**Range Filter**

**1. Task Description**

The task is to build a Django web application that stores products with attributes such as name, price, and size, and dynamically segments these products into user-defined ranges based on price and size. The number of segments is configurable via a web interface, allowing flexible segmentation for analysis.

The application includes:

* A Product model with fields: name, price, and size.
* A seed management command to auto-generate 30 random products for testing.
* Views and templates to display segmented products grouped by price and size.
* Dynamic form controls on the web page to specify the number of segments.

This project demonstrates Django's capabilities in database modeling, management commands, dynamic filtering, and template rendering.

**2. Task Output Screenshot**

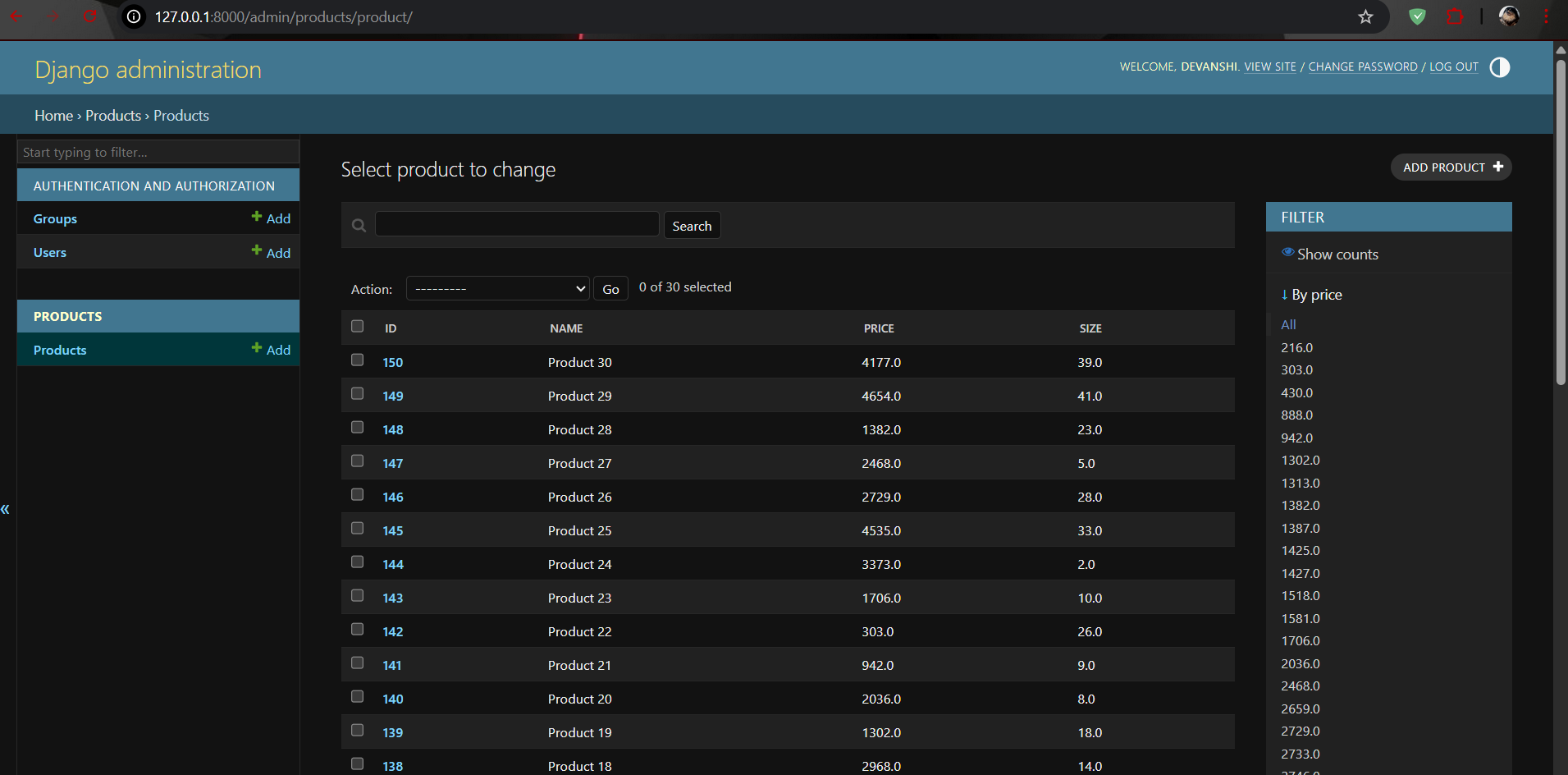
The project generates a web page that displays products grouped into ranges for both price and size. The output includes segments dynamically calculated based on the user-specified number of buckets. Each segment lists the products falling into that range.

**Example usage:**

1. Run `python manage.py seed` to populate 30 products.
2. Run the server with `python manage.py runserver`.
3. Open `http://127.0.0.1:8000/admin/` in the browser.

A screenshot of a computer program

AI-generated content may be incorrect.



**3. Widget/Algorithm Used In Task**

The task uses Django framework for building the application. The key functions and algorithms are:  
  
- Django ORM: Handles product model creation, querying, and filtering.  
- Range Calculation: Divides min-to-max price and size values into N equal ranges (segments).  
- Template Rendering: Renders buckets and product data dynamically via Jinja-style templates.  
- Bootstrap Components: Used for styling, layout, and responsive design.  
- JSON Fixtures: Provide initial dataset of 30 products.  
  
These components together enable efficient segmentation, data display, and dynamic adjustment of product groups based on user input.