

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and
Engineering

By

Pravenesh B
(Reg. No. 38110428)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMIL NADU**

NOVEMBER 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Pravenesh B (38110428)** who carried out the project entitled “**Quick Aid**” under my supervision from June 2021 to November 2021.

Internal Guide

Dr. N. Srinivasan

Head of the Department

Dr. L. LAKSHMANAN M.E., Ph.D.,

Dr.S.VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **Pravenesh B** hereby declare that the Project Report entitled **Quickaid** done by me under the guidance of **Dr. N. Srinivasan** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the Board **of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean**, School of Computing , **Dr.S.Vigneshwari M.E., Ph.D., and Dr.L.Lakshmanan M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. Mary Paul Selvan** for her valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project

ABSTRACT

Our Aim is to design and create a product rental app that enables the user to rent a product with people in their neighbourhood or require a service, so that the product can be used by the requestor. This system could also help in building an amicable relationship among the neighbours and solve the problem of people not owning non-essential things readily.

Table of content

Chapter no	Title	Page no
	ABSTRACT	i
	LIST OF Table	ii
1	Introduction	1
1.1	Outline of the project	1
1.2	proposed system	1
1.3	Scope of the project	2
2	AIM & SCOPE OF THE PROJECT	3
2.2	AIM	3
2.1	Software requirement	3
2.2	Hardware requirement	3
3	EXPERIMENTAL OR MATERIALS AND METHODS	5
3.1	Flutter	5
3.2	Feature of Flutter	5
3.3	Architecture of Flutter Application	6
3.4	Flutter Packages	10
4	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	13

4.1	Discussion	13
4.2	Result	14
5	SUMMARY AND CONCLUSIONS	15
5.1	Reference	15
5.2	APPENDIX	15
5.2A	Screenshots	15
5.2B	Source code	20

Chapter 1

Introduction

1.1 Outline of the project

Renting and exchanging the goods is a method of using technology to bring people together" despite the geographical barriers The technology has been available for years but the acceptance of it was quite recent. Our project to make neighbourhoods close and earn money . It is made up of flutter made by Google. When we post the product ,other users can get the notification. If they have the product they can communicate with the user. Some may give the product for free rental for the neighbourhoods or pay service ,it can be chosen by the user. First This app sends the notification to a 6 km radius from the user given location. This app was designed to be very user friendly. This software can have further posence , such as delivering the product ,map location.

1.2 proposed system

Resolving the names of all the systems connected in a network and enlisting them.

- Used for communication between multiple systems enlisted in the resolved list.
- The GUI operates in two forms, the ist form & the chat form.
- The List form contains the names of all the systems connected to a network.
- The chat form makes the actual communication possible in the form of text.

Advantage :

- User friendly
- Fast replay
- More clients can use at a time
- Cross platform

Disadvantage :

- Can see post of only near by location
- Some product cannot be rented

1.3 scope of the project:

Where we rent the product from nearby neighbours ,so where they can earn money and make friends with unknown neighbours . By getting the product, the rental person can save money from buying it.

Chapter 2

AIM & SCOPE OF THE PROJECT

2.1 AIM

This is the simple product rental app where the person can rent all simple item with the neighbourhood .So they can save money instead of buying it .By renting the rarely used product they can also earn the money. In this process neighbour can know about ,they can be good friends also.

2.1 Software requirements

Software can be defined as programs which run on our computer. It acts as petrol in the vehicle. It provides the relationship between the human and a computer. It is very important to run software to function on the computer.

Various software are needed in this project for its development. Which are as follows-?

- Operating system- more than Windows 7
- Android studio
- Flutter

Languages : Dart language is used.

2.2 Hardware Requirements

In hardware requirements we require all those components which will provide us the platform for the development of the project.The minimum hardware required for the development of this project as follows-

- Ram minimum 128 MB
- Hard disk -minimum 5 GB

- Processor Pentium 3
- Floppy drive 1,44 inch

These all are the minimum hardware requirements required for our project. We want to make our project to be used in any. Type of computer therefore we have taken minimum configuration to a large extent. 128 MB ram is used so that we can execute our project in the least possible RAMS. Hard disk is used because the project takes less space to be executed or stored. Therefore the minimum hard disk is used. Other enhancements are according to the needs.

Chapter 3

EXPERIMENTAL OR MATERIALS AND METHODS

3.1 FLUTTER

Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia, and the web from a single codebase

Flutter is Google's **portable UI toolkit for crafting beautiful**, natively compiled applications for mobile, web, and desktop from a single codebase. Flutter works with existing code, is used by developers and organizations around the world, and is free and open source.

3.2 Features of Flutter

Flutter framework offers the following features to developers:

- Modern and reactive framework.
- Uses Dart programming language and it is very easy to learn.
- Fast development.
- Beautiful and fluid user interfaces.
- Huge widget catalog.
- Runs the same UI for multiple platforms.
- High performance application

Advantages of Flutter

Flutter comes with beautiful and customizable widgets for high performance and outstanding mobile application. It fulfills all the custom needs and requirements. Besides these, Flutter offers many more advantages as mentioned below:

- Dart has a large repository of software packages which lets you extend the

capabilities of your application.

- Developers need to write just a single code base for both applications (both Android and iOS platforms). Flutter may be extended to other platforms as well in the future.
- Flutter needs less testing. Because of its single code base, it is sufficient if we write automated tests once for both the platforms.
- Flutter's simplicity makes it a good candidate for fast development. Its customization capability and extendibility makes it even more powerful.
- With Flutter, developers have full control over the widgets and its layout.
- Flutter offers great developer tools, with amazing hot reload.

Disadvantages of Flutter

Despite its many advantages, flutter has the following drawbacks:

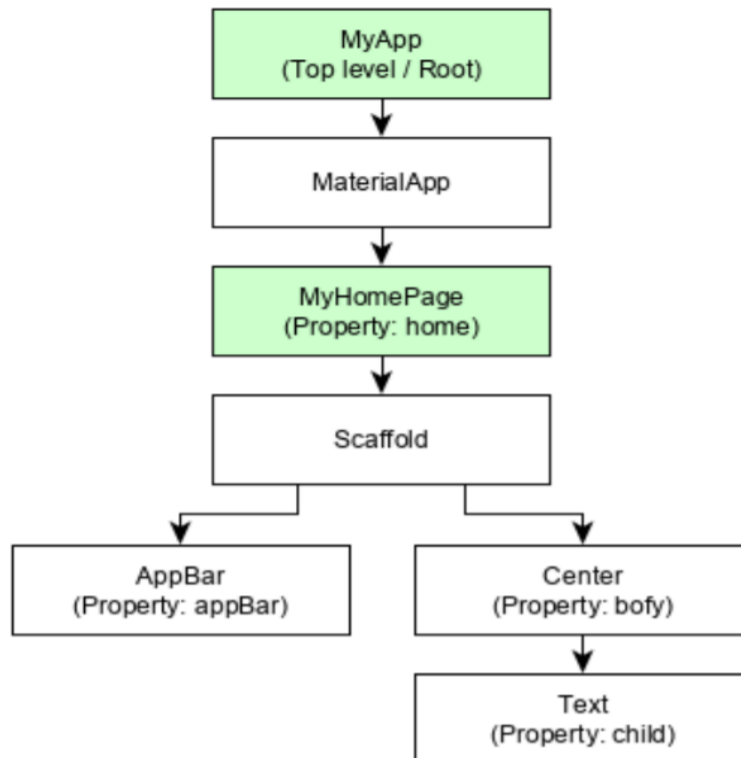
- Since it is coded in Dart language, a developer needs to learn a new language (though it is easy to learn).
- Modern frameworks try to separate logic and UI as much as possible but, in Flutter, user interface and logic is intermixed. We can overcome this using smart coding and using high level modules to separate user interface and logic.
- Flutter is yet another framework to create mobile applications. Developers are having a hard time choosing the right development tools in highly populated segments.

3.3 Flutter – Architecture of Flutter Application

In this chapter, let us discuss the architecture of the Flutter framework.

Widgets

The core concept of the Flutter framework is In Flutter, Everything is a widget. Widgets are basically user interface components used to create the user interface of the application. In Flutter, the application is itself a widget. The application is the top- level widget and its UI is built using one or more children (widgets), which again build using its children widgets. This composability feature helps us to create a user interface of any complexity. For example, the widget hierarchy of the hello world application (created in previous chapter) is as specified in the following diagram:



Here the following points are worth notable:

- MyApp is the user created widget and it is build using the Flutter native widget, MaterialApp.
- MaterialApp has a home property to specify the user interface of the home page, which is again a user created widget, MyHomePage.

- MyHomePage is build using another flutter native widget, Scaffold.
- Scaffold has two properties – body and appBar.
- body is used to specify its main user interface and appBar is used to specify its header user interface.
- Header UI is build using flutter native widget, AppBar and Body UI is build using Center widget.
- The Center widget has a property, Child, which refers to the actual content and it is build using Text widget.

Gestures

Flutter widgets support interaction through a special widget, GestureDetector. GestureDetector is an invisible widget having the ability to capture user interactions such as tapping, dragging, etc., of its child widget. Many native widgets of Flutter support interaction through the use of GestureDetector. We can also incorporate interactive feature into the existing widget by composing it with the GestureDetector widget. We will learn the gestures separately in the upcoming chapters.

Concept of State

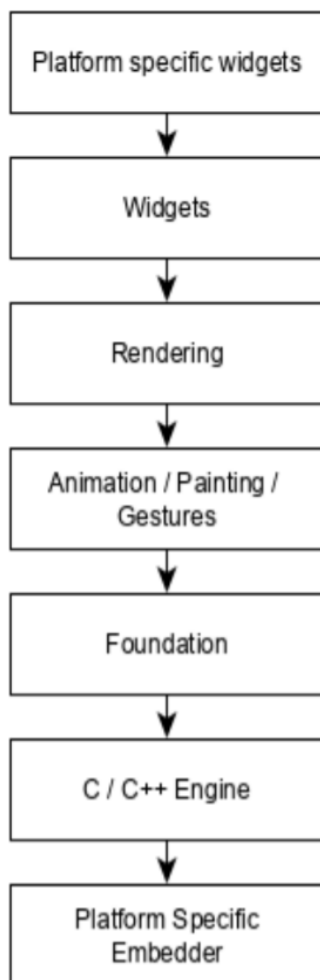
Flutter widgets support State maintenance by providing a special widget, StatefulWidget. Widget needs to be derived from StatefulWidget widget to support state maintenance and all other widget should be derived from StatelessWidget. Flutter widgets are reactive in native. This is similar to reactjs and StatefulWidget will be auto re-rendered whenever its internal state is changed. The re-rendering is optimized by finding the difference between old and new widget UI and rendering only the necessary changes.

Layers

The most important concept of the Flutter framework is that the framework is grouped into multiple categories in terms of complexity and clearly arranged in layers of

decreasing complexity. A layer is build using its immediate next level layer. The top most layer is widget specific to Android and iOS. The next layer has all flutter native widgets. The next layer is the Rendering layer, which is a low level renderer component and renders everything in the flutter app. Layers go down to core platform specific code.

The general overview of a layer in Flutter is specified in the below diagram:



The following points summarize the architecture of Flutter:

- In Flutter, everything is a widget and a complex widget is composed of already existing widgets.
- Interactive features can be incorporated whenever necessary using GestureDetector widget.

- The state of a widget can be maintained whenever necessary using StatefulWidget widget.
- Flutter offers layered design so that any layer can be programmed depending on the complexity of the task.

3.4 Flutter Packages

Dart's way of organizing and sharing a set of functionality is through Package. Dart Package is simply shareable libraries or modules. In general, the Dart Package is the same as that of the Dart Application except that the Dart Package does not have an application entry point, main.

The general structure of Package (consider a demo package, my_demo_package) is as below:

- lib/src/* : Private Dart code files.
- lib/my_demo_package.dart: Main Dart code file. It can be imported into an application as:

```
import 'package:my_demo_package/my_demo_package.dart'
```

- Other private code file may be exported into the main code file (my_demo_package.dart), if necessary as shown below:

```
export src/my_private_code.dart
```

- lib/* : Any number of Dart code files arranged in any custom folder structure. The code can be accessed as,

```
import 'package:my_demo_package/custom_folder/custom_file.dart'
```

- pubspec.yaml: Project specification, same as that of application.

All Dart code files in the Package are simply Dart classes and it does not have any special requirement for a Dart code to include it in a Package.

Types of Packages

Since Dart Packages are basically a small collection of similar functionality, it can be categorized based on its functionality.

Dart Package

Generic Dart code, which can be used in both web and mobile environments. For example, `english_words` is one such package which contains around 5000 words and has basic utility functions like `nouns` (list nouns in the English), `syllables` (specify number of syllables in a word).

Flutter Package

Generic Dart code, which depends on Flutter framework and can be used only in mobile environments. For example, `fluro` is a custom router for flutter. It depends on the Flutter framework.

Flutter Plugin

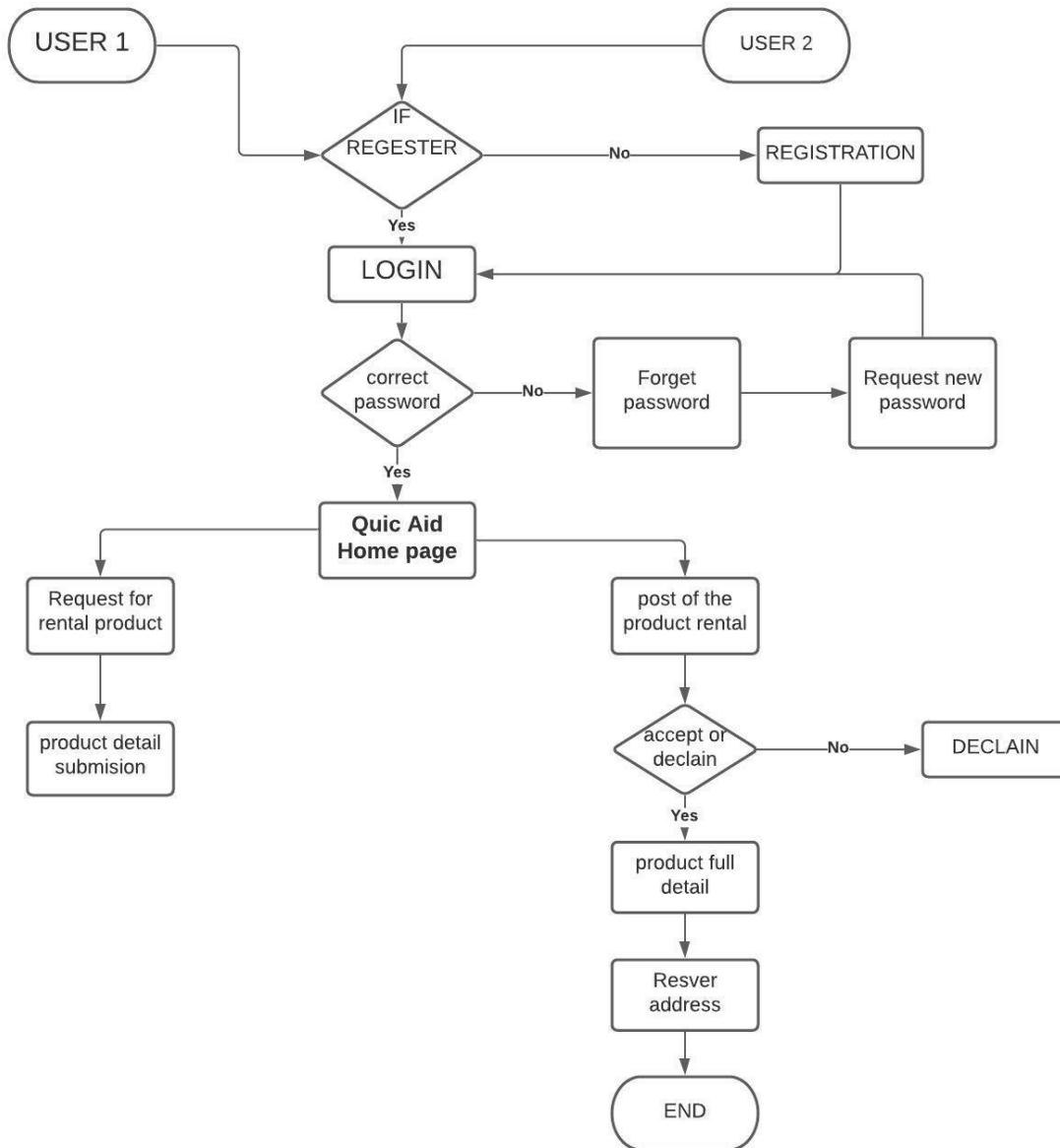
Generic Dart code, which depends on Flutter framework as well as the underlying platform code (Android SDK or iOS SDK). For example, `camera` is a plugin to interact with device cameras. It depends on the Flutter framework as well as the underlying framework to get access to the camera.

CHAPTER 4

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

4.1 DISCUSSION

This is the ideal map of the app running system:



This is the architecture diagram of this app. where this diagram shows the step by step process of the app

- Where user to use quik aid app ,first register in the app
- Login the page ,if the password is correct it takes to the home page. if the password is wrong, it leads to requesting the new password .
- Where we can see the post of the product ,where a user can accept or decline the request from another user .
- From the request form ,we can request the product for the rental.

4.2 RESULT

Product reat system has ready to implement by the user ,so the user can using this they can earn money and save money .In future we can implement this in very bigger way ,so where use will be more comfort by adding Maps, chat .This will be our feature implementation

Chapter 5

SUMMARY AND CONCLUSIONS

5.1 REFERENCES

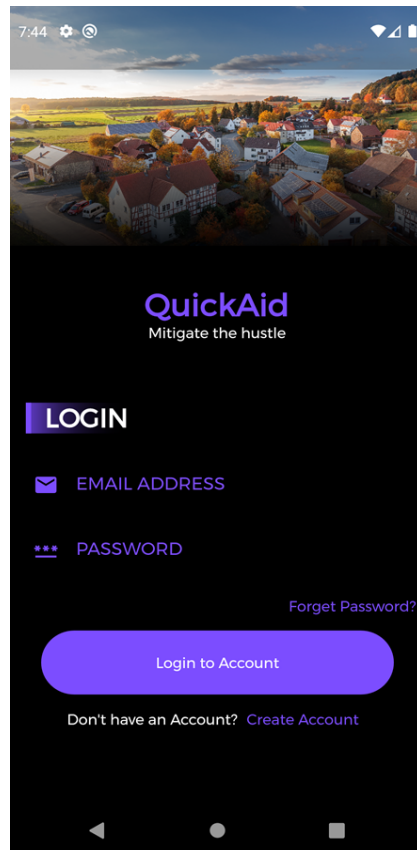
PREXA 365 is a cloud-based equipment rental management software designed to help businesses of all sizes streamline equipment tracking, inventory and contracts management, invoicing, preventive maintenance, and other operations via a unified portal. The platform enables organizations to store and manage leads, opportunities, contacts, customer interaction history, and more in a centralized repository for future reference.

link:<https://prexa365.com/>

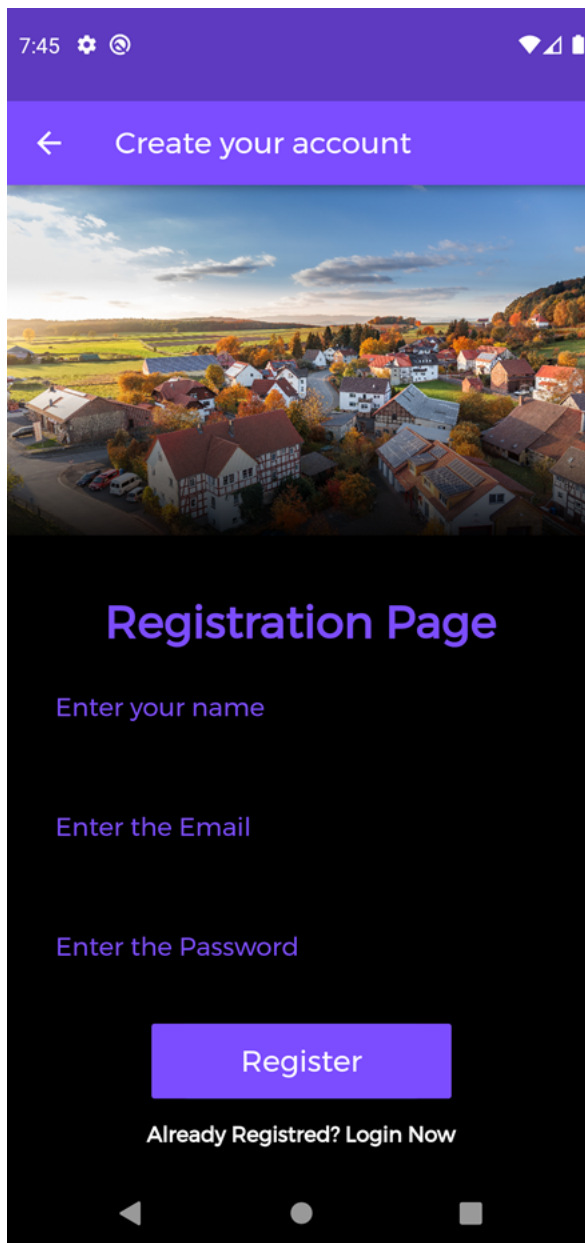
5.2 APPENDIX

5.2A SCREENSHOTS

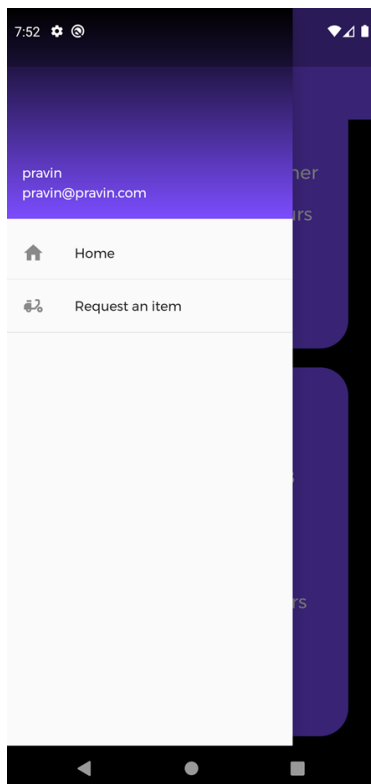
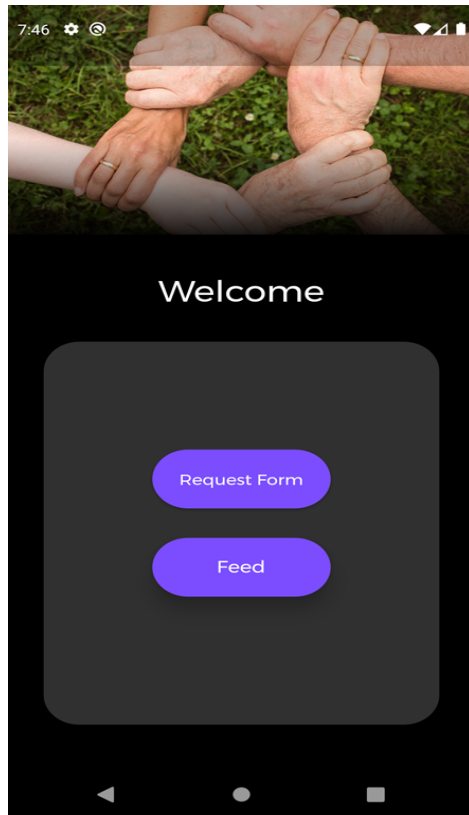
- Login page of Quick Aid



- Registration page



- Home page



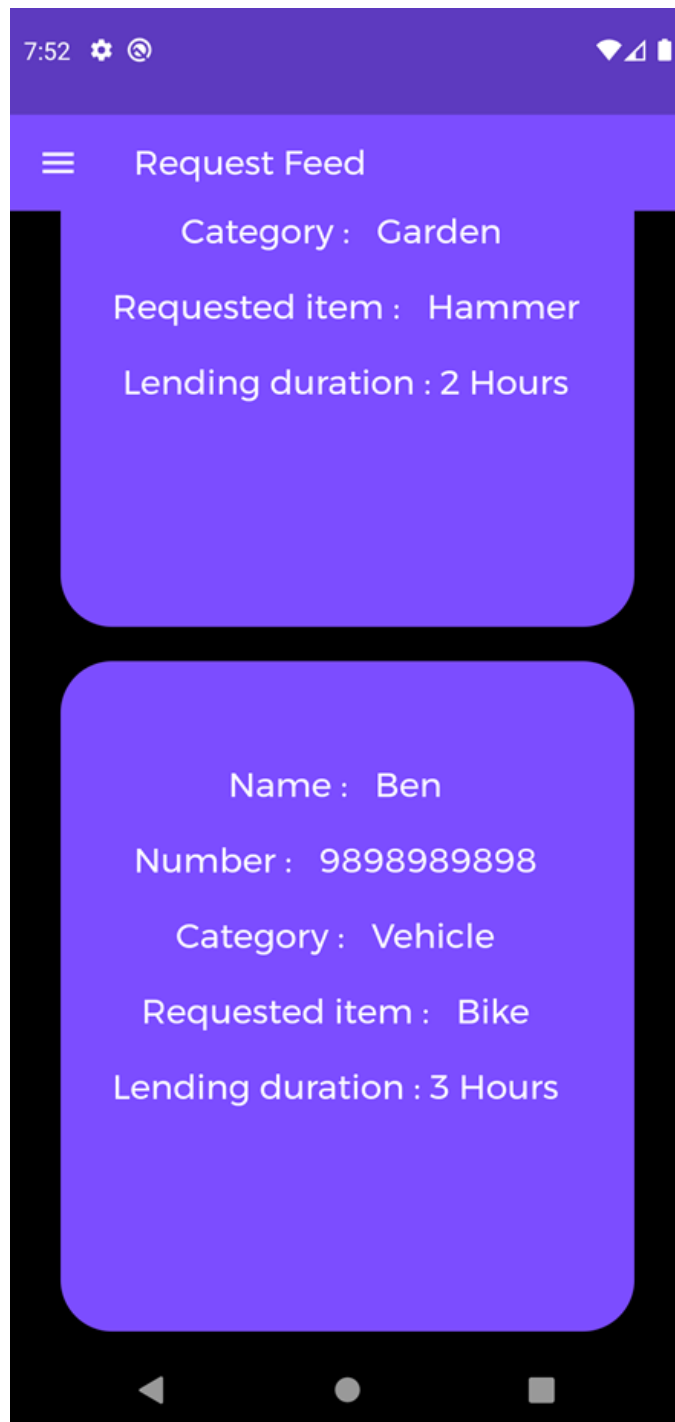
- Request from

The image shows a mobile application interface for a 'Request Form'. The form is displayed on a light gray background with a purple header bar. The header bar contains a back arrow and the title 'Request Form'. The form fields are as follows:

- Name:** Ben
- Phone number:** 9898989898 (with a 10/10 character count indicator)
- what category does the item belongs to?:** Vehicle
- Name of the Item:** Bike
- Lending time (Min/ Hours):** 3 Hours

A numeric keypad is overlaid at the bottom of the screen, featuring digits 1-9, 0, a decimal point, a comma, a minus sign, a backspace key, and a blue checkmark button.

- Post of request



5.2B SOURCE CODE

Main.dart

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

import 'package:quickaid/details.dart';

import 'package:quickaid/forum_screen.dart';

import 'package:quickaid/welcome.dart';

import 'package:quickaid/request_form.dart';

import 'package:quickaid/new_account.dart';

import 'constants.dart';

import 'login_screen.dart';

import 'package:firebase_core/firebase_core.dart';

import 'package:google_fonts/google_fonts.dart';

void main() async {

  WidgetsFlutterBinding.ensureInitialized();

  await Firebase.initializeApp();

  runApp(InitialScreen());

}

class InitialScreen extends StatelessWidget {

  @override
```

```

Widget build(BuildContext context) {

  return MaterialApp(

    home: LoginScreen(),

    title: appName,

    debugShowCheckedModeBanner: false,

    theme: ThemeData(

      appBarTheme:

        const AppBarTheme(backgroundColor: Colors.deepPurpleAccent),

      accentColor: Colors.deepPurpleAccent,

      fontFamily: GoogleFonts.montserrat().fontFamily,

      textTheme: GoogleFonts.montserratTextTheme()),

    );
}
}

```

Loginscreen.dart

```

import 'package:cloud_firestore/cloud_firestore.dart';

import 'package:flutter/cupertino.dart';

import "package:flutter/material.dart";

import 'package:quickaid/forum_screen.dart';

import 'package:firebase_auth/firebase_auth.dart';

import 'package:quickaid/new_account.dart';

import 'package:quickaid/welcome.dart';

import 'constants.dart';

```

```

State<LoginScreen> createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {

  GlobalKey<FormState> formKey = GlobalKey<FormState>();

  final _controlleremail = TextEditingController();

  String? _email, _password;

  void signIn(BuildContext context) async {

    await FirebaseAuth.instance

      .signInWithEmailAndPassword(email: _email!, password: _password!)

      .catchError((onError) {})

      .then((authUser) {

        if (authUser.user != null)

          Navigator.push(

            context, MaterialPageRoute(builder: (context) => Welcome()));

      });

    final _taskemail = _controlleremail.text;

    FirebaseFirestore.instance.collection("Datum").add({"Email": _taskemail});

  }

```

```

@override
Widget build(BuildContext context) {

  double height = MediaQuery.of(context).size.height;

  double width = MediaQuery.of(context).size.width;

return Scaffold(

  backgroundColor: Colors.black,

  resizeToAvoidBottomInset: true,

  body: GestureDetector(

    onTap: () => FocusScope.of(context).unfocus(),

    child: SingleChildScrollView(

      child: Form(

        key: formKey,

        child: Column(

          crossAxisAlignment: CrossAxisAlignment.start,

          mainAxisAlignment: MainAxisAlignment.start,

          children: [

            Stack(

              children: [

                Image.asset(

                  bgimage,

                  height: height * 0.30,

width: width,

                  fit: BoxFit.cover,

                ),

```

```

Container(
  height: height * 0.35,
  width: width,
  decoration: BoxDecoration(
    gradient: LinearGradient(
      stops: [0.4, 0.9],
      begin: Alignment.topCenter,
      end: Alignment.bottomCenter,
      colors: [Colors.transparent, Colors.black]),
    ),
    //color: Colors.indigo.withOpacity(0.2),
  ),
],
),
Center(
  child: Text(
    appName,
    style: TextStyle(
      fontSize: 30,
      fontWeight: FontWeight.w900,
      color: Colors.deepPurpleAccent),
    ),
  ),
Center(
  child: Text(

```

```

        slogan,

        style: TextStyle(color: Colors.white),

    ),

),

    SizedBox(

        height: 60,

    ),

    Padding(

        padding: const EdgeInsets.only(left: 15.0),

        child: Container(

            child: Text(" $loginin",

                style: TextStyle(

                    color: Colors.white,

                    fontSize: 25,

                    fontWeight: FontWeight.bold)),

            decoration: BoxDecoration(

                gradient: LinearGradient(

                    colors: [

                        Colors.deepPurpleAccent.withOpacity(0.8),

                        Colors.transparent

                    ],

                ),

                border: Border(

                    left: BorderSide(

                        color: Colors.deepPurpleAccent, width: 5))),

```

```

    ),

    ),

    Padding(

      padding:

        const EdgeInsets.only(left: 10.0, right: 10.0, top: 20.0),

      child: TextFormField(

        controller: _controlleremail,

        style: TextStyle(color: Colors.white),

        cursorColor: Colors.white,

        onSave: (value) {

          _email = value!;

        },

        validator: (String? email) {

          if (email!.isEmpty)

            return "Please Enter Email";

          else if (!RegExp(

            r"^[a-zA-Z0-9.!#$%&'*/+=?^_`{}~.-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$",

            caseSensitive: false)

            .hasMatch(email)) return "Not a Valid ID";

          return null;

        },

        decoration: InputDecoration(

          focusedBorder: UnderlineInputBorder(

            borderSide:

              BorderSide(color: Colors.deepPurpleAccent)),

```

```

        prefixIcon: Icon(
          Icons.email,
          color: Colors.deepPurpleAccent,
        ),
        labelText: " EMAIL ADDRESS",
        labelStyle: TextStyle(
          color: Colors.deepPurpleAccent, fontSize: 17)),
      ),
    ),
    Padding(
      padding:
        const EdgeInsets.only(left: 10.0, right: 10.0, top: 3.0),
      child: TextFormField(
        style: TextStyle(color: Colors.white),
        cursorColor: Colors.white,
        onSave: (value) {
          _password = value!;
        },
        validator: (password) {
          if (password!.isEmpty)
            return "Please Enter Password";

          else if (password.length < 8 || password.length > 15)
            return "Password criteria not met";

          return null;
        },

```

```

obscureText: true,

decoration: InputDecoration(

  focusedBorder: UnderlineInputBorder(

    borderSide:

      BorderSide(color: Colors.deepPurpleAccent)),

  prefixIcon: Icon(

    Icons.password,

    color: Colors.deepPurpleAccent,

  ),

  labelText: " PASSWORD",

  labelStyle: TextStyle(

    color: Colors.deepPurpleAccent, fontSize: 17)),

),

),

Align(

  alignment: Alignment.centerRight,

  child: TextButton(

    onPressed: () {},

    child: Text(forgettext),

    style: TextButton.styleFrom(

      primary: Colors.deepPurpleAccent))),

Center(

  child: SizedBox(

    height: height * 0.08,

    width: width * 0.85,

```

```

child: ElevatedButton(

  onPressed: () {

    if (formKey.currentState!.validate()) {

      formKey.currentState!.save();

      signIn(context);

      // if (_email == "pravin@g.com" &&

      //   _password == "root12345") {

      //   onTap:

      //     () => FocusScope.of(context).unfocus();

      //   Navigator.push(

      //     context,

      //     MaterialPageRoute(

      //       builder: (context) => Homescreen(),

      //     ),

      //   );

      // } else {

      //   print("invalid login");

      // }

    }

  },

  child: Text("Login to Account"),

  style: ElevatedButton.styleFrom(

    fixedSize: Size(200, 50),

    primary: Colors.deepPurpleAccent,

    shape: RoundedRectangleBorder(

```

```

        borderRadius: BorderRadius.circular(50))),

    ),

    ),

    Row(

      mainAxisAlignment: MainAxisAlignment.center,

      children: [

        Text(

          "Don't have an Account?",

          style: TextStyle(color: Colors.white),

        ),

        TextButton(

          onPressed: () {

            Navigator.push(

              context,

              MaterialPageRoute(

                builder: (context) => Newaccount());

            },

          child: Text("Create Account"),

          style: TextButton.styleFrom(

            primary: Colors.deepPurpleAccent),

          )

      ],

    )

  ],

),

```

```

    ),
    ),
    ),
    );
}
}

```

CreateAccount.dart

```

import 'package:flutter/material.dart';

import 'package:firebase_auth/firebase_auth.dart';

import 'package:quickaid/login_screen.dart';

import 'constants.dart';

class Newaccount extends StatefulWidget {

  @override
  _NewaccountState createState() => _NewaccountState();
}

class _NewaccountState extends State<Newaccount> {

  // final _uname = TextEditingController();

  String? _uname;

  final _auth = FirebaseAuth.instance;

  bool showProgress = false;

```

```

String? email, password;

@override

Widget build(BuildContext context) {

  double height = MediaQuery.of(context).size.height;

  double width = MediaQuery.of(context).size.width;

  return Scaffold(

    backgroundColor: Colors.black,

    appBar: AppBar(

      title: Text("Create your account"),

    ),

    body: SingleChildScrollView(

      child: Center(

        child: Column(

          children: [

            Stack(

              children: [

                Image.asset(

                  bgimage,

                  height: height * 0.30,

                  width: width,

                  fit: BoxFit.cover,

                ),

                Container(

                  height: height * 0.35,

```

```

width: width,

decoration: BoxDecoration(

  gradient: LinearGradient(

    stops: [0.5, 0.9],

    begin: Alignment.topCenter,

    end: Alignment.bottomCenter,

    colors: [Colors.transparent, Colors.black]),

  ),

),

],

),

Text(

  "Registration Page",

  style: TextStyle(

    fontWeight: FontWeight.w800,

    fontSize: 30.0,

    color: Colors.deepPurpleAccent),

),

 SizedBox(

  height: 10.0,

),

 Padding(

  padding: const EdgeInsets.symmetric(horizontal: 20),

  child: TextField(

    // controller: _uname,

```

```

        style: TextStyle(color: Colors.white),

        keyboardType: TextInputType.emailAddress,

        textAlign: TextAlign.center,

        onChanged: (value) {

            _uname = value; //get the value entered by user.

        },

        decoration: InputDecoration(

            labelText: "Enter your name",

            labelStyle: TextStyle(

                color: Colors.deepPurpleAccent, fontSize: 17),

            focusedBorder: UnderlineInputBorder(

                borderSide:

                    BorderSide(color: Colors.deepPurpleAccent)),

            border: OutlineInputBorder(

                borderRadius:

                    BorderRadius.all(Radius.circular(32.0))),

        ),

    ),

    SizedBox(

        height: 20.0,

    ),

    Padding(

        padding: const EdgeInsets.symmetric(horizontal: 20),

        child: TextField(

            style: TextStyle(color: Colors.white),

```

```

keyboardType: TextInputType.emailAddress,

textAlign: TextAlign.center,

onChanged: (value) {

  email = value; //get the value entered by user.

},

decoration: InputDecoration(

  labelText: "Enter the Email",

  labelStyle: TextStyle(

    color: Colors.deepPurpleAccent, fontSize: 17),

  focusedBorder: UnderlineInputBorder(

    borderSide:

      BorderSide(color: Colors.deepPurpleAccent)),

  border: OutlineInputBorder(

    borderRadius:

      BorderRadius.all(Radius.circular(32.0))),

  ),

),

 SizedBox(

  height: 20.0,

),

 Padding(

  padding: const EdgeInsets.symmetric(horizontal: 20),

  child: TextField(

    style: TextStyle(color: Colors.white),

    obscureText: true,

```

```

textAlign: TextAlign.center,

onChanged: (value) {

    password = value; //get the value entered by user.

},

decoration: InputDecoration(

    labelText: "Enter the Password",

    labelStyle: TextStyle(

        color: Colors.deepPurpleAccent, fontSize: 17),

    focusedBorder: UnderlineInputBorder(

        borderSide:

            BorderSide(color: Colors.deepPurpleAccent)),

    border: OutlineInputBorder(

        borderRadius:

            BorderRadius.all(Radius.circular(32.0))),

    ),

),

SizedBox(

    height: 20.0,

),

Material(

    elevation: 5,

    color: Colors.lightBlue,

    borderRadius: BorderRadius.circular(32.0),

    child: MaterialButton(

        color: Colors.deepPurpleAccent,

```

```

onPressed: () async {

  setState(() {

    showProgress = true;

  });

  try {

    final newuser =

      await _auth.createUserWithEmailAndPassword(

        email: email!, password: password!);

    if (newuser != null) {

      setState(() {

        showProgress = false;

      });

    }

  } catch (e) {}

  FirebaseAuth.instance.currentUser!

    .updateDisplayName(_uname);

},

minWidth: 200.0,

height: 50.0,

child: Text(

  "Register",

  style: TextStyle(

    fontWeight: FontWeight.w500,

    fontSize: 20.0,

```

```

        color: Colors.white),

    ),

  ),

  ),

  SizedBox(

    height: 15.0,

  ),

  GestureDetector(

    onTap: () {},

    child: GestureDetector(

      onTap: () => Navigator.push(context,

        MaterialPageRoute(builder: (context) => LoginScreen())),

      child: Text(

        "Already Registred? Login Now",

        style: TextStyle(

          color: Colors.white, fontWeight: FontWeight.w900),

        ),

      ),

    ),

  ),

  ],

),

);

}

```

```
}
```

WelcomeScreen.dart

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

import 'package:quickaid/forum_screen.dart';

import 'package:quickaid/request_form.dart';

import 'constants.dart';

class Welcome extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    double height = MediaQuery.of(context).size.height;

    double width = MediaQuery.of(context).size.width;

    return Scaffold(

      backgroundColor: Colors.black,

      body: SingleChildScrollView(

        physics: AlwaysScrollableScrollPhysics(),

        child: Column(

          children: [

            Stack(

              children: [

                Image.asset(
```

```

        handsimage,

        height: height * 0.30,

        width: width,

        fit: BoxFit.cover,

    ),

    Container(

        height: height * 0.35,

        width: width,

        decoration: BoxDecoration(

            gradient: LinearGradient(

                stops: [0.5, 0.9],

                begin: Alignment.topCenter,

                end: Alignment.bottomCenter,

                colors: [Colors.transparent, Colors.black]),

            ),

        ),

    ],

),

SizedBox(

    child: Text(

        "Welcome",

        style: TextStyle(fontSize: 30, color: Colors.white),

    ),

    height: 40,

),

```

```

Stack(
  children: [
    Padding(
      padding: const EdgeInsets.all(30),
      child: Container(
        height: height * 0.5,
        width: width * 0.85,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(30),
          color: Colors.grey[850],
        ),
      ),
    ),
    Padding(
      padding: const EdgeInsets.only(top: 140),
      child: Column(
        children: [
          Center(
            child: ElevatedButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(
                    builder: (context) => Requestform(),
                  );

```



```

MaterialPageRoute(
    builder: (context) => (Homescreeen()),
);
},
child: Text(
    "Feed",
    style: TextStyle(color: Colors.white, fontSize: 17),
),
style: ButtonStyle(
    backgroundColor: MaterialStateProperty.all(
        Colors.deepPurpleAccent),
    shape: MaterialStateProperty.all(
        RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(50),
        ),
    ),
    fixedSize: MaterialStateProperty.all(
        Size(150, 60),
    ),
    elevation: MaterialStateProperty.all(15)),
),
),
],
),
)

```

```

        ],
      ),
    ],
  ),
);
}
}

```

RequestForm.dart

```

import 'package:cloud_firestore/cloud_firestore.dart';

import 'package:flutter/material.dart';

class Requestform extends StatefulWidget {

  @override

  _RequestformState createState() => _RequestformState();

}

class _RequestformState extends State<Requestform> {

  final _controllerName = TextEditingController();

  final _controllerNumber = TextEditingController();

  final _controllerCategory = TextEditingController();

  final _controllerItem = TextEditingController();

  final _controllerDuration = TextEditingController();

```

```

void _saveTask() async {

  if (_formkey.currentState!.validate()) {

    final taskName = _controllerName.text;

    final taskNumber = _controllerNumber.text;

    final taskCategory = _controllerCategory.text;

    final taskItem = _controllerItem.text;

    final taskDuration = _controllerDuration.text;


    FirebaseFirestore.instance.collection("Datum").add({

      "Name": taskName,

      "PNumber": taskNumber,

      "Category": taskCategory,

      "Item": taskItem,

      "Duration": taskDuration,

      "Server": FieldValue.serverTimestamp(),

      "ID": FieldValue.increment(1)

    });

    // _controllerName.clear();

    // _controllerNumber.clear();

    // _controllerCategory.clear();

    // _controllerItem.clear();

    // _controllerDuration.clear();

  } else {

    ScaffoldMessenger.of(context).showSnackBar(SnackBar(

      content: Text("Fields can't be empty"),

```

```

));

}

}

String? _Name;

String? _Phone_number;

String? _Category;

String? _Item;

String? _Duration;

final GlobalKey<FormState> _formkey = GlobalKey<FormState>();

_buildName() {
  return Padding(
    padding: const EdgeInsets.symmetric(vertical: 10),
    child: TextFormField(
      controller: _controllerName,
      decoration: InputDecoration(labelText: 'Name'),
      validator: (String? Value) {
        if (Value!.isEmpty) {
          return 'Name is Required';
        }
      },
      onSaved: (String? Value) {
        _Name = Value!;

```

```

    },

    ),

);
}

Widget _buildPhone_number() {

  return TextFormField(

    controller: _controllerNumber,

    decoration: InputDecoration(labelText: 'Phone number'),

    maxLength: 10,

    keyboardType: TextInputType.phone,

    validator: (String? Value) {

      if (Value!.isEmpty) {

        return 'Phone number is Required';

      }

    },

    onSave: (String? Value) {

      _Phone_number = Value!;

    },

  );
}

```

```

Widget _buildcategory() {

  return Padding(

    padding: const EdgeInsets.only(bottom: 10),

```

```

child: TextFormField(

  controller: _controllerCategory,

  decoration: InputDecoration(

    labelText: 'what category does the item belongs to?'),

  validator: (String? Value) {

    if (Value!.isEmpty) {

      return 'Category is Required';

    }

  },

  onSave: (String? Value) {

    _Category = Value!;

  },

),

);
}

```

```

Widget _buildItem() {

  return Padding(

    padding: const EdgeInsets.symmetric(vertical: 10),

    child: TextFormField(

      controller: _controllerItem,

      decoration: InputDecoration(labelText: 'Name of the Item'),

      validator: (String? Value) {

        if (Value!.isEmpty) {

          return 'Item name is Required';

        }

      },

    ),

  );
}

```

```

    }

    },

    onSave: (String? Value) {

        _Item = Value!;

    },

),

);

}

Widget _buildDuration() {

    return Padding(

        padding: const EdgeInsets.symmetric(vertical: 10),

        child: TextFormField(

            controller: _controllerDuration,

            decoration: InputDecoration(labelText: 'Lending time (Min/ Hours)'),

            keyboardType: TextInputType.number,

            validator: (String? Value) {

                if (Value!.isEmpty) {

                    return 'Duration is Required';

                }

            },

            onSave: (String? Value) {

                _Duration = Value!;

            },

),

```

```

);
}

@override
Widget build(BuildContext context) {
  double height = MediaQuery.of(context).size.height;
  double width = MediaQuery.of(context).size.width;

  return Scaffold(
    resizeToAvoidBottomInset: true,
    appBar: AppBar(
      title: Text("Request Form"),
    ),
    body: SingleChildScrollView(
      child: Container(
        margin: EdgeInsets.all(24),
        child: Form(
          key: _formkey,
          child: Padding(
            padding: const EdgeInsets.all(8.0),
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                _buildName(),
                _buildPhone_number(),
                _buildcategory(),

```

```

    _buildItem(),

    _buildDuration(),

    SizedBox(
      height: 70,
    ),

    SizedBox(
      height: height * 0.08,
      width: width * 0.6,
      child: ElevatedButton(
        onPressed: () {
          _saveTask();

          if (!_formkey.currentState!.validate()) {
            return;
          }

          _formkey.currentState!.save();

          // print(_Name);

          // print(_Phone_number);

          // print(_Category);

          // print(_Item);

          // print(_Duration);
        },
        child: Text('Submit'),
        style: ElevatedButton.styleFrom(

```

```

        primary: Colors.deepPurpleAccent,

        shape: RoundedRectangleBorder(

            borderRadius: BorderRadius.circular(60))),

    ),

)

],

),

)),

),

),

);

}

}

```

ForumScreen.dart

```

import 'package:firebase_auth/firebase_auth.dart';

import 'package:flutter/material.dart';

import 'package:quickaid/request_form.dart';

import 'package:cloud_firestore/cloud_firestore.dart';

import 'package:quickaid/welcome.dart';

import 'constants.dart';

class Homescreeen extends StatelessWidget {

```

```

@override

Widget build(BuildContext context) {

  final _currentuser = FirebaseAuth.instance.currentUser!.email;

  final _currentname = FirebaseAuth.instance.currentUser!.displayName;

  double height = MediaQuery.of(context).size.height;

  double width = MediaQuery.of(context).size.width;

  return Scaffold(

    appBar: AppBar(

      title: Text("Request Feed"),

      elevation: 0,

    ),

    backgroundColor: Colors.black,

    drawer: Drawer(

      child: Column(

        children: [

          UserAccountsDrawerHeader(

            decoration: BoxDecoration(

              gradient: LinearGradient(

                colors: [Colors.black, Colors.deepPurpleAccent],

                begin: Alignment.topCenter,

                end: Alignment.bottomCenter,

              ),

            ),

          ),

          accountName: Text("pravin"),

          accountEmail: Text(_currentuser!),

```

```

),

ListTile(

  onTap: () {

    Navigator.pop(context);

    Navigator.push(context,

      MaterialPageRoute(builder: (context) => Welcome()));

  },

  title: Text('Home'),

  leading: Icon(Icons.home),

),

Divider(

  height: 0.1,

),

ListTile(

  onTap: () => Navigator.push(context,

    MaterialPageRoute(builder: (context) => Requestform())),

  title: Text('Request an item'),

  leading: Icon(Icons.delivery_dining),

),

Divider(

  height: 0.1,

),

// ListTile(

//   title: Text('Maps'),

//   leading: Icon(Icons.map),

```

```

// ),

Divider(
  height: 0.1,
)
],
),
),
body: SingleChildScrollView(
  physics: AlwaysScrollableScrollPhysics(),
  child: Column(
    children: [
      Stack(
        children: [
          Image.asset(
            handsimage,
            height: height * 0.30,
            width: width,
            fit: BoxFit.cover,
          ),
          Container(
            height: height * 0.35,
            width: width,
            decoration: BoxDecoration(
              gradient: LinearGradient(
                stops: [0.5, 0.9],

```

```

        begin: Alignment.topCenter,

        end: Alignment.bottomCenter,

        colors: [Colors.transparent, Colors.black]),

    ),

),

],

),

StreamBuilder<QuerySnapshot>(

    stream: FirebaseFirestore.instance

        .collection("Datum")

        .orderBy('Server')

        .snapshots(),

    builder: (BuildContext context,

        AsyncSnapshot<QuerySnapshot> snapshot) {

        if (snapshot.hasError) {

            return Text('Something went wrong');

        }

        if (snapshot.connectionState == ConnectionState.waiting) {

            return Text("Loading");

        }

        return ListView(

            shrinkWrap: true,

            physics: NeverScrollableScrollPhysics(),

```

```

children:

    snapshot.data!.docs.map((DocumentSnapshot document) {

        // Map<String, dynamic> data =

        //   document.data()! as Map<String, dynamic>;

        Text("Request section");

        return Center(

            child: SingleChildScrollView(

                child: Stack(

                    children: [

                        Padding(

                            padding: const EdgeInsets.symmetric(vertical: 10),

                            child: Container(

                                height: height * 0.5,

                                width: width * 0.85,

                                decoration: BoxDecoration(

                                    borderRadius: BorderRadius.circular(30),

                                    color: Colors.deepPurpleAccent,

                                ),

                            ),

                        ),

                    ],

                ),

                Positioned(

                    top: 70,

                    left: 30,

                    child: Column(

                        children: [

```

```

Text(
  "Name : " + document['Name'],
  style: TextStyle(
    color: Colors.white, fontSize: 20),
),
 SizedBox(height: 20),
Text(
  "Number : " + document['PNumber'],
  style: TextStyle(
    color: Colors.white, fontSize: 20),
),
 SizedBox(height: 20),
Text(
  "Category : " + document['Category'],
  style: TextStyle(
    color: Colors.white, fontSize: 20),
),
 SizedBox(height: 20),
Text(
  "Requested item : " + document['Item'],
  style: TextStyle(
    color: Colors.white, fontSize: 20),
),
 SizedBox(height: 20),
Text(

```

```

        "Lending duration : " +

        document["Duration"],

        style: TextStyle(

            color: Colors.white, fontSize: 20),

        ),

    ],

),

),

],

),

),

);

}).toList(),

);

},

)

],

),

),

);

}

}

```