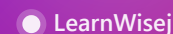


Thèmes et modernisation de l'interface

Thèmes et Theme Builder · Hiérarchie visuelle et finitions sans risque



Thèmes et Theme Builder

Appliquez un thème intégré, voyez la place des fichiers de thème et des options de style, et utilisez Theme Builder à un niveau débutant — sans toucher à la logique métier.



Objectif de la leçon

Appliquer un **thème** Wisej.NET intégré, comprendre la place des fichiers de thème et des options de style, et utiliser Theme Builder à un niveau pratique de débutant — le tout sans toucher à la logique métier.

1. Thème et logique métier

Le thème change *l'apparence* de l'application. La logique métier change *ce qu'elle fait*. Dans une application métier, vous devez pouvoir améliorer les couleurs, les espacements, les cartes, les en-têtes et la navigation sans réécrire les services, les repositories, les règles de validation ou les workflows des tickets.

Domaine	Décision de modernisation de l'interface	À ne pas réécrire
Thème	Choisir un thème intégré comme Bootstrap-4, BootstrapDark-4, Blue-1 ou Classic-2.	Service des tickets, logique d'enregistrement, règles de validation.
Espacement	Utiliser des marges internes, des marges et des espacements de cartes cohérents.	Classes du modèle de données ou méthodes du repository.
Regroupement	Réunir les commandes et informations liées dans des cartes / panneaux.	Workflow de navigation ou contrôles de permissions.
Libellés	Rendre clairs les titres, les libellés de champs et les textes d'aide.	Les calculs eux-mêmes ou les tickets enregistrés.
Commandes	Placer les boutons principaux là où les utilisateurs les attendent ; garder des noms prévisibles.	Les règles métier des gestionnaires d'événements.

2. Fichiers de thème, polices, images et options de style

- Un thème donne à l'application un style de base cohérent pour les contrôles courants.
- Les polices doivent être lisibles et cohérentes — évitez de multiplier les tailles ou les familles.
- Les images et icônes doivent servir le workflow, pas ajouter une décoration qui déroute l'utilisateur.
- Les options de style doivent améliorer la clarté : état actif de la navigation, titres de section clairs, boutons lisibles.

- Gardez un seul système visuel — mêmes tailles de boutons, marges internes des cartes, modèle de libellés et rôle des couleurs sur tous les écrans.

3. Le principe de Theme Builder

Theme Builder sert quand les thèmes intégrés sont proches mais pas exactement adaptés au projet. Un débutant doit d'abord appliquer un thème intégré, puis faire de petits ajustements pratiques — couleur principale, arrière-plans, taille des polices, espacement des contrôles. Ne commencez pas par retoucher chaque écran à la main.



Commencez par un thème intégré

Choisissez d'abord un thème intégré et ajustez quelques valeurs. N'utilisez Theme Builder que lorsque le thème de base ne permet pas d'aller jusqu'au bout.

4. Une structure d'écran moderne

Un écran métier moderne comporte généralement un en-tête clair, une navigation à gauche, une zone de contenu principale, des cartes regroupées et des commandes placées de façon prévisible, pour que les utilisateurs sachent toujours où ils se trouvent et quoi faire ensuite.

En-tête	→ titre, libellé du module, sélecteur du thème courant
Navigation	→ Dashboard · Tickets · Customers · Reports · Settings
Zone de contenu	→ titre de la page + courte description
	→ une rangée de cartes d'indicateurs (grands nombres)
	→ commandes regroupées activité récente



Vous êtes prêt pour la leçon suivante

Vous savez séparer le thème de la logique métier, appliquer un thème intégré et décrire une structure d'écran moderne. Ensuite : les règles de hiérarchie visuelle et la façon de peaufiner un écran sans casser ce qui fonctionne.

Hiérarchie visuelle et finitions sans risque

Les règles de hiérarchie visuelle qui rendent les écrans faciles à parcourir, et comment peaufiner la mise en page sans casser la liaison, la navigation ou la validation.



Objectif de la leçon

Appliquer des règles de hiérarchie visuelle pour rendre les écrans faciles à parcourir, puis moderniser la mise en page **sans** casser la liaison de données, la navigation, la validation ou les règles métier.

1. Règles de hiérarchie visuelle

Règle	Ce que cela signifie	Exemple dans le module 7
Rendre le titre de la page évident	L'utilisateur doit savoir immédiatement sur quel écran il se trouve.	Titres Dashboard, Tickets, Customers, Reports, Settings.
Regrouper le contenu lié	Utiliser des panneaux / cartes pour que les enregistrements et commandes liés restent ensemble.	Cartes d'indicateurs, carte de la file de tickets, carte des réglages d'interface.
Garder des commandes prévisibles	Placer les boutons près du contenu sur lequel ils agissent.	Create / Edit / Close Ticket sous la file de tickets.
Utiliser des espacements cohérents	Les écrans doivent donner l'impression d'appartenir à la même application.	Même marge gauche, mêmes largeurs de cartes et même espacement des lignes sur toutes les pages.
Montrer l'état actif de la navigation	Le bouton de navigation sélectionné doit avoir l'air sélectionné.	Bouton vert sélectionné, avec une bordure, sur la page courante.

2. Peaufiner sans réécrire la logique

- Renommez clairement les contrôles si nécessaire, mais évitez de casser les gestionnaires d'événements existants.
- Déplacez les contrôles liés dans des panneaux ou des cartes sans modifier la couche de service.
- Utilisez **Dock**, **Anchor** et **AutoSize** délibérément pour que l'écran reste utilisable lors d'un redimensionnement.
- Conservez la même source de données et la même logique de liaison lorsque vous modernisez grilles, listes et panneaux de détail.
- Testez chaque bouton de navigation et chaque workflow une fois l'interface peaufinée.



Erreur de débutant à éviter

Ne cachez pas une logique défailante derrière un thème plus joli. Une interface moderne a toujours besoin d'une navigation qui fonctionne, de validation, d'un comportement Save/Cancel correct, d'erreurs sans risque et de messages d'état clairs.

3. Garder les réglages de thème hors de la logique métier

Un sélecteur de thème est un réglage d'interface — gardez-le séparé du service des tickets. Appliquez le thème et mettez à jour un petit libellé d'état ; ne le mêlez pas au code d'enregistrement ou de validation.

```
private void cboTheme_SelectedIndexChanged(object sender, EventArgs e)
{
    string selectedTheme = cboTheme.Text;

    // Apply the selected theme – keep this UI setting separate
    // from ticket service logic.
    Application.Theme.Name = selectedTheme;
    lblCurrentTheme.Text = "Current theme: " + selectedTheme;
}
```

4. Point d'étape du module 7

- Savez-vous expliquer ce que change le thème ?
- Savez-vous désigner la logique métier qui ne doit *pas* changer ?
- Savez-vous identifier la page courante à partir de l'état de la navigation ?
- Savez-vous expliquer pourquoi les cartes, les libellés et les espacements rendent l'application plus facile à utiliser ?



Vous êtes prêt pour le lab

Ensuite : regardez la vidéo pas à pas sur la modernisation, réussissez la vérification des connaissances, puis appliquez un thème au tableau de bord et modernisez-le — thème intégré, indicateurs regroupés en cartes, état actif de la navigation visible, et tous les workflows existants toujours fonctionnels.