Ticketing System Architecture Documentation

1. Introduction

The Ticketing System is a web application designed to facilitate efficient ticket management for customer support. This document outlines the architectural structure of the system, explaining its components, interactions, and deployment procedures.

2. System Overview

The Ticketing System allows users to create tickets, check ticket statuses, and contact support. It utilizes HTML for structure, CSS for styling, JavaScript for client-side interactivity, PHP for server-side processing, and MySQL for database management.

3. Architectural Goals and Constraints

Goals: - Provide a user-friendly interface for ticket creation and management. - Ensure data security and user privacy. - Optimize system performance and responsiveness. Constraints: - Utilize XAMPP as the server environment. - Rely on PHP and MySQL for server-side processing and database management.

4. System Components

4.1 Frontend Components - HTML: Defines the structure and content of web pages. - CSS:Styles the HTML elements, enhancing the user interface. - JavaScript: Provides interactive features and dynamic content on the client side.

4.2 Backend Components - PHP: Handles server-side logic, processes form data, interacts with the database, and generates dynamic content. - MySQL: Stores ticket data, user information, and system logs.

4.3 Database Schema - Tables: - ‘tickets‘: Stores ticket information (ticket ID, user ID, status, description, etc.). - ‘users‘: Manages user data (user ID, username, email, hashed passwords, etc.). - ‘messages‘: Stores the user and admin messages (user ID, useremail, message, etc.).

5. High-Level Architecture

Client-Side Interaction: - Users interact with the web interface (HTML/CSS/JavaScript) to create tickets, check statuses, and contact support. Server-Side Processing: - HTTP requests are processed by the Apache server in XAMPP. - PHP scripts handle form submissions, authenticate users, and interact with the MySQL database. Database Management: - MySQL stores and retrieves ticket data and user information.

6. Security Measures

- Data Sanitization: Input data is sanitized to prevent SQL injections and XSS attacks. - Authentication: User login and session management are implemented securely. - Encryption: Sensitive data (such as passwords) is hashed before storage.

7. Deployment

- The application files (HTML, CSS, JavaScript, PHP) are placed in the XAMPP ‘htdocs‘ directory. - The MySQL database schema is imported and configured via phpMyAdmin. - XAMPP is started, enabling Apache and MySQL services. - Users can access the Ticketing System through ‘http://localhost/Ticket‘.

8. Conclusion

The Ticketing System’s architecture ensures a seamless flow of information between users and support staff. By leveraging HTML, CSS, JavaScript, PHP, and MySQL in the XAMPP environment, the system provides a robust, secure, and user-friendly experience for managing customer support tickets.