**🟢 1. User Side (Mobile App): Reservation Workflow**

This phase starts when the user interacts with the mobile application to reserve a parking slot. The flow is as follows:

* The user **browses and views** all available parking slots (fetched from the slots table).
* They **select a specific slot** and define a reservation time window (reserved\_from, reserved\_to).
* They input their **vehicle license plate number**.
* The system checks if the car already exists in the cars table:
  + If **not found**, a new record is inserted with the user's plate number and default owner ID.
* The reservation is stored in the reservations table with status active.
* The selected slot is immediately marked as **reserved** by setting is\_occupied = true in the slots table.

✅ This prevents double-booking and ensures real-time availability updates.

**⚙️ 2. Entry & Exit Process (AI + IoT + Image Recognition)**

Once the vehicle physically arrives at the parking entrance or exit:

* The **ESP32 microcontroller** detects motion through its sensor.
* A camera connected to the ESP32 captures a **photo** and uploads it to the appropriate Supabase bucket (entrance or exit).
* A **Python background script** automatically processes the new image:
  + It runs the **YOLO model** to detect the license plate region.
  + The plate is **cropped** from the image.
  + The cropped image is sent to **Roboflow OCR** to extract characters.
  + The result is **converted to Arabic plate format**.
* The plate number is **matched with existing cars** in the database.
* If matched:
  + On **entry**: a new row is inserted in parking\_logs with the current entry\_time.
  + On **exit**: the system looks up the most recent log where exit\_time IS NULL, and updates it with the current exit\_time.

✅ This entire process is automatic and contactless.

**💸 3. Billing Process (Database Trigger Logic)**

Once the exit is logged, two triggers are automatically activated:

* calculate\_duration():
  + Calculates the total time the car spent inside the garage, based on the difference between entry\_time and exit\_time.
* calculate\_amount():
  + Computes the final amount to be paid by multiplying the duration by the predefined rate per hour (e.g. 10 EGP/hour).

The updated record in the parking\_logs table includes:

* Entry time
* Exit time
* Duration (in minutes or hours)
* Final billed amount

✅ This logic is handled entirely within the database using **PostgreSQL triggers**, ensuring consistency and real-time billing without manual intervention.

