

## ASSIGNMENT 1 FRONT SHEET

<b>Qualification</b>	<b>BTEC Level 5 HND Diploma in Business</b>		
<b>Unit number and title</b>	<b>Unit 30: Application Development</b>		
<b>Submission date</b>		<b>Date Received 1st submission</b>	
<b>Re-submission Date</b>		<b>Date Received 2nd submission</b>	
<b>Student Name</b>	Truong Tan Phuc	<b>Student ID</b>	GCD210070
<b>Class</b>	GCD1101	<b>Assessor name</b>	Pham Thanh Son
<b>Student declaration</b> <p>I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.</p>			
		<b>Student's signature</b>	

### Grading grid

P1	P2	P3	M1	M2	D1

<input type="checkbox"/> Summative Feedback:		<input type="checkbox"/> Resubmission Feedback:
Grade:	Assessor Signature:	Date:

**Internal Verifier's Comments:**

**Signature & Date:**

## Table of Contents

CHAPTER 1: SOFTWARE REQUIREMENTS SPECIFICATION AND SOFTWARE DESIGN (P1 – P2 – M1).....	8
1. Introduction (P1).....	9
1.1 Document Purpose .....	9
1.2 Product Scope .....	10
1.3 Intended Audience .....	11
1.4 Definitions, Acronyms and Abbreviations .....	11
2. Overall Description (P1).....	12
2.1 Product Overview .....	12
2.2 Product Functionality.....	14
3. Specific Requirements (P1 & M1).....	16
3.1 Functional Requirements (P1) .....	16
3.2 Use Case Model (P1) .....	17
3.3 Wireflow (P1) .....	18
4. Technical Design (M1) .....	21
4.1 Entity Relationship Diagram (ERD).....	21
4.2 Class Diagram.....	22
4.3 Activity Diagram .....	23
4.4 Gantt Chart.....	25
5. Risk Assessment (P2) .....	27
CHAPTER 2: TECHNOLOGIES LIST (P3) .....	30
1. Design Tools .....	30
1.1. Draw.io.....	30

2. Front End Technology Stack .....	32
2.1 HTML .....	32
2.2 CSS .....	34
2.3 Javascript.....	36
2.4 Bootstrap Framework.....	37
3. Back End Technology Stack.....	38
3.1 Programming Language.....	38
3.2 Operating System.....	39
3.3 Web Server.....	40
3.4 Database .....	41
3.5 Hosting.....	42
3.6 Frameworks.....	43
4. Tools For Source Control Management.....	45
4.1 Github .....	45
4.2 Visual Studio 2022.....	46
5. Software Development Models.....	46
5.1 Waterfall Model .....	46
5.2 V Model .....	47
5.3 Spiral Model.....	48
5.4 Select Model .....	49
CHAPTER 3: TECHNOLOGIES EVALUATION (M2).....	52
1. Design Tool – DrawIO.....	52
1.1 Compare.....	52

1.2 Preferred Selection and Methodology .....	53
2. Front End – HTML, CSS, JavaScript, Bootstrap .....	53
1.1 Compare .....	53
1.2 Preferred Selection and Methodology .....	54
3. Back End .....	54
3.1 Programming Language – C# .....	54
3.2 Operating System – Linux .....	55
3.3 Database – SQL Server .....	56
3.4 Frameworks - .NET .....	57
3.5 Web Server – Nginx .....	59
3.6 Hosting – Ubuntu .....	60
4. Tools For Source Control Management .....	61
1.1 Github .....	61
1.2 Visual Studio 2022 .....	62
CHAPTER 5: REFERENCES .....	70
Figure 1: MVC Flow .....	12
Figure 2: Use Case Model .....	18
Figure 3: Wireflow User Side .....	19
Figure 4: Wireflow Admin Side .....	20
Figure 5: ERD .....	21
Figure 6: Class Diagram .....	22
Figure 7: Activity Diagram User Side .....	23
Figure 8: Activity Diagram Admin Side .....	24
Figure 9: Gantt Chart .....	25

Figure 10: Gantt Chart .....	26
Figure 11: Risk Rating .....	29
Figure 12: DrawIO .....	30
Figure 13: CSS .....	34
Figure 14: Javascript .....	36
Figure 15: Bootstrap .....	37
Figure 16: C# .....	39
Figure 17: Linux .....	40
Figure 18: Nginx .....	41
Figure 19: Ubuntu .....	42
Figure 20: ASP .NET .....	44
Figure 21: Github .....	45
Figure 22: Visual Studio .....	46
Figure 23: Waterfall Model .....	47
Figure 24: V Model .....	48
Figure 25: Spiral Model .....	49

## CHAPTER 1: SOFTWARE REQUIREMENTS SPECIFICATION AND SOFTWARE DESIGN (P1 – P2 – M1)

---

# Software Requirements Specification

for

## ShoBee Project

Version <1.0>

Prepared by ShopeBee Group

Group Name: ShopBee Group

<Nguyen Quang Ngoc>  
<Truong Tan Phuc>  
<Duong Van Hieu>

<GCD210069>  
<GCD210070>  
<GCD210486>

<ngocnqgcd210069@fpt.edu.vn>  
<phucttgcd210070@fpt.edu.vn>  
<hieudvgcd210486@fpt.edu.vn>



**Instructor:** Pham Thanh Son

**Course:** 1670 Application Development

**Date:** 01/12/2023

## 1. Introduction (P1)

This Software Requirements Specification (SRS) outlines the development of a C# .NET Core MVC Website Application dedicated to an online book-selling platform. The application is structured into three primary segments: Customer, Store, and Admin.

The primary objective of this document is to articulate the functional and non-functional requirements essential for the successful implementation and operation of this web-based platform. Each segment (Customer, Store, and Admin) will require user authentication before access is granted. Upon login, the system will verify the user's role and direct them to the relevant section accordingly.

The subsequent sections of this SRS will delve into specific requirements, functionalities, and constraints for each segment, elucidating the comprehensive scope of the project.

### 1.1 Document Purpose

This Software Requirements Specification (SRS) pertains to a C# .NET Core MVC Website Application ShopBee designed for an online book-selling platform. The project encompasses the development of a comprehensive system featuring three distinct segments: Customer, Store, and Admin. The Store and Customer roles are essentially a common Route. The Store role will be allocated when the Customer wants to increase their authority. Each segment holds specific functionalities catering to different user roles and responsibilities within the platform.

The scope of this SRS encapsulates the delineation of functional and non-functional requirements essential for the successful construction, implementation, and operation of the aforementioned web-based application. The document specifically details the requirements for user authentication across all segments, ensuring that access to the system is permitted only after successful authentication and role verification. This SRS encompasses the initial phase of the development process, focusing primarily on outlining the requirements for the segmented user areas while providing a foundation for subsequent development phases.

This description aims to define the project, its components, and the scope covered by the Software Requirements Specification (SRS). It highlights the primary focus on user authentication and delineates the specific areas within the system that will be detailed in this document.

## 1.2 Product Scope

The platform aims to streamline the online book purchasing process, offering benefits such as a convenient and intuitive user interface for customers, efficient inventory management for store administrators as well as Comprehensive control and monitoring for system administrators. The goals of the software include enhancing user experience, ensuring secure transactions, efficient book inventory management, and providing a seamless administration experience for all stakeholders. The main goal is to create a powerful, scalable, and reliable platform that meets the needs of book enthusiasts, store managers, and system administrators alike.

### **Benefits for Customers:**

Customers will easily come into contact with a convenient and intuitive interface. They can choose the types of books they want on the website by searching by title, author, and type of book they want. They can also search by the amount they can afford.

### **Benefits for Store Owners:**

Store Owners can manage the number of books they can sell, customize order information easily to notify buyers. In addition, the website will integrate a revenue statistics function to help Store Owners have no difficulty calculating the total amount they will receive.

### **Benefits for Admin:**

Admins can easily manage information on the database without having to access the database system. With an interface that makes it easy to manage the system through

In essence, the software aims to enhance the user experience for customers, facilitate efficient inventory management for store owners, and provide a simplified administrative experience for system administrators. The platform's core strengths lie in its user-centric design, robust inventory controls, and seamless administrative capabilities.

This summary highlights the key objectives, benefits, and user-focused functionalities of the online book-selling platform for customers, store owners, and system administrators.

### 1.3 Intended Audience

The Software Requirements Specification (SRS) is tailored for various stakeholders involved in the development and implementation of the online book-selling platform. The intended audience includes Instructor (Teacher), developers, project managers, system administrators, and the client. Developers will use this document as a guide for technical implementation, while project managers will rely on it for project planning and oversight. System administrators will find information relevant to system management, and the client will gain insights into the project's scope and functionalities.

The SRS is organized into several sections, each serving a distinct purpose in delineating the project's requirements and specifications. The document begins with an introduction encompassing the purpose, product scope, and intended audience overview. Following this, the 'Overall Description' section provides a comprehensive overview of the product, including its functionality and high-level requirements. Subsequently, the 'Specific Requirements' section outlines detailed functional requirements, use case models, and wireflow diagrams essential for developers and testers.

Moving forward, the '**Technical Design**' section delves into the technical aspects, presenting entity relationship diagrams (ERD), class diagrams, activity diagrams, and a Gantt chart crucial for developers and technical staff. The document concludes with a 'Risk Assessment' section outlining potential risks associated with the project.

To optimize understanding, the recommended sequence for reading the document is as follows:

**Introduction:** Provides an overview of the document's purpose, product scope, intended audience, and definitions.

**Overall Description:** Offers a high-level understanding of the product's overview and functionality.

**Specific Requirements:** Delineates detailed functional requirements, use case models, and wireflow diagrams.

**Technical Design:** Focuses on technical aspects such as ERD, class diagrams, activity diagrams, and the Gantt chart.

**Risk Assessment:** Outlines potential risks associated with the project, providing an understanding of potential challenges.

This description outlines the intended readership and offers a recommended reading sequence for different sections within the SRS document based on their relevance to specific stakeholders.

### 1.4 Definitions, Acronyms and Abbreviations

ID	Acronyms and Abbreviations	Explanation
----	----------------------------	-------------

1	SRS	Software Requirement Specifications
2	HR	Human Resource
3	ERD	Entity-Relationship Diagram
4	MVC	Model-View-Controller
5	UC	Use Case

## 2. Overall Description (P1)

### 2.1 Product Overview

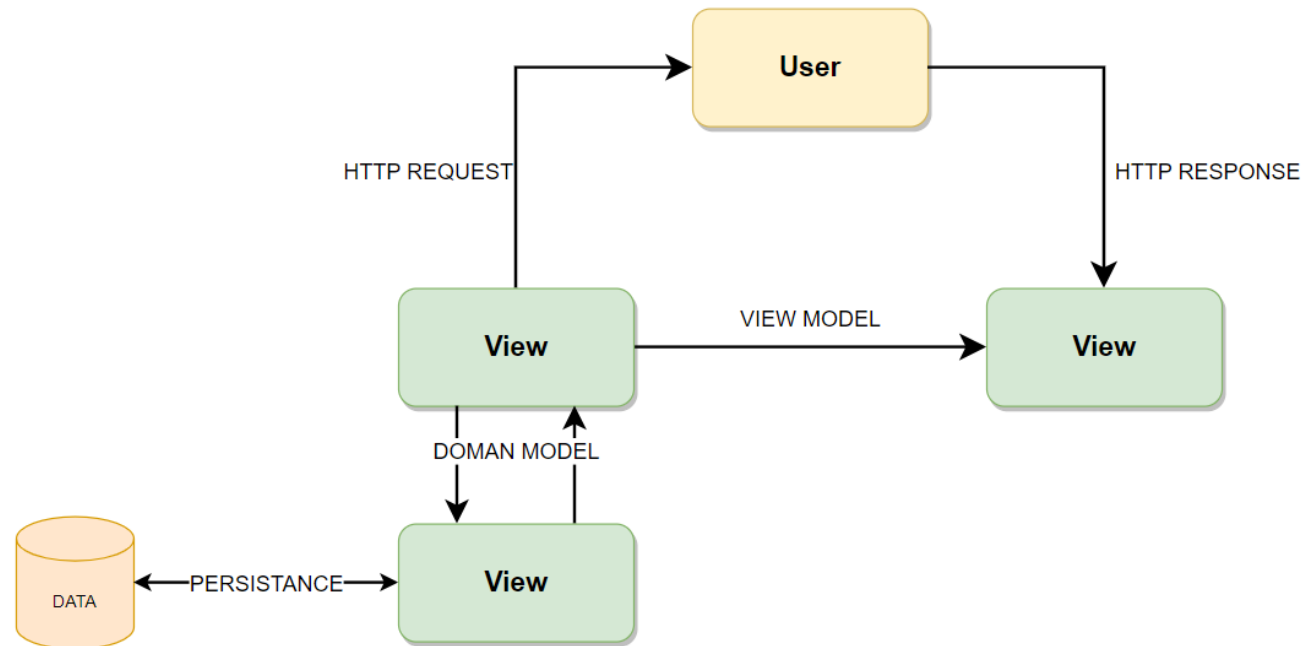


Figure 1: MVC Flow

**From a Product Perspective:**

The product being considered is a cutting-edge online bookstore system that makes use of the Model-View-Controller (MVC) architecture to provide book lovers a smooth and intuitive experience. The idea behind this system is to create a brand-new, standalone product that will update and modernize the book buying process by bringing it online and making it more interactive. Instead than just being a little addition to an already-existing product line, it is a ground-breaking solution that completely changes the way readers interact with the literary canon.

### **Overall Diagram:**

Using the MVC design, we show the key elements of the Online Bookstore System in the diagram. The book database and any other data feeds are represented as the data source by the Model component. Purchases, book choices, and other user inputs are managed by the Controller component. The shopping cart, user profiles, book listings, and user interface are all shown via the View component.

The data and control flow between the Model, View, and Controller are shown by the subsystem linkages. When a user interacts with the system, input is processed by the Controller, which then changes the Model (e.g., modifies the shopping cart) and tells the View to update the presentation. Payment gateways and user authentication services are examples of external interfaces that are portrayed as essential components of the system as a whole.

An overview of how the Online Bookstore System operates in its surroundings is given by this diagram. While the MVC design makes sure that the underlying software is well-structured and easily maintained, users interact with the system via the user interface, search for books, and make purchases. With the help of this creative product viewpoint, book purchases should be more easy, pleasurable, and smoothly incorporated into the digital age.

This software marks our inaugural endeavor in the domain of online book-selling platforms. It stands as a new, self-contained product, not associated with any existing systems or part of a product family. Our platform is conceived as an independent and innovative solution aimed at transforming the traditional book purchasing experience into an intuitive and streamlined online process. With no predecessors or replacements, this product is designed to serve as a standalone entity, revolutionizing how customers interact with online bookstores, how store administrators manage their inventory, and how system administrators oversee the platform's operations.

### **Major Components:**

**Guest User:** Represents users who haven't logged in. They can perform actions like registering an account, browsing the book list, and searching for books based on specific characteristics or criteria.

**Customer:** Logged-in users with an account. They have access to functionalities such as logging in/out, managing their account, viewing order history, making orders, and even transitioning to become a Store Owner.

**Store Owner:** Users who have transitioned from being customers and now have the additional capabilities of managing their own bookstore, selling books, and requesting the Admin to add new book categories to the system.

**Admin:** Represents the system administrator. This role has access to the Admin Panel and is responsible for managing stores, categories, customers, books, and orders within the system.

This description conveys that the software is a new and self-contained product, not connected to any existing systems or part of a product lineage. Additionally, it highlights the innovative approach aimed at enhancing the book purchasing experience. The accompanying general diagram aims to provide a visual representation, depicting major components and system interactions at a high level.

## 2.2 Product Functionality

### a, Functional

#### **Registration and authentication:**

Allows users to register new accounts and authenticate themselves to access the system.

#### **Browse and search:**

Allows guest users and registered customers to browse the list of available books.

Allows users to search for books based on various characteristics such as title, author or genre.

#### **Account management:**

Provides functions for users to log in, log out, and manage their account details.

Allows customers to view their order history and manage their account settings.

**Order management:**

Allow customers to order books and manage their shopping carts.

Allows store owners to monitor and manage orders placed in their respective stores.

**Manage user roles:**

Allows customers to convert to become store owners, granting them additional functionalities to manage their bookstore.

**Store manager:**

Allows store owners to manage their bookstore, including book inventory, sales, and new catalog requests.

**Manage admin panel:**

Allows administrators to access the admin panel to manage stores, categories, customers, books, and orders in the system.

These key functions encapsulate the core capabilities of an online bookselling platform, from user authentication to account management, order processing, and various levels of user role management. Detailed specifications for each function will be provided in Part 3 of the SRS.

**b, Non-Functional**

**Performance:**

Response time: 2 seconds.

Concurrent users: 100 without performance degradation.

**Scalability:**

Handle 20% growth in users and transactions over the next year.

Database scalability for books, users, and orders.

**Reliability:**

System uptime: 99.9%.

Recovery time: < 30 minutes after a failure.

### Security:

Strong authentication and authorization.

Encryption: HTTPS for all communication.

Protection against common web application security threats.

## 3. Specific Requirements (P1 & M1)

### 3.1 Functional Requirements (P1)

ID	AS A < TYPE OF USER >	I WANT TO < PERFORM SOME TASK >	SO THAT I CAN < ACHIEVE SOME GOAL >
1	GUEST USER	Able to sign up, browse for goods, see and add items to their basket, and log in.	<ul style="list-style-type: none"> <li>- Makes it possible for consumers to compile a list of future purchases, improving the whole purchasing experience.</li> <li>- Urge visitors to sign up so they may customize their interactions going forward.</li> </ul>
2	REGISTERED USER	Search for books, View the list of books, Log in, Add books to the cart, Edit profile.	<ul style="list-style-type: none"> <li>- Savor a customized book-searching journey.</li> <li>- Control and monitor a few chosen books in the cart.</li> <li>- Ensure that your profile is up to date to get customized services.</li> </ul>
3	STORE OWNER	CRUD (Create, Read, Update, Delete) books.  View statistics.  Request a new category for books.	Handle the book inventory of the business with efficiency.  Utilize statistics to get insights.  Increase the number of book categories that are offered.
4	ADMIN	Manage stores.	Assure the system's correct operation and arrangement.



		Manage categories.	Monitor and regulate user rights and access.
		Manage customers.	Encourage effective administration of the whole functioning of the bookshop.
		Manage managers.	
		Manage books.	
		Manage orders.	

### Similarities Between User Roles:

**Efficiency:** Whether browsing books, running the shop, or monitoring the system, all user roles strive to be as efficient as possible in their particular jobs.

**Personalization:** Tasks like modifying profiles and setting preferences demonstrate how the system provides users with a tailored experience.

**Management:** Store owners and administrators both concentrate on overseeing various facets of the system, stressing the need of efficient and well-structured operations.

To put it briefly, the Online Bookstore System's defined user roles and duties are made to accommodate a variety of user requirements, from effective administration and supervision to exploration and customization. The system's goal is to provide different parties engaged in the book-buying process a flexible and user-friendly platform.

### 3.2 Use Case Model (P1)

**Guest User:** Register, Book List

**Customer:** Login, Logout, Edit Profile, Search Book, Add Book to Cart, Checkout

**Store Owner:** CRUD Books, Request Category, Statistics

**Admin:** Login, Manager Stores, Manager Categories, Manager Customers, Manager Books, Manager Orders

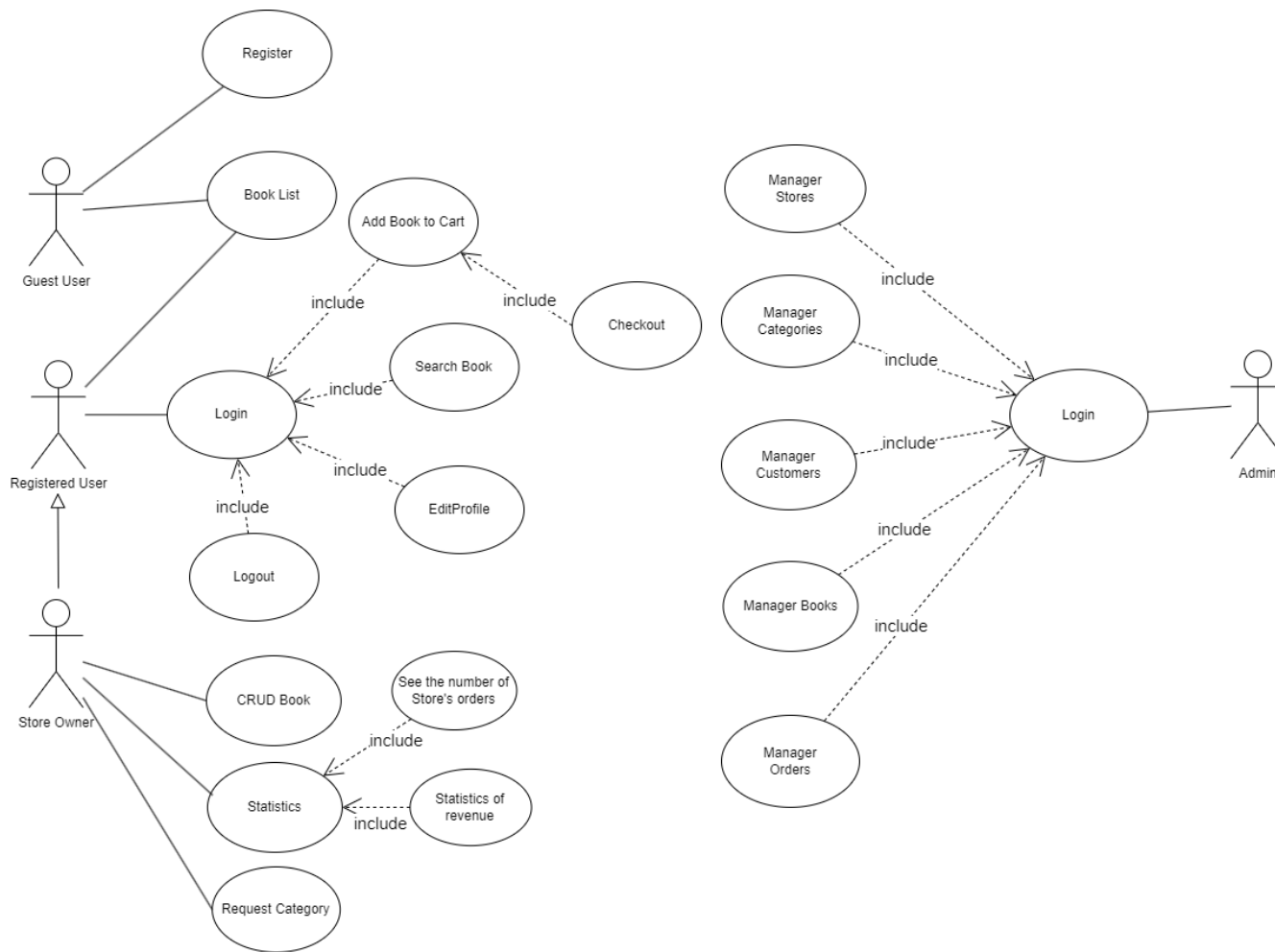


Figure 2: Use Case Model

### 3.3 Wireflow (P1)

[Source](#)

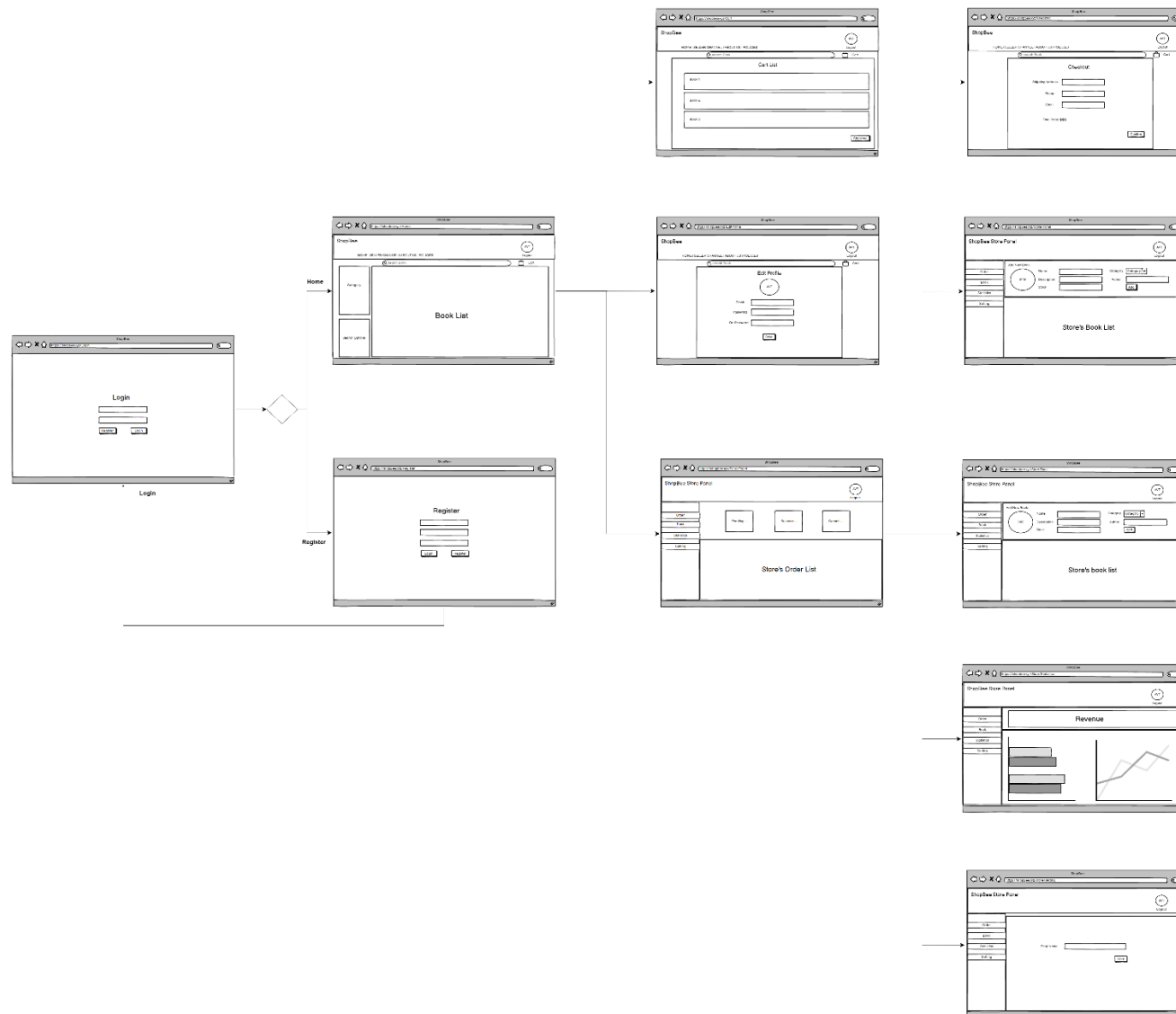


Figure 3: Wireflow User Side

Source

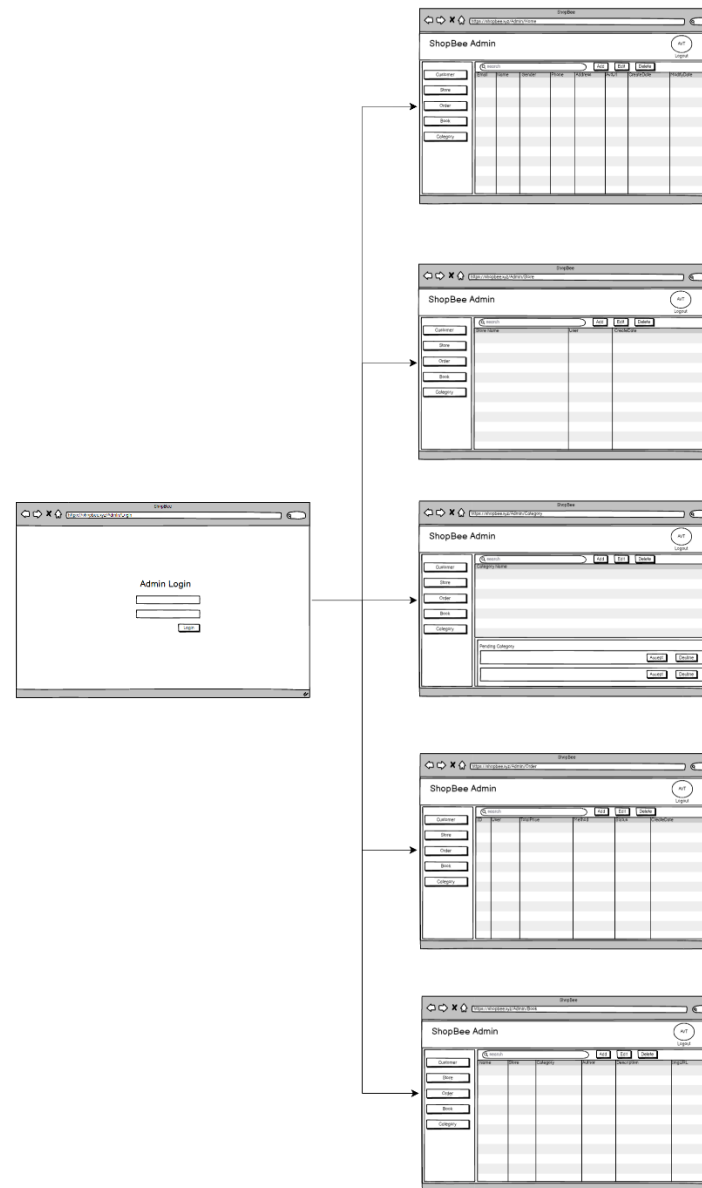


Figure 4: Wireflow Admin Side

## 4. Technical Design (M1)

### 4.1 Entity Relationship Diagram (ERD)

[Source](#)

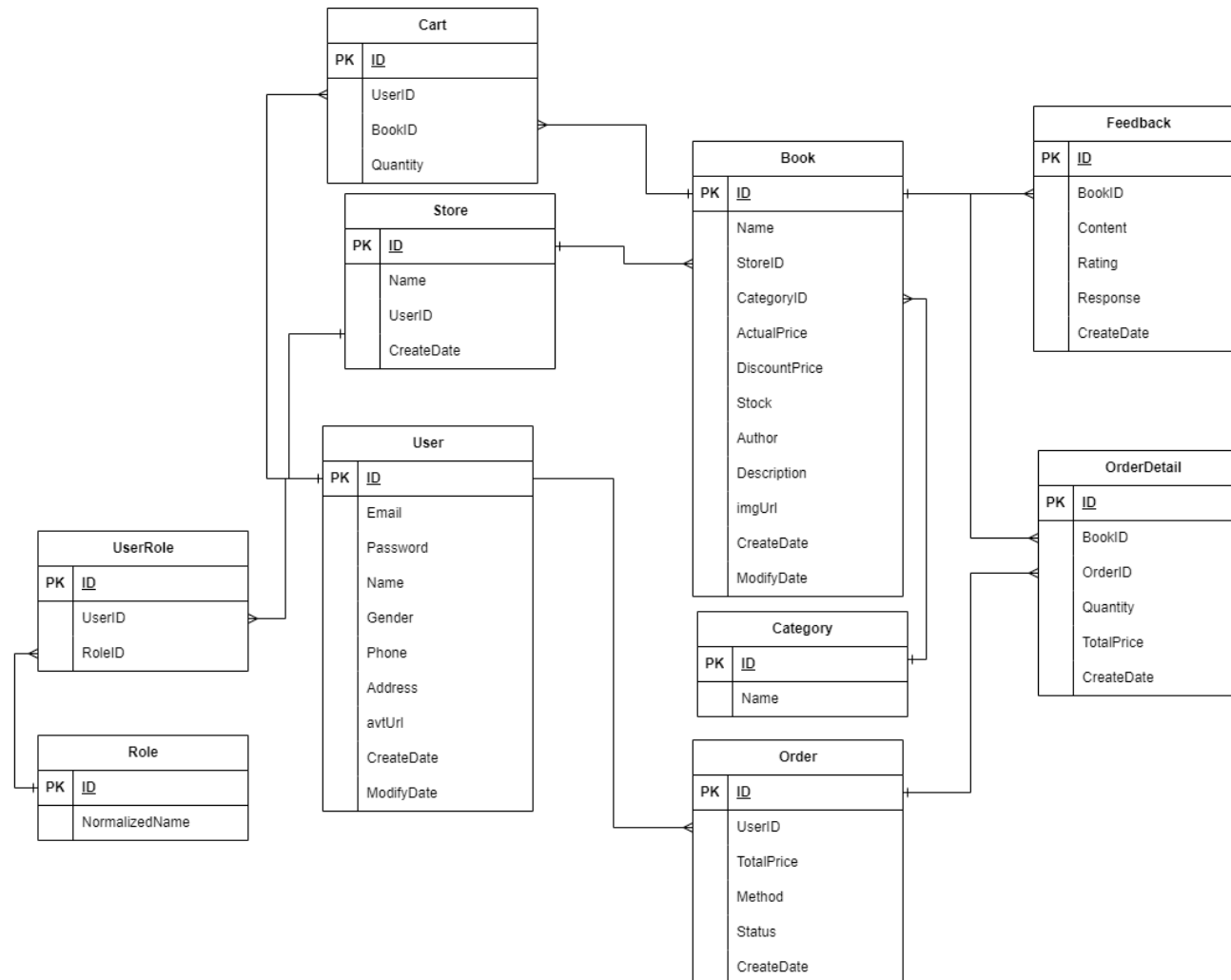


Figure 5: ERD

## 4.2 Class Diagram

### Source

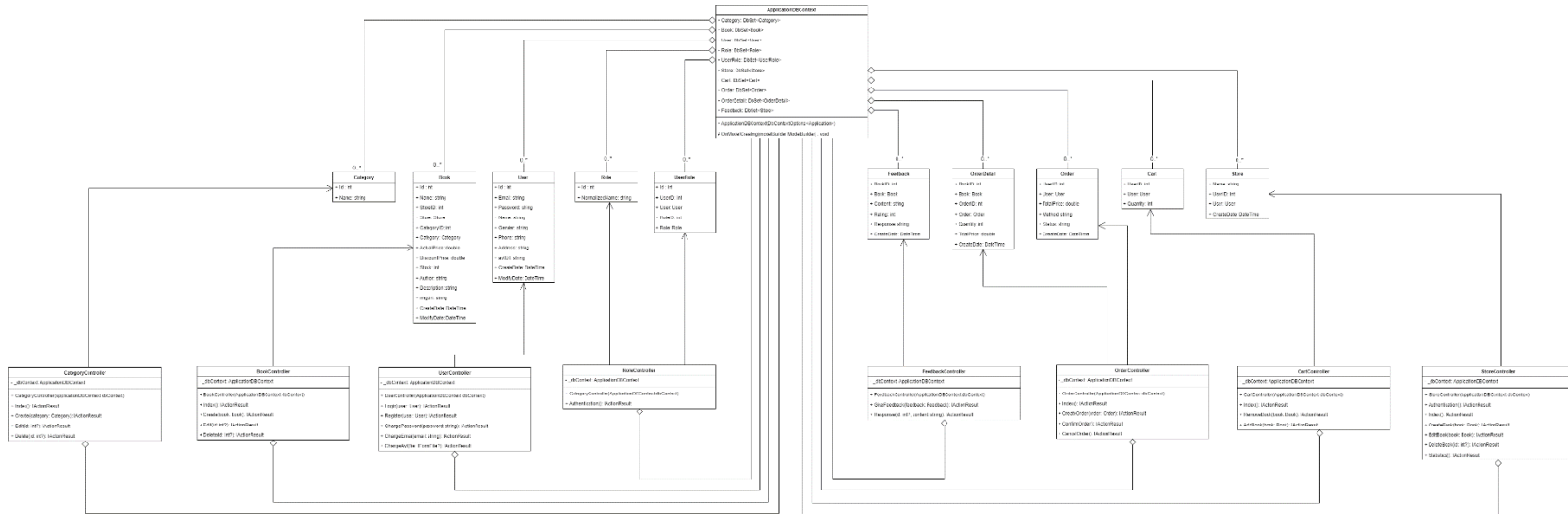


Figure 6: Class Diagram

### 4.3 Activity Diagram

[Source](#)

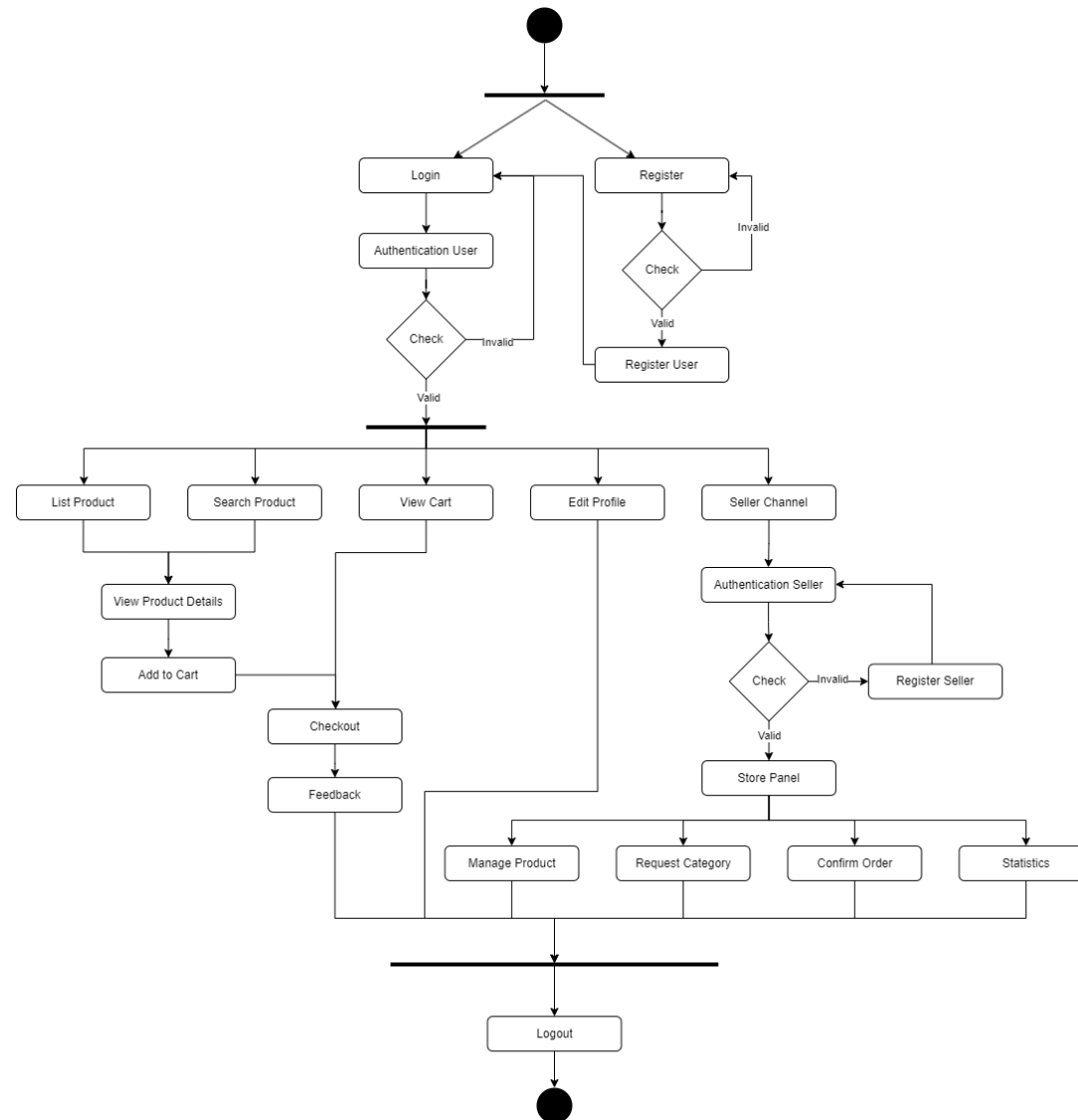


Figure 7: Activity Diagram User Side

[Source](#)

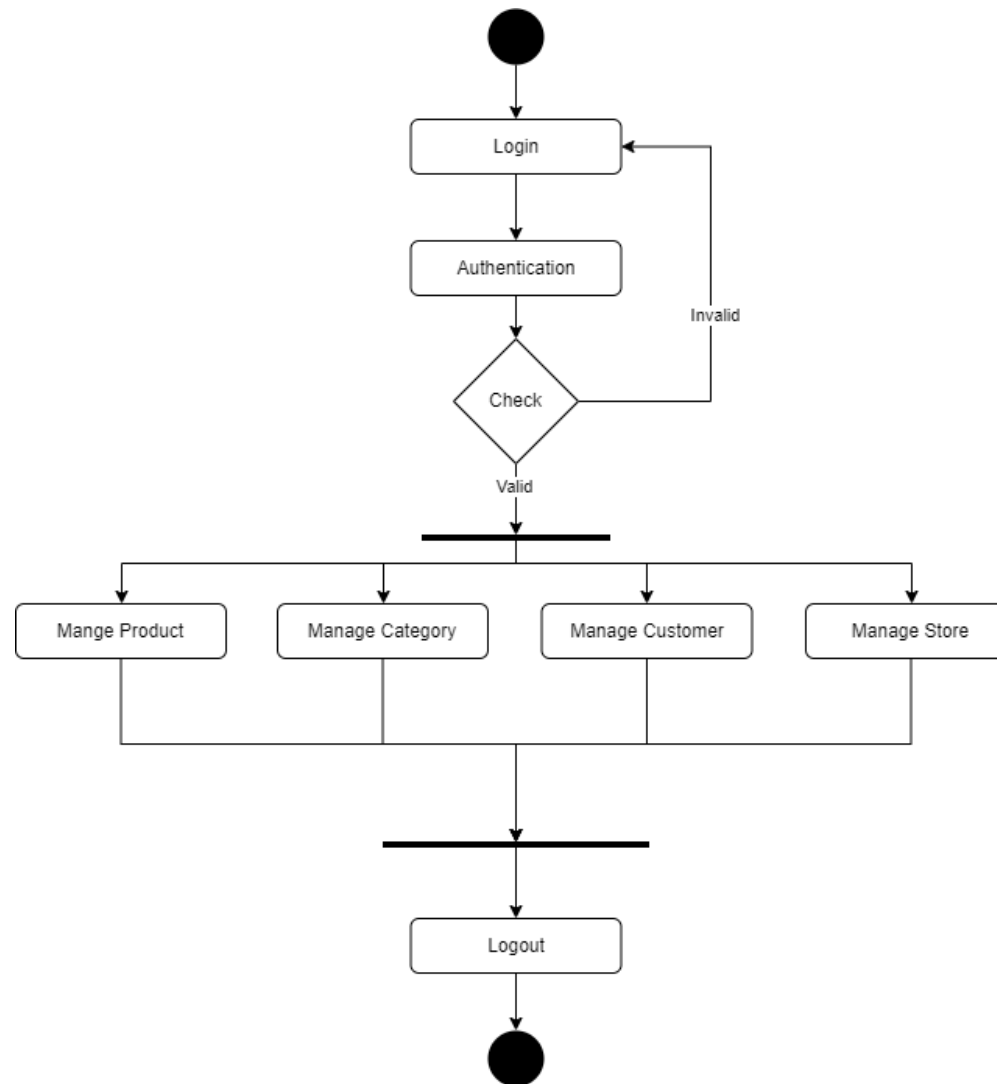


Figure 8: Activity Diagram Admin Side



## 4.4 Gantt Chart

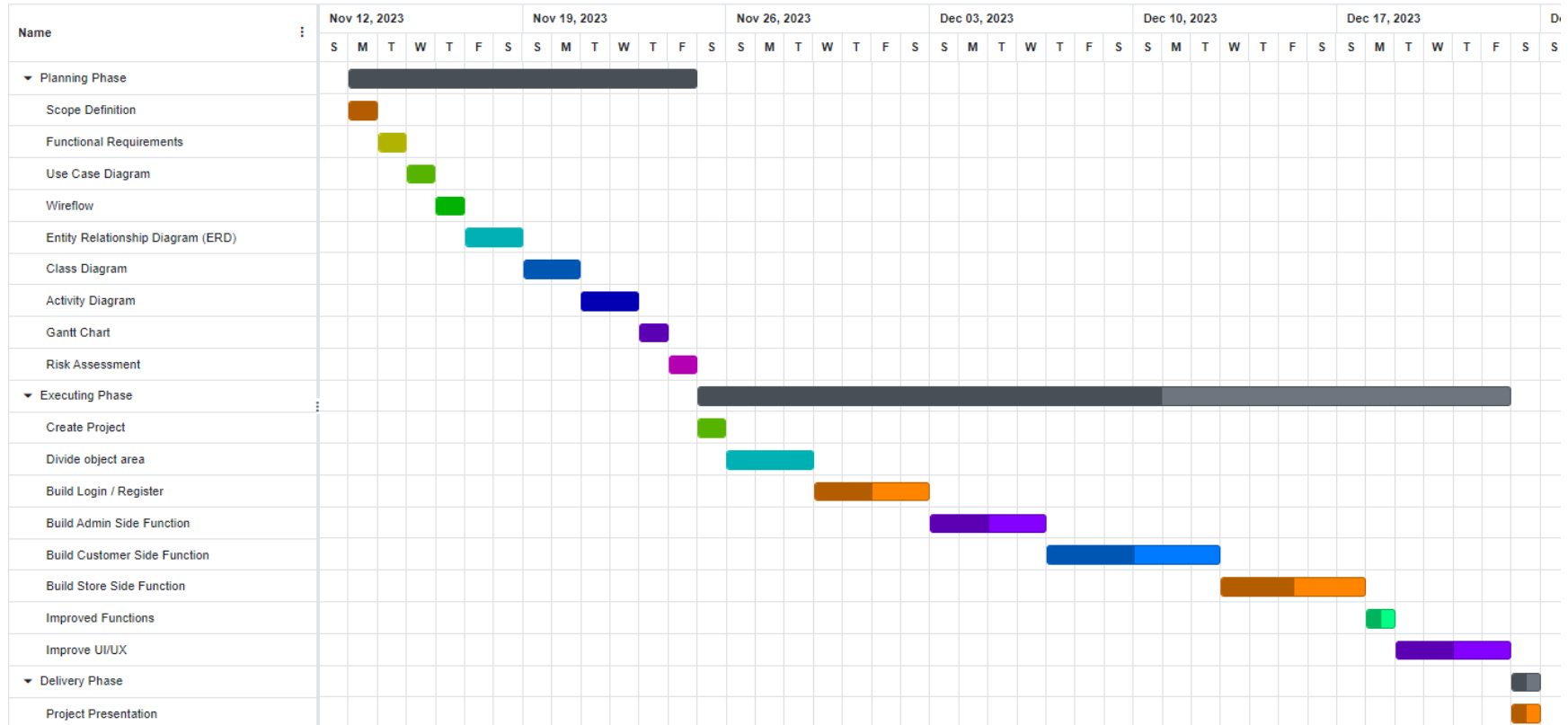


Figure 9: Gantt Chart

Name	Start Date	End Date	Duration	Progress %
▼ Planning Phase	Nov 13, 2023	Nov 24, 2023	12 days	100
Scope Definition	Nov 13, 2023	Nov 13, 2023	1 day	100
Functional Requirements	Nov 14, 2023	Nov 14, 2023	1 day	100
Use Case Diagram	Nov 15, 2023	Nov 15, 2023	1 day	100
Wireflow	Nov 16, 2023	Nov 16, 2023	1 day	100
Entity Relationship Diagram (ERD)	Nov 17, 2023	Nov 18, 2023	2 days	100
Class Diagram	Nov 19, 2023	Nov 20, 2023	2 days	100
Activity Diagram	Nov 21, 2023	Nov 22, 2023	2 days	100
Gantt Chart	Nov 23, 2023	Nov 23, 2023	1 day	100
Risk Assessment	Nov 24, 2023	Nov 24, 2023	1 day	100
▼ Executing Phase	Nov 25, 2023	Dec 22, 2023	28 days	57
Create Project	Nov 25, 2023	Nov 25, 2023	1 day	100
Divide object area	Nov 26, 2023	Nov 28, 2023	3 days	100
Build Login / Register	Nov 29, 2023	Dec 02, 2023	4 days	50
Build Admin Side Function	Dec 03, 2023	Dec 06, 2023	4 days	50
Build Customer Side Function	Dec 07, 2023	Dec 12, 2023	6 days	50
Build Store Side Function	Dec 13, 2023	Dec 17, 2023	5 days	50
Improved Functions	Dec 18, 2023	Dec 18, 2023	1 day	50
Improve UI/UX	Dec 19, 2023	Dec 22, 2023	4 days	50
▼ Delivery Phase	Dec 23, 2023	Dec 23, 2023	1 day	50
Project Presentation	Dec 23, 2023	Dec 23, 2023	1 day	50

Figure 10: Gantt Chart

## 5. Risk Assessment (P2)

RISKS	LIKEDHOOD	IMPACT	RISK RATING	ACTION
<b>Security Risks:</b> Unauthorized access to customer data or payment information. Data breaches leading to the exposure of sensitive information. Malicious attacks such as hacking or phishing.	3	5	15	Implement strong encryption for customer data and payment information. Regularly conduct security audits and vulnerability assessments. Employ firewalls, intrusion detection systems, and other security measures.
<b>Data Integrity Risks:</b> Errors in data entry leading to incorrect order processing. Loss or corruption of customer and inventory data.	4	4	16	Implement data validation checks to ensure accurate information. Regularly backup customer and inventory data to prevent loss. Implement version control to track changes to critical data.
<b>Third-Party Integration Risks:</b> Issues with external services or APIs affecting order fulfillment. Dependency on third-party services that may be prone to failures.	3	5	12	Choose reliable and reputable third-party services and APIs. Have contingency plans for service disruptions and communicate them to users.

				Regularly assess the security practices of third-party providers.
<b>System Performance</b> <b>Risks:</b> Downtime or slow performance during peak usage periods. System crashes leading to the loss of transaction data.	2	4	10	Use load balancing and scalable infrastructure to handle peak loads. Implement monitoring tools to identify and address performance issues proactively. Have a disaster recovery plan in place to minimize downtime.
<b>Payment Processing</b> <b>Risks:</b> Errors in payment processing leading to financial discrepancies. Fraudulent transactions or chargebacks.	4	3	12	Comply with PCI DSS standards for secure payment processing. Implement two-factor authentication for sensitive transactions. Regularly review and update payment processing procedures.

After having the score of LIKELIHOOD and IMPACT of each risk. To find the RISK RATING of each risk we calculate by LIKELIHOOD \* IMPACT. Below is a table showing the level of risk. [[Reference Source](#)]

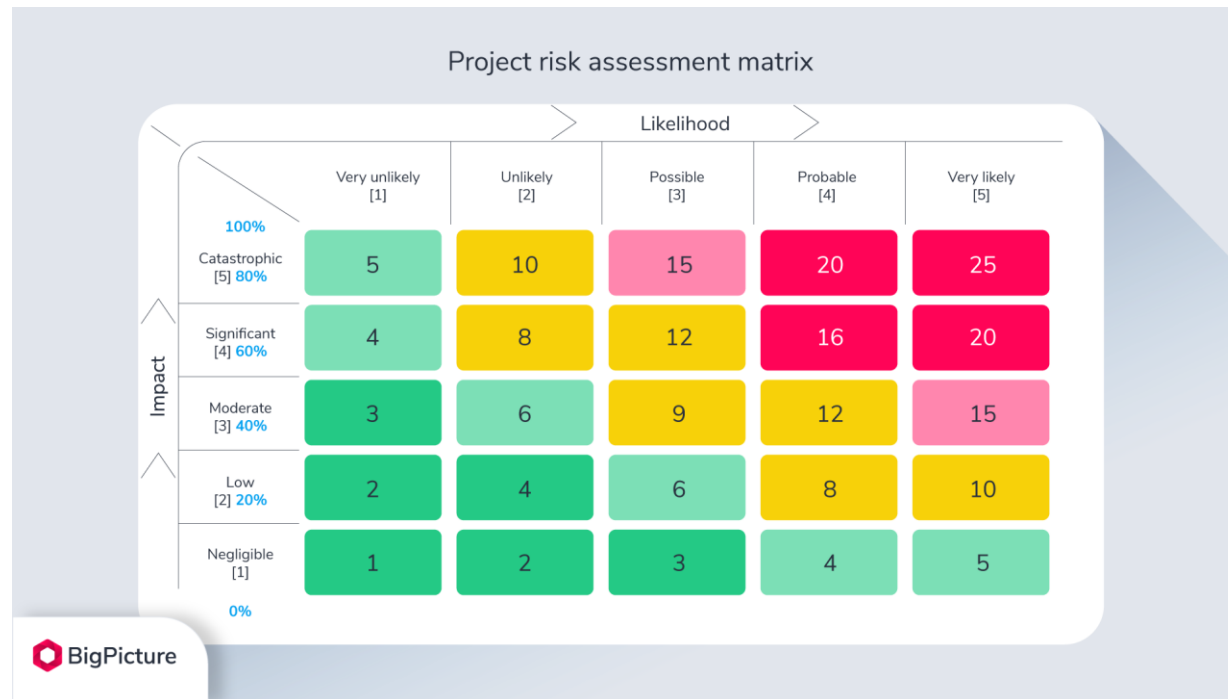


Figure 11: Risk Rating

## CHAPTER 2: TECHNOLOGIES LIST (P3)

### 1. Design Tools

#### 1.1. Draw.io

Seibert Media developed Draw.io, a proprietary tool for making diagrams and charts. You may utilize the program's automatic layout tool or create a custom layout.

To assist you in creating a special diagram or chart, they provide a large range of shapes and many graphic components. The drag-and-drop feature makes it simple to construct an attractive diagram or chart. Draw.io gives you the choice to save saved charts on network storage at a data center, on a server, or in the cloud, according on your needs.



*Figure 12: DrawIO*

#### **Advantages of draw.io:**

**User-Friendly Interface:** Users with different degrees of design expertise may use Draw.io due to its user-friendly interface, which is simple to use and intuitive.

**Online communication:** Draw.io is a web-based platform that facilitates real-time team communication, hence streamlining distant work.

**Vast Selection of Templates:** The application saves time and effort throughout the design process by offering a wide variety of templates for UI mockups, flowcharts, diagrams, and more.

**Cost-effective:** Draw.io is a reasonable choice for small teams and projects because of its extensive capabilities and free edition.

**Versatility:** Draw.io is a great tool for UI design, but it can also be used to create system architecture diagrams, database models, and other things.

**Options for Exporting:** Draw.io allows for the export of designs in a variety of file formats, making it simple to share and showcase design ideas.

**Frequent Updates and Support:** Draw.io is a well-liked tool that is updated and stabilized on a regular basis to provide users with a modern and reliable design experience.

**Integration with Cloud Services:** Draw.io makes file management and sharing easier by integrating with cloud storage services like Dropbox, OneDrive, and Google Drive.

### **Disadvantages of draw.io:**

**Restricted Offline view:** Because Draw.io is largely a web-based application, users may have issues when trying to view their creations offline.

**Not the Best for Advanced capabilities:** Draw.io is a flexible tool, but it may not provide as many sophisticated capabilities as more specialist design tools like Adobe Creative Suite.

**Reduced Support for Print Design:** Draw.io may not be the greatest option for print design projects, while being great for digital design and prototyping.

**Dependency on Internet access:** Because Draw.io is a web-based application, it needs internet connectivity to function. Any interruption to internet connectivity might make design work more difficult.

**Restricted Customization choices:** When compared to more sophisticated design tools, some users may find Draw.io's customization choices to be more limited.

**Learning Curve for sophisticated Designs:** In order to properly use all of the features and functions of the program, novices may need to go through a learning curve for elaborate and sophisticated designs.

Security Issues with Cloud Services: Although cloud integration facilitates collaboration, some users may be wary about putting confidential design data on cloud servers.

### **The reason why I choose Draw.io in this project:**

A single platform for UI and UML design is offered by Draw.io. My team can use Draw.io for all visual design requirements, streamlining their workflow instead of utilizing several tools for different design jobs.

Because of its user-friendly UI, team members with different degrees of design experience may use Draw.io. The familiar drag-and-drop feature makes it easier to create UI mockups and UML diagrams.

A vast collection of templates tailored to UML and UI design is available from Draw.io. By using these templates, my team can save time and make sure that best practices and industry standards are followed.

Working together and providing feedback: My team members can now collaborate in real time on UML diagrams and UI mockups thanks to Draw.io's support for it. This functionality comes in very handy when many stakeholders need to comment on designs and make iterations.

I can export your UI mockups and UML diagrams using Draw.io in PNG, JPEG, PDF, and XML formats. It is simple to share designs with stakeholders or other team members and include them into documentation or presentations thanks to this versatility.

## **2. Front End Technology Stack**

### **2.1 HTML**

Hypertext markup language (HTML) is the standard markup language used to generate web pages that are meant to be seen and navigated online. Being a markup language, it features several tags. Tags may be used to display text, tables, sorted and unordered lists, and more. An HTML page consists of two main sections: the head section and the body portion. While all of the tags needed to express the visual content of the web page are included in the body section, the head section holds the information or data that defines the page. HTML is a platform-independent language, meaning it may be used on any kind of operating system, such as Linux, Windows, etc.

#### **Advantages of HTML:**

- HTML helps to build structure of a website and is a widely used Markup language



- It is easy to learn.
- Every browser supports HTML Language.
- HTML is light weighted and fast to load.
- Storage of big files are allowed because of the application cache feature.
- Do not get to purchase any extra software because it's by default in every window.
- Loose syntax (although, being too flexible won't suit standards).
- HTML is simple to edit as being a plain text.
- It integrates easily with other languages such as JavaScript, CSS etc.

### **Disadvantages of HTML:**

- It cannot produce dynamic output alone, since it is a static language.
- Making the structure of HTML documents becomes tough to understand.
- Errors can be costly.
- It is the time consuming as the time it consume to maintain on the color scheme of a page and to make lists, tables and forms.
- We need to write a lot of code for just creating a simple webpage.
- We have to check up the deprecated tags and confirm not to use them to appear because another language that works with HTML has replaced the first work of the tag, and hence the opposite language needs to be understood and learned.

### **The reason why I choose HTML:**

- The Universal Language of the online: The foundational language used to generate online pages is HTML (Hyper Text Markup Language). All contemporary web browsers support it, therefore a wide audience using a variety of platforms and devices may access your online application.
- Basic and Easy to Learn: HTML is very simple to learn and comprehend, allowing developers of all skill levels to work with it. It is perfect for creating the structure and content of web pages since it employs an easy-to-understand syntax based on tags.
- Integration with Other Technologies: JavaScript provides interaction, while CSS (Cascading Style Sheets) handles style. HTML does this with ease. The basis of contemporary web development is this trio (JavaScript, CSS, and HTML), which enables solid and dynamic user experiences.

- **Rich Structure and Semantic components:** The structure and meaning of the material are described by a variety of semantic components that HTML offers. In addition to aiding in SEO (Search Engine Optimization), using these components correctly increases accessibility for those with impairments.
- **Responsive Web Design:** Creating web pages that adjust to various screen sizes and devices requires the use of HTML. You can design an HTML flexible layout online application that looks fantastic on tablets, smartphones, and PCs.

## 2.2 CSS

With the use of CSS, a technique sheet language, web designers may modify the way an HTML page is presented and behaves when viewed in a web browser. CSS, sometimes referred to as cascading sheets, is a text-based coding language that specifies how webpages are designed and work with web browsers. Web designers may manage a great deal of stylistic features and functionality using this language, including layout, color, and fonts, which in turn affects how HTML information is presented and formatted.

Differentiating between a document's presentation—which include design elements like color, style, and typefaces—and its information was the main goal. CSS is responsible for an internet page's appearance and design. Among other things, you may use CSS to change the text's color, font style, paragraph spacing, column size, and layout.



*Figure 13: CSS*

### **Advantages of CSS:**

- CSS plays an important role, by using CSS you simply got to specify a repeated style for element once & use it multiple times as because CSS will automatically apply the required styles.
- The main advantage of CSS is that style is applied consistently across variety of sites. One instruction can control several areas which is advantageous.
- Web designers needs to use few lines of programming for every page improving site speed.
- Cascading sheet not only simplifies website development, but also simplifies the maintenance as a change of one line of code affects the whole web site and maintenance time.
- It is less complex therefore the effort are significantly reduced.
- It helps to form spontaneous and consistent changes.

### **Disadvantages of CSS:**

- With CSS, what works with one browser might not always work with another. The web developers need to test for compatibility, running the program across multiple browsers.
- There exists a scarcity of security.
- After making the changes we need to confirm the compatibility if they appear. The similar change affects on all the browsers.
- The programming language world is complicated for non-developers and beginners. Different levels of CSS i.e. CSS, CSS 2, CSS 3 are often quite confusing.

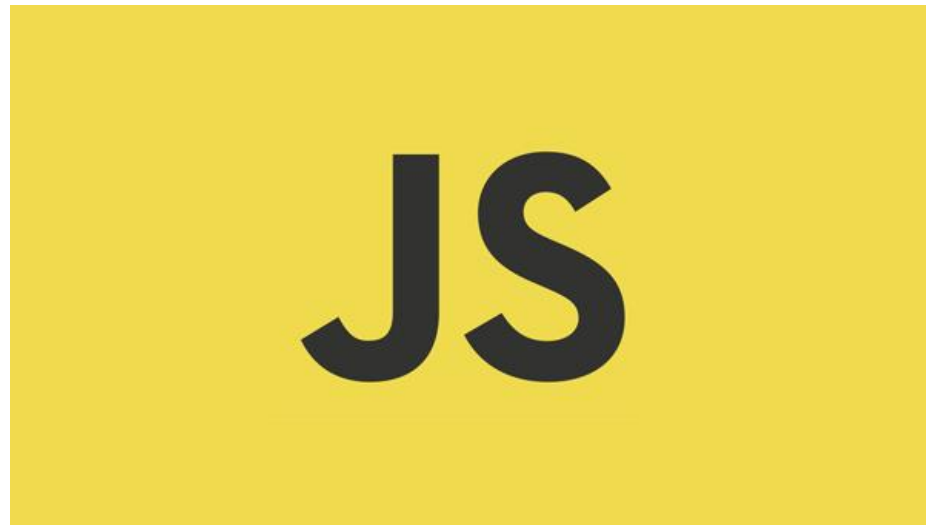
### **The reasons why I choose CSS:**

- Separation of Concerns: CSS allows me to separate the presentation layer from the content layer (HTML). This separation promotes clean and organized code, making it easier to maintain and update the styling without affecting the underlying structure.
- Consistent Design and Branding: CSS enables you to define global styles and apply them consistently throughout the entire web app. This consistency ensures a cohesive design and reinforces the branding of the book store, creating a professional and polished appearance.

- **Responsive Web Design:** CSS plays a vital role in creating responsive web pages that adapt to different screen sizes and devices. By using media queries and responsive design techniques, I can ensure that the web app looks and functions optimally on desktops, tablets, and smartphones.
- **Styling and Theming:** With CSS, you can apply different styles and themes to my web app easily. This versatility allows you to switch between different themes, change color schemes, and adjust the visual aspects of the application to suit different purposes or events.

## 2.3 Javascript

High-level, interpreted programming languages like JavaScript are typically utilized for web development. It is one of the fundamental technologies used to provide dynamic and interactive online content. JavaScript gives programmers the ability to enhance and captivate users by adding features, managing activities, and modifying web page content in real-time.



*Figure 14: Javascript*

### **The reason why I choose Javascript:**

- **Client-Side interaction:** To include interaction into your online application, JavaScript is required. With it, you can build dynamic user interfaces, manage user events (including clicks and form submissions), and react to user activity instantly without having to refresh the page.

- **Wide Browser Support:** JavaScript is a dependable option for guaranteeing your online application functions uniformly across all platforms and devices since it is supported by all current web browsers.
- **Rich Ecosystem of Libraries and Frameworks:** Front-end libraries and frameworks, like React, Angular, and Vue.js, are widely available and well-established in the JavaScript ecosystem. These frameworks may greatly expedite development, make difficult jobs easier, and give your software a unified structure.
- **Form Validation and Error Handling:** By using JavaScript to check user input on forms, you may lower the likelihood of mistakes and ensure that data supplied to the server is correct.
- **Local Storage and Caching:** By caching frequently requested data, JavaScript allows you to store data locally on the user's browser, which may enhance speed and lower server load.

## 2.4 Bootstrap Framework

In programming, a "Bootstrap" is anything that launches another system, usually an operating system. In the actual world, a "bootstrap" is the little piece in the rear of a boot that is used to pull it on. Throughout software development, it is used to define a small system that initiates a much bigger system. Thus, "Bootstrap" refers to a framework that may be used to quickly develop a website. To utilize Bootstrap, you do not need to finish any of the standard prerequisite programs. Alternatively, you may focus entirely on what really counts—customizing it for your project.



*Figure 15: Bootstrap*

### **Advantages of Bootstrap:**

- It's responsive. Today, people are more likely to be browsing the web on a mobile phone or a tablet than on a personal computer. A responsive design will automatically resize to suit the page.
- It automatically resizes images and elements. Nothing has to be handled manually by the developer — it's all done for them.
- It's grid-based. The grid mechanic makes it much easier for developers to manage where elements show up on their site.
- It includes major components. Dropdowns, navigation bars, forms, and progress bars; it's all designed for you. You just need to drop them in.

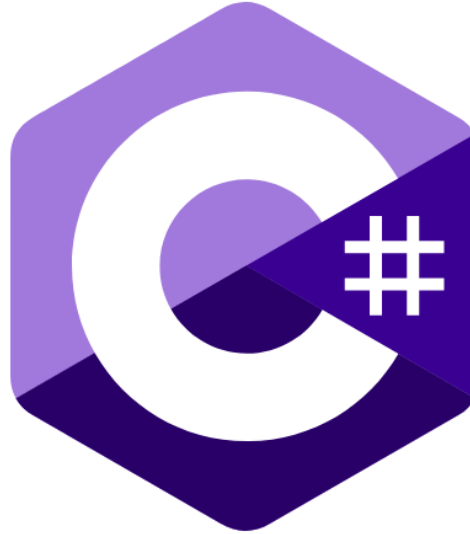
**Disadvantages of Bootstrap:**

- You must be proficient in coding. Bootstrap is not going to be helpful to someone who is not at anyway proficient in coding. If you don't know how to code, a content management system like WordPress is a better option.
- Its naming convention may be unclear. It will take some initial effort to get familiar with everything Bootstrap has to offer. It may need extensive documentation since it is impossible to deduce what is present or its name.
- It may not be as efficient as you would want. Because Bootstrap comes with everything you could possibly need, it is notorious for being bloated. It won't be suitable if you want to make a modest, straightforward website.
- It might grow overly dependent on you. You won't ever learn how to create a website from scratch if you get dependent on it, even if you may never need to.

### 3. Back End Technology Stack

#### 3.1 Programming Language

The term "C sharp" refers to the object-oriented programming language C#. It was created by Microsoft under the direction of Anders Hejlsberg and his team as part of the .Net program, and it has been approved by both the International Standards Organization (ISO) and the European Association of Computer Manufacturers (ECMA). The term's character # takes its name from the sharp musical key, which designates an increase in pitch of one semitone. The letter C is pronounced "see sharp".



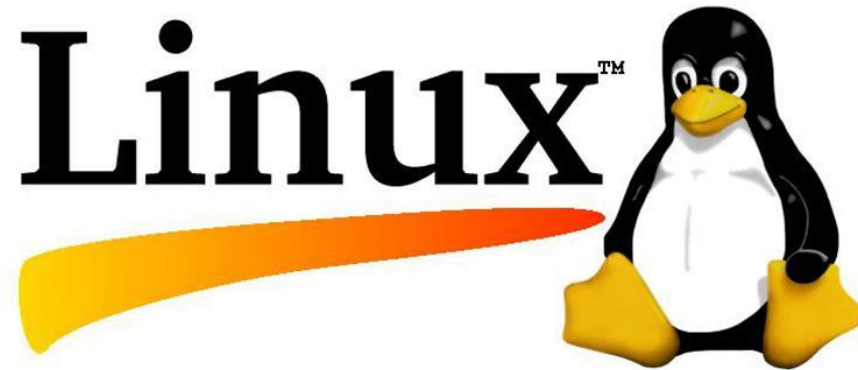
*Figure 16: C#*

### **The reasons why I choose C#:**

- **Robust and Powerful Language:** C# is a robust and powerful programming language thanks to its strong type system. Its modern language qualities allow for the creation of complex, scalable applications.
- **ASP.NET Core Web Framework:** This adaptable web framework was designed especially for web application development. It helps your team to create back-end solutions for your online bookstore that are secure, scalable, and highly effective.
- **Cross-Platform programming:** With the advent of .NET Core, C# enables cross-platform programming, allowing your team to build the back end for Windows, macOS, and Linux. This adaptability will allow your software to run on a wide range of operating systems.

### **3.2 Operating System**

Linux is a C-language computer operating system that was created in 1991 and is based on the Unix operating system. In contrast to industry titans like Windows and macOS, Linux continues to have a strong place in customers' hearts due to its numerous exceptional benefits and free distribution.



*Figure 17: Linux*

### 3.3 Web Server

The open-source program Nginx is used for web serving, load balancing, reverse proxying, caching, and video streaming, among other functions. Originally intended to be the web server with the highest level of reliability and speed, Nginx has developed into a flexible tool. Beyond only being an HTTP server, it may also be used as a load balancer for HTTP, TCP, and UDP servers, a reverse proxy, and an email proxy (IMAP, POP3, and SMTP).





*Figure 18: Nginx*

### **Reason I Choose:**

I firmly feel that Nginx is the best option for my project. Its high performance and parallelism speed up website response times and greatly improve user experience. One major benefit is that it can easily manage both static and dynamic servers, particularly with the help of technologies like PHP and Node.js. Time-saving sophisticated settings may be achieved by using Nginx's basic configuration and its interaction with technologies like cache and reverse proxy, which make system customization and maintenance easy. Nginx's dependability and stability, especially in situations of heavy load, give me confidence to deliver my product without worrying about system problems. Last but not least, Nginx's cross-platform compatibility makes it an adaptable option ideal for a range of deployment scenarios, bringing comfort and productivity to my project.

### **3.4 Database**

### 3.5 Hosting

To put it simply, web applications are hosted on Ubuntu as the operating system. Ubuntu is a Linux distribution that offers a dependable, open-source framework for setting up and maintaining different kinds of online services. It supports several applications and technologies and acts as the cornerstone for hosting solutions.



*Figure 19: Ubuntu*

#### **Benefits:**

Ubuntu's versatility allows it to handle a wide range of web technologies, which makes it appropriate for hosting programs written in many programming languages.

Ubuntu is renowned for its dependability and stability, which guarantees steady performance for hosted web apps.

**Security:** With frequent security updates and a proactive community that quickly fixes vulnerabilities, Ubuntu benefits from the strong security protections built into Linux.

**Cost-Effective:** Ubuntu's open-source nature eliminates license costs and lowers the total cost of hosting, making it an affordable operating system.

**Community Support:** Ubuntu has a large and vibrant community that offers guidance, answers, and documentation for a range of hosting situations.

**Management Ease:** Ubuntu makes managing hosted web applications easier with a variety of management tools and an intuitive command-line interface.

**Negative aspects:**

**Learning Curve:** There may be an initial learning curve for those who are not acquainted with Ubuntu or Linux.

**Restricted Integration with Microsoft Technologies:** Organizations that significantly depend on Microsoft solutions may want to take into account Ubuntu's potential integration constraints with certain Microsoft technologies.

**Dependency on System Administrator Skills:** In order to use Ubuntu for hosting effectively, system administrators may need to be competent, particularly when it comes to troubleshooting and complicated setups.

**Availability of Commercial assistance:** Although there is a lot of community assistance, some businesses may prefer the more restricted commercial support choices available with proprietary systems.

### 3.6 Frameworks

ASP.NET is a web application framework developed by Microsoft that enables developers to build dynamic, robust, and scalable websites, web applications, and web services. ASP.NET allows you to create web applications using various programming languages such as C#, Visual Basic.NET, and F#.

ASP.NET has been widely adopted by developers and organizations, particularly those in the Microsoft ecosystem. With its powerful features and ease of use, ASP.NET remains a popular choice for building modern web applications and services.



*Figure 20: ASP .NET*

### **The reason why I choose ASP .NET:**

**Robust and Mature Framework:** ASP.NET is a mature and well-established framework backed by Microsoft, which means it benefits from continuous improvements, updates, and extensive support. It has a large developer community and a wealth of resources available, including documentation, tutorials, and libraries.

**Flexibility in Language Choice:** ASP.NET allows you to write code in various languages such as C#, Visual Basic.NET, and F#. This flexibility allows you to leverage the expertise of your development team and choose the language that best suits your project requirements.

**Rapid Development with Visual Studio:** Microsoft's Visual Studio IDE provides excellent support for ASP.NET development. It offers powerful tools, code editors, debugging capabilities, and project templates, which can significantly speed up the development process.

**Scalability:** ASP.NET is designed to handle high levels of traffic and can scale to accommodate large numbers of users and requests. This makes it suitable for both small- scale applications.

## 4. Tools For Source Control Management

### 4.1 Github

For software engineers, GitHub is a web-based platform for version control and collaboration. Git is a tool for storing source code for projects and keeping track of all code modifications. By offering tools for handling potentially contradictory modifications from various developers, it enables developers to work on a project more successfully.

Developers can modify, adapt, and enhance software from GitHub's public repositories without paying a fee, but the company offers a number of premium options for private repositories. All of a project's files, together with each file's revision history, are stored in each public and private repository. Repositories can be either public or private, and they can have several collaborators.



*Figure 21: Github*

#### **The reasons why I choose GitHub in this project:**

**Industry Standard:** GitHub is one of the most widely used and respected platforms for source control management. We are already familiar with GitHub, making it easier for new team members to get up to speed quickly.

**Collaboration and Teamwork:** GitHub provides an excellent platform for collaboration. Multiple developers can work on the same project simultaneously, each contributing their changes through pull requests. This allows for easy code review and discussion before merging changes into the main codebase.

**Branching and Merging:** With GitHub, you can create branches for different features or bug fixes, allowing team members to work independently without interfering with each other's code. Once the work is complete, branches can be merged back into the main codebase seamlessly.

## 4.2 Visual Studio 2022

Visual Studio is a powerful developer tool that you can use to complete the entire development cycle in one place. It is a comprehensive integrated development environment (IDE) that you can use to write, edit, debug, and build code, and then deploy your app. Beyond code editing and debugging, Visual Studio includes compilers, code completion tools, source control, extensions, and many more features to enhance every stage of the software development process.

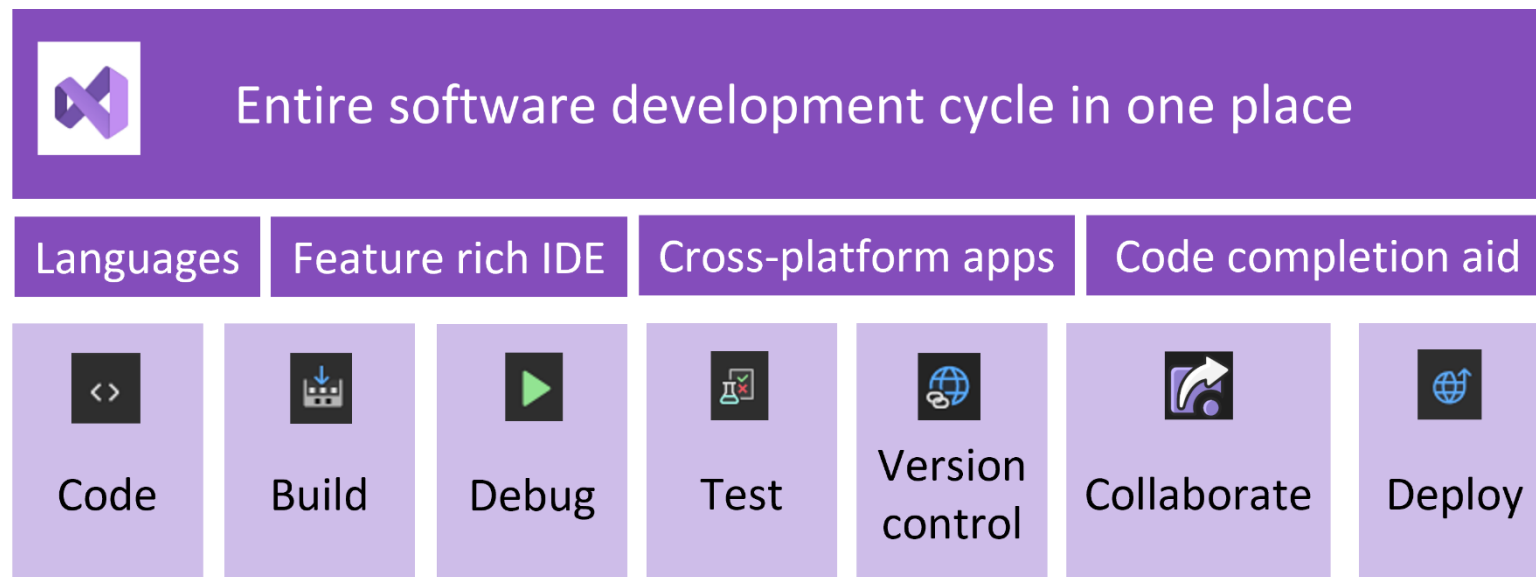


Figure 22: Visual Studio

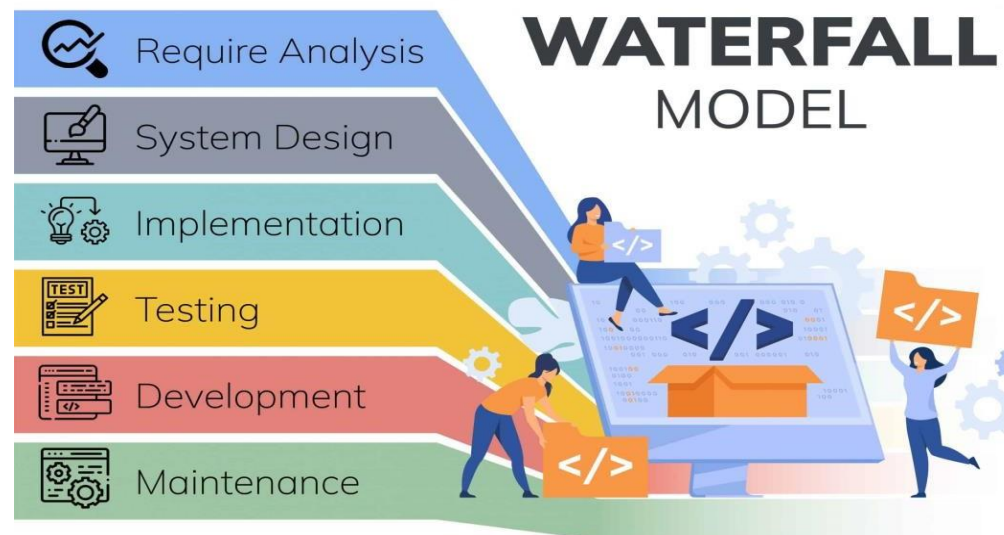
## 5. Software Development Models

### 5.1 Waterfall Model

The Waterfall model is a traditional and sequential software development process that follows a linear and systematic approach. It was one of the earliest software development methodologies and was widely used in the past, though it has been largely replaced by more iterative and agile approaches in modern software development.

The Waterfall model consists of the following phases, each of which must be completed before moving on to the next:

- **Requirements Gathering and Analysis:** In this phase, the project's requirements are gathered from stakeholders, analyzed, and documented in detail. The goal is to define what the software is expected to do and what features it should have.
- **System Design:** Based on the requirements, the system architecture and design are created. It involves breaking down the software into smaller components and defining how they will interact with each other.
- **Implementation:** This phase involves the actual coding and development of the software based on the design specifications. Programmers write code, and the software starts taking shape.
- **Testing:** Once the implementation is complete, the software is rigorously tested to ensure it meets the specified requirements and functions correctly. This phase includes both unit testing (testing individual components) and system testing (testing the entire system as a whole).



*Figure 23: Waterfall Model*

## 5.2 V Model

In the Waterfall model, testing is done in a separate stage. The V-model divides the entire operation into two groups, development and testing, which correspond to the two stages. Every development stage will be followed by a testing phase.

The core concept of the V-model is that, if at all possible, testing and development activities should begin the cycle concurrently. System analysis and design tasks, for example, may be carried out concurrently with system-wide test preparation procedures.

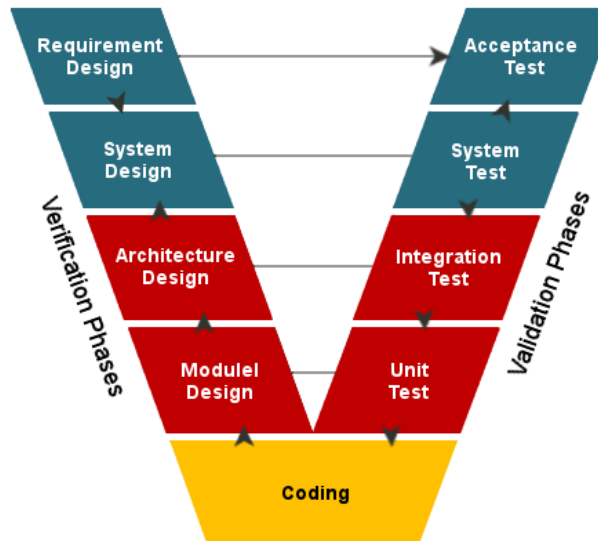


Figure 24: V Model

### 5.3 Spiral Model

The spiral model depicts the software development process as a spiral. Each spiral symbolizes a different stage of the software development process.

It is founded on the concept of risk minimization, which is accomplished by examining the risk factor at each iteration and employing prototyping methodologies. Each iterative phase in the development process begins with planning, risk analysis, prototyping, system refinement and development, review, and so on. The material consists of four major activities:



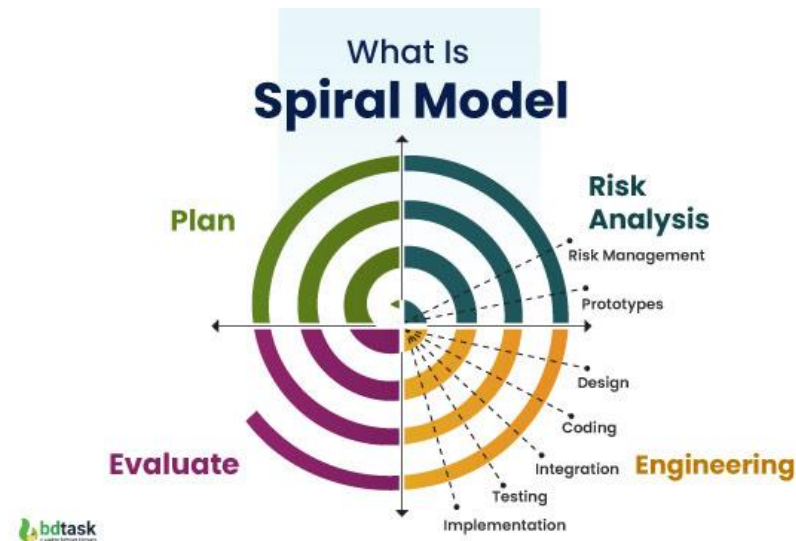


Figure 25: Spiral Model

## 5.4 Select Model

### Benefits:

**Structured and Predictable:** Project schedules and deliverables are more easily predicted thanks to the Waterfall model's structured and linear methodology.

**Unique Milestones:** Every stage in the Waterfall model includes a unique set of milestones that make it easy to assess and monitor progress.

**Well-Suited for Small Projects:** The Waterfall model may be a simple and efficient option for smaller projects with clear needs.

**Documentation:** At every stage, a great deal of documentation is produced to help with comprehension, upkeep, and any future improvements.

**Simplified Project Management:** The Waterfall model's linear structure streamlines project management by making resource allocation and activity planning simpler.

### **Negative aspects:**

**Restricted Flexibility:** Once a project has begun, it might be difficult to modify its needs due to the Waterfall model's rigidity.

**Late Customer Feedback:** Usually collected at the conclusion of the project, customer feedback might cause confusion and discontent if the finished result falls short of expectations.

**Extended Development Time:** Waterfall's sequential design may cause an extended period of development before any tangible products are produced.

**Increased chance of Failure:** The chance of a project failing rises when needs are not clearly specified at the outset or when there are major modifications made while the project is being developed.

**Not Ideal for Complex Projects:** If flexibility and iterative development are essential for large-scale, intricate projects, waterfall development may not be the best option.

### **Ground for Decision:**

I have decided to use the Waterfall technique to create a book-selling website with.NET in 4 weeks since the project requirements are clear and consistent. I think the Waterfall model will provide a simple route to project completion within the allotted time frame if there are defined milestones and an organized approach.

The Waterfall model is a good fit for the project's features because of its modest size and constrained timeline. Waterfall is a good option in this case since there aren't many requirements changes and early deployment is required. In order to guarantee a seamless and well-managed development process, I want to take use of its predictability and documentation advantages.

## 6. Drawing the Overview

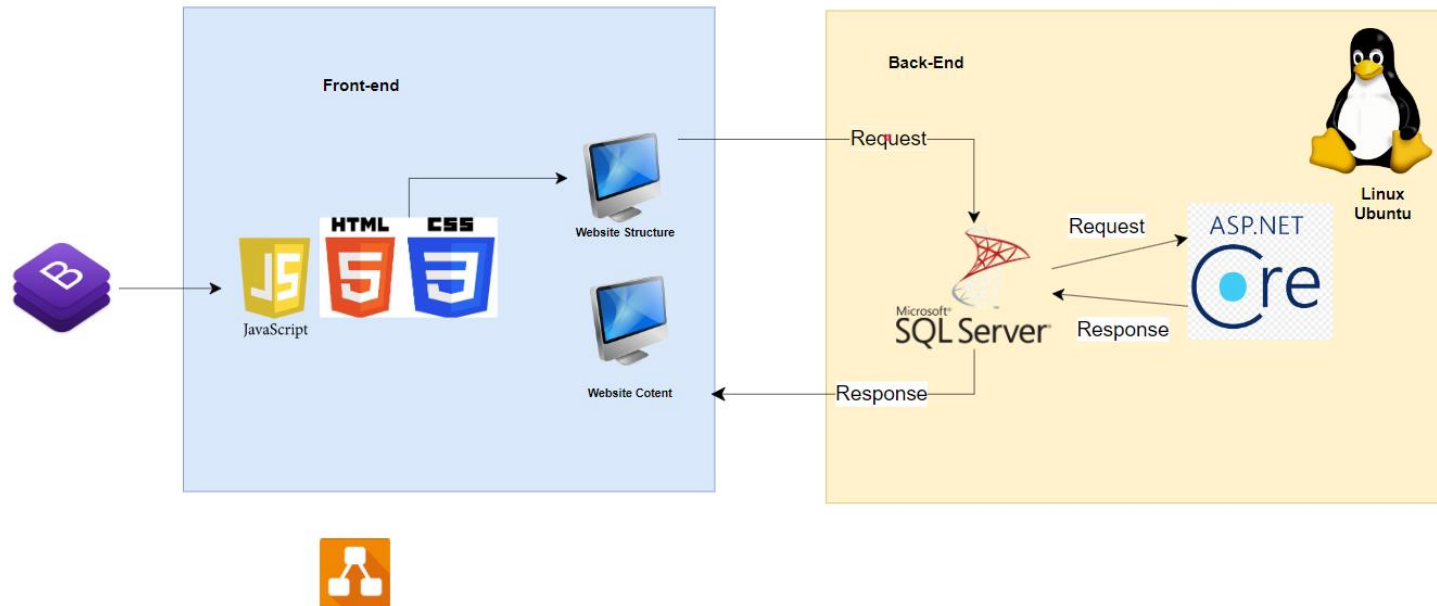


Figure 26: Overview

The meticulous selection of tools and techniques in our project forms a cohesive and tailored technology stack, precisely calibrated to address our unique needs. The front-end technologies have been chosen to ensure an intuitive and engaging user interface, while the design tools facilitate effective conceptualization of the website's layout and features. On the other hand, the back-end technologies provide a scalable and integrated solution, ensuring the website's functionality meets our requirements.

This comprehensive approach is aimed at creating a custom-built website that not only addresses the identified business-related problem but does so efficiently, with scalability, and a focus on user-centric design.

Overall, the harmonized use of these tools and techniques contributes to a development process characterized by precision, reliability, and optimal performance, allowing us to build a website that meets our project objectives with finesse.

## CHAPTER 3: TECHNOLOGIES EVALUATION (M2)

### 1. Design Tool – DrawIO

#### 1.1 Compare

Both Draw.io and Lucidchart are widely used diagramming tools in the software development landscape. They facilitate the creation of various diagrams, including flowcharts, UML diagrams, wireframes, and more. Here's a comparison focusing on their features and suitability within software development processes:

	<b>Draw.io</b>	<b>Lucidchart</b>
<b>Features and Interface</b>	<ul style="list-style-type: none"> <li>• Offers a vast array of diagramming elements and templates.</li> <li>• Open-source and can be used offline after installation.</li> <li>• Integrates with cloud storage services like Google Drive, OneDrive, etc.</li> <li>• User interface might feel less intuitive compared to some other tools.</li> </ul>	<ul style="list-style-type: none"> <li>• Collaborative features for team-based diagramming and editing.</li> <li>• Offers extensive integrations with various tools such as Jira, Confluence, etc.</li> <li>• More templates and shape libraries available compared to Draw.io.</li> </ul>
<b>Collaboration and Sharing</b>	<ul style="list-style-type: none"> <li>• Allows sharing diagrams via links or exporting them in various formats.</li> <li>• Collaboration features are somewhat limited compared to Lucidchart.</li> </ul>	<ul style="list-style-type: none"> <li>• Strong collaborative capabilities with real-time editing and commenting.</li> <li>• Sharing and collaboration options are more robust compared to Draw.io.</li> </ul>
<b>Pricing</b>	<ul style="list-style-type: none"> <li>• Free to use, as it's open-source. No subscription fees.</li> <li>• Offers extensive functionality without any cost.</li> </ul>	<ul style="list-style-type: none"> <li>• Offers a free version with limited functionality.</li> <li>• Subscription-based model for accessing advanced features and full functionality.</li> </ul>

## 1.2 Preferred Selection and Methodology

Draw.io stands out as an exceptional design tool due to its combination of versatility, ease of use, collaboration features, and extensive customization options. Its user-friendly interface and compatibility with various platforms make it an ideal choice for diverse teams working on projects. The seamless integration with cloud storage services and the ability to export files in multiple formats ensure flexibility and ease of access. Additionally, Draw.io's collaborative features foster efficient teamwork, enabling real-time editing and sharing among team members.

In comparison to Lucidchart or other similar tools, Draw.io's robust capabilities, coupled with its free availability and comprehensive functionality, make it a preferred choice for design needs in software development projects.

## 2. Front End – HTML, CSS, JavaScript, Bootstrap

### 1.1 Compare

Criteria	HTML, CSS, JavaScript, Bootstrap	React.js
<b>Ease of Learning</b>	Widely known and easy to start	Steeper learning curve initially, but powerful once mastered
<b>Component-Based</b>	Not inherently component-based	Built on component-based architecture, enabling modular development
<b>Performance</b>	Generally good performance	Virtual DOM for optimized rendering, improving performance
<b>Community Support</b>	Established community support	Strong and active community with extensive libraries and resources
<b>Ecosystem</b>	Mature ecosystem with wide support	Growing ecosystem with evolving best practices
<b>State Management</b>	Requires additional libraries for complex state management	Built-in state management capabilities with Redux or Context API
<b>Tooling</b>	Relatively basic tooling	Robust tooling and developer-friendly environment

## 1.2 Preferred Selection and Methodology

The preferred front-end technology stack for this project remains HTML, CSS, JavaScript, and Bootstrap framework due to its comprehensive advantages in ease of learning, flexibility, responsiveness, community support, and integration capabilities. The stack allows for efficient and rapid development with its well-documented resources, vast community support, and pre-built components provided by Bootstrap, enhancing the development process.

The chosen methodology involves leveraging the strengths of HTML, CSS, JavaScript, and Bootstrap to create a user-centric and responsive interface. Utilizing a mobile-first approach facilitated by Bootstrap ensures cross-device compatibility and responsiveness. Additionally, the extensive support and resources available for this stack empower the development team to deliver a robust and visually appealing front-end for the software.

## 3. Back End

### 3.1 Programming Language – C#

	C#	Java
<b>Syntax and Language Features</b>	C# provides a concise and modern syntax, with features like LINQ (Language Integrated Query) and asynchronous programming with async/await.	Java offers a robust and versatile syntax with a rich set of libraries and frameworks, although it may have more verbose syntax compared to C#.
<b>Platform Independence</b>	C# was initially designed for Windows, but with the advent of .NET Core, it has become more cross-platform.	Java is renowned for its "write once, run anywhere" capability, being highly portable across different platforms and operating systems.
<b>Performance</b>	C# is known for its performance due to its integration with Windows and optimizations.	Java also offers good performance and scalability, but its performance might vary depending on the implementation and runtime environment.
<b>Ecosystem and Libraries</b>	C# has a strong ecosystem, especially with the .NET framework, offering a wide range of libraries and tools for development.	Java boasts a vast ecosystem with numerous libraries and frameworks, providing extensive support for various functionalities.

<b>Community and Support</b>	C# has a dedicated and growing community, backed by Microsoft's support and resources.	Java has a massive and established community, with extensive documentation, support, and a wealth of resources.
------------------------------	--	---

### Preferred Selection and Methodology

The preferred choice for the back-end technology stack in this scenario is C#. While both C# and Java offer robust features and extensive support, C# stands out due to its modern syntax, performance optimizations, and a growing ecosystem, especially with the evolution of .NET Core, enabling cross-platform compatibility.

For this project, leveraging C# allows for efficient development, benefiting from its concise syntax, performance optimizations, and extensive libraries available within the .NET ecosystem. Its compatibility with Windows and the ability to deploy across various platforms using .NET Core make it a suitable choice.

The methodology involves utilizing C# along with the .NET framework or .NET Core to build the back-end functionalities of the software. This includes leveraging C#'s features, libraries, and tools within the .NET ecosystem to create a scalable, performant, and reliable back-end infrastructure for the software application.

### 3.2 Operating System – Linux

	<b>Linux</b>	<b>Windows</b>
<b>Open Source and Flexibility</b>	Linux is open-source, offering flexibility for customization and modification according to specific needs.	Windows, while not open-source, provides a user-friendly interface and a wide range of software compatibility.
<b>Cost</b>	Linux distributions are typically free of cost, offering a cost-effective solution for deployment.	Windows often involves licensing fees, which can contribute to higher operational costs.
<b>Security</b>	Linux is renowned for its robust security features, with fewer vulnerabilities and a strong emphasis on permissions and user management.	Windows has a substantial user base and, as a result, may face more frequent security threats. However, it offers robust security features with regular updates and patches.

<b>Customization and Terminal Access</b>	Linux allows extensive customization and access to the terminal, offering greater control over system configurations and management.	Windows provides a user-friendly interface but may have limitations in terms of customization and terminal access.
<b>Support and Compatibility</b>	Linux has a vast community support base, offering extensive documentation, forums, and resources. Compatibility with various software and hardware may require additional configurations.	Windows benefits from strong vendor support, extensive compatibility with software and hardware, and a large user base, ensuring widespread application support.

### Preferred Selection and Methodology:

The preferred choice for the operating system in this scenario is Linux due to its open-source nature, flexibility, robust security features, and cost-effectiveness. Leveraging Linux as the operating system for the project offers the advantage of customization, strong security, and cost efficiency.

The methodology involves utilizing a Linux distribution such as Ubuntu, CentOS, or Debian as the operating system for hosting the software application. This entails deploying the software on Linux servers or virtual machines, taking advantage of its customizable nature, robust security features, and the vast community support available.

### 3.3 Database – SQL Server

	<b>SQL Server</b>	<b>PostgreSQL</b>
<b>Licensing and Cost</b>	SQL Server often involves licensing fees, especially for enterprise editions, which can contribute to higher operational costs.	PostgreSQL is open-source and free, making it a cost-effective solution without any licensing fees, which can significantly reduce overall expenses.
<b>Performance and Scalability</b>	SQL Server is known for its excellent performance, especially in Windows environments. It offers high scalability and optimization features.	PostgreSQL also offers strong performance and scalability, especially for complex queries and larger datasets, although its performance might vary based on configurations.



<b>Features and Ecosystem</b>	SQL Server provides a comprehensive set of features, tools, and integrations within the Microsoft ecosystem, offering seamless integration with other Microsoft products.	PostgreSQL has a rich set of features and extensions, providing a wide range of functionalities and a strong ecosystem with various tools and integrations.
<b>Platform Independence</b>	SQL Server is primarily optimized for Windows but offers some level of compatibility with Linux and Docker environments.	PostgreSQL is platform-independent and can run on various operating systems, providing greater flexibility for deployment.
<b>Community Support</b>	SQL Server has a robust support system provided by Microsoft, offering extensive documentation, forums, and resources.	PostgreSQL also has a large and active community, offering comprehensive support, frequent updates, and a wealth of resources.

### Preferred Selection and Methodology:

The preferred choice for the database in this scenario is SQL Server due to its comprehensive features, strong performance in Windows environments, robust support from Microsoft, and seamless integration within the Microsoft ecosystem.

The methodology involves utilizing SQL Server as the database management system for the software application. Leveraging SQL Server offers benefits such as optimized performance, scalability, comprehensive toolsets, and integration capabilities with other Microsoft products. It involves setting up and managing databases on SQL Server instances, ensuring efficient data storage and retrieval for the software application.

This comparison and preferred selection emphasize the strengths of SQL Server over PostgreSQL in terms of features, performance, ecosystem, support, and integration within the Microsoft environment, justifying its preference for the database management system in the software development project.

### 3.4 Frameworks - .NET

<b>Framework</b>	<b>.NET Framework</b>	<b>Django</b>
<b>Language Support</b>	.NET Framework supports multiple languages like C#, VB.NET, and F# among others, offering flexibility for	Django is primarily based on Python, which offers simplicity, readability, and a large community base.

	developers to choose the language of their preference.	
<b>Ecosystem and Tools</b>	.NET Framework provides a comprehensive ecosystem with extensive tools, libraries, and integrations within the Microsoft ecosystem.	Django offers a rich ecosystem with various built-in features, libraries, and third-party packages available through PyPI (Python Package Index).
<b>Platform Independence</b>	.NET Framework initially focused on Windows but evolved with .NET Core to become more cross-platform, enabling deployment on Linux and macOS.	Django, being Python-based, is inherently cross-platform and can be deployed on different operating systems.
<b>Scalability</b>	.NET Framework offers strong scalability and performance, especially with its optimization for Windows environments and robust features like Entity Framework for database operations.	Django also provides good scalability and performance, with features like Django ORM (Object-Relational Mapping) for database interactions.
<b>Community Support</b>	.NET Framework benefits from a strong and active community support base, with extensive documentation, forums, and resources provided by Microsoft.	Django has a large and vibrant community, offering comprehensive support, frequent updates, and a wealth of resources.

### Preferred Selection and Methodology:

The preferred choice for the framework in this scenario is .NET Framework due to its versatility, comprehensive tooling, ecosystem, and strong integration within the Microsoft environment.

The methodology involves leveraging .NET Framework to build the software application. Utilizing .NET Framework offers benefits such as multi-language support, a vast ecosystem with tools and libraries, cross-platform compatibility, scalability, and integration capabilities with other Microsoft technologies. It involves developing the application using C#, VB.NET, or other supported languages within the .NET environment, ensuring efficient and effective development of the software.

This comparison and preferred selection emphasize the strengths of .NET Framework over Django in terms of language support, ecosystem, platform compatibility, scalability, community support, and integration within the Microsoft environment, justifying its preference for the framework in the software development project.

### 3.5 Web Server – Nginx

	<b>Nginx</b>	<b>Apache2</b>
<b>Performance</b>	Nginx is known for its high performance and efficiency, particularly in handling concurrent connections and static content. It utilizes an asynchronous, event-driven architecture.	Apache2 is highly versatile but may not perform as well as Nginx in scenarios involving high concurrent connections due to its multi-threaded architecture.
<b>Resource Usage</b>	Nginx typically consumes less memory and CPU resources compared to Apache2, making it more efficient, especially in high-traffic situations.	Apache2 may require more system resources, especially when handling a large number of simultaneous connections.
<b>Configuration</b>	Nginx uses a declarative configuration syntax that is often considered more straightforward and easier to understand.	Apache2 utilizes an extensive configuration file, which may be more complex for some users, but it provides more flexibility and customization options.
<b>Modules and Extensions</b>	Nginx offers a limited set of core modules, but it can be extended through third-party modules.	Apache2 provides a vast array of built-in modules and third-party extensions, offering a wider range of functionalities out-of-the-box.
<b>Usage and Popularity</b>	Nginx has gained popularity in recent years due to its performance, scalability, and use in high-traffic websites and applications.	Apache2 has been a dominant web server for a long time and remains widely used, especially in shared hosting environments and traditional web setups.

#### Preferred Selection and Methodology:

The preferred choice for the web server in this scenario is Nginx due to its high performance, efficiency, lower resource usage, and scalability, particularly in scenarios with high concurrent connections.

The methodology involves deploying Nginx as the web server to host the software application. Leveraging Nginx offers benefits such as superior performance, efficient resource utilization, simpler configuration syntax, and the ability to handle high traffic loads effectively. It involves setting up Nginx to serve web content, manage requests, and act as a reverse proxy if required, ensuring optimal performance and reliability for the software application.

This comparison and preferred selection emphasize the strengths of Nginx over Apache2 in terms of performance, resource usage, configuration, and scalability, justifying its preference for the web server in the software development project.

### 3.6 Hosting – Ubuntu

	Ubuntu	Azure
<b>Operating System</b>	Ubuntu is a Linux distribution, offering an open-source and free operating system widely used for server deployments.	Azure is a cloud computing service by Microsoft, providing a range of cloud-based solutions and services, including infrastructure as a service (IaaS), platform as a service (PaaS), and more.
<b>Deployment Flexibility</b>	Ubuntu allows deployment on physical servers, virtual machines, and cloud environments, providing flexibility in hosting options.	Azure offers a wide range of cloud services and deployment options, including virtual machines, containers, serverless computing, and more, enabling scalability and flexibility.
<b>Management and Control</b>	Ubuntu gives administrators full control over server configurations, allowing customization and direct management of server resources.	Azure provides a managed cloud environment, offering various management tools and services for automated deployment, scaling, and monitoring.
<b>Cost Structure</b>	Ubuntu is typically free to use, with no licensing fees required for the operating system itself. However, costs may incur for hosting infrastructure or support services.	Azure operates on a pay-as-you-go model, where users pay for the resources and services they consume, offering flexibility in cost management but with potential costs based on usage.
<b>Support and Community</b>	Ubuntu benefits from a large and active open-source community, providing extensive documentation, forums, and support resources.	Azure offers comprehensive technical support and documentation, backed by Microsoft's resources and expertise.

#### Preferred Selection and Methodology:

The preferred choice for hosting in this scenario is Ubuntu due to its flexibility, open-source nature, deployment options, and the ability to customize server configurations.

The methodology involves deploying Ubuntu as the hosting solution for the software application. Utilizing Ubuntu allows for deployment on various hosting environments, such as physical servers, virtual machines, or cloud-based solutions. It involves setting up and managing Ubuntu servers, configuring them to host the software application, and ensuring reliability, security, and scalability as needed.

This comparison and preferred selection emphasize the strengths of Ubuntu over Azure in terms of flexibility, control, cost structure, and open-source community support, justifying its preference for the hosting solution in the software development project.

## 4. Tools For Source Control Management

### 1.1 Github

	GitHub	Google Drive
<b>Version Control</b>	GitHub is primarily a version control platform using Git, designed for software development projects, offering features like version history, branching, merging, and collaboration among developers.	Google Drive provides file storage and sharing capabilities but is not primarily designed for version control in software development. It offers basic version history but lacks advanced version control functionalities like branching and merging.
<b>Collaboration and Documentation</b>	GitHub facilitates collaboration among developers with features like pull requests, code reviews, issue tracking, and wikis. It provides a structured environment for team collaboration and project documentation.	Google Drive enables file sharing and collaboration, allowing multiple users to collaborate on documents simultaneously. However, it may not provide specialized tools for code-related collaboration or project management.
<b>Code Management and Hosting</b>	GitHub hosts code repositories, making it a central hub for storing, managing, and sharing code among team members. It allows for code hosting, code reviews, and integration with various development tools.	Google Drive is primarily focused on file storage and lacks the specialized features for code hosting, code reviews, and integration with development workflows like GitHub.
<b>Community and Integration</b>	GitHub has a large community of developers, offering extensive documentation, forums, and resources. It integrates with various development tools	Google Drive integrates with other Google Workspace applications and offers seamless integration within the Google ecosystem but

	and services, fostering a collaborative development environment.	may not have specialized developer-oriented integrations like GitHub.
--	--	---

### Preferred Selection and Methodology:

The preferred choice for hosting in this scenario is Ubuntu due to its flexibility, open-source nature, deployment options, and the ability to customize server configurations.

The methodology involves deploying Ubuntu as the hosting solution for the software application. Utilizing Ubuntu allows for deployment on various hosting environments, such as physical servers, virtual machines, or cloud-based solutions. It involves setting up and managing Ubuntu servers, configuring them to host the software application, and ensuring reliability, security, and scalability as needed.

This comparison and preferred selection emphasize the strengths of Ubuntu over Azure in terms of flexibility, control, cost structure, and open-source community support, justifying its preference for the hosting solution in the software development project.

## 1.2 Visual Studio 2022

	Visual Studio	Rider
<b>Ecosystem and Integration</b>	Visual Studio offers a comprehensive integrated development environment by Microsoft, providing a wide range of tools, languages, and integrations within the Microsoft ecosystem.	Rider is an IDE developed by JetBrains, offering cross-platform capabilities and compatibility with various programming languages and frameworks, including .NET, C#, and others.
<b>Language Support</b>	Visual Studio supports multiple programming languages, including C#, VB.NET, F#, JavaScript, Python, and more, providing extensive language support within the Microsoft ecosystem.	Rider focuses on cross-platform development and supports various programming languages like C#, VB.NET, F#, JavaScript, TypeScript, HTML, CSS, and more, catering to a broader range of developers.
<b>Features and Tooling</b>	Visual Studio provides a wide array of features, debugging tools, code analysis, extensions, and project management tools	Rider emphasizes productivity, offering advanced code analysis, refactorings, debugging tools, and cross-platform

	specifically optimized for .NET development.	development features. It also integrates with ReSharper, a popular productivity extension.
<b>Performance</b>	Visual Studio is known for its robustness and performance, especially for .NET development, offering a stable and feature-rich development environment.	Rider is recognized for its fast performance and responsiveness, providing a smooth and efficient development experience across different platforms.
<b>Community Support</b>	Visual Studio benefits from a large and active community, extensive documentation, forums, and resources provided by Microsoft and the developer community.	Rider has a growing community and receives support from JetBrains, offering documentation, forums, and resources for developers.

### Preferred Selection and Methodology:

The preferred choice for the integrated development environment (IDE) in this scenario is Visual Studio due to its specialization in .NET development, extensive Microsoft ecosystem integration, and robust tooling for .NET development.

The methodology involves utilizing Visual Studio as the primary IDE for software development. This includes leveraging its features, debugging tools, project management capabilities, and integration with the .NET framework to facilitate efficient .NET development. It involves creating and managing projects, writing and debugging code, and utilizing its ecosystem for .NET-based software development.

This comparison and preferred selection emphasize the strengths of Visual Studio over Rider in terms of .NET specialization, extensive Microsoft ecosystem integration, and robust tooling for .NET development, justifying its preference as the primary IDE for software development in the project.



## CHAPTER 4: JUSTIFY THE TOOLS AND TECHNIQUES CHOSEN TO REALISE A CUSTOM BUILT WEBSITE (D1)

### 1. C# .NET

C# .NET Core MVC was chosen as the primary framework for this project for several reasons, providing a solid foundation for the development of a custom-built website:

- **Versatility and Robustness:** C# .NET Core MVC offers a versatile and robust framework for web development. It provides a structured approach to building web applications, enabling developers to create scalable, high-performance websites.
- **Cross-Platform Compatibility:** .NET Core is designed to be cross-platform, allowing the developed application to run on various operating systems, enhancing its flexibility and reach.
- **Ecosystem Integration:** Being part of the Microsoft ecosystem, C# .NET Core MVC integrates seamlessly with other Microsoft technologies, libraries, and tools. This integration simplifies development, enhances productivity, and allows for easy access to a wide range of resources.
- **Security Features:** .NET Core MVC emphasizes security features, offering built-in security mechanisms that aid in mitigating common web vulnerabilities, ensuring a more secure application.
- **Scalability and Performance:** It's known for its scalability and performance optimizations, making it suitable for handling both small-scale and enterprise-level web applications efficiently.
- **Community Support:** C# .NET Core MVC has a large community of developers, providing extensive documentation, forums, and resources. This robust community support facilitates learning, issue resolution, and staying up-to-date with best practices.

Overall, C# .NET Core MVC was chosen as it provides a powerful and comprehensive framework that aligns with the project's requirements for building a feature-rich website. Its versatility, robustness, security features, scalability, and integration within the Microsoft ecosystem contribute to its suitability for developing a custom-built website to meet the project's objectives effectively.

### 2. Visual Studio 2022

Visual Studio 2022 was selected as the integrated development environment (IDE) for several compelling reasons, providing a robust platform for developing the custom-built website:



- **Comprehensive Toolset:** Visual Studio 2022 offers a comprehensive set of tools, features, and functionalities that streamline the development process. It provides an extensive range of integrated tools for coding, debugging, testing, and deployment, enhancing productivity and efficiency.
- **Advanced Debugging Capabilities:** The IDE's advanced debugging capabilities facilitate the identification and resolution of coding errors swiftly. Features such as IntelliSense, live debugging, and error highlighting contribute to a smoother development experience.
- **Seamless .NET Integration:** As a Microsoft product, Visual Studio seamlessly integrates with the .NET framework. This integration simplifies the development of .NET Core MVC applications, providing a conducive environment for coding, testing, and deploying web applications.
- **Cross-Platform Development:** Visual Studio 2022 supports cross-platform development, allowing developers to work on various operating systems. This flexibility is valuable for teams working in diverse environments and aids in the project's scalability.
- **Extensive Extension Ecosystem:** It offers a rich ecosystem of extensions and plugins, allowing developers to customize their development environment according to specific project needs. These extensions enhance functionality, provide additional tools, and support integration with other platforms and services.
- **Microsoft Ecosystem Integration:** Visual Studio is deeply integrated with other Microsoft services, enabling smooth interactions with Azure cloud services, GitHub, and other Microsoft tools. This integration simplifies workflows and enhances collaboration within the Microsoft ecosystem.

Overall, Visual Studio 2022 was chosen for its robustness, comprehensive toolset, advanced debugging capabilities, seamless .NET integration, cross-platform support, extensibility, and integration within the Microsoft ecosystem. These features collectively contribute to an efficient development process, aligning well with the project's objectives of creating a custom-built website.

### 3. GitHub

GitHub was selected as the version control system for this project due to several key reasons, contributing to an effective and collaborative software development process:

- **Specialization in Version Control:** GitHub is specifically designed for version control and collaboration in software development. It offers robust version history, branching, merging, and collaboration features, essential for managing code changes efficiently.

- **Collaborative Development:** GitHub facilitates collaborative development among team members by providing tools for code review, pull requests, and issue tracking. These features enhance collaboration, allowing team members to work concurrently on different branches while maintaining code integrity.
- **Secure and Reliable Platform:** GitHub provides a secure platform for hosting repositories, ensuring data integrity, access control, and secure collaboration. Its reliability, uptime, and disaster recovery mechanisms make it a trusted platform for version control.
- **Extensive Documentation and Community Support:** GitHub offers extensive documentation, tutorials, and resources. Its active community provides forums, discussions, and support, enabling developers to resolve issues efficiently and stay updated with best practices.
- **Integration Capabilities:** GitHub integrates seamlessly with various development tools, continuous integration/continuous deployment (CI/CD) pipelines, and services like Travis CI, Jenkins, and Azure DevOps, streamlining the development workflow.
- **Open Source Collaboration:** GitHub has a strong focus on open source collaboration, encouraging contribution and participation in open source projects. This aligns with the ethos of collaboration, sharing, and transparency in software development.

Overall, GitHub's specialization in version control, collaborative development features, security, reliability, extensive documentation, community support, integration capabilities, and focus on open source collaboration made it an ideal choice for managing version control and facilitating collaboration in the development of the custom-built website.

#### 4. Nginx

Nginx was chosen as the web server for this project for several reasons, contributing to its suitability in hosting and serving the custom-built website:

- **High Performance and Scalability:** Nginx is renowned for its high performance, efficiency, and scalability. It excels in handling concurrent connections and delivering static content swiftly, making it suitable for hosting websites, especially in scenarios with high traffic loads.
- **Efficient Resource Utilization:** Compared to other web servers like Apache2, Nginx typically consumes fewer system resources such as memory and CPU, while efficiently managing and serving web content. This efficiency contributes to better server performance and cost-effectiveness.
- **Asynchronous, Event-Driven Architecture:** Nginx's architecture is designed to handle multiple simultaneous connections efficiently through its asynchronous, event-driven model. This allows it to handle a large number of concurrent requests without consuming excessive server resources.

- **Reverse Proxy Capabilities:** Nginx's reverse proxy capabilities are robust, making it suitable for load balancing, caching, and acting as a reverse proxy server. These features enhance performance, reliability, and security by distributing incoming traffic across multiple servers.
- **Ease of Configuration:** Nginx offers a straightforward and declarative configuration syntax, making it easier to configure and manage compared to some other web servers. This simplicity in configuration facilitates efficient server management.
- **Community Support and Documentation:** Nginx has a strong community and extensive documentation, providing resources, forums, and support for troubleshooting and optimizing server configurations.

Overall, Nginx's high performance, efficiency in resource utilization, asynchronous architecture, reverse proxy capabilities, ease of configuration, community support, and documentation make it an optimal choice for hosting and serving the custom-built website. Its ability to handle high traffic loads efficiently aligns well with the project's requirements for a reliable and high-performing web server.

## 5. Ubuntu for Hosting

Ubuntu was chosen as the hosting platform for this project due to several key reasons, contributing to its suitability for hosting the custom-built website:

- **Flexibility and Compatibility:** Ubuntu, being a Linux distribution, offers flexibility and compatibility with various hosting environments, including physical servers, virtual machines, and cloud platforms. This flexibility allows for seamless deployment and scalability, aligning with the project's needs.
- **Open-Source Nature:** Ubuntu is an open-source operating system, providing cost-effectiveness by eliminating licensing fees. It offers free access to a wide array of tools, packages, and repositories, reducing operational costs for hosting.
- **Stability and Reliability:** Ubuntu is known for its stability and reliability, providing a robust operating system environment for hosting web applications. Its regular updates and Long-Term Support (LTS) releases ensure stability and security patches for an extended period.
- **Community Support and Documentation:** Ubuntu benefits from a vast and active open-source community that offers extensive documentation, forums, and resources. The strong community support facilitates issue resolution, troubleshooting, and staying updated with best practices.
- **Security Features:** Ubuntu emphasizes security, providing built-in security features and regular security updates, ensuring a secure hosting environment for web applications.

- **Compatibility with Development Tools:** Ubuntu is compatible with a wide range of development tools, software, and frameworks commonly used in web development, ensuring compatibility and ease of integration within the development workflow.

Overall, Ubuntu's flexibility, open-source nature, stability, reliability, security features, strong community support, and compatibility with various hosting environments make it an ideal choice for hosting the custom-built website. Its cost-effectiveness, stability, and compatibility align well with the project's requirements for a reliable and scalable hosting platform.

## 6. SQL Server

SQL Server was chosen as the database management system (DBMS) for this project due to several compelling reasons, contributing to its suitability for managing the custom-built website's database:

- **Comprehensive Features and Tools:** SQL Server offers a comprehensive set of features, tools, and functionalities for efficient database management. It provides robust data storage, retrieval, querying, and management capabilities, essential for the website's data requirements.
- **Scalability and Performance:** SQL Server is known for its scalability and performance optimizations. It can efficiently handle large datasets, complex queries, and high transaction volumes, ensuring the website's responsiveness and reliability, even under heavy loads.
- **Security Measures:** SQL Server prioritizes security, offering built-in security features such as data encryption, access control mechanisms, auditing, and compliance tools. These features help in securing sensitive data and complying with industry standards.
- **Integration with Microsoft Ecosystem:** As part of the Microsoft ecosystem, SQL Server seamlessly integrates with other Microsoft technologies, frameworks, and tools like .NET Core, Visual Studio, and Azure services. This integration simplifies development, data access, and application deployment.
- **Reliability and Disaster Recovery:** SQL Server provides reliability features, including backup and recovery options, high availability solutions, and failover clustering, ensuring data integrity and minimal downtime.
- **Support and Documentation:** SQL Server benefits from extensive documentation, resources, and a strong community, facilitating troubleshooting, performance tuning, and staying updated with best practices.

Overall, SQL Server's comprehensive features, scalability, performance optimizations, security measures, integration within the Microsoft ecosystem, reliability, and support align well with the project's requirements for managing a robust and secure database system for the custom-built website. Its capabilities ensure efficient data handling, secure storage, and reliability for the website's data management needs.

## 7. The Waterfall

The Waterfall model was chosen as the software development methodology for this project due to its structured and sequential approach, aligning effectively with the project's specific requirements and objectives:

- **Structured Approach:** The Waterfall model follows a well-defined structure with distinct phases (requirements, design, implementation, testing, deployment, and maintenance), ensuring a systematic and organized development process. This structure facilitates clear deliverables at the end of each phase, aiding in project management and progress tracking.
- **Clarity in Requirements:** The Waterfall model emphasizes gathering and documenting comprehensive requirements at the initial stage. This approach ensures a thorough understanding of the project's scope, reducing the likelihood of scope creep and late-stage requirement changes.
- **Sequential Progression:** The Waterfall model progresses through phases sequentially, ensuring that each phase's outputs serve as inputs for the subsequent phase. This approach allows for better predictability and control over the project timeline and deliverables.
- **Emphasis on Documentation:** The model emphasizes documentation at each phase, providing detailed documentation for requirements, design specifications, and other artifacts. This documentation aids in understanding the project's progress, facilitating future maintenance and upgrades.
- **Risk Mitigation:** While it may be less flexible to changes once a phase is completed, the Waterfall model attempts to mitigate risks by focusing on comprehensive planning and detailed documentation at each stage.
- **Well-Suited for Well-Defined Requirements:** The Waterfall model is particularly suitable for projects with well-defined and stable requirements at the outset. It aligns well with scenarios where the scope and objectives are clear from the project's initiation.

Overall, the structured, sequential, and document-centric nature of the Waterfall model aligns with the project's requirements. Its emphasis on thorough planning, clear milestones, and comprehensive documentation ensures a systematic approach to developing the custom-built website, aiming to meet the project's objectives within defined timelines and requirements.

## CHAPTER 5: REFERENCES

.javatpoint., n.d. *javatpoint*. [Online]

Available at: [https://l.facebook.com/l.php?u=https%3A%2F%2Fwww.javatpoint.com%2Fsoftware-engineering-agile-model%3Ffbclid%3DIwAR1K8L3tyg-IPKglP5CxytINYeJNdt4fNdyAaEQD8vDWONfAZGmTpBpSI4k&h=AT3n6hsHdY-EbYaZHKkvdzRRfLZxt3xD9k4aBaOkKwC0yHkTptNSPxY3LBkaQWr6\\_fJOkpQ50mrVpx-2x](https://l.facebook.com/l.php?u=https%3A%2F%2Fwww.javatpoint.com%2Fsoftware-engineering-agile-model%3Ffbclid%3DIwAR1K8L3tyg-IPKglP5CxytINYeJNdt4fNdyAaEQD8vDWONfAZGmTpBpSI4k&h=AT3n6hsHdY-EbYaZHKkvdzRRfLZxt3xD9k4aBaOkKwC0yHkTptNSPxY3LBkaQWr6_fJOkpQ50mrVpx-2x)  
[Accessed 1 12 2023].

amela, 2023. *amela*. [Online]

Available at: <https://amela.vn/5-phan-mem-thiet-ke-wireframe-mien-phi-tot-nhat-2023/#:~:text=%C4%90a%20n%E1%BB%81n%20t%E1%BA%A3ng%3A%20Mockflow%20l%C3%A0,s%E1%BA%BB%20v%E1%BB%9Bi%20nh%C3%B3m%20l%C3%A0m%20vi%E1%BB%87c>  
[Accessed 1 12 2023].

B., R. B. & A., 2023. *hostinger*. [Online]

Available at:  
[https://l.facebook.com/l.php?u=https%3A%2F%2Fwww.hostinger.co.uk%2Ftutorials%2Fnginx-vs-apache-what-to-use%2F%3Ffbclid%3DIwAR1wJ7ygeGcaWZKoN\\_aMII8pJFM3JmUomYSgyGnM0djrMZGA\\_CsqIUL\\_5e8&h=AT3n6hsHdY-EbYaZHKkvdzRRfLZxt3xD9k4aBaOkKwC0yHkTptNSPxY3LBkaQWr6\\_fJOkp](https://l.facebook.com/l.php?u=https%3A%2F%2Fwww.hostinger.co.uk%2Ftutorials%2Fnginx-vs-apache-what-to-use%2F%3Ffbclid%3DIwAR1wJ7ygeGcaWZKoN_aMII8pJFM3JmUomYSgyGnM0djrMZGA_CsqIUL_5e8&h=AT3n6hsHdY-EbYaZHKkvdzRRfLZxt3xD9k4aBaOkKwC0yHkTptNSPxY3LBkaQWr6_fJOkp)  
[Accessed 1 12 2023].

Duy, K. M., 2023. *Our team*. Da Nang: Greenwich University.

enlabsoftware, 2020. *enlabsoftware*. [Online]

Available at: <https://enlabsoftware.com/offshore-software-development/how-to-write-a-good-srs-for-software-outsourcing-project.html>  
[Accessed 30 11 2023].

GeeksforGeeks, 2023. *geeksforgeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/how-to-write-a-good-srs-for-your-project/>  
[Accessed 30 11 2023].

Herbert, D., 2023. *blog.hubspo*. [Online]

Available at: <https://blog.hubspot.com/website/react-js>  
[Accessed 1 12 2023].

Kravchenko, I., 2020. *diceus*. [Online]

Available at: <https://diceus.com/custom-software-requirements-specification/>  
[Accessed 30 11 2023].

Krüger, G., 2023. *perforce*. [Online]

Available at: <https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document>

[Accessed 30 11 2023].

microsoft, 2023. *learn.microsoft*. [Online]

Available at: <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>

[Accessed 1 12 2023].

Shankar, R., 2023. *hackr.io*. [Online]

Available at: <https://hackr.io/blog/c-sharp-vs-python>

[Accessed 1 12 2023].

Wiegers, K. E., 2002. *Queen School of Computing*. [Online]

Available at: [https://research.cs.queensu.ca/home/emads/teaching/slides/srs\\_template\\_sep14.pdf](https://research.cs.queensu.ca/home/emads/teaching/slides/srs_template_sep14.pdf)

[Accessed 30 11 2023].