

# Django Project Plan: News Aggregator Platform

## Project Idea: Build a Robust News Aggregator Platform

Think of it as a personal 'Flipboard' or feed curator where users can follow topics, read aggregated articles, and get personalized content.

### Why it's a great showcase:

- Realworld value: Aggregates data from multiple sources (APIs or RSS), filters duplicates, categorizes content.
- Django strengths: Leverage Django ORM, custom models, admin customization, background tasks, pagination.
- Advanced features:
  - Scheduled fetches using Celery + CeleryBeat or Django Q.
  - Full-text search with Elasticsearch or PostgreSQL trigram.
  - User auth + personalized topic subscriptions.
  - REST API endpoints via Django REST Framework.
  - Frontend interface with filters/sort, plus a nice UI.

### Insights from the community:

On Reddit, developers highlight:

For a showcase piece I suggest just cloning a major app ... Make your own Pinterest.

A truly standout project ... consists of a unique, or niche problem youve tackled with ingenuity.

### Proposed Feature Set

Feature | Purpose

-----|-----

User accounts | Sign up, login, profile settings

Subscription model | Follow topics/categories (e.g. Tech, Sports)

Fetcher service | Periodic scraping or API-based news ingestion

Backend tasks | Use Celery to schedule and perform ingestion

Search & filters | Full-text search, filters by date or topic

REST API | Expose curated feeds programmatically

Frontend UI | Responsive listing, cards, pagination

Admin dashboard | Monitor feed activity, manage publishers/posts

### **Optional extensions:**

- Personalized email digests or push notifications.
- Graph analytics: user engagement or trending topics.
- User comments or ratings on articles.

### **Why This Stands Out as a Portfolio Project**

- Integrates multiple Django components.
- More meaningful than a to-do app.
- Shows production readiness: task queues, security, deployment.
- Demonstrates modular, scalable design.

### **Suggested Timeline**

1. Planning & Design (12 days)
2. Core MVP (35 days)
3. Advanced Add-ons (35 days)
4. Styling & UX polish (12 days)
5. Testing & QA (1 day)
6. Deployment

### **Ready to Start?**

I can walk you through each step with code examples, deployment instructions, and best practices.