

EX-10

Mini Projects

Aim

To develop mini projects involving flutter multi-platform.

Definitions

Flutter

Flutter is not a programming language. It's a software development kit (SDK) with prewritten code, consisting of ready-to-use and customizable widgets, as well as libraries, tools, and documentation that together serve to build cross-platform apps.

Flutter plugin

A Flutter plugin is a special kind of package that enables Flutter apps to interact with platform-specific APIs (iOS, Android, web, desktop). Plugins can include Dart code, but crucially, they also contain platform-specific implementation code written in Kotlin/Java for Android and Swift/Objective-C for iOS.

Dart plugin

The Dart plugin adds Dart support to IntelliJ Platform-based IDEs developed by JetBrains. These IDEs provide features unique to specific development technologies. The IDEs recommended for Dart and Flutter development include: IntelliJ IDEA which specializes in JVM-based language development.

Flutter SDK

Flutter is Google's free, open-source software development kit (SDK) for cross-platform mobile application development. Using a single platform-agnostic codebase, Flutter helps developers build high-performance, scalable applications with attractive and functional user interfaces for Android or iOS.

Todo Application

A to-do list is a structured list of tasks or activities that a person intends to accomplish, often used as a tool for planning, organization, and task management. It's a way to visually represent and track tasks, facilitating prioritization and progress tracking.

Stopwatch Application

A stopwatch application, often referred to as a stopwatch app, is a mobile application designed to measure elapsed time. It allows users to start, stop, and reset a timer, and typically includes features like lap recording for tracking time intervals during an activity.

A) Todo Application

Procedure

1. Open android studio
2. Click 'new flutter project'
3. Select 'flutter' at the left side of the window
4. Add 'flutter sdk' from the desired location
5. Click 'next' and specify the project name and select language 'java', check only Android, Web and Windows under platforms then click 'create'
6. Create a new dart file under 'lib' folder in the projects window (right click over lib folder -> new -> dart file -> specify the file name as 'todo' -> press 'enter')
7. Type the following codes in the todo.dart file

todo.dart

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

void main() {

  // Entry point of the app
  runApp(MyApp());
}

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(

      // Sets the home screen of the app
      home: HomeScreen(),

      // Removes debug banner
      debugShowCheckedModeBanner: false,

      // Sets the app theme color
      theme: ThemeData(primarySwatch: Colors.indigo),
    );
  }
}

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

```

@override
State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {

  // List to store tasks
  List<String> todoList = [];

  // Controller for text input
  final TextEditingController _controller = TextEditingController();

  // Index to track which task is being edited
  int updateIndex = -1;

  // Function to add a new task to the list
  addList(String task) {
    setState() {
      todoList.add(task);
      _controller.clear();
    });
  }

  // Function to update an existing task
  updateListItem(String task, int index) {
    setState() {
      todoList[index] = task;

      // Reset update index
      updateIndex = -1;
      _controller.clear();
    });
  }

  // Function to delete a task
  deleteItem(index) {
    setState() {
      todoList.removeAt(index);
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(

```

```

"Todo Application",
style: TextStyle(
  fontWeight: FontWeight.bold,
  fontSize: 25,
),
),

// Centers the app bar title
centerTitle: true,
backgroundColor: Colors.green,
foregroundColor: Colors.white,
),
body: Container(
  margin: EdgeInsets.all(10),
  child: Column(
    children: [
      Expanded(
        flex: 90,
        child: ListView.builder(

          // Number of tasks in the list
          itemCount: todoList.length,
          itemBuilder: (context, index) {
            return Card(
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(8),
              ),

              // Card background color
              color: Colors.green,
              child: Container(
                margin: EdgeInsets.only(left: 20),
                alignment: Alignment.center,
                padding: EdgeInsets.all(10),
                child: Row(
                  children: [
                    Expanded(
                      flex: 80,
                      child: Text(

                        // Display the task text
                        todoList[index],
                        style: TextStyle(
                          color: Colors.white,
                          fontWeight: FontWeight.bold,
                          fontSize: 20),

```

```

    ),
  ),

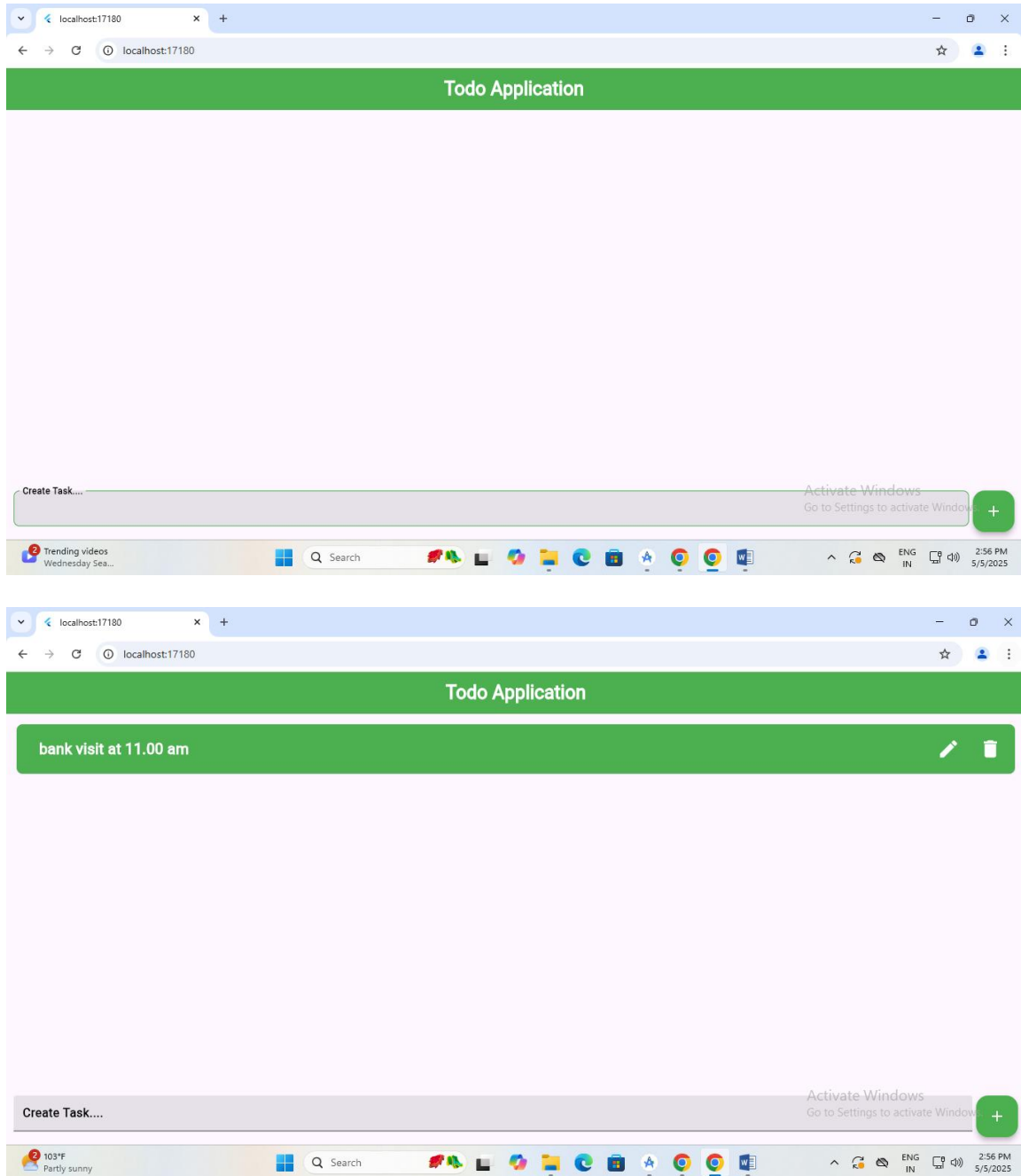
  // Edit button
  IconButton(
    onPressed: () {
      setState(() {
        _controller.clear();
        _controller.text = todoList[index];
        updateIndex = index;
      });
    },
    icon: Icon(
      Icons.edit,
      size: 30,
      color: Colors.white,
    ),
  ),
  SizedBox(width: 10),

  // Delete button
  IconButton(
    onPressed: () {
      deleteItem(index);
    },
    icon: Icon(
      Icons.delete,
      size: 30,
      color: Colors.white,
    ),
  ),
],
),
),
);
}),
),
Expanded(
  flex: 10,
  child: Row(
    children: [
      Expanded(
        flex: 70,
        child: SizedBox(
          height: 60,
          child: TextFormField(

```


8. Save the file todo.dart (click main menu -> saveall)
9. Select device as 'chrome(web)'
10. Click on run/debug configuration -> edit configurations -> specify dart file name (todo.dart) -> browse and set dart entrypoint as todo.dart -> click ok -> click run

Output



B) Stopwatch Application

Procedure

11. Open android studio
12. Click 'new flutter project'
13. Select 'flutter' at the left side of the window
14. Add 'flutter sdk' from the desired location
15. Click 'next' and specify the project name and select language 'java', check only Android, Web and Windows under platforms then click 'create'
16. Create a new dart file under 'lib' folder in the projects window (right click over lib folder -> new -> dart file -> specify the file name as 'stopwatch' -> press 'enter')
17. Type the following codes in the stopwatch.dart file

stopwatch.dart

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

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Stopwatch App',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: StopwatchScreen(),
    );
  }
}

class StopwatchScreen extends StatefulWidget {
  @override
  _StopwatchScreenState createState() => _StopwatchScreenState();
}

class _StopwatchScreenState extends State<StopwatchScreen> {
  late Timer _timer;
  int _start = 0;
  bool _isRunning = false;
```



```

// Start the timer
void _startStopwatch() {
  if (!_isRunning) {
    _timer = Timer.periodic(Duration(seconds: 1), (timer) {
      setState(() {
        _start++;
      });
    });
  } else {
    _timer.cancel();
  }

  setState(() {
    _isRunning = !_isRunning;
  });
}

// Reset the timer
void _resetStopwatch() {
  _timer.cancel();
  setState(() {
    _start = 0;
    _isRunning = false;
  });
}

// Format time in hours, minutes, seconds
String _formatTime(int timeInSeconds) {
  int hours = (timeInSeconds / 3600).floor();
  int minutes = ((timeInSeconds % 3600) / 60).floor();
  int seconds = timeInSeconds % 60;

  return '$hours:${minutes.toString().padLeft(2, '0')}:${seconds.toString().padLeft(2, '0')}';
}

@override
void dispose() {
  _timer.cancel(); // Cancel the timer when widget is disposed
  super.dispose();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(

```

```

        title: Text("Stopwatch"),
        centerTitle: true,
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          // Time display
          Text(
            _formatTime(_start),
            style: TextStyle(fontSize: 60, fontWeight: FontWeight.bold),
          ),
          SizedBox(height: 20),
          // Start/Pause button
          ElevatedButton(
            onPressed: _startStopwatch,
            child: Text(_isRunning ? "Pause" : "Start"),
            style: ElevatedButton.styleFrom(
              minimumSize: Size(150, 50),
              textStyle: TextStyle(fontSize: 20),
            ),
          ),
          SizedBox(height: 20),
          // Reset button
          ElevatedButton(
            onPressed: _resetStopwatch,
            child: Text("Reset"),
            style: ElevatedButton.styleFrom(
              minimumSize: Size(150, 50),
              textStyle: TextStyle(fontSize: 20),
            ),
          ),
        ],
      ),
    ),
  );
}
}

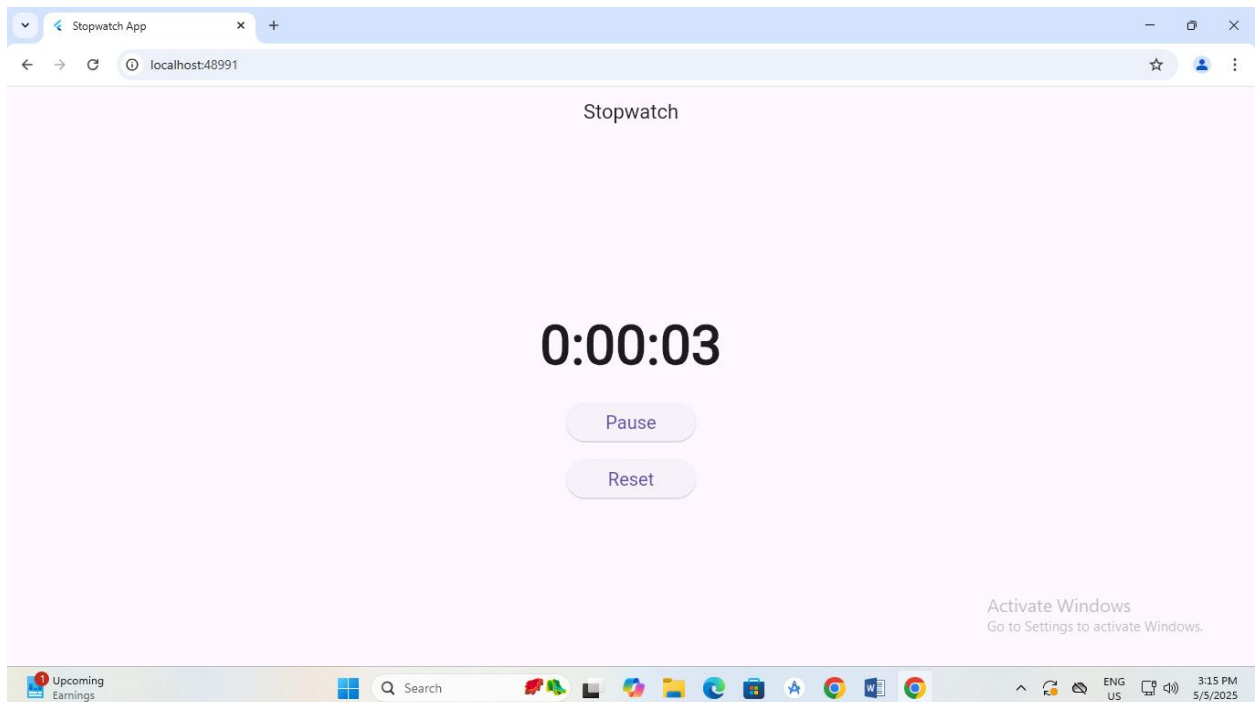
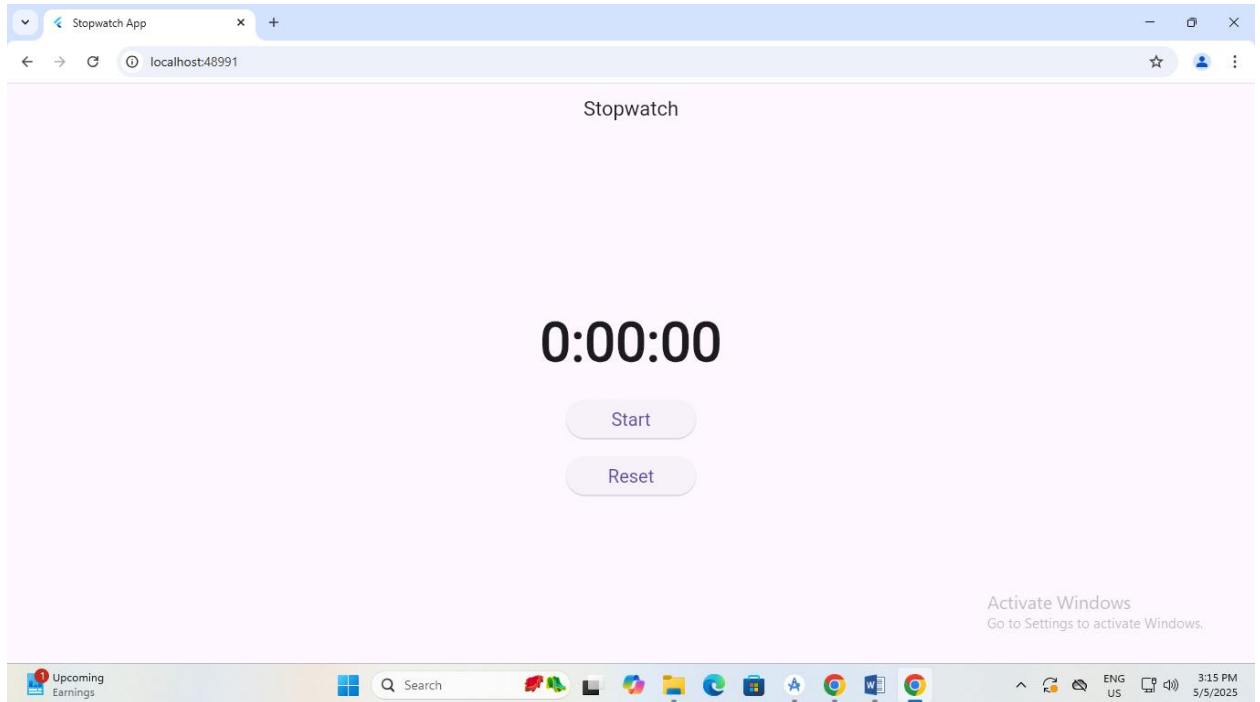
```

18. Save the file stopwatch.dart (click main menu -> saveall)

19. Select device as 'chrome(web)'

20. Click on run/debug configuration -> edit configurations -> specify dart file name (stopwatch.dart) -> browse and set dart entrypoint as stopwatch.dart -> click ok -> click run

Output



Result

Thus, mini projects involving flutter multi-platform have been developed.