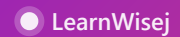


Esercizio di coding con l'IA

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.



Scrivere il codice è ormai la metà facile.

Ottenere codice funzionante per "I pacchetti di icone di Wisej.NET e NuGet" da un assistente moderno richiede un buon prompt. Giudicarlo, correggerlo ed estenderlo richiede tutto quello che questo modulo ha trattato. È su questa divisione che è costruito l'esercizio: tu metti la specifica e la revisione, l'assistente mette la digitazione.

Che cosa stai chiedendo

Aggiungi a IconDesk due package NuGet ufficiali di pacchetti di icone Wisej-4 e costruisci una pagina IconCompare che mostra gli stessi cinque concetti — Save, Delete, Search, User e Settings — presi da entrambe le famiglie, così che la differenza visiva sia impossibile da ignorare. Scegli almeno cinque icone attraverso l'Image Explorer di Visual Studio invece di scrivere il percorso, assegna le restanti dal catalogo del pacchetto in C#, ricolorane due con il suffisso ?color= e osserva nel pannello di rete i file delle icone che arrivano una richiesta alla volta e poi arrivano dalla cache del browser. Chiudi con una breve raccomandazione che indica la famiglia di icone principale del progetto.

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

- ✓ Due package NuGet ufficiali di pacchetti di icone Wisej-4 referenziati da IconDesk
- ✓ Pagina IconCompare che mostra Save, Delete, Search, User e Settings da entrambi i pacchetti
- ✓ Almeno cinque icone scelte tramite l'Image Explorer di Visual Studio invece che scritte a mano
- ✓ Due icone ricolorate con il suffisso di colore e una nota su quale grafica ha accettato la ricolorazione
- ✓ Raccomandazione di un paragrafo che indica la famiglia di icone principale del progetto, con le prove raccolte dalla cache del browser

Perché è fatto così

Dopo tre moduli IconDesk sa già mostrare un'immagine in cinque modi diversi, ma ogni icona che possiede è stata disegnata o scaricata a mano. Un'applicazione vera ha bisogno di un'icona Save, una Delete, una Search, una User e una Settings fin dal primo giorno, e di altre quaranta entro fine trimestre, tutte nello stesso linguaggio visivo. Copiare i file SVG uno alla volta produce una cartella di grafica sCOORDINATA, spessori di tratto incoerenti e nessuna traccia delle licenze.

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
pacchetto di icone	Un normale assembly.NET i cui file delle icone, per lo più SVG, sono compilati dentro come risorse incorporate; referenziarlo aggiunge un solo file alla tua distribuzione, per quante icone contenga.
risorsa incorporata	Un file compilato dentro l'assembly invece che copiato accanto a esso, così le icone non possono essere perse, rinominate o dimenticate quando l'applicazione viene distribuita su un altro server.
resource.wx	L'handler di Wisej.NET che serve una singola risorsa incorporata via HTTP; la sorgente immagine di un pacchetto di icone è un URL resource.wx che indica l'assembly del pacchetto e il file al suo interno.
catalogo delle icone	La classe statica che un pacchetto pubblica perché chi la usa abbia IntelliSense e un nome verificato dal compilatore invece di un percorso scritto a mano; il namespace e il nome esatto della classe variano da pacchetto a pacchetto.
richiesta per singola icona	Le icone dei pacchetti vengono richieste una a una, al primo utilizzo, e poi servite dalla cache del browser: è questo che rende sostenibile un catalogo da migliaia di icone.
precaricamento delle immagini del tema	Le immagini del tema viaggiano con il tema e sono già nella cache del tema sul client quando l'applicazione parte, quindi all'avvio si comportano in modo molto diverso dalle icone dei pacchetti.
famiglia di icone principale	L'unico pacchetto su cui un prodotto si impegna per i suoi comandi di ogni giorno, scelto di proposito per copertura, stile e licenza invece che accumulato pacchetto dopo pacchetto.

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 4. Puoi aprire tu stesso [module-04.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 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 4 - "I pacchetti di icone di Wisej.NET e NuGet" - 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-04.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: Aggiungi a IconDesk due package NuGet ufficiali di pacchetti di icone Wisej-4 e costruisci una pagina IconCompare che mostra gli stessi cinque concetti - Save, Delete, Search, User e Settings - presi da entrambe le famiglie, così che la differenza visiva sia impossibile da ignorare. Scegli almeno cinque icone attraverso l'Image Explorer di Visual Studio invece di scrivere il percorso, assegna le restanti dal catalogo del pacchetto in C#, ricolorane due con il suffisso `?color=` e osserva nel pannello di rete i file delle icone che arrivano una richiesta alla volta e poi arrivano dalla cache del browser. Chiudi con una breve raccomandazione che indica la famiglia di icone principale del progetto.
3. Produci questi deliverable e dimmi dove si trova ciascuno nella soluzione:
 - Due package NuGet ufficiali di pacchetti di icone Wisej-4 referenziati da IconDesk
 - Pagina IconCompare che mostra Save, Delete, Search, User e Settings da entrambi i pacchetti
 - Almeno cinque icone scelte tramite l'Image Explorer di Visual Studio invece che scritte a mano
 - Due icone ricolorate con il suffisso di colore e una nota su quale grafica ha accettato la ricolorazione
 - Raccomandazione di un paragrafo che indica la famiglia di icone principale del progetto, con le prove raccolte dalla cache del browser
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:

- Referenziare un pacchetto da tremila icone aggiunge un solo assembly alla distribuzione. Che cosa costa al browser, e quando?
- Una delle tue due icone ricolorate ha ignorato il suffisso di colore. Che cosa c'è in quella grafica, e che cosa cambieresti per questo nei criteri di scelta dei pacchetti?
- Un collega vuole che ogni icona di comando segua il tema quando l'applicazione cambia tema. Quali delle cinque icone di IconCompare dovrebbero diventare immagini del tema, e quali no?

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:

- ✓ Referenziare un pacchetto da tremila icone aggiunge un solo assembly alla distribuzione. Che cosa costa al browser, e quando?
- ✓ Una delle tue due icone ricolorate ha ignorato il suffisso di colore. Che cosa c'è in quella grafica, e che cosa cambieresti per questo nei criteri di scelta dei pacchetti?
- ✓ Un collega vuole che ogni icona di comando segua il tema quando l'applicazione cambia tema. Quali delle cinque icone di IconCompare dovrebbero diventare immagini del tema, e quali no?

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.

- ✓ **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."*
- ✓ **Aggiungi il secondo caso.** *"Estendi questo progetto IconDesk con un secondo caso che segue lo stesso schema del primo — un'altra entità, schermata o controllo della stessa forma, collegato allo stesso modo. Estrai in una classe base o in un servizio ciò che è davvero condiviso, ma solo dove se lo merita; due cose simili non chiedono sempre un'astrazione. Mostrami il diff e giustifica ogni cosa che hai estratto."*

- ✓ **Dimostra che funziona davvero.** *"Passa in rassegna questo progetto deliverable per deliverable rispetto alla specifica del modulo su <https://learnwisej.com/assets/courses/icons-and-images/specs/module-04.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."*



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.