

רשימת תיקון

BabyList Israel

מסמך אפיון מוצר מלא - גרסה 4.0

🚀-ספרינט 8 ימים ל-MVP

Tech Stack

React + Vite + TypeScript + Supabase + GitHub Pages

Payment Provider

Green Invoice / חשבונית יומת (Morning) - Dynamic Payment Links

언어: JavaScript

모델: Concierge Fulfillment (MVP)

תוכן עניינים

1. סיכום מנהלי ומודל עסקי.
2. חשבונית ירוקה (Green Invoice) אינטגרטיבית.
3. מערכת התראות אימיל.
4. דרישות פונקציונליות מפורטות.
5. Supabase (סכמת בסיס נתונים).
6. UI/UX - כל המרכיבים בעברית.
7. Edge Functions - קוד מלא.
8. לוח זמנים - 8 ימים (שעה אחר שעה).
9. Deployment ל-GitHub Pages.
10. QA בדיקות.
11. MVP-לאחר ה-

1. סיכום מנהלים ומודל עסקי.

1.1 מה אנחנו בונים?

KSP, פלטפורמה ישראלית לניהול רשותות לידי אוניברסיטאות. הורים מוסיפים מוצרים מכל חנות ישראליות (שלב, מוצצים, סופר-פארם, ועוד, (חברים ומשפחה רואים את הרשותה, משלמים דרכנו, ואנו קונים ושולחים את המנתנה.

1.2 איך זה עובד - מודל Concierge

"במקרה לבנות אינטגרציות מורכבות עם חניות, אנחנו מפעילים מודל "קונסיליר".

1. אורח בוחר מתנה מהרשימה
2. אורח משלם דרך חשבונית יroke (Green Invoice)
3. אנחנו מקבלים התראה + כסף
4. אנחנו קונים את המוצר מהחנות המקורית (דנית)
5. אנחנו שולחים ללקוחות שהאורח חזן
6. כולן מקבלים עדכון באימייל

1.3 ללא פעולה - MVP אסטרטגיית

אנו לא גובים פעולה - המחיר שהאורח משלם = המחיר בחנות MVP-ב-

"בל' חיכון של "למה לשלם יותר מה זה? כדי לבדוק Product-Market Fit למה? כדי לבדוק

מה אנחנו מרווחים בזמן זה:

- נתוני לקוחות (כתובות, טלפונים, העדפות)
- הבנת השוק (מה קונים, מאיפה, כמה)
- בניית מותג ואמון
- תוכן לשוק ("500 משפחות כבר משתמשות")
- אמייתי מליקחות Feedback

1.4 למה Green Invoice?

kritirion	Green Invoice	Lemon Squeezy
عملיה	2.5%	5% + \$0.50
תמייה בעברית	מלואה	אין
חשבונות	אוטומטי (חויה בישראל!)	לא רלוונטי
זמן הקמה	כבר יש לכם	יום
לסקום דינמי API	כן ✓	כן ✓
Webhooks	כן ✓	כן ✓
אמון לקוחות	גבוה (מותג ישראלי)	ビルוני

1.5 Stack

שכבה	טכנולוגיה	סיבה
Frontend	React 18 + Vite + TypeScript	Type-safe, מהיר
Styling	Tailwind CSS + RTL	מהיר לפיתוח, תמייה בעברית
Backend	Supabase (PostgreSQL + Auth + Edge Functions)	הכל בمكان אחד
Payments	Green Invoice API	ישראלי, זול, כבר מוגדר
Emails	Resend	נק, API, פשוט, זול
Hosting	GitHub Pages	חינם, פשוט
Domain	MVP אחריו) בהמשך	reshimat-tinok.co.il

2. חשבונית ירוקה (Green Invoice) אינטגרטיבית

2.1 איך זה עובד - תמונה גודלה

Green Invoice: מאפשר ליצור קישור תשלום דינמי לכל סכום. כל פעם שאורח רוצה לקבלת מתנה, אנחנו

1. שלטנו DB-DB (status: pending_payment)
2. עם הסכום הגרפי Green Invoice של קוראים לו
3. ייחודי לתשלום URL מקבלים
4. מעבירים את האורח לדף התשלום
5. עם אישור Webhook אחריו תשלום - מקבלים
6. ושלוחים אימילים DB-עדכנים את ה

2.2 בחשבונית ירוקה API הגדרת

2.2.1 קבלת מפתחות API

בדשبورד של חשבונית ירוקה:

צורך מפתח חדש → API → הגדרות

תקבלו:

- API_KEY (מזהה)
- API_SECRET (ടו)

2.2.2 הגדרת Webhook

בדשבורד:

URL הוסף → URLs → הגדירות

URL: <https://<your-supabase-url>/functions/v1/green-invoice-webhook>

סמן את האירועים: payment.success, payment.failure

2.3 API Reference - תשלום

Endpoint

POST <https://api.greeninvoice.co.il/api/v1/paymentForm/create>

Headers

Authorization: Bearer <JWT_TOKEN>

Content-Type: application/json

Body - דוגמה מלאה

```
{
  "type": 320, // קבילה = 320
  "lang": "he", // עברית
  "currency": "ILS", // שקלים
  "pluginId": "order-uuid", // מזהה הזמנה שלנו

  "client": { // פרטי הלקוח
    "name": "ישראל ישראלי", // שם
    "emails": ["israel@email.com"], // אימייל
    "phone": "0501234567", // טלפון
    "add": false // לא להוסיף לרשימת לקוחות
  },
  "income": [{} // הסכום הדינמי
    "description": "תגלת תינוק Bugaboo", // מילוי
    "quantity": 1, // כמות המוצר
    "price": 1500, // מחיר המוצר
    "currency": "ILS", // מטבע
  ]
}
```

```

    "vatType": 1           // כולל מ"מ
  }],

  "remarks": "מתנה מרשות הלידה של שרה ודני",
  "successUrl": "https://site.com/success?order=xxx",
  "failureUrl": "https://site.com/failed?order=xxx",
  "notifyUrl": "https://supabase.co/functions/v1/webhook",

  "custom": "{\"order_id\":\"xxx\"}" // מילוי מידי נושא לwebhook
}

```

Response

```
{
  "id": "abc123",
  "url": "https://app.greeninvoice.co.il/pay/abc123" // 🔍 לאורח
}
```

2.4 Webhook Payload - מה מקבלים אחרי תשלום

```

{
  "id": "webhook-id",
  "type": "payment.success",
  "data": {
    "id": "payment-id",
    "status": 1,           // 1 = הצלחה
    "amount": 1500,
    "documentId": "doc-id",
    "documentUrl": "https://...", // לינק לקבלת המזהה שלנו!
    "pluginId": "order-uuid",   // המזהה שלנו!
    "custom": "..."}         // המידע הנושא ששלחנו
  }
}

```

3. מערכות התראות אימייל.

3.1 סוגים אימיילים

אימייל	נושא	טריגר	תוכן עיקרי
אישור רכישה	הكونה (אורח)	Webhook success	פרטי הזמנת, לינק לקבללה
התראת מתנה	בעל/ת הרשימה	Webhook success	משהו קנה לך + פרטיים
Admin התראת	אתה	Webhook success	יש הזמנה לטפל - כל הפרטים
עדכן משלוח	הكونה	Admin action	המתנה נשלחה + מספר מעקב
Welcome	הרשמה חדש	After signup	ברוכים הבאים + הסבר

3.2 Resend

נקוי API, לשילוח אימיילים פשוט, זול Resend-נשותמם ב-

1. הרשמה ב resend.com
2. API Key צור
3. Supabase secrets הוסף ל

3.3 תבניות אימייל בעברית

אימייל 1: אישור רכישה (לקוננה)

נושא אישור רכישה - [שם המוצר]

, היי [שם הקוננה],

תודה שרכשת מתנה מרשות הלידת שלך [שם ההורים]

: פרטי הזמנת

- מוצר [שם המוצר]
- סכום תשלום [₪ : מחיר]
- : מס' הזמנה [order_id]

: כתובת למשולח

[שם], [כתובת], [עיר]

. המתנה תישלח תוך 3-5 ימי עסקים.

! נעדכן אותך כשהיא בדרך

[לינק לקבללה מחשבונית י Rogha]

, בברכה,

צוות רשות תינוק 

אימייל 2: התראת מתנה (לבעל הרשימה)

נושא  משהו קנה לך מתנה!

, היי [שם]

! יש בשורות טובות 

: [שם הקוננה] / משהו [רכשה לך]

 [שם המוצר]

המתנה בדרך אלין!

לצפייה ברשימה שלך]: לינק [

בברכה,

צוות רשות תינוק 📝

אליך 3: התראת אימייל

נושא: 🎉 הזמנה חדשה - [מספר הזמן]

===== הזמנה חדשה לטיפול =====

📦: פרטי המוצר

- שם]: שם המוצר
- מחיר]: מחיר
- חנות]: שם החנות
- לינק]: URL לקובייה

🚚: פרטי משלוח

- שם]: שם מלא
- כתובת]: כתובת מלאה
- טלפון]: טלפון

✉️: פרטי תשלה

- שולם]: סכום
- קבלה]: לינק לחשבונית יരוקה

➡️ לטיפול בדשبورד]: לינק [

אימייל 4: עדכון משלוח

נושא: 🚚 המתנה שלך בדרך!

, היי [שם]

המתנה ששלחת לך שם ההורים [יצא לדרך! 🎁

📦: שם המוצר

🚚: מספר מעקב [tracking]

תודה שהשתמשה ברשות תינוק!

בברכה,

צוות רשות תינוק 📝

דרישות פונקציונליות מפורטות 4.

מערכת משתמשים 4.1

הרשמה

שדה	סוג	חובה	Validation
אימייל	email	✓	פורמט תקין
סיסמה	password	✓	מינימום 8 תווים
אישור סיסמה	password	✓	תואם לסיסמה

(אפשרויות נוספת: Google OAuth, Magic Link)

השלמת פרופיל (אחרי הרשמה)

שדה	סוג	חובה	הערות
שם מלא	text	✓	להציג ברשימה
טלפון	tel	✓	לתיאום משלוחים
תאריך לידה משוער	date	✓	לספירה לאחר
מין התינוק	select	X	בן/בת/הפטעה/תאומים
שם בן/בת זוג	text	X	להציג ברשימה

ניהול רשימה 4.2

יצירת רשימה

שדה	סוג	חובה	הערות
כותרת	text	✓	"הרשימה של שרה ודני"
תיאור	textarea	X	הודעה לאורחים
Slug	auto	✓	ನೋಂದುಗೊಳಿಸಿ (sarah-dan-2025)
ציבורי	boolean	✓	ברירת מחדל:כן

הוספת פריט

: שני מצבים

(אוטומטי) URL מצב 1: הבדיקה

- משתמש מודיע לינק מהנות
- מנסה לחלץ: שם, תמונה, מחיר, חנות Function Preview אם הצלח - מציג
- לאישור Preview אם הצלח - מציג
- משתמש יכול לעורר לפני שמיירה

(Fallback) מצב 2: הזנה ידנית

- אם חילוץ נכשל - נפתח טופס ידני
- משתמש מזין את כל הפרטים

שדות פריט

שדה	סוג	חובה	הערות
שם מוצר	text	✓	בערבית
מחיר	number	✓	ב שקלים
لينك מקורי	url	✓	לחנות

שדה	סוג	חובה	הערות
תמונה	url/file	✓	או העלאה URL
חנות	select	✓	שיילב/מוצצים/אחר
קטגוריה	select	✓	צדוד/לבשה/האוכל/בריאות/חדר
כמות	number	X	ברירת מחדל : 1
עדיפות	select	X	גבואה/בינונית/נמוכה
הערות	textarea	X	מידה, צבע, העדפה

4.3 תצוגת אורה (Public)

מה האורח רואה

- cotreta הרשימה + שמות ההורמים
- ספירה לאחר לידה ("עוד ימי(ים")!
- הודעה אישית מההורם
- רשת מוצרים (Grid)
- פילטרים: קטgorיה, חנות, מחיר, זמין/נכנה
- כפתור "קנה מותנה" לכל פריט זמין

זרימת רכישה

- אורח לוחץ "קנה מותנה"
- נפתח מודול - שלב 1: פרטי הקונה
- שלב 2: כתובת למשלו
- שלב 3: סיכום + אישור
- לחיצה על "תשלומים" → "יצירת הזמנה
- Redirect Green Invoice
- עדכן סטטים + אימיילים → תשלום
- דף הצלחה

4.4 דashboard (בשבייר) Admin

- רשימת כל הזמנות לפי סטטוס
- פרטי כל הזמנה: מוצר, לינק, כתובת, טלפון
- (purchased) כפתור "סמן כנשלח" + "הנתן מסך מעקב"
- (shipped) כפתור "סמן כנשלח" + "הנתן מסך מעקב"
- (delivered) כפתור "סמן כהתקבל"
- הערות פנימיות לכל הזמנה

5. סכמת בסיס נתונים (Supabase)

5.1 טבלת profiles

```
CREATE TABLE profiles (
    id UUID PRIMARY KEY REFERENCES auth.users(id) ON DELETE CASCADE,
    email TEXT NOT NULL,
    full_name TEXT NOT NULL,
    phone TEXT,
    partner_name TEXT,
    due_date DATE,
    baby_gender TEXT CHECK (baby_gender IN ('boy', 'girl', 'surprise', 'twins')),
    avatar_url TEXT,
    created_at TIMESTAMPTZ DEFAULT NOW(),
    updated_at TIMESTAMPTZ DEFAULT NOW()
);
```

5.2 טבלת registries

```
CREATE TABLE registries (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL REFERENCES profiles(id) ON DELETE CASCADE,
    title TEXT NOT NULL,
    description TEXT,
    slug TEXT UNIQUE NOT NULL,
    is_public BOOLEAN DEFAULT true,
    created_at TIMESTAMPTZ DEFAULT NOW(),
    updated_at TIMESTAMPTZ DEFAULT NOW()
);
```

5.3 טבלת items

```
CREATE TABLE items (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    registry_id UUID NOT NULL REFERENCES registries(id) ON DELETE CASCADE,
    name TEXT NOT NULL,
    price NUMERIC(10,2) NOT NULL,
    original_url TEXT NOT NULL,
    image_url TEXT,
    store_name TEXT NOT NULL,
    category TEXT NOT NULL CHECK (category IN (
        'gear', 'clothing', 'feeding', 'health', 'nursery', 'other'
    )),
    quantity_needed INTEGER DEFAULT 1,
    quantity_purchased INTEGER DEFAULT 0,
    priority TEXT DEFAULT 'medium' CHECK (priority IN ('high', 'medium', 'low')),
    notes TEXT,
    status TEXT DEFAULT 'available' CHECK (status IN ('available', 'purchased')),
    purchased_by TEXT,
    created_at TIMESTAMPTZ DEFAULT NOW()
);
```

5.4 טבלת orders

```
CREATE TABLE orders (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    item_id UUID NOT NULL REFERENCES items(id),
    registry_id UUID NOT NULL REFERENCES registries(id),
```

```
-- Buyer info
buyer_name TEXT NOT NULL,
buyer_email TEXT NOT NULL,
buyer_phone TEXT NOT NULL,
buyer_message TEXT,
is_anonymous BOOLEAN DEFAULT false,

-- Shipping info
shipping_name TEXT NOT NULL,
shipping_address TEXT NOT NULL,
shipping_city TEXT NOT NULL,
shipping_phone TEXT NOT NULL,

-- Payment info
amount_paid NUMERIC(10,2) NOT NULL,
payment_method TEXT DEFAULT 'green_invoice',
green_invoice_form_id TEXT,
green_invoice_doc_id TEXT,
green_invoice_doc_url TEXT,

-- Status
status TEXT DEFAULT 'pending_payment' CHECK (status IN (
    'pending_payment', -- ממתין לתשלומים
    'paid', -- שולם
    'purchased', -- נרכשה מהחנות
    'shipped', -- נשלח
    'delivered', -- הגיע
    'cancelled' -- בוטל
)),
-- Fulfillment tracking
store_order_id TEXT, -- מספר הזמנה מהחנות
tracking_number TEXT, -- מספר מעקב משולח
admin_notes TEXT, -- הערות פנימיות

-- Timestamps
created_at TIMESTAMPTZ DEFAULT NOW(),
paid_at TIMESTAMPTZ,
purchased_at TIMESTAMPTZ,
shipped_at TIMESTAMPTZ,
delivered_at TIMESTAMPTZ
);
```

5.5 RLS Policies

```
-- Enable RLS
ALTER TABLE profiles ENABLE ROW LEVEL SECURITY;
ALTER TABLE registries ENABLE ROW LEVEL SECURITY;
ALTER TABLE items ENABLE ROW LEVEL SECURITY;
ALTER TABLE orders ENABLE ROW LEVEL SECURITY;

-- Profiles: everyone reads, owner updates
CREATE POLICY "Public read profiles" ON profiles FOR SELECT USING (true);
CREATE POLICY "Owner update profile" ON profiles FOR UPDATE USING (auth.uid() = id);
CREATE POLICY "Owner insert profile" ON profiles FOR INSERT WITH CHECK (auth.uid() = id);

-- Registries: public read if public, owner all
```

```

CREATE POLICY "Public read registries" ON registries FOR SELECT USING (is_public = true);
CREATE POLICY "Owner all registries" ON registries FOR ALL USING (auth.uid() = user_id);

-- Items: public read, owner all
CREATE POLICY "Public read items" ON items FOR SELECT
    USING (EXISTS (SELECT 1 FROM registries WHERE id = registry_id AND is_public = true));
CREATE POLICY "Owner all items" ON items FOR ALL
    USING (EXISTS (SELECT 1 FROM registries WHERE id = registry_id AND user_id = auth.uid()));

-- Orders: anyone insert, owner read
CREATE POLICY "Anyone insert orders" ON orders FOR INSERT WITH CHECK (true);
CREATE POLICY "Owner read orders" ON orders FOR SELECT
    USING (EXISTS (SELECT 1 FROM registries WHERE id = registry_id AND user_id = auth.uid()));

```

5.6 Functions & Triggers

```

-- Auto-create profile on signup
CREATE OR REPLACE FUNCTION public.handle_new_user()
RETURNS TRIGGER AS $$

BEGIN
    INSERT INTO public.profiles (id, email)
    VALUES (NEW.id, NEW.email);
    RETURN NEW;
END;
$$ LANGUAGE plpgsql SECURITY DEFINER;

CREATE TRIGGER on_auth_user_created
AFTER INSERT ON auth.users
FOR EACH ROW EXECUTE FUNCTION public.handle_new_user();

-- Generate unique slug
CREATE OR REPLACE FUNCTION generate_unique_slug(title TEXT)
RETURNS TEXT AS $$

DECLARE
    base_slug TEXT;
    final_slug TEXT;
    counter INTEGER := 0;

BEGIN
    base_slug := lower(regexp_replace(title, '[^a-zA-Z0-9ñ-ñ]+', '-', 'g'));
    final_slug := base_slug;
    WHILE EXISTS (SELECT 1 FROM registries WHERE slug = final_slug) LOOP
        counter := counter + 1;
        final_slug := base_slug || '-' || counter;
    END LOOP;
    RETURN final_slug;
END;
$$ LANGUAGE plpgsql;

```

XU/IU - כל המרכיבים בעברית 6.

6.1 Routes מבנה

/	דף נחיתה →
/login	התחברות →
/signup	הרשמה →
/onboarding	השלמת פרופיל →
/dashboard	לוח בקרה →
/dashboard/add	הוספת פריט →
/dashboard/edit/:id	עריכת פריט →
/dashboard/settings	הגדרות →
/r/:slug	טציגת אורח → (PUBLIC)
/r/:slug/buy/:itemId	רכישת מתנה →
/payment/success	הצלחת תשלום →
/payment/failed	כשל תשלום →
/admin	דשبورד Admin →

6.2 דף נחיתה (/)

[התחברות] [הרשמה] [לוגו]

רישימת הלידה שלך, במקום אחד

הוסף מוצריים מכל חנות, שתפי עם המשפחה
קבלי מתנות בלי כפילותיות

[צרי רשיימה חדשה] [חפשי רשיימה]

מכל חנות ✓ בלי כפילותיות ✓ משלוח ישיר
שילב, מוצאים, אנחנו מטעלים
ונעוד נקנה בהצל KSP,

6.3 Dashboard

[מסך] [יציאה] [לוגו]

שלום שרה! 😊
עוד 45 ימים עד הלידה

12	8	4
זמןינו	nakna	פריטים

[שתף את הרשיימה] [+ הוסיף פריט]

פילטר: הכל [▼] [זמןין]

[תמונה]	[תמונה]	[תמונה]
בקבוקים	עריסה	עגלה
₪1,500	₪2,200	₪89
[שיילב]	[IKEA]	[מושגים]
[נקנה]	[ערוך] [מחק]	✓ [ערוך] [מחק]

6.4 הוספה פריט

הוספי פריט חדש
:הבדיקי לינק מהחנות
https://www.shilav.co.il/...
[דפסי מוצר]
או —————
[הדני ידנית]

// After URL parsing success:

✓ !מצאנו את המוצר
[תמונה]
שם: Bugaboo_____
מחיר: 1500 ₪
▼] ■ שיילב
▼] ■ ציוד
▼] ■ גבואה
[—————] ■ הערות
[■ ביטול ✓] ■ שמרי

6.5 תצוגת אורך (/r/:slug)

הרשימה של שרה ודני!
עוד 45 ימים עד הגעה!

"אנחנו כל כך מתרגשים! תודה לכל מי שיעזר לנו להתארגו!"

[הכל] [ציוד] [הלבשה] [האכלה] [זמן בלבד]

[תמונה] בקבוקים ₪1,500	[תמונה] עריסת ₪2,200	[תמונה] עגלה ₪89
★ גבואה		

[קנה מתנה] [קנה מתנה ✓]

6.6 מודל רכישה

// Step 1: Buyer Info

[X] קנה מתנה 

[תמונה] Bugaboo עגלת תינוק
₪1,500

הפרטים שלך

: שם מלא: [_____]
: אימייל: [_____]
: טלפון: [_____]

: הودעה אישית (אופציונלי)
[_____]

הסתר אתשמי (אונונימי)

[→ הבא - כתובות למשLOW]

// Step 2: Shipping

[←] כתובות למשLOW

: שם מקבל המשLOW [_____]
: רחוב ומספר [_____]
: מיר [_____]
: טלפון למשLOW [_____]

[→ הבא - סיכום]

// Step 3: Summary

[←]

סיכום

מוצר: עגלת תינוק Bugaboo

₪1,500:

סה"כ לתשלום

נשלח ל: שרה כהן

רחוב הרצל 15, תל אביב

[לתשלום מאובטח]

התשלום מאובטח באמצעות חשבוניות יrokeha

7. קוד מלך - Edge Functions

7.1 create-payment (יצירת קישור תשלום)

```
// supabase/functions/create-payment/index.ts

import { serve } from 'https://deno.land/std@0.168.0/http/server.ts';
import { createClient } from 'https://esm.sh/@supabase/supabase-js@2';

const GREEN_INVOICE_API = 'https://api.greeninvoice.co.il/api/v1';
const corsHeaders = {
  'Access-Control-Allow-Origin': '*',
  'Access-Control-Allow-Headers': 'authorization, x-client-info, apikey, content-type',
};

// Get JWT token from Green Invoice
async function getToken(): Promise<string> {
  const res = await fetch(`${
    GREEN_INVOICE_API
  }/account/token`, {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({
      id: Deno.env.get('GREEN_INVOICE_API_KEY'),
      secret: Deno.env.get('GREEN_INVOICE_API_SECRET'),
    }),
  });
  const data = await res.json();
  return data.token;
}

serve(async (req) => {
  // Handle CORS
  if (req.method === 'OPTIONS') {
    return new Response(null, { headers: corsHeaders });
  }

  try {
    const body = await req.json();
    const {
      item_id, item_name, item_price, registry_id, registry_title,
      buyer_name, buyer_email, buyer_phone, buyer_message, is_anonymous,
      shipping_name, shipping_address, shipping_city, shipping_phone,
    } = body;

    // 1. Initialize Supabase
    const supabase = createClient(
      Deno.env.get('SUPABASE_URL')!,
      Deno.env.get('SUPABASE_SERVICE_ROLE_KEY')!
    );

    // 2. Create order in DB
    const { data: order, error } = await supabase
      .from('orders')
      .insert({
        item_id,
        registry_id,
        buyer_name,
      })
  }
})
```

```

        buyer_email,
        buyer_phone,
        buyer_message,
        is_anonymous,
        shipping_name,
        shipping_address,
        shipping_city,
        shipping_phone,
        amount_paid: item_price,
        status: 'pending_payment',
    })
.select()
.single();

if (error) throw error;

// 3. Get Green Invoice token
const token = await getToken();

// 4. Create payment form with DYNAMIC amount
const paymentRes = await fetch(`.${GREEN_INVOICE_API}/paymentForm/create`, {
    method: 'POST',
    headers: {
        'Authorization': `Bearer ${token}`,
        'Content-Type': 'application/json',
    },
    body: JSON.stringify({
        type: 320, // קבלת
        lang: 'he',
        currency: 'ILS',
        pluginId: order.id,
        client: {
            name: buyer_name,
            emails: [buyer_email],
            phone: buyer_phone,
            add: false,
        },
        income: [
            {
                description: item_name,
                quantity: 1,
                price: item_price, // 💡 DYNAMIC PRICE
                currency: 'ILS',
                vatType: 1,
            },
        ],
        remarks: `מתנה מרשות הלאינה: ${registry_title}`,
        successUrl: `.${Deno.env.get('APP_URL')}/payment/success?order=${order.id}`,
        failureUrl: `.${Deno.env.get('APP_URL')}/payment/failed?order=${order.id}`,
        notifyUrl: `.${Deno.env.get('SUPABASE_URL')}/functions/v1/green-invoice-webhook`,
        custom: JSON.stringify({ order_id: order.id, item_id, registry_id }),
    }),
});

const paymentData = await paymentRes.json();

if (paymentData.errorCode) {
    throw new Error(paymentData.errorMessage);
}

```

```
// 5. Update order with payment form ID
await supabase
  .from('orders')
  .update({ green_invoice_form_id: paymentData.id })
  .eq('id', order.id);

// 6. Return payment URL
return new Response(
  JSON.stringify({ success: true, paymentUrl: paymentData.url, orderId: order.id }),
  { headers: { ...corsHeaders, 'Content-Type': 'application/json' } }
);

} catch (error) {
  return new Response(
    JSON.stringify({ success: false, error: error.message }),
    { status: 500, headers: { ...corsHeaders, 'Content-Type': 'application/json' } }
);
}

});
```

7.2 green-invoice-webhook (טיפול בתשלומים)

```
// supabase/functions/green-invoice-webhook/index.ts

import { serve } from 'https://deno.land/std@0.168.0/http/server.ts';
import { createClient } from 'https://esm.sh/@supabase/supabase-js@2';

const RESEND_API = 'https://api.resend.com/emails';

async function sendEmail(to: string, subject: string, html: string) {
  await fetch(RESEND_API, {
    method: 'POST',
    headers: {
      'Authorization': `Bearer ${Deno.env.get('RESEND_API_KEY')}`,
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      from: 'רשות מים <noreply@reshimat-tinok.co.il>',
      to,
      subject,
      html,
    }),
  });
}

serve(async (req) => {
  try {
    const payload = await req.json();
    console.log('Webhook received:', payload);

    // Check if payment successful
    if (payload.type !== 'payment.success' && payload.status !== 1) {
      return new Response(JSON.stringify({ received: true }));
    }

    const orderId = payload.pluginId || payload.data?.pluginId;
    const docUrl = payload.documentUrl || payload.data?.documentUrl;
    const docId = payload.documentId || payload.data?.documentId;

    const supabase = createClient(
      Deno.env.get('SUPABASE_URL')!,
      Deno.env.get('SUPABASE_SERVICE_ROLE_KEY')!
    );

    // 1. Get order with relations
    const { data: order } = await supabase
      .from('orders')
      .select(`*, items(*), registries(*, profiles(*))`)
      .eq('id', orderId)
      .single();

    if (!order) {
  
```

```

    throw new Error(`Order not found: ${orderId}`);
}

// 2. Update order status
await supabase
  .from('orders')
  .update({
    status: 'paid',
    paid_at: new Date().toISOString(),
    green_invoice_doc_id: docId,
    green_invoice_doc_url: docUrl,
  })
  .eq('id', orderId);

// 3. Update item status
const displayName = order.is_anonymous ? 'anonim' : order.buyer_name;
await supabase
  .from('items')
  .update({
    status: 'purchased',
    purchased_by: displayName,
  })
  .eq('id', order.item_id);

// 4. Send email to buyer
await sendEmail(
  order.buyer_email,
  `✓ אישור רכישה - ${order.items.name}`,
  `

# תודה על רכישת ${order.items.name}! 🎁



היה ${order.buyer_name},



תודה לך על הקבלה בהצלחה!



חוצץ: ${order.items.name}



סכום: ${order.amount_paid}



מרשתת הלידה: ${order.registries.title}



כתובת לשלוח: ${order.shipping_name}, ${order.shipping_address}, ${order.shipping_city}



המתנה תישלח תוך 3-5 ימי עסקים.


    ${docUrl ? `

צפייה במסמך

` : ''}
    

צוות רשות מים תינוק/בוככה

`);
}

// 5. Send email to registry owner
const ownerEmail = order.registries.profiles.email;
const ownerName = order.registries.profiles.full_name;
await sendEmail(
  ownerEmail,
  `🎉 ${displayName}, !
  ייש לך מתנה חזקה! 🎁`,
  `

# תודה לך על רכישת ${ownerName}! 🎁



היה ${ownerName},



שם: ${displayName}

`);

```

```

        <p style="font-size:20px;"><strong>${order.items.name}</strong></p>
        ${order.buyer_message} ? `<p style="font-style:italic;">"${order.buyer_message}"</p>` :
        ```
 </div>
 <p>!המתנה בדף אליז</p>
 <p> צוות רשותת תינוק
,ביבכה</p>
</div>` ;
) ;

// 6. Send email to Admin
await sendEmail(
 Deno.env.get('ADMIN_EMAIL')!,
 `✉️ זמנה חדשה - ${orderId.slice(0, 8)} `,
 `<div dir="rtl" style="font-family:Arial;">
 <h1>!הזמנה חדשה לטיול</h1>
 <h2>📦 פרט המוצר</h2>

 מוצר: ${order.items.name}
 מחיר:₪ ${order.items.price}
 שם: ${order.items.store_name}
 ליקן: ליקן

 <h2>🚚 פרטימשלוח</h2>

 שם: ${order.shipping_name}
 כתובת: ${order.shipping_address}, ${order.shipping_city}
 טלפון: ${order.shipping_phone}

 <h2>סכום תשלום:</h2>

 סכום:₪ ${order.amount_paid}
 קבלת: מסמך

 <p>דשbor פתק</p>
 </div>`;
);

return new Response(JSON.stringify({ success: true }));
}

} catch (error) {
 console.error('Webhook error:', error);
 return new Response(
 JSON.stringify({ error: error.message }),
 { status: 500 }
);
}
);
}
);

```

### 7.3 scrape-url (חילוץ מוצר)

```
// supabase/functions/scrape-url/index.ts

import { serve } from 'https://deno.land/std@0.168.0/http/server.ts';
import { DOMParser } from 'https://deno.land/x/deno_dom/deno-dom-wasm.ts';

const corsHeaders = {
 'Access-Control-Allow-Origin': '*',
 'Access-Control-Allow-Headers': 'authorization, x-client-info, apikey, content-type',
};

// Store detection
function detectStore(url: string): string {
 if (url.includes('shilav.co.il')) return 'שילוב';
 if (url.includes('motsesim.co.il')) return 'מויסים';
 if (url.includes('baby-star.co.il')) return 'בָּבִי-סְטָאָר';
 if (url.includes('super-pharm.co.il')) return 'סֶוּפֵר-פַּארָם';
 if (url.includes('ksp.co.il')) return 'KSP';
 if (url.includes('bug.co.il')) return 'בָּגָן';
 if (url.includes('ikea.co.il')) return 'IKEA';
 return 'אַרְזָן';
}

serve(async (req) => {
 if (req.method === 'OPTIONS') {
 return new Response(null, { headers: corsHeaders });
 }

 try {
 const { url } = await req.json();

 // Fetch the page
 const response = await fetch(url, {
 headers: {
 'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36',
 'Accept-Language': 'he-IL,he;q=0.9,en;q=0.8',
 },
 });

 const html = await response.text();
 const doc = new DOMParser().parseFromString(html, 'text/html');

 if (!doc) {
 throw new Error('Failed to parse HTML');
 }

 // Extract OG tags
 const getMeta = (property: string): string | null => {
 const el = doc.querySelector(`meta[property="${property}"]`) ||
 doc.querySelector(`meta[name="${property}"]`);
 return el?.getAttribute('content') || null;
 };

 // Try OG tags first
 let name = getMeta('og:title') || doc.querySelector('h1')?.textContent;
```

```
let image = getMeta('og:image');
let priceStr = getMeta('product:price:amount') ||
 getMeta('og:price:amount');

// Fallback: look for price in page
if (!priceStr) {
 const priceEl = doc.querySelector('[class*="price"]') || doc.querySelector('[itemprop="price"]');
 priceStr = priceEl?.textContent?.replace(/[^0-9.]/g, '') || null;
}

const price = priceStr ? parseFloat(priceStr) : null;
const store = detectStore(url);

if (!name) {
 throw new Error('Could not extract product name');
}

return new Response(
 JSON.stringify({
 success: true,
 data: {
 name: name.trim(),
 price,
 image_url: image,
 store_name: store,
 original_url: url,
 },
 }),
 { headers: { ...corsHeaders, 'Content-Type': 'application/json' } }
);

} catch (error) {
 return new Response(
 JSON.stringify({ success: false, error: error.message }),
 { headers: { ...corsHeaders, 'Content-Type': 'application/json' } }
);
}
});
```

## 8. לוח זמנים - 8 ימים (שעה אחר שעה)

הנחה: עבודה אינטנסיבית ~10 שעות ביום.

### 1: Setup + Auth

שעה	משימה	פלט
1-2	צור Vite + React + TS project	Project folder
2-3	הגדיר Tailwind עם RTL	RTL working
3-4	צור Supabase project + tables	DB ready
4-5	צור RLS policies	Security set
5-6	בנה דף Login	Login UI
6-7	בנה דף Signup	Signup UI
7-8	הוסף Supabase Auth	Auth working
8-9	הוסף Google OAuth	OAuth working
9-10	בנה AuthContext + Protected Routes	Navigation working

✓ Deliverable: משתמש יכול להירשם ולהתחבר

### 2: Profile + Registry + Items (Manual)

שעה	משימה	פלט
1-2	בנה טופס השלמה פרופיל	Profile form
2-3	הוסף Supabase-חבר לשמירה ב	Profile saves
3-4	ראשוונה Registry בונה יצירה	Registry created
4-5	בנה Dashboard layout	Dashboard UI
5-6	בנה ItemCard component	Cards display
6-7	בנה מודל 'הוסף פריט' - ידני	Add modal
7-8	הוסף Insert ל-DB חבר	Items save
8-9	הוסף Edit item	Edit works
9-10	הוסף Delete item	Delete works

✓ Deliverable: אפשר ליצור רשימה ולהוסיף פרטיים ידנית

### 3: URL Scraper

שעה	משימה	פלט
1-2	צור Edge Function scrape-url	Function created
2-3	הוסף parser ל-Shilav כתוב	Shilav works
3-4	הוסף parser ל-Motsesim כתוב	Motsesim works
4-5	הוסף parser ל-Baby-Star כתוב	Baby-Star works
5-6	בנה URL הדבקת UI	Paste UI
6-7	הוסף Loading state	Loading shows
7-8	לפניהם שמירה Preview הצג	Preview works
8-9	הוסף Fallback ל-manual	Fallback works
9-10	20 URLs אמיתיים בדוק	Testing done

✓ Deliverable: הבדיקה URL → Auto-fill URL

### 4: Guest View

שעה	משימה	פלט
1-2	בנה route /r/:slug	Route works

שעה	משימה	פלט
2-3	בנה Guest Header	Header shows
3-4	של פריטים grid הציג	Items display
4-5	Guest Cards עצב	Cards look good
5-6	הוסף פילטרים	Filters work
6-7	'בנה כפתור' קינה מתנה	Button shows
7-8	צור מודול רכישה שלב 1	Step 1 done
8-9	צור מודול רכישה שלב 2	Step 2 done
9-10	צור מודול רכישה שלב 3	Step 3 done

✓ Deliverable: אורךים רואים רשימה ומתחילה קניה

## 5: Green Invoice Payment

שעה	משימה	פלט
1-2	API credentials n-Green Invoice קבל	Credentials ready
2-3	Edge Function create-payment צור	Function created
3-4	getToken כתוב קוד	Auth works
4-5	createPaymentForm כתוב קוד	Payment links work
5-6	-חבר כפתור 'ל转身' function	Button redirects
6-7	payment link יוצרת בדוק	Links created
7-8	Edge Function webhook צור	Webhook created
8-9	status -webhook: ערך טפל ב	Status updates
9-10	flow-test ב דוחן מלא	Flow works

✓ Deliverable: תשלום עובד קצת לפחות

## 6: Emails + Admin

שעה	משימה	פלט
1-2	Resend account + API key הגדר	Resend ready
2-3	כתוב תבנית אימייל לקונה	Buyer email ready
3-4	כתוב תבנית אימייל לבעל רשימה	Owner email ready
4-5	-Admin-כתוב תבנית אימייל ל	Admin email ready
5-6	-חבר שליחה ב webhook	Emails sending
6-7	בדוק קבלת כל 3 אימיילים	Emails received
7-8	Admin Dashboard - list בניית	Admin list
8-9	status הוסף כפתורי	Status buttons
9-10	הוסף שליחת אימייל משולח	Shipped email works

✓ Deliverable: כל האימיילים עובדים + Admin Dashboard

## 7: Sharing + Polish

שעה	משימה	פלט
1-2	בנה מודול Share	Share modal
2-3	הוסף Copy Link	Copy works
3-4	הוסף WhatsApp share	WhatsApp works
4-5	הגדיר OG meta tags	Preview shows
5-6	Responsive audit - mobile	Mobile fixed
6-7	הוסף Loading skeletons	Skeletons show

שעה	משימה	פלט
7-8	הוסף Error states	Errors handled
8-9	RTL issues תקן	RTL perfect
9-10	בנה דף נחיתה	Landing page done

✓ Deliverable: מילוטש UI + שיתוף עובד

## 8 ימים: Test + Deploy 🚀

שעה	משימה	פלט
1-2	כתב 15 test cases	Test list ready
2-4	manual testing בצע	Bugs found
4-6	bugs תקן	Bugs fixed
6-7	RLS security בדוק	Security OK
7-8	Build + Deploy to GitHub Pages	Site live
8-9	production בדוק	Production OK
9-10	צור רשימות בדיקה + שלח לחברים	Beta testing

✓ Deliverable: MVP LIVE!

## 9. Deployment ל-GitHub Pages

### הכנות הפרוייקט 1

```
// vite.config.ts
import { defineConfig } from 'vite';
import react from '@vitejs/plugin-react';

export default defineConfig({
 plugins: [react()],
 base: '/reshimat-tinok/', // ה נוּוּ-repo
});
```

### 9.2 GitHub Actions Workflow

```
.github/workflows/deploy.yml

name: Deploy to GitHub Pages

on:
 push:
 branches: [main]

jobs:
 deploy:
 runs-on: ubuntu-latest
 steps:
 - uses: actions/checkout@v3

 - name: Setup Node
 uses: actions/setup-node@v3
 with:
 node-version: 18

 - name: Install
 run: npm ci

 - name: Build
 run: npm run build
 env:
 VITE_SUPABASE_URL: ${{ secrets.VITE_SUPABASE_URL }}
 VITE_SUPABASE_ANON_KEY: ${{ secrets.VITE_SUPABASE_ANON_KEY }}

 - name: Deploy
 uses: peaceiris/actions-gh-pages@v3
 with:
 github_token: ${{ secrets.GITHUB_TOKEN }}
 publish_dir: ./dist
```

### 9.3 Environment Variables

#### GitHub Secrets (Settings → Secrets)

```
VITE_SUPABASE_URL=https://xxx.supabase.co
VITE_SUPABASE_ANON_KEY=xxx
```

#### Supabase Secrets (Edge Functions)

```
GREEN_INVOICE_API_KEY=xxx
```

```
GREEN_INVOICE_API_SECRET=xxxx
RESEND_API_KEY=xxxx
ADMIN_EMAIL=your@email.com
APP_URL=https://username.github.io/reshimat-tinok
```

## 9.4 לאחר MVP - Custom Domain

משלך domain עובד, תוכל לחבר custom-domain שתה:

1. domain (משלך reshimat-tinok.co.il) קנה
2. CNAME record הגדר username.github.io
3. הוסף public ה-TIINOK
4. APP\_URL עדכן את Supabase secrets
5. successUrl/failureUrl עדכן את Green Invoice

# 10. QA בדיקות.

## 10.1 Test Cases

#	תרחיש	תוצאה צפוייה
1	הרשמה עם אימייל	חשבון נוצר
2	התחברות Google	עובר לדשبورד
3	השלמת פרופיל	נתונים נשמרים
4	יצירת רשימה	ונאש רשימה נוצרת עם
5	שילוב URL הדבקת מוצר	מוצר נטען
6	מוצאים URL הדבקת מוצר	מוצר נטען
7	הוספה ידנית	פריט נשמר
8	עריכת פריט	שינויים נשמרים
9	מחיקת פריט	פריט נמחק
10	צפיה כארוח	רשימה מוצגת
11	פילטרים	מסננים עובדים
12	התחלת רכישה	מודל פתוח
13	(test) תשלום	Redirect ל-Green Invoice
14	Webhook	מתעדכן Order
15	אימייל קונה	מגיע תוך דקה
16	אימייל בעל רשימה	מגיע תוך דקה
17	אימייל Admin	מגיע תוך דקה
18	WhatsApp שיתוף	תיקין Preview
19	Mobile responsive	נראה טוב
20	RTL	הכל בכיוון נכון

## 10.2 Security Checklist

- פועל על כל הtablאות RLS
- של אחרים `orders`-אי אפשר לגשת ל-
- API keys ב `client`-לא חשובים
- Webhook (אופציונלי) מאומנת `signature`
- Input validation בכל השדות

## 11. MVP-לאחר ה-

### 11.1 שבוע 2: Stabilization

- לfy bugs תיקן feedback
- custom domain
- הוסף analytics (Plausible)
- הוסף error tracking (Sentry)

### 11.2 חודש 1: שיפורים

- להנויות נספנות הוסף scrapers
- Admin Dashboard - שפר את ה
- אוטומטי הוסף Welcome אימייל
- הוסף חיפוש רשיימה

### 11.3 חודש 2: פיצ'רים חדשים

- Group Gifting - מספר תורמים
- Cash Fund - 'תרמו לחדר'
- Checklist מובנה
- Thank You tracker

### 11.4 חודש 3: Scale

- אם יש +100 הזמנות - שקל אוטומטית
- Premium tier (אופציונלי)
- שיוק בקבוצות פייסבוק
- שותפות עם בלוגריות

## גנומך: Quick Reference

### A. Commands

```
Setup
npm create vite@latest reshimat-tinok -- --template react-ts
cd reshimat-tinok
npm install @supabase/supabase-js @tanstack/react-query react-router-dom
npm install -D tailwindcss postcss autoprefixer
npx tailwindcss init -p

Supabase
npx supabase init
npx supabase functions new create-payment
npx supabase functions new green-invoice-webhook
npx supabase functions new scrape-url
npx supabase functions deploy --all

Run locally
npm run dev

Deploy
npm run build
git add . && git commit -m 'deploy' && git push
```

### B. חניות נתמכות

חניות	Domain	עדיפות
שילב	shilav.co.il	P0
מוצאים	motsesim.co.il	P0
בייבי-סטאר	baby-star.co.il	P0
סופר-פארם	super-pharm.co.il	P1
KSP	ksp.co.il	P1
IKEA	ikea.co.il	P2

### C. קטגוריות

```
zieg = ציוד (עגלות , כיסאות בטיחות)
clothing = הלבשה
feeding = האכלה (בקבוקים , משאבות)
health = בריאות (מדי חום , מוניטורים)
nursery = חדר תינוק (עיריסה , שידקה)
other = אחר
```

— סוף המסמך —  
!בהצלחה!

8 ימים Um Green Invoice - MVP עם