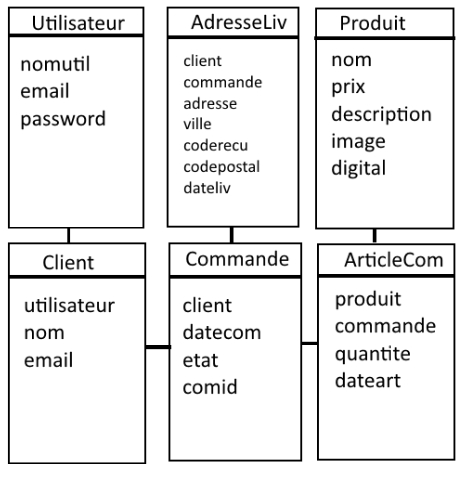
Modèles



Importer le modèle Utilisateur : boutique>models.py

*from* django.db *import* models

*from* django.contrib.auth.models *import* User

class Client(models.Model):

    utilisateur = models.OneToOneField(User, on\_delete=models.CASCADE, null=True, blank=True)

    nom = models.CharField(max\_length=200, null=True)

    email = models.CharField(max\_length=200, null=True)

    def \_\_str\_\_(self):

*return* self.nom

class Produit(models.Model):

    nom = models.CharField(max\_length=200, null=True)

    prix = models.DecimalField(max\_digits=7, decimal\_places=2)

    description = models.CharField(max\_length=1000, null=True)

 digital = models.BooleanField(default=False, null=True, blank=False)

    def \_\_str\_\_(self):

*return* self.nom

class Commande(models.Model):

    client = models.ForeignKey(Client, on\_delete=models.SET\_NULL, null=True, blank=True)

    datecom = models.DateTimeField(auto\_now\_add=True)

    etat = models.BooleanField(default=False, null=True, blank=False)

    transactionid = models.CharField(max\_length=200, null=True)

    def \_\_str\_\_(self):

*return* str(self.id)

class ArticleCom(models.Model):

    produit = models.ForeignKey(Produit, on\_delete=models.SET\_NULL, null=True, blank=True)

    commande = models.ForeignKey(Commande, on\_delete=models.SET\_NULL, null=True, blank=True)

    quantite = models.IntegerField(default=0, null=True, blank=True)

    dateart = models.DateTimeField(auto\_now\_add=True)

class AdresseLiv(models.Model):

    client = models.ForeignKey(Client, on\_delete=models.SET\_NULL, null=True, blank=True)

    commande = models.ForeignKey(Commande, on\_delete=models.SET\_NULL, null=True, blank=True)

    adresse = models.CharField(max\_length=200, null=True)

    ville = models.CharField(max\_length=200, null=True)

    coderecu = models.CharField(max\_length=200, null=True)

    codepostal = models.CharField(max\_length=200, null=True)

    dateliv = models.DateTimeField(auto\_now\_add=True)

    def \_\_str\_\_(self):

*return* self.adresse

la migration de la base de donnée

$ python manage.py makemigrations

Migrations for 'boutique':

boutique\migrations\0001\_initial.py

- Create model Client

- Create model Produit

- Create model Commande

- Create model ArticleCom

- Create model AdresseLiv

$ python manage.py migrate

Operations to perform:

Apply all migrations: admin, auth, boutique, contenttypes, sessions

Running migrations:

Applying contenttypes.0001\_initial... OK

Applying auth.0001\_initial... OK

Applying admin.0001\_initial... OK

Applying admin.0002\_logentry\_remove\_auto\_add... OK

Applying admin.0003\_logentry\_add\_action\_flag\_choices... OK

Applying contenttypes.0002\_remove\_content\_type\_name... OK

Applying auth.0002\_alter\_permission\_name\_max\_length... OK

Applying auth.0003\_alter\_user\_email\_max\_length... OK

Applying auth.0004\_alter\_user\_username\_opts... OK

Applying auth.0005\_alter\_user\_last\_login\_null... OK

Applying auth.0006\_require\_contenttypes\_0002... OK

Applying auth.0007\_alter\_validators\_add\_error\_messages... OK

Applying auth.0008\_alter\_user\_username\_max\_length... OK

Applying auth.0009\_alter\_user\_last\_name\_max\_length... OK

Applying auth.0010\_alter\_group\_name\_max\_length... OK

Applying auth.0011\_update\_proxy\_permissions... OK

Applying auth.0012\_alter\_user\_first\_name\_max\_length... OK

Applying boutique.0001\_initial... OK

Applying sessions.0001\_initial... OK

Dans boutique>admin.py

*from* django.contrib *import* admin

*from* .models *import* \*

admin.site.register(Client)

admin.site.register(Produit)

admin.site.register(Commande)

admin.site.register(ArticleCom)

admin.site.register(AdresseLiv)

demarrer le serveur avec le port 8080

$ python manage.py runserver 8080

$ winpty python manage.py createsuperuser

$ winpty python manage.py createsuperuser

Username (leave blank to use 'azer'): programeraplus

Email address: programeraplus@hotmail.com

Password:

Password (again):

Superuser created successfully.

Puis accéder a l’admin : <http://127.0.0.1:8080/admin>

Puis ajouter 6 produits

### Produit1 / Name : Cornes de gazelle

/ Price : 12 / Description : En forme de petits croissants dont le cœur moelleux contient de la pâte d'amande.

### Produit2 / Name : Chebakia

/ Price : 8 / Description : Un petit délice servi lors du ramadan et très gourmand à la dégustation.

### Produit3 / Name : Baklawa

/ Price : 13 / Description : Est une pâtisserie venue du Moyen-Orient, originaire de la Turquie.

### Produit4 / Name : Fekas

/ Price : 5 / Description : ressemble au croquet sauf que là les biscuits sont plus riche en introduisant les fruits secs.

### Produit5 / Name : Richbond

/ Price : 3 / Description : Gâteau boules de neige, boule de coco ou gâteau Richbond.

### Produit6 / Name : M'Chewek

/ Price : 4 / Description : Biscuits d'amande frais avec de la confiture de fruit.

### Produit7 / Name : Lunes glacées

/ Price : 3 / Description : délicieux sablés fins, aux amandes, agréablement parfumés à l'eau de fleur d'oranger.

### Produit8 / Name : Ghriba

/ Price : 3 / Description : qu’on appelle aussi Bahla, c’est une pâtisserie Marocaine ancestral, pleine de craquelé et délicieuse en bouche.

### Produit9 /

### Name : Ghriba aux noix de coco

/ Price : 3 / Description : la Ghriba noix de coco et semoule est une délicieuse gourmandise Marocaine à déguster à l'heure du thé.

### Produit10 / Name : Gâteau aux Cacahuètes

/ Price : 4 / Description : Servez le Gâteau aux Cacahuètes avec un thé à la menthe.

### Produit11 / Name : Mini Fekkas

/ Price : 2 / Description : À dégustez avec du café noir ou un bon thé a la menthe.

### Produit12 / Name : Briouates

/ Price : 4 / Description : Briouats aux amandes et cacahuètes enrobées de miel.

Dans boutique>views.py

*from* django.shortcuts *import* render

*from* .models *import* \*

def boutique(request):

     produits = Produit.objects.all()

     context = {'produits':produits}

*return* render(request, 'boutique/boutique.html', context)

def chariot(request):

     context = {}

*return* render(request, 'boutique/chariot.html', context)

def caisse(request):

      context = {}

*return* render(request, 'boutique/caisse.html', context)

modifier boutique>templates>boutique>boutique.html

{% *extends* 'boutique/main.html' %}

{% load static %}

{% *block* content %}

    <section class="menu section bd-container" id="menu">

        <span class="section-subtitle">Nos produits Orientales et Maghrébines</span>

        <h2 class="section-title">Nos assortiments</h2>

        <div class="row">

            {% *for* produit in produits %}

            <div class="col-lg-4">

                <img class="thumbnail" src="{{produit.image.url}}">

                <div class="box-element produit">

                    <h6><strong>{{produit.nom}}</strong></h6>

                    <hr>

                    <button  class="btn btn-outline-secondary add-btn">Ajouter Chariot</button>

                    <h4 style="display:inline-block; float: right">MAD{{produit.prix|floatformat:2}}</h4>

                </div>

            </div>

            {% *endfor* %}

        </div>

    </section>

{% *endblock* content %}

Ajouter champ image du produit boutique>models.py

class Produit(models.Model):

    …

image = models.ImageField(null=True, blank=True)

    def \_\_str\_\_(self):

*return* self.nom

installe pillow : $ pip install pillow

$ python manage.py makemigrations

Migrations for 'boutique':

boutique\migrations\0002\_produit\_image.py

- Add field image to produit

$ python manage.py migrate

Operations to perform:

Apply all migrations: admin, auth, boutique, contenttypes, sessions

Running migrations:

Applying boutique.0002\_produit\_image... OK

Ajouter dans Bismi>settings.py

MEDIA\_ROOT = os.path.join(BASE\_DIR, 'static/images')

Puis ajouter les images dans votre application Bismi de même taille

Et n’oublier pas de supprimer les derniers images crées

<http://127.0.0.1:8080/admin/boutique/produit/>

puis ajouter MEDIA\_URL dans Bismi >setting.py

MEDIA\_URL = '/images/'

MEDIA\_ROOT = os.path.join(BASE\_DIR, 'static/images')

Dans Bismi >urls.py ajouter

*…*

*from* django.conf.urls.static *import* static

*from* django.conf *import* settings

urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

templates>boutique>main.html

        <script src="{% static 'js/main.js' %}"></script>

static>js>main.js

*/\*==================== SHOW MENU ====================\*/*

const showMenu = (toggleId, navId) =>{

    const toggle = document.getElementById(toggleId),

    nav = document.getElementById(navId)

*// Validate that variables exist*

*if*(toggle && nav){

        toggle.addEventListener('click', ()=>{

*// We add the show-menu class to the div tag with the nav\_\_menu class*

            nav.classList.toggle('show-menu')

        })

    }

}

showMenu('nav-toggle','nav-menu')

*/\*==================== REMOVE MENU MOBILE ====================\*/*

const navLink = document.querySelectorAll('.nav\_\_link')

function linkAction(){

    const navMenu = document.getElementById('nav-menu')

*// When we click on each nav\_\_link, we remove the show-menu class*

    navMenu.classList.remove('show-menu')

}

navLink.forEach(n => n.addEventListener('click', linkAction))

*/\*==================== CHANGE BACKGROUND HEADER ====================\*/*

function scrollHeader(){

    const nav = document.getElementById('header')

*// When the scroll is greater than 200 viewport height, add the scroll-header class to the header tag*

*if*(this.scrollY >= 200) nav.classList.add('scroll-header'); *else* nav.classList.remove('scroll-header')

}

window.addEventListener('scroll', scrollHeader)

*/\*==================== SHOW SCROLL TOP ====================\*/*

function scrollTop(){

    const scrollTop = document.getElementById('scroll-top');

*// When the scroll is higher than 560 viewport height, add the show-scroll class to the a tag with the scroll-top class*

*if*(this.scrollY >= 560) scrollTop.classList.add('show-scroll'); *else* scrollTop.classList.remove('show-scroll')

}

window.addEventListener('scroll', scrollTop)

*/\*==================== DARK LIGHT THEME ====================\*/*

const themeButton = document.getElementById('theme-button')

const darkTheme = 'dark-theme'

const iconTheme = 'bx-sun'

*// Previously selected topic (if user selected)*

const selectedTheme = localStorage.getItem('selected-theme')

const selectedIcon = localStorage.getItem('selected-icon')

*// We obtain the current theme that the interface has by validating the dark-theme class*

const getCurrentTheme = () => document.body.classList.contains(darkTheme) ? 'dark' : 'light'

const getCurrentIcon = () => themeButton.classList.contains(iconTheme) ? 'bx-moon' : 'bx-sun'

*// We validate if the user previously chose a topic*

*if* (selectedTheme) {

*// If the validation is fulfilled, we ask what the issue was to know if we activated or deactivated the dark*

  document.body.classList[selectedTheme === 'dark' ? 'add' : 'remove'](darkTheme)

  themeButton.classList[selectedIcon === 'bx-moon' ? 'add' : 'remove'](iconTheme)

}

*// Activate / deactivate the theme manually with the button*

themeButton.addEventListener('click', () => {

*// Add or remove the dark / icon theme*

    document.body.classList.toggle(darkTheme)

    themeButton.classList.toggle(iconTheme)

*// We save the theme and the current icon that the user chose*

    localStorage.setItem('selected-theme', getCurrentTheme())

    localStorage.setItem('selected-icon', getCurrentIcon())

})

*/\*==================== SCROLL REVEAL ANIMATION ====================\*/*

const sr = ScrollReveal({

    origin: 'top',

    distance: '30px',

    duration: 2000,

    reset: true

});

sr.reveal(`.home\_\_data, .home\_\_img,

            .about\_\_data, .about\_\_img,

            .services\_\_content, .menu\_\_content,

            .app\_\_data, .app\_\_img,

            .contact\_\_data, .contact\_\_button,

            .footer\_\_content`, {

    interval: 200

})

dans boutique>templates>boutique>boutique.html

Si en modifie une image de notre app Bismi il aura une erreur donc On Ajoute une solution d’erreur pour image

Dans boutique>models.py

Ajouter dans la classe Produit

class Produit(models.Model):

    ...

    @property

    def imageURL(self):

*try*:

            url = self.image.url

*except*:

            url = ''

*return* url

Dans <http://127.0.0.1:8080/admin> ajouter dans Client /Commande /ArticleCom

Modifier boutique>views.py

def chariot(request):

*if* request.user.is\_authenticated:

          client = request.user.client

          commande, created = Commande.objects.get\_or\_create(client=client, etat=False)

          items = commande.articlecom\_set.all()

*else*:

          items = []

     context = {'items':items}

*return* render(request, 'boutique/chariot.html', context)

modifier boutique>templates>boutique>chariot.html

{% *extends* 'boutique/main.html' %}

{% load static %}

{% *block* content %}

<main class="l-main">

    <section class="menu section bd-container" id="menu">

        <span class="section-subtitle">Nos produits Orientales et Maghrébines</span>

        <h2 class="section-title">Nos assortiments</h2>

        <div class="row">

            <div class="col-lg-12">

                <div class="box-element">

                    <a  class="btn btn-outline-dark" href="{% url 'boutique' %}">&#x2190; Acheter Encore</a>

                    <br>

                    <br>

                    <table class="table">

                        <tr>

                            <th><h5>Nombre de produits: <strong>3</strong></h5></th>

                            <th><h5>Total:<strong>42 MAD</strong></h5></th>

                            <th>

                                <a  style="float:right; margin:5px;" class="btn btn-outline-dark" href="{% url 'caisse' %}">Caisse</a>

                            </th>

                        </tr>

                    </table>

                </div>

                <div class="box-element">

                        <div class="chariot-row">

                            <div style="flex:2"></div>

                            <div style="flex:2"><strong>Nombre</strong></div>

                            <div style="flex:1"><strong>Prix</strong></div>

                            <div style="flex:1"><strong>Quantité</strong></div>

                            <div style="flex:1"><strong>Total</strong></div>

                        </div>

                        {% *for* item in items %}

                        <div class="chariot-row">

                            <div style="flex:2"><img class="row-image" src="{{item.produit.imageURL}}"></div>

                            <div style="flex:2"><p>{{item.produit.nom}}</p></div>

                            <div style="flex:1"><p>${{item.produit.prix|floatformat:2}}</p></div>

                            <div style="flex:1">

                                <p class="quantite">{{item.quantite}}</p>

                                <div class="quantite">

                                    <img class="chg-quantite" src="{% static  'images/arrow-up.png' %}">

                                    <p class="quantite">Kilo</p>

                                    <img class="chg-quantite" src="{% static  'images/arrow-down.png' %}">

                                </div>

                            </div>

                            <div style="flex:1"><p>32 MAD</p></div>

                        </div>

                        {% *endfor* %}

                </div>

            </div>

        </div>

    </section>

</main>

{% *endblock* content %}

Ajouter images arrow-up et arrow-down

Ajouter dans boutique>models.py la propriété qui calcule le total

class Commande(models.Model):

    …

    @property

    def get\_chariot\_total(self):

        commandeitems = self.articlecom\_set.all()

        total = sum([item.get\_total *for* item *in* commandeitems])

*return* total

    @property

    def get\_chariot\_items(self):

        commandeitems = self.articlecom\_set.all()

        total = sum([item.quantite *for* item *in* commandeitems])

*return* total

class ArticleCom(models.Model):

    …

    @property

    def get\_total(self):

        total = self.produit.prix \* self.quantite

*return* total

modifier templates>boutique>chariot.html

    <div style="flex:1"><p>{{item.get\_total}}$</p></div>

modifier cette ligne dans boutique>views.py dans chariot et boutique

def chariot(request):

*else*:

          items = []

          commande = {'get\_chariot\_total':0, 'get\_chariot\_items':0}

     context = {'items':items, 'commande':commande}

*return* render(request, 'boutique/chariot.html', context)

def caisse(request):

*if* request.user.is\_authenticated:

          client = request.user.client

          commande, created = Commande.objects.get\_or\_create(client=client, etat=False)

          items = commande.articlecom\_set.all()

*else*:

          items = []

          commande = {'get\_chariot\_total':0, 'get\_chariot\_items':0}

     context = {'items':items, 'commande':commande}

*return* render(request, 'boutique/caisse.html', context)

modifier dans boutique>chariot.html

                             <th><h5>Nombre:<strong>{{commande.get\_chariot\_items}}</strong></h5></th>

                            <th><h5>Total:<strong>{{commande.get\_chariot\_total|floatformat:2}} $</strong></h5></th>

Dans boutique>templates>boutique>caisse.html

{% *extends* 'boutique/main.html' %}

{% load static %}

{% *block* content %}

<main class="l-main">

    <section class="contact section bd-container" id="contact">

            <div class="row">

                <div class="col-lg-6">

                    <div class="box-element" id="form-wrapper">

                        <form id="form">

                            <div id="user-info">

                                <div class="form-field">

                                    <input required class="form-control" type="text" name="nom" placeholder="Nom..">

                                </div>

                                <div class="form-field">

                                    <input required class="form-control" type="email" name="email" placeholder="Email..">

                                </div>

                            </div>

                            <div id="shipping-info">

                                <hr>

                                <p>Information de Livraison:</p>

                                <hr>

                                <div class="form-field">

                                    <input class="form-control" type="text" name="adresse" placeholder="Adresse..">

                                </div>

                                <div class="form-field">

                                    <input class="form-control" type="text" name="ville" placeholder="Ville..">

                                </div>

                                <div class="form-field">

                                    <input class="form-control" type="text" name="coderecu" placeholder="Code Recu..">

                                </div>

                                <div class="form-field">

                                    <input class="form-control" type="text" name="codepostal" placeholder="Code Postal..">

                                </div>

                            </div>

                            <hr>

                            <input id="form-button" class="btn btn-secondary btn-lg btn-block" type="submit" value="Continue">

                        </form>

                    </div>

                    <br>

                    <div class="box-element hidden" id="payment-info">

                        <small>Options Paypal</small>

                    </div>

                </div>

                <div class="col-lg-6">

                    <div class="box-element">

                        <a  class="btn btn-outline-dark" href="{% url 'chariot' %}">&#x2190; Retour au panier</a>

                        <hr>

                        <h3>Sommaire de Commande</h3>

                        <hr>

                        {% *for* item in items %}

                        <div class="chariot-row">

                            <div style="flex:2"><img class="row-image" src="{{item.produit.imageURL}}"></div>

                            <div style="flex:2"><p>{{item.produit.nom}}</p></div>

                            <div style="flex:1"><p>${{item.produit.prix}}</p></div>

                            <div style="flex:1"><p>x{{item.quantite}}</p></div>

                        </div>

                        {% *endfor* %}

                        <h5>Items:  {{commande.get\_chariot\_items}}</h5>

                        <h5>Total:   ${{commande.get\_chariot\_total}}</h5>

                    </div>

                </div>

            </div>

    </section>

</main>

{% *endblock* content %}

Modifier dans styles.css

*/\*===== GOOGLE FONTS =====\*/*

*@import* url("https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap");

*/\*===== VARIABLES CSS =====\*/*

:root {

  --header-height: 3rem;

*/\*========== Colors ==========\*/*

  --first-color: #f2a328;

  --first-color-alt:  #707070;

  --title-color: #393939;

  --text-color: #707070;

  --text-color-light: #A6A6A6;

  --body-color: #FBFEFD;

  --container-color: #FFFFFF;

*/\*========== Font and typography ==========\*/*

  --body-font: 'Poppins', sans-serif;

  --biggest-font-size: 2.25rem;

  --h1-font-size: 1.5rem;

  --h2-font-size: 1.25rem;

  --h3-font-size: 1rem;

  --normal-font-size: .938rem;

  --small-font-size: .813rem;

  --smaller-font-size: .75rem;

*/\*========== Font weight ==========\*/*

  --font-medium: 500;

  --font-semi-bold: 600;

*/\*========== Margenes ==========\*/*

  --mb-1: .5rem;

  --mb-2: 1rem;

  --mb-3: 1.5rem;

  --mb-4: 2rem;

  --mb-5: 2.5rem;

  --mb-6: 3rem;

*/\*========== z index ==========\*/*

  --z-tooltip: 10;

  --z-fixed: 100;

}

*@media* screen and (min-width: 768px){

  :root{

    --biggest-font-size: 4rem;

    --h1-font-size: 2.25rem;

    --h2-font-size: 1.5rem;

    --h3-font-size: 1.25rem;

    --normal-font-size: 1rem;

    --small-font-size: .875rem;

    --smaller-font-size: .813rem;

  }

}

*/\*========== BASE ==========\*/*

\*,::before,::after{

  box-sizing: border-box;

}

html{

  scroll-behavior: smooth;

}

#cart-icon{

  width:25px;

  display: inline-block;

  margin-left: 15px;

}

#cart-total{

  display: block;

  text-align: center;

  color:#fff;

  background-color: #f2a328;

  width: 20px;

  height: 25px;

  border-radius: 50%;

  font-size: 14px;

}

*/\*========== Variables Dark theme ==========\*/*

body.dark-theme{

  --title-color: #F1F3F2;

  --text-color: #C7D1CC;

  --body-color: #1D2521;

  --container-color: #27302C;

}

*/\*========== Button Dark/Light ==========\*/*

.change-theme{

  position: absolute;

  right: 1rem;

  top: 1.8rem;

  color: var(--text-color);

  font-size: 1rem;

  cursor: pointer;

}

body{

  margin: var(--header-height) 0 0 0;

  font-family: var(--body-font);

  font-size: var(--normal-font-size);

  background-color: var(--body-color);

  color: var(--text-color);

  line-height: 1.6;

}

h1,h2,h3,p,ul{

  margin: 0;

}

ul{

  padding: 0;

  list-style: none;

}

a{

  text-decoration: none;

}

img{

  max-width: 100%;

  height: auto;

}

*/\*========== CLASS CSS ==========\*/*

.section{

  padding: 4rem 0 2rem;

}

.section-title, .section-subtitle{

  text-align: center;

}

.section-title{

  font-size: var(--h1-font-size);

  color: var(--title-color);

  margin-bottom: var(--mb-3);

}

.section-subtitle{

  display: block;

  color: var(--first-color);

  font-weight: var(--font-medium);

  margin-bottom: var(--mb-1);

}

*/\*========== LAYOUT ==========\*/*

.bd-container{

  max-width: 960px;

  width: calc(100% - 2rem);

  margin-left: var(--mb-2);

  margin-right: var(--mb-2);

}

.bd-grid{

  display: grid;

  gap: 1.5rem;

}

.l-header{

  width: 100%;

  position: fixed;

  top: 0;

  left: 0;

  z-index: var(--z-fixed);

  background-color: var(--body-color);

}

*/\*========== NAV ==========\*/*

.nav{

  max-width: 1024px;

  height: var(--header-height);

  display: flex;

  justify-content: space-between;

  align-items: center;

}

*@media* screen and (max-width: 768px){

  .nav\_\_menu{

    position: fixed;

    top: -100%;

    left: 0;

    width: 100%;

    padding: 1.5rem 0 1rem;

    text-align: center;

    background-color: var(--body-color);

    transition: .4s;

    box-shadow: 0 4px 4px rgba(0,0,0,.1);

    border-radius: 0 0 1rem 1rem;

    z-index: var(--z-fixed);

  }

}

.nav\_\_item{

  margin-bottom: var(--mb-2);

}

.nav\_\_link, .nav\_\_logo, .nav\_\_toggle{

  color: var(--text-color);

  font-weight: var(--font-medium);

}

.nav\_\_logo:hover{

  color: var(--first-color);

}

.nav\_\_link{

  transition: .3s;

}

.nav\_\_link:hover{

  color: var(--first-color);

}

.nav\_\_toggle{

  font-size: 1.3rem;

  cursor: pointer;

}

*/\* Show menu \*/*

.show-menu{

  top: var(--header-height);

}

*/\* Active menu \*/*

.active-link{

  color: var(--first-color);

}

*/\* Change background header \*/*

.scroll-header{

  box-shadow: 0 2px 4px rgba(0,0,0,.1);

}

*/\* Scroll top \*/*

.scrolltop{

  position: fixed;

  right: 1rem;

  bottom: -20%;

  display: flex;

  justify-content: center;

  align-items: center;

  padding: .3rem;

  background: #d1690c;

  border-radius: .4rem;

  z-index: var(--z-tooltip);

  transition: .4s;

  visibility: hidden;

}

.scrolltop:hover{

  background-color: var(--first-color-alt);

}

.scrolltop\_\_icon{

  font-size: 1.8rem;

  color: var(--body-color);

}

*/\* Show scrolltop \*/*

.show-scroll{

  visibility: visible;

  bottom: 1.5rem;

}

*/\*========== HOME ==========\*/*

.home\_\_container{

  height: calc(100vh - var(--header-height));

  align-content: center;

}

.home\_\_title{

  font-size: var(--biggest-font-size);

  color: var(--first-color);

  margin-bottom: var(--mb-1);

}

.home\_\_subtitle{

  font-size: var(--h2-font-size);

  color: var(--title-color);

  margin-bottom: var(--mb-4);

}

.home\_\_img{

  width: 300px;

  justify-self: center;

}

*/\*========== BUTTONS ==========\*/*

.button{

  display: inline-block;

  background-color: var(--first-color);

  color: #FFF;

  padding: .75rem 1rem;

  border-radius: .5rem;

  transition: .3s;

}

.button:hover{

  background-color: var(--first-color-alt);

}

*/\*========== ABOUT ==========\*/*

.about\_\_data{

  text-align: center;

}

.about\_\_description{

  margin-bottom: var(--mb-3);

}

.about\_\_img{

  width: 280px;

  border-radius: .5rem;

  justify-self: center;

}

*/\*========== SERVICES ==========\*/*

.services\_\_container{

  row-gap: 2.5rem;

  grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));

}

.services\_\_content{

  text-align: center;

}

.services\_\_img{

  width: 64px;

  height: 64px;

  fill: var(--first-color);

  margin-bottom: var(--mb-2);

}

.services\_\_title{

  font-size: var(--h3-font-size);

  color: var(--title-color);

  margin-bottom: var(--mb-1);

}

.services\_\_description{

  padding: 0 1.5rem;

}

*/\*========== MENU ==========\*/*

.menu\_\_container{

  grid-template-columns: repeat(2, 1fr);

  justify-content: center;

}

.menu\_\_content{

  position: relative;

  display: flex;

  flex-direction: column;

  background-color: var(--container-color);

  border-radius: .5rem;

  box-shadow: 0 2px 4px rgba(3,74,40,.15);

  padding: .75rem;

}

.menu\_\_img{

  width: 100px;

  align-self: center;

  margin-bottom: var(--mb-2);

}

.menu\_\_name, .menu\_\_preci{

  font-weight: var(--font-semi-bold);

  color: var(--title-color);

}

.menu\_\_name{

  font-size: var(--normal-font-size);

}

.menu\_\_detail, .menu\_\_preci{

  font-size: var(--small-font-size);

}

.menu\_\_detail{

  margin-bottom: var(--mb-1);

}

.menu\_\_button{

  position: absolute;

  bottom: 0;

  right: 0;

  display: flex;

  padding: .625rem .813rem;

  border-radius: .5rem 0 .5rem 0;

}

*/\*========== APP ==========\*/*

.app\_\_data{

  text-align: center;

}

.app\_\_description{

  margin-bottom: var(--mb-5);

}

.app\_\_stores{

  margin-bottom: var(--mb-4);

}

.app\_\_store{

  width: 120px;

  margin: 0 var(--mb-1);

}

.app\_\_img{

  width: 230px;

  justify-self: center;

}

*/\*========== CONTACT ==========\*/*

.contact\_\_container{

  text-align: center;

}

.contact\_\_description{

  margin-bottom: var(--mb-3);

}

*/\*========== FOOTER ==========\*/*

.footer\_\_container{

  grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));

  row-gap: 2rem;

}

.footer\_\_logo{

  font-size: var(--h3-font-size);

  color: var(--first-color);

  font-weight: var(--font-semi-bold);

}

.footer\_\_description{

  display: block;

  font-size: var(--small-font-size);

  margin: .25rem 0 var(--mb-3);

}

.footer\_\_social{

  font-size: 1.5rem;

  color: var(--title-color);

  margin-right: var(--mb-2);

}

.footer\_\_title{

  font-size: var(--h2-font-size);

  color: var(--title-color);

  margin-bottom: var(--mb-2);

}

.footer\_\_link{

  display: inline-block;

  color: var(--text-color);

  margin-bottom: var(--mb-1);

}

.footer\_\_link:hover{

  color: var(--first-color);

}

.footer\_\_copy{

  text-align: center;

  font-size: var(--small-font-size);

  color: var(--text-color-light);

  margin-top: 3.5rem;

}

*/\*========== MEDIA QUERIES ==========\*/*

*@media* screen and (min-width: 576px){

  .home\_\_container,

  .about\_\_container,

  .app\_\_container{

    grid-template-columns: repeat(2,1fr);

    align-items: center;

  }

  .about\_\_data, .about\_\_initial,

  .app\_\_data, .app\_\_initial,

  .contact\_\_container, .contact\_\_initial{

    text-align: initial;

  }

  .about\_\_img, .app\_\_img{

    width: 380px;

    order: -1;

  }

  .contact\_\_container{

    grid-template-columns: 1.75fr 1fr;

    align-items: center;

  }

  .contact\_\_button{

    justify-self: center;

  }

}

*@media* screen and (min-width: 768px){

  body{

    margin: 0;

  }

  .section{

    padding-top: 8rem;

  }

  .nav{

    height: calc(var(--header-height) + 1.5rem);

  }

  .nav\_\_list{

    display: flex;

  }

  .nav\_\_item{

    margin-left: var(--mb-5);

    margin-bottom: 0;

  }

  .nav\_\_toggle{

    display: none;

  }

  .change-theme{

    position: initial;

    margin-left: var(--mb-2);

  }

  .home\_\_container{

    height: 100vh;

    justify-items: center;

  }

  .services\_\_container,

  .menu\_\_container{

    margin-top: var(--mb-6);

  }

  .menu\_\_container{

    grid-template-columns: repeat(3, 210px);

    column-gap: 4rem;

  }

  .menu\_\_content{

    padding: 1.5rem;

  }

  .menu\_\_img{

    width: 130px;

  }

  .app\_\_store{

    margin: 0 var(--mb-1) 0 0;

  }

}

*@media* screen and (min-width: 960px){

  .bd-container{

    margin-left: auto;

    margin-right: auto;

  }

  .home\_\_img{

    width: 500px;

  }

  .about\_\_container,

  .app\_\_container{

    column-gap: 7rem;

  }

}

h1,h2,h3,h4,h5,h6{

  color:hsl(0, 0%, 30%);

}

.box-element{

  box-shadow:#f2a328;

  background-color: #fff;

  border-radius: 4px;

  padding: 10px;

}

.produit{

  border-radius: 0 0 4px 4px;

}

.bg-dark{

  background-color: #4f868c!important;

}

#chariot-icon{

  width:25px;

  display: inline-block;

  margin-left: 15px;

}

#chariot-total{

  display: block;

  text-align: center;

  color:#fff;

  background-color:#f2a328;

  width: 20px;

  height: 25px;

  border-radius: 50%;

  font-size: 14px;

}

.btn{

  border-radius: 0;

}

.row-image{

  width: 100px;

}

.form-field{

  width:250px;

  display: inline-block;

  padding: 5px;

}

.chariot-row{

  display: flex;

    align-items: flex-stretch;

    padding-bottom: 10px;

    margin-bottom: 10px;

    border-bottom: 1px solid #ececec;

}

.quantite{

  display: inline-block;

  font-weight: 700;

  padding-right:10px;

}

.chg-quantite{

  width: 12px;

  cursor: pointer;

  display: block;

  margin-top: 5px;

  transition:.1s;

}

.chg-quantite:hover{

  opacity: .6;

}

.hidden{

  display: none!important;

}

.col-lg-4, .col-lg-6, .col-lg-8, .col-lg-12{

  margin-top: 10px;

}

.thumbnail{

  width: 100%;

  height: 200px;

  -webkit-box-shadow: -1px -3px 5px -2px rgba(214,214,214,1);

    -moz-box-shadow: -1px -3px 5px -2px #f2a328;

    box-shadow: -1px -3px 5px -2px #f2a328;

}

.box-element{

  box-shadow:#f2a328;

  background-color:#f2a328;

  border-radius: 4px;

  padding: 10px;

}

.produit{

  border-radius: 0 0 4px 4px;

}