

Activitate practica de vara

Ideea proiectului pornește de la crearea unei aplicații Android care integrează un sistem de chat AI pentru interpretarea comenzilor text sau vocale și trimiterea comenzilor corespunzătoare către un microcontroler ESP32, cu scopul de a controla un ring LED WS2812 prin Wi-Fi.

Obiectivele principale ale proiectului sunt:

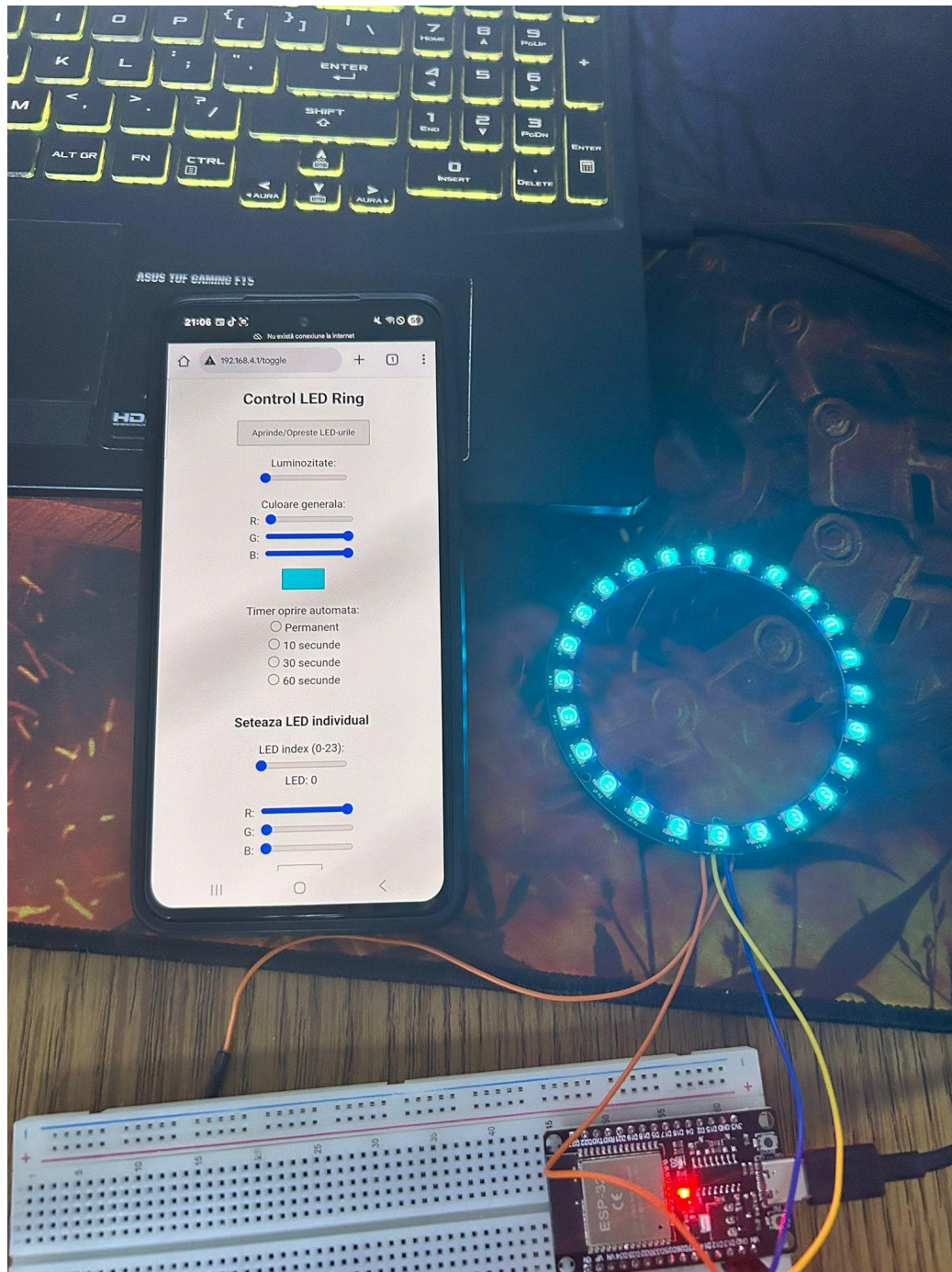
- Dezvoltarea unei aplicații Android care oferă o interfață de chat pentru utilizator;
- Integrarea unui model AI de la Google Gemini pentru interpretarea comenzilor text;
- Trimiterea comenzilor relevante, generate de AI, către un ESP32 aflat în aceeași rețea Wi-Fi;
- Programarea ESP32 pentru a recunoaște comenzile (de exemplu: schimbă culoarea în roșu, oprește toate LED-urile);
- Testarea aplicației și sistemului integrat în diferite scenarii de utilizare;

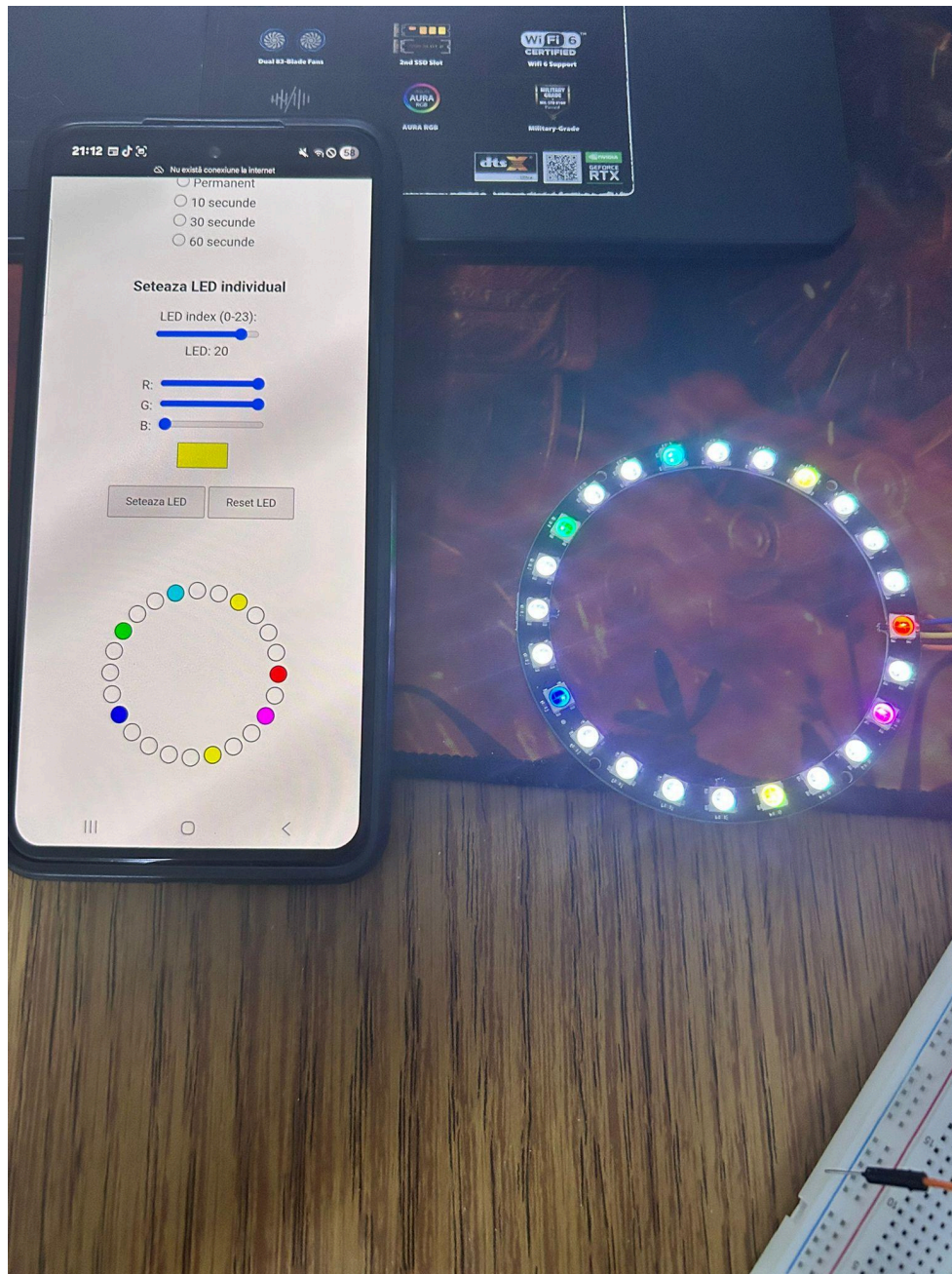
Luni - 30 iunie

- Am făcut rost de toate componentele hardware și software necesare.
- Am dezvoltat o aplicație simplă în MIT App Inventor pentru pornirea și stingerea ring-ului LED WS2812 prin ESP32.
- Am testat cu succes funcționalitatea de pornire și oprire a ring-ului LED.
- Estimat costul total al componentelor: ~100 lei.

Marti - 1 iulie

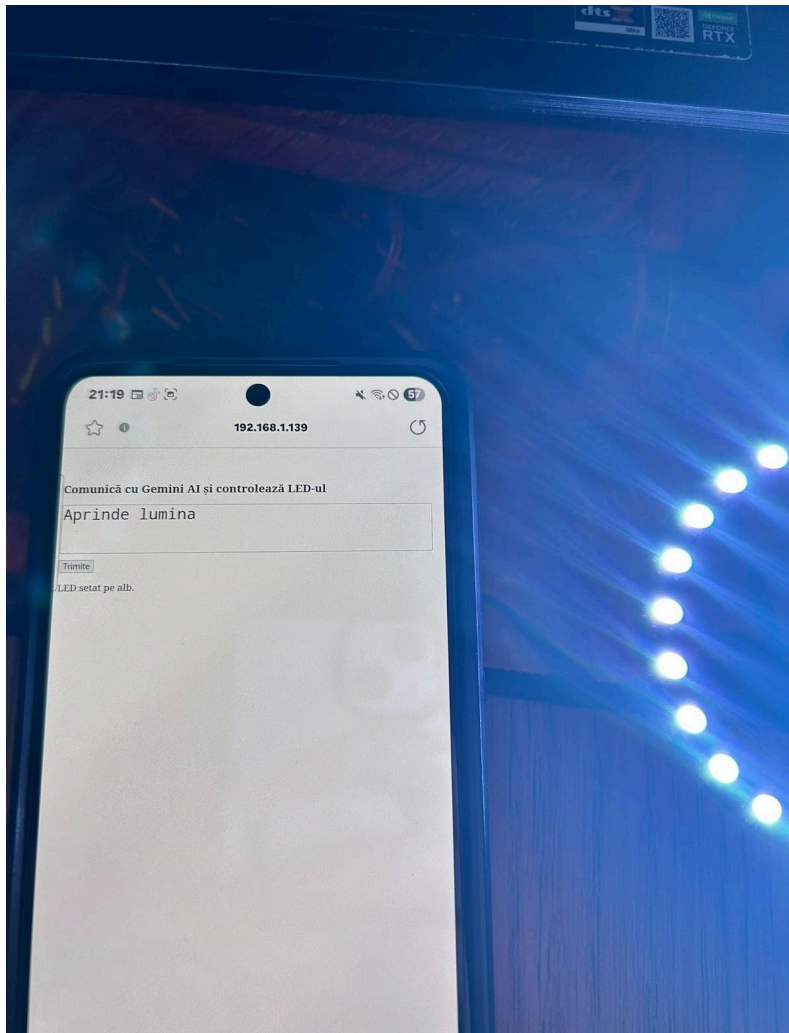
- Conectat ring-ul LED WS2812 (24 LED-uri) la placa ESP32.
- Implementat codul de control LED în Arduino IDE.
- Dezvoltat aplicație web simplă (interfață web locală).
- Adăugat funcționalități: alegere culoare, intensitate, control LED individual, temporizator.
- Testat cu succes controlul LED-urilor din aplicație.
- Interfața aplicației va fi atașată ca imagine în document.

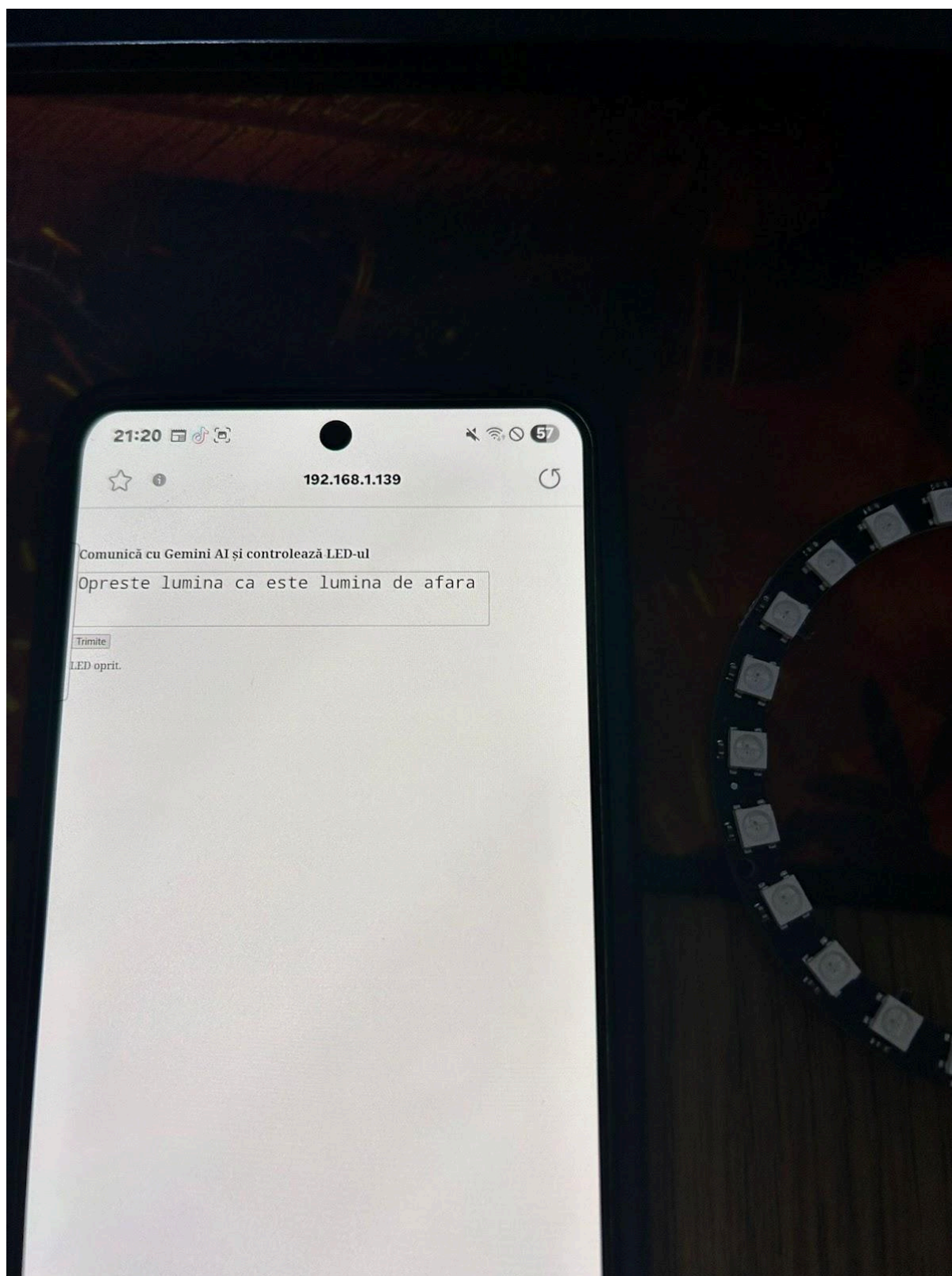


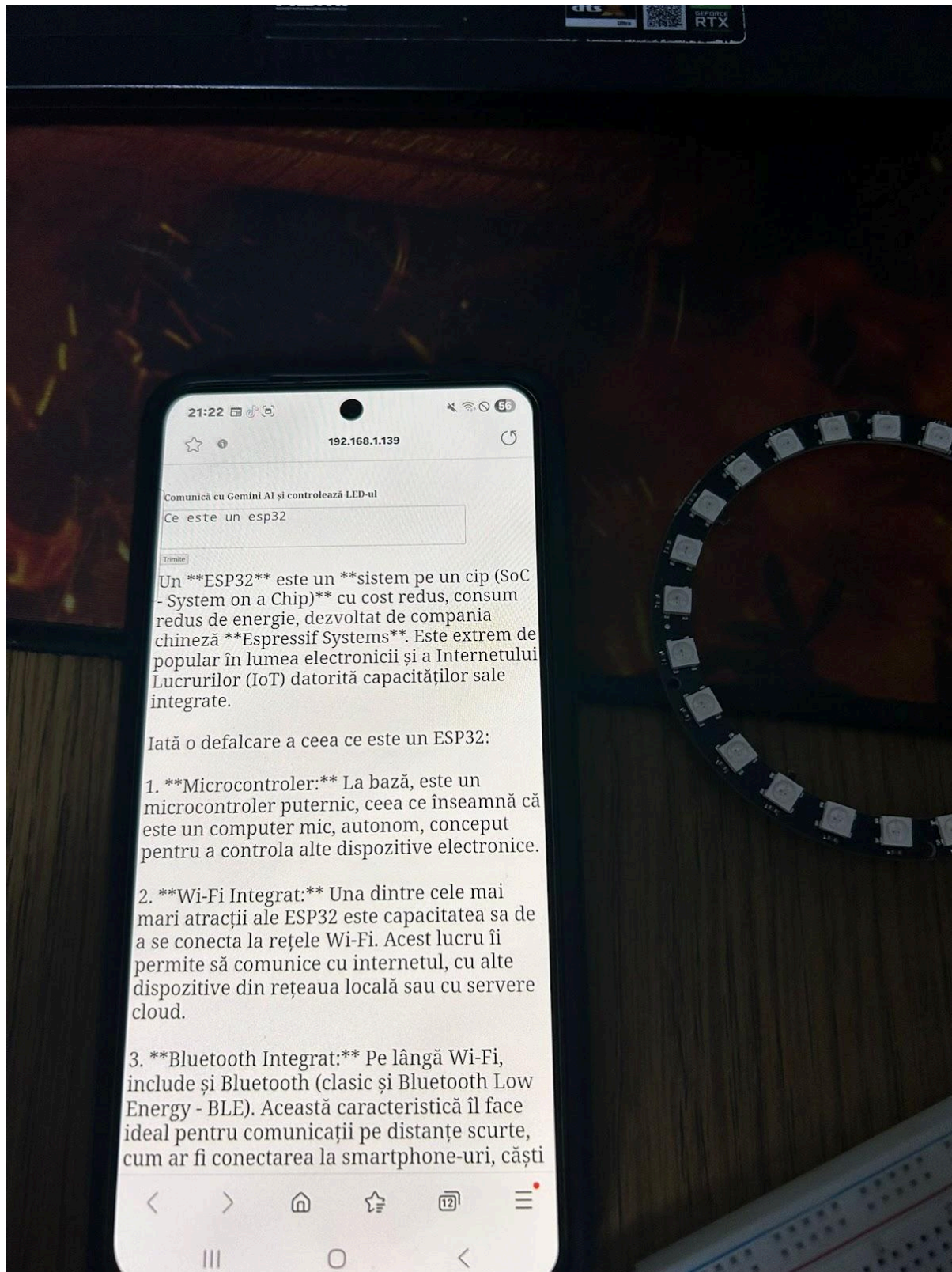


Miercuri - 2 iulie

- Realizat o versiune web funcțională a aplicației.
- Am încercat integrarea cu API-ul OpenAI, dar nu a funcționat conform așteptărilor; am reușit integrarea cu succes folosind Google Gemini AI.
- Implementat logica de trimitere a comenzilor de la AI către ESP32 prin rețea locală.
- Testat cu succes: Gemini răspunde la orice curiozitate și poate acționa LED-ul conform comenzilor (pornire și oprire).
- Realizat capturi de ecran cu aplicația și testele, ce vor fi atașate în document.







Vineri - 4 iulie

- Dezvoltat prima versiune a aplicației Android folosind Android Studio.
- Creat interfața aplicației: ecran principal, câmp pentru comenzi și butoane de control.
- Început integrarea Google Gemini AI pentru interpretarea comenzilor direct în aplicația mobilă.
- Testat parțial funcționalitatea aplicației pe telefon.

Sambata - 5 iulie

- Adăugat funcționalitate de speech-to-text în aplicația Android.
- Reușit integrarea comenzilor vocale cu Google Gemini AI.
- Am configurat și testat comportamentul AI-ului astfel încât să înțeleagă corect comenzi precum: pornire/oprire LED, schimbare culoare și intensitate.
- Urmează ajustări pe partea de design și extinderea funcționalităților aplicației.

Luni - 7 iulie

- Adăugat funcționalitate de răspuns vocal din partea aplicației folosind Gemini AI.
- Optimizat timpul de răspuns dintre comandă și execuție, acum reacționează mai rapid.
- Adăugat un al doilea LED, denumit „bucătărie”, în timp ce ring-ul LED a rămas „baie”.

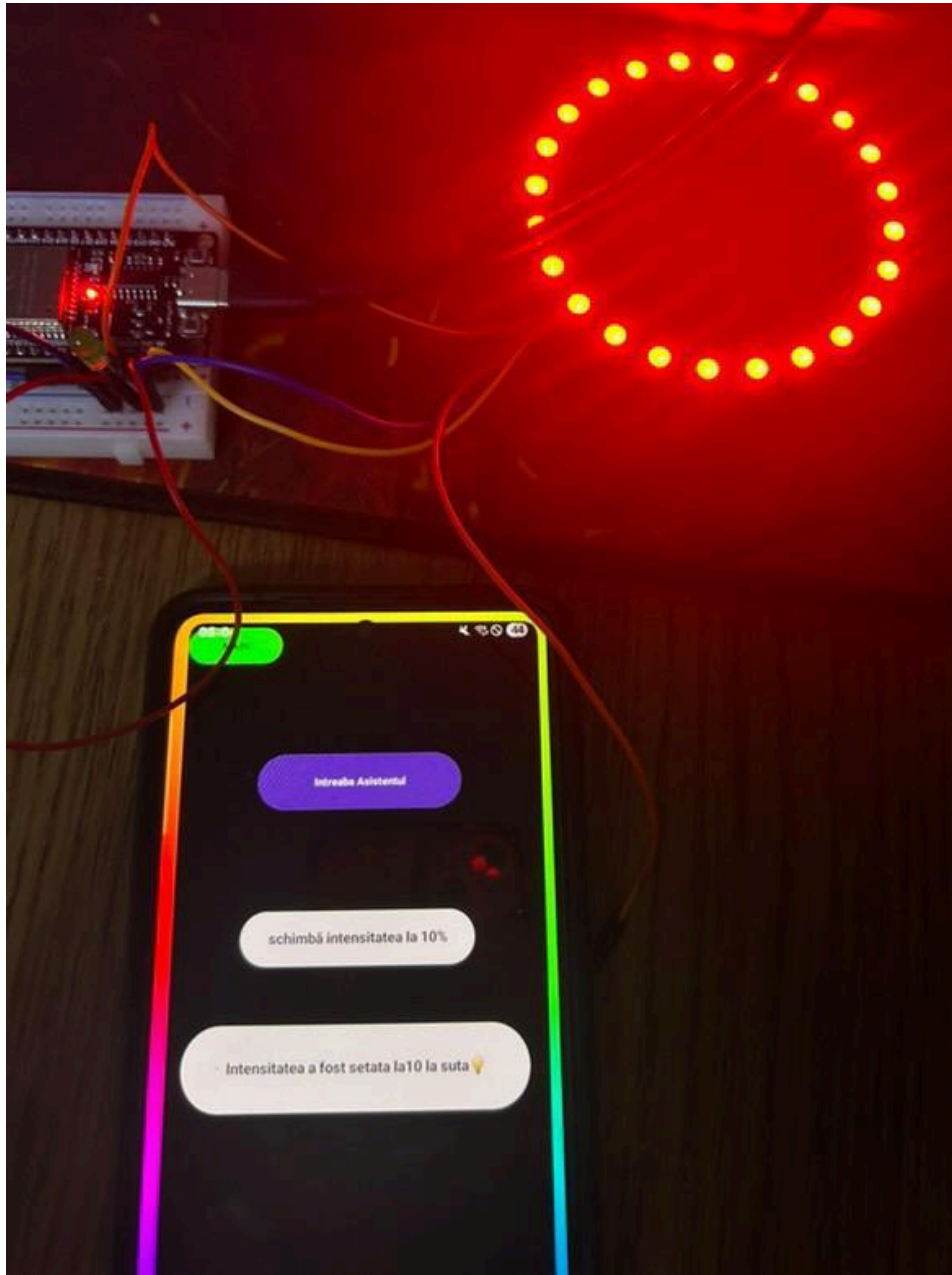
Marti - 8 iulie

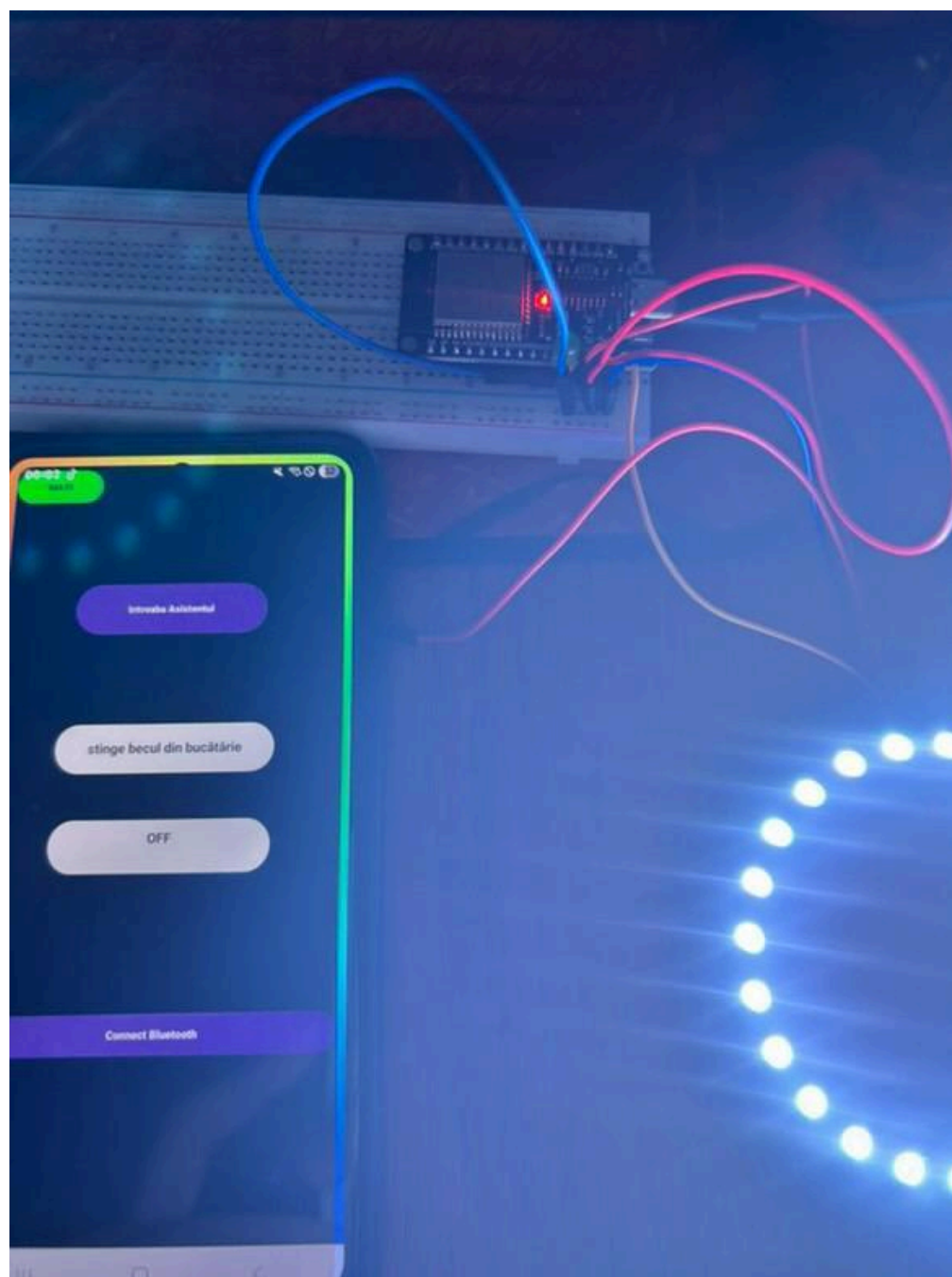
- Am încercat să conectăm aplicația din altă rețea folosind Firebase.

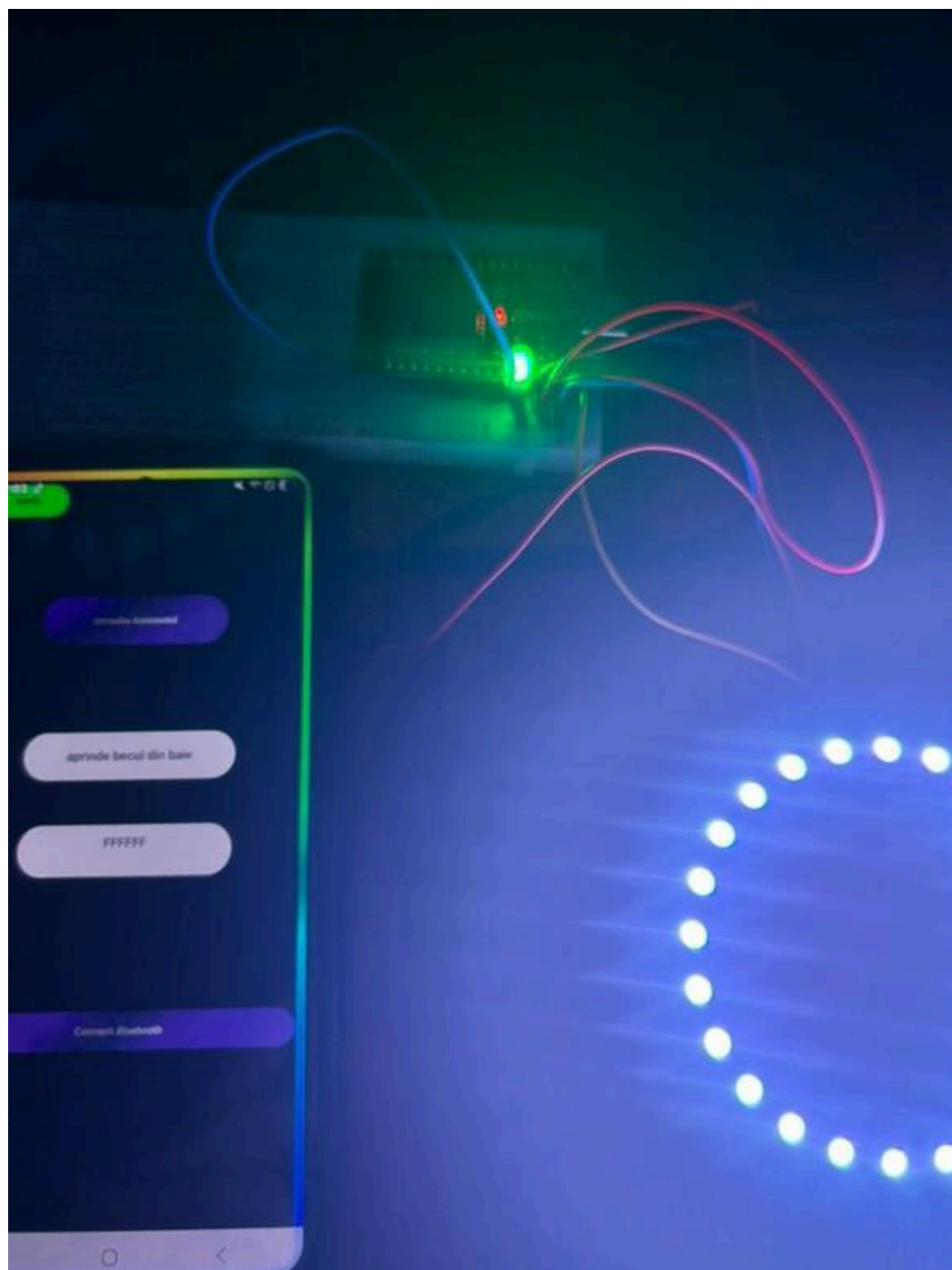
-Am implementat păstrarea intensității setate pentru ring-ul LED indiferent de schimbarea culorii (ex: intensitate 10% pe roșu, intensitatea rămâne 10% când schimb culoarea).

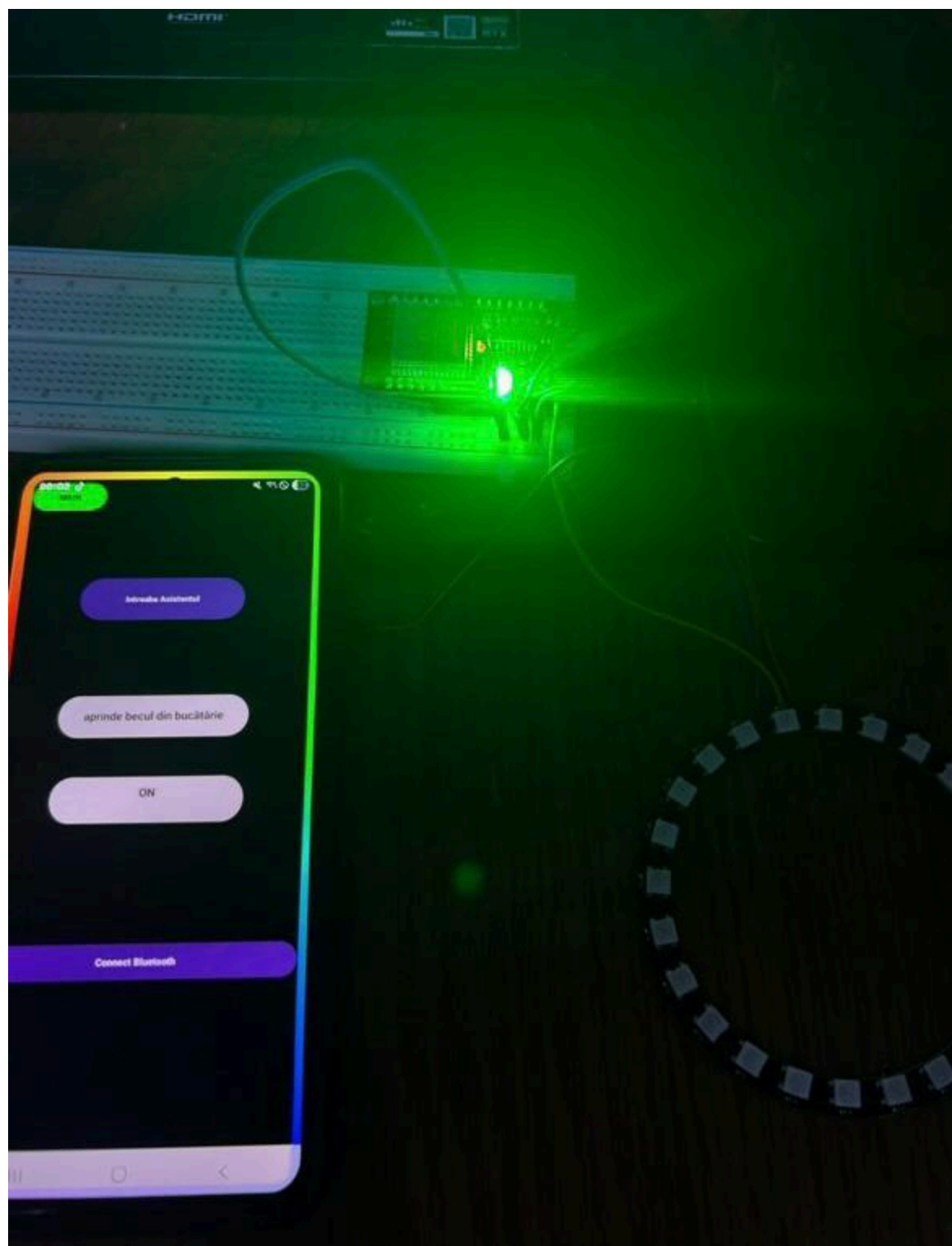
-Am implementat comenzile pentru aprinderea și stingerea LED-urilor din „bucătărie” și „baie”.

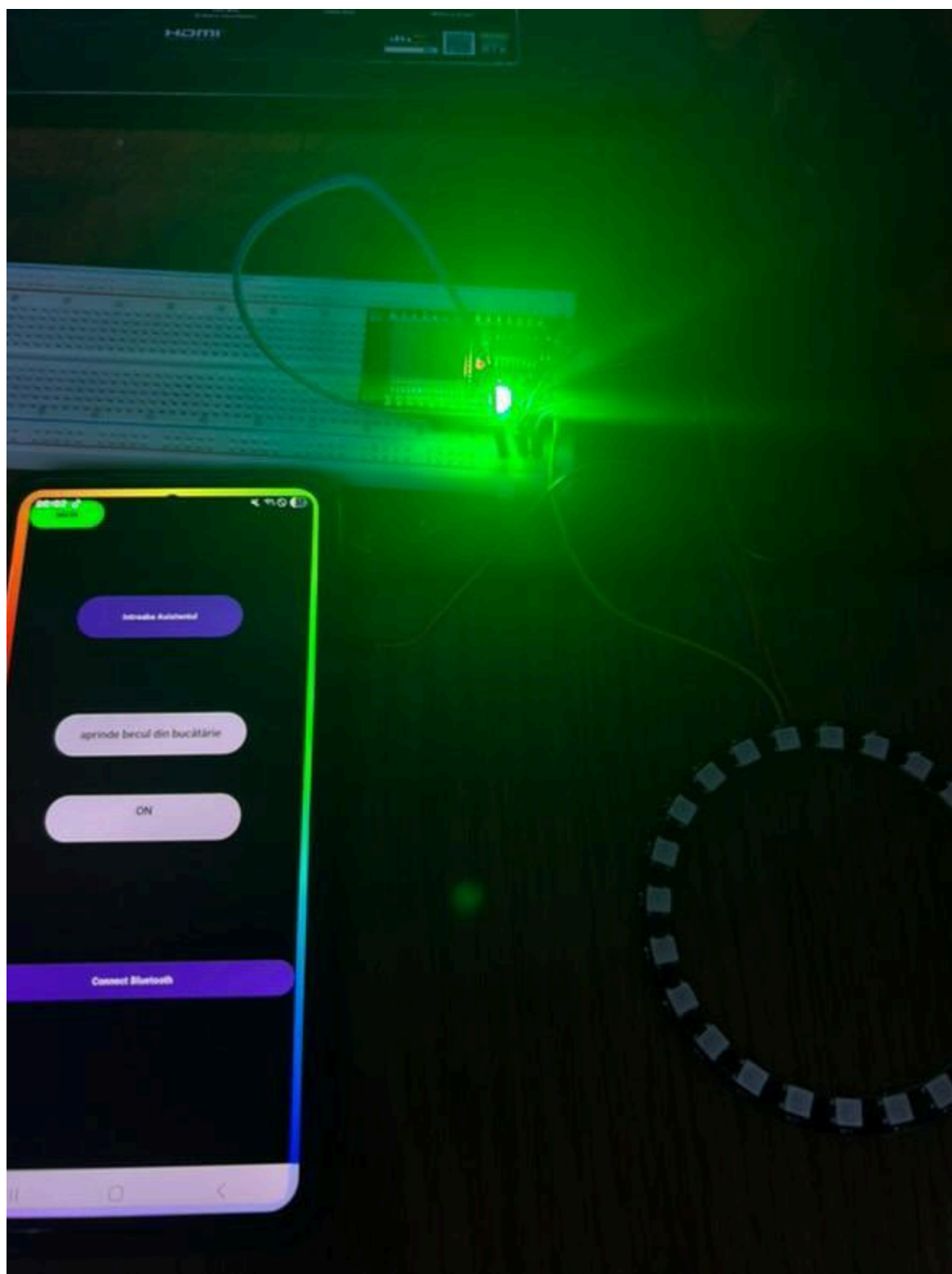
-Atașăm capturi de ecran care demonstrează funcționarea comenzilor.









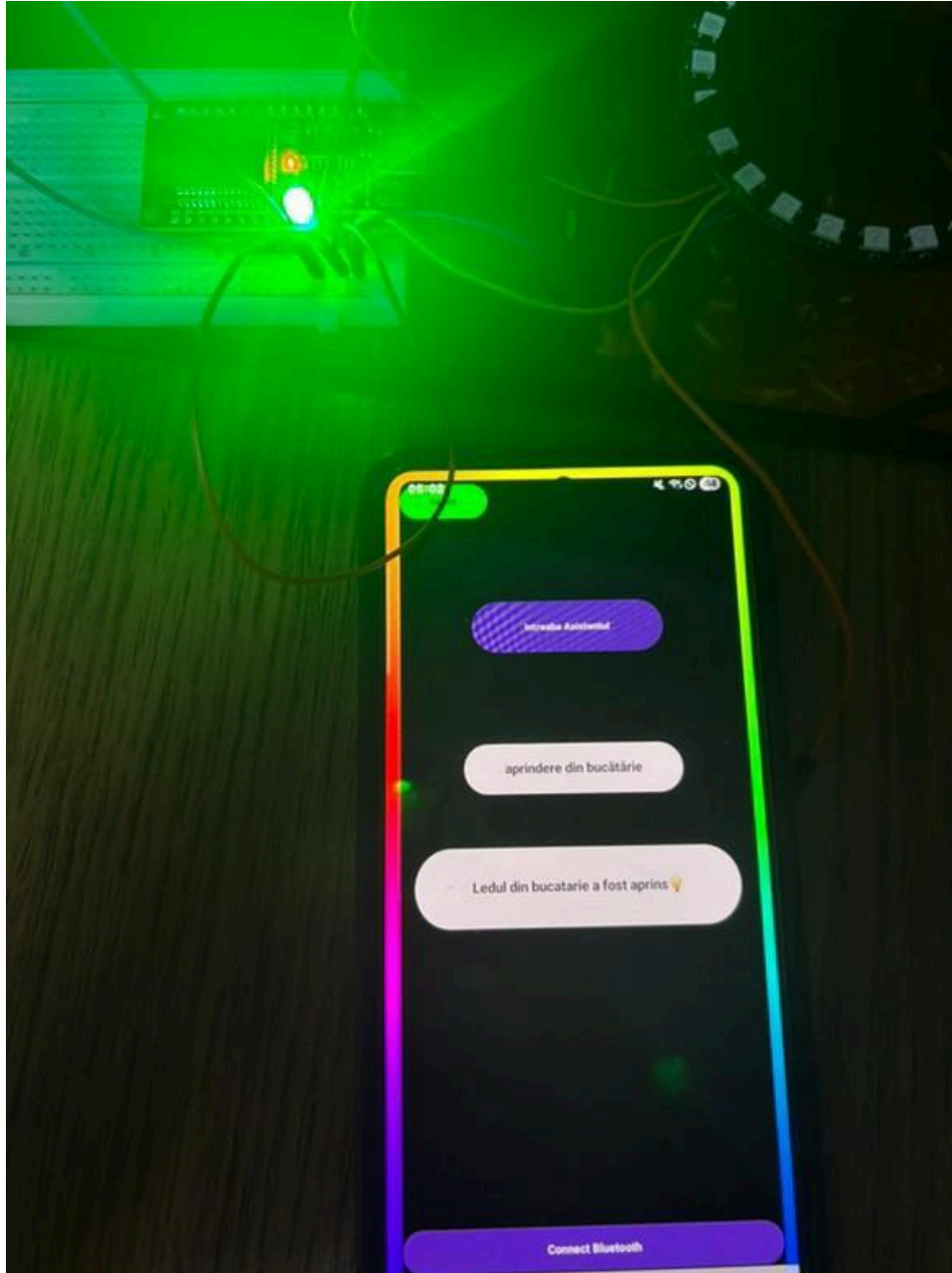


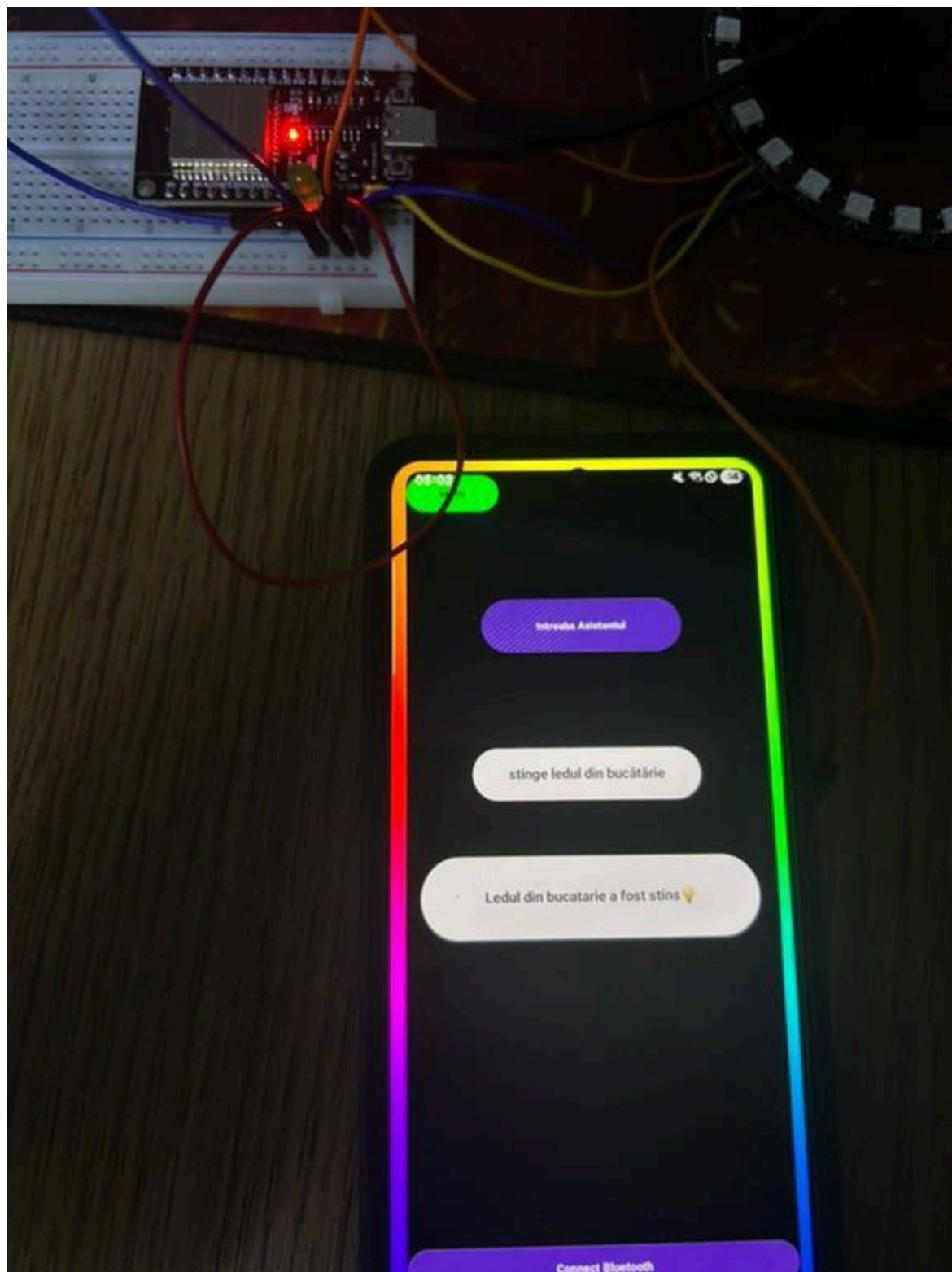
Miercuri - 9 iulie

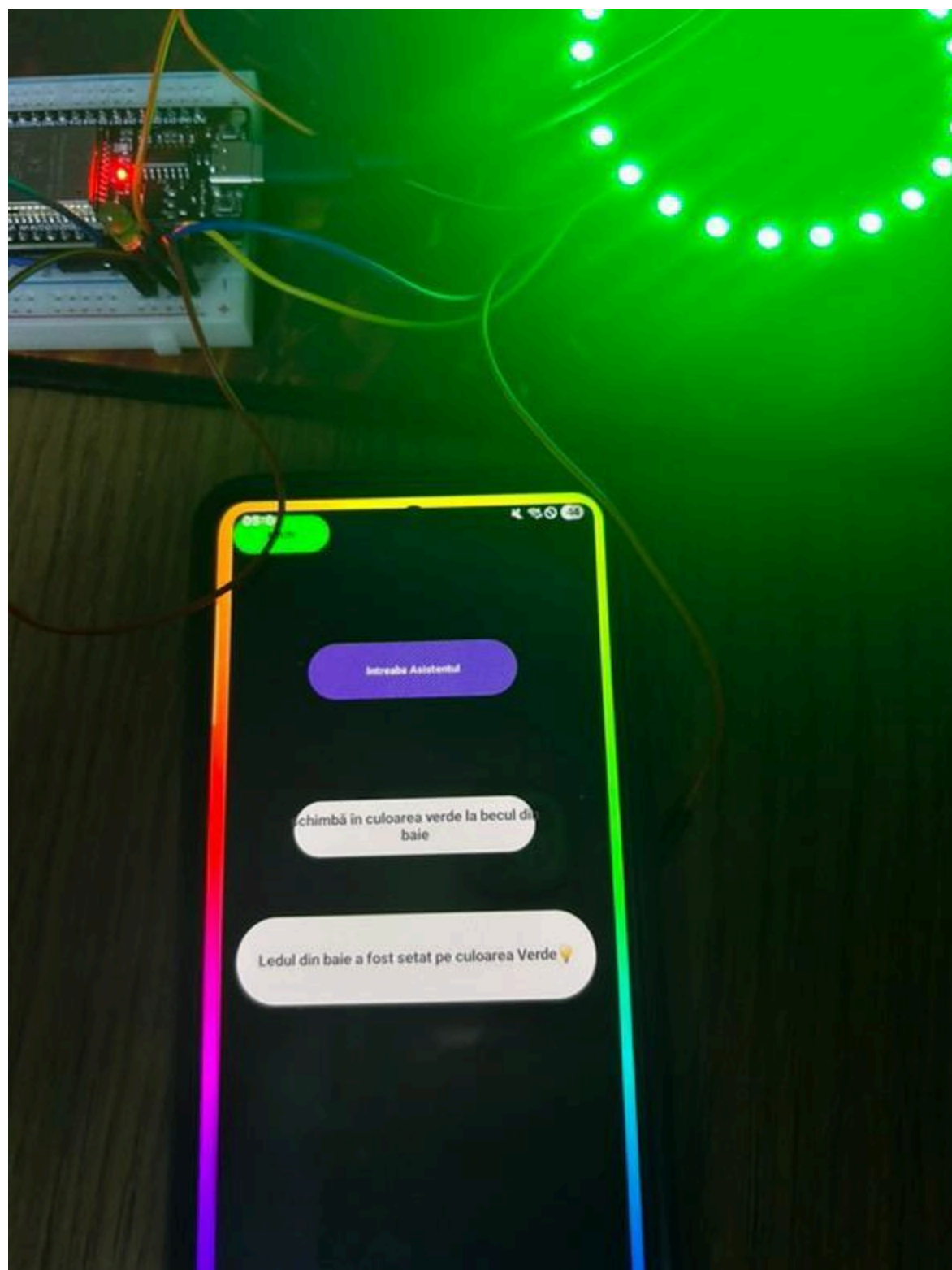
-Am implementat feedback vizual și vocal pentru comenzile executate cu succes și pentru cele eșuate.

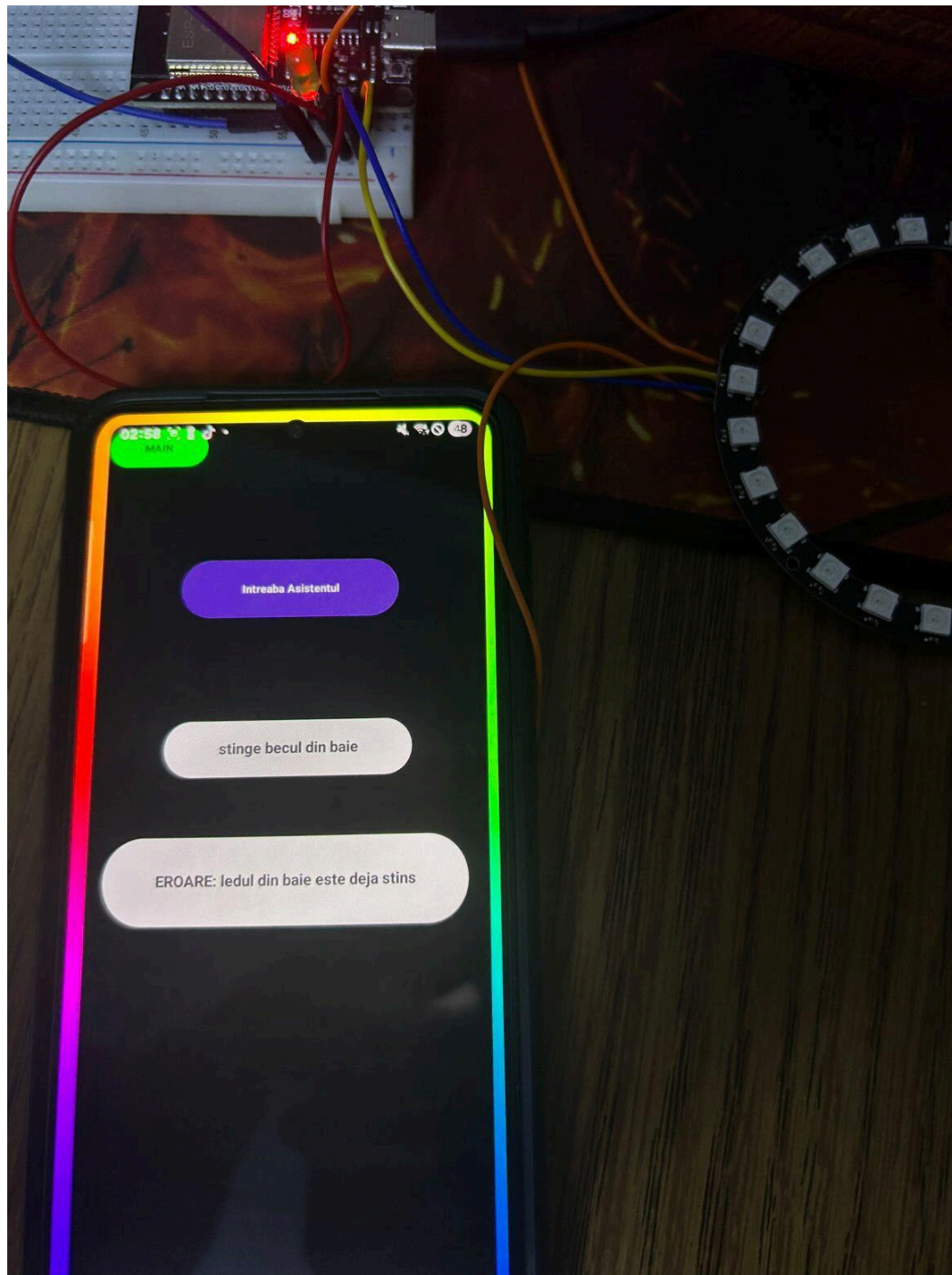
-Aplicația afișează un mesaj text împreună cu o pictogramă animată (ex: „💡 LED bucătărie aprins”).

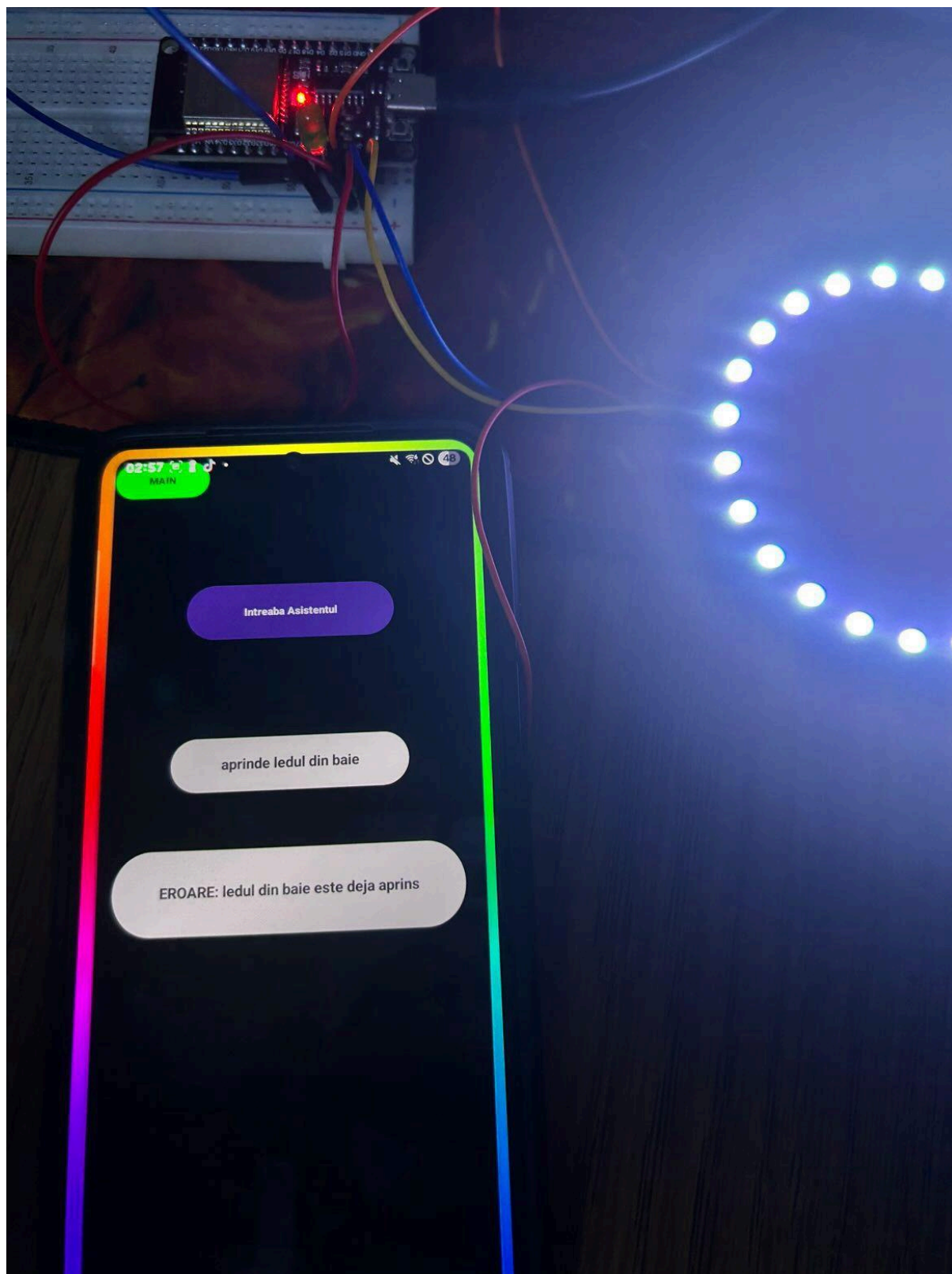
-Capturile de ecran atașate ilustrează acest feedback în aplicație.

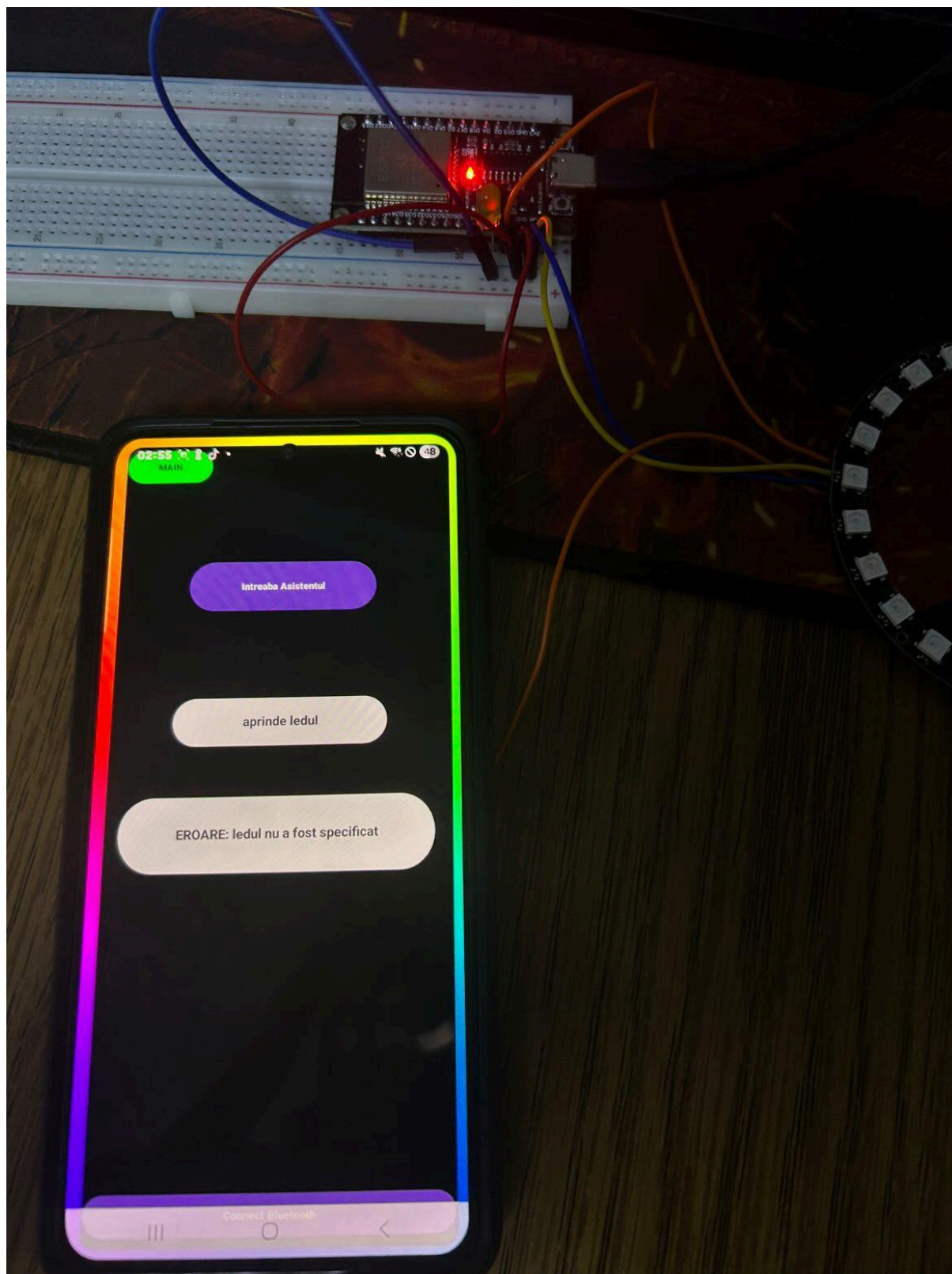










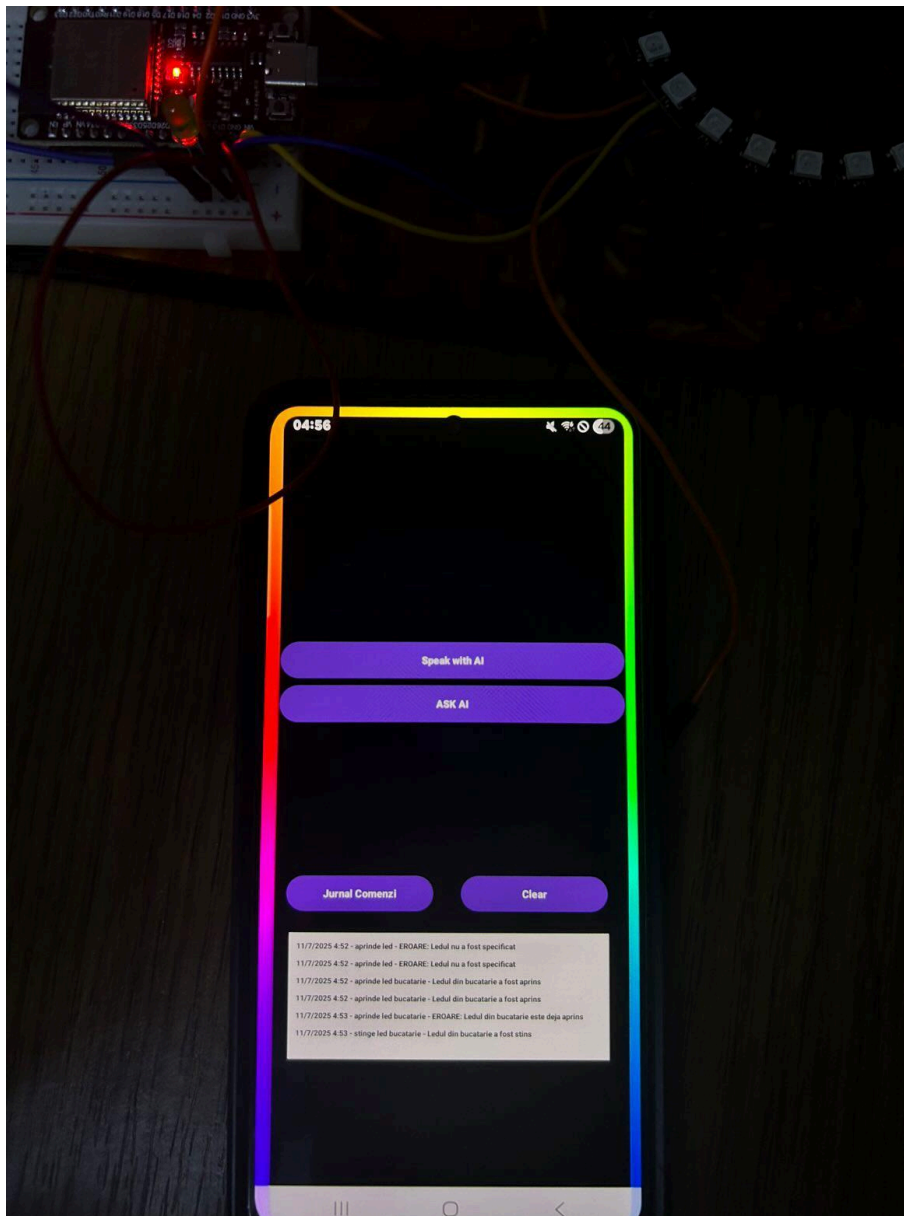


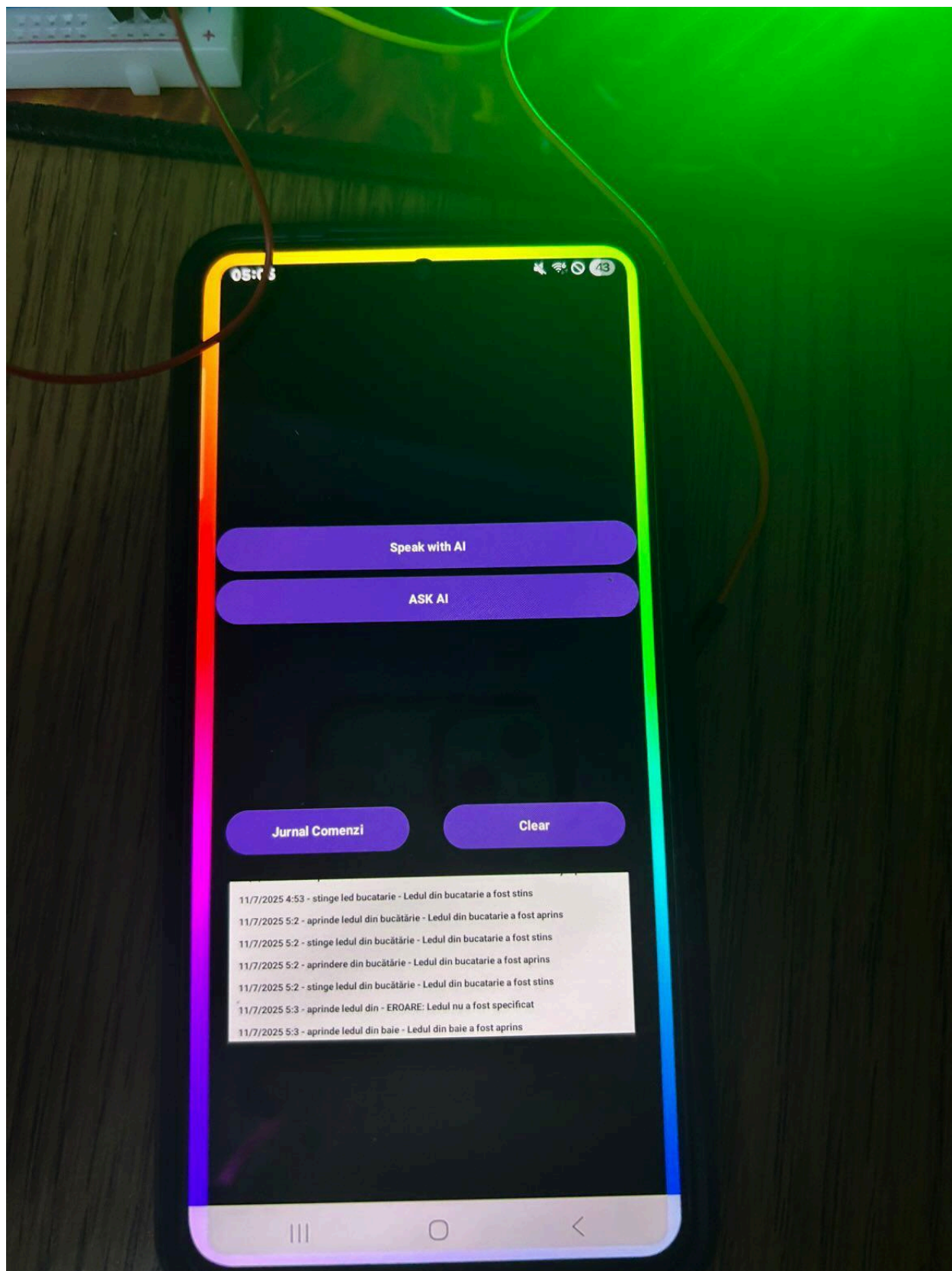
Joi - 10 iulie

-Am implementat o funcționalitate de istoric care salvează comenzile date și răspunsurile oferite de AI.

-Comenzile și răspunsurile sunt stocate local, într-o listă vizibilă în aplicație, cu opțiune de ștergere.

-Am redactat documentația proiectului și am detaliat etapele parcurse pe parcursul practicii într-un repository public pe GitHub.





—
TASKURI pentru urmatoarele 48 ore: