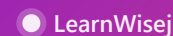


Esercizio di coding con l'IA

Costruisci il progetto AccessOps di questo modulo con ChatGPT o Claude partendo da un prompt già pronto che indirizza il modello alla specifica del modulo, poi rivedi, esegui ed estendi quello che ti restituisce.



Questo progetto non lo scriverai a mano. Lo dirigerai.

Quasi nessuno scrive più codice Wisej.NET da un file vuoto: un assistente scrive la prima bozza e lo sviluppatore decide se è giusta. Funziona però solo se conosci la materia, ed è quello che ti ha dato il resto di questo modulo. Qui affidi quelle conoscenze a ChatGPT o a Claude, ti fai restituire il progetto del modulo come soluzione Visual Studio funzionante, e lo rivedi come rivedresti la pull request di un collega.

Che cosa stai chiedendo

Crea la soluzione AccessOps e scrivine il modello delle minacce: disegna i confini di fiducia, elenca gli asset (anagrafiche dei dipendenti, documenti, approvazioni), enumera i punti di ingresso (login, upload, download, endpoint personalizzati, l'assistente IA), valuta ogni minaccia e associa ogni mitigazione al modulo di questo corso che la implementa. Costruisci lo shell con una pagina di login segnaposto.

Il modulo chiede questi deliverable, e il prompt qui sotto li chiede all'assistente esattamente così:

- ✓ Soluzione AccessOps con una pagina di login segnaposto
- ✓ Diagramma dei confini di fiducia
- ✓ Inventario di asset e punti di ingresso
- ✓ Tabella delle minacce con probabilità, impatto e mitigazione
- ✓ Mappa di posizionamento dei controlli di sicurezza

Perché è fatto così

AccessOps è un portale operativo interno: reparti, dipendenti, documenti riservati, richieste di approvazione e una console di amministrazione. Chiunque possa raggiungerlo lavora già per l'azienda, ed è esattamente per questo che il lavoro sulla sicurezza viene rimandato — e per questo un unico portale finisce per contenere fasce retributive, lettere disciplinari e la traccia delle approvazioni di ogni eccezione che l'azienda abbia mai concesso. Prima che esista una sola riga di codice di autenticazione ti servono tre risposte: che cosa contiene il browser, dove passano i confini, e quali delle cose che potrebbero andare storte vale la pena contrastare per prime.

Questi sono i termini che l'assistente deve azzeccare. Se quello che ti restituisce li usa con leggerezza, non ha capito il modulo:

Termine	Che cosa significa qui
modello di UI lato server	I controlli sono oggetti C# nel processo server; il browser ne tiene uno specchio renderizzato e un canale per gli eventi, quindi il tuo codice e le tue query non vengono mai scaricati.
payload di stato dei widget	I valori delle proprietà che Wisej.NET serializza perché il browser possa disegnare un controllo. Tutto ciò che assegni viaggia in rete, anche sui controlli che l'utente non può vedere.
confine di fiducia	Una linea dove i dati passano tra parti con un diverso livello di fiducia. Tutto ciò che la attraversa viene ricontrollato dalla parte più affidabile, perché l'altra parte può mentire.
confine di sessione	La frontiera tra un browser e il suo stato sul server in Application.Session. Le decisioni nascono dallo stato sul lato server di questo confine, mai da un valore del browser.
punto di ingresso	Qualsiasi via d'accesso raggiungibile da un attaccante: la pagina di login, i gestori di evento, gli upload, i download, gli handler HTTP personalizzati, le chiamate di interop e il prompt dell'assistente IA.
STRIDE	Sei categorie per dare un nome alle minacce — spoofing, manomissione, ripudio, divulgazione di informazioni, denial of service, elevazione dei privilegi — così che una revisione sia sistematica.
posizionamento dei controlli	Scegliere il confine su cui una regola può essere fatta rispettare. Una regola che esiste solo nella UI è un suggerimento; la stessa regola dentro un servizio è un controllo.

Costruiscilo — scegli il tuo assistente

Entrambi i pulsanti aprono una nuova scheda con lo **stesso prompt già pronto**. Il prompt indirizza il modello alla specifica di questo modulo — un unico file Markdown con la lezione, la traccia del lab e i deliverable — così lavora sul materiale vero invece di tirare a indovinare che cosa sia il Modulo 1. Puoi aprire tu stesso [module-01.md](#) per vedere esattamente che cosa leggerà.



Usa ChatGPT

Apri [chatgpt.com](#) con il prompt di questo modulo già caricato. Funziona con GPT-5 o Codex; qualsiasi piano che possa navigare il web recupererà la specifica.



Usa Claude

Apri [claude.ai](#) con lo stesso prompt già caricato. Chi usa Claude Code può invece incollarlo in un terminale dentro una cartella vuota.



Mandalo così com'è e ottieni un progetto, non un frammento

Il prompt si chiude chiedendo l'intera soluzione in un unico `.zip` con i passi per compilarla ed eseguirla. Scaricalo, apri la soluzione in Visual Studio, premi F5 e AccessOps si apre su `localhost`. Ti servono l'SDK .NET e il pacchetto NuGet `Wisej-4`; tutto il resto il prompt chiede all'assistente di scriverlo da sé.

Sto seguendo il Modulo 1 - "Il modello di sicurezza di Wisej.NET" - del corso LearnWisej "Pattern di sicurezza e autenticazione", e voglio costruire con te il progetto del modulo. Rispondimi in italiano.

Comincia leggendo la specifica del modulo. È un unico file Markdown (in inglese) che contiene l'intera lezione, la traccia del lab e i deliverable:

<https://learnwisej.com/assets/courses/security-and-auth-patterns/specs/module-01.md>

Poi costruisci un progetto Wisej.NET 4 completo che io possa aprire in Visual Studio:

1. Progetto "AccessOps", C#, net8.0 o successivo, pacchetto NuGet Wisej-4 4.1.x, host Kestrel che chiama `app.UseWisej()`.
2. Costruisci quello che chiede il lab della specifica: Crea la soluzione AccessOps e scrivine il modello delle minacce: disegna i confini di fiducia, elenca gli asset (anagrafiche dei dipendenti, documenti, approvazioni), enumera i punti di ingresso (login, upload, download, endpoint personalizzati, l'assistente IA), valuta ogni minaccia e associa ogni mitigazione al modulo di questo corso che la implementa. Costruisci lo shell con una pagina di login segnaposto.
3. Produci questi deliverable e dimmi dove si trova ciascuno nella soluzione:
 - Soluzione AccessOps con una pagina di login segnaposto
 - Diagramma dei confini di fiducia
 - Inventario di asset e punti di ingresso
 - Tabella delle minacce con probabilità, impatto e mitigazione
 - Mappa di posizionamento dei controlli di sicurezza
4. Segui la specifica ovunque nomini una classe, una proprietà o un evento di Wisej.NET - usa esattamente quello. Dove lascia una scelta aperta, dimmi quale API hai scelto e perché.
5. Tieni l'autorità sul server: stato e validazione vivono in .NET, il browser disegna e riferisce. Non spostare mai una decisione in JavaScript solo perché lì è più comodo.
6. Rendi il risultato dimostrabile: devo poterlo eseguire e vedere il comportamento di cui parla questo modulo, compreso che cosa succede quando qualcosa va storto.

Prima di scrivere codice, riformula con parole tue che cosa costruirai e dove starà ogni pezzo, così posso correggere il progetto finché costa poco.

Quando il codice è pronto, impacchetta l'intera soluzione in un unico file .zip scaricabile e dammi i passi esatti per compilarla ed eseguirla.

Infine rispondi a queste domande della specifica, usando come prova la tua implementazione:

- La griglia dei dipendenti è collegata ai record completi e la colonna dello stipendio è nascosta nel designer. Descrivi che cosa contiene il browser e indica la modifica che toglie davvero lo stipendio da ciò che viaggia in rete.
- Due persone hanno effettuato l'accesso ad AccessOps sullo stesso server. Spiega quali parti dello stato condividono per impostazione predefinita, e che cosa dovrebbe essere vero perché una delle due veda il reparto corrente dell'altra.
- Scegli la minaccia della tua tabella con l'impatto più alto e sostieni la tesi opposta: che cosa dovrebbe essere vero di AccessOps perché quella mitigazione sia una perdita di tempo per il team?

Chiedimi qualsiasi cosa non ti sia chiara prima di cominciare.

Poi rivedi quello che ti è tornato indietro

Codice che compila non è ancora codice giusto. Le domande interessanti sono domande di progetto: è questo il controllo Wisej.NET che il compito richiede davvero, segue il ciclo di vita che il framework ti offre, lo stato autorevole sta sul server o è stato parcheggiato nel browser perché era comodo? Leggi il risultato da revisore, non da cliente.

Questo modulo ti dà la checklist. Rispondi a ciascuna domanda guardando il codice che hai appena ricevuto:

- ✓ La griglia dei dipendenti è collegata ai record completi e la colonna dello stipendio è nascosta nel designer. Descrivi che cosa contiene il browser e indica la modifica che toglie davvero lo stipendio da ciò che viaggia in rete.
- ✓ Due persone hanno effettuato l'accesso ad AccessOps sullo stesso server. Spiega quali parti dello stato condividono per impostazione predefinita, e che cosa dovrebbe essere vero perché una delle due veda il reparto corrente dell'altra.
- ✓ Scegli la minaccia della tua tabella con l'impatto più alto e sostieni la tesi opposta: che cosa dovrebbe essere vero di AccessOps perché quella mitigazione sia una perdita di tempo per il team?

Tutto ciò che hai dovuto correggere vale la pena di scriverlo. Una breve nota con le convenzioni e le decisioni a cui sei arrivato, incollata nel primo messaggio della sessione successiva, vale più di qualsiasi prompt ingegnoso.



Per questo corso non c'è ancora un esempio di riferimento — il tuo è il primo

Il [repository degli esempi LearnWisej](#) contiene build già fatte per quasi tutti i corsi, ma non ancora per questo. Significa che non c'è niente con cui confrontarsi e nessuna scorciatoia: la specifica e la tua revisione sono tutta la rete di sicurezza. Tieni quello che costruisci — e se viene bene, ci farebbe piacere vederlo.

Tre prompt per andare oltre

Punta il tuo assistente sul progetto che hai appena costruito e prova questi. Ognuno è un esercizio di revisione ed estensione, non una riscrittura.

- ✓ **Dimostra che funziona davvero.** *"Passa in rassegna questo progetto deliverable per deliverable rispetto alla specifica del modulo su <https://learnwisej.com/assets/courses/security-and-auth-patterns/specs/module-01.md> e mostrami la prova che ciascuno è davvero fatto — il percorso nel codice, su che cosa cliccare e che cosa dovrei vedere. Dove un deliverable è soddisfatto solo in parte, dillo chiaramente invece di arrampicarti sugli specchi. Poi aggiungi quello che manca."*
- ✓ **Rivedilo come faresti con una pull request.** *"Accompagnami attraverso questo codice AccessOps come se stessi rivedendo la pull request di un collega e non l'avessi scritto tu. Dove vive lo stato, che cosa succede dall'inizio alla fine su una singola azione dell'utente, che cosa rimanderesti indietro per modifiche, e quali parti non ti sentiresti tranquillo a mettere davanti a un cliente? Sii specifico e cita le righe di cui parli — non riassumermi il progetto."*

- ✓ **Sostituisci un pezzo senza spostare le giunzioni.** *"Scegli la parte di questo progetto AccessOps che più evidentemente sta al posto di qualcosa di reale — una sorgente dati finta, un archivio in memoria, un servizio semplificato — e sostituiscila con un'implementazione di forma produttiva. Il contratto ai due lati non deve cambiare: stessi tipi, stesse chiamate, stesso comportamento in caso di errore. Dimmi che cosa l'implementazione reale ti ha costretto a gestire e che la versione provvisoria non gestiva."*



Portalo dentro il lab

Il lab pratico che chiude questo modulo chiede gli stessi deliverable che hai appena fatto produrre all'assistente. Usa quello che hai qui come forma della tua risposta e come prova a sostegno — non come la risposta in sé. Il lab viene valutato sul tuo ragionamento, e quella parte deve essere tua.