📱 Appigram

A Full-Stack Social Feed Application

🚀 Overview

Appigram is a mini social networking app where users can:

- 🔐 Register & Login (JWT Authentication)
- 📝 Create posts with text + images
- 👍 Like/Unlike posts
- 💬 Comment on posts
- ↩️ Reply to comments
- ❤️ Like comments & replies
- 👤 View feed (public + own private posts)
- 🚪 Logout

The project demonstrates a complete **React + Django full-stack workflow**.

🏗️ Tech Stack

Frontend

- ⚛️ React + Vite
- ⚡ Axios
- 🎨 CSS (based on the provided UI design)
- 🔐 Context API for authentication

Backend

- 🐍 Django

- 🔧 Django REST Framework
- 🔑 SimpleJWT for token authentication
- 🗄 SQLite3
- 🌐 CORS Headers

📂 Project Structure

```
appigram_backend/
├── appigram/ # Django project configuration
├── users/ # Authentication & registration
├── feed/ # Posts, comments, replies
├── media/ # Image uploads
└── manage.py
```

```
appigram_frontend/
├── src/
| ├── components/ # UI components
| ├── contexts/ # AuthContext.jsx
| ├── api/ # Axios instance
| └── App.jsx
└── index.html
```

⚙️ Backend Setup (Django)

1️⃣ Create Virtual Environment

```
cd appigram_social_media

python -m venv venv
```

2️⃣ Activate Environment

```
# Windows

venv\Scripts\activate

# Linux/Mac

source venv/bin/activate
```

3️⃣ Install Dependencies

```
pip install -r requirements.txt
```

4️⃣ Apply Migrations

```
cd appigram
```

```
python manage.py migrate
```

5️⃣ Create Superuser (optional)

```
python manage.py createsuperuser
```

6️⃣ Start the Server

```
python manage.py runserver
```

Backend available at:

👉 http://127.0.0.1:8000/

⚙️ Frontend Setup (React + Vite) 1️⃣ Install Packages

```
cd ..
```

```
cd appigram_frontend
```

```
npm install
```

2️⃣ Start Dev Server

```
npm run dev
```

Frontend available at:

👉 http://localhost:5173/

🔗 API Endpoints

🔐 Authentication

Method	Endpoint	Description
POST	/api/auth/register/	Register new user
POST	/api/auth/token/	Login & get JWT

📝 Posts

Method	Endpoint	Description
GET	/api/posts/	Fetch feed

POST /api/posts/ Create post (with image)

POST /api/posts//toggle_like/ Like/Unlike

💬 Comments

Method Endpoint Description

POST /api/posts/<post_id>/comments/ Create comment

POST /api/comments//toggle_like/ Like/Unlike comment

↩️ Replies

| POST | /api/comments/<comment_id>/replies/ |

| POST | /api/replies//toggle_like/ |

🖼 Media Uploads Images stored in:

appigram/media/posts/

Ensure Django settings include:

```
MEDIA_URL = "/media/"

MEDIA_ROOT = BASE_DIR / "media"
```

and in urls.py:

```
urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

🔐 Authentication Flow

User logs in → React sends:

```
{

  "username": "email@example.com",

  "password": "password123"

}
```

Django issues JWT (access + refresh)

React stores token in localStorage

Axios attaches:

```
Authorization: Bearer <token>
```

🛠 Production Build

Frontend:

```
npm run build
```

Backend:

```
python manage.py collectstatic
```

🧑‍💻 Author Mahafuzur Rahman Full Stack Developer 🌐 React | Django | REST | TailwindCSS