Subject: Embedded systems

(BCSE 305L)

Digital Assignment

TEAM: ALLAM NEHAL VARMA(22BCE0414)

Project Title:

"Mobile Phone Hardware Controller via ADB Interface"

Purpose & Functionality

The purpose of this project is to design and implement a web-based embedded control system that allows a user to interact with and operate key hardware functions of an Android smartphone from a computer. By using Android Debug Bridge (ADB) as the communication protocol and integrating it with a Flask-based backend server and a custom-designed browser UI, this system enables remote control of features such as:

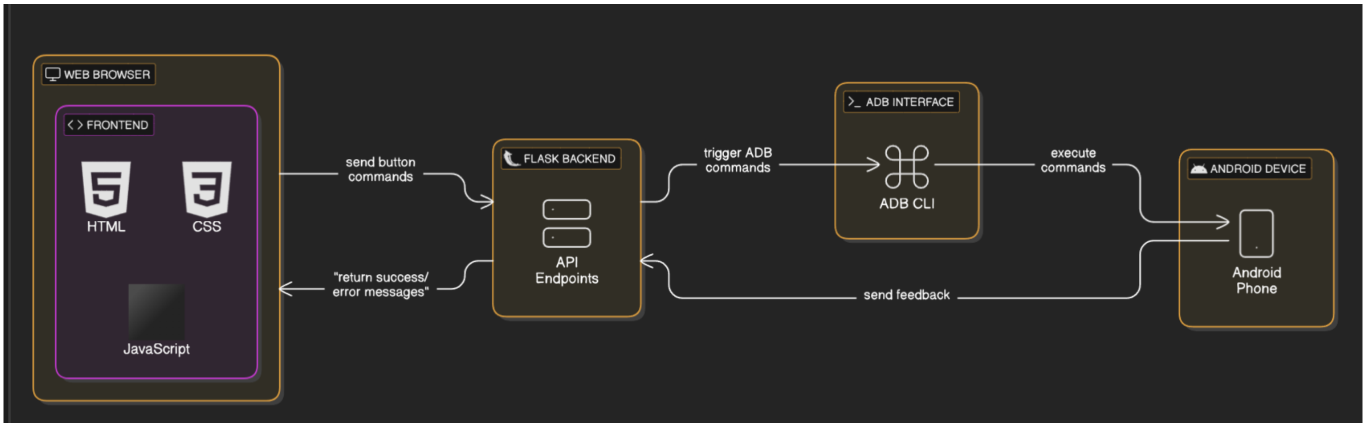
- Volume control (up/down/mute)
- WiFi toggle
- App launching (YouTube, Play Store, Spotify)
- Taking screenshots
- Powering the screen ON/OFF

The frontend is styled like a mobile interface using HTML, CSS, and PNG icons to simulate app-like controls. The entire solution runs without requiring the mobile device to be rooted, making it accessible for users with standard Android configurations.

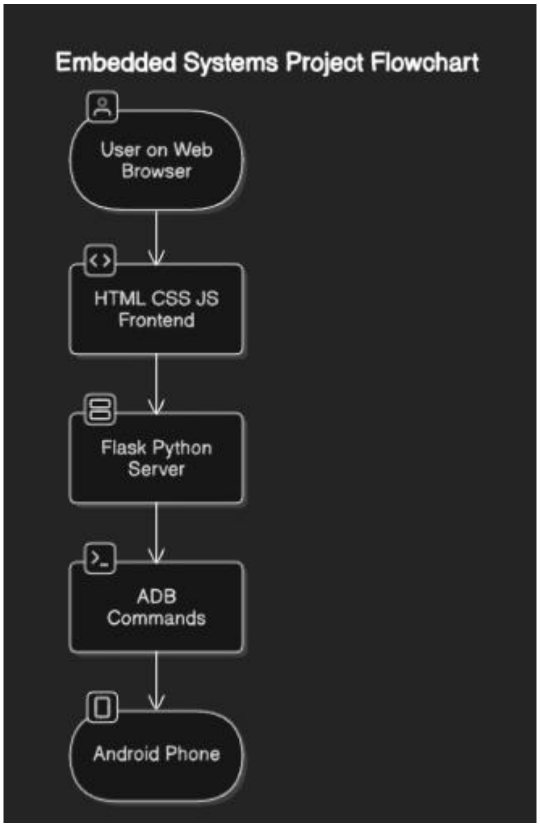
1. Hardware Requirements Specification (HRS):

- Android Phone with Developer Mode enabled
- USB Data Cable
- Laptop/PC with:
    - Python Installed
    - ADB (Android Debug Bridge) Installed
    - Flask (for backend server)
    - Modern browser (for frontend UI)

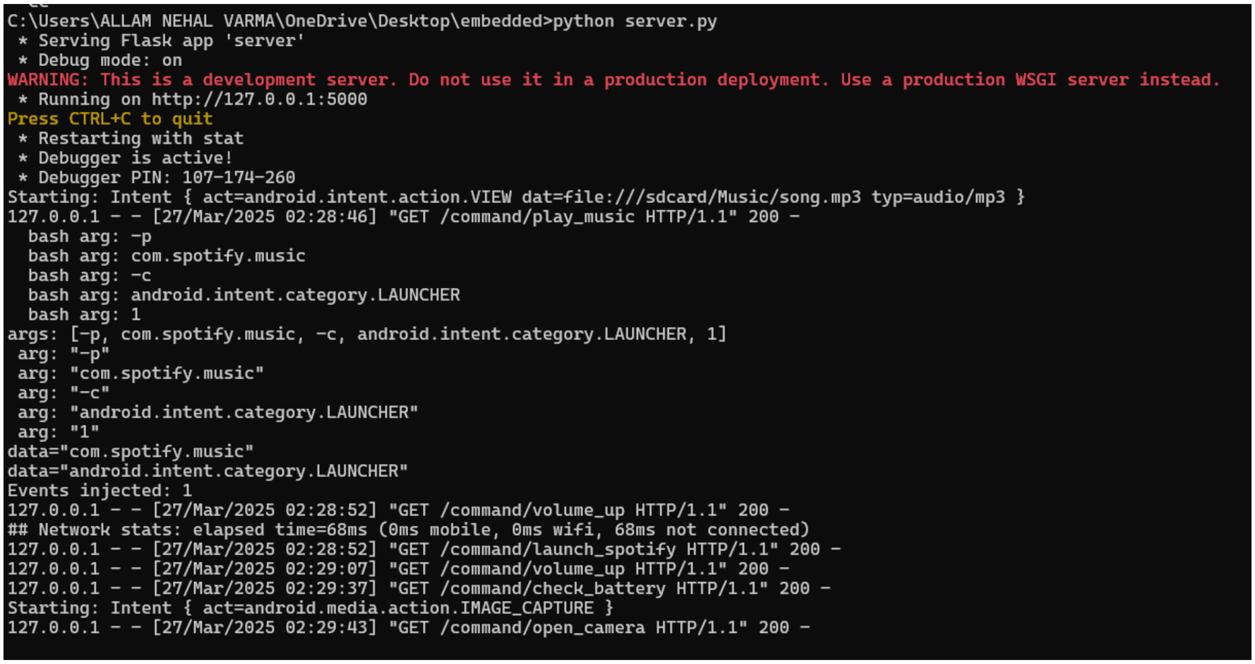
2. Software Diagram (Architecture):



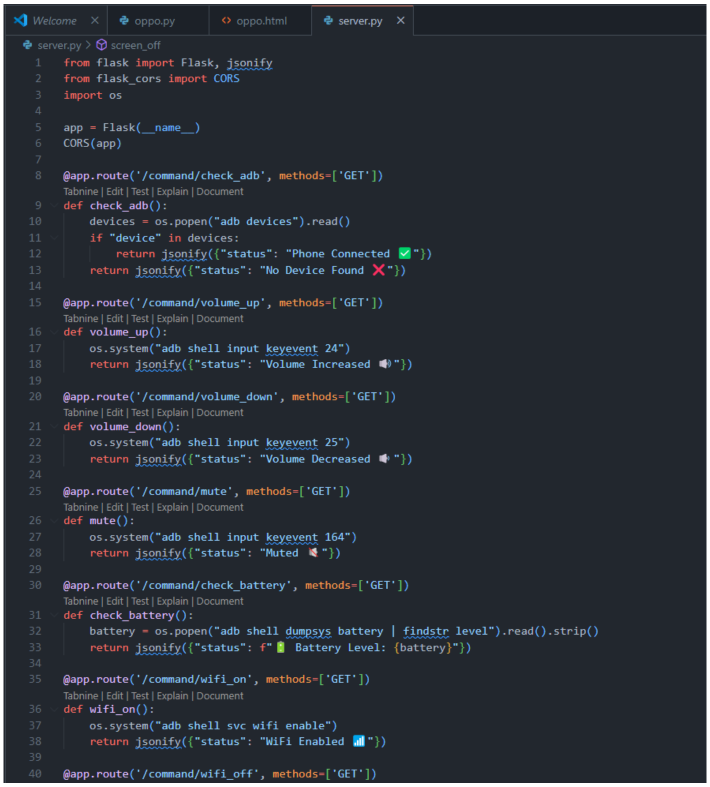
3. Overall Block Diagram:



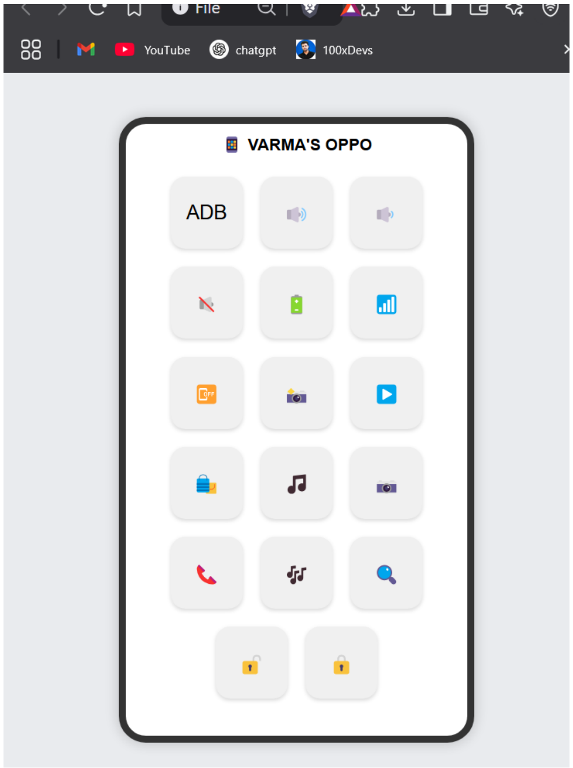
4. Screenshots:

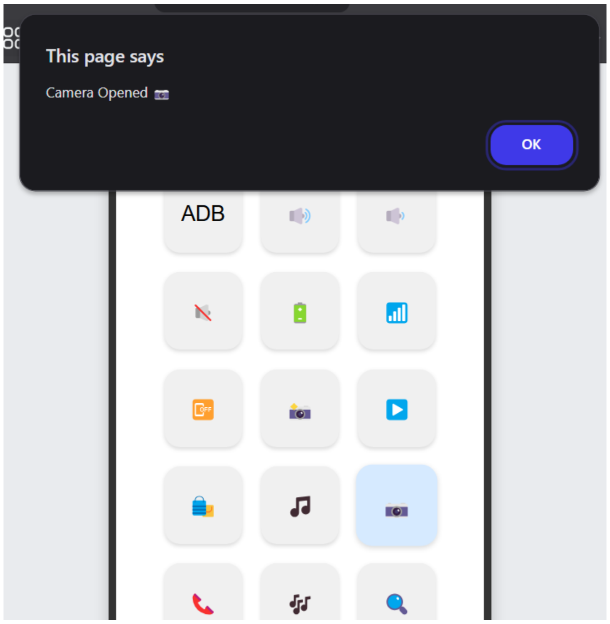


```
C:\Users\ALLAM NEHAL VARMA\OneDrive\Desktop\embedded>python server.py
 * Serving Flask app 'server'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 107-174-260
Starting: Intent { act=android.intent.action.VIEW dat=file:///sdcard/Music/song.mp3 typ=audio/mp3 }
127.0.0.1 - - [27/Mar/2025 02:28:46] "GET /command/play_music HTTP/1.1" 200 -
  bash arg: -p
  bash arg: com.spotify.music
  bash arg: -c
  bash arg: android.intent.category.LAUNCHER
  bash arg: 1
args: [-p, com.spotify.music, -c, android.intent.category.LAUNCHER, 1]
 arg: "-p"
 arg: "com.spotify.music"
 arg: "-c"
 arg: "android.intent.category.LAUNCHER"
 arg: "1"
data="com.spotify.music"
data="android.intent.category.LAUNCHER"
Events injected: 1
127.0.0.1 - - [27/Mar/2025 02:28:52] "GET /command/volume_up HTTP/1.1" 200 -
## Network stats: elapsed time=68ms (0ms mobile, 0ms wifi, 68ms not connected)
127.0.0.1 - - [27/Mar/2025 02:28:52] "GET /command/launch_spotify HTTP/1.1" 200 -
127.0.0.1 - - [27/Mar/2025 02:29:07] "GET /command/volume_up HTTP/1.1" 200 -
127.0.0.1 - - [27/Mar/2025 02:29:37] "GET /command/check_battery HTTP/1.1" 200 -
Starting: Intent { act=android.media.action.IMAGE_CAPTURE }
127.0.0.1 - - [27/Mar/2025 02:29:43] "GET /command/open_camera HTTP/1.1" 200 -
```

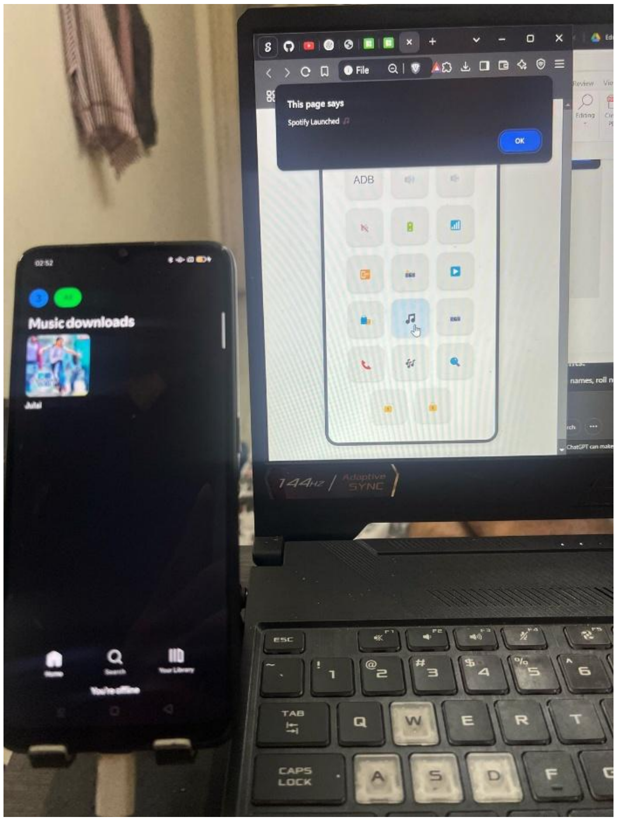


```python
from flask import Flask, jsonify
from flask_cors import CORS
import os

app = Flask(__name__)
CORS(app)

@app.route('/command/check_adb', methods=['GET'])
def check_adb():
    devices = os.popen("adb devices").read()
    if "device" in devices:
        return jsonify({"status": "Phone Connected ✅"})
    return jsonify({"status": "No Device Found ❌"})

@app.route('/command/volume_up', methods=['GET'])
def volume_up():
    os.system("adb shell input keyevent 24")
    return jsonify({"status": "Volume Increased 🔊"})

@app.route('/command/volume_down', methods=['GET'])
def volume_down():
    os.system("adb shell input keyevent 25")
    return jsonify({"status": "Volume Decreased 🔉"})

@app.route('/command/mute', methods=['GET'])
def mute():
    os.system("adb shell input keyevent 164")
    return jsonify({"status": "Muted 🔇"})

@app.route('/command/check_battery', methods=['GET'])
def check_battery():
    battery = os.popen("adb shell dumpsys battery | findstr level").read().strip()
    return jsonify({"status": f"🔋 Battery Level: {battery}"})

@app.route('/command/wifi_on', methods=['GET'])
def wifi_on():
    os.system("adb shell svc wifi enable")
    return jsonify({"status": "WiFi Enabled 📶"})

@app.route('/command/wifi_off', methods=['GET'])
```




This page says

Camera Opened 📷



This page says

Spotify Launched