

## ABSTRACT

Navigating a large educational institution like the Mangalore Institute of Technology and Engineering (MITE) can be challenging, especially for new students, guests, and event participants unfamiliar with the campus layout. To address this, the project titled "**Smart Campus Navigator for MITE**" aims to simplify campus navigation through a mobile-based solution that provides visual route guidance using Google Maps.

This application is developed using **Flutter**, a cross-platform UI toolkit, and integrates the **Google Maps API** to display the MITE campus map with custom markers and route lines (polylines). Users can log in or sign up through a secure **Firebase Authentication** system, select their **starting location** and **destination** from a predefined list of campus landmarks, and view the optimal path visually on the map.

Key features of the application include:

- **User Authentication:** Sign-up and login functionality using Firebase.
- **Location Selection:** Dropdown lists to choose starting and ending points on campus.
- **Route Visualization:** Dynamic map rendering with start and end markers, and a polyline showing the shortest route.
- **Map Integration:** Real-time map support using the Google Maps API, ensuring accurate navigation and an interactive interface.
- **Simple and Intuitive UI:** Designed using Flutter for seamless performance on both Android and iOS platforms.

This project helps students and visitors navigate efficiently within the campus, reducing confusion and saving time. It is particularly helpful during campus events, first-year admissions, and placement drives, where participants often need quick guidance.

The system focuses on **practical usability**, **cost-effectiveness**, and **ease of maintenance**. It does not require any special hardware like GPS devices or QR scanners — all functionality is powered by APIs and cloud services, making it scalable and easy to deploy. While the current version focuses on basic route navigation, future updates can include features like admin-based location updates, real-time directions, and indoor mapping.

# INTRODUCTION

## 1.1 Overview

Navigating large educational campuses such as the Manipal Institute of Technology and Engineering (MITE) can be a challenging experience, particularly for first-time visitors, newly admitted students, or newly recruited staff members. The complex layout, multiple departments, administrative blocks, hostels, libraries, sports complexes, and other facilities can create confusion and lead to time delays in reaching specific destinations. To address this issue effectively, the Smart Campus Navigator mobile application has been developed.

The Smart Campus Navigator is designed to provide an efficient, user-friendly, and real-time navigation solution within the boundaries of the MITE campus. It aims to simplify campus mobility by helping users find the shortest and most efficient paths between two selected points on the campus. Whether it's locating a particular classroom, department, cafeteria, or event venue, the application ensures users can reach their destinations without unnecessary detours or confusion.

This mobile application is built using Flutter, a robust and flexible framework by Google that enables seamless cross-platform development. This ensures the application is compatible with both Android and iOS devices, thus making it accessible to a wide range of users without platform limitations.

At the core of the application is the integration of the Google Maps API, which provides rich map data, real-time routing capabilities, and interactive map components. Users can select a starting point and a destination from a predefined list of locations or directly on the map interface. Once selected, the application computes and displays the most optimal route using visual elements such as markers for locations and polylines to indicate the navigation path. These visual indicators offer a clear and intuitive understanding of the route, improving the overall user experience.

In addition to static pathfinding, the Smart Campus Navigator is designed to handle dynamic interactions such as zooming, panning, and live location tracking. This ensures that users are always aware of their current position relative to their destination, enhancing usability and reliability.

By combining modern mobile development practices with powerful mapping technology, the Smart Campus Navigator serves as a digital guide to improve campus orientation, reduce navigation-related stress, and promote smarter, technology-driven campus experiences.

## 1.2 Problem Statement

Large academic institutions like MITE often consist of a vast and complex network of buildings, departments, labs, classrooms, libraries, hostels, sports arenas, and administrative blocks. For individuals unfamiliar with the campus layout—such as newly admitted students, visiting parents, guest lecturers, event participants, or even new faculty and staff—navigating this environment can be an overwhelming and time-consuming experience.

Traditionally, newcomers rely on static maps displayed on notice boards, printed campus layouts, or by asking for directions from others. However, these methods are inefficient, non-interactive, and often prone to miscommunication. Static maps cannot provide real-time guidance, lack location-specific details, and cannot adapt dynamically based on the user's current position or destination. Additionally, depending on individuals for directions can be unreliable, especially during busy hours or when there are language or communication barriers.

Problem Definition:

There is a significant gap in the availability of a smart, interactive, and user-friendly navigation system tailored specifically for internal campus environments. Existing map solutions like Google Maps primarily focus on external navigation and lack detailed data for internal routes within campuses, such as individual buildings, internal walkways, or designated academic zones.

Due to the absence of such a system, users often face unnecessary delays, confusion, and difficulty accessing specific campus facilities, which can affect punctuality, productivity, and the overall experience of being on campus. For students, this may result in late arrivals to lectures or labs; for visitors, it could mean missing important meetings or events; and for staff, it may impact workflow efficiency.

Therefore, there is a clear need for a mobile-based, intelligent navigation application that provides real-time guidance, interactive route mapping, and visual cues to ensure smooth and accurate movement within the campus.

The Smart Campus Navigator seeks to bridge this gap by offering a solution that is custom-built for the MITE campus, enabling users to navigate effortlessly and confidently without dependence on traditional, outdated methods.

## 1.3 Objectives

### 1. To design a mobile application that offers real-time navigation across the MITE campus.

The core goal is to create a mobile solution that enables users to navigate the campus accurately and in real-time. By allowing users to view their current location and follow the best possible route to their selected destination, the application reduces confusion, saves time, and improves the overall campus experience. The app will be available on both Android and iOS platforms using Flutter, ensuring cross-platform compatibility and wider accessibility.

### 2. To integrate Google Maps API for route display and marker interaction.

By integrating the **Google Maps API**, the application provides a dynamic and visually intuitive way to display the route between selected locations. The use of **markers** allows users to clearly identify both the starting point and destination on the map, while **polylines** illustrate the optimal path. The interactive map also supports panning, zooming, and other gestures to make navigation simple and clear.

### 3. To implement user authentication using Firebase.

To maintain a secure and personalized user experience, **Firebase Authentication** is incorporated into the app. This enables users to register and log in using their credentials, ensuring secure access and allowing for potential future features such as saving favorite locations, tracking usage history, or customizing preferences. Firebase also provides robust backend support and scalability.

### 4. To simplify location selection via dropdown menus.

To improve user experience and reduce manual effort, the app includes **dropdown menus** for selecting the starting point and destination from a predefined list of campus locations. This ensures consistency in naming, eliminates errors in input, and makes the app more accessible, especially for users unfamiliar with the exact names of buildings or departments.

### 5. To assist new students, parents, and guests during events like orientations or college fests.

The Smart Campus Navigator is particularly useful during large events such as **student orientations, cultural fests, technical symposiums, and parent-teacher meetings**, where the number of campus visitors significantly increases. These users often face difficulty locating event venues, hostels, food stalls, or help desks.

## LITERATURE SURVEY

Traditional navigation apps like **Google Maps** and **Apple Maps** are highly effective for city-wide routing but fall short when it comes to indoor or campus-specific navigation. This creates challenges for students, faculty, and visitors trying to find locations within large educational campuses like MITE.

Studies show that **Location-Based Services (LBS)**, when combined with user-friendly interfaces, significantly improve navigation in confined environments. However, existing campus navigation solutions each come with limitations:

### Existing Projects:

- **College Map Navigator (2020):**  
Android-based app using static local map files. Offers basic navigation but lacks real-time updates and scalability.
- **Flutter + Google Maps Integrations:**  
These projects use Google Maps API in Flutter for real-time, scalable, and interactive navigation. They support cross-platform development and reduce complexity.

### Gap Identified:

Most existing solutions are either too basic (static maps) or too complex (hardware-dependent). There is a lack of a **simple, cost-effective, mobile-based navigation system** tailored for campus-level routing.

### Our Approach:

The **Smart Campus Navigator** addresses this gap by offering:

- Cross-platform mobile app using **Flutter**
- Real-time route display with **Google Maps API**
- Easy location selection via **dropdown menus**
- Marker and polyline-based route visualization

This approach ensures an efficient, low-maintenance, and accessible solution tailored to the MITE campus environment.

# SYSTEM REQUIREMENTS SPECIFICATION

## 3.1 Hardware Requirements

- **Mobile Device:**

The application requires a smartphone or tablet running **Android or iOS** operating systems. The device must have internet access to load real-time map data and navigation routes from Google Maps.

- **Processor:**

A processor speed of **1.8 GHz or higher** is recommended to ensure the app runs smoothly without delays, enabling quick loading and responsiveness during navigation.

- **RAM:**

At least **2 GB of RAM** is necessary to handle the app's processes efficiently, including map rendering, route calculations, and user interactions.

- **Storage:**

The device should have a minimum of **100 MB of free storage space** to accommodate the app installation and temporary data such as cached maps, user preferences, and offline information.

## 3.2 Software Requirements

- **Flutter SDK:**

Flutter is a cross-platform development framework that allows building natively compiled applications for both Android and iOS using a single codebase. It offers a rich set of pre-built UI components and ensures high performance.
- **Dart Language:**

Dart is the programming language used with Flutter. It supports reactive programming and provides an easy-to-learn syntax for developing interactive and efficient app features.
- **Firebase Authentication:**

Firebase Auth is used to manage user authentication securely and efficiently. It supports various sign-in methods such as email/password, Google sign-in, and social logins, providing seamless access control.
- **Google Maps API:**

This API integrates comprehensive mapping features into the app, including real-time route display, interactive markers, polylines for routes, and geolocation tracking to assist users in campus navigation.
- **Android Studio:**

Android Studio is the official integrated development environment (IDE) for Flutter and Android app development. It offers tools for writing, debugging, and testing the application, ensuring smooth development workflows.

# METHODOLOGY

## 1. Requirement Analysis

- Conducted a detailed study of navigation challenges faced by students, staff, and visitors within the MITE campus.
- Gathered user requirements to understand the essential features needed in the app such as route guidance, location selection, and real-time updates.

## 2. Design

- Created the app's **User Interface (UI)** prototypes
- Designed the overall **system architecture**, outlining the interaction between the app frontend, Firebase backend, and Google Maps services.

## 3. Development

- Set up the Flutter development environment to build a cross-platform mobile app.
- Integrated the **Google Maps API** for dynamic map rendering, route plotting, and interactive markers within the app.

## 4. Testing

- Performed **UI testing** to ensure all screens and controls are intuitive, responsive, and consistent across devices.
- Tested **route functionality** to verify accurate route calculation, marker placement, and polyline drawing between selected points.

## 5. Deployment

- Prepared the final app build and generated an **APK file** for Android platform distribution.
- Ensured the APK meets performance and security standards before release.



## IMPLEMENTATION

- The **Login and Signup** functionality is implemented using **Firebase Authentication**, enabling secure user access.
- The **Location List Screen** provides dropdown menus for users to select their start and destination points on campus.
- The **Google Maps Widget** displays an interactive map interface.
- When both start and destination points are selected, the app dynamically draws a route on the map using **polylines**.

### CODE SNIPPET:

#### main.dart

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'MITE Campus Navigation',
      theme: ThemeData(primarySwatch: Colors.blue),
      home: LoginPage(),
    );
  }
}
```

#### Login\_page.dart

```
import 'package:flutter/material.dart';
import 'signup_page.dart';
import 'location_list_page.dart';

class LoginPage extends StatelessWidget {
```

```

    final TextEditingController emailController =
TextEditingController();
    final TextEditingController passwordController =
TextEditingController();

LoginPage({super.key});

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      decoration: BoxDecoration(
        image: DecorationImage(
          image: NetworkImage(
            'https://i.ytimg.com/vi/tx61kRUNfCE/maxresdefault.jpg',
          ), // Add your background image URL here
          fit: BoxFit.cover,
        ),
      ),
    child: Center(
      child: Padding(
        padding: const EdgeInsets.all(20.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.min,
          children: [
            // Email input field
            TextField(
              controller: emailController,
              decoration: InputDecoration(
                labelText: 'Email',
                labelStyle: TextStyle(color: Colors.white),
                filled: true,
                fillColor: Colors.black45,
                border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(8.0),
                  borderSide: BorderSide.none,
                ),
                hintStyle: TextStyle(color: Colors.white70),
              ),
              style: TextStyle(color: Colors.white),
            ),
            SizedBox(height: 15),

```

```

// Password input field
TextField(
  controller: passwordController,
  obscureText: true,
  decoration: InputDecoration(
    labelText: 'Password',
    labelStyle: TextStyle(color: Colors.white),
    filled: true,
    fillColor: Colors.black45,
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(8.0),
      borderSide: BorderSide.none,
    ),
    hintStyle: TextStyle(color: Colors.white70),
  ),
  style: TextStyle(color: Colors.white),
),
SizedBox(height: 20),

// Login button
ElevatedButton(
  style: ElevatedButton.styleFrom(
    backgroundColor:
      Colors.blue, // Change to your preferred color
    padding: EdgeInsets.symmetric(vertical: 15,
horizontal: 60),
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(8.0),
    ),
  ),
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => LocationListPage(),
      ),
    );
  },
  child: Text('Login', style: TextStyle(fontSize: 18)),
),
SizedBox(height: 15),

// Sign up button

```

```

        TextButton(
          onPressed: () {
            Navigator.push(
              context,
              MaterialPageRoute(builder: (context) =>
SignupPage()),
            );
          },
          child: Text(
            'Sign Up',
            style: TextStyle(color: Colors.white, fontSize:
16),
          ),
        ),
      ],
    ),
  ),
),
);
}
}

```

#### Signup\_page.dart

```

import 'package:flutter/material.dart';
import 'login_page.dart';

class SignupPage extends StatelessWidget {
  final TextEditingController emailController =
TextEditingController();
  final TextEditingController passwordController =
TextEditingController();

  SignupPage({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        decoration: BoxDecoration(
          image: DecorationImage(

```

```

        image: NetworkImage(
          'https://i.ytimg.com/vi/tx61kRUNfCE/maxresdefault.jpg',
        ), // Add your background image URL here
        fit: BoxFit.cover,
      ),
    ),
    child: Center(
      child: Padding(
        padding: const EdgeInsets.all(20.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.min,
          children: [
            // Email input field
            TextField(
              controller: emailController,
              decoration: InputDecoration(
                labelText: 'Email',
                labelStyle: TextStyle(color: Colors.white),
                filled: true,
                fillColor: Colors.black45,
                border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(8.0),
                  borderSide: BorderSide.none,
                ),
                hintStyle: TextStyle(color: Colors.white70),
              ),
              style: TextStyle(color: Colors.white),
            ),
            SizedBox(height: 15),

            // Password input field
            TextField(
              controller: passwordController,
              obscureText: true,
              decoration: InputDecoration(
                labelText: 'Password',
                labelStyle: TextStyle(color: Colors.white),
                filled: true,
                fillColor: Colors.black45,
                border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(8.0),
                  borderSide: BorderSide.none,
                ),
              ),
            ),
          ],
        ),
      ),
    ),
  ),
);

```

```

        hintStyle: TextStyle(color: Colors.white70),
      ),
      style: TextStyle(color: Colors.white),
    ),
    SizedBox(height: 20),

    // Sign Up button
    ElevatedButton(
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.blue, // Customize button
color
        padding: EdgeInsets.symmetric(vertical: 15,
horizontal: 60),
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(8.0),
        ),
      ),
      onPressed: () {
        Navigator.pop(context);
      },
      child: Text('Sign Up', style: TextStyle(fontSize:
18)),
    ),
    SizedBox(height: 15),

    // Login button (Redirect to LoginPage)
    TextButton(
      onPressed: () {
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) =>
LoginPage()),
        );
      },
      child: Text(
        'Already have an account? Login',
        style: TextStyle(color: Colors.white, fontSize:
16),
      ),
    ),
  ],
),
),

```

```

    ),
  ),
);
}
}

```

Location\_list\_page.dart

```

import 'package:flutter/material.dart';
import 'google_map_page.dart';

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

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

class _LocationListPageState extends State<LocationListPage> {
  String? startLocation; // Variable to hold the selected start
location
  String?
  destinationLocation; // Variable to hold the selected destination
location

  final List<Map<String, dynamic>> locations = [
    {'name': 'Main Block', 'image': '', 'icon': Icons.location_on},
    {'name': 'PG Block', 'image': '', 'icon': Icons.school},
    {'name': 'Boys Hostel', 'image': '', 'icon': Icons.boy},
    {'name': 'Girls Hostel', 'image': '', 'icon': Icons.girl},
    {'name': 'Sports Ground', 'image': '', 'icon':
Icons.sports_cricket},
    {'name': 'MITE GYM', 'image': '', 'icon': Icons.fitness_center},
    {'name': 'MITE Food Court', 'image': '', 'icon': Icons.fastfood},
    {'name': 'MITE Mess', 'image': '', 'icon': Icons.restaurant},
    {'name': 'Mechanical Block', 'image': '', 'icon':
Icons.schedule_rounded},
    {'name': 'Parking Area', 'image': '', 'icon': Icons.local_parking},
    {'name': 'MITE Greens', 'image': '', 'icon': Icons.nature},
    {'name': 'MITE Stationary', 'image': '', 'icon': Icons.book},
    {'name': 'MITE Library', 'image': '', 'icon': Icons.library_books},
  ];
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: Text('Select Start and Destination
Locations')),
    body: Stack(
      children: [
        // Network background image
        Positioned.fill(
          child: Image.network(
            'https://i.ytimg.com/vi/tx6lkRUNfCE/maxresdefault.jpg',
// <-- replace with your image URL
            fit: BoxFit.cover,
          ),
        ),

        // Optional dark overlay for readability
        Positioned.fill(
          child: Container(color: Colors.black.withOpacity(0.4)),
        ),

        // Centered content
        Center(
          child: Padding(
            padding: const EdgeInsets.all(16.0),
            child: Column(
              mainAxisAlignment: MainAxisAlignment.min,
              children: [
                if (startLocation == null)
                  ElevatedButton(
                    onPressed: () async {
                      final selectedStartLocation = await
_selectLocation(
                        context,
                      );
                      setState(() {
                        startLocation = selectedStartLocation;
                      });
                    },
                    child: Text('Select Start Location'),
                  ),
                if (startLocation != null)

```



```

        Text(
          'Start Location: $startLocation',
          style: TextStyle(color: Colors.white, fontSize:
16),
        ),

        SizedBox(height: 20),

        if (startLocation != null && destinationLocation ==
null)

          ElevatedButton(
            onPressed: () async {
              final selectedDestinationLocation =
                await _selectLocation(context);
              setState(() {
                destinationLocation =
selectedDestinationLocation;
              });
            },
            child: Text('Select Destination Location'),
          ),

        if (destinationLocation != null)

          Text(
            'Destination Location: $destinationLocation',
            style: TextStyle(color: Colors.white, fontSize:
16),
          ),

          SizedBox(height: 30),

          if (startLocation != null && destinationLocation !=
null)

            ElevatedButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(
                    builder:
                      (context) => GoogleMapPage(
                        startLocation: startLocation!,
                        destination: destinationLocation!,
                      ),
                  ),
            ),

```

```

        );
      },
      child: Text('View Route on Google Map'),
    ),
  ],
),
),
),
),
],
),
);
}

// Function to show the location list and select a location
Future<String?> _selectLocation(BuildContext context) async {
  return showDialog<String>(
    context: context,
    builder: (context) {
      return AlertDialog(
        title: Text('Select a Location'),
        content: SingleChildScrollView(
          child: Column(
            children:
              locations.map((location) {
                return ListTile(
                  title: Text(location['name']!),
                  onTap: () {
                    Navigator.pop(context, location['name']);
                  },
                );
              }).toList(),
          ),
        ),
      );
    },
  );
}
}

```

Googlemap\_page.dart

```

import 'dart:async';
import 'package:flutter/material.dart';

```

```

import 'package:google_maps_flutter/google_maps_flutter.dart';

class GoogleMapPage extends StatefulWidget {
  final String startLocation;
  final String destination;

  const GoogleMapPage({
    super.key,
    required this.startLocation,
    required this.destination,
  });

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

class _GoogleMapPageState extends State<GoogleMapPage> {
  final Completer<GoogleMapController> _controller = Completer();

  Set<Marker> _markers = {};
  Set<Polyline> _polylines = {};
  LatLng? _startLatLng;
  LatLng? _endLatLng;

  final LatLngBounds _miteCampusBounds = LatLngBounds(
    southwest: LatLng(13.0490, 74.9640),
    northeast: LatLng(13.0535, 74.9670),
  );

  final Map<String, LatLng> locations = {
    'MITE Main Entrance': LatLng(13.050689903907054,
74.96671541061446),
    'Main Block': LatLng(13.05098458113866, 74.96515323346254),
    'PG Block': LatLng(13.050110739655757, 74.96527961934655),
    'Sports Ground': LatLng(13.05121193319527, 74.96590314876343),
    'MITE GYM': LatLng(13.053030521485221, 74.96490000261377),
    'MITE Food Court': LatLng(13.049341072670144, 74.96487049830093),
    'Mechanical Block': LatLng(13.050409764290322, 74.964167759548),
    'MITE Greens': LatLng(13.051619502278417, 74.96472976054392),
    'MITE Stationary': LatLng(13.050594494848506, 74.96494928145448),
    'MITE Library': LatLng(13.051313475325378, 74.96492278755149),
    'Ganapati Temple': LatLng(13.050469133935866, 74.96573652885785),
  };
}

```

```

@override
void initState() {
  super.initState();
  _initializeRouteFromInputs();
  _loadMarkers();
}

void _initializeRouteFromInputs() {
  final start = locations[widget.startLocation];
  final end = locations[widget.destination];

  if (start != null && end != null) {
    _startLatLng = start;
    _endLatLng = end;
    _drawRoute();
  }
}

void _loadMarkers() {
  _markers = locations.entries.map((entry) {
    return Marker(
      markerId: MarkerId(entry.key),
      position: entry.value,
      infoWindow: InfoWindow(title: entry.key),
      onTap: () => _handleMarkerTap(entry.key, entry.value),
    );
  }).toSet();
}

void _handleMarkerTap(String name, LatLng tappedLatLng) {
  setState(() {
    if (_startLatLng == null || (_startLatLng != null && _endLatLng
!= null)) {
      _startLatLng = tappedLatLng;
      _endLatLng = null;
      _polylines.clear();
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Start location selected: $name')),
      );
    } else if (_endLatLng == null && tappedLatLng != _startLatLng) {
      _endLatLng = tappedLatLng;
      _drawRoute();
    }
  });
}

```

```

        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(content: Text('Destination selected: $name')),
        );
      }
    });
  }

void _drawRoute() {
  if (_startLatLng != null && _endLatLng != null) {
    final Polyline polyline = Polyline(
      polylineId: const PolylineId('route_line'),
      color: Colors.blueAccent,
      width: 5,
      patterns: [PatternItem.dot],
      points: [_startLatLng!, _endLatLng!],
    );

    setState(() {
      _polylines = {polyline};
    });
  }
}

void _clearRoute() {
  setState(() {
    _startLatLng = null;
    _endLatLng = null;
    _polylines.clear();
  });

  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(content: Text('Route cleared!')),
  );
}

@override
Widget build(BuildContext context) {
  String routeTitle;

  if (_startLatLng != null && _endLatLng != null) {
    routeTitle = 'Route from ${widget.startLocation} to
    ${widget.destination}';
  } else if (_startLatLng != null) {
    routeTitle = 'Select destination';
  }
}

```

```

    } else {
      routeTitle = 'Select start location';
    }

    return Scaffold(
      appBar: AppBar(
        title: Text(routeTitle),
      ),
      body: GoogleMap(
        initialCameraPosition: const CameraPosition(
          target: LatLng(13.0513, 74.9657),
          zoom: 18,
        ),
        markers: _markers,
        polylines: _polylines,
        mapType: MapType.satellite,
        myLocationEnabled: true,
        myLocationButtonEnabled: true,
        zoomControlsEnabled: true,
        minMaxZoomPreference: MinMaxZoomPreference(17.5, 20),
        cameraTargetBounds: CameraTargetBounds(_miteCampusBounds),
        onMapCreated: (controller) {
          _controller.complete(controller);
        },
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _clearRoute,
        tooltip: 'Clear Route',
        child: const Icon(Icons.clear),
      ),
    );
  }
}

```

## RESULTS AND SNAPSHOTS

### User Authentication:

- The app integrates **Firestore Authentication** to allow users to sign up and log in securely using email and password. The authentication process is smooth and error-handled, ensuring protected access and preventing invalid or duplicate entries.

### Location Selection:

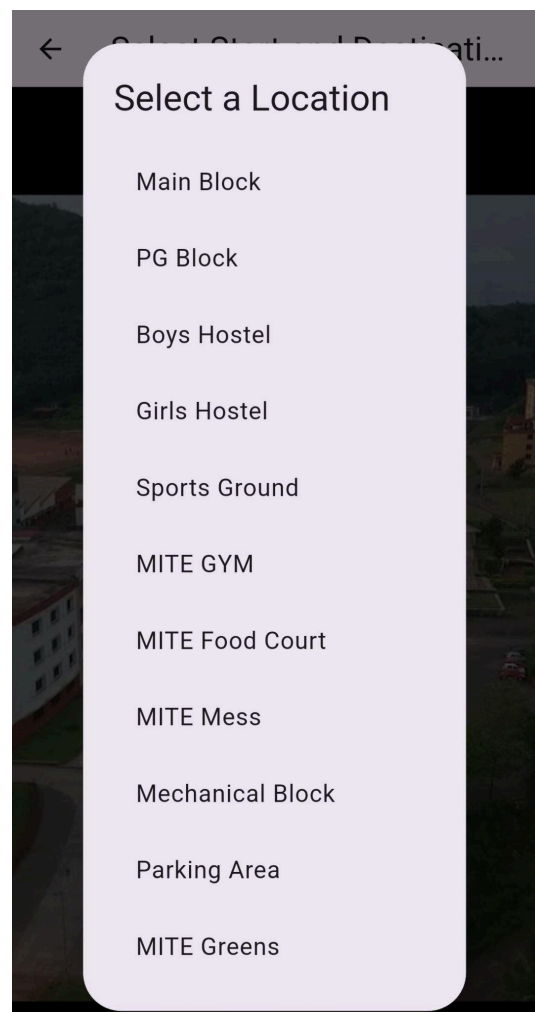
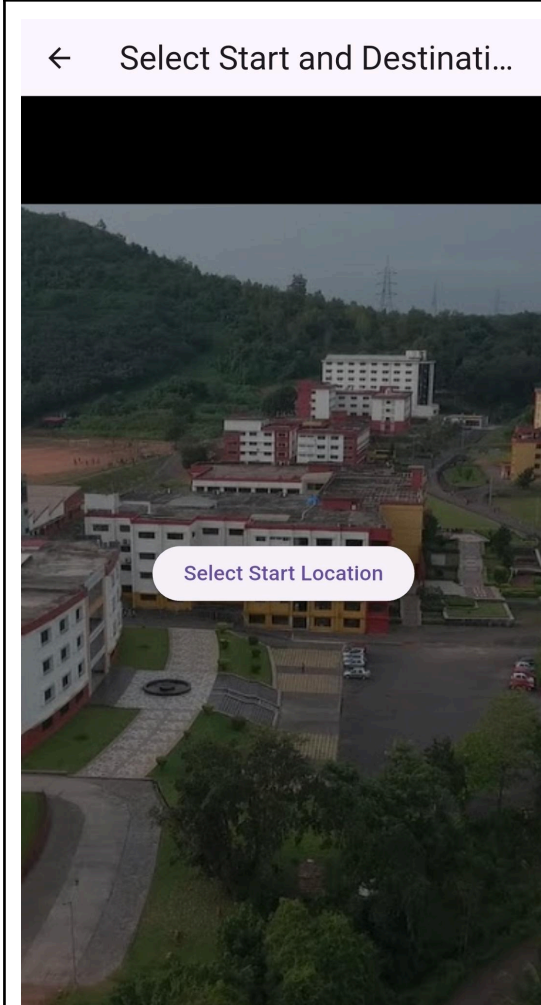
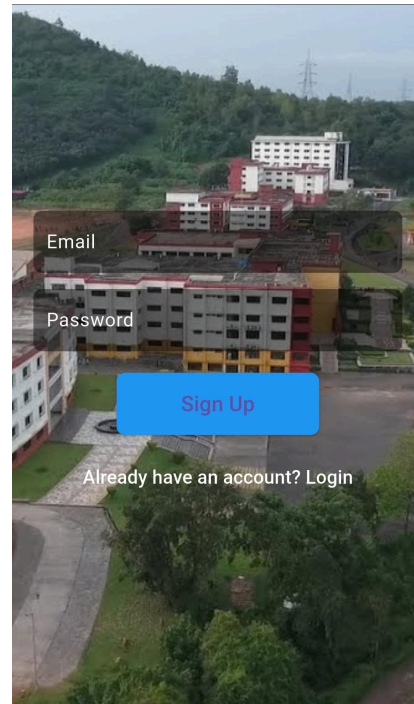
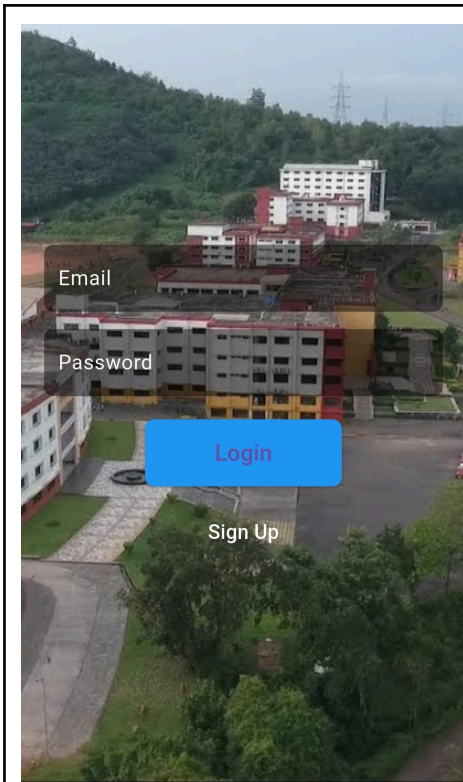
- The **Location List Screen** provides two dropdown menus for selecting the starting point and destination. These are populated with campus locations and designed to be intuitive, helping users—especially newcomers—navigate without confusion.

### Map Loading:

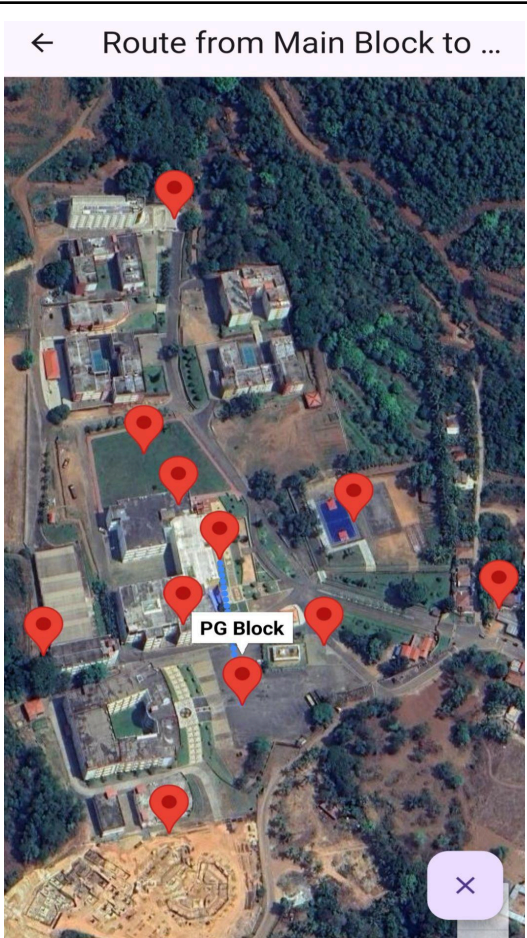
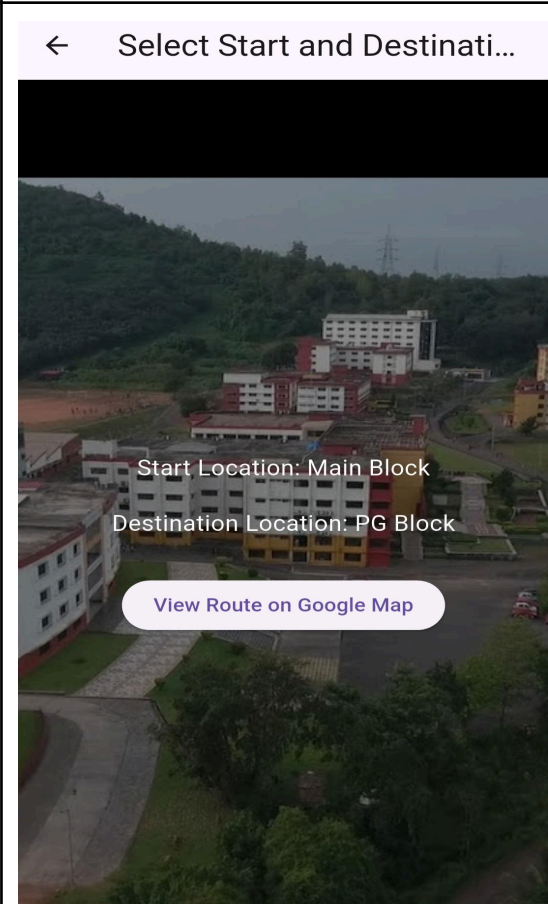
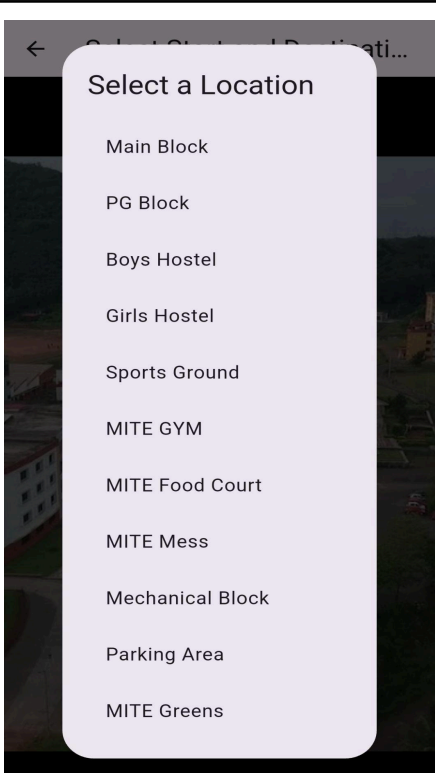
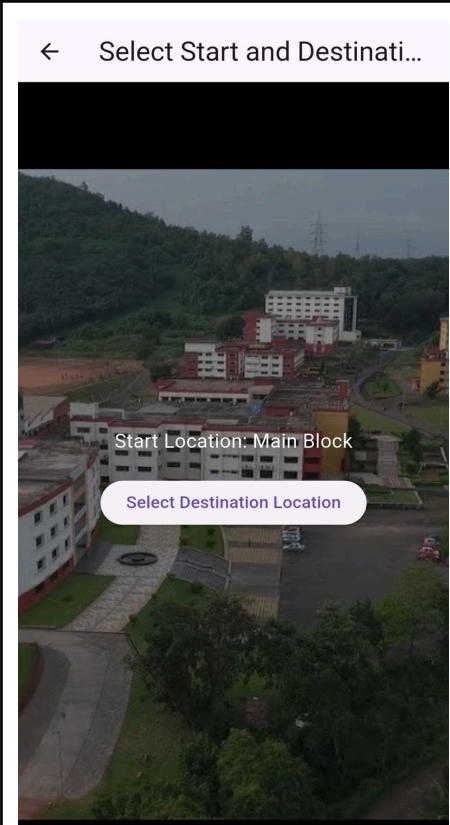
- The **Google Maps widget** loads correctly with optimal zoom focused on the MITE campus. It displays interactive markers for the selected locations, and the map supports pan and zoom gestures for ease of use.

### Route Display:

- When both points are selected, a route is displayed on the map using **blue polylines**. The polyline updates in real time and provides a clear visual guide from start to destination, enhancing spatial understanding and reducing navigation time.







## CONCLUSION AND FUTURE WORK

### 7.1 Conclusion

The **Smart Campus Navigator** project effectively addresses the challenges of navigating large educational campuses such as MITE. By leveraging Flutter for cross-platform development, Google Maps API for real-time route visualization, and Firebase for secure authentication, the app delivers a streamlined and reliable navigation experience.

Through features like interactive location selection, map-based route display, and a user-friendly interface, the application significantly reduces confusion, especially for new students, visitors, and participants during campus events. The system ensures that users can reach their desired destinations with minimal effort and maximum efficiency, contributing to better orientation and time management on campus.

Overall, the project demonstrates how modern mobile technologies and APIs can be effectively integrated to solve real-world problems in educational environments.

### 7.2 Future Work

While the current version of the Smart Campus Navigator meets the core navigation needs, several enhancements can be considered for future development:

- **3D Campus Building Models:**  
Integrating 3D models of campus buildings can provide a more immersive and spatially accurate experience, helping users better understand building structures and interiors.
- **Voice-Based Navigation:**  
Implementing voice-guided navigation will make the app more accessible and convenient, especially while walking or for users with visual challenges.
- **Offline Mode Support:**  
By caching map data and routes, users can navigate the campus even without an active internet connection—ideal for areas with poor connectivity.
- **Integration with Campus Events and Notices:**  
The app could display real-time updates about campus events, fests, and notices directly on the map or home screen, making it a central information hub for students

## REFERENCES

1. **Flutter Documentation**

<https://flutter.dev>

Official documentation for Flutter, used for developing the cross-platform mobile application.

2. **Google Maps API**

<https://developers.google.com/maps>

Provides tools and documentation for integrating maps, markers, and routing features.

3. **Firebase Authentication**

<https://firebase.google.com/docs/auth>

Used for secure user login and authentication in the app.

4. **IEEE Research Paper – Indoor Navigation Using Smartphones**

Available on IEEE Xplore

This paper provided insights into indoor and localized navigation approaches using mobile technologies.

5. **Android Developers Guide**

<https://developer.android.com>

Reference material for mobile app development and best practices in Android environments.