Software & Functional Requirements Specification

FDA – Website

14-07-2023

FDA, Maharashtra
(fda-ITCell@gov.in)

Amendment Log

Version	Date	Change No	Brief Description	Sections Changed
1.1	14 July 2023	-	Draft	-

Approver Details

Sr.No.	Date	Approver Name	Document Version	Comments
1				
2				
3				
4				

Table of Content

FDA Website – Content Management Feature

Introduction

Requirement

The “Food and Drugs Administration, Maharashtra” (FDA), currently has a static website that requires frequent content updates. To facilitate easier content management, FDA wants to introduce a content management system (CMS) into the existing website. The CMS will employ a file/folder structure, enabling the website administrator to access and update files containing text, which will be reflected on the website.

Scope

The scope of this project is to enhance the existing static website of the “Food and Drugs Administration, Maharashtra”, by implementing a CMS. The CMS will provide a user-friendly interface (File / Folder stricture) for the website administrator to manage and update website content efficiently.

Proposed Solution

In order to fulfill the requirements outlined in the given use case, an upgrade to the current website will be implemented using PHP Laravel technology.

The content management process will be facilitated through the utilization of a file structure. The website administrator of the Food and Drugs Administration (FDA) will be granted access to this file structure by logging in through the provided admin login credentials, which will be hosted by cloud providers.

System Requirements

Hardware Requirements

1. Web server or cloud server capable of hosting the website.
2. Sufficient storage space for storing website files and content.

Software Requirements

1. Operating system capable of hosting the web server
2. Web server software (e.g., Apache, Nginx)
3. Text editor for modifying website content files
4. Web browser for accessing the CMS interface

Proposed Development Technology

1. **User Interface**: PHP Laravel
2. **Database**: File Stricture
3. **Development Platform**: Visual Studio Code
4. **UAT Server**: FDA Cloud Server
5. **Deployment Server**: FDA Cloud Server

Server Specification

1. Application Server
    a. **Processor**: A quad-core or higher CPU with a clock speed of at least 2.0 GHz is recommended to handle the web application's processing needs.
    b. **Memory**: At least 8 GB of RAM is recommended to ensure smooth performance and avoid slowdowns due to memory limitations.
    c. **Storage**: An SSD storage with at least 256 GB of capacity would be suitable for storing the web application and related data.
    d. **Operating System**: A modern 64-bit server operating system, such as Windows Server 2016 or Ubuntu Server 18.04 LTS, is recommended.
    e. **Web Server**: A web server, such as Nginx or Apache, would be necessary to serve the web application to clients.
    f. **Database**: NA.
    g. **Network**: A Gigabit Ethernet connection is recommended to ensure fast and reliable data transfer.
    h. **Security**: The server should be secured with up-to-date anti-virus and anti-malware software, firewalls, and regular security patches and updates.

Client-Side PC Requirement

1. Web Browser: Any web browser that supports latest web applications.
2. OS: Any (Windows, iOS, Android etc.)

Functional Requirements

User Roles and Permissions

1. Administrator: Has full access to the CMS and can create, update, and delete website content. Administrator will use cloud server credentials to access the folder stricture.
2. Viewer: Can only view the website content but does not have permission to make any changes.

Content Management

- The CMS shall provide an authentication (cloud server authentication) mechanism to ensure that only authorized administrators can access and modify the website content.
- Upon successful authentication, the administrator shall be presented with a dashboard showing the existing files and folders.
- The administrator shall be able to navigate through the file/folder structure to locate the desired content file for modification.
- The CMS shall provide a text editor or a similar interface to edit the content of the selected file.
- The administrator shall be able to save the changes made to the content file.
- The CMS shall automatically update the website with the modified content, reflecting the changes in real-time.

File and Folder Management

- The administrator shall be able to create new folders to organize the website content.
- The administrator shall be able to upload new files to the appropriate folders.
- The CMS shall support the deletion of files and folders, providing appropriate warnings and confirmation dialogs to prevent accidental deletion.

Constraints

- The CMS development should adhere to the existing website design and branding guidelines.
- The CMS should utilize the existing server infrastructure and available resources.
- The project should be completed within the allocated timeline.

Future Scope

Admin Module

- Develop an Admin Module within the CMS to provide advanced functionalities for website administration.
- Implement user management features, allowing the website administrator to create and manage multiple user accounts with different roles and permissions.
- Enhance content management capabilities by enabling the administrator to schedule content updates, set approval workflows, and track revision history.
- Integrate analytics tools to provide insights into website traffic, user engagement, and content performance.
- Incorporate a media library feature to enable the administrator to manage and organize images, videos, and other media files used on the website.
- Introduce a template management system, allowing the administrator to create and customize website templates for different sections or pages.

Fully Dynamic Website

- Redesign the website architecture to make it fully dynamic, eliminating the need for manual file modifications.
- Implement a database-driven approach to store and manage website content, enabling the administrator to update content directly through a user-friendly interface.
- Develop a flexible page management system, allowing the administrator to create, update, and delete pages as needed, with support for hierarchical navigation structures.
- Implement dynamic content elements, such as news feeds, event calendars, and interactive forms, to enhance user engagement and provide up-to-date information.
- Incorporate search functionality to enable users to find specific content quickly and efficiently.
- Integrate social media sharing capabilities, allowing visitors to share website content on various social platforms.