

# Django Sales App Project Report

Group Members: Name1, Name2, Name3, Name4

April 29, 2025

## 1 Introduction

Provide an overview of your Django Sales App here. Discuss the purpose of the application and its main features. You might mention the key pages (Home, Dashboard, Browse Sales, etc.) and the goals of the project.

## 2 Part a: HTML and Django Template Code Implementation

In this section, include your HTML and Django template code for the project layout.

### 2.1 Base Template Explanation

Explain the structure of your base template (e.g., navigation bar, hero/jumbotron section, footer). Discuss how you set up the content blocks that child templates will override.

### 2.2 Child Template Example

Provide an example of a child template that extends the base template. For example, this could be the HTML for the Home or Browse Sales page. Show how it uses template blocks from the base template.

### 2.3 Use of Template Tags

Describe the Django template tags you used and their purpose. Examples include `{% extends %}`, `{% block %}`, and `{% static %}`. Explain how each tag helps organize your templates.

### 2.4 Bonus: URL and View Structure

(Optional) Show how your `urls.py` and `views.py` relate to the templates above. Include any relevant snippets of URL patterns or view functions.

### 3 Part b: Benefits of Django for Responsive Web Apps

Discuss why Django is beneficial for building responsive web applications. Mention Django's template system (separation of concerns), integration with CSS frameworks (like Bootstrap), and built-in features (such as security and scalability).

### 4 Part c: Linking Static CSS and JavaScript in Django

Explain how to include static files (CSS, JavaScript) in your Django project:

1. Configure `STATIC_URL` in `settings.py`.
2. Use `{% load static %}` at the top of your templates.
3. Reference static files with `{% static 'path/to/filename.css' %}` in your HTML.

### 5 Part d: Components of MTV Architecture in Django

Describe the Model-Template-View (MTV) components in Django:

- **Model:** Define its role in data management (e.g., in this project, Sales models handle product data).
- **View:** Describe how views handle requests and generate responses (e.g., your view functions or classes for Home, Browse, etc.).
- **Template:** Explain the role of templates in rendering HTML (e.g., the HTML files you created for each page).