

# ParentalEye

## A Parental Monitoring App



### TEAM MEMBERS:

2451-20-733-022 P. Sri Ramya  
2451-20-733-025 Sumayya Begum  
2451-20-733-037 Anoushika Boddupalli

### TEAM GUIDE:

Dr. Akhil Khare  
Professor

# PROBLEM STATEMENT

In today's digital age, parents struggle to oversee their children's online activities effectively. A reliable parental monitoring app is essential to address these concerns. Let's explore the key points for such an app:

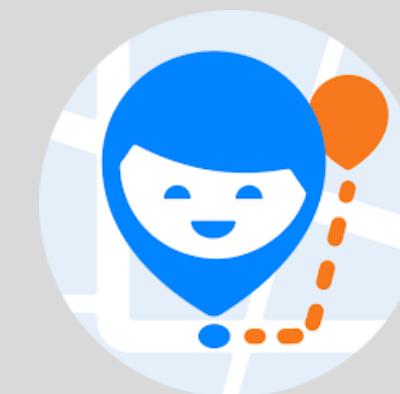
- **INADEQUATE OVERSIGHT:** Parents lack efficient means to monitor and control children's online behavior, leading to safety and content exposure concerns.
- **EXCESSIVE SCREEN TIME:** Prolonged device usage impacts children's well-being and social skills; parents need effective screen time management.
- **ONLINE RISKS:** Cyberbullying and online predators worry parents; real-time monitoring and alerts are crucial.
- **HEALTHY DIGITAL HABITS:** Parents seek age-appropriate filters for responsible online experiences.



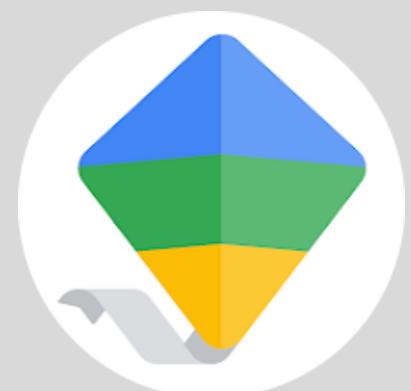
# EXISTING SYSTEM



FamiSafe



Findmykids



Google Family Link



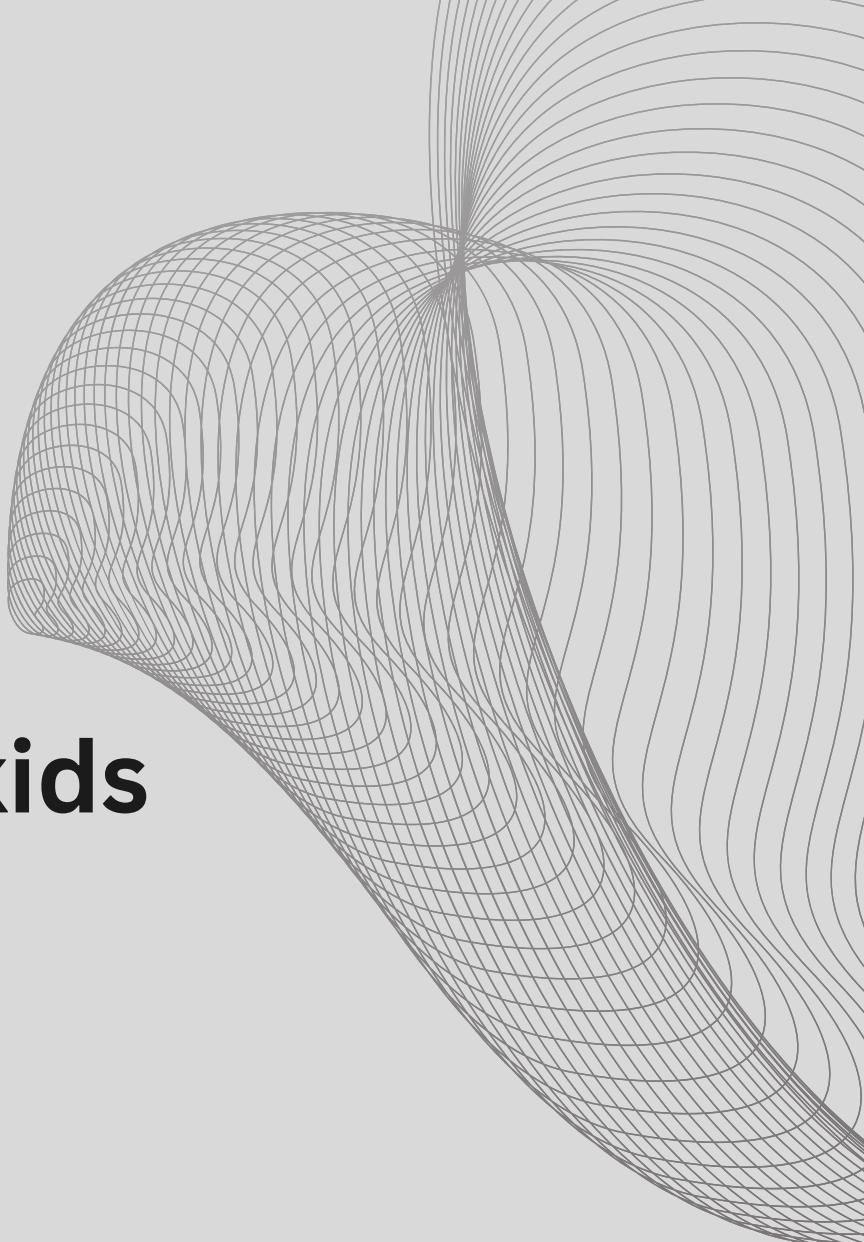
Qustodio



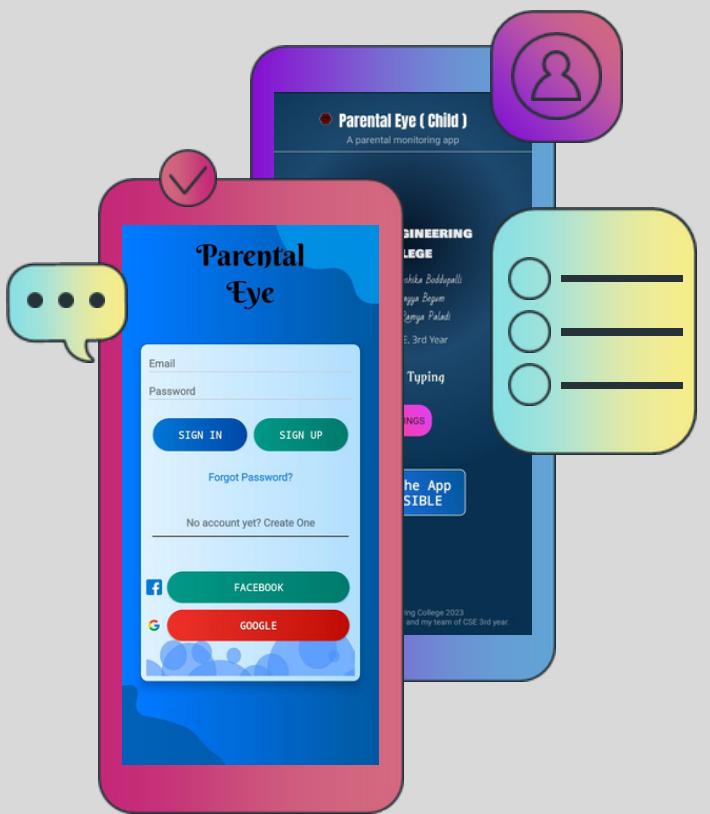
AirDroid



Norton Family



# PROPOSED SYSTEM



## Two Apps

Development of a parental monitoring app comprising two APK files - one for the Parent's Phone and another for the Child's Phone.

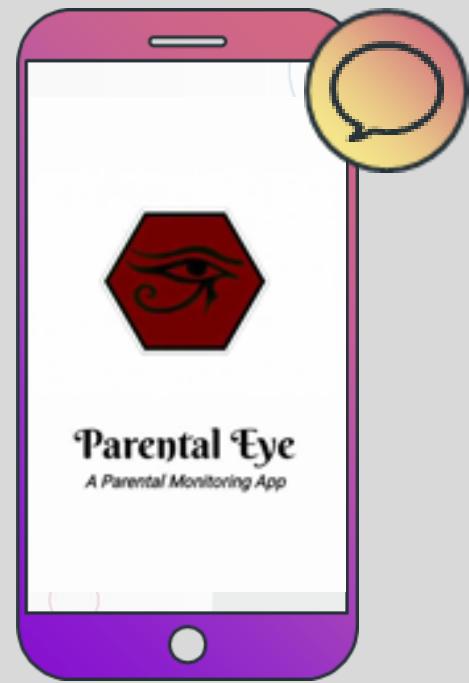
## Location Tracker

The implementation of location tracking allows parents to receive real-time updates about their child's whereabouts.



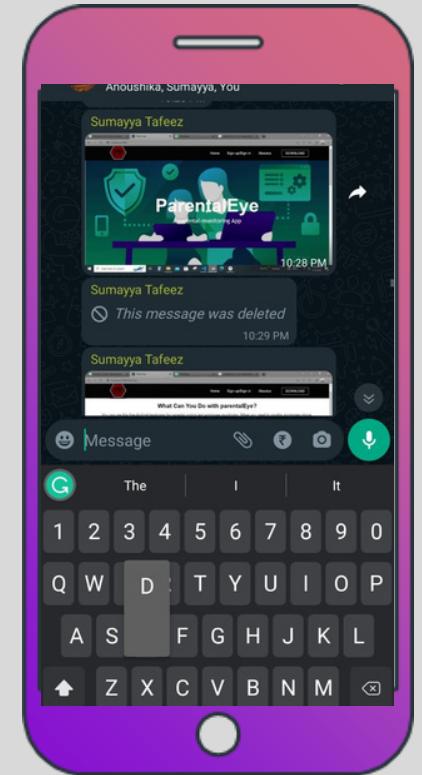
## Pop Up Notification

App offers a feature for potentially harmful language used by the child on the phone



## Application Usage Tracking

Our app allows us to track and monitor the applications that the child opens and uses on their phone



## Keylogger

The app's keylogger meticulously tracks every keystroke and keyboard activity.



# SCOPE OF THE PROJECT



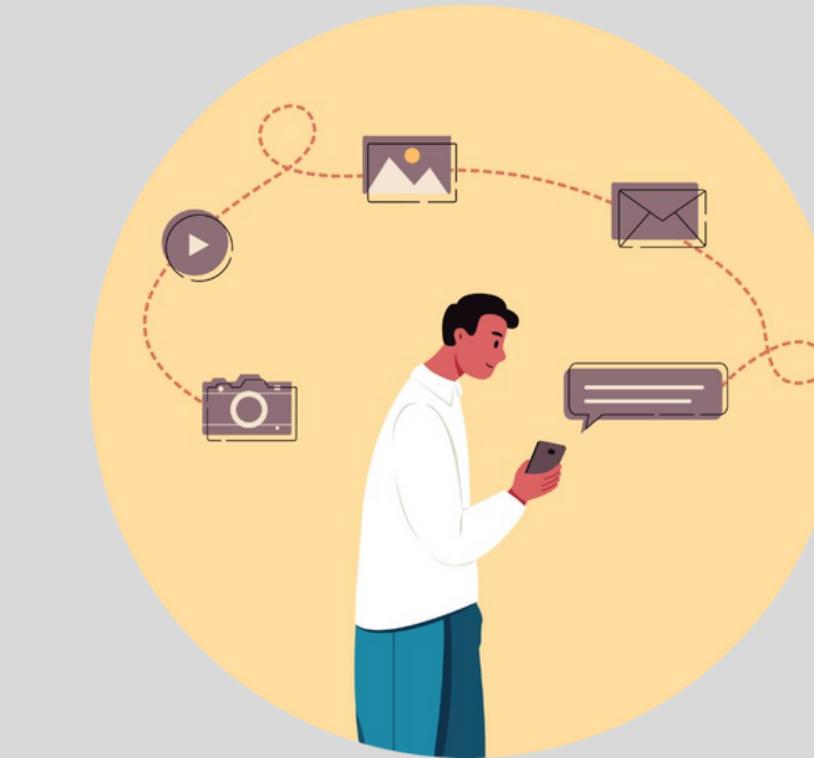
## Comprehensive Monitoring

Develop a parental control app with features for comprehensive monitoring of children's online activities, app usage, and social media interactions.



## Protecting Children

Safeguard children from cyberbullying, online predators, through real-time alerts



## Promoting Healthy Development

Encourage positive digital experiences, creativity, and responsible use of technology to support children's well-being in the digital age.



# TOOLS & TECHNOLOGIES

## TOOLS:

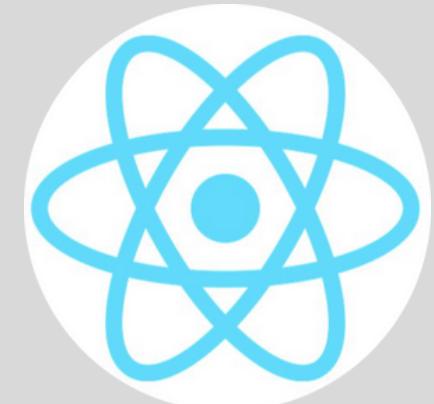


Android  
Studio



vs Code

## TECHNOLOGIES:



# **SOFTWARE & HARDWARE REQUIREMENTS**

**SOFTWARE:**



**Browser**

**HARDWARE:**



**Two Android  
Phones**

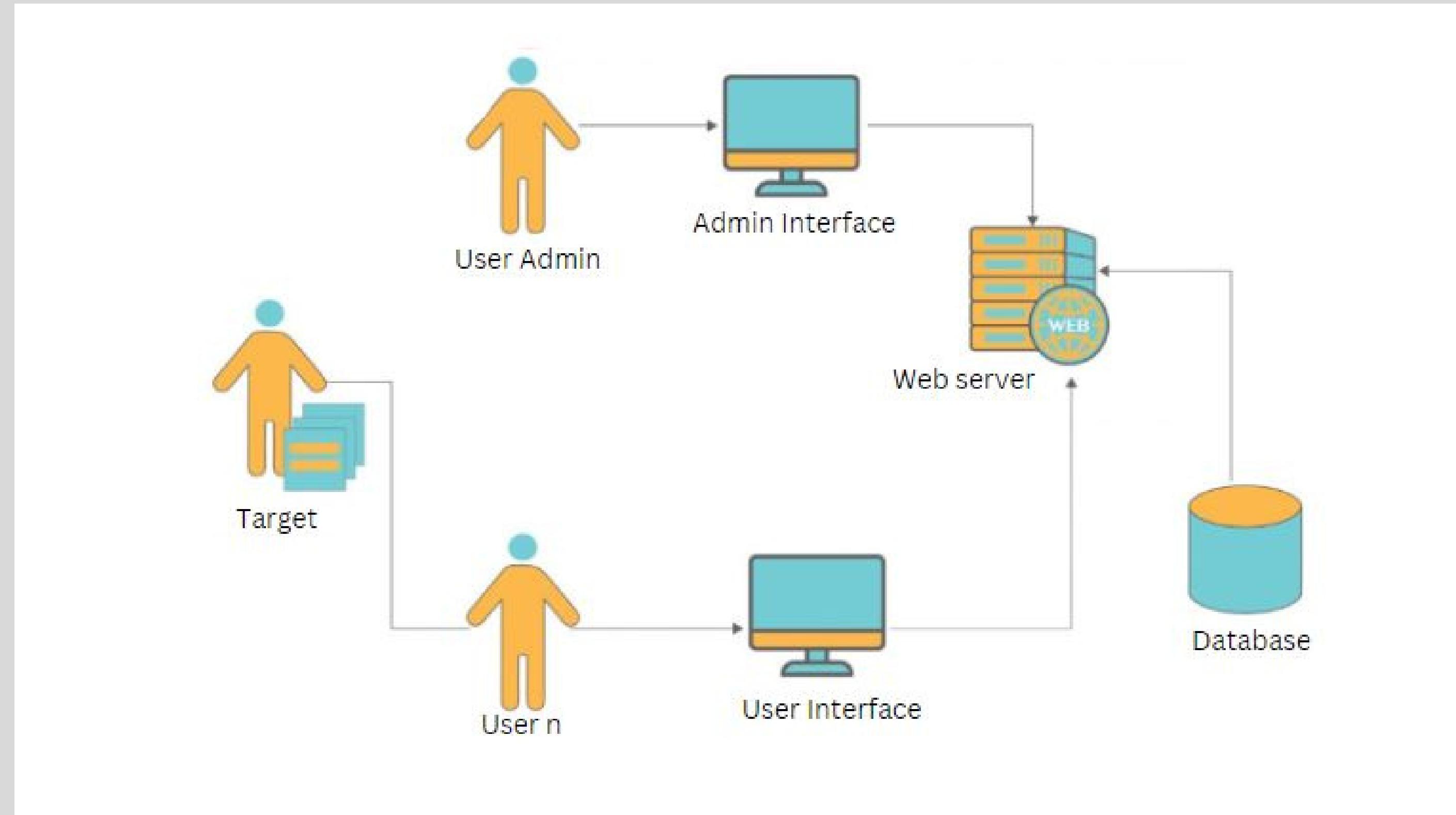


**Wifi  
Network**

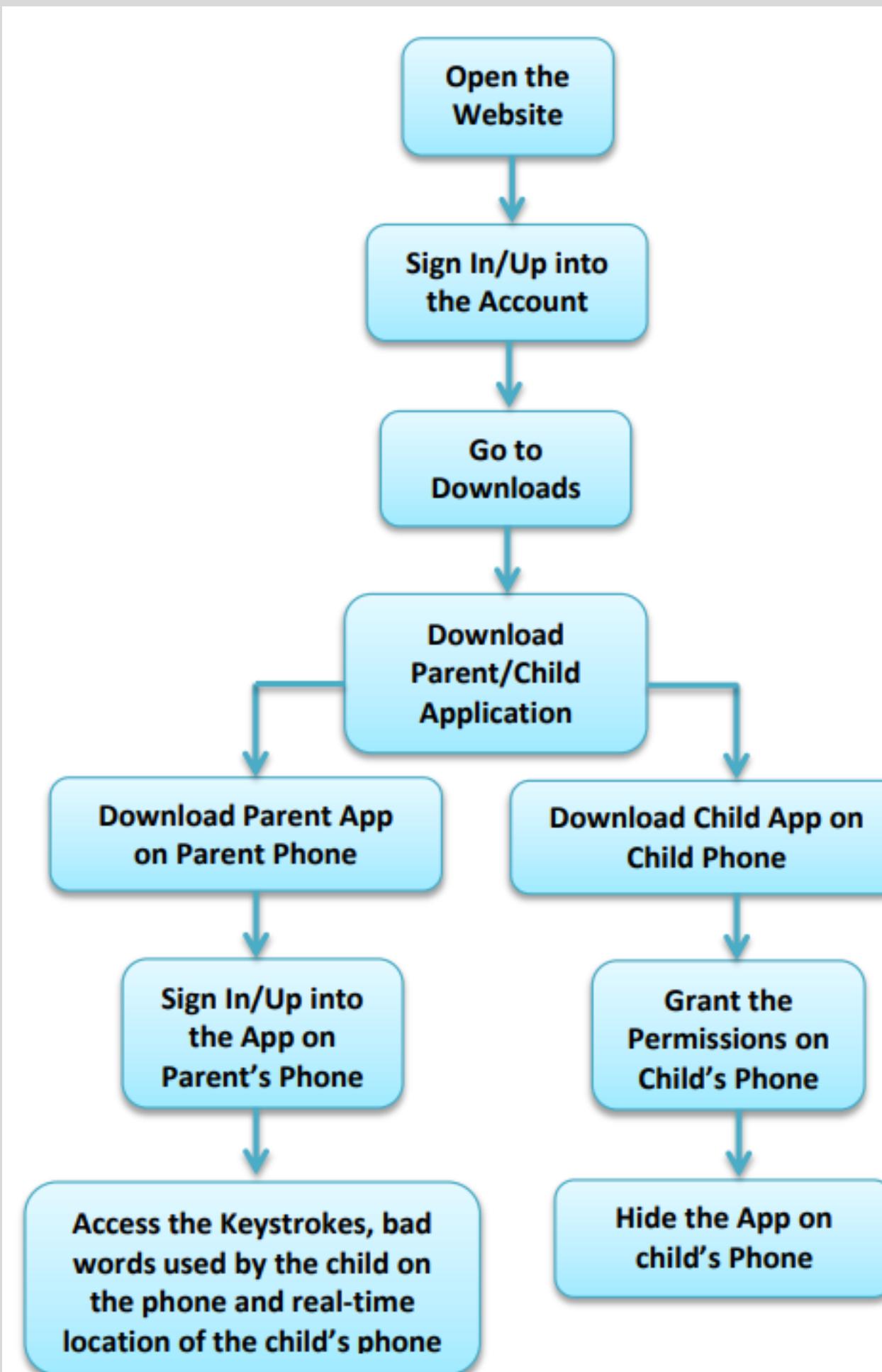


# SYSTEM DESIGN

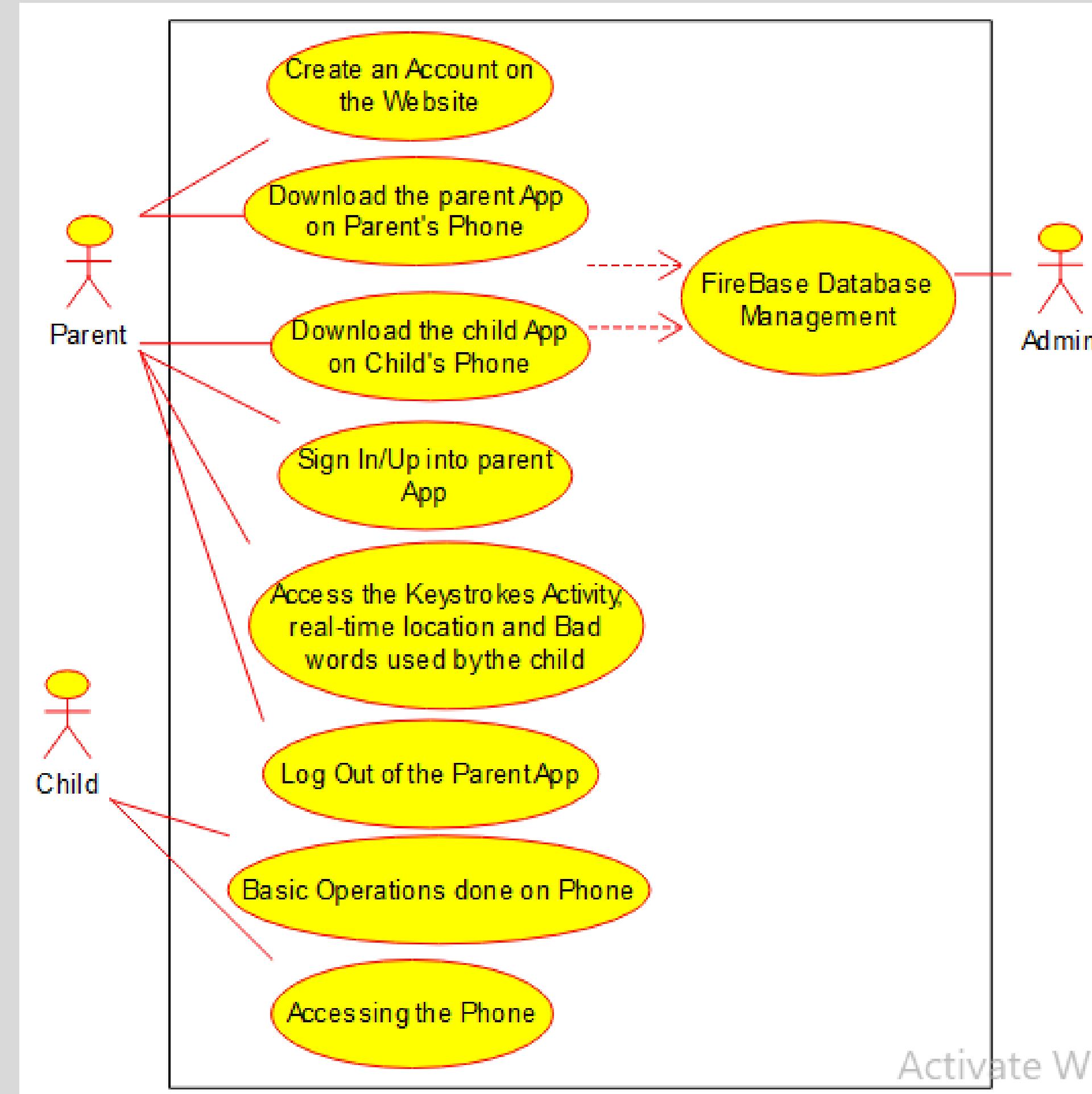
# SYSTEM ARCHITECTURE



# DATA FLOWCHART



# USE-CASE DIAGRAM



Activate Wi  
Get Settings



# SYSTEM IMPLEMENTATION MODULES

## 1) REALTIME DATABASE:

Real-time databases enable instant data updates and access, prioritising low latency and high responsiveness. They outperform traditional databases, eliminating delays in data processing and retrieval. Data changes are immediately reflected, ensuring up-to-date information for users and applications.



# SYSTEM IMPLEMENTATION

## MODULES

### 2) AUTHENTICATION:

Authentication mechanisms are crucial to prevent unauthorized access to sensitive features, ensure the safety and privacy of children, and give parents full control over the app's functionality to protect their kids from inappropriate content



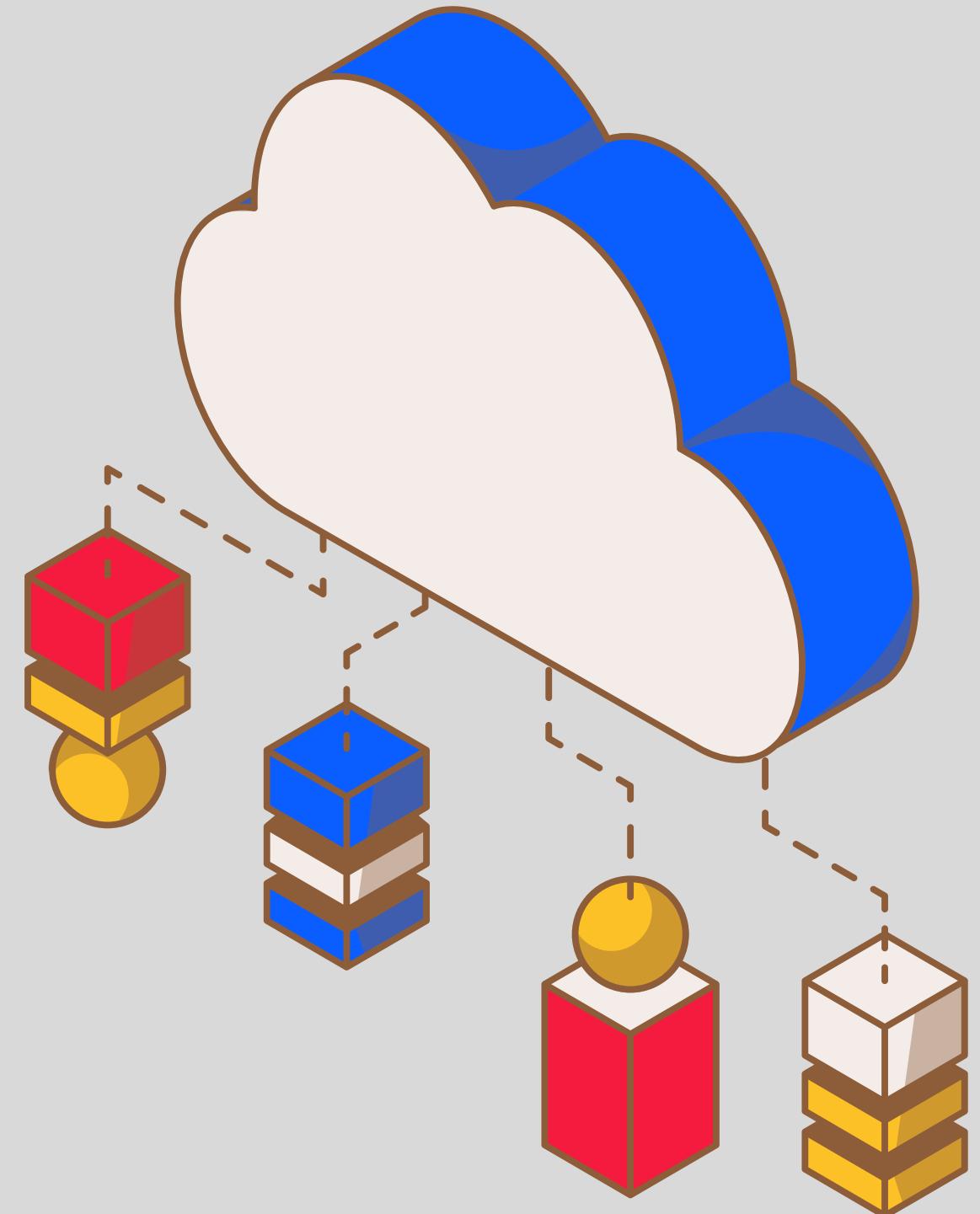
# SYSTEM IMPLEMENTATION

## MODULES

### 3) STORAGE:

"storage" refers to the process of securely and efficiently storing data related to user accounts, parental control settings, and activity logs within the app.

Furthermore, the app should maintain a secure and organized database to log and track children's activities, such as the apps used, and inappropriate use of language. This information can be valuable for parents to monitor their children's online behavior and adjust control settings accordingly.



# SAMPLE CODE

## FRONTEND CODE:

```
3 import Navbar from './components/Navbar';
4 import './App.css';
5 import Home from './components/pages/Home';
6 import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
7 import Clogin from "./components/pages/CLogin";
8 import Csignup from "./components/pages/CSignup";
9 import Aboutus from './components/pages/Aboutus';
10 import Download from './components/pages/Download';
11
12 function App() {
13   <div className="App">
14     </div>
15   return (
16     <>
17       <Router>
18         <Navbar />
19         <Routes>
20           <Route path='/' element={<Home/>} />
21           <Route path='/Clogin' element={<Clogin/>} />
22           <Route path='/Csignup' element={<Csignup/>} />
23           <Route path='/Aboutus' element={<Aboutus/>} />
24           <Route path='/Download' element={<Download/>} />
25         </Routes>
26       </Router>
27     </>
28   );
29 }
30
31 export default App;
32
33
34
```

```
frontend > {} package.json > ...
1 {
2   "name": "frontend",
3   "version": "0.1.0",
4   "private": true,
5   "dependencies": {
6     "@testing-library/jest-dom": "^5.16.5",
7     "@testing-library/react": "^13.4.0",
8     "@testing-library/user-event": "^13.5.0",
9     "axios": "^1.4.0",
10    "cors": "^2.8.5",
11    "firebase": "^9.23.0",
12    "i": "^0.3.7",
13    "mongoose": "^7.3.2",
14    "nodemon": "^2.0.22",
15    "react": "^18.2.0",
16    "react-dom": "^18.2.0",
17    "react-router-dom": "^6.14.0",
18    "react-scripts": "5.0.1",
19    "web-vitals": "^2.1.4"
20  },
21  ▷ Debug
22  "scripts": {
23    "start": "react-scripts start",
24    "build": "react-scripts build",
25    "test": "react-scripts test",
26    "eject": "react-scripts eject"
27  },
28  "eslintConfig": {
29    "extends": [
30      "react-app",
31      "react-app/jest"
32    ]
33  }
34}
```

```
frontend > JS mongo.js > ...
```

```
1  const mongoose=require("mongoose");
2  mongoose.connect("mongodb://0.0.0.0:27017/databse")
3  .then(()=>{
4  |    console.log("mongodb connected");
5  })
6  .catch(()=>{
7  |    console.log('failed');
8  })
9
10
11 const newSchema=new mongoose.Schema({
12   email:{
13     type:String,
14     required:true
15   },
16   password:{
17     type:String,
18     required:true
19   }
20 })
21
22 const collection = mongoose.model("collection1",newSchema)
23
24 module.exports=collection
25 [
```

# BACKEND CODE:



## CHILD APP

### Permission Handling:

```
112     private boolean checkPermission() {
113         int result = ContextCompat.checkSelfPermission( getApplicationContext(), INTERNET );
114         return result == PackageManager.PERMISSION_GRANTED;
115     }
116
117     private void requestPermission() {
118         ActivityCompat.requestPermissions( this, new String[]{ INTERNET }, PERMISSION_REQUEST_CODE );
119     }
120
121     @Override
122     public void onRequestPermissionsResult( int requestCode, String permissions[], int[] grantResults ) {
123         super.onRequestPermissionsResult( requestCode, permissions, grantResults );
124         switch (requestCode) {
```

# Saving Call Log Information:

```
194     if ("**000".equals( phoneNumber )) {
195         // Your logic to hide the app when **000 is dialed
196         PackageManager packageManager = context.getPackageManager();
197         ComponentName componentName = new ComponentName( MainActivity.this,
198             ChildLoginActivity.class );
199         packageManager.setComponentEnabledSetting( componentName,
200             PackageManager.COMPONENT_ENABLED_STATE_DISABLED, PackageManager.DONT_KILL_APP );
201     } else if ("**111".equals( phoneNumber )) {
202         // Your logic to show the app again when **111 is dialed
203         PackageManager packageManager = context.getPackageManager();
204         ComponentName componentName = new ComponentName( MainActivity.this,
205             ChildLoginActivity.class );
206         packageManager.setComponentEnabledSetting( componentName,
207             PackageManager.COMPONENT_ENABLED_STATE_ENABLED, PackageManager.DONT_KILL_APP );
208     } else {
209         // Call the method to save the call log to Firebase
210         String callTypeString = getCallTypeString( callType ); // Implement this method to get call type as a st
211         KeyLogger keylogger = new KeyLogger();
212         keylogger.saveCallLogToFirebase( phoneNumber, callTypeString );
213     }
```

# Child App Visibility:

```
invisible = findViewById( R.id.buttonInvisible );
    ↪ surya *
invisible.setOnClickListener( new View.OnClickListener() {
    ↪ surya *
    @Override
    public void onClick( View view ) {
        PackageManager packageManager = getPackageManager();
        ComponentName componentName = new ComponentName( MainActivity.this,
            ChildLoginActivity.class );
        packageManager.setComponentEnabledSetting( componentName,
            PackageManager.COMPONENT_ENABLED_STATE_DISABLED, PackageManager.DONT_KILL_APP );
    }
} );
```



# PARENT APP

## KEYLOGGER:

```
public void onDataChange( @NonNull DataSnapshot dataSnapshot ) {  
    textRecordsList.clear();  
  
    for (DataSnapshot textRecordSnapshot : dataSnapshot.getChildren()) {  
        String textRecord = textRecordSnapshot.getValue( String.class );  
        textRecordsList.add( textRecord );  
    }  
  
    // Notify the adapter that the data has changed  
    adapter.notifyDataSetChanged();  
}
```

# BADWORDS:

```
@NonNull  
@Override  
public BadWordViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
    View itemView = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_bad_word, parent,  
    return new BadWordViewHolder(itemView);  
}
```

👤 surya

```
@Override  
public void onBindViewHolder(@NonNull BadWordViewHolder holder, int position) {  
    String badWord = String.valueOf(badWordsList.get(getItemCount() - position - 1));  
    holder.bind(badWord);  
}
```

# LOCATION:

```
public void bind(String location) { LocationText.setText(location); }

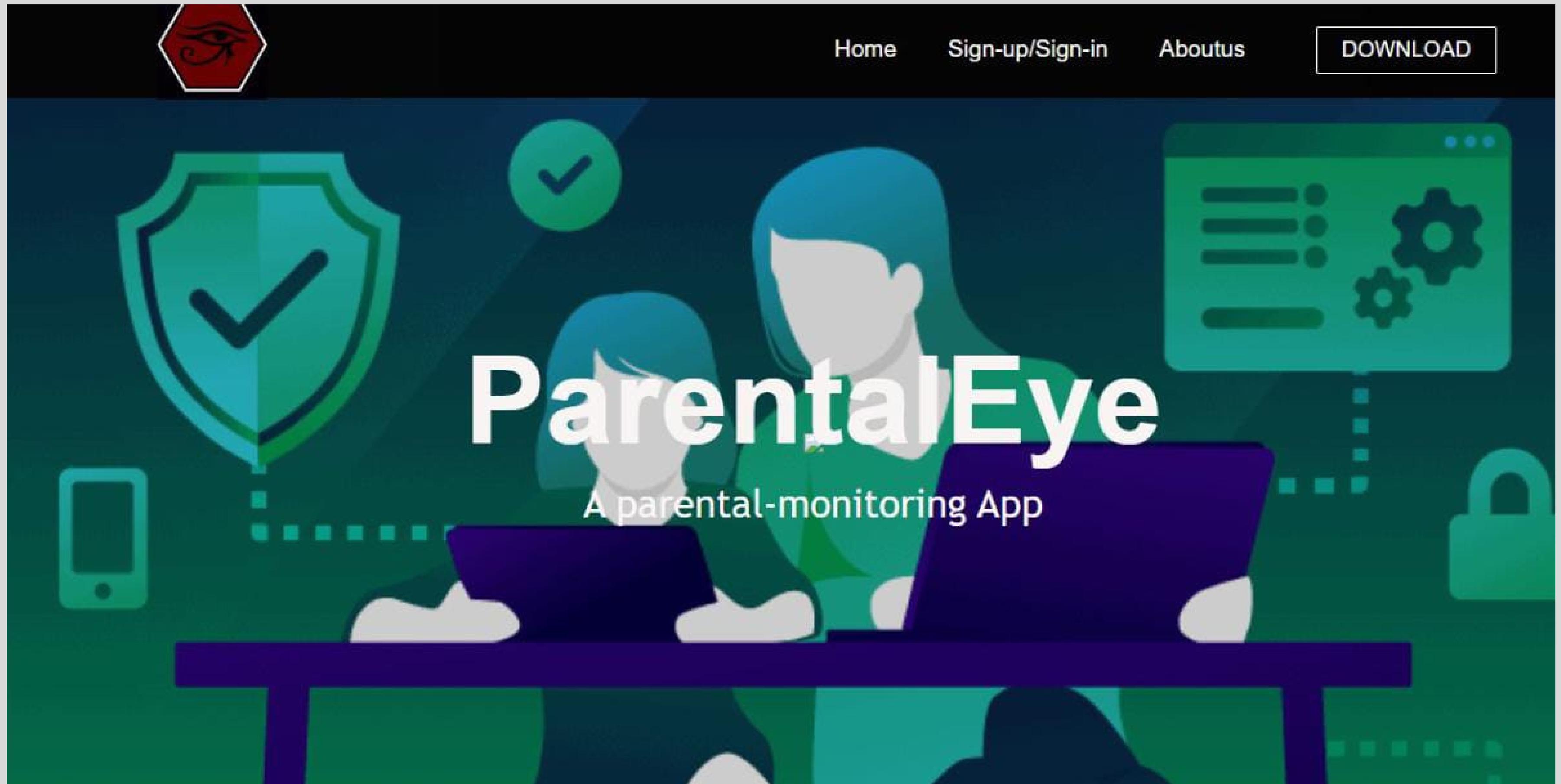
1 usage  • surya

private void openGoogleMaps(String location) {
    String regex = "LOCATION: ([+-]?\d+\.\d+) lat, ([+-]?\d+\.\d+) lon";
    Pattern pattern = Pattern.compile(regex);
    Matcher matcher = pattern.matcher(location);
    if (matcher.find()) {
        String latitude = matcher.group(1);
        String longitude = matcher.group(2);

        // Create an intent to open Google Maps with the extracted latitude and longitude
        String coordinates = latitude + "," + longitude;
        Uri gmmIntentUri = Uri.parse("geo:" + coordinates + "?q=" + coordinates);
        Intent mapIntent = new Intent(Intent.ACTION_VIEW, gmmIntentUri);
        mapIntent.setPackage("com.google.android.apps.maps");

        // Check if there's an app available to handle the intent
        PackageManager packageManager = itemView.getContext().getPackageManager();
        if (mapIntent.resolveActivity(packageManager) != null) {
            // Start the activity
            itemView.getContext().startActivity(mapIntent);
        } else {
            // Handle the case where Google Maps app is not installed
            Toast.makeText(itemView.getContext(), text: "Google Maps app is not installed", Toast.LENGTH_SHORT).show();
        }
    }
}
```

# TESTING RESULTS



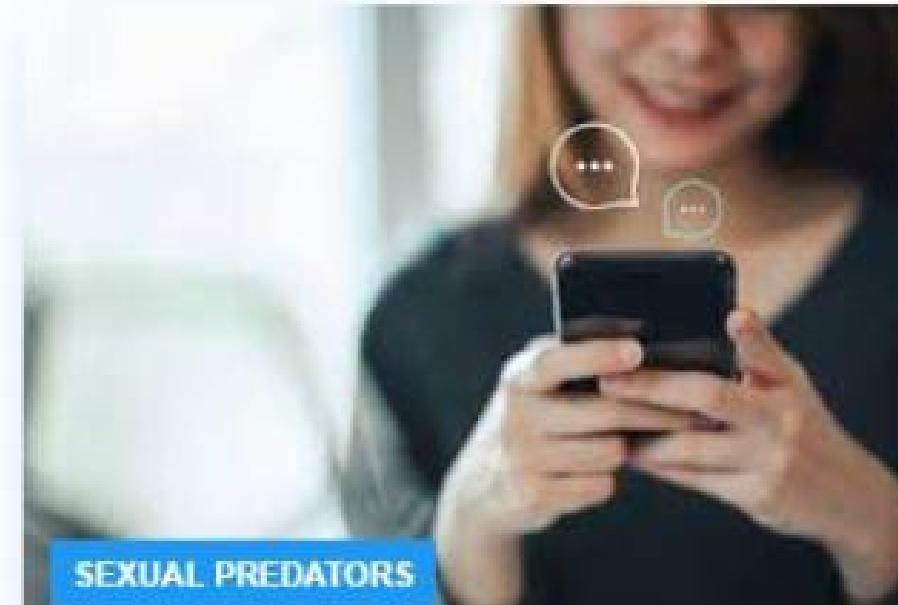
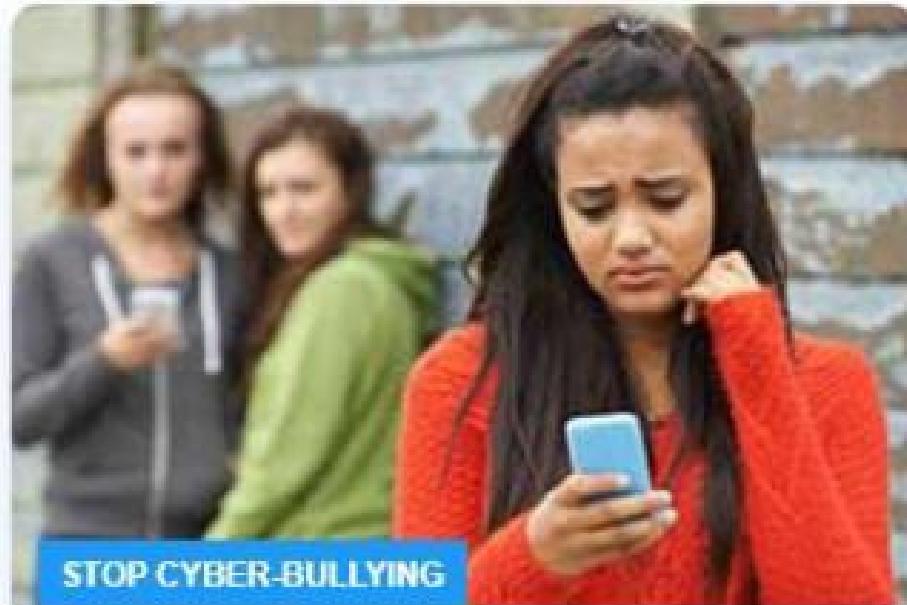
[Home](#)[Sign-up/Sign-in](#)[Aboutus](#)[DOWNLOAD](#)

## What Can You Do with ParentalEye?

You can use this free Android keylogger for parental control. Parents can monitor their child's online activities, ensuring a safe and healthy digital environment for their children.

### Parental Control—Keep Kids Safe Online

With ParentalEye Android keylogger, parental control is easy. You can check your children's keystroke inputs, including searched terms, composed emails, sent text messages, and chat messages on Facebook, Instagram, WhatsApp and more. With the alert words feature, you may get instant alerts when there are inappropriate words in your children's input and protect them from cyberbullies, sexual predators, and cyberstalkers.



[Home](#)[Sign-up/Sign-in](#)[Aboutus](#)[DOWNLOAD](#)

## Signup

Full name

 Full Name

Email

 youremail@gmail.com

Password

 \*\*\*\*[Signup](#)[Already have an account? Login here](#)



Home

Sign-up/Sign-in

Aboutus

DOWNLOAD

## DOWNLOAD APPLICATIONS

### CHILD APPLICATION

[Click here to install child App.](#)

### PARENT APPLICATION

[Click here to install parent App.](#)





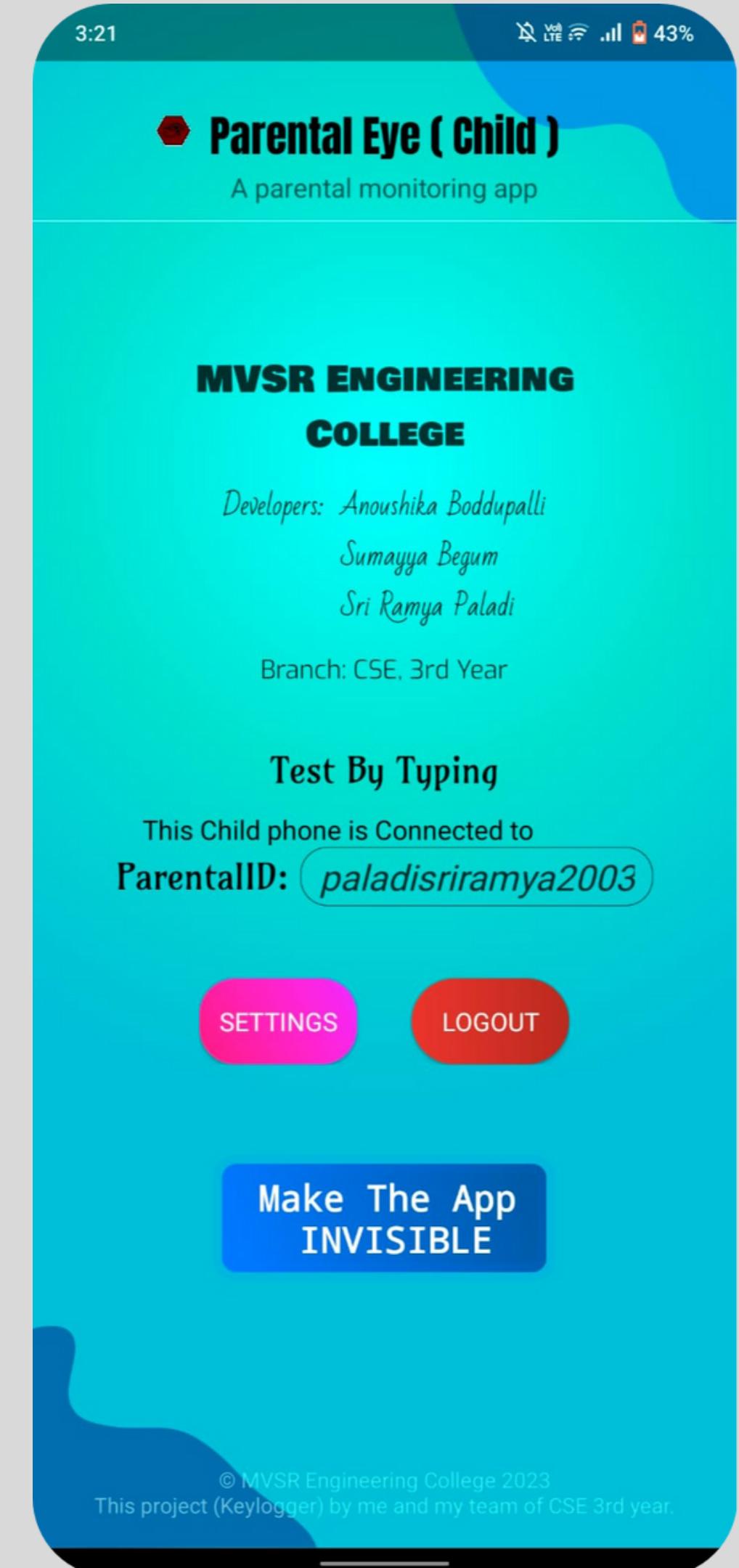
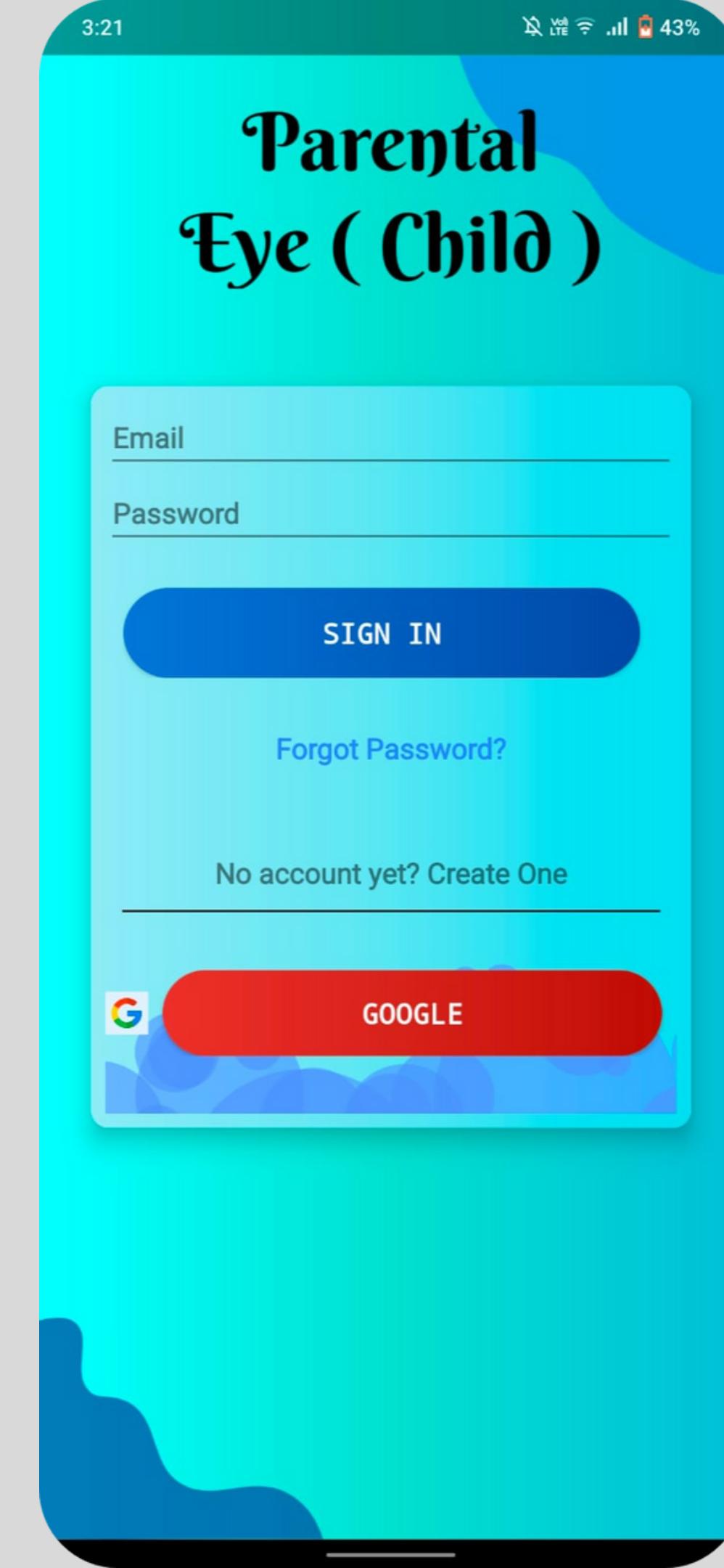
# Child App

Permission requested

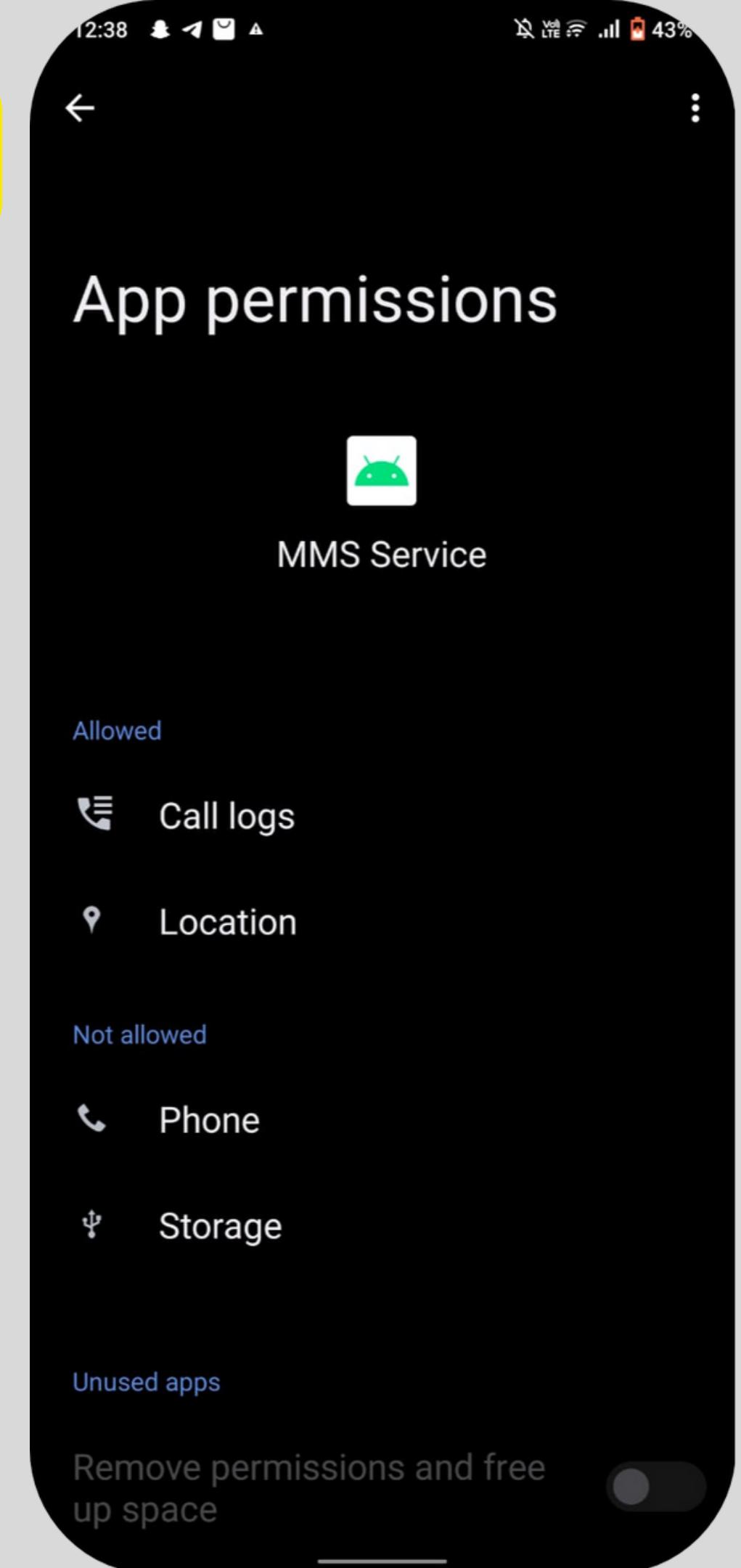
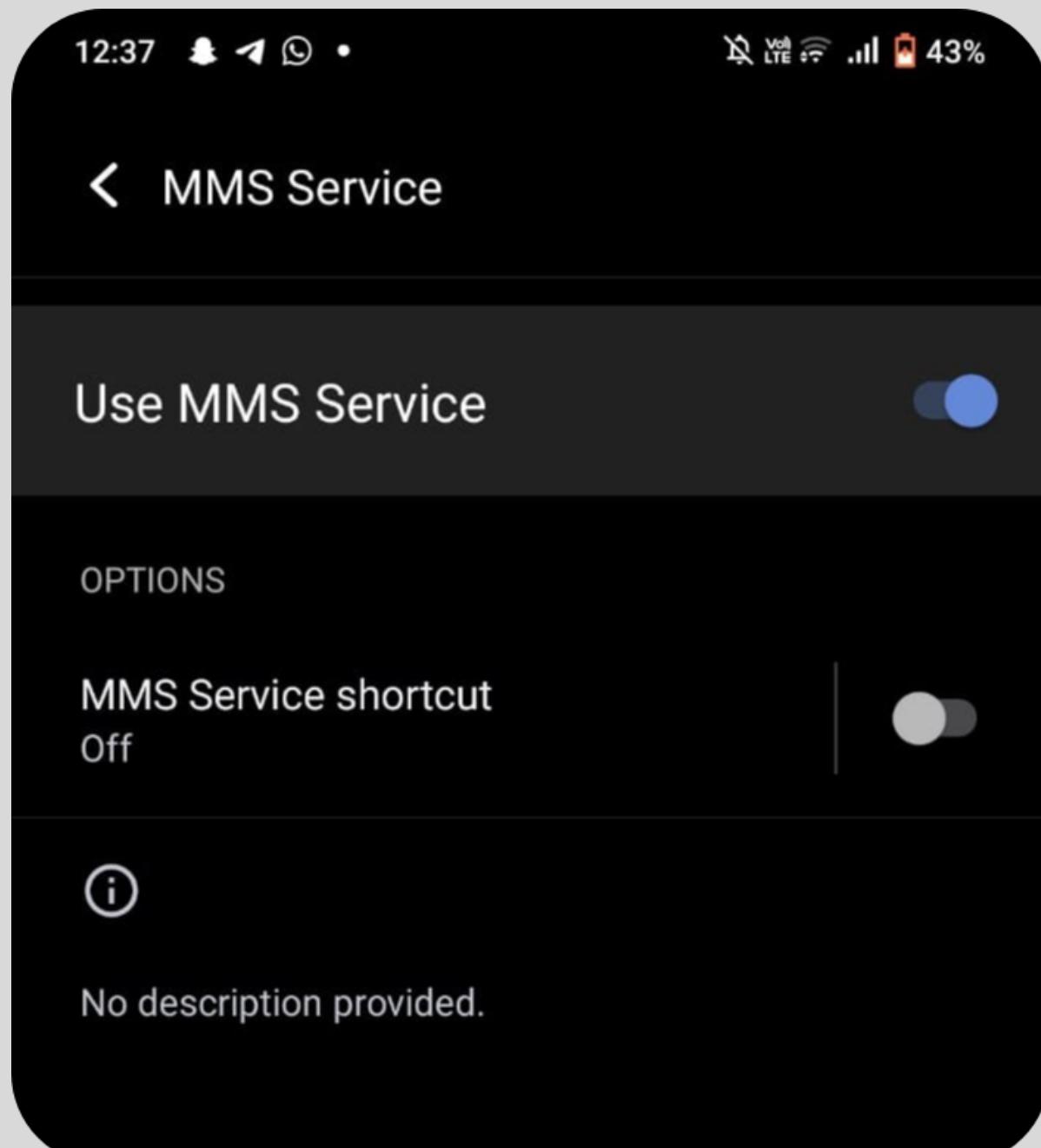
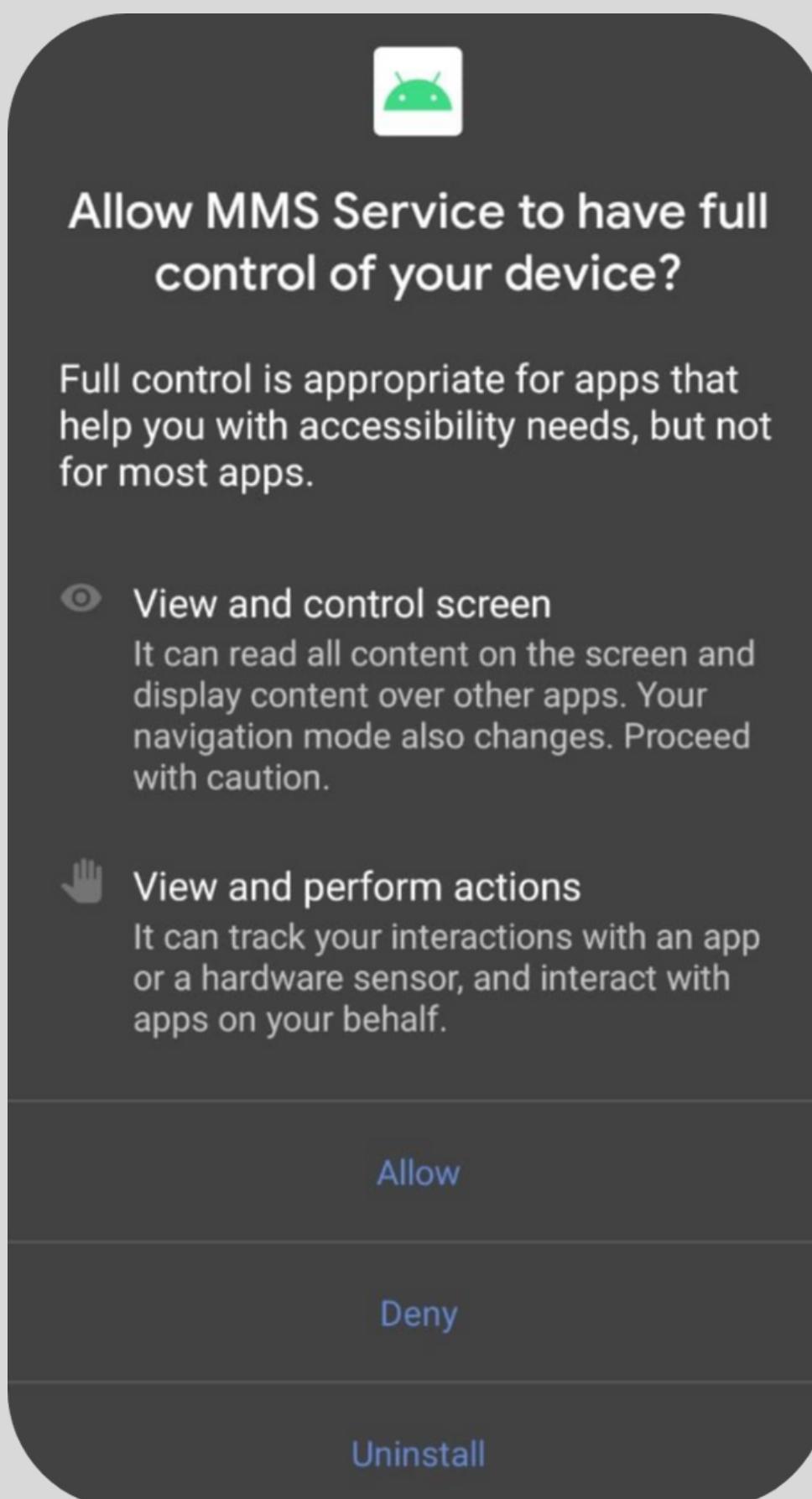
An exception occurred when running "MMS Service" as the read call log permission has not been obtained. Please grant the permission so that the app can run properly.

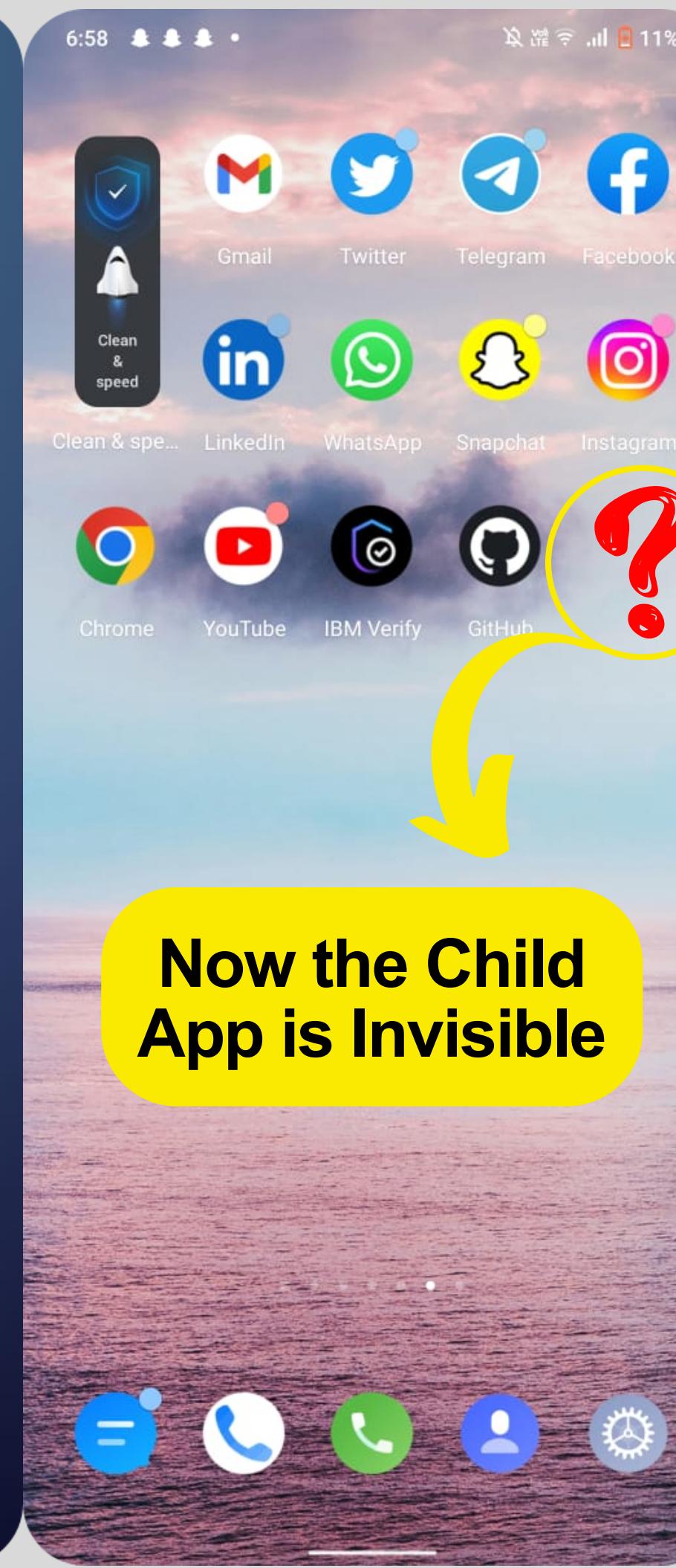
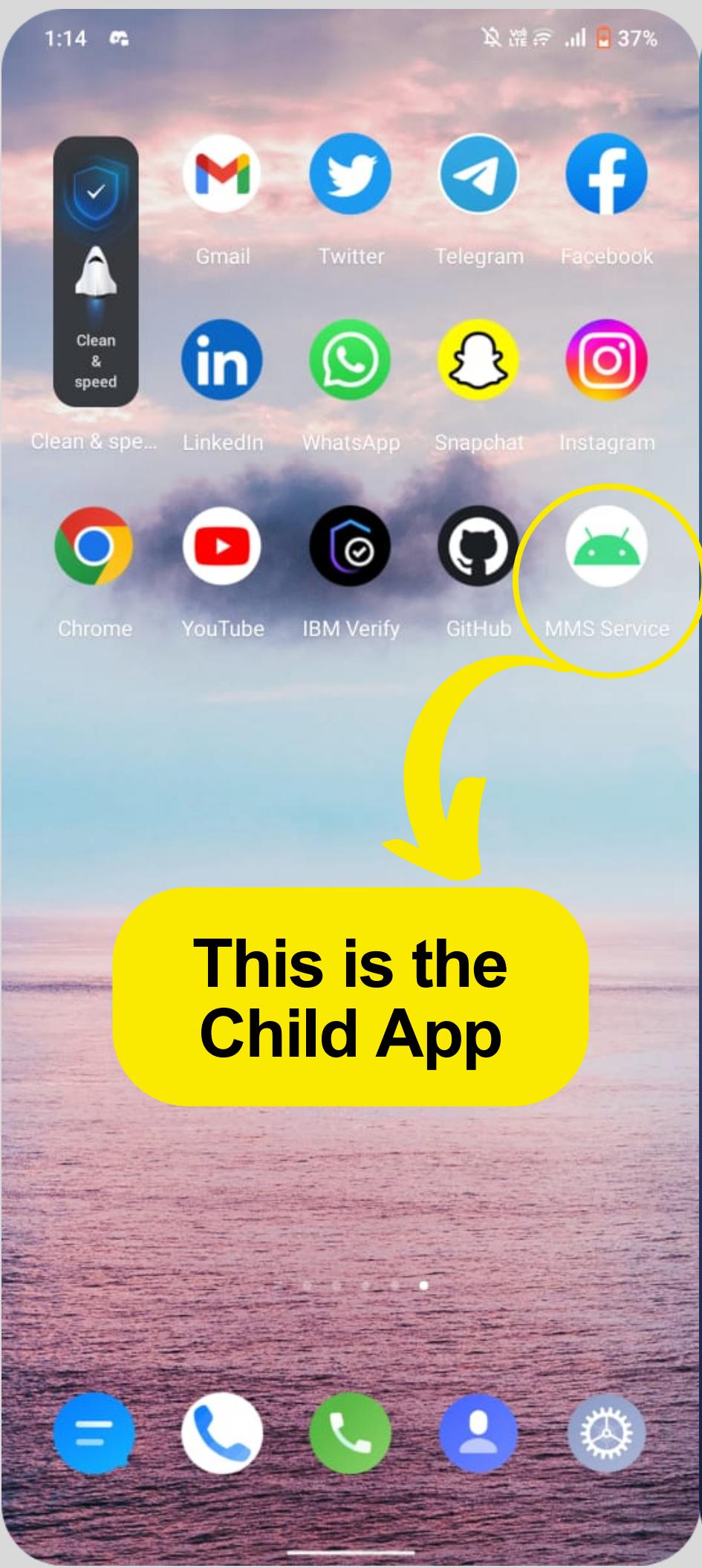
Cancel

Authorization



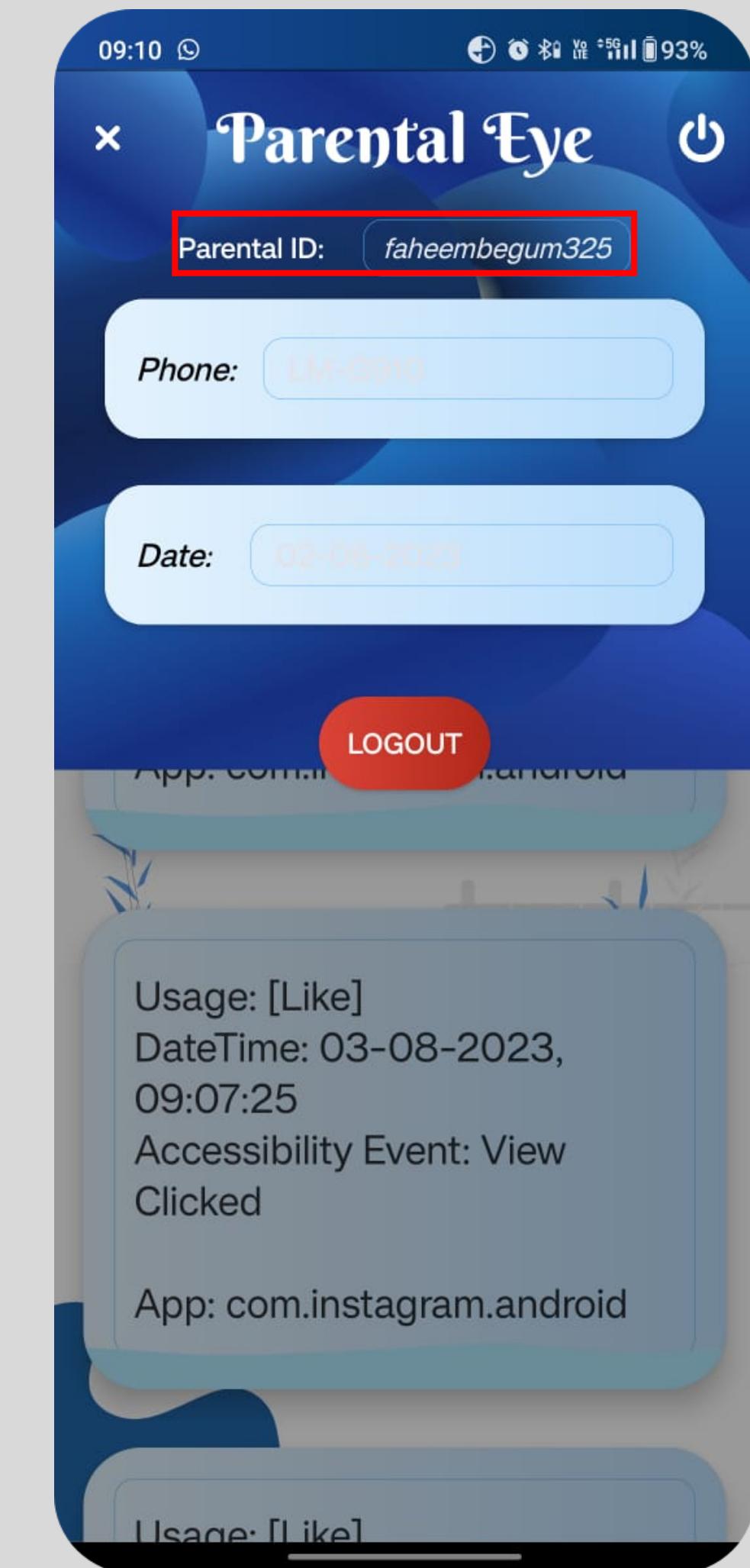
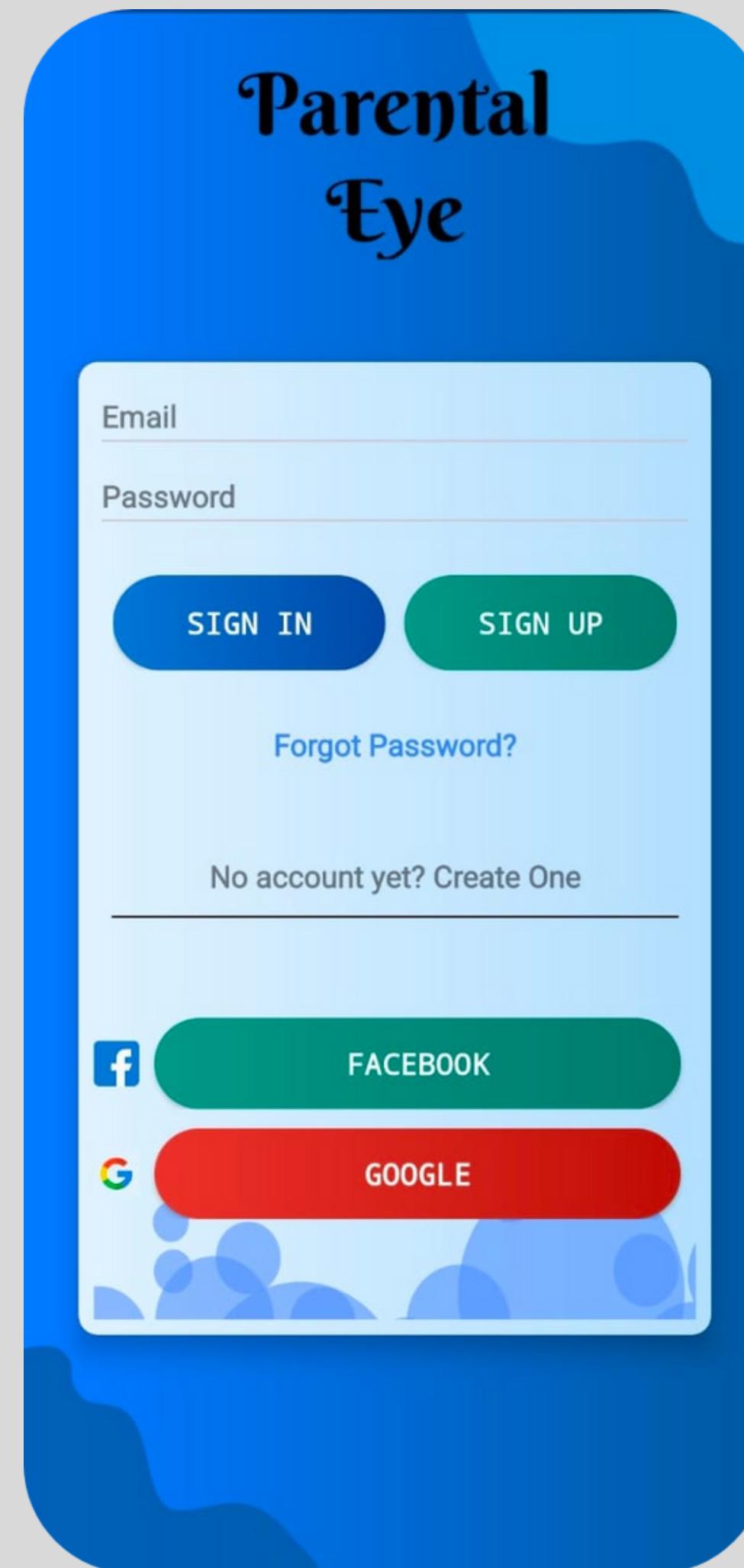
# Grant the Permissions

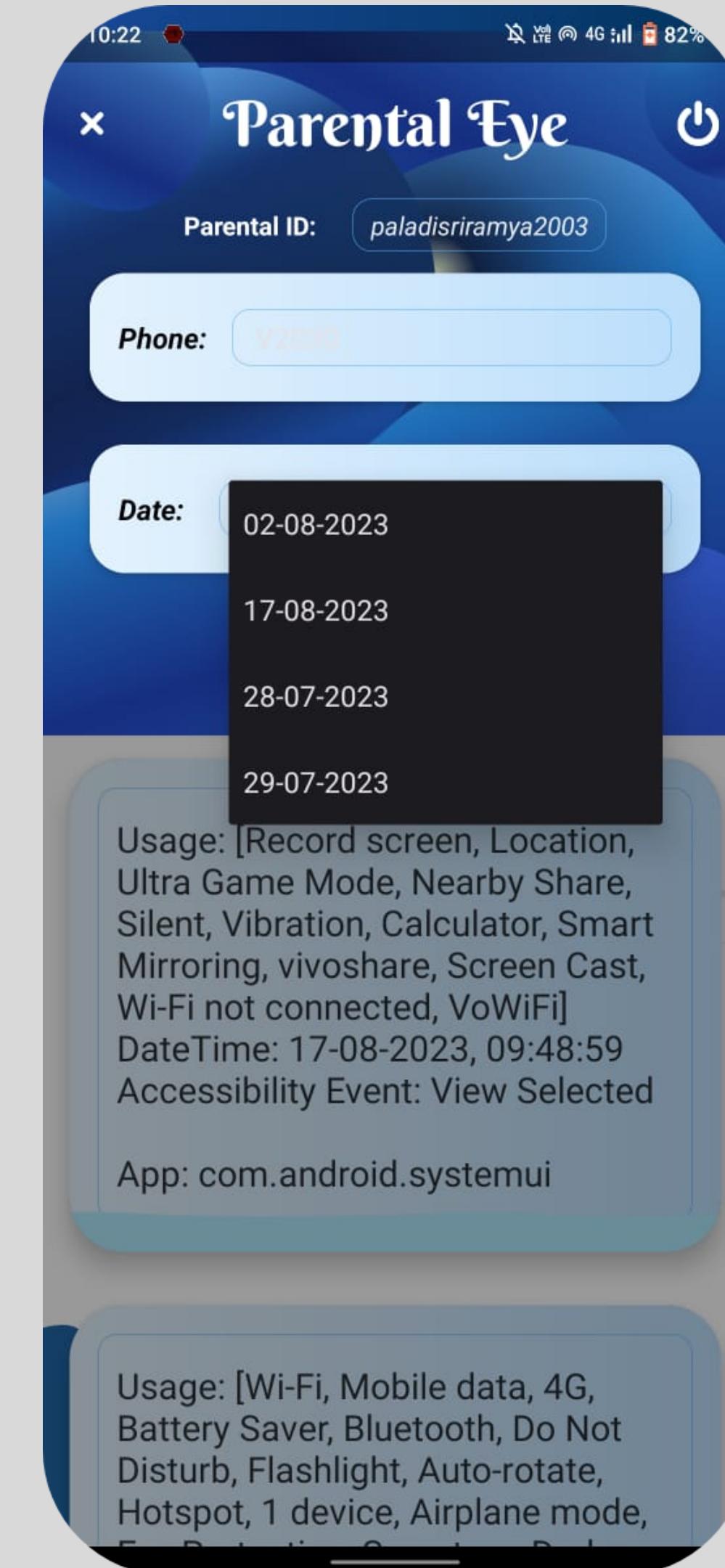
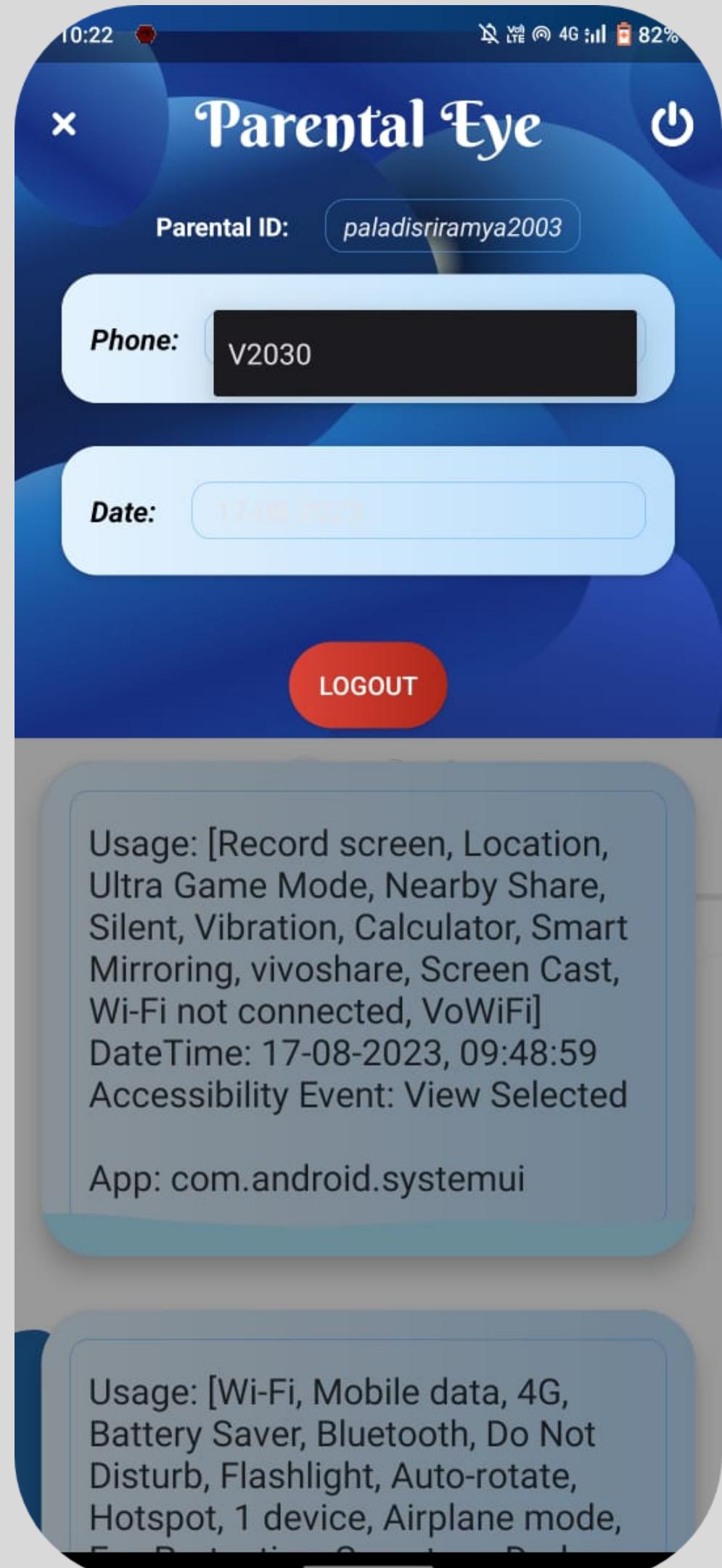


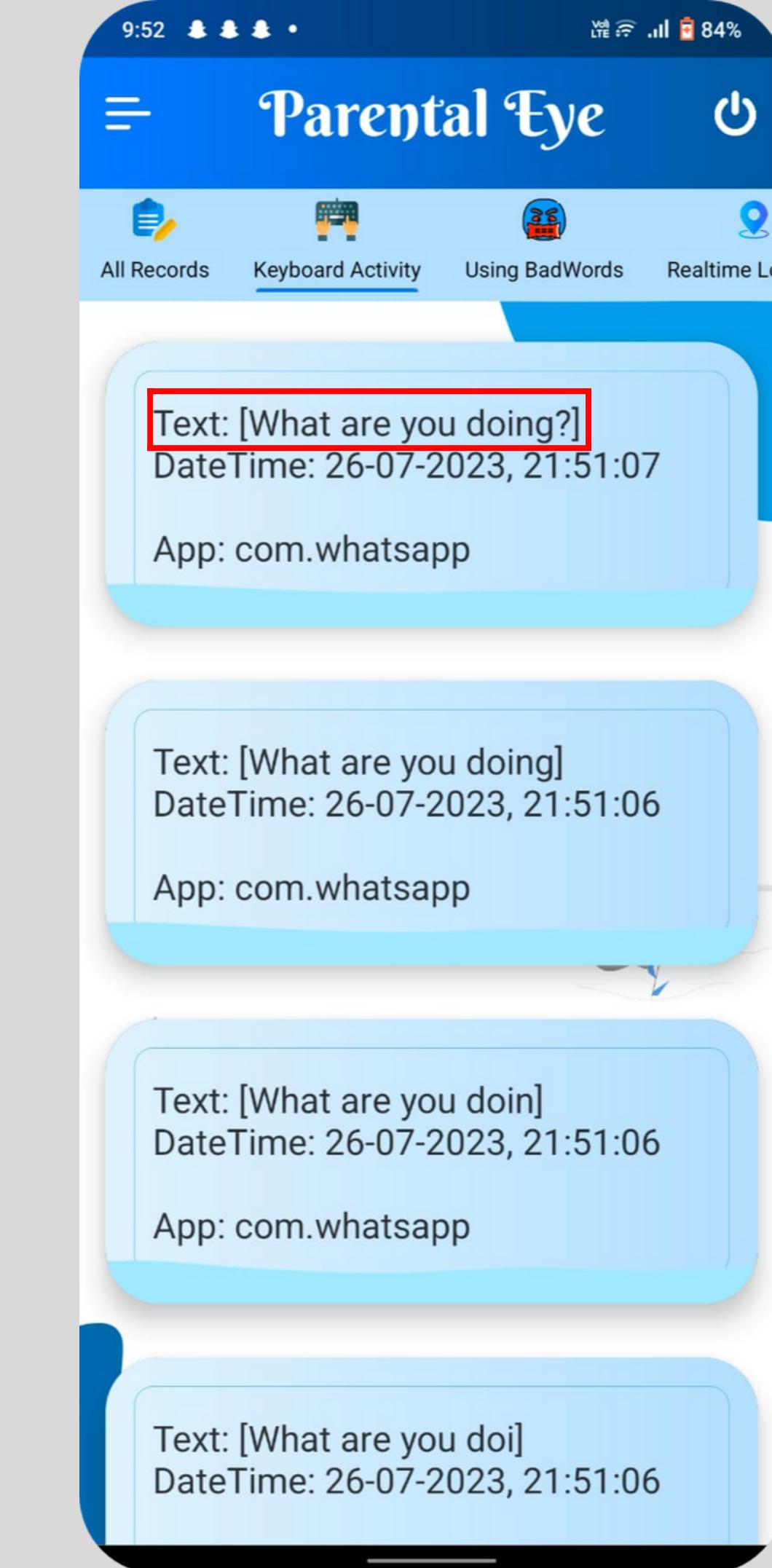
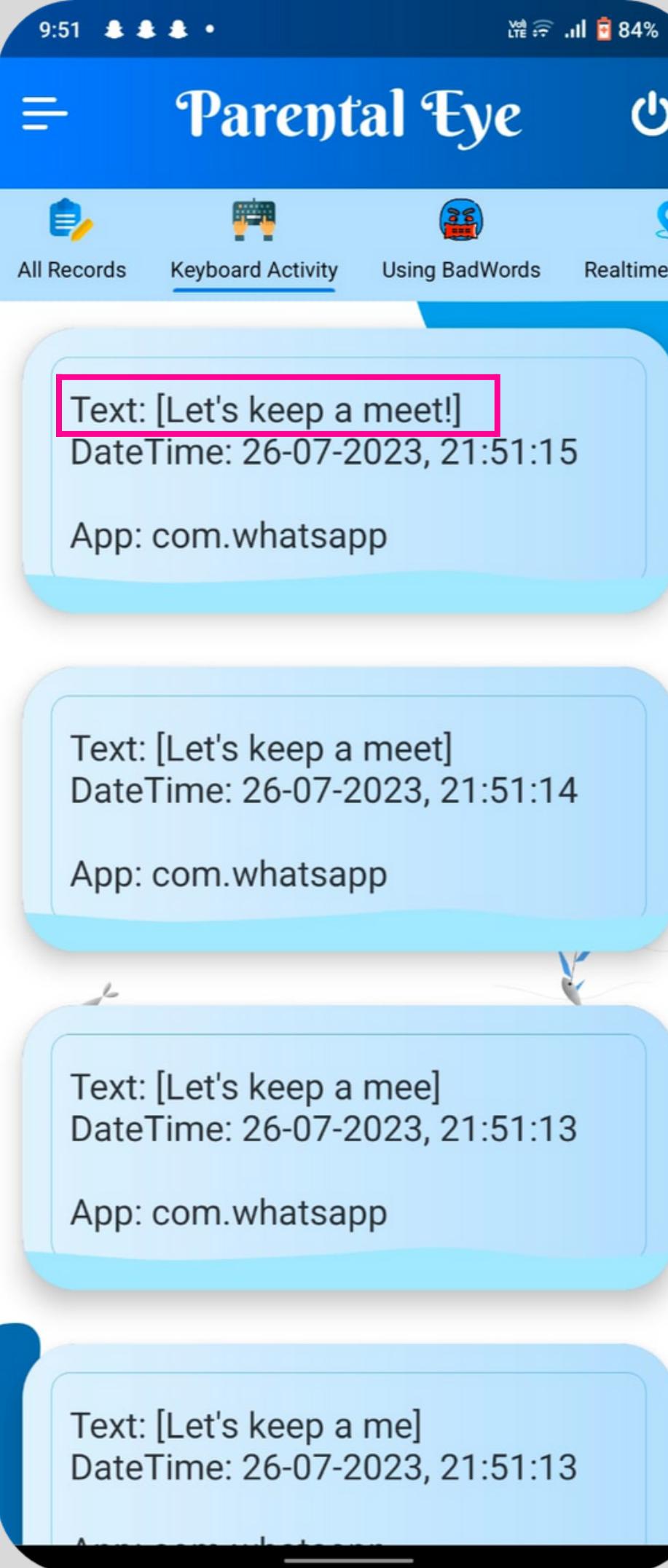
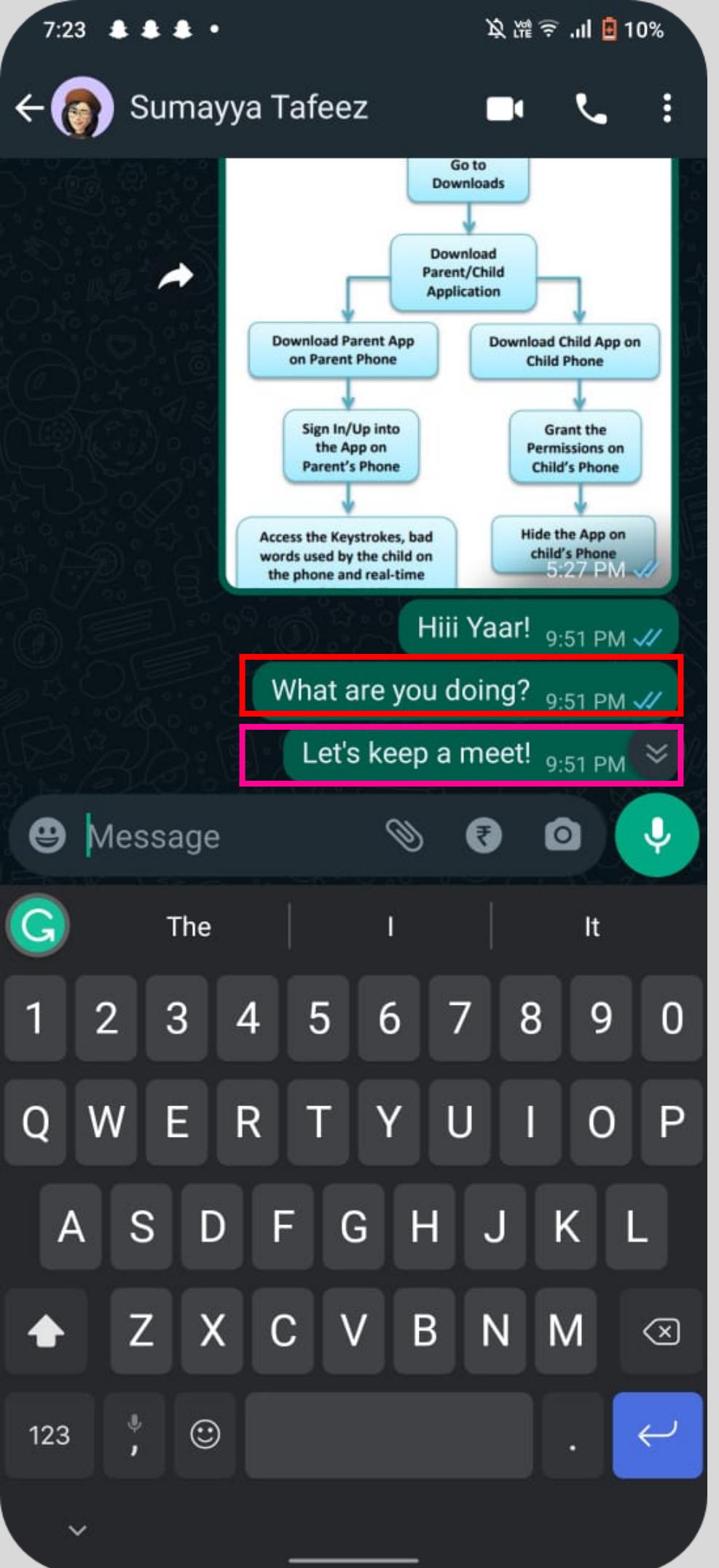


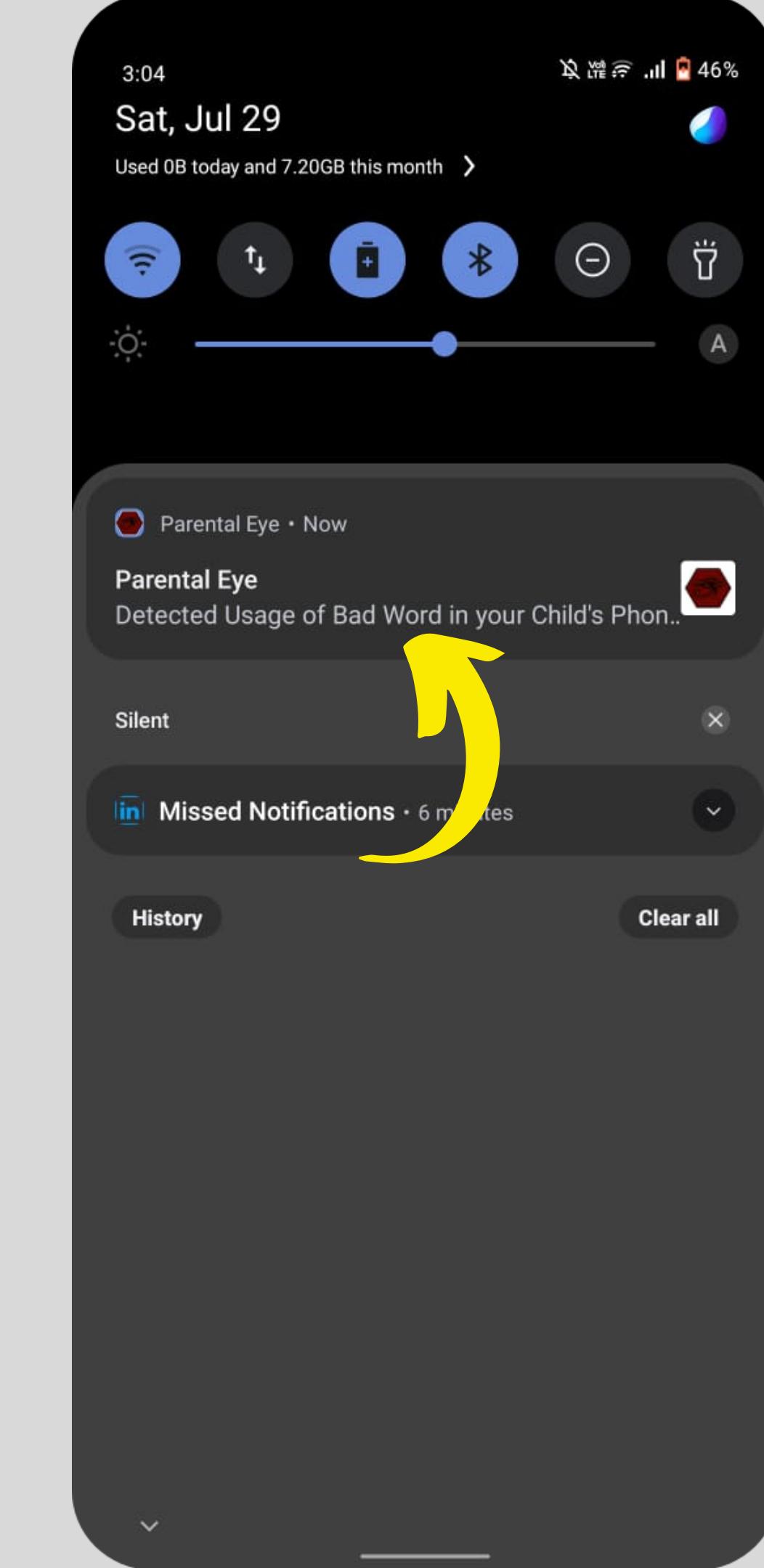
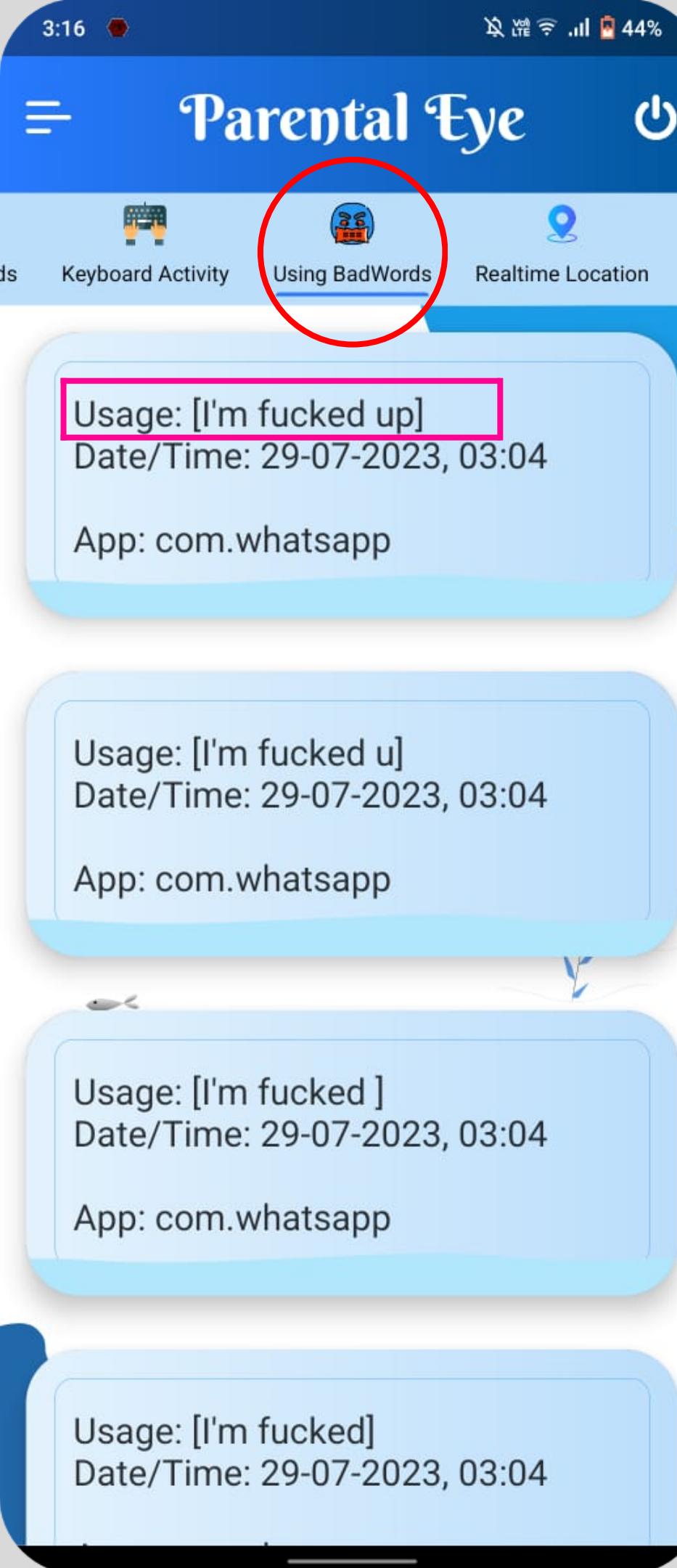
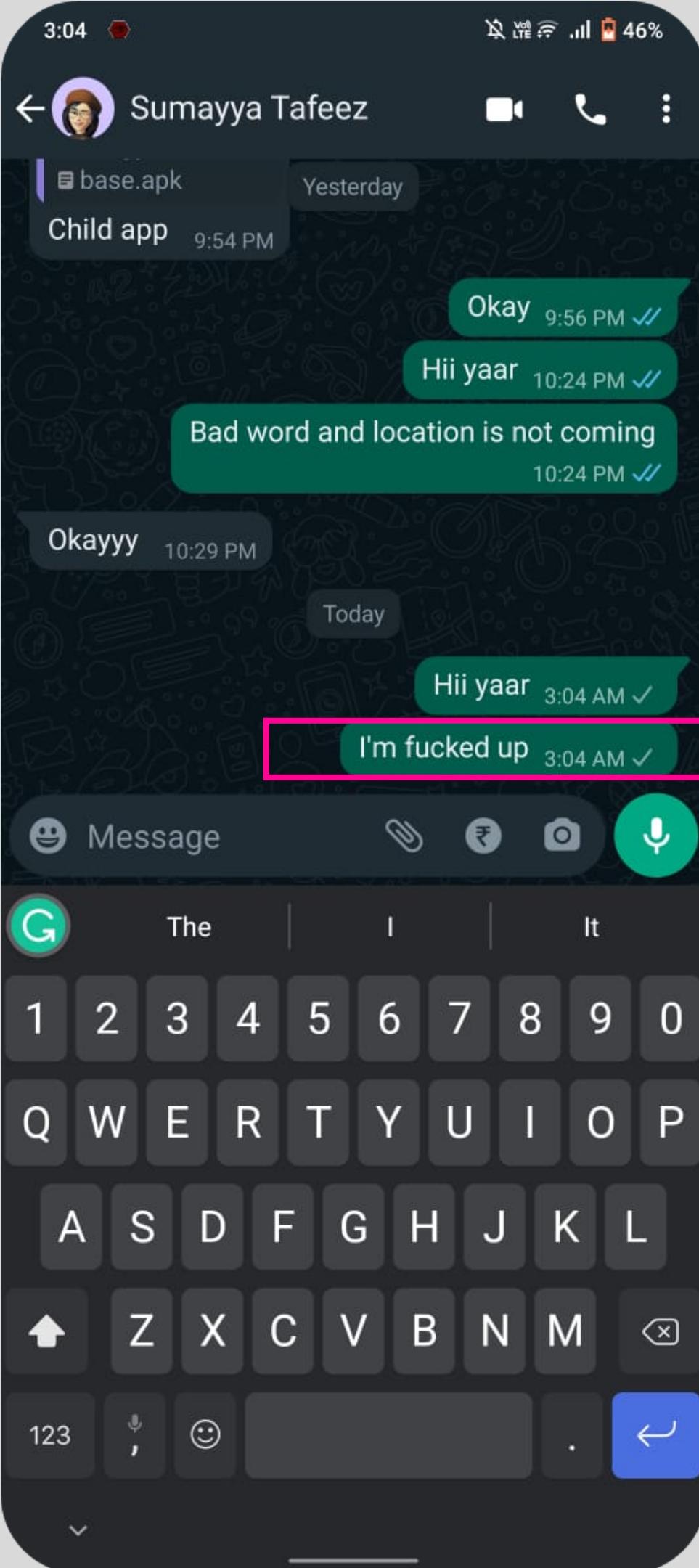


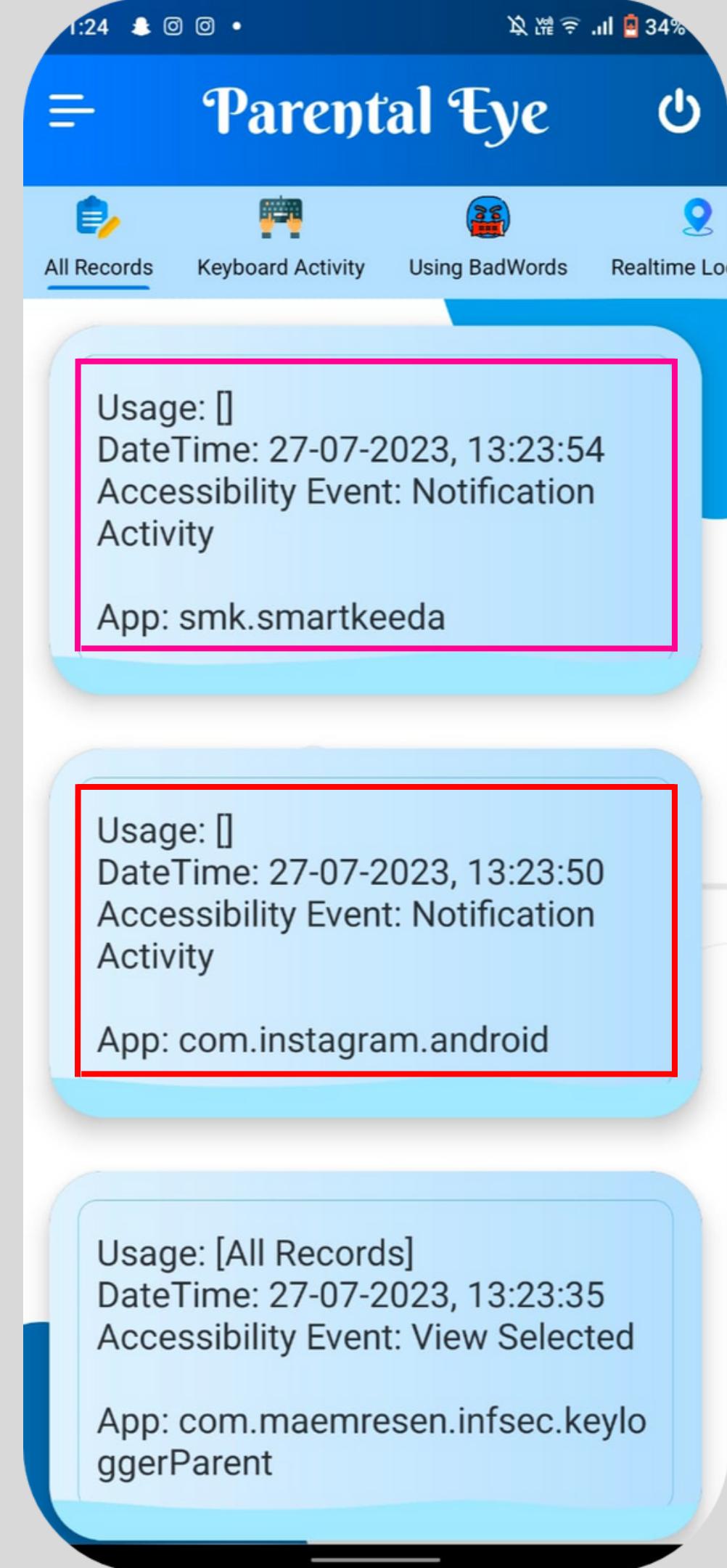
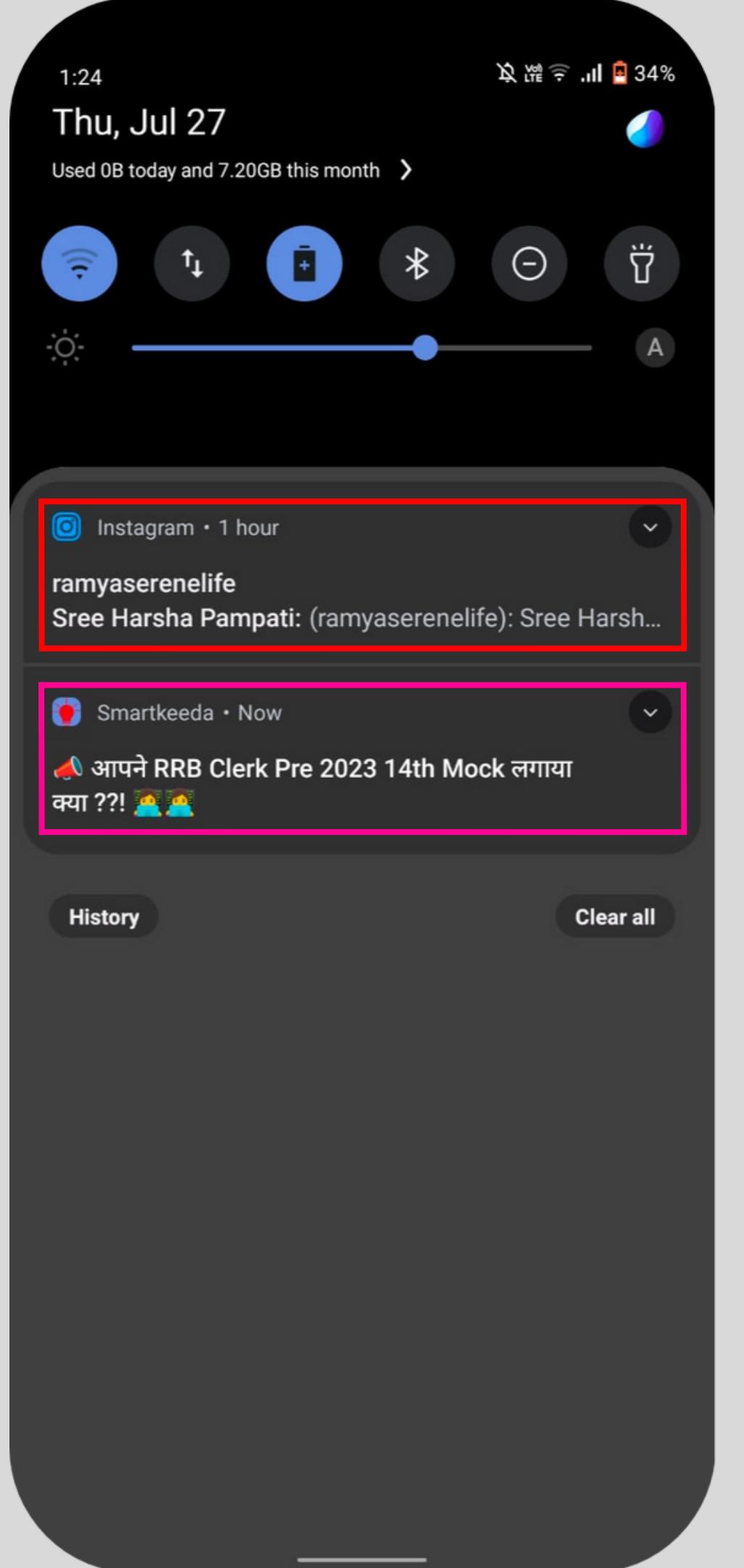
# Parent App

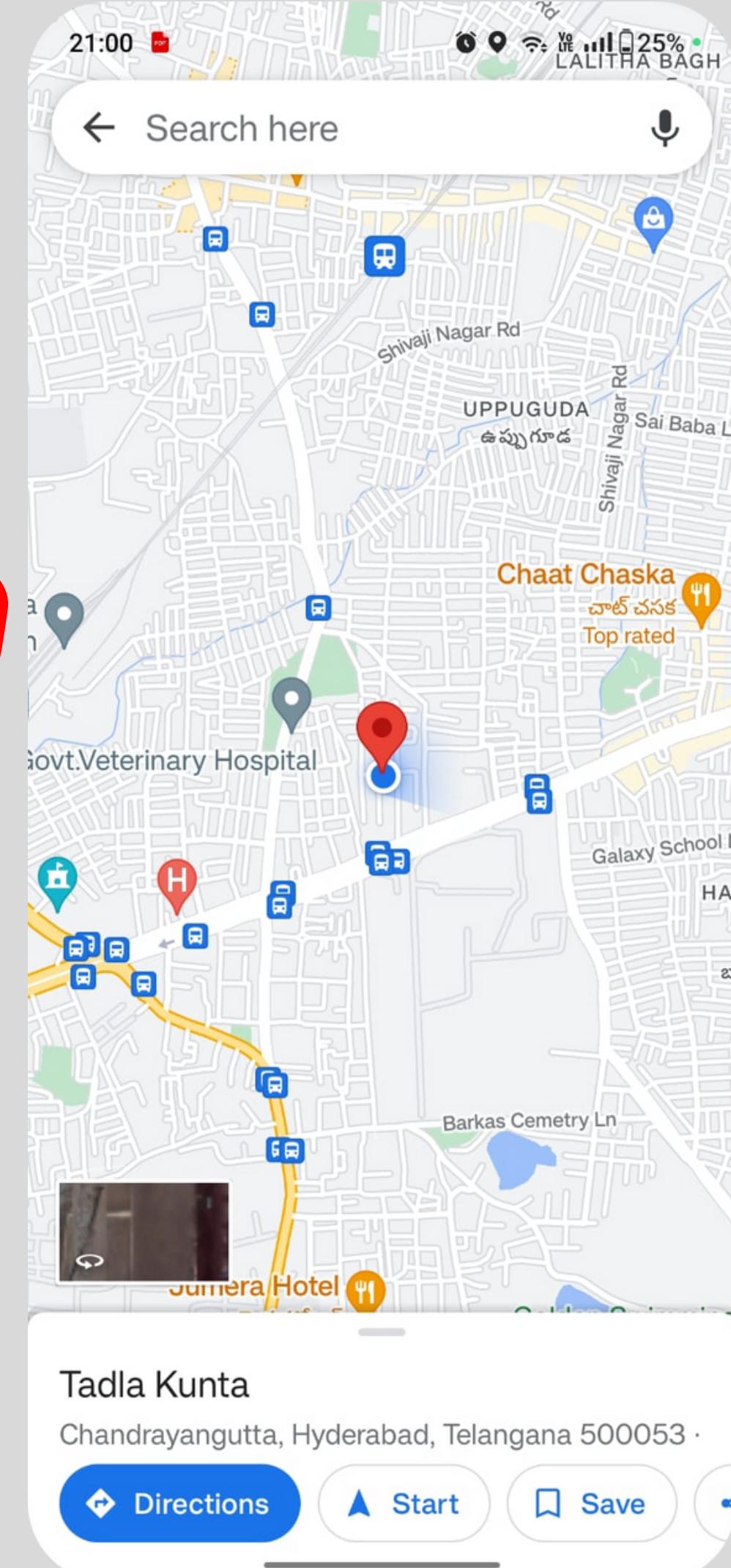
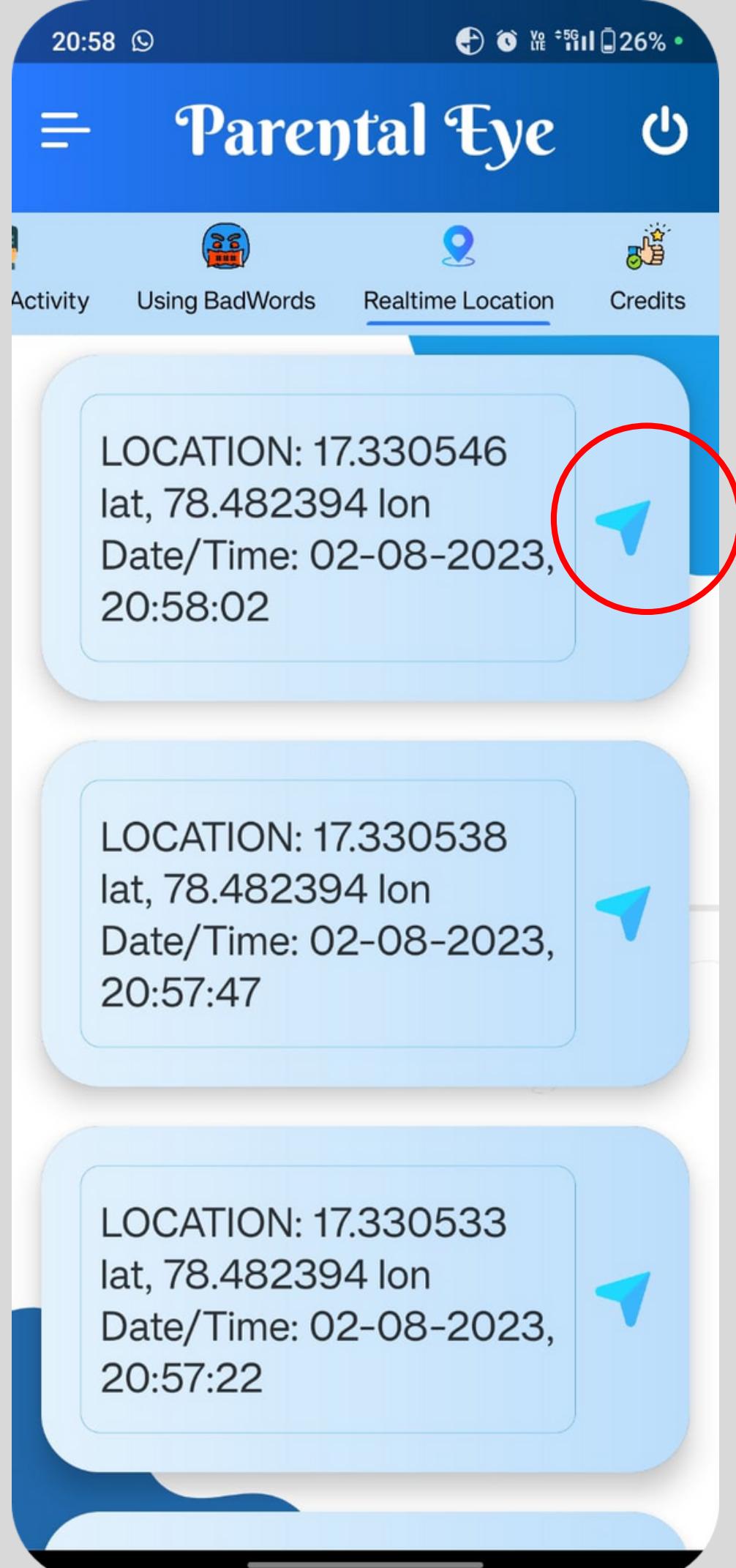


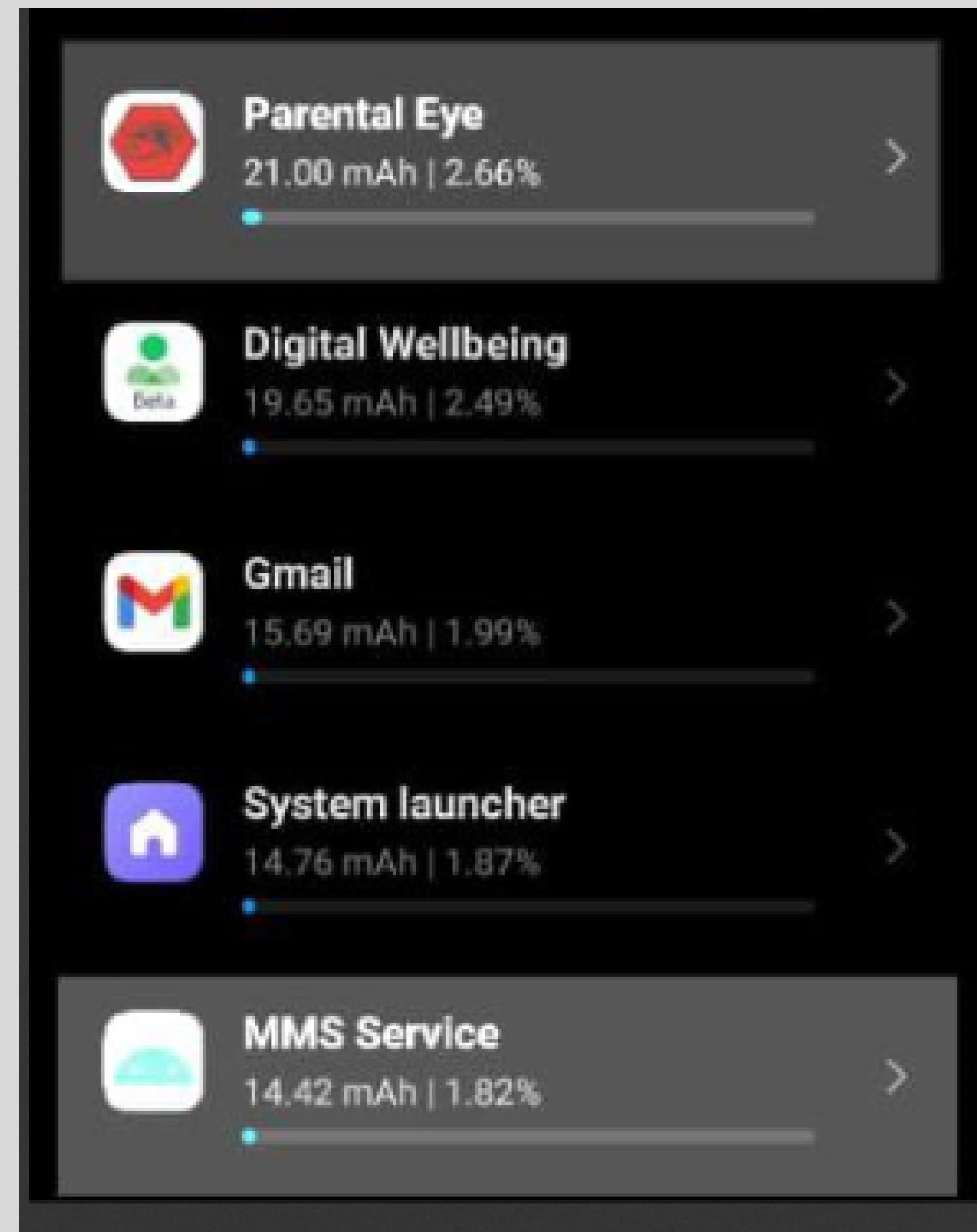
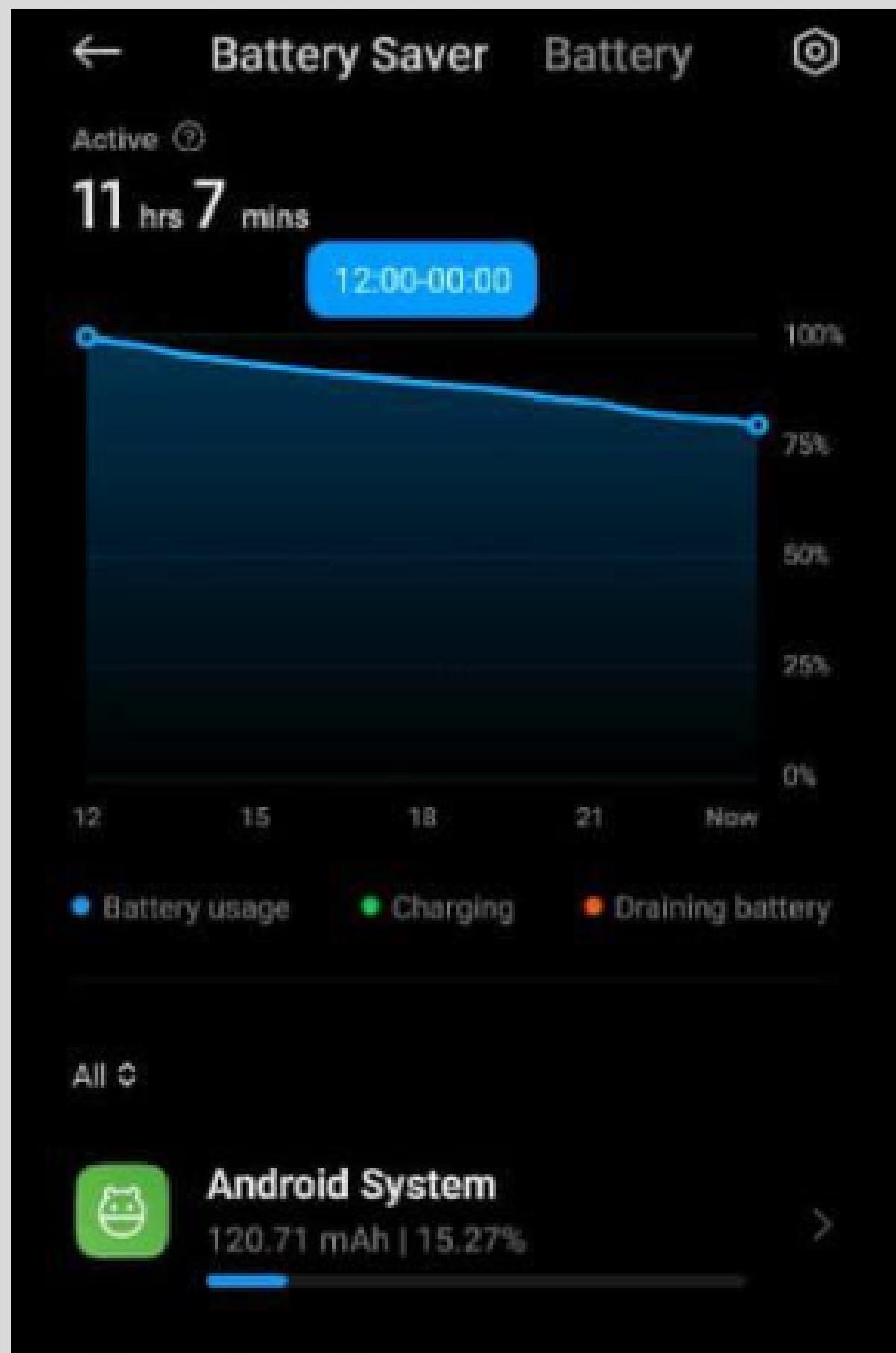






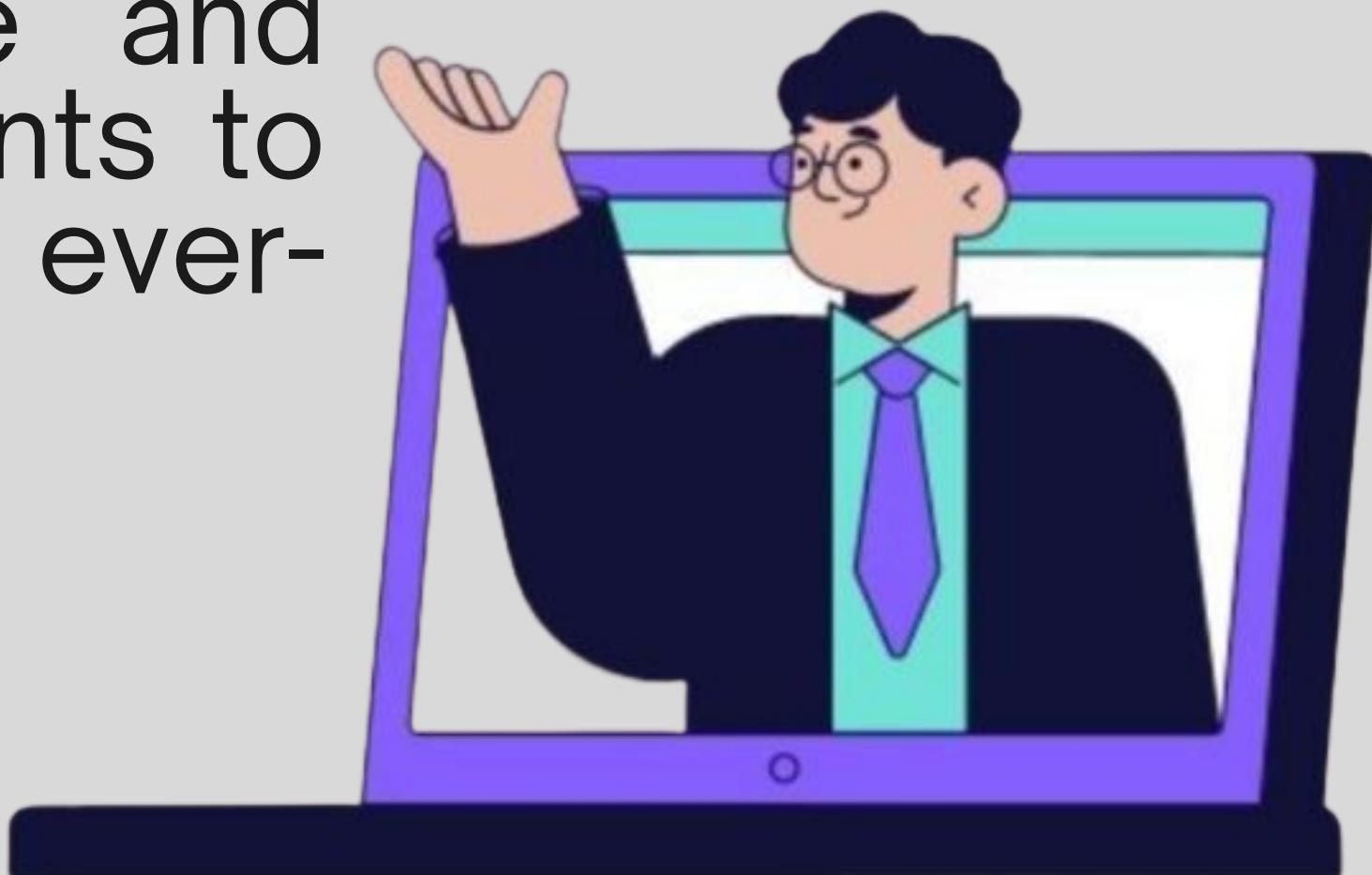






# CONCLUSION

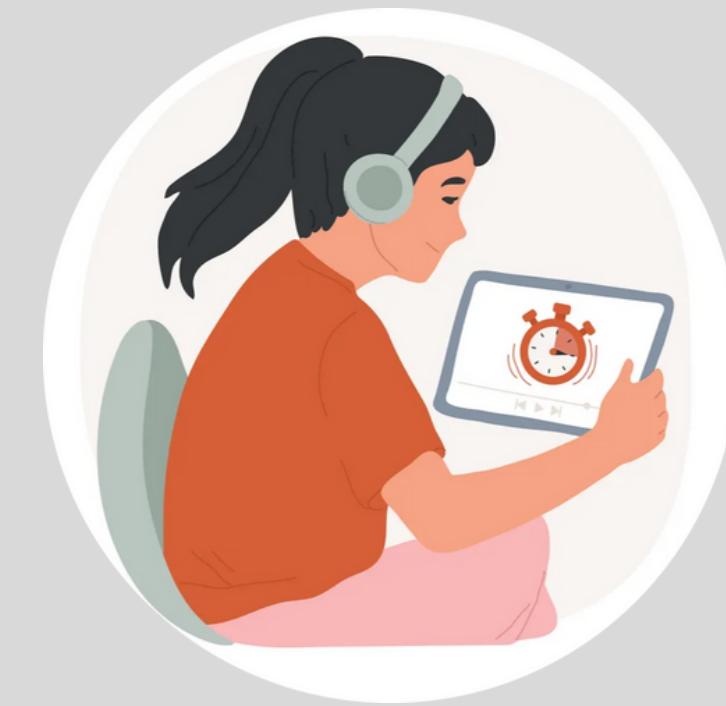
Our project aims to ensure children's safety online and promote responsible digital parenting. By addressing user preferences, improving usability, and embracing technological advancements, the app can be more effective and widely adopted, empowering parents to protect their children in the ever-changing digital world.



# FUTURE SCOPE



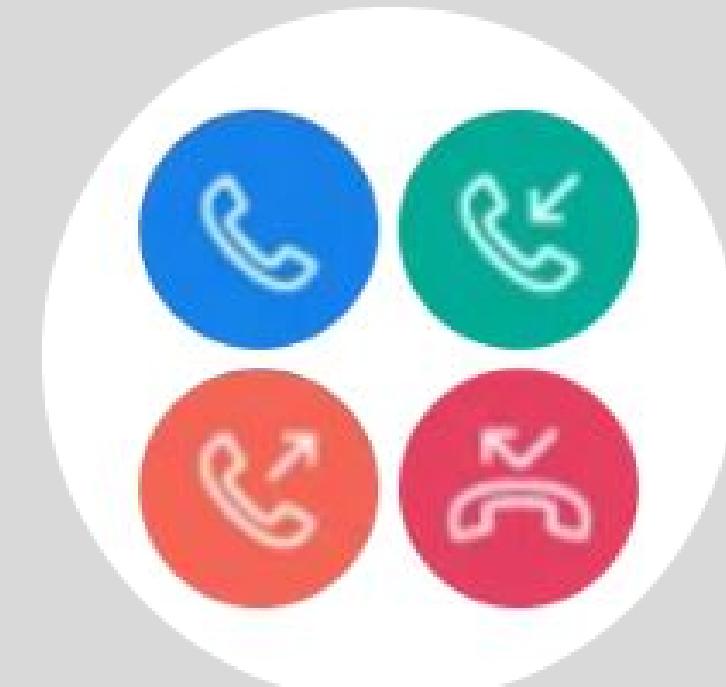
**Content Filtering**



**Screen Time Management**



**App Usage Reports**

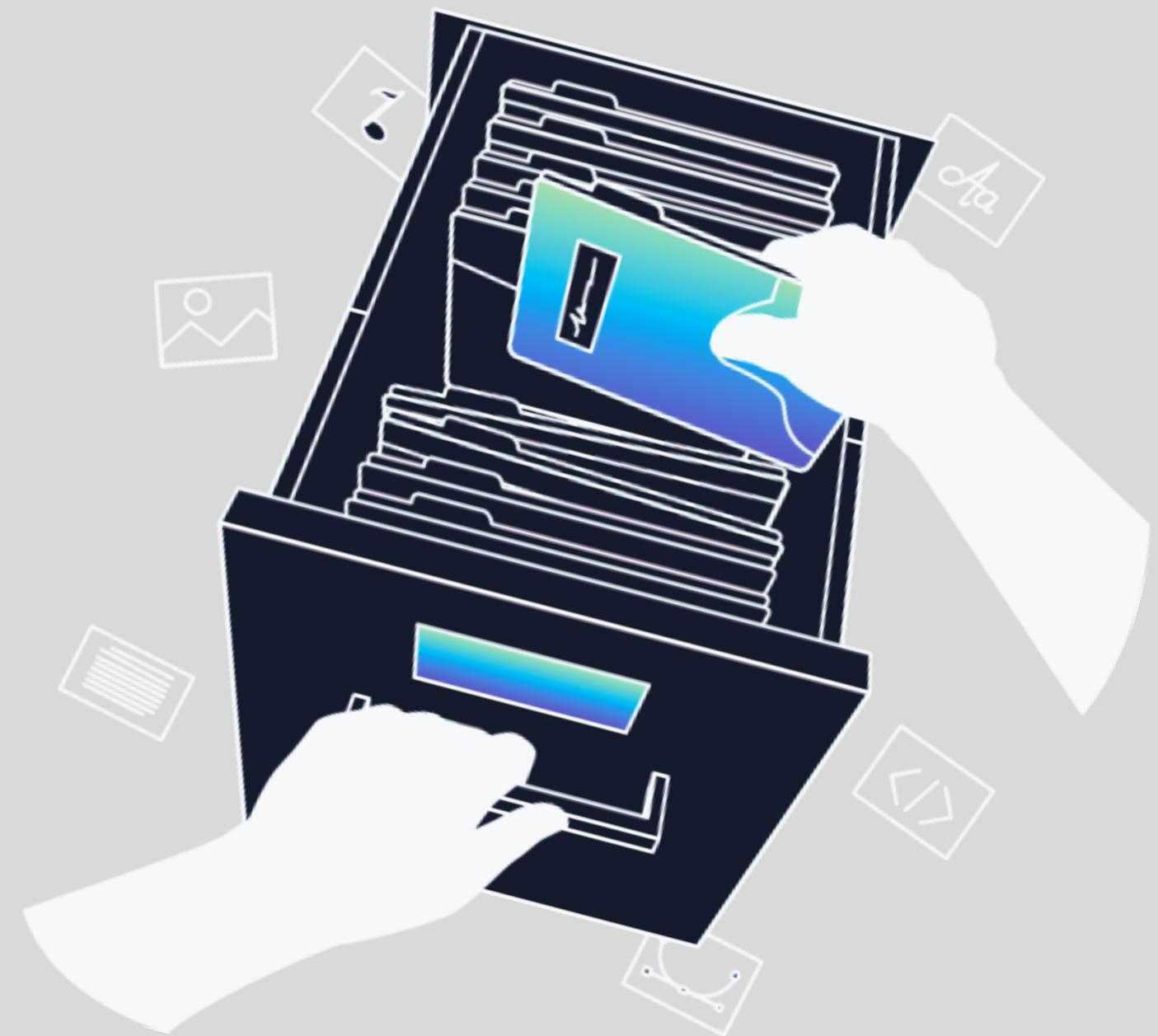


**Call Logs Monitoring**



# REFERENCES

- <https://youtu.be/gB6WLkSrtJk>
- <https://youtu.be/I2UBjN5ER4s>
- [https://youtube.com/playlist?  
list=PLUhfM8afLE\\_Ok-OLx2v9hfrmbxi3GgsX1](https://youtube.com/playlist?list=PLUhfM8afLE_Ok-OLx2v9hfrmbxi3GgsX1)
- [https://www.youtube.com/playlist?  
list=PLYx38U7gxBf3pmsHVTUwRT\\_lGON6ZIB](https://www.youtube.com/playlist?list=PLYx38U7gxBf3pmsHVTUwRT_lGON6ZIB)
- Hi , <https://youtu.be/HGXUwnpaVYA>
- [https://youtube.com/playlist?  
list=PLrnPJCHvNZuA80lNWNCICR3qYzhw3i](https://youtube.com/playlist?list=PLrnPJCHvNZuA80lNWNCICR3qYzhw3i)
- PI,



**THANK YOU**

*Any  
Questions?*