB(16, 16, 16, 0),
child: Padding(
padding: const EdgeInsets.all(12.0),
child: Column(
crossAxisAlignment: CrossAxisAlignment.stretch,
children: [
Text(
quote.text,
style: TextStyle(
fontSize: 18,
color: Colors.grey[600],
),
```

```
),
SizedBox(height: 80,),
Text(
quote.author,
style: TextStyle(
fontSize: 14,
color: Colors.grey[800],
),
),
],
),
),
);
}

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[300],
appBar: AppBar(
```

```
title: Text("Awesome Quote"),
centerTitle: true,
backgroundColor: Colors.redAccent,
),
body: Column(
children:quotes.map((e)=>quoteTemplate(e)).toList(),
));
}
}
```

EXTRACTING WIDGETS

quote template file

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

class QuoteCard extends StatelessWidget {
```

```
final Quote quote;
QuoteCard({required this.quote});

@override
Widget build(BuildContext context) {
return Card(
margin: EdgeInsets.fromLTRB(16, 16, 16, 0),
child: Padding(
padding: const EdgeInsets.all(12.0),
child: Column(
crossAxisAlignment: CrossAxisAlignment.stretch,
children: [
Text(
quote.text,
style: TextStyle(
fontSize: 18,
color: Colors.grey[600],
),
),
SizedBox(
```

```
height: 80,
),
Text(
quote.author,
style: TextStyle(
fontSize: 14,
color: Colors.grey[800],
),
),
],
),
),
);
}
}
```

Main File

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

```
import "quotes.dart";
import 'quotetemplate.dart';

void main() => runApp(MaterialApp(
home: QuoteList(),
));

class QuoteList extends StatefulWidget {
const QuoteList({super.key});

@override
State<QuoteList> createState() => _QuoteListState();
}

class _QuoteListState extends State<QuoteList> {
List<Quote> quotes = [
Quote(text: "Until I win i will never quit", author: "Hydrogen"),
Quote(text: "Sometime we win, sometimes we learn", author:
"Champion"),
Quote(
```

```
text: "The Stronger the storm, the stronger the strength",
author: "Riyard Kimpling")
];

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[300],
appBar: AppBar(
title: Text("Awesome Quote"),
centerTitle: true,
backgroundColor: Colors.redAccent,
),
body: Column(
children: quotes.map((e) => QuoteCard(quote:e)).toList(),
));
}
}
```

FUNCTIONS AS ARGUMENTS

Quote template file

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

class QuoteCard extends StatelessWidget {
final Quote quote;
final Function delete;
QuoteCard({required this.quote, required this.delete});

@override
Widget build(BuildContext context) {
return Card(
margin: EdgeInsets.fromLTRB(16, 16, 16, 0),
child: Padding(
padding: const EdgeInsets.all(12.0),
```

```
child: Column(
crossAxisAlignment: CrossAxisAlignment.stretch,
children: [
Text(
quote.text,
style: TextStyle(
fontSize: 18,
color: Colors.grey[600],
),
),
SizedBox(
height: 80,
),
Text(
quote.author,
style: TextStyle(
fontSize: 14,
color: Colors.grey[800],
),
),
```

```
SizedBox(
height: 8,
),
ElevatedButton.icon(
onPressed: delete(),
icon: Icon(Icons.delete),
label: Text("delete"))
],
),
),
);
}
}
```

Main dart file

```
import 'package:flutter/material.dart';
import "quotes.dart";
import 'quotetemplate.dart';
```

```
void main() => runApp(MaterialApp(
home: QuoteList(),
));

class QuoteList extends StatefulWidget {
const QuoteList({super.key});

@override
State<QuoteList> createState() => _QuoteListState();
}

class _QuoteListState extends State<QuoteList> {
List<Quote> quotes = [
Quote(text: "Until I win i will never quit", author: "Hydrogen"),
Quote(text: "Sometime we win, sometimes we learn", author:
"Champion"),
Quote(
text: "The Stronger the storm, the stronger the strength",
author: "Riyard Kimpling")
```

```
];

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[300],
appBar: AppBar(
title: Text("Awesome Quote"),
centerTitle: true,
backgroundColor: Colors.redAccent,
),
body: Column(
children: quotes
.map((e) => QuoteCard(
quote: e,
delete: () {
setState(() {
quotes.remove(e);
});
}))
```

```
.toList(),
));
}
}
```

MAPS AND ROUTING

MAIN.DART

```
import "package:flutter/material.dart";
import 'pages/home.dart';
import 'pages/choose_locations.dart';
import 'pages/loading.dart';



void main() => runApp(MaterialApp(
```

```
initialRoute: "/home",
routes: {
"/":(context) => Loading(),
"/home":(context) => Home(),
"/location":(context) => ChooseLocation()
},
),);
```

HOME.DART

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

class Home extends StatefulWidget {
const Home({super.key});

@override
State<Home> createState() => _HomeState();
}
```

```
class _HomeState extends State<Home> {
@override
Widget build(BuildContext context) {
return Scaffold(
body: SafeArea(
child: Column(
children: [
ElevatedButton.icon(
onPressed: () {
Navigator.pushNamed(context, "/location");
},
icon: Icon(Icons.edit_location),
label: Text("Edit Location"))
],
)));
}
}
```

LOCATION.DART

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

class ChooseLocation extends StatefulWidget {
const ChooseLocation({super.key});

@override
State<ChooseLocation> createState() => _ChooseLocationState();
}

class _ChooseLocationState extends State<ChooseLocation> {
@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[200],
appBar: AppBar(
backgroundColor: Colors.blue[900],
title: Text("Choose Location"),
centerTitle: true,
elevation: 0,
),
```

```
body: Text("Location screen "),
);
}
}
```

LOADING.DART

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

class Loading extends StatefulWidget {
const Loading({super.key});

@override
State<Loading> createState() => _LoadingState();
}

class _LoadingState extends State<Loading> {
@override
Widget build(BuildContext context) {
```

```
return Scaffold(
body: Text("Loading screen"),
);
}
}
```

WIDGET LIFECYCLE

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

class ChooseLocation extends StatefulWidget {
const ChooseLocation({super.key});

@override
State<ChooseLocation> createState() => _ChooseLocationState();
}

class _ChooseLocationState extends State<ChooseLocation> {
```

```
int counter = 0;

@override
void initState() {
super.initState();
print("Init state function run");
}

@override
Widget build(BuildContext context) {
print("Init state function run build");
return Scaffold(
backgroundColor: Colors.grey[200],
appBar: AppBar(
backgroundColor: Colors.blue[900],
title: Text("Choose Location"),
centerTitle: true,
elevation: 0,
),
body: ElevatedButton(
```

```
onPressed: () {
setState(() {
counter += 1;
});
},
child: Text("Counter is $counter"),
),
);
}
}
```

ASYNCHRONOUS CODE

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

class ChooseLocation extends StatefulWidget {
const ChooseLocation({super.key});

@override
```

```
State<ChooseLocation> createState() => _ChooseLocationState();
}

class _ChooseLocationState extends State<ChooseLocation> {
int counter = 0;

void getData() async {
String username = await Future.delayed(Duration(seconds: 5), () {
return "yoshi";
});

String bio = await Future.delayed(Duration(seconds: 4), () {
return "munkaila";
});
print("$username => $bio");
}

@override
void initState() {
super.initState();
```

```
getData();
print("I will run first in initstate");
}

@override
Widget build(BuildContext context) {
print("I Will run next in the build if async is not ready ");
return Scaffold(
backgroundColor: Colors.grey[200],
appBar: AppBar(
backgroundColor: Colors.blue[900],
title: Text("Choose Location"),
centerTitle: true,
elevation: 0,
),
body: ElevatedButton(
onPressed: () {
setState(() {
counter += 1;
});
```

```
},
child: Text("Counter is $counter"),
),
);
}
}
```

FLUTTER PACKAGES

```
import "package:flutter/material.dart";
import "package:http/http.dart";
import "dart:convert";

class Loading extends StatefulWidget {
const Loading({super.key});

@override
State<Loading> createState() => _LoadingState();
}
```

```
class _LoadingState extends State<Loading> {
void getData() async {
String url = "http://127.0.0.1:8001/notes/5/";
Response kofi = await get(Uri.parse(url));
Map twinkle = jsonDecode(kofi.body);
print(twinkle["id"]);
}

@override
void initState() {
super.initState();
getData();
}

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[700],
appBar: AppBar(
```

```
backgroundColor: Colors.grey[900],
title: Text("Hydot Navi"),
centerTitle: true,
),
body: Column(mainAxisAlignment: MainAxisAlignment.spaceEvenly,
children: [
Container(
child: ElevatedButton.icon(
onPressed: () {
Navigator.pushNamed(context, "/home");
},
icon: Icon(Icons.home),
label: Text("Home"))),
Container(
child: ElevatedButton.icon(
onPressed: () {
Navigator.pushNamed(context, "/location");
},
icon: Icon(Icons.local_airport),
label: Text("Location"))),
```

```
]),
);
}

}
```

WORLD TIME API

```
import 'dart:async';

import "package:flutter/material.dart";
import "package:http/http.dart";
import "dart:convert";

class Loading extends StatefulWidget {
const Loading({super.key});
```

```
@override
State<Loading> createState() => _LoadingState();
}

class _LoadingState extends State<Loading> {
void getTime() async {
String url = "http://worldtimeapi.org/api/timezone/Europe/London";

Response hello = await get(Uri.parse(url));
Map data = jsonDecode(hello.body);

String datetime = data["datetime"];
String offset = data["utc_offset"].substring(1, 3);
print(datetime);
print(offset);

DateTime now = DateTime.parse(datetime);
now = now.add(Duration(hours: int.parse(offset)));
print(now);
```

```
//print(data);
//int week = data["week_number"];
//int day = data["day_of_year"];
//String timezone = data["timezone"];
//print("week => $week day => $day timezone => $timezone");
}

@override
void initState() {
super.initState();
getTime();
}

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[700],
appBar: AppBar(
backgroundColor: Colors.grey[900],
title: Text("Hydot Navi"),
```

```
centerTitle: true,
),
body: Column(mainAxisAlignment: MainAxisAlignment.spaceEvenly,
children: [
Container(
child: ElevatedButton.icon(
onPressed: () {
Navigator.pushNamed(context, "/home");
},
icon: Icon(Icons.home),
label: Text("Home"))),
Container(
child: ElevatedButton.icon(
onPressed: () {
Navigator.pushNamed(context, "/location");
},
icon: Icon(Icons.local_airport),
label: Text("Location"))),
]),
);
```

```
}
}
```

Tutorial 2

SCAFFOLD

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

void main() {
runApp(MaterialApp(
home: Homepage(),
theme: ThemeData(
primarySwatch: Colors.grey,
),
));
}
```

```
class Homepage extends StatelessWidget {
const Homepage({super.key});

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[850],
appBar: AppBar(
title: Text("Home",style: TextStyle(color: Colors.grey),),
actions: [
Icon(Icons.replay_outlined, color: Colors.grey,),
Icon(Icons.restart_alt,color: Colors.grey,),
Icon(Icons.settings,color: Colors.grey,),



],
backgroundColor: Colors.grey[900],
),
);
```

```
}
}
```

CONTAINER

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

void main() {
runApp(MaterialApp(
home: Homepage(),
theme: ThemeData(
primarySwatch: Colors.grey,
),
));
}

class Homepage extends StatelessWidget {
const Homepage({super.key});
```

```
@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[850],
appBar: AppBar(
title: Text("Home",style: TextStyle(color: Colors.grey),),
actions: [
Icon(Icons.replay_outlined, color: Colors.grey,),
Icon(Icons.restart_alt,color: Colors.grey,),
Icon(Icons.settings,color: Colors.grey,),
],
backgroundColor: Colors.grey[900],
),
body:Center(
child:Container(
clipBehavior: Clip.antiAlias,
padding: EdgeInsets.all(10),
width: 100,
height: 100,
```

```
alignment: Alignment.topLeft,
decoration: BoxDecoration(color: Colors.grey,
shape: BoxShape.circle,
//borderRadius: BorderRadius.circular(10),
gradient: LinearGradient(colors:[Colors.pink, Colors.yellow,]),

boxShadow: [BoxShadow(
color: Color.fromARGB(255, 26, 10, 9),
blurRadius: 4
),],


),
child: Text("Box Me"),

)
),
);
}
}
```

ROWS AND COLUMNS

```
i
import "package:flutter/material.dart";

void main() {
runApp(MaterialApp(
home: Homepage(),
theme: ThemeData(
primarySwatch: Colors.grey,
),
));
}

class Homepage extends StatelessWidget {
const Homepage({super.key});

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

```
return Scaffold(
backgroundColor: Colors.grey[850],
appBar: AppBar(
title: Text("Home",style: TextStyle(color: Colors.grey),),
actions: [
Icon(Icons.replay_outlined, color: Colors.grey,),
Icon(Icons.restart_alt,color: Colors.grey,),
Icon(Icons.settings,color: Colors.grey,),
],
backgroundColor: Colors.grey[900],
),
body:Center(
child:Container(
width: 300,
height: 500,
color: Colors.grey,
child: Row(
mainAxisAlignment: MainAxisAlignment.spaceAround,
crossAxisAlignment: CrossAxisAlignment.end,
children: [
```

```
Container(
width: 100,
height:100,
padding: EdgeInsets.all(8),
decoration: BoxDecoration(
shape: BoxShape.circle,
gradient: LinearGradient(colors: [
Colors.pink,
Colors.yellow
])
),
),
Container(
width: 100,
height:100,
padding: EdgeInsets.all(8),
decoration: BoxDecoration(
borderRadius: BorderRadius.circular(10),
gradient: LinearGradient(colors: [
Colors.lightBlue,
```

```
Color.fromARGB(255, 17, 16, 3),

])
),
),
Container(
width: 100,
height:100,
padding: EdgeInsets.all(8),
decoration: BoxDecoration(
shape: BoxShape.circle,
gradient: LinearGradient(colors: [
Colors.yellow,
Colors.pink,
])
),
),
```

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

MATERIAL DRAWER | LISTVIEW | CIRCLE AVATAR | NETWORK IMAGE | FLOATING ACTION BUTTON

PART 1

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

```
void main() {
runApp(MaterialApp(
home: Homepage(),
theme: ThemeData(
primarySwatch: Colors.grey,
),
));
}

class Homepage extends StatelessWidget {
const Homepage({super.key});

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[850],
appBar: AppBar(
title: Text("Home",style: TextStyle(color: Colors.grey),),
actions: [
```

```
Icon(Icons.replay_outlined, color: Colors.grey,),
Icon(Icons.restart_alt,color: Colors.grey,),
Icon(Icons.settings,color: Colors.grey,),
],
backgroundColor: Colors.grey[900],
),
body:Center(
child:Container(
width: 300,
height: 500,



)
),
drawer: Drawer(child: ListView(
padding: EdgeInsets.all(0),
children: [
DrawerHeader(child: Text("Hydot Tech",style: TextStyle(
color: Colors.grey,
fontSize: 30,
```

```
letterSpacing: 2.1
),),
decoration: BoxDecoration(
color: Colors.grey[900]
),),
ListTile(
leading: Icon(Icons.person),
title: Text("Solomon Danso"),
subtitle: Text("Software engineer"),
trailing: Icon(Icons.edit),
),
ListTile(
leading: Icon(Icons.email),
title: Text("Email"),
subtitle: Text("solomondanso217@gmail.com"),
trailing: Icon(Icons.edit),
),
ListTile(
leading: Icon(Icons.phone_callback),
title: Text("Mobile Number"),
```

```
subtitle: Text("0599626272"),
trailing: Icon(Icons.edit),
),
ListTile(
leading: Icon(Icons.whatsapp),
title: Text("Whatsapp"),
subtitle: Text("0599626272"),
trailing: Icon(Icons.edit),
),
],
),),
floatingActionButton: FloatingActionButton(onPressed:
(){},child:Icon(Icons.edit),hoverColor: Colors.grey[900],foregroundColor:
Colors.pink,heroTag: Text("Me"),),
);
}
}
```

PART 2

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

void main() {
runApp(MaterialApp(
home: Homepage(),
theme: ThemeData(
primarySwatch: Colors.grey,
),
));
}

class Homepage extends StatelessWidget {
const Homepage({super.key});

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[850],
appBar: AppBar(
title: Text("Home",style: TextStyle(color: Colors.grey),),
```

```
actions: [
Icon(Icons.replay_outlined, color: Colors.grey,),
Icon(Icons.restart_alt,color: Colors.grey,),
Icon(Icons.settings,color: Colors.grey,),
],
backgroundColor: Colors.grey[900],
),
body:Center(
child:Container(
width: 300,
height: 500,



)
),
drawer: Drawer(child: ListView(
padding: EdgeInsets.all(0),
children: [
UserAccountsDrawerHeader(
accountName: Text("Solomon Danso"),
```

```
accountEmail: Text("solomondanso217@gmail.com"),
currentAccountPicture:CircleAvatar(
backgroundImage: AssetImage("assets/ccna.png"),
),
//decoration: BoxDecoration(color: Colors.grey[900]),
),

ListTile(
leading: Icon(Icons.person),
title: Text("Solomon Danso"),
subtitle: Text("Software engineer"),
trailing: Icon(Icons.edit),
onTap: () {

},
),
ListTile(
leading: Icon(Icons.email),
title: Text("Email"),
subtitle: Text("solomondanso217@gmail.com"),
```

```
trailing: Icon(Icons.edit),
onTap: () {

},
),
ListTile(
leading: Icon(Icons.phone_callback),
title: Text("Mobile Number"),
subtitle: Text("0599626272"),
trailing: Icon(Icons.edit),
onTap: () {

},
),
ListTile(
leading: Icon(Icons.whatsapp),
title: Text("Whatsapp"),
subtitle: Text("0599626272"),
trailing: Icon(Icons.edit),
onTap: () {
```

```
},
),
],
),),
floatingActionButton: FloatingActionButton(onPressed:
(){},child:Icon(Icons.edit),hoverColor: Colors.grey[900],foregroundColor:
Colors.pink,heroTag: Text("Me"),),
);
}
}
```

TEXTFIELD AND SCROLLVIEW

PART 1

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

void main() {
runApp(MaterialApp(
home: Homepage(),
theme: ThemeData(
primarySwatch: Colors.grey,
),
));
}

class Homepage extends StatelessWidget {
const Homepage({super.key});

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[850],
appBar: AppBar(
title: Text("Home",style: TextStyle(color: Colors.grey),),
```

```
actions: [
Icon(Icons.replay_outlined, color: Colors.grey,),
Icon(Icons.restart_alt,color: Colors.grey,),
Icon(Icons.settings,color: Colors.grey,),
],
backgroundColor: Colors.grey[900],
),
body:Padding(
padding: const EdgeInsets.all(16.0),
child: Center(
child:SingleChildScrollView(
child: Card(

child: Column(
children: [
Image.asset("assets/ccna.png",
width: 300,
height: 300,
//fit: BoxFit.cover,
),
```

```
SizedBox(height: 20,),
Text("Change my name",style: TextStyle(fontSize:
20,fontWeight:FontWeight.bold),),
SizedBox(height: 20,),
Padding(
padding: const EdgeInsets.all(16.0),
child: TextField(
decoration: InputDecoration(
hintText: "Enter Your Name",
labelText: "Name",
border: OutlineInputBorder()

),
),
),
],
),
),
),
),
```

```
),
drawer: Drawer(
child: ListView(
padding: EdgeInsets.all(0),
children: [
UserAccountsDrawerHeader(
accountName: Text("Solomon Danso"),
accountEmail: Text("solomondanso217@gmail.com"),
currentAccountPicture:CircleAvatar(
backgroundImage: AssetImage("assets/ccna.png"),
),
//decoration: BoxDecoration(color: Colors.grey[900]),
),

ListTile(
leading: Icon(Icons.person),
title: Text("Solomon Danso"),
subtitle: Text("Software engineer"),
trailing: Icon(Icons.edit),
onTap: () {
```

```
},
),
ListTile(
leading: Icon(Icons.email),
title: Text("Email"),
subtitle: Text("solomondanso217@gmail.com"),
trailing: Icon(Icons.edit),
onTap: () {

},
),
ListTile(
leading: Icon(Icons.phone_callback),
title: Text("Mobile Number"),
subtitle: Text("0599626272"),
trailing: Icon(Icons.edit),
onTap: () {

},
```

```
),
ListTile(
leading: Icon(Icons.whatsapp),
title: Text("Whatsapp"),
subtitle: Text("0599626272"),
trailing: Icon(Icons.edit),
onTap: () {

},
),
],
),),
floatingActionButton: FloatingActionButton(onPressed:
(){},child:Icon(Icons.send),hoverColor:
Colors.grey[900],foregroundColor: Colors.pink,heroTag: Text("Me"),),
);
}
}
```

PART 2

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

void main() {
runApp(MaterialApp(
home: Homepage(),
theme: ThemeData(
primarySwatch: Colors.grey,
),
));
}

class Homepage extends StatefulWidget {
const Homepage({super.key});

@override
State<Homepage> createState() => _HomepageState();
}
```

```
class _HomepageState extends State<Homepage> {
var myText = "<Your Name>";
TextEditingController waw = TextEditingController();

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[850],
appBar: AppBar(
title: Text(
"Home",
style: TextStyle(color: Colors.grey),
),
actions: [
Icon(
Icons.replay_outlined,
color: Colors.grey,
),
Icon(
```

```
Icons.restart_alt,
color: Colors.grey,
),
Icon(
Icons.settings,
color: Colors.grey,
),
],
backgroundColor: Colors.grey[900],
),
body: Padding(
padding: const EdgeInsets.all(16.0),
child: Center(
child: SingleChildScrollView(
child: Card(
child: Column(
children: [
Image.asset(
"assets/ccna.png",
width: 300,
```

```
height: 300,
//fit: BoxFit.cover,
),
SizedBox(
height: 20,
),
Text(
myText,
style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
),
SizedBox(
height: 20,
),
Padding(
padding: const EdgeInsets.all(16.0),
child: TextField(
decoration: InputDecoration(
hintText: "Enter Your Name",
labelText: "Name",
border: OutlineInputBorder()),
```

```
controller:waw,
),
),
],
),
),
),
),
),
),
drawer: Drawer(
child: ListView(
padding: EdgeInsets.all(0),
children: [
UserAccountsDrawerHeader(
accountName: Text("Solomon Danso"),
accountEmail: Text("solomondanso217@gmail.com"),
currentAccountPicture: CircleAvatar(
backgroundImage: AssetImage("assets/ccna.png"),
),
//decoration: BoxDecoration(color: Colors.grey[900]),
```

```
),
ListTile(
leading: Icon(Icons.person),
title: Text("Solomon Danso"),
subtitle: Text("Software engineer"),
trailing: Icon(Icons.edit),
onTap: () {},
),
ListTile(
leading: Icon(Icons.email),
title: Text("Email"),
subtitle: Text("solomondanso217@gmail.com"),
trailing: Icon(Icons.edit),
onTap: () {},
),
ListTile(
leading: Icon(Icons.phone_callback),
title: Text("Mobile Number"),
subtitle: Text("0599626272"),
trailing: Icon(Icons.edit),
```

```
onTap: () {},
),
ListTile(
leading: Icon(Icons.whatsapp),
title: Text("Whatsapp"),
subtitle: Text("0599626272"),
trailing: Icon(Icons.edit),
onTap: () {},
),
],
),
),
floatingActionButton: FloatingActionButton(
onPressed: () {

setState(() {
myText = waw.text;
});
},
child: Icon(Icons.send),
```

```
hoverColor: Colors.grey[900],
foregroundColor: Colors.pink,

),
);
}
}
```

ORGANISING YOUR FLUTTER CODE

MAIN.DART

```
import "package:flutter/material.dart";
import 'package:flutterpur/pages/homepage.dart';

void main() {
```

```
runApp(MyApp());
}

class MyApp extends StatelessWidget {
const MyApp({super.key});

@override
Widget build(BuildContext context) {
return MaterialApp(
home: Homepage(),
theme: ThemeData(
primarySwatch: Colors.grey,
),
);
}
}
```

HOMEPAGE.DART

```dart
import "package:flutter/material.dart";
import "../drawer.dart";
import "../card.dart";
import "../body.dart";

class Homepage extends StatefulWidget {
const Homepage({super.key});

@override
State<Homepage> createState() => _HomepageState();
}

class _HomepageState extends State<Homepage> {
var myText = "<Your Name>";
TextEditingController waw = TextEditingController();

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[850],
```

```
appBar: AppBar(
title: Text("Home",style: TextStyle(color: Colors.grey),),
actions: [
Icon(Icons.replay_outlined,color: Colors.grey,),
Icon(Icons.restart_alt,color: Colors.grey, ),
Icon(Icons.settings,color: Colors.grey,),
],
backgroundColor: Colors.grey[900],
),
body: MyBody(myText: myText, waw: waw),
drawer: MyDrawer(),
floatingActionButton: FloatingActionButton(
onPressed: () {
setState(() {
myText = waw.text;
});
},
child: Icon(Icons.send),
hoverColor: Colors.grey[900],
foregroundColor: Colors.yellow,
```

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

BODY.DART

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

class MyBody extends StatelessWidget {
const MyBody({
Key? key,
required this.myText,
required this.waw,
}) : super(key: key);
```

```
final String myText;
final TextEditingController waw;

@override
Widget build(BuildContext context) {
return Padding(
padding: const EdgeInsets.all(16.0),
child: Center(
child: SingleChildScrollView(
child: MyCard(myText: myText, waw: waw),
),
),
);
}
}
```

CARD.DART

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

```
class MyCard extends StatelessWidget {
const MyCard({
Key? key,
required this.myText,
required this.waw,
}) : super(key: key);

final String myText;
final TextEditingController waw;

@override
Widget build(BuildContext context) {
return Card(
child: Column(
children: [
Image.asset(
"assets/ccna.png",
width: 300,
height: 300,
```

```
//fit: BoxFit.cover,
),
SizedBox(
height: 20,
),
Text(
myText,
style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
),
SizedBox(
height: 20,
),
Padding(
padding: const EdgeInsets.all(16.0),
child: TextField(
decoration: InputDecoration(
hintText: "Enter Your Name",
labelText: "Name",
border: OutlineInputBorder()),
controller:waw,
```

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

DRAWER.DART

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

class MyDrawer extends StatelessWidget {
const MyDrawer({super.key});

@override
Widget build(BuildContext context) {
return Drawer(
child: ListView(
```

```
padding: EdgeInsets.all(0),
children: [
UserAccountsDrawerHeader(
accountName: Text("Solomon Danso"),
accountEmail: Text("solomondanso217@gmail.com"),
currentAccountPicture: CircleAvatar(
backgroundImage: AssetImage("assets/ccna.png"),
),
//decoration: BoxDecoration(color: Colors.grey[900]),
),
ListTile(
leading: Icon(Icons.person),
title: Text("Solomon Danso"),
subtitle: Text("Software engineer"),
trailing: Icon(Icons.edit),
onTap: () {},
),
ListTile(
leading: Icon(Icons.email),
title: Text("Email"),
```

```
subtitle: Text("solomondanso217@gmail.com"),
trailing: Icon(Icons.edit),
onTap: () {},
),
ListTile(
leading: Icon(Icons.phone_callback),
title: Text("Mobile Number"),
subtitle: Text("0599626272"),
trailing: Icon(Icons.edit),
onTap: () {},
),
ListTile(
leading: Icon(Icons.whatsapp),
title: Text("Whatsapp"),
subtitle: Text("0599626272"),
trailing: Icon(Icons.edit),
onTap: () {},
),
],
),
```

```
);
}
}
```

NETWORK AND LISTVIEWS (APIS)

```
import "package:flutter/material.dart";
import "../drawer.dart";
import "../card.dart";
import "../body.dart";
import "package:http/http.dart" as http;
import "dart:convert";

class Homepage extends StatefulWidget {
const Homepage({super.key});
```

```
@override
State<Homepage> createState() => _HomepageState();
}

class _HomepageState extends State<Homepage> {
//var myText = "<Your Name>";
//TextEditingController waw = TextEditingController();
var uri = "http://0.0.0.0:8001/apis/v1/";
var url = "https://jsonplaceholder.typicode.com/photos";
var data;
@override
void initState() {
// TODO: implement initState
super.initState();
fetchData();
}

fetchData() async {
var res = await http.get(Uri.parse(url));
```

```
data = jsonDecode(res.body);
setState(() {});
}

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[850],
appBar: AppBar(
title: Text(
"Home",
style: TextStyle(color: Colors.grey),
),
actions: [
Icon(
Icons.replay_outlined,
color: Colors.grey,
),
Icon(
Icons.restart_alt,
```

```
color: Colors.grey,
),
Icon(
Icons.settings,
color: Colors.grey,
),
],
backgroundColor: Colors.grey[900],
),
body: data != null
? ListView.builder(
itemBuilder: (context, index) {
return ListTile(
title: Text(data[index]["title"]),
subtitle: Text("ID: ${data[index]['id']}"),
leading: Image.network(data[index]["url"]),
);
},
itemCount: data.length,
)
```

```
: Center(
child: CircularProgressIndicator(),
),
drawer: MyDrawer(),
floatingActionButton: FloatingActionButton(
onPressed: () {
setState(() {});
},
child: Icon(Icons.send),
hoverColor: Colors.grey[900],
foregroundColor: Colors.yellow,
),
);
}
}
```

DJANGO BACKEND FLUTTER FRONTEND

```dart
import "package:flutter/material.dart";
import "../drawer.dart";
import "../card.dart";
import "../body.dart";
import "package:http/http.dart" as http;
import "dart:convert";

class Homepage extends StatefulWidget {
const Homepage({super.key});

@override
State<Homepage> createState() => _HomepageState();
}

class _HomepageState extends State<Homepage> {
//var myText = "<Your Name>";
//TextEditingController waw = TextEditingController();
var uri = "http://0.0.0.0:8001/apis/v1/";
var url = "https://jsonplaceholder.typicode.com/photos";
var data;
```

```
@override
void initState() {
// TODO: implement initState
super.initState();
fetchData();
}

fetchData() async {
var res = await http.get(Uri.parse(uri));
data = jsonDecode(res.body);
setState(() {});
}

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[850],
appBar: AppBar(
title: Text(
"Home",
```

```
style: TextStyle(color: Colors.grey),
),
actions: [
Icon(
Icons.replay_outlined,
color: Colors.grey,
),
Icon(
Icons.restart_alt,
color: Colors.grey,
),
Icon(
Icons.settings,
color: Colors.grey,
),
],
backgroundColor: Colors.grey[900],
),
body: data != null
? ListView.builder(
```

```
itemBuilder: (context, i) {
return Card(
child: Column(
mainAxisAlignment: MainAxisAlignment.spaceBetween,
crossAxisAlignment: CrossAxisAlignment.stretch,
children: [
Text(data[i]["title"],style: TextStyle(fontSize: 25, color:
Colors.blue[500]),),
Text("${data[i]['id']}",style: TextStyle(fontSize: 25, color:
Colors.green[500]),),
Text(data[i]["description"],style: TextStyle(fontSize: 25, color:
Colors.pink[500]),),
SizedBox(height: 30,),



],


),
);
},
```

```
itemCount: data.length,
)
: Center(
child: CircularProgressIndicator(),
),
drawer: MyDrawer(),
floatingActionButton: FloatingActionButton(
onPressed: () {
setState(() {});
},
child: Icon(Icons.send),
hoverColor: Colors.grey[900],
foregroundColor: Colors.yellow,
),
);
}
}
```

GRIDVIEW

```dart
import "package:flutter/material.dart";
import "../drawer.dart";
import "../card.dart";
import "../body.dart";
import "package:http/http.dart" as http;
import "dart:convert";

class Homepage extends StatefulWidget {
const Homepage({super.key});

@override
State<Homepage> createState() => _HomepageState();
}

class _HomepageState extends State<Homepage> {
//var myText = "<Your Name>";
//TextEditingController waw = TextEditingController();
var uri = "http://0.0.0.0:8001/apis/v1/";
//var url = "https://jsonplaceholder.typicode.com/photos";
var url = "http://universities.hipolabs.com
```

```
/search?country=United+Kingdom";
var data;
@override
void initState(){
// TODO: implement initState
super.initState();
DataHq();
}

DataHq() async {
var res = await http.get(Uri.parse(url));
data = jsonDecode(res.body);
setState(() {});
}

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[850],
appBar: AppBar(
```

```
title: Text(
"Home",
style: TextStyle(color: Colors.grey),
),
actions: [
Icon(
Icons.replay_outlined,
color: Colors.grey,
),
Icon(
Icons.restart_alt,
color: Colors.grey,
),
Icon(
Icons.settings,
color: Colors.grey,
),
],
backgroundColor: Colors.grey[900],
),
```

```
body: data != null
? GridView.builder(
gridDelegate:
SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 3),
itemBuilder: (context, i) {
return Card(
child: Column(
mainAxisAlignment: MainAxisAlignment.spaceBetween,
crossAxisAlignment: CrossAxisAlignment.stretch,
children: [
Text(data[i]["name"]),
Text(data[i]["alpha_two_code"]),
//Text(data[i]["web_pages"]),
//Text(data[i]["domains"]),
Text(data[i]["country"]),
],
),
);
},
itemCount: data.length,
```

```
)
: Center(
child: CircularProgressIndicator(),
),
drawer: MyDrawer(),
floatingActionButton: FloatingActionButton(
onPressed: () {
setState(() {});
},
child: Icon(Icons.send),
hoverColor: Colors.grey[900],
foregroundColor: Colors.yellow,
),
);
}
}
```

LOGINS AND NAVIGATION

LOGIN PAGE

```
import "package:flutter/material.dart";
import 'package:flutterpur/pages/homepage.dart';

class LoginPage extends StatefulWidget {
@override
State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
final formkey = GlobalKey<FormState>();

final _username = TextEditingController();

final _password = TextEditingController();

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

```
return Scaffold(
appBar: AppBar(title: Text("Login Page")),
body: Stack(
fit: StackFit.expand,
children: [
Image.asset(
"assets/1.jpg",
fit: BoxFit.cover,
),
Center(
child: Padding(
padding: const EdgeInsets.all(8.0),
child: SingleChildScrollView(
child: Form(
key: formkey,
child: Card(
child: Padding(
padding: const EdgeInsets.all(16.0),
child: Column(
children: [
```

```
TextFormField(
controller: _username,
keyboardType: TextInputType.emailAddress,
decoration: InputDecoration(
hintText: "eg. johndoe@gmail.com",
labelText: "Username",
),
validator: (value) {

},
),
SizedBox(
height: 20,
),
TextFormField(
controller: _password,
keyboardType: TextInputType.text,
decoration: InputDecoration(
hintText: "eg. mystrongpass",
labelText: "Password",
```

```
),
validator: (value) {

},
obscureText: true,
),
SizedBox(
height: 20,
),
ElevatedButton(
onPressed: () {
formkey.currentState!.validate();
Navigator.push(
context,
MaterialPageRoute(builder: (context) => Homepage())
);
},
child: Text(
"Sign in",
style: TextStyle(color: Colors.white),
```

```
),
style: ElevatedButton.styleFrom(
backgroundColor: Colors.orange),
)
],
),
),
)),
),
),
),
],
),
);
}
}
```

HOME PAGE

```dart
import "package:flutter/material.dart";
import "../drawer.dart";
import "../card.dart";
import "../body.dart";
import "package:http/http.dart" as http;
import "dart:convert";

class Homepage extends StatefulWidget {
const Homepage({super.key});

@override
State<Homepage> createState() => _HomepageState();
}

class _HomepageState extends State<Homepage> {
//var myText = "<Your Name>";
//TextEditingController waw = TextEditingController();
var uri = "http://0.0.0.0:8001/apis/v1/";
//var url = "https://jsonplaceholder.typicode.com/photos";
var url = "http://universities.hipolabs.com
```

```
/search?country=United+Kingdom";
var data;
@override
void initState() {
// TODO: implement initState
super.initState();
DataHq();
}

DataHq() async {
var res = await http.get(Uri.parse(url));
data = jsonDecode(res.body);
setState(() {});
}

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.grey[100],
appBar: AppBar(
```

```
title: Text(
"Home",
style: TextStyle(color: Colors.grey),
),
actions: [
IconButton(
onPressed: () {
Navigator.pop(context);
},
icon: Icon(Icons.exit_to_app),
),
],
backgroundColor: Colors.grey[900],
),
body: data != null
? GridView.builder(
gridDelegate:
SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 3),
itemBuilder: (context, i) {
return Card(
```

```
child: Column(
mainAxisAlignment: MainAxisAlignment.spaceBetween,
crossAxisAlignment: CrossAxisAlignment.stretch,
children: [
Text(data[i]["name"]),
Text(data[i]["alpha_two_code"]),
//Text(data[i]["web_pages"]),
//Text(data[i]["domains"]),
Text(data[i]["country"]),
],
),
);
},
itemCount: data.length,
)
: Center(
child: CircularProgressIndicator(),
),
drawer: MyDrawer(),
floatingActionButton: FloatingActionButton(
```

```
onPressed: () {
setState(() {});
},
child: Icon(Icons.send),
hoverColor: Colors.grey[900],
foregroundColor: Colors.yellow,
),
);
}
}
```

DJANGO API BACKEND

How to create a django api from scratch

Let go straight to the point.

Before you start you will need to install

1. Python

2. Django

3. Rest framework

PART 1

[1] Start the project by running django-admin startproject MyApi

[2] change directory into the MyApi folder using the command cd MyApi

[3]Perform a migration,and create a super user account

[4] In settings.py add your app and rest_framework to your installed apps

```
INSTALLED_APPS = [
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'API',
"rest_framework",

]
```

[5] In the MyApi folder, open models.py and create your model and register it the admin panel

models.py

```
from django.db import models

class Drinks(models.Model):
```

```
name= models.CharField(max_length=200)
description = models.TextField()

def __str__(self):
return self.name
```

admin.py

```
from django.contrib import admin
from .models import Drinks

admin.site.register(Drinks)
```

[6] Now we have created our models, we need to serialize them so that it can be converted to a JSON (javascript object notation) file

```
from rest_framework import serializers
from .models import Drink
class DrinkSerializer(serializers.ModelSerializers):
    class Meta:
        model= Drink
         fields = "__all__"
```

serializers.py

```
from rest_framework import serializers
from .models import Drinks
class DrinkSerializer(serializers.ModelSerializer):
class Meta:
model= Drinks
fields = "__all__"
```

[7] Create a view for the serializers and assign the serializer in the view. It must return a Json file

views.py

```
from .models import Drinks
from .serializers import DrinkSerializer
from django.http import JsonResponse

#get all the model
#serialize them
#return json

def drink_list(request):
drinks = Drinks.objects.all()
```

```
serializer = DrinkSerializer(drinks,many=True)
return JsonResponse(serializer.data,safe=False)
```

[8] Create a url to route the views

```
from django.contrib import admin
from django.urls import path
from API import views

urlpatterns = [
path('admin/', admin.site.urls),
path("drinks/",views.drink_list)

]
```

[9] run the server using python manage.py runserver and open the url on the browser

http://127.0.0.1:8001/drinks/
[10] The default http method is GET

PART 2

Everything explained in detail in the views.py

```
from .models import Drinks
from .serializers import DrinkSerializer
from django.http import JsonResponse
from rest_framework.decorators import api_view
from rest_framework.response import Response
from rest_framework import status

#get all the model
#serialize them
#return json

@api_view(['GET','POST'])
```

```
def drink_list(request,format=None):
if request.method == "GET":
drinks = Drinks.objects.all()
serializer = DrinkSerializer(drinks,many=True)
return Response(serializer.data)

if request.method == "POST":
serializer = DrinkSerializer(data = request.data)
if serializer.is_valid():
serializer.save()
return Response(serializer.data,status=status.HTTP_201_CREATED)

#put is the same us update

@api_view(['GET','PUT','DELETE'])
def drink_detail(request,id,format=None):
#accessing the database by filtering it with an id and wrapping it in try and
except which is not compulsory
try:
drink = Drinks.objects.get(pk=id)
```

```
except Drinks.DoesNotExist:
return Response(status=status.HTTP_404_NOT_FOUND)

if request.method == "GET":
serializer = DrinkSerializer(drink)
return Response(serializer.data)

elif request.method == "PUT":
serializer = DrinkSerializer(drink, data=request.data)
if serializer.is_valid():
serializer.save()
return Response(serializer.data)
#If serializer is not valid, return this errors
return
Response(serializer.errors,status=status.HTTP_400_BAD_REQUEST)

elif request.method == "DELETE":
drink.delete()
return Response(status=status.HTTP_204_NO_CONTENT)
```

addition routes in the url.py

```
from django.contrib import admin
from django.urls import path
from API import views
from rest_framework.urlpatterns import format_suffix_patterns

urlpatterns = [
path('admin/', admin.site.urls),
path("drinks/",views.drink_list),
path("drinks/<int:id>",views.drink_detail)
]
#It will help us to add .json extension
urlpatterns = format_suffix_patterns(urlpatterns)
```

Django Cors

For django to work, we need to install django-cors

Now in the settings.py add the following
[1]
```
ALLOWED_HOSTS = ["*"]
CORS_ORIGIN_ALLOW_ALL = True
```

[2] In the middleware, include this

```
 'corsheaders.middleware.CorsMiddleware',
    'django.middleware.common.CommonMiddleware',
```

[3] Full Settings.py

```
from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings - unsuitable for production
```

```
# See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-
insecure-$0kxja!oo5z=9r1c1=v6usv46y6zim1vg*tm!r29g*nj5%3^m&'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ["*"]

#Changes 1
CORS_ORIGIN_ALLOW_ALL = True


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
```

```
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'API',
    "rest_framework",

]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',

   #changes 2
    'corsheaders.middleware.CorsMiddleware',
    'django.middleware.common.CommonMiddleware',
```

```
]

ROOT_URLCONF = 'drinks.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

```
WSGI_APPLICATION = 'drinks.wsgi.application'


# Database
# https://docs.djangoproject.com/en/3.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}


# Password validation
# https://docs.djangoproject.com/en/3.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
```

```
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/3.2/topics/i18n/
```

```
LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.2/howto/static-files/

STATIC_URL = '/static/'

# Default primary key field type
# https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

Learn React js

[start project] => npx create-react-app solomon-portfolio

Component

A components represent to the part of the application examples are header, sidebar, footer, main

Components are part of the user interface and they are reusable

Componet types are

stateless funtional component

example

```
function welcome(props){

return <h1>Hello {props.name}</h1>

}
```

stateful class component

```
class Welcome extends React.Component{

render(){

return <h1>Hello {this.props.name}</h1>

}

}
```

Functional Component

```
function Greet(){
    return <h1>Hello Learner</h1>

}

export default Greet
```

APP.JS

```
import Greet from "./components/greet"

const kofi = () =>{
  return (
<div>
<Greet/>
```

```
</div>

    )
}

export default kofi
```

CLASS COMPONENTS

Welcome.js

```
import React, {Component} from "react"

class Welcome extends Component{

    render(){
    return <h1>Hello Welcome </h1>
    }
}
```

```
export default Welcome
```

App.js

```
import Greet from "./components/greet"
import Welcome from "./components/Welcome.js"

const kofi = () =>{
  return (
<div>
<Greet/>
<Welcome/>
</div>

    )
}

export default kofi
```

JSX

JSX MEANS JAVASCRIPT XML AND IT HELPS TO MAKE THE CODE SIMPLER WITHOUT IT, THE CODE WILL BE COMPLEX USING REACT.CREATELEMENT

PROPS

Greet.js

```
function Greet(props){

    return <h1>Hello {props.name}</h1>

}

export default Greet
```

App.js

```
import Greet from "./components/greet"
import Welcome from "./components/Welcome.js"

const kofi = () =>{
  return (
<div>
<Greet name="Solomon"/>
<Greet name="Kofi"/>
<Greet name="Danso"/>
</div>

    )
}

export default kofi
```

FOR ADDITIONAL ATTRIBUTES

APP.js

```
import Greet from "./components/greet"
```

```
import Welcome from "./components/Welcome.js"

const kofi = () =>{
  return (
<div>
<Greet name="Solomon"/>
<Greet name="Kofi"/>
<Greet name="Danso"/>
<Greet name="Kwesi" age="40">
<button>Another Banger</button>
</Greet>
</div>

    )
}

export default kofi
```

Greet.js

```
function Greet(props){
    return(
<div>
<h1>Hello {props.name}, are you {props.age} years old?</h1>
{props.children}
</div>

)}

export default Greet
```

props is immutable and component defined by us must start with capital letter

Dynamic Values

```
const App = () =>{
    const title = "Welcome to dynamic values";
```

```
    const link = "http://192.168.43.186:8000/home/fullstack-hydrogen-drf/";
    return(
<div className="App">

<div className="content">

<h1>App Component </h1>
{title}<br/>
{2+5}<br/>
{Math.random()*19}<br/>

<a href={link}>Dynamic Notes </a>

</div>

</div>



        )
}
```

```
export default App
```

Multiple Components

APP.js

```
import Navbar from "./navbar"
import Home from "./home"

const App = () =>{
    return(
<div className="App">
<Navbar/>
<Home/>
</div>


        )
}
```

```
export default App
```

Home.js

```
const Home = () =>{
    const title = "Welcome to dynamic values";
    const link = "http://192.168.43.186:8000/home/fullstack-hydrogen-
drf/";

return(
<div className="home">
<div className="content">

<h1>App Component </h1>
{title}<br/>
{2+5}<br/>
{Math.random()*19}<br/>

<a href={link}>Dynamic Notes </a>

</div>
```

```
</div>

    )

}

export default Home
```

Navbar.js

```
const Navbar = () =>{
    return (
      <div className="navbar">
<h1>The Dojo Blog</h1>
<div className="links">
<a href="/" >Home</a>
<a href="/create" >New Blog</a>

</div>
      </div>
        )
```

```
}

export default Navbar
```

Adding Styles

index.js

```
import React from "react";
import ReactDOM from "react-dom";
import App from "./App"
import "./index.css"


ReactDOM.render(
<React.StrictMode>
<App/>
</React.StrictMode>,
document.getElementById('root')
  )
```

CSS IN A FILE

index.css

```
body{
    background-color: hsl(0, 0%, 0%);
    color: whitesmoke;
}

.links{
    display: flex;
    flex-direction: row;
    justify-content: space-between;
}
*{
    color: lightskyblue;
}
```

Inline Css

```
const Navbar = () =>{
```

```
    return (
      <div className="navbar">
<h1>The Dojo Blog</h1>
<div className="links">
<a href="/" >Home</a>
<a href="/create" style={{
    color:"whitesmoke",
    backgroundColor:"green",
    borderRadius:"20px"

}}>New Blog</a>

</div>
      </div>
        )
}

export default Navbar
```

Click Events and Functions

Functions

```
const handleClick = () =>{
console.log('Function without parameters');
}

const handleClickAgain = (name) =>{
console.log(name+' Function with parameters');
}


const Home = () =>{

return(
<div className="home">
<div className="content">

<h1>App Component </h1>

<button onClick={handleClick}>'Function without parameters'</button>
```

```
<button onClick={()=>{handleClickAgain("Solomon")}}>'Function with parameters'</button>

<button onClick={()=>{
console.log("This is an anonymous function")
}}>Anonymous Function</button>

</div>

</div>

    )

}

export default Home
```

EVENTS

```
const handleClick = (e) =>{
```

```
console.log('Function without parameters ',e);
}
```

WHEN WE OPEN THE CONSOLE WE WILL SEE THIS EVENTS

Object { _reactName: "onClick", _targetInst: null, type: "click", nativeEvent: click, target: button, currentTarget: null, eventPhase: 3, bubbles: true, cancelable: true, timeStamp: 3864, … }

home.js

```
const handleClick = (Q) =>{
console.log('Function without parameters ',Q);
}

const handleClickAgain = (name, e) =>{
console.log(' Function with parameters '+name,e.target);
}


const Home = () =>{
```

```
return(
<div className="home">
<div className="content">

<h1>App Component </h1>

<button onClick={handleClick}>'Function without parameters'</button>

<button onClick={()=>{handleClickAgain("Solomon")}}>'Function with
parameters'</button>

<button onClick={(E)=>{
console.log("This is an anonymous function WITH THIS EVENTS =>
",E.target)
}}>Anonymous Function</button>

</div>

</div>

    )
```

```
}

export default Home
```

USESTATE HOOK

Hooks start with the name **Use**

First of all we will create a variable that has an initial value set to a usestate function

to change the initial value, we will use the setstate in which we have already assigned to the usestate

```
const [name, setname] = useState("Solomon");
```

name will contain the initial value which is Solomon

setname will then change the name for us

Home.js

```
import {useState} from "react";

const Home = () =>{

//Use state is defined here
const [name, NewName] = useState("Solomon");
const [age, NewAge] = useState(22)

//Functions that will called when button is clicked
const handleClick = () =>{
NewName("Twinkle")
NewAge(21)
}


return(
```

```
<div className="home">
<div className="content">

<h1>App Component </h1>
<p>{name} is {age} years old</p>
<button onClick={handleClick}>Change name </button>

</div>

</div>

    )

}

export default Home
```

OUTPUTING LIST

```
import {useState} from "react";

const Home = () =>{

const [blogs,newblog] = useState([
  {title:"My first blog", body:"My first body", id:1},
  {title:"My second blog", body:"My second body", id:2},
  {title:"My third blog", body:"My third body", id:3},
  {title:"My fourth blog", body:"My fourth body", id:4},
  {title:"My fifth blog", body:"My fifth body", id:5},

    ])

return(
<div className="home">
{blogs.map((blog)=>(
<div className="blog_preview" key={blog.id}>
<h1>Title: {blog.title} </h1>
<h2>Body: {blog.body} </h2>
<br/>
```

```
</div>


    ))}

</div>

    )

}

export default Home
```

PROPS

Home.js

```
import BlockList from "./blocklist"
import {useState} from "react";
```

```
const Home = () =>{

const [blogs,newblog] = useState([
  {title:"My first blog", body:"My first body", id:1},
  {title:"My second blog", body:"My second body", id:2},
  {title:"My third blog", body:"My third body", id:3},
  {title:"My fourth blog", body:"My fourth body", id:4},
  {title:"My fifth blog", body:"My fifth body", id:5},

    ])

return(
<div>
{/*The blogs value is passed to Kofi as a variable*/}
<BlockList Kofi={blogs} title="All my blogs"/>

</div>



    )
```

```
}

export default Home



BookList.js

import {useState} from "react"
const BlockList = (props) =>{
const blogs = props.Kofi
const title = props.title
return(

<div className="home">
<h1>{title}</h1>
{
    blogs.map((blog) => (

<div key={blog.id}>
```

```
<h1>[{blog.id}] Title: {blog.title}</h1>
<h2>Body: {blog.body}</h2>

</div>

        ))
}
</div>

    )




}

export default BlockList
```

Reusable components

Using Filter

```
import BlockList from "./blocklist"
import {useState} from "react";

const Home = () =>{

const [blogs,newblog] = useState([
  {title:"My first blog", body:"My first body", id:1},
  {title:"My second blog", body:"My second body", id:2},
  {title:"My third blog", body:"My third body", id:3},
  {title:"My fourth blog", body:"My fourth body", id:4},
  {title:"My fifth blog", body:"My fifth body", id:5},

    ])

return(
<div>
```

```
{/*The blogs value is passed to Kofi as a variable*/}
<BlockList Kofi = {blogs.filter((blog)=>blog.id===3)} title="Third
Blog"/>
</div>


    )

}

export default Home
```

FUNCTIONS AS PROPS

identity parameter is a counting number first blog identity is 1 and fifth blog identity is 5

setblog will update the state using the id defined in the blog example is

const filteredBlogs = blogs.filter(blog => blog.id==5)
setblog(filteredBlogs)
Only the fifth blog will be stored us an updated state and the rest will be deleted

example 2
const filteredBlogs = blogs.filter(blog => blog.id==identity)
setblog(filteredBlogs)
using this example, if i click on delete for the third blog, only third blog will be saved because the identity condition will be set to 3, the filter will pic the blog with id = 3, which happens to be the third blog

example 3
const filteredBlogs = blogs.filter(blog => blog.id!==identity)
setblog(filteredBlogs)

when i click on 2 it will be deleted because the filter will only work for those whoose blog.id is not equal to their current identity
1,3,4,5 will be saved because i have not clicked on them so their identity is equal to null but as for 2, as at the tme i clicked on it, the identity is 2 and the blog.id is already defined as 2

home.js

```
import BlockList from "./blocklist"
import {useState} from "react";

const Home = () =>{

const [blogs,setblog] = useState([
  {title:"My first blog", body:"My first body", id:1},
  {title:"My second blog", body:"My second body", id:2},
```

```
  {title:"My third blog", body:"My third body", id:3},
  {title:"My fourth blog", body:"My fourth body", id:4},
  {title:"My fifth blog", body:"My fifth body", id:5},

    ])
const handledelete = (identity) =>{
    console.log(identity)
const filteredBlogs = blogs.filter(blog => blog.id!==identity)
setblog(filteredBlogs)

}

return(
<div>
{/*The blogs value is passed to Kofi as a variable*/}
<BlockList Kofi={blogs} title="All my blogs" handledelete=
{handledelete}/>
</div>
```

```
    )

}

export default Home
```

USEEFFECTS

```
import BlockList from "./blocklist"
import {useState,useEffect} from "react";

const Home = () =>{

const [blogs,setblog] = useState([
  {title:"My first blog", body:"My first body", id:1},
  {title:"My second blog", body:"My second body", id:2},
  {title:"My third blog", body:"My third body", id:3},
  {title:"My fourth blog", body:"My fourth body", id:4},
  {title:"My fifth blog", body:"My fifth body", id:5},
```

```
    ])
const handledelete = (identity) =>{
    console.log(identity)
const filteredBlogs = blogs.filter(blog => blog.id!==identity)
setblog(filteredBlogs)

}

useEffect(()=>{
  console.log("Use effects can be used to fetch data and this function run
every render.")
  console.log("For every changes in the applicatin, useEffect will run")
  console.log("We can also access the state in the useEffect")
  console.log(blogs)
  console.log("It can be uesd to run any code that needs to run first")
})

return(
<div>
{/*The blogs value is passed to Kofi as a variable*/}
```

```
<BlockList Kofi={blogs} title="All my blogs" handledelete=
{handledelete}/>
</div>


    )

}

export default Home
```

UseEffect dependencies

If we dont want the useEffect to run after every update we can pass a dependency in it array.

We can pass an empty array, and this will tell useEffect to run only after the first render

```
useEffect(()=>{
```

```
  console.log("Use effects can be used to fetch data and this function run every render.")
  },[])
```

Another example is, use effect can also be rendered only if a certain state changes

```
import BlockList from "./blocklist"
import {useState,useEffect} from "react";

const Home = () =>{

const [blogs,setblog] = useState([
  {title:"My first blog", body:"My first body", id:1},
  {title:"My second blog", body:"My second body", id:2},
  {title:"My third blog", body:"My third body", id:3},
  {title:"My fourth blog", body:"My fourth body", id:4},
  {title:"My fifth blog", body:"My fifth body", id:5},

    ])
```

```
const [name,setname] = useState("Solomon")
const handledelete = (identity) =>{
    console.log(identity)
const filteredBlogs = blogs.filter(blog => blog.id!==identity)
setblog(filteredBlogs)

}

useEffect(()=>{
  console.log(name)
  },[name])

return(
<div>
{/*The blogs value is passed to Kofi as a variable*/}
{name}
<BlockList Kofi={blogs} title="All my blogs" handledelete=
{handledelete}/>
<button onClick={()=>{setname("Twinkle")}}>Change Name</button>
</div>
```

```
    )

}

export default Home
```

Fetch Request Using UseEffect

We will put an empty array at the useEffect because we want the component to run ones if the page loads

```
useEffect(()=>{
  fetch("http://127.0.0.1:8002/drinks/")
  .then((res)=>{return res.json()})
  .then((data)=>{setblog(data)})
},[])
```

BLOCKLIST.JS

```
import {useState} from "react"
const BlockList = (props) =>{
const blogs = props.Kofi
const title = props.title


return(

<div className="home">
<h1>{title}</h1>
{
    blogs.map((blog) => (

<div key={blog.id}>

<h1>[{blog.id}] Title: {blog.name}</h1>
<h2>Body: {blog.description}</h2>
</div>
```

```
        ))
}
</div>

    )




}

export default BlockList



HOME.JS

import BlockList from "./blocklist"
import {useState,useEffect} from "react";

const Home = () =>{
```

```
const [blogs,setblog] = useState(null)



useEffect(()=>{
  fetch("http://127.0.0.1:8002/drinks/")
  .then((res)=>{return res.json()})
  .then((data)=>{setblog(data)})

},[])

return(
<div>
{/*The blogs value is passed to Kofi as a variable*/}
{blogs && <BlockList Kofi={blogs} title="All my blogs"/>
}</div>


    )

}
```

export default Home

CONDITIONAL TEMPLATING

Home.js

```
import BlockList from "./blocklist"
import {useState,useEffect} from "react";

const Home = () =>{

const [blogs,setblog] = useState(null)
const [isPending, setIsPending] = useState(true)


useEffect(()=>{
  setTimeout(()=>{
    fetch("http://127.0.0.1:8002/drinks/")
  .then((res)=>{return res.json()})
  .then((data)=>{
```

```
    setblog(data)
    setIsPending(false)
  })
},2000)


},[])

return(
<div>
{/*Only if the isPending is true before you can load the function at the
right hand side */}
{isPending && <div>Loading Awesome Project.....</div>}
{blogs && <BlockList Kofi={blogs} title="All my blogs"/>}

</div>


    )

}
```

```
export default Home



Blocklist.js

import {useState} from "react"
const BlockList = (props) =>{
const blogs = props.Kofi
const title = props.title


return(

<div className="home">
<h1>{title}</h1>
{
    blogs.map((blog) => (

<div key={blog.id}>
```

```
<h1>[{blog.id}] Title: {blog.name}</h1>
<h2>Body: {blog.description}</h2>
</div>

      ))
}
</div>

   )




}

export default BlockList
```

HANDLING ERRORS

HOME.JS

```
import BlockList from "./blocklist"
import {useState,useEffect} from "react";

const Home = () =>{

const [blogs,setblog] = useState(null)
const [isPending, setIsPending] = useState(true)
const [error, setError] = useState(null)

useEffect(()=>{
  setTimeout(()=>{
    fetch("http://127.0.0.1:8002/drinks/")
  .then((responsefromserver)=>{

    if(!responsefromserver.ok){
      throw Error("Could not fetch data for the result. Please check your
internet connection")
    }

    return responsefromserver.json()})
```

```
  .then((data)=>{
    setblog(data)
    setIsPending(false)
    setError(null)
  })
  .catch((error)=>{
    setError(error.message)
    setIsPending(false)
  })

},2000)


},[])

return(
<div>


{/*Only if the isPending is true before you can load the function at the
```

```
right hand side */}
{isPending && <div>Loading Awesome Project.....</div>}
{error && <div>{error}</div>}
{blogs && <BlockList Kofi={blogs} title="All my blogs"/>}

</div>


    )

}

export default Home
```

CREATING CUSTOM HOOKS

To create a hook start with **use** keyword

useFetch.js

```
import {useState,useEffect} from "react";

const useFetch = (url) =>{

const [data,setData] = useState(null)
const [isPending, setIsPending] = useState(true)
const [error, setError] = useState(null)


    useEffect(()=>{

    fetch(url)
  .then((responsefromserver)=>{

    if(!responsefromserver.ok){
      throw Error("Could not fetch data for the result. Please check your
internet connection")
    }
    return responsefromserver.json()})
  .then((datafromserver)=>{
```

```
    setData(datafromserver)
    setIsPending(false)
    setError(null)
  })
  .catch((err)=>{
    setError(err.message)
    setIsPending(false)
  })



},[url])
    return {data,isPending,error}

}

export default useFetch



Home.js
```

```
import BlockList from "./blocklist"
import {useState,useEffect} from "react";
import useFetch from "./useFetch"


const Home = () =>{
const {data:blogs,isPending,error} = useFetch("http://127.0.0.1:8002
/drinks/")

return(
<div>


{/*Only if the isPending is true before you can load the function at the
right hand side */}
{isPending && <div>Loading Awesome Project.....</div>}
{error && <div>{error}</div>}
{blogs && <BlockList Kofi={blogs} title="All my blogs"/>}

</div>
```

```
    )
}

export default Home
```

THE REACT ROUTER

Install the router using **npm install react-router-dom@5**

App.js

```
import Navbar from "./navbar"
import Home from "./home"
import {BrowserRouter as Router, Route, Switch} from "react-router-dom"
import Create from "./create"
```

```
const App = () =>{
    return(
        <Router>

<div className="App">
<Navbar/>
<Switch>

<Route exact path="/"> <Home/> </Route>
<Route path="/create"> <Create/> </Route>

</Switch>
</div>

        </Router>


        )
}

export default App
```

Navbar.js

```
import {Link} from "react-router-dom"

const Navbar = () =>{
    return (
      <div className="navbar">
<h1>Learn React Blog</h1>
<div className="links">
<Link to="/" >Home</Link>

<Link to="/create" style={{
    color:"whitesmoke",
    backgroundColor:"green",
    borderRadius:"20px"

}}>New Blog</Link>

</div>
      </div>
        )
```

```
}

export default Navbar
```

Create.js

```
import React from 'react'

const Create = () => {
    return (
        <div>
            <h2>Add a New title</h2>

        </div>
    )
}

export default Create
```

useEffect Cleanup(not very useful)

useFetch.js

```
import {useState,useEffect} from "react";

const useFetch = (url) =>{

const [data,setData] = useState(null)
const [isPending, setIsPending] = useState(true)
const [error, setError] = useState(null)


    useEffect(()=>{
    const abortcontstant = new AbortController()

    fetch(url,{signal:abortcontstant.signal})
  .then((responsefromserver)=>{

    if(!responsefromserver.ok){
      throw Error("Could not fetch data for the result. Please check your
internet connection")
```

```
  }
  return responsefromserver.json()})
.then((datafromserver)=>{
  setData(datafromserver)
  setIsPending(false)
  setError(null)
})
.catch((err)=>{
  if (err.name === "AbortError"){
  console.log("Fetch Aborted")
  }
  else{
  setError(err.message)
  setIsPending(false)
  }

})
return ()=>abortcontstant.abort()
```

```
},[url])
    return {data,isPending,error}

}

export default useFetch
```

ROUTE PARAMETERS(Blog Details)

APP.JS

```
import Navbar from "./navbar"
import Home from "./home"
import {BrowserRouter as Router, Route, Switch} from "react-router-
dom"
import Create from "./create"
import BlogDetails from "./BlogDetails"

const App = () =>{
```

```
    return(
        <Router>

<div className="App">
<Navbar/>
<Switch>

<Route exact path="/"> <Home/> </Route>
<Route path="/create"> <Create/> </Route>

<Route path="/blogs/:my_unique_id_from_the_database"><BlogDetails
/></Route>
</Switch>
</div>

        </Router>


        )
}
```

```
export default App
```

BLOCKLIST.JS

```
import {useState} from "react"
import {Link} from "react-router-dom"

const BlockList = (props) =>{
const blogs = props.Kofi
const title = props.title


return(

<div className="home">
<h1>{title}</h1>
{
    blogs.map((blog) => (
```

```
<div key={blog.id}>
{/*At this point, I have access to the id of the blog using the id assigned by the database and i will append this id to the link */}
<Link to={`/blogs/${blog.id}`}>
{/*blog.id is already predefined by the backend*/}

<h1>{blog.name}</h1>

</Link>

</div>

      ))
}
</div>

   )
```

```
}

export default BlockList



BLOGDETAILS

import {useParams} from "react-router-dom"
import useFetch from "./useFetch.js"
import React from 'react'

const BlogDetails = () => {

const {my_unique_id_from_the_database} = useParams()
const {data,isPending,error} = useFetch("http://127.0.0.1:8002/drinks
/"+my_unique_id_from_the_database)
{/*
The useFetch will connect to it unique api url as if it is a new request
The request has my_unique_id_from_the_database attached to it which
was been accesible in the bloglist LINK <Link to={`/blogs/${blog.id}`}>
```

```
*/}


    return (
        <div className="BlogDetails">
    {isPending && <div>Loading ...</div>}
    {error && <div>{error}</div>}
    {data && (

    <article>
  <h1>{data.name}</h1>
    <h2>{data.description}</h2>
    </article>

        )}


        </div>
    )
}
```

```
export default BlogDetails
```

CONTROLLED INPUTS

```
import React from 'react'
import "./create.css"
import {useState} from "react"


const Create = () => {
    {/*The title is set to an empty string. As the letters inside the input field keeps changing, it is sent to the set state which will then update the dom */}
const [title, setTitle] = useState("")
const [body, setBody] = useState("")


    return (
```

```
<div>
    <h2>Add a New title</h2>
<form action="" className="myForm">

<label>Blog title</label>
<input
type="text"
required
value={title}
onChange={(e)=>setTitle(e.target.value)}
/>

<label>Blog Body</label>
<textarea
required
value={body}
onChange={(e)=>setBody(e.target.value)}
></textarea>

<button>Add blog</button>
```

```
        </form>
        {/*Beginning of the application, the title was an empty string, now it will be updated with the setTitle */}
        <p>Title: {title}</p>
        <p>Body: {body}</p>

        </div>
    )
}

export default Create
```

Submitting a Form

Create.js

```
import React from 'react'
import "./create.css"
import {useState} from "react"
```

```
const Create = () => {
    {/*The title is set to an empty string. As the letters inside the input field keeps changing, it is sent to the set state which will then update the dom */}
const [name, setName] = useState("")
const [description, setDescription] = useState("")

const handlesubmit = (event) =>{
event.preventDefault()
const blog = {name,description}
console.log(blog)

}

    return (
        <div>
            <h2>Add a New title</h2>
        <form action="" className="myForm" onSubmit={handlesubmit}>
```

```
        <label>Blog title</label>
        <input
        type="text"
        required
        value={name}
        onChange={(event)=>setName(event.target.value)}
        />

        <label>Blog Body</label>
       <textarea
       required
       value={description}
       onChange={(event)=>setDescription(event.target.value)}
       ></textarea>

       <button type="submit">Add blog</button>

        </form>
        {/*Beginning of the application, the title was an empty string, now it will be updated with the setTitle */}
```

```
        </div>
    )
}

export default Create
```

Make a post request

```
import React from 'react'
import "./create.css"
import {useState} from "react"


const Create = () => {
    {/*The title is set to an empty string. As the letters inside the input field keeps changing, it is sent to the set state which will then update the dom */}
```

```
const [name, setName] = useState("")
const [description, setDescription] = useState("")
const [beforesendingdata, sendingdataprocess] = useState(true)


const handlesubmit = (event) =>{
event.preventDefault()
const blog = {name,description}
sendingdataprocess(false)


fetch("http://127.0.0.1:8002/drinks/",{
    method: "POST",
    headers: {"Content-Type":"application/json"},
    body: JSON.stringify(blog)
}).then(()=>{
    sendingdataprocess(true)
    console.log("New Blog Added")

})
```

```
}

    return (
        <div>
            <h2>Add a New title</h2>
        <form action="" className="myForm" onSubmit={handlesubmit}>

        <label>Blog title</label>
        <input
        type="text"
        required
        value={name}
        onChange={(event)=>setName(event.target.value)}
        />

        <label>Blog Body</label>
       <textarea
       required
       value={description}
       onChange={(event)=>setDescription(event.target.value)}
```

```
        ></textarea>

        {beforesendingdata && <button type="submit">Add blog</button>}
        {!beforesendingdata && <button disabled type="submit">Adding
Blog .....</button>}

        </form>
        {/*Beginning of the application, the title was an empty string, now it
will be updated with the setTitle */}


        </div>
    )
}

export default Create
```

PROGRAMMATIC REDIRECTS

(useHistory)

```
import React from 'react'
import "./create.css"
import {useState} from "react"
import {useHistory} from "react-router-dom"


const Create = () => {
    {/*The title is set to an empty string. As the letters inside the input field keeps changing, it is sent to the set state which will then update the dom */}
const [name, setName] = useState("")
const [description, setDescription] = useState("")
const [beforesendingdata, sendingdataprocess] = useState(true)
const [successMessage, updateSuccessMessage] = useState(null)
const history = useHistory()


const handlesubmit = (event) =>{
```

```
event.preventDefault()
const blog = {name,description}
sendingdataprocess(false)


fetch("http://127.0.0.1:8002/drinks/",{
    method: "POST",
    headers: {"Content-Type":"application/json"},
    body: JSON.stringify(blog)
}).then(()=>{
    sendingdataprocess(true)
    updateSuccessMessage("Your Blogs Were Successfully Added To This Site, Hurray!!!")
    {/*history.go(-1), this function will send you tothe previous page  */}
    //history.go(-1)
    history.push('/')

})
```

```
}

    return (
        <div>
            <h2>Add a New title</h2>
            {successMessage && <div>{successMessage}</div>}
        <form action="" className="myForm" onSubmit={handlesubmit}>

        <label>Blog title</label>
        <input
        type="text"
        required
        value={name}
        onChange={(event)=>setName(event.target.value)}
        />

        <label>Blog Body</label>
       <textarea
       required
       value={description}
```

```
        onChange={(event)=>setDescription(event.target.value)}
        ></textarea>

        {beforesendingdata && <button type="submit">Add blog</button>}
        {!beforesendingdata && <button disabled type="submit">Adding
Blog .....</button>}
         </form>
         {/*Beginning of the application, the title was an empty string, now it
will be updated with the setTitle */}


         </div>
    )
}

export default Create
```

Delete Blogs

Blockdetails.js

```
import {useParams,useHistory} from "react-router-dom"
import useFetch from "./useFetch.js"
import React from 'react'

const BlogDetails = () => {

const {my_unique_id_from_the_database} = useParams()
const {data,isPending,error} = useFetch("http://127.0.0.1:8002/drinks
/"+my_unique_id_from_the_database)
{/*
The useFetch will connect to it unique api url as if it is a new request
The request has my_unique_id_from_the_database attached to it which
was been accesible in the bloglist LINK <Link to={`/blogs/${blog.id}`}>

*/}
const history = useHistory()

const handleDelete = () =>{
    fetch("http://127.0.0.1:8002/drinks
```

```
/"+my_unique_id_from_the_database,{
        method:"DELETE",

    }).then(()=>{
   history.push("/")
    })

}

    return (
        <div className="BlogDetails">
    {isPending && <div>Loading ...</div>}
    {error && <div>{error}</div>}
    {data && (

    <article>
  <h1>{data.name}</h1>
    <h2>{data.description}</h2>
    <button onClick={handleDelete}>Delete</button>
    </article>
```

```
        )}


        </div>
    )
}

export default BlogDetails
```

404 PAGES NOT FOUND

notfound.js

```
import React from 'react'
import {Link} from "react-router-dom"

const NotFound = () => {
    return (
```

```
        <div className="not found">
        <h2>Sorry </h2>
        <p>The page you are requesting cannot be found on the server</p>
        <Link to="/">Back to the HomePage ....</Link>
        </div>
    )
}

export default NotFound
```

App.js

```
import Navbar from "./navbar"
import Home from "./home"
import {BrowserRouter as Router, Route, Switch} from "react-router-dom"
import Create from "./create"
import BlogDetails from "./BlogDetails"
import NotFound from "./notfound"
```

```
const App = () =>{
    return(
        <Router>

<div className="App">
<Navbar/>
<Switch>

<Route exact path="/"> <Home/> </Route>
<Route path="/create"> <Create/> </Route>
<Route path="/blogs/:my_unique_id_from_the_database"><BlogDetails
/></Route>
<Route path="*"><NotFound/></Route>

</Switch>
</div>

        </Router>
```

```
        )
}

export default App
```

6 days, 6 hours ago