Blog Website
“TroubleShoot”

Software Requirements Specification

22/4/2024

Name: Chinmaye HG
Registration number:12215959

Prepared for

Continuous Assessment 3

Spring 2024

Revision History

Date	Description	Author	Comments
10/3/24	VERSION 1	Chinmaye	Basic html css
17/3/24	VERSION 2	Chinmaye	Multiple pages
2/4/24	VERSION 3	Chinmaye	BootStrap and jQuery
10/4/24	VERSION 4	Chinmaye	Backend

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to outline the software requirements for the "TroubleShoot" blog website. It serves as a comprehensive reference for understanding the project's scope, functional and non-functional requirements, design constraints, and other relevant information.

1.2 Scope

The "TroubleShoot" blog website is a platform where users can create and share blog posts on various topics related to technology, artificial intelligence, data science, and programming. It aims to foster a community of tech enthusiasts, learners, and experts by providing a space for knowledge-sharing, discussions, and networking.

1.3 Definitions, Acronyms, and Abbreviations

SRS: Software Requirements Specification

UI: User Interface

CRUD: Create, Read, Update, Delete

API: Application Programming Interface

1.4 References

W3Schools (https://www.w3schools.com/)

W3Schools provides tutorials and references on various web development technologies, including HTML, CSS, JavaScript, and more. It was used for learning and referencing web development concepts. GitHub (https://github.com/)

GitHub

gitHub is a platform for hosting and collaborating on code repositories. It was utilized for version control and sharing code during the development process.

Bootstrap Documentation

(https://getbootstrap.com/docs/5.0/getting-started/introduction/) Bootstrap is a popular CSS framework for building responsive and mobile-first websites. Its documentation was referenced for implementing styling and layout components in the project.

JavaPoint (https://www.javatpoint.com/)

JavaPoint provides tutorials and articles on programming languages, web development, and other technical topics. It was used as a supplementary resource for learning and understanding certain concepts.

YouTube (https://www.youtube.com/)

YouTube is a video-sharing platform where tutorials and instructional videos on various topics, including web development, can be found. It was used for accessing video tutorials and demonstrations related to specific development tasks and techniques.

1.5 Overview

The remainder of this document is organized as follows:

Section 2 provides a general description of the product.

Section 3 outlines the specific requirements, including external interfaces, functional requirements, non-functional requirements, design constraints, and other relevant requirements.

Section 4 presents analysis models, such as data flow diagrams, if applicable.

Section 5 includes the GitHub link for the project repository.

Appendices contain additional information or supporting materials, if needed.

2. General Description

2.1 Product Perspective

The "TroubleShoot" blog website is a web-based application developed using modern web technologies, including HTML, CSS, JavaScript, and Bootstrap for the frontend, and Node.js, Express, and MongoDB for the backend.

2.2 Product Functions

The main functions of the "TroubleShoot" blog website include:

1. User registration and authentication
2. Creating, editing, and deleting blog posts
3. Categorizing and tagging blog posts
4. Commenting and interacting with blog posts
5. Searching and browsing blog posts
6. User profile management
7. Administrative features for moderating content and user activities

2.3 User Characteristics

The target users of the "TroubleShoot" blog website include:

Tech enthusiasts interested in learning about and discussing various technology-related topics

Professionals and experts in fields such as artificial intelligence, data science, and programming

Students and learners seeking educational resources and knowledge-sharing opportunities

2.4 General Constraints

The website should be responsive and accessible across different devices and screen sizes.

The website should adhere to web accessibility standards and best practices.

The website should be secure and protect user data and information.

2.5 Assumptions and Dependencies

The website will be hosted on a reliable web server or cloud platform.

Users will have access to a modern web browser and an internet connection.

The MongoDB database will be hosted and maintained separately.

3. Specific Requirements

3.1 External Interface Requirements

Bootstrap:

The Bootstrap framework was utilized for designing and implementing the user interface components of the software product. Bootstrap provides a collection of pre-styled components and layout utilities, facilitating the creation of responsive and visually appealing user interfaces.

3.1.3 Software Interfaces

jQuery:

jQuery, a fast and feature-rich JavaScript library, was employed to enhance the interactivity and functionality of the user interface elements. It provides simplified methods for DOM manipulation, event handling, and AJAX requests, contributing to the dynamic behavior of the software application.

3.1.4 Communications Interfaces

MongoDB:

MongoDB, a NoSQL database management system, served as the backend data store for the software product. Communications with MongoDB were established through the MongoDB Node.js driver, allowing the application to perform CRUD (Create, Read, Update, Delete) operations on the database, store and retrieve data, and handle data persistence.

3.1.1 User Interfaces

The "TroubleShoot" blog website will have the following user interfaces:

Landing page

Blog listing page

Individual blog post page

User registration and login forms

User profile page

Create/edit blog post form

Search and filtering interface

Comment section for blog posts

3.1.2 Hardware Interfaces

The website will be accessed through various devices, including desktops, laptops, tablets, and smartphones. No specific hardware interfaces are required.

3.1.3 Software Interfaces

The website will interact with the following software components:

MongoDB database for storing and retrieving data

Node.js and Express for server-side logic and API development

Third-party libraries and frameworks (e.g., Passport.js for authentication, Multer for file uploads)

3.1.4 Communications Interfaces

The website will communicate with users through standard web protocols, such as HTTP and HTTPS.

3.2 Functional Requirements

3.2.1 User Authentication and Authorization

Users should be able to register and create an account.

Users should be able to log in and log out of their accounts.

User roles and permissions should be implemented (e.g., regular users, administrators).

Password hashing and secure authentication mechanisms should be employed.

3.2.2 Blog Post Management

Users with appropriate permissions should be able to create, edit, and delete blog posts.

Blog posts should have fields for title, content, author, tags, and other metadata.

Rich text formatting and image uploads should be supported for blog post content.

Users should be able to categorize blog posts by topics or categories.

Blog posts should be displayed with appropriate formatting and styling

3.2.3 Commenting and Interaction

Users should be able to comment on blog posts.

Comment moderation features should be available for administrators.

Users should be able to like or dislike blog posts and comments.

Social sharing options for blog posts should be provided

3.2.4 Search and Filtering

Users should be able to search for blog posts by title, content, tags, or other relevant criteria.

Filtering options should be available to sort blog posts by date, popularity, or other parameters.

3.2.5 User Profile Management

Users should be able to view and update their profile information.

Users should be able to view a list of their published blog posts.

Users should be able to manage their account settings and preferences.

3.2.6 Administration and Moderation

Administrators should have access to a control panel or dashboard.

Administrators should be able to manage and moderate user-generated content.

Administrators should be able to ban or suspend users for violating rules or guidelines.

Administrators should be able to configure website settings and preferences.

3.5 Non-Functional Requirements

3.5.1 Performance

The website should have reasonable load times and response times for various user interactions.

Optimization techniques should be employed to ensure efficient data retrieval and rendering.

Caching mechanisms should be implemented where appropriate.

3.5.2 Reliability

The website should be available and accessible with minimal downtime.

Appropriate error handling and logging mechanisms should be implemented.

Regularly scheduled backups and disaster recovery plans should be in place.

3.5.3 Availability

The website should be available 24/7, with occasional maintenance windows scheduled in advance.

3.5.4 Security

The website should follow best practices for web application security, including input validation, protection against common vulnerabilities (e.g., XSS, CSRF), and secure communication protocols.

User data and sensitive information should be encrypted and protected.

Regular security audits and penetration testing should be conducted.

3.5.5 Maintainability

The codebase should be well-organized, modular, and follow coding standards and best practices.

Proper documentation and comments should be provided to facilitate future maintenance and updates.

Automated testing frameworks should be implemented to ensure code quality and regression testing.

3.5.6 Portability

The website should be compatible with popular web browsers and devices.

The backend components should be easily deployable on different hosting environments or cloud platforms.

3.7 Design Constraints

The website should follow responsive web design principles to ensure a consistent user experience across different devices and screen sizes.

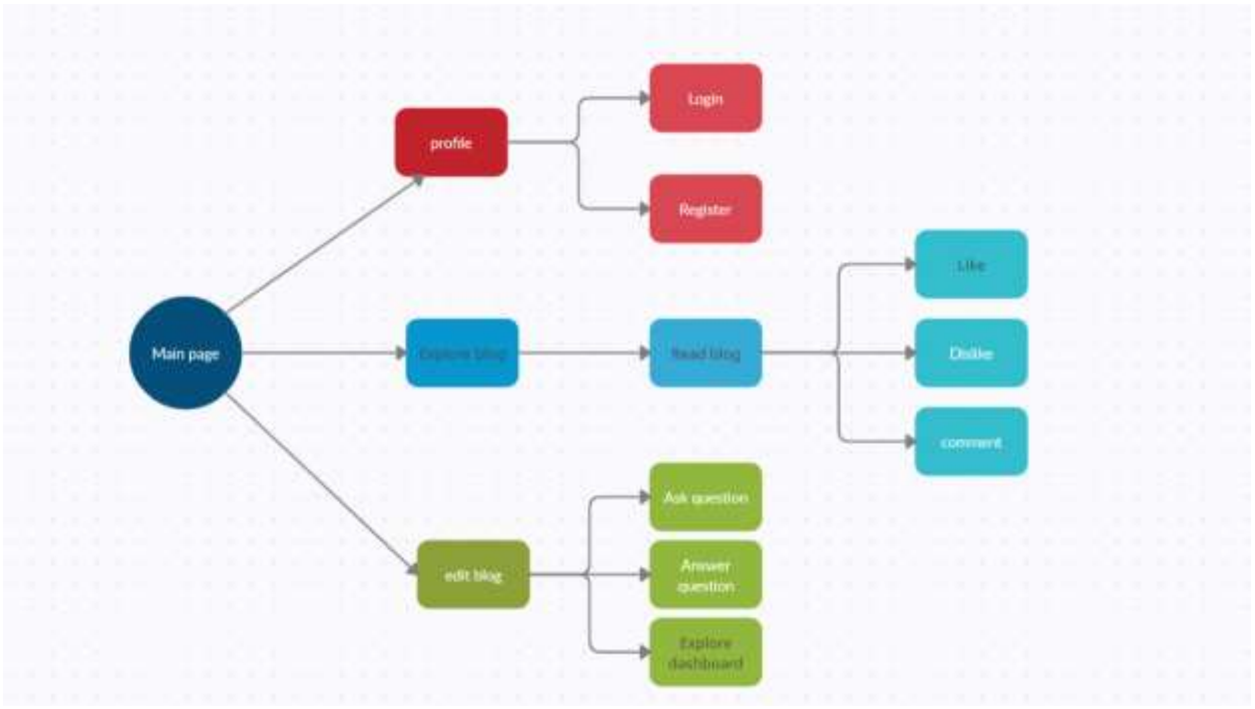
The user interface should adhere to modern design principles and provide an intuitive and user-friendly experience.

3.9 Other Requirements

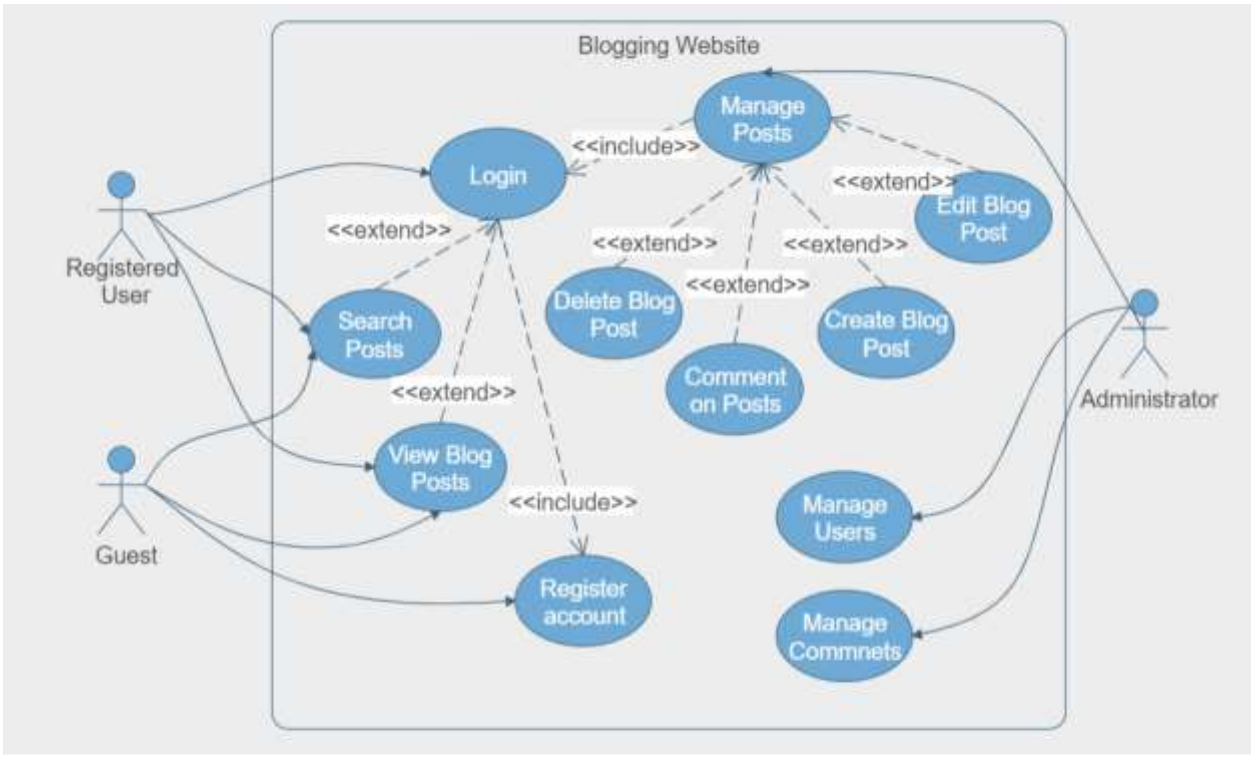
The website should comply with relevant laws and regulations, including data privacy laws and intellectual property rights.

Accessibility standards should be followed to ensure the website is inclusive and usable for users with disabilities.

4.1 Data Flow Diagrams (DFD)



4.2 Use case diagram:



5. GitHub Link

The source code for the "TroubleShoot" blog website is hosted on GitHub:

https://github.com/chinmaye5/TroubleShoot.git

6. CLIENT APPROVAL PROOF: Pending.