# ShutterRent Backend - README

# ShutterRent Backend  
  
This is the backend for \*\*ShutterRent\*\*, a full-stack camera rental web application built with Node.js, Express, and MongoDB. It provides all API endpoints to support the user authentication, listing management, booking system, and admin dashboard.  
  
---  
  
## Technologies Used  
  
- Node.js  
- Express.js  
- MongoDB + Mongoose  
- Multer (Image Uploads)  
- JSON Web Tokens (JWT)  
- bcrypt.js (Password Hashing)  
- Dotenv (Environment Variables)  
- CORS  
  
---  
  
## Folder Structure  
  
```  
shutterrent-backend/  
│  
├── controllers/ # Route logic for auth, users, bookings, listings, admin  
├── middleware/ # Custom middleware (auth, admin check, multer config)  
├── models/ # Mongoose schemas: User, Listing, Booking  
├── routes/ # Express route definitions for all features  
├── uploads/ # Uploaded user profile & listing images  
├── .env # Environment variables (not committed)  
├── server.js # Entry point to the backend server  
```  
  
---  
  
## .env Configuration  
  
Create a `.env` file in the root directory with the following:  
  
```  
MONGO\_URI=mongodb+srv://ptk:HESoyam243@shutterrent-cluster.lproqul.mongodb.net/?retryWrites=true&w=majority&appName=shutterrent-cluster  
JWT\_SECRET=shutterrentsecret123  
```  
  
---  
  
## How to Run Locally  
  
1. Navigate to the backend directory:  
  
```bash  
cd shutterrent-backend  
```  
  
2. Install dependencies:  
  
```bash  
npm install  
```  
  
3. Start the server:  
  
```bash  
node server.js  
```  
  
Server will run on `http://localhost:5000` (or your specified PORT).  
  
---  
  
## Available API Routes  
  
### Auth Routes (`/api/auth`)  
- `POST /register`  
- `POST /login`  
  
### User Routes (`/api/user`)  
- `GET /profile` (Protected)  
- `PUT /update` (Update name, phone, address, profile picture)  
  
### Listing Routes (`/api/listings`)  
- `GET /` (All listings)  
- `POST /` (Create new listing) [Admin]  
- `PUT /:id` (Update listing) [Admin]  
- `DELETE /:id` (Delete listing) [Admin]  
  
### Booking Routes (`/api/bookings`)  
- `POST /` (Create booking)  
- `GET /` (Get user bookings)  
- `DELETE /:id` (Cancel booking)  
  
### Admin Routes (`/api/admin`)  
- `GET /users` (View all users)  
- `DELETE /user/:id`  
- `GET /listings`  
- `GET /bookings`  
- `PUT /bookings/:id/status`  
  
---  
  
## Security Features  
  
- JWT Auth middleware to protect private routes  
- Admin middleware to restrict admin-only actions  
- bcrypt password hashing  
- Validation and error handling  
  
---  
  
## Hosting  
  
This backend is deployed on \*\*Render\*\* and works with the frontend hosted on \*\*Vercel\*\*.  
  
---  
  
## Notes  
  
- Make sure to allow `https://shutterrent-frontend.vercel.app` and `http://localhost:5173` in CORS  
- Images are served statically via `/uploads/<filename>`  
- Profile image and listing uploads use Multer middleware  
  
---  
  
## Author  
  
Built by Phyo Thet Khaing for the University of Sunderland Final-Year Project - 2025.