

Nama: Zaky Mahfudz Pasha

NIM: 1301194315

FIF-43-06

To Do List_1301194315
delectus aut autem Not Completed
quis ut nam facilis et officia qui Not Completed
fugiat veniam minus Not Completed
et porro tempora Completed

Code:

```
// By: Zaky Mahfudz Pasha
// FIF-43-06
import 'dart:convert';

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

```

class Todo {
    final int userId;
    final int id;
    final String title;
    final bool completed;
    const Todo({
        required this.userId,
        required this.id,
        required this.title,
        required this.completed,
    });

    factory Todo.fromJson(Map<String, dynamic> json) {
        return Todo(
            userId: json['userId'],
            id: json['id'],
            title: json['title'],
            completed: json['completed']);
    }
}

Future<List<Todo>> getData() async {
    final response =
        await http.get(Uri.parse('https://jsonplaceholder.typicode.com/todos'));
    if (response.statusCode == 200) {
        List jsonResponse = json.decode(response.body);
        return jsonResponse.map((e) => Todo.fromJson(e)).toList();
    } else {
        throw Exception('Failed to load data');
    }
}

void main() => runApp(const MyApp());

class MyApp extends StatefulWidget {
    const MyApp({Key? key}) : super(key: key);

    @override
    // ignore: library_private_types_in_public_api
    _MyAppState createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
    late Future<List<Todo>> futureTodo;
}

```

```

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

@override
Widget build(BuildContext context) {
  return MaterialApp(
    debugShowCheckedModeBanner: false,
    title: 'Fetch Data Example',
    theme: ThemeData(
      primarySwatch: Colors.blue,
    ),
    home: Scaffold(
      appBar: AppBar(
        title: const Center(
          child: Text('To Do List_1301194315'),
        ),
      ),
      body: Center(
        child: FutureBuilder<List<Todo>>(
          future: futureTodo,
          builder: (context, snapshot) {
            if (snapshot.hasData) {
              return TodosScreen(todos: snapshot.data!);
            } else if (snapshot.hasError) {
              return Text('${snapshot.error}');
            }
            // By default, show a loading spinner.
            return const CircularProgressIndicator();
          },
        ),
      ),
    ),
  );
}

class TodosScreen extends StatelessWidget {
  const TodosScreen({Key? key, required this.todos});

  final List<Todo> todos;

  @override

```

```

Widget build(BuildContext context) {
  return ListView.builder(
    itemCount: todos.length,
    itemBuilder: (context, index) {
      return GestureDetector(
        child: Container(
          height: 150,
          width: 150,
          margin: const EdgeInsets.all(15),
          decoration:
            BoxDecoration(border: Border.all(color: Colors.blue)),
        child: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text(todos[index].title),
              Text(todos[index].completed
                ? "Completed"
                : "Not Completed")
            ],
          ),
        )),
    onTap: () {
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: ((context) =>
            DetailScreen(todo: todos[index]))));
    },
  );
});
}
}

```

```

class DetailScreen extends StatelessWidget {
  const DetailScreen({Key? key, required this.todo});
  final Todo todo;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('To Do List Detail'),
      ),
      body: Container(
        margin: const EdgeInsets.all(15),

```

```
width: 500,
height: 700,
decoration: BoxDecoration(border: Border.all(color: Colors.blue)),
child: Container(
  margin: const EdgeInsets.all(20),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    children: [
      Text('User ID : ${todo.userId}'),
      Text('ID : ${todo.id}'),
      Text('Title : ${todo.title}'),
      Text(todo.completed ? '(Completed)' : '(Not Completed)')
    ],
  ),
)),
);
}
```