# PANDTRACK (Pandemic Tracking System)

\*\*PANDTRACK\*\* is a web-based application built with PHP, MySQL, HTML, CSS, JavaScript, and Bootstrap. It helps administrators, doctors, and patients track pandemic-related data (e.g., patient details, symptoms, contact lists, hotspot areas) in Kerala. The system is designed to simplify and automate day-to-day healthcare workflows and reporting.

---

## Table of Contents

1. [Features](#features)

2. [Tech Stack](#tech-stack)

3. [Prerequisites](#prerequisites)

4. [Installation & Setup](#installation--setup)

5. [Database Configuration](#database-configuration)

6. [Project Structure](#project-structure)

7. [Default Credentials](#default-credentials)

8. [Usage](#usage)

9. [Roles & Permissions](#roles--permissions)

10. [License](#license)

---

## Features

- \*\*Admin\*\*

- Approve/reject new doctor registrations

- Manage users (add, view, update, delete)

- Configure doctor types (add, edit, delete)

- Add/view hotspot areas

- Add/view primary contact lists

- View/update/delete any record in the system

- \*\*Doctor\*\*

- Register, then wait for Admin approval

- Login and add/view patient details (symptoms, medicines, history)

- View list of doctors and consultation schedules

- \*\*Patient\*\*

- Register and login

- View personal consultation history, medicine history, disease details

- Share disease details (symptoms, source, contact list)

- \*\*Other\*\*

- Weekly reports of pandemic cases

- Role‐based access control: each user sees only what their role permits

---

## Tech Stack

- \*\*Language:\*\* PHP, HTML, CSS, JavaScript

- \*\*Framework/Library:\*\* Bootstrap

- \*\*Backend:\*\* MySQL (using WAMP/XAMPP)

- \*\*IDE:\*\* VS Code (or Notepad++)

---

## Prerequisites

1. \*\*Web Server:\*\* WAMP Server (or XAMPP) installed on Windows.

2. \*\*PHP:\*\* Version 7.x or above.

3. \*\*MySQL:\*\* Version 5.7 or above (ensure the `mysqli` extension is enabled).

4. \*\*Browser:\*\* Chrome, Firefox, or any modern browser.

---

## Installation & Setup

1. \*\*Clone or Download\*\*

```bash

git clone https://github.com/YourUsername/PANDTRACK.git

Or download and extract the ZIP into your web server’s document root (e.g., C:\wamp64\www\).

1. **Move Files**
   * Ensure the entire project folder (PANDTRACK/) is inside www/ (for WAMP) or htdocs/ (for XAMPP).
   * The final path should look like:
   * C:\wamp64\www\PANDTRACK\
   * ├─ css\
   * ├─ fonts\
   * ├─ HTML\
   * ├─ images\
   * ├─ includes\
   * ├─ js\
   * ├─ less\
   * ├─ SSLA\
   * ├─ SSLD\
   * ├─ SSLP\
   * ├─ dbconnection.php
   * ├─ footer.php
   * ├─ header.php
   * ├─ header2.php
   * ├─ index.php
   * ├─ login.php
   * ├─ phpcodes-Delete.php
   * ├─ phpcodes-Insert.php
   * ├─ phpcodes-Login.php
   * ├─ phpcodes-Update.php
   * ├─ register.php
   * └─ success.php
2. **Start WAMP/XAMPP**
   * Launch the control panel.
   * Start Apache and MySQL services.
3. **Import Database**
   * Open **phpMyAdmin** (<http://localhost/phpmyadmin>).
   * Create a new database named pantra.
   * Click **Import** → **Choose File** → select pantra.sql (provided in the database/ folder) → **Go**.
   * This will create all necessary tables and seed any default data.
4. **Configure dbconnection.php**
   * Open dbconnection.php in VS Code.
   * Update the MySQL credentials if necessary (default WAMP credentials):
   * <?php
   * // dbconnection.php
   * $host = "localhost";
   * $username = "root";
   * $password = ""; // default is empty for WAMP
   * $database = "pantra";
   * $conn = new mysqli($host, $username, $password, $database);
   * if ($conn->connect\_error) {
   * die("Connection failed: " . $conn->connect\_error);
   * }
   * ?>
   * Save and close.
5. **Access the Application**
   * In your browser, navigate to http://localhost/PANDTRACK/.
   * You should see the landing page

**Database Configuration**

* **Database Name:** pantra

**Project Structure**

PANDTRACK/

│

├─ .vscode/

├─ ajax-load/

├─ css/

├─ fonts/

├─ HTML/

├─ images/

├─ includes/

├─ js/

├─ less/

├─ SSLA/

├─ SSDL/

├─ SSLP/

│

├─ dbconnection.php

├─ footer.php

├─ header.php

├─ header2.php

├─ index.php

├─ login.php

├─ phpcodes-Delete.php

├─ phpcodes-Insert.php

├─ phpcodes-Login.php

├─ phpcodes-Update.php

├─ register.php

└─ success.php

* **css/**: Stylesheets and Bootstrap overrides.
* **js/**: JavaScript files (AJAX calls, form validation, etc.).
* **includes/**: Reusable components (e.g., navigation menus, sidebars).
* **SSLA/, SSLD/, SSLP/**: Submodules/pages for different user roles or modules.
* **dbconnection.php**: Establishes a connection to MySQL.
* **phpcodes-\*.php**: CRUD operations for different user actions (Insert, Update, Delete, Login).
* **register.php**: User registration form (doctor/patient).
* **login.php**: Login form for all user roles.
* **index.php**: Landing page/dashboard (role‐based redirect).
* **header.php**, **footer.php**, **header2.php**: Common header/footer templates.

**Default Credentials**

* **Admin**
  + Email: admin@gmail.com
  + Password: admin123

*(After first login, Admin can create additional doctor or patient accounts.)*

**Usage**

1. **Admin Login**
   * Go to http://localhost/PANDTRACK/login.php
   * Enter Admin credentials (admin@gmail.com / admin123).
   * After logging in, the Admin dashboard shows options to manage users, hotspots, contacts, and doctor types.
2. **Doctor Registration & Approval**
   * A new doctor visits <http://localhost/PANDTRACK/login.php>.
   * Admin can add doctors their number is password
3. **Patient Registration & Usage**
   * A new patient visits register.php, signs up, and logs in immediately (no approval needed).
   * Patient dashboard allows viewing consultation history, medicine history, and adding source/contact details.
4. **Adding Hotspot Areas & Contacts (Admin)**
   * Navigate to **Hotspot Management** → Add/Edit/Delete hotspot areas.
   * Navigate to **Primary Contacts** → Add/Edit/Delete primary contact entries.
5. **Doctor Dashboard**
   * doctors log in and are redirected to their dashboard.
   * They can add a new patient, update symptoms, assign medicines, and view history.
   * “Doctor List” page displays all doctors, their departments, and consultation schedules.
6. **Patient Dashboard**
   * After login, patients see their “Consultation History”, “Medicine History”, and “Disease Details”.
   * They can update personal profile and share any new symptoms or contact lists.

**Roles & Permissions**

1. **Admin**
   * Full access: Manage users (doctors/patients), hotspots, contacts, doctor types.
   * View or delete any record in the system.
2. **Doctor**
   * Add/view patient details, symptoms, medication history.
   * View list of all doctors (including consultation schedules).
3. **Patient**
   * View personal consultation/medicine history.
   * Share disease/symptom details.