



گزارش پروژه درس برنامه‌نویسی پیشرفته

جناب آقای دکتر حبیبی
جناب آقای دکتر نبویان

400103419	مهسا سرسنگی
400103187	مبینا باطبی
400103381	ساغر رضائواز
400170277	رژین صوفی‌آبادی

3	لینک‌های مهم
3	کتابخانه‌های استفاده شده در پروژه
3	مدل‌ها
3	:Category Model
3	:Product Model
3	:Cart Model
4	:Order Model
4	:Storage Model
4	فرم‌ها
4	:RegistrationForm
4	:LoginForm
4	صفحات سایت
4	:Home
5	:Contact
5	:Add_product
6	:Detail
6	:All_categories
7	:Category_products
7	:Registration
7	:Profile
8	:Cart
9	:Storage
9	:Store management
9	:Login
10	:Logout
10	:Navbar HTML
10	:Footer HTML
10	:Base HTML

کتابخانه‌های استفاده شده در پروژه

Matplotlib: از این کتابخانه برای رسم نمودار مربوط به صفحه مدیریت فروشگاه و تحلیل فروش محصولات استفاده شده است.

pillow: از این کتابخانه برای آپلود و پردازش تصاویر استفاده شده است.

مدل‌ها

در ادامه به توضیح مدل‌ها می‌پردازیم:

Category Model

این مدل برای ذخیره و مدیریت دسته‌بندی‌های محصولات استفاده می‌شود و شامل ویژگی‌های زیر است:

- title: مشخص کننده عنوان دسته‌بندی محصولات می‌باشد.
- slug: یک نام یکتا برای دسته‌بندی است که برای ایجاد URL استفاده می‌شود.
- description: توضیحات کوتاهی درباره دسته‌بندی محصولات ارائه می‌دهد.
- category_image: تصویر یا آیکون مرتبط با دسته‌بندی محصولات را ذخیره می‌کند.

Product Model

این مدل جزئیات محصولات را شامل می‌شود و برای نمایش و مدیریت محصولات در فروشگاه آنلاین استفاده می‌شود. ویژگی‌های آن شامل این موارد است:

- title: عنوان محصول را نشان می‌دهد.
- sugar,coffee,flour,chocolate,milk: این ویژگی‌ها نشان دهنده مواد مورد استفاده در تهیه محصول می‌باشند.
- slug: یک نام یکتا برای محصول است که برای ایجاد URL استفاده می‌شود.
- short_description: توضیحی کوتاه درباره محصول را ارائه می‌دهد.
- product_image: تصویر محصول را نگهداری می‌کند.
- price: قیمت محصول را نشان می‌دهد.
- category: دسته‌بندی محصول را نشان می‌دهد که با کلید خارجی به مدل دسته‌بندی متصل است.

Cart Model

این مدل مربوط به سبد خرید کاربران است که اطلاعات محصولاتی که کاربر انتخاب کرده است، را نگهداری می‌کند. ویژگی‌های آن شامل این موارد است:

- user: کاربری که محصولات را به سبد خرید اضافه کرده است.
- product: محصول انتخاب شده که در سبد خرید قرار دارد.
- quantity: تعداد محصولات موجود در سبد خرید را نشان می‌دهد.

- `total_price`: قیمت کل سبد خرید را با در نظر گرفتن تعداد و قیمت محصولات محاسبه می‌کند.

:Order Model

این مدل برای ثبت سفارشات کاربران در فروشگاه آنلاین استفاده می‌شود.

- `user`: کاربری که سفارش را ثبت کرده است.
- `product`: محصول مورد سفارش را نشان می‌دهد.
- `quantity`: تعداد محصولاتی که در سفارش وجود دارد را نشان می‌دهد.
- `ordered_date`: تاریخ و زمان ثبت سفارش را نگهداری می‌کند.
- `status`: وضعیت فعلی سفارش، به عنوان مثال "Pending".

:Storage Model

این مدل برای ذخیره و مدیریت اطلاعات مربوط به موجودی انبار استفاده می‌شود.

- `name`: نام محصول یا آیتم موجود در انبار را نشان می‌دهد.
- `amount`: مقدار یا تعداد موجودی انبار برای آن مورد را ذخیره می‌کند.

فرم‌ها

در ادامه به توضیح فرم‌ها پرداخته می‌شود.

:RegistrationForm

این فرم برای ثبت نام کاربران جدید استفاده می‌شود. و دارای فیلدهای `full_name`, `username`, `email`, `password1` برای ورود اطلاعات کاربر مورد استفاده قرار می‌گیرند. از ویژگی‌های `label` و `widget` برای تنظیم ظاهر فرم استفاده شده است.

:LoginForm

این فرم برای ورود به سیستم کاربران موجود استفاده می‌شود. فیلدهای `username` و `password` برای ورود اطلاعات کاربر مورد استفاده قرار می‌گیرند. در این فرم، یک متد `clean` تعریف شده است که برای بررسی اینکه آیا نام کاربری وارد شده واقعاً یک ایمیل است یا نه، استفاده می‌شود.

:ProductForm

این فرم برای ایجاد و ویرایش محصولات در کافی شاپ توسط ادمین استفاده می‌شود. این فرم از مدل `Product` برای تعریف فیلدها استفاده می‌کند.

صفحات سایت

:Home

این صفحه محصولات و دسته‌بندی‌های اصلی را نمایش می‌دهد. در ادامه به اجزای آن می‌پردازیم:

- View:

تابع View داده‌ها را بازیابی می‌کند و محیط را برای رندر شدن در تمپلیت آماده می‌کند. این تابع سه دسته را از مدل Category بازیابی کرده و محصولات را بر اساس مجموع تعداد موجودی فروش مرتب می‌کند و 12 محصول پرفروش را نشان می‌دهد. همچنین همه دسته‌ها را برای منوی Navigator بازیابی می‌کند. در نهایت، این داده‌ها را به `context` منتقل کرده و تمپلیت `index.html` را رندر می‌کند.

- HTML:

تمپلیت `base.html` شامل هدر و فوتر و منو (Navbar.html) را با استفاده از تگ URL گسترش می‌دهیم. محتوای واقعی صفحه در داخل تگ‌های `{% block content %}` و `{% endblock %}` قرار داده می‌شود. ابتدا اسلایدر خودکار را با استفاده از Bootstrap ایجاد می‌کنیم. سپس سه کتگوری را در یک بخش نمایش می‌دهد، با حلقه زدن روی دسته‌ها که در محیط ارسال می‌شود. برای هر دسته بررسی می‌کند که آیا به هر تصویر کتگوری مرتبط شده و در این صورت، آن را نمایش می‌دهد؛ در غیر این صورت تصویر پیش‌فرضی را نشان می‌دهد. بعداً محصولات پرفرمدار را در بخش دیگری نمایش می‌دهد. برای هر محصول در لیست `products`، تصویر محصول، عنوان، قیمت را نشان داده و با استفاده از یک فرم یک دکمه "افزودن به سبد خرید" ارائه می‌دهد.

- CSS:

در تگ Style، ارتفاع عکسها و اسلایدر تنظیم شده است. همچنین تگ Static به منظور ارتباط با فولدر Static شامل CSS است.

Contact

این صفحه اطلاعات تماس را نمایش می‌دهد.

- View:

اطلاعات categories_menu به عنوان context برای نمایش منو در صفحه فرستاده شده است.

- HTML:

{% 'extends' base.html %}: این تگ به فایل base.html ارجاع می‌دهد و از آن به عنوان الگوی پایه برای این صفحه استفاده می‌کند. {% load static %}: این تگ فایل‌های استاتیک مورد نیاز مانند CSS و JavaScript را بارگذاری می‌کند. اطلاعات درباره ما، ساعات کاری، آدرس، نقشه مکانی و اطلاعات تماس فروشگاه با قالب‌ها و سبک‌های مختلف درون تگ‌های HTML نوشته شده‌اند. تگ <div class="map-container"> برای نمایش نقشه استفاده شده است. درون این تگ، یک فریم <iframe> از Google Maps قرار داده شده است که نقشه مکان فروشگاه قهوه‌ای با عنوان "Coffee Shop" را نمایش می‌دهد. این فریم <iframe> از Google Maps URL به آدرس ...=https://www.google.com/maps/embed?pb برای نمایش نقشه دانشکده صنایع استفاده می‌کند.

Add_product

- View:

صفحه "Add Product" به View ای با نام add_product مربوط می‌شود. به دلیل استفاده از دکوراتور @user_passes_test اگر کاربر از نوع staff باشد، به این ویو دسترسی خواهد داشت. این ویو در صفحه Add Product ابتدا تمام دسته‌بندی‌های موجود را از پایگاه داده استخراج می‌کند. وقتی کاربر فرم را پر کرده و ارسال می‌کند (متود POST)، ویو فرم را با استفاده از اطلاعات ارسالی از فرم ProductForm ایجاد می‌کند و اگر فرم معتبر باشد، اطلاعات فرم را ذخیره می‌کند و کاربر را به صفحه اصلی هدایت می‌کند که فهرستی از محصولات را نمایش می‌دهد. در غیر این صورت (وقتی که متد درخواست برابر POST نباشد)، ویو یک نمونه خالی از فرم ProductForm ایجاد می‌کند.

• HTML:

در (add-product.html)، از base.html به عنوان تمپلیت پایه استفاده شده و فایل‌های استاتیک برای CSS بارگذاری شده‌اند. بخش اصلی محتوای صفحه در block content قرار دارد. این بخش شامل یک فرم برای اضافه کردن محصول است که از روش POST استفاده می‌کند و مجموعه‌ای از فیلدها و دکمه‌ای برای ارسال فرم را نمایش می‌دهد. در این فرم، از {% csrf_token %} برای افزودن توکن CSRF استفاده شده و فیلدهای مختلف فرم به صورت iterator پردازش، نمایش داده شده‌اند. در صورت بروز خطاها، پیامهای خطا نیز نمایش داده شده‌اند. در نهایت، ویو داده‌های لازم را به تمپلیت ارسال کرده و صفحه نهایی به کاربر نمایش داده می‌شود.

:Detail

این صفحه جزئیات یک محصول خاص را بر اساس slug نمایش می‌دهد.

• View:

یک ویو با نام detail تعریف شده است که دو پارامتر request و slug دریافت می‌کند. این ویو به وسیله متد get_object_or_404 یک محصول با slug مشخص را از پایگاه داده استخراج می‌کند. همچنین محصولات مرتبط با محصول اصلی (دارای کتگوری یکسان) بر اساس دسته‌ی محصول اصلی را نیز استخراج می‌کند. دسته‌های موجود نیز از جدول Category در پایگاه داده گرفته می‌شوند.

• HTML:

سپس یک context شامل اطلاعات مربوط به محصول اصلی، محصولات مرتبط و دسته‌ها به تمپلیت detail.html ارسال می‌شود و تمپلیت نهایی به کاربر نمایش داده می‌شود. تمپلیت مربوطه که با استفاده از {% extends 'base.html' %} از تمپلیت اصلی base.html ارث‌بری می‌کند، بخشی از محتوای خود را با {% block content %} جایگزین می‌کند. در این بخش برای اضافه کردن محصول به سبد خرید نمایش داده می‌شود. اطلاعات فرم که با استفاده از ایتريتور for از فیلدهای فرم تولید می‌شوند، به‌مراه توکن CSRF جهت امنیت فرم، نمایش داده می‌شوند. در صورت وجود خطاها در فرم، پیام خطا نیز نمایش داده می‌شود. با کلیک بر روی دکمه Add to cart، اطلاعات فرم به ویو مربوطه ارسال شده و پس از پردازش، به کاربر بسته به نتیجه، اطلاعات لازم نشان داده می‌شود.

:All_categories

این صفحه تمام محصولات از تمام دسته‌بندی‌ها را نمایش می‌دهد.

• View:

این ویو با نام all_categories، دو مدل محصول و کتگوری از پایگاه داده را استخراج می‌کند و فهرست دسته‌بندی‌ها را از جدول Category (برای منو صفحه) و تمام محصولات (برای نمایش) را از جدول Product دریافت می‌کند. سپس این اطلاعات را به تمپلیت categories.html ارسال می‌کند.

• HTML:

تمپلیت categories.html که از base.html ارث‌بری می‌کند، در بخش content قسمت‌های صفحه را نمایش می‌دهد. در این تمپلیت، یک صفحه HTML ساده طراحی شده است که شامل دسته‌بندی و محصولات مرتبط با آن دسته‌بندی است. بخش Hero شامل اطلاعات مربوط به دسته‌بندی نمایشی است، از جمله عنوان و توضیحات آن دسته‌بندی. بخش Breadcrumb یک نوبری ساده است که به کاربر امکان می‌دهد به راحتی بین صفحات جابه‌جا شود و ارتباط صفحه فعلی با صفحات دیگر را نمایش می‌دهد. با استفاده از breadcrumb از bootstrap، کاربران می‌توانند به راحتی مسیری که در آن پیش رفته‌اند را متوجه شوند و به صفحات قبلی یا صفحه اصلی بازگردند، که این امر از تجربه کاربری بهتری برخوردار است. بخش Product Listing شامل لیستی از محصولات مرتبط با دسته‌بندی است که به کاربران نمایش داده می‌شود. برای هر محصول، تصویر، عنوان، قیمت و گزینه‌های افزودن به سبد خرید نمایش داده می‌شود. همچنین یک استایل CSS نیز در آخر تمپلیت برای تنظیم اندازه تصاویر دسته‌بندی و محصول‌ها به عنوان الگوی ظاهری اضافه شده است.

:Category_products

این صفحه محصولات مرتبط با یک دسته‌بندی خاص را بر اساس slug نمایش می‌دهد.

• View

با دریافت یک پارامتر slug، دسته‌بندی مورد نظر را از مدل Category دریافت می‌کند و سپس محصولات مرتبط با آن دسته‌بندی را از مدل Product استخراج کرده و در متغیر products ذخیره می‌کند. سپس تمام دسته‌بندی‌ها را در categories_menu ذخیره می‌کند و یک تمپلیت به نام category_products.html با اطلاعات محصولات، دسته‌بندی و فهرست دسته‌بندی‌ها را به کاربر نمایش می‌دهد.

• HTML

این تمپلیت یک صفحه وب برای نمایش محصولات دسته‌بندی شده استفاده می‌شود. در این صفحه، محصولات با تصویر، عنوان، قیمت و گزینه افزودن به سبد خرید نشان داده شده‌اند. استایل CSS نیز برای تنظیم اندازه تصاویر محصولات استفاده شده است.

:Registration

این صفحه برای ثبت‌نام کاربران جدید استفاده می‌شود.

• View

RegistrationView از کلاس View ارث‌بری شده و دارای دو متد get و post است. در متد get، یک فرم ثبت نام ایجاد شده و به عنوان محتوای اصلی تمپلیت register.html ارسال می‌شود. در متد post، ابتدا اعتبارسنجی فرم انجام می‌شود. اگر فرم معتبر باشد، پیام موفقیت‌آمیز بودن ثبت نام نمایش داده شده و فرم ثبت نام ارسال می‌شود. در هر صورت، تمپلیت register.html به همراه فرم و فهرست دسته‌بندی‌ها نمایش داده می‌شود.

• HTML

تمپلیت register.html نیز از base.html به عنوان قالب استفاده می‌کند و شامل یک فرم ثبت نام است که از ورودی‌های مختلفی مانند نام کاربری، ایمیل و رمز عبور تشکیل شده است. در صورت وجود خطاهای اعتبارسنجی، آن‌ها نمایش داده خواهند شد و همچنین پیوند "ورود" برای کسانی که قبلاً ثبت نام کرده‌اند، نمایش داده شده است.

:Profile

این صفحه اطلاعات پروفایل کاربر و سفارش‌های وی را نمایش می‌دهد. فقط کاربران لاگین کرده می‌توانند به این صفحه دسترسی داشته باشند.

• View

این ویو محتوای پروفایل کاربر را نمایش می‌دهد. ویو با استفاده از ویژگی @login_required اطمینان حاصل می‌کند که تنها کاربران وارد شده به این صفحه دسترسی داشته باشند. ویو اطلاعات سفارشات کاربر جاری را از مدل Order با فیلتر بر اساس کاربر نمایش می‌دهد. همچنین فهرست دسته‌بندی‌ها از مدل Category استخراج می‌شود.

• HTML

تمپلیت مربوط به این ویو از قالب base.html ارث‌بری شده و شامل دو بخش اصلی است. ابتدا یک حاشیه کلی با اطلاعات ساده‌ای مانند ایمیل و نام کاربری کاربر جاری نمایش داده می‌شود. سپس یک بخش جدول اطلاعات سفارشات با استفاده از متغیر orders نمایش داده می‌شود. در جدول سفارشات، اگر کاربر سفارشی داشته باشد، جزئیات هر سفارش به همراه تعداد و تاریخ سفارش نمایش داده می‌شود. اگر کاربر هنوز سفارشی نداشته باشد، پیام "شما هنوز چیزی سفارش نداده‌اید." نمایش داده می‌شود.

این صفحه محصولات موجود در سبد خرید کاربر را نمایش می‌دهد.

• View

Add_to_cart: از دکوریتر `login_required` استفاده شده است که اطمینان حاصل می‌کند که تنها کاربران ورود کرده به این صفحه دسترسی داشته باشند. این تابع وظیفه اضافه کردن محصول به سبد را دارد. ابتدا محصولی که قرار است به سبد اضافه شود را از درخواست دریافت می‌کند. سپس با استفاده از توابع دیگر (مانند `check_ingredients_availability` و `add_product_to_cart_and_deduct_ingredients`) بررسی می‌شود که آیا اجزای لازم برای تهیه این محصول در انبار موجود است یا خیر. در صورت موجود بودن، محصول به سبد اضافه می‌شود و مقدار اجزای مورد نیاز از انبار کم می‌شود.

add_product_to_cart_and_deduct_ingredients(product, user): این تابع وظیفه اضافه کردن محصول به سبد و کاهش موجودی اجزای لازم در انبار را دارد. ابتدا بررسی می‌شود که آیا محصول مورد نظر قبلاً در سبد قرار دارد یا خیر. در صورت وجود، تعداد آن افزایش می‌یابد در غیر این صورت یک رکورد جدید برای محصول در سبد ایجاد می‌شود.

deduct_ingredient_amounts(product): این تابع وظیفه کاهش مقدار اجزای مورد نیاز برای تهیه محصول از انبار را دارد. برای هر اجزا، مقدار مورد نیاز برای تولید محصول از مقدار موجود در انبار کم می‌شود.

check_ingredients_availability(product): این تابع وظیفه بررسی می‌کند که آیا مقدار اجزای لازم برای تهیه محصول در انبار موجود است یا خیر. برای هر اجزا، ابتدا مشخص می‌شود که آیا آن اجزا در انبار موجود است و سپس بررسی می‌شود که آیا مقدار موجود کافی است یا خیر. اگر مقدار برای حتی یک اجزا کافی نباشد، تابع `False` برمی‌گرداند، در غیر این صورت `True` برمی‌گرداند.

remove_cart: این صفحه محصول را از سبد خرید کاربر حذف می‌کند. این تابع وظیفه حذف یک محصول از سبد خرید است. ابتدا با استفاده از `get_object_or_404` شیء مورد نظر از مدل `Cart` با شناسه `cart_id` یافته می‌شود. سپس این شیء حذف می‌شود و یک پیام موفقیت آمیز به کاربر نمایش داده می‌شود.

plus_cart: این صفحه تعداد یک محصول در سبد خرید را افزایش می‌دهد. این تابع وظیفه افزایش تعداد یک محصول در سبد خرید است. با استفاده از `get_object_or_404` شیء مورد نظر از مدل `Cart` با شناسه `cart_id` یافته و تعداد آن افزایش می‌یابد.

minus_cart: این صفحه تعداد یک محصول در سبد خرید را کاهش می‌دهد. این تابع وظیفه کاهش تعداد یک محصول در سبد خرید است.

checkout: این صفحه فرآیند تکمیل خرید و ثبت سفارش‌ها را انجام می‌دهد. این تابع وظیفه تکمیل خریدهای کاربر است. ابتدا سبد خرید کاربر با فیلتر کردن بر اساس کاربر جاری (`user`) یافته می‌شود. برای هر مورد در سبد، یک سفارش جدید با اطلاعات محصول و تعداد آن ایجاد می‌شود.

Orders: این صفحه سفارش‌های کاربر را نمایش می‌دهد. دسته‌های منو (`categories_menu`) و تمام سفارش‌های کاربر از مدل `Order` با فیلتر بر اساس کاربر جاری گرفته شده و به صورت مرتب بر اساس تاریخ سفارش نمایش داده می‌شوند. سپس یک پیام موفقیت آمیز برای کاربر نمایش داده می‌شود و اطلاعات مربوط به سفارش‌ها به تمپلیت `cart.html` ارسال می‌شود.

• HTML: این تمپلیت به نمایش سبد خرید کاربر می‌پردازد. به طور خلاصه، این تمپلیت دارای محتوای زیر است:

Shopping cart: نمایش محتوای سبد خرید که شامل یک جدول اطلاعات محصولات است. برای هر محصول نمایش تصویر، عنوان، قیمت، تعداد، مجموع قیمت و گزینه حذف محصول از سبد خرید انجام می‌شود.

Cart Nav: دکمه برای ادامه خرید یا انتقال به صفحه تکمیل خرید (پرداخت).

Order Total: نمایش جمع کل قیمت محصولات در سبد. قابلیت انتخاب روش پرداخت (حضوری یا تحویل درب منزل) و دکمه ارسال سفارش (کاربر حتماً باید یکی از روش‌های ارسال را انتخاب کند در غیر این صورت دکمه ثبت سفارش فعال نمی‌شود)

Your cart is empty: در صورت خالی بودن سبد خرید، یک پیغام مناسب به کاربر نمایش داده می‌شود.

Messages: اگر پیامی برای کاربر وجود داشته باشد (به عنوان مثال، پیام‌های موفقیت برای ادا شدن محصول به سبد خرید یا خرید محصولات سبد).

:Storage

این صفحه برای مدیریت موجودی انبار است و فقط کاربران با سطح دسترسی staff می‌توانند به آن دسترسی داشته باشند.

• View

این ویو تنها برای کاربرانی با ویژگی staff قابل دسترسی است. در این ویو، ابتدا تمام اقلام موجود در انبار (storage) از مدل Storage گرفته می‌شود و در متغیر storage_items ذخیره می‌شود. همچنین تمام دسته‌های منو از مدل Category گرفته و در categories_menu ذخیره می‌شود. اگر درخواست به صورت POST ارسال شود، ابتدا بررسی می‌شود که آیا فیلدهای item_id و new_amount ارسال شده‌اند یا خیر. در صورت وجود این دو فیلد، مقدار جدید تعداد محصول با شناسه مشخص در انبار به تعداد مورد نظر بروزرسانی می‌شود. اگر فیلدهای item_id و new_amount ارسال نشده باشند، مقادیر name و amount اقلام جدید اضافه شده به انبار گرفته می‌شود و در انبار ذخیره می‌شود. در نهایت، تمپلیت storage.html با ارسال storage_items و categories_menu به کاربر نمایش داده می‌شود.

• HTML

این تمپلیت از base.html ارث بری کرده و از فایل‌های استاتیک استفاده می‌کند. در بخش محتوا (content) تمپلیت، یک جدول نمایش داده می‌شود که شامل ردیف‌های مربوط به اقلام موجود در انبار است. برای هر محصول، نام و تعداد موجودی نمایش داده می‌شود و یک فرم برای به‌روزرسانی تعداد محصول وجود دارد. برای اضافه کردن محصول جدید به انبار، یک فرم اضافه کردن نمایش داده می‌شود که شامل ورود اطلاعات نام و تعداد محصول است.

:Store management

• View

این صفحه برای مدیریت فروشگاه است و اطلاعات مربوط به محصولات و فروش را نمایش می‌دهد. فقط کاربران با سطح دسترسی staff می‌توانند به این صفحه دسترسی داشته باشند. در واقع این تابع لیستی از محصولات را از مدل دریافت می‌کند و اطلاعات مربوط به فروش یک محصول خاص را بر اساس product_id جمع‌آوری و نموداری از تاریخ سفارشات و مقدار آن‌ها رسم می‌کند. در این ویو، ابتدا همه محصولات از مدل Product گرفته می‌شود و در products ذخیره می‌شود. اطلاعات فروش در sales_data ذخیره می‌شود و چارت نیز تهیه می‌شود. چارت با استفاده از پکیج matplotlib ایجاد شده است. به منظور استفاده از matplotlib، یک فایل utils.py در اپ ایجاد کرده و سپس فانکشن get_plot در Views از utils ایمپورت می‌کنیم. انتخاب محصول برای نمایش چارت و فروش‌ها بر اساس آی‌دی محصول انجام می‌شود. اگر آی‌دی محصول انتخاب شود، فروش‌های مربوط به آن محصول استخراج می‌شود و یک چارت بر اساس تاریخ‌های سفارش ایجاد می‌شود. اگر هیچ سفارشی برای محصول انتخاب شده وجود نداشته باشد، پیام "No orders found for the selected product" چاپ می‌شود.

• HTML

این تمپلیت از base.html ارث بری کرده و در قسمت مربوط به محتوا (content) اطلاعات نمایش داده می‌شود. در بخش محتوا، یک سایدبار و یک بخش اصلی داریم. سایدبار حاوی لینک‌های "Storage Management" و "Add Product" است. در بخش اصلی، محصولات به صورت یک لیست کشویی (select) نمایش داده شده‌اند. جاوااسکریپت نیز برای بروزرسانی چارت براساس محصول انتخابی و تغییرات در لیست کشویی استفاده شده است. اطلاعات فروش مربوط به محصول انتخابی و چارت (اگر وجود داشته باشد) نمایش داده می‌شود.

:Login

این صفحه برای ورود کاربران استفاده می‌شود.

• View:

این تابع ورودی کاربر را بررسی می‌کند و اگر این ورودی بر حسب ایمیل باشد نام کاربری مرتبط به ایمیل را پیدا کرده و اگر رمز عبور نیز مطابقت داشت، کاربر را وارد سایت می‌کند. همچنین کاربران با سطح دسترسی staff به صفحه مدیریت فروشگاه و مشتریان را به صفحه پروفایل هدایت می‌کند.

• HTML:

یک عنوان Log In to Your Account به عنوان عنوان صفحه در سایت آورده شده است. یک فرم با دو فیلد نام کاربری یا ایمیل و رمز عبور برای ورود به حساب کاربری ایجاد شده است که در بخش فرم توضیح داده شده است و یک دکمه ثبت برای ورود به حساب کاربری قرار داده شده است. یک شرط login_failed وجود دارد که اگر ورود ناموفق بود، یک پیام خطا رنگی از نوع Invalid username or password. Please try again. نمایش می‌دهد. یک لینک جهت ایجاد حساب جدید برای افرادی که عضو جدید هستند نیز وجود دارد. این تمپلیت به کاربران امکان ورود به حساب کاربری خود را ارائه می‌دهد و اگر ورود ناموفق باشد، آنها را به ایجاد حساب جدید هدایت می‌کند.

Logout:

این صفحه برای خروج کاربران از سیستم استفاده می‌شود و آنها را به صفحه اصلی برمی‌گرداند. یک تابع ویو ساده به این منظور ایجاد شده است که در navbar قرار دارد.

Navbar HTML:

این قسمت HTML شامل کدهای مربوط به یک نوار ناوبری (Navigation bar) برای وبسایت است. این کد به وبسایت اجازه می‌دهد تا از فونت‌های گوگل استفاده کند. در این حالت، از فونت Material Symbols Outlined استفاده شده است. کلاس navbar یک نوار ناوبری برای وبسایت ایجاد می‌کند که شامل لوگو و گزینه‌های منوی مختلف می‌باشد. این کد categories_menu را از ویو دریافت نموده و یک منو می‌سازد. این تمپلیت شامل نوار ناوبری با عنوان "MRS.M Coffee Shop" است که شامل گزینه‌های مختلفی مانند خانه، دسته‌بندی‌ها، تماس، مدیریت فروشگاه، سبد خرید، تاریخچه سفارشات، ورود و خروج کاربران است.

Footer HTML:

این قسمت کد، بخش فوتر یک وبسایت را نشان می‌دهد که شامل اطلاعات مربوط به خدمات مشتری و پیوندهای اجتماعی می‌باشد. استایل آن به صورت سیاه با متن سفید تنظیم شده است و شامل دو ستون است. در ستون اول خدمات مشتری و در ستون دوم پیوندهای اجتماعی نمایش داده می‌شود.

Base HTML:

این فایل اجزای ثابت همه صفحه‌ها را شامل می‌شود، مانند هدر، فوتر، تگهای اولیه و در این کد همچنین می‌توان قالبهای Bootstrap را بارگذاری کرد. همچنین Footer.html و Navbar.html در این کد بارگذاری می‌شود.

با تشکر از توجه شما