**MES’s INSTITUTE OF MANAGEMENT AND CAREER COURSES**



**“Exlog – Expense Logger”**

Submitted By

**Rahul Kumar 2201110**

Guided By

**Dr. Swapnaja Patwardhan**

Submitted To

**MES’s IMCC**

131, Mayur Colony, Kothrud, Pune-411 038, MH, INDIA

2022-2023

**Mini Project Index**

|  |  |  |
| --- | --- | --- |
| **Activity No.** | **Description** | **Page No.** |
| 1 | Chapter 1 – Introduction   * 1. Project Scope |  |
| 2 | Chapter 2 – Design  2.1 Business Use Case  2.2 Class diagram  2.3 Database Design/ Data Dictionary |  |
| 3 | Chapter 3 - User Interface  3.1 Input design  3.2 Output Design |  |
| 4 | Chapter 4 – Code |  |
| 5 | Bibliography |  |

**Introduction**

The primary purpose of the Exlog – Expense Logger application is to keep track of all the expenses one pays daily. The main elements of the apps are as follows:

1. User Registration

2. User Login

3. User Authentication

4. Dashboard

5. Expense addition

6. View expenses

**Project Scope**

1. User Registration: Allow users to create accounts using name, email and password.
2. User Login: Allow users to sign-in to their accounts using email and password.
3. User Authentication: Allow users to validate their credentials.
4. Dashboard: Allow users to create a dashboard where amount related information can be viewed along with paths to different pages.
5. Expense Addition: Allow users to add daily expenses using a form.
6. View Expenses: Allow users to view all expenses in a list view.

**Technologies Used**

1. Language : Dart
2. Framework: Flutter
3. Tools : VS Code
4. Database : SQLite

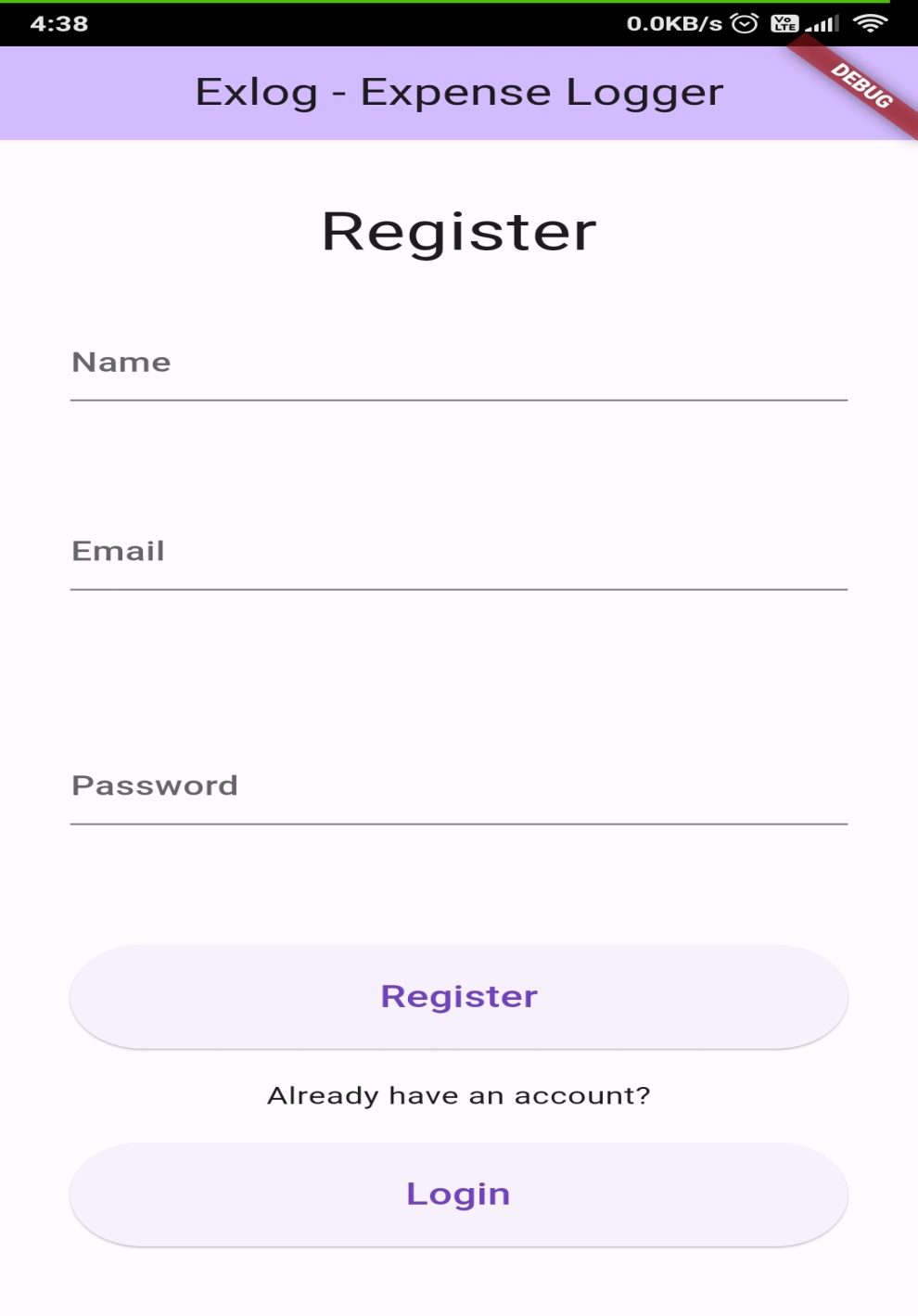
**Design**

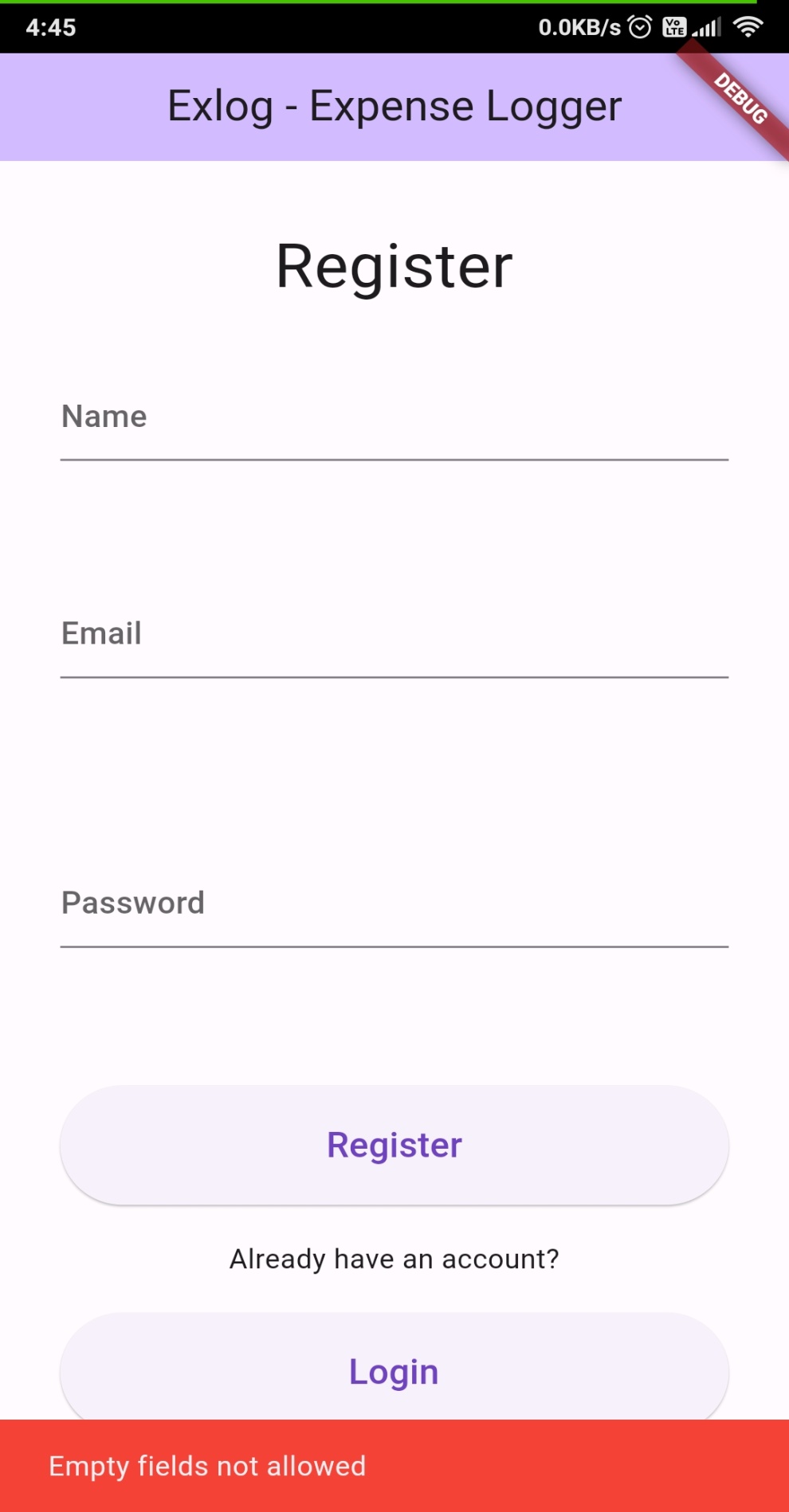
Use Case

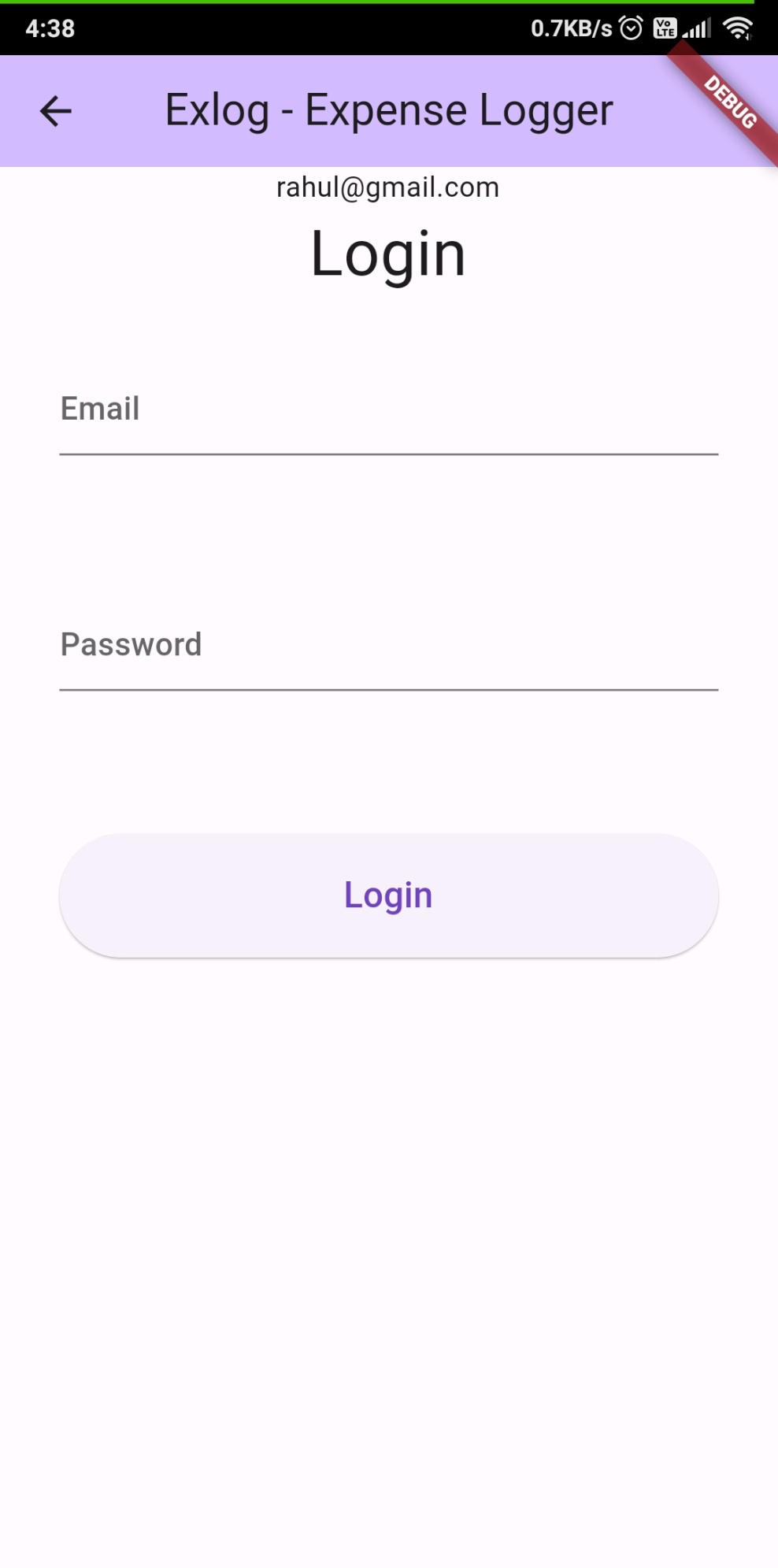
Class diagram

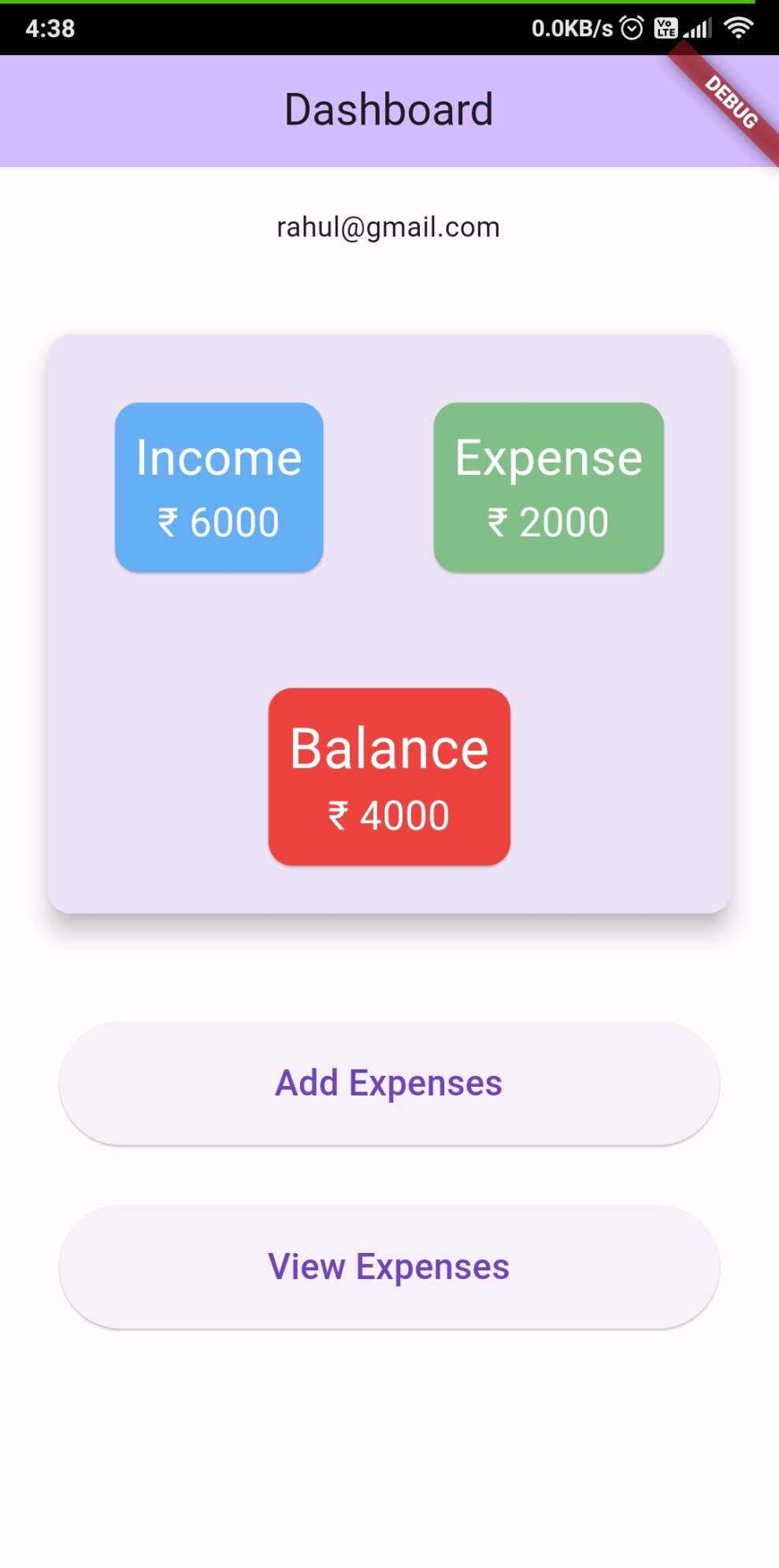
Database Design/ Data Dictionary

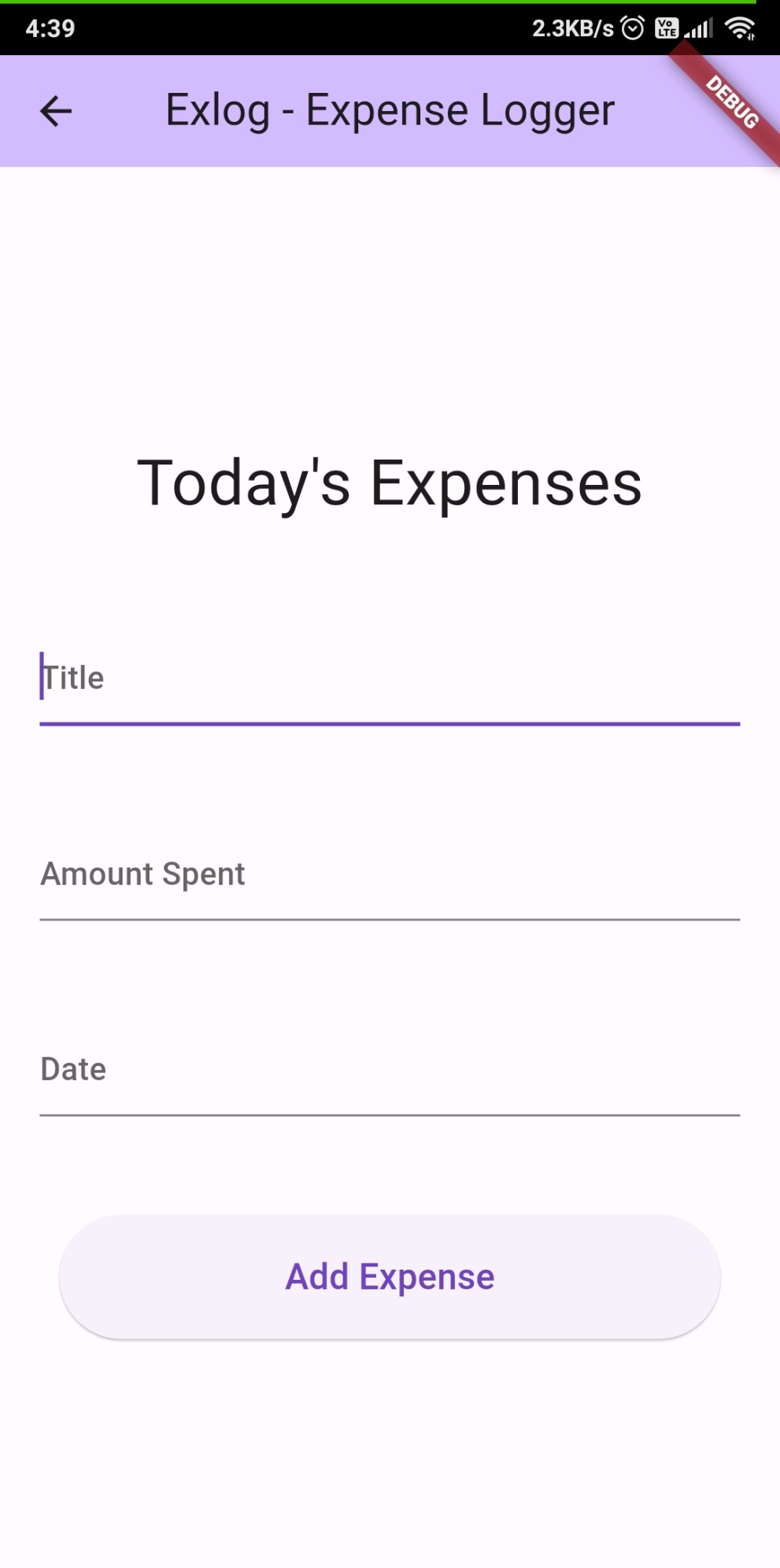
**User Interface**





****

****

****

**Code**

Main.dart

// import 'dart:math';

import 'package:flutter/material.dart';

import 'package:sqflite/sqflite.dart';

import 'login.dart';

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return SafeArea(

child: MaterialApp(

title: 'Exlog',

theme: ThemeData(

colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),

useMaterial3: true,

),

home: const RegistrationPage(title: 'Exlog - Expense Logger'),

),

);

}

}

class RegistrationPage extends StatefulWidget {

const RegistrationPage({super.key, required this.title});

final String title;

@override

State<RegistrationPage> createState() => \_RegistrationPageState();

}

class \_RegistrationPageState extends State<RegistrationPage> {

// Database helper

final \_formKey = GlobalKey<FormState>();

TextEditingController nameController = TextEditingController();

TextEditingController emailController = TextEditingController();

TextEditingController passwordController = TextEditingController();

\_RegistrationPageState() {

initDb();

}

late Database \_db;

Future<void> initDb() async {

\_db = await openDatabase('registration.db');

await \_db.execute(

"CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY, name TEXT, email TEXT)");

}

// Insert a new user record

void \_registerUser(String name, String email) async {

await \_db.insert('users', {'name': name, 'email': email});

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

backgroundColor: Theme.of(context).colorScheme.inversePrimary,

title: Text(widget.title),

centerTitle: true,

),

body: Center(

child: SingleChildScrollView(

child: Form(

key: \_formKey,

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

const Text(

'Register',

style: TextStyle(fontSize: 32.0),

),

Padding(

padding: const EdgeInsets.all(30.0),

child: TextField(

controller: nameController,

decoration: const InputDecoration(hintText: 'Name'),

),

),

const SizedBox(height: 5.0),

Padding(

padding: const EdgeInsets.all(30.0),

child: TextField(

controller: emailController,

decoration: const InputDecoration(hintText: 'Email'),

keyboardType: TextInputType.emailAddress,

),

),

const SizedBox(height: 32.0),

Padding(

padding: const EdgeInsets.all(30.0),

child: TextField(

controller: passwordController,

obscureText: true,

decoration: const InputDecoration(hintText: 'Password'),

keyboardType: TextInputType.visiblePassword,

),

),

const SizedBox(height: 32.0),

Column(

crossAxisAlignment: CrossAxisAlignment.stretch,

children: [

Padding(

padding: const EdgeInsets.symmetric(horizontal: 30, vertical: 10),

child: ElevatedButton(

onPressed: () {

// Validate form for empty fields

if (nameController.text.isEmpty ||

emailController.text.isEmpty ||

passwordController.text.isEmpty) {

ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(

content: Text('Empty fields not allowed'),

backgroundColor: Colors.red,

),

);

} else {

// navigate to login page

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => LoginPage(

email: emailController.text,

password: passwordController.text,

),

),

);

}

},

child: const Padding(

padding: EdgeInsets.all(18.0),

child: Text(

"Register",

style: TextStyle(

fontSize: 18,

),

),

),

),

),

],

),

const SizedBox(height: 8.0),

const Text('Already have an account?'),

const SizedBox(height: 8.0),

Column(

crossAxisAlignment: CrossAxisAlignment.stretch,

children: [

Padding(

padding: const EdgeInsets.symmetric(horizontal: 30, vertical: 10),

child: ElevatedButton(

onPressed: () {

// Navigate to login page

String email = 'rahul@gmail.com';

String password = '1234';

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => LoginPage(

email: email,

password: password,

),

),

);

},

child: const Padding(

padding: EdgeInsets.all(18.0),

child: Text(

"Login",

style: TextStyle(

fontSize: 18,

),

),

),

),

),

],

)

],

),

),

),

),

);

}

}

Login.dart

import 'package:exlog/dashboard.dart';

import 'package:flutter/material.dart';

class LoginPage extends StatefulWidget {

final String email;

final String password;

const LoginPage({super.key, required this.email, required this.password});

@override

State<LoginPage> createState() => \_LoginPageState();

}

class \_LoginPageState extends State<LoginPage> {

final \_formKey = GlobalKey<FormState>();

TextEditingController emailController = TextEditingController();

TextEditingController passwordController = TextEditingController();

@override

Widget build(BuildContext context) {

return SafeArea(

child: Scaffold(

appBar: AppBar(

backgroundColor: Theme.of(context).colorScheme.inversePrimary,

title: const Text('Exlog - Expense Logger'),

centerTitle: true,

),

body: Center(

child: Column(

children: [

Text(widget.email),

SingleChildScrollView(

child: Form(

key: \_formKey,

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

const Text(

'Login',

style: TextStyle(fontSize: 32.0),

),

Padding(

padding: const EdgeInsets.all(30.0),

child: TextField(

controller: emailController,

decoration: const InputDecoration(hintText: 'Email'),

keyboardType: TextInputType.emailAddress,

),

),

const SizedBox(height: 10.0),

Padding(

padding: const EdgeInsets.all(30.0),

child: TextField(

controller: passwordController,

obscureText: true,

decoration:

const InputDecoration(hintText: 'Password'),

),

),

const SizedBox(height: 32.0),

Column(

crossAxisAlignment: CrossAxisAlignment.stretch,

children: [

Padding(

padding: const EdgeInsets.symmetric(horizontal: 30, vertical: 10),

child: ElevatedButton(

onPressed: () {

// Validate form for empty fields

if (emailController.text.isEmpty ||

passwordController.text.isEmpty) {

ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(

content: Text('Empty fields not allowed'),

backgroundColor: Colors.red,

),

);

} else if (\_formKey.currentState!.validate()) {

if (emailController.text == widget.email &&

passwordController.text == widget.password) {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => DashboardPage(

email: emailController.text,

password: passwordController.text,

),

),

);

} else {

ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(

content: Text('Invalid Data'),

backgroundColor: Colors.red,

),

);

}

}

},

child: const Padding(

padding: EdgeInsets.all(18.0),

child: Text(

"Login",

style: TextStyle(

fontSize: 18,

),

),

),

),

),

],

),

],

),

),

),

],

),

),

),

);

}

}

Dashboard.dart

import 'package:exlog/add\_expenses.dart';

import 'package:exlog/view\_expenses.dart';

import 'package:flutter/material.dart';

class DashboardPage extends StatelessWidget {

final String email;

final String password;

const DashboardPage({super.key, required this.email, required this.password});

@override

Widget build(BuildContext context) {

return WillPopScope(

onWillPop: () async => false,

child: SafeArea(

child: Scaffold(

appBar: AppBar(

automaticallyImplyLeading: false,

backgroundColor: Theme.of(context).colorScheme.inversePrimary,

title: const Text('Dashboard'),

centerTitle: true,

),

body: Center(

child: Column(

children: [

Padding(

padding: const EdgeInsets.all(20.0),

child: Text(email),

),

Padding(

padding: const EdgeInsets.all(20.0),

child: Card(

elevation: 10,

child: Column(

children: [

Padding(

padding: const EdgeInsets.all(30.0),

child: Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

Column(

children: [

Card(

color: Colors.blue[300],

child: const Padding(

padding: EdgeInsets.all(10.0),

child: Column(

children: [

Row(

children: [

Text(

"Income",

style: TextStyle(

fontSize: 25,

color: Colors.white,

),

),

],

),

Row(

children: [

Text(

"₹ 6000",

style: TextStyle(

fontSize: 20,

color: Colors.white,

),

),

],

),

],

),

),

),

],

),

Column(

children: [

Card(

color: Colors.green[300],

child: const Padding(

padding: EdgeInsets.all(10.0),

child: Column(

children: [

Row(

children: [

Text(

"Expense",

style: TextStyle(

fontSize: 25,

color: Colors.white,

),

),

],

),

Row(

children: [

Text(

"₹ 2000",

style: TextStyle(

fontSize: 20,

color: Colors.white,

),

),

],

),

],

),

),

),

],

),

],

),

),

Padding(

padding: const EdgeInsets.all(20.0),

child: Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Card(

color: Colors.red[500],

child: const Padding(

padding: EdgeInsets.all(10.0),

child: Column(

children: [

Row(

children: [

Text(

"Balance",

style: TextStyle(

fontSize: 28,

color: Colors.white,

),

),

],

),

Row(

children: [

Text(

"₹ 4000",

style: TextStyle(

fontSize: 20,

color: Colors.white,

),

),

],

),

],

),

),

),

],

),

),

],

),

),

),

Padding(

padding: const EdgeInsets.all(30.0),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

crossAxisAlignment: CrossAxisAlignment.stretch,

children: [

ElevatedButton(

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => const AddExpenses(),

),

);

},

child: const Padding(

padding: EdgeInsets.all(18.0),

child: Text(

"Add Expenses",

style: TextStyle(

fontSize: 18,

),

),

),

),

const SizedBox(

height: 30,

),

ElevatedButton(

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => const ViewExpenses(),

),

);

},

child: const Padding(

padding: EdgeInsets.all(18.0),

child: Text(

"View Expenses",

style: TextStyle(

fontSize: 18,

),

),

),

),

],

),

),

],

),

),

),

),

);

}

}

**Bibliography**

1. Dart Official Documentation
2. Flutter Official Documentation
3. Sqlite Official Documentation
4. Sqflite Official Documentation
5. Dartpad.io
6. Github.com
7. Google bard
8. Youtube.com