accounts\templates\accounts\2fa_status.html

```
{# templates/accounts/2fa_status.html #}
{% extends 'accounts/base_accounts.html' %}

{% block title %}2FA Status{% endblock %}

{% block content %}
<div class="container mt-5">
    <div class="row justify-content-center">
        <div class="col-md-6">
            <div class="card">
                <div class="card-header bg-primary text-white">
                    <h4 class="mb-0">Two-Factor Authentication</h4>
                </div>
                <div class="card-body">
                    {% if is_2fa_enabled %}
                        <div class="alert alert-success">
                            <h5 class="alert-heading">✓ 2FA is Enabled</h5>
                            <p>Your account is protected with two-factor authentication.</p>
                        </div>

                        <div class="mb-4">
                            <h6 class="text-muted">Authenticator App</h6>
                            <p>You're using an authenticator app to generate security codes.</p>
                        </div>

                        <hr>

                        <div class="d-grid gap-2">
                            <a href="{% url 'accounts:show_recovery_codes' %}" class="btn btn-outline-primary">
                                <i class="bi bi-key"></i> View Recovery Codes
                            </a>
                            <a href="{% url 'accounts:regenerate_recovery_codes' %}" class="btn btn-outline-secondary">
                                <i class="bi bi-arrow-clockwise"></i> Regenerate Recovery Codes
                            </a>
                            <form method="post" action="{% url 'accounts:disable_2fa' %}" style="display: inline;">
                                {% csrf_token %}
                                <button type="submit" class="btn btn-outline-danger w-100"
                                                onclick="return confirm('Are you sure you want to disable 2FA? This will reduce your account security.');">
                                    <i class="bi bi-shield-x"></i> Disable 2FA
                                </button>
                            </form>
                        </div>
                    {% else %}
                        <div class="alert alert-warning">
                            <h5 class="alert-heading">⚠ 2FA is Not Enabled</h5>
                            <p>Your account is not currently protected with two-factor authentication.</p>
                        </div>

                        <p>Enabling 2FA adds an extra layer of security to your account by requiring a code from your phone in addition to your password.</p>

                        <div class="d-grid">
                            <a href="{% url 'accounts:setup_2fa' %}" class="btn btn-primary btn-lg">
                                <i class="bi bi-shield-check"></i> Enable 2FA Now
                            </a>
                        </div>

                        <hr>
                        <h6>Why enable 2FA?</h6>
                        <ul class="small text-muted">
                            <li>Protects against unauthorized access even if your password is compromised</li>
                            <li>Required for enhanced account security</li>
                            <li>Takes less than 2 minutes to set up</li>
                        </ul>
                    {% endif %}
                </div>
            </div>
        </div>
    </div>
</div>
{% endblock %}
```

templates\errors\400.html

```
{# templates/errors/400.html #}
{% extends "accounts/base_accounts.html" %}
{% block title %}Bad request · Home Craft 3D{% endblock %}

{% block content %}
  <div class="card" style="max-width:680px;margin:40px auto;padding:28px;">
    <div class="text-center">
      <div class="text-uppercase text-secondary fw-bold" style="font-size:0.75rem;letter-spacing:0.08em;">Error 400</div>
      <h1 style="margin:8px 0 10px;font-size:30px;font-weight:600;">Bad request</h1>
      <p class="text-secondary" style="margin:0 auto 18px;max-width:520px;">
        The request couldn’t be processed. Please check the URL and try again.
      </p>
      <div class="d-flex justify-content-center gap-2 flex-wrap">
        <a class="btn" href="{% url 'core:home' %}">Go home</a>
        <a class="btn secondary" href="{% url 'products:list' %}">Browse products</a>
      </div>
    </div>
  </div>
{% endblock %}
```

templates\errors\403.html

```
{# templates/errors/400.html #}
{% extends "accounts/base_accounts.html" %}
{% block title %}Access denied · Home Craft 3D{% endblock %}

{% block content %}
  <div class="card" style="max-width:680px;margin:40px auto;padding:28px;">
    <div class="text-center">
      <div class="text-uppercase text-secondary fw-bold" style="font-size:0.75rem;letter-spacing:0.08em;">Error 403</div>
      <h1 style="margin:8px 0 10px;font-size:30px;font-weight:600;">Access denied</h1>
      <p class="text-secondary" style="margin:0 auto 18px;max-width:520px;">
        You don’t have permission to view this page.
      </p>
      <div class="d-flex justify-content-center gap-2 flex-wrap">
        <a class="btn" href="{% url 'core:home' %}">Go home</a>
        <a class="btn secondary" href="{% url 'products:list' %}">Browse products</a>
      </div>
    </div>
  </div>
{% endblock %}
```

templates\errors\404.html

```
{# templates/errors/404.html #}
{% extends "accounts/base_accounts.html" %}
{% block title %}Page not found · Home Craft 3D{% endblock %}

{% block content %}
  <div class="card" style="max-width:680px;margin:40px auto;padding:28px;">
    <div class="text-center">
      <div class="text-uppercase text-secondary fw-bold" style="font-size:0.75rem;letter-spacing:0.08em;">Error 404</div>
      <h1 style="margin:8px 0 10px;font-size:30px;font-weight:600;">Page not found</h1>
      <p class="text-secondary" style="margin:0 auto 18px;max-width:520px;">
        We couldn’t find the page you’re looking for. It may have been moved or removed.
      </p>
      <div class="d-flex justify-content-center gap-2 flex-wrap">
        <a class="btn" href="{% url 'core:home' %}">Go home</a>
        <a class="btn secondary" href="{% url 'products:list' %}">Browse products</a>
      </div>
    </div>
  </div>
{% endblock %}
```

templates\errors\500.html

```
{# templates/errors/500.html #}
{% extends "accounts/base_accounts.html" %}
{% block title %}Server error · Home Craft 3D{% endblock %}

{% block content %}
  <div class="card" style="max-width:680px;margin:40px auto;padding:28px;">
    <div class="text-center">
      <div class="text-uppercase text-secondary fw-bold" style="font-size:0.75rem;letter-spacing:0.08em;">Error 500</div>
      <h1 style="margin:8px 0 10px;font-size:30px;font-weight:600;">Something went wrong</h1>
      <p class="text-secondary" style="margin:0 auto 18px;max-width:520px;">
        We’re having trouble loading this page. Please try again in a few minutes.
      </p>
      <div class="d-flex justify-content-center gap-2 flex-wrap">
        <a class="btn" href="{% url 'core:home' %}">Go home</a>
        <a class="btn secondary" href="{% url 'products:list' %}">Browse products</a>
      </div>
    </div>
  </div>
{% endblock %}
```

legal\templates\legal_legal_base.html

```
{# templates/legal/_legal_base.html #}
{% extends "base.html" %}
{% block title %}{{ doc.title }} · Home Craft 3D{% endblock %}
{% block meta_title %}{{ doc.title }} · Home Craft 3D{% endblock %}
{% block meta_description %}
  {% if doc.summary %}
    {{ doc.summary|striptags|truncatechars:160 }}
  {% else %}
    {{ doc.title }} for Home Craft 3D.
  {% endif %}
{% endblock %}

{% block content %}
<div class="container-fluid" style="max-width: 980px;">
  <div class="d-flex flex-wrap align-items-center justify-content-between gap-2 mb-3">
    <div>
      <h1 class="h4 mb-1">{{ doc.title }}</h1>
      <div class="text-secondary small">
        Version {{ doc.version }} · Published
      </div>
    </div>

    <div class="d-flex gap-2 flex-wrap">
      <a class="btn btn-outline-dark btn-sm" href="{% url 'legal:terms' %}">Terms</a>
      <a class="btn btn-outline-dark btn-sm" href="{% url 'legal:privacy' %}">Privacy</a>
      <a class="btn btn-outline-dark btn-sm" href="{% url 'legal:refund' %}">Refund</a>
      <a class="btn btn-outline-dark btn-sm" href="{% url 'legal:content' %}">Content</a>
    </div>
  </div>

  <div class="card">
    <div class="card-body">
      {% block doc_content %}
        {{ doc.body }}
      {% endblock %}
    </div>

    {% if request.GET.next %}
      <div class="card-footer bg-white">
        <form method="post" action="{% url 'legal:accept' %}">
          {% csrf_token %}
          <input type="hidden" name="next" value="{{ request.GET.next }}">

          {% if not request.user.is_authenticated %}
            <div class="mb-2">
              <label class="form-label small text-secondary mb-1">Email (required for guest acceptance)</label>
              <input class="form-control" type="email" name="guest_email" placeholder="you@example.com"
required>
              <div class="form-text">Used only to record acceptance and for guest order access.</div>
            </div>
          {% endif %}

          <div class="form-check mb-2">
            <input class="form-check-input" type="checkbox" id="acceptLegal" required>
            <label class="form-check-label" for="acceptLegal">
              I have read and agree to the Terms, Privacy Policy, Refund Policy, and Content & Safety Policy.
            </label>
          </div>
          <button class="btn btn-dark" type="submit">Accept and continue</button>
          <a class="btn btn-link" href="{{ request.GET.next }}">Go back</a>
        </form>
      </div>
    {% endif %}
  </div>

  {% if request.GET.next %}
    <div class="text-secondary small mt-3">
      Tip: accept once and you’ll be returned automatically.
    </div>
  {% endif %}
</div>
{% endblock %}
```

products\templates\products_product_card.html

```
{# products/templates/products/_product_card.html #}
{# Expects: p (Product) #}

{% if show_category is None %}
  {% with True as show_category %}{% endwith %}
{% endif %}

<div class="card h-100">
  {% if p.primary_image %}
      <a href="{% url 'products:go' p.pk p.slug %}" class="d-block">
        <img
          src="{{ p.primary_image.image.url }}"
          class="card-img-top"
          alt="{{ p.primary_image.alt_text|default:p.title }}"
          style="width:100%; height:auto; max-height:{{ image_height|default:'240px' }}; object-fit:contain;"
        >
      </a>
  {% else %}
    <div class="bg-light d-flex align-items-center justify-content-center" style="height:{{ image_height|default:'180px' }};">
      <span class="text-secondary small">No image</span>
    </div>
  {% endif %}

  <div class="card-body">
    <div class="d-flex justify-content-between align-items-start gap-2">
      <h2 class="h6 mb-1">
        <a class="text-decoration-none" href="{% url 'products:go' p.pk p.slug %}">{{ p.title }}</a>
      </h2>

      <div class="d-flex flex-column align-items-end gap-1">
        {% if p.trending_badge %}
          <span class="badge" style="background:#f59e0b; color:#111827;"> Trending</span>
        {% endif %}
        {% if p.kind == "FILE" %}
          <span class="badge text-bg-success">
            <i class="bi bi-lightning-fill me-1"></i>Instant
          </span>
        {% else %}
          <span class="badge text-bg-secondary">{{ p.get_kind_display }}</span>
        {% endif %}
      </div>
    </div>

    <div class="d-flex justify-content-between align-items-center mt-1">
      <div class="text-secondary small">
        by <strong><a class="text-decoration-none" href="{% url 'products:seller_shop' p.seller_id %}">{{ p.seller_public_name }}</a></strong>
        {% if show_category|default:True %}
          {% if p.category.parent %}
            · {{ p.category.parent.name }} › {{ p.category.name }}
          {% else %}
            · {{ p.category.name }}
          {% endif %}
        {% endif %}
      </div>

      <div class="small text-secondary" style="white-space:nowrap;">
        {% if p.review_count and p.review_count > 0 %}
    ★ {{ p.avg_rating|floatformat:1 }}/5 ({{ p.review_count }})
   {% endif %}
  </div>
 </div>

 {% if p.seller_review_count and p.seller_review_count > 0 %}
  <div class="d-flex justify-content-between align-items-center mt-1">
   <div class="small text-secondary" style="white-space:nowrap;">
    Seller ★ {{ p.seller_avg_rating|floatformat:1 }}/5 ({{ p.seller_review_count }})
   </div>
  </div>
 {% endif %}

 {% if p.kind == "FILE" and p.file_types_display %}
  <div class="mt-2 d-flex flex-wrap gap-1">
   {% for file_type in p.file_types %}
    <span class="badge text-bg-light text-dark">{{ file_type }}</span>
   {% endfor %}
  </div>
 {% endif %}

 {% if p.kind == "FILE" and p.digital and p.digital.license_type %}
  <div class="mt-2">
   <span class="badge text-bg-info">{{ p.digital.get_license_type_display }}</span>
  </div>
 {% endif %}

 {% if p.complexity_level or p.print_time_hours %}
  <div class="small text-secondary mt-2">
   {% if p.complexity_level %}
    <span class="badge text-bg-secondary">{{ p.get_complexity_level_display }}</span>
   {% endif %}
   {% if p.print_time_hours %}
    <span class="badge text-bg-secondary">{{ p.print_time_hours }}h print</span>
   {% endif %}
  </div>
 {% endif %}

 {% if p.short_description %}
  <div class="text-secondary small mt-2">{{ p.short_description }}</div>
 {% endif %}
</div>

<div class="card-footer bg-white d-flex justify-content-between align-items-center">
 <strong>{{ p.display_price }}</strong>

 <div class="d-flex gap-2 align-items-center">
  {% if show_add_to_cart %}
   {% if p.can_buy %}
    <form method="post" action="{% url 'cart:add' %}" class="m-0">
     {% csrf_token %}
     <input type="hidden" name="product_id" value="{{ p.id }}">
     <input type="hidden" name="quantity" value="1">
     <input type="hidden" name="next" value="{{ request.get_full_path }}">
     <button class="btn btn-sm btn-dark" type="submit">
      <i class="bi bi-cart-plus me-1"></i>Add
     </button>
    </form>
   {% else %}
    <button class="btn btn-sm btn-secondary" type="button" disabled title="Seller not ready">
     <i class="bi bi-cart-x me-1"></i>Add
     </button>
    {% endif %}
   {% endif %}
   <!-- Store button removed from card. Only shown in product detail view. -->
    <a class="btn btn-sm btn-outline-dark" href="{% url 'products:go' p.pk p.slug %}" aria-label="View product details">View</a>
  </div>
 </div>
</div>
```

qa\templates\qa_products_tab.html

```
{# qa/templates/qa/_products_tab.html #}
{% load humanize %}

<div id="qa">
  <div class="d-flex align-items-center justify-content-between mb-2">
    <div class="fw-semibold">Product Q&amp;A</div>
    <div class="text-secondary small">{{ qa_thread_count }} thread{{ qa_thread_count|pluralize }}</div>
  </div>

  {% if user.is_authenticated %}
    <div class="card shadow-sm mb-3">
      <div class="card-body">
        <form method="post" action="{% url 'qa:thread_create' product.id %}">
          {% csrf_token %}
          <div class="row g-2">
            <div class="col-12">
              <input type="text" name="subject" class="form-control" placeholder="Optional subject" maxlength="180">
            </div>
            <div class="col-12">
              <textarea name="body" class="form-control" rows="3" placeholder="Ask a question..." required></textarea>
            </div>
            <div class="col-12 d-flex justify-content-end">
              <button class="btn btn-dark" type="submit">
                <i class="bi bi-chat-left-text me-1"></i>Post question
              </button>
            </div>
          </div>
        </form>
        <div class="text-secondary small mt-2">Only you and the seller can reply. Messages are visible on this page.</div>
      </div>
    </div>
  {% else %}
    <div class="alert alert-light border mb-3">
      <div class="d-flex align-items-center justify-content-between">
        <div>
          <div class="fw-semibold">Want to ask a question?</div>
          <div class="small text-secondary">Log in to post. Only the buyer and seller can reply.</div>
        </div>
        <a class="btn btn-outline-dark" href="{% url 'accounts:login' %}?next={{ request.path }}#qa">Login</a>
      </div>
    </div>
  {% endif %}

  {% for thread in qa_threads %}
    <div class="card shadow-sm mb-3">
      <div class="card-body">
        <div class="d-flex justify-content-between align-items-start gap-2">
          <div>
            <div class="fw-semibold">
              {% if thread.subject %}{{ thread.subject }}{% else %}Question{% endif %}
            </div>
            <div class="small text-secondary">
              Started by <strong>{{ thread.buyer.username }}</strong> · {{ thread.created_at|naturaltime }}
            </div>
          </div>
          <button class="btn btn-sm btn-outline-secondary" type="button" data-bs-toggle="collapse"
```

```
data-bs-target="#thread{{ thread.id }}">
            View
          </button>
        </div>

        <div class="collapse mt-3" id="thread{{ thread.id }}">
          {% for m in thread.messages.all %}
            <div class="border rounded p-2 mb-2 {% if m.is_deleted %}opacity-75{% endif %}">
              <div class="d-flex justify-content-between align-items-center">
                <div class="small fw-semibold">{{ m.author.username }}</div>
                <div class="small text-secondary">{{ m.created_at|naturaltime }}</div>
              </div>

              {% if m.is_deleted %}
                <div class="small text-secondary fst-italic mt-1">(Message deleted)</div>
              {% else %}
                <div class="small mt-2" style="white-space: pre-wrap;">{{ m.body }}</div>

                <div class="d-flex gap-2 justify-content-end mt-2">
                  {% if user.is_authenticated %}
                    <form method="post" action="{% url 'qa:message_report' m.id %}">
                      {% csrf_token %}
                      <input type="hidden" name="reason" value="other">
                      <input type="hidden" name="details" value="">
                      <button class="btn btn-sm btn-outline-danger" type="submit">
                        <i class="bi bi-flag me-1"></i>Report
                      </button>
                    </form>

                    {% if user.id == m.author_id and m.can_author_delete_now %}
                      <form method="post" action="{% url 'qa:message_delete' m.id %}">
                        {% csrf_token %}
                        <button class="btn btn-sm btn-outline-secondary" type="submit">Delete</button>
                      </form>
                    {% elif user.is_staff or user.is_superuser %}
                      <form method="post" action="{% url 'qa:message_delete' m.id %}">
                        {% csrf_token %}
                        <button class="btn btn-sm btn-outline-secondary" type="submit">Staff delete</button>
                      </form>
                    {% endif %}
                  {% endif %}
                </div>
              {% endif %}
            </div>
          {% endfor %}

          {% if user.is_authenticated %}
            {% if user.id == thread.buyer_id or user.id == product.seller_id or user.is_staff or user.is_superuser %}
              <form method="post" action="{% url 'qa:reply_create' thread.id %}" class="mt-2">
                {% csrf_token %}
                <div class="input-group">
                  <textarea name="body" class="form-control" rows="2" placeholder="Reply…" required></textarea>
                  <button class="btn btn-dark" type="submit">Reply</button>
                </div>
              </form>
            {% else %}
              <div class="text-secondary small">Only the buyer and seller can reply in this thread.</div>
            {% endif %}
          {% endif %}
        </div>
      </div>
```

```
    </div>
  {% empty %}
    <div class="text-secondary small">No Q&amp;A yet.</div>
  {% endfor %}

  <div class="text-secondary small mt-3">
  </div>
</div>
```

reviews\admin.py

```
# reviews/admin.py
from django.contrib import admin

from .models import Review, SellerReview


@admin.register(Review)
class ReviewAdmin(admin.ModelAdmin):
    list_display = ("id", "product", "buyer", "rating", "created_at")
    list_filter = ("rating", "created_at")
    search_fields = ("product__title", "buyer__username", "title", "body")
    readonly_fields = ("created_at", "updated_at")


@admin.register(SellerReview)
class SellerReviewAdmin(admin.ModelAdmin):
    list_display = ("id", "seller", "buyer", "rating", "created_at")
    list_filter = ("rating", "created_at")
    search_fields = ("seller__username", "buyer__username", "title", "body")
```

dashboards\templates\dashboards\admin_dashboard.html

```
{# dashboards/templates/dashboards/admin_dashboard.html #}
{% extends "base.html" %}
{% block title %}Admin Dashboard · Home Craft 3D{% endblock %}

{% block content %}
<div class="container-fluid">
  <div class="d-flex flex-wrap align-items-center justify-content-between gap-2 mb-3">
    <div>
      <h1 class="h4 mb-1">Admin Dashboard</h1>
      <div class="text-secondary small">Marketplace health and operating metrics.</div>
    </div>

    <div class="d-flex flex-wrap gap-2">
      <a class="btn btn-primary" href="{% url 'dashboards:admin_settings' %}">
        <i class="bi bi-sliders me-1"></i>Admin Settings
      </a>
      <a class="btn btn-outline-dark" href="{% url 'admin:index' %}">
        <i class="bi bi-gear me-1"></i>Django Admin
      </a>
      <a class="btn btn-dark" href="{% url 'products:list' %}">
        <i class="bi bi-shop me-1"></i>View Store
      </a>
    </div>
  </div>

  <div class="row g-3 mb-3">
    <div class="col-12 col-md-3">
      <div class="card h-100">
        <div class="card-body">
          <div class="text-secondary small">Products (total)</div>
          <div class="h4 mb-0">{{ products_total|default:"0" }}</div>
          <div class="text-secondary small mt-1">Active: {{ products_active|default:"0" }}</div>
        </div>
      </div>
    </div>

    <div class="col-12 col-md-3">
      <div class="card h-100">
        <div class="card-body">
          <div class="text-secondary small">Sellers (with listings)</div>
          <div class="h4 mb-0">{{ sellers_total|default:"0" }}</div>
        </div>
      </div>
    </div>

    <div class="col-12 col-md-3">
      <div class="card h-100">
        <div class="card-body">
          <div class="text-secondary small">Orders</div>
          <div class="d-flex justify-content-between mt-1">
            <span class="text-secondary">Paid</span>
            <span class="fw-semibold">{{ orders_paid|default:"0" }}</span>
          </div>
          <div class="d-flex justify-content-between">
            <span class="text-secondary">Pending</span>
            <span class="fw-semibold">{{ orders_pending|default:"0" }}</span>
          </div>
        </div>
      </div>
    </div>

    <div class="col-12 col-md-3">
      <div class="card h-100">
        <div class="card-body">
          <div class="text-secondary small">Revenue (last {{ since_days }} days)</div>
          <div class="h4 mb-0">${{ revenue_30|default:"0" }}</div>
        </div>
      </div>
    </div>
  </div>

  <div class="row g-3 mb-3">
    <div class="col-12 col-lg-4">
      <div class="card h-100">
        <div class="card-header bg-white">
          <div class="fw-semibold">Marketplace Settings (Live)</div>
        </div>
        <div class="card-body">
          <div class="d-flex justify-content-between">
            <span class="text-secondary">Sales % (platform cut)</span>
            <span class="fw-semibold">{{ marketplace_sales_percent }}%</span>
          </div>
          <div class="d-flex justify-content-between mt-2">
            <span class="text-secondary">Flat fee</span>
            <span class="fw-semibold">
              {% if platform_fee_cents|default:0|add:0 > 0 %}
                ${{ platform_fee_cents|floatformat:0 }}
              {% else %}
                $0
              {% endif %}
            </span>
          </div>

          <div class="text-secondary small mt-3">
            These are DB-backed settings (edit via “Site Settings”).
          </div>
        </div>
      </div>
    </div>

    <div class="col-12 col-lg-8">
      <div class="card h-100">
        <div class="card-header bg-white">
          <div class="fw-semibold">Top Sellers (last {{ since_days }} days)</div>
        </div>

        <div class="table-responsive">
          <table class="table table-hover align-middle mb-0">
            <thead class="table-light">
              <tr>
                <th>Seller</th>
                <th class="text-end">Revenue</th>
                <th class="text-end">Qty</th>
                <th class="text-end">Orders</th>
              </tr>
            </thead>
            <tbody>
              {% for s in top_sellers %}
                <tr>
                  <td class="fw-semibold">{{ s.seller__username }}</td>
```

```
                  <td class="text-end"><strong>${{ s.revenue }}</strong></td>
                  <td class="text-end">{{ s.qty }}</td>
                  <td class="text-end">{{ s.orders }}</td>
                </tr>
              {% empty %}
                <tr>
                  <td colspan="4" class="text-center text-secondary py-4">No sales yet.</td>
                </tr>
              {% endfor %}
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </div>

  <div class="row g-3 mb-3">
    <div class="col-12">
      <div class="card">
        <div class="card-header bg-white d-flex flex-wrap justify-content-between align-items-center gap-2">
          <div class="fw-semibold">Analytics (Plausible)</div>
          <div class="d-flex flex-wrap align-items-center gap-2">
            <button class="btn btn-sm btn-outline-secondary" type="button" data-bs-toggle="collapse" data-bs-target="#plausibleFilters" aria-expanded="false" aria-controls="plausibleFilters">
              <i class="bi bi-funnel me-1"></i>Filters
            </button>
          </div>
          <div class="collapse w-100" id="plausibleFilters">
            <form class="d-flex flex-wrap align-items-center gap-2 mt-2" method="get">
            <select class="form-select form-select-sm" name="period" style="min-width:160px;">
              <option value="7d" {% if plausible_period == "7d" %}selected{% endif %}>Last 7 days</option>
              <option value="30d" {% if plausible_period == "30d" %}selected{% endif %}>Last 30 days</option>
              <option value="90d" {% if plausible_period == "90d" %}selected{% endif %}>Last 90 days</option>
              <option value="6mo" {% if plausible_period == "6mo" %}selected{% endif %}>Last 6 months</option>
              <option value="12mo" {% if plausible_period == "12mo" %}selected{% endif %}>Last 12 months</option>
              <option value="year" {% if plausible_period == "year" %}selected{% endif %}>Year to date</option>
              <option value="today" {% if plausible_period == "today" %}selected{% endif %}>Today</option>
              <option value="yesterday" {% if plausible_period == "yesterday" %}selected{% endif %}>Yesterday</option>
              <option value="custom" {% if plausible_period == "custom" %}selected{% endif %}>Custom range</option>
            </select>
            {% if plausible_period == "custom" %}
              <input class="form-control form-control-sm" type="date" name="from" value="{{ plausible_from }}" title="From date">
              <input class="form-control form-control-sm" type="date" name="to" value="{{ plausible_to }}" title="To date">
            {% endif %}
            <input class="form-control form-control-sm" type="text" name="page" value="{{ plausible_page_filter }}" placeholder="Page filter" style="min-width:160px;">
            <select class="form-select form-select-sm" name="limit" style="width:110px;">
              <option value="5" {% if plausible_top_limit == 5 %}selected{% endif %}>Top 5</option>
              <option value="10" {% if plausible_top_limit == 10 %}selected{% endif %}>Top 10</option>
              <option value="15" {% if plausible_top_limit == 15 %}selected{% endif %}>Top 15</option>
              <option value="20" {% if plausible_top_limit == 20 %}selected{% endif %}>Top 20</option>
              <option value="30" {% if plausible_top_limit == 30 %}selected{% endif %}>Top 30</option>
              <option value="40" {% if plausible_top_limit == 40 %}selected{% endif %}>Top 40</option>
              <option value="50" {% if plausible_top_limit == 50 %}selected{% endif %}>Top 50</option>
            </select>
            <button class="btn btn-sm btn-outline-dark" type="submit">Apply</button>
            <a class="btn btn-sm btn-outline-secondary" href="{% url 'dashboards:admin' %}">Reset</a>
            </form>
          </div>
          <div class="d-flex gap-2">
            {% if plausible_shared_url %}
              <a class="btn btn-outline-primary btn-sm" href="{{ plausible_shared_url }}" target="_blank" rel="noopener">
                <i class="bi bi-graph-up me-1"></i>Open Plausible
              </a>
              {% if plausible_embed_url %}
                <button class="btn btn-outline-secondary btn-sm" type="button" data-bs-toggle="collapse" data-bs-target="#plausibleEmbed" aria-expanded="false" aria-controls="plausibleEmbed">
                  <i class="bi bi-layout-text-sidebar-reverse me-1"></i>Embed
                </button>
              {% endif %}
            {% else %}
              <a class="btn btn-outline-secondary btn-sm" href="{% url 'dashboards:admin_settings' %}">
                <i class="bi bi-sliders me-1"></i>Add Plausible link in Admin Settings
              </a>
            {% endif %}
          </div>
        </div>

        <div class="card-body">
          {% if plausible_api_enabled %}
            {% if plausible_api_error %}
              <div class="alert alert-warning">
                {{ plausible_api_error }}
              </div>
            {% endif %}
            <div class="row g-3 mb-3">
              <div class="col-6 col-md-4 col-xl-2">
                <div class="border rounded p-3 h-100">
                  <div class="text-secondary small">Visitors</div>
                  <div class="h5 mb-0">{{ plausible_summary.visitors|default:"0" }}</div>
                </div>
              </div>
              <div class="col-6 col-md-4 col-xl-2">
                <div class="border rounded p-3 h-100">
                  <div class="text-secondary small">Pageviews</div>
                  <div class="h5 mb-0">{{ plausible_summary.pageviews|default:"0" }}</div>
                </div>
              </div>
              <div class="col-6 col-md-4 col-xl-2">
                <div class="border rounded p-3 h-100">
                  <div class="text-secondary small">Visits</div>
                  <div class="h5 mb-0">{{ plausible_summary.visits|default:"0" }}</div>
                </div>
              </div>
              <div class="col-6 col-md-4 col-xl-2">
                <div class="border rounded p-3 h-100">
                  <div class="text-secondary small">Bounce</div>
                  <div class="h5 mb-0">{{ plausible_summary.bounce_rate|default:"0" }}%</div>
                </div>
              </div>
              <div class="col-6 col-md-4 col-xl-2">
                <div class="border rounded p-3 h-100">
                  <div class="text-secondary small">Avg. Visit</div>
                  <div class="h5 mb-0">{{ plausible_summary.visit_duration|default:"0s" }}</div>
                </div>
              </div>
              <div class="col-6 col-md-4 col-xl-2">
                <div class="border rounded p-3 h-100">
                  <div class="text-secondary small">Period</div>
                  <div class="h6 mb-0">{{ plausible_period_label }}</div>
                </div>
              </div>
            </div>

            <div class="row g-3 mb-3">
              <div class="col-12 col-lg-6">
                <div class="border rounded h-100">
                  <div class="px-3 py-2 border-bottom bg-light">
                    <span class="fw-semibold">Top Pages</span>
                  </div>
                  <div class="table-responsive">
                    <table class="table table-sm align-middle mb-0">
                      <thead>
                        <tr class="text-secondary small">
                          <th>Page</th>
                          <th class="text-end">Views</th>
                          <th class="text-end">Visitors</th>
                        </tr>
                      </thead>
                      <tbody>
                        {% for row in plausible_top_pages %}
                          <tr>
                            <td class="text-truncate" style="max-width: 280px;">{{ row.page|default:"" }}</td>
                            <td class="text-end">{{ row.pageviews|default:"0" }}</td>
                            <td class="text-end">{{ row.visitors|default:"0" }}</td>
                          </tr>
                        {% empty %}
                          <tr>
                            <td colspan="3" class="text-center text-secondary py-3">No data yet.</td>
                          </tr>
                        {% endfor %}
                      </tbody>
                    </table>
                  </div>
                </div>
              </div>
            </div>
          {% else %}
            <div class="alert alert-warning d-flex flex-wrap align-items-center justify-content-between gap-2">
              <div>
                <div class="fw-semibold">Plausible API not configured</div>
                <div class="small">Set PLAUSIBLE_API_KEY and PLAUSIBLE_SITE_ID in your environment to enable API stats.</div>
              </div>
              <a class="btn btn-outline-dark btn-sm" href="https://plausible.io/docs/stats-api" target="_blank" rel="noopener">
                Docs
              </a>
            </div>
          {% endif %}

          {% if plausible_shared_url %}
            <div class="text-secondary small mb-2">
              Uses your Plausible shared dashboard link (read-only).
            </div>

            {% if plausible_embed_url %}
```

```
              <div class="collapse" id="plausibleEmbed">
                <div class="ratio ratio-16x9 border rounded">
                  <iframe
                    src="{{ plausible_embed_url }}"
                    loading="lazy"
                    referrerpolicy="no-referrer"
                    style="border:0;"
                    title="Plausible Analytics"
                  ></iframe>
                </div>
                <div class="text-secondary small mt-2">
                  If the embed ever shows blank, use “Open Plausible” instead (some browsers/privacy settings block embeds).
                </div>
              </div>
            {% endif %}
          {% else %}
            <div class="text-secondary">
              Add your Plausible shared dashboard URL in Admin Settings to enable this section.
            </div>
          {% endif %}
        </div>
      </div>
    </div>
  </div>

</div>
{% endblock %}
```

dashboards\templates\dashboards\admin_settings.html

```
{# templates/dashboards/admin_settings.html #}
{% extends "base.html" %}
{% block title %}Admin Settings · Home Craft 3D{% endblock %}

{% block content %}
<style>
  input[type="text"],
  input[type="number"],
  input[type="email"],
  input[type="password"],
  input[type="url"],
  input[type="date"],
  input[type="time"],
  select,
  textarea {
    background-color: #ffffff !important;
    color: #333333 !important;
  }
</style>

<div class="container-fluid">
  <div class="d-flex flex-wrap align-items-center justify-content-between gap-2 mb-3">
    <div>
      <h1 class="h4 mb-1">Admin Settings</h1>
      <div class="text-secondary small">
        Manage store-wide settings without the Django admin.
        {% if site_config_updated_at %}
          <span class="ms-2">Last saved: {{ site_config_updated_at|date:"M j, Y g:i A" }}</span>
        {% endif %}
      </div>
    </div>
    <div class="d-flex flex-wrap gap-2">
      <a class="btn btn-outline-secondary" href="{% url 'dashboards:admin' %}">
        <i class="bi bi-arrow-left me-1"></i>Back to Admin
      </a>
      <a class="btn btn-outline-secondary" href="{% url 'admin:index' %}">
        <i class="bi bi-gear me-1"></i>Django Admin
      </a>
    </div>
  </div>

  {#  SHOW DJANGO MESSAGES (this was missing) #}
  {% if messages %}
    <div class="mb-3">
      {% for message in messages %}
        <div class="alert alert-{{ message.tags }} mb-2" role="alert">
          {{ message }}
        </div>
      {% endfor %}
    </div>
  {% endif %}

  <div class="card">
    <div class="card-body">
      <form method="post" novalidate>
        {% csrf_token %}

        {% if form.non_field_errors %}
          <div class="alert alert-danger">{{ form.non_field_errors }}</div>
```

```
{% endif %}

<div class="row g-3">
  <div class="col-12 col-lg-4">
    <label class="form-label">Marketplace sales percent</label>
    {{ form.marketplace_sales_percent }}
    {% if form.marketplace_sales_percent.errors %}
      <div class="text-danger small mt-1">{{ form.marketplace_sales_percent.errors }}</div>
    {% endif %}
  </div>

  <div class="col-12 col-lg-4">
    <label class="form-label">Platform fee (cents)</label>
    {{ form.platform_fee_cents }}
    {% if form.platform_fee_cents.errors %}
      <div class="text-danger small mt-1">{{ form.platform_fee_cents.errors }}</div>
    {% endif %}
  </div>

  <div class="col-12 col-lg-4">
    <label class="form-label">Default currency</label>
    {{ form.default_currency }}
    {% if form.default_currency.errors %}
      <div class="text-danger small mt-1">{{ form.default_currency.errors }}</div>
    {% endif %}
  </div>

  <div class="col-12">
    <label class="form-label">Allowed shipping countries</label>
    {{ form.allowed_shipping_countries_csv }}
    <div class="form-text">{{ form.allowed_shipping_countries_csv.help_text }}</div>
    {% if form.allowed_shipping_countries_csv.errors %}
      <div class="text-danger small mt-1">{{ form.allowed_shipping_countries_csv.errors }}</div>
    {% endif %}
  </div>
</div>

<div class="col-12 mt-3">
  <label class="form-label">Plausible shared dashboard URL</label>
  {{ form.plausible_shared_url }}
  {% if form.plausible_shared_url.help_text %}
    <div class="form-text">{{ form.plausible_shared_url.help_text }}</div>
  {% else %}
    <div class="form-text">
      Paste your Plausible shared link (ex: https://plausible.io/share/homecraft3d.com?auth=...).
    </div>
  {% endif %}
  {% if form.plausible_shared_url.errors %}
    <div class="text-danger small mt-1">{{ form.plausible_shared_url.errors }}</div>
  {% endif %}
</div>

<hr class="my-4">

<h2 class="h6 mb-3">Home Page</h2>

<div class="row g-3">
  <div class="col-12 col-lg-6">
    <label class="form-label">Hero title</label>
    {{ form.home_hero_title }}
    {% if form.home_hero_title.errors %}
```

```
      <div class="text-danger small mt-1">{{ form.home_hero_title.errors }}</div>
    {% endif %}
    {% if form.home_hero_title.help_text %}
      <div class="form-text">{{ form.home_hero_title.help_text }}</div>
    {% endif %}
  </div>

  <div class="col-12">
    <label class="form-label">Hero subtitle</label>
    {{ form.home_hero_subtitle }}
    {% if form.home_hero_subtitle.errors %}
      <div class="text-danger small mt-1">{{ form.home_hero_subtitle.errors }}</div>
    {% endif %}
    {% if form.home_hero_subtitle.help_text %}
      <div class="form-text">{{ form.home_hero_subtitle.help_text }}</div>
    {% endif %}
  </div>
</div>

<hr class="my-4">

<h5 class="mb-3">Promo Banner (sitewide above navbar)</h5>

<div class="mb-3 form-check">
  {{ form.promo_banner_enabled }}
  <label class="form-check-label" for="{{ form.promo_banner_enabled.id_for_label }}">
    Enable promo banner
  </label>
</div>

<div class="mb-3">
  <label class="form-label" for="{{ form.promo_banner_text.id_for_label }}">
    Promo banner text
  </label>
  {{ form.promo_banner_text }}
  <div class="form-text">
    Example: Sellers pay 0% marketplace fees for their first 30 days!
  </div>
  {% if form.promo_banner_text.errors %}
    <div class="text-danger small mt-1">{{ form.promo_banner_text.errors }}</div>
  {% endif %}
</div>

<hr class="my-4">

<h5 class="mb-3">Home Page Banner (home page only)</h5>

<div class="mb-3 form-check">
  {{ form.home_banner_enabled }}
  <label class="form-check-label" for="{{ form.home_banner_enabled.id_for_label }}">
    Enable home page banner
  </label>
</div>

<div class="mb-3">
  <label class="form-label" for="{{ form.home_banner_text.id_for_label }}">
    Home page banner text
  </label>
  {{ form.home_banner_text }}
  <div class="form-text">
    Example: <em>First 30 days FREE!</em>
```

```
          </div>
          {% if form.home_banner_text.errors %}
            <div class="text-danger small mt-1">{{ form.home_banner_text.errors }}</div>
          {% endif %}
        </div>

        <hr class="my-4">

        <h2 class="h6 mb-3">Theme</h2>

        <div class="row g-3">
          <div class="col-12 col-lg-4">
            <label class="form-label">Default mode</label>
            {{ form.theme_default_mode }}
          </div>

          <div class="col-12 col-lg-4">
            <label class="form-label">Primary (buttons / CTAs)</label>
            {{ form.theme_primary }}
          </div>

          <div class="col-12 col-lg-4">
            <label class="form-label">Accent</label>
            {{ form.theme_accent }}
          </div>

          <div class="col-12 col-lg-4">
            <label class="form-label">Success</label>
            {{ form.theme_success }}
          </div>

          <div class="col-12 col-lg-4">
            <label class="form-label">Danger</label>
            {{ form.theme_danger }}
          </div>
        </div>

        <div class="row g-3 mt-1">
          <div class="col-12 col-lg-6">
            <div class="border rounded p-3">
              <div class="fw-semibold mb-2">Light mode surfaces</div>
              <div class="row g-2">
                <div class="col-12 col-md-6"><label class="form-label small">BG</label>{{ form.theme_light_bg
}}</div>
                <div class="col-12 col-md-6"><label class="form-label small">Surface</label>{{
form.theme_light_surface }}</div>
                <div class="col-12 col-md-6"><label class="form-label small">Text</label>{{ form.theme_light_text
}}</div>
                <div class="col-12 col-md-6"><label class="form-label small">Muted</label>{{
form.theme_light_text_muted }}</div>
                <div class="col-12 col-md-6"><label class="form-label small">Border</label>{{
form.theme_light_border }}</div>
              </div>
            </div>
          </div>

          <div class="col-12 col-lg-6">
            <div class="border rounded p-3">
              <div class="fw-semibold mb-2">Dark mode surfaces</div>
              <div class="row g-2">
                <div class="col-12 col-md-6"><label class="form-label small">BG</label>{{ form.theme_dark_bg
}}</div>
                  <div class="col-12 col-md-6"><label class="form-label small">Surface</label>{{
form.theme_dark_surface }}</div>
                  <div class="col-12 col-md-6"><label class="form-label small">Text</label>{{ form.theme_dark_text
}}</div>
                  <div class="col-12 col-md-6"><label class="form-label small">Muted</label>{{
form.theme_dark_text_muted }}</div>
                  <div class="col-12 col-md-6"><label class="form-label small">Border</label>{{
form.theme_dark_border }}</div>
              </div>
            </div>
          </div>
        </div>

        <hr class="my-4">

        <h2 class="h6 mb-3">Social Links</h2>
        <div class="row g-3">
          <div class="col-12 col-md-6">
            <label class="form-label">Facebook</label>
            {{ form.facebook_url }}
          </div>
          <div class="col-12 col-md-6">
            <label class="form-label">Instagram</label>
            {{ form.instagram_url }}
          </div>
          <div class="col-12 col-md-6">
            <label class="form-label">TikTok</label>
            {{ form.tiktok_url }}
          </div>
          <div class="col-12 col-md-6">
            <label class="form-label">YouTube</label>
            {{ form.youtube_url }}
          </div>
          <div class="col-12 col-md-6">
            <label class="form-label">X (Twitter)</label>
            {{ form.x_url }}
          </div>
          <div class="col-12 col-md-6">
            <label class="form-label">LinkedIn</label>
            {{ form.linkedin_url }}
          </div>
        </div>

        <div class="mt-4 d-flex gap-2">
          <button class="btn btn-primary" type="submit">
            <i class="bi bi-save me-1"></i>Save Settings
          </button>
          <a class="btn btn-outline-secondary" href="{% url 'dashboards:admin' %}">Cancel</a>
        </div>
      </form>
    </div>
  </div>
</div>
{% endblock %}
```

templates\admin\analytics_index.html

```
{# templates/admin/analytics_index.html #}
{% extends "admin/index.html" %}
{% load static %}

{% block extrastyle %}
  {{ block.super }}
  <style>
    .analytics-dashboard {
      display: grid;
      grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));
      gap: 20px;
      margin-bottom: 30px;
    }

    .metric-card {
      background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
      border-radius: 8px;
      padding: 24px;
      color: white;
      box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
      transition: transform 0.2s, box-shadow 0.2s;
    }

    .metric-card:hover {
      transform: translateY(-2px);
      box-shadow: 0 8px 12px rgba(0, 0, 0, 0.15);
    }

    .metric-card.revenue {
      background: linear-gradient(135deg, #f093fb 0%, #f5576c 100%);
    }

    .metric-card.orders {
      background: linear-gradient(135deg, #4facfe 0%, #00f2fe 100%);
    }

    .metric-card.products {
      background: linear-gradient(135deg, #43e97b 0%, #38f9d7 100%);
    }

    .metric-card.sellers {
      background: linear-gradient(135deg, #fa709a 0%, #fee140 100%);
    }

    .metric-label {
      font-size: 14px;
      font-weight: 500;
      opacity: 0.9;
      margin-bottom: 8px;
      text-transform: uppercase;
      letter-spacing: 0.5px;
    }

    .metric-value {
      font-size: 32px;
      font-weight: 700;
      margin-bottom: 12px;
    }
    .metric-subtext {
      font-size: 12px;
      opacity: 0.8;
    }

    .analytics-tables {
      display: grid;
      grid-template-columns: 1fr 1fr;
      gap: 20px;
      margin-top: 30px;
    }

    @media (max-width: 1200px) {
      .analytics-tables {
        grid-template-columns: 1fr;
      }
    }

    .dashboard-module h2 {
      color: #333;
      margin-top: 0;
      padding-bottom: 12px;
      border-bottom: 3px solid #667eea;
    }

    .module {
      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
      border-radius: 4px;
    }
  </style>
{% endblock %}

{% block content %}
  <div id="content-main">
    <h1>Analytics Dashboard</h1>
    <p>Marketplace health and operating metrics.</p>

    <!-- Metrics Grid -->
    <div class="analytics-dashboard">
      <div class="metric-card products">
        <div class="metric-label">Total Orders</div>
        <div class="metric-value">{{ total_orders }}</div>
        <div class="metric-subtext">All-time paid orders</div>
      </div>

      <div class="metric-card revenue">
        <div class="metric-label">Total Revenue</div>
        <div class="metric-value">{{ total_revenue }}</div>
        <div class="metric-subtext">All-time gross revenue</div>
      </div>

      <div class="metric-card orders">
        <div class="metric-label">Last 30 Days Orders</div>
        <div class="metric-value">{{ orders_30d }}</div>
        <div class="metric-subtext">30-day trend</div>
      </div>

      <div class="metric-card sellers">
        <div class="metric-label">Last 30 Days Revenue</div>
        <div class="metric-value">{{ revenue_30d }}</div>
        <div class="metric-subtext">30-day trend</div>
      </div>
    </div>

    <!-- Engagement Metrics -->
    <div class="module">
      <h2>Engagement Activity</h2>
      <table style="width: 100%; border-collapse: collapse;">
        <tr style="background-color: #f8f9fa; border-bottom: 2px solid #667eea;">
          <td style="padding: 12px; font-weight: 600; color: #333;">Metric</td>
          <td style="padding: 12px; font-weight: 600; color: #333;">Count</td>
        </tr>
        <tr style="border-bottom: 1px solid #e0e0e0;">
          <td style="padding: 12px;">Product Views</td>
          <td style="padding: 12px;"><strong>{{ total_views }}</strong></td>
        </tr>
        <tr style="border-bottom: 1px solid #e0e0e0;">
          <td style="padding: 12px;">Product Clicks</td>
          <td style="padding: 12px;"><strong>{{ total_clicks }}</strong></td>
        </tr>
        <tr>
          <td style="padding: 12px;">Add-to-Cart Events</td>
          <td style="padding: 12px;"><strong>{{ total_add_to_cart }}</strong></td>
        </tr>
      </table>
    </div>

    <!-- Top Products by Revenue -->
    <div class="analytics-tables">
      <div class="module">
        <h2>Top Products by Revenue</h2>
        {% if top_products %}
          <table style="width: 100%; border-collapse: collapse;">
            <tr style="background-color: #f8f9fa; border-bottom: 2px solid #667eea;">
              <td style="padding: 12px; font-weight: 600; color: #333;">Product</td>
              <td style="padding: 12px; font-weight: 600; color: #333; text-align: right;">Revenue</td>
              <td style="padding: 12px; font-weight: 600; color: #333; text-align: right;">Qty</td>
            </tr>
            {% for product in top_products %}
              <tr style="border-bottom: 1px solid #e0e0e0;">
                <td style="padding: 12px;">{{ product.product__title|truncatewords:5 }}</td>
                <td style="padding: 12px; text-align: right;">${{ product.revenue_cents|add:0|divisibleby:100 }}</td>
                <td style="padding: 12px; text-align: right;">{{ product.quantity }}</td>
              </tr>
            {% endfor %}
          </table>
        {% else %}
          <p style="color: #999; padding: 20px; text-align: center;">No sales yet.</p>
        {% endif %}
      </div>

      <!-- Top Products by Engagement -->
      <div class="module">
        <h2>Top Products by Engagement</h2>
        {% if top_engagement %}
          <table style="width: 100%; border-collapse: collapse;">
            <tr style="background-color: #f8f9fa; border-bottom: 2px solid #667eea;">
              <td style="padding: 12px; font-weight: 600; color: #333;">Product</td>
              <td style="padding: 12px; font-weight: 600; color: #333; text-align: right;">Events</td>
            </tr>
            {% for product in top_engagement %}
              <tr style="border-bottom: 1px solid #e0e0e0;">
```

```
                <td style="padding: 12px;">{{ product.product__title|truncatewords:5 }}</td>
                <td style="padding: 12px; text-align: right;"><strong>{{ product.events }}</strong> (V: {{
product.views }} | C: {{ product.clicks }} | A: {{ product.add_to_cart }})</td>
              </tr>
            {% endfor %}
          </table>
        {% else %}
          <p style="color: #999; padding: 20px; text-align: center;">No engagement yet.</p>
        {% endif %}
      </div>
    </div>
  </div>
{% endblock %}
```

reviews\apps.py

```
# reviews/apps.py
from django.apps import AppConfig


class ReviewsConfig(AppConfig):
    default_auto_field = "django.db.models.BigAutoField"
    name = "reviews"
```

config\asgi.py

```
# config/asgi.py
"""
ASGI config for config project.

It exposes the ASGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/6.0/howto/deployment/asgi/
"""

import os

from django.core.asgi import get_asgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'config.settings.base')

application = get_asgi_application()
```

payments\management\commands\backfill_sale_entries.py

```
# payments/management/commands/backfill_sale_entries.py

from __future__ import annotations

from collections import defaultdict

from django.core.management.base import BaseCommand
from django.db import transaction

from orders.models import Order
from payments.models import SellerBalanceEntry


class Command(BaseCommand):
    help = "Backfill missing SellerBalanceEntry SALE credits for PAID orders."

    def add_arguments(self, parser):
        parser.add_argument("--dry-run", action="store_true", help="Do not write changes.")
        parser.add_argument("--limit", type=int, default=0, help="Limit number of orders processed.")
        parser.add_argument("--order-id", type=str, default="", help="Process a single order UUID.")

    @transaction.atomic
    def handle(self, *args, **opts):
        dry_run: bool = bool(opts["dry_run"])
        limit: int = int(opts["limit"] or 0)
        order_id: str = (opts["order_id"] or "").strip()

        qs = Order.objects.filter(status=Order.Status.PAID).prefetch_related("items")
        if order_id:
            qs = qs.filter(pk=order_id)

        if limit > 0:
            qs = qs.order_by("-paid_at")[:limit]
        else:
            qs = qs.order_by("-paid_at")

        created = 0
        skipped = 0

        for order in qs:
            totals_by_seller: dict[str, int] = defaultdict(int)
            for it in order.items.all():
                seller_id = getattr(it, "seller_id", None)
                if not seller_id:
                    continue
                net = int(getattr(it, "seller_net_cents", 0) or 0)
                if net <= 0:
                    continue
                totals_by_seller[str(seller_id)] += net

            for seller_id, net_cents in totals_by_seller.items():
                if net_cents <= 0:
                    continue

                exists = SellerBalanceEntry.objects.filter(
                    seller_id=seller_id,
                    order=order,
                    reason=SellerBalanceEntry.Reason.SALE,
                ).exists()
                if exists:
                    skipped += 1
                    continue

                self.stdout.write(
                    f"[CREATE] order={order.pk} seller={seller_id} sale=+{net_cents}c"
                )

                if not dry_run:
                    SellerBalanceEntry.objects.create(
                        seller_id=seller_id,
                        order=order,
                        reason=SellerBalanceEntry.Reason.SALE,
                        amount_cents=int(net_cents),
                        note=f"Backfill SALE credit for paid order {order.pk}",
                    )
                created += 1

        if dry_run:
            self.stdout.write(self.style.WARNING("DRY RUN: no changes written."))

        self.stdout.write(self.style.SUCCESS(f"Done. created={created} skipped_existing={skipped}"))
```

payments\management\commands\backfill_seller_balance.py

```
from django.core.management.base import BaseCommand
from orders.models import Order
from payments.models import SellerBalanceEntry
from django.db import transaction

class Command(BaseCommand):
    help = "Backfill SellerBalanceEntry for all paid orders that are missing them."

    def handle(self, *args, **options):
        count_created = 0
        with transaction.atomic():
            paid_orders = Order.objects.filter(status=Order.Status.PAID)
            for order in paid_orders:
                for item in order.items.all():
                    exists = SellerBalanceEntry.objects.filter(order_item=item, reason=SellerBalanceEntry.Reason.ADJUSTMENT).exists()
                    if not exists:
                        SellerBalanceEntry.objects.create(
                            seller=item.seller,
                            amount_cents=item.seller_net_cents,
                            reason=SellerBalanceEntry.Reason.ADJUSTMENT,
                            order=order,
                            order_item=item,
                            note=f"Backfill: Order paid {order.pk} (item {item.pk})"
                        )
                        count_created += 1
        self.stdout.write(self.style.SUCCESS(f"Backfilled {count_created} SellerBalanceEntry records."))
```

templates\emails\base.html

```
{# templates/emails/base.html #}
{% load static %}
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>{{ subject|default:"Home Craft 3D" }}</title>
  </head>
  <body style="margin:0;padding:0;background:#f8fafc;font-family:Arial,Helvetica,sans-serif;color:#111827;">
    <div style="max-width:640px;margin:0 auto;padding:24px;">
      <div style="background:#ffffff;border:1px solid #e5e7eb;border-radius:12px;overflow:hidden;">
        <div style="padding:20px 24px;border-bottom:1px solid #e5e7eb;display:flex;align-items:center;gap:10px;">
          <img src="{{ logo_url }}" alt="Home Craft 3D" style="height:28px;width:auto;display:block;" />
          <div style="font-weight:700;font-size:16px;">Home Craft 3D</div>
        </div>
        <div style="padding:24px;">
          {% block content %}{% endblock %}
        </div>
      </div>
      <div style="text-align:center;color:#6b7280;font-size:12px;margin-top:12px;">
        © {% now "Y" %} Home Craft 3D
      </div>
    </div>
  </body>
</html>
```

accounts\templates\accounts\base_accounts.html

```
{# templates/accounts/base_accounts.html #}
{% load static %}
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>{% block title %}Home Craft 3D{% endblock %}</title>
  <link rel="icon" href="{% static 'images/favicon.ico' %}">
  {% block meta_head %}
  <meta name="description" content="{% block meta_description %}Home Craft 3D is a marketplace for makers featuring 3D prints and digital files.{% endblock %}">
  <link rel="canonical" href="{{ request.build_absolute_uri }}">

  <meta property="og:type" content="website">
  <meta property="og:site_name" content="Home Craft 3D">
  {% block meta_title_og %}<meta property="og:title" content="Home Craft 3D">{% endblock %}
  <meta property="og:description" content="{{ block.super }}">
  <meta property="og:url" content="{{ request.build_absolute_uri }}">
  <meta property="og:image" content="{{ request.scheme }}://{{ request.get_host }}{% static 'images/homecraft3d_navbar.svg' %}">

  <meta name="twitter:card" content="summary_large_image">
  {% block meta_title %}<meta name="twitter:title" content="Home Craft 3D">{% endblock %}>
  <meta name="twitter:description" content="{{ block.super }}">
  <meta name="twitter:image" content="{{ request.scheme }}://{{ request.get_host }}{% static 'images/homecraft3d_navbar.svg' %}">
  {% endblock %}>

  {% if ga_measurement_id %}
    <script async src="https://www.googletagmanager.com/gtag/js?id={{ ga_measurement_id }}"></script>
    <script>
      window.dataLayer = window.dataLayer || [];
      function gtag(){dataLayer.push(arguments);}
      gtag('js', new Date());
      gtag('config', '{{ ga_measurement_id }}');
    </script>
  {% endif %}
  <link rel="icon" href="{% static 'images/favicon.ico' %}">
  <!-- Bootstrap CSS (CDN) -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">

  <!-- Bootstrap Icons (CDN) -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css">

  {# Theme variables from SiteConfig #}
  {% include "includes/theme_css_vars.html" %}

  {# Theme mapping to Bootstrap tokens #}
  <link rel="stylesheet" href="{% static 'css/theme.css' %}">

  <!-- Site overrides -->
  <link rel="stylesheet" href="{% static 'css/site.css' %}">

  {# Apply color mode as early as possible to avoid flash #}
  <script defer src="{% static 'js/theme.js' %}"></script>
  <style>
    body { font-family: system-ui, -apple-system, Segoe UI, Roboto, Arial, sans-serif; margin: 0; padding: 0; }
    .wrap { max-width: 900px; margin: 0 auto; padding: 24px; }
    .card { border: 1px solid #e5e7eb; border-radius: 10px; padding: 18px; }
    .form-row { display: grid; grid-template-columns: 1fr 1fr; gap: 12px; }
    .form-row-1 { display: grid; grid-template-columns: 1fr; gap: 12px; }
    label { font-size: 13px; color: #374151; display: block; margin-bottom: 6px; }
    input, select { width: 100%; padding: 10px; border: 1px solid #d1d5db; border-radius: 8px; }
    .actions { margin-top: 14px; display: flex; gap: 10px; }
    .btn { display: inline-block; padding: 10px 14px; border-radius: 8px; border: 1px solid #111827; background: #111827; color: white; text-decoration: none; cursor: pointer; }
    .btn.secondary { background: white; color: #111827; }
    .help { color: #6b7280; font-size: 13px; margin-top: 6px; }
    .msg { padding: 10px 12px; border-radius: 8px; margin-bottom: 12px; border: 1px solid #e5e7eb; }
    .msg.success { border-color: #bbf7d0; background: #f0fdf4; }
    .msg.error { border-color: #fecaca; background: #fef2f2; }
  </style>
</head>
<body class="bg-body">
  {% include "partials/navbar.html" %}

  <div class="wrap">

    {% if messages %}
      {% for message in messages %}
        <div class="msg {% if message.tags %}{{ message.tags }}{% endif %}">{{ message }}</div>
      {% endfor %}
    {% endif %}

    {% block content %}{% endblock %}
  </div>

  {% include "partials/footer.html" %}

  <!-- Bootstrap JS bundle -->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

  <!-- Sidebar expand/collapse (safe no-op on auth pages) -->
  <script defer src="{% static 'js/sidebar.js' %}"></script>
</body>
</html>
```

products\templates\products\seller\base_seller.html

```
{# products/templates/products/seller/base_seller.html #}
{% extends "base.html" %}

{% block title %}{% block seller_title %}Seller · Home Craft 3D{% endblock %}{% endblock %}
```

refunds\templates\refunds\buyer_detail.html

```
{# refunds/templates/refunds/buyer_detail.html #}
{% extends "base.html" %}
{% load money %}

{% block title %}Refund Request · Home Craft 3D{% endblock %}

{% block content %}
<div class="container py-4">
  <div class="d-flex justify-content-between align-items-center mb-3">
    <h1 class="h4 mb-0">Refund request</h1>

    {# If you arrived from an order link, best UX is to go back to the order. #}
    <a class="btn btn-outline-secondary btn-sm"
       href="{% url 'orders:detail' order_id=rr.order_id %}{% if order_token %}?t={{ order_token }}{% endif %}">
      Back to order
    </a>
  </div>

  <div class="card">
    <div class="card-body">
      <div class="fw-semibold">{{ rr.order_item.product.title }}</div>
      <div class="text-muted small">Order #{{ rr.order_id }}</div>

      <div class="mt-3">
        <div><strong>Status:</strong> {{ rr.get_status_display }}</div>
        <div><strong>Reason:</strong> {{ rr.get_reason_display }}</div>

        {% if rr.notes %}
          <div class="mt-2"><strong>Your notes:</strong><br>{{ rr.notes|linebreaksbr }}</div>
        {% endif %}
      </div>

      <hr>

      <div class="text-muted small">
        Refund snapshot:
        <div>Line: ${{ rr.line_subtotal_cents_snapshot|cents_to_dollars }}</div>
        <div>Tax: ${{ rr.tax_cents_allocated_snapshot|cents_to_dollars }}</div>
        <div>Shipping: ${{ rr.shipping_cents_allocated_snapshot|cents_to_dollars }}</div>
        <div class="fw-semibold">Total: ${{ rr.total_refund_cents_snapshot|cents_to_dollars }}</div>
      </div>

      {% if rr.seller_decision_note %}
        <div class="mt-3">
          <strong>Seller note:</strong><br>{{ rr.seller_decision_note|linebreaksbr }}
        </div>
      {% endif %}

      {% if rr.stripe_refund_id %}
        <div class="mt-3">
          <span class="badge text-bg-success">Refunded</span>
          <div class="text-muted small mt-1">Stripe refund id: {{ rr.stripe_refund_id }}</div>
        </div>
      {% endif %}
    </div>
  </div>
</div>
{% endblock %}
```

refunds\templates\refunds\buyer_list.html

```
{# refunds/templates/refunds/buyer_list.html #}
{% extends "base.html" %}
{% load money %}

{% block title %}My Refund Requests · Home Craft 3D{% endblock %}

{% block content %}
<div class="container py-4">
  <div class="d-flex justify-content-between align-items-center mb-3">
    <div>
      <h1 class="h4 mb-0">My Refund Requests</h1>
      <div class="text-muted small">Requests you’ve submitted for physical items.</div>
    </div>
    <a class="btn btn-outline-secondary btn-sm" href="{% url 'orders:my_orders' %}">Order history</a>
  </div>

  {% if refunds %}
    <div class="list-group">
      {% for rr in refunds %}
        <a class="list-group-item list-group-item-action" href="{% url 'refunds:buyer_detail' refund_id=rr.pk %}">
          <div class="d-flex justify-content-between gap-3">
            <div>
              <div class="fw-semibold">{{ rr.order_item.product.title }}</div>
              <div class="text-muted small">
                {{ rr.get_status_display }} · {{ rr.get_reason_display }} · Order #{{ rr.order_id }}
              </div>
            </div>
            <div class="text-end">
              <div class="fw-semibold">${{ rr.total_refund_cents_snapshot|cents_to_dollars }}</div>
              <div class="text-muted small">{{ rr.created_at|date:"M j, Y" }}</div>
            </div>
          </div>
        </a>
      {% endfor %}
    </div>
  {% else %}
    <div class="alert alert-info mb-0">You don’t have any refund requests yet.</div>
  {% endif %}
</div>
{% endblock %}
```

templates\emails\buyer_shipped.html

```
{# templates/emails/buyer_shipped.html #}
{% extends "emails/base.html" %}
{% block content %}
  <h1 style="margin:0 0 12px;font-size:20px;">Your order has shipped</h1>
  <p style="margin:0 0 12px;">Order <strong>#{{ order_id }}</strong></p>

  <div style="margin:12px 0;">
    <div style="font-size:12px;color:#6b7280;">Item</div>
    <div><strong>{{ item_title }}</strong> (Qty {{ quantity }})</div>
  </div>

  {% if carrier or tracking_number %}
    <div style="margin:16px 0;padding:16px;border:1px solid #e5e7eb;border-radius:10px;background:#f9fafb;">
      {% if carrier %}<div><strong>Carrier:</strong> {{ carrier }}</div>{% endif %}
      {% if tracking_number %}
        <div><strong>Tracking:</strong> {{ tracking_number }}</div>
        <div style="margin-top:6px;">
          <a href="https://trackingmore.com/track/?tracknumber={{ tracking_number|urlencode }}"
style="color:#2563eb;text-decoration:none;">Track package</a>
        </div>
      {% endif %}
    </div>
  {% endif %}

  <div style="margin:16px 0;padding:16px;border:1px solid #e5e7eb;border-radius:10px;background:#f9fafb;">
    <div style="font-size:12px;color:#6b7280;margin-bottom:6px;">Order link</div>
    <a href="{{ order_link }}" style="color:#2563eb;text-decoration:none;">View your order</a>
  </div>
{% endblock %}
```

cart\cart.py

```
# cart/cart.py
from __future__ import annotations

from dataclasses import dataclass
from decimal import Decimal, InvalidOperation
from typing import Any, Dict, List

from products.models import Product

CART_SESSION_KEY = "hc3_cart_v1"


def _to_decimal_money(value: Any) -> Decimal:
    """
    Convert user/session value to a safe Decimal dollars amount.
    Always returns >= 0.00.
    """
    if value is None:
        return Decimal("0.00")

    if isinstance(value, Decimal):
        d = value
    else:
        s = str(value).strip()
        if not s:
            return Decimal("0.00")
        # allow "$12.34" style input
        s = s.replace("$", "").replace(",", "").strip()
        try:
            d = Decimal(s)
        except (InvalidOperation, ValueError):
            return Decimal("0.00")

    if d.is_nan() or d.is_infinite():
        return Decimal("0.00")

    if d < 0:
        return Decimal("0.00")

    # currency-safe rounding to cents
    return d.quantize(Decimal("0.01"))


def product_unit_price(product: Product) -> Decimal:
    # Free items are always 0.00
    if getattr(product, "is_free", False):
        return Decimal("0.00")
    return _to_decimal_money(getattr(product, "price", "0.00"))


@dataclass(frozen=True)
class CartLine:
    product: Product
    quantity: int
    buyer_notes: str = ""
    tip_amount: Decimal = Decimal("0.00")

    @property
    def is_tip(self) -> bool:
        return bool(self.tip_amount and self.tip_amount > 0)

    @property
    def unit_price(self) -> Decimal:
        return product_unit_price(self.product)

    @property
    def line_total(self) -> Decimal:
        # Product total only (tips shown separately)
        return (self.unit_price * Decimal(int(self.quantity or 0))).quantize(Decimal("0.01"))

    @property
    def tip_total(self) -> Decimal:
        # Tip is always a single amount per product line (not multiplied by quantity)
        return _to_decimal_money(self.tip_amount)


class Cart:
    """
    Session-backed cart.

    Session format:
      {
        "<product_id>": {
            "qty": 2,
            "notes": "optional",
            "tip": "5.00"   # string dollars (optional)
        }
      }
    """

    def __init__(self, request):
        self.request = request
        self.session = request.session
        raw = self.session.get(CART_SESSION_KEY, {})
        self.data: Dict[str, Dict[str, Any]] = raw if isinstance(raw, dict) else {}

    def _save(self) -> None:
        self.session[CART_SESSION_KEY] = self.data
        self.session.modified = True

    def clear(self) -> None:
        self.data = {}
        self._save()

    def add(
        self,
        product: Product,
        quantity: int = 1,
        buyer_notes: str = "",
        is_tip: bool = False,
        tip_amount: Any = "0.00",
    ) -> None:
        if not getattr(product, "is_active", False):
            return

        pid = str(product.pk)

        # Digital files forced to qty=1
        try:
            if product.kind == Product.Kind.FILE:
                quantity = 1
        except Exception:
            pass

        quantity = max(int(quantity or 1), 1)
        notes = (buyer_notes or "").strip()

        tip_dec = _to_decimal_money(tip_amount) if is_tip else Decimal("0.00")

        if pid in self.data:
            payload = self.data[pid]
            # quantity accumulation for physical only
            try:
                if product.kind == Product.Kind.FILE:
                    payload["qty"] = 1
                else:
                    payload["qty"] = max(1, int(payload.get("qty", 1)) + quantity)
            except Exception:
                payload["qty"] = max(1, int(payload.get("qty", 1)) + quantity)

            # notes: only overwrite when provided
            if notes:
                payload["notes"] = notes

            # tip: overwrite if provided & > 0
            if tip_dec > 0:
                payload["tip"] = str(tip_dec)
            # do not auto-clear tip here; use set_tip()
            self.data[pid] = payload
        else:
            payload: Dict[str, Any] = {"qty": quantity}
            if notes:
                payload["notes"] = notes
            if tip_dec > 0:
                payload["tip"] = str(tip_dec)
            self.data[pid] = payload

        self._save()

    def set_quantity(self, product: Product, quantity: int) -> None:
        pid = str(product.pk)
        if pid not in self.data:
            return

        try:
            if product.kind == Product.Kind.FILE:
                self.data[pid]["qty"] = 1
            else:
                q = int(quantity)
                if q <= 0:
                    self.remove(product)
                    return
                self.data[pid]["qty"] = q
        except Exception:
            q = int(quantity or 1)
            if q <= 0:
                self.remove(product)
                return
            self.data[pid]["qty"] = q

        self._save()
```

```
def set_notes(self, product: Product, buyer_notes: str) -> None:
    pid = str(product.pk)
    if pid not in self.data:
        return

    notes = (buyer_notes or "").strip()
    if notes:
        self.data[pid]["notes"] = notes
    else:
        self.data[pid].pop("notes", None)

    self._save()

def set_tip(self, product: Product, tip_amount: Any) -> None:
    """
    Set or clear a tip for a product line.
    - empty/0 => clears
    - >0 => stores as string dollars
    """
    pid = str(product.pk)
    if pid not in self.data:
        return

    d = _to_decimal_money(tip_amount)
    if d <= 0:
        self.data[pid].pop("tip", None)
    else:
        self.data[pid]["tip"] = str(d)

    self._save()

def remove(self, product: Product) -> None:
    pid = str(product.pk)
    if pid in self.data:
        del self.data[pid]
        self._save()

def product_ids(self) -> List[int]:
    ids: List[int] = []
    for k in self.data.keys():
        try:
            ids.append(int(k))
        except ValueError:
            continue
    return ids

def lines(self) -> List[CartLine]:
    ids = self.product_ids()

    products = (
        Product.objects.filter(pk__in=ids, is_active=True)
        .select_related("category", "seller")
        .prefetch_related("images")
    )
    by_id = {p.pk: p for p in products}

    result: List[CartLine] = []
    dirty = False

    for pid_str, payload in list(self.data.items()):
        try:
            pid = int(pid_str)
        except ValueError:
            del self.data[pid_str]
            dirty = True
            continue

        product = by_id.get(pid)
        if not product:
            # deleted or inactive -> remove
            del self.data[pid_str]
            dirty = True
            continue

        qty = int(payload.get("qty", 1) or 1)

        # Digital files forced to qty=1
        try:
            if product.kind == Product.Kind.FILE:
                qty = 1
        except Exception:
            pass

        qty = max(1, qty)

        notes = str(payload.get("notes", "") or "")
        tip = _to_decimal_money(payload.get("tip", "0.00"))

        result.append(
            CartLine(
                product=product,
                quantity=qty,
                buyer_notes=notes,
                tip_amount=tip,
            )
        )

    if dirty:
        self._save()

    return result

def items_subtotal(self) -> Decimal:
    """
    Product totals only (no tips).
    """
    total = Decimal("0.00")
    for line in self.lines():
        total += line.line_total
    return total.quantize(Decimal("0.01"))

def tips_total(self) -> Decimal:
    total = Decimal("0.00")
    for line in self.lines():
        total += line.tip_total
    return total.quantize(Decimal("0.01"))

def grand_total(self) -> Decimal:
    return (self.items_subtotal() + self.tips_total()).quantize(Decimal("0.01"))

def subtotal(self) -> Decimal:
        """
        Backwards-compatible: keep name subtotal() but return product subtotal (no tips).
        Your template/view can show tips separately + grand total.
        """
        return self.items_subtotal()

    def count_items(self) -> int:
        # count distinct product lines
        return len(self.data)
```

cart\templates\cart\cart_detail.html

```
{# cart/templates/cart/cart_detail.html #}
{% extends "base.html" %}
{% block title %}Cart · Home Craft 3D{% endblock %}

{% block content %}
<div class="container-fluid" style="max-width: 1100px;">
  {% if messages %}
    <div class="mb-3">
      {% for message in messages %}
        <div class="alert alert-{{ message.tags }} mb-2">{{ message }}</div>
      {% endfor %}
    </div>
  {% endif %}

  <div class="d-flex flex-wrap align-items-center justify-content-between gap-2 mb-3">
    <div>
      <h1 class="h4 mb-1">Cart</h1>
      <div class="text-secondary small">Review your items before checkout.</div>
    </div>

    <div class="d-flex gap-2">
      <a class="btn btn-outline-dark" href="{% url 'products:list' %}">
        <i class="bi bi-arrow-left me-1"></i>Continue shopping
      </a>
      <form method="post" action="{% url 'cart:clear' %}">
        {% csrf_token %}
        <button class="btn btn-outline-danger" type="submit" {% if not cart_lines %}disabled{% endif %}>
          <i class="bi bi-trash me-1"></i>Clear cart
        </button>
      </form>
    </div>
  </div>

  {% if unready_sellers %}
    <div class="alert alert-warning d-flex flex-wrap align-items-center justify-content-between gap-2">
      <div>
        <div class="fw-semibold">Checkout blocked</div>
        <div class="small">
          One or more sellers in your cart haven’t completed Stripe payout setup yet:
          <strong>{{ unready_sellers|join:", " }}</strong>
        </div>
        <div class="small mt-1">
          You can still keep these items in your cart, but you must remove them to proceed to checkout.
        </div>
      </div>
      <a class="btn btn-dark" href="{% url 'products:list' %}">
        <i class="bi bi-shop me-1"></i>Browse other items
      </a>
    </div>
  {% endif %}

  <div class="card">
    <div class="table-responsive">
      <table class="table align-middle mb-0">
        <thead class="table-light">
          <tr>
            <th style="min-width: 320px;">Item</th>
            <th>Type</th>
            <th style="width: 180px;">Quantity</th>
      <th class="text-end">Unit</th>
      <th class="text-end">Total</th>
      <th class="text-end">Remove</th>
     </tr>
    </thead>
    <tbody>
     {% for line in cart_lines %}
      <tr>
       <td>
        <div class="d-flex align-items-center gap-3">
         {% if line.product.primary_image %}
          <img src="{{ line.product.primary_image.image.url }}" alt="{{ line.product.title }}"
            style="width:64px;height:64px;object-fit:cover;" class="rounded border">
         {% else %}
          <div class="rounded border bg-light d-flex align-items-center justify-content-center"
            style="width:64px;height:64px;">
           <span class="text-secondary small">No image</span>
          </div>
         {% endif %}

         <div class="w-100">
          <div class="fw-semibold">
               <a class="text-decoration-none" href="{{ line.product.get_absolute_url }}">{{ line.product.title
}}</a>
          </div>
          <div class="text-secondary small">
           by {{ line.product.seller.username }} · {{ line.product.category.name }}
          </div>

          {% if line.is_tip and line.tip_total and line.tip_total > 0 %}
           <div class="small mt-1">
            <span class="badge bg-primary me-1">Tip</span>
            <span class="text-secondary">Tip amount:</span>
            <strong>${{ line.tip_total|floatformat:2 }}</strong>
           </div>
          {% endif %}

          {% if line.buyer_notes %}
           <div class="alert alert-info mt-2 mb-0 p-2">
            <div class="small fw-semibold">
             <i class="bi bi-chat-left-text me-1"></i>Buyer notes
            </div>
            <div class="small" style="white-space: pre-wrap;">{{ line.buyer_notes }}</div>
           </div>
          {% endif %}

          <form method="post" action="{% url 'cart:update' %}" class="mt-2">
           {% csrf_token %}
           <input type="hidden" name="product_id" value="{{ line.product.pk }}">
           <input type="hidden" name="quantity" value="{{ line.quantity }}">
           <div class="input-group input-group-sm">
            <span class="input-group-text">Notes</span>
             <input class="form-control buyer-notes-input" name="buyer_notes" value="{{ line.buyer_notes }}"
placeholder="Add a note for the seller (optional)" maxlength="1000" data-maxlength="1000">
            <button class="btn btn-outline-dark" type="submit">Save</button>
           </div>
           <div class="form-text d-flex justify-content-between">
            <span>Shown to the seller after purchase.</span>
            <span class="buyer-notes-remaining" data-maxlength="1000">1000 characters remaining</span>
           </div>
          </form>
```

```
                  </div>
                </div>
              </td>

              <td>
                <span class="badge text-bg-secondary">{{ line.product.get_kind_display }}</span>
              </td>

              <td>
                {% if line.product.kind == KIND_FILE %}
                  <div class="text-secondary small">1 (digital)</div>
                {% else %}
                  <form method="post" action="{% url 'cart:update' %}" class="d-flex gap-2">
                    {% csrf_token %}
                    <input type="hidden" name="product_id" value="{{ line.product.pk }}">
                    <input class="form-control form-control-sm" type="number" name="quantity" min="1" value="{{ line.quantity }}">
                    <button class="btn btn-sm btn-outline-dark" type="submit">Update</button>
                  </form>
                  <div class="form-text">Use the remove button to delete an item.</div>
                  {% if line.product.max_purchases_per_buyer %}
                    <div class="small text-secondary mt-1">
                      Limit: {{ line.product.max_purchases_per_buyer }} per buyer.
                    </div>
                  {% endif %}
                {% endif %}
              </td>

              <td class="text-end"><strong>${{ line.unit_price|floatformat:2 }}</strong></td>
              <td class="text-end">
                <div><strong>${{ line.line_total|floatformat:2 }}</strong></div>
                {% if line.is_tip and line.tip_total and line.tip_total > 0 %}
                  <div class="small text-secondary">+ Tip: ${{ line.tip_total|floatformat:2 }}</div>
                {% endif %}
              </td>

              <td class="text-end">
                <form method="post" action="{% url 'cart:remove' line.product.pk %}">
                  {% csrf_token %}
                    <button class="btn btn-sm btn-outline-danger" type="submit" title="Remove" aria-label="Remove {{ line.product.title }} from cart">
                    <i class="bi bi-x-lg"></i>
                  </button>
                </form>
              </td>
            </tr>
          {% empty %}
            <tr>
              <td colspan="6" class="text-center text-secondary py-4">Your cart is empty.</td>
            </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>

    <div class="card-footer bg-white">
      <div class="d-flex justify-content-between align-items-center">
        <div class="text-secondary">Subtotal</div>
        <div class="h6 mb-0">${{ subtotal|floatformat:2 }}</div>
      </div>
      {% if tips_total and tips_total > 0 %}
        <div class="d-flex justify-content-between align-items-center mt-1">
          <div class="text-secondary">Tips</div>
          <div class="h6 mb-0">${{ tips_total|floatformat:2 }}</div>
        </div>
      {% endif %}

      <div class="d-flex justify-content-between align-items-center mt-2 pt-2 border-top">
        <div class="fw-semibold">Total</div>
        <div class="h5 mb-0">${{ grand_total|floatformat:2 }}</div>
      </div>
    </div>
  </div>

  <div class="mt-3 d-flex justify-content-end">
    <form id="place-order-form" method="post" action="{% url 'orders:place' %}" style="min-width: 420px;">
      {% csrf_token %}
      <input type="hidden" name="recaptcha_token" value="">

      {% if not user.is_authenticated %}
        <div class="mb-2">
          <label class="form-label small text-secondary mb-1">Email (required for guest checkout)</label>
          <input class="form-control" type="email" name="guest_email" placeholder="you@example.com" required>
          <div class="form-text">We’ll email you a secure link for order status and downloads.</div>
        </div>
      {% endif %}

      <div class="alert alert-light border small mb-2">
        By placing an order you agree to our
        <a href="{% url 'legal:terms' %}" class="text-decoration-none">Terms</a> and
        <a href="{% url 'legal:privacy' %}" class="text-decoration-none">Privacy Policy</a>.
      </div>

      <button class="btn btn-dark w-100" type="submit" {% if not can_checkout %}disabled{% endif %}>
        <i class="bi bi-receipt me-1"></i>Place order (pending payment)
      </button>

      {% if not cart_lines %}
        <div class="small mt-2 text-secondary">
          Add items to your cart to continue.
        </div>
      {% elif cart_lines and not can_checkout %}
        <div class="small mt-2" style="color:#b45309;">
          Remove items from sellers who haven’t finished Stripe setup to proceed.
        </div>
      {% endif %}
    </form>
  </div>

  {% if recaptcha_site_key %}
    <script src="https://www.google.com/recaptcha/api.js?render={{ recaptcha_site_key }}"></script>
    <script>
      (function () {
        var form = document.getElementById('place-order-form');
        if (!form) return;
        var tokenField = form.querySelector('input[name="recaptcha_token"]');

        form.addEventListener('submit', function (e) {
          if (!tokenField) return;
          if (tokenField.value) return;
```

```
          e.preventDefault();
          if (!window.grecaptcha || !grecaptcha.ready) {
            form.submit();
            return;
          }

          grecaptcha.ready(function () {
            grecaptcha.execute('{{ recaptcha_site_key }}', { action: 'checkout_place_order' }).then(function (token) {
              tokenField.value = token;
              form.submit();
            });
          });
        });
      })();
    </script>
  {% endif %}

  <script>
    (function () {
      var inputs = document.querySelectorAll('.buyer-notes-input');
      if (!inputs.length) return;

      function updateRemaining(input) {
        var max = parseInt(input.getAttribute('data-maxlength') || input.maxLength || '0', 10) || 0;
        var value = input.value || '';
        var remaining = Math.max(0, max - value.length);
        var remainingEl = input.closest('form')?.querySelector('.buyer-notes-remaining');
        if (remainingEl) {
          remainingEl.textContent = remaining + ' characters remaining';
        }
      }

      inputs.forEach(function (input) {
        updateRemaining(input);
        input.addEventListener('input', function () { updateRemaining(input); });
      });
    })();
  </script>
</div>
{% endblock %}
```

catalog\templates\catalog\category_detail.html

```
{# catalog/templates/catalog/category_detail.html #}
{% extends "base.html" %}
{% block title %}{{ category.name }} · Home Craft 3D{% endblock %}
{% block meta_title %}{{ category.name }} · Home Craft 3D{% endblock %}
{% block meta_description %}
  {% if category.description %}
    {{ category.description|striptags|truncatechars:160 }}
  {% else %}
    Explore {{ category.name }} on Home Craft 3D.
  {% endif %}
{% endblock %}

{% block content %}
<div class="container-fluid">
  <div class="d-flex flex-wrap align-items-center justify-content-between gap-2 mb-3">
    <div>
      <h1 class="h4 mb-1">{{ category.name }}</h1>
      <div class="text-secondary small">
        {{ category.get_type_display }}
        {% if category.parent %}
          · Parent: <a href="{{ category.parent.get_absolute_url }}" class="text-decoration-none">{{ category.parent.name }}</a>
        {% endif %}
      </div>
      {% if category.description %}
        <div class="text-secondary mt-2">{{ category.description }}</div>
      {% endif %}
    </div>

    <div class="d-flex gap-2">
      <a class="btn btn-outline-dark" href="{% url 'catalog:category_list' %}">
        <i class="bi bi-arrow-left me-1"></i>All Categories
      </a>
      <a class="btn btn-outline-dark" href="{% url 'products:list' %}">
        <i class="bi bi-grid me-1"></i>All Products
      </a>
    </div>
  </div>

  {% if children %}
    <div class="mb-3">
      <div class="text-secondary small mb-2">Sub-categories</div>
      <div class="d-flex flex-wrap gap-2">
        {% for c in children %}
          <a class="btn btn-sm btn-outline-secondary" href="{{ c.get_absolute_url }}">
            {{ c.name }}
          </a>
        {% endfor %}
      </div>
    </div>
  {% endif %}

  <div class="row g-3">
    {% for p in products %}
      <div class="col-12 col-sm-6 col-lg-4 col-xl-3">
        <div class="card h-100">
          {% if p.primary_image %}
            <img src="{{ p.primary_image.image.url }}" class="card-img-top" alt="{{ p.primary_image.alt_text|default:p.title }}">
          {% else %}
            <div class="bg-light d-flex align-items-center justify-content-center" style="height:180px;">
              <span class="text-secondary small">No image</span>
            </div>
          {% endif %}

          <div class="card-body">
            <div class="d-flex justify-content-between align-items-start gap-2">
              <h2 class="h6 mb-1">
                <a class="text-decoration-none" href="{{ p.get_absolute_url }}">{{ p.title }}</a>
              </h2>
              <span class="badge text-bg-secondary">{{ p.get_kind_display }}</span>
            </div>

            <div class="text-secondary small mb-2">
              by <strong>{{ p.seller.username }}</strong>
              · {{ p.category.name }}
            </div>

            {% if p.short_description %}
              <div class="text-secondary small">{{ p.short_description }}</div>
            {% endif %}
          </div>

          <div class="card-footer bg-white d-flex justify-content-between align-items-center">
            <strong>{{ p.display_price }}</strong>
            <a class="btn btn-sm btn-outline-dark" href="{{ p.get_absolute_url }}">View</a>
          </div>
        </div>
      </div>
    {% empty %}
      <div class="col-12">
        <div class="alert alert-secondary mb-0">No products in this category yet.</div>
      </div>
    {% endfor %}
  </div>
</div>
{% endblock %}
```

catalog\templates\catalog\category_list.html

```
{# catalog/templates/catalog/category_list.html #}
{% extends "base.html" %}
{% block title %}Categories · Home Craft 3D{% endblock %}
{% block meta_title %}Browse Categories · Home Craft 3D{% endblock %}
{% block meta_description %}Browse 3D model and 3D file categories on Home Craft 3D.{% endblock %}

{% block content %}
  <div style="max-width:1200px; margin:0 auto;">
    <h1 style="margin-top:0;">Browse Categories</h1>
    <p style="color:#6b7280; margin-top:6px;">
      Select a category from the sidebar, or browse the top-level groups below.
    </p>

    <div style="display:grid; grid-template-columns: 1fr 1fr; gap: 14px; margin-top: 14px;">
      <div style="background:#fff; border:1px solid #e5e7eb; border-radius:16px; padding:16px;">
        <h2 style="margin-top:0;">3D Models</h2>
        <ul>
          {% for c in model_roots %}
            <li>
              <a href="{{ c.get_absolute_url }}">{{ c.name }}</a>
              {% if c.children.all %}
                <ul>
                  {% for child in c.children.all %}
                    {% if child.is_active %}
                      <li><a href="{{ child.get_absolute_url }}">{{ child.name }}</a></li>
                    {% endif %}
                  {% endfor %}
                </ul>
              {% endif %}
            </li>
          {% empty %}
            <li style="color:#6b7280;">No model categories yet.</li>
          {% endfor %}
        </ul>
      </div>

      <div style="background:#fff; border:1px solid #e5e7eb; border-radius:16px; padding:16px;">
        <h2 style="margin-top:0;">3D Files</h2>
        <ul>
          {% for c in file_roots %}
            <li>
              <a href="{{ c.get_absolute_url }}">{{ c.name }}</a>
              {% if c.children.all %}
                <ul>
                  {% for child in c.children.all %}
                    {% if child.is_active %}
                      <li><a href="{{ child.get_absolute_url }}">{{ child.name }}</a></li>
                    {% endif %}
                  {% endfor %}
                </ul>
              {% endif %}
            </li>
          {% empty %}
            <li style="color:#6b7280;">No file categories yet.</li>
          {% endfor %}
        </ul>
      </div>
    </div>
  </div>
```

{% endblock %}

config\settings\celery.py

```
# config/settings/celery.py
```

orders\templates\orders\checkout_success.html

```
{# orders/templates/orders/checkout_success.html #}
{% extends "base.html" %}

{% block title %}Payment successful · Home Craft 3D{% endblock %}

{% block content %}
<div class="container py-5" style="max-width: 900px;">
  <div class="text-center">
    <h1 class="h3 mb-2">Payment successful</h1>
    <p class="text-muted mb-4">Thanks, your order is confirmed.</p>

    {% if order %}
      <div class="card shadow-sm mx-auto" style="max-width: 720px;">
        <div class="card-body">
          <div class="row g-2">
            <div class="col-md-6">
              <div class="text-muted small">Order</div>
              <div class="fw-semibold">{{ order.id }}</div>
            </div>
            <div class="col-md-6">
              <div class="text-muted small">Status</div>
              <div class="fw-semibold">{{ order.get_status_display }}</div>
            </div>
          </div>

          <hr>

          <a class="btn btn-primary" href="{% url 'orders:detail' order_id=order.id %}">
            View order
          </a>

          {% if user.is_authenticated %}
            <a class="btn btn-outline-secondary ms-2" href="{% url 'orders:purchases' %}">
              Go to purchases
            </a>
          {% endif %}
        </div>
      </div>
    {% else %}
      <div class="alert alert-info mx-auto" style="max-width: 720px;">
        Your payment completed. If your order doesn’t appear immediately, refresh in a moment.
      </div>
    {% endif %}

    <div class="mt-4">
      <a href="{% url 'products:list' %}" class="btn btn-outline-secondary">Continue shopping</a>
    </div>
  </div>
</div>

{% if ga_measurement_id and order %}
  <script>
    if (typeof gtag === 'function') {
      gtag('event', 'purchase', {
        transaction_id: '{{ order.id }}',
        value: {{ order.total_cents|default:0|floatformat:2 }} / 100,
        currency: '{{ order.currency|default:"usd"|upper }}'
      });
    }
  </script>
{% endif %}
{% endblock %}
```

templates\coming_soon.html

```
{# templates/coming_soon.html #}
{% extends "base.html" %}
{% block title %}Coming Soon · Home Craft 3D{% endblock %}
{% block content %}
<div class="container py-5 text-center">
  <h1 class="display-4 mb-3"> Coming Soon!</h1>
  <p class="lead mb-4">We're building something awesome for the Home Craft 3D community.</p>
  <div class="row justify-content-center mb-4">
    <div class="col-md-6">
      <div class="card shadow-sm">
        <div class="card-body">
          <h2 class="h4 mb-2">Blog</h2>
           <p>Get ready for expert tips, maker stories, product spotlights, and the latest 3D printing news. Our blog will inspire, inform, and connect the Home Craft 3D community!</p>
        </div>
      </div>
    </div>
    <div class="col-md-6 mt-4 mt-md-0">
      <div class="card shadow-sm">
        <div class="card-body">
          <h2 class="h4 mb-2">Community Chat Board</h2>
          <p>Share your projects, ask questions, and connect with fellow makers! Our chat board will be the go-to place for collaboration, support, and fun challenges.</p>
        </div>
      </div>
    </div>
  </div>
        <p class="mt-4">Stay tuned—launching soon! Want to be the first to know? <a href="#" class="fw-semibold">Sign up for updates</a>.</p>
</div>
{% endblock %}
```

core\config.py

```
# core/config.py
from __future__ import annotations

from decimal import Decimal
from typing import Any

from django.core.cache import cache

from .models import SiteConfig

CACHE_KEY = "core:site_config:v1"
CACHE_TTL_SECONDS = 30  # short TTL so admin changes take effect quickly


def get_site_config(*, use_cache: bool = True) -> SiteConfig:
    """
    Returns the singleton SiteConfig.

    Fresh DB case: auto-creates one row with model defaults.
    This avoids boot errors on a brand-new DB.
    """
    if use_cache:
        cached = cache.get(CACHE_KEY)
        if isinstance(cached, SiteConfig):
            return cached

    obj = SiteConfig.objects.first()
    if obj is None:
        obj = SiteConfig.objects.create()

    cache.set(CACHE_KEY, obj, CACHE_TTL_SECONDS)
    return obj


def invalidate_site_config_cache() -> None:
    cache.delete(CACHE_KEY)


def get_marketplace_sales_percent() -> Decimal:
    cfg = get_site_config()
    return Decimal(cfg.marketplace_sales_percent or Decimal("0"))


def get_marketplace_sales_rate() -> Decimal:
    # 10.00 -> 0.10
    pct = get_marketplace_sales_percent()
    try:
        return pct / Decimal("100")
    except Exception:
        return Decimal("0")


def get_platform_fee_cents() -> int:
    cfg = get_site_config()
    try:
        return int(cfg.platform_fee_cents or 0)
    except Exception:
        return 0
```

```
def get_allowed_shipping_countries() -> list[str]:
    cfg = get_site_config()
    return cfg.allowed_shipping_countries


def get_affiliate_sidebar_links() -> dict[str, Any]:
    """
    Convenience accessor for templates.
    Returns: {enabled, title, disclosure, links:[{label,url,note}]}
    """
    cfg = get_site_config()
    return {
        "enabled": bool(getattr(cfg, "affiliate_links_enabled", False)),
        "title": str(getattr(cfg, "affiliate_links_title", "") or "").strip() or "Recommended Filament & Gear",
        "disclosure": str(getattr(cfg, "affiliate_disclosure_text", "") or "").strip(),
        "links": list(getattr(cfg, "affiliate_links", []) or []),
    }
```

payments\templates\payments\connect_status.html

```
{# payments/templates/payments/connect_status.html #}
{% extends "base.html" %}
{% block title %}Stripe Payouts · Home Craft 3D{% endblock %}

{% block content %}
<div class="container-fluid" style="max-width: 900px;">
  <div class="d-flex flex-wrap align-items-center justify-content-between gap-2 mb-3">
    <div>
      <h1 class="h4 mb-1">Stripe Payouts</h1>
      <div class="text-secondary small">Connect your Stripe Express account to receive payouts.</div>
    </div>
  </div>

  {% if messages %}
    <div class="mb-3">
      {% for message in messages %}
        <div class="alert alert-info mb-2">{{ message }}</div>
      {% endfor %}
    </div>
  {% endif %}

  <div class="card mb-3">
    <div class="card-body">
      <div class="row g-3">
        <div class="col-md-6">
          <div class="text-secondary small mb-1">Account</div>
          {% if stripe.stripe_account_id %}
            <div class="fw-semibold">{{ stripe.stripe_account_id }}</div>
          {% else %}
            <div class="text-secondary">Not connected yet.</div>
          {% endif %}
        </div>

        <div class="col-md-6">
          <div class="text-secondary small mb-1">Status</div>

          <div class="d-flex flex-column gap-1">
            <div>
              {% if stripe.details_submitted %}
                <span class="badge text-bg-success">Details submitted</span>
              {% else %}
                <span class="badge text-bg-secondary">Details not submitted</span>
              {% endif %}
            </div>

            <div>
              {% if stripe.charges_enabled %}
                <span class="badge text-bg-success">Charges enabled</span>
              {% else %}
                <span class="badge text-bg-secondary">Charges disabled</span>
              {% endif %}
            </div>

            <div>
              {% if stripe.payouts_enabled %}
                <span class="badge text-bg-success">Payouts enabled</span>
              {% else %}
                <span class="badge text-bg-secondary">Payouts disabled</span>
              {% endif %}
            </div>

            {% if ready %}
              <div class="mt-2">
                <span class="badge text-bg-success">Ready to sell</span>
              </div>
            {% endif %}
          </div>
        </div>
      </div>

      <hr>

      <form method="post" action="{% url 'payments:connect_start' %}">
        {% csrf_token %}
        <button class="btn btn-dark" type="submit">
          <i class="bi bi-link-45deg me-1"></i>
          {% if stripe.stripe_account_id %}Continue Stripe setup{% else %}Connect Stripe{% endif %}
        </button>
      </form>

      <div class="text-secondary small mt-2">
            You’ll be redirected to Stripe to complete onboarding. Home Craft 3D never stores your sensitive banking details.
      </div>
    </div>
  </div>

  <div class="alert alert-warning">
    <div class="fw-semibold mb-1">Stripe setup status</div>
     You’re ready to sell when all three badges show: Details submitted, Charges enabled, and Payouts enabled (or when “Ready to sell” appears).
  </div>
</div>
{% endblock %}
```

dashboards\templates\dashboards\consumer_dashboard.html

```
{# dashboards/templates/dashboards/consumer_dashboard.html #}
{% extends "base.html" %}
{% load money %}
{% block title %}Consumer Dashboard · Home Craft 3D{% endblock %}

{% block content %}
<div class="container-fluid">
  <div class="d-flex flex-wrap align-items-center justify-content-between gap-2 mb-3">
    <div>
      <h1 class="h4 mb-1">Consumer Dashboard</h1>
      <div class="text-secondary small">Your purchases, downloads, and account activity.</div>
    </div>

    <div class="d-flex flex-wrap gap-2">
      <a class="btn btn-outline-dark" href="{% url 'orders:my_orders' %}">
        <i class="bi bi-bag-check me-1"></i>My Orders
      </a>
      <a class="btn btn-outline-dark" href="{% url 'accounts:profile' %}">
        <i class="bi bi-person-circle me-1"></i>Profile
      </a>
      <a class="btn btn-dark" href="{% url 'products:list' %}">
        <i class="bi bi-shop me-1"></i>Browse Store
      </a>
    </div>
  </div>

  <div class="row g-3 mb-3">
    <div class="col-12 col-md-4">
      <div class="card h-100">
        <div class="card-body">
          <div class="d-flex align-items-center justify-content-between">
            <div>
              <div class="text-secondary small">Total spent</div>
              <div class="h4 mb-0">${{ total_spent|default:"0" }}</div>
            </div>
            <i class="bi bi-currency-dollar fs-2 text-secondary"></i>
          </div>
        </div>
      </div>
    </div>

    <div class="col-12 col-md-4">
      <div class="card h-100">
        <div class="card-body">
          <div class="d-flex align-items-center justify-content-between">
            <div>
              <div class="text-secondary small">Paid orders</div>
              <div class="h4 mb-0">{{ paid_count|default:"0" }}</div>
            </div>
            <i class="bi bi-receipt fs-2 text-secondary"></i>
          </div>
        </div>
      </div>
    </div>

    <div class="col-12 col-md-4">
      <div class="card h-100">
        <div class="card-body">
          <div class="d-flex align-items-center justify-content-between">
```

```
            <div>
              <div class="text-secondary small">Quick links</div>
              <div class="d-flex flex-wrap gap-2 mt-2">
                <a class="btn btn-sm btn-outline-dark" href="{% url 'products:models' %}">Models</a>
                <a class="btn btn-sm btn-outline-dark" href="{% url 'products:files' %}">Files</a>
              </div>
            </div>
            <i class="bi bi-link-45deg fs-2 text-secondary"></i>
          </div>
        </div>
      </div>
    </div>
  </div>

  {% if not user.profile.is_seller %}
  <div class="card mb-3 border-primary bg-light">
    <div class="card-body">
      <div class="d-flex align-items-center justify-content-between">
        <div>
          <h5 class="card-title mb-2">Ready to become a seller?</h5>
          <p class="text-secondary mb-3">Upload your designs, reach customers worldwide, and start earning today.</p>
          <a href="{% url 'payments:connect_status' %}" class="btn btn-primary btn-sm">
            <i class="bi bi-shop me-1"></i>Set Up Selling
          </a>
        </div>
        <i class="bi bi-rocket fs-2 text-primary"></i>
      </div>
    </div>
  </div>
  {% endif %}

  <div class="card">
    <div class="card-header bg-white d-flex align-items-center justify-content-between">
      <div class="fw-semibold">Recent Orders</div>
      <a class="small text-decoration-none" href="{% url 'orders:my_orders' %}">View all</a>
    </div>

    <div class="table-responsive">
      <table class="table table-hover align-middle mb-0">
        <thead class="table-light">
          <tr>
            <th>Ordered</th>
            <th>Order</th>
            <th>Status</th>
            <th>Items</th>
            <th class="text-end">Total</th>
            <th class="text-end">Action</th>
          </tr>
        </thead>
        <tbody>
          {% for o in orders %}
            <tr>
              <td class="text-muted small">{{ o.created_at|date:"M j, Y" }}</td>
              <td class="fw-semibold">#{{ o.id }}</td>
              <td>
                {% if o.status == "PAID" %}
                  <span class="badge text-bg-success">{{ o.get_status_display }}</span>
                {% elif o.status == "PENDING" %}
                  <span class="badge text-bg-secondary">{{ o.get_status_display }}</span>
                {% else %}
```

```
                  <span class="badge text-bg-secondary">{{ o.get_status_display }}</span>
                {% endif %}
              </td>
              <td class="text-secondary">
                {{ o.items.count }} item{{ o.items.count|pluralize }}
              </td>
              <td class="text-end"><strong>${{ o.total_cents|cents_to_dollars }}</strong></td>
              <td class="text-end">
                <a class="btn btn-sm btn-outline-dark" href="{% url 'orders:detail' o.id %}">View</a>
              </td>
            </tr>
          {% empty %}
            <tr>
              <td colspan="6" class="text-center text-secondary py-4">No orders yet.</td>
            </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>
  </div>
</div>
{% endblock %}
```

legal\templates\legal\content.html

```
{# templates/legal/content.html #}
{% extends "legal/_legal_base.html" %}

{% block doc_content %}
<p>We require all listings and user content to be safe, lawful, and respectful.</p>

<h2>Prohibited Content</h2>
<p>Listings and user content must not include:</p>
<ul>
  <li>Illegal items or instructions</li>
  <li>Intellectual property infringement</li>
  <li>Weapons intended for harm</li>
  <li>Hate, harassment, or abusive content</li>
  <li>Explicit adult content</li>
  <li>Malware or harmful files</li>
</ul>

<h2>Seller Responsibilities</h2>
<ul>
  <li>Ensure you have rights to sell your designs</li>
  <li>Provide accurate, clear listings</li>
  <li>Respond to customer issues in a timely manner</li>
</ul>

<h2>Enforcement</h2>
<p>We may remove listings or suspend accounts that violate this policy.</p>

<p><strong>Contact:</strong>                                                                    <a
href="mailto:homecraft3dstore@gmail.com">homecraft3dstore@gmail.com</a></p>
{% endblock %}
```

payments\context_processors.py

```
# payments/context_processors.py
from __future__ import annotations

from django.urls import NoReverseMatch, reverse

from payments.models import SellerStripeAccount
from products.permissions import is_owner_user, is_seller_user


def seller_stripe_status(request):
    """Global template context.

    - seller_stripe_ready: True/False/None
        None => not a seller (hide badge)
    - has_connect_sync: bool (payments:connect_sync exists)
    - user_is_owner: bool (owner/admin override)
    - user_is_seller: bool (seller OR owner/admin)

    Notes:
    - These flags avoid templates touching `user.profile` directly, which can raise
      RelatedObjectDoesNotExist if a Profile row isn't created yet.
    """
    user = getattr(request, "user", None)

    user_is_owner = False
    user_is_seller = False
    seller_stripe_ready = None

    if user and getattr(user, "is_authenticated", False):
        user_is_owner = is_owner_user(user)
        user_is_seller = is_seller_user(user)

        if user_is_owner:
            seller_stripe_ready = True
        elif user_is_seller:
            acct = SellerStripeAccount.objects.filter(user=user).first()
            seller_stripe_ready = bool(acct and acct.is_ready)

    try:
        reverse("payments:connect_sync")
        has_connect_sync = True
    except NoReverseMatch:
        has_connect_sync = False

    return {
        "seller_stripe_ready": seller_stripe_ready,
        "has_connect_sync": has_connect_sync,
        "user_is_owner": user_is_owner,
        "user_is_seller": user_is_seller,
    }
```

payments\decorators.py

```
# payments/decorators.py

from __future__ import annotations

from functools import wraps

from django.contrib import messages
from django.shortcuts import redirect

from products.permissions import is_owner_user
from payments.models import SellerStripeAccount


def stripe_ready_required(view_func):
    """
    Gate seller uploads/publishing until Stripe Connect is fully enabled.

    Owner/admin bypasses the gate.
    """

    @wraps(view_func)
    def _wrapped(request, *args, **kwargs):
        if request.user.is_authenticated and is_owner_user(request.user):
            return view_func(request, *args, **kwargs)

        acct = SellerStripeAccount.objects.filter(user=request.user).first()
        if not acct or not acct.is_ready:
            messages.warning(
                request,
                "You must finish Stripe onboarding before you can create or modify listings.",
            )
            return redirect("payments:connect_status")

        return view_func(request, *args, **kwargs)

    return _wrapped
```

config\settings\dev.py

```
# config/settings/dev.py
"""
Development settings.
These settings are for local development only.
"""

from .base import *

DEBUG = True

ALLOWED_HOSTS = [
    "localhost",
    "127.0.0.1",
    "homecraft3d.onrender.com",
]

CSRF_TRUSTED_ORIGINS = [
    # keep empty for localhost; add ngrok/cloudflare tunnel here if used
]

EMAIL_BACKEND = "django.core.mail.backends.console.EmailBackend"

SECURE_SSL_REDIRECT = False
SESSION_COOKIE_SECURE = False
CSRF_COOKIE_SECURE = False

LOGGING = {
    "version": 1,
    "disable_existing_loggers": False,
    "handlers": {"console": {"class": "logging.StreamHandler"}},
    "root": {"handlers": ["console"], "level": "INFO"},
}
```

accounts\templates\accounts\disable_2fa.html

```
{# templates/accounts/disable_2fa.html #}
{% extends 'accounts/base_accounts.html' %}

{% block title %}Disable 2FA{% endblock %}

{% block content %}
<div class="container mt-5">
    <div class="row justify-content-center">
        <div class="col-md-6">
            <div class="card border-danger">
                <div class="card-header bg-danger text-white">
                    <h4 class="mb-0">Disable Two-Factor Authentication</h4>
                </div>
                <div class="card-body">
                    <div class="alert alert-warning" role="alert">
                        <strong>⚠ Warning:</strong> Disabling 2FA will reduce the security of your account.
                        Anyone who obtains your password will be able to access your account.
                    </div>

                    <p>To confirm, please enter your password:</p>

                    <form method="post">
                        {% csrf_token %}

                        {% if form.non_field_errors %}
                            <div class="alert alert-danger">
                                {{ form.non_field_errors }}
                            </div>
                        {% endif %}

                        <div class="mb-3">
                            {{ form.password }}
                            {% if form.password.errors %}
                                <div class="invalid-feedback d-block">
                                    {{ form.password.errors }}
                                </div>
                            {% endif %}
                        </div>

                        <div class="d-grid gap-2">
                            <button type="submit" class="btn btn-danger"
                                      onclick="return confirm('Are you sure you want to disable 2FA? This will reduce your account
security.');">
                                Disable 2FA
                            </button>
                            <a href="{% url 'accounts:view_2fa_status' %}" class="btn btn-secondary">
                                Cancel
                            </a>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>
</div>
{% endblock %}
```

templates\emails\download_reminder.html

```
{# templates/emails/download_reminder.html #}
{% extends "emails/base.html" %}
{% block content %}
  <h1 style="margin:0 0 12px;font-size:20px;">Your downloads are ready</h1>
  <p style="margin:0 0 12px;">Your order <strong>#{{ order_id }}</strong> includes downloadable files.</p>

  <div style="margin:16px 0;padding:16px;border:1px solid #e5e7eb;border-radius:10px;background:#f9fafb;">
    <div style="font-size:12px;color:#6b7280;margin-bottom:6px;">Order link</div>
    <a href="{{ order_link }}" style="color:#2563eb;text-decoration:none;">View your order</a>
  </div>

  {% if downloads %}
    <h2 style="margin:16px 0 8px;font-size:16px;">Downloads</h2>
    <ul style="padding-left:18px;margin:0;">
      {% for d in downloads %}
        <li style="margin:6px 0;">
          <strong>{{ d.filename }}</strong><br>
          <a href="{{ d.url }}" style="color:#2563eb;text-decoration:none;">Download</a>
        </li>
      {% endfor %}
    </ul>
  {% endif %}

  {% if is_guest %}
    <p style="margin:16px 0 0;color:#6b7280;font-size:13px;">Keep this link safe so you can access your downloads
later.</p>
  {% endif %}
{% endblock %}
```

accounts\management\commands\enable_admin_2fa.py

```
"""Management command to enable 2FA for admin/superuser accounts."""
import io
import qrcode
from base64 import b64encode
from django.core.management.base import BaseCommand
from django.contrib.auth.models import User
from django_otp.plugins.otp_totp.models import TOTPDevice
from django_otp.plugins.otp_static.models import StaticDevice, StaticToken
from django_otp.util import random_hex


class Command(BaseCommand):
    help = 'Setup 2FA for superuser/admin accounts'

    def add_arguments(self, parser):
        parser.add_argument(
            'username',
            nargs='?',
            type=str,
            help='Username to enable 2FA for (if not provided, enables for all superusers)'
        )
        parser.add_argument(
            '--force',
            action='store_true',
            help='Force regeneration of 2FA even if already configured'
        )

    def handle(self, *args, **options):
        username = options.get('username')
        force = options.get('force', False)

        # Get users to process
        if username:
            try:
                users = [User.objects.get(username=username)]
            except User.DoesNotExist:
                self.stdout.write(self.style.ERROR(f'User "{username}" not found'))
                return
        else:
            users = User.objects.filter(is_superuser=True)

        if not users:
            self.stdout.write(self.style.WARNING('No superusers found'))
            return

        for user in users:
            self.stdout.write(f'\nProcessing user: {user.username} ({user.email})')

            # Check if already has 2FA
            device = TOTPDevice.objects.filter(user=user, name='default').first()
            if device and device.confirmed and not force:
                self.stdout.write(self.style.WARNING('  → 2FA already enabled (use --force to regenerate)'))
                continue

            # Generate or regenerate device
            if device and force:
                device.delete()
                self.stdout.write('  → Clearing existing 2FA')
            device, created = TOTPDevice.objects.get_or_create(
                user=user,
                name='default',
                defaults={'confirmed': False}
            )

            # Generate secret key
            if not device.key or force:
                device.key = random_hex(20)
                device.save()
                self.stdout.write('  → Generated new secret key')

            # Generate backup codes
            recovery_device, _ = StaticDevice.objects.get_or_create(
                user=user,
                name='recovery'
            )
            StaticToken.objects.filter(device=recovery_device).delete()
            backup_codes = []
            for i in range(10):
                token = StaticToken.random_token()
                StaticToken.objects.create(device=recovery_device, token=token)
                backup_codes.append(token)

            self.stdout.write(self.style.SUCCESS('✓ 2FA Setup Complete'))
            self.stdout.write(f'\nAccount: {user.email}')
            self.stdout.write(f'Secret Key: {device.key}')
            self.stdout.write(f'\nSetup URL: {device.config_url}')

            # Generate and display QR code
            qr = qrcode.QRCode(version=1, box_size=10, border=5)
            qr.add_data(device.config_url)
            qr.make(fit=True)
            img = qr.make_image(fill_color="black", back_color="white")
            buffer = io.BytesIO()
            img.save(buffer, format='PNG')
            qr_code_base64 = b64encode(buffer.getvalue()).decode()

            self.stdout.write('\n--- BACKUP RECOVERY CODES ---')
            self.stdout.write('Save these in a secure location:')
            for i, code in enumerate(backup_codes, 1):
                self.stdout.write(f'{i:2d}. {code}')

            self.stdout.write('\n--- QR CODE (Base64) ---')
            self.stdout.write(f'data:image/png;base64,{qr_code_base64}')

            self.stdout.write(f'\n--- NEXT STEPS ---')
            self.stdout.write('1. Use the above secret key or QR code with your authenticator app')
            self.stdout.write('2. Save the recovery codes in a secure location')
            self.stdout.write('3. Visit /accounts/2fa/status/ to verify setup')
            self.stdout.write('4. First login will require 2FA token verification')

        self.stdout.write(self.style.SUCCESS('\n✓ 2FA setup complete for all selected users'))
```

templates\partials\footer.html

```
{# templates/partials/footer.html #}
{% load static %}
<footer class="border-top bg-body-tertiary mt-auto">
  <div class="container py-4">
    <div class="row g-3">
      <div class="col-md-3">
        <a class="text-decoration-none d-inline-flex align-items-center gap-2 mb-1" href="{% url 'core:home' %}">
          <img src="{% static 'images/homecraft3d_icon.svg' %}" alt="Home Craft 3D" style="height:26px;width:auto;">
          <span class="fw-semibold small text-body">Home Craft 3D</span>
        </a>
        <div class="text-secondary" style="font-size:0.8rem;">Marketplace for makers - 3D prints and digital files.</div>
      </div>

      <div class="col-md-2">
        <div class="text-uppercase text-secondary fw-bold mb-1" style="font-size:0.7rem;letter-spacing:0.06em;">Company</div>
        <div class="d-flex flex-column gap-1" style="font-size:0.8rem;">
          <a class="text-secondary text-decoration-none" href="{% url 'core:home' %}">Home</a>
          <a class="text-secondary text-decoration-none" href="{% url 'products:list' %}">Shop</a>
          <a class="text-secondary text-decoration-none" href="mailto:homecraft3dstore@gmail.com">Contact</a>
        </div>
      </div>

      <div class="col-md-2">
        <div class="text-uppercase text-secondary fw-bold mb-1" style="font-size:0.7rem;letter-spacing:0.06em;">Support</div>
        <div class="d-flex flex-column gap-1" style="font-size:0.8rem;">
          <a class="text-secondary text-decoration-none" href="{% url 'orders:my_orders' %}">My Orders</a>
          <a class="text-secondary text-decoration-none" href="{% url 'legal:refund' %}">Refund Policy</a>
          <a class="text-secondary text-decoration-none" href="{% url 'legal:content' %}">Content Policy</a>
        </div>
      </div>

      <div class="col-md-2">
        <div class="text-uppercase text-secondary fw-bold mb-1" style="font-size:0.7rem;letter-spacing:0.06em;">Legal</div>
        <div class="d-flex flex-column gap-1" style="font-size:0.8rem;">
          <a class="text-secondary text-decoration-none" href="{% url 'legal:terms' %}">Terms & Conditions</a>
          <a class="text-secondary text-decoration-none" href="{% url 'legal:privacy' %}">Privacy Policy</a>
          <a class="text-secondary text-decoration-none" href="/docs/seller_fees/">Seller Fees</a>
        </div>
      </div>

      <div class="col-md-3">
        {% if site_config.facebook_url or site_config.instagram_url or site_config.tiktok_url or site_config.youtube_url or site_config.x_url or site_config.linkedin_url %}
          <div class="text-uppercase text-secondary fw-bold mb-1" style="font-size:0.7rem;letter-spacing:0.06em;">Follow Us</div>
          <div class="d-flex align-items-center gap-2">
            {% if site_config.facebook_url %}
              <a class="text-secondary" href="{{ site_config.facebook_url }}" target="_blank" rel="noopener" aria-label="Facebook" style="font-size:1rem;">
                <i class="bi bi-facebook"></i>
              </a>
            {% endif %}
            {% if site_config.instagram_url %}
              <a class="text-secondary" href="{{ site_config.instagram_url }}" target="_blank" rel="noopener"
```

```
aria-label="Instagram" style="font-size:1rem;">
                <i class="bi bi-instagram"></i>
              </a>
            {% endif %}
            {% if site_config.tiktok_url %}
              <a class="text-secondary" href="{{ site_config.tiktok_url }}" target="_blank" rel="noopener"
aria-label="TikTok" style="font-size:1rem;">
                <i class="bi bi-tiktok"></i>
              </a>
            {% endif %}
            {% if site_config.youtube_url %}
              <a class="text-secondary" href="{{ site_config.youtube_url }}" target="_blank" rel="noopener"
aria-label="YouTube" style="font-size:1rem;">
                <i class="bi bi-youtube"></i>
              </a>
            {% endif %}
            {% if site_config.x_url %}
              <a class="text-secondary" href="{{ site_config.x_url }}" target="_blank" rel="noopener" aria-label="X"
style="font-size:1rem;">
                <i class="bi bi-twitter-x"></i>
              </a>
            {% endif %}
            {% if site_config.linkedin_url %}
              <a class="text-secondary" href="{{ site_config.linkedin_url }}" target="_blank" rel="noopener"
aria-label="LinkedIn" style="font-size:1rem;">
                <i class="bi bi-linkedin"></i>
              </a>
            {% endif %}
          </div>
        {% endif %}
      </div>
    </div>

    <div class="border-top mt-3 pt-3 d-flex flex-wrap justify-content-between align-items-center text-secondary
gap-2" style="font-size:0.75rem;">
      <div>© {% now "Y" %} Home Craft 3D. All rights reserved.</div>
      <div class="d-flex gap-3">
        <a class="text-secondary text-decoration-none" href="{% url 'legal:terms' %}">Terms</a>
        <a class="text-secondary text-decoration-none" href="{% url 'legal:privacy' %}">Privacy</a>
        <a class="text-secondary text-decoration-none" href="{% url 'legal:refund' %}">Refund</a>
        <a class="text-secondary text-decoration-none" href="{% url 'legal:content' %}">Content</a>
      </div>
    </div>
  </div>
</footer>
```

reviews\forms.py

```
# reviews/forms.py
from __future__ import annotations

from django import forms

from .models import Review, SellerReview


class ReviewForm(forms.ModelForm):
    class Meta:
        model = Review
        fields = ["rating", "title", "body"]
        widgets = {
            "rating": forms.NumberInput(attrs={"min": 1, "max": 5, "class": "form-control"}),
            "title": forms.TextInput(attrs={"class": "form-control"}),
            "body": forms.Textarea(attrs={"class": "form-control", "rows": 5}),
        }


class SellerReviewForm(forms.ModelForm):
    class Meta:
        model = SellerReview
        fields = ["rating", "title", "body"]
        widgets = {
            "rating": forms.NumberInput(attrs={"min": 1, "max": 5, "class": "form-control"}),
            "title": forms.TextInput(attrs={"class": "form-control"}),
            "body": forms.Textarea(attrs={"class": "form-control", "rows": 5}),
        }
```

accounts\forms_2fa.py

```
# accounts/forms_2fa.py
from django import forms


class TOTPVerificationForm(forms.Form):
    """Form to verify TOTP token."""
    token = forms.CharField(
        label='6-Digit Code',
        max_length=6,
        min_length=6,
        widget=forms.TextInput(attrs={
            'class': 'form-control text-center',
            'placeholder': '000000',
            'inputmode': 'numeric',
            'autocomplete': 'off',
        }),
        help_text='Enter the 6-digit code from your authenticator app'
    )

    def clean_token(self):
        """Clean and validate token."""
        token = self.cleaned_data.get('token', '').replace(' ', '')
        if not token.isdigit():
            raise forms.ValidationError('Code must contain only digits.')
        return token


class RecoveryCodeForm(forms.Form):
    """Form to use a recovery code."""
    code = forms.CharField(
        label='Recovery Code',
        max_length=50,
        widget=forms.TextInput(attrs={
            'class': 'form-control',
            'placeholder': 'Enter your recovery code',
            'autocomplete': 'off',
        }),
        help_text='Enter one of your recovery codes if you lose access to your authenticator app'
    )


class Disable2FAForm(forms.Form):
    """Form to disable 2FA (requires password confirmation)."""
    password = forms.CharField(
        label='Confirm Password',
        widget=forms.PasswordInput(attrs={
            'class': 'form-control',
            'placeholder': 'Enter your password to disable 2FA',
        }),
        help_text='Enter your password to confirm 2FA disabling'
    )

    def __init__(self, user=None, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.user = user

    def clean_password(self):
        """Verify password is correct."""
        password = self.cleaned_data.get('password')
```

```
if self.user and not self.user.check_password(password):
    raise forms.ValidationError('Password is incorrect.')
return password
```

orders\forms_seller.py

```
# orders/forms_seller.py
from __future__ import annotations

from django import forms


class MarkShippedForm(forms.Form):
    tracking_number = forms.CharField(required=False, max_length=80)
    carrier = forms.CharField(required=False, max_length=40)
```

core\templates\core\home.html

```
{# core/templates/core/home.html #}
{% extends "base.html" %}
{% load static %}

{% block title %}Home Craft 3D · 3D Prints & Digital Files Marketplace{% endblock %}
{% block meta_title %}Home Craft 3D · 3D Prints & Digital Files{% endblock %}
{% block og_title %}{{ product.title }} · Home Craft 3D{% endblock %}
{% block og_description %}{{ product.short_description }}{% endblock %}
{% block meta_description %}Discover high-quality 3D printed models and digital files. Buy from makers worldwide or sell your designs. Trending products, new releases, and more.{% endblock %}

{% block content %}

  {% include "includes/home_banner.html" %}
  <br>

  {% if ad_banner %}
  <div class="container mb-4">
    <div class="row justify-content-center">
      <div class="col-12 col-md-10">
        <div class="alert alert-warning p-0 border-0 rounded-3 overflow-hidden position-relative">
          {% if ad_banner.link_url %}
            <a href="{{ ad_banner.link_url }}" target="_blank" rel="noopener" class="d-block text-decoration-none">
              <img src="{{ ad_banner.image.url }}" alt="{{ ad_banner.title }}" class="img-fluid w-100" style="max-height:180px;object-fit:cover;">
            </a>
          {% else %}
            <img src="{{ ad_banner.image.url }}" alt="{{ ad_banner.title }}" class="img-fluid w-100" style="max-height:180px;object-fit:cover;">
          {% endif %}
          {% if ad_banner.title %}
            <div class="position-absolute bottom-0 start-0 w-100 bg-dark bg-opacity-50 text-white p-2 small text-center">{{ ad_banner.title }}</div>
          {% endif %}
        </div>
      </div>
    </div>
  </div>
  {% endif %}
<!-- Hero Section -->
<div class="bg-light border-bottom mb-5">
  <div class="container py-5">
    <div class="row align-items-center g-4">
      <div class="col-lg-6">
        <h1 class="display-5 fw-bold mb-3">
          <strong> {{ site_config.home_hero_title|default:"Welcome to Home Craft 3D" }} </strong>
        </h1>
        <p class="lead text-secondary mb-4">
          <strong> {{ site_config.home_hero_subtitle|default:"Discover high-quality 3D printable models and digital files. Buy from makers worldwide or sell your designs."|linebreaksbr|safe }} </strong>
        </p>

        <div class="d-flex flex-wrap gap-2">
          <a class="btn btn-dark btn-lg" href="{% url 'products:list' %}">
            <i class="bi bi-search me-2"></i>Browse All Products
          </a>
          <a class="btn btn-outline-dark btn-lg" href="{% url 'catalog:category_list' %}">
            <i class="bi bi-grid me-2"></i>Explore Categories
          </a>
        </div>
      </div>

      <div class="col-lg-6 text-center">
        <div class="bg-secondary-subtle p-4 rounded-3 h-100 d-flex align-items-center justify-content-center">
          <img
            src="{% static 'images/homecraft3d_facebook.png' %}"
            alt="3D printer in action printing a dragon"
            class="img-fluid rounded-3 hero-image">
        </div>
      </div>

    </div>
  </div>
</div>

<!-- Category Quick Links -->
<div class="container mb-5">
  <div class="row g-3">
    <div class="col-md-4">
      <div class="card h-100 text-center p-4 border-0 bg-light">
        <h5 class="card-title mb-2">3D Models</h5>
        <p class="text-secondary small mb-3">Ready-to-print designs</p>
        <a href="{% url 'products:models' %}" class="btn btn-sm btn-outline-dark">Browse Models</a>
      </div>
    </div>
    <div class="col-md-4">
      <div class="card h-100 text-center p-4 border-0 bg-light">
        <h5 class="card-title mb-2">Digital Files</h5>
        <p class="text-secondary small mb-3">Instant downloads</p>
        <a href="{% url 'products:files' %}" class="btn btn-sm btn-outline-dark">Browse Files</a>
      </div>
    </div>
    <div class="col-md-4">
      <div class="card h-100 text-center p-4 border-0 bg-light">
        <h5 class="card-title mb-2">Top Sellers</h5>
        <p class="text-secondary small mb-3">Trusted creators</p>
        <a href="{% url 'products:top_sellers' %}" class="btn btn-sm btn-outline-dark">View Sellers</a>
      </div>
    </div>
  </div>
</div>

<div class="container">
  {% if trending %}
    <div class="mb-5">
      <div class="d-flex justify-content-between align-items-center mb-4">
        <div>
          <h2 class="h3 mb-1"> Trending Now</h2>
          <p class="text-secondary">Hot products everyone's talking about</p>
        </div>
        <a href="{% url 'products:list' %}?sort=trending" class="btn btn-sm btn-outline-secondary">View all</a>
      </div>
      <div class="row g-3">
        {% for p in trending %}
          <div class="col-12 col-sm-6 col-lg-3">
                  {%  include  "products/_product_card.html"  with  p=p  show_add_to_cart=True  show_category=False
image_height="200px" %}
          </div>
        {% endfor %}
      </div>
    </div>
  {% endif %}

  {% if new_items %}
    <div class="mb-5">
      <div class="d-flex justify-content-between align-items-center mb-4">
        <div>
          <h2 class="h3 mb-1"> Just Added</h2>
          <p class="text-secondary">Latest designs uploaded</p>
        </div>
        <a href="{% url 'products:list' %}?sort=new" class="btn btn-sm btn-outline-secondary">View all</a>
      </div>
      <div class="row g-3">
        {% for p in new_items %}
          <div class="col-12 col-sm-6 col-lg-3">
            {% include "products/_product_card.html" with p=p show_add_to_cart=True show_category=False image_height="200px" %}
          </div>
        {% endfor %}
      </div>
    </div>
  {% endif %}

  {% if recently_purchased_list %}
    <div class="mb-5">
      <div class="d-flex justify-content-between align-items-center mb-4">
        <div>
          <h2 class="h3 mb-1"> Customer Favorites</h2>
          <p class="text-secondary">Popular purchases in the last 30 days</p>
        </div>
        <a href="{% url 'products:list' %}" class="btn btn-sm btn-outline-secondary">View all</a>
      </div>
      <div class="row g-3">
        {% for p in recently_purchased_list %}
          <div class="col-12 col-sm-6 col-lg-3">
            {% include "products/_product_card.html" with p=p show_add_to_cart=True show_category=False image_height="200px" %}
          </div>
        {% endfor %}
      </div>
    </div>
  {% endif %}

  {% if most_downloaded_list %}
    <div class="mb-5">
      <div class="d-flex justify-content-between align-items-center mb-4">
        <div>
          <h2 class="h3 mb-1"> Most Downloaded</h2>
          <p class="text-secondary">Top digital files everyone's getting</p>
        </div>
        <a href="{% url 'products:files' %}?sort=trending" class="btn btn-sm btn-outline-secondary">View all</a>
      </div>
      <div class="row g-3">
        {% for p in most_downloaded_list %}
          <div class="col-12 col-sm-6 col-lg-3">
            {% include "products/_product_card.html" with p=p show_add_to_cart=True show_category=False image_height="200px" %}
          </div>
        {% endfor %}
      </div>
    </div>
  {% endif %}

  {% if featured %}
    <div class="mb-5">
      <div class="d-flex justify-content-between align-items-center mb-4">
        <div>
          <h2 class="h3 mb-1"> Featured Picks</h2>
          <p class="text-secondary">Curated selections from our store</p>
        </div>
        <a href="{% url 'products:list' %}" class="btn btn-sm btn-outline-secondary">View all</a>
      </div>
      <div class="row g-3">
        {% for p in featured %}
          <div class="col-12 col-sm-6 col-lg-3">
            {% include "products/_product_card.html" with p=p show_add_to_cart=True show_category=False image_height="200px" %}
          </div>
        {% endfor %}
      </div>
    </div>
  {% endif %}

  <br>

  {% if product.filament_recommendations.all %}
  <section class="mt-4">
    <h4>Recommended Filament</h4>

    <ul class="list-unstyled">
      {% for rec in product.filament_recommendations.all %}
        {% if rec.is_active %}
          <li class="mb-2">
            <strong>{{ rec.get_material_display }}</strong>
            {% if rec.brand %} — {{ rec.brand }}{% endif %}
            {% if rec.notes %}<div class="text-muted small">{{ rec.notes }}</div>{% endif %}
            <div>
              <a href="{{ rec.url }}" target="_blank" rel="noopener noreferrer">
                View filament ↗
              </a>
            </div>
          </li>
        {% endif %}
      {% endfor %}
    </ul>
  </section>
  {% endif %}

  <br>

  <div class="bg-dark text-white rounded-3 p-5 mb-5 text-center">
    <h2 class="mb-3">Ready to start creating or selling?</h2>
    <p class="lead text-light mb-4">Join our community of creators. Upload your designs, connect with customers, and grow your business.</p>
    {% if not user.is_authenticated %}
      <a href="{% url 'accounts:register' %}" class="btn btn-light btn-lg">Get Started Today</a>
    {% else %}
      <a href="{% url 'payments:connect_status' %}" class="btn btn-light btn-lg">Become a Seller</a>
    {% endif %}
  </div>
</div>
{% endblock %}
```

templates\includes\home_banner.html

```
{# templates/includes/home_banner.html #}
{% if site_config and site_config.home_banner_enabled and site_config.home_banner_text %}
  <div class="hc3-home-banner text-center" role="alert">
    <div class="container-fluid">
      <div class="fw-bold">
        {{ site_config.home_banner_text }}
      </div>
    </div>
  </div>
{% endif %}
```

accounts\templates\accounts\login.html

```
{# templates/accounts/login.html #}
{% extends "accounts/base_accounts.html" %}
{% load static %}

{% block title %}Login · Home Craft 3D{% endblock %}

{% block content %}
<div style="max-width: 540px; margin: 40px auto;">
  <div style="text-align: center; margin-bottom: 32px;">
    <div style="margin:0 auto 16px;">
      <img
        src="{% static 'images/homecraft3d_icon.svg' %}"
        alt="Home Craft 3D"
        style="height:60px;width:auto;"
      >
    </div>
    <h1 style="margin:0 0 8px;font-size:28px;font-weight:600;color:#111827;">Welcome back</h1>
    <p style="margin:0;color:#6b7280;font-size:15px;">Sign in to manage orders and your listings.</p>
  </div>

  <div class="card" style="padding:32px;">
    <h2 style="margin-top:0;">Login</h2>

    <form method="post" novalidate>
      {% csrf_token %}

      {# Preserve ?next=... through POST #}
      <input type="hidden" name="next" value="{{ request.GET.next }}">

      {% if form.non_field_errors %}
        <div class="msg error">{{ form.non_field_errors }}</div>
      {% endif %}

      <div class="form-row-1">
        <div>
          <label for="{{ form.username.id_for_label }}">Username</label>
          {{ form.username }}
          {% for e in form.username.errors %}
            <div class="help" style="color:#b91c1c;">{{ e }}</div>
          {% endfor %}
        </div>

        <div>
          <label for="{{ form.password.id_for_label }}">Password</label>
          {{ form.password }}
          {% for e in form.password.errors %}
            <div class="help" style="color:#b91c1c;">{{ e }}</div>
          {% endfor %}
        </div>
      </div>

      <div class="actions">
        <button class="btn" type="submit">Login</button>
        <a class="btn secondary" href="{% url 'accounts:register' %}">Create account</a>
      </div>
    </form>
  </div>

  <div style="text-align:center;margin-top:24px;">
```

```
    <a href="{% url 'core:home' %}" style="color:#6b7280;font-size:14px;text-decoration:none;">
      ← Back to store
    </a>
  </div>
</div>
{% endblock %}
```

reviews\models.py

```
# reviews/models.py
from __future__ import annotations

from django.conf import settings
from django.core.validators import MaxValueValidator, MinValueValidator
from django.db import models


class Review(models.Model):
    """Buyer review for a purchased product."""

    product = models.ForeignKey(
        "products.Product",
        on_delete=models.CASCADE,
        related_name="reviews",
    )

    order_item = models.OneToOneField(
        "orders.OrderItem",
        on_delete=models.CASCADE,
        related_name="review",
        help_text="Enforces one review per purchased line item.",
    )

    buyer = models.ForeignKey(
        settings.AUTH_USER_MODEL,
        on_delete=models.CASCADE,
        related_name="reviews",
    )

    rating = models.PositiveSmallIntegerField(
        validators=[MinValueValidator(1), MaxValueValidator(5)],
        help_text="1–5 stars",
    )

    title = models.CharField(max_length=120, blank=True, default="")
    body = models.TextField(blank=True, default="")

    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    class Meta:
        ordering = ["-created_at"]
        indexes = [
            models.Index(fields=["product", "created_at"]),
            models.Index(fields=["buyer", "created_at"]),
            models.Index(fields=["rating"]),
        ]

    def __str__(self) -> str:
        return f"Review<{self.product_id}> by {self.buyer_id} ({self.rating}/5)"


class SellerReview(models.Model):
    """Purchased-only rating for a seller.

    Rules:
    - Only an authenticated buyer (Order.buyer) can rate.
    - Order must be PAID.
    - Order must include at least one OrderItem for that seller.

    We bind to the Order itself (not a specific item) so a buyer can leave ONE
    seller rating per order for that seller.
    """

    seller = models.ForeignKey(
        settings.AUTH_USER_MODEL,
        on_delete=models.CASCADE,
        related_name="seller_reviews_received",
    )

    buyer = models.ForeignKey(
        settings.AUTH_USER_MODEL,
        on_delete=models.CASCADE,
        related_name="seller_reviews_written",
    )

    order = models.ForeignKey(
        "orders.Order",
        on_delete=models.CASCADE,
        related_name="seller_reviews",
    )

    rating = models.PositiveSmallIntegerField(
        validators=[MinValueValidator(1), MaxValueValidator(5)],
        help_text="1–5 stars",
    )

    title = models.CharField(max_length=120, blank=True, default="")
    body = models.TextField(blank=True, default="")

    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    class Meta:
        ordering = ["-created_at"]
        constraints = [
            models.UniqueConstraint(fields=["buyer", "seller", "order"], name="uniq_seller_review_per_order"),
        ]
        indexes = [
            models.Index(fields=["seller", "created_at"]),
            models.Index(fields=["buyer", "created_at"]),
            models.Index(fields=["rating"]),
        ]

    def __str__(self) -> str:
        return f"SellerReview<seller={self.seller_id} buyer={self.buyer_id} order={self.order_id}> ({self.rating}/5)"
```

core\models_advert.py

```
# core/models_advert.py
from django.db import models
from django.utils import timezone

class AdvertisementBanner(models.Model):
    title = models.CharField(max_length=200)
    image = models.ImageField(upload_to="ad_banners/")
    link_url = models.URLField(blank=True)
    is_active = models.BooleanField(default=True)
    start_date = models.DateTimeField(default=timezone.now)
    end_date = models.DateTimeField(null=True, blank=True)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    class Meta:
        ordering = ["-created_at"]

    def __str__(self):
        return self.title

    def is_currently_active(self):
        now = timezone.now()
        if not self.is_active:
            return False
        if self.start_date and self.start_date > now:
            return False
        if self.end_date and self.end_date < now:
            return False
        return True
```

core\models_email.py

```
# core/models_email.py
from django.db import models

class SiteEmailTemplate(models.Model):
    site_config = models.ForeignKey(
        'core.SiteConfig',
        on_delete=models.CASCADE,
        related_name='email_templates',
        help_text='Site config this template belongs to.',
        null=True,
        blank=True
    )
    name = models.CharField(max_length=100, unique=True, help_text="Template identifier (e.g. 'staff_announcement')")
    subject = models.CharField(max_length=200, help_text="Email subject line")
    body = models.TextField(help_text="Email body. You can use {{ user }}, {{ site_name }}, etc.")
    is_active = models.BooleanField(default=True)
    updated_at = models.DateTimeField(auto_now=True)

    class Meta:
        verbose_name = "Site Email Template"
        verbose_name_plural = "Site Email Templates"
        ordering = ["-updated_at"]

    def __str__(self):
        return f"{self.name} ({'active' if self.is_active else 'inactive'})"
```

orders\templatetags\money.py

```
# orders/templatetags/money.py
from __future__ import annotations

from decimal import Decimal, ROUND_HALF_UP

from django import template

register = template.Library()


@register.filter(name="cents_to_dollars")
def cents_to_dollars(value) -> str:
    """
    12345 -> "123.45"
    Safely formats cents as dollars for display.
    """
    try:
        cents = int(value or 0)
    except (TypeError, ValueError):
        cents = 0

    dollars = (Decimal(cents) / Decimal("100")).quantize(Decimal("0.01"), rounding=ROUND_HALF_UP)
    return f"{dollars}"
```

orders\templates\orders\my_orders.html

```
{# orders/templates/orders/my_orders.html #}
{% extends "base.html" %}
{% load money %}
{% block title %}My Orders · Home Craft 3D{% endblock %}

{% block content %}
<div class="container py-4">
  <div class="d-flex align-items-center justify-content-between mb-3">
    <h1 class="h4 mb-0">My Orders</h1>
  </div>

  <div class="card">
    <div class="table-responsive">
      <table class="table table-hover align-middle mb-0">
        <thead class="table-light">
          <tr>
            <th>Order</th>
            <th>Status</th>
            <th>Created</th>
            <th class="text-end">Total</th>
            <th class="text-end"></th>
          </tr>
        </thead>
        <tbody>
          {% for o in page_obj.object_list %}
            <tr>
              <td class="fw-semibold">#{{ o.id }}</td>
              <td>{{ o.get_status_display }}</td>
              <td class="text-secondary">{{ o.created_at|date:"M j, Y" }}</td>
              <td class="text-end fw-semibold">${{ o.total_cents|cents_to_dollars }}</td>
              <td class="text-end">
                <a class="btn btn-sm btn-outline-dark" href="{% url 'orders:detail' o.id %}">View</a>
              </td>
            </tr>
          {% empty %}
            <tr>
              <td colspan="5" class="text-center text-secondary py-4">No orders yet.</td>
            </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>
  </div>

  {% if page_obj.has_other_pages %}
    <nav class="mt-3">
      <ul class="pagination mb-0">
        {% if page_obj.has_previous %}
          <li class="page-item">
            <a class="page-link" href="?page={{ page_obj.previous_page_number }}">Previous</a>
          </li>
        {% else %}
          <li class="page-item disabled"><span class="page-link">Previous</span></li>
        {% endif %}

        <li class="page-item disabled"><span class="page-link">
          Page {{ page_obj.number }} of {{ page_obj.paginator.num_pages }}
        </span></li>
        {% if page_obj.has_next %}
          <li class="page-item">
            <a class="page-link" href="?page={{ page_obj.next_page_number }}">Next</a>
          </li>
        {% else %}
          <li class="page-item disabled"><span class="page-link">Next</span></li>
        {% endif %}
      </ul>
    </nav>
  {% endif %}
</div>
{% endblock %}
```

templates\partials\navbar.html

```
{# templates/partials/navbar.html #}
{% load static %}

<nav class="navbar navbar-expand-lg bg-body-tertiary border-bottom" style="min-height: 90px;">
  <div class="container-fluid">

    <!-- BOX 1: LEFT (brand + mobile buttons) -->
    <div class="d-flex align-items-center gap-2">
      {# Send logo to the store home (safe for logged-in and logged-out users) #}
      <a class="navbar-brand py-0 m-0" href="{% url 'core:home' %}">
        <img src="{% static 'images/homecraft3d_navbar.svg' %}" alt="Home Craft 3D"
             style="height:70px !important;width:auto;max-height:70px;">
      </a>

      <button class="btn btn-sm btn-outline-secondary d-lg-none" type="button"
          data-bs-toggle="offcanvas" data-bs-target="#sidebarOffcanvas"
          aria-controls="sidebarOffcanvas" aria-label="Open sidebar menu">
        <i class="bi bi-list" aria-hidden="true"></i>
      </button>

      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#mainNavbar"
          aria-controls="mainNavbar" aria-expanded="false" aria-label="Toggle main navigation">
        <span class="navbar-toggler-icon" aria-hidden="true"></span>
      </button>
    </div>

    <div class="collapse navbar-collapse" id="mainNavbar">
      <!-- 3 BOX LAYOUT -->
      <div class="w-100 d-flex flex-column flex-lg-row align-items-stretch align-items-lg-center gap-2">

        <!-- BOX 2: CENTER (menu items) -->
        <div class="flex-grow-1 d-flex justify-content-lg-center">
          <ul class="navbar-nav mb-2 mb-lg-0 hc3-nav-center">
            {# "Store" should be the store, not the marketing home #}
            <li class="nav-item"><a class="nav-link" href="{% url 'core:home' %}">Storefront</a></li>
            <li class="nav-item"><a class="nav-link" href="{% url 'products:list' %}">All Products</a></li>
            <li class="nav-item"><a class="nav-link" href="{% url 'products:models' %}">3D Models</a></li>
            <li class="nav-item"><a class="nav-link" href="{% url 'products:files' %}">3D Files</a></li>

            <li class="nav-item"><a class="nav-link" href="{% url 'core:coming_soon'
%}?feature=blog">Blog</a></li>
            <li class="nav-item"><a class="nav-link" href="{% url 'core:coming_soon'
%}?feature=community">Community</a></li>

            {% if user.is_authenticated %}
              {% with user_is_seller=user_is_seller|default:False user_is_owner=user_is_owner|default:False %}
                {% if user_is_seller or user_is_owner %}
                  <li class="nav-item dropdown">
                    <a class="nav-link dropdown-toggle" href="#" id="myStoreDropdown" role="button"
data-bs-toggle="dropdown" aria-expanded="false">
                      My Store
                    </a>
                    <ul class="dropdown-menu" aria-labelledby="myStoreDropdown">
                      <li>
                        <a class="dropdown-item" href="{% url 'products:seller_shop' user.id %}">
                          <i class="bi bi-shop me-1"></i>Seller Storefront
                        </a>
                      </li>
                      <li>
                        <a class="dropdown-item" href="{% url 'products:seller_list' %}">
                          <i class="bi bi-card-list me-1"></i>My Listings
                        </a>
                      </li>
                    </ul>
                  </li>
                {% endif %}

                <li class="nav-item dropdown">
                  <a class="nav-link dropdown-toggle" href="#" id="myOrdersDropdown" role="button"
                     data-bs-toggle="dropdown" aria-expanded="false">
                    My Orders
                  </a>
                  <ul class="dropdown-menu" aria-labelledby="myOrdersDropdown">
                    <li>
                      <a class="dropdown-item" href="{% url 'orders:my_orders' %}">
                        <i class="bi bi-receipt me-1"></i>Order history
                      </a>
                    </li>
                    <li>
                      <a class="dropdown-item" href="{% url 'orders:purchases' %}">
                        <i class="bi bi-download me-1"></i>Purchases & downloads
                      </a>
                    </li>
                    <li>
                      <a class="dropdown-item" href="{% url 'orders:refunds:buyer_list' %}">
                        <i class="bi bi-arrow-counterclockwise me-1"></i>Refund requests
                      </a>
                    </li>

                    {% if user_is_seller or user_is_owner %}
                      <li><hr class="dropdown-divider"></li>
                      <li class="dropdown-header">Seller</li>
                      <li>
                        <a class="dropdown-item" href="{% url 'orders:seller_orders_list' %}">
                          <i class="bi bi-box-seam me-1"></i>Seller orders
                        </a>
                      </li>
                      <li>
                        <a class="dropdown-item" href="{% url 'orders:refunds:seller_queue' %}">
                          <i class="bi bi-arrow-counterclockwise me-1"></i>Seller refunds
                        </a>
                      </li>
                    {% endif %}

                    {% if user.is_staff or user.is_superuser %}
                      <li><hr class="dropdown-divider"></li>
                      <li class="dropdown-header">Staff</li>
                      <li>
                        <a class="dropdown-item" href="{% url 'orders:refunds:staff_queue' %}">
                          <i class="bi bi-shield-check me-1"></i>Refund safety valve
                        </a>
                      </li>
                    {% endif %}
                  </ul>
                </li>
              {% endwith %}
            {% endif %}
          </ul>
        </div>
```

```
        <!-- BOX 3: RIGHT (dashboard + search + cart + profile + theme) -->
        <div class="d-flex flex-column flex-lg-row align-items-stretch align-items-lg-center gap-2 ms-lg-auto">

          {% if user.is_authenticated %}
            <a class="nav-link px-0 px-lg-2" href="{% url 'dashboards:home' %}">Dashboard</a>
          {% endif %}

          <form class="d-flex" method="get" action="{% url 'products:list' %}" role="search" style="min-width: 260px;">
            <input class="form-control form-control-sm me-2" type="search" name="q" placeholder="Search"
                  aria-label="Search">
            <button class="btn btn-sm btn-outline-primary" type="submit">Search</button>
          </form>

          <a class="nav-link px-0 px-lg-2 position-relative" href="{% url 'cart:detail' %}">
            <i class="bi bi-bag me-1"></i>Cart
            {% if cart_item_count and cart_item_count > 0 %}
              <span class="position-absolute top-0 start-100 translate-middle badge rounded-pill bg-danger">
                {{ cart_item_count }}
                <span class="visually-hidden">items in cart</span>
              </span>
            {% endif %}
          </a>

          {% if user.is_authenticated %}
            <div class="nav-item dropdown">
              <a class="nav-link dropdown-toggle px-0 px-lg-2" href="#" id="userDropdown" role="button"
                 data-bs-toggle="dropdown" aria-expanded="false">
                {% if user.profile.avatar %}
                  <img src="{{ user.profile.avatar.url }}" alt="{{ user.username }}" class="rounded-circle me-1" style="width: 32px; height: 32px; object-fit: cover;">
                {% else %}
                  <i class="bi bi-person-circle me-1"></i>
                {% endif %}
                {{ user.username }}
              </a>
              <ul class="dropdown-menu dropdown-menu-end" aria-labelledby="userDropdown">
                <li>
                  <a class="dropdown-item" href="{% url 'accounts:profile' %}">
                    <i class="bi bi-person me-1"></i>Profile
                  </a>
                </li>
                <li>
                  <a class="dropdown-item" href="{% url 'dashboards:consumer' %}">
                    <i class="bi bi-person-badge me-1"></i>Account
                  </a>
                </li>
                {% if user.is_staff or user.is_superuser %}
                  <li>
                    <a class="dropdown-item" href="{% url 'dashboards:admin' %}">
                      <i class="bi bi-shield-lock me-1"></i>Admin
                    </a>
                  </li>
                {% endif %}
                <li><hr class="dropdown-divider"></li>
                <li>
                  <a class="dropdown-item" href="{% url 'accounts:logout' %}">
                    <i class="bi bi-box-arrow-right me-1"></i>Logout
                  </a>
                </li>
              </ul>
          </div>
        {% else %}
          <a class="nav-link px-0 px-lg-2" href="{% url 'accounts:login' %}">Login</a>
          <a class="btn btn-primary btn-sm" href="{% url 'accounts:register' %}">Sign up</a>
        {% endif %}

        <button id="hcThemeToggle" class="btn btn-sm btn-outline-secondary" type="button"
                aria-pressed="false" aria-label="Toggle dark/light theme" title="Toggle theme">
          <i class="bi bi-moon-stars" aria-hidden="true"></i>
        </button>

      </div>
    </div>
   </div>
  </div>
</nav>
```

templates\emails\order_canceled.html

```
{# templates/emails/order_canceled.html #}
{% extends "emails/base.html" %}
{% block content %}
  <h1 style="margin:0 0 12px;font-size:20px;">Order canceled</h1>
  <p style="margin:0 0 12px;">Your order <strong>#{{ order_id }}</strong> was canceled.</p>

  <div style="margin:16px 0;padding:16px;border:1px solid #e5e7eb;border-radius:10px;background:#f9fafb;">
    <div style="font-size:12px;color:#6b7280;margin-bottom:6px;">Order details</div>
    <a href="{{ order_link }}" style="color:#2563eb;text-decoration:none;">View your order</a>
  </div>

  <p style="margin:0;color:#6b7280;font-size:13px;">If this was a mistake, you can return to your cart and try again.</p>
{% endblock %}
```

orders\templates\orders\seller\order_detail.html

```
{# orders/templates/orders/seller_order_detail.html #}
{% extends "base.html" %}
{% load money %}
{% block title %}Seller Order #{{ order.id }} · Home Craft 3D{% endblock %}

{% block content %}
<div class="container-fluid">
  <div class="d-flex flex-wrap align-items-center justify-content-between gap-2 mb-3">
    <div>
      <h1 class="h4 mb-1">Order #{{ order.id }}</h1>
      <div class="text-secondary small">
        Status: <span class="fw-semibold">{{ order.get_status_display }}</span>
        · Your total: <span class="fw-semibold">${{ seller_total|floatformat:2 }}</span>
      </div>
    </div>

    <div class="d-flex gap-2">
      <a class="btn btn-outline-dark" href="{% url 'orders:seller_orders_list' %}">
        <i class="bi bi-arrow-left me-1"></i>Back to orders
      </a>
    </div>
  </div>

  {% if order.status != "PAID" %}
    <div class="alert alert-warning">
      This order is not paid yet. Shipping actions are disabled until payment is complete.
    </div>
  {% endif %}

  <div class="row g-3">
    <div class="col-lg-8">
      <div class="card">
        <div class="card-header d-flex align-items-center justify-content-between">
          <div class="fw-semibold">Your items in this order</div>
        </div>
        <div class="card-body">
          {% if items %}
            <div class="table-responsive">
              <table class="table table-sm align-middle">
                <thead>
                  <tr>
                    <th>Item</th>
                    <th>Kind</th>
                    <th class="text-end">Qty</th>
                    <th class="text-end">Line total</th>
                    <th>Fulfillment</th>
                  </tr>
                </thead>
                <tbody>
                  {% for it in items %}
                    <tr>
                      <td class="fw-semibold">
                        <a class="text-decoration-none" href="{% url 'products:detail' pk=it.product_id slug=it.product.slug
%}">{{ it.product.title }}</a>
                        {% if it.carrier or it.tracking_number %}
                          <div class="text-secondary small d-flex flex-wrap align-items-center gap-2">
                            <div>
                              {% if it.carrier %}Carrier: <span class="font-monospace">{{ it.carrier }}</span>{% endif %}
                              {% if it.tracking_number %}
```

```
                                {% if it.carrier %} · {% endif %}
                                Tracking: <span class="font-monospace" data-tracking-value="{{ it.tracking_number }}">{{
it.tracking_number }}</span>
                              {% endif %}
                            </div>
                            {% if it.tracking_number %}
                                <a href="https://trackingmore.com/track/?tracknumber={{ it.tracking_number|urlencode }}"
target="_blank" rel="noopener">Track</a>
                                <button class="btn btn-sm btn-outline-secondary" type="button" data-copy-tracking="{{
it.tracking_number }}">Copy</button>
                            {% endif %}
                          </div>
                        {% endif %}
                        {% if it.buyer_notes %}
                          <div class="alert alert-info alert-sm mt-2 mb-0 p-2">
                            <div class="small fw-semibold">
                              <i class="bi bi-chat-left-text me-1"></i>Buyer's Special Instructions:
                            </div>
                            <div class="small" style="white-space: pre-wrap;">{{ it.buyer_notes }}</div>
                          </div>
                        {% endif %}
                      </td>
                      <td>
                        {% if it.is_digital %}
                          <span class="badge text-bg-info">Digital</span>
                        {% else %}
                          <span class="badge text-bg-secondary">Physical</span>
                        {% endif %}
                      </td>
                      <td class="text-end">{{ it.quantity }}</td>
                      <td class="text-end"><strong>${{ it.line_total_cents|cents_to_dollars }}</strong></td>
                      <td>
                        {% if it.is_digital %}
                          <span class="badge text-bg-success">Delivered</span>
                          <div class="text-secondary small">Downloads handled automatically.</div>
                        {% else %}
                          {% if it.fulfillment_status == "delivered" %}
                            <span class="badge text-bg-success">Delivered</span>
                          {% elif it.fulfillment_status == "shipped" %}
                            <span class="badge text-bg-success">Shipped</span>
                            {% if it.shipped_at %}
                              <div class="text-secondary small">{{ it.shipped_at|date:"M j, Y" }}</div>
                            {% endif %}
                            {% if order.status == "PAID" %}
                              <form class="mt-2" method="post" action="{% url 'orders:mark_item_delivered' order.id it.id
%}">
                                {% csrf_token %}
                                <button class="btn btn-sm btn-outline-dark" type="submit">
                                  <i class="bi bi-check2-circle me-1"></i>Mark delivered
                                </button>
                              </form>
                            {% endif %}
                          {% else %}
                            <span class="badge text-bg-warning text-dark">Unfulfilled</span>

                            {% if order.status == "PAID" %}
                              <form class="mt-2" method="post" action="{% url 'orders:mark_item_shipped' order.id it.id
%}">
                                {% csrf_token %}
                                <div class="row g-2">
                                  <div class="col-md-5">
```

```
                                    <select class="form-select form-select-sm" name="carrier">
                                      <option value="">Carrier (optional)</option>
                                      <option>USPS</option>
                                      <option>UPS</option>
                                      <option>FedEx</option>
                                      <option>DHL</option>
                                      <option>Other</option>
                                    </select>
                                    <input class="form-control form-control-sm mt-2 d-none" name="carrier_other" placeholder="Enter carrier" data-carrier-other>
                                  </div>
                                  <div class="col-md-7">
                                    <input class="form-control form-control-sm" name="tracking_number" placeholder="Tracking number (optional)" />
                                  </div>
                                </div>
                                  <div class="form-text small">Add carrier and tracking number from your label/receipt. We’ll email it to the buyer.</div>
                                <div class="mt-2">
                                  <button class="btn btn-sm btn-dark" type="submit">
                                    <i class="bi bi-truck me-1"></i>Mark shipped
                                  </button>
                                </div>
                              </form>
                            {% endif %}
                          {% endif %}
                        {% endif %}
                      </td>
                    </tr>
                  {% endfor %}
                </tbody>
              </table>
            </div>
          {% else %}
            <div class="text-secondary">No items for you in this order.</div>
          {% endif %}
        </div>
      </div>
    </div>

    <div class="col-lg-4">
      <div class="card mb-3">
        <div class="card-header"><div class="fw-semibold">Buyer & Shipping</div></div>
        <div class="card-body">
          <div class="mb-2">
            <div class="text-secondary small">Buyer</div>
            <div class="fw-semibold">
              {% if order.buyer %}{{ order.buyer.username }}{% else %}Guest{% endif %}
            </div>
          </div>

          <div>
            <div class="text-secondary small">Ship to</div>
            {% if order.ship_line1 or order.ship_city or order.ship_postal_code %}
              <div class="small">
                <div class="fw-semibold">{{ order.ship_name }}</div>
                <div>{{ order.ship_line1 }}</div>
                {% if order.ship_line2 %}<div>{{ order.ship_line2 }}</div>{% endif %}
                <div>{{ order.ship_city }}, {{ order.ship_state }} {{ order.ship_postal_code }}</div>
                <div>{{ order.ship_country }}</div>
                {% if order.ship_phone %}<div class="mt-1">Phone: {{ order.ship_phone }}</div>{% endif %}
              </div>
            {% else %}
              <div class="text-secondary small">No shipping address recorded.</div>
            {% endif %}
          </div>
        </div>
      </div>

      <div class="card">
        <div class="card-header"><div class="fw-semibold">Order snapshot</div></div>
        <div class="card-body">
          <div class="d-flex justify-content-between">
            <div class="text-secondary">Created</div>
            <div class="fw-semibold">{{ order.created_at|date:"M j, Y" }}</div>
          </div>
          {% if order.paid_at %}
          <div class="d-flex justify-content-between">
            <div class="text-secondary">Paid</div>
            <div class="fw-semibold">{{ order.paid_at|date:"M j, Y" }}</div>
          </div>
          {% endif %}
        </div>
      </div>

    </div>
  </div>
</div>

<script>
  document.addEventListener('click', function (event) {
    const btn = event.target.closest('[data-copy-tracking]');
    if (!btn) return;

    const value = btn.getAttribute('data-copy-tracking') || '';
    if (!value) return;

    navigator.clipboard.writeText(value).then(function () {
      const original = btn.textContent;
      btn.textContent = 'Copied';
      setTimeout(function () {
        btn.textContent = original;
      }, 1200);
    });
  });

  document.addEventListener('change', function (event) {
    const select = event.target.closest('select[name="carrier"]');
    if (!select) return;
    const form = select.closest('form');
    if (!form) return;
    const otherInput = form.querySelector('[data-carrier-other]');
    if (!otherInput) return;

    if (select.value === 'Other') {
      otherInput.classList.remove('d-none');
      otherInput.focus();
    } else {
      otherInput.classList.add('d-none');
      otherInput.value = '';
    }
  });
  document.addEventListener('submit', function (event) {
    const form = event.target.closest('form');
    if (!form) return;
    const select = form.querySelector('select[name="carrier"]');
    const otherInput = form.querySelector('[data-carrier-other]');
    if (!select || !otherInput) return;
    if (select.value === 'Other' && otherInput.value.trim()) {
      select.value = otherInput.value.trim();
    }
  });
</script>
{% endblock %}
```

templates\emails\order_failed.html

```
{# templates/emails/order_failed.html #}
{% extends "emails/base.html" %}
{% block content %}
  <h1 style="margin:0 0 12px;font-size:20px;">Payment failed</h1>
  <p style="margin:0 0 12px;">We couldn't process payment for order <strong>#{{ order_id }}</strong>.</p>

  <p style="margin:0 0 12px;color:#6b7280;">{{ reason }}</p>

  <div style="margin:16px 0;padding:16px;border:1px solid #e5e7eb;border-radius:10px;background:#f9fafb;">
    <div style="font-size:12px;color:#6b7280;margin-bottom:6px;">Order details</div>
    <a href="{{ order_link }}" style="color:#2563eb;text-decoration:none;">View your order</a>
  </div>

  <p style="margin:0;color:#6b7280;font-size:13px;">You can retry checkout from your order page.</p>
{% endblock %}
```

orders\templates\orders\seller\order_list.html

```
{# orders/templates/orders/seller/order_list.html #}
{% extends "base.html" %}
{% block title %}Seller Orders · Home Craft 3D{% endblock %}

{% block content %}
<div class="container-fluid">
  <div class="d-flex flex-wrap align-items-center justify-content-between gap-2 mb-3">
    <div>
      <h1 class="h4 mb-1">Seller Orders</h1>
      <div class="text-secondary small">Orders that include your items.</div>
    </div>

    <form class="d-flex gap-2" method="get">
      <input class="form-control" name="q" value="{{ q }}" placeholder="Search orders, items, buyer, tracking..." style="min-width:280px;">
      <button class="btn btn-outline-dark" type="submit">Search</button>
    </form>
  </div>

  <div class="card">
    <div class="table-responsive">
      <table class="table table-hover align-middle mb-0">
        <thead class="table-light">
          <tr>
            <th>Order</th>
            <th>Status</th>
            <th>Buyer</th>
            <th class="text-end">Your total</th>
            <th class="text-end">Action</th>
          </tr>
        </thead>
        <tbody>
          {% for o in orders %}
            <tr>
              <td class="fw-semibold">#{{ o.id }}</td>
              <td>
                {% if o.status == "PAID" %}
                  <span class="badge text-bg-success">{{ o.get_status_display }}</span>
                {% elif o.status == "PENDING" %}
                  <span class="badge text-bg-secondary">{{ o.get_status_display }}</span>
                {% else %}
                  <span class="badge text-bg-secondary">{{ o.get_status_display }}</span>
                {% endif %}
              </td>
              <td class="text-secondary">
                {% if o.buyer %}
                  {{ o.buyer.username }}
                {% else %}
                  Guest
                {% endif %}
              </td>
              <td class="text-end"><strong>${{ o.seller_total|floatformat:2 }}</strong></td>
              <td class="text-end">
                <a class="btn btn-sm btn-outline-dark" href="{% url 'orders:seller_order_detail' o.id %}">
                  View
                </a>
              </td>
            </tr>
          {% empty %}
```

```
            <tr>
              <td colspan="5" class="text-center text-secondary py-4">No seller orders yet.</td>
            </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>
  </div>
</div>
{% endblock %}
```

templates\emails\order_paid.html

```
{# templates/emails/order_paid.html #}
{% extends "emails/base.html" %}
{% block content %}
  <h1 style="margin:0 0 12px;font-size:20px;">Thanks for your purchase!</h1>
  <p style="margin:0 0 12px;">Your order <strong>#{{ order_id }}</strong> is confirmed.</p>

  <div style="margin:16px 0;padding:16px;border:1px solid #e5e7eb;border-radius:10px;background:#f9fafb;">
    <div style="font-size:12px;color:#6b7280;margin-bottom:6px;">Order link</div>
    <a href="{{ order_link }}" style="color:#2563eb;text-decoration:none;">View your order</a>
  </div>

  {% if downloads %}
    <h2 style="margin:16px 0 8px;font-size:16px;">Your downloads</h2>
    <ul style="padding-left:18px;margin:0;">
      {% for d in downloads %}
        <li style="margin:6px 0;">
          <strong>{{ d.filename }}</strong><br>
          <a href="{{ d.url }}" style="color:#2563eb;text-decoration:none;">Download</a>
        </li>
      {% endfor %}
    </ul>
  {% endif %}

  {% if is_guest %}
    <p style="margin:16px 0 0;color:#6b7280;font-size:13px;">Keep this link safe so you can access your order later.</p>
  {% endif %}
{% endblock %}
```

templates\emails\payout_sent.html

```
{# templates/emails/payout_sent.html #}
{% extends "emails/base.html" %}
{% block content %}
  <h1 style="margin:0 0 12px;font-size:20px;">Transfer sent</h1>
  <p style="margin:0 0 12px;">A transfer was sent for order <strong>#{{ order_id }}</strong>.</p>

  <div style="margin:16px 0;padding:16px;border:1px solid #e5e7eb;border-radius:10px;background:#f9fafb;">
    <div style="font-size:12px;color:#6b7280;margin-bottom:6px;">Payout summary</div>
    <div style="font-weight:600;">{{ payout_amount }}</div>
    <div style="color:#6b7280;font-size:13px;">Balance before transfer: {{ balance_before }}</div>
    {% if transfer_id %}
      <div style="color:#6b7280;font-size:12px;">Transfer ID: {{ transfer_id }}</div>
    {% endif %}
  </div>

  <a href="{{ order_link }}" style="color:#2563eb;text-decoration:none;">View order details</a>
{% endblock %}
```

payments\templates\payments\payouts_dashboard.html

```
{# payments/templates/payments/payouts_dashboard.html #}
{% extends "base.html" %}
{% block title %}Payouts & Balance · Home Craft 3D{% endblock %}

{% block content %}
<div class="container py-4">

  <div class="d-flex justify-content-between align-items-start gap-3 flex-wrap">
    <div>
      <h1 class="h3 mb-1">Payouts & Balance</h1>
      <div class="text-muted">This is your platform ledger. Stripe transfers and refund/chargeback adjustments show up here.</div>
    </div>

    <div class="text-end">
      <div class="small text-muted">Current balance</div>
      <div class="h4 mb-0 {% if balance_cents < 0 %}text-danger{% elif balance_cents > 0 %}text-success{% endif %}">
        {{ balance_cents|cents_to_dollars }}

      </div>
      <div class="small text-muted">
        {% if balance_cents > 0 %}
          Platform owes you
        {% elif balance_cents < 0 %}
          You owe the platform
        {% else %}
          Settled
        {% endif %}
      </div>
    </div>
  </div>

  {% if not stripe_ready %}
    <div class="alert alert-warning mt-3">
      <div class="d-flex justify-content-between align-items-center flex-wrap gap-2">
        <div>
          <strong>Stripe onboarding not complete.</strong>
          You won’t be able to receive payouts until Stripe is fully enabled.
        </div>
        <a class="btn btn-sm btn-outline-dark" href="{% url 'payments:connect_status' %}">Finish Stripe setup</a>
      </div>
    </div>
  {% endif %}

  <div class="card mt-3">
    <div class="card-body">
      <form method="get" class="row g-2 align-items-end">
        <div class="col-12 col-md-4">
          <label class="form-label">Reason</label>
          <select name="reason" class="form-select">
            <option value="">All</option>
            {% for val,label in reasons %}
              <option value="{{ val }}" {% if reason == val %}selected{% endif %}>{{ label }}</option>
            {% endfor %}
          </select>
        </div>

        <div class="col-12 col-md-5">
```

```
          <label class="form-label">Search</label>
          <input name="q" class="form-control" value="{{ q }}" placeholder="Search notes or order id...">
        </div>

        <div class="col-12 col-md-3 d-flex gap-2">
          <button class="btn btn-primary w-100" type="submit">Filter</button>
          <a class="btn btn-outline-secondary w-100" href="{% url 'payments:payouts_dashboard' %}">Reset</a>
        </div>
      </form>
    </div>
  </div>

  <div class="card mt-3">
    <div class="card-header d-flex justify-content-between align-items-center">
      <div class="fw-semibold">Ledger entries</div>
      {% if page_obj %}
        <div class="small text-muted">
          Page {{ page_obj.number }} of {{ page_obj.paginator.num_pages }}
        </div>
      {% endif %}
    </div>

    <div class="table-responsive">
      <table class="table table-sm mb-0 align-middle">
        <thead>
          <tr class="small text-muted">
            <th style="width: 170px;">Date</th>
            <th style="width: 140px;">Reason</th>
            <th style="width: 160px;" class="text-end">Amount</th>
            <th>Order</th>
            <th>Note</th>
          </tr>
        </thead>
        <tbody>
          {% for e in entries %}
            <tr>
              <td class="small">{{ e.created_at|date:"Y-m-d H:i" }}</td>
              <td class="small">{{ e.get_reason_display }}</td>
              <td class="text-end fw-semibold {% if e.amount_cents < 0 %}text-danger{% elif e.amount_cents > 0
%}text-success{% endif %}">
                <span class="{% if e.amount_cents < 0 %}text-danger{% elif e.amount_cents > 0 %}text-success{%
endif %}">
  {{ e.amount_cents|cents_to_dollars }}
</span>

              </td>
              <td class="small">
                {% if e.order %}
                  <a href="{% url 'orders:detail' e.order.id %}">Order {{ e.order.id }}</a>
                {% else %}
                  N/A
                {% endif %}
              </td>
              <td class="small text-muted">{{ e.note|default:"N/A" }}</td>
            </tr>
          {% empty %}
            <tr>
              <td colspan="5" class="text-center text-muted py-4">No ledger entries yet.</td>
            </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>

    {% if page_obj and page_obj.paginator.num_pages > 1 %}
      <div class="card-footer">
        <nav>
          <ul class="pagination pagination-sm mb-0">
            {% if page_obj.has_previous %}
              <li class="page-item">
                <a class="page-link" href="?page={{ page_obj.previous_page_number }}&reason={{ reason }}&q={{ q }}">Prev</a>
              </li>
            {% else %}
              <li class="page-item disabled"><span class="page-link">Prev</span></li>
            {% endif %}

            <li class="page-item disabled"><span class="page-link">{{ page_obj.number }} / {{ page_obj.paginator.num_pages }}</span></li>

            {% if page_obj.has_next %}
              <li class="page-item">
                <a class="page-link" href="?page={{ page_obj.next_page_number }}&reason={{ reason }}&q={{ q }}">Next</a>
              </li>
            {% else %}
              <li class="page-item disabled"><span class="page-link">Next</span></li>
            {% endif %}
          </ul>
        </nav>
      </div>
    {% endif %}
  </div>

  <div class="mt-3 small text-muted">
    <div><strong>Note:</strong> Positive amounts mean the platform owes you; negative amounts mean you owe the platform.</div>
    <div>Stripe may show payout timing separately in your Stripe Express dashboard.</div>
  </div>

</div>
{% endblock %}
```

products\permissions.py

```
# products/permissions.py
from __future__ import annotations

from functools import wraps
from typing import Callable

from django.contrib.auth.decorators import login_required
from django.http import HttpRequest, HttpResponse
from django.shortcuts import redirect
from django.urls import reverse


def _get_profile(user):
    # Works with a typical OneToOne Profile named `profile`
    return getattr(user, "profile", None)


def is_owner_user(user) -> bool:
    if not user or not user.is_authenticated:
        return False
    # Owner/admin override paths
    if getattr(user, "is_superuser", False) or getattr(user, "is_staff", False):
        return True
    profile = _get_profile(user)
    return bool(getattr(profile, "is_owner", False))


def is_seller_user(user) -> bool:
    if not user or not user.is_authenticated:
        return False
    if is_owner_user(user):
        return True
    profile = _get_profile(user)
    return bool(getattr(profile, "is_seller", False))


def seller_required(view_func: Callable[[HttpRequest, ...], HttpResponse]):
    """
    Requires:
      - authenticated
      - seller OR owner/admin
    """
    @login_required
    @wraps(view_func)
    def _wrapped(request: HttpRequest, *args, **kwargs):
        if not is_seller_user(request.user):
            return redirect(reverse("home"))
        return view_func(request, *args, **kwargs)

    return _wrapped
```

dashboards\plausible.py

```
# dashboards/plausible.py
from __future__ import annotations

from typing import Any, Dict, List, Optional

import requests
from django.conf import settings

BASE_URL = "https://plausible.io/api/v1/stats"


def is_configured() -> bool:
    return bool(settings.PLAUSIBLE_API_KEY and settings.PLAUSIBLE_SITE_ID)


def _headers() -> Dict[str, str]:
    return {"Authorization": f"Bearer {settings.PLAUSIBLE_API_KEY}"}


def _get(path: str, params: Dict[str, Any]) -> Dict[str, Any]:
    url = f"{BASE_URL}/{path.lstrip('/')}"
    response = requests.get(url, headers=_headers(), params=params, timeout=10)
    response.raise_for_status()
    return response.json()


def get_summary(
    period: str = "30d",
    *,
    from_date: Optional[str] = None,
    to_date: Optional[str] = None,
    filters: Optional[str] = None,
) -> Dict[str, Any]:
    if not is_configured():
        return {}

    params = {
        "site_id": settings.PLAUSIBLE_SITE_ID,
        "period": period,
        "metrics": "visitors,pageviews,visits,bounce_rate,visit_duration",
    }
    if from_date and to_date:
        params["date"] = f"{from_date},{to_date}"
    elif from_date:
        params["date"] = from_date
    if filters:
        params["filters"] = filters
    data = _get("aggregate", params)
    return data.get("results", {}) or {}


def get_top_pages(
    period: str = "30d",
    limit: int = 10,
    *,
    from_date: Optional[str] = None,
    to_date: Optional[str] = None,
    filters: Optional[str] = None,
) -> List[Dict[str, Any]]:
```

```
if not is_configured():
    return []

params = {
    "site_id": settings.PLAUSIBLE_SITE_ID,
    "period": period,
    "property": "event:page",
    "metrics": "pageviews,visitors",
    "limit": str(limit),
}
if from_date and to_date:
    params["date"] = f"{from_date},{to_date}"
elif from_date:
    params["date"] = from_date
if filters:
    params["filters"] = filters
data = _get("breakdown", params)
return data.get("results", []) or []
```

legal\templates\legal\privacy.html

```
{# templates/legal/privacy.html #}
{% extends "legal/_legal_base.html" %}

{% block doc_content %}
<h2>1. Information We Collect</h2>
<ul>
  <li>Account details (email, username)</li>
  <li>Order and payment details (processed securely by our payment providers)</li>
  <li>Usage data (pages visited, actions taken)</li>
  <li>Device and browser information</li>
</ul>

<h2>2. How We Use Information</h2>
<ul>
  <li>To process orders and deliver digital downloads</li>
  <li>To provide customer support</li>
  <li>To improve site performance and security</li>
  <li>To comply with legal obligations</li>
</ul>

<h2>3. Sharing</h2>
<p>We share only what's necessary with sellers (to fulfill orders) and payment providers (to process payments). We
do not sell personal data.</p>

<h2>4. Cookies</h2>
<p>We use cookies for login sessions, cart functionality, and analytics. You can manage cookies in your browser
settings.</p>

<h2>5. Data Retention</h2>
<p>We keep data as long as needed for orders, legal compliance, and support.</p>

<h2>6. Your Rights</h2>
<p>You can request access, corrections, or deletion of your data by contacting support.</p>

<p><strong>Contact:</strong> <a
href="mailto:homecraft3dstore@gmail.com">homecraft3dstore@gmail.com</a></p>
{% endblock %}
```

config\settings\prod.py

```
# config/settings/prod.py
"""
Production settings.

These settings are for the live environment.
"""

from .base import *  # noqa
import os

# ------------------------------------------------------------------------------
# CORE
# ------------------------------------------------------------------------------

DEBUG = False

ALLOWED_HOSTS = ["homecraft3d.com", "www.homecraft3d.com", "homecraft3d.onrender.com"]

CSRF_TRUSTED_ORIGINS = ["https://homecraft3d.com", "https://www.homecraft3d.com",
"https://homecraft3d.onrender.com"]

# ------------------------------------------------------------------------------
# SECURITY
# ------------------------------------------------------------------------------

SECURE_SSL_REDIRECT = True
SESSION_COOKIE_SECURE = True
CSRF_COOKIE_SECURE = True

SESSION_COOKIE_DOMAIN = ".homecraft3d.com"
CSRF_COOKIE_DOMAIN = ".homecraft3d.com"

SECURE_BROWSER_XSS_FILTER = True
SECURE_CONTENT_TYPE_NOSNIFF = True
X_FRAME_OPTIONS = "DENY"

# If you're behind a reverse proxy / load balancer:
SECURE_PROXY_SSL_HEADER = ("HTTP_X_FORWARDED_PROTO", "https")
USE_X_FORWARDED_HOST = True

# Throttling should trust proxy headers only when you control them.
THROTTLE_TRUST_PROXY_HEADERS = True

# ------------------------------------------------------------------------------
# EMAIL (real backend required)
# ------------------------------------------------------------------------------

EMAIL_BACKEND = "django.core.mail.backends.smtp.EmailBackend"

EMAIL_HOST = os.getenv("EMAIL_HOST", "smtp.gmail.com")
EMAIL_PORT = int(os.getenv("EMAIL_PORT", "587"))
EMAIL_USE_TLS = os.getenv("EMAIL_USE_TLS", "True").strip().lower() in ("1", "true", "yes", "on")

EMAIL_HOST_USER = os.getenv("EMAIL_HOST_USER", "homecraft3dstore@gmail.com")
EMAIL_HOST_PASSWORD = os.getenv("EMAIL_HOST_PASSWORD", "")
DEFAULT_FROM_EMAIL = os.getenv("DEFAULT_FROM_EMAIL", "homecraft3dstore@gmail.com")

# ------------------------------------------------------------------------------
# LOGGING
# ------------------------------------------------------------------------------

LOGGING = {
    "version": 1,
    "disable_existing_loggers": False,
    "handlers": {
        "console": {
            "class": "logging.StreamHandler",
        },
    },
    "root": {
        "handlers": ["console"],
        "level": "WARNING",
    },
}
```

products\templates\products\seller\product_assets.html

```
{# products/templates/products/seller/product_assets.html #}
{% extends "base.html" %}
{% load static %}
{% block title %}Seller · Assets · {{ product.title }} · Home Craft 3D{% endblock %}

{% block content %}
<div class="container-fluid" style="max-width: 980px;">
  <div class="d-flex align-items-center justify-content-between mb-3">
    <div>
      <h1 class="h4 mb-1">Digital Assets</h1>
      <div class="text-secondary small">{{ product.title }}</div>
    </div>
    <div class="d-flex gap-2">
      <a class="btn btn-outline-dark" href="{% url 'products:seller_edit' product.pk %}">
        <i class="bi bi-pencil me-1"></i>Edit Listing
      </a>
      <a class="btn btn-outline-dark" href="{% url 'products:seller_images' product.pk %}">
        <i class="bi bi-images me-1"></i>Manage Images
      </a>
      <a class="btn btn-outline-dark" href="{% url 'products:seller_specs' product.pk %}">
        <i class="bi bi-rulers me-1"></i>Manage Specs
      </a>
      <a class="btn btn-outline-dark" href="{% url 'products:seller_list' %}">
        <i class="bi bi-arrow-left me-1"></i>Back
      </a>
    </div>
  </div>

  <div class="card mb-3">
    <div class="card-body">
      <h2 class="h6">Upload Asset</h2>
      <form method="post" enctype="multipart/form-data">
        {% csrf_token %}
        <div class="row g-3">
          <div class="col-12">
            <label class="form-label">File</label>
            {{ form.file }}
            {% if form.file.errors %}<div class="text-danger small">{{ form.file.errors }}</div>{% endif %}
          </div>
          <div class="col-12 col-md-4">
            <label class="form-label">File type</label>
            {{ form.file_type }}
            {% if form.file_type.errors %}<div class="text-danger small">{{ form.file_type.errors }}</div>{% endif %}
          </div>
          <div class="col-12">
            <label class="form-label">Original filename (optional)</label>
            {{ form.original_filename }}
          </div>
        </div>
        <button class="btn btn-dark mt-3" type="submit">
          <i class="bi bi-upload me-1"></i>Upload
        </button>
      </form>
    </div>
  </div>

  <div class="card">
    <div class="card-body">
      <h2 class="h6 mb-3">Current Assets</h2>
      <div class="table-responsive">
        <table class="table table-sm align-middle">
          <thead class="table-light">
            <tr>
              <th>ID</th>
              <th>Filename</th>
              <th>Type</th>
              <th>Contents</th>
              <th>Stored Path</th>
              <th class="text-end">Action</th>
            </tr>
          </thead>
          <tbody>
            {% for a in assets %}
              <tr>
                <td>{{ a.id }}</td>
                <td>{{ a.original_filename|default:"(not set)" }}</td>
                <td>{% if a.file_type %}.{{ a.file_type }}{% else %}N/A{% endif %}</td>
                <td class="small text-secondary">
                  {% if a.file_type == "zip" and a.zip_contents %}
                    <ul class="mb-0">
                      {% for item in a.zip_contents|slice:":8" %}
                        <li>{{ item }}</li>
                      {% endfor %}
                      {% if a.zip_contents|length > 8 %}
                        <li>+ {{ a.zip_contents|length|add:"-8" }} more...</li>
                      {% endif %}
                    </ul>
                  {% else %}
                    N/A
                  {% endif %}
                </td>
                <td class="text-secondary small">{{ a.file.name }}</td>
                <td class="text-end">
                  <form method="post" action="{% url 'products:seller_asset_delete' product.pk a.pk %}">
                    {% csrf_token %}
                    <button class="btn btn-sm btn-outline-danger" type="submit">
                      <i class="bi bi-trash me-1"></i>Delete
                    </button>
                  </form>
                </td>
              </tr>
            {% empty %}
              <tr>
                <td colspan="4" class="text-center text-secondary py-3">No assets uploaded yet.</td>
              </tr>
            {% endfor %}
          </tbody>
        </table>
      </div>
    </div>
  </div>

  {% if messages %}
    <div class="mt-3">
      {% for message in messages %}
        <div class="alert alert-info mb-2">{{ message }}</div>
      {% endfor %}
    </div>
  {% endif %}
```

```
</div>
{% endblock %}
```

products\templates\products\product_detail.html

```
{# templates/products/product_detail.html #}
{% extends "base.html" %}
{% load static %}
{% block title %}{{ product.title }} · Home Craft 3D{% endblock %}
{% block meta_title %}{{ product.title }} · Home Craft 3D{% endblock %}
{% block og_type %}product{% endblock %}

{% block og_title %}
{{ product.title }} – Home Craft 3D
{% endblock %}

{% block og_description %}
{{ product.short_description|default:product.description|truncatechars:160 }}
{% endblock %}

{% block og_image %}
  {% if product.images.first %}
    https://{{ request.get_host }}{{ product.images.first.image.url }}
  {% else %}
    https://{{ request.get_host }}{% static 'images/homecraft3d_facebook.png' %}
  {% endif %}
{% endblock %}

{% block twitter_title %}{{ product.title }} – Home Craft 3D{% endblock %}
{% block twitter_description %}{{ product.short_description|default:product.description|striptags|truncatechars:160
}}{% endblock %}
{% block twitter_image %}
  {% if product.images.first %}
    https://{{ request.get_host }}{{ product.images.first.image.url }}
  {% else %}
    https://{{ request.get_host }}{% static 'images/homecraft3d_facebook.png' %}
  {% endif %}
{% endblock %}

{% block meta_description %}
  {% if product.short_description %}
    {{ product.short_description }}
  {% elif product.description %}
    {{ product.description|striptags|truncatechars:160 }}
  {% else %}
    Explore this product on Home Craft 3D.
  {% endif %}
{% endblock %}

{% block content %}
<div class="container-fluid">
  {% if messages %}
    <div class="mb-3">
      {% for message in messages %}
        <div class="alert alert-{{ message.tags }} mb-2">{{ message }}</div>
      {% endfor %}
    </div>
  {% endif %}

  {% if is_preview %}
    <div class="alert alert-warning mb-3">
      <strong>Preview:</strong> This listing is inactive and not visible to buyers.
    </div>
  {% endif %}
  <div class="row g-3">
    <div class="col-12 col-lg-7">
      <div class="card">
        {% if product.images.all %}
          <div id="productCarousel" class="carousel slide" data-bs-ride="carousel">
            <div class="carousel-inner">
              {% for img in product.images.all %}
                <div class="carousel-item {% if forloop.first %}active{% endif %}">
                  <img src="{{ img.image.url }}" class="d-block w-100" alt="{{ img.alt_text|default:product.title }}">
                </div>
              {% endfor %}
            </div>
            <button class="carousel-control-prev" type="button" data-bs-target="#productCarousel" data-bs-slide="prev">
              <span class="carousel-control-prev-icon" aria-hidden="true"></span>
              <span class="visually-hidden">Previous</span>
            </button>
            <button class="carousel-control-next" type="button" data-bs-target="#productCarousel" data-bs-slide="next">
              <span class="carousel-control-next-icon" aria-hidden="true"></span>
              <span class="visually-hidden">Next</span>
            </button>
          </div>

          {% if product.images.all|length > 1 %}
          <div class="card-body pt-2 pb-2">
            <div class="d-flex gap-2 flex-wrap">
              {% for img in product.images.all %}
                <button
                  type="button"
                  class="border rounded p-1 bg-white thumbnail-btn {% if forloop.first %}active{% endif %}"
                  data-bs-target="#productCarousel"
                  data-bs-slide-to="{{ forloop.counter0 }}"
                  style="width: 70px; height: 70px; cursor: pointer; opacity: {% if forloop.first %}1{% else %}0.6{% endif %};"
                  onclick="this.parentElement.querySelectorAll('.thumbnail-btn').forEach(b => b.style.opacity='0.6'); this.style.opacity='1';"
                >
                  <img src="{{ img.image.url }}" class="w-100 h-100 rounded" style="object-fit: cover;" alt="{{ img.alt_text|default:product.title }}">
                </button>
              {% endfor %}
            </div>
          </div>
          {% endif %}
        {% else %}
          <div class="bg-light d-flex align-items-center justify-content-center" style="height:360px;">
            <span class="text-secondary small">No images uploaded yet.</span>
          </div>
        {% endif %}
      </div>
    </div>

    <div class="col-12 col-lg-5">
      <div class="card">
        <div class="card-body">
          <div class="d-flex justify-content-between align-items-start gap-2">
            <div>
              <h1 class="h4 mb-1">{{ product.title }}</h1>
              <div class="text-secondary small">
                by <strong>{{ product.seller_public_name }}</strong> ·
                {% if product.category.parent %}
                  {{ product.category.parent.name }} › {{ product.category.name }}
                {% else %}
                  {{ product.category.name }}
                {% endif %}
              </div>
              <div class="mt-2">
                <a class="btn btn-sm btn-outline-secondary" href="{% url 'products:seller_shop' product.seller_id %}">
                  <i class="bi bi-shop me-1"></i>Visit seller store
                </a>
              </div>

              <div class="mt-1 small">
                {% if review_count and review_count > 0 %}
                  <span class="badge text-bg-dark me-1">{{ avg_rating|floatformat:1 }}/5</span>
                  <span class="text-secondary">{{ review_count }} review{{ review_count|pluralize }}</span>
                  <a class="ms-2 text-decoration-none" href="{% url 'reviews:product_reviews' product.id %}">View reviews</a>
                {% else %}
                  <span class="text-secondary">No product reviews yet.</span>
                {% endif %}
              </div>

              <div class="mt-1 small">
                {% if seller_review_count and seller_review_count > 0 %}
                  <span class="badge text-bg-secondary me-1">Seller {{ seller_avg_rating|floatformat:1 }}/5</span>
                  <span class="text-secondary">{{ seller_review_count }} seller rating{{ seller_review_count|pluralize }}</span>
                {% endif %}
              </div>
            </div>
            <span class="badge text-bg-secondary">{{ product.get_kind_display }}</span>
          </div>

          {% if product.complexity_level or product.print_time_hours %}
            <div class="mt-2 d-flex flex-wrap gap-2">
              {% if product.complexity_level %}
                <span class="badge text-bg-secondary">{{ product.get_complexity_level_display }}</span>
              {% endif %}
              {% if product.print_time_hours %}
                <span class="badge text-bg-secondary">
                  <i class="bi bi-clock me-1"></i>{{ product.print_time_hours }}h print time
                </span>
              {% endif %}
            </div>
          {% endif %}

          <div class="mt-3">
            <div class="h5 mb-2">{{ product.display_price }}</div>

            {# Keep Add-to-cart behavior for now, but your new free-download buttons make checkout unnecessary for free FILEs #}

            {% if can_buy %}
              <form method="post" action="{% url 'cart:add' %}" class="flex-grow-1">
                {% csrf_token %}
                <input type="hidden" name="product_id" value="{{ product.pk }}">
                <input type="hidden" name="next" value="{{ request.get_full_path }}">
                {% if product.kind == "MODEL" %}
                  <div class="mb-2">
                    <label class="form-label fw-semibold">Desired Color(s)</label>
          <input class="form-control mb-2" type="text" name="custom_colors" placeholder="e.g., Red, Blue,
Custom mix">
         </div>
        {% endif %}
        <div class="mb-2">
         <label class="form-label fw-semibold">Notes for Seller</label>
         <textarea class="form-control mb-2" name="buyer_notes" rows="2" placeholder="Special instructions,
requests, etc."></textarea>
        </div>
        <div class="mb-2">
         <label class="form-label fw-semibold">Quantity</label>
         <input class="form-control mb-2" type="number" name="quantity" min="1" value="1"
style="max-width:120px; display:inline-block; width:auto;"
           {% if product.max_purchases_per_buyer %}max="{{ product.max_purchases_per_buyer }}"{% endif
%} {% if product.kind == "FILE" %}readonly value="1"{% endif %}>
         {% if product.max_purchases_per_buyer %}
          <div class="small text-secondary">Limit: {{ product.max_purchases_per_buyer }} per buyer</div>
         {% endif %}
         {% if product.kind == "FILE" %}
          <div class="text-secondary small mt-2">Digital items are limited to 1 per cart.</div>
         {% endif %}
        </div>
        <div class="mb-2">
         <label class="form-label fw-semibold">Tip for Seller</label>
         <div class="input-group" style="max-width:180px;">
          <span class="input-group-text">$</span>
          <input class="form-control" type="number" name="tip_amount" min="0" step="0.01" value="0.00"
placeholder="0.00">
         </div>
         <div class="form-text">Optional. Show your appreciation with a tip.</div>
        </div>
        <button class="btn btn-dark flex-grow-1 mb-2" type="submit" style="width:100%">
         <i class="bi bi-cart-plus me-1"></i>Add to cart
        </button>
       </form>
      {% else %}
       <div class="d-flex gap-2">
        <button class="btn btn-dark flex-grow-1" type="button" disabled style="opacity:.6; cursor:not-allowed;
width:100%">
         <i class="bi bi-cart-plus me-1"></i>Add to cart
        </button>
        <button class="btn btn-outline-primary flex-grow-1" type="button" disabled style="opacity:.6;
cursor:not-allowed; width:100%">
         <i class="bi bi-cash-coin me-1"></i>Tip this seller
        </button>
       </div>
       {% if is_preview %}
        <div class="small mt-2" style="color:#b45309;">This listing is inactive.</div>
       {% else %}
        <div class="small mt-2" style="color:#b45309;">Seller not ready (Stripe onboarding incomplete).</div>
       {% endif %}
      {% endif %}

      <div class="mt-2">
       <a class="btn btn-outline-dark w-100" href="{% url 'cart:detail' %}">
        <i class="bi bi-bag me-1"></i>View cart
       </a>
      </div>
     </div>

     {% if product.short_description %}
```

```
            <hr>
            <div class="text-secondary">{{ product.short_description }}</div>
          {% endif %}

          {% if product.description %}
            <hr>
            <div style="white-space: pre-wrap;">{{ product.description }}</div>
          {% endif %}

          <hr>
          <div class="fw-semibold mb-2">What you get</div>
          {% if product.kind == "FILE" %}
            <ul class="small text-secondary mb-2">
                 <li>{% if product.is_free %}Instant free download.{% else %}Instant digital download after purchase.{%
endif %}</li>
              {% if product.file_types_display %}
                <li>File formats: {{ product.file_types_display }}</li>
              {% endif %}
              {% if product.digital and product.digital.license_type %}
                <li>License: {{ product.digital.get_license_type_display }}</li>
              {% endif %}
            </ul>
            {% if product.digital and product.digital.license_text %}
              <div class="text-secondary small" style="white-space: pre-wrap;">{{ product.digital.license_text }}</div>
            {% endif %}
          {% else %}
            <ul class="small text-secondary mb-2">
              <li>One physical 3D printed item per quantity.</li>
              <li>Made to order by the seller.</li>
            </ul>
          {% endif %}

          {% if product.kind == "FILE" %}
            <hr>
            <div class="fw-semibold mb-2">Files</div>

            {% if product.digital_assets.all %}
              <div class="table-responsive">
                <table class="table table-sm align-middle mb-0">
                  <thead class="table-light">
                    <tr>
                      <th>File</th>
                      <th style="width:110px;">Type</th>
                      <th class="text-center" style="width:140px;">Downloads</th>
                      <th class="text-end" style="width:190px;">Action</th>
                    </tr>
                  </thead>
                  <tbody>
                    {% for a in product.digital_assets.all %}
                      <tr>
                        <td>
                          <div class="fw-semibold">{{ a.original_filename|default:a.file.name }}</div>
                          {% if a.file_type == "zip" and a.zip_contents %}
                            <div class="text-secondary small mt-1">
                              <div class="text-muted">Includes:</div>
                              <ul class="mb-0">
                                {% for item in a.zip_contents|slice:":8" %}
                                  <li>{{ item }}</li>
                                {% endfor %}
                                {% if a.zip_contents|length > 8 %}
                                  <li>+ {{ a.zip_contents|length|add:"-8" }} more…</li>
                                {% endif %}
                              </ul>
                            </div>
                          {% endif %}
                        </td>

                        <td class="text-muted">
                          {% if a.file_type %}.{{ a.file_type }}{% else %}—{% endif %}
                        </td>

                        <td class="text-center">
                          <span class="badge text-bg-light text-dark">{{ a.download_count|default:0 }}</span>
                        </td>

                        <td class="text-end">
                          {% if product.is_free and product.is_active %}
                            <a class="btn btn-sm btn-dark"
                               href="{% url 'products:free_asset_download' product.pk product.slug a.pk %}">
                              <i class="bi bi-download me-1"></i>Download
                            </a>
                          {% else %}
                            <button class="btn btn-sm btn-outline-secondary" type="button" disabled style="opacity:.7;">
                              <i class="bi bi-lock me-1"></i>Locked
                            </button>
                          {% endif %}
                        </td>
                      </tr>
                    {% endfor %}
                  </tbody>
                </table>
              </div>

              {% if not product.is_free %}
                <div class="text-secondary small mt-2">
                  <i class="bi bi-info-circle me-1"></i>Downloads unlock when this item is free (or later: after purchase).
                </div>
              {% endif %}
            {% else %}
              <div class="text-secondary small">No files uploaded yet.</div>
            {% endif %}
          {% endif %}

          {# Recommended Filament #}
          {% if product.kind == "FILE" and filament_recommendations %}
            <hr>
            <div class="fw-semibold mb-2">Recommended Filament</div>
            <div class="small text-secondary mb-2">
              Optional links to help you print this file successfully.
            </div>

            <div class="list-group list-group-flush">
              {% for f in filament_recommendations %}
                <div class="list-group-item px-0">
                  <div class="d-flex justify-content-between align-items-start gap-2">
                    <div>
                      <div class="fw-semibold">
                        {{ f.get_material_display }}
                        {% if f.brand %}<span class="text-muted"> · {{ f.brand }}</span>{% endif %}
                      </div>
                      {% if f.notes %}
                        <div class="text-secondary small" style="white-space: pre-wrap;">{{ f.notes }}</div>
```

```
                      {% endif %}
                    </div>
                    <a href="{{ f.url }}"
                       class="btn btn-sm btn-outline-dark"
                       target="_blank"
                       rel="noopener noreferrer nofollow">
                      Buy filament ↗
                    </a>
                  </div>
                </div>
              {% endfor %}
            </div>
          {% endif %}

          {% if product.kind == "MODEL" and product.physical %}
            <hr>
            <div class="fw-semibold mb-3">Specifications</div>
            <div class="row g-3 text-secondary small">
              {% if product.physical.material %}
              <div class="col-6">
                <div class="text-muted">Material</div>
                <div class="fw-semibold text-dark">{{ product.physical.material }}</div>
              </div>
              {% endif %}

              {% if product.physical.color %}
              <div class="col-6">
                <div class="text-muted">Color</div>
                <div class="fw-semibold text-dark">{{ product.physical.color }}</div>
              </div>
              {% endif %}

              {% if product.physical.num_colors %}
              <div class="col-6">
                <div class="text-muted">Number of Colors</div>
                <div class="fw-semibold text-dark">{{ product.physical.num_colors }}</div>
              </div>
              {% endif %}

              {% if product.physical.width_mm or product.physical.height_mm or product.physical.depth_mm %}
              <div class="col-6">
                <div class="text-muted">Dimensions (mm)</div>
                <div class="fw-semibold text-dark">
                  {% if product.physical.width_mm %}W:{{ product.physical.width_mm }}{% endif %}
                  {% if product.physical.height_mm %}{% if product.physical.width_mm %} × {% endif %}H:{{ product.physical.height_mm }}{% endif %}
                  {% if product.physical.depth_mm %}{% if product.physical.width_mm or product.physical.height_mm %} × {% endif %}D:{{ product.physical.depth_mm }}{% endif %}
                </div>
              </div>
              {% endif %}

              {% if product.physical.weight_grams %}
              <div class="col-6">
                <div class="text-muted">Estimated Weight</div>
                <div class="fw-semibold text-dark">{{ product.physical.weight_grams }}g</div>
              </div>
              {% endif %}

              {% if product.physical.support_required %}
              <div class="col-6">
                <div class="text-muted">Support Structures</div>
                <div class="fw-semibold text-dark">
                  <span class="badge text-bg-warning">Required</span>
                </div>
              </div>
              {% endif %}
            </div>

            {% if product.physical.specifications %}
            <hr>
            <div class="fw-semibold mb-2">Additional Details</div>
            <div style="white-space: pre-wrap;" class="text-secondary small">{{ product.physical.specifications }}</div>
            {% endif %}
          {% endif %}

          {% if product.kind == "FILE" and product.digital %}
            <hr>
            <div class="fw-semibold mb-3">File Specifications</div>
            <div class="row g-3 text-secondary small">
              {% if product.digital.software_requirements %}
              <div class="col-12">
                <div class="text-muted">Software Required</div>
                <div class="fw-semibold text-dark">{{ product.digital.software_requirements }}</div>
              </div>
              {% endif %}

              {% if product.digital.compatible_software %}
              <div class="col-12">
                <div class="text-muted">Compatible With</div>
                <div class="fw-semibold text-dark">{{ product.digital.compatible_software }}</div>
              </div>
              {% endif %}

              {% if product.digital.license_type %}
              <div class="col-6">
                <div class="text-muted">License Type</div>
                <div class="fw-semibold text-dark">
                  <span class="badge text-bg-info">{{ product.digital.get_license_type_display }}</span>
                </div>
              </div>
              {% endif %}
            </div>

            {% if product.digital.requirements %}
            <hr>
            <div class="fw-semibold mb-2">Requirements & Details</div>
            <div style="white-space: pre-wrap;" class="text-secondary small">{{ product.digital.requirements }}</div>
            {% endif %}
          {% endif %}

          <hr>

          <!-- Tabs: Q&A + Reviews -->
          <ul class="nav nav-tabs" id="productTabs" role="tablist">
            <li class="nav-item" role="presentation">
              <button class="nav-link active" id="qa-tab" data-bs-toggle="tab" data-bs-target="#qa-pane" type="button" role="tab" aria-controls="qa-pane" aria-selected="true">
                Q&amp;A <span class="badge text-bg-light text-dark ms-1">{{ qa_thread_count }}</span>
              </button>
            </li>
```

```
            <li class="nav-item" role="presentation">
              <button class="nav-link" id="reviews-tab" data-bs-toggle="tab" data-bs-target="#reviews-pane" type="button" role="tab" aria-controls="reviews-pane" aria-selected="false">
                Reviews <span class="badge text-bg-light text-dark ms-1">{{ review_count }}</span>
              </button>
            </li>
          </ul>

          <div class="tab-content pt-3" id="productTabsContent">
            <div class="tab-pane fade show active" id="qa-pane" role="tabpanel" aria-labelledby="qa-tab" tabindex="0">
              {% include "qa/_products_tab.html" %}
            </div>

            <div class="tab-pane fade" id="reviews-pane" role="tabpanel" aria-labelledby="reviews-tab" tabindex="0">
              <div class="d-flex align-items-center justify-content-between">
                <div class="fw-semibold">Recent reviews</div>
                <a class="text-decoration-none" href="{% url 'reviews:product_reviews' product.id %}">All reviews</a>
              </div>

              <div class="mt-2">
                {% for r in recent_reviews %}
                  <div class="border rounded p-2 mb-2">
                    <div class="d-flex justify-content-between align-items-center">
                      <div class="small fw-semibold">{{ r.buyer.username }}</div>
                      <div class="small text-secondary">{{ r.created_at|date:"M j, Y" }}</div>
                    </div>
                    <div class="mt-1">
                      <span class="badge text-bg-dark">{{ r.rating }}/5</span>
                      {% if r.title %}
                        <span class="ms-2 fw-semibold">{{ r.title }}</span>
                      {% endif %}
                    </div>
                    {% if r.body %}
                      <div class="small mt-2">{{ r.body|linebreaksbr }}</div>
                    {% endif %}
                  </div>
                {% empty %}
                  <div class="text-secondary small">No reviews yet.</div>
                {% endfor %}
              </div>
            </div>
          </div>

        </div>
      </div>
    </div>
  </div>

  {% if related_same_seller or related_same_category %}
    <div class="mt-4">
      {% if related_same_seller %}
        <div class="d-flex align-items-center justify-content-between mb-2">
          <h2 class="h5 mb-0">More from this seller</h2>
          <a class="text-decoration-none" href="{% url 'products:seller_shop' product.seller_id %}">View seller shop</a>
        </div>
        <div class="row g-3">
          {% for p in related_same_seller %}
            <div class="col-12 col-sm-6 col-lg-3">
              {% include "products/_product_card.html" with p=p show_add_to_cart=True show_category=True
```

```
image_height="160px" %}
            </div>
          {% endfor %}
        </div>
      {% endif %}

      {% if related_same_category %}
        <div class="d-flex align-items-center justify-content-between mb-2 mt-4">
          <h2 class="h5 mb-0">More in this category</h2>
        </div>
        <div class="row g-3">
          {% for p in related_same_category %}
            <div class="col-12 col-sm-6 col-lg-3">
                          {%  include  "products/_product_card.html"  with  p=p  show_add_to_cart=True  show_category=True
image_height="160px" %}
            </div>
          {% endfor %}
        </div>
      {% endif %}
    </div>
  {% endif %}
</div>
{% endblock %}
```

products\templates\products\seller\product_form.html

```
{# templates/products/seller/product_form.html #}
{% extends "base.html" %}
{% load static %}

{% block title %}Seller · {% if mode == "create" %}New Listing{% else %}Edit Listing{% endif %} · Home Craft 3D{% endblock %}

{% block content %}
<div class="container-fluid" style="max-width: 980px;">
  <div class="d-flex align-items-center justify-content-between mb-3">
    <div>
      <h1 class="h4 mb-1">{% if mode == "create" %}New Listing{% else %}Edit Listing{% endif %}</h1>
      <div class="text-secondary small">Fill in the details for your listing.</div>
    </div>
    <a class="btn btn-outline-dark" href="{% url 'products:seller_list' %}">
      <i class="bi bi-arrow-left me-1"></i>Back
    </a>
  </div>

  <div class="card">
    <div class="card-body">
      <form method="post" novalidate>
        {% csrf_token %}

        {% if form.non_field_errors %}
          <div class="alert alert-danger mb-3">
            {{ form.non_field_errors }}
          </div>
        {% endif %}

        <!-- Basic Information -->
        <h5 class="border-bottom pb-2 mb-3">Basic Information</h5>
        <div class="row g-3 mb-4">

          <div class="col-12">
            <label class="form-label fw-semibold">Kind <span class="text-danger">*</span></label>
            {{ form.kind }}
            {% if form.kind.errors %}<div class="text-danger small mt-1">{{ form.kind.errors }}</div>{% endif %}
            <div class="form-text">File (digital download) or Model (physical print).</div>
          </div>

          <div class="col-12">
            <label class="form-label fw-semibold">Category <span class="text-danger">*</span></label>
            <select name="category" id="id_category" class="form-select">
              <option value="">Loading categories...</option>
            </select>
            {% if form.category.errors %}<div class="text-danger small mt-1">{{ form.category.errors }}</div>{% endif %}
            <div class="form-text">Choose a main category (filtered by Kind).</div>
          </div>

          <div class="col-12">
            <label class="form-label fw-semibold">Subcategory</label>
            <select name="subcategory" id="id_subcategory" class="form-select" disabled>
              <option value="">Select a category first...</option>
            </select>
            {% if form.subcategory.errors %}<div class="text-danger small mt-1">{{ form.subcategory.errors }}</div>{% endif %}
            <div class="form-text">Locked until you pick a Category.</div>
          </div>

          <div class="col-12">
            <label class="form-label fw-semibold">Title <span class="text-danger">*</span></label>
            {{ form.title }}
            {% if form.title.errors %}<div class="text-danger small mt-1">{{ form.title.errors }}</div>{% endif %}
            <div class="form-text">Give your listing a clear, descriptive title.</div>
          </div>

          <div class="col-12">
            <label class="form-label fw-semibold">Short Description</label>
            {{ form.short_description }}
             {% if form.short_description.errors %}<div class="text-danger small mt-1">{{ form.short_description.errors
}}</div>{% endif %}
            <div class="form-text">A brief summary shown in search results and cards.</div>
          </div>

          <div class="col-12">
            <label class="form-label fw-semibold">Full Description</label>
            {{ form.description }}
            {% if form.description.errors %}<div class="text-danger small mt-1">{{ form.description.errors }}</div>{%
endif %}
            <div class="form-text">Detailed information about your product.</div>
          </div>
        </div>

        <!-- Pricing -->
        <h5 class="border-bottom pb-2 mb-3">Pricing</h5>
        <div class="row g-3 mb-4">
          <div class="col-12 col-md-6">
            <label class="form-label fw-semibold">Price <span class="text-danger">*</span></label>
            <div class="input-group">
              <span class="input-group-text">$</span>
              {{ form.price }}
            </div>
            {% if form.price.errors %}<div class="text-danger small mt-1">{{ form.price.errors }}</div>{% endif %}
            <div class="form-text">Set to $0.00 if offering for free.</div>
          </div>

          <div class="col-12 col-md-6">
            <div class="form-check mt-4">
              {{ form.is_free }}
              <label class="form-check-label fw-semibold" for="{{ form.is_free.id_for_label }}">
                <i class="bi bi-gift me-1"></i>Free item
              </label>
            </div>
               {% if form.is_free.errors %}<div class="text-danger small mt-1">{{ form.is_free.errors }}</div>{% endif
%}
            <div class="form-text ms-4">Check this if offering your product for free.</div>
          </div>
        </div>

        <!-- Visibility Settings -->
        <h5 class="border-bottom pb-2 mb-3">Visibility & Settings</h5>
        <div class="row g-3 mb-4">
          <div class="col-12">
            <div class="form-check">
              {{ form.is_active }}
              <label class="form-check-label fw-semibold" for="{{ form.is_active.id_for_label }}">
                <i class="bi bi-eye me-1"></i>Active (visible)
              </label>
            </div>
            {% if form.is_active.errors %}<div class="text-danger small mt-1">{{ form.is_active.errors }}</div>{%
endif %}
            <div class="form-text">Make this listing visible to buyers.</div>
          </div>
          <div class="col-12 col-md-6">
            <label class="form-label fw-semibold">Max units per buyer</label>
            {{ form.max_purchases_per_buyer }}
            {% if form.max_purchases_per_buyer.errors %}<div class="text-danger small mt-1">{{
form.max_purchases_per_buyer.errors }}</div>{% endif %}
            <div class="form-text">Limit how many units a single buyer can purchase. Leave blank or set to 0 for no
limit.</div>
          </div>
        </div>

        <!-- Advanced Settings -->
        <div class="accordion mb-3" id="advancedSettings">
          <div class="accordion-item">
            <h2 class="accordion-header">
              <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse"
data-bs-target="#collapseAdvanced">
                <i class="bi bi-gear me-2"></i>Advanced Settings (Optional)
              </button>
            </h2>
            <div id="collapseAdvanced" class="accordion-collapse collapse" data-bs-parent="#advancedSettings">
              <div class="accordion-body">
                <div class="row g-3">
                  <div class="col-12">
                    <label class="form-label fw-semibold">URL Slug</label>
                    {{ form.slug }}
                    {% if form.slug.errors %}<div class="text-danger small mt-1">{{ form.slug.errors }}</div>{% endif
%}
                    <div class="form-text">
                      <i class="bi bi-info-circle me-1"></i>
                      <strong>For advanced users:</strong> Customize the URL for this product. Leave blank to
auto-generate from the title.
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>

        <!-- Action Buttons -->
        <div class="d-flex gap-2 pt-3 border-top">
          <button class="btn btn-dark btn-lg" type="submit">
            <i class="bi bi-save me-1"></i>Save Listing
          </button>

          {% if mode == "edit" %}
            <a class="btn btn-outline-dark" href="{% url 'products:seller_images' product.pk %}">
              <i class="bi bi-images me-1"></i>Manage Images
            </a>
            <a class="btn btn-outline-dark" href="{% url 'products:seller_specs' product.pk %}">
              <i class="bi bi-rulers me-1"></i>Manage Specs
            </a>
            {% if product.kind == "FILE" %}
              <a class="btn btn-outline-dark" href="{% url 'products:seller_assets' product.pk %}">
                <i class="bi bi-paperclip me-1"></i>Manage Assets
              </a>
            {% endif %}
          {% endif %}

          <a class="btn btn-outline-secondary ms-auto" href="{% url 'products:seller_list' %}">
            Cancel
          </a>
        </div>
      </form>
    </div>
  </div>

  {% if messages %}
    <div class="mt-3">
      {% for message in messages %}
        <div class="alert alert-info mb-2">{{ message }}</div>
      {% endfor %}
    </div>
  {% endif %}
</div>

<script>
document.addEventListener("DOMContentLoaded", function () {
  const kindSelect = document.getElementById("id_kind");
  const categorySelect = document.getElementById("id_category");
  const subcategorySelect = document.getElementById("id_subcategory");

  if (!kindSelect || !categorySelect || !subcategorySelect) return;

  const CATEGORIES_URL = "{% url 'catalog:api_categories' %}";
  const SUBCATEGORIES_URL = "{% url 'catalog:api_subcategories' %}";

  // Preserve selection on edit/postback:
  // Prefer POSTed values, then initial, then blank.
  const initialKind = (kindSelect.value || "").trim() || "FILE";
  const initialCategory =
    ("{{ form.data.category|default_if_none:'' }}".trim()) ||
    ("{{ form.initial.category|default_if_none:'' }}".trim()) ||
    "";
  const initialSubcategory =
    ("{{ form.data.subcategory|default_if_none:'' }}".trim()) ||
    ("{{ form.initial.subcategory|default_if_none:'' }}".trim()) ||
    "";

  function setCategoryOptions(items) {
    categorySelect.innerHTML = "";
    const ph = document.createElement("option");
    ph.value = "";
    ph.textContent = "Select a category…";
    categorySelect.appendChild(ph);

    for (const c of items) {
      const opt = document.createElement("option");
      opt.value = String(c.id);
      opt.textContent = c.name;
      categorySelect.appendChild(opt);
    }
  }

  function setSubcategoryOptions(items) {
    subcategorySelect.innerHTML = "";
    const ph = document.createElement("option");
    ph.value = "";
    ph.textContent = items.length ? "Select a subcategory…" : "No subcategories";
    subcategorySelect.appendChild(ph);

    for (const s of items) {
      const opt = document.createElement("option");
      opt.value = String(s.id);
      opt.textContent = s.name;
      subcategorySelect.appendChild(opt);
    }

    subcategorySelect.disabled = (items.length === 0);
  }

  async function fetchJSON(url) {
    const resp = await fetch(url, {
      credentials: "same-origin",
      headers: { "Accept": "application/json" }
    });
    if (!resp.ok) return null;
    try { return await resp.json(); } catch (e) { return null; }
  }

  async function loadCategories(kind, desiredCategoryId) {
    // Reset
    categorySelect.innerHTML = '<option value="">Loading categories…</option>';
    subcategorySelect.innerHTML = '<option value="">Select a category first…</option>';
    subcategorySelect.disabled = true;

    const data = await fetchJSON(CATEGORIES_URL + "?kind=" + encodeURIComponent(kind));
    const cats = (data && data.categories) ? data.categories : [];

    setCategoryOptions(cats);

    // Determine selection:
    // 1) desiredCategoryId (edit/postback)
    // 2) first category
    // 3) blank
    let selectId = (desiredCategoryId || "").trim();
    if (selectId && !Array.from(categorySelect.options).some(o => o.value === selectId)) {
      selectId = "";
    }
    if (!selectId && cats.length) {
      selectId = String(cats[0].id);
    }

    categorySelect.value = selectId;

    // Load subcategories for selected category (if any)
    await loadSubcategories(categorySelect.value, initialSubcategory);
  }

  async function loadSubcategories(categoryId, desiredSubId) {
    if (!categoryId) {
      setSubcategoryOptions([]);
      return;
    }

    const data = await fetchJSON(SUBCATEGORIES_URL + "?category_id=" + encodeURIComponent(categoryId));
    const subs = (data && data.subcategories) ? data.subcategories : [];
```

```
    setSubcategoryOptions(subs);

    // Determine selection:
    // 1) desiredSubId (edit/postback)
    // 2) first subcategory
    // 3) blank
    let subId = (desiredSubId || "").trim();
    if (subId && !Array.from(subcategorySelect.options).some(o => o.value === subId)) {
      subId = "";
    }
    if (!subId && subs.length) {
      subId = String(subs[0].id);
    }

    subcategorySelect.value = subId;
  }

  // Events
  kindSelect.addEventListener("change", async function () {
    const kind = (kindSelect.value || "FILE").trim().toUpperCase();
    await loadCategories(kind, "");
  });

  categorySelect.addEventListener("change", async function () {
    await loadSubcategories((categorySelect.value || "").trim(), "");
  });

  // Init: default to FILE on create
  kindSelect.value = (initialKind || "FILE").trim().toUpperCase();
  loadCategories(kindSelect.value, initialCategory);
});
</script>
{% endblock %}
```

products\templates\products\seller\product_images.html

```
{# products/templates/products/seller/product_images.html #}
{% extends "base.html" %}
{% load static %}
{% block title %}Images · {{ product.title }} · Home Craft 3D{% endblock %}

{% block content %}
<div class="container py-4" style="max-width: 1100px;">
  {% if messages %}
    <div class="mb-3">
      {% for message in messages %}
        <div class="alert alert-{{ message.tags }} mb-2">{{ message }}</div>
      {% endfor %}
    </div>
  {% endif %}

  <div class="d-flex flex-wrap align-items-start justify-content-between gap-2 mb-3">
    <div>
      <h1 class="h4 mb-1">Images</h1>
      <div class="text-secondary small">{{ product.title }}</div>
      <div class="text-secondary small">Upload images one at a time.</div>
    </div>

    <div class="d-flex flex-wrap gap-2">
      <a class="btn btn-outline-dark btn-sm" href="{% url 'products:seller_edit' product.pk %}">
        <i class="bi bi-pencil me-1"></i>Edit listing
      </a>
      <a class="btn btn-outline-dark btn-sm" href="{% url 'products:seller_specs' product.pk %}">
        <i class="bi bi-rulers me-1"></i>Manage specs
      </a>
      {% if product.kind == "FILE" %}
        <a class="btn btn-outline-dark btn-sm" href="{% url 'products:seller_assets' product.pk %}">
          <i class="bi bi-folder2-open me-1"></i>Digital assets
        </a>
      {% endif %}
      <a class="btn btn-outline-dark btn-sm" href="{% url 'products:seller_list' %}">
        <i class="bi bi-arrow-left me-1"></i>Back to listings
      </a>
    </div>
  </div>

  <div class="row g-3">
    <div class="col-12 col-lg-6">
      <div class="card shadow-sm">
        <div class="card-body">
          <div class="d-flex align-items-center justify-content-between mb-2">
            <div class="fw-semibold">Upload image</div>
            <div class="text-secondary small">Add images one at a time</div>
          </div>

          <form method="post" enctype="multipart/form-data" novalidate>
            {% csrf_token %}

            {% if form.non_field_errors %}
              <div class="alert alert-danger mb-2">
                {{ form.non_field_errors }}
              </div>
            {% endif %}

            <div class="mb-2">
```

```
              {{ form.image.label_tag }}
              {{ form.image }}
              {% for error in form.image.errors %}
                <div class="text-danger small">{{ error }}</div>
              {% endfor %}
            </div>

            <div class="mb-2">
              {{ form.alt_text.label_tag }}
              {{ form.alt_text }}
              {% for error in form.alt_text.errors %}
                <div class="text-danger small">{{ error }}</div>
              {% endfor %}
            </div>

            <div class="row g-2">
              <div class="col-6">
                {{ form.sort_order.label_tag }}
                {{ form.sort_order }}
                {% for error in form.sort_order.errors %}
                  <div class="text-danger small">{{ error }}</div>
                {% endfor %}
              </div>
              <div class="col-6 d-flex align-items-end">
                <div class="form-check">
                  {{ form.is_primary }}
                  <label class="form-check-label" for="{{ form.is_primary.id_for_label }}">Primary image</label>
                </div>
              </div>
            </div>

            <div class="mt-3">
              <button class="btn btn-outline-dark" type="submit" name="single_upload" value="1">
                <i class="bi bi-plus-circle me-1"></i>Add image
              </button>
            </div>
          </form>

        </div>
      </div>
    </div>

    <div class="col-12 col-lg-6">
      <div class="card shadow-sm">
        <div class="card-body">
          <div class="d-flex align-items-center justify-content-between">
            <div class="fw-semibold">Current images</div>
            <div class="text-secondary small">{{ images|length }} total</div>
          </div>

          {% if images %}
            <div class="row g-2 mt-2">
              {% for img in images %}
                <div class="col-6 col-md-4">
                  <div class="border rounded p-2 h-100">
                    <div class="ratio ratio-4x3 bg-light rounded overflow-hidden">
                      <img src="{{ img.image.url }}" alt="{{ img.alt_text|default:product.title }}" style="object-fit: contain; width: 100%; height: 100%; background: #f8f9fa;">
                    </div>

                    <div class="d-flex justify-content-between align-items-center mt-2">
```

```
           <div class="small">
            {% if img.is_primary %}
             <span class="badge text-bg-dark">Primary</span>
            {% else %}
             <span class="badge text-bg-light text-dark">Image</span>
            {% endif %}
            <span class="text-secondary">#{{ img.sort_order }}</span>
           </div>

           <form method="post" action="{% url 'products:seller_image_delete' product.pk img.pk %}">
            {% csrf_token %}
            <button class="btn btn-sm btn-outline-danger" type="submit" title="Delete" aria-label="Delete
image {{ forloop.counter }}">
             <i class="bi bi-trash"></i>
            </button>
           </form>
          </div>

          <form class="mt-2" method="post" action="{% url 'products:seller_image_update' product.pk img.pk
%}">
           {% csrf_token %}
           <div class="input-group input-group-sm">
            <span class="input-group-text">Order</span>
            <input type="number" class="form-control" name="sort_order" value="{{ img.sort_order }}"
min="0">
            <button class="btn btn-outline-dark" type="submit" name="action"
value="save_order">Save</button>
           </div>
           <div class="mt-2 d-grid">
            {% if img.is_primary %}
            <button class="btn btn-sm btn-outline-secondary" type="button" disabled>Primary
image</button>
            {% else %}
            <button class="btn btn-sm btn-outline-dark" type="submit" name="action"
value="make_primary">Make primary</button>
            {% endif %}
           </div>
          </form>

          {% if img.alt_text %}
           <div class="text-secondary small mt-1">{{ img.alt_text|truncatechars:42 }}</div>
          {% endif %}
         </div>
        </div>
       {% endfor %}
      </div>
     {% else %}
      <div class="alert alert-secondary mt-3 mb-0">
       No images uploaded yet.
      </div>
     {% endif %}
    </div>
   </div>
  </div>
 </div>
</div>
{% endblock %}
```

products\templates\products\seller\product_list.html

```
{# products/templates/products/seller/product_list.html #}
{% extends "base.html" %}
{% load static %}
{% block title %}Seller · Listings · Home Craft 3D{% endblock %}

{% block content %}
<div class="container-fluid">
  <div class="d-flex flex-wrap align-items-center justify-content-between gap-2 mb-3">
    <div>
      <div class="d-flex align-items-center gap-2">
        <h1 class="h4 mb-0">Seller Listings</h1>

        {% if stripe_ready %}
          <span class="badge text-bg-success">
            <i class="bi bi-check2-circle me-1"></i>Stripe: Ready
          </span>
        {% else %}
          <span class="badge text-bg-warning text-dark">
            <i class="bi bi-exclamation-triangle me-1"></i>Stripe: Action Required
          </span>
        {% endif %}
      </div>

      <div class="text-secondary small">Create, edit, and manage your listings.</div>
      {% if stripe_account and stripe_account.stripe_account_id %}
        <div class="text-secondary small mt-1">
          Connect ID: <span class="font-monospace">{{ stripe_account.stripe_account_id }}</span>
        </div>
      {% endif %}
    </div>

    <div class="d-flex gap-2">
      <form class="d-flex gap-2" method="get">
                <input class="form-control" name="q" value="{{ q }}" placeholder="Search your listings..."
style="min-width:260px;">
        <select class="form-select" name="file_type" style="min-width:170px;">
          <option value="">All file types</option>
          {% for t in file_type_options %}
            <option value="{{ t }}" {% if file_type == t %}selected{% endif %}>.{{ t }}</option>
          {% endfor %}
        </select>
        <button class="btn btn-outline-dark" type="submit">Search</button>
      </form>

      <a class="btn btn-outline-dark" href="{% url 'payments:connect_status' %}">
        <i class="bi bi-credit-card-2-front me-1"></i>Stripe Payouts
      </a>

      {% if stripe_ready %}
        <a class="btn btn-dark" href="{% url 'products:seller_create' %}">
          <i class="bi bi-plus-circle me-1"></i>New Listing
        </a>
      {% else %}
        <a class="btn btn-dark disabled" href="#" tabindex="-1" aria-disabled="true">
          <i class="bi bi-plus-circle me-1"></i>New Listing
        </a>
      {% endif %}
    </div>
  </div>
  {% if not stripe_ready %}
    <div class="alert alert-warning d-flex flex-wrap align-items-center justify-content-between gap-2">
      <div>
        <div class="fw-semibold">Finish Stripe setup to sell</div>
        <div class="small">
          You can browse your listings, but uploading, editing, activating, and checkout require Stripe payouts enabled.
        </div>
      </div>
      <a class="btn btn-dark" href="{% url 'payments:connect_status' %}">
        <i class="bi bi-arrow-right-circle me-1"></i>Finish Stripe setup
      </a>
    </div>
  {% endif %}

  {% if messages %}
    <div class="mb-3">
      {% for message in messages %}
        <div class="alert alert-info mb-2">{{ message }}</div>
      {% endfor %}
    </div>
  {% endif %}

  <div class="card">
    <div class="table-responsive">
      <table class="table table-hover align-middle mb-0">
        <thead class="table-light">
          <tr>
            <th>Title</th>
            <th>Kind</th>
            <th>Category</th>
            <th>Price</th>
            <th>Status</th>
            <th>Units Sold</th>
            <th>Downloads</th>
            <th class="text-end">Actions</th>
          </tr>
        </thead>
        <tbody>
          {% for p in products %}
            <tr>
              <td class="fw-semibold">
                <div class="d-flex align-items-center gap-2">
                  {% if p.primary_image %}
                    <img src="{{ p.primary_image.image.url }}" alt="{{ p.title }}" class="rounded border"
style="width:48px;height:48px;object-fit:cover;">
                  {% else %}
                    <div class="rounded border bg-light d-flex align-items-center justify-content-center"
style="width:48px;height:48px;">
                      <span class="text-secondary small">No image</span>
                    </div>
                  {% endif %}
                  <div>
                    <a class="text-decoration-none" href="{{ p.get_absolute_url }}">{{ p.title }}</a>
                    <div class="text-secondary small">
                      by {{ p.seller.username }}
                    </div>
                    {% if p.is_active and not p.has_specs %}
                      <div class="small text-warning">
                        <i class="bi bi-exclamation-triangle me-1"></i>Specs missing
                      </div>
                    {% endif %}
                  </div>
                </div>
              </td>
              <td><span class="badge text-bg-secondary">{{ p.get_kind_display }}</span></td>
              <td>
                {% if p.category.parent %}
                  <div class="small">{{ p.category.parent.name }}</div>
                  <div class="text-secondary small">› {{ p.category.name }}</div>
                {% else %}
                  {{ p.category.name }}
                {% endif %}
              </td>
              <td><strong>{{ p.display_price }}</strong></td>
              <td>
                {% if p.is_active %}
                  <span class="badge text-bg-success">Active</span>
                {% else %}
                  <span class="badge text-bg-secondary">Inactive</span>
                {% endif %}
              </td>
              <td>
                {{ p.units_sold|default:0 }}
              </td>
              <td>
                {{ p.total_downloads }}
              </td>
              <td class="text-end">
                <div class="btn-group">
                  <button class="btn btn-sm btn-outline-dark dropdown-toggle" type="button" data-bs-toggle="dropdown" aria-expanded="false">
                    Actions
                  </button>
                  <ul class="dropdown-menu dropdown-menu-end">
                    <li>
                      <a class="dropdown-item" href="{% url 'products:seller_preview' p.pk %}"><i class="bi bi-box-arrow-up-right me-1"></i>Preview</a>
                    </li>
                    {% if stripe_ready %}
                      <li><a class="dropdown-item" href="{% url 'products:seller_edit' p.pk %}"><i class="bi bi-pencil me-1"></i>Edit</a></li>
                      <li><a class="dropdown-item" href="{% url 'products:seller_images' p.pk %}"><i class="bi bi-images me-1"></i>Images</a></li>
                      <li><a class="dropdown-item" href="{% url 'products:seller_specs' p.pk %}"><i class="bi bi-rulers me-1"></i>Specs</a></li>
                      {% if p.kind == "FILE" %}
                        <li><a class="dropdown-item" href="{% url 'products:seller_assets' p.pk %}"><i class="bi bi-paperclip me-1"></i>Assets</a></li>
                      {% endif %}
                      <li>
                        <form method="post" action="{% url 'products:seller_duplicate' p.pk %}" class="d-inline">
                          {% csrf_token %}
                          <button class="dropdown-item" type="submit"><i class="bi bi-files me-1"></i>Copy</button>
                        </form>
                      </li>
                      <li>
                        <form method="post" action="{% url 'products:seller_toggle_active' p.pk %}" class="d-inline">
                          {% csrf_token %}
                          <button class="dropdown-item" type="submit">
                            {% if p.is_active %}
                              <i class="bi bi-eye-slash me-1"></i>Deactivate
                            {% else %}
                              <i class="bi bi-eye me-1"></i>Activate
                            {% endif %}
                          </button>
                        </form>
                      </li>
                    {% else %}
                      <li><a class="dropdown-item disabled" href="#" tabindex="-1" aria-disabled="true"><i class="bi bi-pencil me-1"></i>Edit</a></li>
                      <li><a class="dropdown-item disabled" href="#" tabindex="-1" aria-disabled="true"><i class="bi bi-images me-1"></i>Images</a></li>
                      {% if p.kind == "FILE" %}
                        <li><a class="dropdown-item disabled" href="#" tabindex="-1" aria-disabled="true"><i class="bi bi-paperclip me-1"></i>Assets</a></li>
                      {% endif %}
                      <li><a class="dropdown-item" href="{% url 'payments:connect_status' %}"><i class="bi bi-credit-card-2-front me-1"></i>Finish Stripe</a></li>
                    {% endif %}
                  </ul>
                  <button class="btn btn-sm btn-outline-danger" type="button" data-bs-toggle="modal" data-bs-target="#deleteModal{{ p.pk }}">
                    <i class="bi bi-trash me-1"></i>Delete
                  </button>
                </div>
              </td>
            </tr>
          {% empty %}
            <tr>
              <td colspan="6" class="text-center text-secondary py-4">No listings yet.</td>
            </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>
  </div>
</div>

<!-- Delete Confirmation Modals -->
{% for p in products %}
<div class="modal fade" id="deleteModal{{ p.pk }}" tabindex="-1" aria-labelledby="deleteModalLabel{{ p.pk }}" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="deleteModalLabel{{ p.pk }}">
          <i class="bi bi-exclamation-triangle-fill text-danger me-2"></i>Delete Product?
        </h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <p class="mb-2">Are you sure you want to permanently delete this product?</p>
        <div class="alert alert-warning mb-0">
          <strong>{{ p.title }}</strong>
          <div class="small mt-1">This will delete:</div>
          <ul class="small mb-0">
            <li>All product images</li>
            <li>All specifications</li>
            {% if p.kind == "FILE" %}<li>All digital assets</li>{% endif %}
            <li>Product listing data</li>
          </ul>
        </div>
        <p class="text-danger small mt-2 mb-0"><strong>This action cannot be undone.</strong></p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancel</button>
        <form method="post" action="{% url 'products:seller_delete' p.pk %}" class="d-inline">
          {% csrf_token %}
          <button type="submit" class="btn btn-danger">
            <i class="bi bi-trash me-1"></i>Delete Permanently
          </button>
        </form>
      </div>
    </div>
  </div>
</div>
{% endfor %}
{% endblock %}
```

reviews\templates\reviews\product_reviews.html

```
{# reviews/templates/reviews/product_reviews.html #}
{% extends "base.html" %}
{% block title %}Reviews · Home Craft 3D{% endblock %}

{% block content %}
<div class="container py-4" style="max-width: 900px;">
  <div class="d-flex align-items-center justify-content-between mb-3">
    <div>
      <h1 class="h4 mb-1">Product Reviews</h1>
      <div class="text-secondary small">
        {{ review_count }} review{{ review_count|pluralize }} · Avg {{ avg_rating|floatformat:1 }}/5
      </div>
    </div>
    <a class="btn btn-outline-dark" href="{% url 'products:detail' pk=product_id slug='-' %}" style="display:none;"></a>
  </div>

  <div class="card">
    <div class="card-body">
      {% for r in reviews %}
        <div class="mb-3 pb-3 border-bottom">
          <div class="d-flex align-items-center justify-content-between">
            <div class="fw-semibold">{{ r.buyer.username }}</div>
            <div class="text-secondary small">{{ r.created_at|date:"M j, Y" }}</div>
          </div>
          <div class="mt-1">
            <span class="badge text-bg-dark">{{ r.rating }}/5</span>
            {% if r.title %}
              <span class="ms-2 fw-semibold">{{ r.title }}</span>
            {% endif %}
          </div>
          {% if r.body %}
            <div class="mt-2">{{ r.body|linebreaksbr }}</div>
          {% endif %}
        </div>
      {% empty %}
        <div class="text-secondary">No reviews yet.</div>
      {% endfor %}
    </div>
  </div>
</div>
{% endblock %}
```

products\templates\products\seller\product_specs_form.html

```
{# products/templates/products/seller/product_specs_form.html #}
{% extends "products/seller/base_seller.html" %}

{% block title %}Product Specifications · Home Craft 3D{% endblock %}

{% block content %}
<div class="container my-4">
  <div class="d-flex align-items-center justify-content-between mb-4">
    <div>
      <h1 class="h4 mb-0">Edit Specifications</h1>
      <div class="text-secondary small mt-1">{{ product.title }}</div>
    </div>
    <div class="d-flex gap-2">
      <a href="{% url 'products:seller_edit' product.pk %}" class="btn btn-outline-dark btn-sm">
        <i class="bi bi-pencil me-1"></i>Edit Listing
      </a>
      <a href="{% url 'products:seller_images' product.pk %}" class="btn btn-outline-dark btn-sm">
        <i class="bi bi-images me-1"></i>Manage Images
      </a>
      {% if product.kind == "FILE" %}
        <a href="{% url 'products:seller_assets' product.pk %}" class="btn btn-outline-dark btn-sm">
          <i class="bi bi-paperclip me-1"></i>Manage Assets
        </a>
      {% endif %}
      <a href="{% url 'products:seller_list' %}" class="btn btn-outline-secondary btn-sm">
        <i class="bi bi-arrow-left me-1"></i>Back to Products
      </a>
    </div>
  </div>

  {% if messages %}
    <div class="mb-3">
      {% for message in messages %}
        <div class="alert alert-info mb-2">{{ message }}</div>
      {% endfor %}
    </div>
  {% endif %}

  <div class="row">
    <div class="col-lg-8">
      <div class="card">
        <div class="card-header bg-light">
          <h5 class="mb-0">{{ product.title }}</h5>
          <small class="text-secondary">
            {% if kind == "MODEL" %}
              Physical Model Specifications
            {% else %}
              Digital File Specifications
            {% endif %}
          </small>
        </div>
        <div class="card-body">
          <form method="post" novalidate>
            {% csrf_token %}

            {% if form.non_field_errors %}
              <div class="alert alert-danger">{{ form.non_field_errors }}</div>
            {% endif %}
      {% if kind == "MODEL" %}
       <!-- Physical Product Specs -->
       <!-- Material & Color -->
       <div class="row">
        <div class="col-md-6 mb-3">
         <label for="{{ form.material.id_for_label }}" class="form-label">Material</label>
         {{ form.material }}
                {% if form.material.errors %}<div class="small text-danger mt-1">{{ form.material.errors.0
}}</div>{% endif %}
         <small class="form-text text-muted d-block mt-1">E.g., PLA, ABS, Resin, Metal, Wood</small>
        </div>

        <div class="col-md-6 mb-3">
         <label for="{{ form.color.id_for_label }}" class="form-label">Available Colors</label>
         {{ form.color }}
          {% if form.color.errors %}<div class="small text-danger mt-1">{{ form.color.errors.0 }}</div>{% endif
%}
         <small class="form-text text-muted d-block mt-1">E.g., White, Multi-color, Black</small>
        </div>
       </div>

       <!-- Number of Colors -->
       <div class="mb-3">
        <label for="{{ form.num_colors.id_for_label }}" class="form-label">Number of Colors in Print</label>
        {{ form.num_colors }}
            {% if form.num_colors.errors %}<div class="small text-danger mt-1">{{ form.num_colors.errors.0
}}</div>{% endif %}
        <small class="form-text text-muted d-block mt-1">How many different colors are in this print?</small>
       </div>

       <!-- Dimensions -->

       <div class="row">
        <div class="col-md-3 mb-3">
         <label for="{{ form.width.id_for_label }}" class="form-label">Width</label>
         {{ form.width }}
            {% if form.width.errors %}<div class="small text-danger mt-1">{{ form.width.errors.0 }}</div>{%
endif %}
        </div>
        <div class="col-md-3 mb-3">
         <label for="{{ form.height.id_for_label }}" class="form-label">Height</label>
         {{ form.height }}
           {% if form.height.errors %}<div class="small text-danger mt-1">{{ form.height.errors.0 }}</div>{%
endif %}
        </div>
        <div class="col-md-3 mb-3">
         <label for="{{ form.depth.id_for_label }}" class="form-label">Depth</label>
         {{ form.depth }}
            {% if form.depth.errors %}<div class="small text-danger mt-1">{{ form.depth.errors.0 }}</div>{%
endif %}
        </div>
        <div class="col-md-3 mb-3">
         <label for="{{ form.dimension_unit.id_for_label }}" class="form-label">Units</label>
         {{ form.dimension_unit }}
                          {% if form.dimension_unit.errors %}<div class="small text-danger mt-1">{{
form.dimension_unit.errors.0 }}</div>{% endif %}
        </div>
       </div>

       <!-- Weight -->
       <div class="mb-3">
```

```
          <label for="{{ form.weight_grams.id_for_label }}" class="form-label">Weight (grams)</label>
          {{ form.weight_grams }}
          {% if form.weight_grams.errors %}<div class="small text-danger mt-1">{{ form.weight_grams.errors.0
}}</div>{% endif %}
          <small class="form-text text-muted d-block mt-1">Estimated weight of the printed model</small>
        </div>

        <!-- Support Required -->
        <div class="mb-3">
          <div class="form-check">
            {{ form.support_required }}
            <label class="form-check-label" for="{{ form.support_required.id_for_label }}">
              Requires support structures for printing
            </label>
          </div>
          <small class="form-text text-muted d-block mt-1">Check this if the model needs supports when 3D
printing</small>
        </div>

        <!-- Additional Specifications -->
        <div class="mb-3">
          <label for="{{ form.specifications.id_for_label }}" class="form-label">Additional Specifications</label>
          {{ form.specifications }}
          {% if form.specifications.errors %}<div class="small text-danger mt-1">{{ form.specifications.errors.0
}}</div>{% endif %}
          <small class="form-text text-muted d-block mt-1">Assembly instructions, scale info, requested colors,
special notes, etc.</small>
        </div>

      {% else %}
        <!-- Digital File Specs -->
        <!-- Software Requirements -->
        <div class="mb-3">
          <label for="{{ form.software_requirements.id_for_label }}" class="form-label">Software
Required</label>
          {{ form.software_requirements }}
          {% if form.software_requirements.errors %}<div class="small text-danger mt-1">{{
form.software_requirements.errors.0 }}</div>{% endif %}
          <small class="form-text text-muted d-block mt-1">E.g., Fusion 360, FreeCAD, Blender,
Meshmixer</small>
        </div>

        <!-- Compatible Platforms -->
        <div class="mb-3">
          <label for="{{ form.compatible_software.id_for_label }}" class="form-label">Compatible
Platforms</label>
          {{ form.compatible_software }}
          {% if form.compatible_software.errors %}<div class="small text-danger mt-1">{{
form.compatible_software.errors.0 }}</div>{% endif %}
          <small class="form-text text-muted d-block mt-1">E.g., Windows, macOS, Linux, All Platforms</small>
        </div>

        <!-- License Type -->
        <div class="mb-3">
          <label for="{{ form.license_type.id_for_label }}" class="form-label">License Type</label>
          {{ form.license_type }}
          {% if form.license_type.errors %}<div class="small text-danger mt-1">{{ form.license_type.errors.0
}}</div>{% endif %}
          <small class="form-text text-muted d-block mt-1">What buyers can use this file for</small>
        </div>
```

```
              <!-- Requirements & Notes -->
              <div class="mb-3">
                <label for="{{ form.requirements.id_for_label }}" class="form-label">System Requirements & Notes</label>
                {{ form.requirements }}
                {% if form.requirements.errors %}<div class="small text-danger mt-1">{{ form.requirements.errors.0 }}</div>{% endif %}
                <small class="form-text text-muted d-block mt-1">Installation steps, dependencies, system specs, technical notes</small>
              </div>
            {% endif %}

            <!-- Actions -->
            <div class="d-flex gap-2">
              <button type="submit" class="btn btn-primary">
                <i class="bi bi-check me-1"></i>Save Specifications
              </button>
              <a href="{% url 'products:seller_list' %}" class="btn btn-outline-secondary">Cancel</a>
            </div>
          </form>
        </div>
      </div>
    </div>

    <div class="col-lg-4">
      <div class="card bg-light">
        <div class="card-body">
          <h6 class="card-title mb-3">Tips</h6>
          {% if kind == "MODEL" %}
            <ul class="small">
              <li>Accurate dimensions help buyers understand scale</li>
              <li>Material info is important for buyers' printing setup</li>
              <li>Note if supports are needed to save buyers time</li>
              <li>Use the specifications field for assembly instructions or other details</li>
            </ul>
          {% else %}
            <ul class="small">
              <li>Be clear about software requirements</li>
              <li>Specify which operating systems work with your file</li>
              <li>License type helps buyers know their usage rights</li>
              <li>Include any special installation or compatibility notes</li>
            </ul>
          {% endif %}
        </div>
      </div>
    </div>
  </div>
</div>
{% endblock %}
```

accounts\templates\accounts\profile.html

```
{# templates/accounts/profile.html #}
{% extends "accounts/base_accounts.html" %}
{% block title %}Profile · Home Craft 3D{% endblock %}

{% block content %}
  <div class="card">
    <h2 style="margin-top:0;">Profile</h2>
    <div class="help">Username (public): <strong>{{ request.user.username }}</strong></div>

    <form method="post" enctype="multipart/form-data" novalidate style="margin-top:14px;">
      {% csrf_token %}

      {% if form.non_field_errors %}
        <div class="msg error">{{ form.non_field_errors }}</div>
      {% endif %}

      <div class="form-row">
        <div>
          <label for="{{ form.first_name.id_for_label }}">First name</label>
          {{ form.first_name }}
          {% for e in form.first_name.errors %}<div class="help" style="color:#b91c1c;">{{ e }}</div>{% endfor %}
        </div>

        <div>
          <label for="{{ form.last_name.id_for_label }}">Last name</label>
          {{ form.last_name }}
          {% for e in form.last_name.errors %}<div class="help" style="color:#b91c1c;">{{ e }}</div>{% endfor %}
        </div>

        <div>
          <label for="{{ form.email.id_for_label }}">Email</label>
          {{ form.email }}
          {% for e in form.email.errors %}<div class="help" style="color:#b91c1c;">{{ e }}</div>{% endfor %}
        </div>

        <div>
          <label for="{{ form.phone_1.id_for_label }}">Phone 1</label>
          {{ form.phone_1 }}
          {% for e in form.phone_1.errors %}<div class="help" style="color:#b91c1c;">{{ e }}</div>{% endfor %}
        </div>

        <div>
          <label for="{{ form.phone_2.id_for_label }}">Phone 2</label>
          {{ form.phone_2 }}
          {% for e in form.phone_2.errors %}<div class="help" style="color:#b91c1c;">{{ e }}</div>{% endfor %}
        </div>

        <div>
          <label for="{{ form.address_1.id_for_label }}">Address 1</label>
          {{ form.address_1 }}
          {% for e in form.address_1.errors %}<div class="help" style="color:#b91c1c;">{{ e }}</div>{% endfor %}
        </div>

        <div>
          <label for="{{ form.address_2.id_for_label }}">Address 2</label>
          {{ form.address_2 }}
          {% for e in form.address_2.errors %}<div class="help" style="color:#b91c1c;">{{ e }}</div>{% endfor %}
        </div>
        <div>
          <label for="{{ form.city.id_for_label }}">City</label>
          {{ form.city }}
          {% for e in form.city.errors %}<div class="help" style="color:#b91c1c;">{{ e }}</div>{% endfor %}
        </div>

        <div>
          <label for="{{ form.state.id_for_label }}">State</label>
          {{ form.state }}
          {% for e in form.state.errors %}<div class="help" style="color:#b91c1c;">{{ e }}</div>{% endfor %}
        </div>

        <div>
          <label for="{{ form.zip_code.id_for_label }}">ZIP</label>
          {{ form.zip_code }}
          {% for e in form.zip_code.errors %}<div class="help" style="color:#b91c1c;">{{ e }}</div>{% endfor %}
        </div>

        <div>
          <label for="{{ form.avatar.id_for_label }}">Profile picture</label>
          {% if form.avatar.value %}
            <div class="d-flex align-items-center gap-3" style="margin: 6px 0 10px;">
              <img src="{{ form.avatar.value.url }}" alt="Current profile picture" class="rounded border" style="width:64px;height:64px;object-fit:cover;">
              <div class="small text-secondary">Current image</div>
            </div>
          {% endif %}
          {{ form.avatar }}
          {% for e in form.avatar.errors %}<div class="help" style="color:#b91c1c;">{{ e }}</div>{% endfor %}
        </div>
      </div>

      <div style="margin-top:12px;">
        <label for="{{ form.is_seller.id_for_label }}">Seller account</label>
        {{ form.is_seller }}
        <div class="help">Sellers will be required to complete Stripe onboarding before selling.</div>
      </div>

      <div class="actions">
        <button class="btn" type="submit">Save</button>
      </div>
    </form>
  </div>
{% endblock %}
```

templates\includes\promo_banner.html

```
{# templates/includes/promo_banner.html #}
{% with cfg=site_config %}
  {% if cfg and cfg.promo_banner_enabled and cfg.promo_banner_text %}
    <div class="hc3-promo-banner">
      <div class="container-fluid text-center">
        <span class="hc3-promo-badge">PROMO</span>
        <span class="hc3-promo-text">{{ cfg.promo_banner_text }}</span>
      </div>
    </div>
  {% endif %}
{% endwith %}
```

products\management\commands\prune_engagement_events.py

```
# products/management/commands/prune_engagement_events.py
from __future__ import annotations

from datetime import timedelta

from django.core.management.base import BaseCommand
from django.utils import timezone

from products.models import ProductEngagementEvent


class Command(BaseCommand):
    help = "Delete old ProductEngagementEvent rows older than N days (default 90)."

    def add_arguments(self, parser):
        parser.add_argument(
            "--days",
            type=int,
            default=90,
            help="Delete events older than this many days (default: 90).",
        )
        parser.add_argument(
            "--dry-run",
            action="store_true",
            help="Show how many rows would be deleted without deleting them.",
        )
        parser.add_argument(
            "--chunk",
            type=int,
            default=5000,
            help="Delete in chunks of this size to avoid long locks (default: 5000).",
        )

    def handle(self, *args, **options):
        days: int = options["days"]
        dry_run: bool = options["dry_run"]
        chunk: int = options["chunk"]

        cutoff = timezone.now() - timedelta(days=days)

        qs = ProductEngagementEvent.objects.filter(created_at__lt=cutoff).order_by("id")
        total = qs.count()

        if total == 0:
            self.stdout.write(self.style.SUCCESS("No engagement events to prune."))
            return

        if dry_run:
            self.stdout.write(
                self.style.WARNING(
                    f"[DRY RUN] Would delete {total} engagement events older than {days} days (cutoff={cutoff})."
                )
            )
            return

        deleted = 0
        while True:
            ids = list(qs.values_list("id", flat=True)[:chunk])
            if not ids:
```

```
        break
    ProductEngagementEvent.objects.filter(id__in=ids).delete()
    deleted += len(ids)
    self.stdout.write(f"Deleted {deleted}/{total}...")

self.stdout.write(
    self.style.SUCCESS(
        f"Pruned {deleted} engagement events older than {days} days (cutoff={cutoff})."
    )
)
```

orders\templates\orders\purchases.html

```
{# orders/templates/orders/purchases.html #}
{% extends "base.html" %}
{% load money %}

{% block title %}Purchases · Home Craft 3D{% endblock %}

{% block content %}
<div class="container py-4">
  <div class="d-flex align-items-center justify-content-between mb-3">
    <h1 class="h4 mb-0">Purchases</h1>
  </div>

  <div class="card shadow-sm">
    <div class="card-body">
      <p class="text-secondary mb-0">Paid orders only. This is your download hub.</p>
    </div>

    <div class="table-responsive">
      <table class="table mb-0 align-middle">
        <thead>
          <tr>
            <th>Paid</th>
            <th>Order</th>
            <th>Items</th>
            <th class="text-end">Total</th>
            <th class="text-end">Downloads</th>
          </tr>
        </thead>
        <tbody>
          {% for order in orders %}
            <tr>
              <td class="text-muted small">{{ order.paid_at|date:"M j, Y" }}</td>
              <td>
                <a class="text-decoration-none" href="{% url 'orders:detail' order_id=order.id %}">
                  {{ order.id }}
                </a>
              </td>
              <td class="small">
                {{ order.items.count }} item{{ order.items.count|pluralize }}
              </td>
              <td class="text-end fw-semibold">${{ order.total_cents|cents_to_dollars }}</td>
              <td class="text-end">
                <div class="d-flex flex-column gap-1 align-items-end">
                  {% if order.has_digital_assets %}
                    <a class="btn btn-sm btn-outline-primary"
                       href="{% url 'orders:download_all_assets' order_id=order.id %}">
                      <i class="bi bi-download me-1"></i>Download all
                    </a>
                  {% endif %}
                  {% for item in order.items.all %}
                    {% if item.is_digital %}
                      {% for a in item.product.digital_assets.all %}
                        <a class="btn btn-sm btn-outline-primary"
                           href="{% url 'orders:download_asset' order_id=order.id asset_id=a.id %}">
                          Download{% if a.original_filename %}: {{ a.original_filename }}{% endif %}
                        </a>
                      {% empty %}
                        <span class="text-muted small">No files</span>
                      {% endfor %}
                    {% endif %}
                  {% endfor %}
                </div>
              </td>
            </tr>
          {% empty %}
            <tr>
              <td colspan="5" class="text-center text-muted py-4">No purchases yet.</td>
            </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>

    {% if page_obj and page_obj.paginator.num_pages > 1 %}
      <div class="card-footer bg-white">
        <nav aria-label="Purchases pages">
          <ul class="pagination mb-0">
            {% if page_obj.has_previous %}
              <li class="page-item"><a class="page-link" href="?page={{ page_obj.previous_page_number }}">Previous</a></li>
            {% else %}
              <li class="page-item disabled"><span class="page-link">Previous</span></li>
            {% endif %}

            <li class="page-item disabled"><span class="page-link">Page {{ page_obj.number }} of {{ page_obj.paginator.num_pages }}</span></li>

            {% if page_obj.has_next %}
              <li class="page-item"><a class="page-link" href="?page={{ page_obj.next_page_number }}">Next</a></li>
            {% else %}
              <li class="page-item disabled"><span class="page-link">Next</span></li>
            {% endif %}
          </ul>
        </nav>
      </div>
    {% endif %}
  </div>
</div>
{% endblock %}
```

core\recaptcha.py

```
# core/recaptcha.py
from __future__ import annotations

import json
import logging
import urllib.parse
import urllib.request
from dataclasses import dataclass
from typing import Callable

from django.conf import settings
from django.contrib import messages
from django.http import HttpRequest, HttpResponse
from django.shortcuts import redirect

logger = logging.getLogger(__name__)


@dataclass(frozen=True)
class RecaptchaResult:
    ok: bool
    score: float
    action: str
    error: str


def _get_client_ip(request: HttpRequest) -> str:
    # If later behind a proxy, swap to X-Forwarded-For parsing
    return request.META.get("REMOTE_ADDR") or ""


def verify_recaptcha_v3(*, request: HttpRequest, token: str, expected_action: str) -> RecaptchaResult:
    """
    Verify reCAPTCHA v3 token against Google.

    Requires settings:
      - RECAPTCHA_V3_SECRET_KEY
      - RECAPTCHA_V3_MIN_SCORE (default 0.5)
      - RECAPTCHA_ENABLED (default True)
    """
    enabled = bool(getattr(settings, "RECAPTCHA_ENABLED", True))
    if not enabled:
        return RecaptchaResult(ok=True, score=1.0, action=expected_action, error="disabled")

    site_key = (getattr(settings, "RECAPTCHA_V3_SITE_KEY", "") or "").strip()
    secret = (getattr(settings, "RECAPTCHA_V3_SECRET_KEY", "") or "").strip()

    # If keys aren't configured, don't block checkout.
    if not site_key or not secret:
        logger.warning("reCAPTCHA keys missing; bypassing verification.")
        return RecaptchaResult(ok=True, score=1.0, action=expected_action, error="recaptcha_not_configured")

    token = (token or "").strip()
    if not token:
        return RecaptchaResult(ok=False, score=0.0, action="", error="missing_token")

    data = urllib.parse.urlencode(
        {
            "secret": secret,
            "response": token,
            "remoteip": _get_client_ip(request),
        }
    ).encode("utf-8")

    try:
        req = urllib.request.Request(
            "https://www.google.com/recaptcha/api/siteverify",
            data=data,
            headers={"Content-Type": "application/x-www-form-urlencoded"},
            method="POST",
        )
        with urllib.request.urlopen(req, timeout=6) as resp:
            raw = resp.read().decode("utf-8")
        payload = json.loads(raw)
    except Exception as e:
        logger.exception("reCAPTCHA verification failed: %s", e)
        return RecaptchaResult(ok=False, score=0.0, action="", error="verify_error")

    success = bool(payload.get("success"))
    score = float(payload.get("score") or 0.0)
    action = str(payload.get("action") or "")
    min_score = float(getattr(settings, "RECAPTCHA_V3_MIN_SCORE", 0.5))

    if not success:
        return RecaptchaResult(ok=False, score=score, action=action, error="not_success")

    # v3 action must match or it’s likely scripted reuse
    if expected_action and action and action != expected_action:
        return RecaptchaResult(ok=False, score=score, action=action, error="action_mismatch")

    if score < min_score:
        return RecaptchaResult(ok=False, score=score, action=action, error="low_score")

    return RecaptchaResult(ok=True, score=score, action=action, error="")


def require_recaptcha_v3(action: str) -> Callable:
    """
    Decorator for POST endpoints. Expects token in POST:
      - "g-recaptcha-response" (common)
      - OR "recaptcha_token"
    """
    def decorator(view_func: Callable) -> Callable:
        def wrapped(request: HttpRequest, *args, **kwargs) -> HttpResponse:
            if request.method != "POST":
                return view_func(request, *args, **kwargs)

            token = (request.POST.get("g-recaptcha-response") or request.POST.get("recaptcha_token") or "").strip()
            result = verify_recaptcha_v3(request=request, token=token, expected_action=action)

            if not result.ok:
                # Friendly UX
                messages.error(request, "We couldn’t verify you’re human. Please try again.")
                # safest redirect: back to referer, else home
                back = request.META.get("HTTP_REFERER") or "/"
                return redirect(back)

            return view_func(request, *args, **kwargs)

        wrapped.__name__ = getattr(view_func, "__name__", "wrapped")
        wrapped.__doc__ = getattr(view_func, "__doc__", "")
        return wrapped
    return decorator
```

accounts\templates\accounts\recovery_codes.html

```
{# templates/accounts/recovery_codes.html #}
{% extends 'accounts/base_accounts.html' %}

{% block title %}Recovery Codes{% endblock %}

{% block content %}
<div class="container mt-5">
    <div class="row justify-content-center">
        <div class="col-md-6">
            <div class="card">
                <div class="card-header bg-warning text-dark">
                    <h4 class="mb-0">Recovery Codes</h4>
                </div>
                <div class="card-body">
                    <div class="alert alert-danger">
                            <strong>Important:</strong> Save these codes in a secure location. Each code can only be used once.
                        If you lose access to your authenticator app, use these codes to sign in.
                    </div>

                    <div class="alert alert-light border">
                        <div id="codes-list">
                            {% for code in codes %}
                                <div class="code-item mb-2">
                                    <code>{{ code }}</code>
                                </div>
                            {% endfor %}
                        </div>
                    </div>

                    <div class="mb-3">
                        <button class="btn btn-outline-secondary me-2" onclick="copyAllCodes()">
                            <i class="bi bi-clipboard"></i> Copy All Codes
                        </button>
                        <button class="btn btn-outline-secondary" onclick="printCodes()">
                            <i class="bi bi-printer"></i> Print
                        </button>
                    </div>

                    <hr>

                    <p class="text-muted small mb-0">
                        <i class="bi bi-info-circle"></i>
                        Store these codes securely (password manager, safe, etc.).
                    </p>
                </div>
                <div class="card-footer">
                    <a href="{% url 'accounts:view_2fa_status' %}" class="btn btn-secondary">
                        Back to 2FA Settings
                    </a>
                </div>
            </div>
        </div>
    </div>
</div>

<script>
function copyAllCodes() {
    const codes = document.querySelectorAll('.code-item code');
    const text = Array.from(codes).map(c => c.textContent.trim()).join('\n');
    navigator.clipboard.writeText(text).then(() => {
        alert('Recovery codes copied to clipboard!');
    }).catch(() => {
        alert('Failed to copy codes. Please try again.');
    });
}

function printCodes() {
    const printWindow = window.open('', '', 'height=600,width=600');
    const codes = document.querySelectorAll('.code-item code');
    const codeList = Array.from(codes).map(c => c.textContent.trim()).join('<br>');

    printWindow.document.write(`
        <html>
        <head>
            <title>2FA Recovery Codes</title>
            <style>
                body { font-family: Arial, sans-serif; padding: 20px; }
                h2 { margin-bottom: 20px; }
                .warning { color: red; font-weight: bold; margin-bottom: 20px; }
                pre { font-size: 14px; background: #f5f5f5; padding: 10px; border-radius: 5px; }
            </style>
        </head>
        <body>
            <h2>Two-Factor Authentication Recovery Codes</h2>
            <p class="warning">⚠ KEEP THESE CODES SAFE</p>
            <p>These codes can be used to access your account if you lose your authenticator app.</p>
            <pre>${codeList}</pre>
            <p style="margin-top: 30px; font-size: 12px; color: #666;">
                Generated: ${new Date().toLocaleString()}
            </p>
        </body>
        </html>
    `);
    printWindow.document.close();
    setTimeout(() => printWindow.print(), 250);
}
</script>
{% endblock %}
```

templates\admin\redirect.html

```
{# templates/admin/redirect.html #}
<!DOCTYPE html>
<html>
<head>
    <title>Redirect</title>
    <meta http-equiv="refresh" content="0;url={{ redirect_url }}">
</head>
<body>
    <p><a href="{{ redirect_url }}">Redirecting...</a></p>
</body>
</html>
```

legal\templates\legal\refund.html

```
{# templates/legal/refund.html #}
{% extends "legal/_legal_base.html" %}

{% block doc_content %}
<h2>Digital Files</h2>
<p>Digital downloads are generally non-refundable once delivered. Refunds may be granted if a file is corrupted or materially different from the listing.</p>

<h2>Physical Items</h2>
<p>Returns are accepted within 14 days of delivery if the item is defective or significantly not as described. Buyers are responsible for return shipping unless the seller states otherwise.</p>

<h2>How to Request a Refund</h2>
<ol>
  <li>Go to <strong>My Orders</strong> and submit a refund request with details</li>
  <li>Our team will review and coordinate with the seller</li>
</ol>

<p><strong>Contact:</strong> <a href="mailto:homecraft3dstore@gmail.com">homecraft3dstore@gmail.com</a></p>
{% endblock %}
```

templates\emails\refund_approved.html

```
{# templates/emails/payout_sent.html #}
{% extends "emails/base.html" %}
{% block content %}
  <h1 style="margin:0 0 12px;font-size:20px;">Refund approved</h1>
  <p style="margin:0 0 12px;">Your refund for order <strong>#{{ order_id }}</strong> was approved.</p>

  <div style="margin:16px 0;padding:16px;border:1px solid #e5e7eb;border-radius:10px;background:#f9fafb;">
    <div style="font-size:12px;color:#6b7280;margin-bottom:6px;">Refund details</div>
    <div><strong>Item:</strong> {{ item_title }}</div>
    <div><strong>Amount:</strong> {{ refund_amount }}</div>
    {% if decision_note %}
      <div style="margin-top:6px;color:#6b7280;">Seller note: {{ decision_note }}</div>
    {% endif %}
  </div>

  <a href="{{ order_link }}" style="color:#2563eb;text-decoration:none;">View order</a>
{% endblock %}
```

templates\emails\refund_declined.html

```
{# templates/emails/payout_sent.html #}
{% extends "emails/base.html" %}
{% block content %}
  <h1 style="margin:0 0 12px;font-size:20px;">Refund declined</h1>
  <p style="margin:0 0 12px;">Your refund for order <strong>#{{ order_id }}</strong> was declined.</p>

  <div style="margin:16px 0;padding:16px;border:1px solid #e5e7eb;border-radius:10px;background:#f9fafb;">
    <div style="font-size:12px;color:#6b7280;margin-bottom:6px;">Refund details</div>
    <div><strong>Item:</strong> {{ item_title }}</div>
    <div><strong>Amount:</strong> {{ refund_amount }}</div>
    {% if decision_note %}
      <div style="margin-top:6px;color:#6b7280;">Seller note: {{ decision_note }}</div>
    {% endif %}
  </div>

  <a href="{{ order_link }}" style="color:#2563eb;text-decoration:none;">View order</a>
{% endblock %}
```

templates\emails\refund_processed.html

```
{# templates/emails/refund_processed.html #}
{% extends "emails/base.html" %}
{% block content %}
  <h1 style="margin:0 0 12px;font-size:20px;">Refund processed</h1>
  <p style="margin:0 0 12px;">Your refund for order <strong>#{{ order_id }}</strong> has been processed.</p>

  <div style="margin:16px 0;padding:16px;border:1px solid #e5e7eb;border-radius:10px;background:#f9fafb;">
    <div style="font-size:12px;color:#6b7280;margin-bottom:6px;">Refund details</div>
    <div><strong>Item:</strong> {{ item_title }}</div>
    <div><strong>Amount:</strong> {{ refund_amount }}</div>
  </div>

  <a href="{{ order_link }}" style="color:#2563eb;text-decoration:none;">View order</a>
{% endblock %}
```

templates\emails\refund_requested.html

```
{# templates/emails/refund_processed.html #}
{% extends "emails/base.html" %}
{% block content %}
  <h1 style="margin:0 0 12px;font-size:20px;">Refund requested</h1>
  <p style="margin:0 0 12px;">A refund was requested for order <strong>#{{ order_id }}</strong>.</p>

  <div style="margin:16px 0;padding:16px;border:1px solid #e5e7eb;border-radius:10px;background:#f9fafb;">
    <div style="font-size:12px;color:#6b7280;margin-bottom:6px;">Request details</div>
    <div><strong>Item:</strong> {{ item_title }}</div>
    <div><strong>Amount:</strong> {{ refund_amount }}</div>
    <div><strong>Reason:</strong> {{ reason }}</div>
    {% if notes %}
      <div style="margin-top:6px;color:#6b7280;">Notes: {{ notes }}</div>
    {% endif %}
  </div>

  <a href="{{ order_link }}" style="color:#2563eb;text-decoration:none;">Review refund request</a>
{% endblock %}
```

orders\refunds.py

```python
# orders/refunds.py

from __future__ import annotations

from typing import Iterable

from django.db import transaction

from orders.models import Order, OrderItem, OrderEvent
from payments.models import SellerBalanceEntry


@transaction.atomic
def refund_order_items(
    *,
    order: Order,
    items: Iterable[OrderItem],
    reason: str,
    refund_marketplace_fee: bool = False,
) -> None:
    """
    Refunds specific order items.

    Assumptions:
    - Stripe refund is handled upstream (this function is ledger + audit only)
    - Items belong to the given order
    """
    if order.status not in {Order.Status.PAID, Order.Status.REFUNDED}:
        raise ValueError("Cannot refund an unpaid order.")

    items_list = list(items)

    for item in items_list:
        if item.order_id != order.id:
            raise ValueError("OrderItem does not belong to this order.")

        SellerBalanceEntry.objects.create(
            seller=item.seller,
            amount_cents=-int(item.seller_net_cents),
            reason=SellerBalanceEntry.Reason.REFUND,
            order=order,
            order_item=item,
            note=reason,
        )

        if refund_marketplace_fee and int(item.marketplace_fee_cents or 0) > 0:
            SellerBalanceEntry.objects.create(
                seller=item.seller,
                amount_cents=int(item.marketplace_fee_cents),
                reason=SellerBalanceEntry.Reason.ADJUSTMENT,
                order=order,
                order_item=item,
                note="Marketplace fee refunded",
            )

    OrderEvent.objects.create(
        order=order,
        type=OrderEvent.Type.REFUNDED,
        message=reason,
    )

    if len(items_list) == order.items.count():
        order.status = Order.Status.REFUNDED
        order.save(update_fields=["status", "updated_at"])
```

accounts\templates\accounts\register.html

```
{# templates/accounts/register.html #}
{% extends "accounts/base_accounts.html" %}
{% load static %}
{% block title %}Register · Home Craft 3D{% endblock %}

{% block content %}
  <div style="max-width: 540px; margin: 40px auto;">
    <div style="text-align: center; margin-bottom: 32px;">
      <div style="margin:0 auto 16px;">
        <img src="{% static 'images/homecraft3d_icon.svg' %}" alt="Home Craft 3D" style="height:60px;width:auto;">
      </div>
      <h1 style="margin:0 0 8px;font-size:28px;font-weight:600;color:#111827;">Create account</h1>
      <p style="margin:0;color:#6b7280;font-size:15px;">
        Join to purchase downloads, track orders, and sell your models
      </p>
    </div>

    <div class="card" style="padding:32px;">

    <form method="post" novalidate style="margin-top: 14px;">
      {% csrf_token %}

      {% if form.non_field_errors %}
        <div class="msg error">{{ form.non_field_errors }}</div>
      {% endif %}

      <div style="display:flex;flex-direction:column;gap:20px;">
        <div>
          <label for="{{ form.username.id_for_label }}">Username</label>
          {{ form.username }}
          {% for e in form.username.errors %}<div class="help" style="color:#dc2626;margin-top:4px;">{{ e }}</div>{% endfor %}
        </div>

        <div>
          <label for="{{ form.email.id_for_label }}">Email</label>
          {{ form.email }}
           {% for e in form.email.errors %}<div class="help" style="color:#dc2626;margin-top:4px;">{{ e }}</div>{% endfor %}
        </div>

        <div style="display:grid;grid-template-columns:1fr 1fr;gap:12px;">
          <div>
            <label for="{{ form.first_name.id_for_label }}">First name <span style="color:#9ca3af;">(optional)</span></label>
            {{ form.first_name }}
            {% for e in form.first_name.errors %}<div class="help" style="color:#dc2626;margin-top:4px;">{{ e }}</div>{% endfor %}
          </div>

          <div>
            <label for="{{ form.last_name.id_for_label }}">Last name <span style="color:#9ca3af;">(optional)</span></label>
            {{ form.last_name }}
            {% for e in form.last_name.errors %}<div class="help" style="color:#dc2626;margin-top:4px;">{{ e }}</div>{% endfor %}
          </div>
        </div>
        <div>
          <label for="{{ form.password1.id_for_label }}">Password</label>
          {{ form.password1 }}
          {% for e in form.password1.errors %}<div class="help" style="color:#dc2626;margin-top:4px;">{{ e }}</div>{% endfor %}
        </div>

        <div>
          <label for="{{ form.password2.id_for_label }}">Confirm password</label>
          {{ form.password2 }}
          {% for e in form.password2.errors %}<div class="help" style="color:#dc2626;margin-top:4px;">{{ e }}</div>{% endfor %}
        </div>
      </div>

      <div style="margin-top:24px;padding:20px;background:#fefce8;border:1px solid #fde047;border-radius:8px;">
        <label style="display:flex;align-items:start;gap:12px;cursor:pointer;margin:0;">
          <div style="flex-shrink:0;margin-top:2px;">
            {{ form.register_as_seller }}
          </div>
          <div style="flex:1;">
            <div style="font-weight:600;color:#854d0e;margin-bottom:4px;font-size:14px;">
              Register as seller
            </div>
            <div style="font-size:13px;color:#a16207;line-height:1.5;">
              Enable selling to upload 3D models and physical prints to the marketplace. You'll complete Stripe onboarding after registration.
            </div>
          </div>
        </label>
      </div>

      <button class="btn" type="submit" style="width:100%;margin-top:24px;font-weight:500;">
        Create account
      </button>

      <div style="text-align:center;margin-top:20px;">
        <span style="color:#6b7280;font-size:14px;">Already have an account?</span>
        <a href="{% url 'accounts:login' %}" style="color:#111827;font-weight:500;margin-left:6px;">Sign in</a>
      </div>

      <div style="text-align:center;margin-top:20px;font-size:13px;color:#9ca3af;">
        By creating an account, you agree to our
        <a href="{% url 'legal:terms' %}" style="color:#6b7280;">Terms</a> and
        <a href="{% url 'legal:privacy' %}" style="color:#6b7280;">Privacy Policy</a>.
      </div>
    </form>
  </div>

  <div style="text-align:center;margin-top:24px;">
    <a href="{% url 'core:home' %}" style="color:#6b7280;font-size:14px;text-decoration:none;">
      ← Back to store
    </a>
  </div>
</div>
{% endblock %}
```

refunds\templates\refunds\request_create.html

```
{# templates/refunds/request_create.html #}
{% extends "base.html" %}
{% load money %}

{% block title %}Request Refund · Home Craft 3D{% endblock %}

{% block content %}
<div class="container py-4">
  <div class="d-flex justify-content-between align-items-center mb-3">
    <h1 class="h4 mb-0">Request refund</h1>
    <a class="btn btn-outline-secondary btn-sm"
       href="{% url 'orders:detail' order_id=order.id %}{% if order_token %}?t={{ order_token }}{% endif %}">
      Back to order
    </a>
  </div>

  <div class="card mb-3">
    <div class="card-body">
      <div class="fw-semibold">{{ item.product.title }}</div>
      <div class="text-muted small">Order #{{ order.id }}</div>

      <div class="mt-3">
        <div><strong>Item total:</strong> ${{ item.line_total_cents|cents_to_dollars }}</div>
        <div class="text-muted small">
          Note: Digital products are not refundable. Refunds are full refunds per physical item only.
        </div>
      </div>
    </div>
  </div>

  <form method="post"
        action="{% url 'refunds:buyer_create' order_id=order.id item_id=item.id %}{% if order_token %}?t={{ order_token }}{% endif %}">
    {% csrf_token %}

    <div class="card">
      <div class="card-body">
        {% if is_guest %}
          <div class="mb-3">
            <label class="form-label">Confirm checkout email</label>
            <input type="email"
                   name="guest_email"
                   value="{{ guest_email }}"
                   class="form-control"
                   required
                   placeholder="you@example.com">
            <div class="form-text">
              For guest orders, the email must match the one used at checkout.
            </div>
          </div>
        {% endif %}

        <div class="mb-3">
          <label class="form-label">Reason</label>
          {{ form.reason }}
        </div>

        <div class="mb-3">
          <label class="form-label">Notes (optional)</label>
          {{ form.notes }}
        </div>

        <button class="btn btn-danger">Submit refund request</button>
      </div>
    </div>
  </form>
</div>
{% endblock %}
```

reviews\templates\reviews\review_form.html

```
{# reviews/templates/reviews/review_form.html #}
{% extends "base.html" %}
{% block title %}Leave a Review · Home Craft 3D{% endblock %}

{% block content %}
<div class="container py-4" style="max-width: 760px;">
  <div class="d-flex align-items-center justify-content-between mb-3">
    <div>
      <h1 class="h4 mb-1">Leave a Review</h1>
      <div class="text-secondary small">
        For: <strong>{{ product.title }}</strong>
      </div>
      <div class="text-secondary small">
        Order item #{{ item.id }} · Qty {{ item.quantity }}
      </div>
    </div>
    <a class="btn btn-outline-dark" href="{{ product.get_absolute_url }}">Back to product</a>
  </div>

  <div class="card">
    <div class="card-body">
      <form method="post" novalidate>
        {% csrf_token %}

        {% if form.non_field_errors %}
          <div class="alert alert-danger">{{ form.non_field_errors }}</div>
        {% endif %}

        <div class="mb-3">
          <label class="form-label">Rating</label>
          {{ form.rating }}
          {% if form.rating.errors %}<div class="text-danger small">{{ form.rating.errors }}</div>{% endif %}
        </div>

        <div class="mb-3">
          <label class="form-label">Title (optional)</label>
          {{ form.title }}
          {% if form.title.errors %}<div class="text-danger small">{{ form.title.errors }}</div>{% endif %}
        </div>

        <div class="mb-3">
          <label class="form-label">Review (optional)</label>
          {{ form.body }}
          {% if form.body.errors %}<div class="text-danger small">{{ form.body.errors }}</div>{% endif %}
        </div>

        <button class="btn btn-dark" type="submit">
          Post Review
        </button>
      </form>
    </div>
  </div>
</div>
{% endblock %}
```

templates\emails\review_request.html

```
{# templates/emails/review_request.html #}
{% extends "emails/base.html" %}
{% block content %}
  <h1 style="margin:0 0 12px;font-size:20px;">How was your order?</h1>
  <p style="margin:0 0 12px;">We'd love your feedback on <strong>{{ product_title }}</strong>.</p>

  <div style="margin:16px 0;padding:16px;border:1px solid #e5e7eb;border-radius:10px;background:#f9fafb;">
    <div style="font-size:12px;color:#6b7280;margin-bottom:6px;">Leave a review</div>
    <a href="{{ review_link }}" style="color:#2563eb;text-decoration:none;">Write your review</a>
  </div>

  <p style="margin:0;color:#6b7280;font-size:13px;">Thanks for shopping at Home Craft 3D.</p>
{% endblock %}
```

core\security_headers.py

```
# core/security_headers.py
from __future__ import annotations

from django.conf import settings


class SecurityHeadersMiddleware:
    """
    Lightweight baseline. Safe for dev; improves safety for prod.

    NOTE:
    - CSP defaults are intentionally "works-first".
    - Tighten CSP once templates/assets stabilize.
    """

    def __init__(self, get_response):
        self.get_response = get_response

        # Stripe-friendly baseline CSP.
        # Override per environment with SECURITY_CSP if you want stricter rules.
        self.csp = getattr(
            settings,
            "SECURITY_CSP",
            "default-src 'self'; "
            "img-src 'self' data: blob: https:; "
            "style-src 'self' 'unsafe-inline' https:; "
            "script-src 'self' 'unsafe-inline' https://js.stripe.com https:; "
            "font-src 'self' https: data:; "
            "connect-src 'self' https://api.stripe.com https:; "
            "frame-src https://js.stripe.com https://hooks.stripe.com https://plausible.io; "
            "frame-ancestors 'none'; "
            "base-uri 'self';",
        )

        # Permissions-Policy: disable high-risk APIs by default
        self.permissions_policy = getattr(
            settings,
            "SECURITY_PERMISSIONS_POLICY",
            "camera=(), microphone=(), geolocation=(), payment=()",
        )

        self.referrer_policy = getattr(
            settings,
            "SECURE_REFERRER_POLICY",
            "strict-origin-when-cross-origin",
        )

    def __call__(self, request):
        resp = self.get_response(request)

        # Clickjacking
        resp["X-Frame-Options"] = "DENY"

        # MIME sniffing
        resp["X-Content-Type-Options"] = "nosniff"

        # Referrer leakage
        resp["Referrer-Policy"] = self.referrer_policy
```

```
# Modern browser XSS protection is handled by CSP. This disables legacy buggy behavior.
resp["X-XSS-Protection"] = "0"

# CSP
resp["Content-Security-Policy"] = self.csp

# Some sensible cross-origin defaults
resp["Cross-Origin-Opener-Policy"] = "same-origin"
resp["Cross-Origin-Resource-Policy"] = "same-origin"

# Permissions policy
resp["Permissions-Policy"] = self.permissions_policy

# Cache-Control for static and media files
if request.path.startswith(settings.STATIC_URL) or request.path.startswith(settings.MEDIA_URL):
    resp["Cache-Control"] = "public, max-age=31536000, immutable"

# Only set HSTS when HTTPS is truly enforced.
hsts_seconds = int(getattr(settings, "SECURE_HSTS_SECONDS", 0) or 0)
if hsts_seconds > 0:
    include_sub = bool(getattr(settings, "SECURE_HSTS_INCLUDE_SUBDOMAINS", True))
    preload = bool(getattr(settings, "SECURE_HSTS_PRELOAD", False))

    value = f"max-age={hsts_seconds}"
    if include_sub:
        value += "; includeSubDomains"
    if preload:
        value += "; preload"

    resp["Strict-Transport-Security"] = value

return resp
```

catalog\management\commands\seed_categories.py

```
# catalog/management/commands/seed_categories.py
from __future__ import annotations

from dataclasses import dataclass
from typing import Any

from django.core.management.base import BaseCommand
from django.db import transaction
from django.utils.text import slugify

from catalog.models import Category


# One master tree. We seed it twice: once for MODEL and once for FILE.
CATEGORY_TREE: list[dict[str, Any]] = [
    {
        "name": "Art & Sculptures",
        "children": [
            "Abstract",
            "Busts & Statues",
            "Wall Art",
            "Vases",
            "Figurative",
            "Anime & Fan Art",
            "Fantasy",
            "Sci-Fi",
        ],
    },
    {
        "name": "Miniatures & Terrain",
        "children": [
            "Tabletop Minis",
            "Terrain Tiles",
            "Buildings",
            "Scatter Props",
            "Bases",
            "Dungeon",
            "Sci-Fi Terrain",
            "Fantasy Terrain",
        ],
    },
    {
        "name": "Figurines",
        "children": [
            "Animals",
            "Characters",
            "Chibi",
            "Robots",
            "Monsters",
            "Superheroes",
            "Video Game",
            "Collectibles",
        ],
    },
    {
        "name": "Cosplay & Props",
        "children": [
            "Armor Parts",
            "Helmets & Masks",
            "Weapons (Props)",
            "Accessories",
            "Emblems & Badges",
            "Costume Hardware",
            "Display Stands",
        ],
    },
    {
        "name": "Toys & Games",
        "children": [
            "Board Game Inserts",
            "Dice & Accessories",
            "Puzzles",
            "Educational Toys",
            "Building Blocks",
            "Fidget Toys",
            "Game Tokens",
        ],
    },
    {
        "name": "Household & Organization",
        "children": [
            "Hooks & Hangers",
            "Storage Bins",
            "Drawer Organizers",
            "Cable Management",
            "Bathroom",
            "Laundry",
            "Garage",
            "Wall Mounts",
        ],
    },
    {
        "name": "Home & Decor",
        "children": [
            "Planters",
            "Candles & Holders",
            "Frames",
            "Clocks",
            "Seasonal Decor",
            "Signs & Plaques",
            "Lighting Shades",
        ],
    },
    {
        "name": "Kitchen & Dining",
        "children": [
            "Utensils",
            "Containers",
            "Coffee & Tea",
            "Spice Racks",
            "Cookie Cutters",
            "Bottle Openers",
            "Stands & Holders",
        ],
    },
    {
        "name": "Office & Desk",
        "children": [
            "Phone Stands",
            "Tablet Stands",
            "Pen Holders",
            "Desk Organizers",
            "Headphone Stands",
            "Monitor Accessories",
            "Laptop Risers",
        ],
    },
    {
        "name": "Electronics & Tech",
        "children": [
            "Cases",
            "Mounts",
            "Adapters",
            "Raspberry Pi",
            "Arduino",
            "Cable Clips",
            "Camera Mounts",
            "Battery Holders",
        ],
    },
    {
        "name": "Tools & Workshop",
        "children": [
            "Tool Holders",
            "Jigs",
            "Templates",
            "Pegboard",
            "Measuring Aids",
            "Bit Holders",
            "Workbench Accessories",
        ],
    },
    {
        "name": "Automotive",
        "children": [
            "Interior Accessories",
            "Phone Mounts",
            "Clips & Fasteners",
            "Organizers",
            "Detailing Tools",
            "Gauges & Pods",
        ],
    },
    {
        "name": "Outdoors & Garden",
        "children": [
            "Garden Tools",
            "Irrigation",
            "Camping",
            "Hiking",
            "Birdhouses & Feeders",
            "Outdoor Mounts",
        ],
    },
    {
        "name": "Pets & Animals",
        "children": [
            "Pet Toys",
            "Feeding Accessories",
            "Aquarium",
            "Leash & Collar",
            "Grooming",
            "Pet Organization",
        ],
    },
    {
        "name": "Jewelry & Accessories",
        "children": [
            "Rings",
            "Pendants",
            "Earrings",
            "Bracelets",
            "Jewelry Boxes",
            "Display Stands",
        ],
    },
    {
        "name": "Fashion & Wearables",
        "children": [
            "Belt Clips",
            "Buttons",
            "Buckles",
            "Shoe Accessories",
            "Wearable Tech Mounts",
        ],
    },
    {
        "name": "Sports & Fitness",
        "children": [
            "Bottle Holders",
            "Gym Accessories",
            "Bike Accessories",
            "Skate Accessories",
            "Sports Gear Mounts",
        ],
    },
    {
        "name": "Photography & Video",
        "children": [
            "Tripod Accessories",
            "GoPro",
            "Light Mounts",
            "Camera Rigs",
            "Cable/Accessory Holders",
        ],
    },
    {
        "name": "Music & Audio",
        "children": [
            "Instrument Accessories",
            "Mic Mounts",
            "Headphone Accessories",
            "Cable Management (Audio)",
        ],
    },
    {
        "name": "Science & Education",
        "children": [
            "STEM Models",
            "Molecules",
            "Anatomy",
            "Math Aids",
            "Physics Demonstrations",
        ],
    },
    {
        "name": "Robotics & Drones",
        "children": [
            "Frames",
            "Mounts",
            "Arms & Brackets",
            "Battery Trays",
            "Sensor Mounts",
        ],
    },
    {
        "name": "RC & Model Kits",
        "children": [
            "Airplanes",
            "Cars",
            "Boats",
            "Upgrades",
            "Parts & Spares",
        ],
    },
    {
        "name": "Architecture & Models",
        "children": [
            "Buildings",
            "Furniture Miniatures",
            "City Models",
            "Scale Accessories",
        ],
    },
    {
        "name": "Holiday & Seasonal",
        "children": [
            "Christmas",
            "Halloween",
            "Easter",
            "Valentine’s",
            "Thanksgiving",
            "New Year",
        ],
    },
    {
        "name": "Replacement Parts",
        "children": [
            "Appliance Parts",
            "Furniture Parts",
            "Electronics Parts",
            "Automotive Clips",
            "Knobs & Handles",
        ],
    },
    {
        "name": "Functional Parts",
        "children": [
            "Brackets",
            "Clamps",
            "Gears",
            "Enclosures",
            "Connectors",
            "Spacers & Shims",
        ],
    },
]


@dataclass(frozen=True)
class SeedStats:
    created: int = 0
    updated: int = 0


def _slug(name: str) -> str:
    return (slugify(name) or "").strip()[:140] or "category"


def _upsert_category(*, type_value: str, name: str, parent: Category | None, sort_order: int) -> tuple[Category, bool]:
    """
    Upsert by unique_together (type, parent, slug).
    Name/description/is_active/sort_order are enforced on every run.
    """
    slug = _slug(name)
    obj, created = Category.objects.update_or_create(
        type=type_value,
        parent=parent,
        slug=slug,
        defaults={
            "name": name.strip(),
            "is_active": True,
            "sort_order": int(sort_order),
        },
    )
    return obj, created


class Command(BaseCommand):
    help = "Seed a robust set of categories + subcategories for both MODEL and FILE types."

    def add_arguments(self, parser):
        parser.add_argument(
            "--wipe",
            action="store_true",
            help="Delete all categories before seeding (DANGEROUS).",
        )

    @transaction.atomic
    def handle(self, *args, **options):
        wipe = bool(options.get("wipe"))

        if wipe:
            self.stdout.write(self.style.WARNING("WIPING ALL Category rows..."))
            Category.objects.all().delete()

        created = 0
        updated = 0

        # Seed for both types using the same tree
        for type_value in (Category.CategoryType.MODEL, Category.CategoryType.FILE):
            type_prefix = "MODEL" if type_value == Category.CategoryType.MODEL else "FILE"

            for i, node in enumerate(CATEGORY_TREE, start=1):
                root_name = str(node["name"]).strip()
                children = list(node.get("children") or [])

                root_obj, root_created = _upsert_category(
                    type_value=type_value,
                    name=root_name,
                    parent=None,
                    sort_order=i,
                )
                if root_created:
                    created += 1
                else:
                    updated += 1

                # children
                for j, child_name in enumerate(children, start=1):
                    child_obj, child_created = _upsert_category(
                        type_value=type_value,
                        name=str(child_name).strip(),
                        parent=root_obj,
                        sort_order=j,
                    )
                    if child_created:
                        created += 1
                    else:
                        updated += 1

            self.stdout.write(self.style.SUCCESS(f"{type_prefix}: seeded OK"))

        total = Category.objects.count()
        roots = Category.objects.filter(parent__isnull=True).count()
        subs = total - roots

        self.stdout.write(
            self.style.SUCCESS(
                f"Seed complete. wipe={wipe} created={created} updated={updated} total={total} roots={roots} subs={subs}"
            )
        )
```

products\management\commands\seed_demo_products.py

```
# products/management/commands/seed_demo_products.py
from __future__ import annotations

import base64
from decimal import Decimal
from typing import Optional

from django.contrib.auth import get_user_model
from django.core.files.base import ContentFile
from django.core.management.base import BaseCommand
from django.db import transaction
from django.utils.text import slugify

from products.models import Product, ProductImage

# Optional models in your project
try:
    from payments.models import SellerStripeAccount
except Exception:  # pragma: no cover
    SellerStripeAccount = None  # type: ignore

try:
    from catalog.models import Category
except Exception:  # pragma: no cover
    Category = None  # type: ignore


# 1x1 PNG (transparent) to attach to ImageField so UI renders
_ONE_BY_ONE_PNG = base64.b64decode(

b"iVBORw0KGgoAAAANSUhEUgAAAAEAAAABCAQAAAC1HAwCAAAAC0lEQVR4nGNgYAAAAAMAASsJTYQAAAAASUVORK
5CYII="
)


class Command(BaseCommand):
    help = "Seed demo sellers + categories + products + images for local smoke testing."

    def add_arguments(self, parser):
        parser.add_argument(
            "--products",
            type=int,
            default=12,
            help="Total products to create (default 12). Will split evenly between MODEL and FILE when possible.",
        )
        parser.add_argument(
            "--reset",
            action="store_true",
            help="Delete existing demo users/products created by this command before reseeding.",
        )

    def handle(self, *args, **options):
        total_products: int = max(2, int(options["products"]))
        reset: bool = bool(options["reset"])

        with transaction.atomic():
            seller_ready = self._get_or_create_user(
                username="demo_seller_ready",
                email="demo_seller_ready@example.com",
                password="demo12345",
                is_staff=True,
            )
            seller_not_ready = self._get_or_create_user(
                username="demo_seller_not_ready",
                email="demo_seller_not_ready@example.com",
                password="demo12345",
                is_staff=True,
            )

            if reset:
                self._reset_demo_data({seller_ready.id, seller_not_ready.id})

            self._ensure_seller_ready_flag(seller_ready)

            model_cat = self._get_or_create_category_for_kind("MODEL", name="Demo Models")
            file_cat = self._get_or_create_category_for_kind("FILE", name="Demo Files")

            # Split products roughly evenly
            n_model = total_products // 2
            n_file = total_products - n_model

            created = 0
            created += self._seed_products_for_kind(
                seller=seller_ready,
                kind="MODEL",
                category=model_cat,
                count=max(1, n_model // 2),
                title_prefix="Ready Model",
                price_base=Decimal("24.99"),
                featured=True,
            )
            created += self._seed_products_for_kind(
                seller=seller_not_ready,
                kind="MODEL",
                category=model_cat,
                count=max(1, n_model - (n_model // 2)),
                title_prefix="NotReady Model",
                price_base=Decimal("19.99"),
                featured=False,
            )
            created += self._seed_products_for_kind(
                seller=seller_ready,
                kind="FILE",
                category=file_cat,
                count=max(1, n_file // 2),
                title_prefix="Ready File",
                price_base=Decimal("4.99"),
                featured=False,
            )
            created += self._seed_products_for_kind(
                seller=seller_not_ready,
                kind="FILE",
                category=file_cat,
                count=max(1, n_file - (n_file // 2)),
                title_prefix="NotReady File",
                price_base=Decimal("3.99"),
                featured=False,
            )

        self.stdout.write(self.style.SUCCESS("Seed complete."))
```

```
    self.stdout.write("Demo logins:")
    self.stdout.write("  demo_seller_ready / demo12345  (Stripe-ready where supported)")
    self.stdout.write("  demo_seller_not_ready / demo12345")
    self.stdout.write(f"Created/ensured ~{created} products + images.")

def _get_or_create_user(self, *, username: str, email: str, password: str, is_staff: bool) -> object:
    User = get_user_model()

    # Try to find by username or email
    user = User.objects.filter(username=username).first()
    if not user:
        user = User.objects.filter(email=email).first()

    if user:
        # Ensure basics
        if getattr(user, "username", None) != username:
            try:
                user.username = username  # type: ignore[attr-defined]
            except Exception:
                pass
        if getattr(user, "email", None) != email:
            try:
                user.email = email
            except Exception:
                pass
        try:
            user.is_staff = bool(is_staff)
        except Exception:
            pass
        user.save()
        return user

    # Create
    try:
        user = User.objects.create_user(username=username, email=email, password=password)
    except TypeError:
        # If custom user creation differs, fallback to minimal create + set_password
        user = User(username=username, email=email)
        user.set_password(password)
        user.save()

    try:
        user.is_staff = bool(is_staff)
        user.save()
    except Exception:
        pass

    return user

def _reset_demo_data(self, seller_ids: set[int]) -> None:
    # Delete demo products for our demo sellers
    qs = Product.objects.filter(seller_id__in=seller_ids)
    count = qs.count()
    qs.delete()
    self.stdout.write(self.style.WARNING(f"Reset: deleted {count} existing demo products."))

def _ensure_seller_ready_flag(self, seller) -> None:
if SellerStripeAccount is None:
    self.stdout.write(self.style.WARNING("SellerStripeAccount model not available; skipping ready flag."))
    return
    try:
        SellerStripeAccount.objects.update_or_create(
            user_id=seller.id,
            defaults={
                "charges_enabled": True,
                "payouts_enabled": True,
                "details_submitted": True,
            },
        )
    except Exception:
        self.stdout.write(self.style.WARNING("Could not set Stripe readiness flags (schema mismatch)."))

    def _get_or_create_category_for_kind(self, kind: str, *, name: str):
        """
        Best-effort:
        - Prefer existing categories for the requested kind/type.
        - If none exist and Category model is compatible, create a simple root category.
        """
        if Category is None:
            self.stdout.write(self.style.WARNING("Category model not available; cannot attach categories."))
            return None

        # First: try to find anything matching kind
        # Your Category probably has CategoryType MODEL/FILE stored in field `type`.
        cat = None
        try:
            cat = Category.objects.filter(type=kind).order_by("id").first()
        except Exception:
            cat = None

        if cat:
            return cat

        # Try to create a minimal category. This depends on your schema.
        data = {}
        # Common fields
        if hasattr(Category, "name"):
            data["name"] = name
        if hasattr(Category, "slug"):
            data["slug"] = slugify(name)
        if hasattr(Category, "type"):
            data["type"] = kind
        # Tree fields (optional)
        if hasattr(Category, "parent_id"):
            data["parent_id"] = None

        try:
            return Category.objects.create(**data)
        except Exception:
                        self.stdout.write(self.style.WARNING(f"Could not auto-create Category for kind={kind}; using first
available."))
            try:
                return Category.objects.order_by("id").first()
            except Exception:
                return None

    def _seed_products_for_kind(
        self,
        *,
        seller,
        kind: str,
    category,
    count: int,
    title_prefix: str,
    price_base: Decimal,
    featured: bool,
) -> int:
    created = 0
    for i in range(1, count + 1):
        title = f"{title_prefix} #{i}"
        slug = slugify(title)[:180]

        # Ensure uniqueness per seller+slug
        existing = Product.objects.filter(seller_id=seller.id, slug=slug).first()
        if existing:
            p = existing
            p.title = title
            p.kind = kind
            if category is not None:
                p.category = category
            p.price = price_base + Decimal(i)
            p.is_free = False
            p.is_active = True
            p.is_featured = bool(featured and i <= 2)
            p.is_trending = bool(i == 1)  # seed a couple manual trending flags
            p.short_description = "Seeded demo listing for UI testing."
            p.description = "This is demo data generated by seed_demo_products."
            p.save()
        else:
            kwargs = dict(
                seller_id=seller.id,
                kind=kind,
                title=title,
                slug=slug,
                short_description="Seeded demo listing for UI testing.",
                description="This is demo data generated by seed_demo_products.",
                price=price_base + Decimal(i),
                is_free=False,
                is_active=True,
                is_featured=bool(featured and i <= 2),
                is_trending=bool(i == 1),
            )
            if category is not None:
                kwargs["category"] = category
            p = Product.objects.create(**kwargs)

        # Ensure an image exists
        if not p.images.exists():
            img_name = f"demo_{p.kind.lower()}_{p.id}.png"
            content = ContentFile(_ONE_BY_ONE_PNG, name=img_name)
            ProductImage.objects.create(
                product=p,
                image=content,
                alt_text=p.title,
                is_primary=True,
                sort_order=0,
            )

        created += 1

    return created
```

dashboards\templates\dashboards\seller_dashboard.html

```
{# templates/dashboards/seller_dashboard.html #}
{% extends "base.html" %}
{% load money %}
{% block title %}Seller Dashboard · Home Craft 3D{% endblock %}


{% block content %}
<div class="container-fluid">
  <div class="alert alert-info d-flex align-items-center gap-2 mb-4" role="alert">
    <i class="bi bi-cash-coin fs-4 text-primary"></i>
    <div>
      <strong>Seller Fees:</strong> Home Craft 3D charges a 7.5% fee per sale. See <a href="/docs/seller_fees/" class="alert-link">Seller Fees &amp; Examples</a> for details.
    </div>
  </div>

  <!-- Existing header stays unchanged -->

  <div class="row g-3 mb-3">
    <div class="col-12 col-md-3">
      <div class="card h-100">
        <div class="card-body">
          <div class="text-secondary small">Available for payout</div>
          <div class="h4 mb-0">
            {% if payout_available < 0 %}
              <span class="text-danger">-${{ payout_available_abs }}</span>
            {% else %}
              ${{ payout_available }}
            {% endif %}
          </div>
          <div class="text-secondary small mt-1">
            Balance:
            {% if balance < 0 %}
              <span class="text-danger">-${{ balance_abs }}</span>
            {% else %}
              ${{ balance }}
            {% endif %}
          </div>
        </div>
      </div>
    </div>

    <div class="col-12 col-md-3">
      <div class="card h-100">
        <div class="card-body">
          <div class="text-secondary small">Net earnings (last {{ since_days }} days)</div>
          <div class="h4 mb-0">${{ net_revenue }}</div>
        </div>
      </div>
    </div>
  </div>

  <div class="card mb-3">
    <div class="card-header bg-white">
      <div class="fw-semibold">Balance Activity</div>
    </div>
    <div class="table-responsive">
      <table class="table table-sm table-hover mb-0">
        <thead class="table-light">
```

```
          <tr>
            <th>Date</th>
            <th>Reason</th>
            <th class="text-end">Amount</th>
          </tr>
        </thead>
        <tbody>
          {% for e in ledger_entries %}
            <tr>
              <td class="small">{{ e.created_at|date:"Y-m-d" }}</td>
              <td class="small">{{ e.get_reason_display }}</td>
              <td class="text-end {% if e.amount_cents < 0 %}text-danger{% else %}text-success{% endif %}">
                ${{ e.amount_cents|cents_to_dollars }}
              </td>
            </tr>
          {% empty %}
            <tr>
              <td colspan="3" class="text-center text-secondary py-3">
                No balance activity yet.
              </td>
            </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>
  </div>

<div class="container-fluid py-4 px-0">
    <div class="row gx-4">
      <div class="col-12 col-md-6 d-flex">
        <div class="card h-100 w-100">
          <div class="card-header bg-white">
            <h4 class="mb-0">Grant Free Download to User</h4>
          </div>
          <div class="card-body">
            <form method="post" class="mt-3" id="grant-free-unlock-form">
              {% csrf_token %}
              <div class="mb-3">
                <label for="id_product" class="form-label">Product</label>
                {{ grant_form.product }}
              </div>

              <div class="mb-3">
                <label for="id_username" class="form-label">Username</label>
                <div class="input-group">
                  {{ grant_form.username }}
                  <button type="button" class="btn btn-outline-secondary" id="verify-username-btn">
                    Verify
                  </button>
                </div>
                <div id="username-feedback" class="form-text text-danger"></div>
              </div>

              <div class="mb-3">
                <label for="id_user_email" class="form-label">User Email</label>
                {{ grant_form.user_email }}
              </div>

              <div class="form-check mb-3">
                {{ grant_form.send_email }}
                {{ grant_form.send_email.label_tag }}
              </div>

              <button type="submit" class="btn btn-primary">
                Grant Free Download
              </button>
            </form>
          </div>
        </div>
      </div>

      <div class="col-12 col-md-6 d-flex">
        <div class="border rounded bg-light p-4 h-100 w-100 d-flex align-items-center justify-content-center text-muted">
          <span>More seller tools coming soon…</span>
        </div>
      </div>
    </div>
  </div>

  <div class="card mb-4">
    <form method="post" id="bulk-action-form">
      {% csrf_token %}
      <div class="card-header bg-white d-flex justify-content-between align-items-center">
        <span class="fw-semibold">Your Listings</span>
        <div class="d-flex gap-2">
          <button type="submit" name="bulk_action" value="activate" class="btn btn-sm btn-success">Activate Selected</button>
          <button type="submit" name="bulk_action" value="deactivate" class="btn btn-sm btn-secondary">Deactivate Selected</button>
        </div>
      </div>
      <div class="table-responsive">
        <table class="table table-sm align-middle mb-0">
          <thead class="table-light">
            <tr>
              <th style="width:2%"><input type="checkbox" id="select-all"></th>
              <th>Title</th>
              <th>Status</th>
              <th>Checklist</th>
            </tr>
          </thead>
          <tbody>
            {% for row in listings_with_checklist %}
              {% with product=row.product checklist=row.checklist %}
              <tr>
                <td><input type="checkbox" name="selected_ids" value="{{ product.id }}"></td>
                <td>
                  <a href="{{ product.get_absolute_url }}" target="_blank">{{ product.title }}</a>
                  <div class="text-secondary small">{{ product.get_kind_display }}</div>
                </td>
                <td>
                  {% if product.is_active %}
                    <span class="badge bg-success">Active</span>
                  {% else %}
                    <span class="badge bg-secondary">Inactive</span>
                  {% endif %}
                </td>
                <td>
                  <div class="d-flex align-items-center gap-2">
                    <span title="Checklist status">
                      {% if checklist.has_image and checklist.has_specs and checklist.has_assets and checklist.is_active %}
                        <i class="bi bi-circle-fill text-success"></i>
                      {% else %}
                        <i class="bi bi-circle-fill text-secondary"></i>
                      {% endif %}
                    </span>
                    <div class="d-flex flex-row gap-3 small">
                      <span title="Image">
                              <i class="bi {% if checklist.has_image %}bi-check-circle-fill text-success{% else %}bi-x-circle-fill
    text-danger{% endif %}"></i> Image
                      </span>
                      <span title="Specs">
                              <i class="bi {% if checklist.has_specs %}bi-check-circle-fill text-success{% else %}bi-x-circle-fill
    text-danger{% endif %}"></i> Specs
                      </span>
                      <span title="Digital Asset">
                              <i class="bi {% if checklist.has_assets %}bi-check-circle-fill text-success{% else %}bi-x-circle-fill
    text-danger{% endif %}"></i> Digital Asset
                      </span>
                      <span title="Active">
                                <i class="bi {% if checklist.is_active %}bi-check-circle-fill text-success{% else %}bi-x-circle-fill
    text-danger{% endif %}"></i> Active
                      </span>
                    </div>
                  </div>
                </td>
              </tr>
              {% endwith %}
            {% empty %}
              <tr>
                <td colspan="4" class="text-center text-secondary py-3">No listings yet.</td>
              </tr>
            {% endfor %}
          </tbody>
        </table>
      </div>
    </form>
    <script>
      document.getElementById('select-all').addEventListener('change', function(e) {
        const checkboxes = document.querySelectorAll('input[name="selected_ids"]');
        for (const cb of checkboxes) { cb.checked = e.target.checked; }
      });
    </script>
  </div>

  <hr class="my-5">

  <script>
  document.getElementById('verify-username-btn').addEventListener('click', function() {
    var usernameInput = document.getElementById('id_username');
    var feedback = document.getElementById('username-feedback');
    var emailInput = document.getElementById('id_user_email');
    var username = usernameInput.value.trim();
    feedback.textContent = '';
    emailInput.value = '';
    if (!username) {
      feedback.textContent = 'Please enter a username.';
      return;
    }
    fetch('{% url "dashboards:ajax_verify_username" %}?username=' + encodeURIComponent(username))
      .then(response => response.json())
      .then(data => {
        if (data.success) {
          emailInput.value = data.email;
          feedback.textContent = '';
        } else {
          feedback.textContent = data.error || 'User not found.';
          emailInput.value = '';
        }
      })
      .catch(() => {
        feedback.textContent = 'Error verifying username.';
        emailInput.value = '';
      });
  });
  </script>

</div>
{% endblock %}
```

refunds\templates\refunds\seller_detail.html

```
{# refunds/templates/refunds/seller_detail.html #}
{% extends "base.html" %}
{% load money %}

{% block title %}Refund Request · Seller · Home Craft 3D{% endblock %}

{% block content %}
<div class="container py-4">
  <div class="d-flex justify-content-between align-items-center mb-3">
    <div>
      <h1 class="h4 mb-0">Refund request</h1>
      <div class="text-muted small">Refund #{{ rr.pk }}</div>
    </div>
    <a class="btn btn-outline-secondary btn-sm" href="{% url 'orders:refunds:seller_queue' %}">Back to queue</a>
  </div>

  <div class="card mb-3">
    <div class="card-body">
      <div class="fw-semibold">{{ rr.order_item.product.title }}</div>
      <div class="text-muted small">Order #{{ rr.order_id }}</div>

      <div class="mt-3">
        <div><strong>Status:</strong> {{ rr.get_status_display }}</div>
        <div><strong>Reason:</strong> {{ rr.get_reason_display }}</div>
        <div class="text-muted small">
          Buyer:
          {% if rr.buyer_id %}
            {{ rr.buyer.username }}
          {% else %}
            Guest ({{ rr.requester_email }})
          {% endif %}
        </div>

        {% if rr.notes %}
          <div class="mt-2"><strong>Buyer notes:</strong><br>{{ rr.notes|linebreaksbr }}</div>
        {% endif %}
      </div>

      <hr>

      <div class="text-muted small">
        Refund snapshot:
        <div>Line: ${{ rr.line_subtotal_cents_snapshot|cents_to_dollars }}</div>
        <div>Tax: ${{ rr.tax_cents_allocated_snapshot|cents_to_dollars }}</div>
        <div>Shipping: ${{ rr.shipping_cents_allocated_snapshot|cents_to_dollars }}</div>
        <div class="fw-semibold">Total: ${{ rr.total_refund_cents_snapshot|cents_to_dollars }}</div>
      </div>

      {% if rr.seller_decision_note %}
        <div class="mt-3">
          <strong>Decision note:</strong><br>{{ rr.seller_decision_note|linebreaksbr }}
        </div>
      {% endif %}

      {% if rr.stripe_refund_id %}
        <div class="mt-3">
          <span class="badge text-bg-success">Refunded</span>
          <div class="text-muted small mt-1">Stripe refund id: {{ rr.stripe_refund_id }}</div>
        </div>
      {% endif %}
    </div>
  </div>

  {% if rr.status == "requested" %}
    <div class="card mb-3">
      <div class="card-body">
        <h2 class="h6 mb-2">Approve or decline</h2>
        <p class="text-muted small mb-3">
          Add an optional note. This note is stored with your decision.
        </p>

        <form method="post" action="{% url 'refunds:seller_approve' refund_id=rr.pk %}" class="mb-2">
          {% csrf_token %}
          {{ decision_form.decision_note }}
          <button class="btn btn-success mt-2">Approve</button>
        </form>

        <form method="post" action="{% url 'refunds:seller_decline' refund_id=rr.pk %}">
          {% csrf_token %}
          {{ decision_form.decision_note }}
          <button class="btn btn-danger mt-2">Decline</button>
        </form>
      </div>
    </div>

  {% elif rr.status == "approved" %}
    <form method="post" action="{% url 'refunds:seller_trigger_refund' refund_id=rr.pk %}" class="card">
      {% csrf_token %}
      <div class="card-body">
        <div class="alert alert-warning mb-2">
          Approved. Click below to process the Stripe refund for the full snapshot amount.
        </div>
        <button class="btn btn-primary">Refund in Stripe</button>
      </div>
    </form>

  {% elif rr.status == "refunded" %}
    <div class="alert alert-success mb-0">
      Refunded via Stripe{% if rr.stripe_refund_id %} ({{ rr.stripe_refund_id }}){% endif %}.
    </div>

  {% elif rr.status == "declined" %}
    <div class="alert alert-secondary mb-0">
      Declined.
    </div>

  {% else %}
    <div class="alert alert-secondary mb-0">Status: {{ rr.get_status_display }}</div>
  {% endif %}
</div>
{% endblock %}
```

dashboards\templates\dashboards\seller_grant_free_unlock.html

```
{# dashboards/templates/dashboards/seller_grant_free_unlock.html #}
{% extends "base.html" %}
{% block title %}Grant Free Download · Seller Dashboard{% endblock %}
{% block content %}
<div class="container py-4">
  <h2>Grant Free Download</h2>
  <form method="post" class="mt-3" id="grant-free-unlock-form">
    {% csrf_token %}
    <div class="mb-3">
      <label for="id_product" class="form-label">Product</label>
      {{ form.product }}
    </div>
    <div class="mb-3">
      <label for="id_username" class="form-label">Username</label>
      <div class="input-group">
        {{ form.username }}
        <button type="button" class="btn btn-outline-secondary" id="verify-username-btn">Verify</button>
      </div>
      <div id="username-feedback" class="form-text text-danger"></div>
    </div>
    <div class="mb-3">
      <label for="id_user_email" class="form-label">User Email</label>
      {{ form.user_email }}
    </div>
    <div class="form-check mb-3">
      {{ form.send_email }}
      {{ form.send_email.label_tag }}
    </div>
    <button type="submit" class="btn btn-primary">Grant Free Download</button>
    <a href="{% url 'dashboards:seller' %}" class="btn btn-secondary ms-2">Back to Dashboard</a>
  </form>
</div>
<script>
document.getElementById('verify-username-btn').addEventListener('click', function() {
  var usernameInput = document.getElementById('id_username');
  var feedback = document.getElementById('username-feedback');
  var emailInput = document.getElementById('id_user_email');
  var username = usernameInput.value.trim();
  feedback.textContent = '';
  emailInput.value = '';
  if (!username) {
    feedback.textContent = 'Please enter a username.';
    return;
  }
  fetch('{% url "dashboards:ajax_verify_username" %}?username=' + encodeURIComponent(username))
    .then(response => response.json())
    .then(data => {
      if (data.success) {
        emailInput.value = data.email;
        feedback.textContent = '';
      } else {
        feedback.textContent = data.error || 'User not found.';
        emailInput.value = '';
      }
    })
    .catch(() => {
      feedback.textContent = 'Error verifying username.';
      emailInput.value = '';
    });
});
</script>
{% endblock %}
```

templates\emails\seller_new_order.html

```
{# templates/emails/refund_processed.html #}
{% extends "emails/base.html" %}
{% block content %}
  <h1 style="margin:0 0 12px;font-size:20px;">New order received</h1>
  <p style="margin:0 0 12px;">Order <strong>#{{ order_id }}</strong> has items for you to ship.</p>

  <div style="margin:16px 0;padding:16px;border:1px solid #e5e7eb;border-radius:10px;background:#f9fafb;">
    <div style="font-size:12px;color:#6b7280;margin-bottom:6px;">Fulfillment link</div>
    <a href="{{ fulfillment_link }}" style="color:#2563eb;text-decoration:none;">View and fulfill</a>
  </div>

  <h2 style="margin:16px 0 8px;font-size:16px;">Items to ship</h2>
  <ul style="padding-left:18px;margin:0;">
    {% for item in items %}
      <li style="margin:6px 0;">{{ item }}</li>
    {% endfor %}
  </ul>

  {% if shipping_lines %}
    <h2 style="margin:16px 0 8px;font-size:16px;">Shipping address</h2>
    <div style="white-space:pre-line;">{{ shipping_lines }}</div>
  {% endif %}
{% endblock %}
```

orders\templates\orders\seller_orders_detail.html

```
{# orders/templates/orders/seller_orders_detail.html #}
{% extends "base.html" %}
{% block title %}Order · Home Craft 3D{% endblock %}

{% block content %}
<div class="container-fluid py-3">
  <div class="d-flex align-items-center justify-content-between mb-3">
    <div>
      <h1 class="h4 mb-0">Order #{{ order.id }}</h1>
      <div class="text-secondary small">{{ order.get_status_display }}{% if order.paid_at %} · Paid {{ order.paid_at }}{% endif %}</div>
    </div>
    <a class="btn btn-outline-dark" href="{% url 'orders:seller_orders_list' %}">Back</a>
  </div>

  <!-- Shipping Address -->
  {% if order.shipping_name %}
  <div class="card mb-3">
    <div class="card-header bg-light">
      <h5 class="mb-0">Shipping Address</h5>
    </div>
    <div class="card-body">
      <p class="mb-1"><strong>{{ order.shipping_name }}</strong></p>
      <p class="mb-1">{{ order.shipping_line1 }}{% if order.shipping_line2 %}<br>{{ order.shipping_line2 }}{% endif %}</p>
      <p class="mb-1">{{ order.shipping_city }}, {{ order.shipping_state }} {{ order.shipping_postal_code }}</p>
      <p class="mb-0">{{ order.shipping_country }}</p>
      {% if order.shipping_phone %}<p class="mb-0 small text-secondary">{{ order.shipping_phone }}</p>{% endif %}
    </div>
  </div>
  {% endif %}

  <!-- Items Table -->
  <div class="card">
    <div class="table-responsive">
      <table class="table table-hover align-middle mb-0">
        <thead class="table-light">
          <tr>
            <th>Item</th>
            <th class="text-end">Qty</th>
            <th class="text-end">Unit</th>
            <th class="text-end">Marketplace fee</th>
            <th class="text-end">Seller net</th>
            <th>Status</th>
            <th>Tracking</th>
          </tr>
        </thead>
        <tbody>
          {% for it in seller_items %}
            <tr>
              <td class="fw-semibold">{{ it.product.title }}</td>
              <td class="text-end">{{ it.quantity }}</td>
              <td class="text-end">${{ it.unit_price_cents|add:0 }}</td>
              <td class="text-end">${{ it.marketplace_fee_cents|add:0 }}</td>
              <td class="text-end fw-semibold">${{ it.seller_net_cents|add:0 }}</td>
              <td>
                {% if it.requires_shipping %}
                  <span class="badge {% if it.fulfillment_status == 'shipped' %}bg-success{% elif it.fulfillment_status ==
'pending' %}bg-warning{% else %}bg-secondary{% endif %}">
                    {{ it.get_fulfillment_status_display }}
                  </span>
                {% else %}
                  <span class="badge bg-info">Digital</span>
                {% endif %}
              </td>
              <td>
                {% if it.requires_shipping %}
                  {% if it.fulfillment_status == 'pending' %}
                    <!-- Show form to mark as shipped -->
                     <form method="post" action="{% url 'orders:mark_item_shipped' order.id it.id %}" class="d-flex gap-2
flex-wrap">
                      {% csrf_token %}
                      <select name="carrier" class="form-select form-select-sm" style="max-width: 160px;">
                        <option value="">Carrier (optional)</option>
                        <option>USPS</option>
                        <option>UPS</option>
                        <option>FedEx</option>
                        <option>DHL</option>
                        <option>Other</option>
                      </select>
                      <input type="text" name="carrier_other" class="form-control form-control-sm d-none"
placeholder="Enter carrier" style="max-width: 160px;" data-carrier-other>
                      <input type="text" name="tracking_number" class="form-control form-control-sm"
placeholder="Tracking #" style="max-width: 160px;">
                      <button type="submit" class="btn btn-sm btn-primary">Ship</button>
                    </form>
                  {% elif it.fulfillment_status == 'shipped' %}
                    <div class="small mb-1">
                      {% if it.carrier %}<span class="text-monospace">{{ it.carrier }}</span>{% endif %}
                      {% if it.tracking_number %}
                        {% if it.carrier %} · {% endif %}
                        <span class="text-monospace">{{ it.tracking_number }}</span>
                        <a class="ms-2" href="https://trackingmore.com/track/?tracknumber={{
it.tracking_number|urlencode }}" target="_blank" rel="noopener">Track</a>
                        <button class="btn btn-sm btn-outline-secondary ms-2" type="button" data-copy-tracking="{{
it.tracking_number }}">Copy</button>
                      {% endif %}
                    </div>
                    <form method="post" action="{% url 'orders:mark_item_delivered' order.id it.id %}" class="d-inline">
                      {% csrf_token %}
                      <button type="submit" class="btn btn-sm btn-outline-dark">Mark delivered</button>
                    </form>
                  {% elif it.carrier or it.tracking_number %}
                    <div class="small">
                      {% if it.carrier %}<span class="text-monospace">{{ it.carrier }}</span>{% endif %}
                      {% if it.tracking_number %}
                        {% if it.carrier %} · {% endif %}
                        <span class="text-monospace">{{ it.tracking_number }}</span>
                        <a class="ms-2" href="https://trackingmore.com/track/?tracknumber={{
it.tracking_number|urlencode }}" target="_blank" rel="noopener">Track</a>
                        <button class="btn btn-sm btn-outline-secondary ms-2" type="button" data-copy-tracking="{{
it.tracking_number }}">Copy</button>
                      {% endif %}
                    </div>
                  {% else %}
                    <small class="text-secondary">N/A</small>
                  {% endif %}
                {% endif %}
              </td>
```

```
            </tr>
          {% empty %}
            <tr>
              <td colspan="7" class="text-center text-secondary py-4">No items for this seller.</td>
            </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>
  </div>
</div>

<script>
  document.addEventListener('click', function (event) {
    const btn = event.target.closest('[data-copy-tracking]');
    if (!btn) return;

    const value = btn.getAttribute('data-copy-tracking') || '';
    if (!value) return;

    navigator.clipboard.writeText(value).then(function () {
      const original = btn.textContent;
      btn.textContent = 'Copied';
      setTimeout(function () {
        btn.textContent = original;
      }, 1200);
    });
  });

  document.addEventListener('change', function (event) {
    const select = event.target.closest('select[name="carrier"]');
    if (!select) return;
    const form = select.closest('form');
    if (!form) return;
    const otherInput = form.querySelector('[data-carrier-other]');
    if (!otherInput) return;

    if (select.value === 'Other') {
      otherInput.classList.remove('d-none');
      otherInput.focus();
    } else {
      otherInput.classList.add('d-none');
      otherInput.value = '';
    }
  });

  document.addEventListener('submit', function (event) {
    const form = event.target.closest('form');
    if (!form) return;
    const select = form.querySelector('select[name="carrier"]');
    const otherInput = form.querySelector('[data-carrier-other]');
    if (!select || !otherInput) return;
    if (select.value === 'Other' && otherInput.value.trim()) {
      select.value = otherInput.value.trim();
    }
  });
</script>
{% endblock %}
```

orders\templates\orders\seller_orders_list.html

```
{# orders/templates/orders/seller_orders_list.html #}
{% extends "base.html" %}
{% load money %}
{% block title %}Fulfillment · Home Craft 3D{% endblock %}

{% block content %}
<div class="container-fluid py-3">
  <div class="d-flex align-items-center justify-content-between mb-3">
    <h1 class="h4 mb-0">Fulfillment</h1>
  </div>

  <div class="alert alert-info d-flex flex-wrap align-items-center gap-2">
    <div>
      <div class="fw-semibold">Order & Tracking helpers</div>
      <div class="small">Click an order to add tracking. We’ll email the buyer once you mark it shipped.</div>
    </div>
  </div>

  <div class="card">
    <div class="table-responsive">
      <table class="table table-hover align-middle mb-0">
        <thead class="table-light">
          <tr>
            <th>Order</th>
            <th>Item</th>
            <th class="text-end">Qty</th>
            <th class="text-end">Unit</th>
            <th class="text-end">Line</th>
            <th class="text-end"></th>
          </tr>
        </thead>
        <tbody>
          {% for it in items %}
            <tr>
              <td class="fw-semibold">#{{ it.order_id }}</td>
              <td>
                <div class="fw-semibold">{{ it.product.title }}</div>
                <div class="text-secondary small">
                  {% if it.is_digital %}Digital{% else %}Physical{% endif %}
                </div>
              </td>
              <td class="text-end">{{ it.quantity }}</td>
              <td class="text-end">${{ it.unit_price_cents|cents_to_dollars }}</td>
              <td class="text-end fw-semibold">
                ${{ it.line_total_cents|cents_to_dollars }}
              </td>
              <td class="text-end">
                <a class="btn btn-sm btn-outline-dark" href="{% url 'orders:seller_order_detail' it.order_id %}">View</a>
              </td>
            </tr>
          {% empty %}
            <tr>
              <td colspan="6" class="text-center text-secondary py-4">No paid orders yet.</td>
            </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>
  </div>
```

```
</div>
{% endblock %}
```

refunds\templates\refunds\seller_queue.html

```
{# refunds/templates/refunds/seller_detail.html #}
{% extends "base.html" %}
{% load money %}

{% block title %}Refund Requests · Seller · Home Craft 3D{% endblock %}

{% block content %}
<div class="container py-4">
  <div class="d-flex justify-content-between align-items-center mb-3">
    <h1 class="h4 mb-0">Refund requests</h1>
    <a class="btn btn-outline-secondary btn-sm" href="{% url 'orders:seller_orders_list' %}">Fulfillment</a>
  </div>

  {% if refunds %}
    <div class="list-group">
      {% for rr in refunds %}
        <a class="list-group-item list-group-item-action" href="{% url 'refunds:seller_detail' refund_id=rr.pk %}">
          <div class="d-flex justify-content-between gap-3">
            <div>
              <div class="fw-semibold">{{ rr.order_item.product.title }}</div>
              <div class="text-muted small">
                {{ rr.get_status_display }} · {{ rr.get_reason_display }} · Order #{{ rr.order_id }}
              </div>
              <div class="text-muted small">
                Buyer:
                {% if rr.buyer_id %}
                  {{ rr.buyer.username }}
                {% else %}
                  Guest ({{ rr.requester_email }})
                {% endif %}
              </div>
            </div>
            <div class="text-end">
              <div class="fw-semibold">${{ rr.total_refund_cents_snapshot|cents_to_dollars }}</div>
              <div class="text-muted small">{{ rr.created_at|date:"M j, Y" }}</div>
            </div>
          </div>
        </a>
      {% endfor %}
    </div>
  {% else %}
    <div class="alert alert-info mb-0">No refund requests.</div>
  {% endif %}
</div>
{% endblock %}
```

reviews\templates\reviews\seller_review_form.html

```
{# reviews/templates/reviews/seller_review_form.html #}
{% extends "base.html" %}
{% block title %}Rate Seller · Home Craft 3D{% endblock %}

{% block content %}
<div class="container py-4" style="max-width: 760px;">
  <div class="d-flex align-items-center justify-content-between mb-3">
    <div>
      <h1 class="h4 mb-1">Rate Seller</h1>
      <div class="text-secondary small">Order #{{ order.id }}</div>
    </div>
    <a class="btn btn-outline-dark" href="{% url 'orders:detail' order.id %}">Back to order</a>
  </div>

  <div class="card">
    <div class="card-body">
      <form method="post" novalidate>
        {% csrf_token %}

        {% if form.non_field_errors %}
          <div class="alert alert-danger">{{ form.non_field_errors }}</div>
        {% endif %}

        <div class="mb-3">
          <label class="form-label">Rating</label>
          {{ form.rating }}
          {% if form.rating.errors %}<div class="text-danger small">{{ form.rating.errors }}</div>{% endif %}
        </div>

        <div class="mb-3">
          <label class="form-label">Title (optional)</label>
          {{ form.title }}
          {% if form.title.errors %}<div class="text-danger small">{{ form.title.errors }}</div>{% endif %}
        </div>

        <div class="mb-3">
          <label class="form-label">Feedback (optional)</label>
          {{ form.body }}
          {% if form.body.errors %}<div class="text-danger small">{{ form.body.errors }}</div>{% endif %}
        </div>

        <button class="btn btn-dark" type="submit">Post Seller Rating</button>
      </form>
    </div>
  </div>
</div>
{% endblock %}
```

products\templates\products\seller_shop.html

```
{# products/templates/products/seller_shop.html #}
{% extends "base.html" %}
{% block title %}{{ seller.username }}'s Shop · Home Craft 3D{% endblock %}
{% block meta_title %}{{ seller.username }}'s Shop · Home Craft 3D{% endblock %}
{% block og_title %}{{ product.title }} · Home Craft 3D{% endblock %}
{% block og_description %}{{ product.short_description }}{% endblock %}
{% block meta_description %}Explore {{ seller.username }}'s 3D models and files on Home Craft 3D.{% endblock %}

{% block content %}
<div class="container py-4">
  <div class="row g-4">
    <!-- Seller Info -->
    <div class="col-12 col-lg-4">
      <div class="card shadow-sm">
        <div class="card-body text-center">
          {% if profile and profile.avatar %}
            <img src="{{ profile.avatar.url }}" alt="{{ profile.public_seller_name }} avatar" class="rounded-circle mb-2" style="width: 96px; height: 96px; object-fit: cover;">
          {% endif %}
          <div class="h5 mb-1">{{ profile.public_seller_name|default:seller.username }}</div>
          {% if profile and profile.city %}
            <div class="text-secondary small mb-1">{{ profile.city }}{% if profile.state %}, {{ profile.get_state_display }}{% endif %}</div>
          {% endif %}
          <div class="text-secondary small mb-1">Member since {{ seller.date_joined|date:"F Y" }}</div>
          {% if profile and profile.bio %}
            <div class="mb-2 small">{{ profile.bio }}</div>
          {% endif %}
          <div class="text-secondary small mb-2">
            {{ products|length }} product{{ products|pluralize }}
          </div>
          {% if profile and profile.website or profile.social_instagram or profile.social_twitter or profile.social_facebook or profile.social_youtube %}
            <div class="mb-2">
              {% if profile.website %}
                <a href="{{ profile.website }}" class="me-2" target="_blank" rel="noopener"><i class="bi bi-globe"></i> Website</a>
              {% endif %}
              {% if profile.social_instagram %}
                <a href="{{ profile.social_instagram }}" class="me-2" target="_blank" rel="noopener"><i class="bi bi-instagram"></i></a>
              {% endif %}
              {% if profile.social_twitter %}
                <a href="{{ profile.social_twitter }}" class="me-2" target="_blank" rel="noopener"><i class="bi bi-twitter-x"></i></a>
              {% endif %}
              {% if profile.social_facebook %}
                <a href="{{ profile.social_facebook }}" class="me-2" target="_blank" rel="noopener"><i class="bi bi-facebook"></i></a>
              {% endif %}
              {% if profile.social_youtube %}
                <a href="{{ profile.social_youtube }}" class="me-2" target="_blank" rel="noopener"><i class="bi bi-youtube"></i></a>
              {% endif %}
            </div>
          {% endif %}
          {% if seller_review_count > 0 %}
            <div class="mb-3">
```

```
              <div class="h6 mb-1">★ {{ seller_avg_rating|floatformat:1 }}/5</div>
                <div class="text-secondary small">{{ seller_review_count }} seller rating{{ seller_review_count|pluralize }}</div>
            </div>
          {% endif %}
          {% if recent_reviews %}
            <hr>
            <div class="text-start">
              <div class="small fw-semibold mb-2">Recent Feedback</div>
              {% for review in recent_reviews %}
                <div class="border-top pt-2 mb-2">
                  <div class="d-flex justify-content-between align-items-center mb-1">
                    <div class="small fw-semibold">{{ review.buyer.username }}</div>
                    <span class="badge text-bg-dark">{{ review.rating }}/5</span>
                  </div>
                  {% if review.title %}
                    <div class="small fw-semibold">{{ review.title }}</div>
                  {% endif %}
                  {% if review.body %}
                    <div class="small text-secondary">{{ review.body|truncatewords:20 }}</div>
                  {% endif %}
                </div>
              {% endfor %}
            </div>
          {% endif %}
        </div>
      </div>
    </div>

    <!-- Products Grid -->
    <div class="col-12 col-lg-8">
      <h2 class="h4 mb-3">Products</h2>

      {% if products %}
        <div class="row g-3">
          {% for p in products %}
            <div class="col-12 col-sm-6">
              {% include "products/_product_card.html" with p=p show_add_to_cart=True show_category=False %}
            </div>
          {% endfor %}
        </div>
      {% else %}
        <div class="alert alert-info">This seller hasn't listed any products yet.</div>
      {% endif %}
    </div>
  </div>
</div>
{% endblock %}
```

orders\management\commands\send_download_reminders.py

```
# orders/management/commands/send_download_reminders.py
from __future__ import annotations

from datetime import timedelta

from django.core.management.base import BaseCommand
from django.utils import timezone

from orders.models import Order, OrderEvent, _send_download_reminder_email


class Command(BaseCommand):
    help = "Send download reminder emails for paid digital orders."

    def add_arguments(self, parser):
        parser.add_argument(
            "--days",
            type=int,
            default=3,
            help="Minimum age in days since payment before sending reminders (default: 3).",
        )
        parser.add_argument(
            "--limit",
            type=int,
            default=200,
            help="Maximum number of reminders to send in one run (default: 200).",
        )
        parser.add_argument(
            "--dry-run",
            action="store_true",
            help="Only show how many reminders would be sent.",
        )

    def handle(self, *args, **options):
        days = int(options.get("days") or 3)
        limit = int(options.get("limit") or 200)
        dry_run = bool(options.get("dry_run"))

        cutoff = timezone.now() - timedelta(days=days)

        qs = (
            Order.objects.filter(status=Order.Status.PAID, paid_at__lte=cutoff)
            .filter(kind__in=[Order.Kind.DIGITAL, Order.Kind.MIXED])
            .order_by("paid_at")
        )

        candidates = []
        for order in qs:
            if not order.items.filter(is_digital=True).exists():  # type: ignore[attr-defined]
                continue

            already_sent = order.events.filter(  # type: ignore[attr-defined]
                type=OrderEvent.Type.WARNING,
                message="Download reminder sent",
            ).exists()
            if already_sent:
                continue

            candidates.append(order)
            if len(candidates) >= limit:
                break

        if dry_run:
            self.stdout.write(f"{len(candidates)} reminder(s) would be sent.")
            return

        sent = 0
        for order in candidates:
            if _send_download_reminder_email(order):
                OrderEvent.objects.create(
                    order=order,
                    type=OrderEvent.Type.WARNING,
                    message="Download reminder sent",
                )
                sent += 1

        self.stdout.write(f"Sent {sent} download reminder(s).")
```

reviews\services.py

```
# reviews/services.py
from __future__ import annotations

from django.core.exceptions import PermissionDenied

from orders.models import Order, OrderItem


def get_reviewable_order_item_or_403(*, user, order_item_id: int) -> OrderItem:
    """Return an OrderItem the user is allowed to review, else raise PermissionDenied.

    Rules:
    - User must be authenticated.
    - OrderItem must exist.
    - The parent order must be PAID.
    - The order's buyer must be the requesting user.

    Notes:
    - One review per OrderItem is enforced in models via OneToOneField.
    """

    if not user or not getattr(user, "is_authenticated", False):
        raise PermissionDenied("Authentication required")

    item = (
        OrderItem.objects.select_related("order", "product")
        .filter(id=order_item_id)
        .first()
    )
    if not item:
        raise PermissionDenied("Order item not found")

    order = item.order
    if not order or order.status != Order.Status.PAID:
        raise PermissionDenied("Order not paid")

    if getattr(order, "buyer_id", None) != getattr(user, "id", None):
        raise PermissionDenied("Not your order")

    return item


def get_rateable_seller_order_or_403(*, user, order_id: int, seller_id: int) -> Order:
    """Return an Order the user can use to rate a seller, else raise PermissionDenied.

    Rules:
    - User must be authenticated.
    - Order must exist and be PAID.
    - Order.buyer must be the requesting user.
    - Order must include at least one OrderItem with product.seller_id == seller_id.
    """

    if not user or not getattr(user, "is_authenticated", False):
        raise PermissionDenied("Authentication required")

    order = Order.objects.filter(id=order_id).first()
    if not order:
        raise PermissionDenied("Order not found")
```

```
if order.status != Order.Status.PAID:
    raise PermissionDenied("Order not paid")

if getattr(order, "buyer_id", None) != getattr(user, "id", None):
    raise PermissionDenied("Not your order")

has_seller_item = (
    OrderItem.objects.filter(order_id=order.id, product__seller_id=seller_id)
    .exists()
)
if not has_seller_item:
    raise PermissionDenied("Seller not in this order")

return order
```

payments\services_fee_waiver.py

```python
# payments/services_fee_waiver.py
from __future__ import annotations

from decimal import Decimal

from django.utils import timezone

from core.config import get_site_config
from payments.models import SellerFeeWaiver


def get_effective_marketplace_sales_percent_for_seller(*, seller_user) -> Decimal:
    """
    Returns the marketplace sales percent for a seller, after applying fee-waiver logic.

    - If waiver enabled and seller has an active waiver window => 0%
    - Else => SiteConfig.marketplace_sales_percent
    """
    cfg = get_site_config()
    default_pct = Decimal(getattr(cfg, "marketplace_sales_percent", Decimal("0.00")) or Decimal("0.00"))

    waiver_enabled = bool(getattr(cfg, "seller_fee_waiver_enabled", True))
    if not waiver_enabled:
        return default_pct

    waiver = SellerFeeWaiver.objects.filter(user=seller_user).first()
    if waiver and waiver.is_active:
        return Decimal("0.00")

    return default_pct


def ensure_fee_waiver_for_new_seller(*, seller_user) -> None:
    """
    Idempotently create a waiver record for a seller based on SiteConfig.seller_fee_waiver_days.
    """
    cfg = get_site_config()
    if not bool(getattr(cfg, "seller_fee_waiver_enabled", True)):
        return

    try:
        days = int(getattr(cfg, "seller_fee_waiver_days", 30) or 30)
    except Exception:
        days = 30

    days = max(0, min(days, 365))
    # Only create if missing
    if not SellerFeeWaiver.objects.filter(user=seller_user).exists():
        SellerFeeWaiver.ensure_for_seller(user=seller_user, waiver_days=days)
```

accounts\templates\accounts\setup_2fa.html

```
{# templates/accounts/setup_2fa.html #}
{% extends 'accounts/base_accounts.html' %}
{% load static %}

{% block title %}Setup Two-Factor Authentication{% endblock %}

{% block content %}
<div class="container mt-5">
    <div class="row justify-content-center">
        <div class="col-md-8">
            <div class="card">
                <div class="card-header bg-primary text-white">
                    <h4 class="mb-0">Setup Two-Factor Authentication</h4>
                </div>
                <div class="card-body">
                    <div class="alert alert-info">
                        <strong>Two-Factor Authentication (2FA)</strong> adds an extra layer of security to your account.
                        You'll need an authenticator app like Google Authenticator, Microsoft Authenticator, or Authy.
                    </div>

                    <div class="row mb-4">
                        <div class="col-md-6">
                            <h5 class="mb-3">Step 1: Scan QR Code</h5>
                            <p>Open your authenticator app and scan this QR code:</p>
                            <div class="text-center">
                                {% if qr_code %}
                                                  <img src="data:image/png;base64,{{ qr_code }}" alt="QR Code" class="img-fluid"
style="max-width: 300px;">
                                {% endif %}
                            </div>
                        </div>

                        <div class="col-md-6">
                            <h5 class="mb-3">Step 2: Manual Entry</h5>
                            <p>Or manually enter this code in your authenticator app:</p>
                            <div class="alert alert-light border">
                                <code style="font-size: 14px; word-break: break-all;">{{ device.key }}</code>
                            </div>
                            <p class="text-muted small">Account: {{ user.email }}</p>
                        </div>
                    </div>

                    <hr>

                    <h5 class="mb-3">Step 3: Backup Recovery Codes</h5>
                    <p class="text-danger">
                        <strong>Save these codes in a safe place.</strong> If you lose access to your authenticator app,
                        you can use these codes to regain access to your account.
                    </p>

                    <div class="alert alert-warning">
                        <div class="row" id="backup-codes">
                            {% for code in backup_codes %}
                                <div class="col-md-6 mb-2">
                                    <code>{{ code }}</code>
                                </div>
                            {% endfor %}
                        </div>
                    </div>
                    <button class="btn btn-sm btn-outline-secondary" onclick="copyBackupCodes()">
                        <i class="bi bi-clipboard"></i> Copy Codes
                    </button>
                    <button class="btn btn-sm btn-outline-secondary" onclick="printBackupCodes()">
                        <i class="bi bi-printer"></i> Print Codes
                    </button>

                    <hr>

                    <h5 class="mb-3">Step 4: Verify Token</h5>
                    <p>Enter the 6-digit code from your authenticator app to confirm setup:</p>

                    <form method="post">
                        {% csrf_token %}
                        <div class="mb-3">
                            <input type="text"
                                   name="token"
                                   class="form-control form-control-lg text-center"
                                   placeholder="000000"
                                   maxlength="6"
                                   inputmode="numeric"
                                   autocomplete="off"
                                   required>
                        </div>
                        <button type="submit" class="btn btn-primary btn-lg w-100">
                            Verify & Enable 2FA
                        </button>
                    </form>
                </div>
            </div>
        </div>
    </div>
</div>

<script>
function copyBackupCodes() {
    const codes = document.querySelectorAll('#backup-codes code');
    const text = Array.from(codes).map(c => c.textContent).join('\n');
    navigator.clipboard.writeText(text).then(() => {
        alert('Backup codes copied to clipboard!');
    });
}

function printBackupCodes() {
    const printWindow = window.open('', '', 'height=400,width=600');
    const codes = document.querySelectorAll('#backup-codes code');
    const codeList = Array.from(codes).map(c => c.textContent).join('<br>');

    printWindow.document.write('<html><head><title>2FA Backup Codes</title></head><body>');
    printWindow.document.write('<h2>Two-Factor Authentication Backup Codes</h2>');
    printWindow.document.write('<p>Save these codes in a safe place:</p>');
    printWindow.document.write('<pre style="font-size: 14px;">' + codeList + '</pre>');
    printWindow.document.write('</body></html>');
    printWindow.document.close();
    printWindow.print();
}
</script>
{% endblock %}
```

templates\partials\sidebar.html

```
{# templates/partials/sidebar.html #}
{% comment %}
Sidebar router.

- Dashboard pages -> dashboard action sidebar
- Everything else -> storefront category sidebar

We check `app_name` because `namespace` is often empty unless you explicitly pass a namespace in include().
{% endcomment %}

{% if request.resolver_match and request.resolver_match.app_name == "dashboards" %}
  {% include "partials/sidebar_dashboard.html" %}
{% else %}
  {% include "partials/sidebar_store.html" %}
{% endif %}
```

templates\partials\sidebar_dashboard.html

```
{# templates/partials/sidebar_dashboard.html #}
<div class="p-3">

  <div class="d-flex align-items-center justify-content-between mb-2">
    <div class="text-uppercase small text-secondary fw-semibold">
      <i class="bi bi-speedometer2 me-1"></i>Dashboard
    </div>
    <a class="text-decoration-none small" href="{% url 'core:home' %}">
      <i class="bi bi-arrow-left"></i> Store
    </a>
  </div>

  <ul class="list-unstyled mb-4">
    <li class="mb-1">
      <a class="d-flex align-items-center gap-2 py-2 px-2 rounded text-decoration-none text-dark hc3-tree-row"
         href="{% url 'dashboards:home' %}">
        <i class="bi bi-house-door"></i>
        <span>Dashboard Home</span>
      </a>
    </li>
  </ul>

  <div class="text-uppercase small text-secondary fw-semibold mb-2">
    <i class="bi bi-person me-1"></i>Consumer
  </div>

  <ul class="list-unstyled mb-4">
    <li class="mb-1">
      <a class="d-flex align-items-center gap-2 py-2 px-2 rounded text-decoration-none text-dark hc3-tree-row"
         href="{% url 'dashboards:consumer' %}">
        <i class="bi bi-bar-chart"></i>
        <span>Overview</span>
      </a>
    </li>

    <li class="mb-1">
      <a class="d-flex align-items-center gap-2 py-2 px-2 rounded text-decoration-none text-dark hc3-tree-row"
         href="{% url 'orders:my_orders' %}">
        <i class="bi bi-bag-check"></i>
        <span>My Orders</span>
      </a>
    </li>
  </ul>

  {% if user_is_seller %}
    <div class="text-uppercase small text-secondary fw-semibold mb-2">
      <i class="bi bi-shop me-1"></i>Seller
    </div>

    <ul class="list-unstyled mb-4">
      <li class="mb-1">
        <a class="d-flex align-items-center gap-2 py-2 px-2 rounded text-decoration-none text-dark hc3-tree-row"
           href="{% url 'dashboards:seller' %}">
          <i class="bi bi-graph-up"></i>
          <span>Seller Overview</span>
        </a>
      </li>

      <li class="mb-1">
        <a class="d-flex align-items-center gap-2 py-2 px-2 rounded text-decoration-none text-dark hc3-tree-row"
           href="{% url 'products:seller_list' %}">
          <i class="bi bi-box"></i>
          <span>My Listings</span>
        </a>
      </li>

      <li class="mb-1">
        <a class="d-flex align-items-center gap-2 py-2 px-2 rounded text-decoration-none text-dark hc3-tree-row ms-3"
           href="{% url 'products:seller_shop' user.id %}">
          <i class="bi bi-shop"></i>
          <span>My Storefront</span>
        </a>
      </li>

      <li class="mb-1">
        <a class="d-flex align-items-center gap-2 py-2 px-2 rounded text-decoration-none text-dark hc3-tree-row"
           href="{% url 'orders:seller_orders_list' %}">
          <i class="bi bi-truck"></i>
          <span>Fulfillment</span>
        </a>
      </li>

      <li class="mb-1">
        <a class="d-flex align-items-center gap-2 py-2 px-2 rounded text-decoration-none text-dark hc3-tree-row"
           href="{% url 'payments:connect_status' %}">
          <i class="bi bi-credit-card-2-front"></i>
          <span>Stripe Payouts</span>
          {% if seller_stripe_ready is not None and not seller_stripe_ready %}
            <span class="ms-auto badge text-bg-warning text-dark">Action</span>
          {% endif %}
        </a>
      </li>
    </ul>
  {% endif %}

  {% if user_is_owner %}
    <div class="text-uppercase small text-secondary fw-semibold mb-2">
      <i class="bi bi-shield-lock me-1"></i>Admin
    </div>

    <ul class="list-unstyled mb-0">
      <li class="mb-1">
        <a class="d-flex align-items-center gap-2 py-2 px-2 rounded text-decoration-none text-dark hc3-tree-row"
           href="{% url 'dashboards:admin' %}">
          <i class="bi bi-clipboard-data"></i>
          <span>Admin Overview</span>
        </a>
      </li>

      <li class="mb-1">
        <a class="d-flex align-items-center gap-2 py-2 px-2 rounded text-decoration-none text-dark hc3-tree-row"
           href="{% url 'dashboards:admin_settings' %}">
          <i class="bi bi-sliders"></i>
          <span>Admin Settings</span>
        </a>
      </li>

      <li class="mb-1">
        <a class="d-flex align-items-center gap-2 py-2 px-2 rounded text-decoration-none text-dark hc3-tree-row"
           href="{% url 'admin:index' %}">
```

```
          <i class="bi bi-gear"></i>
          <span>Django Admin</span>
        </a>
      </li>
    </ul>
  {% endif %}

</div>
```

templates\partials\sidebar_store.html

```
{# templates/partials/sidebar_store.html #}
{% load static %}

<div class="p-3 hc3-sidebar">

  <div class="text-uppercase small text-secondary fw-semibold mb-2">
    <i class="bi bi-compass me-1"></i>Explore
  </div>

  <ul class="list-unstyled mb-3">
    <li class="mb-1">
      <a class="hc3-side-link" href="{% url 'products:list' %}">
        <i class="bi bi-grid"></i>
        <span>All Products</span>
      </a>
    </li>
    <li class="mb-1">
      <a class="hc3-side-link" href="{% url 'products:list' %}?sort=trending">
        <i class="bi bi-fire"></i>
        <span>Trending</span>
      </a>
    </li>
    <li class="mb-1">
      <a class="hc3-side-link" href="{% url 'products:list' %}?sort=top">
        <i class="bi bi-star"></i>
        <span>Top Rated</span>
      </a>
    </li>
    <li class="mb-1">
      <a class="hc3-side-link" href="{% url 'products:list' %}?sort=new">
        <i class="bi bi-clock"></i>
        <span>New Arrivals</span>
      </a>
    </li>
  </ul>

  {# Only the filter/search is collapsible, categories always show #}
  <details class="mb-3">
    <summary class="small text-secondary fw-semibold">Filters</summary>
        <form class="mt-2 d-grid gap-2" method="get" action="{% if request.path|slice:':9' == '/products' %}{{ request.path }}{% else %}{% url 'products:list' %}{% endif %}">
      <input class="form-control form-control-sm" name="q" value="{{ request.GET.q }}" placeholder="Search...">
      <div class="input-group input-group-sm mb-2">
        <span class="input-group-text"><i class="bi bi-funnel"></i></span>
        <input id="hc3SidebarFilter"
              class="form-control"
              type="search"
              placeholder="Filter categories..."
              aria-label="Filter categories">
        <button class="btn btn-outline-secondary" type="button" id="hc3SidebarFilterClear" title="Clear">
          <i class="bi bi-x-lg"></i>
        </button>
      </div>

      {% if request.path|slice:':15' == '/products/models' %}
        <input type="hidden" name="kind" value="MODEL">
      {% elif request.path|slice:':14' == '/products/files' %}
        <input type="hidden" name="kind" value="FILE">
      {% else %}
```

```
        <select class="form-select form-select-sm" name="kind">
          <option value="">All</option>
          <option value="MODEL" {% if request.GET.kind == 'MODEL' %}selected{% endif %}>3D Models</option>
          <option value="FILE" {% if request.GET.kind == 'FILE' %}selected{% endif %}>3D Files</option>
        </select>
      {% endif %}

      {% if request.path|slice:':15' != '/products/models' %}
        <select class="form-select form-select-sm" name="file_type">
          <option value="">All file types</option>
          {% for t in sidebar_file_type_options %}
            <option value="{{ t }}" {% if request.GET.file_type == t %}selected{% endif %}>.{{ t }}</option>
          {% endfor %}
        </select>
      {% endif %}

      <div class="d-flex gap-2">
        <input type="number" class="form-control form-control-sm" name="price_min" value="{{
request.GET.price_min }}" placeholder="Min price">
        <input type="number" class="form-control form-control-sm" name="price_max" value="{{
request.GET.price_max }}" placeholder="Max price">
      </div>

      <select class="form-select form-select-sm" name="complexity">
        <option value="">All levels</option>
        {% for value, label in sidebar_complexity_options %}
          <option value="{{ value }}" {% if request.GET.complexity == value %}selected{% endif %}>{{ label
}}</option>
        {% endfor %}
      </select>

      {% if request.path|slice:':15' != '/products/models' %}
        <select class="form-select form-select-sm" name="license_type">
          <option value="">All licenses</option>
          {% for value, label in sidebar_license_type_options %}
            <option value="{{ value }}" {% if request.GET.license_type == value %}selected{% endif %}>{{ label
}}</option>
          {% endfor %}
        </select>

        <label class="small d-flex align-items-center gap-2">
          <input type="checkbox" name="instant" value="1" {% if request.GET.instant == '1' %}checked{% endif
%}>
          Instant download only
        </label>
      {% endif %}

      <select class="form-select form-select-sm" name="rating_min">
        <option value="">All ratings</option>
        <option value="4" {% if request.GET.rating_min == '4' %}selected{% endif %}>4★ & up</option>
        <option value="3" {% if request.GET.rating_min == '3' %}selected{% endif %}>3★ & up</option>
        <option value="2" {% if request.GET.rating_min == '2' %}selected{% endif %}>2★ & up</option>
        <option value="1" {% if request.GET.rating_min == '1' %}selected{% endif %}>1★ & up</option>
      </select>

      <select class="form-select form-select-sm" name="sort">
          <option value="new" {% if request.GET.sort == 'new' or not request.GET.sort %}selected{% endif %}>Sort:
New</option>
          <option value="trending" {% if request.GET.sort == 'trending' %}selected{% endif %}>Sort:
Trending</option>
        <option value="top" {% if request.GET.sort == 'top' %}selected{% endif %}>Sort: Top Rated</option>
```

```
    </select>

    <div class="d-flex gap-2">
      <button class="btn btn-sm btn-dark" type="submit">Apply</button>
      {% if request.path|slice:':15' == '/products/models' %}
        <a class="btn btn-sm btn-outline-secondary" href="{% url 'products:models' %}">Clear</a>
      {% elif request.path|slice:':14' == '/products/files' %}
        <a class="btn btn-sm btn-outline-secondary" href="{% url 'products:files' %}">Clear</a>
      {% else %}
        <a class="btn btn-sm btn-outline-secondary" href="{% url 'products:list' %}">Clear</a>
      {% endif %}
    </div>
  </form>
</details>

{# -------------------- 3D MODELS -------------------- #}
<div class="text-uppercase small text-secondary fw-semibold mb-2">
  <i class="bi bi-box-seam me-1"></i>3D Models
</div>

<ul class="list-unstyled mb-3" id="hc3ModelsList">
  {% for cat in sidebar_model_categories %}
    {% if forloop.counter0 == 8 %}
      </ul>

      <details class="hc3-more mb-3" id="hc3ModelsMore">
        <summary class="hc3-more-summary">More...</summary>
        <ul class="list-unstyled mt-2" id="hc3ModelsMoreList">
    {% endif %}

    <li class="mb-1 hc3-filter-parent" data-name="{{ cat.name|lower }}">
      <div class="hc3-cat-row">
        <a class="hc3-cat-link" href="{{ cat.get_absolute_url }}">{{ cat.name }}</a>

        {% if cat.children.all|length %}
          <button class="btn btn-sm hc3-tree-toggle hc3-cat-toggle"
                  type="button"
                  data-target="cat-{{ cat.id }}"
                  aria-expanded="false"
                  aria-controls="cat-{{ cat.id }}"
                  title="Show subcategories">
            <i class="bi bi-caret-right-fill"></i>
          </button>
        {% endif %}
      </div>

      {% if cat.children.all|length %}
        <ul class="list-unstyled ms-2 mt-1 hc3-tree-children"
            id="cat-{{ cat.id }}"
            hidden>
          {% for child in cat.children.all %}
            <li class="mb-1 hc3-filter-child" data-name="{{ child.name|lower }}">
              <a class="hc3-child-link" href="{{ child.get_absolute_url }}">
                {{ child.name }}
              </a>
            </li>
          {% endfor %}
        </ul>
      {% endif %}
    </li>
  {% empty %}
```

```
    <li class="text-secondary small">No model categories yet.</li>
  {% endfor %}
</ul>

{% if sidebar_model_categories|length > 8 %}
      </ul>
    </details>
{% endif %}

{# -------------------- 3D FILES -------------------- #}
<div class="text-uppercase small text-secondary fw-semibold mb-2">
  <i class="bi bi-file-earmark-zip me-1"></i>3D Files
</div>

<ul class="list-unstyled mb-0" id="hc3FilesList">
  {% for cat in sidebar_file_categories %}
    {% if forloop.counter0 == 8 %}
      </ul>

      <details class="hc3-more mb-0" id="hc3FilesMore">
        <summary class="hc3-more-summary">More…</summary>
        <ul class="list-unstyled mt-2" id="hc3FilesMoreList">
    {% endif %}

    <li class="mb-1 hc3-filter-parent" data-name="{{ cat.name|lower }}">
      <div class="hc3-cat-row">
        <a class="hc3-cat-link" href="{{ cat.get_absolute_url }}">{{ cat.name }}</a>

        {% if cat.children.all|length %}
          <button class="btn btn-sm hc3-tree-toggle hc3-cat-toggle"
                  type="button"
                  data-target="cat-{{ cat.id }}"
                  aria-expanded="false"
                  aria-controls="cat-{{ cat.id }}"
                  title="Show subcategories">
            <i class="bi bi-caret-right-fill"></i>
          </button>
        {% endif %}
      </div>

      {% if cat.children.all|length %}
        <ul class="list-unstyled ms-2 mt-1 hc3-tree-children"
            id="cat-{{ cat.id }}"
            hidden>
          {% for child in cat.children.all %}
            <li class="mb-1 hc3-filter-child" data-name="{{ child.name|lower }}">
              <a class="hc3-child-link" href="{{ child.get_absolute_url }}">
                {{ child.name }}
              </a>
            </li>
          {% endfor %}
        </ul>
      {% endif %}
    </li>
  {% empty %}
    <li class="text-secondary small">No file categories yet.</li>
  {% endfor %}
</ul>

{% if sidebar_file_categories|length > 8 %}
      </ul>
      </details>
  {% endif %}


  {# -------------------- AMAZON / AFFILIATE LINKS -------------------- #}
  {% if site_config.affiliate_links_enabled and site_config.affiliate_links %}
    <hr class="my-3">

    <div class="text-uppercase small text-secondary fw-semibold mb-2">
      <i class="bi bi-bag-check me-1"></i>{{ site_config.affiliate_links_title|default:"Recommended Filament & Gear"
}}
    </div>

    <ul class="list-unstyled mb-2">
      {% for item in site_config.affiliate_links %}
        <li class="mb-1">
          <a class="hc3-side-link" href="{{ item.url }}" target="_blank" rel="nofollow sponsored noopener">
            <i class="bi bi-box-arrow-up-right"></i>
            <span>{{ item.label }}</span>
          </a>
          {% if item.note %}
            <div class="small text-secondary ms-4">{{ item.note }}</div>
          {% endif %}
        </li>
      {% endfor %}
    </ul>

    {% if site_config.affiliate_disclosure_text %}
      <div class="small text-secondary">
        {{ site_config.affiliate_disclosure_text }}
      </div>
    {% endif %}
  {% endif %}

  <br>
  <br>

  <!-- Square ads -->
  <ins class="adsbygoogle"
    style="display:block"
    data-ad-client="ca-pub-9022631332194777"
    data-ad-slot="6271230306"
    data-ad-format="auto"
    data-full-width-responsive="true"></ins>
  <script>
     (adsbygoogle = window.adsbygoogle || []).push({});
  </script>

</div>
```

payments\signals.py

```
# payments/signals.py
from __future__ import annotations

from django.db.models.signals import post_save
from django.dispatch import receiver

from payments.models import SellerStripeAccount
from payments.services_fee_waiver import ensure_fee_waiver_for_new_seller


@receiver(post_save, sender=SellerStripeAccount)
def _start_fee_waiver_when_stripe_row_created(sender, instance: SellerStripeAccount, created: bool, **kwargs):
    """
    Practical trigger: when a SellerStripeAccount row exists, we treat the user as a seller.
    That lets us automatically start the waiver without depending on user/profile role plumbing.

    If your "seller role" is tracked elsewhere (e.g., profile.is_seller), we can move this signal
    to that model instead. This works immediately with what you already have.
    """
    if created:
        ensure_fee_waiver_for_new_seller(seller_user=instance.user)
```

core\site_settings.py

```
# core/site_settings.py
from __future__ import annotations

from decimal import Decimal
from typing import Dict, Tuple

from django.db import transaction

from .models import SiteConfig

# (default_value, description)
DEFAULTS: Dict[str, Tuple[str, str]] = {
    # Platform cut taken from sales (percent). Example: "10.0" => 10%
    "marketplace_sales_percent": ("10.0", "Platform cut of each sale, as a percent (e.g. 10.0)."),

    # You mentioned adding a platform fee later (flat fee). Leave default at 0 for now.
    "order_platform_fee_cents": ("0", "Optional flat fee per order in cents (0 disables)."),
}


def ensure_defaults_exist() -> None:
    """
    Ensures all DEFAULTS keys exist in the DB.
    Safe to call at runtime.
    """
    # Avoid wrapping in atomic unless you want strict consistency; this is fine.
    SiteConfig.objects.get_or_create(pk=1)


def get_str(key: str, default: str = "") -> str:
    obj = SiteConfig.objects.filter(pk=1).first()
    if obj is None:
        return default
    val = getattr(obj, key, None)
    if val is None:
        return default
    return str(val).strip()


def get_int(key: str, default: int = 0) -> int:
    obj = SiteConfig.objects.filter(pk=1).first()
    if obj is None:
        return default
    try:
        val = getattr(obj, key, None)
        if val is None:
            return default
        return int(val)
    except (ValueError, TypeError):
        return default


def get_decimal(key: str, default: Decimal = Decimal("0")) -> Decimal:
    obj = SiteConfig.objects.filter(pk=1).first()
    if obj is None:
        return default
    try:
        val = getattr(obj, key, None)
        if val is None:
            return default
        return Decimal(str(val))
    except Exception:
        return default


def get_bool(key: str, default: bool = False) -> bool:
    obj = SiteConfig.objects.filter(pk=1).first()
    if obj is None:
        return default
    try:
        val = getattr(obj, key, None)
        if val is None:
            return default
        if isinstance(val, bool):
            return val
        return str(val).lower() in ("true", "1", "yes")
    except Exception:
        return default


def marketplace_sales_percent() -> Decimal:
    """
    The sales cut percent (e.g. 10.0).
    """
    ensure_defaults_exist()
    return get_decimal("marketplace_sales_percent", default=Decimal(DEFAULTS["marketplace_sales_percent"][0]))


def marketplace_sales_rate() -> Decimal:
    """
    The sales cut rate (e.g. 0.10).
    """
    pct = marketplace_sales_percent()
    try:
        return (pct / Decimal("100"))
    except Exception:
        return Decimal("0.10")
```

refunds\templates\refunds\staff_queue.html

```
{# refunds/templates/refunds/staff_queue.html #}
{% extends "base.html" %}
{% load money %}

{% block title %}Refund Requests · Staff · Home Craft 3D{% endblock %}

{% block content %}
<div class="container py-4">
  <div class="d-flex justify-content-between align-items-center mb-3">
    <div>
      <h1 class="h4 mb-0">Staff refund safety valve</h1>
      <div class="text-muted small">
        Use only if the seller is unavailable (inactive/suspended) or cannot complete the refund.
      </div>
    </div>
    <a class="btn btn-outline-secondary btn-sm" href="{% url 'orders:refunds:buyer_list' %}">Buyer list</a>
  </div>

  {% if refunds %}
    <div class="list-group">
      {% for rr in refunds %}
        <div class="list-group-item">
          <div class="d-flex justify-content-between gap-3">
            <div>
              <div class="fw-semibold">{{ rr.order_item.product.title }}</div>
              <div class="text-muted small">
                {{ rr.get_status_display }} · {{ rr.get_reason_display }} · Order #{{ rr.order_id }}
              </div>
              <div class="text-muted small">
                Seller: {{ rr.seller.username }} · Buyer:
                {% if rr.buyer_id %}
                  {{ rr.buyer.username }}
                {% else %}
                  Guest ({{ rr.requester_email }})
                {% endif %}
              </div>

              {% if rr.seller_decision_note %}
                <div class="mt-2 small">
                  <strong>Seller note:</strong> {{ rr.seller_decision_note|linebreaksbr }}
                </div>
              {% endif %}
            </div>

            <div class="text-end">
              <div class="fw-semibold">${{ rr.total_refund_cents_snapshot|cents_to_dollars }}</div>

              {% if rr.status == "approved" and not rr.stripe_refund_id %}
                <form method="post" action="{% url 'refunds:staff_trigger_refund' refund_id=rr.pk %}" class="mt-2">
                  {% csrf_token %}
                  <button class="btn btn-sm btn-danger">Trigger refund (staff)</button>
                </form>
              {% elif rr.stripe_refund_id %}
                <div class="text-muted small mt-2">Refunded: {{ rr.stripe_refund_id }}</div>
              {% endif %}
            </div>
          </div>
        </div>
      {% endfor %}
    </div>
  {% else %}
    <div class="alert alert-info mb-0">No refund requests.</div>
  {% endif %}
</div>
{% endblock %}
```

qa\templates\qa\staff_reports_queue.html

```
{# qa/templates/qa/staff_reports_queue.html #}
{% extends "base.html" %}

{% block title %}Q&A Reports · Home Craft 3D{% endblock %}

{% block content %}
<div class="container py-4">
  <div class="d-flex align-items-center justify-content-between mb-3">
    <h1 class="h4 mb-0">Q&amp;A Reports</h1>
    <span class="badge text-bg-secondary">Open: {{ reports|length }}</span>
  </div>

  <div class="card shadow-sm">
    <div class="card-body p-0">
      <div class="table-responsive">
        <table class="table mb-0">
          <thead>
            <tr>
              <th>When</th>
              <th>Product</th>
              <th>Reporter</th>
              <th>Reason</th>
              <th>Message</th>
              <th class="text-end">Action</th>
            </tr>
          </thead>
          <tbody>
            {% for r in reports %}
              <tr>
                <td class="small text-secondary">{{ r.created_at|date:"M j, Y" }}</td>
                <td>
                  <a class="text-decoration-none" href="{{ r.message.thread.product.get_absolute_url }}#qa">
                    {{ r.message.thread.product.title }}
                  </a>
                </td>
                <td>{{ r.reporter.username }}</td>
                <td>{{ r.get_reason_display }}</td>
                <td class="small" style="max-width:420px; white-space:nowrap; overflow:hidden; text-overflow:ellipsis;">
                  {{ r.message.body }}
                  {% if r.details %}<div class="text-secondary">Details: {{ r.details }}</div>{% endif %}
                </td>
                <td class="text-end">
                  <form method="post" action="{% url 'qa:staff_report_resolve' r.id %}">
                    {% csrf_token %}
                    <button class="btn btn-sm btn-outline-success" type="submit">Resolve</button>
                  </form>
                </td>
              </tr>
            {% empty %}
              <tr>
                <td colspan="6" class="text-center text-secondary py-4">No open reports.</td>
              </tr>
            {% endfor %}
          </tbody>
        </table>
      </div>
    </div>
  </div>
</div>
```

{% endblock %}

core\storage_backends.py

```
# core/storage_backends.py
from __future__ import annotations

from django.conf import settings
from django.core.files.storage import FileSystemStorage

try:
    from storages.backends.s3boto3 import S3Boto3Storage
except Exception:  # pragma: no cover
    S3Boto3Storage = None  # type: ignore


class MediaStorage(S3Boto3Storage):  # type: ignore[misc]
    """
    Media bucket storage (images, avatars, etc).

    Reads from Django settings to properly inherit AWS config.
    """
    @property
    def bucket_name(self):
        return getattr(settings, "AWS_S3_MEDIA_BUCKET", "")

    @property
    def access_key(self):
        return getattr(settings, "AWS_ACCESS_KEY_ID", "")

    @property
    def secret_key(self):
        return getattr(settings, "AWS_SECRET_ACCESS_KEY", "")

    @property
    def region_name(self):
        return getattr(settings, "AWS_S3_REGION_NAME", "us-east-2")

    default_acl = None
    file_overwrite = False
    location = "media"


class DownloadsStorage(S3Boto3Storage):  # type: ignore[misc]
    """
    Private downloads bucket storage for paid digital assets.

    Always uses signed URLs.
    """
    @property
    def bucket_name(self):
        return getattr(settings, "AWS_S3_DOWNLOADS_BUCKET", "")

    @property
    def access_key(self):
        return getattr(settings, "AWS_ACCESS_KEY_ID", "")

    @property
    def secret_key(self):
        return getattr(settings, "AWS_SECRET_ACCESS_KEY", "")

    @property
    def region_name(self):
        return getattr(settings, "AWS_S3_REGION_NAME", "us-east-2")

    default_acl = None
    file_overwrite = False
    location = "downloads"
    querystring_auth = True
    custom_domain = None  # ensures signed S3 URLs


def get_media_storage():
    """
    Returns the correct storage for general media.
    - If USE_S3=True: MediaStorage (S3)
    - Else: local filesystem under MEDIA_ROOT
    """
    if getattr(settings, "USE_S3", False) and S3Boto3Storage is not None:
        return MediaStorage()
    return FileSystemStorage(location=settings.MEDIA_ROOT, base_url=settings.MEDIA_URL)


def get_downloads_storage():
    """
    Returns the correct storage for paid downloads.
    - If USE_S3=True: DownloadsStorage (S3 private bucket)
    - Else: local filesystem under MEDIA_ROOT (dev)
    """
    if getattr(settings, "USE_S3", False) and S3Boto3Storage is not None:
        return DownloadsStorage()
    return FileSystemStorage(location=settings.MEDIA_ROOT, base_url=settings.MEDIA_URL)
```

payments\stripe_connect.py

```python
# payments/stripe_connect.py
from __future__ import annotations

import stripe
from django.conf import settings
from django.urls import reverse


def _base_url() -> str:
    if not (base := (getattr(settings, "SITE_BASE_URL", "") or "").strip()):
        raise RuntimeError("SITE_BASE_URL is required for Stripe Connect links.")
    else:
        return base.rstrip("/")


def configure_stripe() -> None:
    stripe.api_key = settings.STRIPE_SECRET_KEY


def create_express_account(*, email: str, country: str = "US") -> stripe.Account:
    configure_stripe()

    return stripe.Account.create(
        type="express",
        country=country,
        email=email,
        capabilities={
            "card_payments": {"requested": True},
            "transfers": {"requested": True},
        },
        business_type="individual",  # MVP default; Stripe will ask for details
    )


def create_account_link(*, stripe_account_id: str) -> stripe.AccountLink:
    """
    Returns a Stripe-hosted onboarding link for an Express account.
    """
    configure_stripe()

    refresh_url = f"{_base_url()}{reverse('payments:connect_refresh')}"
    return_url = f"{_base_url()}{reverse('payments:connect_return')}"

    return stripe.AccountLink.create(
        account=stripe_account_id,
        refresh_url=refresh_url,
        return_url=return_url,
        type="account_onboarding",
    )


def retrieve_account(stripe_account_id: str) -> stripe.Account:
    configure_stripe()
    return stripe.Account.retrieve(stripe_account_id)
```

refunds\stripe_service.py

```
# refunds/stripe_service.py

from __future__ import annotations

import stripe
from django.conf import settings

from .models import RefundRequest


def _init_stripe() -> None:
    key = getattr(settings, "STRIPE_SECRET_KEY", None)
    if not key:
        raise RuntimeError("STRIPE_SECRET_KEY is not configured.")
    stripe.api_key = key


def create_stripe_refund_for_request(*, rr: RefundRequest) -> str:
    """
    Create a Stripe refund for this request.

    Source of truth:
      - refund amount = rr.total_refund_cents_snapshot
      - target payment = rr.order.stripe_payment_intent_id

    Idempotency:
      - idempotency_key=f"refundreq-{rr.pk}"
    """
    _init_stripe()

    order = rr.order
    payment_intent = (getattr(order, "stripe_payment_intent_id", "") or "").strip()

    # FREE checkouts have no Stripe refund
    if payment_intent == "FREE":
        raise ValueError("Order was a free checkout; no Stripe refund is possible.")

    if not payment_intent:
        raise ValueError("Order has no Stripe payment intent id; cannot refund.")

    amount = int(getattr(rr, "total_refund_cents_snapshot", 0) or 0)
    if amount <= 0:
        raise ValueError("Refund amount must be > 0.")

    refund = stripe.Refund.create(
        payment_intent=payment_intent,
        amount=amount,
        metadata={
            "refund_request_id": str(rr.pk),
            "order_id": str(order.pk),
            "order_item_id": str(rr.order_item_id),
            "seller_id": str(rr.seller_id),
        },
        idempotency_key=f"refundreq-{rr.pk}",
    )

    refund_id = str(getattr(refund, "id", "") or "").strip()
    if not refund_id:
        raise ValueError("Stripe did not return a refund id.")
```

```
return refund_id
```

legal\templates\legal\terms.html

```
{# templates/legal/terms.html #}
{% extends "legal/_legal_base.html" %}

{% block doc_content %}
<h2>1. Marketplace Role</h2>
<p>Home Craft 3D is a marketplace that connects buyers and independent sellers. Sellers are responsible for their listings, fulfillment, and compliance with laws.</p>
<p>Home Craft 3D provides the platform only and does not manufacture, inspect, or quality-control items sold by sellers. Any guarantees, product quality, safety, or regulatory compliance are the sole responsibility of the seller.</p>

<h2>2. Accounts</h2>
<p>You're responsible for your account activity, keeping your credentials secure, and providing accurate information.</p>

<h2>3. Listings & Purchases</h2>
<p>Sellers set prices, descriptions, and file contents. Buyers should review listing details before purchase. Digital files are delivered electronically; physical items are shipped by sellers.</p>

<h2>4. Payments</h2>
<p>Payments are processed at checkout. Order status is "pending payment" until payment is confirmed.</p>

<h2>4.1 Fees & Payouts</h2>
<p>
Home Craft 3D charges a marketplace fee on each completed sale, calculated as a percentage of the order total before applicable taxes and shipping. Payment processing fees (such as Stripe fees) are deducted separately and are not controlled by Home Craft 3D.
</p>
<p>
<strong>New Seller Promotional Waiver.</strong> From time to time, Home Craft 3D may offer promotional fee waivers for new sellers. When such a promotion is active, eligible new sellers may receive a <strong>0% marketplace fee for their first 30 days</strong> from the date they upload their first product. During this promotional period, Home Craft 3D retains no marketplace fee; payment processing fees still apply.
</p>
<p>
After the promotional period ends, standard marketplace fees automatically apply to subsequent sales unless otherwise stated. Eligibility, duration, and availability of promotional waivers are determined solely by Home Craft 3D and may be modified, suspended, or discontinued at any time.
</p>
<p>
Net proceeds (after applicable fees) are paid out to sellers according to the platform’s payout schedule and payment processor requirements. For additional details, see our <a href="/docs/seller_fees/">Seller Fees</a> page.
</p>

<h2>5. Digital Items</h2>
<p>Digital downloads are intended for personal use unless the listing specifies otherwise. Sharing, reselling, or redistributing files without permission is prohibited.</p>

<h2>6. Shipping (Physical Items)</h2>
<p>Shipping timelines and handling are set by sellers. Delays may occur due to carrier issues or production time.</p>

<h2>7. Prohibited Content</h2>
<p>Listings and user content must comply with our <a href="{% url 'legal:content' %}">Content & Safety Policy</a>.</p>

<h2>8. Refunds</h2>
<p>Refund eligibility is described in the <a href="{% url 'legal:refund' %}">Refund Policy</a>.</p>
```

```
<h2>9. Intellectual Property</h2>
<p>Sellers must own or have rights to sell their content. Buyers must respect seller IP rights.</p>

<h2>10. Limitation of Liability</h2>
<p>Home Craft 3D is not liable for indirect damages, lost profits, or issues arising from seller listings or fulfillment. To the maximum extent permitted by law, Home Craft 3D and its staff disclaim liability for product quality, defects, or seller performance.</p>

<h2>11. Changes</h2>
<p>We may update these Terms. Continued use of the site means you accept the updated Terms.</p>

<h2>12. Disclosures</h2>
<p>Some product links may be affiliate links. These help support the platform at no additional cost to you.</p>

<p><strong>Contact:</strong> <a
href="mailto:homecraft3dstore@gmail.com">homecraft3dstore@gmail.com</a></p>
{% endblock %}
```

reviews\tests.py

```
# cart/tests.py
from django.test import TestCase

# Create your tests here.
```

templates\includes\theme_css_vars.html

```
{# templates/includes/theme_css_vars.html #}
<style>
  :root {
    --hc-primary: {{ site_config.theme_primary }};
    --hc-accent: {{ site_config.theme_accent }};
    --hc-success: {{ site_config.theme_success }};
    --hc-danger: {{ site_config.theme_danger }};

    --hc-light-bg: {{ site_config.theme_light_bg }};
    --hc-light-surface: {{ site_config.theme_light_surface }};
    --hc-light-text: {{ site_config.theme_light_text }};
    --hc-light-text-muted: {{ site_config.theme_light_text_muted }};
    --hc-light-border: {{ site_config.theme_light_border }};

    --hc-dark-bg: {{ site_config.theme_dark_bg }};
    --hc-dark-surface: {{ site_config.theme_dark_surface }};
    --hc-dark-text: {{ site_config.theme_dark_text }};
    --hc-dark-text-muted: {{ site_config.theme_dark_text_muted }};
    --hc-dark-border: {{ site_config.theme_dark_border }};

    --hc-default-mode: "{{ site_config.theme_default_mode }}";
  }
</style>
```

core\throttle.py

```python
# core/throttle.py
from __future__ import annotations

import time
from dataclasses import dataclass
from typing import Callable

from django.conf import settings
from django.core.cache import cache
from django.http import HttpRequest, HttpResponse


@dataclass(frozen=True)
class ThrottleRule:
    key_prefix: str
    limit: int
    window_seconds: int


def _get_client_ip(request: HttpRequest) -> str:
    """
    Best-effort client IP.

    If THROTTLE_TRUST_PROXY_HEADERS=True (prod behind your own proxy),
    we trust X-Forwarded-For / X-Real-IP. Otherwise use REMOTE_ADDR.
    """
    trust_proxy = bool(getattr(settings, "THROTTLE_TRUST_PROXY_HEADERS", False))

    if trust_proxy:
        xff = (request.META.get("HTTP_X_FORWARDED_FOR") or "").strip()
        if xff:
            # first IP is the original client
            ip = xff.split(",")[0].strip()
            if ip:
                return ip

        xri = (request.META.get("HTTP_X_REAL_IP") or "").strip()
        if xri:
            return xri

    return (request.META.get("REMOTE_ADDR") or "ip-unknown").strip() or "ip-unknown"


def _client_fingerprint(request: HttpRequest) -> str:
    """
    Fingerprint is stable enough to throttle abuse, but not overly unique.

    Includes:
    - client ip
    - short user agent prefix
    - user id if authenticated
    """
    ip = _get_client_ip(request)
    ua = (request.META.get("HTTP_USER_AGENT") or "")[:60]
    user_part = f"user:{request.user.id}" if getattr(request.user, "is_authenticated", False) else "anon"
    return f"{ip}|{ua}|{user_part}"


def throttle(rule: ThrottleRule) -> Callable:
```

```
"""
Cache-based throttle.

Intended for POST/PUT/PATCH/DELETE endpoints that can be abused:
- Q&A create/reply/report/delete
- refund create/approve/decline/trigger
- checkout place + checkout start
- auth login/register
"""
def decorator(view_func: Callable) -> Callable:
    def wrapped(request: HttpRequest, *args, **kwargs) -> HttpResponse:
        if request.method not in ("POST", "PUT", "PATCH", "DELETE"):
            return view_func(request, *args, **kwargs)

        fp = _client_fingerprint(request)
        bucket = int(time.time() // max(1, rule.window_seconds))
        cache_key = f"throttle:{rule.key_prefix}:{bucket}:{fp}"

        current = int(cache.get(cache_key, 0) or 0)
        if current >= rule.limit:
            # For typical browser POST form flows, redirect back and show a friendly message.
            # Fall back to 429 if we can't.
            try:
                accept = (request.META.get("HTTP_ACCEPT") or "").lower()
                referer = (request.META.get("HTTP_REFERER") or "").strip()
                if "text/html" in accept and referer:
                    try:
                        from django.contrib import messages
                        messages.error(request, "Too many requests. Please try again in a moment.")
                    except Exception:
                        pass
                    from django.shortcuts import redirect
                    return redirect(referer)
            except Exception:
                pass

            return HttpResponse("Too many requests. Please try again shortly.", status=429)

        cache.set(cache_key, current + 1, timeout=rule.window_seconds + 5)
        return view_func(request, *args, **kwargs)

    wrapped.__name__ = getattr(view_func, "__name__", "wrapped")
    wrapped.__doc__ = getattr(view_func, "__doc__", "")
    wrapped.__module__ = getattr(view_func, "__module__", "")
    return wrapped
return decorator
```

products\templates\products\top_sellers.html

```
{# products/templates/products/top_sellers.html #}
{% extends "base.html" %}
{% block title %}Top Sellers · Home Craft 3D{% endblock %}
{% block content %}
<div class="container py-4">
  <h1 class="mb-4">Top Sellers</h1>
  <div class="row g-4">
    {% for seller in top_sellers %}
      <div class="col-12 col-md-6 col-lg-4">
        <div class="card h-100">
          <div class="card-body text-center">
            {% if seller.profile.avatar %}
              <img src="{{ seller.profile.avatar.url }}" alt="{{ seller.profile.public_seller_name }} avatar" class="rounded-circle mb-2" style="width: 72px; height: 72px; object-fit: cover;">
            {% endif %}
            <div class="h5 mb-1">{{ seller.profile.public_seller_name|default:seller.username }}</div>
            <div class="text-secondary small mb-1">Member since {{ seller.date_joined|date:"F Y" }}</div>
            <div class="mb-2 small">{{ seller.product_count }} product{{ seller.product_count|pluralize }}</div>
            <div class="mb-2 small">{{ seller.seller_review_count }} review{{ seller.seller_review_count|pluralize }}</div>
            <div class="mb-2 small">Average rating: {{ seller.seller_avg_rating|floatformat:1 }}/5</div>
            <a href="{% url 'products:seller_shop' seller.id %}" class="btn btn-sm btn-outline-primary">View Shop</a>
          </div>
        </div>
      </div>
    {% empty %}
      <div class="col-12">
        <div class="alert alert-secondary text-center">No top sellers found.</div>
      </div>
    {% endfor %}
  </div>
</div>
{% endblock %}
```

products\management\commands\update_trending_products.py

```
# products/management/commands/update_trending_products.py
"""
Update trending products based on recent engagement (views).

Marks products as trending if they have high engagement in the last 7 days.
Run periodically (e.g., via celery beat or cron): every 6 hours.
"""
from __future__ import annotations

from datetime import timedelta

from django.core.management.base import BaseCommand
from django.db import models
from django.utils import timezone

from products.models import Product, ProductEngagementEvent


class Command(BaseCommand):
    help = "Update trending products based on recent engagement metrics"

    def add_arguments(self, parser):
        parser.add_argument(
            "--days",
            type=int,
            default=7,
            help="Number of days to look back for engagement (default: 7)",
        )
        parser.add_argument(
            "--top-count",
            type=int,
            default=20,
            help="Number of products to mark as trending (default: 20)",
        )
        parser.add_argument(
            "--min-views",
            type=int,
            default=5,
            help="Minimum views required to be considered for trending (default: 5)",
        )

    def handle(self, *args, **options):
        days = options["days"]
        top_count = options["top_count"]
        min_views = options["min_views"]

        cutoff_date = timezone.now() - timedelta(days=days)

        # Get engagement events from the past N days
        recent_events = ProductEngagementEvent.objects.filter(
            created_at__gte=cutoff_date,
            event_type=ProductEngagementEvent.EventType.VIEW,
        )

        # Count views per product
        view_counts = (
            recent_events.values("product_id")
            .annotate(view_count=models.Count("id"))
            .filter(view_count__gte=min_views)
            .order_by("-view_count")[: top_count]
        )

        trending_product_ids = [item["product_id"] for item in view_counts]

        # Mark trending products
        Product.objects.filter(id__in=trending_product_ids).update(is_trending=True)

        # Unmark non-trending products
        Product.objects.exclude(id__in=trending_product_ids).update(is_trending=False)

        self.stdout.write(
            self.style.SUCCESS(
                f"✓ Updated trending products: {len(trending_product_ids)} products "
                f"marked as trending (based on {days} days, min {min_views} views)"
            )
        )

        # Show top products
        if trending_product_ids:
            top_products = Product.objects.filter(id__in=trending_product_ids[:5])
            self.stdout.write("\nTop trending products:")
            for i, product in enumerate(top_products, 1):
                view_count = next(
                    (item["view_count"] for item in view_counts if item["product_id"] == product.id),
                    0,
                )
                self.stdout.write(f"  {i}. {product.title} ({view_count} views)")
```

reviews\urls.py

```
# reviews/urls.py
from django.urls import path

from . import views

app_name = "reviews"

urlpatterns = [
    path("product/<int:product_id>/", views.product_reviews, name="product_reviews"),
    path("order-item/<int:order_item_id>/new/", views.review_create_for_order_item, name="review_for_item"),

    # Purchased-only seller rating
    path("seller/<uuid:order_id>/<int:seller_id>/new/", views.seller_review_create, name="seller_review_new"),
]
```

orders\urls_buyer.py

```
# orders/urls_buyer.py
from django.urls import path
from . import views_buyer

urlpatterns = [
    path("my/", views_buyer.my_orders_list, name="my_orders"),
]
```

products\urls_seller.py

```
# products/urls_seller.py
from django.urls import path

from . import views_seller

urlpatterns = [
    path("", views_seller.seller_product_list, name="seller_list"),
    path("new/", views_seller.seller_product_create, name="seller_create"),
    path("<int:pk>/edit/", views_seller.seller_product_edit, name="seller_edit"),
    path("<int:pk>/duplicate/", views_seller.seller_product_duplicate, name="seller_duplicate"),

    path("<int:pk>/images/", views_seller.seller_product_images, name="seller_images"),
    path("<int:pk>/images/<int:image_id>/delete/", views_seller.seller_product_image_delete, name="seller_image_delete"),
    path("<int:pk>/images/<int:image_id>/update/", views_seller.seller_product_image_update, name="seller_image_update"),

    path("<int:pk>/assets/", views_seller.seller_product_assets, name="seller_assets"),
    path("<int:pk>/assets/<int:asset_id>/delete/", views_seller.seller_product_asset_delete, name="seller_asset_delete"),

    path("<int:pk>/toggle-active/", views_seller.seller_product_toggle_active, name="seller_toggle_active"),
]
```

payments\utils.py

```
# payments/utils.py
from __future__ import annotations

from decimal import Decimal, ROUND_HALF_UP
from typing import Union

from payments.models import SellerStripeAccount
from products.permissions import is_owner_user


def seller_is_stripe_ready(seller_user) -> bool:
    """
    True if seller can receive payouts (Stripe Connect fully enabled).

    Owner/admin bypass is treated as ready.
    """
    if seller_user and is_owner_user(seller_user):
        return True

    acct = SellerStripeAccount.objects.filter(user=seller_user).first()
    return bool(acct and acct.is_ready)


MoneyLike = Union[Decimal, int, float, str, None]


def money_to_cents(value: MoneyLike) -> int:
    """Convert a currency amount (e.g. dollars) to integer cents safely.

    Accepts Decimal, int, float, or numeric string. Uses ROUND_HALF_UP.

    NOTE:
    - If caller passes an int, we assume it is already cents (by convention).
    """
    if value is None:
        return 0

    if isinstance(value, int):
        return int(value)

    dec = value if isinstance(value, Decimal) else Decimal(str(value))
    cents = (dec * Decimal("100")).quantize(Decimal("1"), rounding=ROUND_HALF_UP)
    return int(cents)


def cents_to_money(cents: int) -> Decimal:
    return (Decimal(int(cents)) / Decimal("100")).quantize(Decimal("0.01"))
```

reviews\views.py

```
# reviews/views.py
from __future__ import annotations

from django.contrib import messages
from django.core.exceptions import PermissionDenied
from django.http import Http404
from django.shortcuts import get_object_or_404, redirect, render
from django.views.decorators.http import require_http_methods

from .forms import ReviewForm, SellerReviewForm
from .models import Review, SellerReview
from .services import get_rateable_seller_order_or_403, get_reviewable_order_item_or_403


def product_reviews(request, product_id: int):
    qs = (
        Review.objects.select_related("buyer")
        .filter(product_id=product_id)
        .order_by("-created_at")
    )

    from django.db.models import Avg, Count

    summary = qs.aggregate(avg=Avg("rating"), count=Count("id"))
    avg_rating = summary.get("avg") or 0
    review_count = summary.get("count") or 0

    return render(
        request,
        "reviews/product_reviews.html",
        {"reviews": qs, "avg_rating": avg_rating, "review_count": review_count, "product_id": product_id},
    )


@require_http_methods(["GET", "POST"])
def review_create_for_order_item(request, order_item_id: int):
    try:
        item = get_reviewable_order_item_or_403(user=request.user, order_item_id=order_item_id)
    except PermissionDenied:
        raise Http404("Not found")

    if hasattr(item, "review"):
        messages.info(request, "You already reviewed this item.")
        return redirect(item.product.get_absolute_url())

    if request.method == "POST":
        form = ReviewForm(request.POST)
        if form.is_valid():
            review: Review = form.save(commit=False)
            review.product = item.product
            review.order_item = item
            review.buyer = request.user
            review.save()
            messages.success(request, "Thanks — your review was posted.")
            return redirect(item.product.get_absolute_url())
    else:
        form = ReviewForm()

    return render(
        request,
        "reviews/review_form.html",
        {"form": form, "item": item, "product": item.product},
    )


@require_http_methods(["GET", "POST"])
def seller_review_create(request, order_id: int, seller_id: int):
    """Create a seller rating for a seller within a specific PAID order."""
    try:
        order = get_rateable_seller_order_or_403(user=request.user, order_id=order_id, seller_id=seller_id)
    except PermissionDenied:
        raise Http404("Not found")

    # Prevent duplicates per order
    existing = SellerReview.objects.filter(order_id=order.id, seller_id=seller_id, buyer_id=request.user.id).first()
    if existing:
        messages.info(request, "You already rated this seller for this order.")
        return redirect("orders:detail", order_id=order.id)

    if request.method == "POST":
        form = SellerReviewForm(request.POST)
        if form.is_valid():
            sr: SellerReview = form.save(commit=False)
            sr.order = order
            sr.seller_id = seller_id
            sr.buyer = request.user
            sr.save()
            messages.success(request, "Thanks — your seller rating was posted.")
            return redirect("orders:detail", order_id=order.id)
    else:
        form = SellerReviewForm()

    return render(
        request,
        "reviews/seller_review_form.html",
        {"form": form, "order": order, "seller_id": seller_id},
    )
```

accounts\views_2fa.py

```
# accounts/views_2fa.py
import io
import qrcode
from base64 import b64encode
from django.contrib.auth.decorators import login_required
from django.contrib.auth.models import User
from django.contrib import messages
from django.http import JsonResponse, HttpResponse
from django.shortcuts import render, redirect
from django.views.decorators.http import require_http_methods
from django.views.decorators.csrf import csrf_protect
from django_otp.plugins.otp_totp.models import TOTPDevice
from django_otp.plugins.otp_static.models import StaticDevice, StaticToken
from django_otp.util import random_hex
from .forms_2fa import TOTPVerificationForm, RecoveryCodeForm, Disable2FAForm


@login_required
@require_http_methods(['GET', 'POST'])
def setup_2fa(request):
    """
    Setup two-factor authentication for the current user.
    GET: Display QR code and recovery codes
    POST: Verify TOTP token to enable 2FA
    """
    user = request.user

    # Check if user already has 2FA enabled
    if user.totpdevice_set.filter(confirmed=True).exists():
        messages.info(request, '2FA is already enabled for your account.')
        return redirect('view_2fa_status')

    if request.method == 'GET':
        # Generate new device if not exists or not confirmed
        device, created = TOTPDevice.objects.get_or_create(
            user=user,
            name='default',
            defaults={'confirmed': False}
        )

        # Generate secret key if not exists
        if not device.key:
            device.key = random_hex(20)
            device.save()

        # Generate or reuse backup codes
        recovery_device, _ = StaticDevice.objects.get_or_create(
            user=user,
            name='recovery'
        )
        backup_codes = list(recovery_device.token_set.all().values_list('token', flat=True))
        if not backup_codes:
            for i in range(10):
                token = StaticToken.random_token()
                StaticToken.objects.create(device=recovery_device, token=token)
                backup_codes.append(token)

        # Generate QR code
        totp_string = device.config_url
        qr = qrcode.QRCode(version=1, box_size=10, border=5)
        qr.add_data(totp_string)
        qr.make(fit=True)

        img = qr.make_image(fill_color="black", back_color="white")
        buffer = io.BytesIO()
        img.save(buffer, format='PNG')
        qr_code_base64 = b64encode(buffer.getvalue()).decode()

        context = {
            'device': device,
            'qr_code': qr_code_base64,
            'backup_codes': backup_codes,
            'totp_url': totp_string,
        }
        return render(request, 'accounts/setup_2fa.html', context)

    elif request.method == 'POST':
        form = TOTPVerificationForm(request.POST)
        if form.is_valid():
            token = form.cleaned_data['token']
            device = TOTPDevice.objects.get(user=user, name='default')

            # Verify token
            if device.verify_token(token):
                device.confirmed = True
                device.save()
                messages.success(request, '2FA has been successfully enabled!')
                return redirect('view_2fa_status')
            else:
                messages.error(request, 'Invalid code. Please try again.')
                return redirect('setup_2fa')

    return render(request, 'accounts/verify_2fa_token.html', {
        'form': TOTPVerificationForm(),
        'step': 'verify'
    })


@login_required
@require_http_methods(['GET'])
def view_2fa_status(request):
    """Display current 2FA status and options."""
    user = request.user
    device = TOTPDevice.objects.filter(user=user, name='default').first()
    is_enabled = device and device.confirmed

    context = {
        'is_2fa_enabled': is_enabled,
        'device': device,
    }
    return render(request, 'accounts/2fa_status.html', context)


@login_required
@require_http_methods(['POST'])
def disable_2fa(request):
    """Disable 2FA for the current user."""
    user = request.user
    form = Disable2FAForm(user=user, data=request.POST)
    if form.is_valid():
        # Delete all OTP devices
        TOTPDevice.objects.filter(user=user).delete()
        StaticDevice.objects.filter(user=user).delete()
        messages.success(request, '2FA has been disabled.')
        return redirect('view_2fa_status')

    return render(request, 'accounts/disable_2fa.html', {
        'form': form,
        'errors': form.errors
    })


@login_required
@require_http_methods(['GET'])
def show_recovery_codes(request):
    """Show recovery codes for the current user."""
    user = request.user
    device = TOTPDevice.objects.filter(user=user, name='default').first()

    if not device or not device.confirmed:
        messages.error(request, '2FA must be enabled first.')
        return redirect('view_2fa_status')

    # Get existing tokens
    recovery_device = StaticDevice.objects.filter(user=user, name='recovery').first()
    codes = list(recovery_device.token_set.all().values_list('token', flat=True)) if recovery_device else []

    context = {
        'codes': codes,
    }
    return render(request, 'accounts/recovery_codes.html', context)


@login_required
@require_http_methods(['POST'])
def regenerate_recovery_codes(request):
    """Regenerate recovery codes."""
    user = request.user
    device = TOTPDevice.objects.filter(user=user, name='default').first()

    if not device or not device.confirmed:
        return JsonResponse({'error': '2FA must be enabled'}, status=400)

    # Delete old tokens
    recovery_device = StaticDevice.objects.filter(user=user, name='recovery').first()
    if not recovery_device:
        recovery_device = StaticDevice.objects.create(user=user, name='recovery')

    StaticToken.objects.filter(device=recovery_device).delete()

    # Generate new tokens
    new_codes = []
    for i in range(10):
        token = StaticToken.random_token()
        StaticToken.objects.create(device=recovery_device, token=token)
        new_codes.append(token)

    messages.success(request, 'Recovery codes have been regenerated.')

    context = {
```

```
    'codes': new_codes,
}
return render(request, 'accounts/recovery_codes.html', context)
```

orders\views_buyer.py

```
# orders/views_buyer.py
from __future__ import annotations

from django.contrib.auth.decorators import login_required
from django.db.models import Prefetch
from django.shortcuts import render

from .models import Order, OrderItem


@login_required
def my_orders_list(request):
    orders = (
        Order.objects.filter(buyer=request.user)
        .prefetch_related(
            Prefetch(
                "items",
                queryset=OrderItem.objects.select_related("product", "seller").order_by("id"),
            )
        )
        .order_by("-created_at")
    )

    return render(request, "orders/buyer/order_list.html", {"orders": orders})
```

products\views_seller.py

```
# products/views_seller.py
from __future__ import annotations

import uuid
from pathlib import Path
from typing import Optional

from django.contrib import messages
from django.core.files import File
from django.db import transaction
from django.db.models import Q
from django.http import Http404, JsonResponse
from django.shortcuts import get_object_or_404, redirect, render

from catalog.models import Category
from core.throttle import ThrottleRule, throttle
from payments.models import SellerStripeAccount
from .forms import (
    ProductForm,
    ProductImageUploadForm,
    ProductImageBulkUploadForm,
    DigitalAssetUploadForm,
    ProductPhysicalForm,
    ProductDigitalForm,
)
from .models import Product, ProductImage, DigitalAsset, ALLOWED_ASSET_EXTS
from .permissions import seller_required, is_owner_user
from .views import _render_product_detail


SELLER_PRODUCT_MUTATE_RULE = ThrottleRule(key_prefix="seller_product_mutate", limit=20,
window_seconds=60)
SELLER_UPLOAD_RULE = ThrottleRule(key_prefix="seller_upload", limit=12, window_seconds=60)
SELLER_DELETE_RULE = ThrottleRule(key_prefix="seller_delete", limit=20, window_seconds=60)
SELLER_CATEGORY_AJAX_RULE = ThrottleRule(key_prefix="seller_category_ajax", limit=60, window_seconds=60)


def _file_type_options() -> list[str]:
    preferred = ["stl", "3mf", "obj", "zip"]
    allowed = [t for t in preferred if t in ALLOWED_ASSET_EXTS] + sorted(
        t for t in ALLOWED_ASSET_EXTS if t not in preferred
    )
    return allowed


def _can_edit_product(user, product: Product) -> bool:
    if is_owner_user(user):
        return True
    return product.seller == user


def _get_owned_product_or_404(request, pk: int) -> Product:
    product = get_object_or_404(Product, pk=pk)
    if not _can_edit_product(request.user, product):
        raise Http404("Not found")
    return product


def _safe_storage_copy(*, source_field, dest_prefix: str) -> Optional[str]:
    """
    Copy a stored file to a new key/path and return the new name.

    - If using django-storages S3Boto3Storage, try server-side bucket copy (fast).
    - Otherwise fall back to reading + saving (streaming via File wrapper).
    """
    if not source_field:
        return None

    storage = getattr(source_field, "storage", None)
    name = getattr(source_field, "name", None)
    if not storage or not name:
        return None

    ext = Path(name).suffix
    new_name = f"{dest_prefix}/{uuid.uuid4().hex}{ext}"

    bucket = getattr(storage, "bucket", None)
    location = (getattr(storage, "location", "") or "").strip("/")

    if bucket is not None:
        src_key = f"{location}/{name}".lstrip("/") if location else name
        try:
            bucket.copy(
                {"Bucket": bucket.name, "Key": src_key},
                f"{location}/{new_name}".lstrip("/") if location else new_name,
            )
            return new_name
        except Exception:
            pass

    try:
        source_field.open("rb")
        storage.save(new_name, File(source_field))
        return new_name
    except Exception:
        return None
    finally:
        try:
            source_field.close()
        except Exception:
            pass


# -------------------------
# Seller AJAX endpoint
# -------------------------
@seller_required
@throttle(SELLER_CATEGORY_AJAX_RULE)
def seller_subcategories_for_category(request):
    """
    Seller-only JSON endpoint for dependent dropdown:
    Category -> Subcategory options.
    """
    raw_id = (request.GET.get("category_id") or "").strip()
    try:
        category_id = int(raw_id)
    except Exception:
        return JsonResponse({"results": []})

    parent = Category.objects.filter(pk=category_id).only("id", "type").first()
    if not parent:
        return JsonResponse({"results": []})

    qs = (
        Category.objects.filter(parent_id=parent.id, is_active=True, type=parent.type)
        .only("id", "name")
        .order_by("sort_order", "name")
    )
    return JsonResponse({"results": [{"id": c.id, "text": c.name} for c in qs]})


@seller_required
def seller_product_list(request, *args, **kwargs):
    """
    Seller dashboard: list your products.
    Owner/admin sees all products.
    """
    from django.db.models import Sum, Q as DjQ

    qs = (
        Product.objects.select_related("category", "category__parent", "seller", "digital", "physical")
        .prefetch_related("images", "digital_assets")
        .annotate(
            units_sold=Sum(
                "order_items__quantity",
                filter=DjQ(order_items__order__status="PAID", order_items__order__paid_at__isnull=False),
            )
        )
        .order_by("-created_at")
    )
    if not is_owner_user(request.user):
        qs = qs.filter(seller=request.user)

    q = (request.GET.get("q") or "").strip()
    if q:
        qs = qs.filter(DjQ(title__icontains=q) | DjQ(short_description__icontains=q) | DjQ(description__icontains=q))

    file_type = (request.GET.get("file_type") or "").strip().lower().lstrip(".")
    if file_type:
        qs = qs.filter(
            DjQ(digital_assets__file_type=file_type) | DjQ(digital_assets__file__iendswith=f".{file_type}")
        ).distinct()

    stripe_account = None
    stripe_ready = True
    if not is_owner_user(request.user):
        stripe_account = SellerStripeAccount.objects.filter(user=request.user).first()
        stripe_ready = bool(stripe_account and stripe_account.is_ready)

    return render(
        request,
        "products/seller/product_list.html",
        {
            "products": qs,
            "q": q,
            "file_type": file_type,
            "file_type_options": _file_type_options(),
            "stripe_account": stripe_account,
            "stripe_ready": stripe_ready,
        },
    )
```

```
@seller_required
@throttle(SELLER_PRODUCT_MUTATE_RULE)
def seller_product_create(request, *args, **kwargs):
    """
    Draft-first create:
    - Create listing with is_active=False by default.
    """
    if request.method == "POST":
        form = ProductForm(request.POST, user=request.user)
        if form.is_valid():
            product = form.save(commit=False)
            product.seller = request.user

            # Draft-first
            product.is_active = False

            # If slug left blank, model will generate from title (slug field should be blank=True in model)
            if not (product.slug or "").strip():
                product.slug_is_manual = False
                product.slug = ""

            product.full_clean()
            product.save()

            if product.kind == Product.Kind.FILE:
                messages.success(
                    request,
                    f"Draft created for '{product.title}' (digital file listing). Add files and images from My Listings.",
                )
            else:
                messages.success(
                    request,
                    f"Draft created for '{product.title}' (physical listing). Add images and specs from My Listings.",
                )
            return redirect("products:seller_list")
    else:
        form = ProductForm(user=request.user)

    return render(request, "products/seller/product_form.html", {"form": form, "mode": "create"})


@seller_required
@throttle(SELLER_PRODUCT_MUTATE_RULE)
def seller_product_edit(request, pk: int):
    product = _get_owned_product_or_404(request, pk)

    if request.method == "POST":
        form = ProductForm(request.POST, instance=product, user=request.user)
        if form.is_valid():
            obj = form.save(commit=False)
            obj.full_clean()
            obj.save()
            messages.success(request, f"Listing saved for '{product.title}'.")
            return redirect("products:seller_list")
    else:
        form = ProductForm(instance=product, user=request.user)

    return render(
        request,
        "products/seller/product_form.html",
        {"form": form, "mode": "edit", "product": product},
    )


@seller_required
@throttle(SELLER_PRODUCT_MUTATE_RULE)
def seller_product_specs(request, pk: int):
    """Edit product specifications (different forms for MODEL vs FILE)."""
    product = _get_owned_product_or_404(request, pk)

    if product.kind == Product.Kind.MODEL:
        from .models import ProductPhysical

        physical, _created = ProductPhysical.objects.get_or_create(product=product)

        if request.method == "POST":
            form = ProductPhysicalForm(request.POST, instance=physical)
            if form.is_valid():
                form.save()
                messages.success(request, f"Specs saved for '{product.title}' (physical).")
                return redirect("products:seller_list")
        else:
            form = ProductPhysicalForm(instance=physical)

        return render(
            request,
            "products/seller/product_specs_form.html",
            {"form": form, "product": product, "kind": "MODEL"},
        )

    if product.kind == Product.Kind.FILE:
        from .models import ProductDigital

        digital, _created = ProductDigital.objects.get_or_create(product=product)

        if request.method == "POST":
            form = ProductDigitalForm(request.POST, instance=digital)
            if form.is_valid():
                form.save()
                messages.success(request, f"Specs saved for '{product.title}' (digital file).")
                return redirect("products:seller_list")
        else:
            form = ProductDigitalForm(instance=digital)

        return render(
            request,
            "products/seller/product_specs_form.html",
            {"form": form, "product": product, "kind": "FILE"},
        )

    messages.warning(request, "Product kind not recognized.")
    return redirect("products:seller_list")


@seller_required
@throttle(SELLER_UPLOAD_RULE)
def seller_product_images(request, pk: int):
    """
    Image upload/manage page.
    IMPORTANT UX:
    - After uploading an image, stay on this same page so the seller can keep uploading.
    """
    product = _get_owned_product_or_404(request, pk)

    def _next_sort_order() -> int:
        last = product.images.order_by("-sort_order", "-id").first()
        if not last:
            return 0
        try:
            return int(last.sort_order) + 1
        except Exception:
            return product.images.count()

    bulk_form = ProductImageBulkUploadForm()  # keep defined for template

    if request.method == "POST":
        form = ProductImageUploadForm(request.POST, request.FILES)
        if form.is_valid():
            with transaction.atomic():
                img: ProductImage = form.save(commit=False)
                img.product = product

                if img.sort_order is None:
                    img.sort_order = _next_sort_order()

                img.full_clean()
                img.save()

                if img.is_primary:
                    ProductImage.objects.filter(product=product).exclude(pk=img.pk).update(is_primary=False)

                if not ProductImage.objects.filter(product=product, is_primary=True).exists():
                    img.is_primary = True
                    img.save(update_fields=["is_primary"])

            messages.success(request, f"Image uploaded for '{product.title}'.")
            return redirect("products:seller_images", pk=product.pk)
    else:
        form = ProductImageUploadForm()

    images = product.images.all().order_by("sort_order", "id")
    return render(
        request,
        "products/seller/product_images.html",
        {"product": product, "form": form, "bulk_form": bulk_form, "images": images},
    )


@seller_required
@throttle(SELLER_DELETE_RULE)
def seller_product_image_delete(request, *args, **kwargs):
    pk = kwargs.get("pk")
    image_id = kwargs.get("image_id")
    product = _get_owned_product_or_404(request, pk)
    img = get_object_or_404(ProductImage, pk=image_id, product=product)

    if request.method == "POST":
        was_primary = img.is_primary
        img.delete()
        if was_primary:
            next_img = ProductImage.objects.filter(product=product).order_by("sort_order", "id").first()
            if next_img:
                next_img.is_primary = True
                next_img.save(update_fields=["is_primary"])
        messages.success(request, f"Image deleted for '{product.title}'.")
        return redirect("products:seller_images", pk=product.pk)

    return redirect("products:seller_images", pk=product.pk)


@seller_required
@throttle(SELLER_PRODUCT_MUTATE_RULE)
def seller_product_image_update(request, pk: int, image_id: int):
    product = _get_owned_product_or_404(request, pk)
    img = get_object_or_404(ProductImage, pk=image_id, product=product)

    if request.method != "POST":
        return redirect("products:seller_images", pk=product.pk)

    action = (request.POST.get("action") or "").strip()
    sort_value = (request.POST.get("sort_order") or "").strip()

    updated_fields: list[str] = []

    if sort_value:
        try:
            sort_order = int(sort_value)
            if sort_order < 0:
                sort_order = 0
            if img.sort_order != sort_order:
                img.sort_order = sort_order
                updated_fields.append("sort_order")
        except ValueError:
            messages.error(request, "Sort order must be a whole number.")

    if action == "make_primary":
        ProductImage.objects.filter(product=product).exclude(pk=img.pk).update(is_primary=False)
        if not img.is_primary:
            img.is_primary = True
            updated_fields.append("is_primary")

    if updated_fields:
        img.save(update_fields=updated_fields)
        if action == "make_primary":
            messages.success(request, f"Primary image updated for '{product.title}'.")
        elif "sort_order" in updated_fields:
            messages.success(request, f"Image order updated for '{product.title}'.")
    else:
        messages.info(request, "No changes to save.")

    return redirect("products:seller_images", pk=product.pk)


@seller_required
@throttle(SELLER_UPLOAD_RULE)
def seller_product_assets(request, pk: int):
    product = _get_owned_product_or_404(request, pk)

    if product.kind != Product.Kind.FILE:
        messages.info(request, "Assets are only for FILE listings.")
        return redirect("products:seller_list")
```

```python
    if request.method == "POST":
        form = DigitalAssetUploadForm(request.POST, request.FILES)
        if form.is_valid():
            asset: DigitalAsset = form.save(commit=False)
            asset.product = product
            asset.full_clean()
            asset.save()
            messages.success(request, f"Digital asset uploaded for '{product.title}'.")
            return redirect("products:seller_assets", pk=product.pk)
    else:
        form = DigitalAssetUploadForm()

    assets = product.digital_assets.all().order_by("id")
    return render(
        request,
        "products/seller/product_assets.html",
        {"product": product, "form": form, "assets": assets},
    )


@seller_required
def seller_product_preview(request, pk: int):
    product = _get_owned_product_or_404(request, pk)
    return _render_product_detail(request=request, product=product, log_event=False)


@seller_required
@throttle(SELLER_DELETE_RULE)
def seller_product_asset_delete(request, *args, **kwargs):
    pk = kwargs.get("pk")
    asset_id = kwargs.get("asset_id")
    product = _get_owned_product_or_404(request, pk)
    asset = get_object_or_404(DigitalAsset, pk=asset_id, product=product)

    if request.method == "POST":
        asset.delete()
        messages.success(request, f"Digital asset deleted for '{product.title}'.")
        return redirect("products:seller_assets", pk=product.pk)

    return redirect("products:seller_assets", pk=product.pk)


@seller_required
@throttle(SELLER_PRODUCT_MUTATE_RULE)
def seller_product_toggle_active(request, pk: int):
    product = _get_owned_product_or_404(request, pk)
    if request.method == "POST":
        product.is_active = not product.is_active
        product.save(update_fields=["is_active"])
        messages.success(
            request,
            f"Listing '{product.title}' is now {'active' if product.is_active else 'inactive (draft)'}.",
        )
    return redirect("products:seller_list")


@seller_required
@throttle(SELLER_DELETE_RULE)
def seller_product_delete(request, pk: int):
    """Delete a product listing (cascades to images/assets/specs)."""
    product = _get_owned_product_or_404(request, pk)

    if request.method == "POST":
        product_title = product.title
        product.delete()
        messages.success(request, f"Product '{product_title}' has been deleted.")
        return redirect("products:seller_list")

    return redirect("products:seller_list")


@seller_required
@throttle(SELLER_PRODUCT_MUTATE_RULE)
def seller_product_duplicate(request, pk: int):
    """
    Create a copy of a product as a new draft listing.
    """
    original = _get_owned_product_or_404(request, pk)

    if request.method != "POST":
        return redirect("products:seller_list")

    with transaction.atomic():
        new_product = Product(
            seller=request.user,
            kind=original.kind,
            title=f"{original.title} (Copy)",
            slug="",
            slug_is_manual=False,
            short_description=original.short_description,
            description=original.description,
            category=original.category,
            subcategory=original.subcategory,
            price=original.price,
            is_free=original.is_free,
            is_active=False,
            complexity_level=original.complexity_level,
            print_time_hours=original.print_time_hours,
            max_purchases_per_buyer=original.max_purchases_per_buyer,
            is_featured=False,
            is_trending=False,
        )

        new_product.slug_is_manual = False
        new_product.slug = ""

        new_product.full_clean()
        new_product.save()

        for img in original.images.all().order_by("sort_order", "id"):
            new_name = _safe_storage_copy(source_field=img.image, dest_prefix="product_images")
            if not new_name:
                continue
            ProductImage.objects.create(
                product=new_product,
                image=new_name,
                alt_text=img.alt_text,
                is_primary=img.is_primary,
                sort_order=img.sort_order,
            )
        if new_product.images.exists() and not new_product.images.filter(is_primary=True).exists():
            first = new_product.images.order_by("sort_order", "id").first()
            if first:
                first.is_primary = True
                first.save(update_fields=["is_primary"])

        if original.kind == Product.Kind.FILE:
            for asset in original.digital_assets.all().order_by("id"):
                new_asset_name = _safe_storage_copy(source_field=asset.file, dest_prefix="digital_assets")
                if not new_asset_name:
                    continue
                DigitalAsset.objects.create(
                    product=new_product,
                    file=new_asset_name,
                    original_filename=asset.original_filename,
                    file_type=asset.file_type,
                    zip_contents=asset.zip_contents,
                )

    messages.success(request, "Product duplicated as a draft. Edit files, images, and specs from My Listings.")
    return redirect("products:seller_list")
```

orders\webhooks.py

```
# orders/webhooks.py

from __future__ import annotations

import logging
from dataclasses import dataclass

from django.db import transaction
from django.http import HttpRequest, HttpResponse, HttpResponseBadRequest
from django.views.decorators.csrf import csrf_exempt

from payments.models import SellerBalanceEntry
from payments.services import ensure_sale_balance_entries_for_paid_order

from .models import Order, OrderEvent, StripeWebhookEvent, _send_order_failed_email
from .stripe_service import create_transfers_for_paid_order, verify_and_parse_webhook

logger = logging.getLogger(__name__)


def _extract_shipping_from_session_obj(session_obj: dict) -> dict:
    shipping_details = session_obj.get("shipping_details") or {}
    customer_details = session_obj.get("customer_details") or {}

    name = shipping_details.get("name") or customer_details.get("name") or ""
    phone = customer_details.get("phone") or ""
    addr = shipping_details.get("address") or customer_details.get("address") or {}

    return {
        "name": name,
        "phone": phone,
        "line1": addr.get("line1") or "",
        "line2": addr.get("line2") or "",
        "city": addr.get("city") or "",
        "state": addr.get("state") or "",
        "postal_code": addr.get("postal_code") or "",
        "country": addr.get("country") or "",
    }


def _get_order_id_from_event(event: dict) -> str:
    obj = (event.get("data") or {}).get("object") or {}
    metadata = obj.get("metadata") or {}

    order_id = (metadata.get("order_id") or "").strip()
    if order_id:
        return order_id

    order_id = (obj.get("client_reference_id") or "").strip()
    if order_id:
        return order_id

    return ""


def _record_event_once(*, stripe_event_id: str, event_type: str) -> bool:
    _, created = StripeWebhookEvent.objects.get_or_create(
        stripe_event_id=stripe_event_id,
        defaults={"event_type": event_type or ""},
    )
    return created


def _transfers_already_created(order: Order) -> bool:
    return order.events.filter(type=OrderEvent.Type.TRANSFER_CREATED).exists()


@dataclass(frozen=True)
class _RefundAllocation:
    order_item_id: str
    seller_id: str
    debit_cents: int


def _allocate_refund_across_items(*, order: Order, refund_total_cents: int) -> list[_RefundAllocation]:
    """
    Allocate refund across eligible items proportional to seller_net_cents.

    Integer-safe:
      - share_i = floor(refund_total * net_i / total_net) for all but last
      - last gets remainder
      - each seller debit is capped at that line's net
    """
    refund_total_cents = max(0, int(refund_total_cents or 0))
    items_all = list(order.items.all())
    if refund_total_cents <= 0 or not items_all:
        return []

    eligible = [it for it in items_all if int(getattr(it, "seller_net_cents", 0) or 0) > 0]
    if not eligible:
        return []

    nets = [int(it.seller_net_cents or 0) for it in eligible]
    total_net = sum(nets)
    if total_net <= 0:
        return []

    allocations: list[_RefundAllocation] = []
    remaining = refund_total_cents

    for idx, it in enumerate(eligible):
        net = int(it.seller_net_cents or 0)
        if net <= 0:
            continue

        if idx == len(eligible) - 1:
            share = remaining
        else:
            share = (refund_total_cents * net) // total_net
            share = max(0, min(share, remaining))

        remaining -= share

        debit = min(share, net)
        if debit <= 0:
            continue

        allocations.append(
            _RefundAllocation(
                order_item_id=str(it.pk),
                seller_id=str(it.seller_id),
                debit_cents=int(debit),
            )
        )

    return allocations


def _maybe_mark_order_refunded(*, order: Order, refunded_total_cents: int, note: str) -> None:
    refunded_total_cents = int(refunded_total_cents or 0)
    if refunded_total_cents <= 0:
        return

    total = int(order.total_cents or 0)
    if total > 0 and refunded_total_cents < total:
        OrderEvent.objects.create(
            order=order,
            type=OrderEvent.Type.WARNING,
            message=f"Partial refund observed ({refunded_total_cents}c of {total}c). {note}",
        )
        return

    if order.status != Order.Status.REFUNDED:
        order.status = Order.Status.REFUNDED
        order.save(update_fields=["status", "updated_at"])
        OrderEvent.objects.create(order=order, type=OrderEvent.Type.REFUNDED, message=note)


@csrf_exempt
def stripe_webhook(request: HttpRequest) -> HttpResponse:
    payload = request.body
    sig_header = request.headers.get("Stripe-Signature", "")

    if not sig_header:
        return HttpResponseBadRequest("Missing signature")

    try:
        event = verify_and_parse_webhook(payload, sig_header)
    except Exception:
        return HttpResponseBadRequest("Invalid signature")

    stripe_event_id = (event.get("id") or "").strip()
    event_type = (event.get("type") or "").strip()
    if not stripe_event_id or not event_type:
        return HttpResponse(status=200)

    if not _record_event_once(stripe_event_id=stripe_event_id, event_type=event_type):
        return HttpResponse(status=200)

    obj = (event.get("data") or {}).get("object") or {}
    order_id = _get_order_id_from_event(event)

    if not order_id and event_type == "payment_intent.payment_failed":
        payment_intent_id = (obj.get("id") or "").strip()
        if payment_intent_id:
            fallback = Order.objects.filter(stripe_payment_intent_id=payment_intent_id).first()
            if fallback:
                order_id = str(fallback.pk)

    if not order_id:
        logger.warning("Stripe event %s (%s) missing order_id mapping", stripe_event_id, event_type)
        return HttpResponse(status=200)

    try:
        with transaction.atomic():
            order = Order.objects.select_for_update().get(pk=order_id)

            if event_type == "checkout.session.completed":
                session_id = (obj.get("id") or "").strip()
                payment_intent_id = (obj.get("payment_intent") or "").strip()

                updated_fields: list[str] = []
                if session_id and not order.stripe_session_id:
                    order.stripe_session_id = session_id
                    updated_fields.append("stripe_session_id")
                if payment_intent_id and not order.stripe_payment_intent_id:
                    order.stripe_payment_intent_id = payment_intent_id
                    updated_fields.append("stripe_payment_intent_id")
                if updated_fields:
                    updated_fields.append("updated_at")
                    order.save(update_fields=updated_fields)

                # 1) Mark paid (this should also compute OrderItem snapshots/ledger fields)
                order.mark_paid(payment_intent_id=payment_intent_id, session_id=session_id)

                # 2) Save shipping snapshot if present
                ship = _extract_shipping_from_session_obj(obj)
                if any([ship["line1"], ship["city"], ship["postal_code"], ship["country"]]):
                    order.set_shipping_from_stripe(**ship)

                # 3) IMPORTANT: record SALE credits in the seller ledger BEFORE payouts
                ensure_sale_balance_entries_for_paid_order(order=order)

                # 4) Create transfers/payouts (this is where your -PAYOUT entries are created)
                create_transfers_for_paid_order(order=order, payment_intent_id=payment_intent_id)
                return HttpResponse(status=200)

            if event_type == "checkout.session.expired":
                order.mark_canceled(note="Checkout session expired")
                return HttpResponse(status=200)

            if event_type == "payment_intent.payment_failed":
                failure_message = (obj.get("last_payment_error") or {}).get("message") or "Payment failed"
                OrderEvent.objects.create(
                    order=order,
                    type=OrderEvent.Type.WARNING,
                    message=f"Payment failed (event={stripe_event_id})",
                )
                _send_order_failed_email(order, reason=failure_message)
                return HttpResponse(status=200)

            if event_type in {"charge.refunded", "refund.created", "refund.updated"}:
                refunded_cents = int(obj.get("amount_refunded") or obj.get("amount") or 0)
                payout_created = _transfers_already_created(order)

                if payout_created and refunded_cents > 0:
                    allocs = _allocate_refund_across_items(order=order, refund_total_cents=refunded_cents)

                    for a in allocs:
                        SellerBalanceEntry.objects.create(
                            seller_id=a.seller_id,
                            amount_cents=-int(a.debit_cents),
                        reason=SellerBalanceEntry.Reason.REFUND,
                        order=order,
                        order_item_id=a.order_item_id,
                        note=f"Stripe refund via {event_type} (event={stripe_event_id})",
                    )

                OrderEvent.objects.create(
                    order=order,
                    type=OrderEvent.Type.WARNING,
                    message=(
                        f"Refund received after payout. Recorded seller debits "
                        f"(refund={refunded_cents}c, event={stripe_event_id}, type={event_type})."
                    ),
                )
            else:
                OrderEvent.objects.create(
                    order=order,
                    type=OrderEvent.Type.WARNING,
                    message=(
                        f"Refund received (refund={refunded_cents}c, type={event_type}, event={stripe_event_id}). "
                        f"No seller debits recorded (payout_created={payout_created})."
                    ),
                )

            _maybe_mark_order_refunded(
                order=order,
                refunded_total_cents=refunded_cents,
                note=f"Stripe refund observed ({event_type}, {refunded_cents}c, event={stripe_event_id})",
            )
            return HttpResponse(status=200)

        if event_type in {"charge.dispute.created", "charge.dispute.updated"}:
            status = (obj.get("status") or "").strip().lower()
            OrderEvent.objects.create(
                order=order,
                type=OrderEvent.Type.WARNING,
                message=f"Dispute event: {event_type} status={status or 'unknown'} event={stripe_event_id}",
            )
            return HttpResponse(status=200)

        if event_type == "charge.dispute.closed":
            status = (obj.get("status") or "").strip().lower()
            payout_created = _transfers_already_created(order)

            if status == "lost":
                if payout_created:
                    for it in order.items.all():
                        net = int(it.seller_net_cents or 0)
                        if net <= 0:
                            continue

                        SellerBalanceEntry.objects.create(
                            seller_id=str(it.seller_id),
                            amount_cents=-net,
                            reason=SellerBalanceEntry.Reason.CHARGEBACK,
                            order=order,
                            order_item=it,
                            note=f"Chargeback lost (event={stripe_event_id})",
                        )

                    OrderEvent.objects.create(
                                order=order,
                                type=OrderEvent.Type.WARNING,
                                message=(
                                    "Chargeback lost. Seller debited net (payout already created). "
                                    "Dispute fee may require manual adjustment."
                                ),
                            )
                        else:
                            OrderEvent.objects.create(
                                order=order,
                                type=OrderEvent.Type.WARNING,
                                message="Chargeback lost before payout. No seller debits recorded (no payout created).",
                            )

                        if order.status != Order.Status.REFUNDED:
                            order.status = Order.Status.REFUNDED
                            order.save(update_fields=["status", "updated_at"])
                                OrderEvent.objects.create(order=order, type=OrderEvent.Type.REFUNDED, message="Chargeback
    lost")

                        return HttpResponse(status=200)

                    OrderEvent.objects.create(
                        order=order,
                        type=OrderEvent.Type.WARNING,
                        message=f"Chargeback closed with status={status or 'unknown'} event={stripe_event_id}",
                    )
                    return HttpResponse(status=200)

                return HttpResponse(status=200)

        except Order.DoesNotExist:
            return HttpResponse(status=200)
        except Exception:
            logger.exception(
                "Stripe webhook processing failed event=%s type=%s order=%s",
                stripe_event_id,
                event_type,
                order_id,
            )
            return HttpResponse(status=200)
```

templates\emails\welcome.html

```
{# templates/emails/welcome.html #}
{% extends "emails/base.html" %}
{% block content %}
  <h1 style="margin:0 0 12px;font-size:20px;">Welcome to Home Craft 3D</h1>
  <p style="margin:0 0 12px;">Hi{% if username %} {{ username }}{% endif %}, thanks for joining our maker community.</p>

  <div style="margin:16px 0;padding:16px;border:1px solid #e5e7eb;border-radius:10px;background:#f9fafb;">
    <div style="font-size:12px;color:#6b7280;margin-bottom:6px;">Get started</div>
    <div style="display:flex;flex-wrap:wrap;gap:8px;">
      <a href="{{ shop_url }}" style="color:#2563eb;text-decoration:none;">Browse products</a>
      <span style="color:#9ca3af;">|</span>
      <a href="{{ profile_url }}" style="color:#2563eb;text-decoration:none;">Complete your profile</a>
      <span style="color:#9ca3af;">|</span>
      <a href="{{ seller_url }}" style="color:#2563eb;text-decoration:none;">Become a seller</a>
    </div>
  </div>

  <p style="margin:16px 0 0;color:#6b7280;font-size:13px;">If you did not create this account, you can ignore this email.</p>
{% endblock %}
```

config\wsgi.py

```
# config/wsgi.py
"""
WSGI config for config project.

It exposes the WSGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/6.0/howto/deployment/wsgi/
"""

import os

from django.core.wsgi import get_wsgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'config.settings.base')

application = get_wsgi_application()
```