

FaundaTrek Backend - Requirements & Structure

Project Overview

FaundaTrek is a social e-commerce platform with five main sections:

1. Profiles - Entrepreneur & investor profiles with personal and business details.
2. Stories - A content-sharing section where users post updates, photos, and progress.
3. Pitch Platform - Entrepreneurs upload business pitches for potential investors/supporters.
4. Donation Platform - A system where users can make monetary or material contributions.
5. Connect - Online chat system for real-time communication between users.

Tech Stack

- Backend Framework: Django + Django REST Framework (DRF)
- Database: PostgreSQL (preferred)
- Authentication: JWT (via `django-rest-framework-simplejwt`)
- File Storage: Local for development, AWS S3/DigitalOcean Spaces for production
- API Format: RESTful JSON
- Deployment Target: Cloud-based (Heroku, Railway, or AWS)

Core Modules & Features

A. Profiles

- User registration (email, password)
- Role selection (Entrepreneur / Investor / Donor / Admin)
- Profile fields: Name, bio, location, profile picture, business/organization name, category/industry tags, social media links
- Edit profile & update picture
- View other users' profiles

Example Endpoints:

POST `/api/auth/register/`

POST `/api/auth/login/`

GET `/api/profile/{id}/`

PUT `/api/profile/{id}/`

B. Stories

- Users can post text, images, and short videos
- Like & comment system
- Feed showing stories from all users or connections
- Pagination for faster loading

Example Endpoints:

GET /api/stories/

POST /api/stories/

GET /api/stories/{id}/

POST /api/stories/{id}/like/

POST /api/stories/{id}/comment/

C. Pitch Platform

- Entrepreneurs can upload pitches: title, description, pitch video or PDF, funding goal
- Investors/supporters can view and save pitches
- Search & filter pitches by category, funding stage, or location

Example Endpoints:

GET /api/pitches/

POST /api/pitches/

GET /api/pitches/{id}/

D. Donation Platform

- Supports both monetary donations and material donations
- Donation listing includes: title, description, category, image, target goal, deadline
- Users can donate via integrated payment gateway (Phase 2)
- Track donation progress

Example Endpoints:

GET /api/donations/

POST /api/donations/

POST /api/donations/{id}/contribute/

E. Connect (Online Chat)

- Real-time 1-on-1 messaging

- Message fields: sender_id, receiver_id, content, timestamp, read status
- Optional WebSocket integration for instant updates (Phase 2)

Example Endpoints:

GET /api/messages/{user_id}/

POST /api/messages/

Database Schema (High-Level)

Users Table: id, name, email, password, role, bio, location, profile_pic, created_at

Stories Table: id, user_id (FK), content, image/video, created_at

Pitches Table: id, user_id (FK), title, description, pitch_file, funding_goal, created_at

Donations Table: id, user_id (FK), title, description, type, target_goal, deadline, created_at

Messages Table: id, sender_id, receiver_id, content, timestamp, read_status

Development Guidelines

- Use DRF serializers for all models
- JWT for authentication
- Use pagination on all list endpoints
- Allow CORS for Flutter frontend
- Separate .env for secrets
- Follow RESTful naming conventions