



# PROJECT NAME

BG Remover Portal – User Login, Image Upload & Delivery System

---



## PROJECT OBJECTIVE

Create a portal where:

1. When someone clicks the link, the **first screen must be Sign Up.**
  2. No password is required — only **Full Name + Email (Gmail ID).**
  3. After sign up, the user sees **one upload button** with a sample image on the background.
  4. The user uploads their photo.
  5. After upload, a popup should say:  
**“We have received your photo. You will get your edited image on this portal within 3 days.”**
  6. In the backend, admin receives the image, edits it, re-uploads the final version for that user.
  7. The user logs in anytime and can **download only their own edited image.**
  8. No user should be able to see any other user's photos.
  9. Admin should be able to see everything.
- 



## FULL PROJECT ARCHITECTURE



### 1. Authentication (No Password Login)

Fields:

- Full Name
- Email ID

Process:

- User enters Full Name + Email
  - A 6-digit OTP is sent to email (for verification)
  - User enters OTP → Login successful
  - Creates user profile in database
- 



## 2. User Dashboard (After Login)

UI Components:

- Background sample image (template you already have)
- **Upload Image Button**
- Upload notes written inside the card (example):

“Please upload your photo. You will receive your edited image within 3 days.”

Actions:

- User uploads **1 image (JPG/PNG)**
  - After upload → Save image to server / cloud
  - Show popup:  
**“We have received your photo. You will get the edited image in 3 days.”**
- 



## 3. Backend Processing

Admin Panel (for you/team):

Features for Admin:

- See all registered users
- See each user's uploaded image
- Download it
- Edit it manually
- Upload the final edited image back
- Status options:
  - Pending
  - Completed

Backend sets:

- Timestamp of upload
- Edited file URL
- Auto-send email to user after upload:

“Your edited photo is now available. Login to your portal to download.”

---



## 4. User Image Download

User logs in → sees:

- “Your Uploaded Image”
- “Your Edited Image” (if uploaded by admin)
- Download button

User should only see **their own image**, no others.

---



# TECH STACK (Recommended)

## Frontend:

- React.js or Next.js
- Tailwind CSS
- Axios for API calls

## Backend:

- Node.js + Express.js
- OR Firebase Functions (If you want very quick setup)

## Database:

- Firebase Firestore  
or
- MongoDB

## Storage:

- Firebase Storage  
or
- AWS S3 Bucket



# DATABASE STRUCTURE

## Users Collection

```
{  
  userId: "unique-id",  
  fullName: "Somesh Chaudhari",
```

```
        email: "abc@gmail.com",
        createdAt: timestamp,
    }
```

## Images Collection

```
{
    userId: "unique-id",
    originalImageURL: "...",
    editedImageURL: "...",
    status: "Pending/Completed",
    uploadedAt: timestamp,
    editedAt: timestamp
}
```

---



## API ENDPOINTS

1. POST `/auth/signup`
  2. POST `/auth/verify-otp`
  3. POST `/image/upload`
  4. GET `/image/user/:id`
  5. POST `/image/admin/upload-edited`
  6. GET `/admin/users`
- 



## WORKFLOW SUMMARY (Simple)

### User Side

1. Open link → Sign Up

2. Enter name + email → OTP login
3. Dashboard → Upload photo
4. Show success message
5. After 3 days → Login again → Download edited photo

## **Admin Side**

1. Login to admin panel
2. See all images
3. Download → Edit → Upload edited file
4. Change status to Completed
5. User receives email and can download