PhotoFrame Project Documentation

**1. Introduction <a name="introduction"></a>**

PhotoFrame is a web application designed for managing and selling photo frames. It includes a Django backend for managing data and a React frontend for the user interface. This documentation provides an overview of the project, installation instructions, API documentation, and more.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2. Features <a name="features"></a>**

Admin Panel: Manage photo frames, including adding new frames, updating prices, and enabling top sales.

User Interface: View a list of available photo frames, explore details, and make purchases.

Responsive Design: The frontend is responsive, supporting various devices and screen sizes.

REST API: Backend API built with Django Rest Framework for data management.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3. Technologies Used <a name="technologies-used"></a>**

Frontend: React, Axios

Backend: Django, Django Rest Framework

Database: SQLite (for local development)

Styling: CSS, Bootstrap (React-Bootstrap for frontend)

Other Tools: Axios for HTTP requests, React Router for navigation

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**4. Installation <a name="installation"></a>**

Backend (Django)

* Clone the repository:

git clone <repo-url> cd photoframe-backend

* Install Python dependencies:

pip install -r requirements.txt

* Apply database migrations:

python manage.py migrate

* Create a superuser account:

python manage.py createsuperuser

* Run the development server:

python manage.py runserver

* Access the Django admin panel at http://localhost:8000/admin/ to add and manage frames.

**Frontend (React)**

* Open a new terminal and navigate to the frontend directory:

cd photoframe-frontend

* Install npm dependencies:

npm install

* Start the development server:

npm start

* Access the frontend application at http://localhost:3000/.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**5. Backend <a name="backend"></a>**

**Setup <a name="setup"></a>**

The backend of the PhotoFrame project is built using Django and Django Rest Framework.

* Models: The main model is Frame, which includes fields for title, image, original price, offer price, and top sales.
* Serializers: Convert model instances to JSON format.
* Views: Define the API views using Django Rest Framework viewsets.
* URLs: Map the views to URLs using Django's URL patterns.
* API Documentation <a name="api-documentation"></a>
* The API endpoints available are:
* GET /api/frames/: Fetch all frames.
* POST /api/frames/: Add a new frame.
* PATCH /api/frames/{id}/: Update a frame (toggle top sales).
* DELETE /api/frames/{id}/: Delete a frame.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**6. Frontend <a name="frontend"></a>**

**Setup <a name="setup-1"></a>**

The frontend of the PhotoFrame project is built using React.

Components: Various components such as FrameList, FrameForm, etc.

State Management: Utilizes React Hooks (useState, useEffect) for state management.

HTTP Requests: Uses Axios for making HTTP requests to the backend.

Routing: Uses React Router for navigation between different pages.