**📄 Project: Live Location Tracker**

Developed by [**Pandit Programmer**](https://panditprogrammer.com)  
Date: **11/07/2025  
  
Documentation** of your **Live Location Tracker** project built using **PHP, MySQL, JavaScript, jQuery, Bootstrap, and Google Maps API**.

**🔧 OVERVIEW**

This is a **web-based real-time location tracking system**. It uses:

* **Google Maps JavaScript API** to display the map and marker
* **Geolocation API** to track the user's live location
* **PHP + MySQL** to store and manage location history
* **Bootstrap & Alertify.js** for UI components and notifications

**🛠️ SETUP GUIDE**

**✅ Step 1: Create Project File**

* Save the full code in a file called index.php.
* Place it in your project’s root directory.

**✅ Step 2: Create Database Table**

Ensure your MySQL database (trackdb) has the table below:

CREATE TABLE `user\_locations` (

`id` INT(11) NOT NULL AUTO\_INCREMENT,

`coordinates` VARCHAR(255) NOT NULL,

`address` TEXT NOT NULL,

`created\_at` TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP(),

PRIMARY KEY (`id`)

);

**✅ Step 3: Google Maps API Key**

* Replace YOUR\_API\_KEY in the <script src="https://maps.googleapis.com/..."> with your actual **Google Maps API key**.
* Go to [Google Cloud Console](https://console.cloud.google.com/):
  + Restrict your API key to your domain.
  + Enable Maps JavaScript API and Geocoding API.

**🔐 SECURITY WARNING**

* **Database Credentials**: Use limited-permission users for live servers.
* **API Key Exposure**: Always **restrict API keys** by domain and usage scope.

**📦 FILE STRUCTURE**

project-root/

└── index.php ← contains all PHP, HTML, JS

**📋 FUNCTIONALITY BREAKDOWN**

**🧠 PHP Server Logic (Backend)**

**🔹 insert\_location**

* Saves a location (coordinates + address) to the database.

**🔹 get\_logs**

* Retrieves the last 20 saved locations in JSON format.

**🔹 delete\_location**

* Deletes a specific location by its ID.

**🔹 export\_csv**

* Exports all saved logs to a CSV file for download.

**🌍 Frontend Interface (HTML +CSS+ JS)**

**📌 Map Area**

* ID: #map
* Google Map rendered and controlled via JavaScript.

**📌 Control Buttons**

| **Button** | **Functionality** |
| --- | --- |
| 📍 Locate Me | Center map on current marker |
| ⏸️ Pause/Resume | Toggle location tracking |
| 🕓 Show Logs | Opens modal with past locations |
| 📄 Export CSV | Downloads log as CSV |

**📌 Modal: #logsModal**

* Shows the last 20 location entries from the database.

**🔄 JavaScript Functional Breakdown**

**✅ initMap()**

* Called by Google Maps API when loaded.
* Sets default location (India).
* Starts geolocation tracking via startTracking().

**✅ startTracking()**

* Uses navigator.geolocation.watchPosition() to track movement.
* On first location or if moved >10 meters:
  + Calls updateMap() to show new marker.
  + Calls insertLocation() to save the location to DB.

**✅ updateMap(lat, lng)**

* Adds or moves the map marker.
* Zooms in on first use.
* Pans the map to the current position.

**✅ reverseGeocode(lat, lng, callback)**

* Uses Google’s Geocoder to convert coordinates to a human-readable address.
* Calls a callback with the address string.

**✅ insertLocation(lat, lng)**

* Sends POST request to index.php to save:
  + coordinates (lat,lng)
  + address (via reverse geocode)
* Refreshes logs if the modal is open.

**✅ fetchLogs()**

* Calls get\_logs via POST.
* Populates modal (#logsList) with results.
* Provides a delete button for each log.

**✅ deleteLocation(id)**

* Confirms and deletes a location from the database.
* Reloads log list on success.

**✅ toggleTracking()**

* Pauses or resumes geolocation tracking.
* Updates button icon and status.

**📊 Visual Representation**

[User Location] -> [Google Maps Marker] -> [Reverse Geocode]

↓ ↓

[updateMap] [Address String]

↓ ↓

[insertLocation()] ----> [PHP -> DB]

↓

[Fetch Logs (JSON)] <------ fetchLogs()

↓

[Logs Modal]

**🧪 Technical Notes**

**Database Queries (MySQL)**

* INSERT INTO user\_locations
* SELECT \* FROM user\_locations ORDER BY id DESC LIMIT 20
* DELETE FROM user\_locations WHERE id = ?

**PHP Features Used**

* mysqli with prepared statements
* $\_POST for AJAX
* Output buffering for CSV export

**JavaScript Libraries**

* jQuery for AJAX & DOM
* Alertify.js for notifications
* Bootstrap (modal, tooltips, buttons)
* Google Maps JS API

**✅ Best Practices Applied**

* ✔️ **Prepared statements** (SQL injection protection)
* ✔️ **AJAX-based frontend/backend communication**
* ✔️ **User experience**: Status indicators, tooltips, modals
* ✔️ **Code organization**: Logical separation of JS/PHP

**🧪 Testing Checklist**

| **Feature** | **Status** |
| --- | --- |
| Location tracking | ✅ |
| Reverse geocoding | ✅ |
| Insert into DB | ✅ |
| Fetch recent logs | ✅ |
| Delete individual log | ✅ |
| Export CSV | ✅ |
| Mobile-friendly layout | ✅ |

**📌 Final Recommendation**

* 🛡️ **Restrict Google API key** as described.
* 🗄️ Move database credentials to an external config file (for production).
* 🧹 Separate JS and CSS into their own files for better maintainability.
* 📤 Add pagination or date filters to the logs modal for scalability.