# Objectifs

* Affichage des données des snows dans des cards.
* Affichage du détail d’un article sélectionné dans une vue snow\_details.php

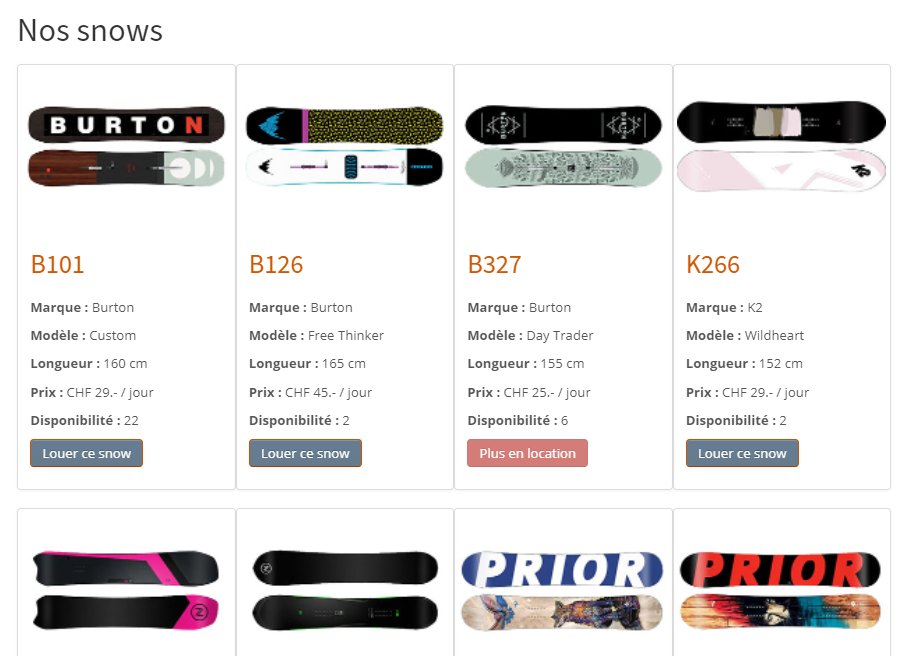
# PARTIE 2

## 1) Affichage sous forme de cards

### Affichage de la page snows pour le client du site

Important : Faites une copie de votre vue snows.php actuelle et renommez la snows\_admin.php. Nous utiliserons ultérieurement cette vue sous forme tabulaire pour la partie CRUD accessible par les vendeurs et administrateurs du site.

Modifiez la page snows.php pour afficher le contenu des snows dans des cards comme ci-dessus. Utilisez la page portfolio.html du gabarit fourni comme base pour créer les cards du snow.



## 2) Affichage du détail d’un snow

### Affichage de la page snow\_detail.php



Sur la page précédente (snows.php), lorsque nous cliquons sur l’id du snow (B101 par exemple), nous sommes redirigés vers la vue snows\_detail.php qui affiche le détail du snow. Le lien devra donc faire passer en querystring l’id du snow.

**<a href="index.php?action=displayASnow&code=B101">**

**L’exemple de lien ci-dessus devra être généré de façon dynamique, en modifiant la valeur de l’id.**

La page index.php devra donc également être modifiée pour récupérer ces valeurs :

**case 'displayASnow' :  
 displayASnow($\_GET['code']);  
 break;**

Le controleur devra aussi être modifié en y ajoutant la fonction displayASnow.

Créez cette fonction et ajoutez-y l’appel à la méthode du modèle getASnow

**function getASnow($snow\_code){**

**$strgSeparator = '\'';  
  
$snowQuery = 'SELECT code, brand, model, snowLength, dailyPrice, qtyAvailable, description, photo FROM snows WHERE code='.$strgSeparator.$snow\_code.$strgSeparator.'AND active=1';  
  
   
$snowResults = executeQuerySelect($snowQuery);  
  
return $snowResults;**

**}**