

The system we will be making is a simple, **not-too-complicated** Social Media Web Application that will allow users to sign up for an account and create user profiles.

This Web Application will take into account, the Security aspect of the modern Web Application and consider several important Search Engine Optimization (SEO) goals with respect to ranking in Google Search and other Search Engines.

After developing our own Social Media Application, we will look at how to deploy the Web Application in the production environment so that users can start signing up for accounts and start using the application. In the deployment section, we will look at how to use modern tools like Docker, and Docker Compose in a Linux environment to scale your application. Yeah, excited? Let's go!

## **Description**

Who this book is for; If you are a beginner like how I started Software Development, then this book will teach you the requirements needed to succeed in the Information Technology Career space. If you have experience with one or two programming languages and know the terms used in Software Development this book will have an impact on your skill.

The Author got tired of the jargon used in Information Technology and hoped for a simpler way to explain the basic truth of what is viewed as a complex area of study.

This book does exactly that, it teaches you the basic concept of Programming your Social Media Web Application and provides insights into the more complex topics such as Security and Scalability. The Author wrote this book to reflect on what had been accumulated over the years of

Software Development to bring important topics to concentrate on for optimal results.

This book does not promise you results overnight, it takes years to achieve desired results, however, this book helps you to jump-start your career to another level.

Through curiosity and research, the author shared a grim amount of tips and tricks for making a Secure Application. Inside the book, you will find illustrations with pictures to convey the best message to what is articulated.

This a Test/ Testing

## **Preface**

### **Understanding ASP.NET Core:**

ASP.NET Core is an open-source web framework developed by Microsoft that allows you to build web applications, including social media platforms, using C# or any other .NET programming language. It provides a robust and scalable environment for building modern web applications that can run on Windows, macOS, or Linux. In the coming chapters, we will explore the basics of ASP.NET Core, including its architecture, MVC pattern, and key concepts such as middleware and dependency injection

### **Setting Up Your Development Environment:**

Before you start building your social media platform, you will need to set up your development environment. The coming chapters will guide you through the process of installing ASP.NET Core SDK, configuring your development environment, and creating a new ASP.NET Core project using Visual Studio or Visual Studio Code. We will also cover essential tools and libraries that you might

need during development, such as Entity Framework Core for database access and Bootstrap for front-end design.

## **Designing Your Social Media Platform:**

The design of your social media platform is crucial in attracting and retaining users. In the coming chapter, we will discuss the key features and functionalities of a typical social media platform, such as user registration and authentication, profile creation and management, friend requests and messaging, a news feed, and post creation and interaction. We will also explore best practices for designing a user-friendly and visually appealing interface using Razor Pages or Angular, a popular front-end framework.

## **Building the Backend Functionality:**

The backend of your social media platform is responsible for processing user requests, managing data, and handling business logic. In the coming chapter, we will dive into the implementation of the backend functionality using ASP.NET Core. We will cover topics such as creating RESTful APIs for handling user actions, implementing authentication and authorization using ASP.NET Core Identity, integrating with third-party authentication providers, managing user profiles and relationships using Entity Framework Core, and implementing real-time notifications using SignalR.

## **Implementing Frontend Features:**

The front end of your social media platform is what users see and interact with. In the chapter to come, we will discuss how to implement frontend features using Razor Pages or Angular. We will cover topics such as user registration and login pages, profile creation and management pages, friend requests and messaging pages, news feed and post creation pages, and user settings and preferences

pages. We will also explore how to consume RESTful APIs from the backend to fetch and update data, and how to use client-side libraries and frameworks for enhancing the user experience, such as Bootstrap for responsive design and Toastr for notifications.

## **Securing Your Social Media Platform:**

Security is a critical aspect of any web application, including social media platforms. In the coming chapter, we will explore how to secure your social media platform using best practices in ASP.NET Core. We will cover topics such as protecting against cross-site scripting (XSS) and cross-site request forgery (CSRF) attacks, securing user passwords using ASP.NET Core Identity, implementing role-based authorization, and securing APIs using token-based authentication with JSON Web Tokens (JWT). We will also discuss other security considerations, such as input validation, data encryption, and logging.

## **Testing and Deployment:**

Testing and deployment are essential steps in the development process of any web application. We will look at how to do exactly that.

---

**DISCLAIMER:**

*The information provided in this book is for general informational purposes only and does not constitute professional advice or establish any professional-client relationship. The author and publisher of this book are not responsible for any errors or omissions, or for any actions taken based on the information provided in this book.*

*The content of this book is based on the author's knowledge and experience in the field of software development, and it may not necessarily reflect the most current industry standards, best practices, or technological advancements. The field of software development is constantly evolving, and readers are encouraged to seek updated information and consult with qualified professionals before making any decisions or taking any actions based on the content of this book.*

*The author and publisher of this book make no warranties, express or implied, regarding the accuracy, completeness, reliability, or suitability of the information provided, and they disclaim any liability for any loss or damages, including but not limited to, direct, indirect, incidental, special, or consequential damages, arising out of or in connection with the use of this book or the information contained herein.*

*All trademarks, service marks, product names, and company names mentioned in this book are the property of their respective owners and are used for identification purposes only. The use of such names does not imply endorsement, sponsorship, or affiliation with this book, its author, or its publisher.*

*Readers are advised to use their own discretion and judgment when applying the information provided in this book to their specific situations or projects and to always comply with applicable laws, regulations, and industry standards.*

*By using this book, readers acknowledge and accept the above disclaimer and agree to use the information provided at their own risk.*

*Please note that this is a generic disclaimer and may need to be customized to fit the specific context and requirements of your Software Development book. It's always recommended to seek legal advice from a qualified professional when drafting legal disclaimers.*



---

# Chapter 1: Computer Programming

## 2. What is Computer Programming

*Category: Computer Programming*

Computer programming, also known as coding or software development, is the process of creating software applications or computer programs by writing instructions that are executed by a computer.

These instructions, written in a programming language, enable a computer to perform specific tasks or solve specific problems.

Computer programming involves designing, coding, testing, and debugging software applications. It requires a deep understanding of programming concepts, algorithms, data structures, and problem-solving skills.

Programming can be done in various programming languages, such as C++, Java, Python, C#, JavaScript, and many others.

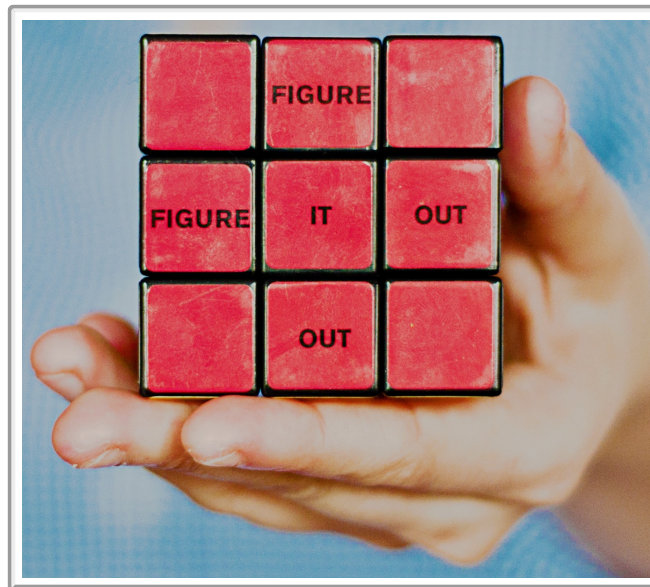
The process of computer programming typically involves the following steps:



1. **Problem Identification:** Identifying the problem or task that needs to be solved by the software application. This involves understanding the requirements and goals of the application and determining what needs to be achieved.
2. **Design:** Planning and designing the structure, flow, and functionality of the software application. This includes defining the user interface, data



structures, algorithms, and overall system architecture.

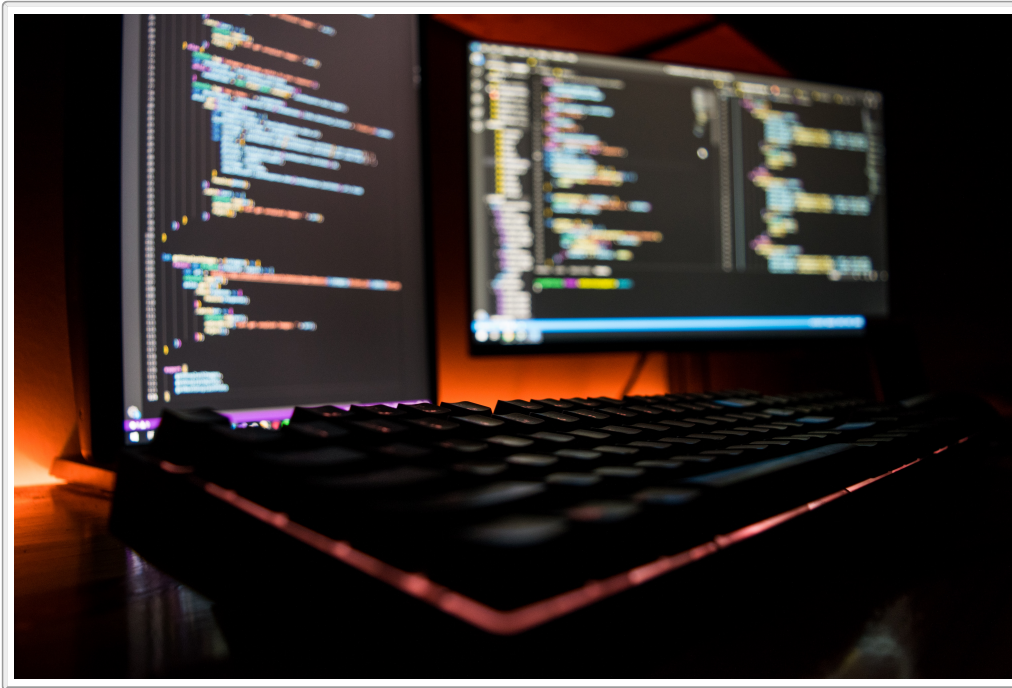


3. **Coding:** Writing the actual code in a programming language based on the design. This involves using the syntax and rules of the programming language to create the instructions that the computer can understand and execute.



4. **Testing:** Testing the software application to identify and fix any errors, bugs, or issues. This involves running various test cases and scenarios to ensure that the software works as expected and meets the requirements.

5. **Debugging:** Identifying and fixing any errors or issues that occur during testing or when the software is running. This involves analyzing the code and identifying the root cause of the problem, and then making the necessary changes to resolve it.



6. **Documentation:** Documenting the code, design, and other aspects of the software application for reference, maintenance, and future development.
7. **Deployment:** Preparing the software application for release and deploying it to the target environment, such as a server or a mobile device, so that users can start using it.

In a very simple form, computer programming is basically writing computer instructions in form of code, these computer instructions or code is then compiled into an application that runs as a program on the computer.

Computer instructions can be written/coded using many different languages called "Programming Languages". There are many Programming Languages that can be used to write a program, to name a few:

- Java
- C-Sharp or C#

- Python
- C
- C++
- JavaScript
- HTML/HTML5

The above-named Programming Languages can be used differently depending on the task needed to be completed. For example, many Software Developers or Computer Programmers prefer to use C-Sharp or C# as a Server-side Programming Language.

We will explain the three stacks of Software Development in the coming chapters. Every programming language has that one thing that a Computer Programmer likes, for example, we might conclude that the reason why most Software Developers like programming in C# is because of its rich intelligence or friendly programming environment.

This goes with many languages too, other Software Developers would prefer to use Java because of its Cross-Platform nature. As mentioned before, every Programming Language has its pros and cons.

Computer programming is a constantly evolving field, with new programming languages, frameworks, and tools being developed regularly.

It requires continuous learning and improvement to keep up with the latest trends and technologies. Programmers work in various industries, including software development, web development, mobile app development, game development, and many others, to create a wide range of applications that are used in our daily lives.

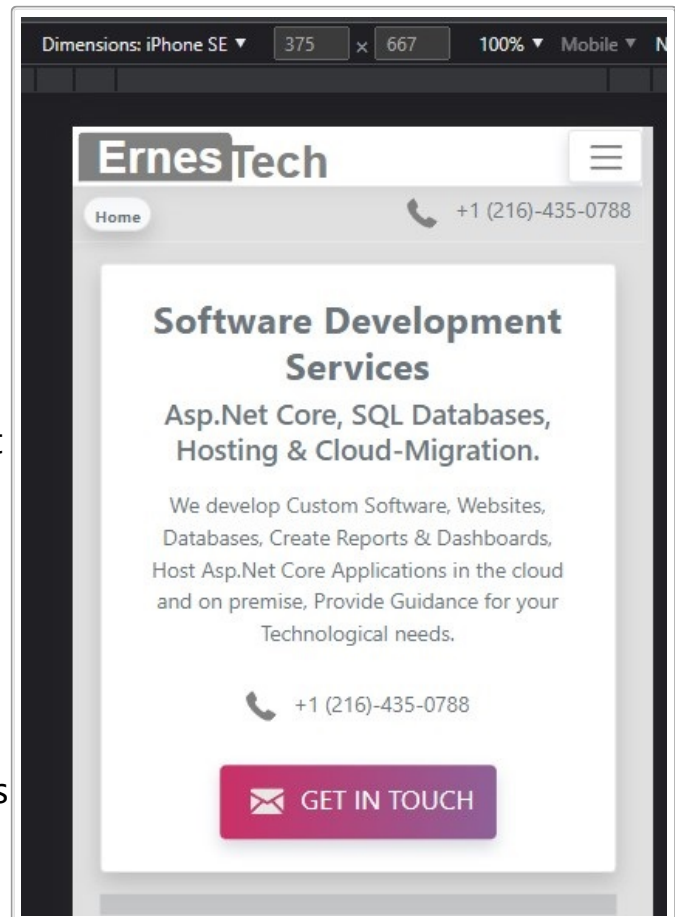
### 3. How to Setup Front-End Development

*Category: Computer Programming*

In this book, we will be using HTML, JavaScript, and CSS3 to program the front end. The front end, also known as the browser side, is what you see in your web browser when you visit a website.

It's important to put the same amount of effort into the front end as it is the layer of the web application that users interact with, and it should convey the right message to the end user.

If a user visits a website with an unfriendly user interface (UI), they may leave, which can be costly to any business. As a Software Developer, prioritizing user experience should always be a priority, as users are what drive the usability.



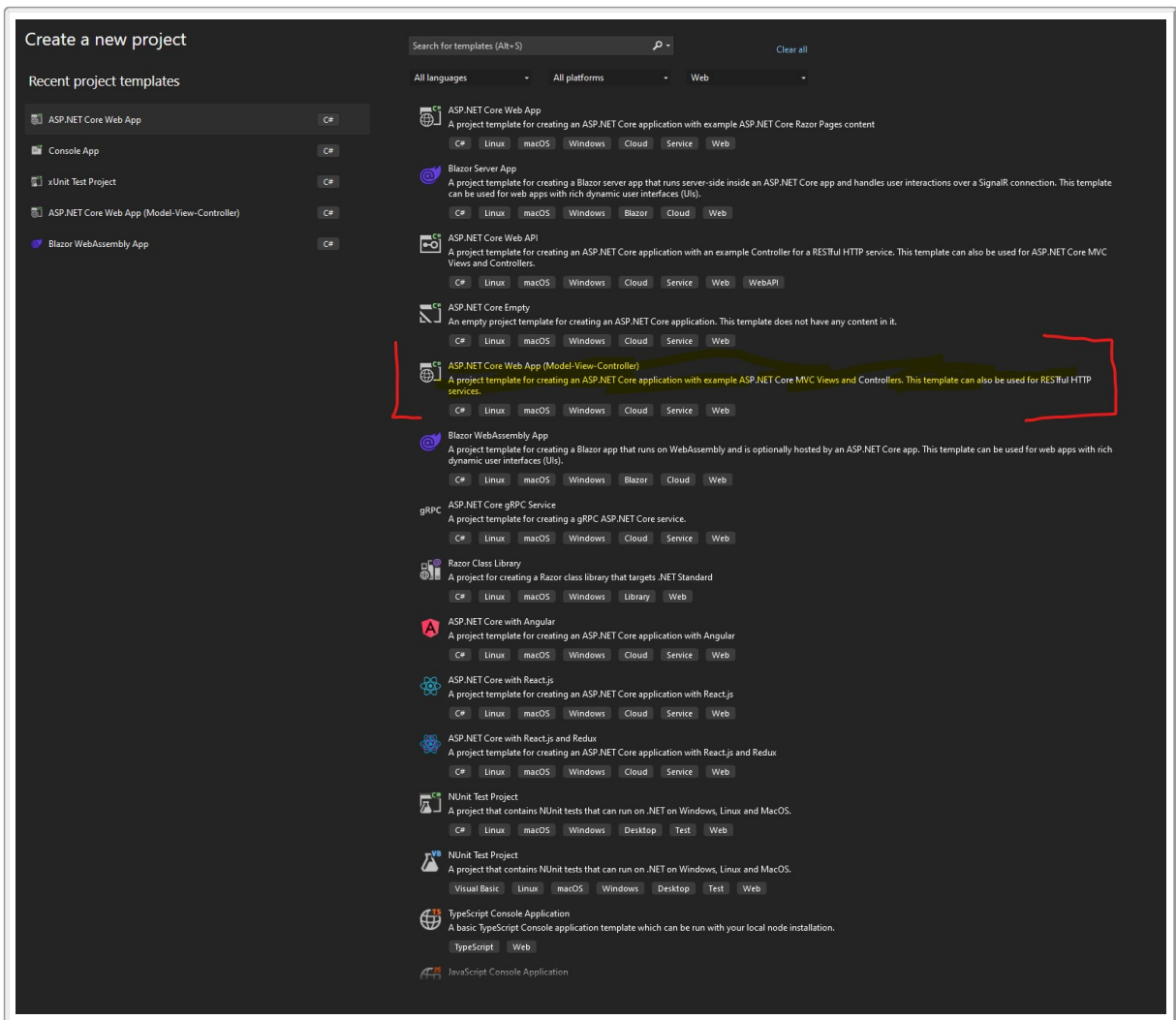
Here's a step-by-step outline for creating the front end of a social media application using ASP.NET Core:

#### **Step 1: Set up ASP.NET Core Project**

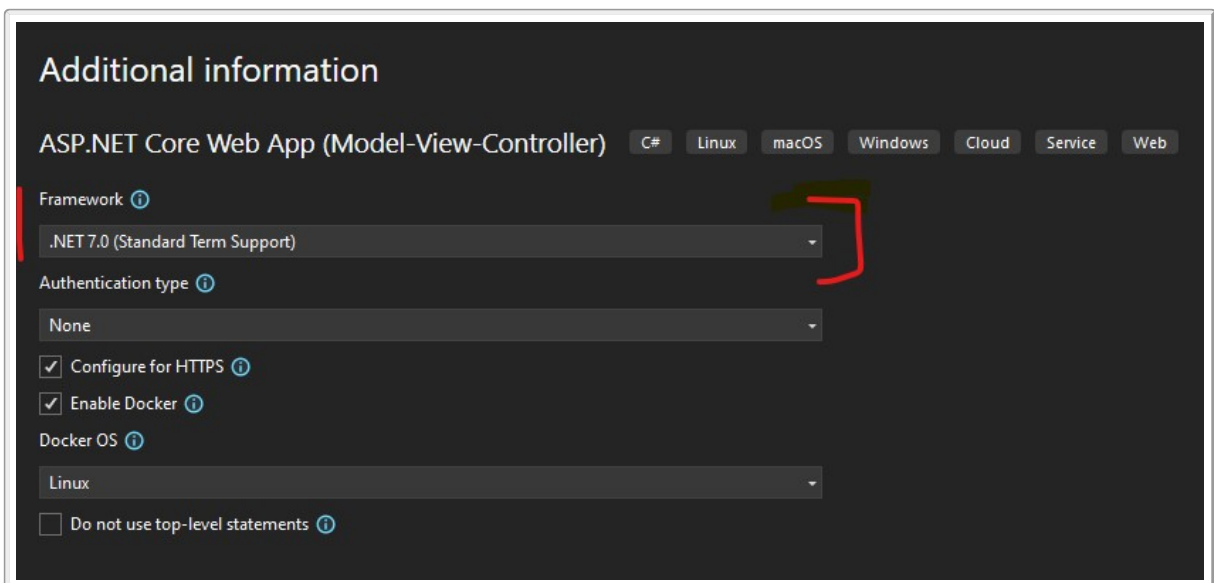
1. Install the latest version of Visual Studio and ASP.NET Core SDK on your machine.

[Link]: <https://visualstudio.microsoft.com/downloads/>

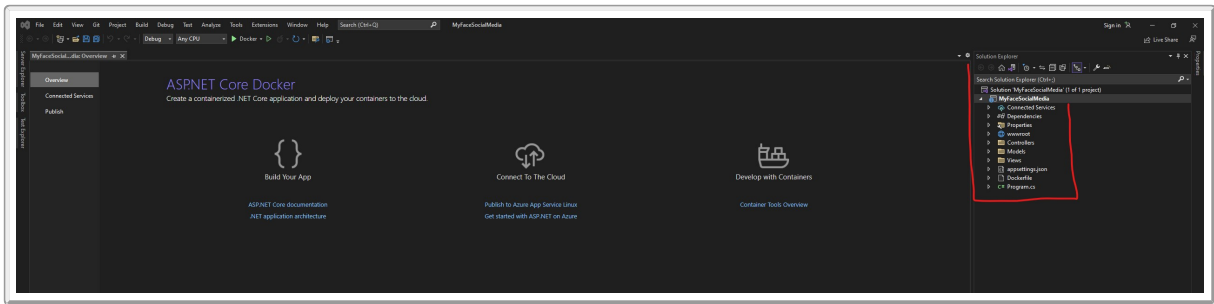
2. Create a new ASP.NET Core project using the ASP.NET Core Web Application template.



3. Choose MVC as the project template and select .Net 7.0 (Standard Term Support).



- You should now have a project that looks like below:



## Step 2: Design the User Interface

4. Define the overall layout and design of your social media application, including the home page, login/register pages, user profile page, timeline/feed page, and other relevant pages.

5. Create HTML views using Razor syntax, which is the default view engine in ASP.NET Core, to generate dynamic HTML content. In this section, we can start by modifying the `_Layout.cshtml` file. This file defines how the overall site will look like



At this stage in Development, you will have a default `_Layout.cshtml` file, we need to modify the file to suit our design.

Here is the default file: