SLA
EASY WAY TO IT JOB

Dot Net Full Stack Developer Tutorial

Published On: July 19, 2024

Dot Net Full Stack Developer Tutorial

In the software development sector, there is a constant need for DotNet developers. This Dot Net Full Stack Developer tutorial will teach you how to create web applications independently, starting from scratch.

DOTNET Full Stack Tutorial PDF

Introduction to Dot Net Full Stack

A software engineer using DotNet technology to create full-stack web applications is known as a ".NET Full Stack Developer." Their extensive knowledge spans several aspects of web development, including middleware, database coding, and UI/UX.

This Dot Net Full Stack Developer tutorial covers the following:

- Front-end technologies: HTML, CSS, Bootstrap, and JavaScript
- Layout design using C#.Net, WCF, WPF, ASP.Net, ADO.Net, and SQL Server.
- Understanding the MVC Framework for Dot Net Development
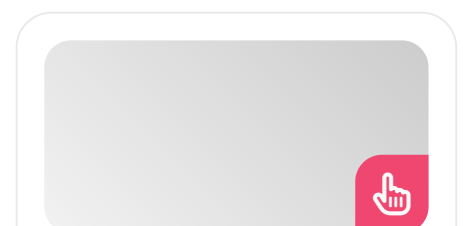
Front End Technologies in Dot Net Full Stack Development

Creating and managing applications constructed with the .NET framework is the responsibility of a full stack .NET developer.

- They should be well-versed in the front-end and back-end aspects of application development.
- They need to be proficient in HTML, CSS, and JavaScript front-end design and implementation of user interfaces.
- Their ability to work with databases, online services, and server-side resources must extend to the back end.

HTML, CSS, Bootstrap, and JavaScript

Working as a .NET full stack developer requires a strong foundation in front-end programming concepts. Because they enable the creation of aesthetically stunning user interfaces, HTML, CSS, and JavaScript are the fundamental technologies of web development.

HTML stands for Hypertext Markup Language, and it establishes the foundation for creating web documents by defining the format and content of web pages.

CSS (Cascading Style Sheets): CSS controls how web pages appear visually, enabling developers to add styles, modify elements, and guarantee a uniform look across platforms.

JavaScript: JavaScript is a powerful programming language that adds interactivity to websites by enabling additional features such as event handling, dynamic behavior, and DOM manipulation.

Programming languages such as JavaScript are widely used in front-end, back-end, and

application development. A.NET full stack developer ought to be proficient in the use of JavaScript in the creation of online applications.

Bootstrap: The name Bootstrap refers to an open-source, free front-end programming framework used to create webpages and web apps. Bootstrap allows the responsive construction of mobile-first websites with its set of template design languages.

Download DOTNET FullStack Syllabus PDF

Layout design using C#.Net, WCF, WPF, ASP.Net, ADO.Net, and SQL Server.

The placement of visual components on a grid to express a certain idea is referred to as layout. No matter how good a design looks, it is useless if the layout is hard to understand.

A layout design, which can also enhance performance by ensuring a fluid message flow, can specify the overall appearance and the connections between graphic elements.

- Layout Design
    - Text
    - Images
    - Lines
    - Shapes
    - White Space
- Hierarchy
- Balance
- Alignment
- Proximity

Importance of Layout Design

The following are a few of the most evident advantages of having a fantastic layout design:

- Make a lovely harmony with the fonts, colors, backgrounds, and other elements.
- Facilitating the material flow will make it easier

for consumers to comprehend and assimilate brand information.
- Highlight the important details in a lengthy piece to keep readers interested.
- Provide factual information without interfering with the section.
- Organizing communication publications better will draw in more readers.

C#.Net

C# may be used to construct a wide range of programs and applications, including games, cloud-based services, mobile and desktop apps, websites, enterprise software, and cloud-based apps. C# is most typically employed in three areas:

Windows Applications

Developing apps and programs that are tailored to the Microsoft platform's architecture is the finest use case for C# because these applications depend on the Windows Dot Net framework to perform at their best.

Since Microsoft developed C# for Microsoft, it makes sense that the majority of apps developed with it are Windows desktop programs.

C# for Games

One of the greatest programming languages for games may be C#. Popular games like Rimworld using the Unity Game Engine are made a lot with this language.

- Unity is the most widely used game engine out there. Over one-third of the best and most played games in the business were created using Unity.
- Because of cross-platform technologies like Xamarin, C# can be utilized on almost any modern mobile device or console and integrates flawlessly with the Unity engine.

C# for website development

C# is frequently used to create open-source

software and polished, dynamic websites on the.NET framework.

You can still utilize C# to develop a completely functional website with Microsoft architecture.

Because of its object-oriented architecture, this language is well-liked for building incredibly efficient, easily scalable, and stable websites.

WCF: Windows Communication Foundation

Application developers utilize the WCF architecture to develop service-oriented apps. With WCF, data can be sent asynchronously between service endpoints in the form of messages.

A service endpoint might be part of an IIS-hosted continuously accessible service or one that is hosted by an application.

Benefits of WCF

- A safe platform for handling financial transactions.
- A service, such as a traffic report or other monitoring service, that provides others with up-to-date information.
- A real-time data sharing or communication tool for two people via chat.
- A dashboard application that gathers data by polling one or more services and displays it logically.
- Exposing as a WCF service a workflow that was built with Windows Workflow Foundation.

WPF: Windows Presentation Foundation

A free and open-source user interface framework for desktop Windows apps is called WPF. WPF apps are typically created with C# and XAML and are built on the.NET framework.

- WPF is a framework for creating sophisticated Windows client applications.
- It offers a standardized programming paradigm for creating graphical user

interfaces for desktop applications.
- WPF separates UI design from logic by using XAML (eXtensible Application Markup Language) to specify the user interface's layout and appearance.
- WPF programs can construct interactive and aesthetically pleasing user interfaces by utilizing advanced features like data binding, styling, templating, animation, and multimedia support.
- WPF is commonly utilized in the development of stand-alone desktop programs that execute on Windows and fully utilize the Windows operating system.

ASP.Net

ASP.NET is a free online framework that uses HTML, CSS, and JavaScript to create amazing websites and web applications. Additionally, real-time technologies like Web sockets and Web APIs can be created. One substitute for ASP.NET is ASP.NET Core.

ASP.NET enhances the.NET platform in the following ways:

> **Basic C# or F# framework for Web Applications**
>
> *<html>*
>
> *<body>*
>
> *<h1>Hello Web Pages</h1>*
>
> *<p>The time is @DateTime.Now</p>*
>
> *</body>*
>
> *</html>*

A simple method for combining HTML, CSS, and server code is through web pages:

- Simple to use, comprehend, and learn
- It employs the Single Page Application (SPA) application architecture.
- Comparable to the scripting languages PHP and Classic ASP, VB (Visual Basic), and C# (C sharp)

Razor is a web page templating vocabulary used in C# to create dynamic web pages.

> A straightforward markup syntax called Razor is used to incorporate server code (C# or VB) into ASP.NET web pages.
>
> *<!DOCTYPE html>*
>
> *<html lang="en">*
>
> *<head>*
>
>   *<meta charset="utf-8" />*
>
>   *<title>Web Pages Demo</title>*
>
> *</head>*
>
> *<body>*
>
>   *<h1>Hello Web Pages</h1>*
>
>   *<p>The time is @DateTime.Now</p>*
>
> *</body>*
>
> *</html>*

Razor Syntax for C#

- Code blocks in C# are delimited with @{… }.
- Variables and functions that are inline expressions begin with @
- A semicolon is used to conclude code statements.
- The datatype (int, string, etc.) or the var keyword is used to declare variables.
- Quotation marks are used to surround strings.
- Case matters in C# code
- The file extension for C# files is '.cshtml.'

A database, libraries, and template pages are all part of this authentication system, which can handle logins and support multi-factor authentication as well as external authentication through Google, X, and other services.

Extensions for editors that offer code completion, syntax highlighting, and other features tailored to web page development.

ADO.Net

Consistent access to data sources like OLE DB and

ODBC, as well as data sources like SQL Server and XML, is made possible via ADO.NET.

Consumer data-sharing apps can use ADO.NET to connect to several data sources and retrieve, control, and update the data they hold.

ADO.Net Components

The DataSet and .NET Framework data providers are the two primary ADO.NET components for data access and manipulation.

- The components known as the.NET Framework Data Providers were created specifically to manipulate data and provide quick, read-only, forward-only access to it.
- Data access from any data source is explicitly intended for the ADO.NET DataSet.

SQL Server for Dot Net Full Stack Development

A strong database management solution called SQL Server is frequently used with .NET applications. To properly query and handle data, a full stack.NET developer has to have a solid working knowledge of SQL Server.

Features of SQL Server

Here are the features and functionalities of Microsoft SQL Server:

- Database Management and Data Storage
  - Creating and managing databases is easy.
  - Wide range of data types and storage options.
- Security and User Management
  - Implementing security measures.
  - Managing user access and permissions.

<DOTNET Full Stack Salary>

Understanding the MVC Framework for Dot Net Development

MVC features are combined to create a dot net framework. MVC is a design pattern that helps accomplish the separation of responsibilities in the design of data, application logic, and user interfaces.

Applications are divided into three component groups using MVC: Model, View, and Controller.

Benefits of MVC for ASP.NET

Flexibility: HTML needs to be extremely flexible to manage the evolution of user interfaces and usability in various business domains. The front-end components need to be changeable at all times.

Test-Driven Development: Testing is a key component of ASP.Net MVC development. By simplifying testing, it guarantees the project's development will be of the highest caliber.

Easily integrates with other technologies: It is possible to easily combine technologies like jQuery, JSON, and AngularJS with ASP.Net MVC and provide mobile functionality.

Example for Models and Data

Model classes can be created simply, and they can be bound to your database. Validation rules are defined declaratively and applied to both the client and the server using C# attributes.

```csharp
public class Person

{

  public int PersonId { get; set; }

  [Required]

  [MinLength(2)]

  public string Name { get; set; }

  [Phone]

  public string PhoneNumber { get; set; }

  [EmailAddress]

  public string Email { get; set; }
```

```
    }
```

Example for Controllers

Requests are simply routed to controller actions, which are implemented as standard C# methods. Method arguments automatically bind data from the request body, query string, and request path.

```
public class PeopleController : Controller

{

  private readonly AddressBookContext _context;

  public PeopleController(AddressBookContext context)

  {

    _context = context;

  }

  // GET: /people

  public async Task Index()

  {

    return View(await _context.People.ToListAsync());

  }

  // GET: /people/details/5

  public async Task Details(int id)

  {

    var person = await _context.People.Find(id);

    if (person == null)

    {

      return NotFound();

    }

    return View(person);

  }

}
```
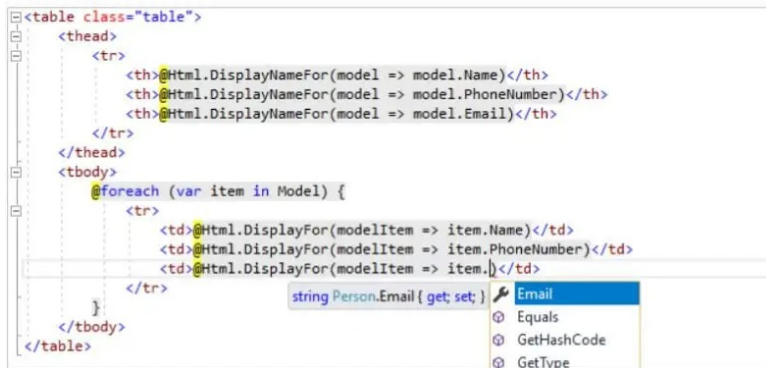
Views with Razor

Rendering HTML text based on your view is made easy, clear, and lightweight with the Razor syntax.

You may create completely HTML5-compatible web pages with Razor by rendering a page in C#.

Dot Net Full Stack Developer Tutorial

Conclusion

We hope this Dot Net Full Stack Developer tutorial is helpful for you to get started on your learning journey. Gain expertise with our **Dot Net Full Stack Developer training in Chennai.**

Share on your Social Media

Softlogic Academy

Navigation

About Us

Blog Posts

Careers

Contact

Placement Training

Softlogic Systems

KK Nagar [Corporate Office]

No.10, PT Rajan Salai, K.K. Nagar, Chennai
– 600 078.
Landmark: Karnataka Bank Building
Phone: +91 86818 84318
Email: enquiry@softlogicsys.in
Map: Google Maps Link

OMR

No. E1-A10, RTS Food Street
92, Rajiv Gandhi Salai (OMR),
Navalur, Chennai - 600 130.
Landmark: Adj. to AGS Cinemas
Phone: +91 89256 88858
Email: info@softlogicsys.in
Map: Google Maps Link

Courses

- Python
- Software Testing
- Full Stack Developer
- Java
- Power BI
- Clinical SAS
- Data Science
- Embedded
- Cloud Computing
- Hardware and Networking
- VBA Macros
- Mobile App Development
- DevOps

- Corporate Training
- Hire With Us
- Job Seekers
- SLA's Recently Placed Students
- Reviews
- Sitemap

Important Links

- Disclaimer
- Privacy Policy
- Terms and Conditions

Social Media Links

Review Sources

- Google
- Trustpilot
- Glassdoor
- Mouthshut
- Sulekha
- Justdial
- Ambitionbox
- Indeed
- Software Suggest
- Sitejabber