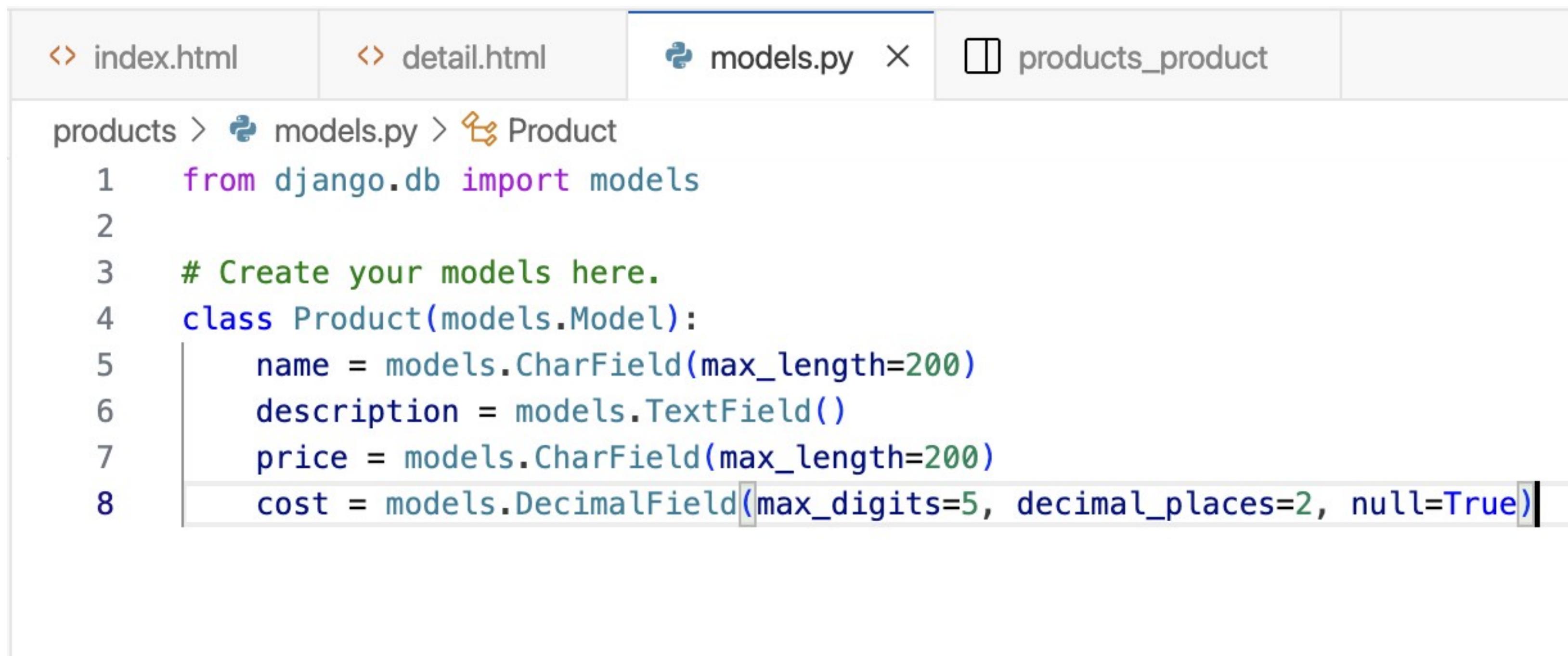


# Assignment 1: full CRUML with Styles

Requirement 1: Read object information with List and Detail Views. (35%)

Product models.py



The screenshot shows a code editor interface with the following details:

- Tab bar: index.html, detail.html, models.py (selected), products\_product
- Breadcrumbs: products > models.py > Product
- Code content:

```
1 from django.db import models
2
3 # Create your models here.
4 class Product(models.Model):
5     name = models.CharField(max_length=200)
6     description = models.TextField()
7     price = models.CharField(max_length=200)
8     cost = models.DecimalField(max_digits=5, decimal_places=2, null=True)
```

# Assignment 1: full CRUCL with Styles

Database connection successful with the new table products\_product

The screenshot shows a database management interface with a sidebar on the left listing databases and tables. The main area displays a table named 'products\_product' with the following schema:

	* id	* name	* description	* price	cost
	INTEGER	varchar(200)	TEXT	varchar(200)	
1	1	MAKE UP FOR EVER	Ultra HD Invisible Cover Foun	\$40.00	100
2	2	ANASTASIA BEVERLY HILLS	Ultra HD Invisible Cover FOUn	\$18.00	100
3	3	SURRAT BEAUTY	Lid Lacquer	\$100.00	100
4	4	BECCA	Be A Light Face Palette	\$46.00	100
8	8	BENEFIT COSMETIC'S	Hello Happy Soft Blur Foundat	\$29.00	100
10	10	FENTY BEAUTY BY RIHANNA	Pro Filt'r Soft Matte Longwear	\$34.00	100
14	14	New Product added	Description	50.00	100

A new row is being inserted with the following values:

- id: 14
- name: New Product added
- description: Description
- price: 50.00
- cost: 100

# Assignment 1: full CRUML with Styles

Admin URL with Products added

The screenshot shows the Django admin dashboard at `localhost:8000/admin/`. The top navigation bar includes links for 'View site / Change password / Log out'. On the left, there are two main sections: 'AUTHENTICATION AND AUTHORIZATION' containing 'Groups' and 'Users', and 'PRODUCTS' containing 'Products'. Each section has 'Add' and 'Change' buttons. To the right, there's a sidebar titled 'Recent actions' which is currently empty, followed by 'My actions' and a message 'None available'.

← → ⌛ ⓘ localhost:8000/admin/ 🔑 ☆ 📸

Django administration

WELCOME, ADMIN. [VIEW SITE / CHANGE PASSWORD / LOG OUT](#)

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#) [Change](#)

Users [+ Add](#) [Change](#)

PRODUCTS

Products [+ Add](#) [Change](#)

Recent actions

My actions

None available

# Assignment 1: full CRUML with Styles

Admin URL with Products list and its details

The screenshot displays two side-by-side views of the Django admin interface.

**Left View (Products List):**

- Header:** WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
- Breadcrumbs:** Home > Products > Products
- Search:** Start typing to filter...
- Actions:** ADD PRODUCT +
- Table Headers:** Action, NAME
- Table Data:** New Product added, FENTY BEAUTY BY RIHANNA, BENEFIT COSMETIC'S, BECCA, Surrat Beauty, ANASTASIA BEVERLY HILLS, MAKE UP FOR EVER
- Bottom:** 7 products

**Right View (Product Detail):**

- Header:** WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
- Breadcrumbs:** Home > Products > Products > Product object (10)
- Search:** Start typing to filter...
- Actions:** HISTORY
- Form Fields:**
  - Name:** FENTY BEAUTY BY RIHANNA
  - Description:** Pro Filt'r Soft Matte Longwear Foundation
  - Price:** \$34.00
  - Cost:** 100.00
- Buttons:** SAVE, Save and add another, Save and continue editing, Delete

# Assignment 1: full CRUML with Styles

## Home Page - Beauty Product use case

The screenshot shows a web browser window for localhost:8000. The header includes navigation icons (back, forward, search), a star icon, and a user profile icon. The main content features a large banner with four women's faces and the text "SKINCARE THAT'S SO YOU" and "Brighter, clearer, younger-looking skin happens when you find your perfect products." Below the banner are four buttons: "PORES", "WRINKLES", "SIGNS OF AGING", and "DULLNESS". At the bottom left is a "WEEKLY SPECIALS" section with a "USE CODE SKINSAVER" offer and a "FREE with \$25 purchase" offer. At the bottom right is a "BRANDS TO KNOW" section featuring products from brands like KOPARI and Skin Laundry.

← → ⌂ ⓘ localhost:8000

★

SEPHORA

Home

Beauty must-haves

SKINCARE THAT'S SO YOU

Brighter, clearer, younger-looking skin happens when you find your perfect products.

SELECT YOUR CONCERN

PORES ▶ WRINKLES ▶ SIGNS OF AGING ▶ DULLNESS ▶

WEEKLY SPECIALS

Try a deluxe sample to address your skin concern.

USE CODE **SKINSAVER** ▶

FREE with \$25 purchase

BRANDS TO KNOW

SHOP SCOUTED BY SEPHORA ▶

# Assignment 1: full CRUML with Styles

## Detail view: Product list view

localhost:8000/smart/products

Home    Beauty must-haves    SEPHORA

Add new product

Product	Description	Price
MAKE UP FOR EVER Ultra HD Invis...	\$40.00	
ANASTASIA BEVERLY HILLS Ultra HD Invis...	\$18.00	
SURRAT BEAUTY Lid Lacquer	\$100.00	

PDF Reader X

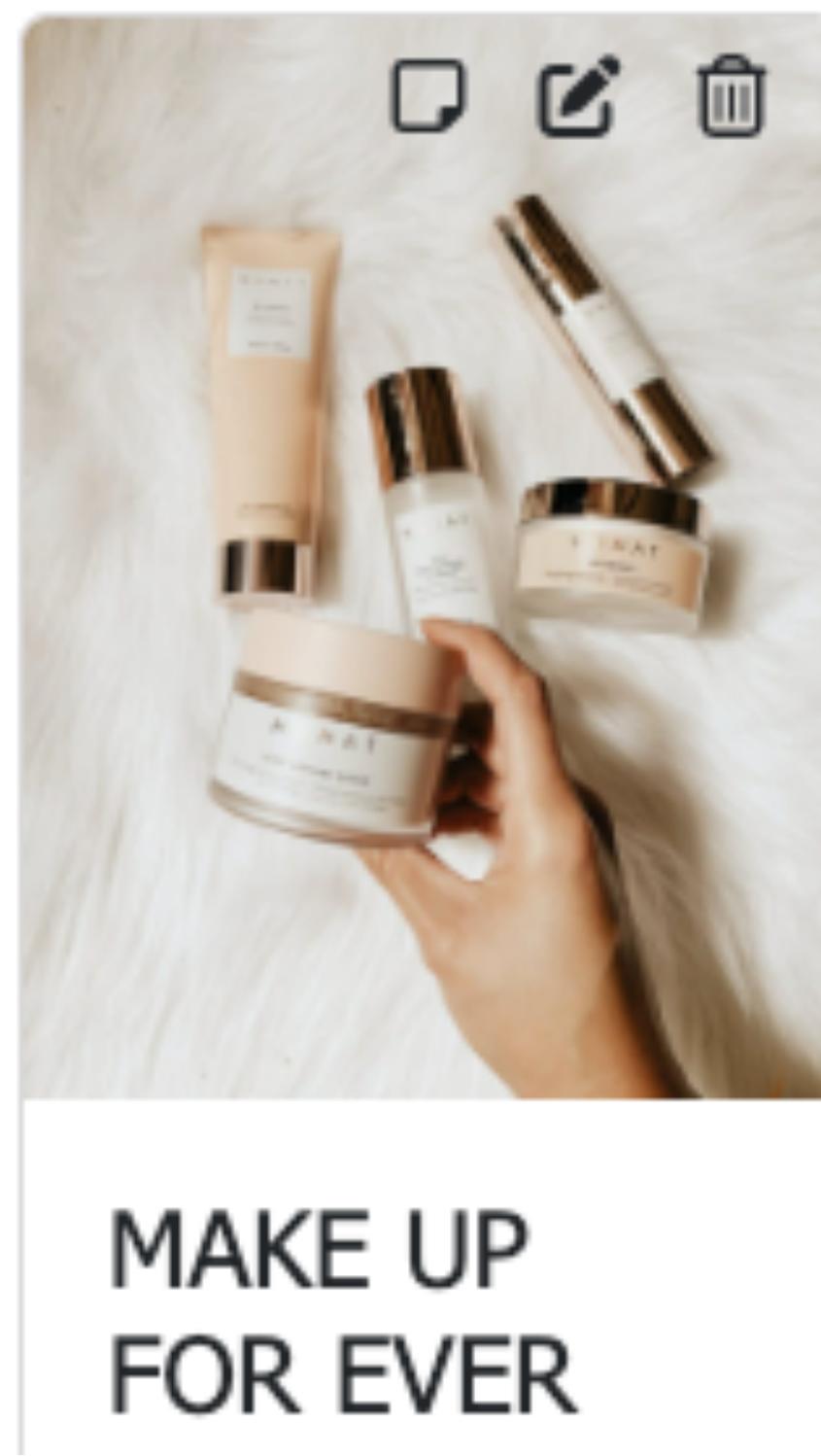
# Assignment 1: full CRUML with Styles

Detail view: Product detail list view and also a button to go back to Product list page



# Assignment 1: full CRUML with Styles

Requirement 2: Write Objects and Object Information with Create, Update and Delete Views.



# Assignment 1: full CRUML with Styles

Requirement 2: Write Objects and Object Information with Create, Update and Delete Views.

Create new Product:

The screenshot shows a web browser window with the URL `localhost:8000/smart/products/new` in the address bar. The page has a light gray header with a dark green navigation bar containing 'Home' and 'Beauty must-haves' links, and the 'SEPHORA' logo. The main content area contains three input fields with placeholder text: 'What is the Product name?', 'What is the description of your product?', and 'What is the price of the product?'. Below these fields is a blue rectangular button labeled 'Add new Product'.

What is the Product name?

What is the description of your product?

What is the price of the product?

Add new Product

# Assignment 1: full CRUML with Styles

Requirement 2: Write Objects and Object Information with Create, Update and Delete Views.

Create new Product with validation check:

The screenshot shows a web browser window with the URL `localhost:8000/smart/products/new`. The page is styled to look like the Sephora website, featuring a green header bar with the Sephora logo and navigation links for 'Home' and 'Beauty must-haves'. The main content area contains a form for creating a new product. The fields are as follows:

- Product name:** A text input field containing the value "New Product added to test the length and check if validation works".
- Description:** A text input field containing the value "sd".
- Price:** A text input field containing the value "50".

At the bottom of the form is a blue button labeled "Add new Product". Below the form, a red horizontal bar displays a validation error message: "\* name must not be more than 200 characters".

# Assignment 1: full CRUML with Styles

Requirement 2: Write Objects and Object Information with Create, Update and Delete Views.

Update Product:

The screenshot shows a web browser window with the URL `localhost:8000/smart/products/14/update`. The page has a header with the Sephora logo and navigation links for Home and Beauty must-haves. The main content area contains fields for updating a product:

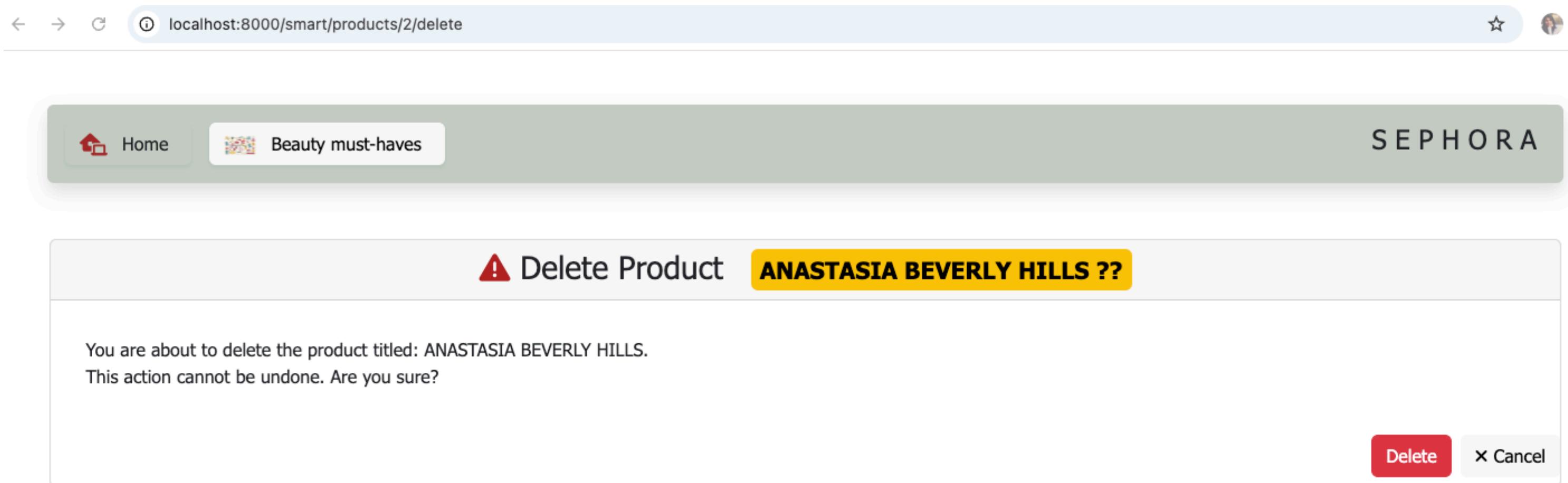
- What is the Product name?**: A text input field containing "New Product added".
- What is the description of your product?**: A large text area labeled "Description" which is currently empty.
- What is the price of the product?**: A text input field containing "50.00".

At the bottom left is a blue button labeled "Add new Product".

# Assignment 1: full CRUML with Styles

Requirement 2: Write Objects and Object Information with Create, Update and Delete Views.

Delete Product:



# Assignment 1: full CRUML with Styles

Requirement 3: Develop effective UIX Design (25%).

- Have used static images from google for background images and product images
- Have Used font awesome for other icons like update, create, delete, Home
- Have also used color picker to get some particular hex code value
- Have used card layout for listing the products and its detail