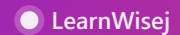


KI-Coding-Übung

Erstellen Sie das Beispielprojekt IconDesk dieses Moduls mit ChatGPT oder Claude aus einem fertigen Prompt, der das Modell auf die Spezifikation des Moduls verweist – und prüfen, starten und erweitern Sie dann, was Sie erhalten.



Der Assistent liefert die schnellen Hände. Der Architekt bleiben Sie.

Ein Assistent erstellt das Beispielprojekt dieses Moduls in wenigen Minuten. Was er nicht kann, ist zu entscheiden, ob das Ergebnis der richtige Entwurf für Ihre Anwendung ist – diese Entscheidung bleibt bei Ihnen. Das Urteilsvermögen bringen Sie mit: Instruieren Sie ihn sorgfältig und prüfen Sie dann, was zurückkommt, anhand dessen, was Ihnen Modul 7 gerade vermittelt hat.

Worum Sie bitten

Schließen Sie das Abschlussprojekt IconDesk ab. Referenzieren Sie ein Icon-Font-Stylesheet aus Default.html, setzen Sie AllowHtml auf einer kleinen Toolbar oder Karte und rendern Sie deren Icons als Font-Glyphen; bauen Sie dann eine DataGridView-Spalte, deren Zellen Icon-Font-Aktionen enthalten, und lesen Sie aus, welches Element angeklickt wurde, damit Edit und Delete Unterschiedliches tun. Bauen Sie eine Übersichtsseite, die alle Mechanismen des Kurses auf einmal zeigt: ein Theme-Bild, ein Icon aus einem offiziellen Pack, ein Icon aus dem eigenen Pack, ein eingebettetes Logo und ein umgefärbtes SVG. Schreiben Sie anschließend die Icon-Policy des Projekts für den Produktivbetrieb und führen Sie einen QA-Durchgang in einem hellen und einem dunklen Theme durch.

Das Modul verlangt diese Ergebnisse, und der Prompt unten fordert genau diese vom Assistenten an:

- ✓ Icon-Font-Stylesheet, das aus Default.html referenziert wird, und eine Toolbar oder Karte mit AllowHtml, die Font-Glyphen rendert
- ✓ DataGridView-Spalte, deren Zellen Icon-Font-Aktionen rendern und auf das angeklickte Element reagieren
- ✓ IconDesk-Übersichtsseite mit einem Theme-Bild, einem Icon aus einem offiziellen Pack, einem Icon aus dem eigenen Pack, einem eingebetteten Asset und einem umgefärbten SVG
- ✓ Schriftliche Icon-Policy für den Produktivbetrieb zu Barrierefreiheit, Lizenzen, Caching und externen Abhängigkeiten
- ✓ QA-Durchgang in einem hellen und einem dunklen Theme mit festgehaltenen Ergebnissen

Warum es so aufgebaut ist

Nach sechs Modulen kann IconDesk ein Icon auf fünf verschiedene Arten zeichnen, und jede davon funktioniert. Was dem Projekt noch fehlt, ist eine Entscheidungsregel: Wenn ein neuer Bildschirm ein Save-Symbol braucht, sollte niemand raten müssen, ob es aus dem Theme, aus einem offiziellen Wisej-4-Pack

oder aus IconDesk.Icons kommt. Dieses Modul ergänzt die letzte Möglichkeit – Icon-Fonts, die als HTML gerendert werden – und schreibt dann die Policy, die sagt, wozu man greift und was vor der Auslieferung eines Bildschirms zu prüfen ist.

Diese Begriffe muss der Assistent richtig verwenden. Geht die Antwort nachlässig mit ihnen um, hat er das Modul nicht verstanden:

Begriff	Was er hier bedeutet
Icon-Font	Eine Webschrift, deren Glyphen Bilder statt Buchstaben sind, ausgewählt über CSS-Klassennamen und vom Browser in HTML-Inhalten gezeichnet statt über eine Bildeigenschaft.
AllowHtml	Der Schalter pro Steuerelement, der Wisej.NET anweist, den Text eines Steuerelements als Markup zu interpretieren; ohne ihn erscheinen die Tags als wörtliche Zeichen, denn Text ist standardmäßig reiner Text.
Head von Default.html	Die Hostseite der Anwendung, in der ein einziger Stylesheet-Link die CSS- und Font-Dateien des Icon-Fonts für jeden Bildschirm der Session verfügbar macht.
rollenbasierter Element-Klick	Ein Klick in HTML-Inhalt kann melden, welches Inline-Element getroffen wurde – eine Grid-Zelle kann also ein Ziel für Edit und eines für Delete anbieten, ohne zwei getrennte Steuerelemente.
CDN gegenüber Selbst-Hosting	Den Link auf ein öffentliches CDN zu richten ist eine Entscheidung für den Produktivbetrieb über Verfügbarkeit, Datenschutz, Content Security Policy und Lizenzen; interne Anwendungen hosten die Dateien oft selbst.
gestufte Icon-Policy	Eine schriftliche Regel, die jeder Art von Symbol einen Ort gibt: Theme-Bilder für Semantiken, ein primäres Pack für Befehle, das eigene Pack für Produktgrafiken, Rasterdateien für Fotos.
Icon-QA-Durchgang	Ein Gang durch die fertigen Bildschirme in einem hellen und einem dunklen Theme und bei mehr als einer Anzeigeskalierung, mit der Prüfung, dass jedes Icon noch sichtbar ist und noch eine Textalternative hat.

Selbst bauen – wählen Sie Ihren Assistenten

Beide Buttons öffnen einen neuen Tab, in dem **derselbe fertige Prompt** bereits eingetragen ist. Der Prompt verweist das Modell auf die Spezifikation dieses Moduls – eine einzelne Markdown-Datei mit der Lektion, der Lab-Aufgabe und den geforderten Ergebnissen –, sodass es mit dem echten Material arbeitet, statt zu raten, worum es in Modul 7 geht. Sie können [module-07.md](#) selbst öffnen, um genau zu sehen, was es lesen wird.



ChatGPT verwenden

Öffnet [chatgpt.com](#) mit dem Prompt dieses Moduls. Funktioniert mit GPT-5 oder



Claude verwenden

Öffnet [claude.ai](#) mit demselben Prompt. Wenn Sie Claude Code nutzen, können Sie

Codex; jeder Tarif mit Websuche ruft die Spezifikation ab.

ihn stattdessen in einem leeren Ordner in ein Terminal einfügen.



Unverändert abgeschickt, erhalten Sie ein Projekt, kein Code-Fragment

Der Prompt verlangt am Ende die gesamte Projektmappe als einzelne `.zip`-Datei mit Anleitung zum Kompilieren und Starten. Laden Sie sie herunter, öffnen Sie die Projektmappe in Visual Studio, drücken Sie F5, und IconDesk startet auf `localhost`. Sie benötigen das .NET SDK und das NuGet-Paket `Wisej-4`; alles andere soll der Assistent laut Prompt selbst schreiben.

Ich arbeite gerade Modul 7 – „Icon-Fonts, HTML-Inhalte und Icon-Strategie für den Produktivbetrieb“ – des LearnWisej-Kurses „Icons und Bilder in Wisej.NET“ durch und möchte das Beispielprojekt des Moduls gemeinsam mit dir erstellen. Bitte antworte auf Deutsch.

Lies zuerst die Modulspezifikation. Sie ist eine einzelne Markdown-Datei (auf Englisch) mit der vollständigen Lektion, der Lab-Aufgabe und den geforderten Ergebnissen:

<https://learnwisej.com/assets/courses/icons-and-images/specs/module-07.md>

Erstelle dann ein vollständiges Wisej.NET-4-Projekt, das ich in Visual Studio öffnen kann:

1. Projekt „IconDesk“, C#, net8.0 oder neuer, NuGet-Paket Wisej-4 4.1.x, Kestrel-Host, der `app.UseWisej()` aufruft.
2. Setze um, was das Lab der Spezifikation verlangt: Schließen Sie das Abschlussprojekt IconDesk ab. Referenzieren Sie ein Icon-Font-Stylesheet aus `Default.html`, setzen Sie `AllowHtml` auf einer kleinen Toolbar oder Karte und rendern Sie deren Icons als Font-Glyphen; bauen Sie dann eine `DataGridView`-Spalte, deren Zellen Icon-Font-Aktionen enthalten, und lesen Sie aus, welches Element angeklickt wurde, damit Edit und Delete Unterschiedliches tun. Bauen Sie eine Übersichtsseite, die alle Mechanismen des Kurses auf einmal zeigt: ein Theme-Bild, ein Icon aus einem offiziellen Pack, ein Icon aus dem eigenen Pack, ein eingebettetes Logo und ein umgefärbtes SVG. Schreiben Sie anschließend die Icon-Policy des Projekts für den Produktivbetrieb und führen Sie einen QA-Durchgang in einem hellen und einem dunklen Theme durch.
3. Liefere diese Ergebnisse und sag mir, wo sich jedes davon in der Projektmappe befindet:
 - Icon-Font-Stylesheet, das aus `Default.html` referenziert wird, und eine Toolbar oder Karte mit `AllowHtml`, die Font-Glyphen rendert
 - `DataGridView`-Spalte, deren Zellen Icon-Font-Aktionen rendern und auf das angeklickte Element reagieren
 - IconDesk-Übersichtsseite mit einem Theme-Bild, einem Icon aus einem offiziellen Pack, einem Icon aus dem eigenen Pack, einem eingebetteten Asset und einem umgefärbten SVG
 - Schriftliche Icon-Policy für den Produktivbetrieb zu Barrierefreiheit, Lizenzen, Caching und externen Abhängigkeiten
 - QA-Durchgang in einem hellen und einem dunklen Theme mit festgehaltenen Ergebnissen
4. Halte dich an die Spezifikation, wo immer sie eine Wisej.NET-Klasse, -Eigenschaft oder ein -Event nennt – verwende genau diese. Wo sie eine Wahl offenlässt, sag mir, welche API du gewählt hast und warum.
5. Der Server bleibt maßgeblich: Zustand und Validierung leben in .NET, der Browser stellt dar und meldet zurück. Verlagere nie eine Entscheidung nach JavaScript, nur weil es dort einfacher ist.
6. Mach das Ergebnis vorführbar: Ich muss es starten und das Verhalten sehen können, um das es in diesem Modul geht – auch, was passiert, wenn etwas schiefgeht.

Bevor du Code schreibst, fasse in eigenen Worten zusammen, was du bauen wirst und wo jedes Teil liegen wird, damit ich den Entwurf korrigieren kann, solange das noch wenig kostet.

Wenn der Code fertig ist, packe die gesamte Projektmappe in eine einzelne .zip-Datei zum Herunterladen und nenne mir die genauen Schritte, um sie zu kompilieren und auszuführen.

Beantworte zum Schluss diese Fragen aus der Spezifikation und nutze dabei deine eigene Implementierung als Beleg:

- Der Stylesheet-Link steht in `Default.html` und die Klassennamen stimmen, aber ein Label zeigt die rohen Tags, während der Button daneben das Glyph zeigt. Was unterscheidet die beiden Steuerelemente, und warum gilt die Einstellung pro Steuerelement und nicht pro

Anwendung?

- Ihre Aktionsspalte funktioniert, und ein Kollege möchte den Notiztext der Kundin in dieselbe Zelle setzen, damit er sich gestalten lässt. Was sagen Sie ihm, und was würden Sie stattdessen bauen?
- Welche der fünf Bilder auf der Übersichtsseite ändern sich beim Theme-Wechsel der Anwendung, und welche bleiben exakt so, wie sie gezeichnet wurden? Was sagt Ihnen das darüber, wo jedes davon gespeichert ist?

Frag mich nach allem, was unklar ist, bevor du anfängst.

Prüfen Sie dann, was zurückkommt

Code, der kompiliert, ist noch lange kein richtiger Code. Die spannenden Fragen sind Designfragen: Ist das wirklich das Wisej.NET-Control, das die Aufgabe verlangt? Folgt der Code dem Lebenszyklus, den das Framework vorgibt? Liegt der maßgebliche Zustand auf dem Server – oder wurde er im Browser abgelegt, weil das bequemer war? Lesen Sie das Ergebnis als Reviewer, nicht als Kunde.

Die Checkliste liefert Ihnen dieses Modul. Beantworten Sie jeden Punkt anhand des Codes, den Sie gerade erhalten haben:

- ✓ Der Stylesheet-Link steht in Default.html und die Klassennamen stimmen, aber ein Label zeigt die rohen Tags, während der Button daneben das Glyph zeigt. Was unterscheidet die beiden Steuerelemente, und warum gilt die Einstellung pro Steuerelement und nicht pro Anwendung?
- ✓ Ihre Aktionsspalte funktioniert, und ein Kollege möchte den Notiztext der Kundin in dieselbe Zelle setzen, damit er sich gestalten lässt. Was sagen Sie ihm, und was würden Sie stattdessen bauen?
- ✓ Welche der fünf Bilder auf der Übersichtsseite ändern sich beim Theme-Wechsel der Anwendung, und welche bleiben exakt so, wie sie gezeichnet wurden? Was sagt Ihnen das darüber, wo jedes davon gespeichert ist?

Alles, was Sie korrigieren mussten, sollten Sie festhalten. Eine kurze Notiz zu den Konventionen und Entscheidungen, auf die Sie sich festgelegt haben – eingefügt in die erste Nachricht Ihrer nächsten Sitzung –, ist mehr wert als jeder noch so clevere Prompt.



Für diesen Kurs gibt es noch kein Referenzbeispiel – Ihres ist das erste

Das [LearnWisej-Beispiel-Repository](#) enthält ausgearbeitete Builds für die meisten Kurse, für diesen aber noch nicht. Es gibt also nichts zum Vergleichen und keine Abkürzung: Die Spezifikation und Ihr eigenes Review sind das ganze Sicherheitsnetz. Behalten Sie, was Sie bauen – und wenn es gut wird, würden wir es gern sehen.

Drei Prompts für den nächsten Schritt

Richten Sie Ihren Assistenten auf das Projekt, das Sie gerade gebaut haben, und probieren Sie diese Prompts aus. Jeder ist eine Übung zum Prüfen und Erweitern, kein Neuschreiben.

- ✓ **Prüfen Sie es wie einen Pull-Request.** *"Führ mich durch diesen IconDesk-Code, als würdest du den Pull-Request eines Kollegen prüfen und hättest ihn nicht selbst geschrieben. Wo lebt der Zustand, was passiert bei einer einzelnen Benutzeraktion von Anfang bis Ende, was würdest du zur*

Überarbeitung zurückgeben, und welche Teile würdest du nicht guten Gewissens einem Kunden vorlegen? Sei konkret und zitiere die Zeilen, die du meinst – fasse mir nicht den Entwurf zusammen."

- ✓ **Tauschen Sie ein Teil aus, ohne die Schnittstellen zu verschieben.** "Wähle den Teil dieses IconDesk-Projekts, der am offensichtlichsten nur ein Platzhalter für etwas Echtes ist – eine Fake-Datenquelle, ein In-Memory-Speicher, ein vereinfachter Service – und ersetze ihn durch eine praxisnahe Implementierung. Der Vertrag auf beiden Seiten darf sich nicht ändern: dieselben Typen, dieselben Aufrufe, dasselbe Verhalten im Fehlerfall. Sag mir, was du durch die echte Implementierung behandeln musstest, was der Platzhalter nicht behandelt hat."
- ✓ **Machen Sie es fit für einen schlechten Tag.** "Untersuche dieses IconDesk-Projekt auf Fehlerszenarien, die es noch nicht behandelt – fehlerhafte oder fehlende Eingaben, eine Abhängigkeit, die langsam oder nicht erreichbar ist, Arbeit, die eintrifft, nachdem der Benutzer die Seite verlassen hat, und zwei Aktionen, die sich gegenseitig in die Quere kommen. Sag mir für jedes zuerst, ob es in Wisej.NET 4 ein echtes Risiko ist und warum, und schreib erst dann die Korrektur. Wo du entscheidest, dass nichts behandelt werden muss, sag das und begründe es, statt defensiven Code hinzuzufügen."



Nehmen Sie es mit ins Lab

Das praktische Lab am Ende dieses Moduls verlangt dieselben Ergebnisse, die Sie gerade vom Assistenten haben erstellen lassen. Nutzen Sie, was Sie hier haben, als Gerüst für Ihre Antwort und als Beleg dafür – nicht als die Antwort selbst. Bewertet wird im Lab Ihre Argumentation, und dieser Teil muss von Ihnen stammen.