

# Django Marketplace Website Project Scope

## Overview

This project is to build a Swappa-style marketplace using Django. The goal is to launch a platform to buy, sell, and manage used Apple devices. The website must be scalable, mobile-friendly, and API-ready for a future mobile app.

## Core Backend Features (Django)

- Django latest LTS version + Django REST Framework (DRF)
- Custom user model with login/registration/password reset
- Secure authentication system
- Database: PostgreSQL or MySQL
- REST API for mobile app integration

## User Features

- User registration, login, and email verification
- Seller verification (optional ID upload)
- User profile with dashboard and listings
- Password reset and account management

## Product Listings

- Add/Edit/Delete device listings
- Images, price, description, condition, IMEI
- Device categories and tags
- Auto status (sold, pending, available)

## Search and Filters

- Search by brand/model, condition, carrier, price

- Filters for product attributes
- Sorting (price, popularity, date)

## **Marketplace Functionality**

- Buy Now option with Stripe or PayPal integration
- Buyer-Seller messaging (optional admin mediation)
- Ratings and reviews for sellers
- Order status tracking

## **Admin Tools**

- Admin dashboard to manage users, products, sales
- Ban or verify sellers
- Export CSV reports for sales and inventory
- Moderate listings and reviews

## **Frontend**

- Responsive UI (Bootstrap or Tailwind)
- Homepage with featured listings
- Product details and checkout pages
- User dashboard and profile pages

## **Deployment**

- Live deployment on Heroku, DigitalOcean, or AWS
- Custom domain setup + SSL
- Gunicorn + Nginx for production
- GitHub version control with access to code

## **Mobile App Readiness**

- All features exposed via REST API

- Token authentication (JWT or similar)
- API documentation with Swagger or Postman

## **Ongoing Support and Handoff**

- Provide GitHub access and full documentation
- README with setup and deployment instructions
- 2-4 weeks post-launch support for fixes and questions
- Clean, well-commented code for future development