Module 8: Application de styles aux applications Web ASP.NET MVC 5

# **Contenu:**

|  |  |
| --- | --- |
|  | [Aperçu du module](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C08.html#P1) |
| **Leçon 1:** | [Utilisation des mises en page](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C08.html#P2) |
| **Leçon 2:** | [Application de styles CSS à une application MVC](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C08.html#P3) |
| **Lecon 3:** | [Création d'une interface utilisateur adaptative](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C08.html#P4) |
| **Laboratoire:** | [Application de styles aux applications Web MVC 5](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C08.html#P5) |
|  | [Examen du module et points à retenir](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C08.html#P6) |

# **Aperçu du module**

Lors de la création d'applications Web, vous devez appliquer une apparence cohérente à l'application. Vous devez inclure des sections d'en-tête et de pied de page cohérentes dans toutes les vues. ASP.NET MVC 5 inclut des fonctionnalités telles que des styles et des mises en page de feuilles de style en cascade (CSS) qui améliorent l'apparence et la convivialité de votre application Web. ASP.NET MVC 5 comprend également des fonctionnalités telles que des vues spécifiques aux mobiles qui vous permettent de créer des applications pour différents navigateurs ou appareils mobiles.

### **Objectifs**

Après avoir terminé ce module, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Appliquez une mise en page cohérente aux applications ASP.NET MVC 5 à l'aide de CSS et d'une mise en page. |
| • | Développez des vues spécifiques à l'appareil pour prendre en charge divers navigateurs. |
| • | Créez une interface utilisateur adaptative pour les appareils mobiles. |

# Leçon 1: Utilisation des mises en page

Vous devez créer plusieurs vues pour prendre en charge les opérations de l'application, telles que la création d'une commande et l'interrogation de l'historique des commandes. Cependant, plusieurs problèmes de maintenance surviennent lors de la modification de la partie commune de la présentation de l'application, à cause desquels vous devez mettre à jour chaque vue. Pour résoudre ces problèmes de maintenance, vous pouvez créer un module commun ou une vue partagée. Une vue partagée qui aide à stocker la logique de l'application est appelée une présentation. ASP.NET MVC 5 comprend des fonctionnalités qui simplifient le processus de création et d'utilisation de mises en page. Vous pouvez simplifier davantage le processus de gestion des applications en utilisant le fichier \_ViewStart, pour appliquer la mise en page à chaque vue, au lieu de modifier individuellement chaque vue.

## **Objectifs de la leçon**

Après avoir terminé cette leçon, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Décrivez les mises en page. |
| • | Décrivez comment utiliser les mises en page. |
| • | Décrivez le fichier \_ViewStart. |

## **Que sont les mises en page?**

Le moteur ASP.NET MVC 5 Razor comprend une fonctionnalité appelée dispositions. Les mises en page sont également appelées vues de modèle. Les mises en page vous permettent de définir un modèle de style commun, puis de l'appliquer à toutes les vues d'une application Web. La fonctionnalité des mises en page est similaire à celle de la page maître dans une application Web ASP.NET traditionnelle. Vous pouvez utiliser des mises en page pour définir la mise en page ou la logique du contenu qui est partagée entre les vues.

Vous pouvez définir plusieurs mises en page dans une application ASP.NET MVC 5, et chaque mise en page peut avoir plusieurs sections. Vous pouvez définir ces sections n'importe où dans le fichier de mise en page, même dans la section <head> du HTML. Les sections vous permettent de générer du contenu dynamique vers plusieurs régions non contiguës de la réponse finale.

**Question**: Quels sont les scénarios courants dans lesquels vous utiliseriez des mises en page?

## **Créer une mise en page**

Lors de la création de mises en page, vous devez stocker les fichiers de mise en page dans le dossier \ Views \ Shared du projet. Le dossier \ Views \ Shared est l'emplacement par défaut, dans lequel vous pouvez stocker des fichiers ou des modèles de vue communs.

**Une vue de mise en page**

<! DOCTYPE html>

<html>

<head>

<meta name = "viewport" content = "width = device-width" />

<title> @ ViewBag.Title </title>

</head>

<body>

<div>

@RenderBody ()

</div>

</body>

</ html

Dans l'exemple précédent, la méthode @RenderBody () indique au moteur de rendu où va le contenu de la vue.

**ViewBag**est un objet que la mise en page et la vue partagent. Vous pouvez utiliser l'objet ViewBag pour transmettre des informations entre une vue et une mise en page. Pour transmettre des informations, vous devez ajouter une propriété à l'objet ViewBag, dans la page ViewPage du fichier ViewController ou View, et utiliser la même propriété dans le fichier de disposition. Les propriétés vous aident à contrôler le contenu de la mise en page, pour rendre dynamiquement les pages Web à partir du code dans le fichier de vue. Par exemple, considérez que le modèle utilise la propriété ViewBag.Title pour restituer le contenu <title> dans la vue. Cette propriété permet de définir la propriété Title de l'objet ViewBag dans la vue et de récupérer la propriété dans la mise en page. Cette récupération est possible car le code du fichier de vue s'exécute avant l'exécution de la mise en page.

**Utilisation de sections dans les mises en page**

<! DOCTYPE html>

<html>

<head>

<meta name = "viewport" content = "width = device-width" />

<title> @ ViewBag.Title </title>

</head>

<body>

<div id = "menu">

@RenderSection ("MenuBar", obligatoire: false)

</div>

<div>

@RenderBody ()

</div>

</body>

</html>

Le paramètre MenuBar dans la méthode d'assistance RenderSection () spécifie le nom de la section que vous souhaitez rendre dans la disposition. Le paramètre requis est facultatif; il vous permet de déterminer si la section que vous affichez est requise. Considérez qu'une section est requise et que vous n'implémentez pas la section dans le fichier de mise en page. Dans ce cas, ASP.NET MVC 5 affiche l'exception de section non définie au moment de l'exécution. L'implémentation de sections dans un fichier de mise en page facilite le suivi des erreurs de contenu. Si une section n'est pas requise, vous pouvez choisir de ne pas l'inclure dans la mise en page.

**Question**: Pourquoi avez-vous plusieurs sections dans une mise en page?

## **Lier des vues et des mises en page**

Après avoir défini la mise en page, vous devez lier la mise en page aux fichiers de vue. Vous devez d'abord supprimer le contenu qui n'est plus nécessaire dans la vue. Ensuite, vous devez créer le lien entre la vue et la mise en page, afin que le contenu supprimé de la vue ne soit pas reflété dans la mise en page.

**Lier à une vue**

@ {

ViewBag.Title = "Détails";

Layout = "~ / Views / Shared / SiteLayout.cshtml";

}

<h2> Détails </h2>

Vous pouvez utiliser la propriété ViewBag.Title pour transmettre des informations de titre de page de la vue à la mise en page. Vous pouvez définir d'autres propriétés avec l'objet ViewBag, tels que les éléments <meta> dans la section <head>, et leur permettre de transmettre des informations à la mise en page.

**Utilisation de la directive @section**

@ {

ViewBag.Title = "Détails";

Layout = "~ / Views / Shared / SiteLayout.cshtml";

}

<h2> Détails </h2>

@section MenuBar {

<p> c'est le menu </p>

}

Dans l'exemple précédent, vous définissez le fichier de disposition pour afficher des sections en haut de chaque fichier de vue. Habituellement, vous avez la même mise en page dans l'ensemble de l'application ou de la section Web. Vous pouvez définir la disposition d'une application ou d'une section à l'aide du fichier \_ViewStart. Pendant l'exécution, le code du fichier \_ViewStart s'exécute avant toutes les autres vues de l'application Web. Par conséquent, vous pouvez placer toutes les logiques d'application courantes dans le fichier \_ViewStart.

**Le fichier \_ViewStart**

@ {

Layout = "~ / Views / Shared / SiteLayout.cshtml";

}

**Question**: Quand devez-vous utiliser le fichier \_Viewstart?

# Leçon 2: Appliquer des styles CSS à une application MVC

Les feuilles de style en cascade (CSS) sont une norme de l'industrie pour l'application de styles aux pages HTML. Différentes méthodes d'application de CSS à une page Web sont disponibles. Ces méthodes incluent le fichier CSS externe, le CSS en ligne et le bloc de code CSS en HTML. Les développeurs utilisent généralement un fichier CSS externe, car ce fichier est partagé sur plusieurs pages et permet d'appliquer un style cohérent dans toute l'application. Vous devez savoir comment importer des styles dans une application Web, pour garantir la cohérence de l'apparence de l'application.

## **Objectifs de la leçon**

Après avoir terminé cette leçon, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Importez des styles dans une application Web ASP.NET MVC 5. |
| • | Appliquez une apparence cohérente à une application Web MVC 5. |

## **Importation de styles dans une application Web MVC**

Après avoir créé des styles CSS, vous devez importer ces styles dans l'application Web. Après avoir importé le fichier CSS dans l'application Web, vous devez modifier la mise en page de l'application Web afin de pouvoir utiliser les styles CSS que vous avez importés. Vous pouvez modifier la mise en page d'une application Web à l'aide de l'élément <link>.

**Lier à une feuille de style**

<! DOCTYPE html>

<html>

<head>

<meta name = "viewport" content = "width = device-width" />

<title> @ ViewBag.Title </title>

<link href = "~ / Views / Shared / StyleSheet1.css" rel = "stylesheet" type = "text / css" />

</head>

<body>

<div id = "menu">

@RenderSection ("MenuBar", obligatoire: false)

</div>

<div>

@RenderBody ()

</div>

</body>

</html>

Les sélecteurs CSS aident les navigateurs à déterminer comment les styles CSS doivent être appliqués. Vous pouvez utiliser divers sélecteurs, tels que des sélecteurs de classe et d'identifiant, pour appliquer des styles aux éléments HTML.

**Sélecteur de classe CSS**

Vous pouvez définir un sélecteur de classe CSS pour spécifier un style pour un groupe d'éléments. Pour appliquer le sélecteur de classe à un élément HTML, vous devez ajouter l'attribut class à l'élément HTML. Vous pouvez utiliser la syntaxe. <classe> pour ajouter le style dans le fichier CSS.

**Utiliser une classe**

@ {

ViewBag.Title = "Détails";

Layout = "~ / Views / Shared / SiteLayout.cshtml";

}

<h2> Détails </h2>

@section MenuBar {

<p class = ”menu”> c'est le menu </p>

}

**Application d'un style à une classe**

.menu

{

poids de la police: gras;

}

**Sélecteur d'identifiant CSS**

Vous pouvez utiliser le sélecteur d'identifiant CSS pour spécifier un style pour tout élément unique de votre code HTML. Pour appliquer le sélecteur d'identifiant à un élément HTML, vous devez ajouter l'attribut id et un nom unique à l'élément HTML. Vous pouvez utiliser la syntaxe # <id> pour ajouter le style dans le fichier CSS.

**Utilisation du sélecteur d'identifiant**

@ {

ViewBag.Title = "Détails";

Layout = "~ / Views / Shared / SiteLayout.cshtml";

}

<h2> Détails </h2>

@section MenuBar {

<p id = "leftmenu"> c'est le menu </p>

}

**Création d'un style d'identification**

#leftmenu

{

taille de la police: 16px;

}

**Question**: Quels sont les scénarios courants dans lesquels vous utiliseriez le sélecteur de classe? Quels sont les scénarios courants dans lesquels vous utiliseriez le sélecteur d'identifiant?

## **Démonstration: comment appliquer une apparence et une sensation cohérentes**

Dans cette démonstration, vous verrez comment:

|  |  |
| --- | --- |
| • | Créez une mise en page. |
| • | Appliquez la mise en page à une vue MVC. |

### **Étapes de démonstration**

|  |  |
| --- | --- |
| • | Vous trouverez les étapes dans la section «Leçon 2: Appliquer des styles CSS à une application MVC» sur la page suivante:<https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD08_DEMO.md>. |

# Leçon 3: Création d'une interface utilisateur adaptative

Les applications ASP.NET MVC 5 facilitent l'interface utilisateur adaptative pour rendre le contenu sur différents appareils. L'interface utilisateur adaptative est un type d'interface utilisateur qui restitue le contenu en fonction de la capacité du navigateur Web ou de l'appareil cible. Vous devez vous assurer que votre application prend en charge les appareils mobiles, afin qu'elle atteigne tous les types d'audience. Vous devez également savoir comment utiliser les requêtes multimédias et les vues spécifiques aux mobiles pour vous assurer que votre application est efficace sur les appareils mobiles.

## **Objectifs de la leçon**

Après avoir terminé cette leçon, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Décrivez l'attribut de la fenêtre HTML5. |
| • | Expliquez comment les requêtes multimédias CSS appliquent des CSS spécifiques en fonction des capacités du navigateur. |
| • | Expliquez comment vous pouvez utiliser les modèles MVC 5 pour rendre des vues en fonction de la taille de l'écran de l'appareil mobile. |

## **L'attribut HTML5 Viewport**

Le rendu adaptatif vous permet de personnaliser votre application Web pour qu'elle s'affiche différemment, en fonction des capacités du navigateur Web ou de l'appareil.

Les navigateurs mobiles tels qu'Internet Explorer utilisent l'attribut viewport pour afficher les pages Web dans une fenêtre virtuelle. Cette fenêtre virtuelle est généralement plus large que l'écran de l'application. L'attribut viewport permet d'éliminer le besoin de réduire la taille de la mise en page de chaque page. La réduction de la taille de la mise en page peut interrompre ou déformer l'affichage des applications Web non optimisées pour les mobiles. La création de l'interface d'application à l'aide de l'attribut viewport permet aux utilisateurs de zoomer sur les différentes zones d'une page Web.

**Lecture supplémentaire:**Pour plus d'informations sur l'attribut viewport, visitez:<http://go.microsoft.com/fwlink/?LinkID=288967&clcid=0x409>

La balise viewport est une balise Meta qui permet de contrôler la largeur et la hauteur des pages Web, lors du rendu dans les navigateurs Web.

**Utilisation de la balise Viewport**

<meta name = "viewport" content = "width = device-width, initial-scale = 1, maximum-scale = 1">

Les propriétés de largeur et de hauteur aident à spécifier la largeur et la hauteur de la fenêtre de la fenêtre virtuelle. Vous pouvez spécifier la largeur en pixels. Vous pouvez utiliser le mot-clé largeur-appareil pour permettre au contenu de s'adapter à la taille d'écran native du navigateur.

La propriété d'échelle initiale contrôle l'échelle initiale ou le niveau de zoom de la page Web. Les propriétés d'échelle maximale, d'échelle minimale et évolutive par l'utilisateur contrôlent les autres fonctionnalités d'évolutivité de la page Web.

**Question**: Comment contrôler la taille de la fenêtre de la fenêtre virtuelle?

## **Requêtes multimédias CSS**

Il se peut que vous deviez parfois appliquer différents styles CSS dans votre application, pour prendre en charge différents navigateurs. HTML5 inclut les requêtes multimédias CSS, qui sont des sélecteurs spéciaux commençant par @media. Les requêtes multimédias permettent l'application conditionnelle de styles CSS, en fonction des conditions de l'appareil ou des capacités du navigateur. Vous pouvez appliquer des requêtes multimédias en CSS et HTML.

**Utilisation d'une requête multimédia**

@media uniquement écran et (max-width: 500px) {

entête{

float: aucun;

}

}

**Utilisation d'une requête multimédia dans l'élément de lien**

<link rel = "stylesheet" type = "text / css" href = "smallscreen.css" media = "uniquement écran et (max-width: 500px)" />

Vous pouvez utiliser des requêtes multimédias CSS pour appliquer des styles CSS lorsque la taille de l'écran est inférieure à 500 pixels. Cependant, vous pouvez utiliser les requêtes multimédias CSS uniquement pour la mise en page de l'écran, mais pas pour la mise en page d'impression.

Le tableau suivant décrit les propriétés que vous pouvez inclure dans une requête multimédia.

| **Biens** | **La description** |
| --- | --- |
| **largeur** | La largeur de la zone d'affichage ciblée, qui comprend la fenêtre du navigateur sur les ordinateurs de bureau et les appareils mobiles. Sur les ordinateurs de bureau, lorsque vous redimensionnez la fenêtre du navigateur, la largeur du navigateur change. Cependant, sur la plupart des navigateurs mobiles, vous ne pouvez pas redimensionner la fenêtre du navigateur. Cela implique que la largeur du navigateur reste constante. |
| **la taille** | La hauteur de la zone d'affichage ciblée, qui comprend la fenêtre du navigateur sur les ordinateurs de bureau et les appareils mobiles. |
| **largeur de l'appareil** | La largeur de l'écran entier d'un appareil. Pour un ordinateur de bureau avec une résolution d'écran de 1 024 x 768, la largeur de l'appareil est généralement de 1 024 pixels. |
| **hauteur de l'appareil** | La hauteur de l'écran entier d'un appareil. Pour un bureau avec une résolution d'écran de 1 024 x 768, la hauteur de l'appareil est généralement de 768 pixels. |
| **orientation** | L'orientation de l'appareil. Si la largeur de l'appareil est supérieure à la hauteur de l'appareil, la valeur d'orientation est définie sur paysage; sinon, il est défini sur portrait. |
| **ratio d'aspect** | Le rapport des propriétés de largeur et de hauteur. |
| **rapport d'aspect de l'appareil** | Le rapport des propriétés de largeur de périphérique et de hauteur de périphérique. L'exemple suivant illustre le rapport hauteur / largeur de l'appareil pour un appareil avec une résolution d'écran de 1 280 x 720.  @ écran multimédia et (ratio d'aspect de l'appareil: 16/9) {}  @ écran multimédia et (rapport hauteur / largeur de l'appareil: 1280/720) {}  @ écran multimédia et (ratio d'aspect de l'appareil: 2560/1440) {} |
| **Couleur** | Le nombre de bits par composant de couleur du périphérique. Si le périphérique n'est pas un périphérique couleur, la valeur est zéro. |
| **indice de couleur** | Le nombre d'entrées dans la table de recherche de couleurs, du périphérique de sortie. |
| **monochrome** | Le nombre de bits par pixel dans un tampon de trame monochrome. Pour les périphériques non monochromes, cette valeur est égale à zéro. |
| **résolution** | La résolution du périphérique de sortie ou la densité des pixels. Les unités communes pour cette propriété incluent le dpi et le dpcm. |
| **analyse** | Le processus de numérisation des périphériques de sortie TV. |
| **la grille** | Propriété qui détecte si la sortie est au format grille ou bitmap. Les périphériques basés sur la grille renvoient une valeur de un; tous les autres appareils renvoient une valeur de zéro. |

**Question**: Pourquoi choisiriez-vous d'utiliser des requêtes multimédias CSS, au lieu d'utiliser du code C #, pour définir des styles pour des navigateurs spécifiques?

## **Modèles MVC 5 et vues spécifiques aux mobiles**

ASP.NET MVC 5 comprend deux fonctionnalités, le mode d'affichage mobile et le mode d'affichage personnalisé, qui vous aident à créer des pages Web pour les appareils mobiles et différents navigateurs.

ASP.NET MVC 5 vous permet de remplacer les vues pour les appareils mobiles en utilisant un ensemble différent de fichiers d'affichage, plutôt qu'en utilisant une configuration. Lorsque ASP.NET MVC 5 reçoit une demande d'un navigateur mobile, il analyse la demande de vues avec la convention de dénomination [view] .mobile.cshtml. Si ASP.NET MVC 5 détecte une vue avec la convention de dénomination mentionnée, ASP.NET MVC 5 servira la demande en utilisant la version mobile de la vue; sinon, il renvoie la demande à la vue standard.

Considérez que votre application Web comprend une mise en page spécifique à un navigateur. Dans ce cas, vous pouvez créer des vues spécifiques au navigateur pour ce navigateur en vérifiant la chaîne UserAgent de ce navigateur. La chaîne UserAgent aide à identifier un navigateur.

**Question**: Pourquoi choisiriez-vous des modes d'affichage spécifiques à l'appareil plutôt que des requêtes multimédias CSS?

# **Atelier: Application de styles aux applications Web MVC 5**

### **Scénario**

Vous avez créé une bonne partie de la fonctionnalité de gestion des photos pour l'application Web Partage de photos. Cependant, les parties prenantes sont préoccupées par l'apparence de base en noir et blanc de l'application. De plus, ils souhaitent que les titres et les menus apparaissent sur chaque page.

Pour résoudre ces problèmes, votre responsable vous a demandé de mettre en œuvre les fonctionnalités d'interface utilisateur suivantes:

|  |  |
| --- | --- |
| • | Une mise en page pour toutes les pages Web. La mise en page doit inclure des éléments communs, tels que le menu principal et les commandes de fil d'Ariane, qui doivent apparaître sur chaque page de l'application. |
| • | Une feuille de style et des images pour toutes les pages Web. L'équipe de conception Web a fourni une application de maquette HTML pour montrer à quoi devrait ressembler le produit final. Cette maquette comprend une feuille de style et des fichiers image. Vous devez importer ces fichiers et les appliquer à chaque page de l'application. |
| • | Une vue spécifique aux mobiles. L'application Web doit être accessible à partir d'appareils mobiles tels que les téléphones portables et les tablettes. En particulier, vous devez vous assurer que les appareils dotés d'écrans étroits peuvent accéder facilement aux photos. |

### **Objectifs**

Après avoir terminé cet atelier, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Appliquez une apparence cohérente à l'application Web. |
| • | Utilisez des mises en page pour vous assurer que les fonctionnalités d'interface communes, telles que les en-têtes, sont cohérentes dans l'ensemble de l'application Web. |
| • | Assurez-vous que l'application Web s'affiche correctement sur des écrans de différentes tailles et proportions. |

##### **Configuration du laboratoire**

Durée estimée: 60 minutes

Vous trouverez les étapes de haut niveau sur la page suivante:<https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD08_LAB_MANUAL.md>.

Vous trouverez les étapes détaillées sur la page suivante:<https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD08_LAK.md>.

### **Exercice 1: Création et application de mises en page**

##### **Scénario**

Dans cet exercice, vous allez:

|  |  |
| --- | --- |
| • | Parcourez l'application Web de partage de photos sans mise en page appliquée. |
| • | Créez une nouvelle disposition et liez l'application à la vue à l'aide d'un fichier \_ViewStart.cshtml. |
| • | Modifiez l'index d'accueil et les vues d'affichage des photos pour utiliser la nouvelle mise en page. |
| • | Parcourez l'application Web résultante. |

### **Exercice 2: Application de styles à une application Web MVC**

##### **Scénario**

Dans cet exercice, vous allez:

|  |  |
| --- | --- |
| • | Examinez une application Web de maquette qui montre l'aspect et la convivialité que les concepteurs Web ont créés pour l'application de partage de photos. |
| • | Importez une feuille de style, avec les fichiers graphiques associés de l'application de maquette, dans votre application Web, puis mettez à jour les classes d'éléments HTML pour appliquer ces styles aux éléments dans les vues. |
| • | Examinez les modifications apportées à l'interface utilisateur une fois les styles appliqués. |

### **Exercice 3: Facultatif - Adaptation de pages Web pour les navigateurs mobiles**

##### **Scénario**

Dans cet exercice, vous allez:

|  |  |
| --- | --- |
| • | Créez une nouvelle mise en page pour les appareils mobiles. |
| • | Ajoutez une requête multimédia à la feuille de style de l'application Web pour vous assurer que l'index des photos s'affiche sur de petits écrans. |
| • | Testez les paramètres appliqués à l'application à l'aide d'un petit navigateur et en modifiant la chaîne de l'agent utilisateur. |

Effectuez cet exercice si le temps le permet.

### **Question (s) de révision**

**Vérifiez vos connaissances**

**Découverte**

**Lorsque vous avez parcouru l'application Web pour la première fois dans l'exercice 1, pourquoi le menu et le fil d'Ariane étaient-ils visibles sur la page d'accueil, mais pas sur la page Toutes les photos ou sur toute autre page?**

Afficher la solution Réinitialiser

**Vérifiez vos connaissances**

**Découverte**

**Lorsque vous avez consulté le site pour la première fois en tant que navigateur mobile dans l'exercice 3, quels sont les problèmes que vous avez rencontrés avec l'affichage de l'en-tête et du menu du site?**

Afficher la solution Réinitialiser

# **Examen du module et points à retenir**

Dans ce module, vous avez appris à appliquer une apparence cohérente à une application Web et à partager d'autres composants communs, tels que des en-têtes et des pieds de page, entre toutes les vues. Vous avez également appris à utiliser le CSS et les modes d'affichage pour adapter l'application Web aux petits écrans et aux appareils mobiles. Vous vous êtes également familiarisé avec les éléments HTML5 qui vous permettent de développer des applications Web fonctionnant sur divers navigateurs et appareils.

### **Problèmes et scénarios du monde réel**

Lorsque vous développez des applications Web, vous devez créer des applications qui fonctionnent sur différents appareils et navigateurs, tels que iPhone, iPad, Google Chrome, Microsoft Edge et Internet Explorer. Dans de tels cas, vous pouvez utiliser les éléments et fonctionnalités HTML5 de MVC 5, tels que les vues spécifiques aux mobiles et les requêtes multimédias pour créer des applications qui fonctionnent bien dans divers navigateurs et appareils.

### **Question (s) de révision**