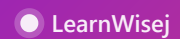


Esercizio di coding con l'IA

Costruisci il progetto IconDesk 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 IconDesk. Referenzia un foglio di stile di font di icone da Default.html, imposta AllowHtml su una piccola toolbar o su una card e disegna le sue icone come glifi del font, poi costruisci una colonna di DataGridView le cui celle contengono azioni realizzate con il font di icone e leggi quale elemento ha cliccato l'utente, così che Edit e Delete facciano cose diverse. Costruisci una pagina di riepilogo che dimostra tutti insieme i meccanismi visti nel corso: un'immagine del tema, un'icona di un pacchetto ufficiale, un'icona del tuo pacchetto, un logo incorporato e un SVG ricolorato. Poi scrivi la policy delle icone in produzione del progetto ed esegui un giro di QA con un tema chiaro e uno scuro.

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

- ✓ Foglio di stile di font di icone referenziato da Default.html e una toolbar o card con AllowHtml che disegna glifi del font
- ✓ Colonna di DataGridView le cui celle mostrano azioni con font di icone e rispondono all'elemento cliccato
- ✓ Pagina di riepilogo di IconDesk che dimostra un'immagine del tema, un'icona di un pacchetto ufficiale, un'icona del pacchetto personalizzato, un asset incorporato e un SVG ricolorato
- ✓ Policy scritta delle icone in produzione che copre accessibilità, licenze, caching e dipendenze esterne
- ✓ Giro di QA eseguito con un tema chiaro e uno scuro, con i risultati annotati

Perché è fatto così

Dopo sei moduli IconDesk sa disegnare un'icona in cinque modi diversi, e funzionano tutti. Quello che al progetto manca ancora è una regola di decisione: quando una nuova schermata ha bisogno di un simbolo Save, nessuno dovrebbe stare a indovinare se arriva dal tema, da un pacchetto ufficiale Wisej-4 o da IconDesk.Icons. Questo modulo aggiunge l'ultima opzione, i font di icone disegnati come HTML, e poi

scrive la policy che dice a quale ricorrere e che cosa controllare prima che una schermata vada in produzione.

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
font di icone	Un web font i cui glifi sono immagini invece che lettere, scelti con nomi di classe CSS e disegnati dal browser dentro il contenuto HTML invece che tramite una qualsiasi proprietà immagine.
AllowHtml	L'interruttore per singolo controllo che dice a Wisej.NET di interpretare il testo di un controllo come markup; senza di esso i tag compaiono come caratteri letterali, perché per impostazione predefinita il testo è testo semplice.
head di Default.html	La pagina host dell'applicazione, dove un unico link a un foglio di stile rende il CSS e i file del font di icone disponibili a ogni schermata che la sessione apre.
click sull'elemento distinto per ruolo	Un click dentro il contenuto HTML può riferire quale elemento è stato colpito, così che una sola cella della griglia offra un bersaglio Edit e un bersaglio Delete senza due controlli separati.
CDN contro hosting proprio	Puntare il link a una CDN pubblica è una decisione di produzione su disponibilità, privacy, Content Security Policy e licenze; le applicazioni interne ospitano spesso i file da sé.
policy delle icone a livelli	Una regola scritta che dà a ogni tipo di simbolo una casa sola: immagini del tema per i significati, un pacchetto principale per i comandi, il pacchetto personalizzato per la grafica di prodotto, file raster per le fotografie.
giro di QA delle icone	Un passaggio sulle schermate finite con un tema chiaro e uno scuro a più di una scala di visualizzazione, verificando che ogni icona sia ancora visibile e abbia ancora un'alternativa testuale.

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



Usa Claude

Apri [claude.ai](#) con lo stesso prompt già caricato. Chi usa Claude Code può invece

Codex; qualsiasi piano che possa navigare il web recupererà la specifica.

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 IconDesk 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 - "Font di icone, contenuto HTML e strategia delle icone in produzione" - del corso LearnWisej "Icone e immagini in Wisej.NET", 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/icons-and-images/specs/module-07.md>

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

1. Progetto "IconDesk", 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 IconDesk. Referenzia un foglio di stile di font di icone da `Default.html`, imposta `AllowHtml` su una piccola toolbar o su una card e disegna le sue icone come glifi del font, poi costruisci una colonna di `DataGridView` le cui celle contengono azioni realizzate con il font di icone e leggi quale elemento ha cliccato l'utente, così che `Edit` e `Delete` facciano cose diverse. Costruisci una pagina di riepilogo che dimostra tutti insieme i meccanismi visti nel corso: un'immagine del tema, un'icona di un pacchetto ufficiale, un'icona del tuo pacchetto, un logo incorporato e un SVG ricolorato. Poi scrivi la policy delle icone in produzione del progetto ed esegui un giro di QA con un tema chiaro e uno scuro.
3. Produci questi deliverable e dimmi dove si trova ciascuno nella soluzione:
 - Foglio di stile di font di icone referenziato da `Default.html` e una toolbar o card con `AllowHtml` che disegna glifi del font
 - Colonna di `DataGridView` le cui celle mostrano azioni con font di icone e rispondono all'elemento cliccato
 - Pagina di riepilogo di IconDesk che dimostra un'immagine del tema, un'icona di un pacchetto ufficiale, un'icona del pacchetto personalizzato, un asset incorporato e un SVG ricolorato
 - Policy scritta delle icone in produzione che copre accessibilità, licenze, caching e dipendenze esterne
 - Giro di QA eseguito con un tema chiaro e uno scuro, con i risultati annotati
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:

- Il link al foglio di stile è in `Default.html` e i nomi di classe sono giusti, ma una `Label` mostra i tag grezzi mentre il `Button` accanto mostra il glifo. Che cosa differisce tra i due controlli, e perché l'impostazione è per singolo controllo invece che per applicazione?
- La tua colonna di azioni funziona, e un collega chiede di mettere nella stessa cella il testo della nota scritta dal cliente, così da poterlo formattare. Che cosa gli rispondi, e che cosa costruiresti al suo posto?

– Nella pagina di riepilogo, quali delle cinque immagini cambiano quando l'applicazione cambia tema e quali restano esattamente come disegnate? Che cosa ti dice questo su dove ciascuna è conservata?

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:

- ✓ Il link al foglio di stile è in Default.html e i nomi di classe sono giusti, ma una Label mostra i tag grezzi mentre il Button accanto mostra il glifo. Che cosa differisce tra i due controlli, e perché l'impostazione è per singolo controllo invece che per applicazione?
- ✓ La tua colonna di azioni funziona, e un collega chiede di mettere nella stessa cella il testo della nota scritta dal cliente, così da poterlo formattare. Che cosa gli rispondi, e che cosa costruiresti al suo posto?
- ✓ Nella pagina di riepilogo, quali delle cinque immagini cambiano quando l'applicazione cambia tema e quali restano esattamente come disegnate? Che cosa ti dice questo su dove ciascuna è conservata?

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 IconDesk 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 IconDesk 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 IconDesk 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.