OBITUARY MANAGER DOCUMENTATION

**Introduction:**

This document outlines the development of an "Obituary Web Application," a platform for users to create, manage, and share obituaries online. The application aims to offer a user-friendly experience while maximizing visibility through search engine optimization (SEO) and social media integration.

**Technology Stack:**

* **Frontend:** HTML provides the web page structure, CSS styles the content visually, and JavaScript enhances user interaction.
* **Backend:** Python's Django framework handles server-side logic, URL routing, and database communication.
* **Database:** MySQL stores obituary data, user accounts, and other relevant information. phpMyAdmin, a tool bundled with Xampp, allows database management.
* **Development Environment:** Visual Studio Code is the chosen integrated development environment (IDE).
* **Operating System:** The development is done on a Windows platform.

**Project Structure:**

* **Frontend:**
  + **HTML Files:** Define the basic structure and layout of web pages.
  + **CSS Files:** Control the visual presentation and styling of the pages.
  + **JavaScript Files:** Add interactivity and dynamic features to enhance user experience.
* **Backend:**
  + **Django Framework:** Handles the core logic of the application, including:
    - **Views:** Manage how data is displayed (templates) and handle user input (forms).
    - **Models:** Define the data structure for storing obituary details in the database.
* **Database:**
  + **MySQL:** Stores all application data, including obituary details, user accounts, and potentially other related information.
  + **phpMyAdmin:** Provides a user interface for managing the MySQL database within the Xampp environment.

**Features:**

* **User Authentication:**
  + Users can register and log in to manage their submitted obituaries, ensuring data privacy.
* **Obituary Submission:**
  + A user-friendly form enables adding obituary details, including:
    - Name of the deceased
    - Date of birth
    - Date of death
    - Biography/ Content/ Description of the deceased
* **Obituary Management:**
  + Users can edit or delete their submitted obituaries for any necessary changes.
  + An admin panel allows for overall management of all submitted obituaries.
* **Obituary Display:**
  + Publicly accessible web pages showcase obituary details.
  + **SEO Optimization:** Improves search engine ranking for better visibility.
  + **Social Media Sharing:** Enables easy sharing on social media platforms to increase awareness.

**SEO and Social Media Optimization:**

* **SEO Techniques:**
  + Implement meta tags, alt text for images, and appropriate heading tags (H1, H2, etc.) for better search engine crawlability.
  + Utilize clean and descriptive URLs for each obituary.
  + Generate a sitemap and robots.txt file to guide search engine bots.
* **Social Media Optimization:**
  + Integrate Open Graph tags to ensure proper previews when shared on social media platforms.
  + Include social media sharing buttons for easy posting of obituaries.

**Development Workflow:**

1. **Setup:**
   * Install Django and configure the project environment.
   * Set up MySQL database and phpMyAdmin within Xampp.
2. **Frontend Development:**
   * Create HTML templates for the web page structure.
   * Design CSS styles for a visually appealing presentation.
   * Implement JavaScript features for user interaction and dynamic elements.
3. **Backend Development:**
   * Define Django models to represent obituary data and its structure.
   * Create views to handle different user requests, such as displaying forms or processing submitted data.
   * Implement URL routes to map user requests to specific views.
   * Integrate forms for user input and data validation to ensure information accuracy.
4. **Database Management:**
   * Design and set up necessary MySQL database tables for storing obituary data and other relevant information.
   * Use phpMyAdmin for database operations like creating tables, adding data, and managing users.
5. **Testing and Debugging:**
   * Conduct thorough testing of the application's functionality across different scenarios.
   * Identify and fix any bugs encountered during testing to ensure a smooth user experience.
   * Test for browser compatibility to ensure the application works well on various browsers and devices.
6. **Deployment:**
   * Deploy the application to a web server to make it publicly accessible.
   * Implement appropriate security measures to protect user data and ensure secure operation of the application.