

BP Cleaning App - Riassunto Progetto

Repository GitHub

<https://github.com/Catholically/bp-cleaning-app>

Deploy Vercel

<https://bp-cleaning-app.vercel.app>

Database Supabase

- **URL:** <https://nesvkpyngurlkmduywy.supabase.co>
 - **Progetto:** BP Cleaning
-

Tabelle Database

1. profiles

Utenti dell'app

- `[id]` (UUID, collegato a auth.users)
- `[email]`
- `[full_name]`
- `[role]` ('user' o 'superuser')

2. suppliers

Fornitori prodotti

- `[id]`, `[name]`, `[contact_email]`, `[phone]`, `[notes]`

3. products

Prodotti magazzino

- `[id]`, `[barcode]` (univoco)
- `[name]`, `[description]`
- `[category]` (detergenti, carta, attrezature, DPI, varie)
- `[unit]` (pz, lt, kg, rotolo, pacco)
- `[current_stock]`, `[min_stock]`

- `cost_per_unit`
- `supplier_id`

4. worksites

Cantieri/clienti

- `id`, `code` (es. CANT-001)
- `name`, `address`
- `contact_name`, `contact_phone`
- `monthly_budget`
- `is_active`

5. movements

Movimenti carico/scarico

- `id`, `type` ('carico' o 'scarico')
- `product_id`, `worksite_id` (solo per scarico)
- `quantity`, `unit_cost`, `total_cost` (calcolato auto)
- `notes`, `created_by`

Environment Variables (Vercel)

env

```
NEXT_PUBLIC_SUPABASE_URL=https://nesvkpyngurlkmduywy.supabase.co
NEXT_PUBLIC_SUPABASE_ANON_KEY=[inserisci la chiave anon]
```

Pagine App

Route	Descrizione
<code>/login</code>	Autenticazione email/password
<code>/</code>	Dashboard home
<code>/prodotti</code>	Lista prodotti, filtro scorte basse

Route	Descrizione
/movimenti	Storico tutti i movimenti
/movimenti/carico	Form carico merce
/movimenti/scarico	Form scarico per cantiere
/cantieri	Lista cantieri con costi mensili
/impostazioni	Settings e logout

SQL Trigger per Auto-Creazione Profili

Eseguire in **Supabase → SQL Editor:**

```
sql
```

```
-- Funzione per creare automaticamente il profilo utente
CREATE OR REPLACE FUNCTION public.handle_new_user()
RETURNS trigger AS $$

BEGIN
    INSERT INTO public.profiles (id, email, full_name, role)
    VALUES (
        NEW.id,
        NEW.email,
        COALESCE(NEW.raw_user_meta_data->>'full_name', NEW.email),
        'user'
    );
    RETURN NEW;
END;

$$ LANGUAGE plpgsql SECURITY DEFINER;
```

```
-- Trigger che si attiva alla registrazione
DROP TRIGGER IF EXISTS on_auth_user_created ON auth.users;
CREATE TRIGGER on_auth_user_created
AFTER INSERT ON auth.users
FOR EACH ROW EXECUTE FUNCTION public.handle_new_user();
```

```
-- RLS policies
ALTER TABLE profiles ENABLE ROW LEVEL SECURITY;
```

```
DROP POLICY IF EXISTS "Users can view own profile" ON profiles;
CREATE POLICY "Users can view own profile" ON profiles
FOR SELECT USING (auth.uid() = id);
```

```
DROP POLICY IF EXISTS "Enable insert for authentication" ON profiles;
CREATE POLICY "Enable insert for authentication" ON profiles
FOR INSERT WITH CHECK (true);
```

Branding

- **Logo:** Goccia blu con "BP" bianco
- **Colori:** Blu primario ( #3B82F6), sfondo grigio chiaro

Prossimi Step Suggeriti

1. Eseguire il trigger SQL per auto-creazione profili
 2. Aggiungere scanner barcode per mobile
 3. Gestione fornitori completa
 4. Report PDF mensili per cantiere
 5. Notifiche scorte basse
-

Ultimo aggiornamento: Gennaio 2026