

## Backend Setup Document

This will guide us as we start developing the backend part of the project. The backend will handle all data processes, user authentication, and connections between the system and the database. We decided to use Laravel with PHP since it's beginner-friendly and already has tools for routing, validation, and authentication.

### Framework and Language

- The backend will be developed using Laravel version 12 and PHP version 8.2 or higher. Laravel follows the MVC (Model-View-Controller) structure, which will help us organize our files properly and make the backend easier to maintain. It's also good for handling API's, database queries, and validation.

#### 1. Server and Environment

- ❖ For local development, we'll use XAMPP. This setup will let us run and test the backend on our local machines before deployment.
- ❖ When the project is ready to go or to be deployed, we plan to host it using Hostinger since it supports both PHP and MySQL.
- ❖ Laravel also uses a ".env" file where we'll store important configuration details like the database name, username, and password. This helps us keep sensitive information secure and easy to change when we move to production.

#### 2. Database Setup

- ❖ Our backend will use MySQL as the database. This will handle all user information, lost and found reports, and other related data.

#### 3. API and Authentication

- ❖ The backend will follow a RESTful API structure for communication between the frontend and backend.
- ❖ We'll use token-based authentication through Laravel Sanctum. After a user log in, the system will generate a unique token that will be used to verify their identity for every API request.
- ❖ API endpoint is declared in ./recorever-api-docs.

#### 4. Data Flow

- ❖ Here's how the backend will process requests:
- ❖ The frontend sends an HTTP request (like submit report) to the backend API.
- ❖ The backend validates the data and processes it.
- ❖ If the user is authenticated, their token is verified before the request is accepted.
- ❖ The backend communicates with the MySQL database to save or retrieve data.
- ❖ The backend returns a JSON response to the frontend.

## 5. Security and Validation

For security, we'll implement:

- ❖ Token-based authentication using Laravel Sanctum
- ❖ Server-side validation for all inputs
- ❖ Password hashing
- ❖ Protected routes for authenticated users only

## 6. File and Media Handling

Uploaded file photos will be stored in Laravel's `/storage/app/public`

In summary, our backend setup will use Laravel 12, PHP, and MySQL following a token-based authentication system using Laravel Sanctum. The backend will manage all data operations, handle secure API communication, and ensure proper validation and access control. We'll use Hostinger for hosting and XAMPP for local development, this plan will guide us as we start working on the backend throughout our sprint.

It serves as our guide before we perform the actual technical setup and coding in Laravel.