

This five-day course provides a comprehensive and practical hands-on introduction to developing Web applications using ASP.NET 4.0 and C#. It includes an introduction to ASP.NET MVC, a new Web programming framework that incorporates use of the Model-View-Controller (MVC) pattern. It also includes coverage of using ASP.NET AJAX to build rich client applications. The fundamentals of Web applications are reviewed, and a testbed is established for ASP.NET and Internet Information Services. The architecture of ASP.NET is outlined, including the role of compilation, the Page class and code-behind. Web Forms are introduced, including server controls, view state, life cycle, and the event model. Request/response HTTP programming using ASP.NET is covered as are ASP.NET Web applications, caching in ASP.NET, and the fundamentals of configuration and security. Data access is covered in some detail in two chapters, including an introduction to ADO.NET, Language Integrated Query (LINQ), and the powerful data access controls provided by ASP.NET 4.0. The final section of the course introduces rich client-side development with ASP.NET AJAX and use of the ASP.NET MVC framework.

Course Objectives:

- Gain a thorough understanding of the philosophy and architecture of Web applications using ASP.NET.
- Acquire a working knowledge of Web application development using Web Forms and Visual Studio 2010.
- Optimize an ASP.NET Web application using configuration, security, and caching.
- Access databases using ADO.NET and LINQ.
- Use newer features in ASP.NET.
- Implement rich client applications using ASP.NET AJAX.
- Create Web applications using the Model-View-Controller design pattern.

Audience: Experienced application developers or architects.

Prerequisites: A working knowledge of C# and .NET Framework. Knowledge of JavaScript is recommended.

Number of Days: 5 days

<p>1. Introduction to ASP.NET Web Application Fundamentals Benefits of ASP.NET An Echo Program ASP.NET Features Compiled Code Server Controls Browser Independence Separation of Code and Content State Management</p>	<p>Web Forms Architecture HelloCodebehind.aspx HelloCodebehind.aspx.cs Page Class Code-Behind Inheritance Model Web Forms Page Life Cycle View State Enabling View State for Controls Web Forms Event Model Page Processing Page Events</p>
<p>2. Web Forms Architecture</p>	

- Page Properties
- Page Directive
- Tracing
- 3. ASP.NET and HTTP**
- Classical Web Programming
- Active Server Pages Object Model
- Request and Response Objects
- Request/Response in ASP.NET
- HttpRequest Class
- Properties of HttpRequest
- Using HttpRequest Class
- HTTP Collections
- HttpResponse Class
- Redirect
- HttpUtility
- Echo.aspx
- EchoBack.aspx
- GET and POST Compared
- QueryString and Forms Collections
- 4. Web Applications Using Visual Studio**
- Using Visual Studio
- Visual Web Developer
- Starter Web Site
- ASP.NET Empty Web Site
- Adding a Web Form
- Using Components in ASP.NET
- Compilation Error
- Shadow Copying
- Temporary Copy of the Component
- ASP.NET Applications
- Global.asax
- Web Application Life Cycle
- Code in Global.asax
- Log Class
- Adding Global.asax File
- ListBox
- Data Binding
- Items Collection
- XHTML
- XHTML in Visual Studio
- Absolute Positioning
- 5. State Management and Web Applications**
- Session and Application State
- Session Object
- Page_Load
- Session Variable Issues
- Session State and Cookies
- Session State Timeout
- Session State Store
- Application State
- Implementing Application State
- Global.asax
- Users.aspx.cs
- Multithreading Issues
- Bouncing the Web Server
- Cookies
- Cookies and ASP.NET
- HttpCookie Properties
- Acme Travel Agency Case Study
- State Management Techniques
- 6. Server Controls**
- Server Controls in ASP.NET
- HTML Server Controls
- Using HTML Server Controls
- HTML vs. Web Forms Server Control
- Code for Login
- HTML Controls in Visual Studio
- Using HTML Controls
- Web Controls
- Validation Controls
- Required Field Validation
- Regular Expression Validation
- Rich Controls
- Copying a Web Site
- User Controls
- Using a User Control
- Copyright.ascx
- Copyright.ascx.cs
- 7. Caching in ASP.NET**
- What Is Caching?
- Need for Caching (Why Cache?)
- Data to be Cached – Time Frame
- ASP vs. ASP.NET Response Model
- Caching in ASP.NET
- Three Types of Caching in ASP.NET
- Output Caching
- @ OutputCache Directive
- @ OutputCache – Attributes in Detail
- VaryByParam in Detail
- HttpCachePolicy Class
- Page Fragment Caching
- Common Mistakes in Using Fragment Caching
- Data Caching or Application Caching
- Add an Item to the Cache Object

- Insert and Add Methods
- Expiration
- Problems in Caching
- 8. ASP.NET Configuration and Security**
- One-minute Introduction to XML!
- ASP.NET Configuration - Overview
- Multi-level Configuration
- Configuration Hierarchy
- Web.Config File Structure
- Web.Config Sections
- Application Settings
- ASP.NET Security – Overview
- Role-Based Security and CAS
- Types and Steps
- Steps in Enabling Role-Based Security
- Three Ways to Authenticate
- Forms Authentication – Default.aspx
- Forms Authentication – Web.Config
- Features of Forms Authentication
- Forms Authentication Classes
- Customizing Forms Authentication
- Authentication Source
- Forms Authentication – Analysis
- Windows Authentication
- Windows Authentication – Analysis
- Passport Authentication
- Passport Authentication – Analysis
- Authorization
- 9. Debugging, Diagnostics and Error Handling**
- ASP.NET Diagnostics
- Debugging Using Visual Studio
- Debugging Calculator
- Application-Level Tracing
- Tracing Calculator
- Using the Page Cache
- Preparing to Debug
- Trace Messages
- Tracing the Calculator Page
- Conditional Tracing
- Trace Category
- Trace Warning
- Exceptions in Trace
- Errors in ASP.NET
- Uncaught Exception
- Custom Error Pages
- 10. More Server Controls**
- ASP.NET Control Improvements
- Newer Controls in ASP.NET
- Master Pages
- Creating Content Pages
- TreeView Control
- Master Page Application
- 11. ADO.NET and LINQ**
- ADO.NET
- ADO.NET Architecture
- .NET Data Providers
- .NET Namespaces
- Connected Data Access
- AcmePub Database
- Creating a Connection
- Using Server Explorer
- Performing Queries
- ADO.NET with ASP.NET
- Web Client Isolation
- Web Client Database Code
- Using Commands
- Creating a Command Object
- Using a Data Reader
- Use of Session State
- Generic Collections
- Executing Commands
- Parameterized Queries
- DataSet
- DataSet Architecture
- Why DataSet?
- DataSet Components
- DataAdapter
- Data Access Class
- Retrieving the Data
- Filling a DataSet
- Accessing a DataSet
- Using a Standalone DataTable
- Adding a New Row
- Searching and Updating a Row
- Deleting a Row
- Row Versions
- Row State
- Iterating Through DataRows
- Command Builders
- Updating a Database
- Language Integrated Query (LINQ)
- Bridging Objects and Data
- Object Relational Designer
- IntelliSense
- Basic LINQ Query Operators

- Obtaining a Data Source
- Filtering
- Ordering
- Aggregation
- Obtaining Lists and Arrays
- Deferred Execution
- Modifying a Data Source
- Performing Inserts via LINQ to SQL
- Performing Deletes via LINQ to SQL
- Performing Updates via LINQ to SQL
- 12. Data Access in ASP.NET 4.0**
 - Data Access in ASP.NET
 - SQL Generation Options
 - Enable Edit and Delete
 - Editing Records
 - GridView Control
 - DetailsView Control
 - Storing the Connection String
 - Protecting the Configuration String
 - FormView Control
 - Master/Detail Web Pages
 - Data Binding
 - Template Editing
 - Using XML Data
 - Multiple-Tier Data Access
 - Object Data Source
 - Data Access in ASP.NET 4.0
 - Using the Entity Data Model
 - EntityDataSource Control
 - ListView Edit
 - DataPager Control
 - LinqDataSource Control
 - Configuring LinqDataSource
 - QueryExtender Control
- 13. Personalization and Security**
 - Themes
 - Control Skins
 - Applying Themes
 - Security in ASP.NET 4.0
 - ASP.NET Membership
 - Login Controls
 - Web Site Administration Tool
 - Access Rules
 - Profile Properties
 - Using ASP.NET Profile Properties
- 14. ASP.NET AJAX**
 - Desktop Applications
 - Web Applications
- Plug-Ins
- Client-Side Scripting
- Script Code
- JavaScript in ASP.NET
- Dynamic Pages
- Efficient Page Redraws
- AJAX
- Google Maps
- ASP.NET AJAX
- Partial Page Rendering
- UpdatePanel Control
- AJAX Extensions Controls
- AJAX Client Library
- Using the Client Library
- ScriptManager Control
- Client Library Namespaces
- Sys.Debug Tracing
- Document Object Model
- JavaScript for Simple Calculator
- Using the Client Library
- AJAX Control Toolkit
- ACT Controls in Visual Studio
- AjaxControlToolkit.dll
- Registering AjaxControlToolkit.dll
- Extender Controls
- NumericUpDownExtender Control
- 15. ASP.NET MVC**
 - Model-View-Controller (MVC)
 - What is ASP.NET MVC?
 - Advantages of ASP.NET MVC
 - Advantages of Web Forms
 - Visual Studio ASP.NET MVC Project
 - Starter Application
 - Simple App with Controller Only
 - Action Methods and Routing
 - Action Method Return Type
 - Rendering a View
 - Creating a View in Visual Studio
 - The View Web Page
 - Passing Data from Controller to View
 - Embedded Scripts
 - Using a Model
 - Controller Using the Model
 - View Using the Model
 - Strongly Typed View
 - Creating Views in Visual Studio
 - Using Forms
 - HTML Helper Functions

- Displaying Sign In Form
- Handling Form Submission
- Model Binding
- Greet View
- Input Validation
- Nullable Type
- Checking Model Validity
- Validation Summary

16. Appendix A – Learning Resources

17. Appendix B – HTTP Pipeline

- Web Applications
- Customizing the HTTP Pipeline
- Customizing Applications
- Customizing a Welcome Application
- Logger Class
- Custom Handlers
- IHttpHandler Interface
- .ashx Files
- Custom Modules
- Using DemoModule