

## 1. Project Idea

- **Concept:** AutoGuard Alert System is a mobile app for monitoring and alerting users about security threats in connected vehicles, such as CAN bus intrusions, unauthorized Wi-Fi access, and GPS spoofing.
- **Features:**
  - Real-time threat alerts with a user-friendly interface.
  - Alert details, event logs, and customizable monitoring settings (CAN, Wi-Fi, Bluetooth).
  - Secure login system for user authentication.
- **Purpose:**
  - Provide real-time security alerts.
  - Enable alert investigation and historical log analysis.
  - Allow configuration of monitoring preferences.
- **Target Audience:** Vehicle owners, fleet managers, and cybersecurity researchers.

## 2. Prototype (Screens and Interactions)

- **Screens:**
  - **Login:** Email/password input, "Forgot Password?" link, and login button. Validates credentials and navigates to Dashboard on success.

## Login

# IDS Login

Email

Password

[Forgot Password?](#)

Login

Test Credentials: mariam@example.com / password123

- **Dashboard:** Displays current alerts (FlatList), with buttons for Logs and Settings. Alerts link to AlertDetails.

← Dashboard

## IDS Dashboard

Current Alerts

Message injection on CAN bus at 10:00 AM

Unauthorized Wi-Fi access at 09:45 AM

View Logs

Settings

- **AlertDetails:** Shows selected alert's message and severity, with a button to return to Dashboard.

← AlertDetails

## Alert Details

Message: Message injection on CAN bus at  
10:00 AM

Severity: high

[Back to Dashboard](#)

- **Logs:** Lists past security events (FlatList), with a back button to Dashboard.

← Logs

## Attack Logs

10:00 AM - CAN bus intrusion

(Severity: high)

09:45 AM - Unauthorized Wi-Fi access attempt

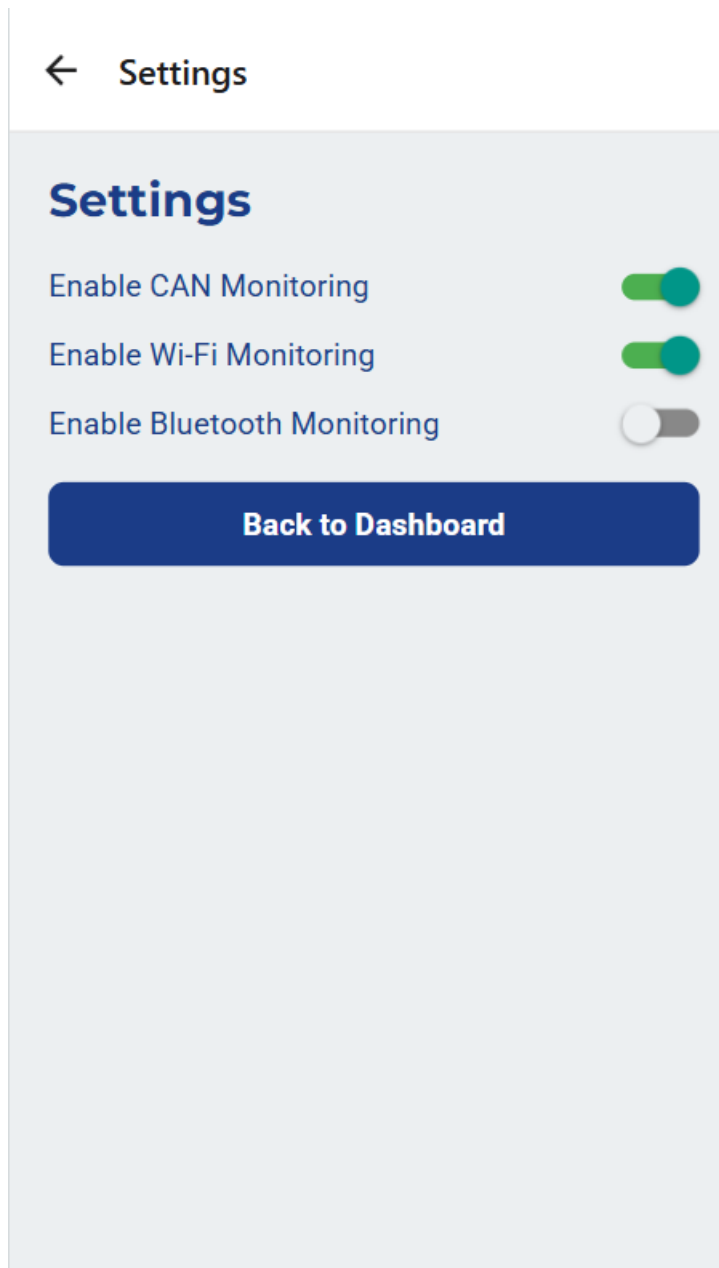
(Severity: low)

09:30 AM - GPS signal spoofing detected

(Severity: medium)

Back to Dashboard

- **Settings:** Toggle switches for monitoring (CAN, Wi-Fi, Bluetooth), with a back button to Dashboard.



- **Navigation:** Uses @react-navigation/stack. Flow: Login → Dashboard → AlertDetails/Logs/Settings → Back to Dashboard.
- **Screenshots:** Confirm UI elements (headers, buttons, lists) and navigation setup in App.js.

### 3. Project Colors

- **Primary:** #1B3C87 (deep blue) – headers, buttons, labels.
- **Secondary:** #F9602E (orange-red) – error messages, high-severity alerts.

- **Neutral:** #ECEFF1 (light gray) – background.
- **Accent:** #4CAF50 (green) – low-severity alerts, "Forgot Password?" link.
- **Usage:** Consistent across screens (e.g., Login errors in #F9602E, Dashboard alerts in #F9602E/#4CAF50).

#### 4. Project Fonts

- **Title:** Montserrat-Bold – headers (e.g., "IDS Login").
- **Header:** Montserrat-Regular – subheaders (e.g., "Current Alerts").
- **Regular:** Roboto-Regular – body text, labels.
- **Bold:** Roboto-Bold – button text, emphasized text.
- **Storage:** Fonts in assets/fonts/ (Montserrat-Bold.ttf, etc.).
- **Loading:** Handled in App.js with expo-font, displaying a loading message until fonts load.

#### 5. Custom Buttons

- **Component:** AlertButton in components/AlertButton.js.
- **Features:**
  - Props: title, onPress, style, severity.
  - Dynamic colors: #F9602E (high severity), #4CAF50 (low severity), #1B3C87 (default).
  - Consistent styling: White text, Roboto-Bold, rounded corners.
- **Usage:** Login ("Login"), Dashboard (alerts, navigation), and back buttons in other screens.
- **Storage:** In components/ folder.

#### 6. Initial Screens (JSX with States)

- **Login** (screens/Login.js):
  - States: username, password, error.
  - Validates mock credentials, navigates to Dashboard.
- **Dashboard** (screens/Dashboard.js):

- State: alerts (mock array).
  - Lists alerts, navigates to AlertDetails/Logs/Settings.
- **AlertDetails** (screens/AlertDetails.js):
  - No state; uses route.params for alert data.
  - Shows alert details, returns to Dashboard.
- **Logs** (screens/Logs.js):
  - No state; uses hardcoded logs.
  - Displays logs, returns to Dashboard.
- **Settings** (screens/Settings.js):
  - States: isCANEnabled, isWiFiEnabled, isBluetoothEnabled.
  - Toggles monitoring settings, returns to Dashboard.
- **Storage:** All screens in screens/ folder.
- **Navigation:** Defined in App.js with Stack.Navigator, starting at Login.

### **Project State Summary**

- A vehicle security app with 5 screens, basic navigation, and a consistent design.
- Uses a defined color scheme (#1B3C87, #F9602E, #ECEFF1, #4CAF50) and fonts (Montserrat, Roboto).
- Features a custom AlertButton for dynamic interactions.
- Screens are fully implemented with states for Login, Dashboard, and Settings, stored in appropriate folders.