

Developing ASP.NET Core MVC Web Applications

CODICE	MOC20486
DURATA	5 gg
PREZZO	1.600,00 €
EXAM	

DESCRIZIONE

In questo corso di 5 giorni, gli sviluppatori web professionisti imparano a sviluppare applicazioni MVC ASP.NET Core avanzate utilizzando strumenti e tecnologie .NET Core. Il corso è focalizzato sulle attività di codifica che possono migliorare le prestazioni e la scalabilità dell'applicazione del sito Web. Questo corso prepara lo studente anche per l'esame 70-486.

OBIETTIVI RAGGIUNTI

- Descrivere lo stack di Microsoft Web Technologies e selezionare una tecnologia appropriata da utilizzare per sviluppare un'applicazione specifica.
- Progettare l'architettura e l'implementazione di un'applicazione Web che soddisfi una serie di requisiti funzionali, i requisiti dell'interfaccia utente e tenga conto dei modelli di business.
- Configurare la pipeline di applicazioni Web ASP.NET Core utilizzando il middleware e sfruttare l'inserimento delle dipendenze nell'applicazione MVC.
- Aggiungere controller a un'applicazione MVC per gestire l'interazione dell'utente, aggiornare i modelli, selezionare e restituire le visualizzazioni.
- Sviluppare un'applicazione Web che utilizzi il motore di routing ASP.NET Core per presentare agli utenti URL intuitivi e una gerarchia di navigazione logica.
- Creare viste in un'applicazione MVC che visualizzino e modifichino i dati e interagiscano con modelli e controller.
- Creare modelli MVC e scrivere codice che implementi la logica aziendale all'interno di metodi, proprietà ed eventi del modello.
- Connettere un'applicazione ASP.NET Core a un database utilizzando Entity Framework Core.
- Implementare un aspetto coerente in un'intera applicazione Web MVC.
- Scrivere codice JavaScript che venga eseguito sul lato client e utilizzi la libreria di script jQuery per ottimizzare la reattività di un'applicazione Web MVC.
- Aggiungere pacchetti lato client e configurare Task Runner.
- Eseguire test di unità e usare strumenti di debug su un'applicazione Web in Visual Studio 2017.
- Scrivere un'applicazione MVC che autentichi e autorizzi gli utenti ad accedere ai contenuti in modo

sicuro utilizzando Identity.

- Creare un'applicazione MVC che resista agli attacchi dannosi.
- Utilizzare la memorizzazione nella cache per accelerare le risposte alle richieste degli utenti.
- Utilizzare SignalR per abilitare la comunicazione bidirezionale tra client e server.
- Descrivere cos'è un'API Web e perché gli sviluppatori possono aggiungere un'API Web a un'applicazione.
- Descrivere come impacchettare e distribuire un'applicazione Web ASP.NET Core MVC da un computer di sviluppo a un server Web.

TARGET

Questo corso è destinato a sviluppatori Web professionisti che utilizzano Microsoft Visual Studio in un ambiente di sviluppo individuale o di gruppo, dalle piccole alle grandi dimensioni. I candidati a questo corso sono interessati a sviluppare applicazioni Web avanzate e vogliono gestire l'HTML renderizzato in modo completo. Vogliono creare siti Web che separino l'interfaccia utente, l'accesso ai dati e la logica dell'applicazione.

PREREQUISITI

Prima di partecipare a questo corso, gli studenti devono avere:

- Esperienza di Visual Studio 2017.
- Esperienza di programmazione C# e di concetti come espressioni Lambda, LINQ e tipi anonimi.
- Esperienza dell'uso di .NET Framework.
- Esperienza di HTML, CSS e JavaScript.
- Esperienza di interrogazione e manipolazione dei dati con ADO.NET.
- Conoscenza delle strutture di dati XML e JSON.

CONTENUTI

Modulo 1: Esplorazione di ASP.NET Core MVC

Microsoft ASP.NET Core MVC e le altre tecnologie Web di ASP.NET Core possono aiutare a creare e ospitare applicazioni Web dinamiche, potenti ed estensibili. ASP.NET Core, di cui ASP.NET Core MVC è un framework multiplatforma open source che consente di creare applicazioni Web. È possibile sviluppare ed eseguire applicazioni Web ASP.NET Core su Windows, macOS, Linux o qualsiasi altra piattaforma che lo supporti.

ASP.NET Core MVC supporta un ciclo di sviluppo agile e guidato dai test. Consente inoltre di utilizzare i più recenti standard HTML e framework front-end come Angular, React e altri.

Lezioni

- Panoramica di Microsoft Web Technologies
- Panoramica di ASP.NET 4.x

- Introduzione ad ASP.NET Core MVC

Lab: Analisi di ASP.NET Core MVC

- Analisi di un'applicazione Razor Pages
- Analisi di un'applicazione API Web
- Analisi di un'applicazione MVC

Al termine di questo corso, gli studenti saranno in grado di:

- Comprendere la varietà di tecnologie disponibili nello stack Web di Microsoft.
- Descrivere i diversi modelli di programmazione disponibili per gli sviluppatori in ASP.NET.
- Scegliere tra ASP.NET Core e ASP.NET 4.x.
- Descrivere il ruolo di ASP.NET Core MVC nello stack delle tecnologie Web e come utilizzare ASP.NET Core MVC per creare applicazioni Web.
- Distinguere tra modelli MVC, controller MVC e viste MVC.

Modulo 2: Progettazione di applicazioni Web ASP.NET Core MVC

Microsoft ASP.NET Core MVC è un modello di programmazione che è possibile utilizzare per creare applicazioni Web potenti e complesse. Tuttavia, tutti i progetti di sviluppo complessi, in particolare i grandi progetti, possono essere difficili da comprendere appieno. Senza una completa comprensione degli scopi di un progetto, non è possibile sviluppare una soluzione efficace al problema del cliente. È necessario sapere come identificare una serie di esigenze aziendali e pianificare l'applicazione Web Model-View-Controller (MVC) per soddisfare tali esigenze. Il piano di progetto assicura agli stakeholder di avere compreso i loro requisiti e comunica agli sviluppatori la funzionalità dell'applicazione Web, la sua interfaccia utente, la struttura e l'archiviazione dei dati. Scrivendo un piano di progetto dettagliato e accurato, è possibile garantire che le potenti funzionalità di MVC vengano utilizzate in modo efficace per risolvere i problemi aziendali del cliente.

Lezioni

- Pianificazione in fase di progettazione
- Progettazione di modelli, controller e viste

Lab: Progettazione di applicazioni Web ASP.NET Core MVC

- Classi del modello di pianificazione
- Controllori di pianificazione
- Viste di pianificazione
- Architettura e applicazione MVC Web

Al termine di questo modulo gli studenti saranno in grado di:

- Pianificare l'architettura generale di un'applicazione Web ASP.NET Core MVC e considerare aspetti

come la gestione dello stato.

- Pianificare i modelli, i controller e le viste necessari per implementare un determinato set di requisiti funzionali.

Modulo 3: Configurazione di middleware e servizi in ASP.NET Core

ASP.NET Core è un framework che consente di creare diversi tipi di applicazioni. Questo modulo mostra come sfruttare il framework ASP.NET Core per gestire le richieste e le risposte tramite middleware esistente e personalizzato e come configurare i servizi da utilizzare nel middleware e in altre parti dell'applicazione, come i controller. Un middleware è un segmento di codice che può essere utilizzato come parte della pipeline di richiesta e risposta che consente di gestirli secondo qualsiasi parametro rilevante. Ciò consente potenzialmente di gestire più richieste separate in modo completamente diverso e ricevere risposte separate. I servizi sono classi che espongono funzionalità che è possibile utilizzare successivamente in diverse parti dell'applicazione, senza dover tenere traccia manuale dell'ambito di ogni singola posizione e creare istanze di dipendenze. Ciò viene fatto usando l'inserimento delle dipendenze. L'inserimento delle dipendenze è una tecnica utilizzata da ASP.NET Core che consente di aggiungere dipendenze nel codice senza doversi preoccupare di creare istanze di oggetti, tenerli in memoria o passare dipendenze richieste. Ciò consente all'applicazione di diventare più flessibile e di ridurre potenziali punti di errore ogni volta che si cambia un servizio.

Lezioni

- Configurazione dei middleware
- Configurazione dei servizi

Lab: Configurazione di middleware e servizi in ASP.NET Core

- Lavorare con file statici
- Creazione di middleware personalizzato
- Utilizzo dell'inserimento di dipendenze
- Inserimento di un servizio su un controller

Al termine di questo modulo gli studenti saranno in grado di:

- Utilizzare il middleware esistente per configurare un'applicazione ASP.NET Core.
- Creare il proprio middleware e usarlo per definire comportamenti personalizzati.
- Comprendere i principi di base dell'inserimento di una dipendenza e come viene utilizzato in ASP.NET Core.
- Sapere come creare un servizio personalizzato, configurarne l'ambito e inserirlo sia nel middleware che nei controller MVC ASP.NET Core.

Modulo 4: Sviluppo dei controller

ASP.NET Core MVC è un framework per la creazione di applicazioni Web utilizzando il modello di architettura Model-View-Controller (MVC). Il controller è essenzialmente responsabile dell'elaborazione di una richiesta Web interagendo con il modello e quindi passando i risultati alla vista. Il modello rappresenta il livello aziendale, a volte indicato come dominio, e può includere oggetti dati, logica dell'applicazione e regole aziendali. La vista utilizza i dati che riceve dal controller per produrre l'HTML o altro output che viene inviato al browser. Questo modulo mostra come sviluppare i controller. I controller sono fondamentali per le applicazioni MVC. Comprendere come funzionano i controller è fondamentale per poter creare gli oggetti modello appropriati, manipolarli e passarli alle viste appropriate. Un controller è una classe. Contiene diversi metodi. Questi metodi sono chiamati azioni. Quando un'applicazione MVC riceve una richiesta, identifica il controller e l'azione che devono gestire la richiesta. Determina ciò utilizzando il routing URL (Uniform Resource Locator). Il routing URL è un altro concetto molto importante e necessario per lo sviluppo di applicazioni MVC. Il framework ASP.NET Core MVC include un sistema di routing URL flessibile che consente di definire regole di mapping URL all'interno delle applicazioni. Per massimizzare il riutilizzo del codice nei controller, è importante sapere come scrivere i filtri di azione. È possibile utilizzare i filtri di azione per eseguire il codice prima o dopo ogni azione nell'applicazione Web, per ogni azione in un controller o in altre combinazioni di azioni del controller.

Lezioni

- Scrittura di controller e azioni
- Configurazione dei percorsi
- Scrittura di filtri di azione

Lab: Sviluppo di controller

- Aggiunta di controller e azioni a un'applicazione MVC
- Configurazione dei percorsi utilizzando la tabella di routing
- Configurazione dei percorsi utilizzando gli attributi
- Aggiunta di un filer di azioni

Al termine di questo modulo gli studenti saranno in grado di:

- Aggiungere un controller a un'applicazione Web che risponda alle azioni dell'utente specificate nello sviluppo del progetto.
- Aggiungere route al motore di routing ASP.NET Core e assicurarsi che gli URL siano di facile utilizzo in un'applicazione Web MVC.
- Scrivere codice nei filtri di azione che viene eseguito prima o dopo un'azione del controller.

Modulo 5: Sviluppo delle viste

Le viste sono uno dei tre componenti principali del modello di programmazione Model-View-Controller (MVC). Creando delle viste è possibile definire l'interfaccia utente dell'applicazione Web. Una vista è una combinazione di markup HTML e codice C# che viene eseguita su un server Web. Pertanto, per creare una vista, è necessario sapere come scrivere il markup HTML e il codice C# e utilizzare le varie classi di

supporto integrate in MVC. Occorre anche sapere come creare viste parziali e visualizzare componenti che creino sezioni di HTML che possano essere riutilizzate nella propria applicazione web.

Lezioni

- Creazione di viste con Razor Syntax
- Utilizzo di helper HTML e tag helper
- Riutilizzo del codice nelle viste

Lab: Sviluppo di viste

- Aggiunta di visualizzazioni a un'applicazione MVC
- Aggiunta di una vista parziale
- Aggiunta di un componente vista

Al termine di questo modulo gli studenti saranno in grado di:

- Creare una vista MVC e aggiungere il markup Razor per far visualizzare i dati agli utenti.
- Utilizzare gli helper HTML e i tag helper in una vista. Riutilizzare il markup Razor in più punti di un'applicazione.

Modulo 6: Sviluppo dei modelli

La maggior parte delle applicazioni Web interagisce con vari tipi di dati o di oggetti. Un'applicazione di e-commerce, ad esempio, gestisce prodotti, carrelli della spesa, clienti e ordini. Un'applicazione di social network potrebbe aiutare a gestire utenti, aggiornamenti di stato, commenti, foto e video. Un blog viene utilizzato per gestire post, commenti, categorie e tag. Quando si scrive un'applicazione Web Model-View-Controller (MVC) si crea un modello MVC per modellare i dati per l'applicazione Web. All'interno di questo modello si crea una classe di modello per ciascun tipo di oggetto. La classe del modello descrive le proprietà di ciascun tipo di oggetto e può includere la logica aziendale che corrisponde ai processi aziendali. Pertanto, in un'applicazione MVC il modello è un componente fondamentale. Questo modulo illustra come creare il codice per i modelli.

Lezioni

- Creazione di modelli MVC
- Lavorare con i moduli
- Convalida dell'applicazione MVC

Lab: Sviluppo di modelli

- Aggiunta di un modello
- Lavorare con i moduli
- Aggiungere convalida

Al termine di questo modulo gli studenti saranno in grado di:

- Aggiungere un modello a un'applicazione MVC e scrivere il codice per implementare la logica aziendale.
- Utilizzare, visualizzare e modificare le annotazioni dei dati.
- Convalidare l'input dell'utente con annotazioni di dati.

Modulo 7: Utilizzo di Entity Framework Core in ASP.NET Core

Le applicazioni Web utilizzano spesso le informazioni e di solito richiedono un archivio dati per tali informazioni. Eseguendo il rendering di pagine Web che utilizzano dati da un archivio è possibile creare un'applicazione Web che cambi continuamente in risposta all'input dell'utente, alle azioni amministrative e agli eventi di pubblicazione. L'archivio dati è generalmente un database, ma occasionalmente vengono utilizzati altri tipi di archivi. Nelle applicazioni Model-View-Controller (MVC) è possibile creare un modello che implementi la logica di accesso ai dati e la logica aziendale. In alternativa, è possibile separare la logica aziendale dalla logica di accesso ai dati utilizzando un repository. Un repository è una classe che un controller può chiamare per leggere dati da un archivio e scrivere dati in un archivio. Quando si scrive un'applicazione ASP.NET è possibile utilizzare le tecnologie Entity Framework Core (EF Core) e Language Integrated Query (LINQ) che rendono il codice di accesso ai dati molto veloce da scrivere e semplice da comprendere. Questo modulo mostra come creare un sito Web basato su database in MVC.

Lezioni

- Introduzione a Entity Framework Core
- Lavorare con Entity Framework Core
- Utilizzo di Entity Framework Core per connettersi a Microsoft SQL Server

Lab: Utilizzo di Entity Framework Core in ASP.NET Core

- Aggiunta di Entity Framework Core
- Utilizzo di Entity Framework Core per recuperare e archiviare i dati
- Utilizzo di Entity Framework Core per connettersi a Microsoft SQL Server

Al termine di questo modulo gli studenti saranno in grado di:

- Collegare un'applicazione a un database per accedere e archiviare i dati.
- Spiegare EF Core.
- Lavorare con Entity Framework Core.
- Utilizzare EF Core per connettersi a un database, incluso Microsoft SQL Server.

Modulo 8: Utilizzo di layout, CSS e JavaScript in ASP.NET Core MVC

Durante la creazione di applicazioni Web, è necessario dare all'intera applicazione un aspetto coerente. È necessario includere sezioni di intestazione e piè di pagina coerenti in tutte le viste. Microsoft ASP.NET

Core MVC include funzionalità come stili e layout dei fogli di stile CSS (CSS) che migliorano l'aspetto e l'usabilità dell'applicazione Web. In ASP.NET Core MVC, è possibile creare elementi HTML interattivi utilizzando JavaScript. Occorre sapere come usare JavaScript nella propria applicazione web. Per semplificare l'aggiunta di JavaScript all'applicazione Web, è necessario sapere come utilizzare librerie come jQuery.

Lezioni

- Utilizzo dei layout
- Utilizzo di CSS e JavaScript
- Utilizzo di jQuery

Lab: Utilizzo di layout, CSS e JavaScript in ASP.NET Core

- Applicazione di un layout e collegamenti
- Utilizzo di CSS
- Utilizzo di JavaScript
- Utilizzo di jQuery

Al termine di questo modulo gli studenti saranno in grado di:

- Applicare un layout coerente alle applicazioni ASP.NET Core MVC.
- Aggiungere codice JavaScript alla propria applicazione web.
- Utilizzare la libreria jQuery nell'applicazione Web.

Modulo 9: Sviluppo lato client

Quando si crea un'applicazione è importante sapere come sviluppare l'applicazione sia lato client sia lato server. Questo modulo illustra gli strumenti lato client che permettono di creare applicazioni web complesse su qualsiasi scala. Questi strumenti si basano sugli argomenti trattati nel Modulo 8, "Utilizzo di layout, CSS e JavaScript in ASP.NET Core MVC". Questo modulo mostra come utilizzare il framework Bootstrap per modellare la propria applicazione web. Pertanto spiega come usare Sass e Less, due preprocessori CSS (Cascading Style Sheets) comuni che aggiungono ai CSS una serie di funzionalità come variabili, regole nidificate e funzioni. Queste funzionalità migliorano notevolmente la manutenibilità di applicazioni web complesse. Successivamente, il corso mostra come impostare i runner di attività come Grunt e gulp e come usarli per compilare i file Sass durante la compilazione di Microsoft Visual Studio. Spiega anche come utilizzare il task runner gulp per eseguire il raggruppamento e la minimizzazione di file CSS e JavaScript e come impostare un'attività di controllo per compilare i file Sass mentre si scrive il codice, senza la necessità di ricostruire la soluzione. Infine, tratterà alcuni strumenti di progettazione responsive che consentono di personalizzare la visualizzazione della propria applicazione web in base alle capacità e alle specifiche di un browser Web o di un dispositivo. Lo studente impara a scrivere query multimediali CSS, apprende come utilizzare il sistema a griglia responsive Bootstrap e come applicare il layout del flexbox CSS alle proprie viste.

Lezioni

- Applicazione di stili
- Utilizzo di Task Runners
- Progettazione responsive

Lab: Sviluppo lato client

- Utilizzo di gulp per eseguire attività
- Applicazione di stili tramite Sass
- Utilizzo di Bootstrap

Al termine di questo modulo gli studenti saranno in grado di:

- Utilizzare Bootstrap, Sass e Less in un'applicazione Microsoft ASP.NET Core.
- Utilizzare i task runner in un'applicazione ASP.NET Core.
- Accertarsi che un'applicazione Web venga visualizzata correttamente su dispositivi con schermi di dimensioni diverse.

Modulo 10: Test e risoluzione dei problemi

I sistemi software come le applicazioni Web sono complessi e richiedono l'interazione tra più componenti spesso scritte da sviluppatori diversi. Ipotesi errate, comprensione imprecisa, errori di codifica e molti altri fattori possono creare bug che provocano eccezioni o comportamenti imprevisti. Per migliorare la qualità della propria applicazione web e creare un'esperienza utente soddisfacente occorre identificare i bug di qualsiasi provenienza ed eliminarli. Tradizionalmente, i tester eseguono la maggior parte delle prove alla fine di un progetto di sviluppo. Tuttavia, recentemente è stato ampiamente riconosciuto che i test eseguiti durante l'intero ciclo di vita del progetto migliorano la qualità e assicurano l'assenza di bug nel software di produzione. È necessario comprendere come eseguire i test su piccoli componenti dell'applicazione Web per assicurarsi che funzionino come previsto prima di assemblarli in un'applicazione Web completa. È anche importante sapere come gestire le eccezioni mentre si verificano. Durante l'esecuzione di un'applicazione è possibile che si verifichino eventi imprevisti. È importante gestire correttamente le eccezioni e fornire un buon feedback agli utenti evitando al contempo di far trasparire informazioni sulla struttura dell'applicazione. Infine, utilizzando i log in tutta l'applicazione, è possibile monitorare le attività degli utenti che potrebbero portare a problemi imprevisti e quindi è possibile trovare soluzioni ai bug, che normalmente non si è sicuri su come riprodurli, seguendo i flussi che si sono verificati nell'ambiente di produzione e trovando ulteriori errori.

Lezioni

- Test delle applicazioni MVC
- Implementazione di una strategia di gestione delle eccezioni
- Registrazione delle applicazioni MVC

Lab: Test e risoluzione dei problemi

- Test di un modello
- Test di un controller utilizzando un repository falso
- Implementazione di un repository nel progetto MVC
- Aggiunta della gestione delle eccezioni
- Aggiunta della registrazione

Al termine di questo corso, gli studenti saranno in grado di:

- Eseguire test unitari sui componenti Model – View – Controller (MVC) come classi di modelli e controller e individuare potenziali bug.
- Creare un'applicazione MVC Microsoft ASP.NET Core che gestisca le eccezioni in modo fluido e affidabile.
- Eseguire i provider di registrazione a vantaggio delle proprie applicazioni ed eseguirli utilizzando un'API di registrazione comune.

Modulo 11: Gestire la sicurezza

Poiché le applicazioni Web sono normalmente indirizzate agli utenti che utilizzano solo un browser per far girare l'applicazione è probabile che ci siano molti più utenti rispetto al caso delle applicazioni installate. Tuttavia, la natura aperta di un'applicazione web significa che la sicurezza deve essere sempre presente nella mente dello sviluppatore. Come parte della sicurezza è necessario decidere quali utenti possano eseguire le azioni, il tutto proteggendo gli utenti e l'applicazione rispetto a terze parti dannose con possibile accesso all'applicazione. L'autenticazione è l'atto di utilizzare diversi parametri per assicurarsi che un utente sia chi afferma di essere. Implementando l'autenticazione, è possibile verificare l'identità degli utenti e fornire loro i contenuti appropriati durante l'utilizzo delle applicazioni. L'autorizzazione è il processo in cui un utente già autenticato nell'applicazione può avere accesso ad azioni o risorse specifiche. Utilizzando l'autorizzazione, è possibile impedire agli utenti di accedere a materiale sensibile non previsto per loro o di eseguire azioni che non dovrebbero essere in grado di fare. Infine, a un certo punto della loro vita, le applicazioni potrebbero essere attaccate da utenti malintenzionati. Questi attacchi possono variare nei mezzi e nelle intenzioni, ma il costo dell'assenza di difese può essere elevato. Si potrebbero perdere potenziali utenti interessati, dati preziosi potrebbero essere erroneamente modificati e, nei casi peggiori, l'intera applicazione potrebbe diventare inutilizzabile. Questo modulo esamina le soluzioni ad alcuni degli attacchi più comuni.

Lezioni

- Autenticazione in ASP.NET Core
- Autorizzazione in ASP.NET Core
- Difesa dagli attacchi

Lab: Gestione della sicurezza

- Utilizzo dell'identità
- Aggiunta dell'autorizzazione
- Evitare l'attacco contraffatto con richieste incrociate

Al termine di questo modulo gli studenti saranno in grado di:

- Aggiungere l'autenticazione di base alla propria applicazione.
- Configurare Microsoft ASP.NET Core Identity.
- Aggiungere l'autorizzazione di base alla propria applicazione.
- Utilizzare diversi approcci di autorizzazione.
- Conoscere il funzionamento degli attacchi alla sicurezza e sapere come difendersi meglio.

Modulo 12: Prestazioni e comunicazione

Le moderne applicazioni Web richiedono interazioni complesse con gli utenti. Gli utenti richiedono spesso molti dati in un breve lasso di tempo, aspettandosi anche dati pertinenti non appena si collegano online. Ciò può facilmente causare un notevole carico su un server non preparato, causando anche operazioni inutilmente complesse o ripetute. Fortunatamente, esistono diversi modi per ridurre il carico. La memorizzazione nella cache consente di archiviare richieste ripetute comunemente, evitando la necessità di eseguire ripetutamente la stessa logica. Utilizzando la memorizzazione nella cache è possibile riutilizzare i dati già caricati e presentarli all'utente. Ciò fornisce all'utente un tempo di risposta rapido e riduce le risorse di sistema utilizzate per eseguire la logica dell'azione. Nel frattempo lo stato consente di ottenere la coerenza tra le diverse richieste. Utilizzando varie forme di gestione dello stato è possibile trasformare un'esperienza Web normalmente senza stato in un'esperienza personalizzata per i singoli clienti, con utenti diversi che godono di un'esperienza separata e pertinente nella stessa applicazione. Infine, SignalR è un framework che consente l'astrazione di diversi protocolli di comunicazione in un'API di facile utilizzo, che consente di creare facilmente un singolo set di strumenti sul server e sul client per facilitare le comunicazioni bidirezionali. Ciò consente di concentrarsi sulla logica che si desidera implementare evitando al contempo di preoccuparsi di browser specifici.

Lezioni

- Implementazione di una strategia di memorizzazione in cache
- Gestione dello stato
- Comunicazione bidirezionale

Lab: Prestazioni e comunicazione

- Implementazione di una strategia di memorizzazione in cache
- Gestione dello stato
- Comunicazione bidirezionale

Al termine di questo modulo gli studenti saranno in grado di:

- Implementare la memorizzazione nella cache in un'applicazione Microsoft ASP.NET Core.
- Utilizzare le tecnologie di gestione dello stato per migliorare l'esperienza del cliente fornendo un'esperienza coerente per l'utente.
- Implementare la comunicazione bidirezionale utilizzando SignalR, consentendo al server di avvisare il

client quando si verificano eventi importanti.

Modulo 13: Implementazione delle API Web

La maggior parte delle applicazioni Web richiede l'integrazione con sistemi esterni come le applicazioni mobili. È necessario sapere come utilizzare le API Web per promuovere l'interazione delle applicazioni con sistemi esterni. È possibile utilizzare l'API Web per implementare i servizi REST (Representational State Transfer) nell'applicazione. I servizi REST aiutano a ridurre il sovraccarico delle applicazioni e limitare i dati trasmessi tra i sistemi client e server. È necessario sapere come chiamare un'API Web utilizzando il codice lato server e il codice jQuery per implementare in modo efficace API Web in stile REST nella propria applicazione.

Lezioni

- Presentazione delle API Web
- Sviluppo di un'API Web
- Chiamata di un'API Web

Lab: Implementazione delle API Web

- Aggiunta di azioni e chiamate tramite Microsoft Edge
- Chiamata di un'API Web utilizzando il codice lato server
- Chiamata di un'API Web utilizzando jQuery

Al termine di questo modulo gli studenti saranno in grado di:

- Creare servizi utilizzando l'API Web ASP.NET Core.
- Chiamare un'API Web dal codice lato server e da jQuery.

Modulo 14: Hosting e distribuzione

Le applicazioni ASP.NET Core MVC sono progettate per fornire un servizio a più utenti contemporaneamente richiedendo solo l'installazione del server e ai clienti di utilizzare i browser per l'accesso. Ciò si traduce in applicazioni altamente desiderabili che non si affidano al fatto che l'utente installi software dedicato e garantiscono l'accessibilità da parte dei clienti da un'ampia varietà di macchine. Per configurare un'applicazione ASP.NET Core per un ambiente di produzione sarà necessario compilare il codice e comprimerlo, quindi configurarlo e eseguirlo su un server dedicato. L'hosting prevede l'utilizzo di un server dedicato per contenere l'applicazione compilata e offrirla agli utenti come servizio basato sul web. Esistono molte tecnologie diverse che possono essere utilizzate per ospitare la propria applicazione. Occorre sceglierne quella adatta alle proprie esigenze. La distribuzione è il processo in cui il codice del progetto viene compilato e quindi trasferito al server di hosting in modo da funzionare all'interno dei requisiti dell'ambiente di hosting. Microsoft Azure è un servizio cloud fornito da Microsoft. Può essere utilizzato per ospitare applicazioni ASP.NET Core ed è uno strumento popolare nel mercato della tecnologia cloud. Fornisce comodi servizi relativi alle applicazioni Web con opzioni multiple di fatturazione in base alle esigenze.

Lezioni

- Hosting e distribuzione in loco
- Distribuzione su Microsoft Azure
- Fondamenti di Microsoft Azure

Lab: Hosting e distribuzione

- Distribuzione di un'applicazione Web su Microsoft Azure
- Caricamento di un'immagine nell'archivio BLOB di Azure

Al termine di questo modulo gli studenti saranno in grado di:

- Ospitare e distribuire un'applicazione ASP.NET Core MVC su IIS.
- Ospitare e distribuire un'applicazione ASP.NET Core MVC su Microsoft Azure.
- Essere in grado di utilizzare i servizi offerti da Microsoft Azure per migliorare le funzionalità delle proprie applicazioni Web.