

GeoTrackPro – Bus Tracking App

GeoTrackPro is a real-time bus tracking app built using **React Native (Expo)** and **Firebase Realtime Database**, made specifically for schools and colleges. It allows parents or students to view the live location of the bus, get personalized ETAs, receive alarms for nearby stops or delays, and confirm boarding status — all dynamically based on each user.

Features:

1. **Real-Time Bus Tracking:** Shows a single bus's live location on the map.
2. **User Stop Selection:** Each user selects their own stop from a list.
3. **Dynamic ETA:** Calculates ETA based on the selected stop.
4. **Alarms:** Notifies users when the bus is near their stop or delayed.
5. **Parent Confirmation:** Lets parents confirm if their child boarded or missed the bus.
6. **Firebase Integration:** Stores stop locations and user selections in Firebase Realtime Database.

Screens in the App:

1. **Home Screen** – Starting point of the app.
2. **Stop Selection Screen** – User enters their name and picks their stop.
3. **Map Screen** – Displays:
 - Bus location
 - Selected stop
 - ETA
 - Alarms
 - Bus Status
4. **Parent Confirmation Prompt** – Lets parents confirm boarding status.

Tech Stack:

1. **Frontend:** React Native (using Expo)
2. **Backend:** Firebase Realtime Database

Additional Packages Used:

1. react-native-maps – for map display
2. expo-location – for accessing GPS/location
3. expo-av – for sound alerts
4. react-native core modules

Setup & Installation

1. Clone the Repository

```
git clone https://github.com/deekshitha-ganji/GeoTrackPro-Bus-Tracking-App.git
cd GeoTrackPro-Bus-Tracking-App
```

2. Install Dependencies

```
npm install
```

You'll need the following packages:

- react-native
- react-native-maps
- expo-av
- expo-location
- firebase

If you're using Expo Go for Android/Web:

```
npx expo install react-native-maps expo-av expo-location
npm install firebase
```

3. Set Up Firebase

1. Go to Firebase Console.
2. Create a **new Firebase project for Android App**.
3. Enable **Realtime Database** and set the rules to public for testing:

```
{
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

4. Go to **Project Settings** → **Web App**, and register a web app.
5. Copy the **Firebase config** and place it inside a file like:

```
// utils/firebaseConfig.js
```

```
export const firebaseConfig = {
  apiKey: "YOUR_API_KEY",
  authDomain: "your-project.firebaseio.com",
  databaseURL: "https://your-project.firebaseio.com",
  projectId: "your-project-id",
  storageBucket: "your-project.appspot.com",
```

```
messagingSenderId: "XXXXXXX",
appId: "YOUR_APP_ID",
measurementId: "G-XXXXXXXXXX"
};
```

Firestore Realtime Database Structure

```
{
  "stops": {
    "stop_no_1": {
      "lat": ...,
      "lng": ...
    },
    "stop_no_2": {
      "lat": ...,
      "lng": ...
    },
    ...
    "stop_no_n": {
      "lat": ...,
      "lng": ...
    }
  },
  "users": {
    "student1_name": {
      "lat": ...,
      "lng": ...,
      "stop_name": "...",
      "stop_no": "stop_no_1"
    },
    "student2_name": {
      "lat": ...,
      "lng": ...,
      "stop_name": "...",
      "stop_no": "stop_no_2"
    },
    ...
    "studentn_name": {
      "lat": ...,
      "lng": ...,
      "stop_name": "...",
      "stop_no": "stop_no_n"
    }
  }
}
```

Explanation:

- **stops:** Contains all predefined bus stop coordinates labeled by stop number.
- **users:** Each student/user entry includes:
 - Their selected stop's coordinates (lat, lng)
 - The name of the stop (stop_name)
 - The reference to the stop number (stop_no) they chose from the list

4. Run the App (Android + Web)

For Android (using Expo Go):

`npx expo start`

- Scan the QR code using the **Expo Go** app on your Android phone.
- The app will launch on your phone.

App Flow

1. **HomeScreen**
→ Welcome screen
2. **UserSelectionScreen**
→ Users enter their **name**.
→ Based on the name, their **assigned stop** is automatically selected from the Firebase database.
3. **MapScreen**
→ Displays the following in real time:
 - **Live bus location**
 - **ETA to your selected stop**
 - **Alarm when bus approaches the stop**
 - **Parent confirmation prompt** to confirm if the student boarded the bus or not

Project Structure

```
GeoTrack/  
├── App.js  
├── screens/  
│   ├── HomeScreen.js  
│   ├── UserSelectionScreen.js  
│   └── MapScreen.js
```

```
|— utils/  
|— firebaseConfig.js
```

Notes

- App supports **a single bus** and **multiple users** with personalized views.
- Alerts and ETA are based on each user's selected stop.
- Firebase handles stop data and user selections.

Future Enhancements

- Multi-bus support
- SMS notifications
- Admin dashboard
- Authentication (parent, driver, admin roles)