

MINI-PROJECT
(2020-2021)

Interval Timer App

FINAL REPORT



SUBMITTED BY:
AYURISH CHANDNA
(181500166)
AMAN KUMAR
(181500072)

SUBMITTED TO:
Dr. MANOJ VARSHNEY
(Asst. Professor)

Department of Computer Engineering and Applications
Institute of Engineering and Technology

Acknowledgment

I wish to express my profound gratitude and indebtedness to Dr. Manoj Varshney, Department of Computer Engineering and Applications, GLA University for introducing the present topic and for their inspiring guidance and valuable suggestion throughout the project work.

Last but not least, my sincere thanks to all our friends who have patiently extended all sorts of help for accomplishing this undertaking.

Ayurish Chandna(181500166)

Aman Kumar(181500072)

Dept. of Computer Engineering and Applications

GLA University



Department of computer Engineering and Applications
GLA University, Mathura

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,
Mathura – 281406**

CERTIFICATE

This is to certify that the Final Report entitled, “Interval Timer App” submitted by Mr./Ms. “Ayurish Chandna” and “Aman Kumar” to GLA University, Mathura, is a record of bonafide Project work carried out by them under the supervision of “Dr. Manoj Varshney” and guidance and is worthy of consideration for the award of the degree of Bachelor of Technology in Computer Science & Engineering of the Institute.

Signature of Candidate: Ayurish Chandna

Aman Kumar

Name of Candidate: Ayurish Chandna (181500166)

Aman Kumar (181500072)

Course: B.Tech (CS)

Year: 2020-2021

Semester: 6th

Index

Acknowledgment

Certificate

Project Description

Working code

Working Screenshots

Conclusion

Project Description

1. The Problem Statement

In our daily life we see that we are so busy in our daily activities that we have forgotten about our health which is the most important priority. During Covid we saw how people were suffering and so many people died also. This time made us think about the importance of health. How important it is to stay fit ? So keeping this in mind we came up with an idea to develop Fitness Interval Timer App.

2. Objectives of the project

The main objective of this app is for people who want to become consistent in their workout but they can't do due to lack of planning and laziness. People who are very conscious for their health we decided to make an app which will help them in maintaining their daily routine workout by setting a list of exercises and set a duration for each instructions. For example you have woken up early in the morning and decided to do some workout. You have planned a list of exercise that will be done on a particular day. This app will help you to follow those instructions by setting a timer for each exercises and tell you about ending of particular timer. It is important to have Interval training because it trains your heart to pump more blood to the muscles and it trains your muscles to extract that oxygen more efficiently, making all your other workouts easier to handle.

Flutter helps in building apps very efficiently. Flutter allows developers to build apps for both major platforms, including Android and iOS, with a single code base, therefore allowing start-ups to save a high cost on development services. Flutter has an efficient yet portable GPU rendering UI that allows it to work on multiple interfaces.

Why Flutter ?

Open-Source and Free to Use

Both Flutter and the underlying programming language, Dart are open-source and free to use. This empowers developers to view the documentation, post their

queries in the open developer forum, and eventually, learn and implement the code.

Hot Reload Functionality

One of the biggest reasons why Google Flutter is preferred by mobile application development agencies is its hot reloading functionality. The functionality makes it easier for anyone to make any change at the backend and view the difference at the front-end simultaneously. And this way, easily decide whether the change looks perfect with other elements of the application or not.

Lower Development Timeline and Cost

One of the prime benefits of focusing on Flutter development is that it renders opportunities to create native-like apps for both iOS and Android using a single codebase. Something that makes the cost to create a Flutter mobile app considerably low.

Support for Google Firebase

With the facility of Firebase integration, Flutter enables developers to add backend facility into the development environment.

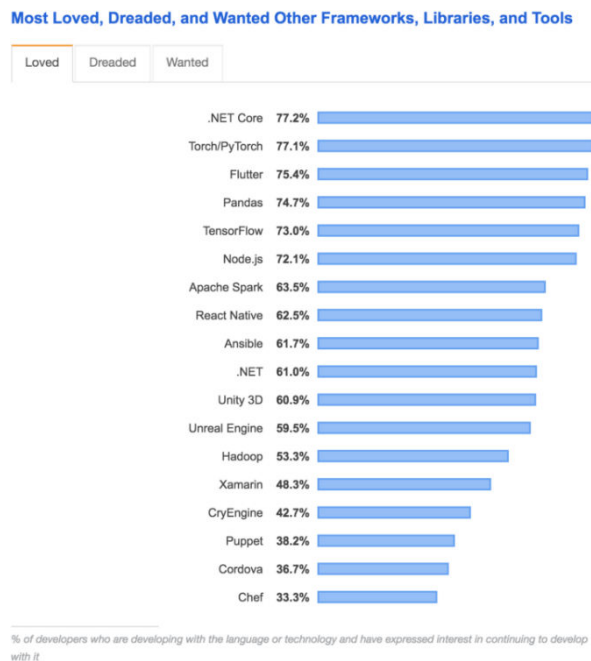
Flutter Testing Support

It is commonly known among the developers that Native Android tools support testing. Testing for native apps can be done using the command line of Gradle. One can add elements, integration and UI tests for Android apps. Numerous frameworks in Java and Kotlin are available for lower-level testing and also for UI testing. There are various levels of testing Android apps.

Flutter provides solid testing that allows quality analysts to write tests at various levels of such as unit, UI. One of the best features Flutter offers is Widget testing which runs UI test fast, equivalent to unit tests. If you are a beginner Flutter developer, you can refer to several levels of testing Android apps in this [shell](#). It runs in a separate process that can be run on real devices or emulators. Flutter has a separate package called Flutter Driver to drive the test in an easy manner.

According to Stack overflow :

Flutter ranks higher with 75.4% and React Native also made the cut with 62.5% among most loved frameworks.



3. Future Scope

Interval training enables you to complete an effective workout in less time than a standard cardiovascular workout. For example, you might complete a workout in about 15 to 20 minutes or less instead of 40 minutes. You'll improve your aerobic capacity.

So the future scope of fitness apps is very vast and it has an infinite number of benefits, especially for young people. So here are some of the benefits that fitness app provides -

- It can help you burn more calories. ...
- It may help you lose weight. ...
- It builds many muscle groups. ...
- It improves oxygen and blood flow. ...
- It can help lower blood pressure and blood sugar.
- Fewer Sick Days

4. Hardware & Software to be used

Hardware requirements –

1. System Software

- Operating System (Windows 10, Linux)
- Core i3 Processor (minimum)
- Minimum 8 GB Ram

Software Requirements

1. Application Software

- Android Studio
- VS Code

2. Programming Language

- Flutter
- Dart

5. What contribution would the project make and where?

This project will give a very crucial contribution to the field of health care. It will help people improve their health which will help them maintain strong immunity. It will boost the quality of life and will become confident and energetic. It will maximize cardiovascular benefits, so it can quickly increase stroke volume, making your heart stronger and more efficient. It also maximizes the other benefits of cardiovascular exercise, including decreasing your risk of both heart disease and high blood pressure

6. Literature Survey

This interval training method combines short, high intensity bursts of speed with slow recovery periods of mild activity or rest and helps to improve your performance incredibly. By varying the intensity of your workout, you'll reap the benefits of both aerobic and anaerobic training. Overtime, HIIT can help improve your speed, strength, and endurance.

Interval training (IT) has been used for many decades with the purpose of increasing performance and promoting health benefits while demanding a relatively small amount of time. IT can be defined as intermittent periods of intense exercise separated by periods of recovery and has been divided into high-intensity interval training (HIIT), sprint interval training (SIT), and repeated sprint training (RST). There is an disadvantage as well -

Interval training is tough on your body. If performed too often they increase the risk of overtraining.

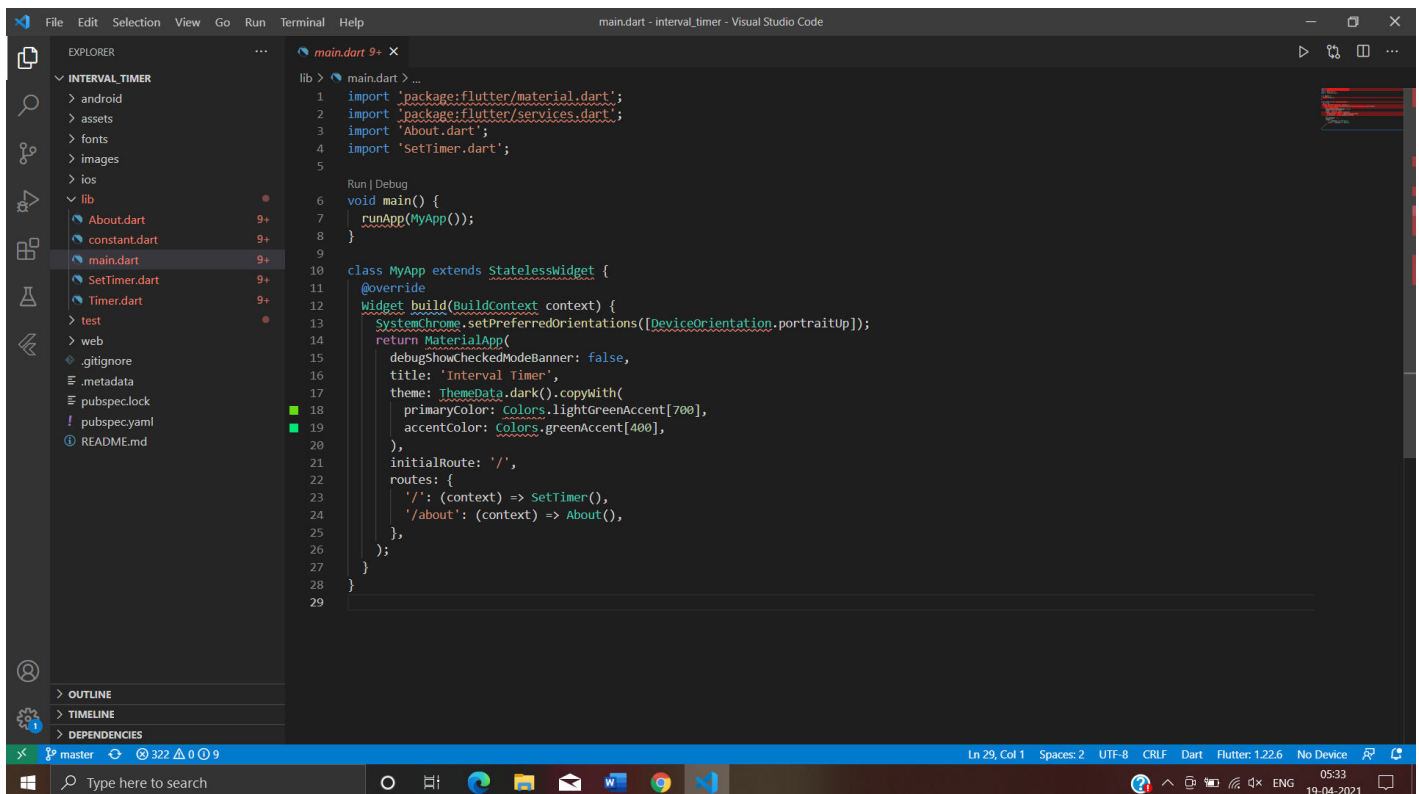
Implementation Details

Part1: Designing UI

- Main.dart
- SetTimer.dart
- Timer.dart
- Constant.dart
- About.dart

Working Code and Working App Screenshots

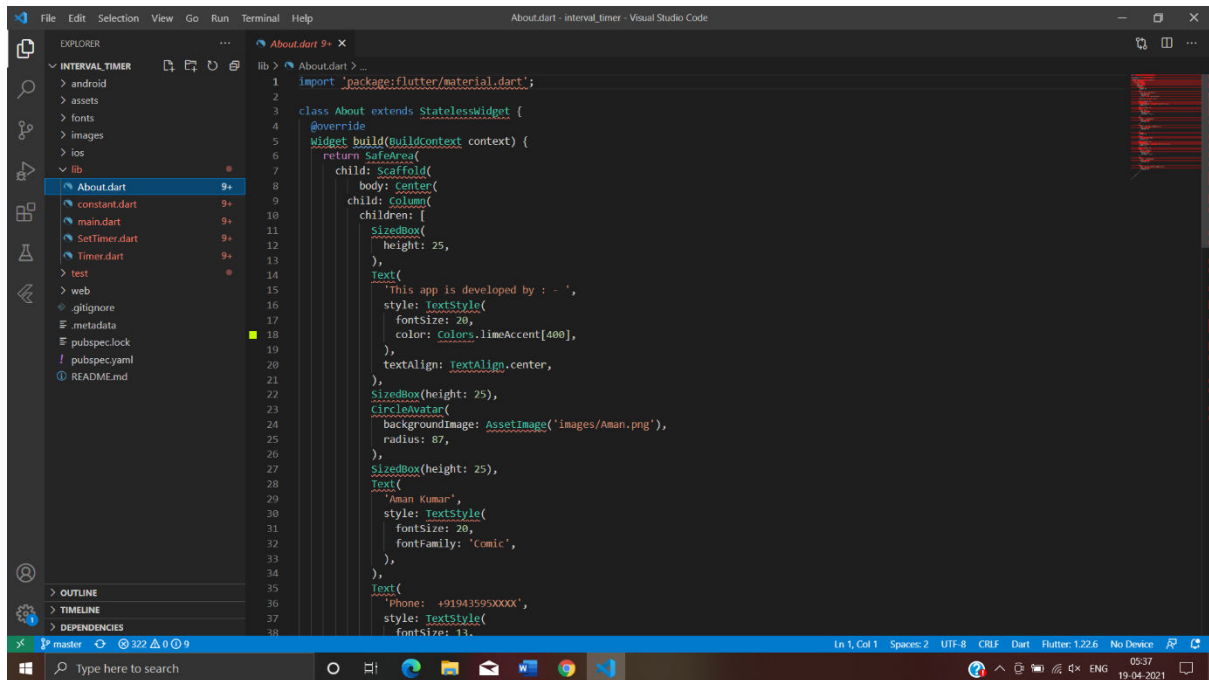
File name: main.dart



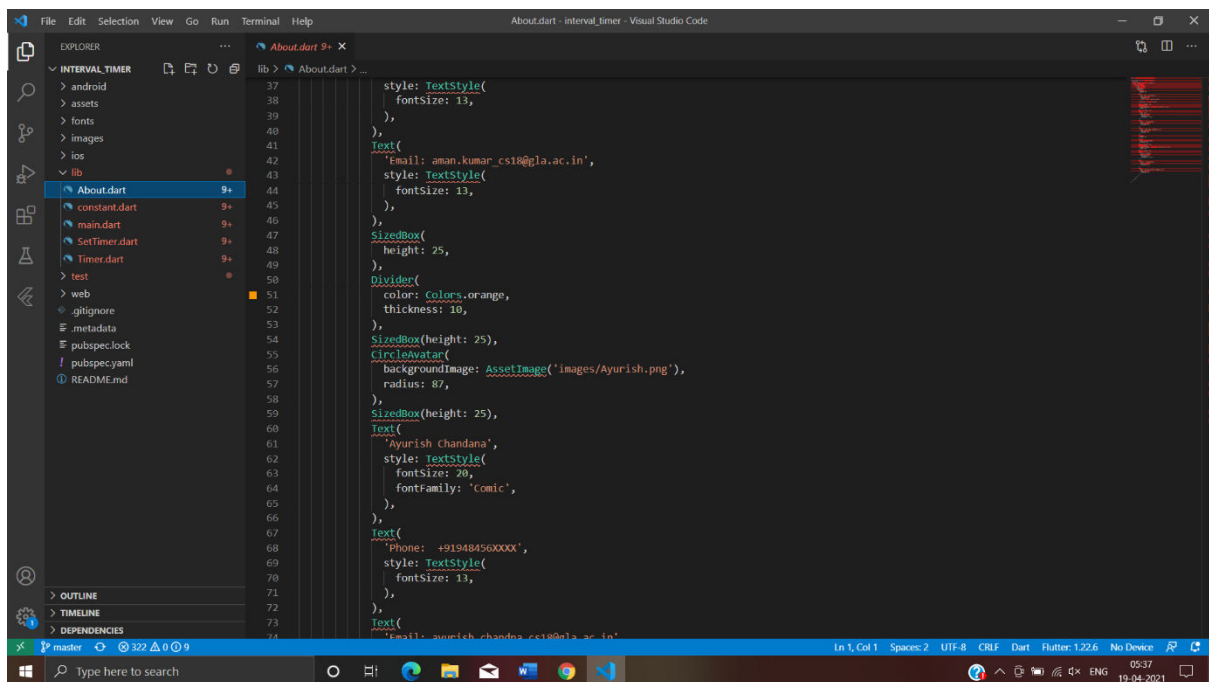
The screenshot displays the Visual Studio Code editor interface for a Flutter project named 'interval_timer'. The Explorer panel on the left shows the project structure, including the 'lib' directory with files like 'About.dart', 'constant.dart', 'main.dart', 'SetTimer.dart', and 'Timer.dart'. The 'main.dart' file is selected and its content is shown in the main editor area. The code defines a Flutter application with a dark theme, light green accents, and two routes: '/' leading to 'SetTimer()' and '/about' leading to 'About()'. The status bar at the bottom indicates the current file is 'main.dart' at line 29, column 1, with 2 spaces, UTF-8 encoding, and CRLF line endings. The Flutter version is 1.22.6, and the device is set to 'No Device'.

```
1 import 'package:flutter/material.dart';
2 import 'package:flutter/services.dart';
3 import 'About.dart';
4 import 'SetTimer.dart';
5
6 void main() {
7   runApp(MyApp());
8 }
9
10 class MyApp extends StatelessWidget {
11   @override
12   Widget build(BuildContext context) {
13     SystemChrome.setPreferredOrientations([DeviceOrientation.portraitUp]);
14     return MaterialApp(
15       debugShowCheckedModeBanner: false,
16       title: 'Interval Timer',
17       theme: ThemeData.dark().copyWith(
18         primaryColor: colors.lightGreenAccent[700],
19         accentColor: colors.greenAccent[400],
20       ),
21       initialRoute: '/',
22       routes: {
23         '/': (context) => SetTimer(),
24         '/about': (context) => About(),
25       },
26     );
27   }
28 }
29
```

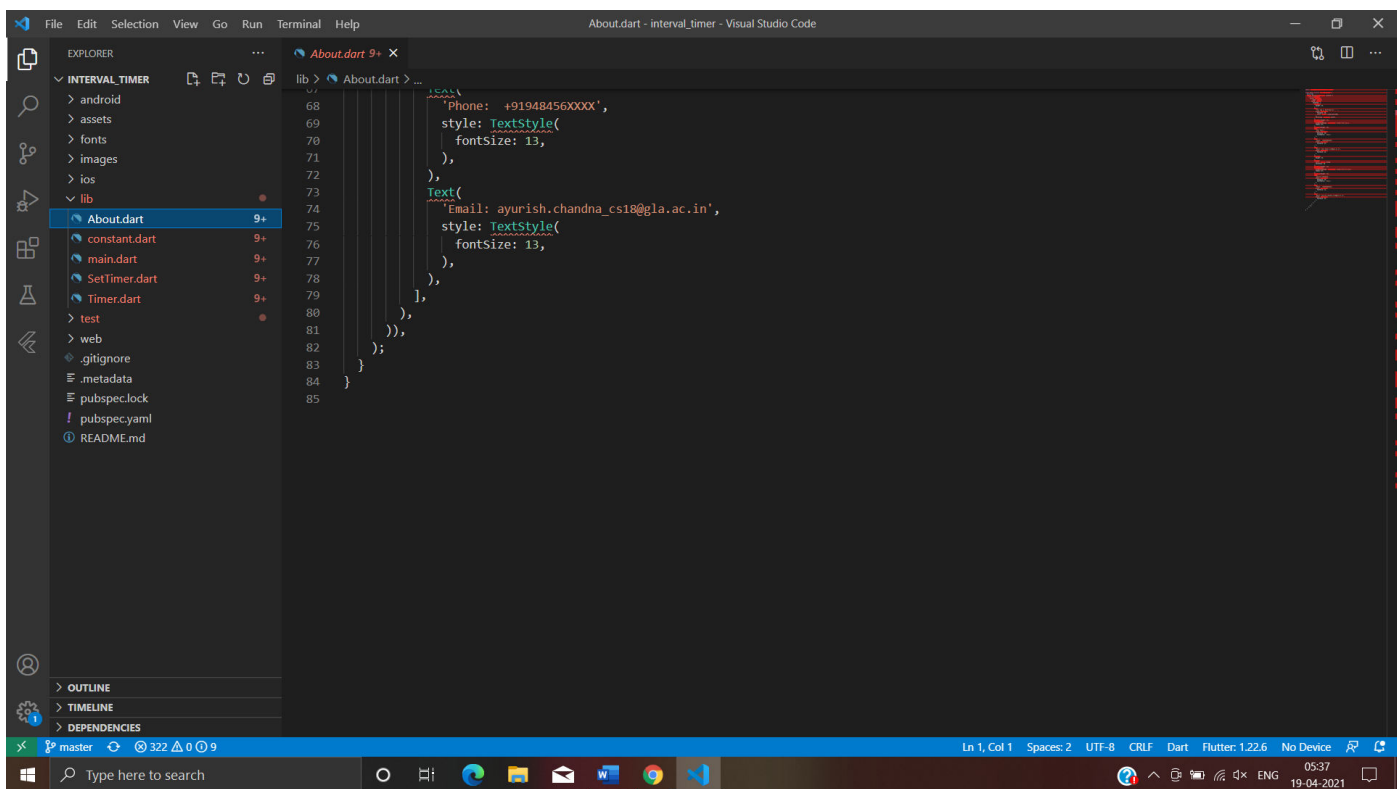
File name: About.dart



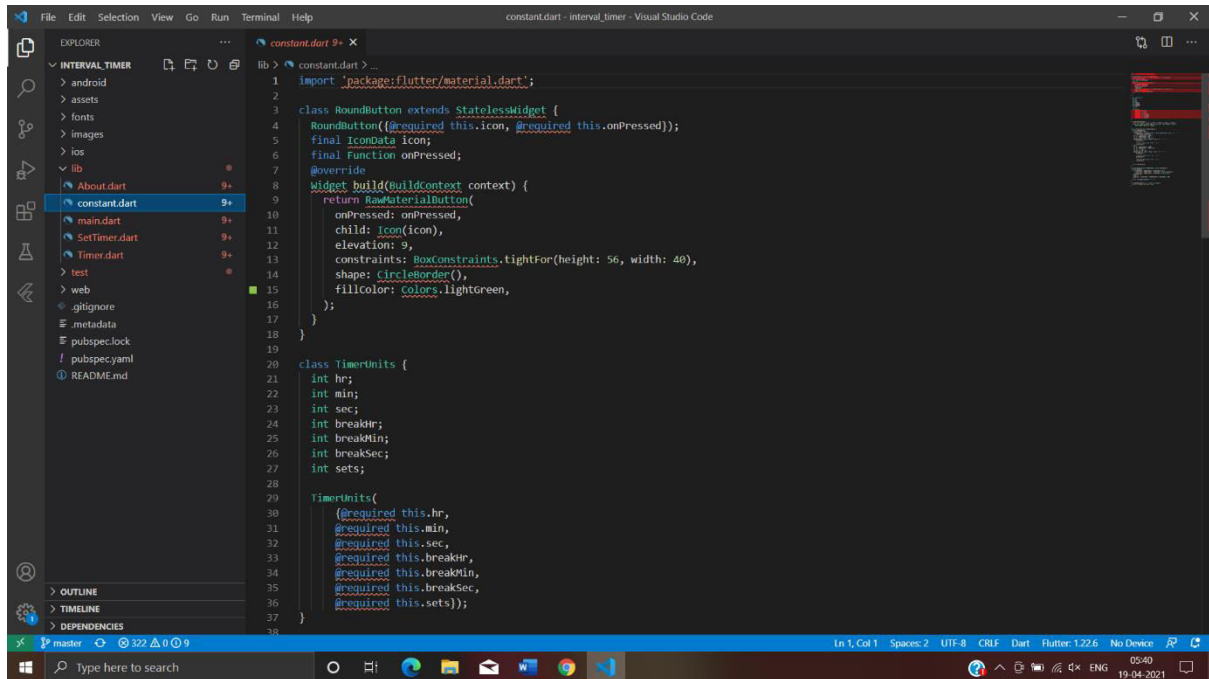
```
1 import 'package:flutter/material.dart';
2
3 class About extends StatelessWidget {
4   @override
5   Widget build(BuildContext context) {
6     return Scaffold(
7       child: Scaffold(
8         body: Center(
9           child: Column(
10            children: [
11              SizedBox(
12                height: 25,
13              ),
14              Text(
15                'This app is developed by - ',
16                style: TextStyle(
17                  fontSize: 20,
18                  color: Colors.limeAccent[400],
19                ),
20                textAlign: TextAlign.center,
21              ),
22              SizedBox(height: 25),
23              CircleAvatar(
24                backgroundImage: AssetImage('images/Aman.png'),
25                radius: 87,
26              ),
27              SizedBox(height: 25),
28              Text(
29                'Aman Kumar',
30                style: TextStyle(
31                  fontSize: 20,
32                  fontFamily: 'Comic',
33                ),
34              ),
35              Text(
36                'Phone: +91943595XXXX',
37                style: TextStyle(
38                  fontSize: 13,
```



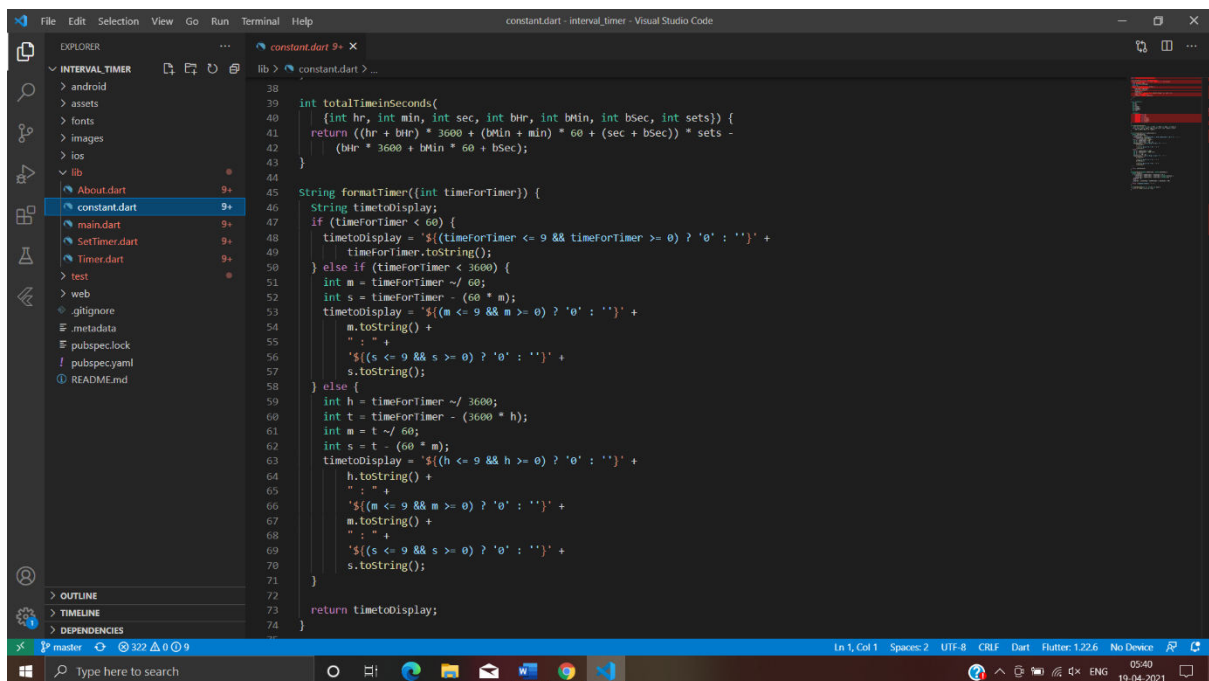
```
38          fontSize: 13,
39        ),
40      ),
41      Text(
42        'Email: aman.kumar_cs18@glac.ac.in',
43        style: TextStyle(
44          fontSize: 13,
45        ),
46      ),
47      SizedBox(
48        height: 25,
49      ),
50      Divider(
51        color: Colors.orange,
52        thickness: 10,
53      ),
54      SizedBox(height: 25),
55      CircleAvatar(
56        backgroundImage: AssetImage('images/Ayurish.png'),
57        radius: 87,
58      ),
59      SizedBox(height: 25),
60      Text(
61        'Ayurish Chandana',
62        style: TextStyle(
63          fontSize: 20,
64          fontFamily: 'Comic',
65        ),
66      ),
67      Text(
68        'Phone: +91948456XXXX',
69        style: TextStyle(
70          fontSize: 13,
71        ),
72      ),
73      Text(
74        'Email: ayurish.chandana_cs18@glac.ac.in'
```



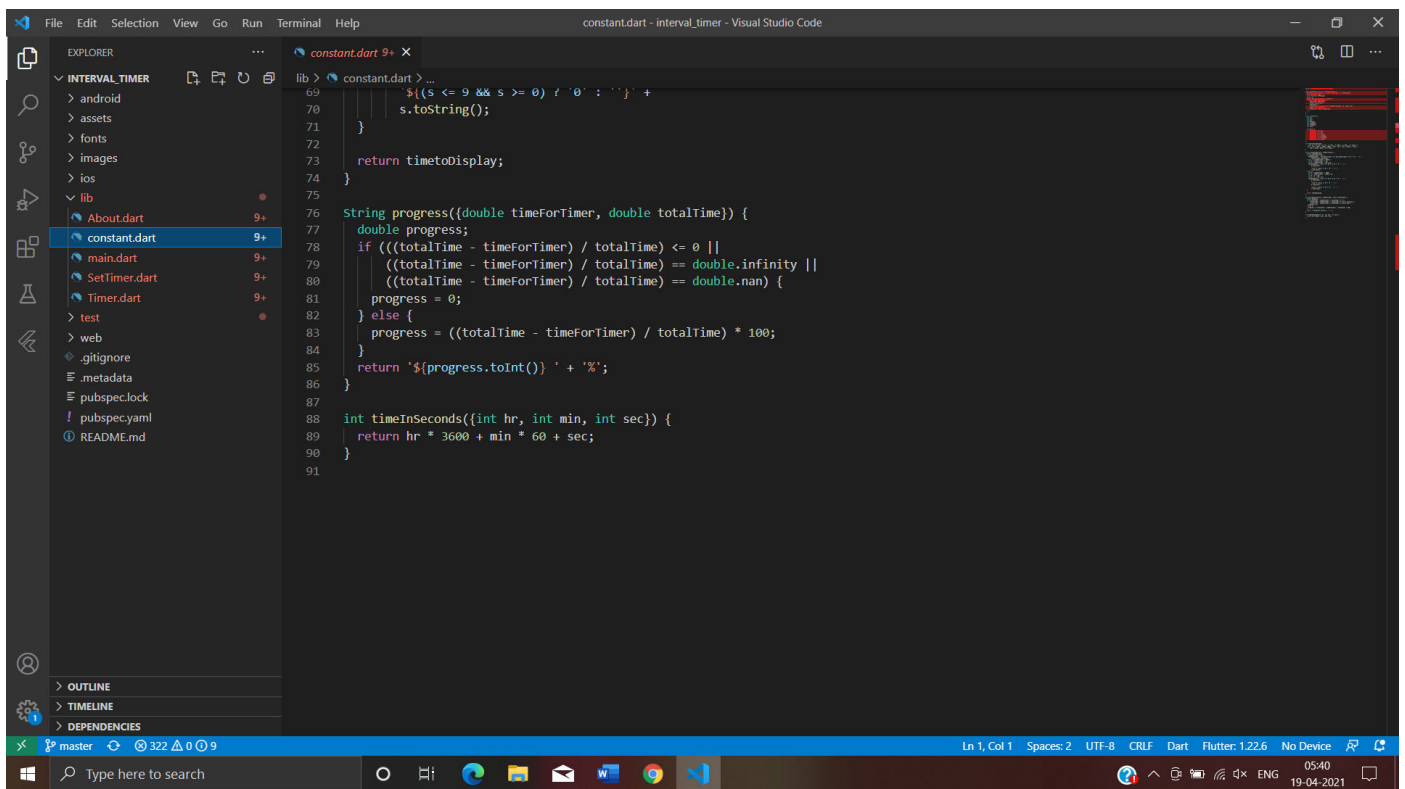
Filename : constant.dart



```
1 import 'package:flutter/material.dart';
2
3 class RoundButton extends StatelessWidget {
4   RoundButton({@required this.icon, @required this.onPressed});
5   final IconData icon;
6   final Function onPressed;
7   @override
8   Widget build(BuildContext context) {
9     return RawMaterialButton(
10       onPressed: onPressed,
11       child: Icon(icon),
12       elevation: 9,
13       constraints: BoxConstraints.tightFor(height: 56, width: 40),
14       shape: CircleBorder(),
15       fillColor: Colors.lightGreen,
16     );
17   }
18 }
19
20 class TimerUnits {
21   int hr;
22   int min;
23   int sec;
24   int breakHr;
25   int breakMin;
26   int breakSec;
27   int sets;
28
29   TimerUnits(
30     @required this.hr,
31     @required this.min,
32     @required this.sec,
33     @required this.breakHr,
34     @required this.breakMin,
35     @required this.breakSec,
36     @required this.sets);
37 }
38
```



```
38
39 int totalTimeInSeconds(
40   {int hr, int min, int sec, int bHr, int bMin, int bSec, int sets}) {
41   return ((hr + bHr) * 3600 + (min + bMin) * 60 + (sec + bSec)) * sets -
42     ((bHr * 3600 + bMin * 60 + bSec));
43 }
44
45 String formatTimer([int timeForTimer]) {
46   String timeToDisplay;
47   if (timeForTimer < 60) {
48     timeToDisplay = '${(timeForTimer <= 9 && timeForTimer >= 0) ? '0' : ''}' +
49       timeForTimer.toString();
50   } else if (timeForTimer < 3600) {
51     int m = timeForTimer ~/ 60;
52     int s = timeForTimer - (60 * m);
53     timeToDisplay = '${(m <= 9 && m >= 0) ? '0' : ''}' +
54       m.toString() +
55       ':' +
56       '${(s <= 9 && s >= 0) ? '0' : ''}' +
57       s.toString();
58   } else {
59     int h = timeForTimer ~/ 3600;
60     int t = timeForTimer - (3600 * h);
61     int m = t ~/ 60;
62     int s = t - (60 * m);
63     timeToDisplay = '${(h <= 9 && h >= 0) ? '0' : ''}' +
64       h.toString() +
65       ':' +
66       '${(m <= 9 && m >= 0) ? '0' : ''}' +
67       m.toString() +
68       ':' +
69       '${(s <= 9 && s >= 0) ? '0' : ''}' +
70       s.toString();
71   }
72
73   return timeToDisplay;
74 }
75
```



Filename : SetTimer.dart

```

1  import 'constant.dart';
2  import 'package:flutter/material.dart';
3  import 'package:numberpicker/numberpicker.dart';
4  import 'timer.dart';
5
6  class SetTimer extends StatefulWidget {
7    @override
8    _SetTimerState createState() => _SetTimerState();
9  }
10
11  class _SetTimerState extends State<SetTimer> {
12    int hr = 0;
13    int min = 0;
14    int sec = 0;
15    int breakHr = 0;
16    int breakMin = 0;
17    int breakSec = 0;
18    int sets = 1;
19    @override
20    Widget build(BuildContext context) {
21      return Scaffold(
22        body: Container(
23          child: ListView(
24            scrollDirection: Axis.vertical,
25            children: [
26              SizedBox(height: 30),
27              Row(
28                children: [
29                  Padding(
30                    padding: const EdgeInsets.only(left: 30),
31                    child: Text(
32                      'Timer',
33                      style: TextStyle(
34                        fontSize: 30,
35                        fontWeight: FontWeight.bold,
36                        letterSpacing: 2),
37                    ),
38                ],
39              ),
40            ],
41          ),
42        ),
43      );
44    }
45  }

```

File Edit Selection View Go Run Terminal Help

SetTimer.dart - interval_timer - Visual Studio Code

EXPLORER

INTERVAL_TIMER

- android
- assets
- fonts
- images
- ios
- lib
 - About.dart
 - constant.dart
 - main.dart
 - SetTimer.dart**
 - Timer.dart
- test
- web
- .gitignore
- metadata
- pubspec.lock
- pubspec.yaml
- README.md

lib > SetTimer.dart

```
//
import 'package:flutter/material.dart';
import 'package:icons_plus/icons.dart';
import 'package:timer/timer.dart';
import 'package:timer/timer.dart';

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

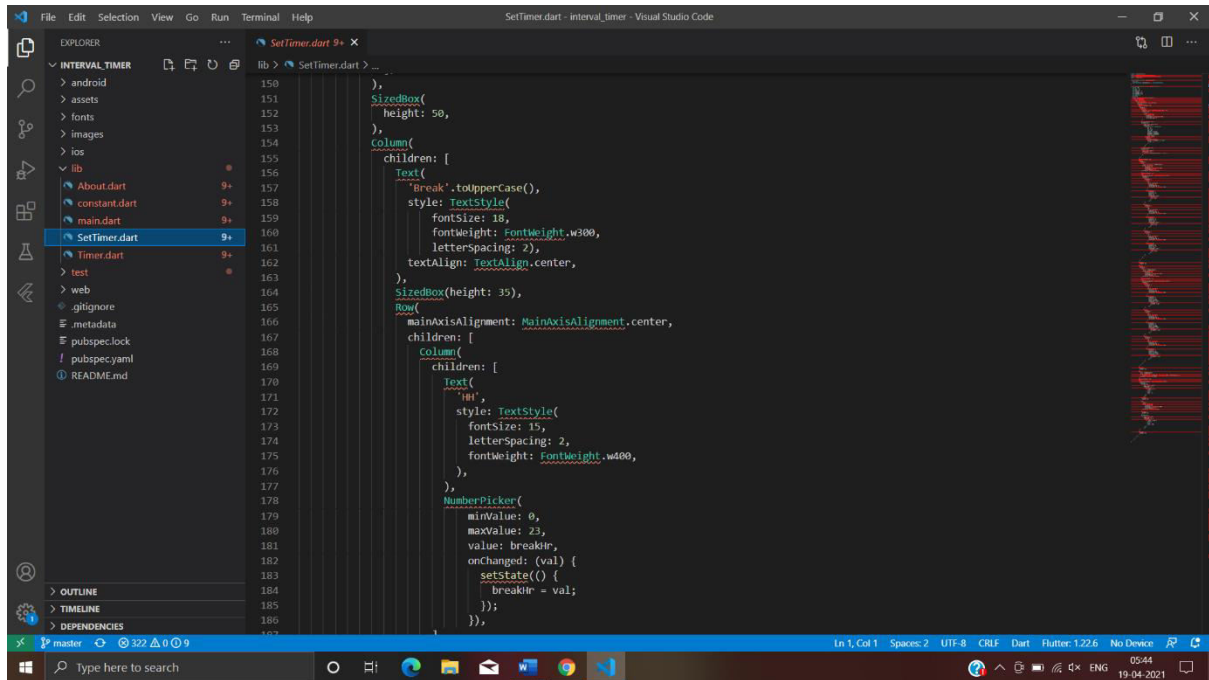
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Timer App',
      theme: ThemeData(
        primarySwatch: Colors.pink,
      ),
      home: MyHomePage(),
    );
  }
}

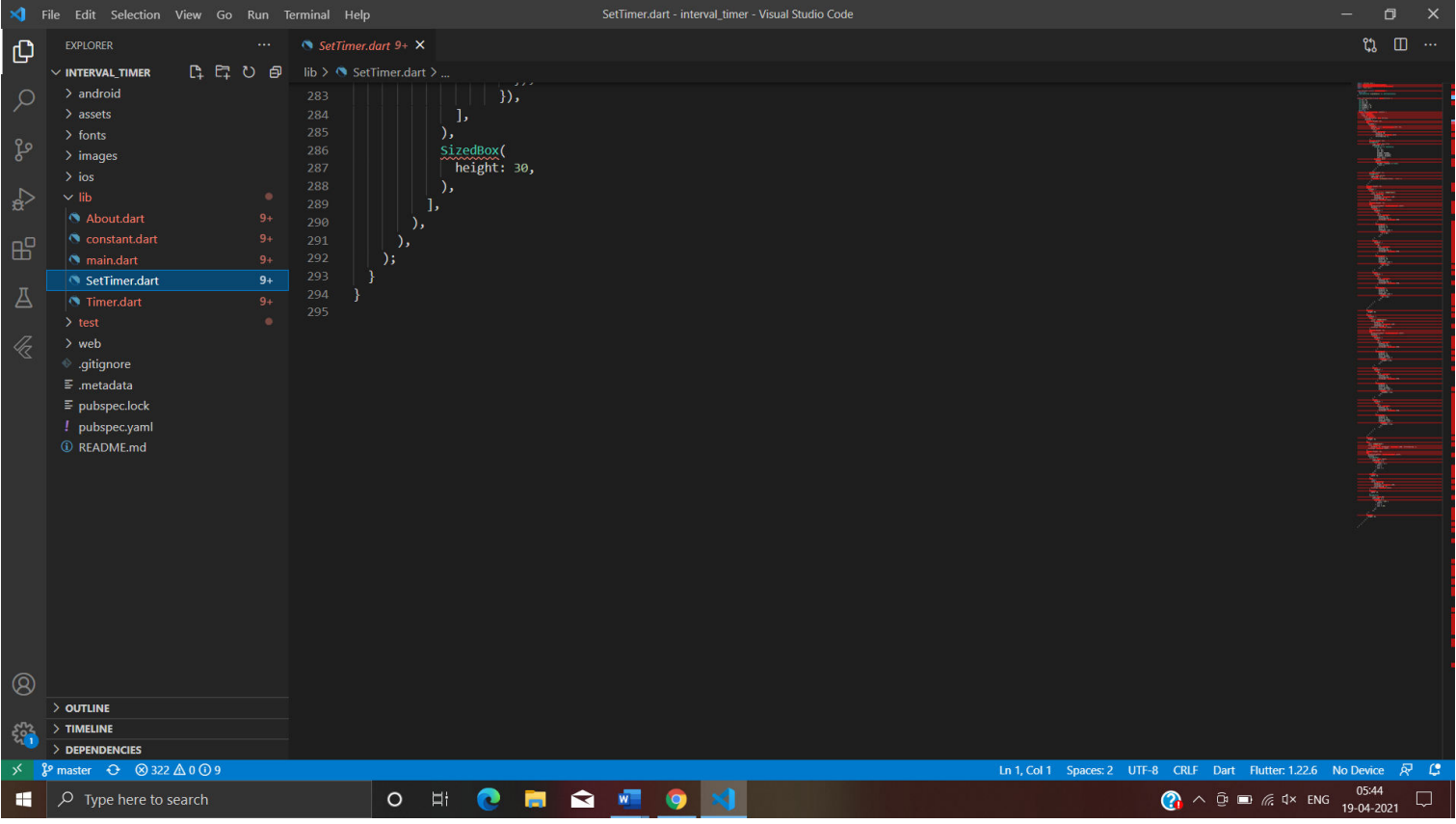
class MyHomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: 'Timer App',
      ),
      body: Column(
        children: [
          Text(
            'Time for workout'.toUpperCase(),
            style: TextStyle(
              color: Colors.pink,
            ),
          ),
          Row(
            children: [
              IconButton(
                icon: Icons.play_arrow,
                onPressed: () {
                  Timer(
                    10,
                    () {
                      Navigator.pushNamed(
                        context,
                        '/about';
                      );
                    },
                  );
                },
              ),
              IconButton(
                icon: Icons.person,
                onPressed: () {
                  Navigator.pushNamed(
                    context,
                    '/about';
                  );
                },
              ),
            ],
          ),
        ],
      ),
    );
  }
}
```

In 1, Col 1 Spaces: 2 UTF-8 CRLF Dart Flutter 1.22.6 No Device

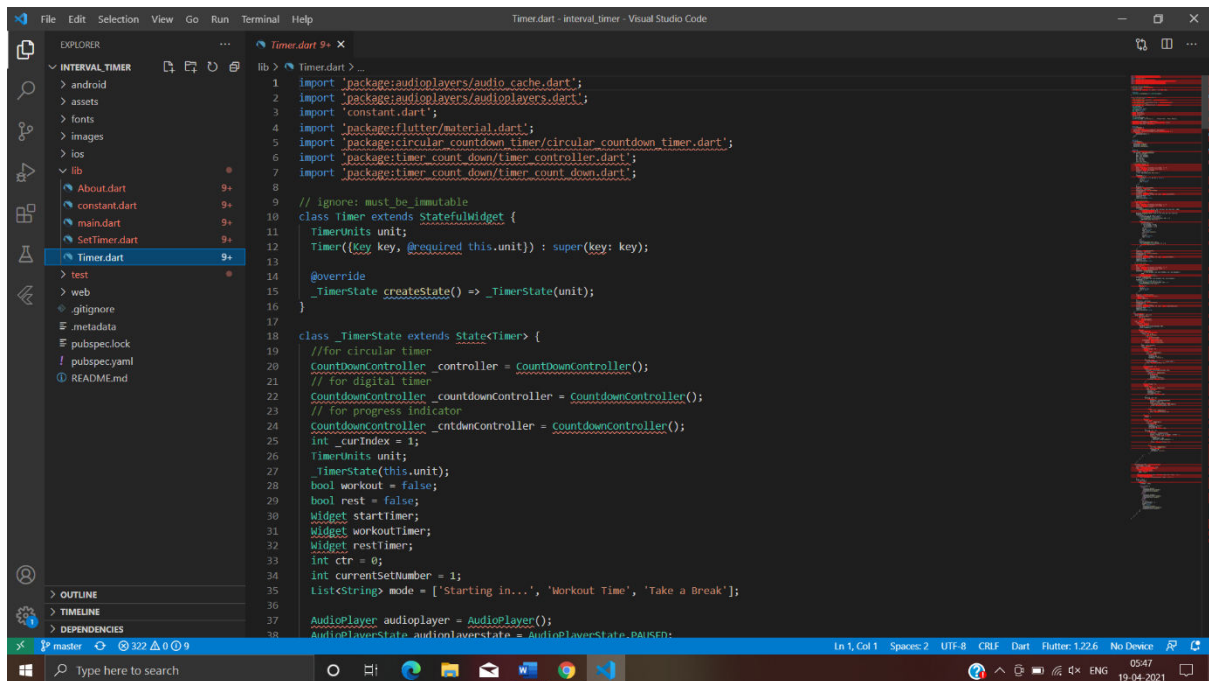
```
75      fontSize: 18,  
76      fontWeight: FontWeight.w300,  
77      letterSpacing: 2),  
78      textAlign: TextAlign.center,  
79    ),  
80    SizedBox(height: 35),  
81    Row(  
82      mainAxisAlignment: MainAxisAlignment.center,  
83      children: [  
84        Column(  
85          children: [  
86            Text(  
87              'H',  
88              style: TextStyle(  
89                fontSize: 15,  
90                letterSpacing: 2,  
91                fontWeight: FontWeight.w400,  
92              ),  
93            ),  
94            NumberPicker(  
95              minValue: 0,  
96              maxValue: 23,  
97              value: hr,  
98              onChanged: (val) {  
99                setState(() {  
100                  hr = val;  
101                });  
102              }),  
103            ],  
104          ),  
105        ],  
106      ),  
107      Column(  
108        children: [  
109          Text(  
110            'M',  
111            style: TextStyle(  
112              fontSize: 15,  
113              letterSpacing: 2,  
114              fontWeight: FontWeight.w400,
```

```
111      letterSpacing: 2,  
112      fontWeight: FontWeight.w400,  
113    ),  
114    ),  
115    NumberPicker(  
116      minValue: 0,  
117      maxValue: 59,  
118      value: min,  
119      onChanged: (val) {  
120        setState(() {  
121          min = val;  
122        });  
123      }),  
124    ),  
125    Column(  
126      children: [  
127        Text(  
128          'SS',  
129          style: TextStyle(  
130            fontSize: 15,  
131            letterSpacing: 2,  
132            fontWeight: FontWeight.w400,  
133          ),  
134        ),  
135        NumberPicker(  
136          minValue: 0,  
137          maxValue: 59,  
138          value: sec,  
139          onChanged: (val) {  
140            setState(() {  
141              sec = val;  
142            });  
143          }),  
144        ],  
145      ),  
146    ),  
147    ],  
148  ),
```

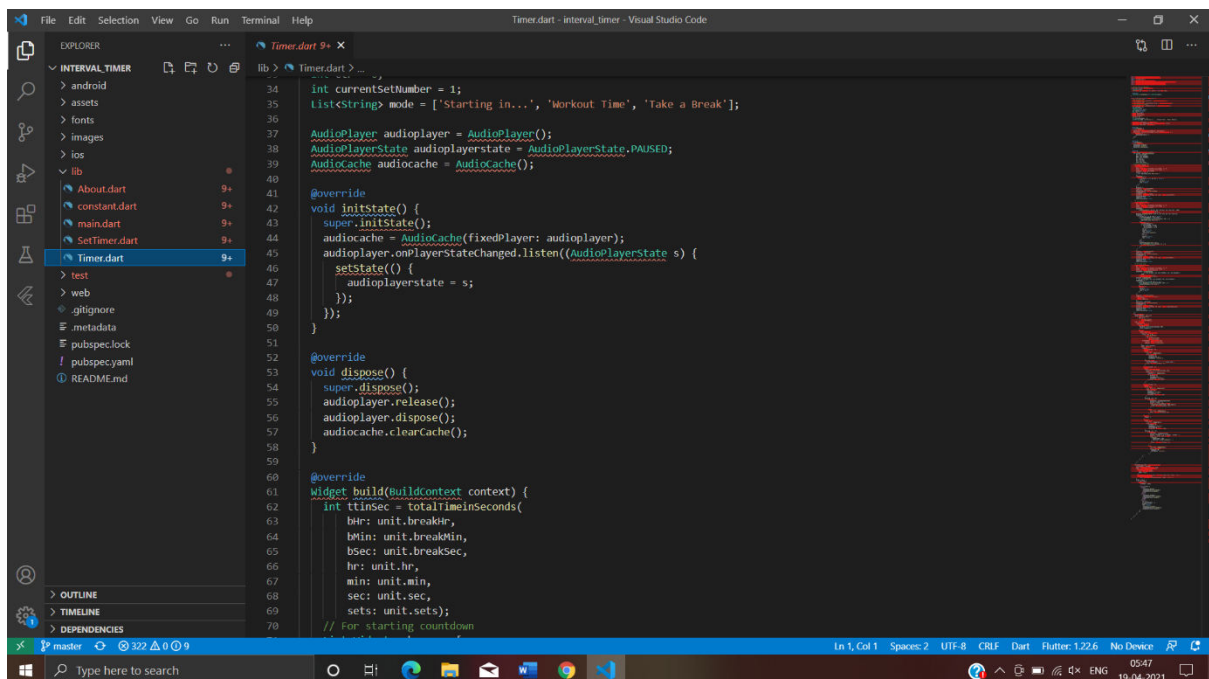





Filename : Timer.dart



```
1 import 'package:audioplayers/audio_cache.dart';
2 import 'package:audioplayers/audioplayers.dart';
3 import 'package:constant.dart';
4 import 'package:flutter/material.dart';
5 import 'package:circular_countdown_timer/circular_countdown_timer.dart';
6 import 'package:timer_count_down/timer_controller.dart';
7 import 'package:timer_count_down/timer_count_down.dart';
8
9 // ignore: must_be_immutable
10 class Timer extends StatefulWidget {
11   TimerUnits unit;
12   Timer(Key key, @required this.unit) : super(key: key);
13
14   @override
15   _TimerState createState() => _TimerState(unit);
16 }
17
18 class _TimerState extends State<Timer> {
19   //for circular timer
20   CountdownController _controller = CountdownController();
21   // for digital timer
22   CountdownController countdownController = CountdownController();
23   // for progress indicator
24   CountdownController cntdwnController = CountdownController();
25   int _currentIndex = 1;
26   TimerUnits unit;
27   _TimerState(this.unit);
28   bool workout = false;
29   bool rest = false;
30   Widget startTimer;
31   Widget workoutTimer;
32   Widget restTimer;
33   int ctr = 0;
34   int currentSetNumber = 1;
35   List<String> mode = ['Starting in...', 'Workout Time', 'Take a Break'];
36
37   AudioPlayer audioplayer = AudioPlayer();
38   AudioPlayerState audioplayerstate = AudioPlayerState.PAUSED;
```



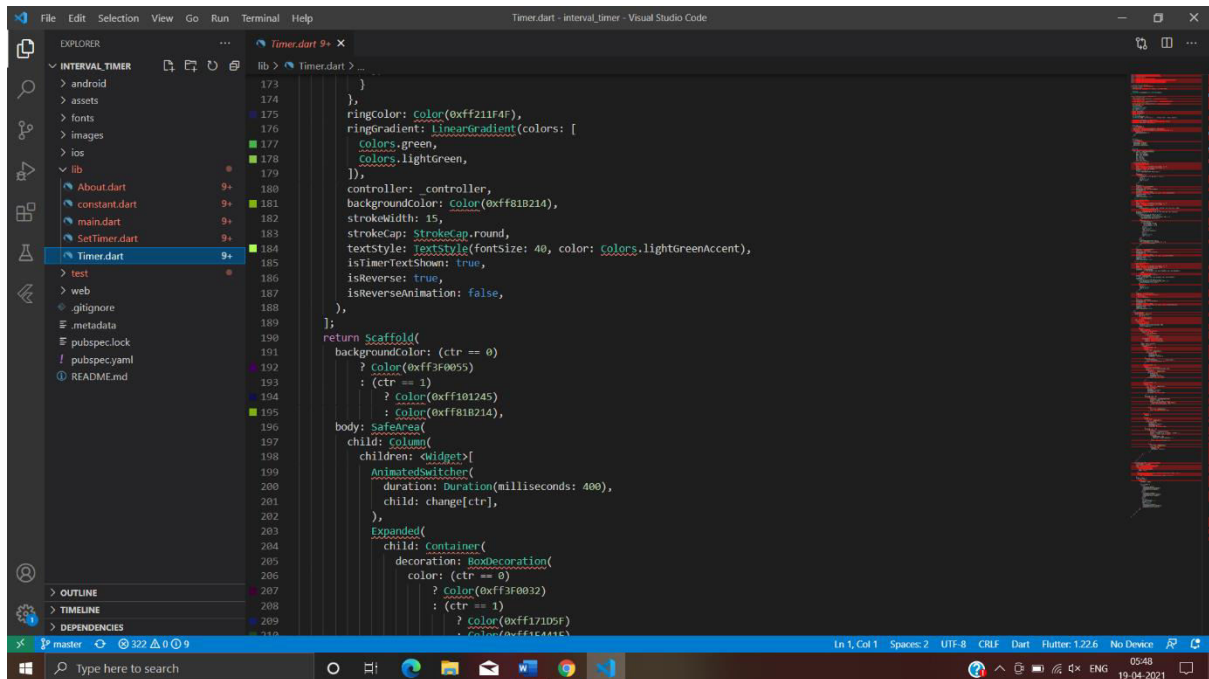
```
34   int currentSetNumber = 1;
35   List<String> mode = ['Starting in...', 'Workout Time', 'Take a Break'];
36
37   AudioPlayer audioplayer = AudioPlayer();
38   AudioPlayerState audioplayerstate = AudioPlayerState.PAUSED;
39   AudioCache audiocache = AudioCache();
40
41   @override
42   void initState() {
43     super.initState();
44     audiocache = AudioCache(fixedPlayer: audioplayer);
45     audioplayer.onPlayerStateChanged.listen((AudioPlayerState s) {
46       setState(() {
47         audioplayerstate = s;
48       });
49     });
50   }
51
52   @override
53   void dispose() {
54     super.dispose();
55     audioplayer.release();
56     audioplayer.dispose();
57     audiocache.clearCache();
58   }
59
60   @override
61   Widget build(BuildContext context) {
62     int ttinSec = totalTimeInSeconds(
63       bhr: unit.breakHr,
64       bmin: unit.breakMin,
65       bsec: unit.breakSec,
66       hr: unit.hr,
67       min: unit.min,
68       sec: unit.sec,
69       sets: unit.sets);
70     // For starting countdown
```

The screenshot shows the Visual Studio Code editor with the file `Timer.dart` open. The Explorer sidebar on the left shows a project structure for an Android app, with the `lib` directory expanded and `Timer.dart` selected. The code in the editor is as follows:

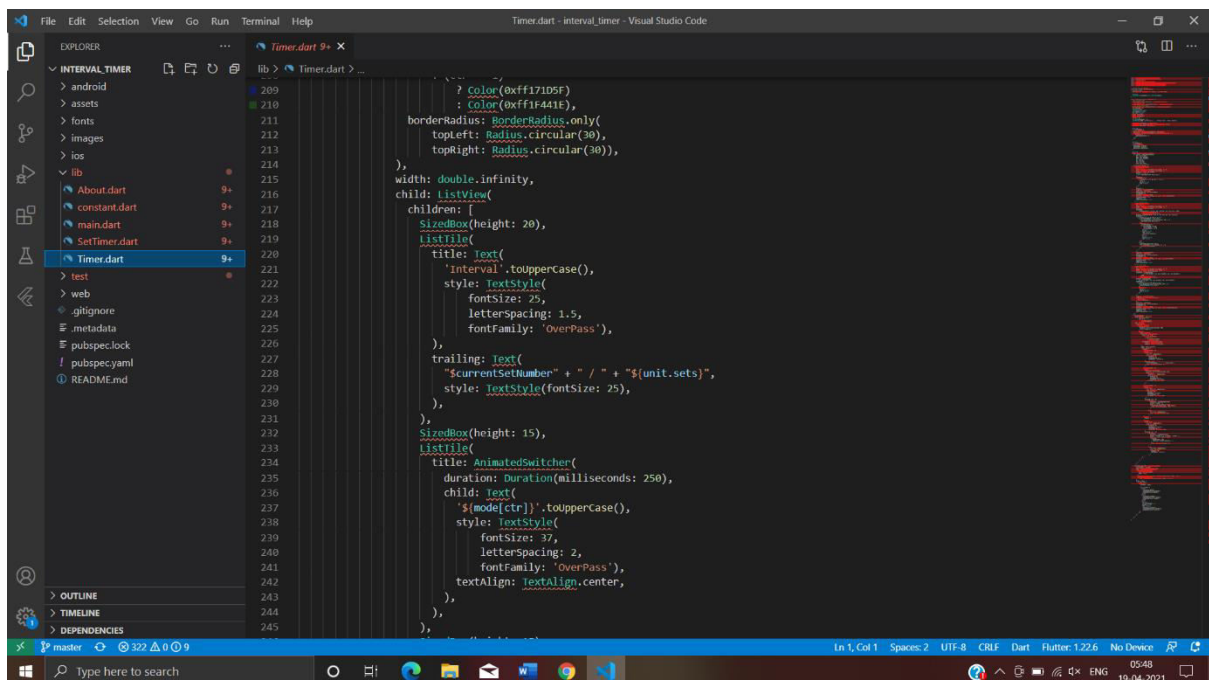
```
lib > Timer.dart > ...
99
100 // For workout timer
101 CircularCountdownTimer(
102   key: Key('2'),
103   width: MediaQuery.of(context).size.width / 2 + 50,
104   height: MediaQuery.of(context).size.height / 2,
105   fillColor: color(0xff218DFA),
106   textFormat:
107     (timeInSeconds(hr: unit.hr, min: unit.min, sec: unit.sec) < 3600)
108       ? CountdownTextFormat.MM_SS
109       : CountdownTextFormat.HH_MM_SS,
110   duration: timeInSeconds(hr: unit.hr, min: unit.min, sec: unit.sec),
111   onComplete: () {
112     if (workout == true && rest == false) {
113       if (currentSetNumber < unit.sets) {
114         Future.delayed(Duration(milliseconds: 250), () {
115           audiocache.play('Beep2.mpeg');
116         }); // Future.delayed
117         setState(() {
118           currentSetNumber++;
119           if (unit.breakHr == 0 &&
120             unit.breakMin == 0 &&
121             unit.breakSec == 0) {
122             ctr = 1;
123             workout = true;
124             rest = false;
125             _controller.restart();
126           } else {
127             ctr = 2;
128             workout = false;
129             rest = true;
130           }
131         });
132       } else {
133         audiocache.play('Beep3.mpeg');
134         Future.delayed(Duration(seconds: 1), () {
135           Navigator.pop(context);
136         }); // Future.delayed
137       }
138     }
139   },
140   ringColor: color(0xff211F4F),
141   backgroundColor: color(0xff101245),
142   strokeWidth: 15,
143   strokeCap: StrokeCap.round,
144   textStyle: TextStyle(fontSize: 40, color: color(0xff218DFA)),
145   isTimerTextShown: true,
146   isReverse: true,
147   isReverseAnimation: false,
148 ),
149 // for break timer
150 CircularCountdownTimer(
151   key: Key('3'),
152   width: MediaQuery.of(context).size.width / 2 + 50,
153   height: MediaQuery.of(context).size.height / 2,
154   fillColor: Colors.lightGreenAccent,
155   textFormat:
156     (timeInSeconds(
157       hr: unit.breakHr, min: unit.breakMin, sec: unit.breakSec) <
158       3600)
159       ? CountdownTextFormat.MM_SS
160       : CountdownTextFormat.HH_MM_SS,
161   duration: timeInSeconds(
162     hr: unit.breakHr, min: unit.breakMin, sec: unit.breakSec),
163   onComplete: () {
164     if (workout == false && rest == true) {
165       Future.delayed(Duration(milliseconds: 250), () {
166         audiocache.play('Beep4.mpeg');
167       }); // Future.delayed
168       setState(() {
169         ctr = 1;
170         workout = true;
171         rest = false;
172       });
173     }
174   },
175 );
```

The screenshot shows the Visual Studio Code editor with the file `Timer.dart` open. The Explorer sidebar on the left shows the same project structure, with `Timer.dart` selected. The code in the editor is as follows:

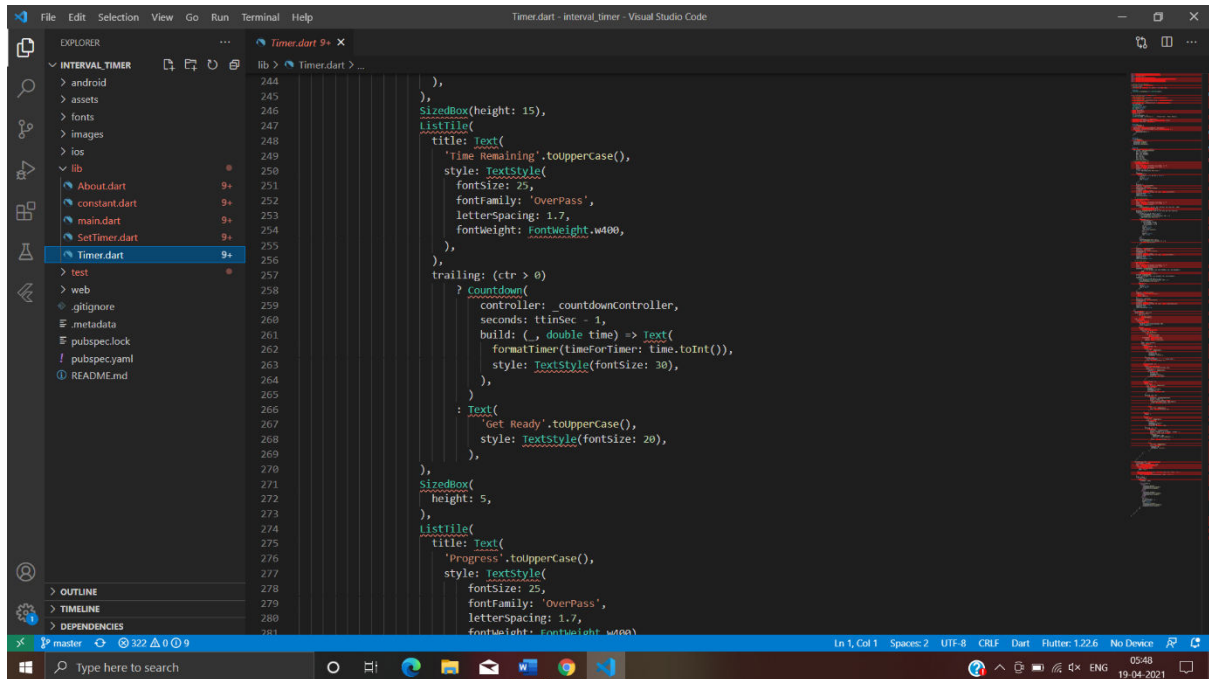
```
lib > Timer.dart > ...
135 Navigator.pop(context);
136 }); // Future.delayed
137 }
138 }
139 },
140 ringColor: color(0xff211F4F),
141 backgroundColor: color(0xff101245),
142 strokeWidth: 15,
143 strokeCap: StrokeCap.round,
144 textStyle: TextStyle(fontSize: 40, color: color(0xff218DFA)),
145 isTimerTextShown: true,
146 isReverse: true,
147 isReverseAnimation: false,
148 ),
149 // for break timer
150 CircularCountdownTimer(
151   key: Key('3'),
152   width: MediaQuery.of(context).size.width / 2 + 50,
153   height: MediaQuery.of(context).size.height / 2,
154   fillColor: Colors.lightGreenAccent,
155   textFormat:
156     (timeInSeconds(
157       hr: unit.breakHr, min: unit.breakMin, sec: unit.breakSec) <
158       3600)
159       ? CountdownTextFormat.MM_SS
160       : CountdownTextFormat.HH_MM_SS,
161   duration: timeInSeconds(
162     hr: unit.breakHr, min: unit.breakMin, sec: unit.breakSec),
163   onComplete: () {
164     if (workout == false && rest == true) {
165       Future.delayed(Duration(milliseconds: 250), () {
166         audiocache.play('Beep4.mpeg');
167       }); // Future.delayed
168       setState(() {
169         ctr = 1;
170         workout = true;
171         rest = false;
172       });
173     }
174   },
175 );
```



```
173 }
174 },
175 ringcolor: color(0xff211f4f),
176 ringgradient: LinearGradient(colors: [
177   colors.green,
178   colors.lightgreen,
179 ]),
180 controller: _controller,
181 backgroundColor: color(0xff81b214),
182 strokeWidth: 15,
183 strokeCap: StrokeCap.round,
184 textStyle: TextStyle(fontSize: 40, color: colors.lightGreenAccent),
185 isTimerTextShown: true,
186 isReverse: true,
187 isReverseAnimation: false,
188 ),
189 ];
190 return Scaffold(
191   backgroundColor: (ctr == 0)
192     ? color(0xff3f0055)
193     : (ctr == 1)
194       ? color(0xff101245)
195       : color(0xff81b214),
196   body: SafeArea(
197     child: Column(
198       children: <widget>[
199         AnimatedSwitcher(
200           duration: Duration(milliseconds: 400),
201           child: change[ctr],
202         ),
203         Expanded(
204           child: Container(
205             decoration: BoxDecoration(
206               color: (ctr == 0)
207                 ? color(0xff3f0032)
208                 : (ctr == 1)
209                   ? color(0xff171d5f)
210                   : color(0xff1f441e),
211             borderRadius: BorderRadius.only(
212               topLeft: Radius.circular(30),
213               topRight: Radius.circular(30)),
214             width: double.infinity,
215             child: ListView(
216               children: [
217                 SizedBox(height: 20),
218                 ListTile(
219                   title: Text(
220                     'Interval'.toUpperCase(),
221                     style: TextStyle(
222                       fontSize: 25,
223                       letterSpacing: 1.5,
224                       fontFamily: 'OverPass'),
225                   ),
226                   trailing: Text(
227                     '$currentSetNumber' + ' / ' + '$unit.sets',
228                     style: TextStyle(fontSize: 25),
229                   ),
230                 ),
231                 SizedBox(height: 15),
232                 ListTile(
233                   title: AnimatedSwitcher(
234                     duration: Duration(milliseconds: 250),
235                     child: Text(
236                       '${mode[ctr]}'.toUpperCase(),
237                       style: TextStyle(
238                         fontSize: 37,
239                         letterSpacing: 2,
240                         fontFamily: 'OverPass'),
241                       textAlign: TextAlign.center,
242                     ),
243                   ),
244                 ),
245               ],
246             ),
247           ),
248         ),
249       ],
250     ),
251   ),
252 );
```



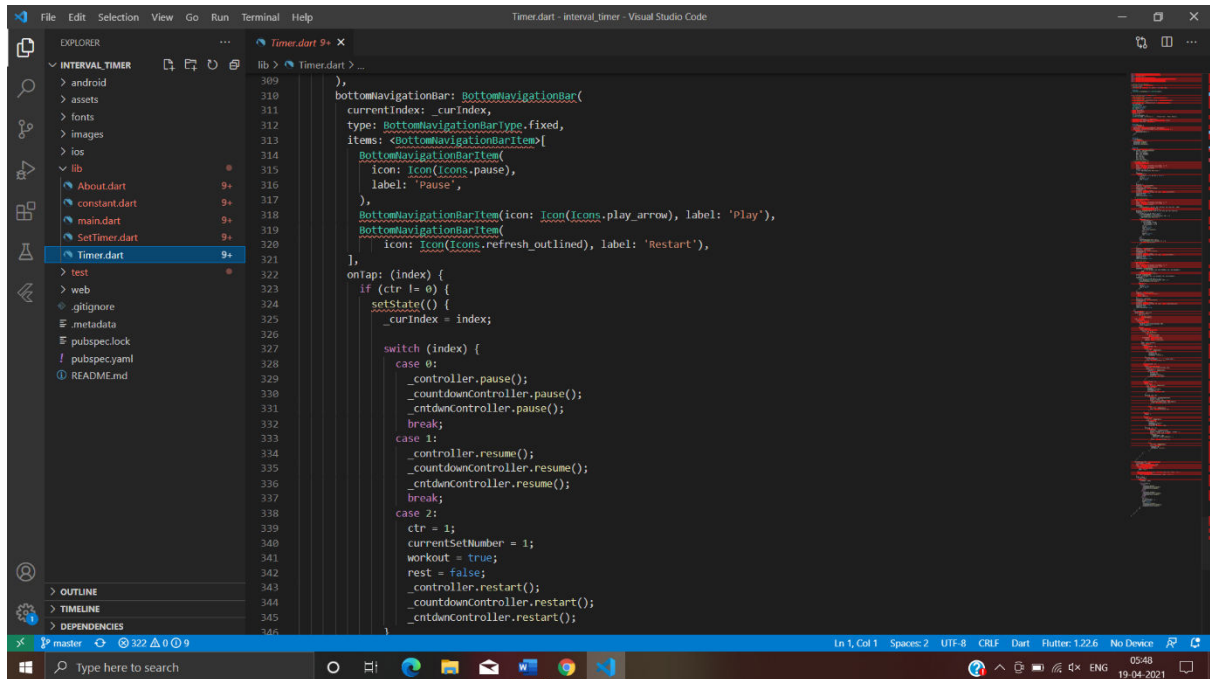
```
209   ? color(0xff171d5f)
210   : color(0xff1f441e),
211   borderRadius: BorderRadius.only(
212     topLeft: Radius.circular(30),
213     topRight: Radius.circular(30)),
214   width: double.infinity,
215   child: ListView(
216     children: [
217       SizedBox(height: 20),
218       ListTile(
219         title: Text(
220           'Interval'.toUpperCase(),
221           style: TextStyle(
222             fontSize: 25,
223             letterSpacing: 1.5,
224             fontFamily: 'OverPass'),
225         ),
226         trailing: Text(
227           '$currentSetNumber' + ' / ' + '$unit.sets',
228           style: TextStyle(fontSize: 25),
229         ),
230       ),
231       SizedBox(height: 15),
232       ListTile(
233         title: AnimatedSwitcher(
234           duration: Duration(milliseconds: 250),
235           child: Text(
236             '${mode[ctr]}'.toUpperCase(),
237             style: TextStyle(
238               fontSize: 37,
239               letterSpacing: 2,
240               fontFamily: 'OverPass'),
241             textAlign: TextAlign.center,
242           ),
243         ),
244       ),
245     ],
246   ),
247 ),
248 ),
249 ],
250 ),
251 ),
252 );
```

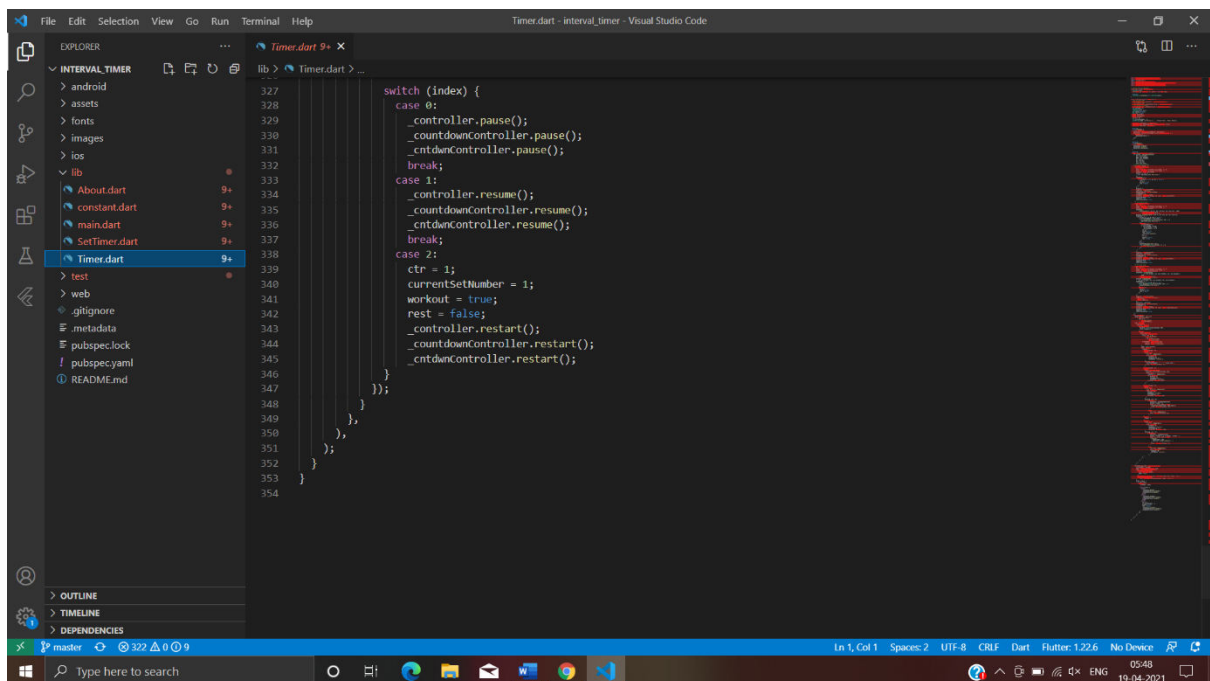
This screenshot shows the Visual Studio Code editor with the file `Timer.dart` open. The Explorer sidebar on the left shows a project structure with folders like `android`, `assets`, `fonts`, `images`, `ios`, `lib`, `test`, and `web`. The `lib` folder is expanded, showing files like `About.dart`, `constant.dart`, `main.dart`, `SetTimer.dart`, and `Timer.dart`. The `Timer.dart` file is selected and its content is displayed in the editor. The code defines a widget with a title, a progress bar, and a countdown timer. The status bar at the bottom indicates the file is on line 1, column 1, using UTF-8 encoding, with 2 spaces, CRLF line endings, and is compatible with Dart and Flutter 1.22.6. The system tray shows the date and time as 05:48 on 19-04-2021.

```
244      ),
245    ),
246    SizedBox(height: 15),
247    ListTile(
248      title: Text(
249        'Time Remaining'.toUpperCase(),
250        style: TextStyle(
251          fontSize: 25,
252          fontFamily: 'OverPass',
253          letterSpacing: 1.7,
254          fontWeight: FontWeight.w400,
255        ),
256      ),
257      trailing: (ctr > 0)
258        ? Countdown(
259          controller: _countdowncontroller,
260          seconds: ttinsec - 1,
261          build: (_, double time) => Text(
262            formatTimer(timeForTimer: time.toInt()),
263            style: TextStyle(fontSize: 30),
264          ),
265        )
266        : Text(
267          'Get Ready'.toUpperCase(),
268          style: TextStyle(fontSize: 20),
269        ),
270    ),
271    SizedBox(
272      height: 5,
273    ),
274    ListTile(
275      title: Text(
276        'Progress'.toUpperCase(),
277        style: TextStyle(
278          fontSize: 25,
279          fontFamily: 'OverPass',
280          letterSpacing: 1.7,
281          fontWeight: FontWeight.w400,
282        ),
283      ),
284    ),
285    ),
286  ], // <Widget>[]
287),
288),
289),
290),
291),
292),
293),
294),
295),
296),
297),
298),
299),
300),
301),
302),
303),
304),
305),
306),
307),
308),
309),
310),
311),
312),
313),
314),
315),
316),
317),
318),
319),
320),
321),
322),
323),
324),
325),
326),
327),
328),
329),
330),
331),
332),
333),
334),
335),
336),
337),
338),
339),
340),
341),
342),
343),
344),
345),
346),
347),
348),
349),
350),
351),
352),
353),
354),
355),
356),
357),
358),
359),
360),
361),
362),
363),
364),
365),
366),
367),
368),
369),
370),
371),
372),
373),
374),
375),
376),
377),
378),
379),
380),
381),
382),
383),
384),
385),
386),
387),
388),
389),
390),
391),
392),
393),
394),
395),
396),
397),
398),
399),
400),
401),
402),
403),
404),
405),
406),
407),
408),
409),
410),
411),
412),
413),
414),
415),
416),
417),
418),
419),
420),
421),
422),
423),
424),
425),
426),
427),
428),
429),
430),
431),
432),
433),
434),
435),
436),
437),
438),
439),
440),
441),
442),
443),
444),
445),
446),
447),
448),
449),
450),
451),
452),
453),
454),
455),
456),
457),
458),
459),
460),
461),
462),
463),
464),
465),
466),
467),
468),
469),
470),
471),
472),
473),
474),
475),
476),
477),
478),
479),
480),
481),
482),
483),
484),
485),
486),
487),
488),
489),
490),
491),
492),
493),
494),
495),
496),
497),
498),
499),
500),
501),
502),
503),
504),
505),
506),
507),
508),
509),
510),
511),
512),
513),
514),
515),
516),
517),
518),
519),
520),
521),
522),
523),
524),
525),
526),
527),
528),
529),
530),
531),
532),
533),
534),
535),
536),
537),
538),
539),
540),
541),
542),
543),
544),
545),
546),
547),
548),
549),
550),
551),
552),
553),
554),
555),
556),
557),
558),
559),
560),
561),
562),
563),
564),
565),
566),
567),
568),
569),
570),
571),
572),
573),
574),
575),
576),
577),
578),
579),
580),
581),
582),
583),
584),
585),
586),
587),
588),
589),
590),
591),
592),
593),
594),
595),
596),
597),
598),
599),
600),
601),
602),
603),
604),
605),
606),
607),
608),
609),
610),
611),
612),
613),
614),
615),
616),
617),
618),
619),
620),
621),
622),
623),
624),
625),
626),
627),
628),
629),
630),
631),
632),
633),
634),
635),
636),
637),
638),
639),
640),
641),
642),
643),
644),
645),
646),
647),
648),
649),
650),
651),
652),
653),
654),
655),
656),
657),
658),
659),
660),
661),
662),
663),
664),
665),
666),
667),
668),
669),
670),
671),
672),
673),
674),
675),
676),
677),
678),
679),
680),
681),
682),
683),
684),
685),
686),
687),
688),
689),
690),
691),
692),
693),
694),
695),
696),
697),
698),
699),
700),
701),
702),
703),
704),
705),
706),
707),
708),
709),
710),
711),
712),
713),
714),
715),
716),
717),
718),
719),
720),
721),
722),
723),
724),
725),
726),
727),
728),
729),
730),
731),
732),
733),
734),
735),
736),
737),
738),
739),
740),
741),
742),
743),
744),
745),
746),
747),
748),
749),
750),
751),
752),
753),
754),
755),
756),
757),
758),
759),
760),
761),
762),
763),
764),
765),
766),
767),
768),
769),
770),
771),
772),
773),
774),
775),
776),
777),
778),
779),
780),
781),
782),
783),
784),
785),
786),
787),
788),
789),
790),
791),
792),
793),
794),
795),
796),
797),
798),
799),
800),
801),
802),
803),
804),
805),
806),
807),
808),
809),
810),
811),
812),
813),
814),
815),
816),
817),
818),
819),
820),
821),
822),
823),
824),
825),
826),
827),
828),
829),
830),
831),
832),
833),
834),
835),
836),
837),
838),
839),
840),
841),
842),
843),
844),
845),
846),
847),
848),
849),
850),
851),
852),
853),
854),
855),
856),
857),
858),
859),
860),
861),
862),
863),
864),
865),
866),
867),
868),
869),
870),
871),
872),
873),
874),
875),
876),
877),
878),
879),
880),
881),
882),
883),
884),
885),
886),
887),
888),
889),
890),
891),
892),
893),
894),
895),
896),
897),
898),
899),
900),
901),
902),
903),
904),
905),
906),
907),
908),
909),
910),
911),
912),
913),
914),
915),
916),
917),
918),
919),
920),
921),
922),
923),
924),
925),
926),
927),
928),
929),
930),
931),
932),
933),
934),
935),
936),
937),
938),
939),
940),
941),
942),
943),
944),
945),
946),
947),
948),
949),
950),
951),
952),
953),
954),
955),
956),
957),
958),
959),
960),
961),
962),
963),
964),
965),
966),
967),
968),
969),
970),
971),
972),
973),
974),
975),
976),
977),
978),
979),
980),
981),
982),
983),
984),
985),
986),
987),
988),
989),
990),
991),
992),
993),
994),
995),
996),
997),
998),
999),
1000),
1001),
1002),
1003),
1004),
1005),
1006),
1007),
1008),
1009),
1010),
1011),
1012),
1013),
1014),
1015),
1016),
1017),
1018),
1019),
1020),
1021),
1022),
1023),
1024),
1025),
1026),
1027),
1028),
1029),
1030),
1031),
1032),
1033),
1034),
1035),
1036),
1037),
1038),
1039),
1040),
1041),
1042),
1043),
1044),
1045),
1046),
1047),
1048),
1049),
1050),
1051),
1052),
1053),
1054),
1055),
1056),
1057),
1058),
1059),
1060),
1061),
1062),
1063),
1064),
1065),
1066),
1067),
1068),
1069),
1070),
1071),
1072),
1073),
1074),
1075),
1076),
1077),
1078),
1079),
1080),
1081),
1082),
1083),
1084),
1085),
1086),
1087),
1088),
1089),
1090),
1091),
1092),
1093),
1094),
1095),
1096),
1097),
1098),
1099),
1100),
1101),
1102),
1103),
1104),
1105),
1106),
1107),
1108),
1109),
1110),
1111),
1112),
1113),
1114),
1115),
1116),
1117),
1118),
1119),
1120),
1121),
1122),
1123),
1124),
1125),
1126),
1127),
1128),
1129),
1130),
1131),
1132),
1133),
1134),
1135),
1136),
1137),
1138),
1139),
1140),
1141),
1142),
1143),
1144),
1145),
1146),
1147),
1148),
1149),
1150),
1151),
1152),
1153),
1154),
1155),
1156),
1157),
1158),
1159),
1160),
1161),
1162),
1163),
1164),
1165),
1166),
1167),
1168),
1169),
1170),
1171),
1172),
1173),
1174),
1175),
1176),
1177),
1178),
1179),
1180),
1181),
1182),
1183),
1184),
1185),
1186),
1187),
1188),
1189),
1190),
1191),
1192),
1193),
1194),
1195),
1196),
1197),
1198),
1199),
1200),
1201),
1202),
1203),
1204),
1205),
1206),
1207),
1208),
1209),
1210),
1211),
1212),
1213),
1214),
1215),
1216),
1217),
1218),
1219),
1220),
1221),
1222),
1223),
1224),
1225),
1226),
1227),
1228),
1229),
1230),
1231),
1232),
1233),
1234),
1235),
1236),
1237),
1238),
1239),
1240),
1241),
1242),
1243),
1244),
1245),
1246),
1247),
1248),
1249),
1250),
1251),
1252),
1253),
1254),
1255),
1256),
1257),
1258),
1259),
1260),
1261),
1262),
1263),
1264),
1265),
1266),
1267),
1268),
1269),
1270),
1271),
1272),
1273),
1274),
1275),
1276),
1277),
1278),
1279),
1280),
1281),
1282),
1283),
1284),
1285),
1286),
1287),
1288),
1289),
1290),
1291),
1292),
1293),
1294),
1295),
1296),
1297),
1298),
1299),
1300),
1301),
1302),
1303),
1304),
1305),
1306),
1307),
1308),
1309),
1310),
1311),
1312),
1313),
1314),
1315),
1316),
1317),
1318),
1319),
1320),
1321),
1322),
1323),
1324),
1325),
1326),
1327),
1328),
1329),
1330),
1331),
1332),
1333),
1334),
1335),
1336),
1337),
1338),
1339),
1340),
1341),
1342),
1343),
1344),
1345),
1346),
1347),
1348),
1349),
1350),
1351),
1352),
1353),
1354),
1355),
1356),
1357),
1358),
1359),
1360),
1361),
1362),
1363),
1364),
1365),
1366),
1367),
1368),
1369),
1370),
1371),
1372),
1373),
1374),
1375),
1376),
1377),
1378),
1379),
1380),
1381),
1382),
1383),
1384),
1385),
1386),
1387),
1388),
1389),
1390),
1391),
1392),
1393),
1394),
1395),
1396),
1397),
1398),
1399),
1400),
1401),
1402),
1403),
1404),
1405),
1406),
1407),
1408),
1409),
1410),
1411),
1412),
1413),
1414),
1415),
1416),
1417),
1418),
1419),
1420),
1421),
1422),
1423),
1424),
1425),
1426),
1427),
1428),
1429),
1430),
1431),
1432),
1433),
1434),
1435),
1436),
1437),
1438),
1439),
1440),
1441),
1442),
1443),
1444),
1445),
1446),
1447),
1448),
1449),
1450),
1451),
1452),
1453),
1454),
1455),
1456),
1457),
1458),
1459),
1460),
1461),
1462),
1463),
1464),
1465),
1466),
1467),
1468),
1469),
1470),
1471),
1472),
1473),
1474),
1475),
1476),
1477),
1478),
1479),
1480),
1481),
1482),
1483),
1484),
1485),
1486),
1487),
1488),
1489),
1490),
1491),
1492),
1493),
1494),
1495),
1496),
1497),
1498),
1499),
1500),
1501),
1502),
1503),
1504),
1505),
1506),
1507),
1508),
1509),
1510),
1511),
1512),
1513),
1514),
1515),
1516),
1517),
1518),
1519),
1520),
1521),
1522),
1523),
1524),
1525),
1526),
1527),
1528),
1529),
1530),
1531),
1532),
1533),
1534),
1535),
1536),
1537),
1538),
1539),
1540),
1541),
1542),
1543),
1544),
1545),
1546),
1547),
1548),
1549),
1550),
1551),
1552),
1553),
1554),
1555),
1556),
1557),
1558),
1559),
1560),
1561),
1562),
1563),
1564),
1565),
1566),
1567),
1568),
1569),
1570),
1571),
1572),
1573),
1574),
1575),
1576),
1577),
1578),
1579),
1580),
1581),
1582),
1583),
1584),
1585),
1586),
1587),
1588),
1589),
1590),
1591),
1592),
1593),
1594),
1595),
1596),
1597),
1598),
1599),
1600),
1601),
1602),
1603),
1604),
1605),
1606),
1607),
1608),
1609),
1610),
1611),
1612),
1613),
1614),
1615),
1616),
1617),
1618),
1619),
1620),
1621),
1622),
1623),
1624),
1625),
1626),
1627),
1628),
1629),
1630),
1631),
1632),
1633),
1634),
1635),
1636),
1637),
1638),
1639),
1640),
1641),
1642),
1643),
1644),
1645),
1646),
1647),
1648),
1649),
1650),
1651),
1652),
1653),
1654),
1655),
1656),
1657),
1658),
1659),
1660),
1661),
1662),
1663),
1664),
1665),
1666),
1667),
1668),
1669),
1670),
1671),
1672),
1673),
1674),
1675),
1676),
1677),
1678),
1679),
1680),
1681),
1682),
1683),
1684),
1685),
1686),
1687),
1688),
1689),
1690),
1691),
1692),
1693),
1694),
1695),
1696),
1697),
1698),
1699),
1700),
1701),
1702),
1703),
1704),
1705),
1706),
1707),
1708),
1709),
1710),
1711),
1712),
1713),
1714),
1715),
1716),
1717),
1718),
1719),
1720),
1721),
1722),
1723),
1724),
1725),
1726),
1727),
1728),
1729),
1730),
1731),
1732),
1733),
1734),
1735),
1736),
1737),
1738),
1739),
1740),
1741),
1742),
1743),
1744),
1745),
1746),
1747),
1748),
1749),
1750),
1751),
1752),
1753),
1754),
1755),
1756),
1757),
1758),
1759),
1760),
1761),
1762),
1763),
1764),
1765),
1766),
1767),
1768),
1769),
1770),
1771),
1772),
1773),
1774),
1775),
1776),
1777),
1778),
1779),
1780),
1781),
1782),
1783),
1784),
1785),
1786),
1787),
1788),
1789),
1790),
1791),
1792),
1793),
1794),
1795),
1796),
1797),
1798),
1799),
1800),
1801),
1802),
1803),
1804),
1805),
1806),
1807),
1808),
1809),
1810),
1811),
1812),
1813),
1814),
1815),
1816),
1817),
1818),
1819),
1820),
1821),
1822),
1823),
1824),
1825),
1826),
1827),
1828),
1829),
1830),
1831),
1832),
1833),
1834),
1835),
1836),
1837),
1838),
1839),
1840),
1841),
1842),
1843),
1844),
1845),
1846),
1847),
1848),
1849),
1850),
1851),
1852),
1853),
1854),
1855),
1856),
1857),
1858),
1859),
1860),
1861),
1862),
1863),
1864),
1865),
1866),
1867),
1868),
1869),
1870),
1871),
1872),
1873),
1874),
1875),
1876),
1877),
1878),
1879),
1880),
1881),
1882),
1883),
1884),
1885),
1886),
1887),
1888),
1889),
1890),
1891),
1892),
1893),
1894),
1895),
1896),
1897),
1898),
1899),
1900),
1901),
1902),
1903),
1904),
1905),
1906),
1907),
1908),
1909),
1910),
1911),
1912),
1913),
1914),
1915),
1916),
1917),
1918),
1919),
1920),
1921),
1922),
1923),
1924),
1925),
1926),
1927),
1928),
1929),
1930),
1931),
1932),
1933),
1934),
1935),
1936),
1937),
1938),
1939),
1940),
1941),
1942),
1943),
1944),
1945),
1946),
1947),
1948),
1949),
1950),
1951),
1952),
1953),
1954),
1955),
1956),
1957),
1958),
1959),
1960),
1961),
1962),
1963),
1964),
1965),
1966),
1967),
1968),
1969),
1970),
1971),
1972),
1973),
1974),
1975),
1976),
1977),
1978),
1979),
1980),
1981),
1982),
1983),
1984),
1985),
1986),
1987),
1988),
1989),
1990),
1991),
1992),
1993),
1994),
1995),
1996),
1997),
1998),
1999),
2000),
2001),
2002),
2003),
2004),
2005),
2006),
2007),
2008),
2009),
2010),
2011),
2012),
2013),
2014),
2015),
2016),
2017),
2018),
2019),
2020),
2021),
2022),
2023),
2024),
2025),
2026),
2027),
2028),
2029),
2030),
2031),
2032),
2033),
2034),
2035),
2036),
2037),
2038),
2039),
2040),
2041),
2042),
2043),
2044),
2045),
2046),
2047),
2048),
2049),
2050),
2051),
2052),
2053),
2054),
2055),
2056),
2057),
2058),
2059),
2060),
2061),
2062),
2063),
2064),
2065),
2066),
2067),
2068),
2069),
2070),
2071),
2072),
2073),
2074),
2075),
2076),
2077),
2078),
2079),
2080),
2081),
2082),
2083),
2084),
2085),
2086),
2087),
2088),
2089),
2090),
2091),
2092),
2093),
2094),
2095),
2096),
2097),
2098),
2099),
2100),
2101),
2102),
2103),
2104),
2105),
2106),
2107),
2108),
2109),
2110),
2111),
2112),
2113),
2114),
2115),
2116),
2117),
2118),
2119),
2120),
2121),
2122),
2123),
2124),
2125),
2126),
2127),
2128),
2129),
2130),
2131),
2132),
2133),
2134),
2135),
2136),
2137),
2138),
2139),
2140),
2141),
2142),
2143),
2144),
2145),
2146),
2147),
2148),
2149),
2150),
2151),
2152),
2153),
2154),
2155),
2156),
2157),
2158),
2159),
2160),
2161),
2162),
2163),
2164),
2165),
2166),
2167),
2168),
2169),
2170),
2171),
2172),
2173),
2174),
2175),
2176),
2177),
2178),
2179),
2180),
2181),
2182),
2183),
2184),
2185),
2186),
2187),
2188),
2189),
2190),
2191),
2192),
2193),
2194),
2195),
2196),
2197),
2198),
2199),
2200),
2201),
2202),
2203),
2204),
2205),
2206),
2207),
2208),
2209),
2210),
2211),
2212),
2213),
2214),
2215),
2216),
2217),
2218),
2219),
2220),
2221),
2222),
2223),
2224),
2225),
2226),
2227),
2228),
2229),
2230),
2231),
2232),
2233),
2234),
2235),
2236),
2237),
2238),
2239),
2240),
2241),
2242),
2243),
2244),
2245),
2246),
2247),
2248),
2249),
2250),
2251),
2252),
2253),
2254),
2255),
2256),
2257),
2258),
2259),
2260),
2261),
2262),
2263),
2264),
2265),
2266),
2267),
2268),
2269),
2270),
2271),
2272),
2273),
2274),
2275),
2276),
2277),
2278),
2279),
2280),
2281),
2282),
2283),
2284),
2285),
2286),
2287),
2288),
2289),
2290),
2291),
2292),
2293),
2294),
2295),
2296),
2297),
2298),
2299),
2300),
2301),
2302),
2303),
2304),
2305),
2306),
2307),
2308),
2309),
2310),
2311),
2312),
2313),
2314),
2315),
2316),
2317),
2318),
2319),
2320),
2321),
2322),
2323),
2324),
2325),
2326),
2327),
2328),
2329),
2330),
2331),
2332),
2333),
2334),
2335),
2336),
2337),
2338),
2339),
2340),
2341),
2342),
2343),
2344),

```

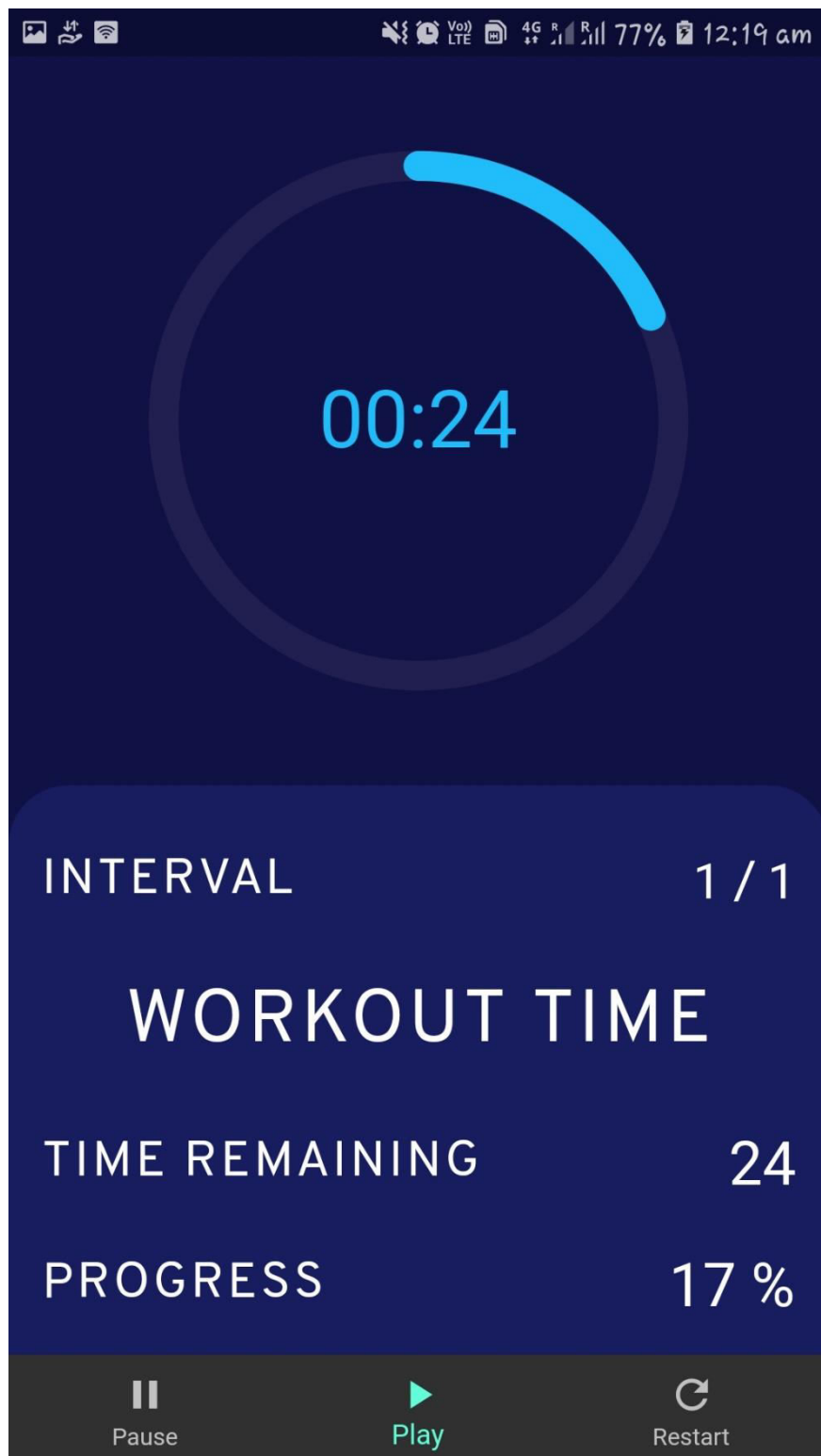


```
309 ),
310 bottomNavigationBar: BottomNavigationBar(
311   currentIndex: _currentIndex,
312   type: BottomNavigationBarType.fixed,
313   items: <BottomNavigationBarItem>[
314     BottomNavigationBarItem(
315       icon: Icon(Icons.pause),
316       label: 'Pause',
317     ),
318     BottomNavigationBarItem(icon: Icon(Icons.play_arrow), label: 'Play'),
319     BottomNavigationBarItem(
320       icon: Icon(Icons.refresh_outlined), label: 'Restart'),
321   ],
322   onTap: (index) {
323     if (ctr != 0) {
324       setState(() {
325         _currentIndex = index;
326       });
327       switch (index) {
328         case 0:
329           _controller.pause();
330           _countdownController.pause();
331           _cntdwnController.pause();
332           break;
333         case 1:
334           _controller.resume();
335           _countdownController.resume();
336           _cntdwnController.resume();
337           break;
338         case 2:
339           ctr = 1;
340           currentSetNumber = 1;
341           workout = true;
342           rest = false;
343           _controller.restart();
344           _countdownController.restart();
345           _cntdwnController.restart();
346       }
347     }
348   }
349 ),
350 ],
351 ),
352 },
353 ],
354 }
```



```
327 switch (index) {
328   case 0:
329     _controller.pause();
330     _countdownController.pause();
331     _cntdwnController.pause();
332     break;
333   case 1:
334     _controller.resume();
335     _countdownController.resume();
336     _cntdwnController.resume();
337     break;
338   case 2:
339     ctr = 1;
340     currentSetNumber = 1;
341     workout = true;
342     rest = false;
343     _controller.restart();
344     _countdownController.restart();
345     _cntdwnController.restart();
346   }
347 }
348 }
349 },
350 ],
351 ),
352 },
353 ],
354 }
```


Screenshots :





00:21

INTERVAL

2 / 2

TAKE A BREAK

TIME REMAINING

51

PROGRESS

38 %



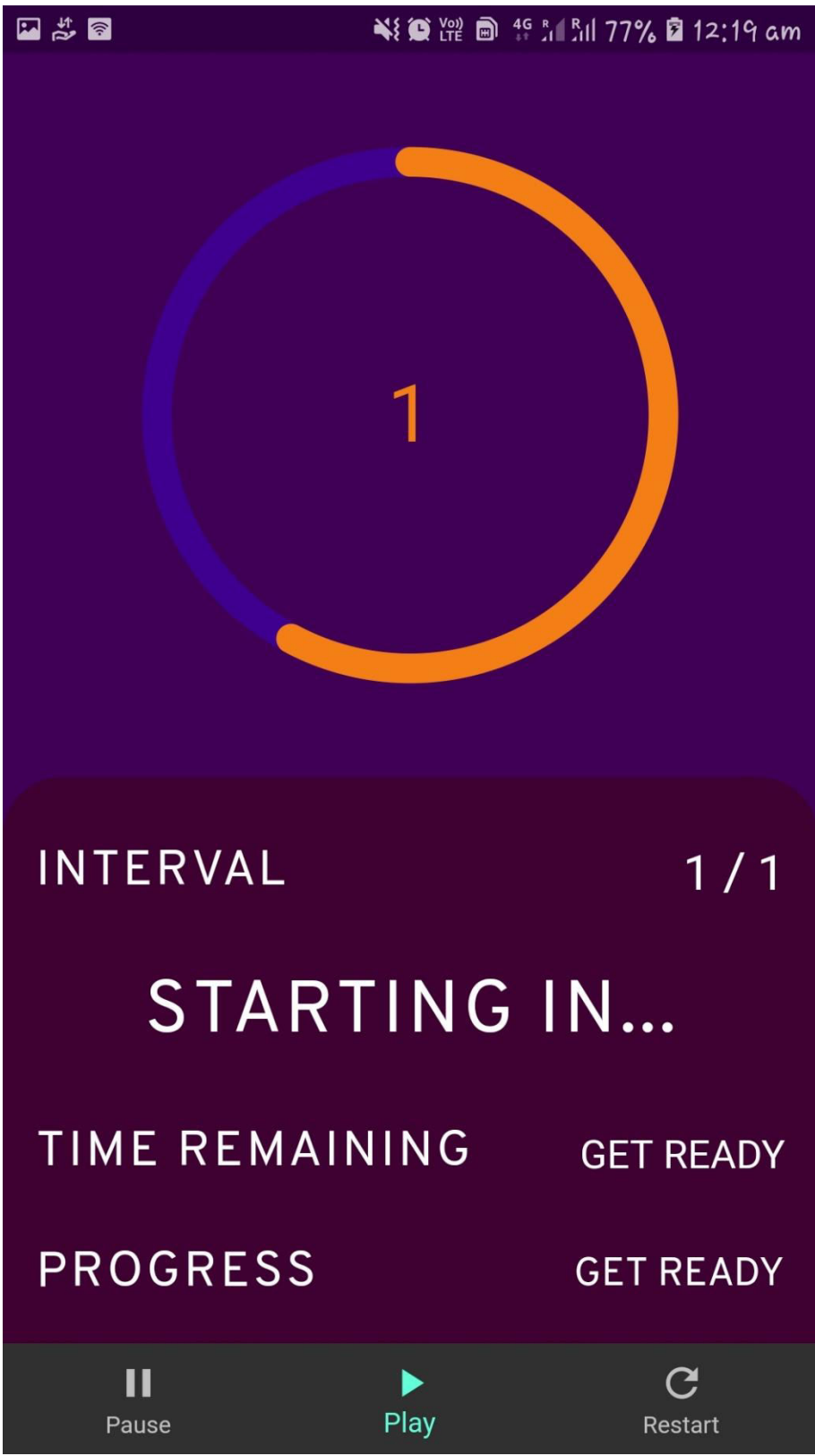
Pause



Play



Restart



12:19 am 77% 4G R R



INTERVAL

1 / 1

STARTING IN...

TIME REMAINING

GET READY

PROGRESS

GET READY

Pause

Play

Restart

76%12:18 am

TIME FOR WORKOUT

HH

MM

SS

0

0

0

1

1

1

BREAK

HH

MM

SS

0

0

0

1

1

1

SETS

-

1

+

VoLTE

4G

76%

12:18 am

Timer

TIME FOR WORKOUT

HH

MM

SS

0

0

0

1

1

1

BREAK

HH

MM

SS

0

0

0

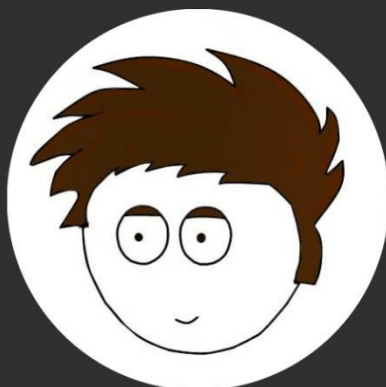
1

1

1

SETS

This app is developed by :-



Aman Kumar

Phone: +91943595XXXX

Email: aman.kumar_cs18@gla.ac.in



Ayurish Chandana

Phone: +91948456XXXX

Email: ayurish.chandna_cs18@gla.ac.in

Conclusion

High-intensity interval training is currently among top fitness trends in the world. It is a training technique that requires high effort through intense burst of exercise followed by short rest period. According to recent researches, supervised HIIT can lead to beneficial impact on athletic performance and fat loss but should not be used by an inexperienced individual as well as elder population.

It is important to have Interval training because it trains your heart to pump more blood to the muscles and it trains your muscles to extract that oxygen more efficiently, making all your other workouts easier to handle.

If we add exercise meditation or yoga in our daily routine it can do wonders. Adopting a healthier lifestyle is a long-term investment but it benefits a lot when during your old age.