
Daffodil International University

# Static Website Builder (CMS)

**Submitted By**

**MD. SAHARUKKARIM**

**133-35-559**

**Supervised By**

**Afsana Begum**

**Assistant Professor & Coordinator M.Sc**

This project report has been submitted in fulfilment of the requirements for the degree of **Bachelor of Science in Software Engineering**

## DAFFODIL INTERNATIONAL UNIVERSITY

### DECLARATION OF THESIS AND COPYRIGHT

I declare that this thesis is classified as:

I acknowledge that Daffodil International University reserves the following rights:

1. The Project is the Property of Daffodil International University.
2. The Library of Daffodil International University has the right to make copies of the Project for the purpose of research only.
3. The Library of Daffodil International University has the right to make copies of the Project for academic exchange.

Certified by:

NOTE: * If the Project is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

# PROJECT DECLARATION LETTER (OPTIONAL)

Librarian,
Daffodil International University,
Daffodil Smart City,
Ashulia.Dhaka,Bangladesh

Dear Sir,

CLASSIFICATION OF Project AS RESTRICTED

Please be informed that the following project is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.
Author's Name
Project Title

Reasons (i)

(ii)

(iii)

Thank you.

Yours faithfully,

_____
(Supervisor's Signature)

Date:

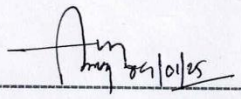
Stamp:

Note: This letter should be written by the supervisor and addressed to the Librarian, *Daffodil International University* with its copy attached to the thesis.

## APPROVAL

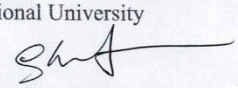
This thesis titled on "**Static Website Builder (CMS)**", submitted by **Md. Saharukkarim (ID: 133-35-559)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

### BOARD OF EXAMINERS



**Professor Dr. Engr. AKM Masum**
**Professor**
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

**Chairman**

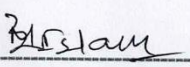


**Md. Shohel Arman**
**Assistant Professor**
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

**Internal Examiner 1**

**Dr. Marzia Ahmed**
**Assistant Professor**
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

**Internal Examiner 2**



**Dr. Md. Monowarul Islam**
**Associate Professor**
Department of Computer Science & Engineering
Jagannath University

**External Examiner**

## SUPERVISOR'S DECLARATION

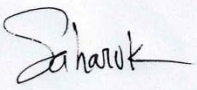
I/We* hereby declare that I/We* have checked this project and in my/our* opinion, this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Science.

(Supervisor's Signature)

Full Name : **Afsana Begum**
Position : Assistant Professor & Coordinator M.Sc
Date : 14 January 2025

## STUDENT'S DECLARATION

I hereby declare that the work in this project is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Daffodil International University or any other institution.



**(Student's Signature)**

Full Name : **Md. Saharukkarim**
ID Number : 133-35-559
Date : 14 January 2025

# Static Website Builder (CMS)


## MD. SAHARUKKARIM


Project submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Science/Master of Science


Department of Software Engineering

DAFFODIL INTERNATIONAL UNIVERSITY


December 2024

# ACKNOWLEDGEMENTS

First of all, we pay our profound thanks to Allah for making us qualified to complete this thesis.

Then we would like to express our acknowledgment of supports to our supervisor "Afsana Begum", "Assistant Professor & Coordinator M.Sc", Department of Software Engineering. We owe a lot of gratitude to her for her expert, sincere and valuable guidance and encouragements to us.

I would also like to thank those people who have completed the survey for this thesis work. Without their active involvement and contribution the validation survey could not have been very successful.

As such, it is humble intention of the authors to register our profound gratitude to all the faculty members of the Department of Software Engineering for their support.

And the last but not the least we would like to thank our parents for their boundless support and love which led us this far.

# DEDICATION

I therefore declare that I have done this project under the oversight of "**Afsana Begum**", **"Assistant Professor & Coordinator M.Sc"**, Department of Software Engineering, Daffodil International University. Also declare that neither entire record nor any portion of this record has been submitted somewhere else for my degree.

# ABSTRACT

The concept behind the Static Website Builder project is to offer a platform where people as well as companies can build, design, and deploy easy static websites with no coding knowledge at all. Predefined functionality includes user identification, selection of templates and types of Web site, preview of the changes introduced and publishing of the Web sites on free or one's own domains.

To cater for this broad market, this project targets constituencies such as small businesses, which require an internet presence, students and working professionals maintaining online portfolios. Users could also download their website as a packaged HTML/ CSS/ JavaScript application for offline use or for hosting on other platforms. A free-trial and paid system of the service's subscription is considered for implementing the availability of the service with the possibility of additional features for paid subscriptions.

# TABLE OF CONTENT

## LIST OF TABLES

## LIST OF FIGURES

## LIST OF SYMBOLS

| | |
|---|---|
| CMS | Content Management System |
| UI | User Interface |
| UX | User Experience |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |
| JS | JavaScript |
| SEO | Search Engine Optimization |
| MVC | Model-View-Controller Architecture |
| SSL | Secure Sockets Layer |
| DB | Database |
| JSON | JavaScript Object Notation |
| SFTP | Secure File Transfer Protocol |
| PHP | Hypertext Preprocessor (Programming Language) |

## LIST OF ABBREVIATIONS

| | |
|---|---|
| CMS | Content Management System |
| UI | User Interface |
| UX | User Experience |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |
| JS | JavaScript |
| SEO | Search Engine Optimization |
| MVC | Model-View-Controller Architecture |
| SSL | Secure Sockets Layer |
| DB | Database |
| JSON | JavaScript Object Notation |

# LIST OF APPENDICES

# CHAPTER 1 INTRODUCTION

## 1.1 Background

Continued development of technology has led to increased necessity of the internet profiles among people, companies and organizations. However, the construction of a website can be a very time consuming task which sometimes demands knowledge of HTML or other programming languages. It's possible to use such ready solutions as CMS platforms (for instance, WordPress) or online builders, which, however, are often costly, require specific skills to work with, or allow for a limited variety of modifications.

The Static Website Builder project attempts to overcome these challenges by providing a simple solution that allows users to design stylish static websites with no programming skills. Since the designing of templates, customization features and real time preview, the users are able to build and deploy websites without much stress. It also provides buckful for download of static files and publishing using free or custom domains whether one is beginning or is a developer.

### 1.1.1 Context and Relevance

The Static Website Builder project can be attributed to the much larger category of web development that has emerged as an essential industry for providing individuals and companies with an opportunity to brand their online platforms. This industry comprises of tools, technologies and platforms that are applied in creating, construction and deployment of websites. I also occasionally discuss with counterparts in related industries including user experience design (UX/UI) industry, cloud computing industry, and the digital marketing industry.

**Trends of the Presenteneration**

- Rise of Static Websites: Static websites are also finding a new home online, primarily because they are fast, easy to manage, and secure. As opposed to the dynamic websites that heavily use dynamic webpage display with server side

processing, static websites are pre-generated and as an outcome, are more amiable to be faster loading and less expensive.

- No-Code/Low-Code Platforms: When it comes to web development, the need for making things as accessible as possible has spurred the no-code and low-code platforms. These tools allow the users who have insufficient programming knowledge to build websites with plenty of simple clicks and templates.
- Mobile-First Design: Taking into consideration the increasing usage of mobile devices in accessing internet, responsive and mobile first approach to site design form a requisite. It looks like the Static Website Builder needs to make websites responsive to the display screen in which they are accessed.
- Integration of AI in Web Development: The use of artificial intelligence (AI) is now invading website builders by performing roles such as content creation, design recommendation, and more. It improves decision making for users and expands on customization potential.

**Challenges in the Industry**

- Accessibility for Non-Technical Users: Most platforms still have high entry barriers, or they are fully coded, meanings that they cannot be used by ordinary people with no coding expertise. One of the greatest difficulties is to create accessible interfaces.
- Customization Limitations: While using templates and pre-defined elements beneficial for web building, they rarely offer further customization options which vex users with high demands of their layouts.
- Scalability and Maintenance: When the user expectations start to rise, simple static websites can be difficult to manage and add new features or compatible integrations without dynamic functionality and back-end access.
- Data Security: Static sites are more secure than dynamic sites and, for example, subscription/paid access management needs much more substantial security.
- Competition in the Market: Website builders are a reasonably saturated market, particularly with the two giants Wix and Squarespace and the platform WordPress. New entrants in any industry environment, therefore, have no option than to have maneuverability strategies, the core one being the value proposition.

**Opportunities**

- Expanding to Emerging Markets: The fact is web development is a luxury that many small business and individuals in emerging markets cannot afford. The solution to this can be had by the increased and affordable and user-friendly of the platforms.
- Customization with AI: AI can be used to improve design, content, and layout through intelligent suggestions shared with a user.
- Hybrid Models: Integrating new and flexible elements into the structure of a website development with the prospect of creating simple sites similar to static ones provides new opportunities for the implementation of powerful, scalable solutions.
- Niche Targeting: Thus, platforms can filter into segments like portfolios, websites for small businesses, or event pages, as it completely separates the market at the moment.
- Environmentally Friendly Solutions: Static websites are conservative in terms of energy consumption because they do not require much from the server. Prospective green users might be interested in the products if sustainability becomes the product's marketing pitch.

### 1.1.2 Problem Identification

Building and maintaining websites remains a significant challenge for many individuals and small businesses due to the following gaps:

- Technical Barrier: With most web development tools, one is forced to learn new programming languages such as HTML, CSS, and JavaScript or at least be well conversant with complicated CMS web development tools like WordPress or Joomla. This poses a sharp learning slope for non-professionals in tech.
- High Cost: Sites specifically developed for companies and individuals are usually expensive to develop and maintain. Website builders such as Wix, Squarespace, etc., have cheaper hefty plans for providing more functions to the user, which can also be unaffordable to some people.
- Limited Customization Options: Most current platforms are designed using the inflexible frame, which hinders creativity. People can't often obtain very specific designs when there is no knowledge of programming or they do not use paid versions.
- Time-Consuming Process: Website construction takes time, particularly in regard to coding, debugging and compilation especially when special coding techniques are used. Busy users consider this process to be time consuming.
- Scalability for Beginners: Instruments usually do not adopt converter from easily comprehensible to the conceptually shallower naives and adequately complex for the more sophisticated experts. C3: New user websites limit users when they require advance features to adopt in their websites .
- Integration Challenges: Select solutions do not integrate well with external services or options including payment gateways or domains, analytics etc. hence a drawback to the overall user experience.

**Limitations:**

**Static Website Builders (Jekyll, Hugo and Gatsby) :**

**Advantages:** Quick to load, safe, and best suited to anyone who knows how to code. **Limitations:** Has to be installed and operated by a technical person, normally needs expert system. Lacking friendly user interface and is not easy to be used in simple command line and configuration file.

**How this Project meet the above stated Gap**

The Static Website Builder project fills these gaps by offering a solution that combines:

- Ease of Use: Most of the platforms have a no-code drag and drop system to enable construction of a website without having to code.
- Affordability: It features a concept of paid plans which are divided by tiers, and even offers a free plan for the users.
- Customizability: It also allows users to choose templates and apply as many customization as they wish without programming interfaces.
- Efficiency: These are the inherent aspects such as real-time preview, templates and auto-publishing which help to reduce the amount of time needed to build and launch Web sites.
- Scalability: The system allows for advanced features, which include a customizable domain, overly premium, and downloadable static files to make sure that it meets the evolution of the user.
- Seamless Integration: The easy integration of payment systems, domain management, and analytics simplifies the overall experience of users.

### 1.1.3 Purpose and Justification

The Static Website Builder project aims to make it easy for everyone to be able to develop, design, and host beautiful static websites without coding knowledge. This is to demystify technical and financial implication often linked to web development to ensure everyone can afford a digital presence online. Through easy to use, cheap, and easily scalable tools, the project ensures that users can create their own online presence within the shortest time possible and depending on their needs for example, portfolios, business, and event websites.

**Justification for the Project**

- Democratizing Web Development
- Cyber-Management Reflection to Digital Transformation
- Tools That Make Non-Technical User Empowered

.

**Why the Project is Important**

The Static Website Builder is significance because the tool stands in tune with present and emerging trends in no-code platforms, the static site generation tools to cater for a growing market of users who require cheap, safe and efficient website builders. It also fits into the concept of empowering individuals and small businesses to be interactive within the new digital frontier without large amounts of time, money or often technical know how.

**How the Project Adds Value**

- Lowering Barriers to Entry: The project solved the problem of not requiring programming knowledge: using the platform, students, and even entrepreneurs will be able to develop websites on their own.

- Cost-Effectiveness: Unlike conventional development or designing of premium website where it can be expensive, this project's clients can select their required and affordable subscriptions that will suit them.
- Faster Website Creation: Some of the aspects such as templates, WYSIWYG editors, and other tools that permit the drag and drop style of web creation have a huge impact on the time it takes to develop a website. What is more important for such users is that the researchers are able to work at such a fast pace and with such effectiveness.
- Enhanced User Customization: This webspacing offers lots of options to change sites according to the requirement and corporate identity avoiding programming knowledge.
- Scalable Solution for Growing Needs: For the users, it provides capabilities for custom domains, static files download and integration with analytics so that the websites can progress with the users.
- Bridging Market Gaps: Here again the project has a niche by giving the ease of drag-and-drop interfaces with the speed of static sites, making it a valuable proposition.
- Security and Performance Benefits: Static websites therefore have several benefits from within, including the ability to load much quicker and the added dimension of being more secure than dynamic websites. This helps in that the end user that is visiting the websites that are generated on the platform will have the best experience possible.

### 1.1.4 Scope

The Static Website Builder is a large-scaled project that aims at enabling users to design, style, and deploy simple websites. These are strictly personal, SMEs and organizations that require an online platform to take their businesses without requiring high IT knowledge, or capital investment.

**The project cover the following key areas:**

1. User-Friendly Website Builder

- Drag-and-Drop Editor: A user interface that is customizable, meaning people with no coding skills can place items such as text, images or videos on the site.
- Pre-Designed Templates: An assortment of 70+ templates in business niches like creative portfolio, food blog, software company, startup, event, and store landing page.
- Real-Time Preview: Because it is an online platform, users are able to view and follow the changes made to their websites as a design runs through seamlessly.

2. Website Management Features

- Static File Download: Some users can download their websites as static HMTL, CSS and JavaScript files for outside hosting.
- Hosting and Publishing Options: Who integrates with hosting providers for free subdomains hosting or custom domains publishing possibilities.
- Responsive Design: Each website created will be mobile responsive, thus aiming at different screen resolutions.

3. Subscriptions and Payments

- Subscription Plans: Several levels of service, from basic (may also be free) to paid, with variations depending on the features as templates and different domain names.

4. Possible Scalability and Growth Features

- Customization Options: It is possible for users to really customize the templates to make them as unique as possible.

5. Security and Performance

- Secure Payment System: Guarantee that user subscriptions will use secure and encrypted process for authenticating the transaction.
- Optimized Performance: As pointed out, static websites are design with reliability, deploys quicker and does not require multiple servers.
- Data Protection: Select the right web hosting for your need and provide the best user data privacy.

6. Administrative tools

- User Management: Also, the administrator can control users, subscriptions, and main activity logs.
- Template Management: In the template library the admins has the permission to add templates, edit them or even delete them.
- Subscription Management: Functionalities to track payments, as well as to execute upgrades/downgrades and account status management.

.

## 1.2 Project Planning and Initiation

### Feasibility Study (Step-by-Step)

Static website builder project will help in building user friendly website for design and publishing of static website effortlessly. It has also undergone a feasibility study to determine if it will be viable and should work at reaching its goals.

### Phase 1 Preliminary Analysis & Project Scope:

The need for the web application that can give non-tech-savvy individuals and businesses the opportunity to create the static websites is growing all the time. Web development with traditional approaches is expensive and time-consuming, while existing no-code platforms are rigid or expensive. It is designed for an easy to use content management system that features drag and drop capabilities, templates, and downloadable static content for self hosting. It meets the needs of the market for benefits that come with static sites – sheer speed,-no susceptibility to hackers, and almost no need for updates. The idea of making it a subscription-based service is feasible for a wide range of users and budget considering the subscription model is extensible enough to allow for continuous growth.

### Scop Description

1. User-Friendly Website Builder

- Drag-and-Drop Editor: A user interface that is customizable, meaning people with no coding skills can place items such as text, images or videos on the site.
- Pre-Designed Templates: An assortment of 70+ templates in business niches like creative portfolio, food blog, software company, startup, event, and store landing page.
- Real-Time Preview: Because it is an online platform, users are able to view and follow the changes made to their websites as a design runs through seamlessly.

2. Website Management Features

- Static File Download: Some users can download their websites as static HMTL, CSS and JavaScript files for outside hosting.

- Hosting and Publishing Options: Who integrates with hosting providers for free subdomains hosting or custom domains publishing possibilities.
- Responsive Design: Each website created will be mobile responsive, thus aiming at different screen resolutions.

3. Subscriptions and Payments

- Subscription Plans: Several levels of service, from basic (may also be free) to paid, with variations depending on the features as templates and different domain names.

4. Possible Scalability and Growth Features

- Customization Options: It is possible for users to really customize the templates to make them as unique as possible.

5. Security and Performance

- Secure Payment System: Guarantee that user subscriptions will use secure and encrypted process for authenticating the transaction.
- Optimized Performance: As pointed out, static websites are design with reliability, deploys quicker and does not require multiple servers.
- Data Protection: Select the right web hosting for your need and provide the best user data privacy.

6. Administrative tools

- User Management: Also, the administrator can control users, subscriptions, and main activity logs.
- Template Management: In the template library the admins has the permission to add templates, edit them or even delete them.
- Subscription Management: Functionalities to track payments, as well as to execute upgrades/downgrades and account status management.

**Phase 2 Market Feasibility Analysis (or Market Research):**

The market for websites creation tool has developed tremendously because of the demands of want to be or new website owners including individuals, freelancers, small business and organizations. In this study, target market, competitors, trends, and opportunities are assessed to evaluate feasibility of the Static Website Builder project.

**1. Target Market**

The primary audience for the project includes:

- Small Businesses: Small scale businessmen and entrepreneurs in search of cheap and easy to use tools to create their web commerce.
- Freelancers: Some of the users are photographers, writers, and designers who require easy and personalized websites primarily to showcase their work.
- Non-Technical Users: People who have no coding background and the are interested in creating personal blogging, events or basic websites.
- Organizations: Interests of non-profit organizations and small groups who only require page-based websites for the provision of information.

**2. Current Market Trends**

- Rise of No-Code Solutions: Now you can build a website using no code at all thanks to apps like Wix or Squarespace. Nevertheless, they provide rather high prices and numerous features that are mostly focused on of large-scale works and initiatives, which creates opportunities for simple and cheap services.
- Demand for Static Websites: They are increasingly used since they are fast, secure and cheaper than dynamic websites.
- Affordability and Accessibility: Customers expect the tools to be easy to use but efficient, which means more and more people need tools that will allow them to create a website effortlessly.
- Customization Needs: Most people need templates that can be greatly edited in terms of color and design to suit the individual brand or personality of the user.

**3. Competitor Analysis**

- Established Platforms: There is a strong preference for Wix, Squarespace, or WordPress but they are characteristic by higher costs or more difficult settings.

- Static Site Generators: Tools such as Jekyll and Hugo provide efficient tools to build static websites but are coded which narrows down the target group.
- Emerging Platforms: Some new entrants are offering comparatively smaller tools, which do not have the growth capabilities or interface standardization.

**4. Gaps in the Market**

- Affordability: Some of the competitors do not have options which are very ‘cheap’ for small-scale projects.
- Simplicity: Not many services provide the ability to get a simple solution for people who are not technical at all, as well as some kind of opportunity for more detailed configuration.
- Ownership: Most of the top-level tools tie users to their hosting platform, and downloadable static files help to avoid this.

**5. Opportunities**

- Market Growth: The fact that there are more and more small businesses, more self-employed people in the world means a huge potential audience.
- Unique Selling Proposition (USP): A combination of easy to use interface, and low cost accomplished by the addition of Download Static Files offer a great value to the user.
- Localized Features: These two features can be used in an attempt to appeal to the users from various markets; for instance, having the localized templates or using the localized language.
- Subscription Model: With the usage of the tiered pricing strategy, the needs of the different users are met while at the same time the firm is enjoying steady revenues.

**6. Challenges**

- Competition: Specifically, the startup is forced to avoid direct rivalry with the established platforms by placing a major emphasis on differentiation and the promotion of the marketplace.
- User Retention: To guarantee users remain interested and subscribe to other levels as they advance in age.
- Awareness: To communicate directly there is need to market and brand the product so as to get to the target clients.

**Phase 3 Technical Feasibility Analysis:**

The static website builder is basically a plan that aims at easing the process of creating websites while using present day technologies and accessories. The technical viability analysis assesses the propriety of the technology, resource, and infrastructure needed in undertaking the project.

**Front-End Development:** HTML5, CSS3 and JavaScript guarantee good quality and adaptivity of static web sites output.

**Back-End Development:** PHP used as the programing Language for manage users, subscription as well as all the admin functionalities. For accounts, subscriptions, and template data MySQL which is a relational database is ideal.

**Database Management:** For data that has relations such as for managing user accounts, subscriptions and template data MySQL relational database is ideal.

**Phase 4 Financial Feasibility Analysis:**

The Static Website Builder project aims at ensuring the creation of cheap, easy to use, and highly extensible static websites.Here is the brief of the complete financial feasibility that includes estimated cost, income and profitability of the budget approach of the project.s the financial feasibility analysis, which outlines the estimated costs, revenue potential, and profitability of the project budget-friendly approach.

1. **Initial Development Costs**

Human Resources:

- Developers (Front-End, Back-End): 50,000 BDT
- UI/UX Designers: 40,000 BDT
- Project Manager: 20,000 BDT

Total: 1,10,000 BDT

Technology and Tools:

- Cloud Hosting Services: 10,000 BDT
- Third-Party API Licenses (Payment Gateway): 5,000 BDT

Total: 10,000 BDT

Marketing and Launch:

- Digital Marketing Campaigns: 5,000 BDT

Total: 5,000 BDT

Miscellaneous Costs:

- Legal and Compliance: 10,000 BDT

Total: 10,000 BDT

**Estimated Total Initial Investment: 1,40,000 BDT**

**2. Operational Costs (Annual)**

Infrastructure and Maintenance:

- Cloud Hosting and Storage: 10,000 BDT
- Domain and SSL Certificates: 2,000 BDT
- Maintenance and Updates: 10,000 BDT

Customer Support:

- Support Team Salaries: 1,20,000 BDT

Marketing and Advertising:

- Ongoing Digital Campaigns: 50,000 BDT

**Estimated Total Annual Cost: 1,92,000 BDT**

**3. Revenue Potential**

Subscription Plans:

- Free Plan (Limited Features): 0 BDT
- Basic Plan (500 BDT/month): Estimated 500 users = 2,50,000 BDT/year
- Premium Plan (1,000 BDT/month): Estimated 200 users = 2,00,000 BDT/year

Additional Revenue Streams:

- Advanced Features and Add-Ons: 50,000 BDT/year

**Estimated Total Annual Revenue: 5,00,000 BDT**

**4. Break-Even Analysis**

- Initial Investment: 1,40,000 BDT
- Annual Operational Costs: 1,92,000 BDT
- Annual Revenue: 5,00,000 BDT

## 1.3 Target User Profile and Tentative Elicitation Process

### 1.3.1 Target User

Static Website Builder project caters heavily towards various groups like small business owners freelancers students nonprofit organizations and bloggers. Users often fall within an age bracket of 18 to 50 possess fairly solid computer skills and aim creation of websites on a budget. Minimal training required for this platform due to its very intuitive interface and overall accessibility makes it pretty suitable for solo use.

### 1.3.2 User profile

Table 1: User Profile for Small business owners

| **User Class** | **Note on Characteristics** |
|---|---|
| Type of user | Small business owners |
| Age range | 18 to 50 years |
| Frequency of use | Moderate to high usage depending on the project (2-5 times per week) |
| Mandatory | Access to an internet connection |
| Computer experience | Knowledge about web browsers and online utilities |
| Education | School, high school, diploma or above |
| goal | Achieve cost-effective web presence |
| Language skills | The communication is done in English (primary interface language). |
| Number of users | Estimated 1,000 users in the first year, scaling up to 15,000 users in five years |
| Training | What about Power Point Guides to Tutorial- like Help Section |
| Others system use | Checkout process Launchers, Content management systems |
| Way of working | Independent and self-paced |

Table 2: User Profile for Students and educators

| **User Class** | **Note on Characteristics** |
|---|---|
| Type of user | Students and educators |
| Age range | 18 to 30 years |

| | |
|---|---|
| Frequency of use | Moderate to high usage depending on the project (2-5 times per week) |
| Mandatory | Access to an internet connection |
| Computer experience | Knowledge about web browsers and online utilities |
| Education | School, high school, diploma or above |
| goal | Achieve cost-effective web presence |
| Language skills | The communication is done in English (primary interface language). |
| Number of users | Estimated 5,000 student in the first year, scaling up to 50,000 student in five years |
| Training | What about Power Point Guides to Tutorial- like Help Section |
| Others system use | Checkout process Launchers, Content management systems |
| Way of working | Independent and self-paced |

Table 3: User Profile for Independents workers

| **User Class** | **Note on Characteristics** |
|---|---|
| Type of user | Independents workers |
| Age range | 18 to 50 years |
| Frequency of use | Moderate to high usage depending on the project (2-5 times per week) |
| Mandatory | Access to an internet connection |
| Computer experience | Knowledge about web browsers and online utilities |
| Education | School, high school, diploma or above |
| goal | Achieve cost-effective web presence |
| Language skills | The communication is done in English (primary interface language). |
| Number of users | Estimated 1,000 users in the first year, scaling up to 15,000 users in five years |
| Training | What about Power Point Guides to Tutorial- like Help Section |
| Others system use | Checkout process Launchers, Content management systems |

| | |
|---|---|
| Way of working | Independent and self-paced |

### 1.3.3 Elicitation Process

**User Interviews:**

- Objective: To get highly detailed information about the needs of users.
- Target Group: Non-technical users, small business owners, freelancers, web agencies.

**Online Surveys and Questionnaires:**

- Objective: To receive numerical data from a larger group.
- Target Group: Business owners, freelancers, web developers.

**Competitor Analysis**

- Objective: Analyze what is already out there in terms of competitor products and outline gaps.

**Focus Groups:**

- Objective: Conduct group discussions with the idea of exploring several user points of view.
- Target Group: Small business owners, freelancers, web designers.

**Usability Testing - Prototyping:**

- Objective: Test early prototypes with real users.
- Target Group: Non-technical users, small business owners.

**Stakeholder Workshop:**

- Objective: Align the project scope and requirements through the consolidation of information from all stakeholders concerned.
- Target audience: This will encompass the project team including business stakeholders to the technical experts.

## 1.4 Project Block Diagram

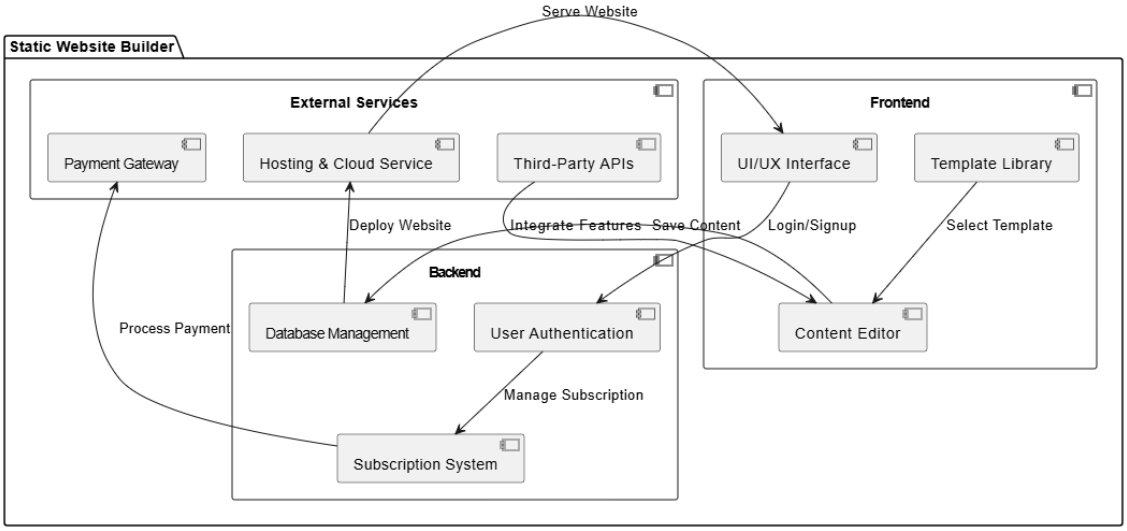



Figure 1: System Block Diagram

## 1.5 System Requirements

### 1.5.1 Hardware Requirements

Such specifications remove some barriers in the development experience including testing and debugging of the application.

- Processor (CPU): Minimum: 8th Generation Intel core i3, AMD Ryzen 5
- Memory (RAM): Minimum: 4 GB
- Storage: From the previous range of 120 GB SSD or at least 256 GB for development tools and software and for local testing.
- Graphics Card: Such one may conclude that IGP is sufficient for web development.
- Operating System: For workloads against Microsoft Windows operating system we would require Windows 10/11 while for Apple we would require MacOS Monterey or later, for Linux we would require Ubuntu 20.04 or newer.
- Network: Dependable internet connectivity of the networks (minimum of 10Mbps for bandwidth) for hosting and API interface of remote cloud services.

### 1.5.2 Software Requirements

Below are the required software specifications for both the development and production environments of the project:

- Operating System: Windows 10/11 or later, Mac Monterey or later, or Linux Ubuntu 20.04 with the preference for a dedicate graphics card.
- Programming Languages: Frontend: HTML 5.0, CSS 3.0, JavaScript; preferred is a JavaScript version 6 and above. Backend: PHP
- Frameworks and Libraries: Backend: Codeigniter (PHP)
- Database Management Systems (DBMS): MySQL
- Development Tools: Visual Studio Code, Notepad++ or atom (in case of coding).
- Version Control and Collaboration: Git (by repositories hosted in GitHub, GitLab, or Bitbucket).
- Browser Support and Debugging: google Chrome, Firefox, or safari web browsers, but with the developers' tools interface displayed.
- Design and Prototyping Tools: MS Word, AI, or Canva for photographs and designs, Figma or XD or Sketch in the case of UI/UX design.
- Testing Tools: Jest, Mocha or Selenium (if you're doing unit testing and end-to-end testing).

### 1.5.3 Constraints and Dependencies

- Hosting Services: For serving and deploying the static websites the project mostly relies on cloud hosting services that are AWS, Google Cloud or Microsoft Azure.
- CDNs (Content Delivery Networks): External CDN services like Cloudflare or AWS CloudFront are needed for enhancing the web resource's performance and speeds.
- Payment Gateways: To handle subscription payments you need integration with third-party payment processors including Stripe, PayPal, and Razorpay.
- Email Services: And while using purely transactional or notification emails, external services like SendGrid, Mailgun, AWS SES are required.
- Analytics Tools: Connectivity with analytic tools such as Google analytics or Mix panel for analyzing users engagement on the application.
- APIs for Third-Party Services: Services icons may be needed for template add-ons like maps, social networks or other widgets from outside application.
- SSL/TLS Certificates Providers: Implementing secure communication for transferring data outside the network is done through external SSL certificate service from Let's Encrypt or DigiCert.

- Version Control and Collaboration Tools: 7 The reliance in such platforms like GitHub, GitLab or Bitbucket for issues tracking, version control and collaboration during software development.
- Browser Compatibility Testing Tools: To check compatibility on various browsers and devices some of the tool available are as follows- BrowserStack or CrossBrowserTesting.
- Backup and Storage Services: Use of remote storage solutions such as AWS S3, Google Cloud Storage for users data and projects backup.

## 1.6 Project Scheduling

**Time Frame**

**Phase 1:** Preproduction (Months 1)

- Identify and close on the features, set up the framework and document the technical specification.

**Phase 2:** Core Development (Months 5)

- Design the construction of the website, the templates available and the accounts for the users.

**Phase 3:** Testing & Beta Launch (Month 3)

- Evaluation of how real users understand the website and identify performance improving and bugs corrections.

**Phase 4:** Full Service (Month 3)

- Deploy marketing efforts and initiate every product promotion process.

**Gantt Chart**

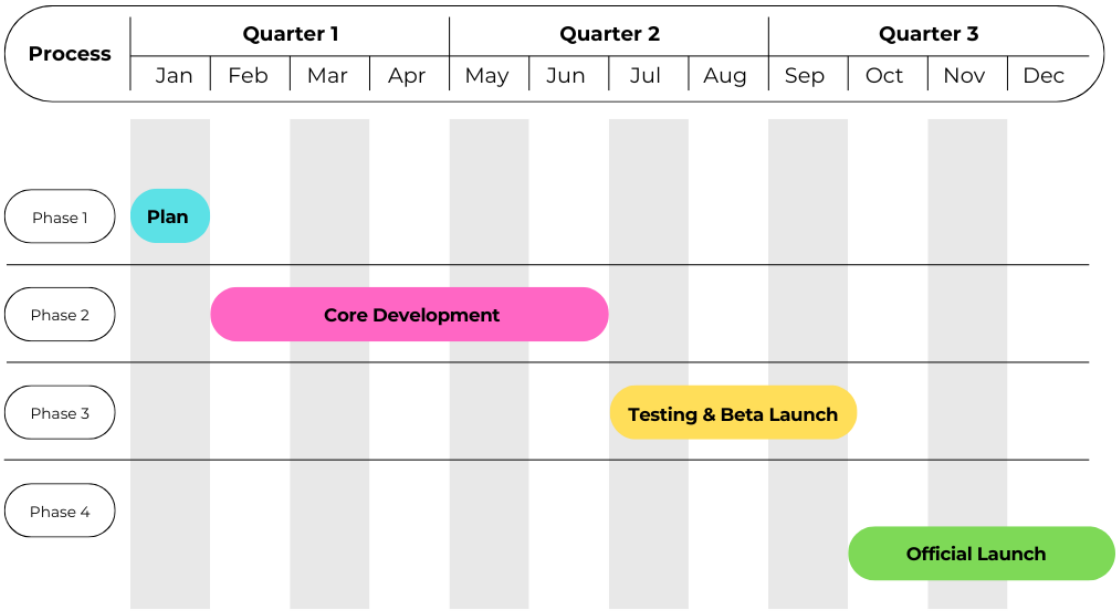



Figure 2: Gantt Chart

**Risk Management**

Risk 1: User Adoption  SMEs using early market analysis and beta testing to obtain feedback.

Risk 2: Business Constraints (for example, accessibility restricts, capacity). Spend time and effort for good testing and performance tuning.

Risk 3: Competition Set fairly reasonable prices for file-sharing and have specific features such as the possibility of downloading or exporting files.

## 1.7 Summary

The following chapter presents the financial viability and operational necessity of the Static Website Builder Project. It shows major development and operating expenses

concerning human resource, cloud, and marketing expenses, which amount to an initial investment of 1,40,000 BDT. Sustaining the operation cost is 1,92,000 BDT per annum and predicted earnings more than 5,00,000 BDT per annum through premium, Paid-on-demand. A breakeven is possible within a timeframe of 3-4 months thus confirming the profitability of a project. Software and hardware requirements are also mentioned, as well as external system interfaces, and various restrictions for dynamic functions and third-party APIs. In conclusion, the property is sustainable and will be ready to grow for the long-run.

# CHAPTER 2 DESIGN AND IMPLEMENTATION

## 2.1 Introduction

Chapter two is devoted to the description of the main development phase of the Static Website Builder Project, including the technical plan and the ways of the platform creation. In this chapter, the system architecture that allows the understanding of the system's design is described, along with the project's block diagram, ER diagram, and class diagram. Also, the specifics of the user interface including the design paradigms, the database schema, and integration with other systems are examined in the chapter. This makes it possible to create a strong and efficient system, that would solve user and business needs in the most efficient manner.

## 2.2 Functional Requirements

The functional requirements of the Static Website Builder System identify the main functions necessary for its functioning. These are; Registration, Login, Profile updates, Password recovery among others to ensure that users are provided with a secure and efficient way. Users can purchase subscriptions, achieve access to the templates database, and integrate their own domain into the platform, which allows them to create Website easy. Furthermore, user can be managed through an admin panel, tutorials and settings also increase the usability. These requirements are as follows and serve to guarantee that the platform provides a convenient solution meeting users' various demands for customization and growth.

Table 4: Functional Requirements

| **FR01** | **User Registration** |
|---|---|
| **Description** | Allow users to register by providing details like username, email, and password. |
| **Stakeholder** | Customer |

| **FR02** | **User Login** |
|---|---|
| **Description** | User gets logged in with the help of username/email address and password. |
| **Stakeholder** | Customer, Admin |

| **FR03** | **Forget Password** |
|---|---|

| | |
|---|---|
| **Description** | Make it possible for the users to constantly change the password using the email confirmation or the recovery URL |
| **Stakeholder** | Customer, Admin |

| **FR04** | **Profile Update** |
|---|---|
| **Description** | Provide orders to allow the users to edit and update their personal information such as name, email and password. |
| **Stakeholder** | Customer, Admin |

| **FR05** | **User List** |
|---|---|
| **Description** | Help administrators to control all registered users by giving them the ability to deactivate or delete the users' account. |
| **Stakeholder** | Admin |

| **FR06** | **Subscription and Plans** |
|---|---|
| **Description** | Show subscription plans (Free, Basic, Premium) with features such as prices that the subscribers are willing to pay. |
| **Stakeholder** | Customer, Admin |

| **FR07** | **Templates** |
|---|---|
| **Description** | A gallery of templates which can be selected, previewed and modified by the user should be developed. |
| **Stakeholder** | Customer, Admin |

| **FR08** | **Custom Domain Integration** |
|---|---|
| **Description** | Let people connect their site to outside domain names of their choice. |
| **Stakeholder** | Customer, Admin |

| **FR09** | **Guidance and Support** |
|---|---|
| **Description** | Promote video tutorials, Frequently asked questions, and useful guides |

| | |
|---|---|
| | for convenient using a particular platform. |
| **Stakeholder** | Customer, Admin |

| | |
|---|---|
| **FR10** | **System Settings** |
| **Description** | Provide them an availing option of Account Settings that enables users adjust their desired settings like sites response |
| **Stakeholder** | Customer, Admin |

## 2.3 Non-Functional Requirements

The Static Website Builder System's non-functional requirements guarantee its dependability, efficiency, and ease of use. Such are high system availability with 99.9% uptime, the response is expected to be under normal conditions of response time that are fitted for 10, 000 concurrent user restrictions. Security is demonstrated by the use of encryption, using HTTPS and role-based access control. It is built to be scalable, modular and device independent so that it may be used uniformly on desktops, tablets and mobiles. Usability requirements which include; accessibility, additional language, or data control laws like the GDPR increase the global user inclusiveness. Further, the data backup routine and modularity of the system guarantee system stability and the ease of later improvements.

### 2.3.1 Performance

In more detail, the system has to offer a fast and efficient environment for the user with the response time of no more than 2 seconds. It should be able to handle many users, say, up to 10, 000 users and still deliver the application with high speed. It is maintained by proper coding, caching and, load balancing.

### 2.3.2 Reliability

To the social platform, a dependability of 99.9% has been programmed in a bid to guarantee that it will always be ready for use by the users. Standby systems, error control mechanisms, and system designs are among the principles implemented to prevent and correct failures to ensure system trust worthiness.

### 2.3.3 Portability

It must also support multiple devices such as, desktop, tablet, and mobile and operating systems such as Windows, macOS, and Linux. It should be compatible with all the

popular browsers today so that regardless of the browser used by a client, they get a seemingly similar experience.

## 2.4 Object-oriented System design using UML

### 2.4.1 Use Case Diagram

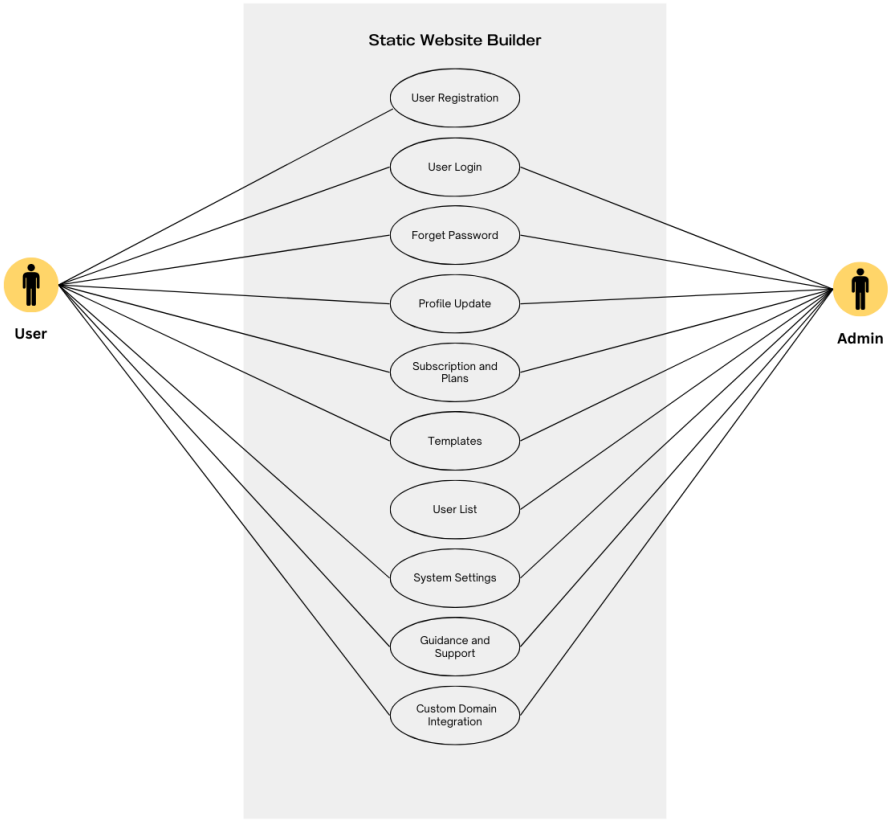



Figure 3: Use case Diagram

### 2.4.2 Case Description

Case Description-01: User Registration

| | |
|---|---|
| Use Case | User Registration |
| Goal | To allow new users to create an account on the platform. |
| Precondition | User must have a valid email address. |

<table>
<tr><td>Success End Condition</td><td>Notification: “The user account is created”</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “The account is not created because of invalid data, which was entered in the server.”</td></tr>
<tr><td>Primary Actors:<br><br>Secondary Actors:</td><td>Customer</td></tr>
<tr><td>Trigger</td><td>User navigates to the sign-up page and clicks at the Sign Up button.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>Press “Sign Up” Button</td></tr>
<tr><td>2.</td><td>Provide registration form</td></tr>
<tr><td>3.</td><td>Enter Information</td></tr>
<tr><td>4.</td><td>Press “Create Account” Button.</td></tr>
<tr><td>5.</td><td>Information saved</td></tr>
<tr><td>6.</td><td>The system saves the details and shows them !!!<br><br>Successfully Registered!!! Notify</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>The user Did not fill up the details!</td></tr>
<tr><td></td><td>4.1.a. Checked By the system & Notify by “Please Enter Full Name”.</td></tr>
<tr><td>5.1</td><td>The system did not respond</td></tr>
<tr><td></td><td>5.1.a. Show Error Message.</td></tr>
<tr><td>6.1</td><td>The system Doesn’t save the details.</td></tr>
<tr><td></td><td>6.1.a. Notification: “Details did not Save”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The registration process should complete within 3 seconds.</td></tr>
</table>

Case Description-02: User Login

<table>
<tr><td>Use Case</td><td>User Login</td></tr>
<tr><td>Goal</td><td>For this purpose, as many users want to access their accounts securely.</td></tr>
<tr><td>Precondition</td><td>The user must be registered and their account verified.</td></tr>
<tr><td>Success End Condition</td><td>User is logged in and redirected to their dashboard.</td></tr>
<tr><td>Failed End Condition</td><td>Web login is declined because User entered wrong password or have not confirmed account.</td></tr>
<tr><td>Primary Actors:<br>Secondary Actors:</td><td>Customer, Admin</td></tr>
<tr><td>Trigger</td><td>User enters their username or email address and password.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table><tr><td>1.</td><td>User inputs right email/username and password.</td></tr><tr><td>2.</td><td>Credentials are endorsed by the System.</td></tr><tr><td>3.</td><td>Logically, user is then taken back to his/her dashboard.</td></tr></table></td></tr>
<tr><td>Alternative Flows</td><td><table><tr><td>1</td><td>When credentials are entered erroneously the system produces an error message on the screen.</td></tr><tr><td>2</td><td>Again, if the user provides 3 incorrect passwords, he or she is required to reset a password.</td></tr></table></td></tr>
<tr><td>Quality Requirements</td><td>Authentication should complete within 2 seconds.</td></tr>
</table>

Case Description-03: Forget Password

<table>
<tr><td>Use Case</td><td>Forget Password</td></tr>
<tr><td>Goal</td><td>To allow users to reset their password.</td></tr>
<tr><td>Precondition</td><td>User must provide a valid registered email.</td></tr>
<tr><td>Success End Condition</td><td>User successfully resets their password.</td></tr>
<tr><td>Failed End Condition</td><td>Reset fails in case the entered email is invalid or the link received for the recovery of account has already expired.</td></tr>
<tr><td>Primary Actors:<br>Secondary Actors:</td><td>Customer, Admin</td></tr>
<tr><td>Trigger</td><td>User enters “Forgot Password” and types his / her email ID.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table><tr><td>1.</td><td>User provides a valid registered email.</td></tr><tr><td>2.</td><td>System sends a password recovery email.</td></tr><tr><td>3.</td><td>User clicks the link and sets a new password.</td></tr></table></td></tr>
<tr><td>Alternative Flows</td><td><table><tr><td>1</td><td>If email is not registered, the system displays an error message.</td></tr><tr><td>2</td><td>If recovery link expires, the user must request a new one.</td></tr></table></td></tr>
<tr><td>Quality Requirements</td><td>Password recovery email should be sent within 1 minute.</td></tr>
</table>

Case Description-04: Profile Update

<table>
<tr><td>Use Case</td><td>Profile Update</td></tr>
<tr><td>Goal</td><td>Entitle users to edit their details.</td></tr>
<tr><td>Precondition</td><td>User must be logged in.</td></tr>
<tr><td>Success End Condition</td><td>User’s profile is updated successfully.</td></tr>
<tr><td>Failed End Condition</td><td>Updates go wrong due to bad parameters or due to problems with the servers.</td></tr>
<tr><td>Primary Actors:<br>Secondary Actors:</td><td>Customer, Admin</td></tr>
<tr><td>Trigger</td><td>User comes to the user profile and modifies them.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table><tr><td>1.</td><td>User navigates to profile settings area.</td></tr><tr><td>2.</td><td>Changes name, email, and/or password.</td></tr><tr><td>3.</td><td>System writes and stores changes to the database.</td></tr></table></td></tr>
<tr><td>Alternative Flows</td><td><table><tr><td>1</td><td>If there’s any invalid input fed into the system, the system will notify the user to correct it.</td></tr><tr><td>2</td><td>In this situation: the URL of the current page is updated and the page is reloaded; in case of a server error – changes are not saved.</td></tr></table></td></tr>
<tr><td>Quality Requirements</td><td>Profile updates should update as when it’s required.</td></tr>
</table>

Case Description-05: Subscription and Plans

<table>
<tr><td>Use Case</td><td>Subscription and Plans</td></tr>
<tr><td>Goal</td><td>Admin user create subscription plan.To show customers available subscription packages and allow them to choose from.</td></tr>
<tr><td>Precondition</td><td>User must be logged in.</td></tr>
<tr><td>Success End Condition</td><td>Admin user create subscription plan and user selects a subscription plan.</td></tr>
<tr><td>Failed End Condition</td><td>User is unable to view or select a plan.</td></tr>
<tr><td>Primary Actors:<br>Secondary Actors:</td><td>Customer<br>Admin</td></tr>
<tr><td>Trigger</td><td>It is then followed by selecting the “Price Plans” in the section.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1</td><td>Admin add plan</td></tr>
<tr><td>2</td><td>User views the list of plans..</td></tr>
<tr><td>3.</td><td>Chooses a plan and go to the payment section.</td></tr>
<tr><td>4.</td><td>System implement the selected plan.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1</td><td>If the selected plan is not available the user is informed that the desired plan is not available.</td></tr>
<tr><td>2</td><td>In case of inability to pay for the plan, the plan is not implemented.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Subscription details should load within 2 seconds.</td></tr>
</table>

Case Description-06: Templates

<table>
<tr><td>Use Case</td><td>Templates</td></tr>
<tr><td>Goal</td><td>Admin work with templates to ensure clients are presented with usable templates on their web projects.</td></tr>
<tr><td>Precondition</td><td>Admin and customer is logged in and customer has an active subscription.</td></tr>
<tr><td>Success End Condition</td><td>admin manage all templates and customer selects and customizes a template.</td></tr>
<tr><td>Failed End Condition</td><td>User is unable to load or customize templates.</td></tr>
<tr><td>Primary Actors:<br>Secondary Actors:</td><td>Customer, Admin</td></tr>
<tr><td>Trigger</td><td>It is then followed by selecting the “Templates Library” in the section.</td></tr>
<tr><td>Description / Main Success Scenario</td><td>
<table>
<tr><td>1</td><td>User becomes aware of the set of templates.</td></tr>
<tr><td>2</td><td>Chooses a layout and adapts in to it.</td></tr>
<tr><td>3.</td><td>Saves the changes and studies their result on the website.</td></tr>
</table>
</td></tr>
</table>

| | |
|---|---|
| Alternative Flows | <table><tr><td>1</td><td>If the template fails to load, the user is notified.</td></tr><tr><td>2</td><td>If customization changes are invalid, they are not saved.</td></tr></table> |
| Quality Requirements | Templates details should load within 2 seconds. |

Case Description-07: User List

| | |
|---|---|
| Use Case | User List |
| Goal | To enable the admin to review, filter and manage the list of users who register with the system. |
| Precondition | Admin, the responsible individual, has to be logged in with the appropriate security privileges.. |
| Success End Condition | Admin can view, search, or update user details successfully. |
| Failed End Condition | User list cannot be accessed by Admin or subjected to management actions as a result of some errors or permissions. |
| Primary Actors:<br>Secondary Actors: | Admin |
| Trigger | Admin chooses the Client List option on the local interface or the administration panel. |

<table>
<tr><td>Description / Main<br>Success<br>Scenario</td><td><table>
<tr><td>1</td><td>Admin then gets a menu with options and then choose the “User List”.</td></tr>
<tr><td>2</td><td>System shows all the users who registered with the system.</td></tr>
<tr><td>3.</td><td>Admin performs search or filter to try and locate a particular user.</td></tr>
<tr><td>4</td><td>Admin change or delete user data whenever is necessary.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1</td><td>In any case the system produces no results, the admin is informed.</td></tr>
<tr><td>2</td><td>For example, if the admin attempts to perform some actions which they are not authorized for, the system protests the action and notifies the admin with an error message.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>User list details should load within 2 seconds.</td></tr>
</table>

Case Description-08: System Settings

| Use Case | System Settings |
|---|---|
| Goal | To allow the admin and customer to configure and manage system-wide settings by own permission. |
| Precondition | Admin and customer have to be connected to their accounts while having the proper permissions. |
| Success End Condition | System settings are updated and applied successfully. |
| Failed End Condition | System settings are also left as they are because of errors or invalid input given by any person. |

<table>
<tr><td>Primary Actors:<br><br>Secondary Actors:</td><td>Admin, Customer</td></tr>
<tr><td>Trigger</td><td>Admin and customer accesses the "System Settings" option from the dashboard.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1</td><td>Admin and customer selects the "System Settings" option.</td></tr>
<tr><td>2</td><td>System displays configurable settings.</td></tr>
<tr><td>3.</td><td>Admin and customer makes the necessary changes and saves them by own permission.</td></tr>
<tr><td>4</td><td>System checks the input and sets the validation according to the settings.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1</td><td>In case the recorded input is invalid, the system informs the admin and requires him/her to input the correct data.</td></tr>
<tr><td>2</td><td>If an update fails the system returns an error to the admin and records the failure in its log file.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>System settings should load within 2 seconds.</td></tr>
</table>

Case Description-09: Guidance and Support

| Use Case | Guidance and Support |
|---|---|
| Goal | To help users understand how to use the platform. |
| Precondition | None |
| Success End Condition | User views and applies the tutorial successfully. |
| Failed End Condition | Tutorials fail to load or are unclear. |

<table>
<tr><td>Primary Actors:<br><br>Secondary Actors:</td><td>Admin, Customer</td></tr>
<tr><td>Trigger</td><td>Admin and customer accesses the "Tutorial" option from the dashboard.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table><tr><td>1</td><td>User selects a tutorial.</td></tr><tr><td>2</td><td>Completes steps outlined in the tutorial.</td></tr></table></td></tr>
<tr><td>Alternative Flows</td><td><table><tr><td>1</td><td>In case the tutorial has not been loaded, the user is informed of the same.</td></tr><tr><td>2</td><td>In this case, if the tutorial is not well understood then some feedback is obtained from the users.</td></tr></table></td></tr>
<tr><td>Quality Requirements</td><td>Tutorial should load within 2 seconds.</td></tr>
</table>

Case Description-10: Custom Domain Integration

| Use Case | Custom Domain Integration |
|---|---|
| Goal | To enable users to link a custom domain to their website. |
| Precondition | User must have an active subscription also admin is logged. |
| Success End Condition | Custom domain request sent to admin panel. |

<table>
<tr><td>Failed End Condition</td><td>Custom domain request not sent due to incorrect configurations.</td></tr>
<tr><td>Primary Actors:<br><br>Secondary Actors:</td><td>Customer,Admin</td></tr>
<tr><td>Trigger</td><td>User decides to choose the “Domain Request” and types in the necessary domain information. Admin choose the “Domain Request” tab to view all the domain request made by the customer.</td></tr>
<tr><td>Description / Main<br><br>Success<br><br>Scenario</td><td><table><tr><td>1</td><td>It is the user who must enter the domain name.</td></tr><tr><td>2</td><td>Sets DNS setting according to the directions given.</td></tr><tr><td>3</td><td>Domain is then checked and associated by the system.</td></tr></table></td></tr>
<tr><td>Alternative Flows</td><td><table><tr><td>1</td><td>If DNS settings are inaccurate the user is informed by the system.</td></tr><tr><td>2</td><td>If the domain is also taken, the user will be asked to resolve the problem.</td></tr></table></td></tr>
<tr><td>Quality Requirements</td><td>Domain linking should complete within 24 hours.</td></tr>
</table>

### 2.4.3 Activity Diagram

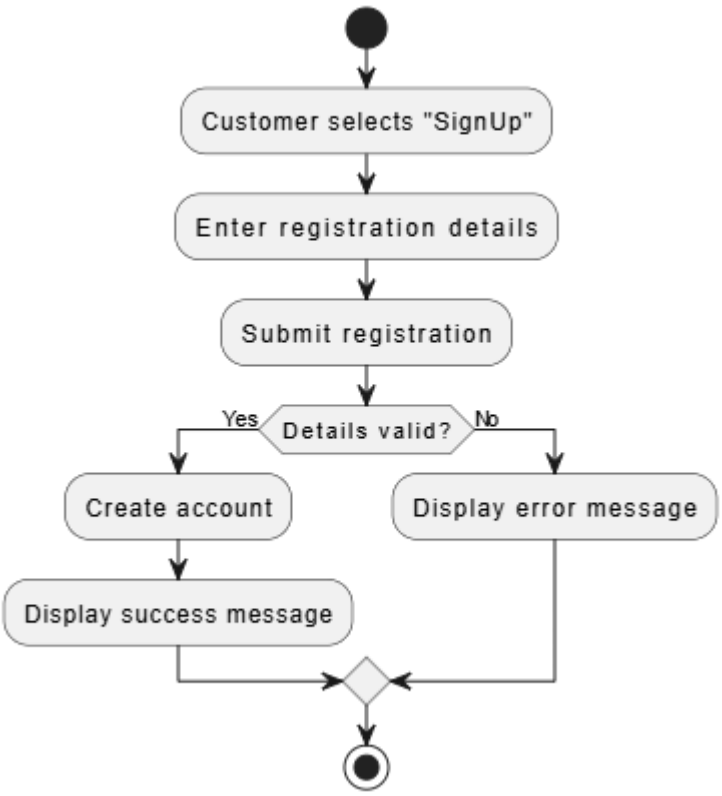



Figure 4: Activity Diagram Customer Registration

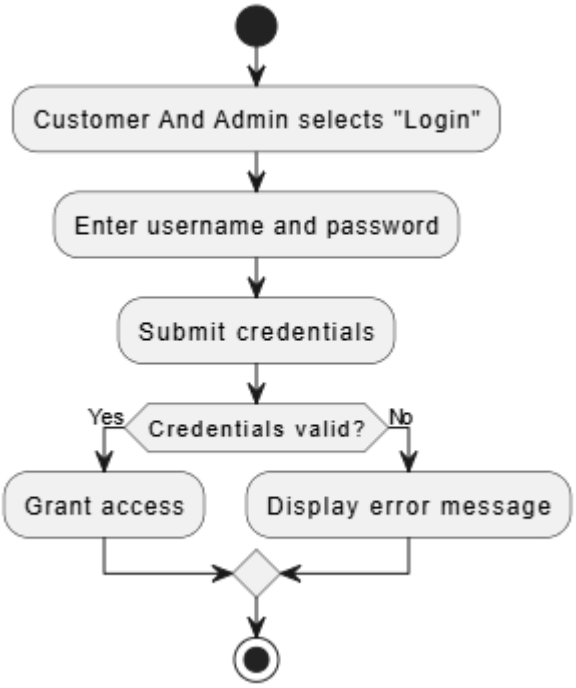



Figure 5: Activity Diagram Admin and Customer Login

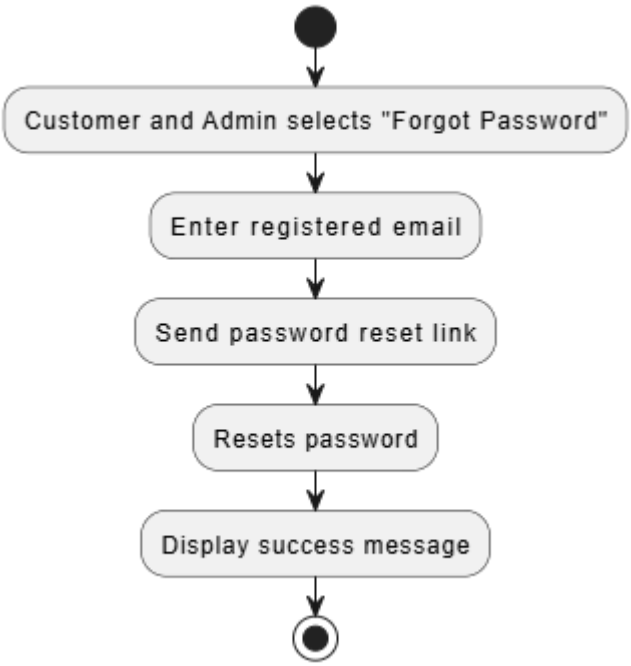



Figure 6: Activity Diagram Admin and Customer Forget Password

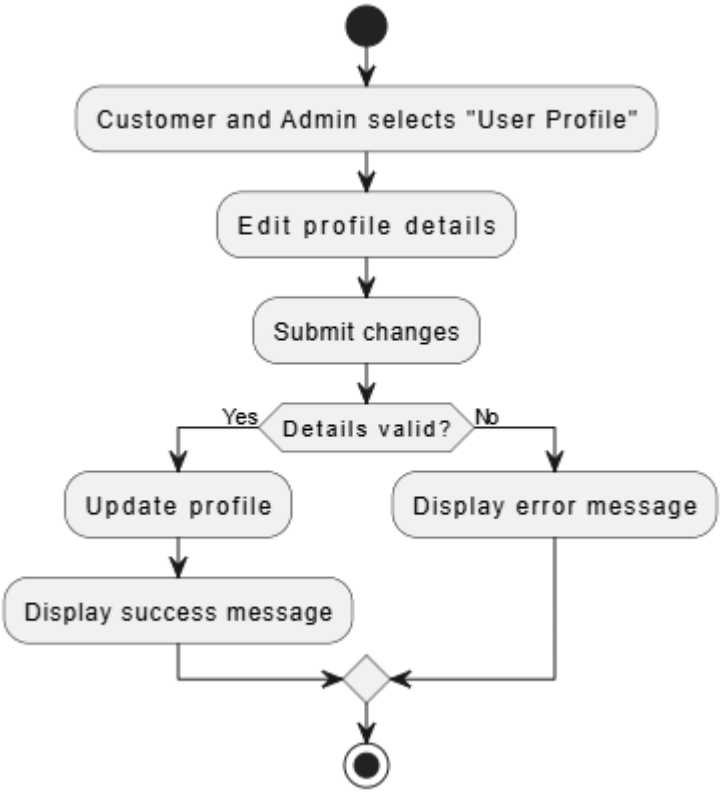



Figure 7: Activity Diagram Admin and Customer Profile Update

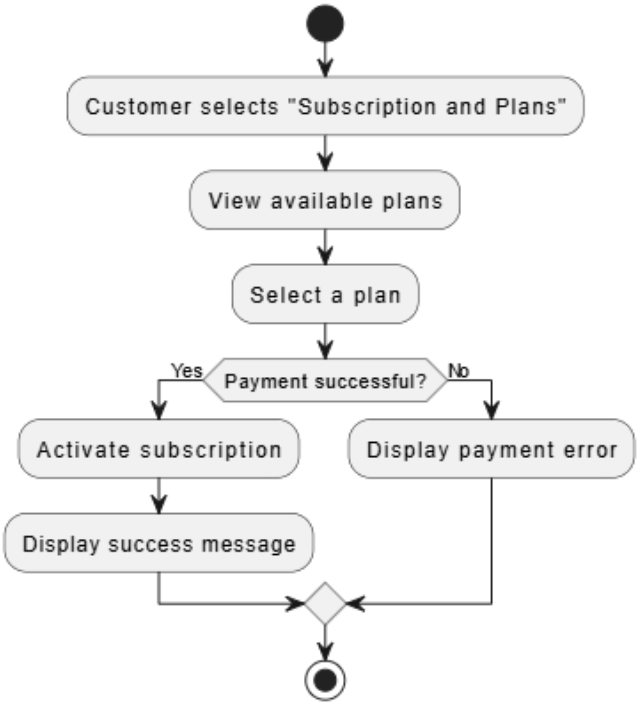



Figure 8: Activity Diagram Customer Subscription and Plans

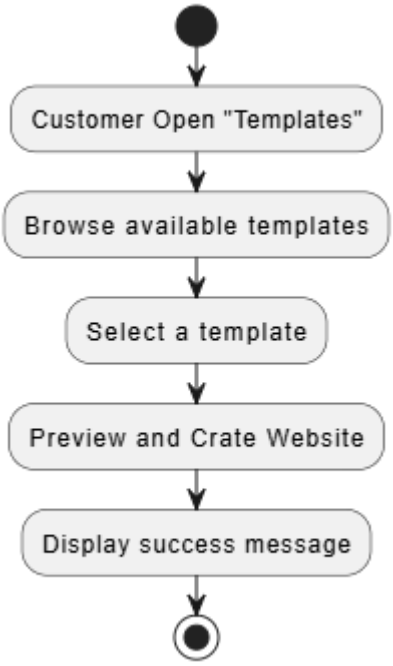



Figure 9: Activity Diagram Customer Templates

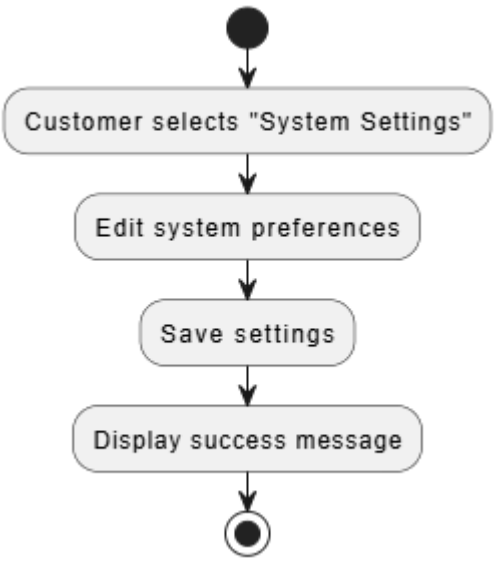



Figure 10: Activity Diagram Customer System Settings

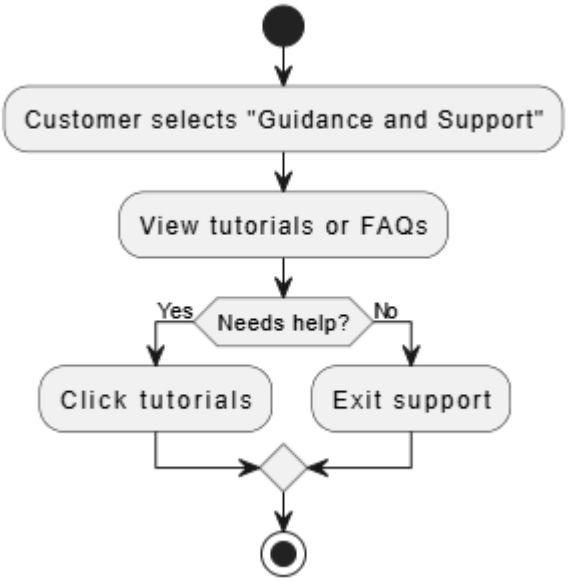



Figure 11: Activity Diagram Customer Tutorials

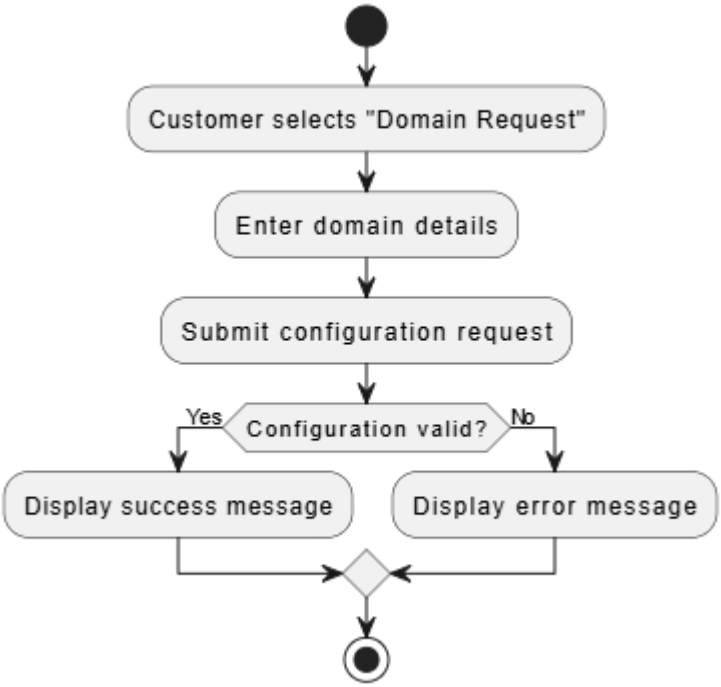



Figure 12: Activity Diagram Customer Domain Request

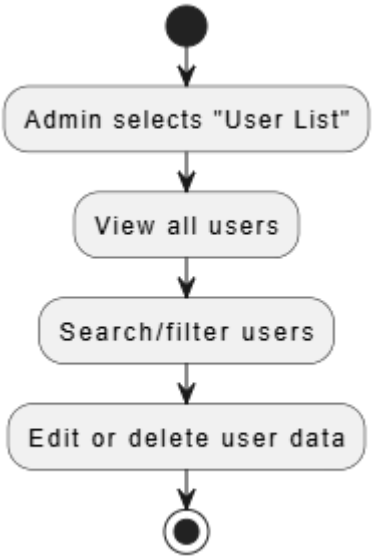



Figure 13: Activity Diagram Admin User List

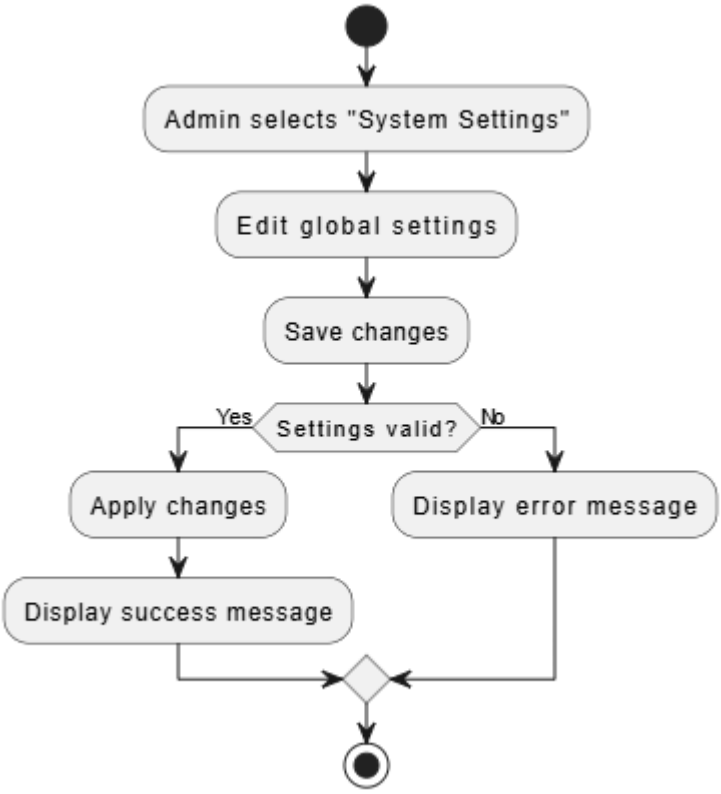



Figure 14: Activity Diagram Admin System Settings

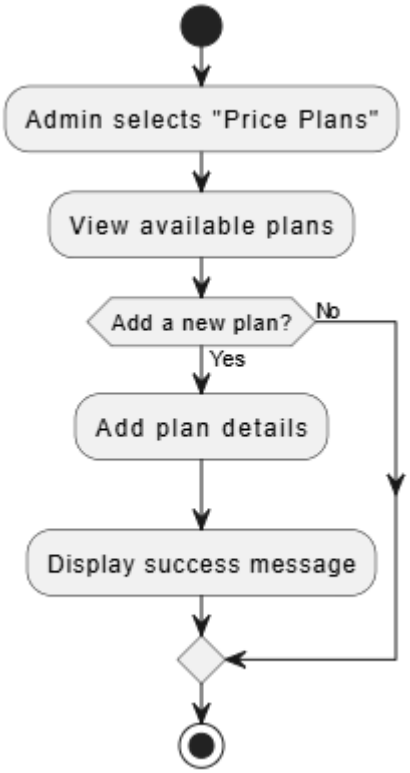



Figure 15: Activity Diagram Admin  Price Plan

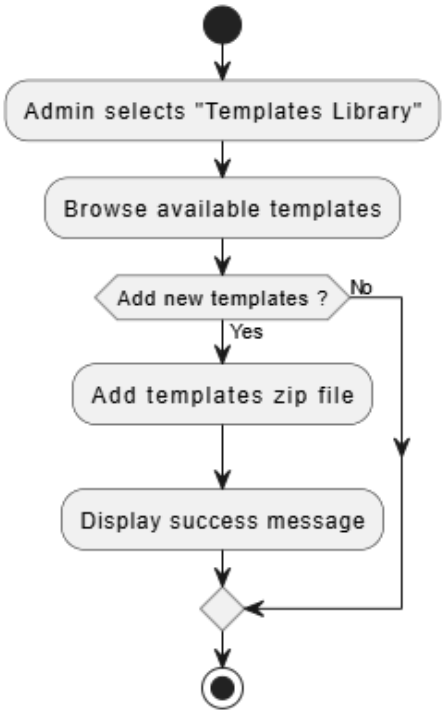



Figure 16: Activity Diagram Admin Templates Library

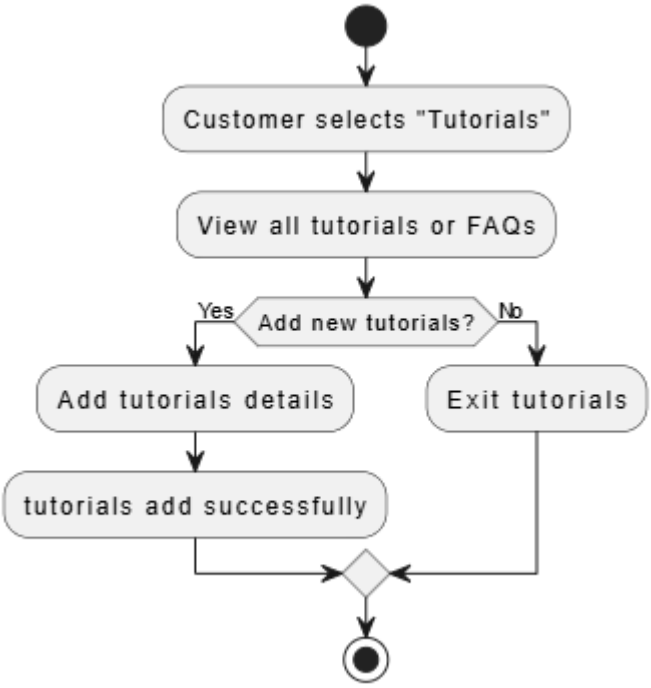



Figure 17: Activity Diagram Admin Tutorials

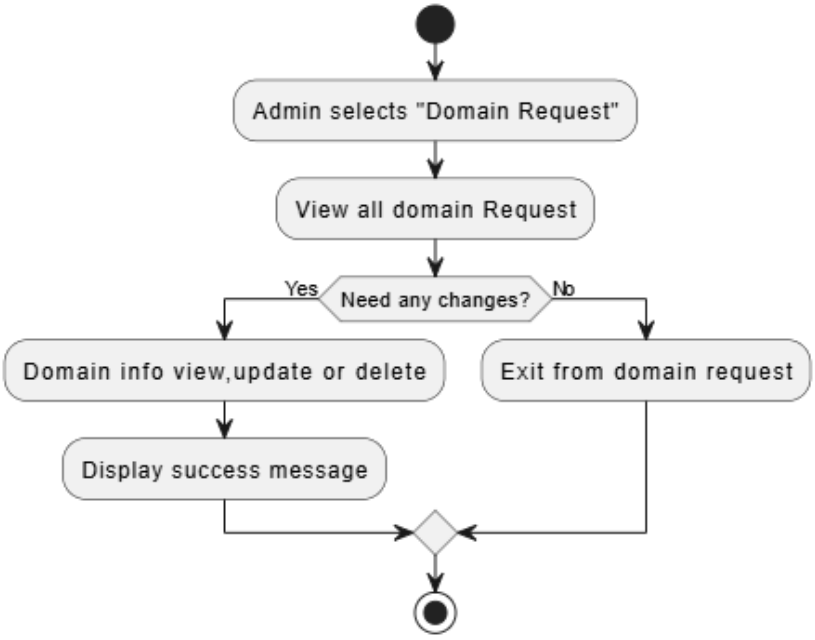



Figure 18: Activity Diagram Admin Domain Request

### 2.4.4 Sequence Diagram

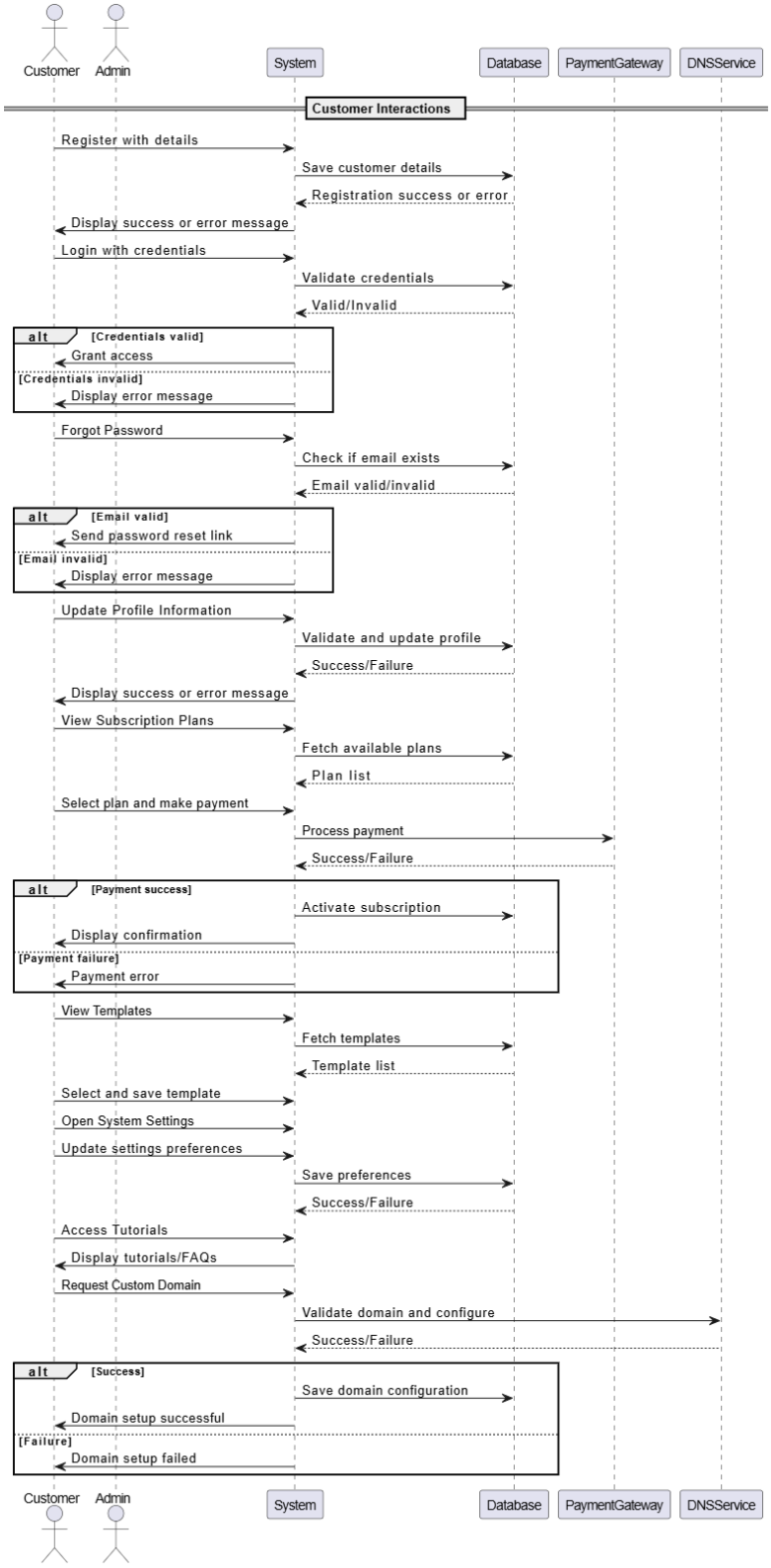



Figure 19: Sequence Diagram Customer

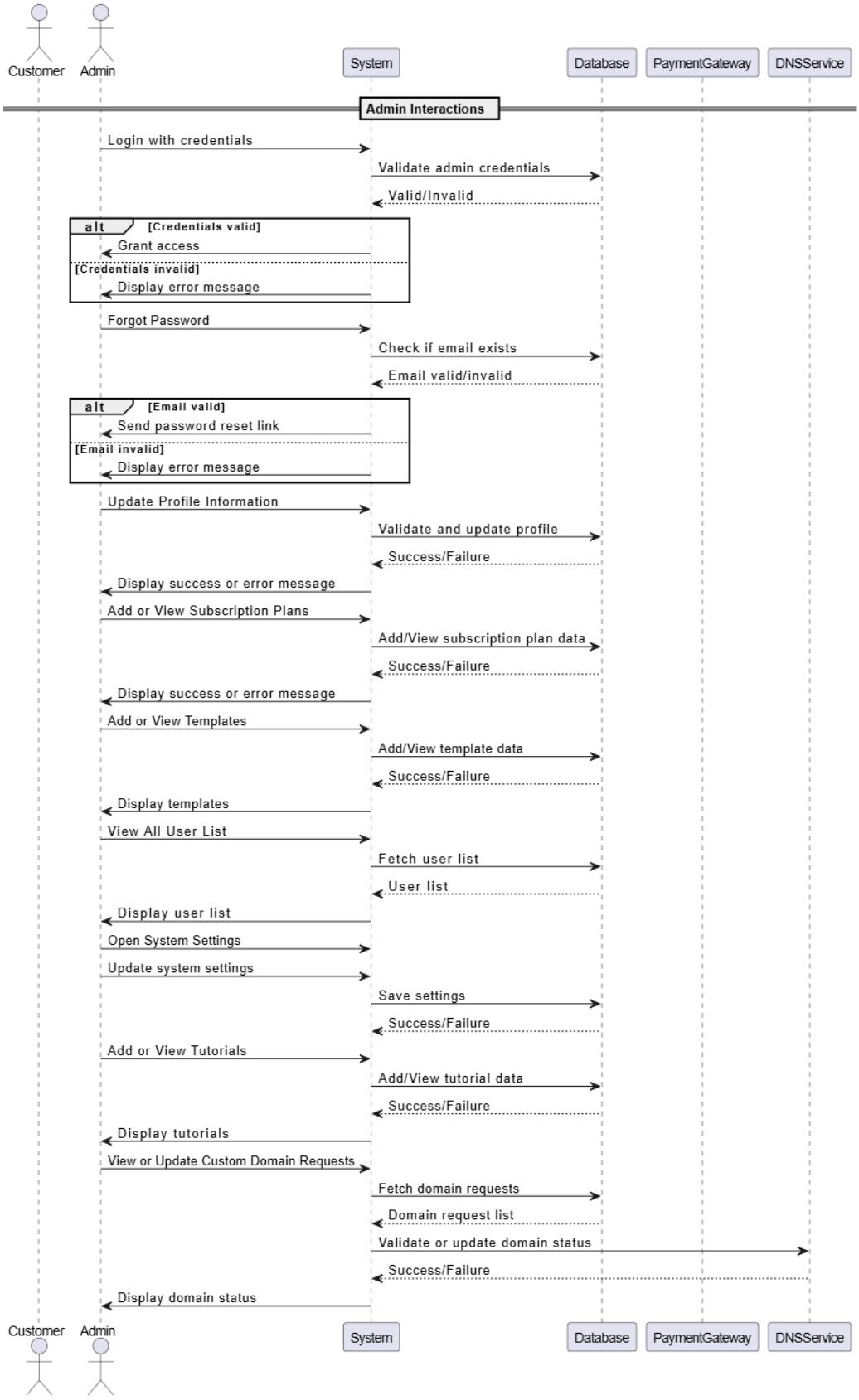



Figure 20: Sequence Diagram Admin

### 2.4.5 Class Diagram

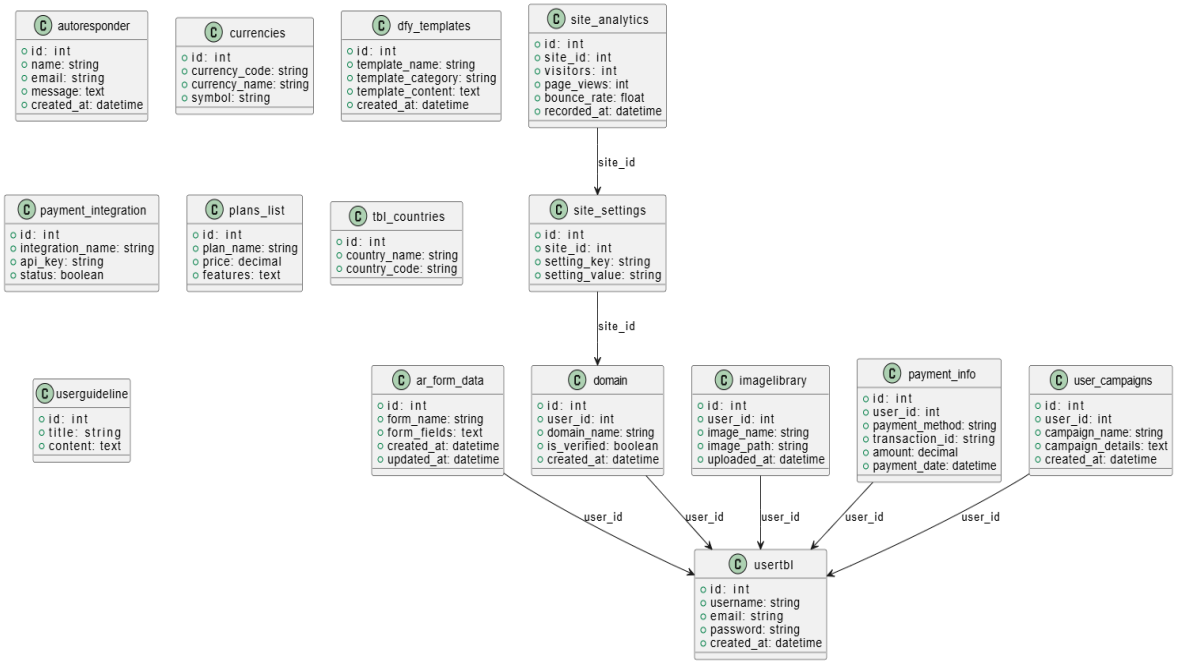



Figure 21: Class Diagram

## 2.4.6 ER Diagram

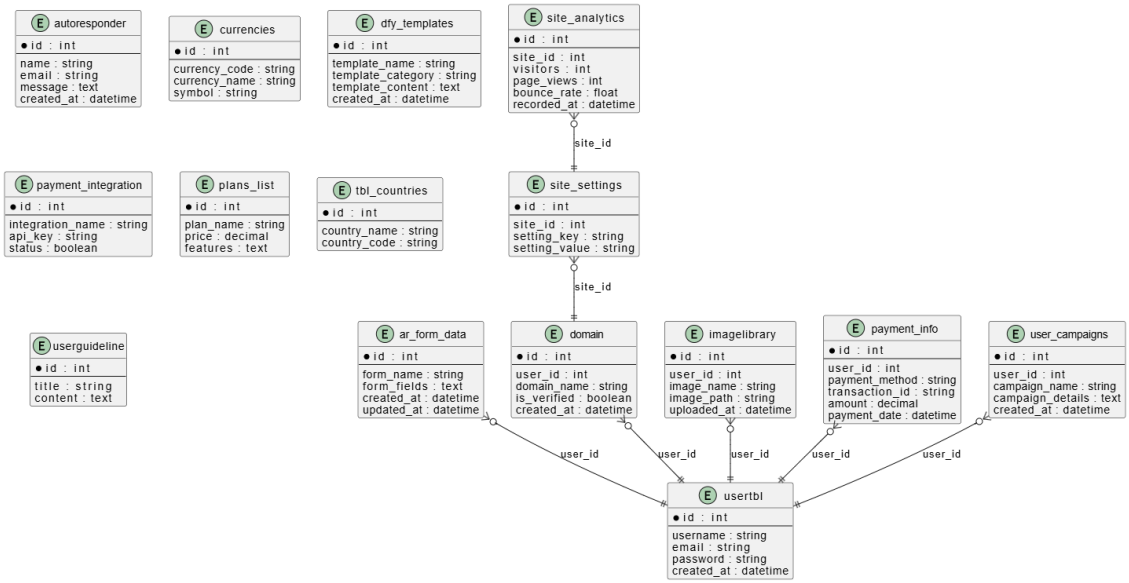



Figure 22: ER Diagram

## 2.5 Coding: Appendix A

**Sample code for Registration**

```
<!--Sign-Up Page Start-->

<div class="pxg_login_main">

        <div class="pxg_auth_login_box">

           <div class="pxg_center_auth_bg">

              <form id="signup_form" data-posturl="authenticate/verifySignUp"
autocomplete="off" >

                <div class="pxg_login_wrapper pxg_signup">

                  <center><a href="<?= base_url() ?>" class="pxg_logo_ wow fadeInUp" data-
wow-delay="300ms" data-wow-duration="1500ms"><img src="<?=
base_url($_SESSION['site_logo'] ) ?>" alt="ltr_sign_up_alt_2"></a></center>

                  <div style="overflow: hidden;">

                    <div class="pxg_auth_input_wrapper wow fadeInUp" data-wow-
delay="300ms" data-wow-duration="1500ms">

                      <div class="pxg_auth_input " >

                        <label><?php echo html_escape($this->lang->line('ltr_sign_up_txt_4'));
?></label>

                        <input class="require valid" type="text" name="pps_name" value=""
placeholder="Enter name here" data-error="Please Enter Full Name" autocomplete="off">

                        <div class="pxg_login_ic">

                          <img src="<?= base_url() ?>assets/images/user.svg"
alt="ltr_sign_up_alt_3">

                        </div>

                      </div>

                      <div class="pxg_auth_input " >

                        <label><?php echo html_escape($this->lang->line('ltr_sign_up_txt_5'));
?></label>

                        <input class="require valid valid_email" type="email"
name="pps_email" value="" placeholder="Enter email here" data-error="Please Enter Email"
autocomplete="off">

                        <div class="pxg_login_ic">

                          <img src="<?= base_url() ?>assets/images/envelope.svg"
alt="ltr_sign_up_alt_4">

                        </div>

                      </div>
```

```
                    <div class="pxg_auth_input">
                        <label><?php echo html_escape($this->lang->line('ltr_sign_up_txt_6')); ?></label>
                        <input class="require m_length_n" data-min_l="8" type="password" name="pps_password" value="" data-fieldname="Password" placeholder="Enter your password" data-error="Please Enter Password" autocomplete="new-password" >
                        <div class="pxg_login_ic">
                            <img src="<?= base_url() ?>assets/images/lock.svg" alt="ltr_sign_up_alt_5">
                        </div>
                    </div>
                </div>
            </div>
            <div class="pxg_btn_area wow fadeInUp" data-wow-delay="300ms" data-wow-duration="1500ms">
                <button data-action="submitMe" data-form="signup_form" class="pxg_btn">
                    <?php echo html_escape($this->lang->line('ltr_sign_up_txt_7')); ?>
                </button>
            </div>
            <p class="get_start wow fadeInUp" data-wow-delay="300ms" data-wow-duration="1500ms"><?php echo html_escape($this->lang->line('ltr_sign_up_txt_8')); ?><b> <a href="<?= base_url('login') ?>"><?php echo html_escape($this->lang->line('ltr_sign_up_txt_9')); ?> </a></b></p>
        </div>
    </form>
    </div>
  </div>
</div>
<!--Sign-Up Page End→
```

## 2.6 Summary

This chapter outlines a description of the way the Static Website Builder project was designed and developed. There are class/actor, activity, and ER diagrams to feature the relationships and integration of the different systems. Business and external requirements are well understood so as to meet the user requirements inclusive of the non-functional ones. The chapter also provides coding examples to illustrate much of the discussed functionality, as well as user registration, login and subscription. In the same manner, architecture requirements and constraints are presented, as well as the measurable attributes affecting both the inner structures and outer appearances of a piece of hardware and the programs that run it.

# Chapter 3 Software Testing

## 3.1 Introduction

Like same with majority of the contemporary software projects, a software like the Static Website Builder requires a phase where it has to be tested in order to determine how effectively it fits into functional, usability, performance as well as the security project requirements. Functionality testing verifies the chosen templates, verifies that the ability to drag and drop elements is functioning alright, and if content has been integrated rightly. Usability tests seek to determine whether the graphical user interface is understandable or not to persons – including specialists and novices such as the author of this paper.

## 3.2 Testing Features

### 3.2.1 Feature to Be Tested

a. Customer Registration
b. Customer Login
c. Customer Website Create
d. Customer Website Download
e. Admin Login
f. Admin Dashboard
g. Add Templates (Admin)
h. Customer Logout
i. Admin Logout

## 3.3 Testing Strategies

### 3.3.1 Test Approach

The test strategy for the Static Website Builder guarantees thorough test of functionality, usability, performance, security and compliance. Ad-hoc testing will prove task such as template selection, drag and drop capabilities, and publishing of website. Automation testing using Selenium will then perform regression or any repetitive test. Functional testing checks that necessary processes work properly and performance testing, with a tool like JMeter, determines how quickly and lightly the site can be produced. Security testing helps in identifying the possible issues that may spoil the content so that it can be protected form such threats. Usability testing is based on the concept of interface friendliness while compatibility testing covers the issue of the product working correctly on different browser and devices. This approach serves as a kind of insurance for building a stable, user-optimized CMS.

### 3.3.2 Pass/Fail Criteria

Pass/Fail criteria for testing the Static Website Builder determine the set of conditions under which a particular test can be considered completed successfully or unsuccessfully.

1. Functional Testing

Pass: Templates, drag-drop mechanism, and publishing paradigms fit all our needs and expectations towards BMC Remedy.

Fail: Every primary function goes awry or does not operate as planned.

2. Usability Testing

Pass: As common with most interface designs, the user does not face a lot of problems when executing tests, and this is arising from a clear and easy to understand interface.

Fail: End-users are involved in the identification of the problems experienced in the system and the users are confused, make errors or less efficient in the use of the system.

3. Performance Testing

Pass: Some results regarding the distribution of the web-site generation time also indicate that it stays within healthy bounds which means that the system under test does not degrade in handling the expected loads.

Fail: The choice of architecture is characterized by low speed, stalling or poor scalability where the load is expected.

4. Security Testing

Pass: Some examples of such weaknesses are: None vulnerabilities such as XSS, CSRF are discovered the system implements correct user authentication and authorization.

Fail: All the possible exploit targets or security breaches of confidential content are detected.

5. Compatibility Testing

Pass: It functions harmoniously in all targeted browser environments, devices and operating systems.

Fail: One or more of the client's platforms does not behave as it should or render client's content incorrectly.

6. Regression Testing

Pass: I know that new changes cannot hurt existing functions or bring them back.

Fail: Current functionalities fail after updates or features inclusion.

## 3.4 System Testing (Test Cases with Report)

**Test Case 01: User Registration**

| # | Name | Email | Password | Response | Pass/Fail | Comment |
|---|---|---|---|---|---|---|
| 1 | Saharuk | Saharukkarim559@diu.edu.bd | 110900Sk@ | Registration successful | Pass | Registration successful with valid info |
| 2 | | Saharukkarim559@diu.edu.bd | 110900Sk@ | Please Enter Full Name | Fail | User must enter full name |
| 3 | Saharuk | Saharukkarim559@diu.edu.bd | | Please Enter Password | Fail | User must enter password |
| Post-condition: The user is successfully registered and the registration process is considered successful with valid information. | | | | | | |

## 3.5 Summary

This chapter therefore explicates on the role and relevance of software testing as a measure of enhancing the quality and suitability of an application software. The reader is introduced to fundamental testing approaches like manual and automated testing with a view to testing functional, performance, security, and usability aspects of a system.

From such examples as customer registration scenarios, this chapter shows how a structured test case can identify problems and verify the operation of the system. This systematic approach makes it possible to deliver a high-quality and friendly to users software solution.

# Chapter 4 Deployment and Maintenance

## 4.1 Introduction

This chapter specifically covers two stages of the software development process: Deployment and Maintenance. Deployment is the process of getting the application ready

for release and making the necessary settings on the environment ready for the application to run. The maintenance phase helps to keep the software running and protected and continually update it when adopted to remove the bugs with the additions of new features that enhance the whole performance of the software. These areas of interest encompass deployment procedures, instruments, management practices, problem solving, and end user feedback. Thus this chapter underlines the potential of the software and acknowledges the need for a good deployment strategy and regular updates for a successful faithful usage of the software in the real world.

## 4.2 Try to follow the SRLC (software release life cycle)

The publication and evolution model of the Static Website Builder is governed by the Software Release Life Cycle (SRLC) plan. Below is a breakdown of the SRLC phases tailored for this project:

**1. Pre-Alpha**

Objective: Preparation and basic features which are common for majority of application development.

Activities:

- What is required is to define the architecture of the system and the schema for the database.
- Some of the basic use cases to apply are selecting templates, the drag drop UI and the static file management.
- Testing that is performed, primarily by developers, to verify that the simplest functionality of the system is operable.

**2. Alpha**

Objective: Almost instant assessment and feedback gathering.

Activities:

- Making the application available to a few people (a restricted internal team, or a selected group of testers).
- Writing pixel perfect, cross-browser HTML and CSS markups.
- Performing usability test and including early feedback in the process.

**3. Beta**

Objective: More testing with real-life participants.

Activities:

- Expanding the customers of this CMS to freelance and entrepreneurs who can benefit from it.
- Surveilling the outcomes of the system performances and acceptability of the whole system, including its reliability.

- Responding to concerns that the application does not meet its users' expectations, adjusting the offer for a better match in a target audience's needs, etc.
- Testing under conditions that mimic traffic from other Web site users.

**4. Release Candidate (RC)**

Objective: Downloading the District CMS in preparation for public use.

Activities:

- Completing all applied features and verification of the project according to the established rules.
- It has to do with performing regression and compatibility testing.
- Providing writing services that can be used to document the software in the project; such as the users' manuals and technical manuals.

**5. General Availability (GA)**

Objective: Official launch of the CMS.

Activities:

- Images and other files running on production servers are in a format for use by browsers, not the editors themselves; Images, other files and content are in a format for use by browsers, not the editors themselves but the CMS to production servers.
- Marketing it to be available to the target group.
- To continuously check on the initial uptake for any problems which may arise after the launch.

**6. Maintenance and Updates**

Objective: Dr. D'Souza explained how he has to maintain and further develop the CMS over the years.

Activities:

- System evaluation, based on the overall performance comparing the results on the monitoring and analysis of system informing and the collected feedback.
- Error and security holes corrections.
- Implementing improvements and new options in the fulfillment of their requirements.

# Chapter 5 User Manual

## 5.1 Introduction

The User Manual is designed to give users adequate information as well as support when using the Static Website Builder. This chapter will showcase detailed procedures that augur for critical functionalities including template selection, website building through drag and drop interface, content management and publishing of static sites and others. These features also consist of troubleshooting methods, often asked questions, and specific guidelines to make particular specifications for better user experience. Even though the target audience of the manual is technical and non-technical users, the author pays much attention to the components' simplicity and clearness, allowing the user to implement the possibilities of the CMS.

## 5.2 Project Functionalities

**Landing Page**

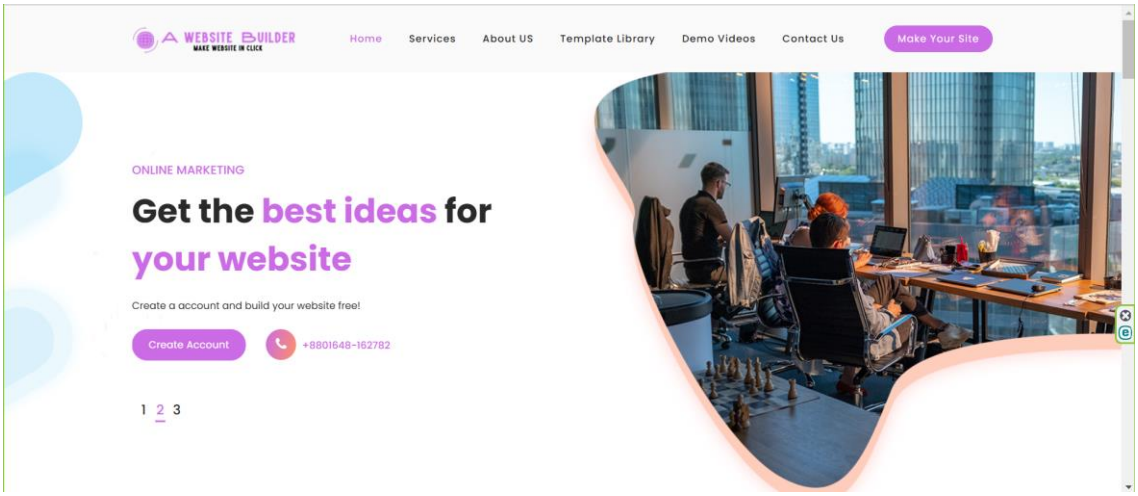



**Figure: Landing Page**

**User Registration**

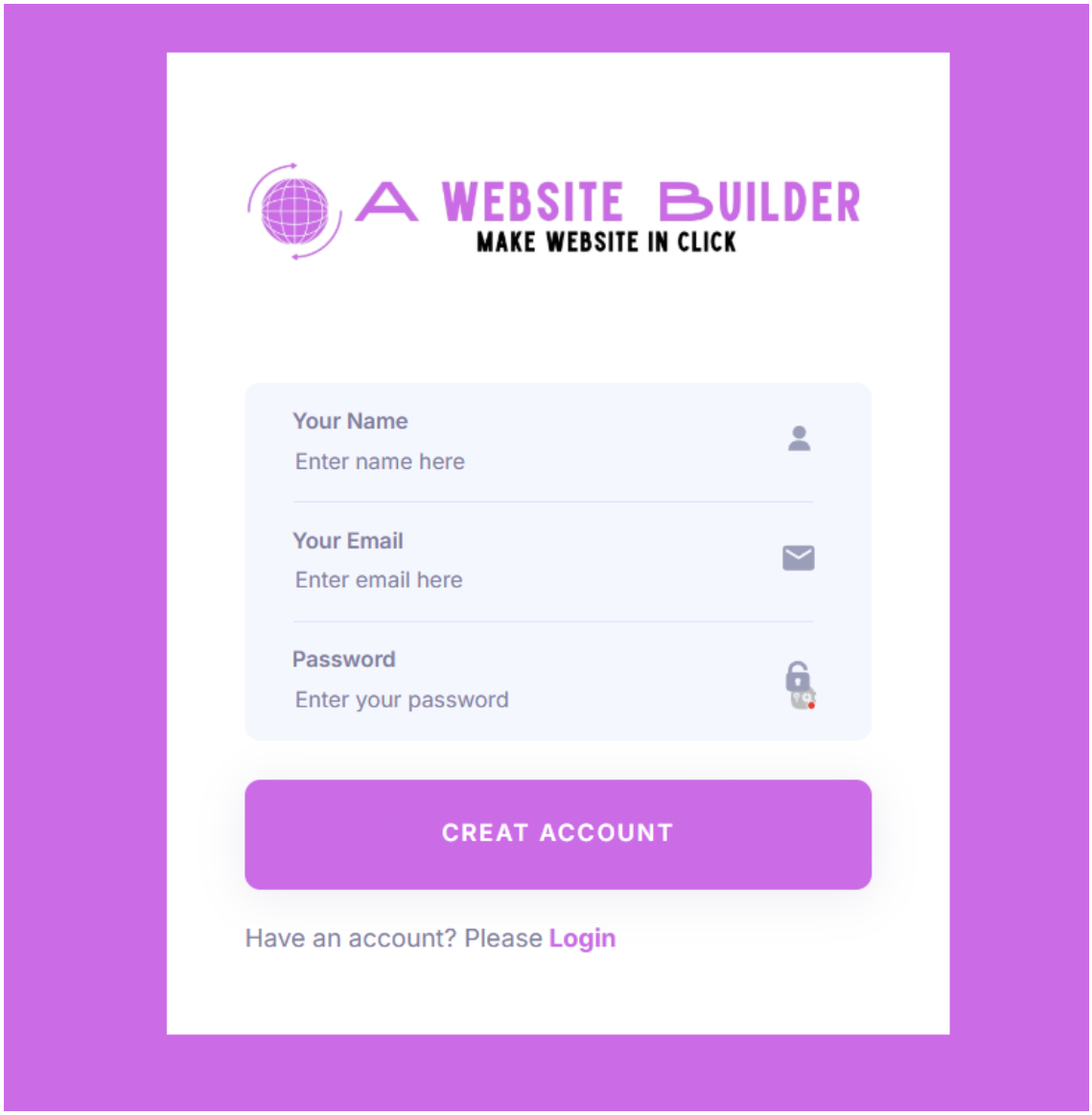



**Figure : User Registration**

**Admin & Customer Login**

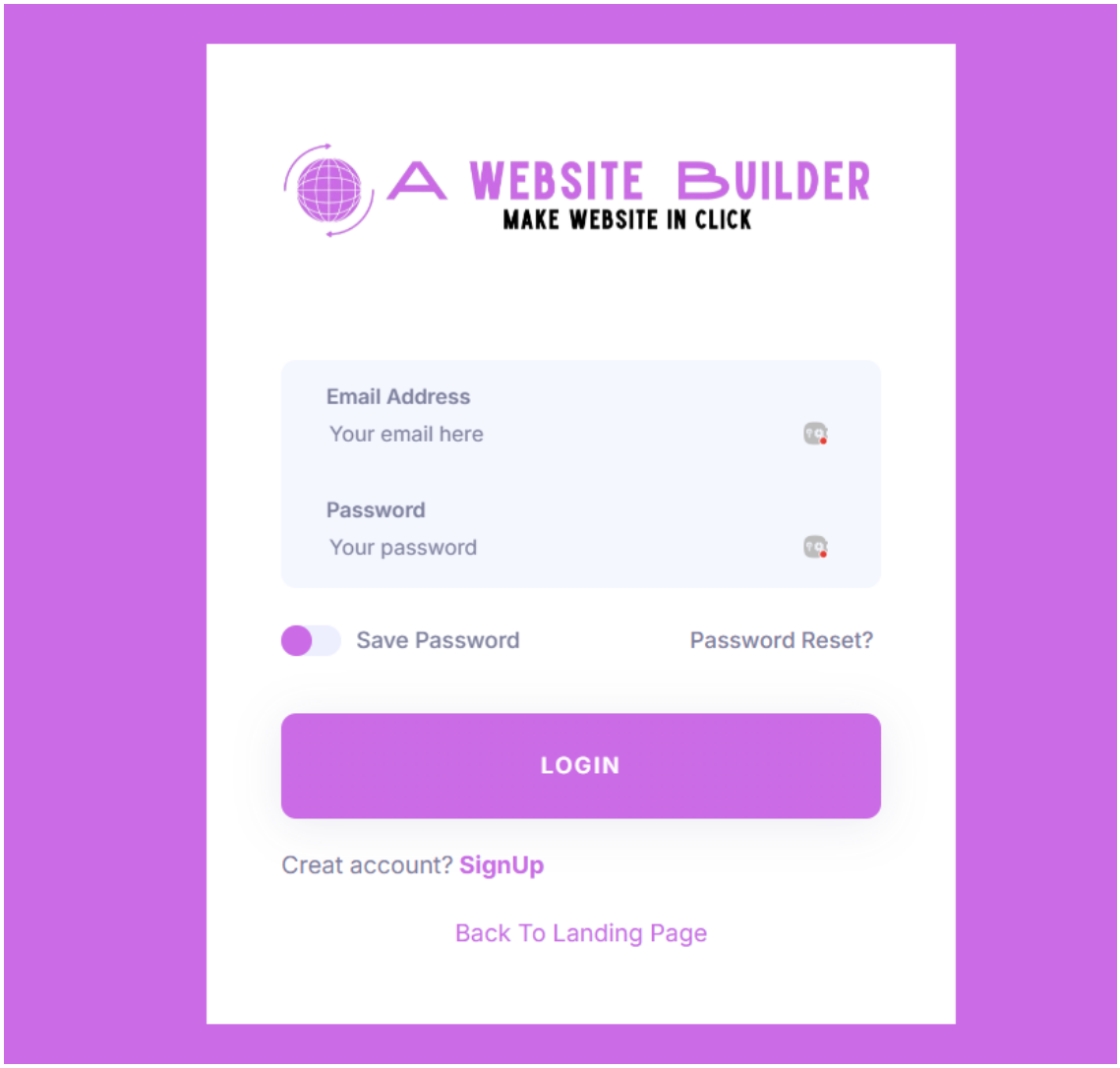



**Figure : Admin & Customer Login**

**Customer Dashboard**

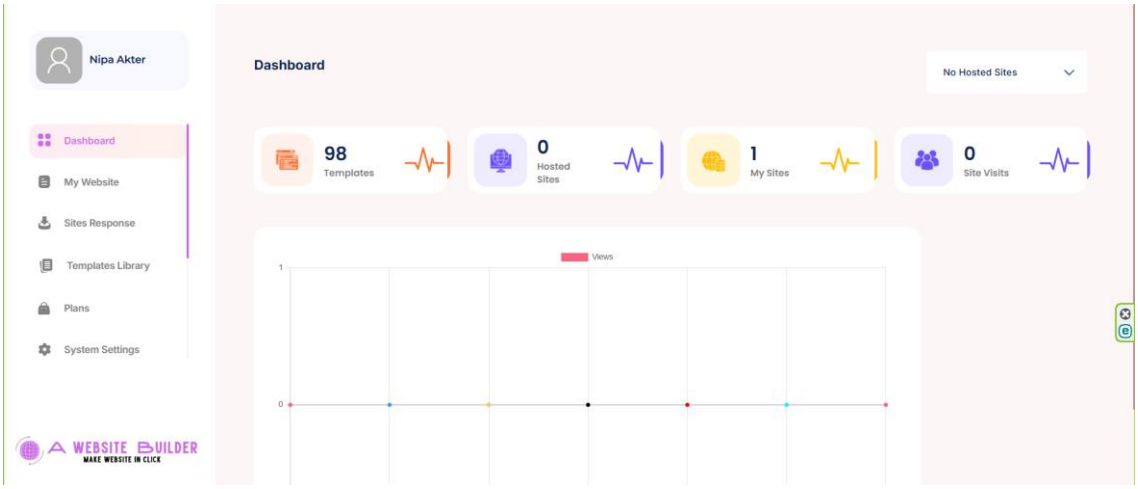



**Figure : Customer Dashboard**

**Customer Website Create**

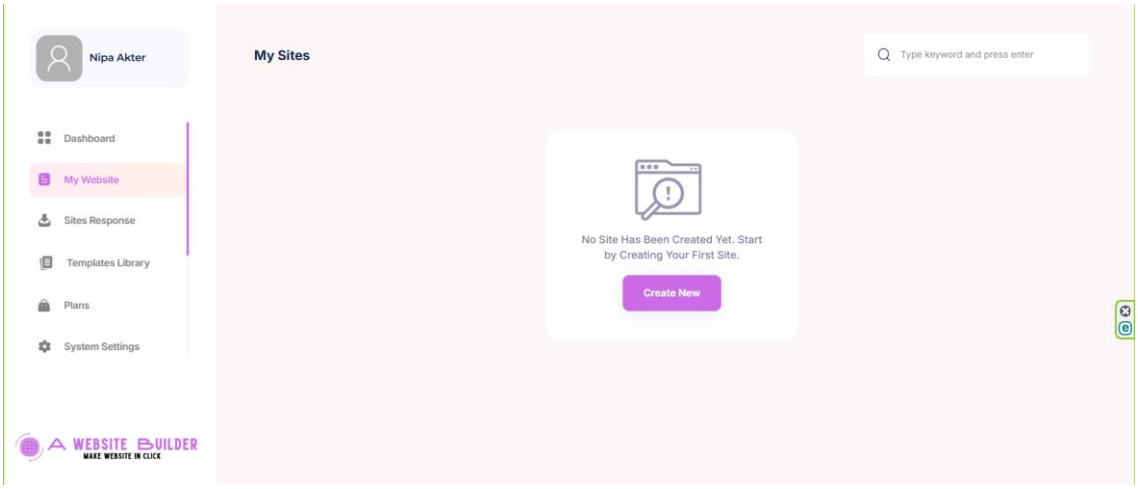



**Figure : Customer Website Create**

## 5.3 Summary

This chapter is actually chapter which is a step by step tutorial on how to use the Static Website Builder making it easy for all type of users even those with little or no technical

skills. For the purpose, it includes comprehensive guidelines for important activities including choosing a template, using the drag and drop functionality, working with content, and publishing a site. In addition to these instructions, the manual contains illustrations and pictorial depictions, problem-solving guidelines, and tips, and advice on optimising the use of the device.

# Chapter 6 Project Summary

## 6.1 Introduction

Here, the main goals, essential functions, work on creating the Static Website Builder project, and its results are briefly described. It describes the fundamental aspects of the system in terms of its use, for instance, creating and managing of simple websites effectively. Also, the processes, methods, tools, techniques, and technology utilized in the development process, testing/ deployment techniques are also discussed in the chapter. In its capacity as an overview and summary of the project's findings and implications, this chapter draws out key lessons from past experiences to gauge the effects of the project as well as suggest areas for further research.

## 6.2 Project Limitation

The Static Website Builder faced several limitations and constraints during its development:

**1. Time Constraints**

The time available for the project implementation was a constraint; therefore, the efforts to implement and optimize features were somewhat constrained.

A few features like auto SEO suggest or ability to work in multiple languages were left out in the basic version.

**2. Budget Constraints**

This led to scarcity of resources in terms of additional third party libraries, professional UX/UI testers and cloud hosting solutions for deployment testing since they are expensive due to limited capital base.

Few flutes like connecting with superior enabling tools were excluded to cut some corners.

**3. Technological Constraints**

The technologies have been selected stably, thus PHP (CodeIgnite) give limitations in compare with modern technologies such Node.js or some frameworks based on Python.

Another key finding was that the system employs manual deployments for deployment since automated DevOps pipelines were not well established.

**4. Unaddressed Requirements**

Specified characteristics from the list below were considered as possible improvements but were not included in this release: Rich features for the dynamic content, real-time capabilities, tight integration with handheadless CMS platforms.

WCAG compliance was an issue in which , in order to eliminate general accessibility tests, they were not performed thoroughly owing to resource constraints.

**5. Incomplete Features**

- Some components are implemented partially: for instance, the extendibility of drag and drop tools and the granularity of the design paradigm of default designs.
- Currently, there is no web traffic monitoring analytics dashboard which remains in the prototype stages and was not incorporated into the program offerings.

## 6.3 Scope

**Inclusions (Project Coverage)**

The CMS associated with the Static Website Builder is aimed at giving easy access to users to create and manage the static websites by using the simple and lucid multimedia templates. The project includes the following features and modules:

- Template Selection: A set of templates that users select and further modify for their purposes or for a particular market segment.
- Drag-and-Drop Editor: Enables the users to create, drag and drop and edit the components of the website without programming knowledge.
- Content Management: Extension required to write or modify text, picture and the multimedia files.
- Static Site Generation: Takes the users' designs and content and compiles them into basic HTML files that are easily downloadable and require little server space.
- Publishing Tools: It is designed to allow for deploying static files to be hosted on web servers or services such as GitHub Pages, or Netlify.
- User Authentication: provide a secure registration and login for the users for them to be able to manage there projects.
- Basic Analytics: This was followed by the first prototype of the system that allows to monitor the simplest parameters, such as the number of views of the corresponding Net website.

**Exclusions (Out of Scope)**

The project excludes certain advanced features and functionalities, including:

- Dynamic Content: The CMS is capable only of managing static Web sites and does not incorporate dynamic features like server side scrape and live database.
- E-Commerce Integration: Some of the options, which are not offered, are a shopping card, a payment gateway, or an inventory tracking system.
- Advanced SEO Tools: It with no recommendation of SEO and there is no provision of more complex metadata than the normal text fields.
- Third-Party Integrations: Such connections as integrating with other external applications like the CRM systems, marketing platforms are not supported.
- Collaborative Editing: There is no functionality for real-time collaborative shared spaces in the current project's design.

## 6.4 Future Work

**Performance Optimization:**

- Probably one of the most important areas for improvement to make the static site generator more efficient is to minimize build times.
- Persist allowing the direct deployment to popular cloud solutions, such as AWS or Netlify.

**Advanced Analytics:**

- Create a statistical dashboard to view more precise data about visitor behavior and traffic source.
- Integrate effectively with other tools from third-party analytics such as google analytics.

**Mobile and Offline Capabilities:**

- Mobile apps should be created for instant website management.
- Must allow the full offline editing and when the device reconnects the work done should update the online version.

**AI and Automation:**

- They are also to use the AI tools for recommending content, for SEO optimization, and for the design of the site.
- Conduct a review on AI automated website creation of contents and structure.

## 6.5 Conclusion

To a large extent, the Static Website Builder achieved its main objective as a tool enabling users to easily create and maintain static web sites. The key features were achieved for the project like the WYSIWYG drag and drop editor, templates with customization controls, static file generation and many more. In terms of functionality, basic components of the system, such as content management, user authorization and basic analytics were implemented successfully with the limits of the project scope.

**Key Achievements**

- User-Centric Design: The system comes with a good user interface meaning firmware developers who lack programming or coding skills in building websites can perform the task easily.
- Efficient Static Site Generation: The system generates high quality and fast loading html files for hosting on different platforms.
- Secure User Authentication: Login page and registration form are also very secure for protecting the data of the users.
- Scalable Architecture: The adaptable design of the architecture also permits future additions and connections.

**Lessons Learned**

- Importance of Planning: Other benefits which were realized included: clarity of the project deliverables and hence proper identification and management of the available time and resources.
- User Feedback Value: The usage of the system in initial testing allowed for recommendability issues to be identified, to avoid them affecting the final product.
- Adaptability in Development: Features were prioritized based on feasibility and user requirements for timely production and incorporation of the application in the university.

**Meeting Objectives**

The project completed the set objectives since the CMS is functional and address the targeted audience. Even such additional functions as dynamic content support and collaboration tools were left the future development of a system defines a stable base for updates. This type of project also highlights future opportunities and enhance/upcoming improvements or additions, through the practical knowledge from this method based on user-centered design, the concept of personalized iteration and development, and problem-solving prior to building issues.

# REFERENCES

Use a reference manager such as *Mendeley, EndNote* or any reference manager software to generate all your list of references here. Once all the references are included then apply *Caption for Reference* style.

Alexandrov, N., & Alexandrov, V. (2015). Computational science research methods for science education at PG level. Procedia Computer Science, 51(1), 1685–1693. https://doi.org/10.1016/j.procs.2015.05.305

Crooks, N. M., & Alibali, M. W. (2014). Defining and measuring conceptual knowledge in mathematics. Developmental Review, 34(4), 344–377. https://doi.org/10.1016/j.dr.2014.10.001

Gertler, M. S. (2003). Tacit knowledge and the economic geography of context, or the undefinable tacitness of being (there). Journal of Economic Geography, 3(1), 75–99. https://doi.org/10.1093/jeg/3.1.7

# Plagiarism Report

sk

ORIGINALITY REPORT

| 10% | 9% | 2% | 8% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | dspace.daffodilvarsity.edu.bd:8080<br>Internet Source | 2% |
| 2 | Submitted to Midlands State University<br>Student Paper | 2% |
| 3 | psm.ump.edu.my<br>Internet Source | 1% |
| 4 | Submitted to Asia Pacific University College of Technology and Innovation (UCTI)<br>Student Paper | <1% |
| 5 | umpir.ump.edu.my<br>Internet Source | <1% |
| 6 | Submitted to University of Greenwich<br>Student Paper | <1% |
| 7 | Submitted to Universiti Malaysia Pahang<br>Student Paper | <1% |
| 8 | www.drnguvenlik.com<br>Internet Source | <1% |
| 9 | dspace.cvut.cz<br>Internet Source | <1% |

| | | |
|---|---|---|
| 10 | capstoneguide.com<br>Internet Source | <1% |
| 11 | Submitted to Arab Open University<br>Student Paper | <1% |
| 12 | Submitted to Macquarie University<br>Student Paper | <1% |
| 13 | Submitted to University of Plymouth<br>Student Paper | <1% |
| 14 | odr.chalmers.se<br>Internet Source | <1% |
| 15 | Submitted to University of Suffolk<br>Student Paper | <1% |
| 16 | github.com<br>Internet Source | <1% |
| 17 | Submitted to KDU College Sdn Bhd<br>Student Paper | <1% |
| 18 | www.coursehero.com<br>Internet Source | <1% |
| 19 | Submitted to University of Westminster<br>Student Paper | <1% |
| 20 | Submitted to NorthTec<br>Student Paper | <1% |
| 21 | manualzz.com<br>Internet Source | <1% |

| | | |
|---|---|---|
| 22 | www.fecontabil.org.br<br>Internet Source | <1% |
| 23 | Submitted to Kingston University<br>Student Paper | <1% |
| 24 | Submitted to Kolej Poly-Tech MARA Kuala Lumpur<br>Student Paper | <1% |
| 25 | Submitted to University of Limerick<br>Student Paper | <1% |
| 26 | Submitted to University of Wales Institute, Cardiff<br>Student Paper | <1% |
| 27 | digitalcollection.utem.edu.my<br>Internet Source | <1% |
| 28 | repositorio.une.edu.pe<br>Internet Source | <1% |
| 29 | Submitted to University of Huddersfield<br>Student Paper | <1% |
| 30 | access.atis.org<br>Internet Source | <1% |
| 31 | www.dell.com<br>Internet Source | <1% |
| 32 | www.solitaireparadise.com<br>Internet Source | <1% |

| | | |
|---|---|---|
| 33 | Li Da Xu. "Frontiers in Enterprise Integration", CRC Press, 2020<br>Publication | <1% |
| 34 | www.intechopen.com<br>Internet Source | <1% |
| 35 | www.simonmetz.com<br>Internet Source | <1% |
| 36 | www.slideshare.net<br>Internet Source | <1% |

| | | | |
|---|---|---|---|
| Exclude quotes | Off | Exclude matches | Off |
| Exclude bibliography | Off | | |

# Accounts Clearance