Art Shop - Django + React Application

A full-stack art marketplace application built with Django REST API backend and React frontend.

Features

- User authentication (login/register)
- Art gallery with image support
- Shopping cart functionality
- Order management system
- Admin dashboard for artwork and order management
- Responsive React frontend
- RESTful API backend

Project Structure

```
artshop-complete/
├── backend/
│   ├── api/                # Django app
│   ├── artshop/            # Django project settings
│   ├── manage.py
│   ├── requirements.txt
│   └── db.sqlite3
├── frontend/
│   ├── src/
│   │   └── App.jsx
│   ├── package.json
│   ├── vite.config.js
│   └── dist/               # Build output
├── Dockerfile
└── README.md
```

Prerequisites

- Python 3.8+
- Node.js 18+ (with npm)
- Git

Setup Methods

Choose one of the following methods to run the project:

Method 1: Docker Setup (Recommended)

Build and Run with Docker

1. **Clone the repository:**

```bash
git clone <your-repo-url>
cd artshop-complete
```

2. **Build the Docker image:**

```bash
docker build -t artshop .
```

3. **Run the container:**

```bash
docker run -p 8000:8000 artshop
```

4. **Access the application:**
   - Main app: http://localhost:8000
   - Admin panel: http://localhost:8000/admin

Login Credentials (Docker)

- Regular user: `demo` / `demo123`
- Admin user: `admin` / `admin123`

Method 2: Local Development Setup

Prerequisites Check

Verify you have the required software:

```bash
```

```
python --version    # Should be 3.8+
node --version      # Should be 18+
npm --version       # Should be 8+
```

Backend Setup

1. **Navigate to backend directory:**

```bash
cd backend
```

2. **Create virtual environment:**

```bash
python -m venv .venv
```

3. **Activate virtual environment: Windows (Command Prompt):**

```cmd
.venv\Scripts\activate.bat
```

Windows (PowerShell):

```powershell
.venv\Scripts\Activate.ps1
```

macOS/Linux:

```bash
source .venv/bin/activate
```

4. **Install Python dependencies:**

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

5. **Run database migrations:**

```bash
python manage.py makemigrations
python manage.py migrate
```

6. **Create sample data:**

```bash
python manage.py seed_data
```

7. **Start Django server:**

```bash
python manage.py runserver 127.0.0.1:8000
```

Frontend Setup

Open a new terminal/command prompt and run:

1. **Navigate to frontend directory:**

```bash
cd frontend
```

2. **Install Node.js dependencies:**

```bash
npm install
```

3. **Start React development server:**

```bash
npm run dev
```

Access the Application (Local Development)

You need BOTH servers running simultaneously:

- **Frontend (React):** http://localhost:5173
- **Backend API:** http://127.0.0.1:8000/api
- **Django Admin:** http://127.0.0.1:8000/admin

Login Credentials (Local Development)

- Regular user: `demo` / `demo123`
- Admin user: `admin` / `admin123`

Method 3: Production Build (Single Server)

If you want to serve the React app from Django (like in Docker):

1. **Complete Backend Setup** (Steps 1-6 from Method 2)
2. **Build Frontend:**

```bash
cd frontend
npm install
npm run build
cd ../backend
```

3. **Collect Static Files:**

```bash
python manage.py collectstatic --noinput
```

4. **Start Django Server:**

```bash
python manage.py runserver 127.0.0.1:8000
```

5. **Access Application:**
   - Everything served from: http://127.0.0.1:8000

API Endpoints

Method	Endpoint	Description
POST	`/api/auth/register/`	User registration
POST	`/api/auth/login/`	User login
GET	`/api/artworks/`	List all artworks
GET/POST	`/api/cart/`	Cart operations
POST	`/api/orders/`	Create order
GET	`/api/orders/list/`	User's orders
GET	`/api/admin/orders/`	Admin: All orders

Environment Variables

For production deployment, set these environment variables:

```bash
SECRET_KEY=your-secret-key-here
DEBUG=False
ALLOWED_HOSTS=yourdomain.com
```

Development Notes

Running Both Servers

- **Backend Terminal:** Keep `python manage.py runserver` running
- **Frontend Terminal:** Keep `npm run dev` running
- Both terminals must stay open during development

Database

- Uses SQLite by default
- Database file: `backend/db.sqlite3`
- Reset database: Delete `db.sqlite3` and run migrations again

Static Files

- React builds to `frontend/dist/`
- Django serves from `/static/` URL
- Run `collectstatic` after frontend builds

Troubleshooting

Common Issues

1. **"npm is not recognized"**
   - Install Node.js from https://nodejs.org
2. **"python is not recognized"**
   - Add Python to your system PATH
3. **Virtual environment activation fails**
   - Use the correct activation command for your OS/shell
4. **React app shows white page**

- Check browser console for JavaScript errors
- Ensure both frontend and backend servers are running

5. **API calls fail with CORS errors**
   - Verify both servers are running on correct ports
   - Check `CORS_ALLOW_ALL_ORIGINS` setting

Windows Users

- Use Command Prompt or PowerShell (not Git Bash) for best compatibility
- Ensure virtual environment activation uses `.bat` extension

Contributing

1. Fork the repository
2. Create a feature branch
3. Make your changes
4. Test both Docker and local development setups
5. Submit a pull request

License

This project is for educational purposes.