BujjiController Setup and Command Log

# 1. Project Overview

- Name: BujjiController  
- Purpose: Android controller app for ESP32-CAM and similar IP-based robots. It's **not limited only to the ESP32-CAM**. While you've currently built it for controlling and streaming from an ESP32-CAM robot, the app can be **expanded to support other ESP32 boards** (like ESP32-WROOM, ESP32-S3, etc.) or **any IP camera-based robotic system**

with minor to moderate changes:

* **Other ESP32 boards** like ESP32-S3, ESP32-WROOM, etc.  
  👉 As long as they serve a stream and accept control commands.
* **Any IP camera or robot**  
  👉 If it uses a compatible video stream (MJPEG/RTSP) and has a simple control API.

- Language: Java  
- Platform: Android

# 2. Dependencies

- Java SDK  
- Android SDK CLI  
- Gradle build tools  
- ESP32 UDP communication  
- IP video streaming

# 3. Code Changes

- Implemented UDP broadcast to auto-detect ESP32-CAM IP address.  
- Added manual IP entry fallback with SharedPreferences saving.  
- Integrated MJPEG video stream in WebView.  
- Added toggle buttons for motion control and flashlight via HTTP.  
- Created an Android app using pure Java (without Android Studio or Flutter).  
- App icon updated to "RV".  
- Added Gradle configuration to build a release APK.  
- Signed APK with a custom release keystore (alias: my-key-alias).

# 4. App Features

- Manual IP entry and auto-saving for future sessions.  
- UDP broadcast IP discovery for plug-and-play functionality.  
- Toggle flashlight (HTTP command to ESP32-CAM).  
- Toggle object tracking (via motion command).  
- Live video feed stream embedded via WebView.