

**Requirements Analysis and Specifications**

Web.

**Abd alfatah aljuaidi**

**Cover Page**

**Revisions Page**

**Table of Contents**

**TABLE OF CONTENTS**

**1. INTRODUCTION……………………………………………...3**

1.1 Project Overview.…………………………………………3

**2. SPECIFIC REQIUREMENTS…………………………………4**

2.1 External Interface Requirements.…………………………………………4

2.1.2 User Interfaces.………………………………………4

2.1.3 Hardware Interfaces……………………………………………4

2.1.4 Software Interfaces……………………………………………5

2.1.5 Communications Protocols……………………………………………5

2.2 Software Product Features.………………………………………5

2.3 Software System Attributes ……………………………………………6

2.3.1 Reliability……………………………………………6

2.3.2 Availability……………………………………………6

2.3.3 Security……………………………………………7

2.3.4 Maintainability……………………………………………7

2.3.5 Portability……………………………………………7

2.3.6 Performance……………………………………………8

2.4 Database Requirements……………………………………………8

**3. ADDITIONAL MATERIAL…………………………………8**

**1. INTRODUCTION**

Requirements analysis in our web development project involves engaging with our clients, potential users, and internal stakeholders to gain a deep understanding of their expectations and aspirations. Through interviews, workshops, and surveys, we aim to elicit their specific requirements, including design preferences, desired features, and target audience. Our team will diligently capture both functional and non-functional requirements to ensure the seamless operation of the platform. We will focus on user experience, performance, security, scalability, and compatibility with various devices and browsers. By collaboratively gathering and analyzing these requirements, we lay the foundation for an effective and successful project execution.

* 1. **Project Overview**

The project aims to develop a user-friendly web platform that enables individuals and businesses to effortlessly create their company's website. The primary objective is to streamline the website development process, making it accessible to users with varying levels of technical expertise. By leveraging modern technologies and efficient design principles, the platform seeks to democratize website creation, empowering users to establish their online presence with ease.

1. **SPECIFIC REQIUREMENTS**

1. User Registration and Authentication:

- The platform shall provide a user registration form with mandatory fields for name, email, and password.

- Users must verify their email addresses during registration to activate their accounts.

- Passwords shall be securely hashed and stored to ensure data privacy.

- The platform should support password recovery options, such as email-based reset links.

* 1. **External Interface Requirements**

User Interface (UI): The platform's user interface shall be intuitive, user-friendly, and visually appealing. All elements, controls, and navigation should be clear and self-explanatory to minimize the learning curve for users. The UI should be responsive and adapt seamlessly to various screen sizes, from desktops to mobile devices. User Authentication and Authorization: The platform shall support user registration and login functionalities to provide personalized access to users. Users will require email verification during the registration process to activate their accounts securely. The platform should implement role-based access control to manage different user roles (e.g., admin, editor, regular user) with appropriate permissions.

**2.1.2 User Interfaces**

**User Registration and Login:**

The user registration interface allows new users to sign up for the platform by providing their name, email, and password.

The login interface enables existing users to access their accounts by entering their credentials.

Password recovery options, such as "Forgot Password," should be available for users to reset their passwords if needed.

**Dashboard:**

After login, users will be directed to their personalized dashboard, which serves as a central hub for managing their websites and projects.

The dashboard will display an overview of the user's created websites, recent activities, and access to various platform features.

**2.1.3 Hardware Interfaces**

Computer, Laptop, and Mobile Devices: Users will access the platform through standard computing devices, such as desktop computers, laptops, and mobile devices (e.g., smartphones and tablets). The platform's user interface should be responsive and adaptable to various screen sizes and resolutions to ensure a consistent experience across devices. Keyboard and Mouse: The primary input devices for users interacting with the platform will be keyboards and mice. Users will utilize the keyboard to input text, navigate through the website builder interface, and manage various form fields and settings. The mouse will be used for selecting and dragging elements, interacting with menus, and other tasks.

**2.1.4 Software Interfaces**

Application Programming Interface (API): The platform shall offer a well-defined API that allows developers and third-party services to integrate with the platform. The API will enable external applications to access platform functionalities, such as website creation, user management, and content generation. Authentication mechanisms (e.g., API keys or OAuth) will be employed to ensure secure data access. Database Interface: The platform will interact with a back-end database to store user accounts, website content, templates, and other relevant data. The database interface should support efficient data retrieval, storage, and updates to handle user-generated content.

**2.1.5Communications Protocols**

Hypertext Transfer Protocol (HTTP) and Hypertext Transfer Protocol Secure (HTTPS): HTTP and HTTPS are the standard protocols used for communication between web browsers and web servers. HTTP is used for regular communication, while HTTPS provides a secure and encrypted channel for sensitive data transmission, such as login credentials and user data. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP): TCP and UDP are transport layer protocols that enable reliable and connection-oriented communication between the platform's server and clients. TCP is suitable for data that requires guaranteed delivery, while UDP is used for real-time data transmission where occasional packet loss is acceptable.

* 1. **Software Product Features**

**User Registration and Authentication:** User-friendly registration process with email verification to activate accounts securely. User login with strong password encryption and optional two-factor authentication for enhanced security.

**Intuitive Website Builder:** Drag-and-drop interface for easily adding, arranging, and customizing website elements. Wide selection of pre-designed templates for various industries and design preferences.

* 1. **Software System Attributes**

**Performance:** The platform should offer responsive and low-latency interactions, providing a smooth user experience during website creation and editing. Fast-loading websites with optimized content and media assets to enhance user engagement and search engine rankings.

**Reliability:** The platform should operate consistently without unexpected downtime or disruptions, ensuring continuous availability for users. Reliable data storage and backup mechanisms to prevent data loss and enable data recovery in case of system failures.

* + 1. **Reliability**

High Availability: The platform should be highly available, accessible to users whenever they need it. Measures like load balancing, redundancy, and failover mechanisms should be in place to handle increased traffic and prevent single points of failure. Error Handling: The platform should have robust error handling and graceful degradation mechanisms. It should gracefully recover from errors and provide meaningful error messages to users when something goes wrong, guiding them on how to resolve issues.

* + 1. **Availability:**

**Redundancy and Load Balancing:** Implementing redundancy and load balancing across multiple servers ensures that if one server fails or experiences high traffic, the platform seamlessly switches to an available server, minimizing downtime.

**Fault Tolerance:** The platform should be designed with fault tolerance in mind, where critical components have backups or alternative routes to continue functioning in the event of failures.

**2.3.4Security**

**Data Encryption:** Sensitive data, such as user passwords and payment information, should be stored and transmitted in encrypted form to protect against data theft.

**Input Validation:** Implementing thorough input validation and sanitization prevents common security vulnerabilities, such as SQL injection and cross-site scripting (XSS) attacks.

* + 1. **Maintainability**

Modularity and Code Organization: The platform's codebase should be organized into well-defined modules and components, making it easier for developers to understand and modify specific functionalities without affecting other parts of the system. Clean and Readable Code: Adhering to coding best practices and maintaining clean, readable code promotes easier maintenance, code reviews, and collaboration among development teams.

* + 1. **Portability**

Browser Compatibility: Websites created on the platform should be compatible with a wide range of modern web browsers, ensuring consistent user experiences across different browsers. Responsive Design: The platform's user interfaces and websites generated by the platform should be responsive, adapting to various screen sizes and resolutions, including desktops, tablets, and smartphones**.**

**2.3.6 Performance**

**Fast Page Load Times:** Websites created on the platform should load quickly to minimize user wait times and improve user experience. Optimizing image sizes, leveraging browser caching, and using content delivery networks (CDNs) can improve page load speed.

**Scalability:** The platform should be able to scale horizontally and vertically to accommodate increasing user traffic and demand without a significant drop in performance. Load balancing and dynamic resource allocation are essential for maintaining optimal performance during peak periods.

* 1. **Database Requirements**

**Data Schema Design:** Designing an optimized and flexible database schema is essential to accommodate various data types and relationships between entities.

**User Data Management:** The database should store user information, including user profiles, authentication credentials, and preferences.

1. **ADDITIONAL MATERIAL**

Design Plan: A design plan that describes the platform's user interface (UI) and user experience (UX) design principles. It includes wireframes, mockups, and design guidelines for a consistent and intuitive user interface.

**Website Development Plan:** A comprehensive plan for developing the platform's core features, modules, and integrations. It includes task breakdown, development timelines, and resource allocation.