# Documentation for Hospital Management System (CodeIgniter)

Project Name: Hospital Management System Platform: PHP Framework: CodeIgniter Used Programming Languages: HTML, HTML5, CSS, CSS3, Bootstrap, JavaScript Database: MySQL Software Tools: Notepad++, Sublime Text, Adobe Dreamweaver Supported Browsers: Internet Explorer, Opera Mini, Google Chrome, Mozilla Firefox Supported Operating Systems: Windows 7, Windows 8, Windows 10, Linux, Mac PHP Software Tools Requirement:

- XAMPP
- LAMP
- WAMP
- Apache
- MySQL

## How to Configure (Run) the Project

- 1. **Download the Project Zip File** Download the zip file containing the source code of the Hospital Management System.
- 2. Unzip and Place in Root Directory Unzip the project file and place the extracted files in the root directory of your web server (e.g., htdocs in XAMPP).
- 3. Create and Import Database
  - Open phpMyAdmin (<u>http://localhost/phpmyadmin</u>).
  - Create a new database. Example: hospital\_db.
  - Import the SQL file into the database. Navigate to Import in phpMyAdmin, choose the SQL file provided, and execute.

## 4. Configure Base URL

Open application/config/config.php and set the \$config['base\_url'] to your project URL. Example: http://localhost/hospital\_management/.

## 5. Set Up Database Configuration

Open application/config/database.php and update the database configuration with your database credentials.

- 6. Access the Application
  - User Login URL: http://localhost/hospital\_management/user
  - Admin Login URL: http://localhost/hospital\_management/admin
- 7. Login Credentials
  - User Login:
    - Username: user

- Password: user
- Admin Panel Login:
  - Username: Admin
  - Password: Admin

# **Project Overview**

0

The Hospital Management System (HMS) is designed to manage hospital operations efficiently. It includes features for patient management, doctor management, appointment scheduling, and more.

## **Key Features**

- 1. Patient Management
  - Register and manage patient information.
  - View patient history and appointments.
- 2. Doctor Management
  - Manage doctor profiles and schedules.
  - Assign patients to doctors.
- 3. Appointment Scheduling
  - Schedule and manage patient appointments.
  - View and edit existing appointments.
- 4. Billing Management
  - Handle patient billing and payments.
  - Generate invoices and payment receipts.
- 5. Reports and Analytics
  - Generate reports on hospital operations, patient statistics, and financials.

# **UI/UX Wireframes**

## Homepage

• Overview of hospital services, patient login, and admin login.

## **Patient Dashboard**

• Patient profile, appointment scheduling, and medical history.

## **Doctor Dashboard**

• Doctor profile, schedule management, and patient list.

## **Admin Dashboard**

• Comprehensive control over hospital operations, including user management, billing, and reports.

## **Error Handling Guidelines**

#### 1. Validation Errors

• Display user-friendly messages for form validation errors (e.g., "Please enter a valid email address").

#### 2. Database Errors

- Show generic error messages and log detailed errors for debugging purposes.
- 3. 404 Errors
  - Display a custom 404 error page indicating that the requested resource was not found.

#### 4. 500 Internal Server Errors

• Display a generic error message and log detailed errors for further analysis.

## **Business Logic**

#### 1. Patient Registration

- Validate patient information.
- Store patient details in the database.

#### 2. Appointment Scheduling

- Check doctor availability.
- Schedule appointments and update doctor and patient records.

#### 3. Billing and Invoicing

- Generate and store billing information.
- Handle payment processing and receipts.

## **Session Management**

#### 1. Session Initialization

• Use Codelgniter's session library to manage user sessions.

#### 2. Session Protection

- Ensure that only logged-in users can access specific pages.
- Implement session timeout and automatic logout for security.

#### 3. Access Control

• Define user roles and permissions to restrict access to sensitive parts of the system.

# **PHP Functions**

#### 1. OnLoad Functions

• Optimize PHP file processing by implementing onload functions for faster server response.

## 2. Database Connectivity

• Use CodeIgniter's database library to connect and interact with the MySQL database.

## 3. Utility Functions

• Implement utility functions for common tasks like data validation, encryption, and session management.