ShutterRent Frontend - README

# Project Overview

ShutterRent is a modern web-based camera rental system built with a React.js frontend and Node.js backend. The frontend allows users to register, log in, browse camera listings, book rentals, manage orders, and view/update their profiles. Admin users have access to an admin dashboard with listing, booking, and user management features.

# Live URL

Frontend Deployment: https://shutterrent-frontend.vercel.app

# Tech Stack

- React.js (Vite) for frontend  
- Tailwind CSS for styling  
- React Router for page navigation  
- Axios for HTTP requests  
- JWT for authentication  
- Cloudinary and Multer for image uploads

# Folder Structure

```  
shutterrent-frontend/  
├── public/ # Static assets  
├── src/  
│ ├── assets/ # Image and static assets  
│ ├── components/ # Reusable UI components (Navbar, AdminRoute)  
│ ├── pages/ # All main page components (Home, Login, Profile, Dashboard...)  
│ ├── App.jsx # Root component  
│ ├── App.css # App-wide styles  
│ └── main.jsx # App entry point  
├── .env # Environment variables (if needed)  
├── tailwind.config.js # Tailwind configuration  
├── vite.config.js # Vite configuration  
└── README.md # This file  
```

# Setup Instructions

To run the frontend locally:  
  
1. Clone the repository:  
 ```bash  
 git clone https://github.com/your-username/shutterrent-frontend.git  
 cd shutterrent-frontend  
 ```  
  
2. Install dependencies:  
 ```bash  
 npm install  
 ```  
  
3. Add a `.env` file (if using environment variables like VITE\_API\_URL)  
  
4. Run the project:  
 ```bash  
 npm run dev  
 ```

# Features Implemented

- Dynamic home page with category filtering  
- Listings page with filtered camera rentals  
- Booking system (create, view, cancel)  
- Order history display  
- User profile with editable fields & profile photo upload  
- Admin dashboard to manage listings, users, bookings, and listing creation  
- Status control for booking requests (pending/confirmed)  
- Responsive design with Tailwind  
- Light/Dark mode toggle  
- Conditional navigation bar for logged-in/admin users

# Integration Notes

The frontend consumes data from the backend API deployed via Render.  
Make sure the backend is running and accessible, especially during local testing.  
Base URL for backend can be managed via Axios defaults or an environment variable.