



Brochure URL — Technical Solutions Report

1. Objective

Currently, brochure links shared with clients expose raw database IDs or Supabase storage URLs, which look unprofessional and are difficult to remember. The goal is to create **clean, branded, SEO-friendly URLs** — similar to how our project website URLs already work.

Current (ugly):

```
https://xyzproject.supabase.co/storage/v1/object/public/brochures/403a576e-8b2f-4a1d-91da-e8c34a10b7f2.pdf
```

Target (professional):

```
https://onward.com/p/quinite-tech/sunrise-residency/brochure
```

2. Proposed Solutions

We have evaluated 3 approaches, ranging from easiest to most integrated:

Solution 1: ID + SEO Slug (Quick & Easy)

Format	https://onward.com/api/brochure/project/403a576e/sunrise-residency.pdf
How it works	Uses the project ID for database lookup + appends a readable name for SEO
Example	onward.com/api/brochure/project/403a576e/sunrise-residency.pdf

✔ Pros:

- Extremely robust — if the project name changes, the link still works
- The .pdf extension makes it clear to clients what they're downloading

✖ Cons:

- URL still contains the UUID, making it somewhat long

Solution 2: Sub-Page Approach ★ Recommended

Format	https://onward.com/p/quinite-tech/sunrise-residency/brochure
How it works	Attaches /brochure to the existing project website URL structure
Example	onward.com/p/quinite-tech/sunrise-residency/brochure

 Pros:

- Matches exactly how our project websites currently work
- Highly branded, extremely clean — completely hides all UUIDs
- Ties directly into the existing `public_pages` system (no new infrastructure needed)

 Cons:

- If a project is renamed, older links may need redirect handling
- Slightly longer due to organization slug (`quinite-tech`)

Solution 3: Direct Slug (Cleanest)

Format	<code>https://onward.com/brochures/sunrise-residency</code>
How it works	A dedicated top-level route that looks up the project by slug alone
Example	<code>onward.com/brochures/sunrise-residency</code>

 Pros:

- Shortest, cleanest, most readable link possible
- Looks fantastic in SMS / WhatsApp

 Cons:

- Slugs must be **100% globally unique** across the entire database
- If two different builders both have "Sunrise Residency", the system cannot distinguish them

3. URL Comparison — What Clients See on WhatsApp

Solution	URL Preview
Current	<code>xyzproject.supabase.co/storage/v1/object/public/brochures/403a576e-...</code>
1	<code>onward.com/api/brochure/project/403a576e/sunrise-residency.pdf</code>
2 	<code>onward.com/p/quinite-tech/sunrise-residency/brochure</code>
3	<code>onward.com/brochures/sunrise-residency</code>

4. Search Capability — Built-in with Supabase

[!IMPORTANT] No external search service is needed. Supabase includes full-text search powered by Postgres out of the box — at no additional cost.

What we get by default:

Feature	Description
Keyword Search	Find projects/brochures by exact words
Fuzzy Matching	"residency" matches "residential", "residence", etc.
Prefix Search	Typing "sun" finds "Sunrise Residency", "Sunflower Heights", etc.
Multi-Column Search	Search across project name + description + location simultaneously
Ranked Results	Most relevant results appear first
Google-style Search	Supports quotes, OR, and negation (e.g., "sunrise" OR "green valley")
Fast Performance	GIN indexes ensure instant results even with large data

How it works from the app:

```
// Search projects by name
const results = await supabase
  .from('projects')
  .select()
  .textSearch('name', 'sunrise residency', { type: 'websearch' })
```

No Algolia, no Meilisearch, no extra cost — it's all included in our existing Supabase plan.

5. Recommendation

Aspect	Recommendation
URL Approach	Solution 2 (Sub-Page) — matches our existing project website URL pattern and leverages the <code>public_pages</code> table already in our database
Search	Use Supabase's built-in full-text search — no external tools needed

Why Solution 2?

- 1. It works **exactly** the way our project website URLs already work
- 2. Our database already has a robust `public_pages` table that maps these slugs safely within an organization
- 3. It is the most **native and scalable** solution for our architecture
- 4. Completely **hides all database IDs** — looks professional and trustworthy to clients

[!NOTE] Both the URL solution and the search capability can be implemented entirely within our existing Supabase infrastructure with **zero additional cost**.