Soggy Potatoes E-Commerce Website - Project Overview

Project Name: *Soggy Potatoes Sticker Shop*
Owner: *Joana*
Developer: *[Your Name]*
Project Start Date: *January 6, 2026*
Current Status: *Phase 0 - Planning & Setup*

🎯 *PROJECT GOALS*

Primary Objectives

1. ***Zero/Minimal Cost****: Keep infrastructure costs at $0 during development and early launch*
2. ***Full E-Commerce Functionality****: Professional shopping cart, checkout, order management*
3. ***Community Features****: Forums with reactions, upvotes, user interactions*
4. ***Scalability****: Start small (20-25 concurrent users) but architect for growth (1000s eventually)*
5. ***Learning Experience****: Serve as template for future business websites*
6. ***Professional Quality****: Industry-standard practices throughout*

Core Requirements

- ✅ *Shopping cart with multiple product pages*
- ✅ *User authentication (login/logout/password recovery)*
- ✅ *User profiles with order history*
- ✅ *Product search functionality*
- ✅ *Community forums (posts, reactions, upvotes)*
- ✅ *About page for Joana*
- ✅ *Admin panel for managing products/orders*
- ✅ *Payment processing (Stripe/PayPal - Phase 2)*
- ✅ *Email notifications for orders*

🏗️ *TECHNOLOGY STACK*

Core Framework: Django 5.x

Rationale:

- *Python-based (continuation from previous work)*
- *Built-in user authentication, admin panel, ORM*
- *Excellent e-commerce plugins (django-oscar, django-shop)*
- *Strong forum solutions (Misago, django-machina)*
- *Runs locally AND deploys to free cloud hosting*
- *Scales from 10 to 10,000+ users*
- *Battle-tested by Instagram, Spotify, Pinterest*

Database: PostgreSQL 16

Rationale:

- *Free, open-source, robust*
- *Runs identically locally and in cloud*
- *Handles complex queries for forums/e-commerce*
- *Excellent Django support*
- *Industry standard*

Frontend: Django Templates + Bootstrap 5

Rationale:

- *Keep it simple initially (no separate React complexity)*
- *Server-side rendering (faster, better SEO)*
- *Bootstrap provides professional UI out-of-box*
- *Can add React/Vue later if needed without rebuilding*

Payment Processing: Stripe (Phase 2)

Rationale:

- *No setup fees, only transaction fees (2.9% + $0.30)*
- *Excellent Django integration*
- *More developer-friendly than PayPal*
- *Handles sales tax automatically*
- *Best documentation*

File Storage: Local → AWS S3 Free Tier (Future)

Rationale:

- *Local storage during development*
- *AWS S3 free tier: 5GB storage, 20k GET requests/month*
- *Easy Django integration with django-storages*

Email: SendGrid Free Tier

Rationale:

- *100 emails/day free forever*
- *Perfect for order confirmations*
- *Professional email delivery*
- *Simple Django integration*

🚀 ***DEPLOYMENT STRATEGY***

Phase 1: Local Development (Current)

- ***Location:*** *Your MacBook Air*
- ***Access:*** *Only when Mac is on*
- ***URL:*** *http://localhost:8000 or http://127.0.0.1:8000*
- ***Auto-Start:*** *Will configure Django to auto-start on Mac boot*
- ***Purpose:*** *Build and test all features safely*

Phase 2: Free Cloud Hosting (When ready for public)

- ***Platform:*** *Railway.app or Render.com*
- ***Free Tier Limits:***
  - *Railway: 500 hours/month, 512MB RAM, 1GB storage*
  - *Render: 750 hours/month, 512MB RAM, persistent storage*
- ***URL:*** *soggypotatoes.up.railway.app (or similar)*
- ***Estimated cost:*** *$0/month for <1000 visitors/month*

Phase 3: Custom Domain (Optional)

- ***Domain:*** *soggypotatoes.com or soggypotatoes.shop*
- ***Cost:*** *~$12/year (.com) or ~$5/year (.shop)*
- ***DNS:*** *Cloudflare (free)*

Phase 4: Paid Hosting (When traffic grows)

- ***Trigger:*** *Exceeding free tier limits OR >10 orders/day*
- ***Platform:*** *DigitalOcean, AWS, or Heroku*
- ***Estimated cost:*** *$5-25/month depending on traffic*

📁 *PROJECT STRUCTURE*

```
soggy-potatoes/

├── docs/                    # All documentation (THIS FOLDER)

│   ├── PROJECT_OVERVIEW.md       # This file - master reference

│   ├── SYSTEM_OPERATIONS_MANUAL.md # How to run/maintain the system

│   ├── TECHNICAL_PROTOCOLS.md     # Technical decisions & configurations

│   ├── DEVELOPMENT_LOG.md        # Chronological progress log

│   └── TROUBLESHOOTING_GUIDE.md   # Common issues & solutions

│

├── soggy_potatoes/            # Django project root

│   ├── manage.py              # Django management script

│   ├── soggy_potatoes/         # Project settings

│   │   ├── settings.py         # Main configuration

│   │   ├── urls.py             # URL routing

│   │   └── wsgi.py             # Web server interface

│   │

│   ├── shop/                  # E-commerce app

│   │   ├── models.py           # Product, Order, Cart models

│   │   ├── views.py            # Shopping logic

│   │   ├── templates/          # HTML templates

│   │   └── static/             # CSS, JS, images
│   │
│   ├── forum/              # Community forum app
│   │   ├── models.py         # Post, Comment, Vote models
│   │   ├── views.py          # Forum logic
│   │   └── templates/        # Forum HTML
│   │
│   ├── users/              # User authentication app
│   │   ├── models.py         # User profile model
│   │   ├── views.py          # Login, register, profile
│   │   └── templates/        # Auth HTML
│   │
│   └── static/             # Global static files
│       ├── css/
│       ├── js/
│       └── images/
│
├── requirements.txt          # Python dependencies
├── .env                    # Environment variables (SECRET!)
├── .gitignore              # Files to ignore in version control
└── README.md                 # Quick start guide
```

🔐 SECURITY CONSIDERATIONS

Development (Local)

- *Django debug mode: ENABLED*
- *Secret keys: Stored in .env file (never commit!)*
- *HTTPS: Not required (localhost)*

Production (Cloud)

- *Django debug mode: DISABLED*
- *Secret keys: Environment variables on hosting platform*
- *HTTPS: Required (free with Railway/Render)*
- *CSRF protection: Enabled*
- *SQL injection protection: Django ORM handles this*
- *Password hashing: Django's built-in PBKDF2*

📊 *DEVELOPMENT PHASES*

Phase 0: Setup & Planning (CURRENT)

- ✅ *Define requirements*
- ✅ *Choose tech stack*
- ✅ *Create documentation structure*
- 🔄 *Set up development environment*
- 🔄 *Create initial Django project*

Phase 1: Core Infrastructure (Week 1-2)

- *Django project initialization*
- *Database setup (PostgreSQL)*
- *User authentication (login/register/logout)*
- *Basic homepage and navigation*
- *Admin panel configuration*

Phase 2: E-Commerce Foundation (Week 2-3)

- *Product model (stickers)*
- *Product listing pages*
- *Product detail pages*
- *Shopping cart functionality*
- *Search functionality*

Phase 3: User Features (Week 3-4)

- *User profiles*

- *Order history*
- *Password recovery*
- *Email verification*

Phase 4: Checkout & Orders (Week 4-5)

- *Checkout page*
- *Order creation (without payment)*
- *Order management in admin*
- *Email notifications*

Phase 5: Community Forum (Week 5-7)

- *Forum structure (categories, threads)*
- *Post creation and replies*
- *Upvote/downvote system*
- *Reactions to posts*

Phase 6: Payment Integration (Week 7-8)

- *Stripe account setup*
- *Payment processing*
- *Order confirmation*
- *Inventory management*

Phase 7: Polish & Testing (Week 8-9)

- *UI refinements*
- *Mobile responsiveness*
- *Bug fixes*
- *Performance optimization*
- *About page for Joana*

Phase 8: Deployment (Week 9-10)

- *Deploy to Railway/Render*
- *Domain configuration (if purchased)*
- *SSL certificate*
- *Final testing*
- *Go live!*

💰 *COST BREAKDOWN (Estimated)*

During Development (Months 1-3)

- ***Hosting:*** *$0 (local development)*
- ***Database:*** *$0 (PostgreSQL local)*
- ***Email:*** *$0 (SendGrid free tier)*
- ***Total:*** *$0/month*

After Launch - Small Scale (<100 orders/month)

- ***Hosting:*** *$0 (Railway/Render free tier)*
- ***Domain:*** *$0-12/year (optional)*
- ***Payment processing:*** *~$0.35/transaction (Stripe fees)*
- ***Email:*** *$0 (SendGrid free tier)*
- ***Total:*** *~$0-1/month*

Growth Phase (>500 orders/month)

- ***Hosting:*** *$15-25/month (paid tier)*
- ***Domain:*** *$12/year*
- ***Payment processing:*** *~$0.35/transaction*
- ***Email:*** *$0 (still free tier)*
- ***Total:*** *~$15-30/month*

🚨 ***CRITICAL SUCCESS FACTORS***

What Makes This Project Work

1. ***Comprehensive Documentation****: Every decision documented before coding*
2. ***Modular Architecture****: Each feature is independent (forum, shop, auth)*
3. ***Version Control****: Git tracks every change*
4. ***Testing at Every Step****: Don't build on broken foundations*
5. ***Single Source of Truth****: This document + technical protocols*
6. ***Clear Rollback Points****: Can always revert to working state*

Common Pitfalls to Avoid

1. *❌ Forgetting what infrastructure we're using → ✅ Check TECHNICAL_PROTOCOLS.md*
2. *❌ Making changes without documenting → ✅ Update DEVELOPMENT_LOG.md*
3. *❌ Choosing incompatible tools mid-project → ✅ All decisions pre-vetted*
4. *❌ Losing track of what works → ✅ Tag working versions in Git*
5. *❌ Rushing without understanding → ✅ Break down complex tasks*

🎓 *LEARNING RESOURCES*

For You (Developer)

- ***Django Official Tutorial**: https://docs.djangoproject.com/en/5.0/intro/tutorial01/*
- ***Django for Beginners Book**: https://djangoforbeginners.com/*
- ***Real Python Django Tutorials**: https://realpython.com/tutorials/django/*

For Troubleshooting

- ***Django Documentation**: https://docs.djangoproject.com/*
- ***Stack Overflow**: Tag questions with [django]*
- ***Django Forum**: https://forum.djangoproject.com/*

📞 *NEXT STEPS*

1. *Review this document with Joana (if needed)*
2. *Set up Mac development environment*
3. *Create remaining documentation artifacts*
4. *Initialize Django project*
5. *Begin Phase 1 development*

📝 *DOCUMENT REVISION HISTORY*

Date	***Version***	***Changes***	***Author***
2026-01-06	*1.0*	*Initial project overview created*	*Claude*

END OF PROJECT OVERVIEW