VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI – 590 014

Internship Project Report

On

“Library Management System”

By:

ENMAZ

Submitted in partial fulfillment of the requirements for the 7th semester of ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi*

Submitted by:

Ishanu Agarwal
1RN18IS054

Under the Guidance of:

Mrs. Vanishri S.
Associate Professor
Dept. of ISE

Department of Information Science and Engineering

RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560 098

2021-2022

RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road,
Bengaluru-560 098

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

ESTD:2001
An Institute with a Difference

CERTIFICATE

Certified that the Internship project work entitled **“*Library Management System*”** has been successfully carried out by **Ishanu Agarwal** bearing USN **1RN18IS054** bonafide students of **RNS Institute of Technology** in partial fulfillment of the requirements for the **7th semester Bachelor of Engineering** in **Information Science and Engineering** of **Visvesvaraya Technological University**, Belagavi, during the academic year 2021-2022. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The project report has been approved as it satisfies the internship project of 7th semester BE in ISE.

Mrs. Vanishri S.

Internship Guide
Associate Professor
Department of ISE

Dr. Suresh L

Professor and HoD
Department of ISE
RNSIT

Dr. M.K. Venkatesha

Principal
RNSIT

External Viva

Name of the Examiners
1. ____________________________
2. ____________________________

Signature with Date
1. _______________________
2. _______________________

ACKNOWLEDGMENT

At the very onset I would like to place our gratefulness to all those people who helped me in making the Internship a successful one. Coming up, this internship to be a success was not easy. Apart from the sheer effort, the enlightenment of the very experienced teachers also plays a paramount role because it is they who guided me in the right direction.

First, we would like to thank the **Management of RNS Institute of Technology** for providing such a healthy environment for the successful completion of internship work.

In this regard, we express sincere gratitude to our beloved Principal **Dr. M K Venkatesha,** for providing us all the facilities.

We are extremely grateful to our own and beloved Professor and Head of Department of Information science and Engineering, **Dr. Suresh L**, for having accepted to patronize me in the right direction with all her wisdom.

We place our heartfelt thanks to **Mrs. Vanishri S.** Professor, Department of Information Science and Engineering for having guided internship and all the staff members of the department of Information Science and Engineering for always helping.

We thank **Mr. Akshay D R, Enmaz**, for providing the opportunity to be a part of the Internship program and having guided me to complete the same successfully.

We also thank our internship coordinator **Dr. R Rajkumar,** Associate Professor, Department of Information Science and Engineering. I would thank my friends for having supported me with all their strength and might. Last but not the least, I thank my parents for supporting and encouraging me throughout. I have made an honest effort in this assignment

Date:	Ishanu Agrawal	1RN18IS054
Place: Bengaluru	Nikhil Singh	1RN18IS069
	Pranshu Shrivastava	1RN18IS077

ABSTRACT

In today’s world, Smart phones have changed our lives and have become an indispensable part of our lives because of its specialty to simplify our routine work and thereby saving our time. A Smartphone with an Android OS offers excellent functionality to the users offering a distinct experience. Android is a Linux based operating system and it was bought by Google in 2007.There are tons of application available and one of the prime reason for this vast number is android being an open source. On the other hand, android based device like mobile, tab are very user friendly. A survey has done by “Light Castle Partners” research wing which indicates that though other operating system mobile users exist but the majority users are goes with android operating system. In this context, Project application is developed based on android platform. The name of application is define as ‘Library Management System’.

It is a simple application made with flutter in which user can select and customize the types of books ,types of books based on his/her wishes. It consists of many categories of topics for books from which user can select one of the book based on his/her learning and requirements. It helps users to gain their knowledge on certain topics.

.

CONTENTS

Description **Page No.**

Acknowledgement i

Abstract ii

Content iii

List of Figures iv

1. Introduction

1.1 Organization 1

1.1.1 Company Profile 1

1.2 Domain Technology 2

1.2.1 Introduction to Flutter 2

1.2.2 History 3

1.2.3 FrameWork-Architecture

2. Literature Survey

2.1 Library Management using Flutter 5

2.2 Functional Requirements 6

2.3 Tools/ Languages/ Platform 9

3. Design and Implementation Definition

3.1 Architecture/ Sequence Diagram/ Flowchart 10

3.2 Algorithm/Methods/ Pseudo code 11

3.3 Libraries Used/ API’s 13

4. Observation And Results

4.1 Testing 14

4.2 Code 14

4.3 Results & Snapshots 20

5. Conclusion And Future Enhancement 25

References 26

LIST OF FIGURES

Fig. No.	Description	Page No
Fig. 3.1	Three Layer Architecture	10
Fig. 4.1	Test Case	14
Fig. 4.2	Landing Page	20
Fig. 4.3	Home Page	21
Fig. 4.4	About Page	22
Fig. 4.5	Login Page	22
Fig. 4.6	Post Page	23

CHAPTER 1

INTRODUCTION

In today’s world, Smart phones have changed our lives and have become an indispensable part of our lives because of its specialty to simplify our routine work and thereby saving our time.

Currently most of the Examination like organizational recruitment, University class test are paper based, which costs time and resources. Questionnaire is developed, printed, and then collect data, entry, editing, cleaning, which time consuming and costly.

Proposed application is the starting for avoid those circumstances which are been currently faced by any organization.

1.1 ORGANIZATION

1.1.1 Company Profile

Enmaz has a simple yet robust solution that helps any Industry / Factory digitise their workfloor in no time. The products offered will help in remote monitoring, controlling and also analysing any machinene parameter or process.

1.2 Domain Technology

1.2.1 Introduction to Flutter

Flutter is Google's Mobile SDK to build native iOS and Android, Desktop (Windows, Linux, macOS), Web apps from a single codebase. When building applications with Flutter everything towards Widgets – the blocks with which the flutter apps are built. They are structural elements that ship with a bunch of material design-specific functionalities and new widgets can be composed out of existing ones too. The process of composing widgets together is called composition. The User Interface of the app is composed of many simple widgets, each of them handling one particular job. That is the reason why Flutter developers tend to think of their flutter app as a tree of widgets.

1.2.2 History

Flutter launched as a project called Sky which at the beginning worked only on Android. Flutter's goal is enabling developers to compile for every platform using its own graphic layer rendered by the Skia engine. Here's a brief presentation of Flutter's relatively short history.

Flutter is a free and open-source mobile UI framework created by Google and released in May 2017. In a few words, this allows you to create a native mobile application with only one code. It means that you can use one programming language and one codebase to create two different apps (IOS and Android).

The first version of Flutter was known by the codename "Sky" and ran on the Android operating system. It was unveiled at the 2015 Dart developer summit[6] with the stated intent of being able to render consistently at 120 frames per second.[7] During the keynote of Google Developer Days in Shanghai in September 2018, Google announced Flutter Release Preview 2, which is the last big release before Flutter 1.0. On December 4th of that year, Flutter 1.0 was released at the Flutter Live event, denoting the first "stable" version of the Framework. On December 11, 2019, Flutter 1.12 was released at the Flutter Interactive event.[8]

On May 6, 2020, the Dart software development kit (SDK) in version 2.8 and the Flutter in version 1.17.0 were released, where support was added to the Metal API, improving performance on iOS devices (approximately 50%), new Material widgets, and new network tracking.

On March 3, 2021, Google released Flutter 2 during an online Flutter Engage event. This major update brought official support for web-based applications with new CanvasKit renderer and web specific widgets, early-access desktop application support for Windows, macOS, and Linux and improved Add-to-App APIs.[9] This release included sound null-safety, which caused many breaking changes and issues with many external packages, but the Flutter team included instructions to mitigate these changes as well.

On September 8th, 2021, the Dart SDK in version 2.14 and Flutter version 2.5 were released by Google. The update brought improvements to the Android Full-Screen mode and the latest version of Google's Material Design called Material You. Dart received two new updates, the newest lint conditions have been standardized and preset as the default conditions as well Dart for Apple Silicon is now stable.

1.2.3 Framework-Architecture

The major components of Flutter include:

- Dart platform
- Flutter engine
- Foundation library
- Design-specific widgets
- Flutter Development Tools (DevTools)

Dart platform

Flutter apps are written in the Dart language and make use of many of the language's more advanced features.

On Windows, macOS, and Linux[11] Flutter runs in the Dart virtual machine, which features a just-in-time execution engine. While writing and debugging an app, Flutter uses Just In Time compilation, allowing for "hot reload", with which modifications to source files can be injected into a running application. Flutter extends this with support for stateful hot reload, where in most cases changes to source code are reflected immediately in the running app without requiring a restart or any loss of state.

For better performance, release versions of Flutter apps targeting Android and iOS are compiled with ahead-of-time (AOT) compilation.

Flutter engine

Flutter's engine, written primarily in C++, provides low-level rendering support using Google's Skia graphics library. Additionally, it interfaces with platform-specific SDKs such as those provided by Android and iOS.[10] The Flutter Engine is a portable runtime for hosting Flutter applications. It implements Flutter's core libraries, including animation and graphics, file and network I/O, accessibility support, plugin architecture, and a Dart runtime and compile toolchain. Most developers interact with Flutter via the Flutter Framework, which provides a reactive framework and a set of platform, layout, and foundation widgets.

Foundation library

The Foundation library, written in Dart, provides basic classes and functions that are used to construct applications using Flutter, such as APIs to communicate with the engine.

Design-specific widgets

The Flutter framework contains two sets of widgets that conform to specific design languages: Material Design widgets implement Google's design language of the same name, and Cupertino widgets implement Apple's iOS Human interface guidelines

Chapter 2

LITERATURE SURVEY

2.1 Library Management using Flutter

The evolution of today's mobile devices increases the number of mobile applications developed and among them the Library managements. Android Mobile hardware and software platforms allow the running of faster and richer applications. This paper presents the main steps in the development of a Library management for Android using Android Studio. This work deals with the development of an Android-based multiple-choice question examination system, namely: Library App. This application is developed for educational purposes, allowing the users to read multiple books for different field interested.

2.1.1 Introduction

The purpose of a library management system is to operate a library with efficiency and at reduced costs. The system being entirely automated streamlines all the tasks involved in operations of the library. The activities of book purchasing, cataloging, indexing, circulation recording and stock checking are done by the software.

The library management system software helps in reducing operational costs. Managing a library manually is labor intensive and an immense amount of paperwork is involved. An automated system reduces the need for manpower and stationery. This leads to lower operational costs.

The system saves time for both the user and the librarian. With just a click the user can search for the books available in the library. The librarian can answer queries with ease regarding the availability of books. Adding, removing or editing the database is a simple process. Adding new members or cancelling existing memberships can be done with ease. Android is software that is built basically for Mobile phones. It's supported by the Linux Kernel and other open-source software and is developed by Google. Android is very popular nowadays among students and students are now choosing Android for his or her projects. It’s greatly important for a beginner to create baby Android apps to learn Android. The evolution of today's mobile devices increases the quantity of mobile applications developed and among them the Library managements. Android Mobile hardware and software platforms allow the running of faster and richer applications. This paper presents the most steps within the development of a Library management for Android using Android Studio. Android could be a software package and Linux based operating system for mobile devices/ equipment such as tablet computers and smartphones. It's developed by Google and later the OHA (Open Handset Alliance). Java language is principally wont to write the android code while other languages are used. The goal of the android project is to form a successful real-world product that improves the mobile experience for end-users.

2.1.2 Literature Survey

Android is rapidly getting famous, and therefore the number of its users are increasing day by day, because it's easy to access the required Android-based apps on tablets and smartphones . Therefore, we found the use of the Android App is less complicated and longer efficient to facilitate the users during this way with none difficulty. Fig 1 Applications and widgets Application Programming Interface Libraries Android runtime Android Architecture reduces the time of development and can reduce the quantity of memory that is required by the appliance. In Future we are able to include redesigned an friendlier user interface because the application targets various levels of users starting from sophisticated programmer to naive users Android is most used Mobile OS worldwide. making an android app isn't as difficult mutually think about. You simply must have minor knowledge about android structure and tiny java programming. Android is an open source and Linuxbased package for mobile devices like smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies. Android offers a unified approach to application development for mobile devices which suggests developers need only develop for Android, and their applications should be ready to run on different devices powered by Android. The very first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 while as because the first working commercial version, Android 1.0, was released in September 2008, and later it kept on changing/updating its rules. On June

27, 2012, at the Google I/O conference, in that meeting, Google announced the subsequent Android version, 4.1. Jelly Bean is an incremental update, with the first aim of improving the program, both in terms of functionality/ activities and performance. The text file i.e source code for Android is out there under free and open source software licenses. Google publishes most of the code under the Apache License version 2.0 and therefore the rest, Linux kernel changes, under the GNU General Public License version.

CHAPTER 3

REQUIREMENT ANALYSIS, TOOLS & TECHNOLOGIES

3.1 Hardware & Software Requirements

The Hardware requirements are very minimal and the program can be run on most of the machines.

Processor	:	Pentium 4 processor
Processor Speed	:	2.4 GHz
RAM	:	4 GB
Storage Space	:	40 GB
Monitor Resolution	:	1024*768 or 1336*768 or 1280*1024

The Software requirements are very minimal and the program can be run on machines with these requirements satisfied.

Editor	:	Visual Studio Code
Operating System	:	Windows Operating System
IDE	:	VS Code
Backend Tool	:	MongoDB

3.2 Tools/ Languages/ Platform

Various tool used in making this project is given below:

Editor/IDE	:	Visual Studio Code
Operating System	:	Windows/Mac OS
Languages	:	Dart,
Backend Tool	:	Open Trivia API

3.3 Functional Requirements

Flutter

Flutter is Google's Mobile SDK to build native iOS and Android apps from a single codebase. When building applications with Flutter everything towards Widgets – the blocks with which the flutter apps are built. The User Interface of the app is composed of many simple widgets, each of them handling one particular job. That is the reason why Flutter developers tend to think of their flutter app as a tree of widgets.

Compared to its contemporary technologies like React Native, Kotlin, and Java, Flutter is much better in regard to having a Single Codebase for Android and iOS, Reusable UI and Business Logic, high compatibility, performance, and productivity.

Dart

Dart is an open-source general-purpose programming language developed by Google. It supports application development in both client and server-side. But it is widely used for the development of android apps, iOS apps, IoT(Internet of Things), and web applications using the Flutter Framework.

Syntactically, Dart bears a strong resemblance to Java, C, and JavaScript. It is a dynamic object-oriented language with closure and lexical scope. The Dart language was released in

2011 but came into popularity after 2015 with Dart 2.0.

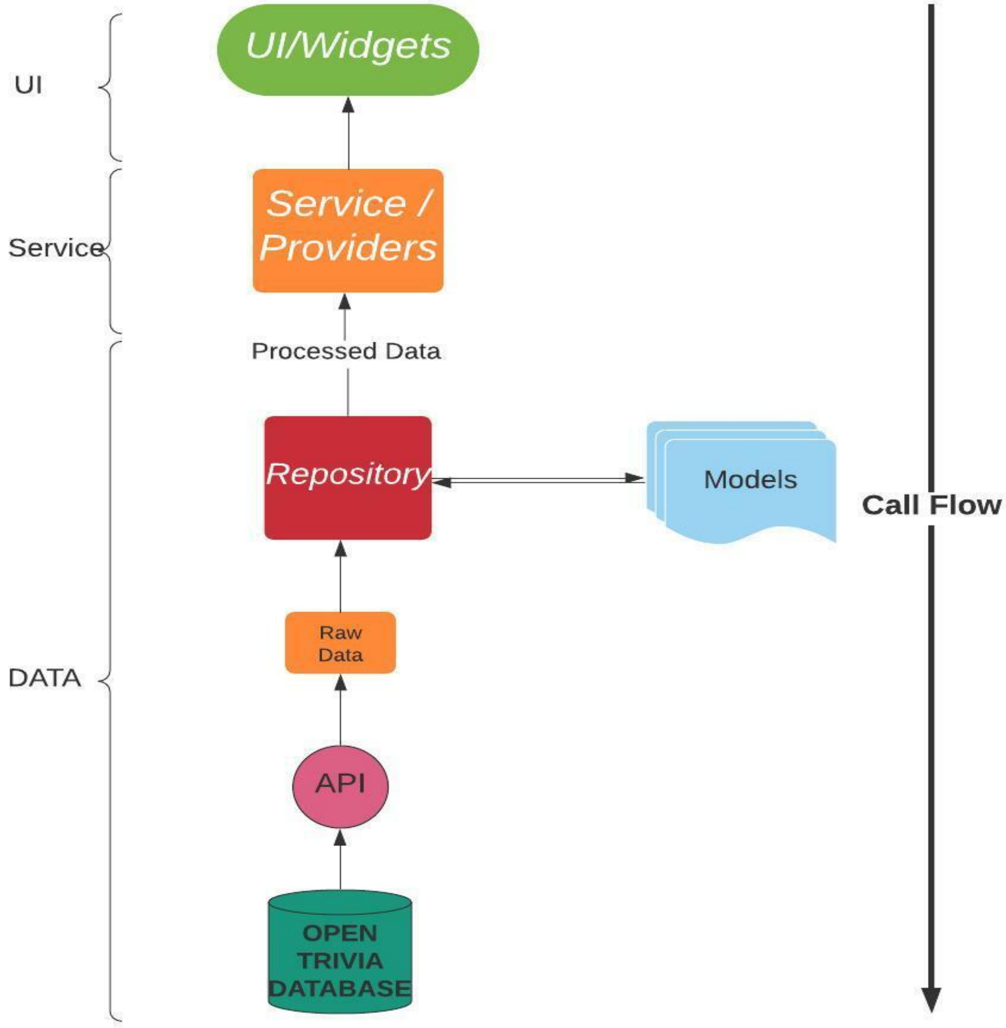
Open Trivia Database API

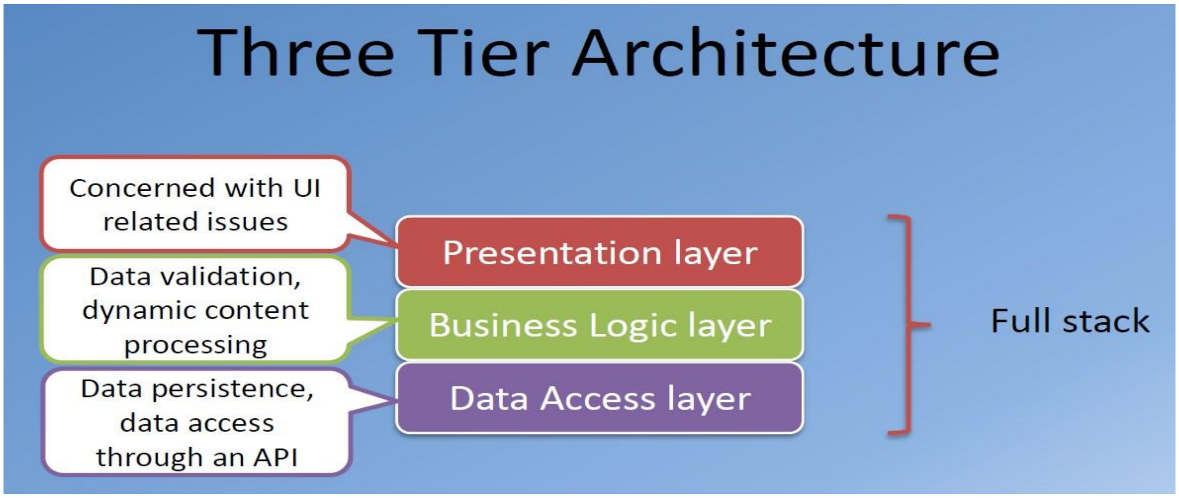
The Open Trivia Database is a free, user contributed trivia database. The Open Trivia Database API offers a specified number of trivia questions via category, difficulty, type, encoding. JSON is returned.

CHAPTER 4

DESIGN AND IMPLIMENTATION

3.1 Architecture/Sequence diagram/Class diagrams /Flowchart





Three Tier Architecture

Concerned with UI related issues

Data validation, dynamic content processing

Data persistence, data access through an API

Presentation layer

Business Logic layer

Data Access layer

Full stack

Fig. 3.1 Three Layer Architecture

Chapter 5

IMPLEMENTATION DETAILS

5.1 main.dart

```
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:flutter_bloc/flutter_bloc.dart';
import 'package:provider/provider.dart';
import 'package:Library/service/question_service_provider.dart';
import 'package:Library/service/Library_customizer_cubit.dart';
import 'package:Library/view/customize_Library_page.dart';
import 'package:Library/view/home_page.dart';
import 'package:Library/view/Library_page.dart';
import 'package:Library/view/score_page.dart';

void main() {
  WidgetsFlutterBinding.ensureInitialized();
  SystemChrome.setPreferredOrientations(
      [DeviceOrientation.portraitUp, DeviceOrientation.portraitDown]).then((_) {
    runApp(new MyApp());
  });
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return BlocProvider.value(
      value: LibraryCustomizerCubit(),
      child: ChangeNotifierProvider<QuestionServiceProvider>(
          create: (_) => QuestionServiceProvider(),
          builder: (context, child) {
            return MaterialApp(
              title: "Library",
              debugShowCheckedModeBanner: false,
              initialRoute: HomePage.routeName,
              routes: {
                HomePage.routeName: (context) => HomePage(),
                CustomizeLibraryPage.routeName: (context) => CustomizeLibraryPage(),
                LibraryPage.routeName: (context) => LibraryPage(),
                ScorePage.routeName: (context) => ScorePage(),
              },
              theme: ThemeData(
                primaryColor: Colors.purple,
                brightness: Brightness.dark,
                accentColor: Colors.deepOrangeAccent,
                backgroundColor: Color(0xFF263058),
                scaffoldBackgroundColor: Color(0xFF263058),
              ),
            );
          }),
    );
  }
}
```

5.2 Homepage.dart

```
import 'package:barcode_scan/barcode_scan.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:library_system/models/Book_model.dart';
import 'package:library_system/pages/addNewBookPage.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:library_system/pages/inventoryPage.dart';
import 'package:library_system/pages/issueBookPage.dart';
import 'package:library_system/pages/profilepage.dart';
import 'package:library_system/pages/returnBookPage.dart';
class HomePage extends StatefulWidget {
  static String id = 'homepage';
  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  Future _scanQR() async {
    try {
      String qrResult = await BarcodeScanner.scan();
      setState(() {
        result = qrResult;
      });
      Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => AddNewBookPage(
              isbn: result,
            ),
          ));
    } on PlatformException catch (ex) {
      if (ex.code == BarcodeScanner.CameraAccessDenied) {
        setState(() {
          result = null;
        });
        Fluttertoast.showToast(
            msg: "Camera permission was denied",
            toastLength: Toast.LENGTH_SHORT,
            gravity: ToastGravity.CENTER,
            timeInSecForIosWeb: 1,
            backgroundColor: Colors.red,
            textColor: Colors.white,
            fontSize: 16.0);
        Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => AddNewBookPage(
                isbn: result,
              ),
            ));
      } else {
        setState(() {
          result = null;
        });
        Fluttertoast.showToast(
            msg: "Unknown Error $ex",
            toastLength: Toast.LENGTH_SHORT,
            gravity: ToastGravity.CENTER,
            timeInSecForIosWeb: 1,
            backgroundColor: Colors.red,
            textColor: Colors.white,
            fontSize: 16.0);
        Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => AddNewBookPage(
                isbn: result,
              ),
            ));
      }
    } on FormatException {
      setState(() {
        result = null;
      });
      Fluttertoast.showToast(
          msg: "You pressed the back button before scanning anything",
          toastLength: Toast.LENGTH_SHORT,
          gravity: ToastGravity.CENTER,
          timeInSecForIosWeb: 1,
          backgroundColor: Colors.red,
          textColor: Colors.white,
          fontSize: 16.0);
    } catch (ex) {
        setState(() {
          result = null;
        });
        Fluttertoast.showToast(
            msg: "Unknown Error $ex",
            toastLength: Toast.LENGTH_SHORT,
            gravity: ToastGravity.CENTER,
            timeInSecForIosWeb: 1,
            backgroundColor: Colors.red,
            textColor: Colors.white,
            fontSize: 16.0);
        Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => AddNewBookPage(
                isbn: result,
              ),
            ));
      }
    }

    Future _scannerQR() async {
      try {
        String mainResult = await BarcodeScanner.scan();
        setState(() {
          reResult = mainResult;
        });
        Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => ReturnBookPage(),
            ));
      } on PlatformException catch (ex) {
        if (ex.code == BarcodeScanner.CameraAccessDenied) {
          setState(() {});
        } else {
          setState(() {});
        }
      } on FormatException {
        setState(() {});
      } catch (ex) {
        setState(() {});
      }
    }

    @override
    Widget build(BuildContext context) {
      return Scaffold(
        body: ListView(
    children: [
      Container(
        padding: EdgeInsets.all(20),
        width: MediaQuery.of(context).size.width,
        child: Row(
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: [
            Column(
              mainAxisSize: MainAxisSize.min,
              mainAxisAlignment: MainAxisAlignment.start,
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                IconButton(
                  onPressed: () async {
                  },
                  icon: Icon(
                    Icons.apps,
                    color: Color(0xFF584846),
                    size: 40,
                  ),
                ),
              ],
            ),
            IconButton(
              icon: Icon(
                Icons.account_circle,
                size: 50,
                color: Color(0xFFDD3617),
              ),
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(builder: (context) => ProfilePage()),
                );
              },
            ),
          ],
        ),
      ),
      Row(
        children: [
          Padding(padding: EdgeInsets.only(left: 20)),
          Text(
            'Welcome!',
            style: TextStyle(
              fontSize: 20,
              color: Color(0xFF584846),
              fontWeight: FontWeight.bold,
            ),
        ),
      ],
    ),
    display1(
      context,
      () {
        _scannerQR();
      },
      "RETURN BOOK",
      Colors.white38,
      Colors.white70,
      Color(0xFF584846),
    ),
    display1(
      context,
      () {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => InventoryPage(),
          ),
        );
      },
      "INVENTORY",
      Colors.white38,
      Colors.white70,
      Color(0xFF584846),
    ),
    display1(
      context,
      () {
        _scanQR();
      },
      "ADD NEW BOOK",
      Colors.white38,
      Colors.white70,
      Color(0xFF584846),
    ),
    display1(
      context,
      () {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) =>
                //  IssueBookPage(barcode:"P2ToQVbmmq1zyX5uV05J")
                DummyIssueBook(),
          ),
        );
```

```
          },
          "ISSUE BOOK",
          Colors.white38,
          Colors.white70,
          Color(0xFF584846),
        ),
      ],
    ),
  );
 }
}

display1(BuildContext context, onTap, String title, Color c1, Color c2,
    Color textc) {
  return Padding(
    padding: EdgeInsets.symmetric(horizontal: 20, vertical: 10),
    child: GestureDetector(
      onTap: onTap,
      child: Card(
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(15),
        ),
        shadowColor: Colors.black,
        elevation: 8,
        child: Container(
          decoration: BoxDecoration(
            gradient: LinearGradient(
                colors: [c1, c2],
                begin: Alignment.topLeft,
                end: Alignment.bottomRight),
            borderRadius: BorderRadius.circular(15),
          ),
          width: MediaQuery.of(context).size.width - 40,
          height: 105,
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: [
                  Padding(
                    padding: const EdgeInsets.all(20.0),
                    child: Text(
                      title,
                      style: TextStyle(
                        fontSize: 27,
                        color: textc,
                        fontWeight: FontWeight.bold,
                      ),
                    ),
                  ),
                ],
              )
            ],
          ),
        ),
      ),
    ),
  );
}
```

5.3 LoginSignup.dart

```
import 'package:flutter/material.dart';
import 'package:library_system/pages/homePage.dart';
import 'package:firebase_auth/firebase_auth.dart';

class LoginSignupPage extends StatefulWidget {
  static String id = 'login_page';
  @override
  _LoginSignupPageState createState() => _LoginSignupPageState();
}

class _LoginSignupPageState extends State<LoginSignupPage> {
  RegExp regExp = RegExp(
      r"^[a-z0-9!#$%&'*+/=?^_`{|}~-]+(?:\.[a-z0-9!#$%&'*+/=?^_`{|}~-]+)*@(?:[a-zA-Z0-9][a-
zA-Z0-9-]{0,253}\.)*[a-zA-Z0-9][a-zA-Z0-9-]{0,253}\.[a-zA-Z0-9]{2,}$");
  final _auth = FirebaseAuth.instance;
  String email;
  String password;
  String emailError = "";
  String passwordError = "";
  String loginSignUpError = "";
  bool busy = false;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Color(0xFFFFFFFF),
      body: SafeArea(
        child: Stack(
          children: [
            Padding(
              padding: const EdgeInsets.only(left: 8.0, right: 8.0),
              child: ListView(
                children: [
                  SizedBox(
                    height: 15,
                  ),
                  Image.asset('assets/images/library.png'),
                  SizedBox(height: 40),
                  (loginSignUpError == "")
                      ? SizedBox(height: 0)
                      : Center(child: Text(loginSignUpError)),
                  Padding(
                    padding: const EdgeInsets.only(left: 10, right: 10),
                    child: TextFormField(
                      onChanged: (value) {
                        if (value.length <= 0) {
                          setState(() {
                            emailError = "Enter Email";
                          });
                        } else if (!regExp.hasMatch(value)) {
                          setState(() {
                            emailError = "Invalid Email";
                          });
                        } else {
                          setState(() {
                            emailError = "";
                            email = value;
                          });
                        }
                      },
                      decoration: InputDecoration(
                        prefixIcon: Icon(
                          Icons.email,
                          color: Color(0xFF584846),
                        ),
                        enabledBorder: UnderlineInputBorder(
                          borderSide: BorderSide(color: Colors.brown),
                        ),
                        hintText: '\tEnter Email-id',
                        errorText: emailError != "" ? emailError : null,
                      ),
                    ),
                  ),
                  SizedBox(height: 14),
                  Padding(
                    padding: const EdgeInsets.only(left: 10, right: 10),
                    child: TextField(
                      onChanged: (value) {
                        if (value.length < 6) {
                          setState(() {
                            passwordError = "Too Short";
                          });
                        } else {
                          setState(() {
                            passwordError = "";
                            password = value;
                          });
                        }
                      },
                      decoration: InputDecoration(
                        prefixIcon: Icon(
                          Icons.lock,
                          color: Color(0xFF584846),
                        ),
                        enabledBorder: UnderlineInputBorder(
                          borderSide: BorderSide(color: Color(0xFF584846)),
                        ),
                        hintText: 'Enter Password',
                        errorText: passwordError == "" ? null : passwordError,
                      ),
                      obscureText: true,
                    ),
                  ),
                  Padding(
                    padding: const EdgeInsets.only(top: 10.0),
                    child: Row(
                      mainAxisAlignment: MainAxisAlignment.center,
                      children: <Widget>[
                        SizedBox(
                          width: 30,
                        ),
                        RaisedButton(
                          padding: EdgeInsets.only(left: 40, right: 40),
                          shape: RoundedRectangleBorder(
                            borderRadius: BorderRadius.circular(14.0),
                          ),
                          color: Color(0xFF584846),
                          onPressed: (emailError==""&&passwordError=="") ?() async {
                            setState(() {
                              busy = true;
                            });
                            await _auth
                                .signInWithEmailAndPassword(
                                    email: email, password: password)
                                .catchError((e) {
                              setState(() {
                                busy = false;
                              });
                              if (e.code == "invalid-email") {
                                setState(() {
                                  loginSignUpError = "Invalid Email";
                                });
                              } else if (e.code == "user-disabled") {
```

```
        setState(() {
          loginSignUpError = "Account Disabled";
        });
      } else if (e.code == "user-not-found") {
        setState(() {
          loginSignUpError = "User Not Registered";
        });
      } else if (e.code == "wrong-password") {
        setState(() {
          loginSignUpError = "Wrong Password Entered";
        });
      } else {
        setState(() {
          loginSignUpError = e.toString();
        });
      }
    }).then((temp) {
      if (temp != null) {
        if (temp.user != null) {
          setState(() {
            loginSignUpError = "";
          });
          Navigator.pushReplacementNamed(context, HomePage.id);
        }
      }
    });
  }:null,
  child: Text('LOGIN',
      style: TextStyle(color: Colors.white)),
),
SizedBox(
  width: 30,
),
RaisedButton(
  padding: EdgeInsets.only(left: 40, right: 40),
  shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(14.0),
  ),
  color: Color(0xFF584846),
  onPressed:  (emailError==""&&passwordError=="") ?(){
    setState(() {
      busy = true;
    });
    _auth
        .createUserWithEmailAndPassword(
            email: email, password: password)
        .catchError((e) {
      setState(() {
        busy = false;
```

```
                  });
                  if (e.code == "invalid-email") {
                    setState(() {
                      loginSignUpError = "Invalid Email";
                    });
                  } else if (e.code == "email-already-in-use") {
                    setState(() {
                      loginSignUpError =
                          "Account Already Exists , please login";
                    });
                  } else if (e.code == "weak-password") {
                    setState(() {
                      loginSignUpError = "User Not Registered";
                    });
                  } else if (e.code == "operation-not-allowed") {
                    setState(() {
                      loginSignUpError =
                          "Signups are temporarily disabled";
                    });
                  } else {
                    setState(() {
                      loginSignUpError = e.toString();
                    });
                  }
                }).then((value) {
                  if (value != null) {
                    if (value.user != null) {
                      setState(() {
                        loginSignUpError = "";
                      });
                      Navigator.pushReplacementNamed(context, HomePage.id);
                    }
                  }
                });
                print('push');
              }:null,
              child: Text(
                'SIGN UP',
                style: TextStyle(color: Colors.white),
              ),
            ),
          ],
        ),
      ),
    ],
  ),
),
busy
    ? Container(
          color: Colors.white54,
          child: Center(
            child: CircularProgressIndicator(),
          ),
        )
       : SizedBox(height: 0)
    ],
   ),
  ),
 );
 }
}
```

5.4 Inventorypage.dart

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

class InventoryPage extends StatefulWidget {
 InventoryPage({Key key}) : super(key: key);

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

class _InventoryPageState extends State<InventoryPage> {
 @override
 Widget build(BuildContext context) {
  return Container(
   child: Scaffold(
    backgroundColor: Colors.grey[200],
    appBar: AppBar(
     title: Text("Inventory Page"),
    ),
    body: StreamBuilder(
     stream: FirebaseFirestore.instance.collection("books").snapshots(),
     builder: (context, snapshot) {
       if(!snapshot.hasData){
        return Center(child: CircularProgressIndicator());
       }
   return Padding(
    padding: const EdgeInsets.all(8.0),
    child: ListView.builder(
      itemCount: snapshot.data.documents.length,
      itemBuilder: (context, index) {
       DocumentSnapshot books =
         snapshot.data.documents[index];
       return Card(
                child: ListTile(
                  title: Text(books['name']),
                  subtitle: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    mainAxisSize: MainAxisSize.min,
                    children: [
                      Text(books['author']),
                      Text(books['genre']),
                      Text(books['isbn']),
                    ],
                  ),
                  trailing: Text(books['publisher']),
                ),
              );
            },
          ),
      );
          },
        ),
      ),
    );
  }
}
```

5.5 issue_book.dart

```
import 'package:flutter/material.dart';
import 'package:Library/model/result.dart';

class ScorePage extends StatefulWidget {
  static const routeName = "/scorePage";

  const ScorePage({Key key}) : super(key: key);

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

class _ScorePageState extends State<ScorePage>
    with SingleTickerProviderStateMixin {
  AnimationController _controller;
  Animation<double> _sizeAnimation;
  Animation<double> _marginAnimation;
  Animation<double> _opacityAnimation;

  @override
  void initState() {
    super.initState();
    _controller =
        AnimationController(vsync: this, duration: Duration(seconds: 5));

    _sizeAnimation = Tween<double>(begin: 0, end: 1).animate(
      CurvedAnimation(
        parent: _controller,
        curve: Interval(0, 0.3, curve: Curves.easeOut),
      ),
    );

    _marginAnimation = Tween<double>(begin: 0, end: 1).animate(
      CurvedAnimation(
        parent: _controller,
        curve: Interval(0.3, 0.7, curve: Curves.bounceOut),
      ),
    );

    _opacityAnimation = Tween<double>(begin: 0, end: 1).animate(
      CurvedAnimation(
        parent: _controller,
        curve: Interval(0.7, 1, curve: Curves.easeOutCubic),
      ),
    );

    _controller.forward();
  }

  void playAgain() {
    Navigator.of(context)
        .popUntil(ModalRoute.withName(Navigator.defaultRouteName));
  }

  @override
  void dispose() {
    _controller.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    final Result result = ModalRoute.of(context).settings.arguments as Result;
    double height = MediaQuery.of(context).size.height;

    Widget marks = Text(
      "${result.marksObtained} / ${result.totalMarks}",
      style: TextStyle(
        fontSize: Theme.of(context).textTheme.headline5.fontSize,
        fontWeight: FontWeight.bold,
      ),
    );

    Widget comment = Text(
      "${result.comment}",
      style: TextStyle(
        fontSize: Theme.of(context).textTheme.headline4.fontSize - 4,
      ),
    );

    Widget playAgainButton = Padding(
      padding: const EdgeInsets.only(bottom: 48.0),
      child: InkWell(
        onTap: playAgain,
        child: Container(
          padding: const EdgeInsets.symmetric(
            horizontal: 32,
            vertical: 16,
          ),
          decoration: BoxDecoration(
            gradient: LinearGradient(
              begin: Alignment.centerLeft,
              end: Alignment.bottomRight,
              colors: [
                Theme.of(context).primaryColor.withOpacity(0.8),
                Theme.of(context).accentColor.withOpacity(0.8),
              ],
            ),
          ),
          child: Text(
            "Play Again",
            style: Theme.of(context).textTheme.headline6,
          ),
        ),
      ),
    );

    return Material(
      child: AnimatedBuilder(
        animation: _controller,
        builder: (_, child) {
          return SafeArea(
            child: Stack(
              fit: StackFit.expand,
              children: [
                Column(
                  children: [
                    ScaleTransition(
                      scale: _sizeAnimation,
                      child: Container(
                        margin: EdgeInsets.only(
                          top: _marginAnimation.value * height * 0.25,
                        ),
                        alignment: Alignment.center,
                        decoration: BoxDecoration(
                          shape: BoxShape.circle,
                          color: Theme.of(context).primaryColor,
                        ),
                        height: 150,
                        width: 150,
                        child: FittedBox(
                          child: marks,
                        ),
                      ),
                    ),
                  ],
                ),
                FadeTransition(
                  opacity: _opacityAnimation,
                  child: child,
                ),
              ],
            ),
          );
        },
        child: Column(
          mainAxisSize: MainAxisSize.max,
          mainAxisAlignment: MainAxisAlignment.end,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            comment,
            SizedBox(height: 80),
            playAgainButton,
          ],
        ),
      ),
    );
  }
}
```

5.5 score_page.dart

```
import 'package:meta/meta.dart';

class LibraryParameter {
  static const API = "https://opentdb.com/api.php";
  final int _amount;
  final int _category;
  final String _difficulty;
  final String _type;

  const LibraryParameter(
      {@required int amount, int category, String difficulty, String type})
      : this._amount = amount,
        this._category = category,
        this._difficulty = difficulty,
        this._type = type,
        assert(amount != null);

  String get type => _type;

  String get difficulty => _difficulty;

  int get category => _category;

  int get amount => _amount;

  Map<String, dynamic> toJson() {
    final Map<String, dynamic> data = new Map<String, dynamic>();
    data['amount'] = this._amount;
    data['category'] = this._category;
    data['difficulty'] = this._difficulty;
    data['type'] = this._type;
    return data;
  }

  @override
  String toString() {
    return API +
        "?amount=$_amount" +
        "${_category == 0 ? "" : "&category=" + _category.toString()}" +
        "${_difficulty == "" ? "" : "&difficulty=" + difficulty}" +
        "${_type == "" ? "" : "&type=" + type}";
  }
}
```

5.6 Libraries used / API'S

```
cached_network_image: ^2.0.0
cloud_firestore: 3.1.6
firebase_auth: 3.3.5
provider: ^3.2.0
shared_preferences: ^0.5.6+3
random_string: ^2.0.1
```

Chapter 6

TESTING

6.1 Introduction

Testing is a process of executing a program with the interest of finding an error. A good test is one that has high probability of finding the yet undiscovered error. Testing should systematically uncover different classes of errors in a minimum amount of time with a minimum number of efforts. Two classes of inputs are provided to test the process

A software configuration that includes a software requirement specification, a design specification and source code.

A software configuration that includes a test plan and procedure, any testing tool and test cases and their expected results.

6.2 Levels of Testing

6.2.1 Unit Testing

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

Unit testing is commonly automated, but may still be performed manually. The objective in unit testing is to isolate a unit and validate its correctness. A manual approach to unit testing may employ a step-by-step instructional document. The unit testing is the process of testing the part of the program to verify whether the program is working correct or not. In this part the main intention is to check the each and every input which we are inserting to our file. Here the validation concepts are used to check whether the program is taking the inputs in the correct format or not.

Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier. Unit test cases embody characteristics that are critical to the success of the unit.

6.2.2 Integration Testing

Integration testing is also taken as integration and testing this is the major testing process where the units are combined and tested. Its main objective is to verify whether the major parts of the program is working fine or not. This testing can be done by choosing the options in the program and by giving suitable inputs.

6.2.3 System Testing

System testing is defined as testing of a complete and fully integrated software product. This testing falls in black-box testing wherein knowledge of the inner design of the code is not a pre-requisite and is done by the testing team. System testing is done after integration testing is complete. System testing should test functional and non-functional requirements of the software.

6.2.4 Validation Testing

In this, requirements established as part of software requirements analysis are validated against the software that has been constructed. Validation testing provides final assurance that software meets all functional, behavioral and performance requirements. Validation can be defined in many ways but a simple definition is that validation succeeds when software Function in a manner that can be reasonably by the customer.

1. Validation test criteria
2. Configuration review
3. Alpha and Beta testing (conducted by end user)

6.2.5 Output Testing

After preparing test data, the system under study is tested using the test data. While testing the system using test data, errors are again uncovered and corrected by using above testing and corrections are also noted for future use.

6.2.6 User Acceptance Testing

User acceptance testing is a type of testing performed by the end user or the client to verify/accept the software application to the production environment.

User Acceptance Testing is done in the final phase of testing.

Chapter 7

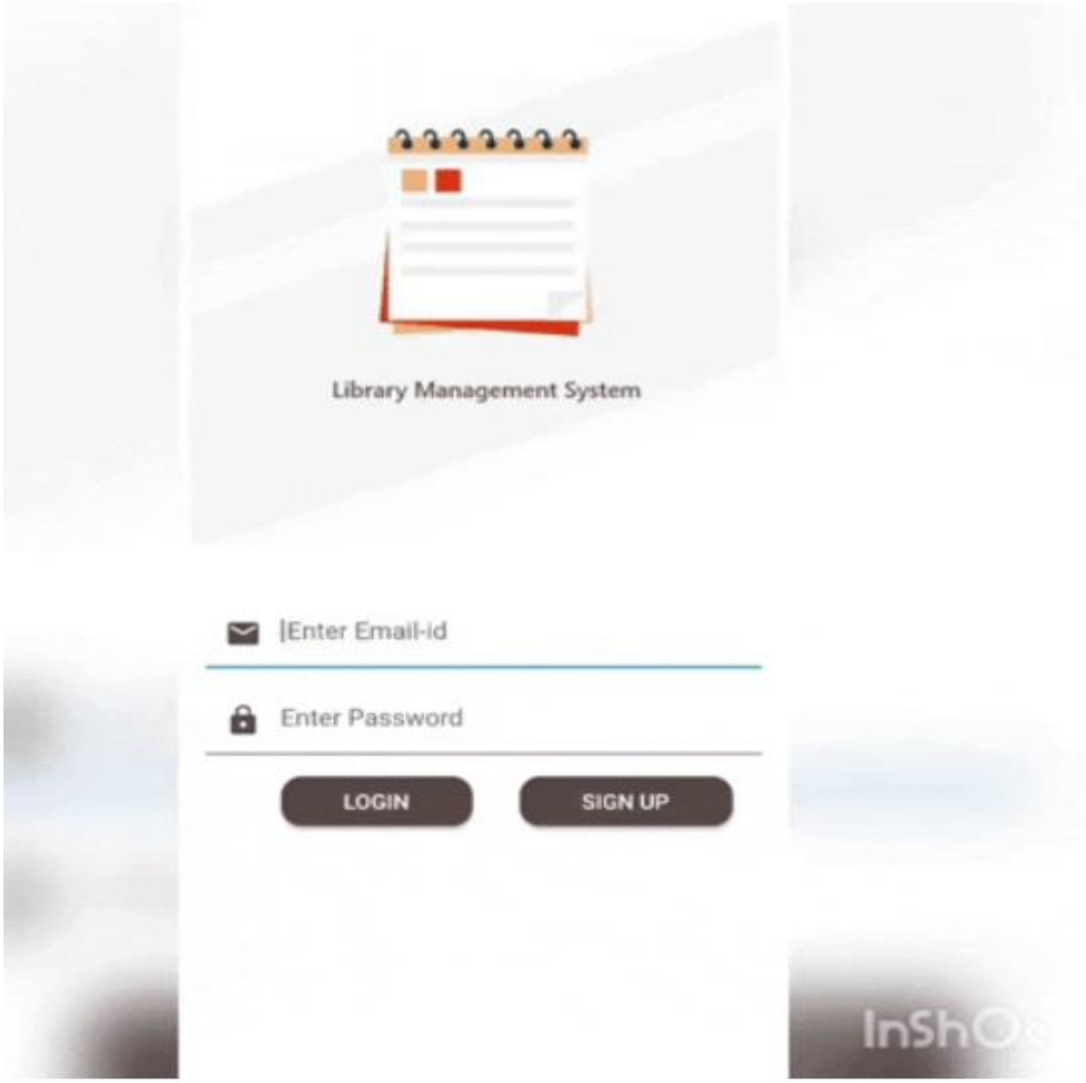
DISCUSSION OF RESULTS

7.1 Home page

This is the landing page of the application where we can select all the categories of the topics available for the Library.

7.2 Login signup page

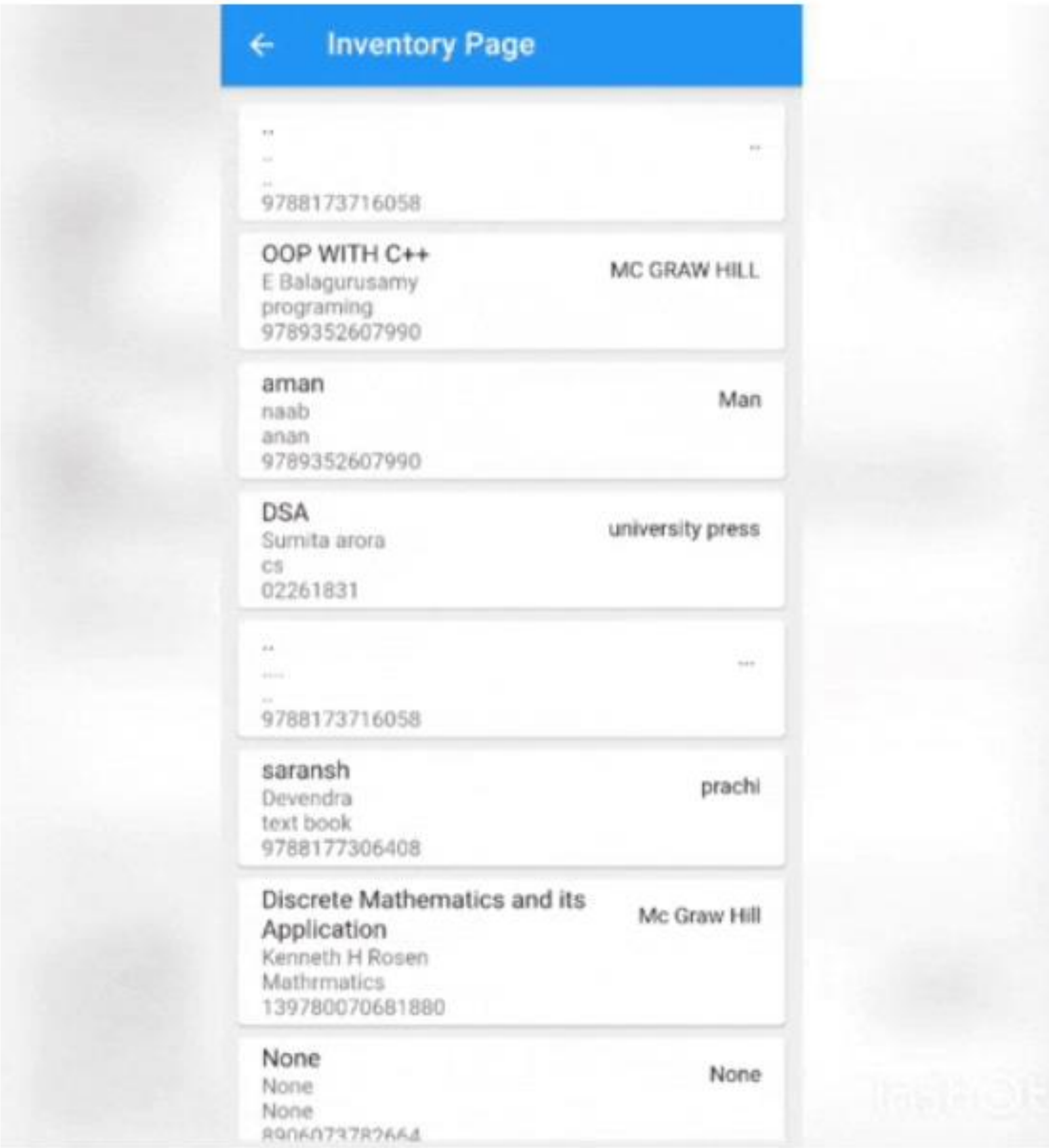
This page allows user to login to their account. In this page user have to enter their email id and password. If user don’t have their account, they can create their new account.



Library Management System

7.3 Inventory page

It shows all the books available in the inventory.



Inventory Page

..
..
..
9788173716058
..

OOP WITH C++
E Balagurusamy
programing
9789352607990
MC GRAW HILL

aman
naab
anan
9789352607990
Man

DSA
Sumita arora
cs
02261831
university press

..
....
..
9788173716058
...

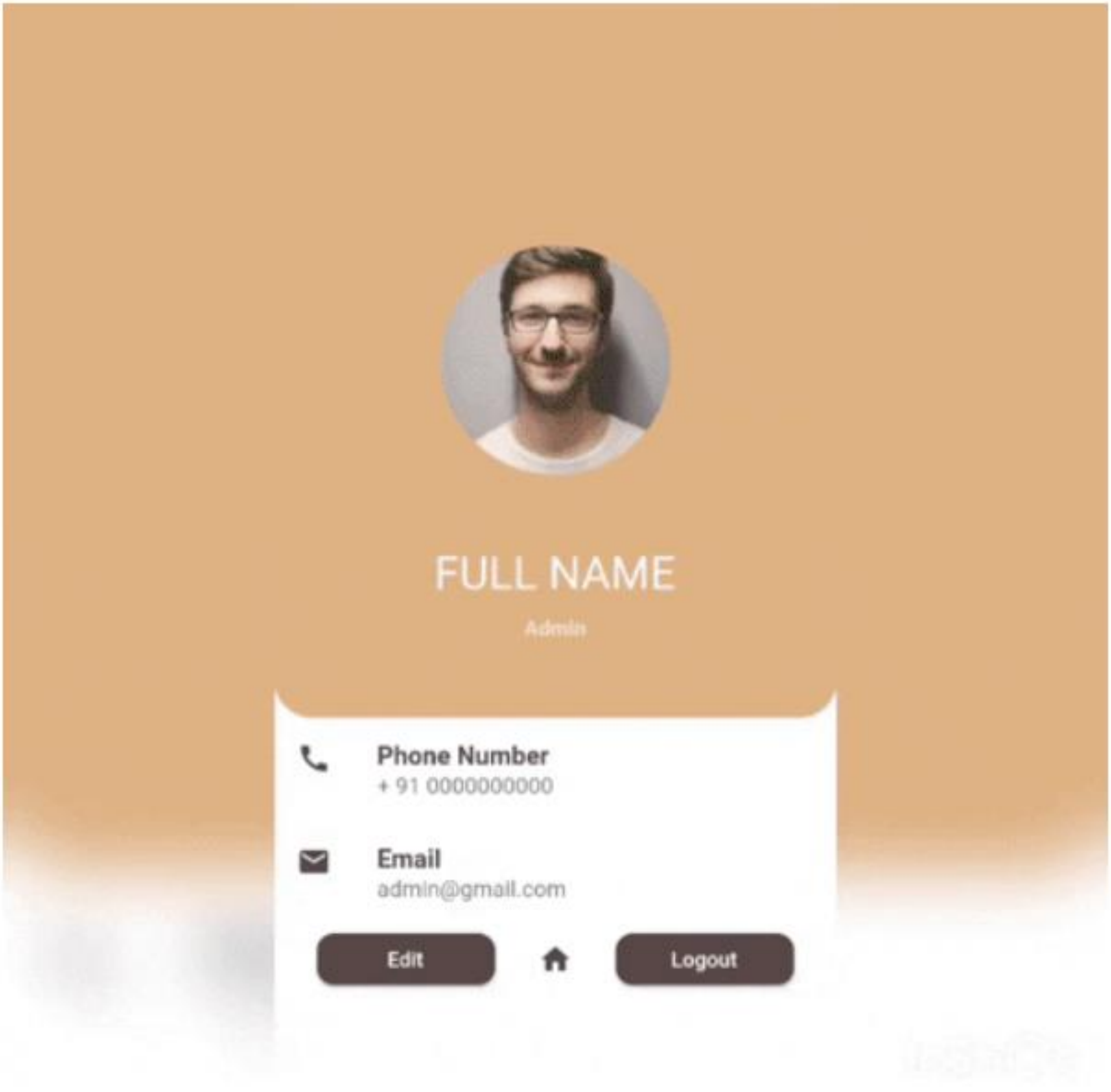
saransh
Devendra
text book
9788177306408
prachi

Discrete Mathematics and its Application
Kenneth H Rosen
Mathrmatics
139780070681880
Mc Graw Hill

None
None
None
8906073782664
None

7.4 Author Page

It shows details of author.



FULL NAME

Admin

Phone Number
+ 91 0000000000

Email
admin@gmail.com

Chapter 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

- The Library App is very important in future learning systems; it can be easily adopted by educational institutions and universities in order to make the Library more flexible and safer.

- The proposed system is easy and flexible for future maintenance and development because each subsystem can be handled separately without influence on another system.

8.2 Future work

- We can add the functionality for a user to access without google or email.

- We can add the functionality for user to make payment online.

- We can have an admin functionality in which admin can check for the reports of the users of the library.

Chapter 9

REFERENCES

- https://flutter.dev/
- https://developers.google.com/learn/pathways/intro-to-flutter
- Beginning Flutter: A Hands On Guide to App Development by Marco L Napoli
- https://stackoverflow.com/
- https://www.geeksforgeeks.or