RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 560054
(Autonomous Institute, Affiliated to VTU)

Department of Computer Science & Engineering

Internship Report

on

Mobile Application Development

INT411: Intra Institutional Internship

STUDENT NAME : RASHMITHA M and SAMPADA B

USN : 1MS22CS113 & 1MS22CS121

Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)
MSR Nagar, MSRIT Post, Bangalore-560054

October-November, 2023

RAMAIAH
Institute of Technology

RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 560054

(Autonomous Institute, Affiliated to VTU)

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that Mr./Ms. ____Rashmitha M & Sampada B a student of Bachelor of Engineering, bearing USN: _____1MS22CS113 & 1MS22CS121 has successfully completed, 20 Hours: from 25.10.2023 to 8.11.2023 Intra Institutional Internship in Mobile Application Development from the Department of Computer Science & Engineering, M S Ramaiah Institute of Technology, Bangalore.

SL No.	Component	Maximum Marks	Marks Obtained
1	Continuous Evaluation	50	
2	Presentation	20	
3	Report	30	
Total Marks		100	

Signature of the Student with Date **Signature of the Faculty Co-Ordinator**

Signature of Head of the Department

OVERVIEW OF INTERNSHIP ACTIVITIES

DATE	DAY	NAME OF THE TOPIC COMPLETED
	Monday	Introduction to Flutter, Dart Basics, Android SDK, VS code Getting Started with Flutter
	Tuesday	Building User Interfaces with Flutter Widgets and Layouts
	Wednesday	Styling and Theming in Flutter
	Thursday	Handling User Input and Navigation User Input and Gesture Handling
	Friday	Navigation and Routing Advanced Topics ➢ Working with animations and custom animation Integrating Flutter Apps with Firebase ➢ Incorporate Firebase Cloud Firestore into your Flutter apps. Implement authentication in your Flutter apps with the Firebase Auth package.
	Saturday	Building a project and deploying.

TABLE OF CONTENTS

Contents **Page No.1-9**

Overview of Stopwatch App

Overview of StopWatch App

Purpose: The Stopwatch App is designed to help users measure and track time with precision. It serves as a digital stopwatch that can be used for various purposes, including timing workouts, tracking event durations, or any activity where precise time measurement is essential.

Key Features:

1. **Start, Pause, and Reset:** Users can start and pause the stopwatch, allowing them to measure time for various activities. The reset button clears the elapsed time.
2. **Customization:** Users may have the option to customize the stopwatch's appearance and settings, including themes and fonts.
3. **Background Operation:** The stopwatch may continue running in the background, even when the app is not in the foreground, to allow users to time activities without keeping the app open.

User Interface: The app's user interface is designed to be intuitive and easy to use. It typically consists of the following elements:

- A large digital display showing the elapsed time.
- Start and pause buttons for controlling the stopwatch.
- A reset button to clear the elapsed time.
- Customization options for themes and appearance.

The Stopwatch App in Flutter leverages the power of the Flutter framework to create a responsive and visually appealing user interface. It uses Flutter's widgets and state management options to efficiently handle timekeeping and user interactions. Additionally, the app may include animations, sound effects, and other user-friendly features to enhance the overall user experience.

Overall, a Stopwatch App in Flutter is a valuable tool for users who need to measure and track time accurately in various contexts, from sports and fitness to cooking and event management. Its simplicity, ease of use, and potential for customization make it a handy utility for a wide range of users.

The primary purpose of a stopwatch app built with Flutter is to provide users with a convenient and accurate timekeeping tool. It allows users to measure and track time intervals with precision, making it useful for various activities and scenarios. Some common use cases include:

1.Fitness and Sports:
Users can time their workouts, runs, or other fitness activities to monitor their performance.

2.Cooking: Stopwatch apps are handy for timing cooking processes, ensuring that recipes are cooked to perfection.

3.Education and Presentations: Teachers, students, and presenters can use stopwatches to time quizzes, speeches, or presentations.

4.Laboratory Experiments: Scientists and researchers can accurately measure time intervals in experiments and data collection.

5.Sports Events: Coaches and referees can use stopwatches to time sporting events, races, or games.

6.Productivity and Time Management: Users can track their work or study sessions to manage time effectively.

IMPLEMENTATION OF STOPWATCH

Step 1: Set Up Your Flutter Project

1. Creating a new Flutter project or use an existing one.

Step 2: Create Stopwatch Screen

Create a new Dart file for your stopwatch screen, for example, `stopwatch_screen.dart`. This file will contain the code for your stopwatch screen. You can use the code provided in the previous response as a starting point.

Step 3: Implement Stopwatch Functionality

Inside your `stopwatch_screen.dart` file, implement the stopwatch functionality. This includes starting, pausing, and resetting the stopwatch. You can use a `Timer` or other timing mechanisms to update the time display. Use the `provider` package for state management if needed.

Step 4: Design the User Interface

Design the user interface of your stopwatch screen. Create a visually appealing layout with appropriate buttons, labels, and any other elements you want to include.

Step 5: Handle User Interactions

Implement the logic to handle user interactions. For example, you should start the timer when the "Start" button is pressed, pause it when "Pause" is pressed, and reset the timer when the "Reset" button is pressed.

Step 6: Display Elapsed Time

Display the elapsed time in a prominent location on the screen. Format the time as needed (hours, minutes, seconds).

Code of Main Module

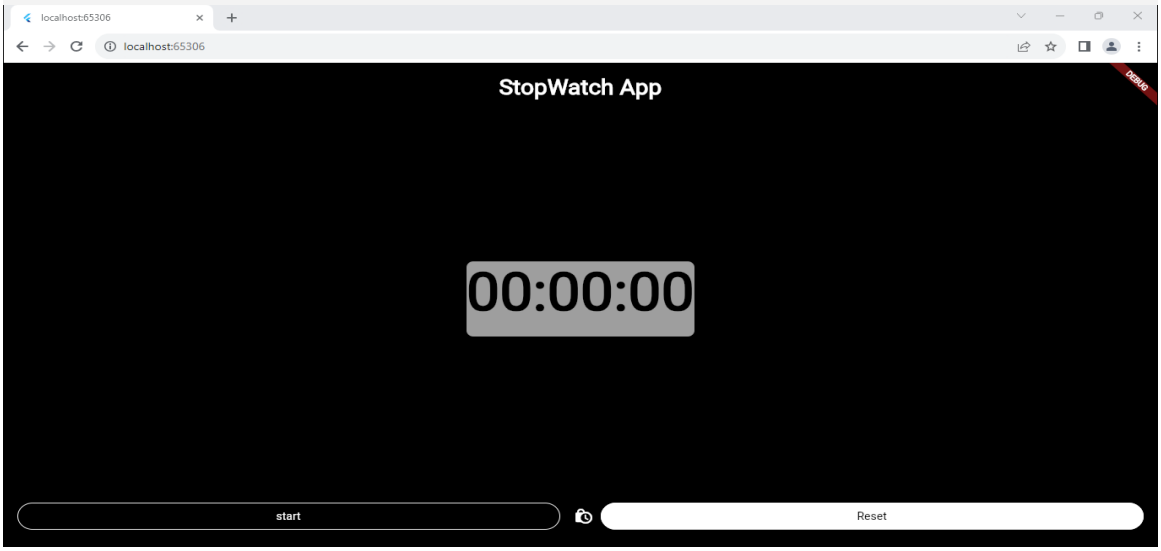
```dart
import 'package:flutter/material.dart';
import 'dart:async';
import 'package:flutter/cupertino.dart';
import 'package:flutter/widgets.dart';

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

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: HomeApp(),
    );
  }
}
class HomeApp extends StatefulWidget{
  const HomeApp({Key? key}) : super(key: key);

  @override
  _HomeAppState createState() => _HomeAppState();
}
class _HomeAppState extends State<HomeApp> {
  int seconds=0,minutes=0,hours=0;
  String digitseconds="00",digitminutes="00",digithours="00";
  Timer? timer;
  bool started=false;




  void stop(){
    timer!.cancel();
    setState(() {
      started=false;
    });
  }


  void reset(){
  timer!.cancel();
  setState(() {
```

```
    seconds=0;
    minutes=0;
    hours=0;
    digitseconds="00";
    digitminutes="00";
    digithours="00";
    started=false;
  });
  }
  void start(){
    started=true;
    timer=Timer.periodic(Duration(seconds: 1)
      , (timer) {int localSeconds=seconds+1;
      int locaMinutes=minutes;
      int localHours=hours;
      if(localSeconds>59){
        if(locaMinutes>59){
          localHours++;
          locaMinutes=0;
        }else{
          locaMinutes++;
          localSeconds=0;
        }
      }
      setState(() {
        seconds=localSeconds;
        minutes=locaMinutes;
        hours=localHours;
        digitseconds=(seconds>10)?"$seconds":"0$seconds";
        digithours=(hours>10)?"$hours":"0$hours";
        digitminutes=(minutes>10)?"$minutes":"0$minutes";
      }); });

  }



  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.black,
      body: SafeArea(
        child: Padding(
          padding: const EdgeInsets.all(16.0) ,
          child: Column(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            crossAxisAlignment: CrossAxisAlignment.center,
            children: [
              Center(child: Text("StopWatch App",style: TextStyle(color:
Colors.white,
              fontSize: 28.0,
```

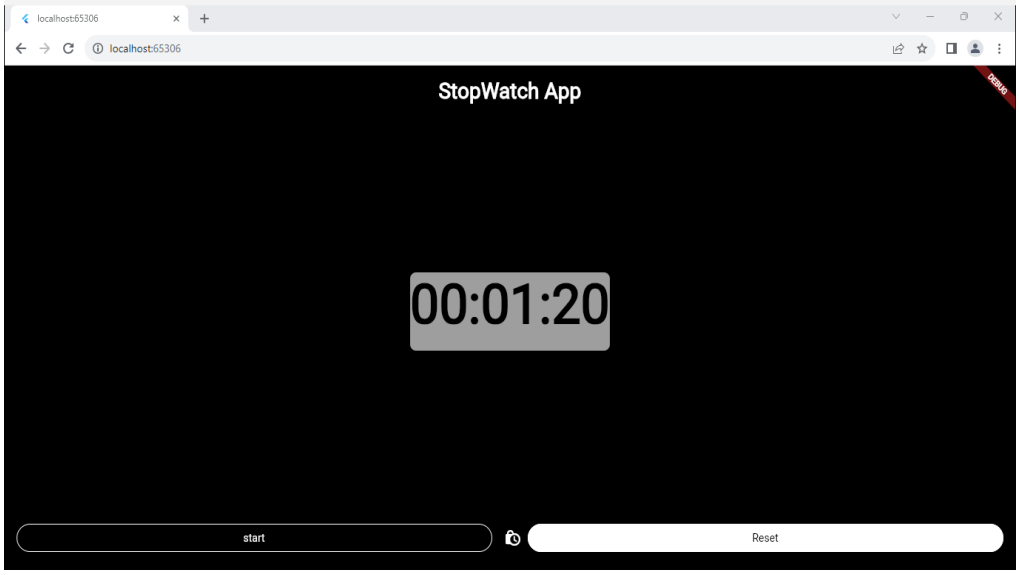
```
            fontWeight: FontWeight.bold,)),),
            SizedBox(height: 20.0
            ,),


            Container(
              height: 100.0,
              decoration: BoxDecoration(color: Colors.grey,
              borderRadius: BorderRadius.circular(8.0,),),
              child: Text("$digithours:$digitminutes:$digitseconds",
            style: TextStyle(color: Colors.black,fontSize: 70.0,
            fontWeight: FontWeight.w600,),),),),

          SizedBox(height: 20.0),
          Row(
           mainAxisAlignment: MainAxisAlignment.spaceBetween,
           children: [
            Expanded(child: RawMaterialButton(onPressed: () {
              (!started)?start():stop();
            },
            shape: StadiumBorder(side: BorderSide(color: Colors.white),),
            child: Text((!started)? "start": "pause",
            style: TextStyle(color: Colors.white))),),
           SizedBox(width: 8.0,),
           IconButton(
            color: Colors.white,
            onPressed: () {},icon: Icon(Icons.lock_clock),

          ),
          Expanded(child: RawMaterialButton(onPressed: () {
           reset();
          },
          fillColor: Colors.white,
            shape: StadiumBorder(side: BorderSide(color: Colors.white),),
            child: Text("Reset",style: TextStyle(color: Colors.black))),),],
          ),
           ],),),),
    );


  }
}
```

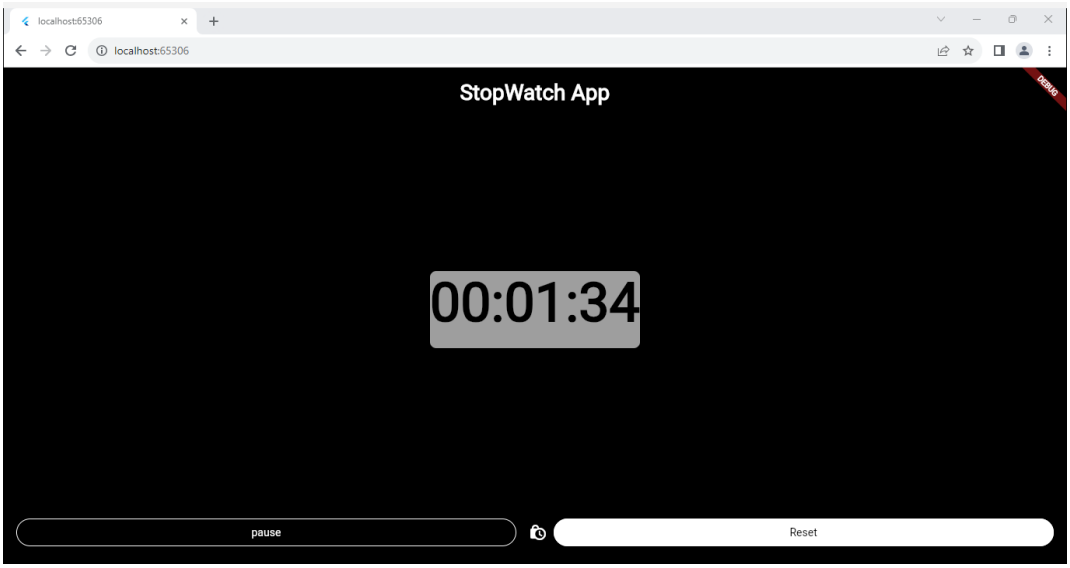
Result Snapshots

As you can see the image below the timer is set to zero.



Now we started the timer





As seen above the timer is paused.

Conclusion :-

StopWatch App:-

This app is used when time periods must be measured precisely and with a minimum of complications .
Laboratory experiments , gaming events and sporting events are some examples where this app can be implemented .

From this intra internship of mobile app development provided by our institution
i learnt few things which is needed to create an app:

- **Dart language which is required for flutter.**
- **The procedure to add images.**
- **The way we can add Icons, buttons, Toast, alert,**
- **Drawer , Navigation are some widgets which we learnt.**
- **Creating a firebase project , packages required for firebase.**
- **Atlast , we learnt how to deploy our project and the link which can be installed and the project can be used by many people.**