**Entry-Log System Documentation**

**System Overview**

The Entry-Log System is a web-based attendance and activity tracking system designed for two user roles: **Field Workers** and **Admin**. It manages sign-ins/outs, field worker information, documentation, and department-specific attendance rules. The system ensures data integrity, enforces controlled access, and streamlines supervision of temporary field-based personnel.

**System Users**

**1. Field Worker**

* Can log in via the common login page.
* On first login:
  + Prompted with a popup to fill in personal details:
    - Full Name
    - Phone Number
    - Department (Dropdown - admin-defined)
    - College/University Name
    - Start Date and End Date
    - Email
    - Group (A or B - pre-assigned or selectable)
  + Forced to change default password.
* After first-time setup:
  + Home page displays all profile details (editable).
  + Options:
    - **Check-In / Check-Out** (based on department time limits).
    - Daily announcements display.
    - Download:
      * Certificate Template
      * Registration Document (Word file).
    - Must before deadline Upload filled documents as **PDF**, named as: Fullname Certificate.pdf and Fullname registration.pdf

**2. Admin**

* Logs in through same login page.
* Redirected to admin dashboard with left-side navigation:

**i. Attendance Details**

* Table displaying all attendance records:
  + Columns: ID, Full Name, Department, Date, Check-In Time, Check-Out Time, Total Hours, Action (delete).
* Features:
  + Filter by Date and Department (with "All Departments" option).
  + Multi-select for bulk delete.

**ii. Add Field Workers**

* Form:
  + Full Name (in capital letters).
  + Default Password: ttcl1234.
  + Support batch addition.
* Table:
  + Columns: Full Name, Department, College/University, Email, Start Date, End Date, Status, Actions (edit/delete).
  + Features: Search, Sort, Multi-select delete.

**iii. Departments**

* Form:
  + Department Name
  + Check-In Limit (Time)
  + Check-Out Limit (Time)
* Table:
  + Columns: ID, Name, Check-In Limit, Check-Out Limit, Actions (edit/delete).

**iv. Announcements**

* Form:
  + Department (select)
  + Date to Display
  + Content
* Table:
  + List of Announcements with edit/delete actions.

**v. Documentation**

* Form:
  + Upload Word versions of Certificate and Registration Document
  + Set Submission Deadline
* Table:
  + Columns: ID, Full Name, Comment (Uploaded/Not Uploaded), Download Option
  + Bulk Actions: Download All, Delete Selected

**Special Features**

**Check-In/Check-Out Logic**

* Field Workers:
  + Can Check-In once per day.
  + Cannot Check-Out unless they previously Checked-In.
  + Optional Rule: Cannot Check-In next day if previous day’s Check-Out is missing.
* All actions are timestamped (Date & Time).

**Location/IP Tracking**

* Verify login/check-in from designated office location via:
  + IP filtering.
  + GPS (for mobile-based access).

**Group Alternation Logic**

* Each department splits Field Workers into Group A and Group B.
* Groups attend work on alternating days.
* Weekends (Saturday & Sunday) are excluded from the attendance cycle.

**System Implementation Steps**

**1. Project Setup**

* Language and Technologies:
  + Frontend: HTML, CSS, JavaScript (jQuery)
  + Backend: PHP (raw PHP)
  + Database: MySQL
* Tools:
  + Backend: PHP (raw PHP)
  + Frontend: HTML, CSS, JavaScript (jQuery)
  + Database: MySQL

**2. Database Design**

Create tables for:

* users (user credentials and roles)
* field\_worker\_profiles
* departments
* attendance
* announcements
* documents
* groups

**3. Authentication System**

* Shared login page.
* Role-based redirect after login.

**4. Field Worker Workflow**

* First login check:
  + If profile incomplete, show modal for data collection.
  + Force password change.
* Home dashboard:
  + Display profile.
  + Buttons for Check-In, Check-Out (only one allowed daily).
  + Announcement section.
  + Document download/upload section.

**5. Admin Dashboard Implementation**

* Sidebar with sections (Attendance, Field Workers, Departments, Announcements, Documentation).
* Admin CRUD functionality for each section.
* DataTables or equivalent for tables (search, filter, sort, bulk actions).

**6. Document Handling**

* Upload system for Word templates.
* Field Worker uploads PDF versions.
* File naming convention check (FullName Certificate.pdf).

**7. Group Logic Implementation**

* Store groups A/B in DB.
* Apply alternate day logic in Check-In eligibility.
* Exclude weekends using date check (date('N') > 5).

**8. Security Features**

* Password hashing.
* Input validation & sanitization.
* File type and size check on upload.
* Access control by user roles.

**9. Location/IP Control**

* Use $\_SERVER['REMOTE\_ADDR'] to check IP.
* Use browser Geolocation API if needed.

**Deployment Considerations**

* Host on LAMP stack or cloud.
* Use .env for sensitive configs.
* Regular backups for uploaded docs and attendance logs.

**Final Notes**

* Make sure to test edge cases (e.g., missed check-out).
* Ensure mobile responsiveness for field users.
* Consider exporting attendance reports as CSV for admin usage.

**DATABASE SCHEMA.**

-- 1. Users

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(100) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL,

role ENUM('admin', 'field\_worker') NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- 2. Field Worker Profiles

CREATE TABLE field\_worker\_profiles (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT UNIQUE NOT NULL,

full\_name VARCHAR(150) NOT NULL,

phone VARCHAR(20) NOT NULL,

department\_id INT NOT NULL,

college\_name VARCHAR(150),

start\_date DATE NOT NULL,

end\_date DATE NOT NULL,

email VARCHAR(100) UNIQUE,

group\_name ENUM('A', 'B') NOT NULL,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES users(id),

FOREIGN KEY (department\_id) REFERENCES departments(id)

);

-- 3. Departments

CREATE TABLE departments (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) UNIQUE NOT NULL,

check\_in\_limit TIME NOT NULL,

check\_out\_limit TIME NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- 4. Attendance

CREATE TABLE attendance (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT NOT NULL,

department\_id INT NOT NULL,

date DATE NOT NULL,

check\_in\_time DATETIME DEFAULT NULL,

check\_out\_time DATETIME DEFAULT NULL,

total\_hours FLOAT GENERATED ALWAYS AS (

TIMESTAMPDIFF(MINUTE, check\_in\_time, check\_out\_time) / 60

) STORED,

FOREIGN KEY (user\_id) REFERENCES users(id),

FOREIGN KEY (department\_id) REFERENCES departments(id)

);

-- 5. Announcements

CREATE TABLE announcements (

id INT AUTO\_INCREMENT PRIMARY KEY,

department\_id INT NOT NULL,

content TEXT NOT NULL,

visible\_date DATE NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (department\_id) REFERENCES departments(id)

);

-- 6. Documents (Submitted by Field Workers)

CREATE TABLE documents (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT NOT NULL,

type ENUM('certificate', 'registration') NOT NULL,

file\_name VARCHAR(255) NOT NULL,

uploaded\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

status ENUM('uploaded', 'missing') DEFAULT 'missing',

FOREIGN KEY (user\_id) REFERENCES users(id)

);

-- 7. System Documents (Uploaded by Admin)

CREATE TABLE system\_documents (

id INT AUTO\_INCREMENT PRIMARY KEY,

type ENUM('certificate', 'registration') NOT NULL,

file\_path VARCHAR(255) NOT NULL,

deadline DATE NOT NULL,

uploaded\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- 8. IP Logs

CREATE TABLE ip\_logs (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT NOT NULL,

action ENUM('login', 'check\_in', 'check\_out') NOT NULL,

ip\_address VARCHAR(45) NOT NULL,

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES users(id)

);

DIRECTORIES

**/entry-log-system**

**│**

**├── /public # Public assets and entry point**

**│ ├── index.php # Main entry point**

**│ ├── login.php # Common login page**

**│ ├── logout.php # Logout handler**

**│ ├── /assets**

**│ │ ├── /css**

**│ │ ├── /js**

**│ │ └── /images**

**│**

**├── /frontend # Frontend UI logic**

**│ ├── /field\_worker**

**│ │ ├── dashboard.php**

**│ │ ├── checkin.js**

**│ │ ├── profile.js**

**│ │ └── documents.js**

**│ ├── /admin**

**│ │ ├── dashboard.php**

**│ │ ├── workers.js**

**│ │ ├── attendance.js**

**│ │ ├── departments.js**

**│ │ ├── announcements.js**

**│ │ └── documents.js**

**│**

**├── /backend # Backend logic (PHP only)**

**│ ├── /controllers # Handle requests**

**│ │ ├── AuthController.php**

**│ │ ├── FieldWorkerController.php**

**│ │ ├── AdminController.php**

**│ │ ├── AttendanceController.php**

**│ │ └── DocumentController.php**

**│ ├── /models # DB logic per table**

**│ │ ├── User.php**

**│ │ ├── FieldWorker.php**

**│ │ ├── Department.php**

**│ │ ├── Attendance.php**

**│ │ ├── Announcement.php**

**│ │ ├── Document.php**

**│ │ └── IPLog.php**

**│ ├── /services # Helper services (file uploads, auth, etc)**

**│ │ └── UploadService.php**

**│ └── /middlewares # Auth and access control**

**│ └── AuthMiddleware.php**

**└──GroupService.php**

**│**

**├── /database**

**│ ├── schema.sql # SQL create table scripts**

**│ ├── seed.sql # Dummy/test data**

**│**

**├── /config**

**│ └── db.php # DB connection config**

**│**

**└── README.md**

**End of Documentation**