

Lag gaya! Backend me jo `photos` ka `path` store ho raha hai (e.g. `uploads/properties/xyz.jpg`), usko **public URL** banana hoga aur static serve karna hoga. Phir frontend me simple `` se render ho jayega.

1) Express me static serving enable karo

```
// app.js / server.js
const express = require('express');
const path = require('path');
const app = express();
// Serve the whole /uploads folder publicly at /uploads/*
app.use('/uploads', express.static(path.join(process.cwd(), 'uploads')));
```

Isse jo bhi file `uploads/...` me hai, wo `http://<your-domain>/uploads/...` se accessible ho jayegi.

2) API responses me file-path → public URL convert karo

Abhi tum absolute filesystem path bhej rahe ho

(e.g. `C:\project\uploads\properties\abc.jpg`). Frontend ko URL chahiye. Ek helper banao:

```
// utils/fileUrl.js
const path = require('path');
const BASE_URL = process.env.BASE_URL || 'http://localhost:3000';
// 👉 apne server ka origin (port same jahan express chal raha ho)
function toPublicUrl(filePath) {
  if (!filePath) return null;
  // Make it relative to /uploads
  const rel = path.relative(path.join(process.cwd(), 'uploads'), filePath);
  // Normalize slashes for URLs
  const urlPath = rel.split(path.sep).join('/');
  return `${BASE_URL}/uploads/${encodeURIComponent(urlPath)}`;
}
module.exports = { toPublicUrl };
```

Create/Update ke response me URLs bhejo

Tumhare controller me:

```
const { toPublicUrl } = require('../utils/fileUrl');
// ...inside createProperty after creating:
res.status(201).json({
  success: true,
  message: "Property created successfully",
  data: {
    id: propertyId,
    ownershipDocUrl: toPublicUrl(ownershipDocPath),
    photoUrls: photoPaths.map(toPublicUrl), // ← array of public URLs
    uploadedFiles: { ownershipDoc: ownershipDocPath ? 1 : 0, photos: photoPaths.length, },
  },
});
```

Read endpoints me URLs map karo

```
const mapPropertyFiles = (p) => ({
  ...p,
  ownership_doc_url: toPublicUrl(p.ownership_doc_path),
  photos: Array.isArray(p.photos) ? p.photos.map(toPublicUrl) : [],
});
const getAllProperties = async (req, res)
```

```
=> { try { const properties = await Property.getAll(); res.json({ success: true, data: properties.map(mapPropertyFiles) }); } catch (error) { /* ... */ } }; const getProperty = async (req, res) => { try { const property = await Property.getById(req.params.id); if (!property) return res.status(404).json({ success: false, message: "Property not found" }); res.json({ success: true, data: mapPropertyFiles(property) }); } catch (error) { /* ... */ } };
```

Note: Agar DB me `photos` JSON me **absolute paths** saved hain, unko bhejte waqt hi URL me convert kar do (upar jaise).

3) Frontend (React) me render

API se aayega:

```
{ "id": 123, "photos": [ "http://localhost:3000/uploads/properties/abc.jpg", "http://localhost:3000/uploads/properties/def.png" ], "ownership_doc_url": "http://localhost:3000/uploads/properties/docs/xyz.pdf" }
```

Render:

```
// Example function PropertyGallery({ property }) { return ( <div className="grid grid-cols-2 md:grid-cols-3 gap-3"> {property.photos?.map((url, idx) => ( <img key={idx} src={url} alt={`Property photo ${idx + 1}`} className="w-full h-40 object-cover rounded-lg border" loading="lazy" /> ))} </div> ); }
```

Ownership document (PDF) download/view:

```
<a href={property.ownership_doc_url} target="_blank" rel="noopener noreferrer" className="text-blue-600 underline" > View Ownership Document </a>
```

4) Photo delete ke liye same URLs bhejo

Tumhare `deletePhotosFromProperty` me `photoUrls` expect hote hain—frontend se wahi URLs array pass kar do jo display ke liye use ho rahe hain. Backend URL → path mapping DB me handle kare (apne `Property.deleteSpecificPhotos` me URL se relative path nikaal lo).

Example mapping utility (agar delete me zarurat pade):

```
function urlToAbsolutePath(publicUrl) { try { const u = new URL(publicUrl); // Expecting /uploads/<relative> const idx = u.pathname.indexOf('/uploads/'); if (idx === -1) return null; const rel = u.pathname.slice(idx +
```

```
'/uploads/'.length); return path.join(process.cwd(), 'uploads',  
decodeURI(rel)); } catch { return null; } }
```

5) Common gotchas

- **CORS:** API and frontend different origins hain to CORS enable rakho.
- **Windows slashes:** \ ko URL banate waqt / me convert karna zaroori.
- **Security:** Sirf uploads ko static serve karo; project root ko kabhi mat expose karo.
- **BASE_URL:** Env me set kar do (https://api.yoursite.com) taki URLs sahi banein.

Itna karte hi images frontend par seedha load ho jayengi. Agar chaho, main tumhare current responses me URL mapping code integrate karke ready-to-paste diffs bhi de sakta/ sakti hoon.