Great! Now everything should be nice and clean. Let me get you caught up. Right now, I have a website at event-aid.org that allows users at large events to report emergencies to first responders. (All the website files are attached for your information). When you go to the website, the first landing page is the emergency tab, which has a google map and current location plotted (as long as the user allows it). There are also 3 other informative tabs at the top of the site (info, venue, history). On the emergency tab, there are two buttons for sending help ("Send Help Now" for life/death emergencies and "Send Help" for less urgent matters). When either "Send Help" button is clicked by the user, it triggers the init\_report.php website file in the IONOS Webspace and a unique identifier, an authentication token (for current browser), the current time, the latitude, and longitude, as well as whether the emergency is life-threatening or not-life threatening (depending upon which button is selected) is entered into an IONOS MariaDB 10.11 database called reports (details provided in screenshot). Once these "Send Help" buttons are hit, on the user end, an automatic pop up is generated, indicating that "Help Is On the Way". It also contains a series of drop down information and fill-in-the-box text fields to collect more information about the emergency. These are filled out by the user and then there are two buttons at the bottom, "Submit Info" and "Cancel Request" When the "Submit Info" button is pressed, it calls the update\_report.php website file in the IONOS Webspace and finds the previous row using the unique ID, and the authentication token, and enters any data entered into the reports database. If the user hits the "Cancel Request" Button, the cancel\_report.php website file is called from the IONOS Webspace and finds the previous row using the unique ID, and the authentication token, and "Canceled" Into the status column in the reports database.

Here’s the entire process, described step by step in plain language:

1. **User taps a “Send Help” button.** The browser asks for the user’s GPS coordinates, then sends “life‑threatening” or “non‑life‑threatening” plus the latitude and longitude to your server.
2. **Server creates a new report.** Your PHP script inserts a row into the database, generating a unique report ID (uid) and an authentication token. It returns those two values to the browser.
3. **Popup appears.** With the uid and token tucked into hidden fields, the form shows dropdowns (awake, breathing, etc.) and text boxes (medications, allergies, name, phone).
4. **User fills in details and submits.** The browser sends every answer along with the uid and token back to the server. A second PHP script matches those credentials and updates that same database row with the new information.
5. **Or the user cancels.** If they click “Cancel Request,” the browser sends the uid, token and a “Canceled” status. A third PHP script finds that row and marks its status as canceled.
6. **All data lives in one table.** Each report row holds the uid, token, timestamp, location, emergency type, all form fields, and status—so you can review or export exactly what happened for every alert.

**What you see on the Emergency tab**

* **Nav buttons** (at the top)  
  - Emergency (tab selector)  
  - Info  
  - History  
  - Venue
* **Alert buttons** (inside Emergency)
  1. 🚨 **Send Help Now** – life‑threatening
  2. 🩹 **Send Help** – non‑life‑threatening
* **Google Map**
  1. Shows your current location as a blue marker
  2. Occupies the full width beneath the two alert buttons

**What you see in the popup form**

1. **Heading & blurb**
   * **“HELP IS ON THE WAY”** (in red)
   * Instruction text: “Please stay nearby and fill out the details…”
2. **Eight dropdowns** (two‑column grid)
   * **Awake** (Unknown, Yes, No)
   * **Breathing** (Unknown, Yes, No)
   * **Vomiting** (Unknown, None, Significant, Minor)
   * **Bleeding** (None, Significant, Minor)
   * **Demeanor** (Unknown, Calm, Hysterical)
   * **Danger** (Unknown, No, Self, Others)
   * **Substances** (Unknown, Drugs, Alcohol, Drugs & Alcohol)
   * **Scene** (Unknown, Safe, Unsafe)
3. **Seven text inputs** (stacked)
   * **Current Medications**
   * **Known Allergies**
   * **Pre‑existing Conditions**
   * **Other Information**
   * **Name of Person Needing Aid**
   * **Name of Person Reporting**
   * **Phone #**
4. **Action buttons** (full‑width at bottom)
   * **Submit Info** (green)
   * **Cancel Request** (red)

**What’s hidden in the form (automatically populated)**

* **uid** – your report’s auto‑increment ID
* **authtok** – server‑generated authentication token
* **etype** – “Life Threatening” or “Non‑Life Threatening”
* **lat**, **long** – your GPS coordinates

**What you don’t see (server/database‑only)**

* **time** – timestamp of when the row was created (set by the database)
* **status** – initially “Pending,” updated to “Canceled” if you cancel
* Any other internal columns or debug logs

—  
So in total you have:

* **Visible buttons:** 4 tab selectors + 2 alert buttons + 2 popup action buttons
* **Visible inputs:** 8 dropdowns + 7 text fields
* **Hidden inputs:** 5 fields (uid, authtok, etype, lat, long)
* **Invisible columns:** time and status in the database

Let me know if you need more detail on any particular element!

**Filled at “Send Help” (init\_report.php)**  
These columns are populated on the initial INSERT:

* **uid** – auto‑incremented report ID generated by MariaDB
* **authtok** – random token created by your PHP and stored alongside the row
* **time** – timestamp defaulting to the current time on insert
* **etype** – set to “Life Threatening” or “Non‑Life Threatening” based on which button was clicked
* **lat, long** – the user’s GPS coordinates at the moment they tapped “Send Help”
* **status** – defaults to 'Pending'

At this point, all the remaining columns use their own default (Unknown, None, or NULL), since the user hasn’t seen the popup yet.

**Filled when the user submits the popup form (update\_report.php)**  
All of these are blank or default until the user fills out the form and clicks **Submit Info**:

* **Enum fields**
  + awake
  + breathing
  + vomiting
  + bleeding
  + mood
  + danger
  + substances
  + scene
* **Text fields**
  + meds (Current Medications)
  + allergies (Known Allergies)
  + conditions (Pre‑existing Conditions)
  + oinfo (Other Information)
  + cname (Name of Person Needing Aid)
  + rname (Name of Person Reporting)
  + rphone (Phone #)

**Updated if the user cancels (cancel\_report.php)**

* **status** – changed from 'Pending' to 'Canceled' when “Cancel Request” is clicked
* Nothing else in the row is modified at cancel time

So in summary:

1. **Init** → sets uid, authtok, time, etype, lat, long, status='Pending'.
2. **Submit** → fills in all eight enums (awake, breathing, vomiting, bleeding, mood, danger, substances, scene and seven text columns (meds, allergies, conditions, oinfo, cname, rname, rphone).
3. **Cancel** → flips only status to 'Canceled'.

**1. Front‑end (index.html) ​**

* **Tabs & Layout**
  + Four tabs (Emergency, Info, History, Venue) with your polished styling.
  + On **Emergency**, two full‑width buttons:
    - 🚨 **Send Help Now** → life‑threatening
    - 🩹 **Send Help** → non‑life‑threatening
  + A Google Map (#map-container) that centers on the user’s current position.
* **Popup Form**
  + Triggered only after init\_report.php succeeds.
  + **Hidden fields** (auto‑populated):
    - uid (int auto\_increment)
    - authtok (varchar token)
    - etype (“Life Threatening” or “Non‑Life Threatening”)
    - lat, long (doubles)
  + **User‑visible inputs**:
    - Eight dropdowns (awake, breathing, vomiting, bleeding, mood, danger, substances, scene)
    - Seven text fields (meds, allergies, conditions, oinfo, cname, rname, rphone)
  + Two buttons:
    - **Submit Info** → calls update\_report.php
    - **Cancel Request** → calls cancel\_report.php

**2. Behind the Scenes: AJAX → PHP → Database**

1. **startReport(type)**
   * Grabs GPS via navigator.geolocation.
   * **POST** etype, lat, long to **init\_report.php**.
   * **init\_report.php** INSERTs a row into reports, auto‑generates uid, saves a new authtok, and returns { status:'success', uid, authtok }.
   * JS fills the hidden fields and shows the popup.
2. **Submit Info**
   * FormData now contains **all** fields (hidden + dropdowns + text).
   * **POST** to **update\_report.php**, which does:

sql

CopyEdit

UPDATE reports

SET awake=?, breathing=?, …, rphone=?

WHERE uid=? AND authtok=?

* + Returns JSON { status:'success' }, JS hides the popup.

1. **Cancel Request**
   * JS builds FormData with uid, authtok, status='Canceled'.
   * **POST** to **cancel\_report.php**, which does:

sql

CopyEdit

UPDATE reports

SET status='Canceled'

WHERE uid=? AND authtok=?

* + Returns JSON { status:'success' }, JS hides the popup.

**3. Database Schema (reports table)**

As shown in phpMyAdmin (you can SHOW CREATE TABLE reports; to confirm), your columns are:

| **Column** | **Type** | **Default** |
| --- | --- | --- |
| uid | INT(4) UNSIGNED ZEROFILL AUTO\_INCREMENT PRIMARY KEY | — |
| authtok | VARCHAR(64) | NULL |
| time | TIMESTAMP | CURRENT\_TIMESTAMP |
| lat | DOUBLE | NULL |
| long | DOUBLE | NULL |
| etype | VARCHAR(50) | NULL |
| awake | ENUM('Unknown','Yes','No') | 'Unknown' |
| breathing | ENUM('Unknown','Yes','No') | 'Unknown' |
| vomiting | ENUM('Unknown','None','Significant','Minor') | 'Unknown' |
| bleeding | ENUM('None','Significant','Minor') | 'None' |
| mood | ENUM('Unknown','Calm','Hysterical') | 'Unknown' |
| substances | ENUM('Unknown','Drugs','Alcohol','Drugs & Alcohol') | 'Unknown' |
| danger | ENUM('Unknown','No','Self','Others') | 'Unknown' |
| scene | ENUM('Unknown','Safe','Unsafe') | 'Unknown' |
| meds | TEXT | NULL |
| allergies | TEXT | NULL |
| conditions | TEXT | NULL |
| oinfo | TEXT | NULL |
| cname | VARCHAR(100) | NULL |
| rname | VARCHAR(100) | NULL |
| rphone | VARCHAR(50) | NULL |
| status | ENUM('Pending','Enroute','On Scene','Egress','Canceled') | 'Pending' |

**4. Connection Details (db\_config.php)**

Your scripts share a single include with these credentials:

php

CopyEdit

<?php

$db = new PDO(

"mysql:host=db5017669720.hosting-data.io;port=3306;dbname=dbs14130702",

"dbu2702584",

"••••••••"

);

// … set error mode, charset, etc.

?>

With all of that mapped out, we’ve fully captured:

* **What the user clicks & fills**
* **What hidden data travels**
* **What your PHP does**
* **Exactly how the row in reports evolves**

Let me know what you’d like to build or optimize next!

Sources

o4-mini-high

**You said:**

can you more succinctly summarize the full operation for me

**ChatGPT said:**

Here’s the end‑to‑end flow in a nutshell:

1. **User clicks “Send Help”** on the Emergency tab.
   * Browser asks for geolocation, then POSTs etype, lat, long to **init\_report.php**.
2. **init\_report.php**
   * INSERTs a new row into reports (auto‑generating uid, storing an authtok),
   * Returns JSON { status:'success', uid, authtok }.
3. **Popup appears** (using your polished UI ​) with hidden fields (uid, authtok, etype, lat, long) plus dropdowns and text inputs for patient details.
4. **Submit Info**
   * FormData (all fields) → **update\_report.php** → runs

sql

CopyEdit

UPDATE reports

SET awake=?, breathing=?, …, rphone=?

WHERE uid=? AND authtok=?

* + Returns { status:'success' }, popup closes.

1. **Cancel Request**
   * POSTs uid, authtok, status='Canceled' to **cancel\_report.php** → runs

sql

CopyEdit

UPDATE reports

SET status='Canceled'

WHERE uid=? AND authtok=?

* + Returns { status:'success' }, popup closes.

1. **Database**
   * One reports table holds every field (from auto‑cols like time to enums/text from the form) so you can query or export it directly.