

# PROJECT REPORT ON BUS TRACKING APP



## SUBMITTED BY:

H. LALHRUAITLUANGA	-	Roll no. 2223BCA004
LALENGZUALA ZADENG	-	Roll no. 2223BCA010
LALRINSANGA SAILO	-	Roll no. 2223BCA012

Under the guidance of:  
**Mr. H. Lalruatkima**  
Assistant Professor  
Bachelor of Computer Application  
(HATIM)



**MIZORAM UNIVERSITY, TANHRIL: AIZAWL**  
**2024**

**HIGHER AND TECHNICAL INSTITUTE, MIZORAM**  
**DEPARTMENT OF COMPUTER SCIENCES**



**CERTIFICATE**

*This is to certify that **H. Lalhrvaitluanga, Lalengzuala Zadeng and Lalrinsanga Sailo** has fully completed the project entitled, “**BUS TRACKING APP**” in order to meet the requirement of the Mizoram University for the V Semester Bachelor of Computer Application in the year 2024. It is to certify that all the corrections/suggestions indicated for internal assessment has been incorporated in the project. The project report has been approved as it satisfies the academic requirements in respects of the project work prescribed for the BCA Course.*

(H. LALRUATKIMA)  
Project Guide  
Dept. of Computer Science

(VUANSANGA VANCHHAWNG)  
Principal

(K. LALMUANPUIA)  
Head of Department  
Dept. of Computer Science

Name of Examiner

Signature with date

---

---

## ACKNOWLEDGEMENT

First, we would like to give thanks to God for guiding us throughout the process of our project, giving us with good health so that we are able to get this far.

We thank to our Principal Mr. Vuansanga Vanchhawng. We thank him for the support, encouragement and permission, which we enjoy freely in order to accomplish our project, and in this time of Pandemic situation.

Our heartfelt gratitude goes to Mr. H. Lalruatkima, Project Guide and Mr. K. Lalmuanpuia, Head of Department in Department of Computer Science, Higher and Technical Institute, Mizoram for their continual advice, backup, confidence, patience and their encouraging words which gives us courage and confidence without which we would be nowhere.

We thank all the people in the Department of Computer Science for being always available and helpful over the semester.

Last but not the least; we thank our parents for their moral support and encouragement.

# CONTENTS

Fig. No.	Figure Name	Page
<b>1</b>	<b>INTRODUCTION</b>	
	Objectives of the Project	2
	Overview of the Project	2
	Scope	2
<b>2</b>	<b>SYSTEM ANALYSIS</b>	
	Hardware Requirement	4
	Software Requirement	4
	Data Flow Diagram	5
<b>3</b>	<b>SYSTEM DESIGN – USER</b>	
	Welcome	7
	User Selection Page	8
	Login Page	9
	Sign Up Page	10
	Home Page	11
	Map Page	12
	Drivers' Details Page	13
	Bus Fare Lists Page	14
<b>4</b>	<b>SYSTEM DESIGN – ADMIN PANEL (DRIVER)</b>	
	Drivers' Login Page	15
	Drivers' Home Page	16
	Drivers' Map Page	17
<b>5</b>	<b>CODING – USER</b>	
	Welcome Page	20-22
	Login Page	22-27
	Sign Up Page	28-35
	Home Page	36-39
	Map Page	40-43
	Driver Details Page	44-46
	Bus Fae List Page	47-50
<b>6</b>	<b>CODING – ADMIN (DRIVER)</b>	
	Drivers' Login Page	52-55
	Drivers' Home Page	56-57
	Drivers' Map Page	58-63

<b>7</b>	<b>TESTING AND IMPLEMENTATION</b>	
	Testing	65
	Possible Problem	65
<b>8</b>	<b>DRAWBACKS AND LIMITATIONS</b>	66
<b>9</b>	<b>FUTURE ENHANCEMENT</b>	67
<b>10</b>	<b>CONCLUSION</b>	68

# 1. INTRODUCTION

## CONTENT

- 
- Overview of the Project
  - Objective of the Project

# INTRODUCTION

## Objectives of the Project

The main objectives of this project is to create an user-friendly and easily accessible application to track the location of the bus.

1. Provide Real-Time Bus Tracking: Develop a system that uses GPS to provide real-time tracking of buses, enabling students to see the current location and movement of their bus.
2. Improve Punctuality: Ensures that students have accurate information about bus arrival times, reducing waiting times and helping them plan their commute more effectively.
3. User-Friendly Interface: Develop a mobile application with an intuitive and user-friendly interface that can be easily used by all students.
4. Details of the bus drivers and bus fares: Providing details of the bus drivers such as mobile no and bus no along with bus fares for different bus stops.

## Overview of the Project

This project mainly focuses on creating a Bus Tracking App in which the students will be able to easily find the location of the bus. This system will provide students with up-to-date information on bus locations, details of the bus drivers and bus fares for different bus stops.

## Scope

The scope of this project is to provide an application for the students to track the location of the buses.

1. Bus Location's.
2. Details for bus fares.
3. Details of the bus drivers.

## 2. SYSTEM ANALYSIS

### CONTENT

- 
- **Hardware Requirement**
  - **Software Requirement**
  - **Entity-Relationship Diagram**
  - **Data Flow Diagram**



## **SYSTEM ANALYSIS**

### **Hardware Requirement**

This project is designed in such a way that it can be implemented almost on any Computer system. But, the requirements for a system to run this is quite high.

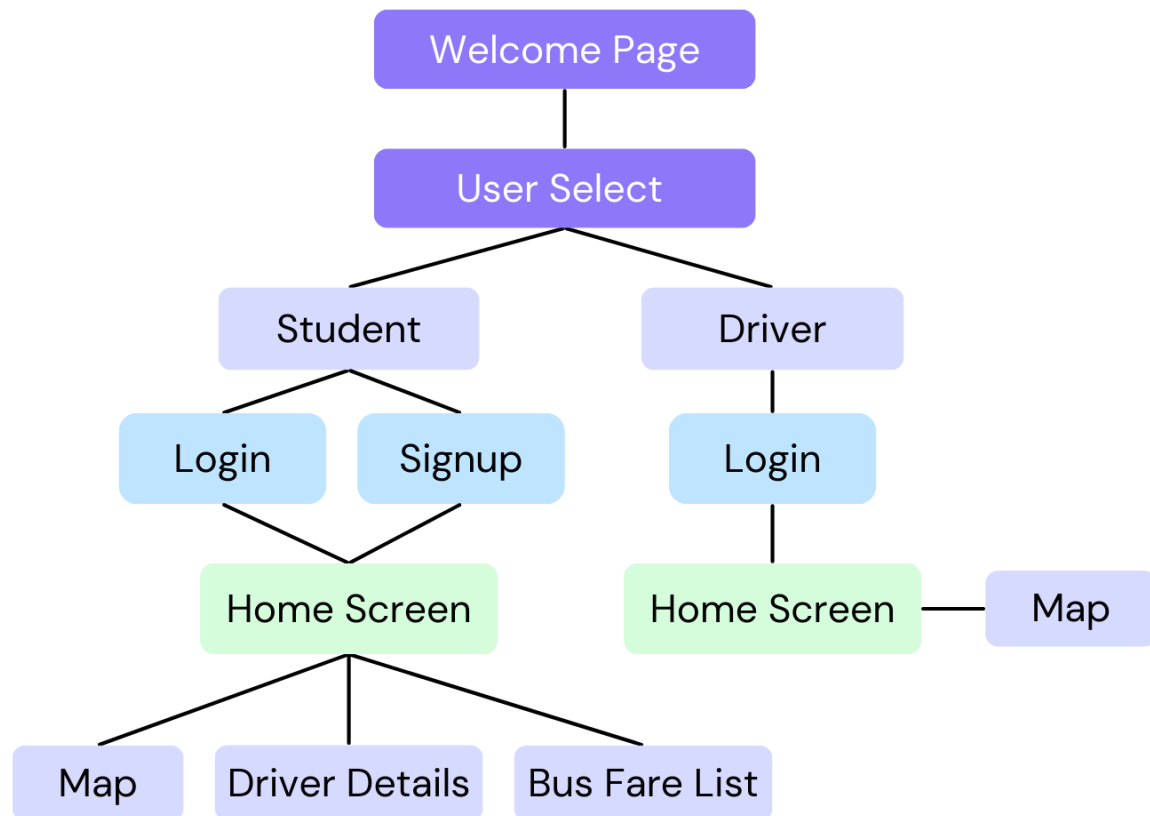
- Coloured monitor.
- Processor- Intel i5 or equivalent (64-bit capable)
- RAM 4gb minimum (8gb is recommended)
- Secondary memory (32GB and above)
- QWERTY or IBM compatible keyboard.
- Optical mouse.

### **Software Requirement**

The software used for designing the Users Interface (front- end) and fir the making of the server side (back-end) used are as follows:

- FIGMA
- FLUTTER
- ANDROID STUDIO

## Data Flow Diagram



### **3. SYSTEM DESIGN – USER (Student)**

#### **CONTENT**

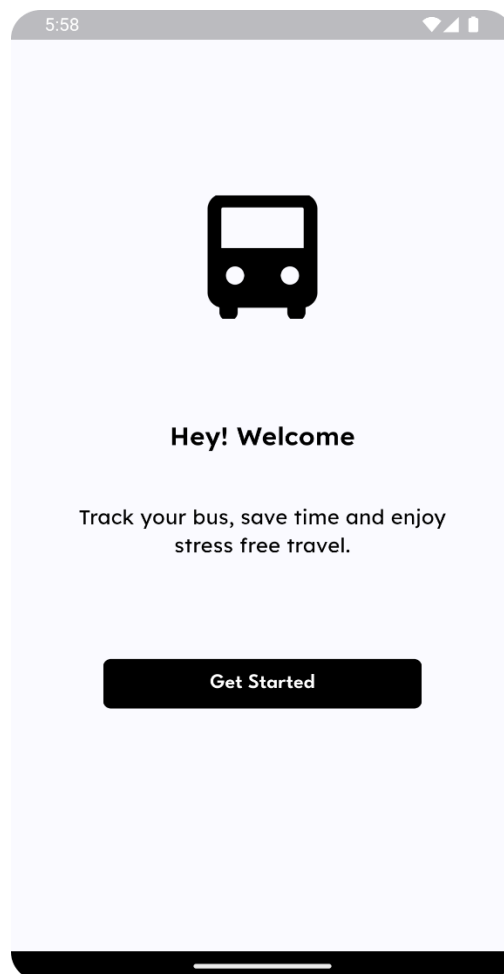
- 
- **Welcome**
  - **User Selection**
  - **Login Page**
  - **Sign Up Page**
  - **Home Page**
  - **Map Page**
  - **Drivers' details**
  - **Bus fare lists**

## SYSTEM DESIGN – USER

The Graphical User Interface of the project is created using Figma.

### Welcome

Our Welcome Page is designed to provide users with a seamless start to their journey. It features an intuitive layout, with an option “Get Started” that leads you to another page when clicked. The prominent logo and tagline immediately inform users about the app's purpose. The clean and minimalistic design, combined with vibrant visuals, ensures an engaging first interaction.



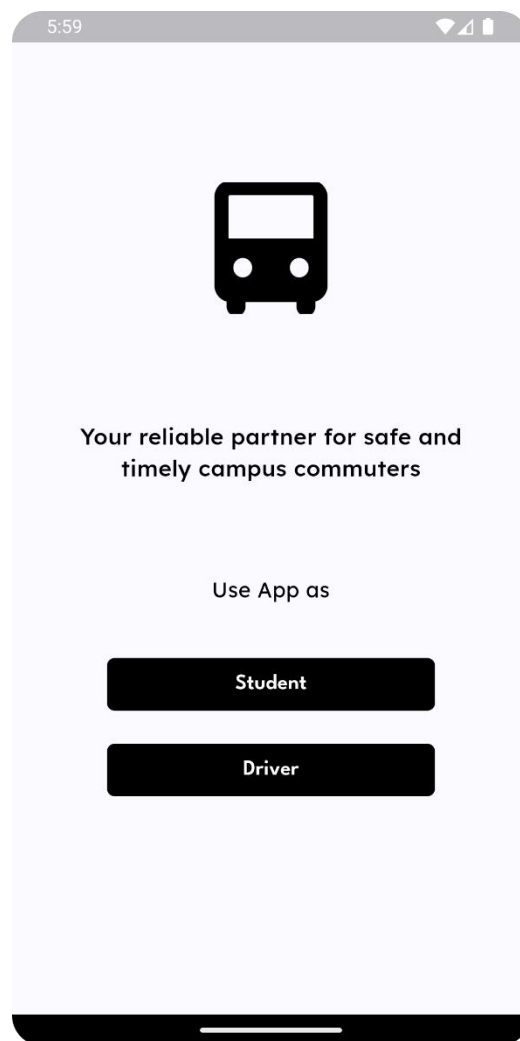
## User Selection

This page allows users to select their role (driver or student) at login which is essential for tailoring their app experience based on their specific needs.

Different roles require distinct functionalities, so role-based login helps streamline the interface for the user.

Role selection ensures that users only access the information relevant to them.

This separation of roles minimizes confusion, reduces clutter, and improves app performance by limiting the features each role accesses.



## Login Page

The students' login page is the gateway for students to access their bus tracking information but only functions for the students who have created their account. After successful authentication, students can view bus locations, drivers' details and the bus fares.

6:00

### STUDENT

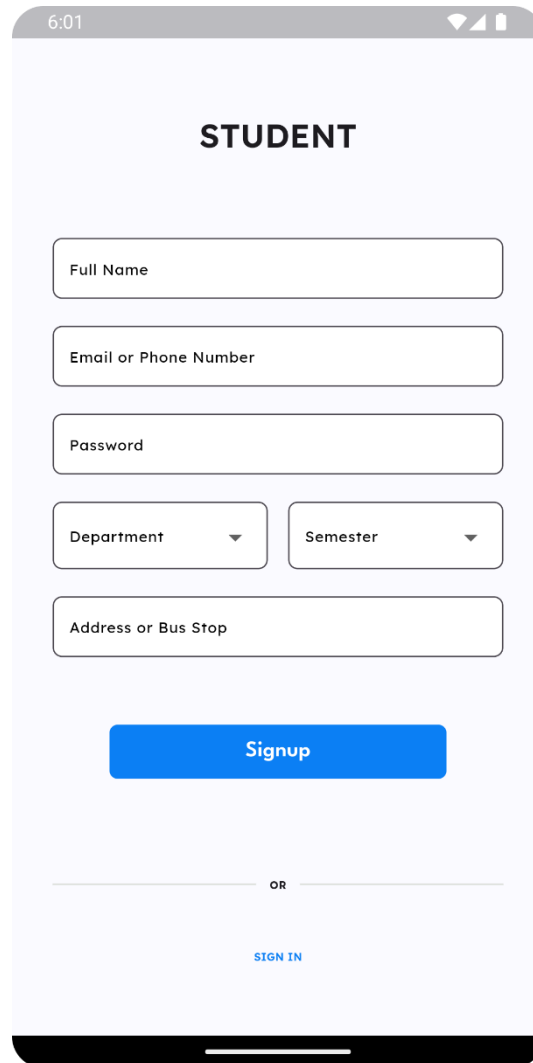
Login

OR

[CREATE NEW ACCOUNT](#)

## SignUp Page

A student signup page is designed to allow students to register and create their profiles, granting them access to features such as viewing bus drivers details, real-time tracking, and bus fares.



The image shows a mobile application interface for a student signup page. At the top, a status bar displays the time 6:01 and signal icons. The page has a light purple background. The word "STUDENT" is centered in bold black text. Below it are five input fields: "Full Name", "Email or Phone Number", "Password", "Department" (with a dropdown arrow), and "Semester" (with a dropdown arrow). Below these is a single-line input field for "Address or Bus Stop". A prominent blue "Signup" button is centered below the address field. At the bottom, a horizontal line separates the signup section from a section containing the text "OR" and a blue "SIGN IN" link.

6:01

### STUDENT

Full Name

Email or Phone Number

Password

Department ▼ Semester ▼

Address or Bus Stop

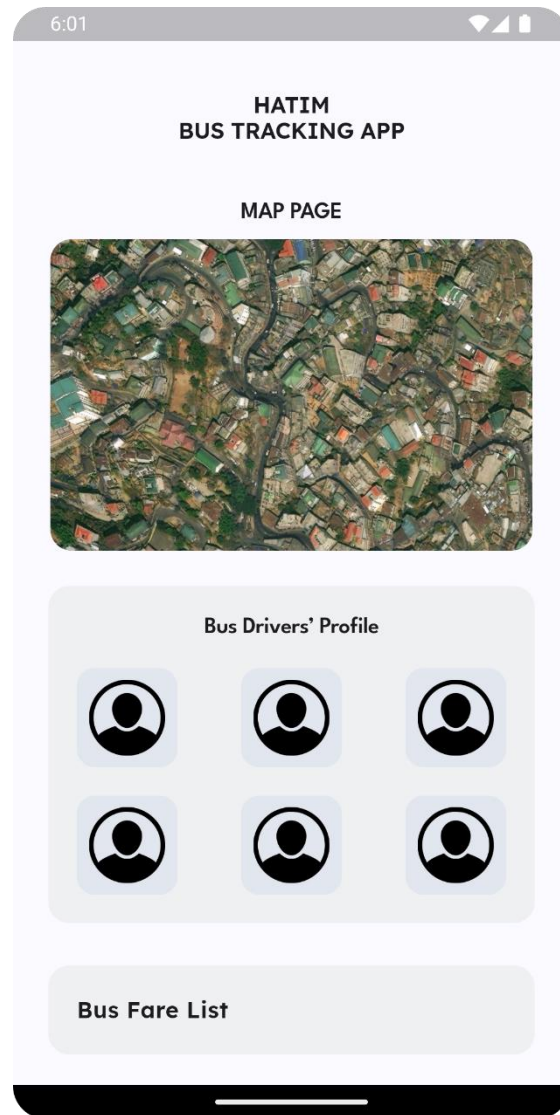
Signup

OR

[SIGN IN](#)

## Home Page

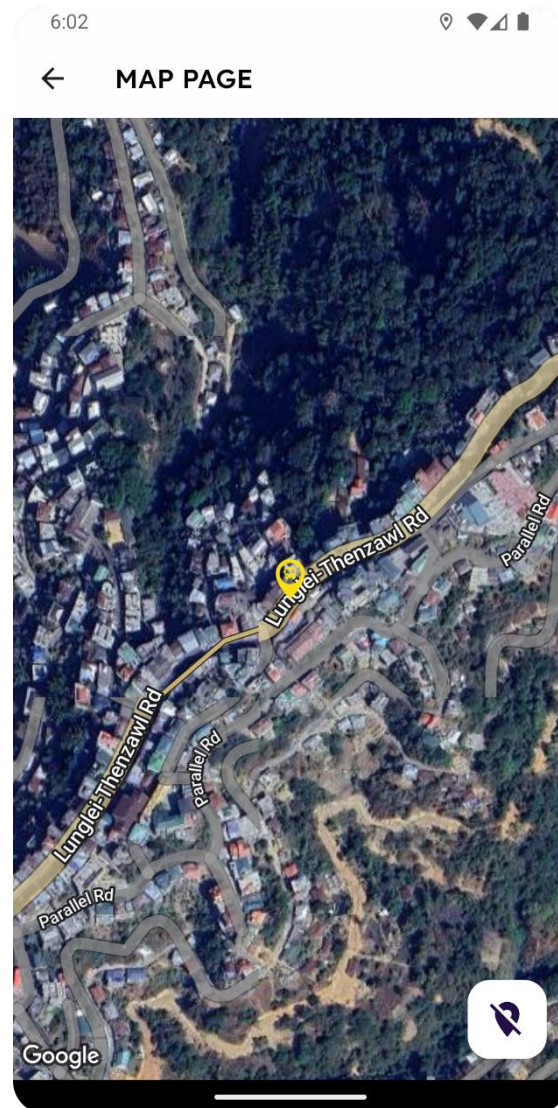
The Home Page provides an intuitive and user-friendly interface for users to access the core functionalities of the bus tracking app. It is designed for quick navigation and displays essential information upfront.





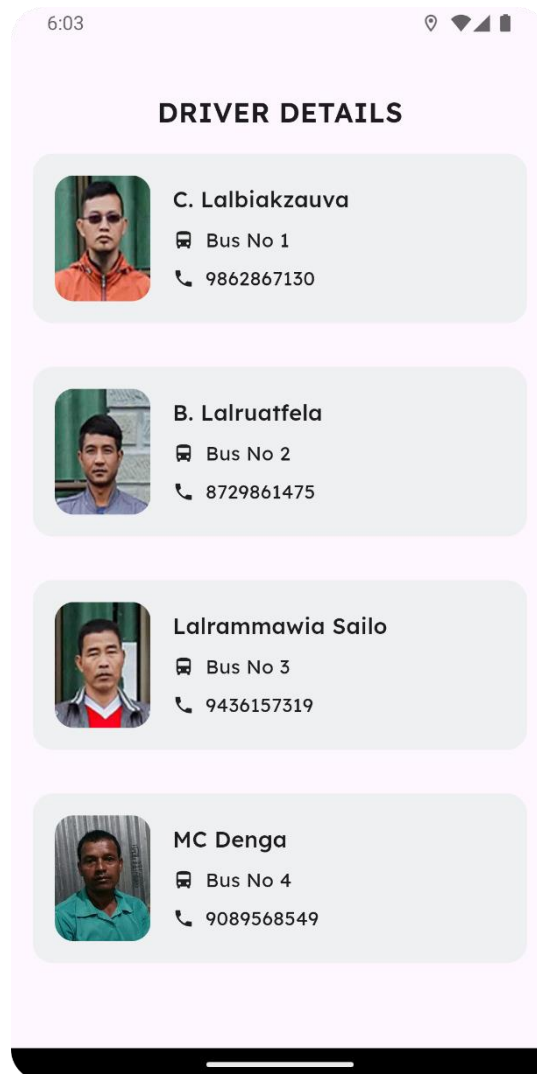
## Map Page

The map page displays the current location of buses in real-time, helping users plan their journeys efficiently.



## Drivers' Details

The Drivers' Details Page in a bus tracking app is designed to provide essential information of the drivers such as contact no, bus no and driver name.



## Bus Fare Lists

The Bus Fare List Page in a bus tracking app serves as a centralized place for users to view detailed fare information for different bus stops.



## 4. SYSTEM DESIGN – ADMIN PANEL (DRIVER)

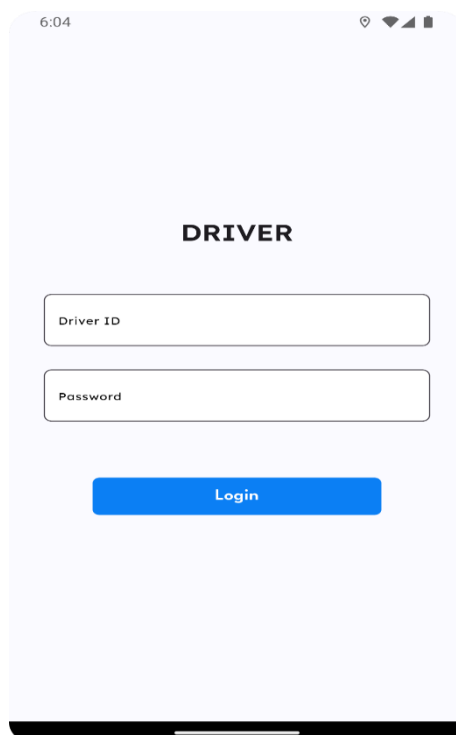
### CONTENT

- 
- Drivers' Login Page
  - Drivers' Home Page
  - Drivers' Map Page

#### Drivers' Login Page

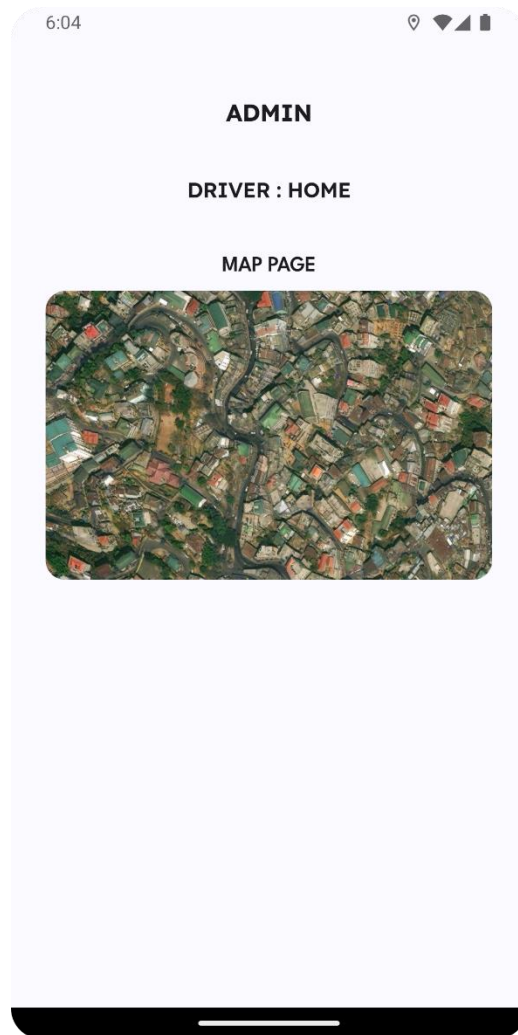
Verifies the identity of the driver to prevent unauthorized access.

Grants drivers access to their specific dashboard



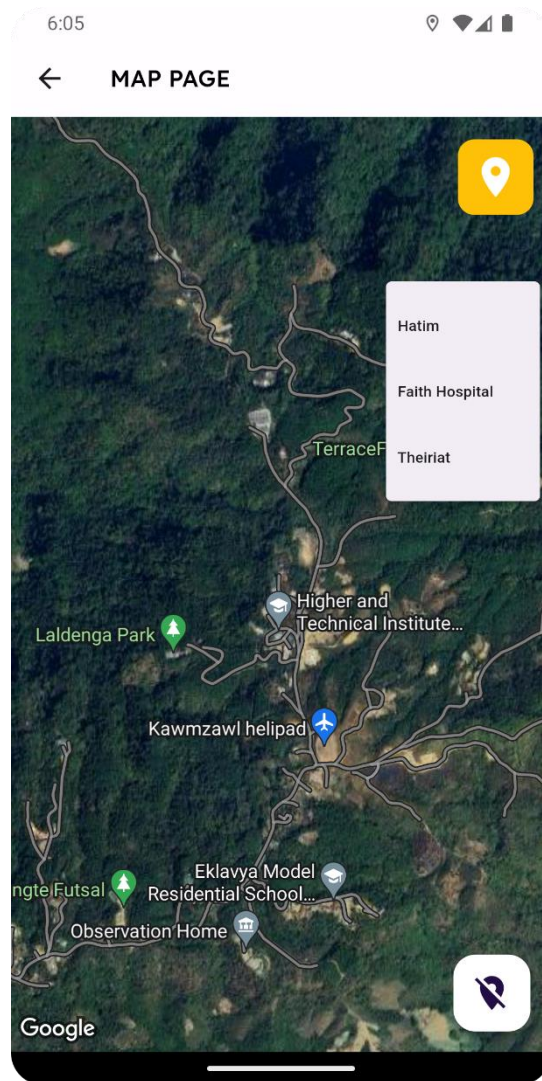
## Drivers' Home Page

The Driver's Home Page in a bus tracking app serves as the main dashboard for the bus drivers.



## Drivers' Map Page

The Driver's Map Page is designed to assist drivers in navigating routes efficiently and providing real-time location to the students.



## **5. CODING – USER (Student)**

### **CONTENT**

- 
- **Welcome**
  - **User Selection**
  - **Login Page**
  - **Sign Up Page**
  - **Home Page**
  - **Map Page**
  - **Drivers' details**
  - **Bus fare lists**

## CODING – USER

Main page with multiple screens for users and drivers. It sets up navigation using named routes, allowing smooth transitions between pages like login, sign-up, home, maps, and fare lists.

```
import 'package:bus_tracking_app/pages/welcome.dart';
import 'package:bus_tracking_app/pages/user.dart';
import 'package:bus_tracking_app/pages/login.dart';
import 'package:bus_tracking_app/pages/signup.dart';
import 'package:bus_tracking_app/pages/driverlogin.dart';
import 'package:bus_tracking_app/pages/home.dart';
import 'package:bus_tracking_app/pages/driverhome.dart';
import 'package:bus_tracking_app/pages/map.dart';
import 'package:bus_tracking_app/pages/drivermap.dart';
import 'package:bus_tracking_app/pages/driversdetails.dart';
import 'package:bus_tracking_app/pages/farelist.dart';
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),

      initialRoute: '/home', // Set the initial route
      routes: {
        '/welcome': (context) => Welcome(),
        '/user': (context) => User(),
        '/login': (context) => Login(),
        '/signup': (context) => SignUp(),
        '/driverlogin': (context) => DriverLogin(),
        '/home': (context) => HomePage(),
        '/driverhome': (context) => DriverHomePage(),
        '/map': (context) => MapPage(),
        '/drivermap': (context) => DriverMapPage(),
```



```

        '/driverdetails': (context) => DriverDetails(),
        '/fare': (context) => FareList(),
      },
    );
  }
}

```

## Welcome

```

import 'package:bus_tracking_app/pages/user.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';

```

```

void main() {
  runApp(Welcome());
}

```

```

class Welcome extends StatelessWidget {
  const Welcome({super.key});

```

```

  @override

```

```

  Widget build(BuildContext context) {

```

```

    return Scaffold(

```

```

      backgroundColor: const Color(0xFFFFFAFAFF),

```

```

      body: SingleChildScrollView(

```

```

        child: Column(

```

```

          children: [

```

```

            // Header - Image

```

```

            const SizedBox(height: 150),

```

```

            Center( // Horizontally center the image

```

```

              child: Container(

```

```

                width: 100,

```

```

                height: 100,

```

```

                decoration: BoxDecoration(

```

```

                  borderRadius: BorderRadius.circular(15),

```

```

                  image: DecorationImage(

```

```

                    image: AssetImage('assets/bus1.png'), // Update with your image path

```

```

                    fit: BoxFit.cover,

```

```

                ),

```

```

            ),

```

```

          ),

```

```

        ),

```

```

        const SizedBox(height: 80),

```

```

        // Welcome Text

```

```

        Padding(

```

```

padding: const EdgeInsets.symmetric(horizontal: 40),
child: Text(
  'Hey! Welcome',
  textAlign: TextAlign.center,
  style: GoogleFonts.lexend(
    fontSize: 20,
    color: Colors.black,
    fontWeight: FontWeight.w600,
  ),
),
),
const SizedBox(height: 40),

// Description Text
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 40),
  child: Text(
    'Track your bus, save time and enjoy stress free travel.',
    textAlign: TextAlign.center,
    style: GoogleFonts.lexend(
      fontSize: 16,
      color: Colors.black,
      fontWeight: FontWeight.w400,
    ),
  ),
),
const SizedBox(height: 80),

// Get Started Button
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 72),
  child: Container(
    width: double.infinity,
    height: 40,
    decoration: BoxDecoration(
      color: Colors.black,
      borderRadius: BorderRadius.circular(6),
    ),
  ),
  child: TextButton(
    onPressed: () {
      Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => User()),
      );
    },
    child: Text(
      'Get Started',

```

```

        style: GoogleFonts.leagueSpartan(
          fontSize: 16,
          color: Colors.white,
          fontWeight: FontWeight.w600,
        ),
      ),
    ),
  ],
),
],
),
);
}
}

```

## Login Page

```

import 'package:bus_tracking_app/pages/signup.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'home.dart';

void main() {
  runApp(MaterialApp(home: LogIn()));
}

class LogIn extends StatefulWidget {
  @override
  _LogInState createState() => _LogInState();
}

class _LogInState extends State<LogIn> {
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();

  String? _errorMessage; // To store error messages for invalid login

  @override
  void dispose() {
    emailController.dispose();
    passwordController.dispose();
    super.dispose();
  }
}

```

```

void _login() {
  String emailOrPhone = emailController.text;
  String password = passwordController.text;

  if (_validateLogin(emailOrPhone, password)) {
    // Handle successful login (for now, just print to console)
    print('Login successful');
    Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => HomePage()),
    );
  } else {
    setState(() {
      _errorMessage = 'Invalid email/phone number or password';
    });
  }
}

bool _validateLogin(String emailOrPhone, String password) {
  // Simple validation logic for demonstration purposes
  // You can replace this with real authentication logic (e.g., API call)

  if (emailOrPhone == 'abc@gmail.com' && password == '12345') {
    return true; // Simulating successful login
  }
  return false; // Simulating failed login
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: const Color(0xFFFFFAFF),
    body: Center(
      child: SingleChildScrollView(
        child: Column(
          children: [
            const SizedBox(height: 40),
            Text(
              'STUDENT',
              style: GoogleFonts.lexend(
                textStyle: Theme.of(context).textTheme.headlineMedium,
                fontSize: 24,
                fontWeight: FontWeight.w600,
              ),
            ),
            const SizedBox(height: 80),
          ],
        ),
      ),
    ),
  );
}

```

```

// Email Input
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 30),
  child: TextField(
    controller: emailController,
    decoration: InputDecoration(
      labelText: 'Email or Phone Number',
      labelStyle: GoogleFonts.lexend(
        fontSize: 11,
        fontWeight: FontWeight.w400,
        color: Colors.black,
      ),
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(7),
        borderSide: const BorderSide(color: Colors.black),
      ),
      filled: true,
      fillColor: Colors.white,
    ),
  ),
),
const SizedBox(height: 30),

// Password Input
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 30),
  child: TextField(
    controller: passwordController,
    obscureText: true,
    decoration: InputDecoration(
      labelText: 'Password',
      labelStyle: GoogleFonts.lexend(
        fontSize: 11,
        fontWeight: FontWeight.w400,
        color: Colors.black,
      ),
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(7),
        borderSide: const BorderSide(color: Colors.black),
      ),
      filled: true,
      fillColor: Colors.white,
    ),
  ),
),
const SizedBox(height: 50),

```

```

// Display error message if login fails
if (_errorMessage != null)
  Padding(
    padding: const EdgeInsets.symmetric(horizontal: 30),
    child: Text(
      _errorMessage!,
      style: TextStyle(color: Colors.red, fontSize: 14),
    ),
  ),
const SizedBox(height: 20),

// Log In Button
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 72),
  child: Container(
    width: double.infinity,
    height: 40,
    decoration: BoxDecoration(
      color: const Color(0xFF0B7FF4),
      borderRadius: BorderRadius.circular(6),
    ),
    child: TextButton(
      onPressed: _login, // Call _signIn on button press
      child: Text(
        'Login',
        style: GoogleFonts.leagueSpartan(
          fontSize: 16,
          color: Colors.white,
          fontWeight: FontWeight.w600,
        ),
      ),
    ),
  ),
),
const SizedBox(height: 70),

// OR Divider
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    const Expanded(
      child: Divider(
        color: Color(0xFFDADDD8),
        thickness: 1,
        indent: 30,
        endIndent: 10,
      ),
    ),
  ],
),

```

```

    ),
    Text(
      'OR',
      style: GoogleFonts.lexend(
        fontSize: 8,
        fontWeight: FontWeight.w700,
      ),
    ),
  ),
  const Expanded(
    child: Divider(
      color: Color(0xFFDADDD8),
      thickness: 1,
      indent: 10,
      endIndent: 30,
    ),
  ),
],
),
const SizedBox(height: 40),

// Create New Account Button
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 72),
  child: Container(
    width: 130,
    height: 30,
    decoration: BoxDecoration(
      color: Colors.transparent,
      borderRadius: BorderRadius.circular(6),
    ),
    child: TextButton(
      onPressed: () {
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) => SignUp()),
        );
      },
      child: Text(
        'CREATE NEW ACCOUNT',
        style: GoogleFonts.lexend(
          fontSize: 8,
          color: const Color(0xFF0B7FF4),
          fontWeight: FontWeight.w600,
        ),
      ),
    ),
  ),
),
),
),
),

```

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



## Signup Page

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'home.dart';
import 'login.dart';

void main() {
  runApp(MaterialApp(home: SignUp()));
}

class SignUp extends StatefulWidget {
  @override
  _SignUpState createState() => _SignUpState();
}

class _SignUpState extends State<SignUp> {
  final TextEditingController fullnameController = TextEditingController();
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();
  final TextEditingController departmentController = TextEditingController();
  final TextEditingController semesterController = TextEditingController();
  final TextEditingController addressController = TextEditingController();

  final List<String> departments = [
    'BCA',
    'BSW',
    'BCom',
    'BA',
  ];

  String? selectedDepartment; // Variable to hold the selected department

  final List<String> semester = [
    'I',
    'II',
    'III',
    'IV',
    'V',
    'VI',
  ];

  String? selectedSemester; // Variable to hold the selected semester

  @override
  void dispose() {
    // Dispose of the controllers when the widget is removed from the widget tree
  }
}
```

```

fullnameController.dispose();
emailController.dispose();
passwordController.dispose();
departmentController.dispose();
semesterController.dispose();
addressController.dispose();
super.dispose();
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Color(0xFFFAFAFF),
    body: Center(
      child: SingleChildScrollView(
        child: Column(
          children: [
            SizedBox(height: 130),
            Text(
              'STUDENT',
              style: GoogleFonts.lexend(
                textStyle: Theme.of(context).textTheme.headlineMedium, // Changed to
headlineMedium
                fontSize: 24,
                fontWeight: FontWeight.w600,
              ),
            ),
            SizedBox(height: 60),

            // Full name Input
            Padding(
              padding: const EdgeInsets.symmetric(horizontal: 30),
              child: Container(
                width: 400,
                height: 45,
                child: TextField(
                  controller: fullnameController, // Use the controller here
                  decoration: InputDecoration(
                    labelText: 'Full Name',
                    labelStyle: GoogleFonts.lexend(
                      fontSize: 11,
                      fontWeight: FontWeight.w400,
                      color: Colors.black,
                    ),
                  ),
                  border: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(7),
                    borderSide: BorderSide(color: Colors.black),

```









[illegible]

```

        indent: 10,
        endIndent: 30,
      ),
    ),
  ],
),
 SizedBox(height: 30),

// Create New Account Button
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 72),
  child: Container(
    width: 60,
    height: 30,
    decoration: BoxDecoration(
      color: Colors.transparent,
      borderRadius: BorderRadius.circular(6),
    ),
    child: TextButton(
      onPressed: () {
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) => LogIn()),
        );
      },
      child: Text(
        'SIGN IN',
        style: GoogleFonts.lexend(
          fontSize: 8,
          color: const Color(0xFF0B7FF4),
          fontWeight: FontWeight.w600,
        ),
      ),
    ),
  ),
),
 SizedBox(height: 80),
],
),
),
),
);
}
}

```



## Home Page

```
import 'package:bus_tracking_app/pages/driversdetails.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'map.dart';
import 'farelist.dart';

void main() {
  runApp(const HomePage());
}

class HomePage extends StatelessWidget {
  const HomePage({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: const Color(0xFFFFFAFAFF),
      body: SingleChildScrollView(
        child: Column(
          children: [
            // Header
            const SizedBox(height: 60),
            Text(
              'HATIM\nBUS TRACKING APP',
              textAlign: TextAlign.center,
              style: GoogleFonts.lexend(
                textStyle: Theme.of(context).textTheme.displayLarge,
                fontSize: 16,
                fontWeight: FontWeight.w600,
              ),
            ),
            const SizedBox(height: 40),
            Text(
              'MAP PAGE',
              textAlign: TextAlign.left,
              style: GoogleFonts.leagueSpartan(
                textStyle: Theme.of(context).textTheme.displayLarge,
                fontSize: 16,
                fontWeight: FontWeight.w600,
              ),
            ),
            const SizedBox(height: 10),

            // Map Image
            GestureDetector(
```

```

onTap: () {
  Navigator.push(
    context,
    MaterialPageRoute(builder: (context) => MapPage()),
  );
},
child: Container(
  width: 340,
  height: 220,
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(15),
    image: DecorationImage(
      image: AssetImage('assets/map.png'), // Update with your image path
      fit: BoxFit.cover,
    ),
  ),
),
),
const SizedBox(height: 25),

// Bus Drivers' Profiles
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 25),
  child: InkWell(
    onTap: () {
      Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => DriverDetails()),
      );
    },
    child: Container(
      padding: const EdgeInsets.all(20),
      decoration: BoxDecoration(
        color: const Color(0xFFEEF0F2),
        borderRadius: BorderRadius.circular(15),
      ),
      child: Column(
        children: [
          Text(
            'Bus Drivers' Profile',
            textAlign: TextAlign.center,
            style: GoogleFonts.leagueSpartan(
              textStyle: Theme.of(context).textTheme.displayLarge,
              fontSize: 16,
              fontWeight: FontWeight.w600,
            ),
          ),
        ],
      ),
    ),
  ),
)

```

```

const SizedBox(height: 20),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    buildProfileImage(),
    buildProfileImage(),
    buildProfileImage(),
  ],
),
const SizedBox(height: 20),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    buildProfileImage(),
    buildProfileImage(),
    buildProfileImage(),
  ],
),
],
),
),
),
),
const SizedBox(height: 30),

// Bus Fare List Section
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 25),
  child: InkWell(
    onTap: () {
      Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => FareList()),
      );
    },
    child: Container(
      width: double.infinity,
      padding: const EdgeInsets.all(20),
      decoration: BoxDecoration(
        color: const Color(0xFFEEF0F2),
        borderRadius: BorderRadius.circular(15),
      ),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text(
            'Bus Fare List',

```

```

        style: GoogleFonts.lexend(
          fontSize: 16,
          fontWeight: FontWeight.w600,
        ),
      ),
    ],
  ),
),
),
const SizedBox(height: 30),
],
),
),
);
}

```

// Helper widget to build profile images

```

Widget buildProfileImage() {
  return Container(
    width: 71,
    height: 70,
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(13),
      image: DecorationImage(
        image: AssetImage('assets/profile.png'), // Update with your image path
        fit: BoxFit.cover,
      ),
    ),
  );
}
}

```

## Map Page

```
import 'package:flutter/material.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';
import 'package:geolocator/geolocator.dart';
import 'package:google_fonts/google_fonts.dart';
import 'dart:async';

class MapPage extends StatefulWidget {
  const MapPage({super.key});

  @override
  State<MapPage> createState() => _MapPageState();
}

class _MapPageState extends State<MapPage> {
  // Destination coordinates
  LatLng hatim = const LatLng(22.94692858593475, 92.76744634309378);

  GoogleMapController? googleMapController;
  Set<Marker> marker = {};

  // Custom marker icons
  BitmapDescriptor destinationIcon = BitmapDescriptor.defaultMarker;
  BitmapDescriptor currentIcon = BitmapDescriptor.defaultMarker;

  bool _isLocationTrackingEnabled = false; // Track location tracking state
  StreamSubscription<Position>? _positionStreamSubscription;

  @override
  void initState() {
    super.initState();
    setCustomMarkerIcon();
  }

  // Set custom marker icons for destination and current location
  void setCustomMarkerIcon() async {
    destinationIcon = await BitmapDescriptor.fromAssetImage(
      const ImageConfiguration(),
      "assets/des.png", // Replace with your actual asset path
    );
    currentIcon = await BitmapDescriptor.fromAssetImage(
      const ImageConfiguration(),
      "assets/bus.png", // Replace with your actual asset path
    );
  }
```

```

// Method to start listening to location updates
void _startLocationTracking() {
  // Check if location service is enabled and permission is granted
  Geolocator.requestPermission().then((permission) {
    if (permission == LocationPermission.denied ||
        permission == LocationPermission.deniedForever) {
      return;
    }

    // Listen to location updates and move marker and camera
    _positionStreamSubscription = Geolocator.getPositionStream(
      locationSettings: const LocationSettings(
        accuracy: LocationAccuracy.high,
        distanceFilter: 10, // Only update if the user moves 10 meters
      ),
    ).listen((Position position) {
      LatLng currentLatLng = LatLng(position.latitude, position.longitude);

      setState(() {
        marker.clear();
        marker.add(
          Marker(
            markerId: const MarkerId("CurrentLocation"),
            icon: currentIcon,
            position: currentLatLng,
          ),
        );
      });

      if (googleMapController != null) {
        googleMapController!.animateCamera(
          CameraUpdate.newCameraPosition(
            CameraPosition(
              target: currentLatLng,
              zoom: 17,
            ),
          ),
        );
      }
    });
  });
}

// Method to stop location tracking
void _stopLocationTracking() {
  if (_positionStreamSubscription != null) {
    _positionStreamSubscription!.cancel();
  }
}

```

```

    _positionStreamSubscription = null;
  }

  setState(() {
    _isLocationTrackingEnabled = false;
    marker.clear();
  });
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      backgroundColor: Colors.white.withOpacity(
        0.7), // Optional: Add a slight transparency for better view of map
      title: Text(
        "MAP PAGE",
        style: GoogleFonts.lexend(
          fontSize: 19,
          fontWeight: FontWeight.w500,
          color: Colors.black,
        ),
      ),
    ),
    body: Stack(
      children: [
        // Google Map
        GoogleMap(
          myLocationButtonEnabled: false,
          zoomControlsEnabled: false,
          markers: {
            Marker(
              markerId: const MarkerId("test"),
              icon: destinationIcon,
              position: hatim,
            ),
            ...marker,
          },
          onMapCreated: (GoogleMapController controller) {
            googleMapController = controller;
          },
          mapType: MapType.hybrid,
          initialCameraPosition: CameraPosition(
            target: hatim,
            zoom: 15,
          ),
        ),
      ],
    ),
  );
}

```

```

// Add a Floating Action Button to toggle location tracking
Positioned(
  right: 15,
  bottom: 15,
  child: FloatingActionButton(
    onPressed: () {
      setState(() {
        _isLocationTrackingEnabled = !_isLocationTrackingEnabled;
      });

      if (_isLocationTrackingEnabled) {
        _startLocationTracking();
      } else {
        _stopLocationTracking();
      }
    },
    backgroundColor: Colors.white,
    child: Icon(
      _isLocationTrackingEnabled
        ? Icons.location_off
        : Icons.location_on,
      size: 30,
    ),
  ),
),
],
),
);
}
}

```



## Driver Details Page

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

void main() {
  runApp(const DriverDetails());
}

class DriverDetails extends StatelessWidget {
  const DriverDetails({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false, // Removes the debug banner
      home: Scaffold(
        body: Center(
          child: SingleChildScrollView(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                // Header Text
                Padding(
                  padding: const EdgeInsets.only(top: 0, bottom: 16), // Add some padding above
and below the header
                  child: Text(
                    'DRIVER DETAILS',
                    style: GoogleFonts.lexend(
                      fontSize: 20,
                      fontWeight: FontWeight.w600,
                    ),
                  ),
                ),
                // Driver Details Boxes
                buildDriverBox(context, 'assets/zauva.jpg', 'C. Lalbiakzauva', 'Bus No 1',
'9862867130'),
                const SizedBox(height: 32), // Space between the box
                buildDriverBox(context, 'assets/fela.jpg', 'B. Lalruatfela', 'Bus No 2', '8729861475'),
                const SizedBox(height: 32),
                buildDriverBox(context, 'assets/mawia.jpg', 'Lalrammawia Sailo', 'Bus No 3',
'9436157319'),
                const SizedBox(height: 32),
                buildDriverBox(context, 'assets/denga.jpg', 'MC Denga', 'Bus No 4', '9089568549'),
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```

```

    ),
  ),
);
}

```

// Method to build the driver details box, with parameters for image, name, bus number, and phone

```

Widget buildDriverBox(BuildContext context, String imagePath, String name, String busNo, String phone) {

```

```

  return Container(
    padding: const EdgeInsets.all(16),
    margin: const EdgeInsets.symmetric(horizontal: 16),
    decoration: BoxDecoration(
      color: const Color(0xFFEEF0F2),
      borderRadius: BorderRadius.circular(15),
    ),
    child: Row(
      children: [
        // Driver Image
        Container(
          width: 70,
          height: 92,
          decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(14),
            image: DecorationImage(
              image: AssetImage(imagePath),
              fit: BoxFit.cover,
            ),
          ),
        ),
        const SizedBox(width: 16),
        // Driver Details
        Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text(
              name,
              style: GoogleFonts.lexend(
                fontSize: 16,
                fontWeight: FontWeight.w500,
              ),
            ),
            const SizedBox(height: 8),
            Row(
              children: [
                const Icon(Icons.directions_bus, size: 16),
                const SizedBox(width: 8),

```

```

        Text(
          busNo,
          style: GoogleFonts.lexend(
            fontSize: 14,
            fontWeight: FontWeight.w400,
          ),
        ),
      ],
    ),
    const SizedBox(height: 8),
    Row(
      children: [
        const Icon(Icons.phone, size: 16),
        const SizedBox(width: 8),
        Text(
          phone,
          style: GoogleFonts.lexend(
            fontSize: 14,
            fontWeight: FontWeight.w400,
          ),
        ),
      ],
    ),
  ],
),
],
),
);
}
}

```

## Bus Fare List Page

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

void main() {
  runApp(const FareList());
}

class FareList extends StatelessWidget {
  const FareList({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        backgroundColor: const Color(0xFFFFFAFF),
        body: SingleChildScrollView(
          child: Center(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                // Header Text
                Padding(
                  padding: const EdgeInsets.only(top: 50), // Add some space from top
                  child: Text(
                    'BUS FARE LIST',
                    style: GoogleFonts.lexend(
                      textStyle: Theme.of(context).textTheme.headlineMedium,
                      fontSize: 18,
                      fontWeight: FontWeight.w600,
                    ),
                  ),
                // Add space between the header and the first box
                SizedBox(height: 15),

                // Fare Boxes
                buildFareBox(
                  context,
                  'Rs. 8,500 per Semester',
                  'Rs. 1,750 per Month',
                  'Rs. 150 One Way',
                  'THEIRIAT',
                ),
                buildFareBox(
                  context,
```

```

        'Rs. 8,000 per Semester',
        'Rs. 1,650 per Month',
        'Rs. 140 One Way',
        'SETHLUN',
    ),
    buildFareBox(
        context,
        'Rs. 7,750 per Semester',
        'Rs. 1,600 per Month',
        'Rs. 130 One Way',
        '3 GATE',
    ),
    buildFareBox(
        context,
        'Rs. 6,750 per Semester',
        'Rs. 1,400 per Month',
        'Rs. 120 One Way',
        'LUNGLAWN',
    ),
    buildFareBox(
        context,
        'Rs. 6,500 per Semester',
        'Rs. 1,350 per Month',
        'Rs. 120 One Way',
        'KAWIZAU',
    ),
    // Add more boxes as needed
    SizedBox(height: 50),
  ],
),
),
),
),
);
}

```

```

// Function to build a fare box
Widget buildFareBox(BuildContext context, String semesterPrice, String monthlyPrice,
String oneWayPrice, String title) {
  return Container(
    width: 330, // Increased width
    height: 120,
    margin: const EdgeInsets.symmetric(vertical: 10),
    decoration: BoxDecoration(
      color: Colors.white,
    ),
    child: Stack(

```

```

children: [
  Positioned(
    left: 0,
    top: 0,
    child: Container(
      width: 330, // Increased width
      height: 120,
      decoration: ShapeDecoration(
        color: Color(0xFFEEF0F2),
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(15),
        ),
      ),
    ),
  ),
  Positioned(
    left: 21,
    top: 41,
    child: Text(
      semesterPrice,
      style: GoogleFonts.lexend(
        textStyle: Theme.of(context).textTheme.headlineMedium,
        fontSize: 14,
        fontWeight: FontWeight.w400,
      ),
    ),
  ),
  Positioned(
    left: 21,
    top: 63,
    child: Text(
      monthlyPrice,
      style: GoogleFonts.lexend(
        textStyle: Theme.of(context).textTheme.headlineMedium,
        fontSize: 14,
        fontWeight: FontWeight.w400,
      ),
    ),
  ),
  Positioned(
    left: 21,
    top: 85,
    child: Text(
      oneWayPrice,
      style: GoogleFonts.lexend(
        textStyle: Theme.of(context).textTheme.headlineMedium,
        fontSize: 14,

```

```

        fontWeight: FontWeight.w400,
      ),
    ),
  ),
  // Centering the title
  Align(
    alignment: Alignment.topCenter,
    child: Padding(
      padding: const EdgeInsets.only(top: 14),
      child: Text(
        title,
        style: GoogleFonts.lexend(
          textStyle: Theme.of(context).textTheme.headlineMedium,
          fontSize: 16,
          fontWeight: FontWeight.w500,
        ),
      ),
    ),
  ),
),
),
),
),
),
),
);
}
}

```

## **6. CODING – ADMIN (Driver)**

### **CONTENT**

- 
- **Drivers' Login Page**
  - **Drivers' Home Page**
  - **Drivers' Map Page**



## CODING – ADMIN (Student)

### Drivers' Login Page

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

void main() {
  runApp(MaterialApp(home: DriverLogin()));
}

class DriverLogin extends StatefulWidget {
  @override
  _DriverLoginState createState() => _DriverLoginState();
}

class _DriverLoginState extends State<DriverLogin> {
  final TextEditingController idController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();

  String? _errorMessage; // To store error messages for invalid login

  @override
  void dispose() {
    idController.dispose();
    passwordController.dispose();
    super.dispose();
  }

  void _login() {
    String driverid = idController.text;
    String password = passwordController.text;

    if (_validateLogin(driverid, password)) {
      // Handle successful login
      print('Login successful');
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => DriverHomePage()), // Navigate to the
DriverHomePage
      );
    } else {
      setState(() {
        _errorMessage = 'Invalid Driver ID or Password'; // Show error message
      });
    }
  }
}
```

```

}

bool _validateLogin(String driverId, String password) {
  // Simple validation logic for demonstration purposes
  // Replace this with real authentication logic
  if (driverId == '12345' && password == 'qwerty') {
    return true; // Simulating successful login
  }
  return false; // Simulating failed login
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: const Color(0xFFFAFAFF),
    body: Center(
      child: SingleChildScrollView(
        child: Column(
          children: [
            const SizedBox(height: 1),
            Text(
              'DRIVER',
              style: GoogleFonts.lexend(
                textStyle: Theme.of(context).textTheme.headlineMedium,
                fontSize: 24,
                fontWeight: FontWeight.w600,
              ),
            ),
            const SizedBox(height: 50),

            // Driver ID Input
            Padding(
              padding: const EdgeInsets.symmetric(horizontal: 30),
              child: TextField(
                controller: idController,
                decoration: InputDecoration(
                  labelText: 'Driver ID',
                  labelStyle: GoogleFonts.lexend(
                    fontSize: 11,
                    fontWeight: FontWeight.w400,
                    color: Colors.black,
                  ),
                ),
                border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(7),
                  borderSide: const BorderSide(color: Colors.black),
                ),
                filled: true,

```

```

        fillColor: Colors.white,
      ),
    ),
  ),
  const SizedBox(height: 25),

  // Password Input
  Padding(
    padding: const EdgeInsets.symmetric(horizontal: 30),
    child: TextField(
      controller: passwordController,
      obscureText: true,
      decoration: InputDecoration(
        labelText: 'Password',
        labelStyle: GoogleFonts.lexend(
          fontSize: 11,
          fontWeight: FontWeight.w400,
          color: Colors.black,
        ),
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(7),
          borderSide: const BorderSide(color: Colors.black),
        ),
        filled: true,
        fillColor: Colors.white,
      ),
    ),
  ),
  const SizedBox(height: 40),

  // Display error message if login fails
  if (_errorMessage != null)
    Padding(
      padding: const EdgeInsets.symmetric(horizontal: 30),
      child: Text(
        _errorMessage!,
        style: TextStyle(color: Colors.red, fontSize: 14),
      ),
    ),
  const SizedBox(height: 20),

  // Log In Button
  Padding(
    padding: const EdgeInsets.symmetric(horizontal: 72),
    child: Container(
      width: double.infinity,
      height: 40,

```

```

decoration: BoxDecoration(
  color: const Color(0xFF0B7FF4),
  borderRadius: BorderRadius.circular(6),
),
child: TextButton(
  onPressed: _login, // Call _signIn on button press
  child: Text(
    'Login',
    style: GoogleFonts.leagueSpartan(
      fontSize: 16,
      color: Colors.white,
      fontWeight: FontWeight.w600,
    ),
  ),
),
),
),
),
),
),
),
),
),
);
}
}

```

## Drivers' Home Page

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

void main() {
  runApp(const DriverHomePage());
}

class DriverHomePage extends StatelessWidget {
  const DriverHomePage({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: const Color(0xFFFAFAFF),
      body: SingleChildScrollView(
        child: Column(
          children: [
            // Header
            const SizedBox(height: 70),
            Center( // Horizontally center the text
              child: Text(
                'ADMIN',
                textAlign: TextAlign.center,
                style: GoogleFonts.lexend(
                  textStyle: Theme.of(context).textTheme.displayLarge,
                  fontSize: 18,
                  fontWeight: FontWeight.w700,
                ),
              ),
            ),
            const SizedBox(height: 40),
            Center( // Horizontally center the text
              child: Text(
                'DRIVER : HOME',
                textAlign: TextAlign.center,
                style: GoogleFonts.lexend(
                  textStyle: Theme.of(context).textTheme.displayLarge,
                  fontSize: 16,
                  fontWeight: FontWeight.w600,
                ),
              ),
            ),
            const SizedBox(height: 40),
            Center( // Horizontally center the text
```

```

child: Text(
  'MAP PAGE',
  textAlign: TextAlign.left,
  style: GoogleFonts.leagueSpartan(
    textStyle: Theme.of(context).textTheme.displayLarge,
    fontSize: 16,
    fontWeight: FontWeight.w600,
  ),
),
),
),
const SizedBox(height: 10),

// Map Image
Center( // Horizontally center the image
  child: GestureDetector(
    onTap: () {
      Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => DriverMapPage()),
      );
    },
    child: Container(
      width: 340,
      height: 220,
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(15),
        image: DecorationImage(
          image: AssetImage('assets/map.png'),
          fit: BoxFit.cover,
        ),
      ),
    ),
  ),
),
const SizedBox(height: 25),
],
),
),
);
}
}

```

## Driver's Map Page

```
import 'package:flutter/material.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';
import 'package:geolocator/geolocator.dart';
import 'package:google_fonts/google_fonts.dart';
import 'dart:async';

class DriverMapPage extends StatefulWidget {
  const DriverMapPage({super.key});

  @override
  State<DriverMapPage> createState() => _DriverMapPageState();
}

class _DriverMapPageState extends State<DriverMapPage> {
  // Destination coordinates
  LatLng hatim = const LatLng(22.94692858593475, 92.76744634309378);
  LatLng faith = const LatLng(22.883923, 92.747116);
  LatLng theiriat = const LatLng(22.866766, 92.783409);

  GoogleMapController? googleMapController;
  Set<Marker> marker = {};

  // Custom marker icons
  BitmapDescriptor destinationIcon = BitmapDescriptor.defaultMarker;
  BitmapDescriptor currentIcon = BitmapDescriptor.defaultMarker;

  bool _isLocationTrackingEnabled = false; // Track location tracking state
  StreamSubscription<Position>? _positionStreamSubscription;
  LatLng? selectedDestination; // Store the selected destination point

  @override
  void initState() {
    super.initState();
    setCustomMarkerIcon();
  }

  // Set custom marker icons for destination and current location
  void setCustomMarkerIcon() async {
    destinationIcon = await BitmapDescriptor.fromAssetImage(
      const ImageConfiguration(),
      "assets/des.png", // Replace with your actual asset path
    );
    currentIcon = await BitmapDescriptor.fromAssetImage(
      const ImageConfiguration(),
      "assets/bus.png", // Replace with your actual asset path
    );
  }
}
```

```

    );
}

// Method to start listening to location updates
void _startLocationTracking() {
    // Check if location service is enabled and permission is granted
    Geolocator.requestPermission().then((permission) {
        if (permission == LocationPermission.denied ||
            permission == LocationPermission.deniedForever) {
            return;
        }

        // Listen to location updates and move marker and camera
        _positionStreamSubscription = Geolocator.getPositionStream(
            locationSettings: const LocationSettings(
                accuracy: LocationAccuracy.high,
                distanceFilter: 10, // Only update if the user moves 10 meters
            ),
        ).listen((Position position) {
            LatLng currentLatLng = LatLng(position.latitude, position.longitude);

            setState(() {
                marker.clear();
                marker.add(
                    Marker(
                        markerId: const MarkerId("CurrentLocation"),
                        icon: currentIcon,
                        position: currentLatLng,
                    ),
                );
            });

            // Add the selected destination marker if destination is selected
            if (selectedDestination != null) {
                marker.add(
                    Marker(
                        markerId: const MarkerId("Destination"),
                        icon: destinationIcon,
                        position: selectedDestination!,
                    ),
                );
            }
        });

        if (googleMapController != null) {
            googleMapController!.animateCamera(
                CameraUpdate.newCameraPosition(
                    CameraPosition(

```



```

        target: currentLatLng,
        zoom: 17,
    ),
),
);
}
});
});
}

// Method to stop location tracking
void _stopLocationTracking() {
  if (_positionStreamSubscription != null) {
    _positionStreamSubscription!.cancel();
    _positionStreamSubscription = null;
  }

  setState(() {
    _isLocationTrackingEnabled = false;
    marker.clear();
  });
}

// Method to select a destination from given LatLng
void _selectDestination(LatLng destination, String name) {
  setState(() {
    selectedDestination = destination;
    // Optionally, move the camera to the selected destination
    googleMapController?.animateCamera(
      CameraUpdate.newCameraPosition(
        CameraPosition(
          target: destination,
          zoom: 15,
        ),
      ),
    );
    // Show a snack bar to confirm the selection
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text("Selected destination: $name")),
    );
  });
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(

```

backgroundColor: Colors.white.withOpacity(0.7), // Optional: Add a slight transparency  
for better view of map

```
title: Text(
  "MAP PAGE",
  style: GoogleFonts.lexend(
    fontSize: 17,
    fontWeight: FontWeight.w500,
    color: Colors.black,
  ),
),
),
body: Stack(
  children: [
    // Google Map
    GoogleMap(
      myLocationButtonEnabled: false,
      zoomControlsEnabled: false,
      markers: marker,
      onMapCreated: (GoogleMapController controller) {
        googleMapController = controller;
      },
      mapType: MapType.hybrid,
      initialCameraPosition: CameraPosition(
        target: hatim,
        zoom: 15,
      ),
    ),
    // Add a Floating Action Button to toggle location tracking
    Positioned(
      right: 15,
      bottom: 15,
      child: FloatingActionButton(
        onPressed: () {
          setState(() {
            _isLocationTrackingEnabled = !_isLocationTrackingEnabled;
          });

          if (_isLocationTrackingEnabled) {
            _startLocationTracking();
          } else {
            _stopLocationTracking();
          }
        },
        backgroundColor: Colors.white,
        child: Icon(
          _isLocationTrackingEnabled
            ? Icons.location_off
```

```

        : Icons.location_on,
        size: 30,
      ),
    ),
  ),
  // Main button for destination selection with custom background color
  Positioned(
    left: 325,
    bottom: 610,
    child: Material(
      color: Colors.amber, // Change this to your desired background color
      borderRadius: BorderRadius.circular(12), // Rounded corners
      child: InkWell(
        borderRadius: BorderRadius.circular(30), // Match the rounded corners
        onTap: () {
          // Open the dropdown menu when the button is tapped
          showMenu(
            context: context,
            position: RelativeRect.fromLTRB(100, 200, 0, 0),
            items: [
              PopupMenuItem<LatLng>(
                value: hatim,
                padding: EdgeInsets.symmetric(vertical: 0), // Even tighter padding
                child: Text(
                  "  Hatim",
                  style: TextStyle(fontSize: 11), // Smaller font size for tighter spacing
                ),
              ),
              PopupMenuItem<LatLng>(
                value: faith,
                padding: EdgeInsets.symmetric(vertical: 0), // Even tighter padding
                child: Text(
                  "  Faith Hospital",
                  style: TextStyle(fontSize: 11), // Smaller font size for tighter spacing
                ),
              ),
              PopupMenuItem<LatLng>(
                value: theiriat,
                padding: EdgeInsets.symmetric(vertical: 0), // Even tighter padding
                child: Text(
                  "  Theiriat",
                  style: TextStyle(fontSize: 11), // Smaller font size for tighter spacing
                ),
              ),
            ],
          ).then((value) {
            if (value != null) {

```

```

        String name = "";
        if (value == hatim) {
            name = "Hatim";
        } else if (value == faith) {
            name = "Faith Hospital";
        } else if (value == theiriat) {
            name = "Theiriat";
        }
        _selectDestination(value, name);
    }
});
},
child: Padding(
    padding: const EdgeInsets.all(10.0),
    child: Icon(Icons.place, size: 35, color: Colors.white), // Custom icon
),
),
),
),
1,
),
);
}
}

```

## 7. TESTING AND IMPLEMENTATION

### CONTENT

- 
- Testing
  - Possible Problem

## TESTING AND IMPLEMENTATION

### Testing

This project has undergone many series of tests, and errors present have been dealt with. But there may be unforeseen errors which are however unlikely. With the amount of time we have, we tried out best to reach as far as we could and run trial many times, so that it will be usable for all in the near future.

### Possible Problem

While the app aims to deliver a seamless and efficient experience, certain challenges may arise that could hinder its functionality and user satisfaction. Still, it faces several challenges that could affect its functionality and user satisfaction. By addressing potential problems like GPS inaccuracy, real-time data delays, and server overloads, the team is working to improve the app's performance and ensure a smooth user experience. Thorough testing, continuous monitoring, and the implementation of proactive solutions will be crucial to minimizing these issues and enhancing the app's reliability and user engagement.

This proactive approach will ensure the app remains competitive and functional in a dynamic real-world environment.

## 8. DRAWBACKS AND LIMITATIONS

This project is undertaken with careful analysis to meet the given criteria of this final project. Corrective measures have been taken. But our insufficiency of knowledge, skill and experience, this project definitely has its own limitations and drawbacks. It definitely cannot have a wide scope as found in other big projects due to the limitation of time. It is primarily a compilation of only what information we thought would be relevant to satisfy the basic requirements of the project.

## 9. FUTURE ENHANCEMENT

Even though the project seems completed and the connection between the frontend and backend works properly, so much more enhancement is a necessity in order to produce this project online.

1. **Real-Time Features**

Enhance GPS accuracy to provide real-time bus location updates with ETAs.

2. **Scalability**

Use cloud databases for scalable storage, ensuring seamless data handling for increasing users.

3. **Enhanced Security**

Log database access and changes for secure monitoring of system usage.

4. **Database Connection**

Connecting to Database for Storing necessary information like student's login info- email, password, department, etc.



## 10. CONCLUSION

This project is basically designed to be used by the students. This chapter seeks to conclude the work that has been done, and to present the areas that mark to the completion of this project work.

This project work is the compilation of the ideas, views and thoughts of our group, Group 1 of 5th Semester BCA, 2024.

We are deeply conscious of the fact that this project would neither have been undertaken nor pursued and completed but for the tremendous support that we received from Mr. H. Lalruatkima, our Project Guide and K. Lalmuanpuia Head of Department, who not only give us their full support but provide us with all kinds of necessities that we required in the project we are working.

We would also like to place and record our sincere thanks and gratitude to our Principal, Mr. Vuansanga Vanchhawng, for extending his full support and consent to this project undertaken.