కళల Bazaar

Sub Title: Handicraft & Artisan Product E-commerce Platform

Team Name : Keystones

Problem Statement & Solution Proposal

An E-Commerce Platform specifically for Handicrafts & Artisan Products that can provide a significant boost to artisans by giving them a dedicated marketplace to showcase and sell their unique creations, ensuring an engaging user experiences?

To design and develop a robust, user-friendly E-Commerce Platform specifically tailored for Selling and Purchasing of Handmade & Artisan Products unlike other available commercial platforms for sale of products for Profits.
The platform aims to connect Artisans with Customers, allowing the showcase of their Unique Products & Artistic Expression.

Developed using Python Django Framework

Our e-commerce platform for handicraft and artisan products is built using the robust Python Django web framework. Django's powerful features, such as its ORM, admin interface, and built-in security, provide a solid foundation for our platform.

Python

Python's simplicity and flexibility make it an ideal choice for web development, allowing our team to write clean, maintainable code.

Django

The Django framework provides a mature, battle-tested solution for building web applications, with a focus on rapid development and best practices.

Database

We securely store product info, user data, and orders using **SQLite3 Database**, ensuring the integrity of our e-commerce ecosystem.

Project Overview

Objective

Create an e-commerce platform for selling handicraft and artisan products.

Key Features

- User registration and email activation.
- Product listing with images, descriptions, and prices.
- Shopping cart and checkout process.

Target Audience

The platform will cater to conscious consumers seeking one-of-a-kind, ethically-sourced handicraft products that support local economies and preserve traditional craftsmanship.

Technology Stack

Backend

The platform is built using the **Python Django framework**, a robust and flexible web framework that simplifies the development of complex e-commerce applications.

Database

The system utilizes a **PostgreSQL database** to store product information, user data, and order details, ensuring reliable and scalable data management.
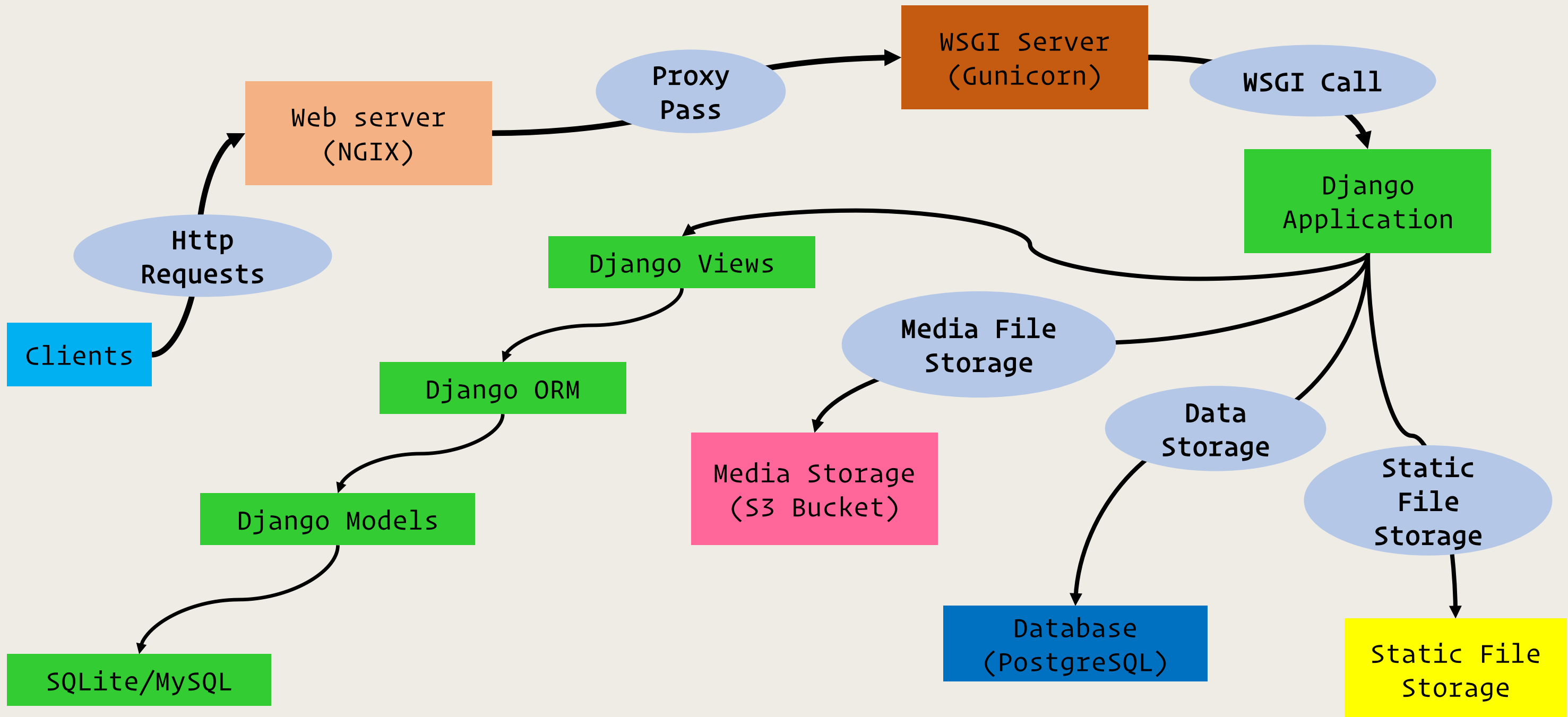
Frontend

The user interface is developed with **HTML**, **CSS**, **Javascript**, **Bootstrap** a popular JavaScript library that enables the creation of dynamic and responsive web pages.

Other Tools

The platform integrates with **Django Admin, Django Authentication.**

Project Architecture

Clients --Http Requests--> **Web server (NGIX)** --Proxy Pass--> **WSGI Server (Gunicorn)** --WSGI Call--> **Django Application**

Django Application --> **Django Views**

Django Views --> **Django ORM** --> **Django Models** --> **SQLite/MySQL**

Django Application --Media File Storage--> **Media Storage (S3 Bucket)**

Django Application --Data Storage--> **Database (PostgreSQL)**

Django Application --Static File Storage--> **Static File Storage**

Product Listing

Features:
- Display products with images, descriptions, and prices.
- Category and size variants.

Shopping Cart and Checkout

Features:

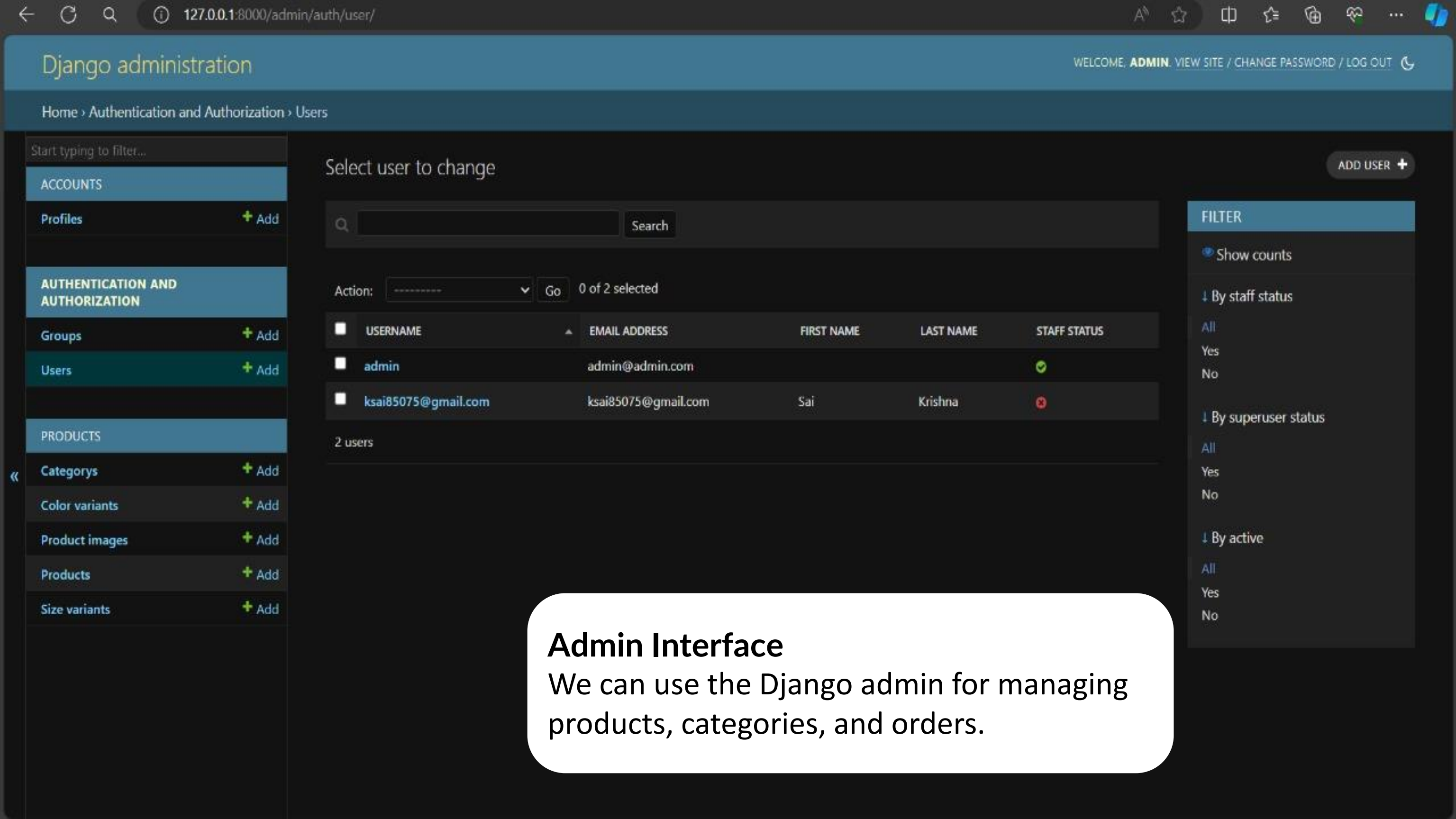- Add products to cart.
- View and manage cart items.
- Proceed to checkout.

Code Snippet:

```
<a href="#" class="btnbtn-outline-primary">Add to cart<i class="fas fa-shooping-cart"></i></a>
```

Admin Interface
We can use the Django admin for managing products, categories, and orders.

Challenges & Solutions

Challenges:

- Email configuration and sending.
- Managing product images and variants.

Solutions:

- Configure Django to use SMTP settings for sending email. Set up environment variables for sensitive information such as email host, port, username, and password to keep them secure.
- Created a Product Image model to associate multiple images to a product. Utilized Django's Image Field for handling image uploads.

Future Enhancements

- Implementing a payment gateway.

- Adding user reviews and ratings.

- Enhancing the user interface.

THANK YOU