**Technical Guidance for ShutterRent Camera Rental System**

### 1. Introduction

ShutterRent is a full-stack web application designed to streamline the process of renting cameras and photography equipment online. This system allows users to browse listings, make bookings, manage their profiles, and for administrators to manage content and users from a dedicated dashboard. The project is built with modern web technologies and follows secure, modular, and scalable development practices.

### 2. Technology Stack

**Frontend**

* Framework: React.js (with Vite)
* Styling: Tailwind CSS
* Routing: React Router DOM
* State: useState and useEffect hooks
* Hosting: Vercel (https://shutterrent-frontend.vercel.app)

**Backend**

* Framework: Node.js with Express.js
* Database: MongoDB (hosted on MongoDB Atlas)
* Authentication: JWT (JSON Web Tokens)
* Middleware: Custom auth & admin checks
* File Uploads: Multer
* Hosting: Render.com

### 3. Project Structure

**Frontend Folder**

* /pages: Contains all main pages (Home, Login, Register, Profile, Listings, Booking, AdminDashboard, etc.)
* /components: Reusable components such as Navbar, Footer, Cards
* /api.js: Axios instance with base URL and token handling

**Backend Folder**

* /routes: Contains route files for auth, users, listings, bookings, admin
* /controllers: Business logic and request handling for each module
* /models: Mongoose models for User, Listing, Booking
* /middleware: Custom middleware for authentication, admin verification, and file uploads
* /uploads: Stores uploaded profile pictures and listing images

### 4. Deployment URLs

**Frontend (Public)**

* https://shutterrent-frontend.vercel.app

This URL is the main access point for all users including examiners. All backend functionality is seamlessly integrated into the frontend; hence no backend URL is necessary to expose.

**Backend (Private)**

* Hosted securely on Render.com
* Connected via environment variable VITE\_API\_BASE\_URL in the frontend
* Not intended for direct public interaction

### 5. Main Features Implemented

#### User Side

* Register/Login using JWT Auth
* Browse camera listings with category filtering
* Book available listings (with conflict prevention)
* View and cancel bookings in Order History
* Edit profile with image upload (Multer)
* Change password securely with verification

#### Admin Side

* Admin-only access to dashboard
* View, edit, delete any listings
* Create listings from dashboard
* View and manage all users
* View and update all bookings with status changes (Pending/Confirmed)

### 6. Access Credentials

You may use the following demo credentials to test the platform:

**Admin**

* Email: aamirptk95@gmail.com
* Password: shutterrent123

**User**

* Email: newuser@gmail.com
* Password: shutterrent321

### 7. Environmental Variables (for deployment only)

**Frontend (.env):**

VITE\_API\_BASE\_URL=https://shutterrent-backend.onrender.com/api

**Backend (.env):**

MONGO\_URI=mongodb+srv://ptk:HESoyam243@shutterrent-cluster.lproqul.mongodb.net/?retryWrites=true&w=majority&appName=shutterrent-cluster

JWT\_SECRET=shutterrentsecret123

These are already configured in the live deployment and do not need modification for examiners.

### 8. Limitations

* Admin account is hardcoded (no role management from UI).
* Image hosting is based on Render static /uploads, which may take time to load.
* No payment integration (only payment method label shown).

### 9. Instructions for Examiner

You do **not** need to clone or run the code locally. Simply visit:

https://shutterrent-frontend.vercel.app

You can register as a user or log in with the admin credentials provided. All features are fully functional and can be tested live.

### 10. Conclusion

This document provides full technical guidance for understanding and reviewing the ShutterRent camera rental web application. The platform was built with best practices in mind, using a modular approach and responsive design to ensure a seamless user and admin experience. Thank you for reviewing the project.