

**1. Project Name:** Git Fighter

**2. Group Information:**

Member 1:

Pavankumar Bhadane

[pavankumar.bhadane20@vit.edu](mailto:pavankumar.bhadane20@vit.edu)

CS - C

12010226

Member 2:

Shreya Roy

[shreya.roy20@vit.edu](mailto:shreya.roy20@vit.edu)

E&TC - D

12010171

Member 3:

Shubham Shah

[shubham.shah20@vit.edu](mailto:shubham.shah20@vit.edu)

CS - D

12010796

**3. Project GitHub Link:** [https://github.com/suhas-kadu/robospark-2021-FT-Git\\_Fighter.git](https://github.com/suhas-kadu/robospark-2021-FT-Git_Fighter.git)

#### 4. Project Workflow:

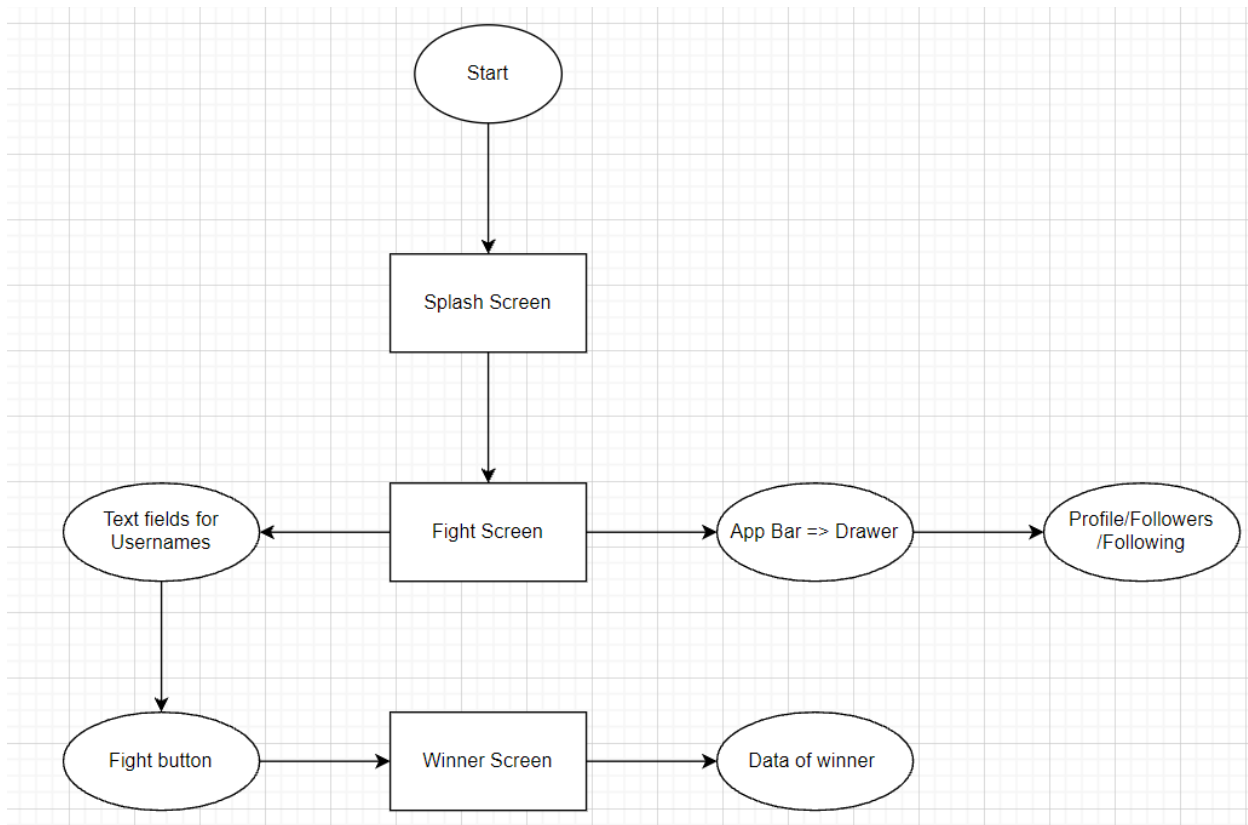


Fig. 01 App Workflow

#### 5. Problems Faced:

- Learning everything on our own and implementing them the right way took time.
- Use of GitHub for pushing the code to our respective branches was quite new for us.
- Visual Studio took a long time to build the project on the mobile emulator every time.
- We were facing a glitch in the winner screen.

#### 6. Alternative Solutions found for problems mentioned above:

- We referred to Dr.Angela's playlist on Udemy for learning flutter.
- Our mentor taught us how to use features of Github not known to us in projects.
- Some of the features to be merged later were implemented individually and run on Chrome which saved time.
- We removed the glitch in the winner screen by referring to stack overflow.

## 7. Code snippet ss:

Main.dart :

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

import 'package:flutter/material.dart';

import 'package:git_fighter/screens/splash_screen.dart';

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

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      title: 'Git Fighter',

      theme: ThemeData(

        brightness: Brightness.dark, primaryColor:

Colors.yellowAccent),

      darkTheme: ThemeData.dark(),

      home: SplashScreen());

  }

}
```

Splash Screen Code:

```
import 'dart:async';
```

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

import 'package:font_awesome_flutter/font_awesome_flutter.dart';

import 'fight_screen.dart';

class SplashScreen extends StatefulWidget {

  @override

  _SplashScreenState createState() => _SplashScreenState();

}

class _SplashScreenState extends State<SplashScreen> {

  @override

  void initState() {

    super.initState();

    Timer(Duration(seconds: 5), () {

      Navigator.pushReplacement(context,

        MaterialPageRoute(builder: (BuildContext context) =>
FightScreen()));

    });

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      body: Stack(
```

```
fit: StackFit.expand,

children: [

  Container(

    decoration: BoxDecoration(color: Colors.white),

  ),

  Column(

    mainAxisAlignment: MainAxisAlignment.start,

    children: [

      Expanded(

        flex: 2,

        child: SingleChildScrollView(

          child: Column(

            mainAxisAlignment: MainAxisAlignment.center,

            children: [

              CircleAvatar(

                backgroundColor: Colors.white,

                radius: 150.0,

                child: FaIcon(

                  FontAwesomeIcons.githubSquare,

                  size: 140.0,

                  color: Colors.black,

                ),

              ),

            ],

          ),

        ),

      ),

    ],

  ),

],
```

```
        Padding(  
          padding: EdgeInsets.only(top: 10.0),  
        ),  
        Text(  
          "Git Fighter",  
          style: TextStyle(  
            color: Colors.grey[900],  
            fontSize: 34.0,  
            fontWeight: FontWeight.bold),  
        )  
      ],  
    ),  
  ),  
),  
Expanded(  
  flex: 1,  
  child: Column(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: [  
      Text(  
        "By TRF Team Flutter",  
        style: TextStyle(  
          color: Colors.grey[900],
```

```

        fontSize: 24.0,

        fontWeight: FontWeight.bold),

        // Padding(padding: EdgeInsets.only(top: 10))

    )

    ],

  ))

  ],

),

],

),

);

}

}

```

Fight Screen code:

```

import 'dart:convert';

import 'package:flutter/material.dart';

import 'package:git_fighter/screens/winner_screen.dart';

import 'package:http/http.dart' as http;

class FightScreen extends StatefulWidget {

  @override

```

```

    _FightScreenState createState() => _FightScreenState();
}

class _FightScreenState extends State<FightScreen> {

    final TextEditingController txtUser1Controller =
    TextEditingController();

    final TextEditingController txtUser2Controller =
    TextEditingController();

    String winner;

    var responseUser1, responseUser2;

    @override

    Widget build(BuildContext context) {

        return Scaffold(

            appBar: AppBar(

                title: Text('Git Fighter'),

            ),

            drawer: Drawer(

                child: Column(

                    children: [

                        Container(

                            width: double.infinity,

```



```
padding: EdgeInsets.all(20),

color: Colors.grey[800],

child: Center(

  child: Column(

    children: [

      Container(

        width: 100,

        height: 100,

        margin: EdgeInsets.only(top: 30, bottom: 10),

        decoration: BoxDecoration(

          shape: BoxShape.circle,

          image: DecorationImage(

            image: NetworkImage(

              'https://avatars.githubusercontent.com/u/90468365?s=400&u=b7c419ea83147ff256e2902208183cd7401c2aa2&v=4'),

            fit: BoxFit.fill),

          ),

        Text(

          'Pavan Bhadane',

          style: TextStyle(fontSize: 22, color: Colors.white),

        ),

        Text(

          'pavan49719@gmail.com',
```

```
        style: TextStyle(fontSize: 15, color: Colors.white),
      ),
    ],
  ),
),
Container(
  width: double.infinity,
  padding: EdgeInsets.all(20),
  color: Colors.grey[700],
  child: Center(
    child: Text(
      'Followers: 48',
      style: TextStyle(fontSize: 20, color: Colors.white),
    ),
  ),
),
Container(
  width: double.infinity,
  padding: EdgeInsets.all(20),
  color: Colors.grey[700],
  child: Center(
    child: Text(
      'Following: 21',
```

```
        style: TextStyle(fontSize: 20, color: Colors.white),
      )),
    ),
    Container(
      width: double.infinity,
      padding: EdgeInsets.all(20),
      color: Colors.grey[700],
      child: Center(
        child: Text(
          'Repos : 11',
          style: TextStyle(fontSize: 20, color: Colors.white),
        )),
      )
    ],
  ),
),
body: Center(
  child: Padding(
    padding: EdgeInsets.all(10),
    child: SingleChildScrollView(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.start,
        children: [
```

```
        SizedBox(height: 60),

        const Text(

          'Git Fighter',

          style: TextStyle(

            color: Colors.greenAccent,

            fontSize: 60,

            fontWeight: FontWeight.bold,

          ),

          textAlign: TextAlign.center,

        ),

        SizedBox(height: 90),

        TextField(

          controller: txtUser1Controller,

          decoration: InputDecoration(

            border: OutlineInputBorder(),

            labelText: 'User 1',

            hintText: 'Enter username'),

        ),

        SizedBox(height: 60),

        InkWell(

          onTap: () {

            startFight();

          },

        ),
```

```
borderRadius: BorderRadius.circular(20),

// hoverColor: Colors.indigoAccent[200],

child: Container(

  clipBehavior: Clip.antiAlias,

  alignment: Alignment.center,

  width: 120,

  height: 120,

  decoration: BoxDecoration(

    borderRadius: BorderRadius.circular(60),

    color: Colors.indigoAccent,

  ),

  child: Row(

    mainAxisAlignment: MainAxisAlignment.center,

    children: [

      Image.asset(

        'images/fight.png',

        width: 120,

        height: 120,

        alignment: Alignment.center,

        fit: BoxFit.contain,

      ),

    ],

  ),

),
```



```

        ],
        ),
    )),
);
}

```

```

void startFight() async {

    if (txtUserController1.text == "" ||

        txtUserController2.text == "" ||

        txtUserController1.text == null ||

        txtUserController2.text == null) {

        // createAlertDialog(context);

        ScaffoldMessenger.of(context).showSnackBar(SnackBar(

            content: Text("Please enter username"),

        ));

    } else {

        responseUser1 = await http

            .get(Uri.http('api.github.com',

'users/${txtUserController1.text}'));

        responseUser2 = await http

            .get(Uri.http('api.github.com',

'users/${txtUserController2.text}'));

        var jsonData1 = jsonDecode(responseUser1.body);

        var jsonData2 = jsonDecode(responseUser2.body);
    }
}

```

```
if (jsonData1["message"] == "Not Found" ||
    jsonData2["message"] == "Not Found") {
    ScaffoldMessenger.of(context).showSnackBar(SnackBar(
        content: Text("Enter valid username"),
    ));
} else {
    int user1Followers = jsonData1['followers'];
    int user2Followers = jsonData2['followers'];
    int user1Following = jsonData1['following'];
    int user2Following = jsonData2['following'];
    setState(
        () {
            if ((user1Followers + user1Following) >
                (user2Followers + user2Following))
                winner = txtUser1Controller.text;
            else
                winner = txtUser2Controller.text;
            print("winner is $winner");
            Navigator.push(
                context,
                MaterialPageRoute(
                    builder: (BuildContext context) => Winner(winner),
                ),
            ),
        },
    );
}
```



```

        );

        },

    );

}

}

}

}

```

#### Winner Screen Code:

```

import 'dart:convert';

import 'package:flutter/material.dart';

import 'package:git_fighter/models/winner_data.dart';

import 'package:git_fighter/widget/all_confetti_widget.dart';

import 'package:http/http.dart' as http;

class Winner extends StatefulWidget {

  String winner;

  Winner(String winner) {

    this.winner = winner;

  }

```

```

@override

_WinnerState createState() => _WinnerState(winner);
}

class _WinnerState extends State<Winner> {

  WinnerData winnerData = new WinnerData();

  String winner;

  _WinnerState(String winner) {

    this.winner = winner;

  }

  @override

  void initState() {

    super.initState();

    getProfileData();

    setState(() {});

  }

  void getProfileData() async {

    var response = await http.get(Uri.http('api.github.com',
'users/$winner'));

    var jsonData = await jsonDecode(response.body);

```

```

    setState(() {
        winnerData = WinnerData.fromJson(jsonData);
    });
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            centerTitle: true,
            title: Center(child: Text("Winner : $winner ")),
        ),
        body: Center(
            child: (SingleChildScrollView(
                child: Padding(
                    padding: const EdgeInsets.all(10.0),
                    child: Column(
                        crossAxisAlignment: CrossAxisAlignment.stretch,
                        mainAxisAlignment: MainAxisAlignment.start,
                        children: <Widget>[
                            AllConfettiWidget(),
                            SizedBox(height: 20),
                            Container(

```

```
        height: 200.0,

        width: 200.0,

        child: winnerData.avatarUrl != null

            ? CircleAvatar(

                backgroundImage:

                    NetworkImage("${winnerData.avatarUrl}"),

                radius: 20,

                backgroundColor: Colors.transparent,

                foregroundColor: Colors.transparent,

            )

            : Container(),

    ),

    Padding(

        padding: EdgeInsets.fromLTRB(10, 0, 0, 0),

    ),

    Text(

        "Followers : ${winnerData.followers}",

        style: TextStyle(fontSize: 30),

        textAlign: TextAlign.center,

    ),

    Text(

        "Following : ${winnerData.following}",

        style: TextStyle(fontSize: 30),
```

```
        textAlign: TextAlign.center,

    ),

    SizedBox(height: 20),

    Container(

        alignment: Alignment.topLeft,

        width: 2000.0,

        height: 6.0,

        color: Colors.blueAccent,

    ),

    SizedBox(height: 20),

    Text(

        "Repos : ${winnerData.publicRepos}",

        style: TextStyle(fontSize: 24),

    ),

    SizedBox(height: 20),

    winnerData.bio != null

        ? Text(

            "Bio : ${winnerData.bio}",

            style: TextStyle(

                fontSize: 24,

            ),

        )

        : Container(),
```

```
Padding(

    padding: EdgeInsets.only(top: 30),

),

winnerData.company != null

    ? Text(

        "Company : " "${winnerData.company}",

        style: TextStyle(

            fontSize: 24,

        ),

    )

    : Container(),

Padding(

    padding: EdgeInsets.only(top: 30),

),

winnerData.location != null

    ? Text(

        "Location : " "${winnerData.location}",

        style: TextStyle(

            fontSize: 24,

        ),

    )

    : Container(),

Padding(
```

```

        padding: EdgeInsets.only(top: 30),
      ),
      Container(
        alignment: Alignment.topLeft,
        width: 2000.0,
        height: 6.0,
        color: Colors.blueAccent,
      ),
    ],
  ),
)),
),
);
}
}

```

### Animation Code:

```

import 'dart:math';

import 'package:confetti/confetti.dart';

import 'package:flutter/material.dart';

import 'package:confetti/src/enums/confetti_controller_state.dart';

```

```
class AllConfettiWidget extends StatefulWidget {  
  
  const AllConfettiWidget({  
  
    Key key,  
  
  }) : super(key: key);  
  
  @override  
  
  _AllConfettiWidgetState createState() => _AllConfettiWidgetState();  
}
```

```
class _AllConfettiWidgetState extends State<AllConfettiWidget> {  
  
  ConfettiController controller;  
  
  @override  
  
  void initState() {  
  
    super.initState();  
  
  
    controller = ConfettiController(duration: Duration(seconds: 5));  
  
    controller.play();  
  
  }
```

```
  static final double right = 0;
```

```
  static final double down = pi / 2;
```

```
  static final double left = pi;
```



```
static final double top = -pi / 2;

final double blastDirection = left;

@override
Widget build(BuildContext context) {

  return GestureDetector(

    behavior: HitTestBehavior.opaque,

    onTap: () {

      if (controller.state == ConfettiControllerState.playing) {

        controller.stop();

      } else {

        controller.play();

      }

    },

    child: Stack(

      children: [

        buildConfetti(),

      ],

    ),

  );

}
```

```
Widget buildConfetti() => Align(

  alignment: Alignment.center,

  child: ConfettiWidget(

    confettiController: controller,

    colors: [

      Colors.red,

      Colors.blue,

      Colors.orange,

      Colors.purple,

      Colors.lightBlue,

    ],

    //blastDirection: blastDirection,

    blastDirectionality: BlastDirectionality.explosive,

    shouldLoop: true,

    emissionFrequency: 0.05,

    numberOfParticles: 5,

    gravity: 0.2,

    maxBlastForce: 2,

    minBlastForce: 1,

    particleDrag: 0.1,

  ),

);

}
```

## 8. Output:



Fig. 02: Splash Screen

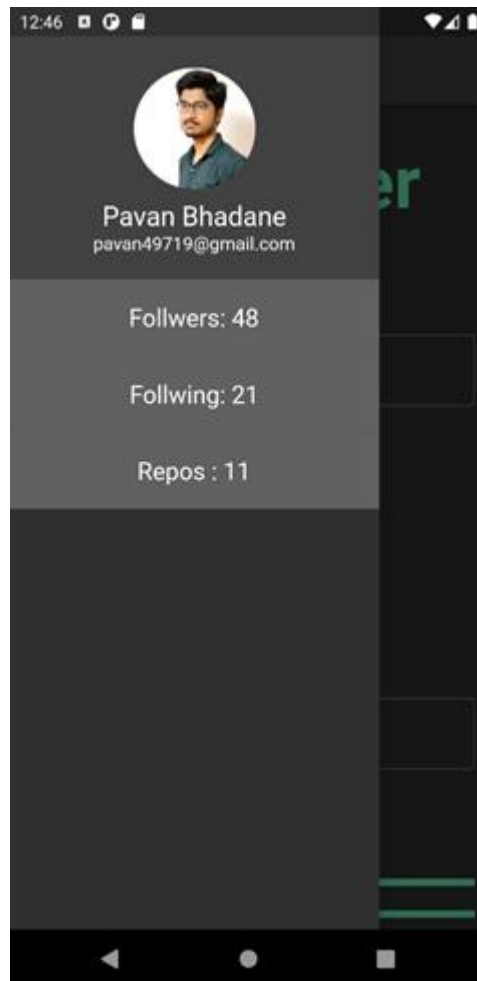


Fig. 03: Drawer

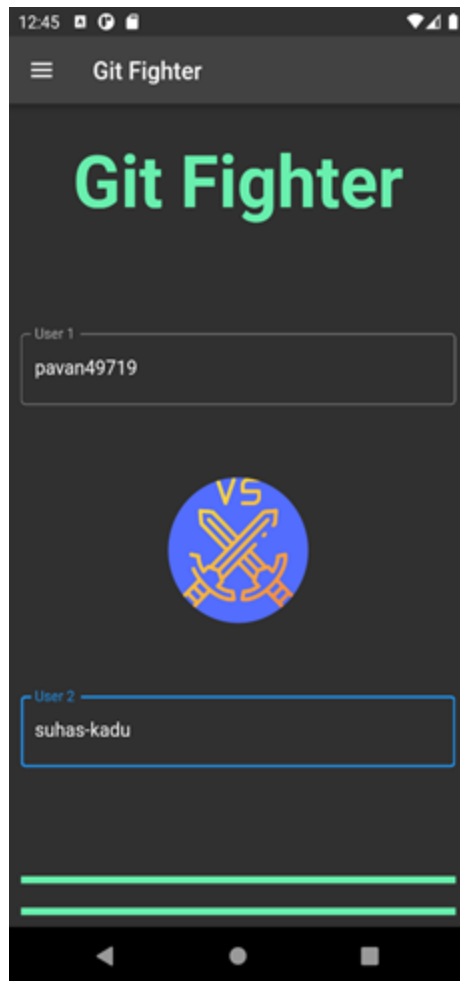


Fig. 04: Fight Screen

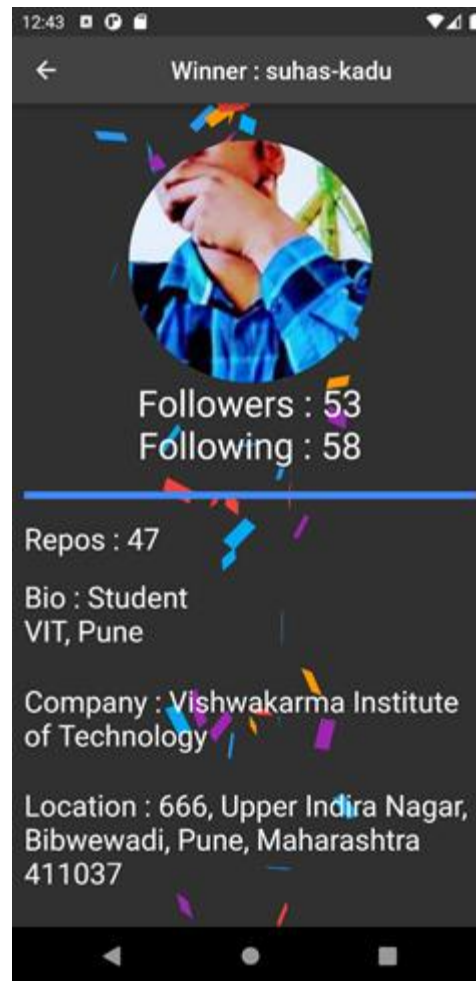


Fig. 05: Winner Screen