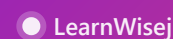


Esercizio di coding con l'IA

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.



Istruiscilo, leggilo, poi mettilo alla prova.

Questo esercizio ha tre parti, e solo la prima riguarda il prompt. Istruisci un assistente con la specifica del modulo e ti fai restituire una soluzione IconDesk; leggi quello che ha scritto e decidi se fa davvero ciò che il modulo chiedeva; poi lo spingi più in là di quanto il modulo chiedesse. La seconda e la terza parte sono dove si impara.

Che cosa stai chiedendo

Riempi in IconDesk una pagina IconGallery usando l'Image Selector del designer invece di scrivere a mano le stringhe delle sorgenti. Assegna le immagini a cinque controlli: un'immagine del tema indicata per nome, un SVG del progetto referenziato con URL relativo, un URL assoluto e almeno due asset di un pacchetto di icone, se ne hai installato uno. Ricolora un SVG monocromatico con il suffisso ?color= e il nome di un colore del tema, apri il Designer.cs generato e leggi le stringhe delle sorgenti prodotte dal selettore, poi passa l'applicazione da un tema Wisej.NET a un altro e annota quali asset sono cambiati e quali sono rimasti identici.

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

- ✓ Pagina IconGallery con cinque controlli le cui immagini sono state assegnate attraverso l'Image Selector del designer
- ✓ Tra di esse un'immagine del tema indicata per nome, un SVG relativo del progetto, un URL assoluto e almeno due asset di un pacchetto di icone
- ✓ SVG monocromatico ricolorato attraverso il suffisso?color= con il nome di un colore del tema
- ✓ Estratto di Designer.cs che mostra le stringhe di sorgente immagine generate dal picker
- ✓ Nota sul cambio di tema che registra quali asset hanno seguito il tema e quali sono rimasti fissi

Perché è fatto così

IconDesk ha ormai un pannello dei comandi e un image lab, e ogni immagine che vi compare è un file distribuito più un percorso che qualcuno ha scritto a mano. Regge finché l'applicazione non viene ritematizzata, distribuita sotto una virtual directory diversa oppure chiamata a mostrare lo stesso simbolo in un colore di avviso su una schermata e in uno neutro su un'altra. Questo modulo sposta l'origine di un'immagine fuori dalla cartella del progetto e dentro il tema, un URL o il designer, così che la stringa portata da un controllo continui a significare qualcosa dopo che la grafica, il tema o il server sono cambiati.

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
immagine del tema indicata per nome	Un nome semantico come icon-print che il tema attivo mappa su una grafica, così un tema diverso può fornire un'immagine diversa senza alcuna modifica in C#.
cache del tema sul client	Le immagini del tema viaggiano con il tema e sono già nel browser quando un controllo ne chiede una: questo rende il tema la casa giusta per un piccolo insieme di simboli di comando e quella sbagliata per migliaia di icone.
sorgente con URL relativo	Una stringa di sorgente immagine risolta rispetto all'applicazione: la forma più semplice per un logo distribuito insieme all'applicazione.
sorgente con URL assoluto	Un indirizzo esterno completo che trasforma l'immagine in una dipendenza da un altro sistema, con conseguenze su disponibilità, privacy, CORS e Content Security Policy.
immagine base64 incorporata	I byte dell'immagine scritti dentro la definizione del tema stessa, così la grafica non richiede una seconda richiesta, al prezzo di un tema più pesante.
Image Selector	Il selettore di Visual Studio in fase di progettazione che esplora le immagini del progetto, le immagini del tema e le risorse dei pacchetti di icone referenziati, poi serializza la stringa di sorgente scelta dentro Designer.cs.
suffisso?color=	Un suffisso su una sorgente immagine che chiede a Wisej.NET di ricolorare un SVG monocromatico compatibile con un colore in stile CSS o con il nome di un colore del tema.

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 3. Puoi aprire tu stesso [module-03.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 3 - "Immagini del tema, URL, colore degli SVG e l'Image Picker del designer" - 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-03.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: Riempi in IconDesk una pagina IconGallery usando l'Image Selector del designer invece di scrivere a mano le stringhe delle sorgenti. Assegna le immagini a cinque controlli: un'immagine del tema indicata per nome, un SVG del progetto referenziato con URL relativo, un URL assoluto e almeno due asset di un pacchetto di icone, se ne hai installato uno. Ricolora un SVG monocromatico con il suffisso `?color=` e il nome di un colore del tema, apri il Designer.cs generato e leggi le stringhe delle sorgenti prodotte dal selettore, poi passa l'applicazione da un tema Wisej.NET a un altro e annota quali asset sono cambiati e quali sono rimasti identici.
3. Produci questi deliverable e dimmi dove si trova ciascuno nella soluzione:
 - Pagina IconGallery con cinque controlli le cui immagini sono state assegnate attraverso l'Image Selector del designer
 - Tra di esse un'immagine del tema indicata per nome, un SVG relativo del progetto, un URL assoluto e almeno due asset di un pacchetto di icone
 - SVG monocromatico ricolorato attraverso il suffisso `?color=` con il nome di un colore del tema
 - Estratto di Designer.cs che mostra le stringhe di sorgente immagine generate dal picker
 - Nota sul cambio di tema che registra quali asset hanno seguito il tema e quali sono rimasti fissi
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 gallery si disegna correttamente sulla tua macchina e non mostra nulla per l'URL assoluto su quella di un collega. Quali dei cinque meccanismi potrebbero produrre quella differenza, e che cosa controlleresti per primo?
- Due delle tue icone si rifiutano di cambiare colore quando aggiungi il suffisso. Che cosa c'è dentro quei file, e quando modificarli è la risposta sbagliata?
- Un designer chiede lo stesso simbolo nella gallery, in una toolbar e in una cella di griglia. Tra la chiave del tema, l'URL relativo e la risorsa del pacchetto, quale costa meno cambiare in futuro, e perché?

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 gallery si disegna correttamente sulla tua macchina e non mostra nulla per l'URL assoluto su quella di un collega. Quali dei cinque meccanismi potrebbero produrre quella differenza, e che cosa controlleresti per primo?
- ✓ Due delle tue icone si rifiutano di cambiare colore quando aggiungi il suffisso. Che cosa c'è dentro quei file, e quando modificarli è la risposta sbagliata?
- ✓ Un designer chiede lo stesso simbolo nella gallery, in una toolbar e in una cella di griglia. Tra la chiave del tema, l'URL relativo e la risorsa del pacchetto, quale costa meno cambiare in futuro, e perché?

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.

- ✓ **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."

- ✓ **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."



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.