|  |
| --- |
| **TRAVEL BOOK IN FLUTTER** |

**AIM:**

The **“NoteHub”** application is a collaborative platform designed for users to share and access notes effortlessly. Users can log in, view, and upload notes, explore bookmarked posts, and interact with their profile showcases. The app ensures seamless access to educational content and personal customization features, providing an engaging and efficient note-sharing experience.

**ALGORITHM:**

**Step 1**: Authenticate the user by verifying their login credentials. Grant access to the application upon successful authentication.

**Step 2**: Display the user's profile, including personal documents and bookmarked posts in separate tabs.

**Step 3**: Fetch posts associated with the user's profile or saved posts using backend controllers and render them in the UI.

**Step 4**: Enable editing, saving, or additional interactions for notes depending on ownership and user preferences.

**Step 5**: Maintain smooth navigation between tabs, ensuring a seamless user experience.

**CODE:**

**main.dart**

import 'package:flutter/material.dart';

import 'package:get/get.dart';

import 'package:hive\_flutter/hive\_flutter.dart';

import 'package:lottie/lottie.dart';

import 'package:notehub/controller/bottom\_navigation\_controller.dart';

import 'package:notehub/core/helper/hive\_boxes.dart';

import 'package:notehub/model/user\_model.dart';

import 'package:notehub/view/splash\_screen/splash.dart';

import 'package:toastification/toastification.dart';

void main() async {

WidgetsFlutterBinding.ensureInitialized();

await Hive.initFlutter();

Hive.registerAdapter(UserModelAdapter());

await Hive.openBox<UserModel>("user");

Get.put(BottomNavigationController());

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return ToastificationWrapper(

child: GetMaterialApp(

title: 'Note Hub',

theme: ThemeData(

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

useMaterial3: true,

),

home: const Splash(),

),

);

}

}

**home.dart**

import 'package:flutter/material.dart';

import 'package:get/get.dart';

import 'package:notehub/controller/document\_controller.dart';

import 'package:notehub/controller/home\_controller.dart';

import 'package:notehub/core/helper/hive\_boxes.dart';

import 'package:notehub/view/home\_screen/widget/home\_document\_section.dart';

import 'package:notehub/view/home\_screen/widget/home\_header.dart';

class Home extends StatefulWidget {

const Home({super.key});

@override

State<Home> createState() => \_HomeState();

}

class \_HomeState extends State<Home> {

@override

void initState() {

super.initState();

Get.put(DocumentController());

Get.put(HomeController());

loadData();

}

loadData() async {

var homeController = Get.find<HomeController>();

if (!homeController.isFetched.value) {

homeController.fetchUpdates();

}

Get.find<DocumentController>().fetchDocsForUsername(

username: HiveBoxes.username,

);

}

@override

Widget build(BuildContext context) {

var homeController = Get.find<HomeController>();

if (!homeController.isFetched.value) {

homeController.fetchUpdates();

}

return const Scaffold(

resizeToAvoidBottomInset: false,

body: Padding(

padding: EdgeInsets.symmetric(horizontal: 12),

child: Column(

children: [

SafeArea(child: SizedBox.shrink()),

HomeHeader(),

Expanded(child: HomeDocumentSection()),

],

),

),

);

}

}

**profile.dart**

import 'package:flutter/material.dart';

import 'package:get/get.dart';

import 'package:notehub/controller/profile\_controller.dart';

import 'package:notehub/controller/showcase\_controller.dart';

import 'package:notehub/core/config/color.dart';

import 'package:notehub/core/helper/custom\_icon.dart';

import 'package:notehub/core/helper/hive\_boxes.dart';

mport 'package:notehub/view/profile\_screen/widget/profile\_header.dart';

import 'package:notehub/view/profile\_screen/widget/profile\_showcase.dart';

import 'package:notehub/view/settings\_screen/settings\_drawer.dart';

import 'package:notehub/view/widgets/normal\_button.dart';

import 'package:notehub/view/widgets/refresher\_widget.dart';

import 'package:shimmer/shimmer.dart';

class Profile extends StatefulWidget {

final String username;

const Profile({super.key, required this.username});

@override

State<Profile> createState() => \_ProfileState();

}

class \_ProfileState extends State<Profile> {

@override

void initState() {

super.initState();

Get.put(SettingsDrawerController());

loadUserData();

}

Future<void> loadUserData() async {

Get.put(SettingsDrawerController());

Get.find<ProfileController>().fetchUserData(username: HiveBoxes.username);

Get.find<ShowcaseController>(tag: HiveBoxes.username)

.fetchProfilePosts(username: HiveBoxes.username);

Get.find<ShowcaseController>(tag: HiveBoxes.username)

.fetchSavedPosts(username: HiveBoxes.username);

}

@override

Widget build(BuildContext context) {

return RefresherWidget(

onRefresh: loadUserData,

child: Scaffold(

key: Get.find<SettingsDrawerController>().scaffoldKey,

resizeToAvoidBottomInset: false,

drawer: const SettingsDrawer(),

body: Column(

children: [

Row(

mainAxisAlignment: MainAxisAlignment.end,

children: [

NormalButton(

onPressed: Get.find<SettingsDrawerController>().openDrawer,

child: const CustomIcon(path: "assets/icons/settings.svg"),

),

const SizedBox(width: 24),

],

),

GetX<ProfileController>(builder: (controller) {

if (controller.isLoading.value) {

return Shimmer.fromColors(

baseColor: GrayscaleWhiteColors.almostWhite,

highlightColor: GrayscaleWhiteColors.white,

child: ProfileHeader(

profileData:

Get.find<ProfileController>().profileData.value,

),

);

}

return ProfileHeader(

profileData: Get.find<ProfileController>().profileData.value,

);

}),

ProfileShowcase(username: widget.username),

],

),

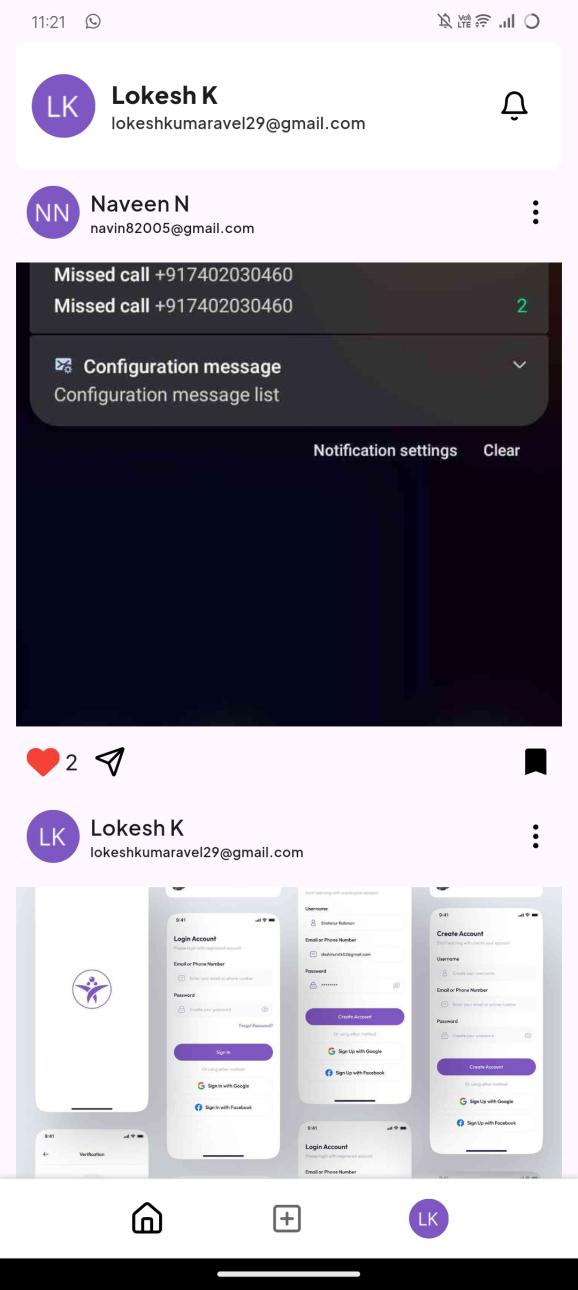
),

);

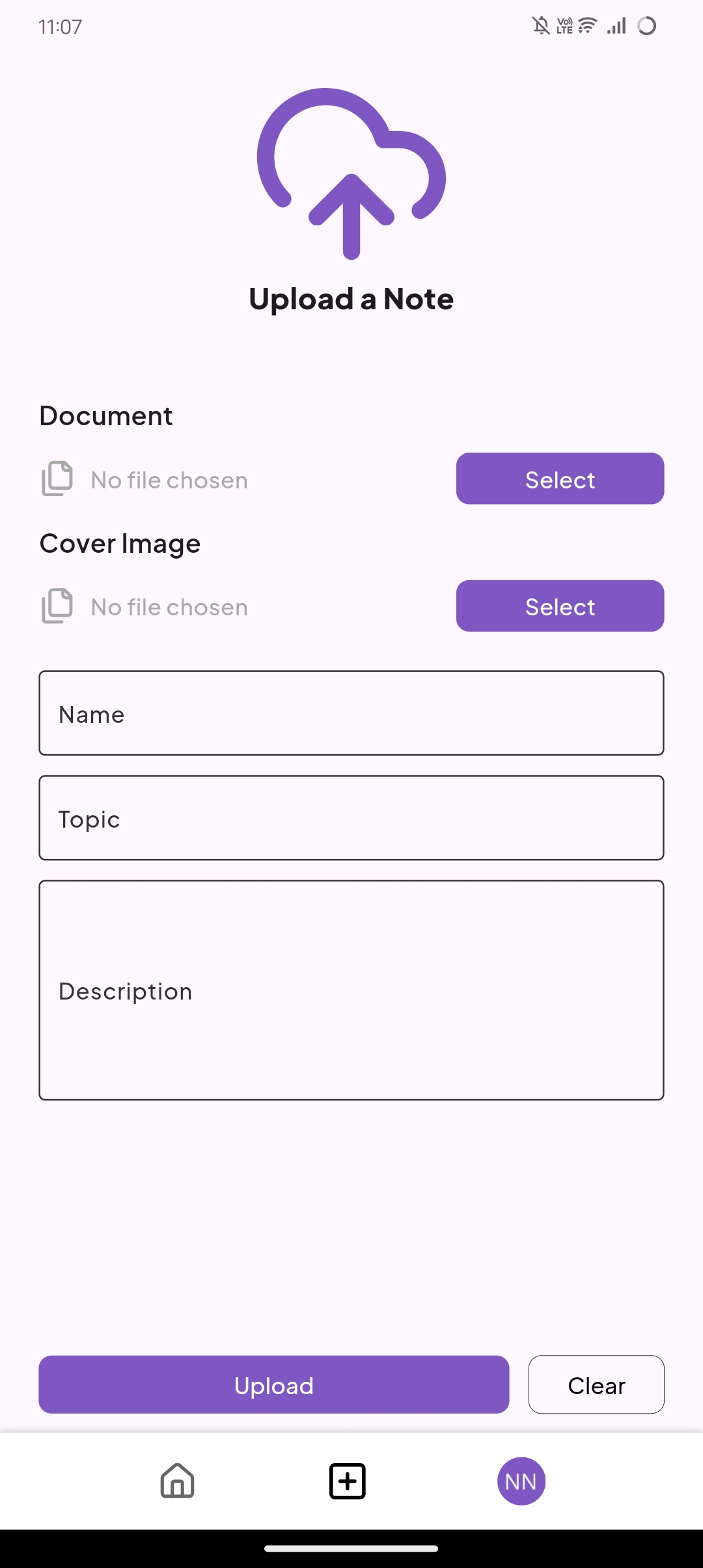
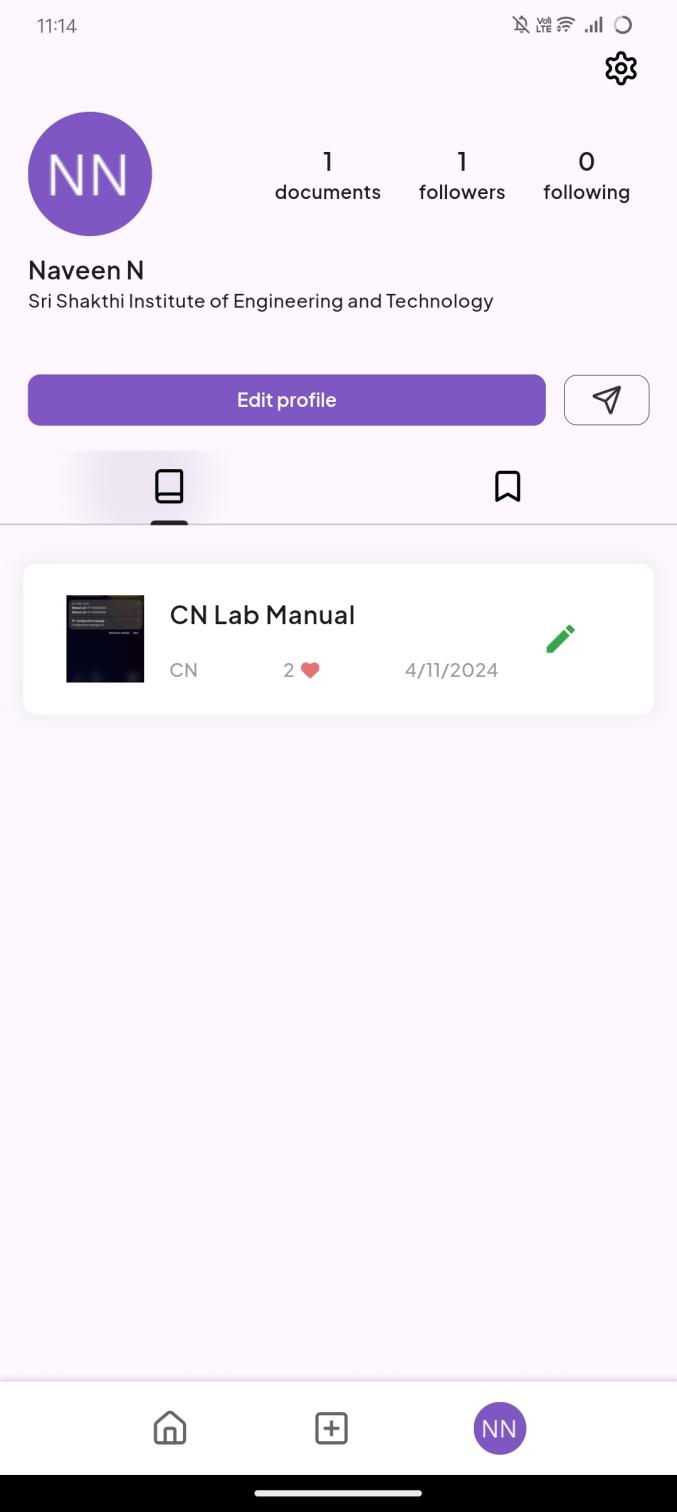
}

}

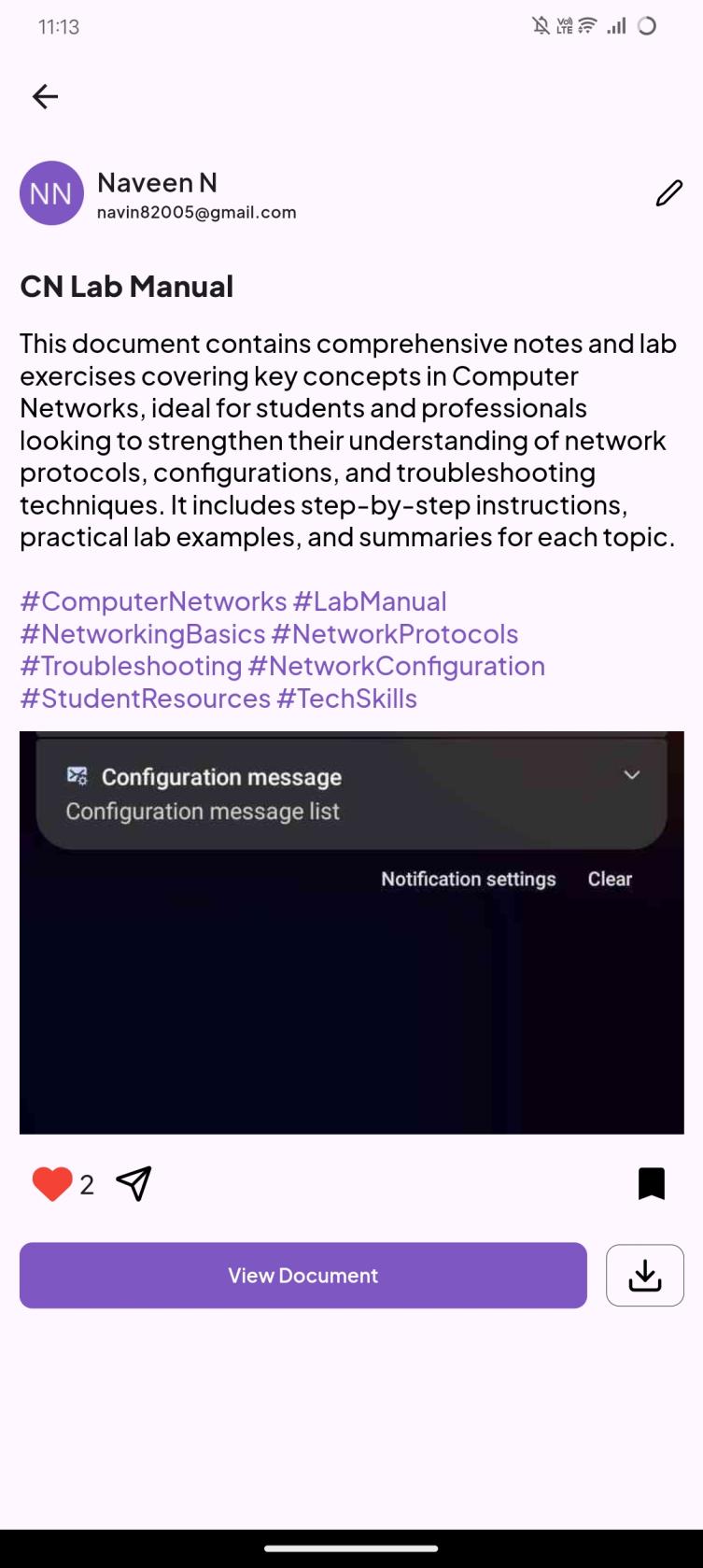
**OUTPUT:**

****

**Fig: Home Screen**

****

**Fig: Profile Screen Fig: Document Upload**

****

**Fig: Document Discription**

**RESULT:**

The **“NoteHub”** application successfully facilitates user authentication and dynamic content rendering. It enables users to explore and manage notes, view bookmarked posts, and interact with profile showcases. The app provides a seamless note-sharing experience, with features like dynamic tabs, custom icons, and smooth scrolling enhancing user engagement and usability.