

CIRCULAR APPLICATION (ASSIST)

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

By

Barbie Saha (Reg. No. 37110078)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

APRIL 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Barbie Saha** (Reg.No. 37110078) and **Bhavya Mishra** (Reg.No. 37110093) who carried out the project entitled "**CIRCULAR APPLICATION**" under our supervision from NOVEMBER 2020 to MARCH 2021.

INTERNAL GUIDE

Murari D Kamalesh, M.E.

HEAD OF THE DEPARTMENT

Dr. S. Vigneshwari, M.E.,Ph.d.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I, **Barbie Saha (Reg.No. 37110078)** hereby declare that the Project Report entitled "**CIRCULAR APPLICATION**" done by us under the guidance of **Murari D Kamalesh M.E.**, is submitted in partial fulfillment of the requirements for the award of **Bachelor of Engineering** degree in **Computer Science and Engineering**.

DATE:

PLACE:

Barbie Saha

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGMENT

I am pleased to acknowledge my sincere thanks to **Sathyabama Institute of Science and Technology Management** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey our thanks to **Dr. T. SASIKALA M.E., Ph.D.**, Dean, School of Computing and **Dr. L. LAKSHMANAN M.E., Ph.D.**, and **Dr. VIGNESHWARI M.E., Ph.D.**, Head of the Department, Department of Computer Science And Engineering for providing us necessary support and details at the right time during the progressive reviews.

I would like to express our sincere and deep sense of gratitude to our Project Guide **Murari D Kamalesh, M.E.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

I wish to express our 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

Mobile applications stood top among usability and user convenience. Many applications are available in the market to socialize a set of people or publically. Not many applications provide a comprehensive view of both use cases. In this project, we develop a mobile application that keeps track of a user's activity with a secured pipeline to communicate or to post his/her personal or official contributions.

This application is just a single hook to a sophisticated backend which is available as endpoints for other projects to hook as well and provide a great integration to their project. There are no constraints bounding this backend, one can simply develop their part of integration/update with time and make a merge. This makes the scope of this application grow even without its developers.

TABLE OF CONTENTS

TITLE	PAGE.NO
ABSTRACT	v
LIST OF FIGURES	viii
ABBREVIATIONS	ix
1. INTRODUCTION	
1.1 AIM	01
1.2 SCOPE	01
1.3 OBJECTIVES	02
2. PROJECT PLANS AND REQUIREMENTS	03
2.1. PLAN	03
2.2. REQUIREMENTS	03
2.3. DESIGN	04
2.4. OVERVIEW OF PLATFORM	04
2.4.1 FLUTTER	04
2.4.2 DART	05
2.4.3 PYTHON	05
2.4.4 REST API	06
2.4.5 JSON	06
2.4.6 POSTGRESQL	07

3. SYSTEM IMPLEMENTATION	08
3.1. ARCHITECTURE DESIGN	08
3.2. DESIGN USER INTERFACE	09
4. RESULTS AND DISCUSSION	13
4.1 RESULTS	13
4.2 DISCUSSION	13
5. CONCLUSION AND SUMMARY	17
5.1 SUMMARY	17
5.2 CONCLUSION	17
REFERENCES	18
APPENDICES	19

LIST OF FIGURES

LIST OF FIGURES

REF. No.	TITLE	PAGE.NO
3.1.1	Architecture Diagram	08
3.2.1.1	Login Screen	09
3.2.1.2	Register Screen	10
3.2.1.3	News Feed Screen	11
3.2.1.4	Search Screen	12
3.2.1.5	Profile Screen	12
3.2.1.6	Add Post Screen	13
4.2.1.1	Main Dashboard	14
4.2.2.1	User Dashboard	15
4.2.3.1	Follow Dashboard	16

LIST OF ABBREVIATIONS

Notation	EXPANSION
APP	Application
REST	Representational State Transfer
API	Application Program Interface
EPT	Endpoint
JSON	JavaScript Object Notation
IDE	Integrated Development Environment
SDK	Software Development Kit
UI	User Interface
UX	User Experience

CHAPTER-1

INTRODUCTION

This project is made with an intent to provide a smart communication application that will bring enhancement to the current circular notification procedure. It will serve as a peer to peer platform, hence no chance of data breach or exploitation.

Data privacy and security is the biggest threat to social media applications which are taken care of using OAuth2 and API key authentications. The second biggest threat comes up as polluted and manipulative content which compromises the authenticity of the platform. These threats are taken care of with precise threads of competitive and contemporary technologies.

1.1 AIM

In this world of growing technologies everything is digitalized. In this era, it's difficult to identify which information is authentic and which is fake. In this project, we have developed a mobile application that allows only authorized users to send information to its followers. Every transaction of information made through this application is secured using OAuth2 and API key authentications.

1.2 SCOPE

Some of the project's scope are:

- Easy to use Interface
- One API - Multiple applications
- Secure transaction of information
- Eliminate fake information
- Mass communication

- Multi-Platform

1.3 OBJECTIVES

The objectives of the system are as follows:

- Design a backend that can handle logins, registrations, and transactions of information.
- Store media files of the user in a secure server.
- Give only admins the privilege to post information.
- Information in multiple formats - text, file, etc.
- Allow users to follow higher authorities who post information.
- One-click Google/Apple login.
- Implement a pleasant user-experience.

CHAPTER 2

PROJECT PLANS AND REQUIREMENTS

2.1 PLANS

Planning is important before starting the project. It ensures that there's a proper plan for executing on strategic goals. The plans created before starting the project helps to manage time, cost, quality, change, risk, and issues.

The plans for this project are as follows:

1. Analyzing the project's objectives
2. Collection of requirements for the project
3. Designing and code the application using flutter.
4. Coding the backend and API using python (Django)
5. Testing

2.2 REQUIREMENTS

Hardware Requirements

- Ram – Above 2gb
- Internal storage – 80 GB (API Hosting)

Software Requirements

- Cloud VM – Ubuntu / Mac OS

Permission requirement

- Google/Apple – Gmail/Apple account

- Access to Internet permission
- API key to access the backend

2.3 DESIGN

For the purposes of this project, "design" means the visual design of the application as well as the implementation details that underlie it. The design should describe what the finished application will look like and how a person will make it work. A vital part of the design will be a set of transitions from one Window to another. The final product of the design phase is a connected Window that contains different structures with different operations.

The Main Window should contain these components:

1. AUTHENTICATION
2. FOLLOWING OPTION
3. UPDATES FEED
4. PLUGINS
5. ADMIN DASHBOARD.

2.4 OVERVIEW OF THE PLATFORM

2.4.1 FLUTTER:

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

Flutter's engine, written primarily in C++, provides low-level rendering support using Google's Skia graphics library. Additionally, it interfaces with platform-specific SDK such as those provided by Android and iOS. 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 will interact with Flutter via the Flutter Framework, which provides a modern, reactive framework, and a rich set of platform, layout and foundation widgets.

2.4.2 DART:

Dart is a client-optimized programming language for apps on multiple platforms. It is developed by Google and is used to build mobile, desktop, server, and web applications.

Dart is an object-oriented, class-based, garbage-collected language with C-style syntax. Dart can compile to either native code or JavaScript. It supports interfaces, mixins, abstract classes, reified generics, and type inference.

2.4.3 PYTHON (Django):

Django is a Python-based free and open-source web framework, which follows the model-template-view (MTV) architectural pattern. It is maintained by the Django Software Foundation(DSF), an independent organization established as a 501(c)(3) non-profit.

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

2.4.4 REST API & ENDPOINTS:

Representational state transfer (REST) is a software architectural style that defines a set of constraints to be used for creating Web services.

Web services that conform to the REST architectural style, called RESTful Web services, provide interoperability between computer systems on the Internet. RESTful Web services allow the requesting systems to access and manipulate textual stateless operations. Other kinds of Web services, such as SOAP Web services, expose their own arbitrary sets of operations.

In a RESTful Web service, requests made to a resource's URI will elicit a response with a payload formatted in HTML, XML, JSON, or some other format. The response can confirm that some alteration has been made to the stored resource, and the response can provide hypertext links to other related resources or collections of resources. When HTTP is used, as is most common, the operations (HTTP methods) available are GET, HEAD, POST, PUT, PATCH, DELETE, CONNECT, OPTIONS and TRACE.

2.4.5 JSON:

JSON is a language-independent data format. It was derived from JavaScript, but many modern programming languages include code to generate and parse JSON- format data. The official Internet media type for JSON is `application/json`. JSON file names use the extension `.json`.

2.4.6 PostgreSQL:

PostgreSQL, also known as Postgres, is a free and open-source relational database management system (RDBMS) emphasizing extensibility and technical standards compliance. It is designed to handle a range of workloads, from single machines to data warehouses or Web services with many concurrent users. It is the default database for macOS Server and is also available for Linux, FreeBSD, OpenBSD, and Windows.

PostgreSQL features transactions with Atomicity, Consistency, Isolation, Durability (ACID) properties, automatically updatable views, materialized views, triggers, foreign keys, and stored procedures.^[14] PostgreSQL is developed by the PostgreSQL Global Development Group, a diverse group of many companies and individual contributors.

2.4.7 Cloud Storage:

Cloud Storage is built for app developers who need to store and serve user-generated content, such as photos or videos.

Cloud Storage for Firebase is a powerful, simple, and cost-effective object storage service built for Google scale. The Firebase SDKs for Cloud Storage add Google security to file uploads and downloads for your Firebase apps, regardless of network quality. You can use our SDKs to store images, audio, video, or other user-generated content. On the server, you can use Google Cloud Storage, to access the same files.

CHAPTER 3

SYSTEM IMPLEMENTATION

3.1 ARCHITECTURE DESIGN

This flowchart represents the workflow of the project. This diagrammatic Representation illustrates a solution to the given problem. This helps in systematic Designing of the project

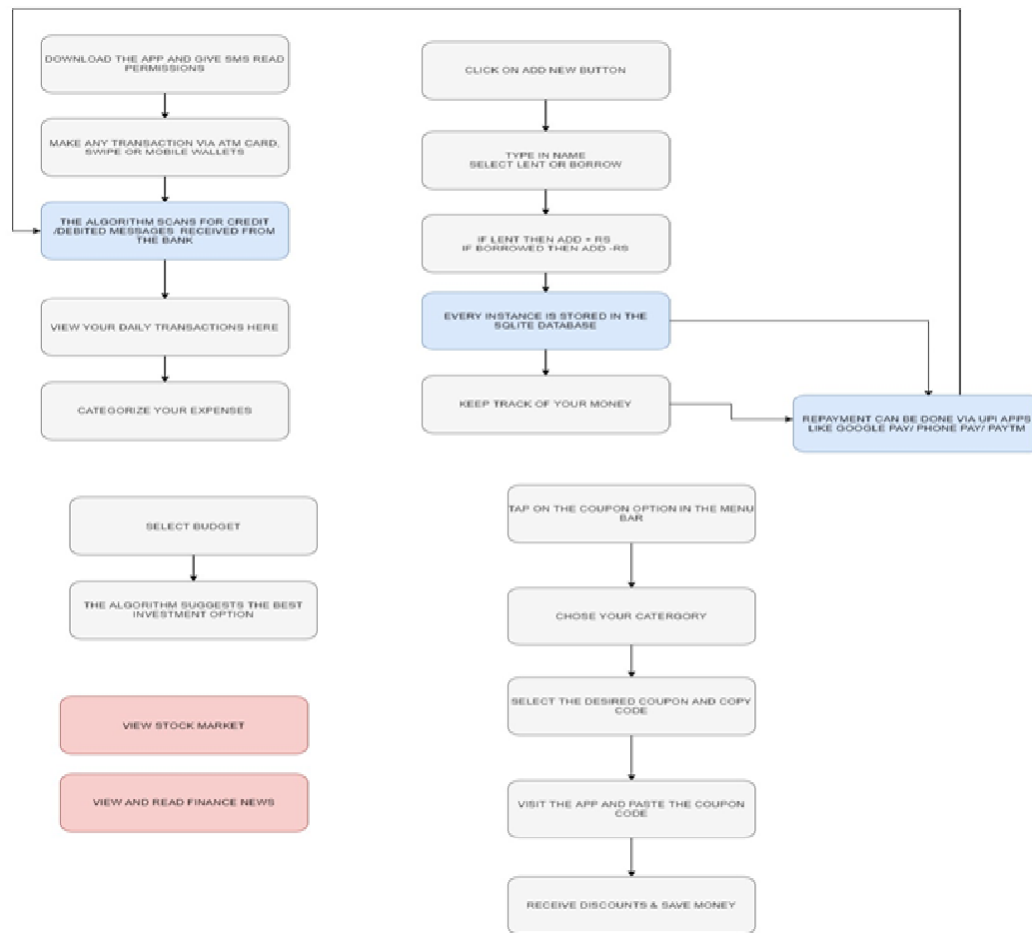


Fig 3.1 Flow diagram

3.2 Design User Interface

The application interface should be minimal and purposeful. The UI should enhance the user experience.

3.2.1 Layout

Designs LOGIN

PAGE

This page communicates with the Google Firebase as well as the backend API to log in the user into the application. First, it signs in the user to the Google account using the Firebase console and gets it's unique user-id then it signs in the user using the user-id into the backend server. This 2-way sign in procedure ensures valid email ids are used to sign in to the application.



Fig 3.2.1.1 Login Screen

REGISTER PAGE

If the user is not registered in the back-end, this page ensures that the user registers in the Google Firebase as well as the backend API to use the application. Then, it signs in the user to the Google account using the Firebase console and gets its unique user-id after which it signs in the user using the user-id into the backend server.

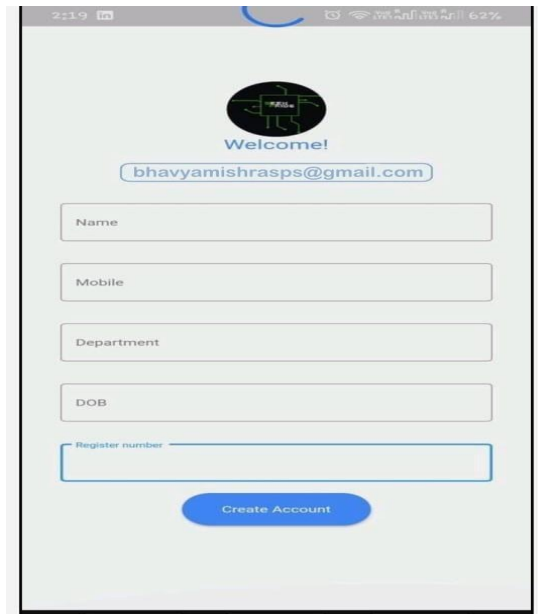
A screenshot of a mobile application's registration screen. At the top, there is a status bar showing the time 2:19, signal strength, Wi-Fi, and battery level at 62%. Below the status bar is a circular logo with a green and black design. Under the logo, the text "Welcome!" is displayed. Below this, the email address "bhavyamishrasps@gmail.com" is shown in a rounded rectangle. Below the email address are four input fields: "Name", "Mobile", "Department", and "DOB". Below these fields is a "Register number" field with a blue border. At the bottom of the form is a blue button labeled "Create Account".

Fig 3.2.1.2 Register Screen

NEWS FEED PAGE

The news feed is where the user can see all the posts from the user they follow. Only the information which has been posted by the authorized user can be seen by the followers after they have followed them.

The information posted by the authorized user before a particular follower has followed them cannot be seen in the news feed by the specific follower. If you are an authorized user, you'll be able to see the add post button where new information can be posted through the application which will be visible to the users they follow.



Fig 3.2.1.3 News Feed Screen

PROFILE PAGE

This is where the user can see their profile details which the application has collected from the user. The user can also log out from the application from the profile page.



Fig 3.2.1.5 Profile Screen

ADD POST PAGE

This page is only accessible by the authorized user from the news feed where they can see an 'Add post'

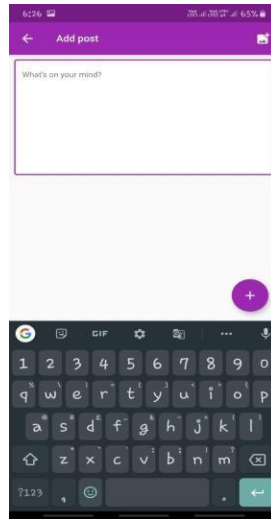


Fig 3.2.1.6 Add Post Screen

CHAPTER 4

RESULTS AND DISCUSSION

4.1 RESULT

The ASSIST application is built with an intent to provide a smart communication application that will bring enhancement to the current circular notification procedure. It will serve as a peer to peer platform, hence no chance of data breach or exploitation.

4.2 DISCUSSION

All the data that is saved by the application can be viewed and accessed by the administrator by logging in to the dashboard. The Dashboard contains all the data collected by the application. There are different sections in which the data are stored making it easier to access the type of data we want to access.

4.2.1 Dashboard

The main dashboard in which all the sections are shown with categories like Auth, Authentication and Authorization and Connection. We can also see the recent actions taken through the dashboard.

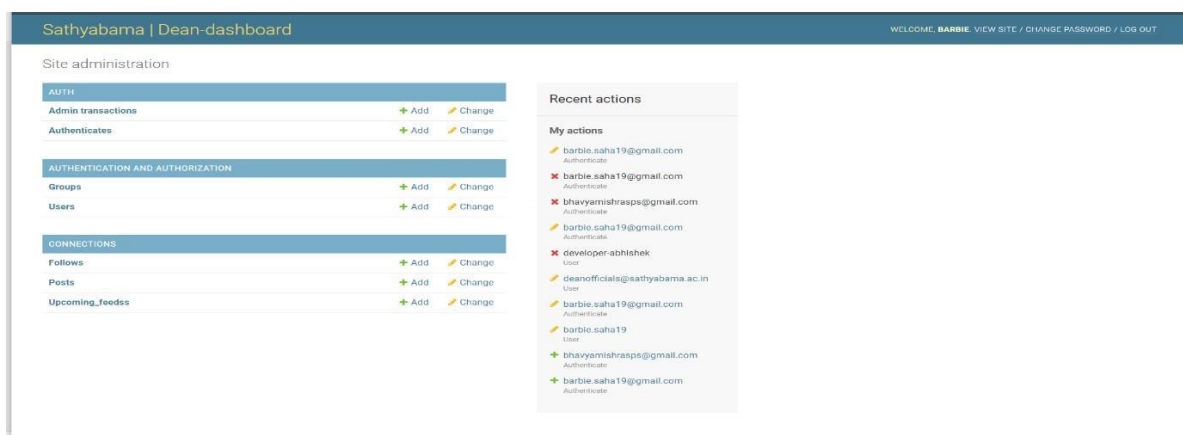


Fig 4.2.1.1 Main Dashboard

The Users dashboard contains all the information collected about the users that have registered in the application and has assigned them an unique uid a.k.a user id.

Sathyabama | Dean-dashboard

WELCOME, **BARBIE** [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Authentication and Authorization > Users

Select user to change

Search

Action: 0 of 2 selected Go

☐	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
☐	barbie.saha19	barbie.saha19@gmail.com	barbie	saha	✓
☐	deanofficials@sathyabama.ac.in	singh.s.kshetriya@gmail.com	Barbie	Saha	✓

2 users

FILTER

By staff status

All
Yes
No

By superuser status

All
Yes
No

By active

All
Yes
No

ADD USER +

Fig 4.2.2.1 User Dashboard

All the information posted by the authorized users can be seen from the Post dashboard. It contains the description of the post, user who has posted it, attachment link(if any), time, date and an unique pid a.k.a post id.

Select post to change

ADD POST +

Action: Go 0 of 100 selected

☐	ID	PID	DESCRIPTION	ATTACHMENT	TIME	DATE
☐	122	6ac6d55c-a138-4922-b007-f19580d73871	ok 2334567		13:17:45	2021-03-09
☐	121	5c196bbf-c615-4810-8b3d-be843a9531e5	ok 233456		13:16:50	2021-03-09
☐	120	847ae13b-ad78-464a-852b-32a3ff0b04b7	ok	https://firebasestorage.googleapis.com/v0/b/dean-officials.appspot.com/o/6d28e371-d176-426d-880c-bd770c9dfa7d%2Fuploads%2FScreenshot_20200309-122622_WhatsApp.jpg?alt=media&token=7c46212d-4286-453d-bb9e-64e46daf1480	13:16:35	2021-03-09
☐	119	e1b02e00-03c9-4c26-ab1f-f19913519e32	ok	https://firebasestorage.googleapis.com/v0/b/dean-officials.appspot.com/o/6d28e371-d176-426d-880c-bd770c9dfa7d%2Fuploads%2FScreenshot_20200309-122622_WhatsApp.jpg?alt=media&token=7c46212d-4286-453d-bb9e-64e46daf1480	13:15:02	2021-03-09
☐	118	f2c81161-7726-418c-b903-3a9a95be0f44	ok	https://firebasestorage.googleapis.com/v0/b/dean-officials.appspot.com/o/6d28e371-d176-426d-880c-bd770c9dfa7d%2Fuploads%2FScreenshot_20200309-122622_WhatsApp.jpg?alt=media&token=7c46212d-4286-453d-bb9e-64e46daf1480	13:13:23	2021-03-09
☐	117	cf9731b5-2895-4df5-9c3a-d8c5d4daed72	ok		13:13:18	2021-03-09
☐	116	d08f39a6-05c6-4f5b-89cb-			13:12:43	2021-03-09

Fig 4.2.2.1 Post Dashboard

The Follow dashboard contains the information regarding who is following whom and since what time. It is used to show posts in the news feed, as user can only see the post of the authorized users that they are following.

Sathyabama | Dean-dashboard

WELCOME, **BARBIE** [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

[Home](#) / [Connections](#) / [Follows](#)

Select follow to change

ADD FOLLOW +

Action: Go0 of 13 selected

☐	TO	BY	DATE
☐	a6f86778-89e9-492d-9a62-bcb03c6f77de	a6f86778-89e9-492d-9a62-bcb03c6f77de	2021-03-09
☐	6d28e371-d176-426d-880c-bd770c9dfa7d	a6f86778-89e9-492d-9a62-bcb03c6f77de	2021-03-09
☐	d2b039d2-ecce-4c23-bed6-fb1c1bb09dc8	a6f86778-89e9-492d-9a62-bcb03c6f77de	2021-03-09
☐	43763467-e61e-4c16-a65b-7f772af8d7f2	a6f86778-89e9-492d-9a62-bcb03c6f77de	2021-03-09
☐	db7348f6-093d-4e8a-952a-aa2955b6f3b2	a6f86778-89e9-492d-9a62-bcb03c6f77de	2021-03-09
☐	d2b039d2-ecce-4c23-bed6-fb1c1bb09dc8	6d28e371-d176-426d-880c-bd770c9dfa7d	2021-03-09
☐	6d28e371-d176-426d-880c-bd770c9dfa7d	6d28e371-d176-426d-880c-bd770c9dfa7d	2021-03-09
☐	db7348f6-093d-4e8a-952a-aa2955b6f3b2	6d28e371-d176-426d-880c-bd770c9dfa7d	2021-03-09
☐	43763467-e61e-4c16-a65b-7f772af8d7f2	6d28e371-d176-426d-880c-bd770c9dfa7d	2021-03-03
☐	d2b039d2-ecce-4c23-bed6-fb1c1bb09dc8	4d94aab2-20a7-4f80-b4dd-5df22dddc878	2021-03-03
☐	991492e6-ea20-4b67-bcf7-80abfe9c2171	991492e6-ea20-4b67-bcf7-80abfe9c2171	2021-03-03
☐	d310a9b5-c1df-469b-99f0-6304e5d91e4c	926808f1-7b17-4b2a-81e5-408e622fca5e	2021-03-03
☐	db7348f6-093d-4e8a-952a-aa2955b6f3b2	926808f1-7b17-4b2a-81e5-408e622fca5e	2021-03-03

13 follows

Fig 4.2.3.1 Follow Dashboard

CHAPTER 5

CONCLUSION AND SUMMARY

5.1 CONCLUSION

The application, ASSIST, with the latest technology stack and is built with an intent to provide a smart communication application that will bring enhancement to the current circular notification procedure. It will serve as a peer to peer platform, hence no chance of data breach or exploitation.

5.2 SUMMARY

In this world of growing technologies everything is digitalized. In this era, it's difficult to identify which information is authentic and which is fake. In this project, we have developed a mobile application that allows only authorized users to send information to its followers. Every transaction of information made through this application is secured using OAuth2 and API key authentications.

Data privacy and security is the biggest threat to social media applications which are taken care of using OAuth2 and API key authentications. The second biggest threat comes up as polluted and manipulative content which compromises the authenticity of the platform. These threats are taken care of with precise threads of competitive and contemporary technologies.

REFERENCES

1. Digvijaysinh M. Rathod, Dr. M S Dahiya, Dr. Satyen M. Parikh, "Towards Composition of RESTful Web Services" 6th ICCCNT – 2015 July 13 - 15, 2015, Denton, U.S.A
2. Park, J., Kim, D., & Yeom, K. (2020). An Approach for Reconstructing Applications to Develop Container-Based Microservices. *Mobile Information Systems*, 2020.
3. Khoonsari, P. E., Moreno, P., Bergmann, S., Burman, J., Capuccini, M., Carone, M., ... & Hankemeier, T. (2019). Interoperable and scalable data analysis with microservices: applications in metabolomics. *Bioinformatics*, 35(19), 3752.
4. Fernando, S., Birch, D., Molina-Solana, M., McIlwraith, D., & Guo, Y. (2019, July). Compositional Microservices for Immersive Social Visual Analytics. In *2019 23rd International Conference Information Visualisation (IV)* (pp. 216-223). IEEE.
5. Di Francesco, P., Lago, P., & Malavolta, I. (2019). Architecting with microservices: A systematic mapping study. *Journal of Systems and Software*, 150, 77-97.
6. Elyas Ben HadjYahia, Laurent Reveillere, Yerom-David Bromberg, Raphael Chevalier, and Alain Cadot, "Medley: An Event- Driven Lightweight Platform For Service Composition"
7. Martin Garriga , CristianMateos , AndresFlores , AlejandraCechich , Alejandro Zunino "RESTful service composition at a glance: A survey" *Journal of Network and Computer Applications* (2016)
8. Bennara, Youssef Amghar, Michael Mrissa "Managing Web Resource Compositions" 2015 IEEE 24th International Conference on Enabling Technologies: Infrastructures for Collaborative Enterprises 978-1-4673-7692-1/15
9. Yu, D., Jin, Y., Zhang, Y., & Zheng, X. (2019). A survey on security issues in services communication of Micro-services are enabled for applications. *Concurrency and Computation: Practice and Experience*, 31(22), e4436.
10. Baresi, L., & Garriga, M. (2020). Microservices: The Evolution and Extinction of Web Services?. In *Microservices* (pp. 3-28). Springer, Cham.

APPENDICES

Home.dart

```
import 'package:dean_app/database/database_helper.dart';
import 'package:dean_app/model/post_model.dart';
import 'package:dean_app/resources/get_posts.dart';
import 'package:dean_app/views/add_post.dart';
import 'package:dean_app/views/login.dart';
import 'package:dean_app/views/post_layout.dart';
import 'package:dean_app/views/search.dart';
import 'package:flutter/material.dart';
import 'package:shared_preferences/shared_preferences.dart';
```

```
class Home extends StatefulWidget {
  @override
  _HomeState createState() => _HomeState();
}
```

```
class _HomeState extends State<Home> {
  List<PostModel> postList = new List<PostModel>();
  GetPosts getPosts = GetPosts();
  DatabaseHelper dbHelper = DatabaseHelper();
```

```
@override
Widget build(BuildContext context) {
  return WillPopScope(
    onWillPop: () => Future.value(false),
    child: Scaffold(
```

```
appBar: AppBar(
```

```

title: Text("Home"),

actions: <Widget>[

  IconButton(

    icon: Icon(

      Icons.search,

    ),

    onPressed: () {

      Navigator.of(context)

        .push(MaterialPageRoute(builder: (context) => Search()));

    },

  ),

  IconButton(

    icon: Icon(

      Icons.store,

    ),

    onPressed: () {

      setSignOut();

      signOutFromGoogle();

      Navigator.of(context)

        .push(MaterialPageRoute(builder: (context) => Login()));

    },

  ),

  IconButton(

    icon: Icon(

      Icons.add,

    ),

    onPressed: () {

      // TODO implement Add post screen

```

Navigator.of(context)

```

        .push(MaterialPageRoute(builder: (context) => AddPost()));
    },
  ),
],
  automaticallyImplyLeading: false,
),
body: RefreshIndicator(
  onRefresh: refreshLayout,
  child: ListView.builder(
    itemCount: postList == null ? 0 : postList.length,
    itemBuilder: (BuildContext context, int position) {
      return PostLayout(
        postList: postList[position],
      );
    }
  ),
),
bottomNavigationBar: BottomNavigationBar(
  items: const <BottomNavigationBarItem>[
    BottomNavigationBarItem(
      icon: Icon(Icons.home),
      title: Text('Home'),
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.favorite), title: Text("Notifications")),
    BottomNavigationBarItem(
      icon: Icon(Icons.account_circle),
      title: Text('Profile'),
    ),
  ],

```

J.

```
    ),  
    ));  
}
```

```
@override  
initState() {  
  super.initState();  
  
  getPosts.getPosts(context).then((value) => this.setState(() {  
    dbHelper.getPosts().then((value) => this.setState(() {  
      postList = value;  
    }));  
  }));  
}
```

```
Future refreshLayout() async {  
  getPosts.getPosts(context).then((value) => this.setState(() {  
    dbHelper.getPosts().then((value) => this.setState(() {  
      postList = value;  
  
      print("Refreshed");  
      print(postList.length);  
    }));  
  }));  
  print("Refresh");  
}
```

```
void signOutFromGoogle() async {  
  await googleSignIn.signOut();  
}
```

```
DatabaseHelper databaseHelper = DatabaseHelper();
```

```

databaseHelper.deletePost();

databaseHelper.getPosts().then((value) => print(value.length));

print("User Sign Out");
}

setSignOut() async {
  SharedPreferences sharedPreferences = await SharedPreferences.getInstance();
  sharedPreferences.clear();
}
}

```

AddPost.dart

```

import 'dart:convert';

import 'package:crypto/crypto.dart';

import 'package:dean_app/model/new_post_response_model.dart';
import 'package:dean_app/model/post_model.dart';
import 'package:http/http.dart' as http;
import 'package:shared_preferences/shared_preferences.dart';

class AddPostClient{
  final String desc;
  final String attachment;

  AddPostClient(this.desc, this.attachment);

  Future addPostReq() async {
    SharedPreferences sharedPreferences = await SharedPreferences.getInstance();

    var data = json.encode({

```

"uid": sharedPreferences.getString("uid"),

"description": desc,

```

        "attachment": attachment
    });

    Map<String, String> head = {
        'Content-type': 'application/json',
        'Authorization': getAuthorizationToken(),
    };

    final response = await http.post(
        "https://sathyabama-dean.herokuapp.com/feeds/posts"
        , body: data,
        headers: head);

    final responseJson = json.decode(response.body);

    CreatePostResponseModel createPostResponseModel = CreatePostResponseModel();
    createPostResponseModel.fromJson(responseJson);
    print(createPostResponseModel.message);
}
}

```

```

String getAuthorizationToken() {
    String salt = '_3bvihz^gyydh86k0+b34xq&+006^m#l)n@!9=s&t@^*xdyn';

    DateTime dateTime = new DateTime.now();

    dateTime = dateTime.toUtc();

    String dateStr = dateTime.toString().substring(0, 10);

    String toEncode = salt + dateStr + salt;

    var bytes = utf8.encode(toEncode);

    var digest = sha256.convert(bytes);

    print(digest.toString().toLowerCase());

    return digest.toString().toLowerCase();
}

```

ApiClient.dart

```
import 'dart:convert';

import 'package:crypto/crypto.dart';

import 'package:dean_app/model/create_response.dart';
import 'package:dean_app/model/login_response.dart';
import 'package:dean_app/model/user_response_model.dart';
import 'package:dean_app/views/home.dart';
import 'package:dean_app/views/login.dart';
import 'package:dean_app/views/userReg.dart';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'package:shared_preferences/shared_preferences.dart';

class ApiClient {

  LoginResponseModel loginResponseModel = LoginResponseModel();

  CreateResponseModel createResponseModel = CreateResponseModel();

  Future loginRequest(context) async {

    Widget nav;

    SharedPreferences sharedPreferences = await SharedPreferences.getInstance();

    var data = json.encode({

      "name": sharedPreferences.getString("fullName"),

      "image": sharedPreferences.getString("image"),

      "email": sharedPreferences.getString("email")

    });

    Map<String, String> head = {
```

'Content-type': 'application/json',

```

'Authorization': getAuthorizationToken(),

};

final response = await http.post(

  "https://sathyabama-dean.herokuapp.com/authenticate/signin",

  body: data,

  headers: head);

final responseJson = json.decode(response.body);

loginResponseModel.fromJson(responseJson);

if (loginResponseModel.action == 0) {

  nav = Home();

  sharedPreferences.setBool("isNew", false);

} else {

  sharedPreferences.setBool("isNew", true);

  nav = UserRegistration();

}

loginResponseModel.message = responseJson['message'];

if (loginResponseModel.action != -1) {

  print(loginResponseModel.action);

}

if (loginResponseModel.message != "") {

  print(loginResponseModel.message);

}

if (loginResponseModel.uid != "") {

  print(loginResponseModel.uid);

  sharedPreferences.setString("uid", loginResponseModel.uid);

}

```

Navigator.of(context).push(

```

MaterialPageRoute(
  builder: (context) {
    return nav;
  },
),
);
}

```

```

void createUser(String name, String uid, String reg, String mobile,
  String dept, String dob, context) async {
  var data = json.encode({
    "name": name,
    "dept": dept,
    "regno": reg,
    "dob": dob,
    "mobile": mobile,
    "uid": uid
  });
  UserModel.fromJson(json.decode(data));
  Map<String, String> head = {
    'Content-type': 'application/json',
    'Authorization': getAuthorizationToken(),
  };
  final response = await http.post(
    "https://sathyabama-dean.herokuapp.com/authenticate/create",
    body: data,
    headers: head);

```

```
createResponseModel.fromJson(json.decode(response.body
)); if (createResponseModel.action == 0) {
```

```

    setSharedPref(googleSignIn.currentUser.email,
        googleSignIn.currentUser.displayName, "");
    Navigator.of(context).push(
        MaterialPageRoute(
            builder: (context) {
                return Home();
            },
        ),
    );
}
}
}

```

```

String getAuthorizationToken() {
    String salt = '_3bvihz^gvydh86k0+b34xq&+006^m#l)n@!9=s&t@^*xdyn';
    DateTime dateTime = new DateTime.now();
    dateTime = dateTime.toUtc();
    String dateStr = dateTime.toString().substring(0, 10);
    String toEncode = salt + dateStr + salt;
    var bytes = utf8.encode(toEncode);
    var digest = sha256.convert(bytes);
    print(digest.toString().toLowerCase());
    return digest.toString().toLowerCase();
}

```

```

setSharedPref(String email, String fullName, String image) async {
    SharedPreferences sharedPreferences = await SharedPreferences.getInstance();
    sharedPreferences.setBool("isLoggedIn", true);
}

```

```
sharedPreferences.setString("email", email);
```

```
sharedPreferences.setString("fullName", fullName);
```

```
sharedPreferences.setString("image", image);  
}
```

GetPosts.dart

```
import 'dart:convert';  
  
import 'package:crypto/crypto.dart';  
  
import 'package:dean_app/database/database_helper.dart';  
  
import 'package:dean_app/model/post_model.dart';  
  
import  
  
'package:dean_app/model/post_response_model.dart';  
  
import 'package:http/http.dart' as http;  
  
import 'package:shared_preferences/shared_preferences.dart';  
  
  
class GetPosts {  
  
  DatabaseHelper dbHelper = DatabaseHelper();  
  
  
  Future getPosts(context) async {  
  
    SharedPreferences sharedPreferences = await SharedPreferences.getInstance();  
  
    var data = json.encode({"uid": sharedPreferences.getString("uid")});  
  
    Map<String, String> head = {  
  
      'Content-type': 'application/json',  
  
      'Authorization': getAuthorizationToken(),  
  
    };  
  
  
    var response = await http.put(
```

"https://sathyabama-dean.herokuapp.com/feeds/posts"

, body: data,

```

        headers: head);

print("Response :");

print(response.body);

final responseJson = json.decode(response.body);

PostResponseModel postResponseModel = new PostResponseModel();

List<PostModel> postmodelList = postResponseModel.fromJson(responseJson);

for (PostModel postModel in postmodelList) {

    dbHelper.savePost(postModel);

}

}

}

```

```

String getAuthorizationToken() {

String salt = '_3bvihz^gvydh86k0+b34xq&+006^m#l)n@!9=s&t@^*xdyn';

DateTime dateTime = new DateTime.now();

dateTime = dateTime.toUtc();

String dateStr = dateTime.toString().substring(0, 10);

String toEncode = salt + dateStr + salt;

var bytes = utf8.encode(toEncode);

var digest = sha256.convert(bytes);

print(digest.toString().toLowerCase());

return digest.toString().toLowerCase();

}

```

PUBLICATION DETAILS

TITLE : REST API DEVELOPMENT USING DJANGO

NAME OF THE CONFERENCE: N/A

STATUS OF THE PAPER: NOT PAID

24

ORIGINALITY REPORT

8%	2%	5%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Usha Sakthivel, Neha Singhal, Pethuru Raj. "RESTful web services composition & performance evaluation with different databases", 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECOT), 2017 Publication	5%
2	blog.appliedinformaticsinc.com Internet Source	2%
3	Submitted to Rutgers University, New Brunswick Student Paper	1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

