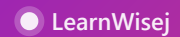


Esercizio di coding con l'IA

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.



L'assistente è la coppia di mani veloci. L'architetto resti tu.

Un assistente può produrre il progetto di questo modulo in pochi minuti. Quello che non può fare è decidere se il risultato è il progetto giusto per la tua applicazione: quella scelta resta tua. Il giudizio lo metti tu: istruiscilo per bene, poi verifica quello che ti torna indietro rispetto a ciò che il Modulo 7 ti ha appena insegnato.

Che cosa stai chiedendo

Completa il capstone di AccessOps: imponi HTTPS con HSTS e imposta gli header di sicurezza, marca i cookie come Secure e HttpOnly con una policy SameSite restrittiva, imposta il timeout di sessione e i limiti di dimensione delle richieste, aggiungi il rate limiting su login e link di condivisione, esegui un audit delle dipendenze e registrane i risultati, poi applica all'app l'intera checklist di questo corso e correggi ogni problema rilevato.

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

- ✓ Configurazione di HTTPS, HSTS e header di sicurezza
- ✓ Impostazioni di irrobustimento di cookie e sessione
- ✓ Rate limiting su login e link di condivisione
- ✓ Report dell'audit delle dipendenze
- ✓ Checklist della revisione di sicurezza completata, con le correzioni

Perché è fatto così

Ora AccessOps verifica le password rispetto a un hash con salt, risolve i permessi in un servizio, invia i documenti solo tramite DocumentService e sottopone ad audit ogni approvazione. Nulla di tutto questo aiuta se la prima richiesta viaggia in chiaro, se uno script può leggere il cookie di sessione, o se la build distribuisce un package con un advisory pubblicato. L'ultimo miglio spetta all'host: il certificato, gli header di risposta, i flag dei cookie, i limiti delle richieste e la versione di ogni elemento del grafo delle dipendenze.

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
HSTS	Un header Strict-Transport-Security che dice al browser di rifiutare HTTP in chiaro per questo host per un tempo dichiarato, così una visita successiva non viene mai intercettata sul redirect.
Content-Security-Policy	Un header che elenca le origini da cui una pagina può caricare script, stili e frame; trasforma uno script iniettato in una richiesta bloccata invece che in codice eseguito in una sessione attiva.
frame-ancestors	La direttiva CSP (X-Frame-Options è l'equivalente più vecchio) che impedisce a un altro sito di inserire AccessOps in un frame e indurre un manager a cliccare Approve.
flag dei cookie	Secure tiene il cookie di sessione lontano da HTTP in chiaro, HttpOnly lo tiene lontano da JavaScript, e SameSite impedisce che venga allegato a una richiesta avviata da un altro sito.
limiti delle richieste	Tetti a livello di host sulla dimensione del corpo e degli header, che rifiutano un upload troppo grande prima che giri una sola riga di codice di AccessOps.
token bucket	Un contatore che concede un tentativo solo finché contiene un token e si ricarica a velocità fissa, così il traffico può avere una piccola raffica e poi rallenta fino quasi a fermarsi.
audit delle dipendenze	Una verifica ripetibile di ogni package diretto e transitivo, e del runtime, rispetto agli advisory pubblicati — registrata con una data e una decisione.

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 7. Puoi aprire tu stesso [module-07.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 7 - "Checklist di irrobustimento e capstone" - 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-07.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: Completa il capstone di AccessOps: imponi HTTPS con HSTS e imposta gli header di sicurezza, marca i cookie come Secure e HttpOnly con una policy SameSite restrittiva, imposta il timeout di sessione e i limiti di dimensione delle richieste, aggiungi il rate limiting su login e link di condivisione, esegui un audit delle dipendenze e registrane i risultati, poi applica all'app l'intera checklist di questo corso e correggi ogni problema rilevato.
3. Produci questi deliverable e dimmi dove si trova ciascuno nella soluzione:
 - Configurazione di HTTPS, HSTS e header di sicurezza
 - Impostazioni di irrobustimento di cookie e sessione
 - Rate limiting su login e link di condivisione
 - Report dell'audit delle dipendenze
 - Checklist della revisione di sicurezza completata, con le correzioni
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:

- Un middleware può applicare il rate limiting all'handler dei link di condivisione ma non al pulsante Sign in. Che cosa hanno di diverso le due richieste in un'app Wisej.NET, e dove deve vivere invece il limite sul login?
- La tua Content-Security-Policy è vincolante e la griglia viene renderizzata senza stili. Quale direttiva è la causa probabile, e come te lo avrebbe detto la modalità report-only prima che lo facesse il deployment?
- L'audit delle dipendenze elenca un package vulnerabile che nessuno in AccessOps referencia direttamente. Come è entrato nella build, e quali sono i due modi onesti per chiudere il problema?

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:

- ✓ Un middleware può applicare il rate limiting all'handler dei link di condivisione ma non al pulsante Sign in. Che cosa hanno di diverso le due richieste in un'app Wisej.NET, e dove deve vivere invece il limite sul login?
- ✓ La tua Content-Security-Policy è vincolante e la griglia viene renderizzata senza stili. Quale direttiva è la causa probabile, e come te lo avrebbe detto la modalità report-only prima che lo facesse il deployment?
- ✓ L'audit delle dipendenze elenca un package vulnerabile che nessuno in AccessOps referencia direttamente. Come è entrato nella build, e quali sono i due modi onesti per chiudere il problema?

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.

- ✓ **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."*

- ✓ **Fallo sopravvivere a una brutta giornata.** *"Esamina questo progetto AccessOps cercando le modalità di guasto che non gestisce ancora — input sbagliato o mancante, una dipendenza lenta o non disponibile, lavoro che arriva dopo che l'utente ha cambiato pagina, e due azioni che si accavallano. Per ciascuna dimmi prima se è un rischio reale in Wisej.NET 4 e perché, e solo dopo scrivi la correzione. Dove decidi che non serve gestire nulla, dillo e spiega il ragionamento invece di aggiungere codice difensivo."*



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.