

רשימת תינוק

BabyList Israel

מסמך אפיון מוצר מלא - גרסה 4.0

 **MVP-ספרינט 8 ימים ל**

Tech Stack

React + Vite + TypeScript + Supabase + GitHub Pages

Payment Provider

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

Language: עברית בלבד

Model: Concierge Fulfillment (MVP-ללא עמלה ב)

תוכן עניינים

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

סיכום מנהלים ומודל עסקי.1

1.1 מה אנחנו בונים

KSP, פלטפורמה ישראלית לניהול רשימות לידה אוניברסלית. הורים מוסיפים מוצרים מכל חנות ישראלית (שילב, מוצצים, סופר-פארם, ועוד), (חברים ומשפחה רואים את הרשימה, משלמים דרכנו, ואנחנו קונים ושולחים את המתנה.

1.2 מודל Concierge - איך זה עובד

"במקום לבנות אינטגרציות מורכבות עם חנויות, אנחנו מפעילים מודל "קונסיירז

1. אורח בוחר מתנה מהרשימה
2. (Green Invoice) אורח משלם דרך חשבונית ירוקה
3. אנחנו מקבלים התראה +כסף
4. אנחנו קונים את המוצר מהחנות המקורית (ידנית)
5. אנחנו שולחים לכתובת שהאורח הזין
6. כולם מקבלים עדכון באימייל

1.3 ללא עמלה - MVP אסטרטגיית

אנחנו לא גובים עמלה - המחיר שהאורח משלם = המחיר בחנות MVP-ב

"בלי חיכוך של "למה לשלם יותר פה Product-Market Fit למה? כדי לבדוק

מה אנחנו מרוויחים בזמן הזה:

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

1.4 למה Green Invoice?

קריטריון	Green Invoice	Lemon Squeezy
עמלה	2.5%	5% + \$0.50
תמיכה בעברית	מלאה	אין
חשבוניות	אוטומטי (חובה בישראל!)	לא רלוונטי
זמן הקמה	!כבר יש לכם	יום
לסכום דינמי API	כן ✓	כן ✓
Webhooks	כן ✓	כן ✓
אמון הלקוחות	גבוה (מותג ישראלי)	בינוני

1.5 טכני Stack

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

2. (חשבונית ירוקה) Green Invoice אינטגרציית

2.1 איך זה עובד - תמונה גדולה

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

1. (status: pending_payment) שלנו DB-יוצרים הזמנה ב
2. עם הסכום הספציפי של Green Invoice של API-קוראים ל
3. ייחודי לתשלום URL מקבלים
4. מעבירים את האורח לדף התשלום
5. עם אישור Webhook אחרי תשלום - מקבלים
6. ושולחים אימיילים DB-מעדכנים את ה

2.2 בחשבונית ירוקה API הגדרת

API שלב 1: קבלת מפתחות

בדשבורד של חשבונית ירוקה:

צור מפתח חדש → API → הגדרות

תקבלו:

- API_KEY (מזהה)
- API_SECRET (סוד)

Webhook שלב 2: הגדרת

בדשבורד:

הגדרות → Webhooks → URL הוסף

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

סמנו את האירועים: payment.success, payment.failure

2.3 API Reference - יצירת קישור תשלום

Endpoint

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

Headers

Authorization: Bearer <JWT_TOKEN>

Content-Type: application/json

Body - מלאה דוגמה

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

  "client": { // פרטי הלקוח
    "name": "ישראל ישראלי",
    "emails": ["israel@email.com"],
    "phone": "0501234567",
    "add": false // לא להוסיף לרשימת לקוחות
  },

  "income": [ // הסכום הדינמי
    {
      "description": "Bugaboo עגלת תינוק",
      "quantity": 1,
      "price": 1500, // מחיר המוצר
      "currency": "ILS",
```

```

    "vatType": 1 // כולל מע"מ
  }],

  "remarks": "מתנה מרשימת 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\"}" // webhook מידע נוסף ל
}

```

Response

```

{
  "id": "abc123",
  "url": "https://app.greeninvoice.co.il/pay/abc123" // לשלוח 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 לקבלה
    "pluginId": "order-uuid", // המזהה שלנו!
    "custom": "{...}" // המידע הנוסף ששלחנו
  }
}

```

מערכת התראות אימייל

סוגי אימיילים

אימייל	נמען	טריגר	תוכן עיקרי
אישור רכישה	הקונה (אורח)	Webhook success	פרטי הזמנה, לינק לקבלה
התראת מתנה	בעל/ת הרשימה	Webhook success	מישהו קנה לך + פרטים
Admin התראת	אתה	Webhook success	יש הזמנה לטפל - כל הפרטים
עדכון משלוח	הקונה	Admin action	המתנה נשלחה + מספר מעקב
Welcome	נרשם חדש	After signup	ברוכים הבאים + הסבר


הגדרה - Resend

נקי API, לשליחת אימיילים. פשוט, זול, Resend-נשתמש ב

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

תבניות אימייל בעברית

אימייל 1: אישור רכישה (לקונה)


נושא:  אישור רכישה - [שם המוצר]

[היי שם הקונה],

תודה שרכשת מתנה מרשימת הלידה של [שם ההורים]

 פרטי ההזמנה:

- מוצר: [שם המוצר]
- סכום ששולם: ₪ [מחיר]
- [order_id]: מספר הזמנה


 כתובת למשלוח:

[שם], [כתובת], [עיר]


המתנה תישלח תוך 3-5 ימי עסקים.
נעדכן אותך כשהיא בדרך!

[לינק לקבלה מחשבונית ירוקה]


בברכה,

 צוות רשימת תינוק


אימייל 2: התראת מתנה (לבעל הרשימה)

נושא:  מישהו קנה לך מתנה!

[היי שם],

 יש בשורות טובות

: [שם הקונה " /מישהו "רכש/ה לך]

[שם המוצר] 

[הודעה אישית - אם יש]

!המתנה בדרך אליך

לצפייה ברשימה שלך]: לינק]

בברכה,

צוות רשימת תינוק 🍼

(אליך) Admin אימייל 3:התראת

נושא: 📧 הזמנה חדשה - [מספר הזמנה]

=== הזמנה חדשה לטיפול ===

📦 פרטי המוצר:

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

🚚 פרטי משלוח:

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

🏠 פרטי תשלום:

- שולם]: שם סכום]
- קבלה]: לינק לחשבונית ירוקה]

➡ לטיפול בדשבורד]: לינק]

אימייל 4:עדכון משלוח

נושא: 🚚 המתנה שלך בדרך

,היי [שם]

📦!המתנה ששלחת ל[שם ההורים] יצאה לדרך

📦 [שם המוצר]

🚚 [tracking]: מספר מעקב

!תודה שהשתמשת ברשימת תינוק

בברכה,

צוות רשימת תינוק 🍼

דרישות פונקציונליות מפורטות

מערכת משתמשים

הרשמה

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

(אופציונלי) Google OAuth, Magic Link: אפשרויות נוספות

השלמת פרופיל (אחרי הרשמה)

שדה	סוג	חובה	הערות
שם מלא	text	✓	להצגה ברשימה
טלפון	tel	✓	לתיאום משלוחים
תאריך לידה משוער	date	✓	לספירה לאחור
מין התינוק	select	✗	בן/בת/הפתעה/תאומים
שם בן/בת זוג	text	✗	להצגה ברשימה

ניהול רשימה

יצירת רשימה

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

הוספת פריט

שני מצבים:

(אוטומטי) URL מצב 1: הדבקת

- משתמש מדביק לינק מחנות
- מנסה לחלץ שם: תמונה, מחיר, חנות Edge Function
- לאישור Preview אם הצליח - מציג
- משתמש יכול לערוך לפני שמירה

(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	✓	בעברית
מחיר	number	✓	בשקלים
לינק מקורי	url	✓	לחנות


שדה	סוג	חובה	הערות
תמונה	url/file	✓	או העלאה URL
חנות	select	✓	שילב/מוצצים/אחר
קטגוריה	select	✓	ציוד/הלבשה/האכלה/בריאות/חדר
כמות	number	X	ברירת מחדל: 1
עדיפות	select	X	גבוהה/בינונית/נמוכה
הערות	textarea	X	מידה, צבע, העדפה

4.3 (Public) תצוגת אורח

מה האורח רואה

- כותרת הרשימה + שמות ההורים
- ספירה לאחור ללידה ("עוד 45 ימים!")
- הודעה אישית מההורים
- רשת מוצרים (Grid)
- פילטרים: קטגוריה, חנות, מחיר, זמין/נקנה
- כפתור "קנה מתנה" לכל פריט זמין

זרימת רכישה

1. "אורח לוחץ  קנה מתנה"
2. נפתח מודל - שלב 1: פרטי הקונה
3. שלב 2: כתובת למשלוח
4. שלב 3: סיכום + אישור
5. לחיצה על "לתשלום" → "יצירת הזמנה"
6. Green Invoice לדף תשלום של Redirect
7. עדכון סטטוס + אימיילים → Webhook → תשלום
8. לדף הצלחה Redirect

4.4 (בשבילך) Admin דשבורד

- רשימת כל ההזמנות לפי סטטוס
- פרטי כל הזמנה: מוצר, לינק, כתובת, טלפון
- (purchased) "כפתור" סמן כנקנה
- (shipped) כפתור "סמן כנשלח" + "הזנת מספר 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', -- מחתין לתשלום
    'paid',             -- שולם
    'purchased',        -- נקנה מהחנות
    'shipped',          -- נשלח
    'delivered',        -- הגיע
    'cancelled'         -- בוטל
)),

-- Fulfillment tracking
store_order_id TEXT,      -- מספר הזמנה מהחנות
tracking_number TEXT,     -- מספר מעקב משלוח
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 $$
DECLARE
  base_slug TEXT;
  final_slug TEXT;
  counter INTEGER := 0;
BEGIN
  base_slug := lower(regexp_replace(title, '^[a-zA-Z0-9n-]+', '-', 'g'));
  final_slug := base_slug;
  WHILE EXISTS (SELECT 1 FROM registries WHERE slug = final_slug) LOOP
    counter := counter + 1;
    final_slug := base_slug || '-' || counter;
  END LOOP;
  RETURN final_slug;
END;
$$ LANGUAGE plpgsql;

```

6. UI/UX - כל המסכים בעברית

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	→ תצוגת אורח (PUBLIC)
/r/:slug/buy/:itemId	→ רכישת מתנה
/payment/success	→ הצלחת תשלום
/payment/failed	→ כשל תשלום
/admin	→ Admin דשבורד

6.2 דף נחיתה (/)

לוגו

התחברות

הרשמה

👶

רשימת הלידה שלך, במקום אחד

הוסיפי מוצרים מכל חנות, שתפי עם המשפחה, קבלי מתנות בלי כפילויות

צרי רשימה חדשה

חפשי רשימה

מכל חנות שילב, מוצעים ועוד

✓ בלי כפילויות, רואים מה כבר נקנה

✓ משלוח ישיר ✓ אנחנו מטפלים בהכל KSP

6.3 Dashboard

לוגו

שם

יציאה

👶

שלום שרה

עוד 45 ימים עד הלידה

12

נקנו

8

זמינים

4

פריטים

+

הוסיפי פריט

שתפי את הרשימה

פילטר

הכל

זמין

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

הוספת פריט

הוסיפי פריט חדש

הדביקי לינק מהחנות

<https://www.shilav.co.il/...>

[חפשי מוצר]

או

[הזיני ידנית]

// After URL parsing success:

✓! מצאנו את המוצר

[תמונה]

שם: [Bugaboo עגלת תינוק] _____

ש [1500]: מחיר

▼ חנות: שילב []

▼ קטגוריה: ציוד []

▼ עדיפות: גבוהה []

הערות: [_____]

[שמרי ✓] [ביטול]

6.5 תצוגת אורח (/r:/slug)

🥰 הרשימה של שרה ודני
!עוד 45 ימים עד ההגעה

"אנחנו כל כך מתרגשים! תודה לכל מי שיעזור לנו להתארגן"		
[הכל]	[ציוד]	[הלבושה] [האכלה]
[זמין בלבד]		

[תמונה] בקבוקים ₪1,500 גבוהה ⭐ [נקנה] ✓	[תמונה] עריסה ₪2,200 קטנה 🎁 [קנה מתנה]	[תמונה] עגלה ₪89 גדולה 🎁 [קנה מתנה]
---	--	---

מודל רכישה 6.6

```
// Step 1: Buyer Info
```

X

קנה מתנה 🎁

Bugaboo עגלת תינוק

1,500₪

תמונה

הפרטים שלך

שם מלא:

אימייל:

טלפון:

הודעה אישית (אופציונלי)

☐ הסתר את שמי (אנונימי)

→ הבא - כתובת למשלוח

```
// Step 2: Shipping
```

[←]	כתובת למשלוח
שם מקבל המשלוח: [_____] רחוב ומספר: [_____] עיר: [_____] טלפון למשלוח: [_____]	
[→ הבא - סיכום]	

// Step 3: Summary

סיכום	
<div> <div>Bugaboo מוצר: עגלת תינוק</div> <div> <div>סה"כ לתשלום</div> <div>1,500₪:</div> </div> </div>	
<p>נשלח ל: שרה כהן</p> <p>רחוב הרצל 15, תל אביב</p> <p>[לתשלום מאובטח]</p> <p>התשלום מאובטח באמצעות חשבונות ירוקה</p>	

7. Edge Functions - קוד מלא

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, // קבלה
    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 מרשימת הלידה: ${registry_title}`,
    successUrl: `${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 בתשלום)

```
// 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: 'רשימת תינוק <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 ? 'אנונימי' : 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}`,
  `<div dir="rtl" style="font-family:Arial;max-width:600px;margin:0 auto;">
    <h1 style="color:#954535;">תודה על הרכישה! 📦</h1>
    <p>היי ${order.buyer_name},</p>
    <p>רכישתך התקבלה בהצלחה!</p>
    <div style="background:#f5f5f5;padding:20px;border-radius:10px;margin:20px 0 0 0;">
      <p><strong>מוצר:</strong> ${order.items.name}</p>
      <p><strong>סכום:</strong> ₪${order.amount_paid}</p>
      <p><strong>מרשימת הלידה של:</strong> ${order.registries.title}</p>
    </div>
    <p>🚚 כתובת למשלוח: ${order.shipping_name}, ${order.shipping_address},
    ${order.shipping_city}</p>
    <p>המחנה תשלח תוך 3-5 ימי עסקים.</p>
    ${docUrl ? `<p><a href="${docUrl}">צפיה בקבלה</a></p>` : ''}
    <p>צוות רשימת תינוק 🍌, בברכה</p>
  </div>`
);

// 5. Send email to registry owner
const ownerEmail = order.registries.profiles.email;
const ownerName = order.registries.profiles.full_name;
await sendEmail(
  ownerEmail,
  `📦 לקנה לך מחנה ${displayName}`,
  `<div dir="rtl" style="font-family:Arial;max-width:600px;margin:0 auto;">
    <h1 style="color:#954535;">יש לך מחנה חדשה! 📦</h1>
    <p>היי ${ownerName},</p>
    <p><strong>${displayName}</strong> לך לרכוש</p>
    <div style="background:#f5f5f5;padding:20px;border-radius:10px;margin:20px 0 0 0;">

```

```

    <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>החמנה בדרך אליך</p>
    <p>צוות רשימת תינוק 🍷</p>
  </div>`
);

// 6. Send email to Admin
await sendEmail(
  Deno.env.get('ADMIN_EMAIL')!,
  `📧 הזמנה חדשה - ${orderId.slice(0, 8)}`,
  `<div dir="rtl" style="font-family:Arial;">
    <h1>הזמנה חדשה לטיפול</h1>
    <h2>📦 פרטי המוצר</h2>
    <ul>
      <li><strong>מוצר</strong> ${order.items.name}</li>
      <li><strong>מחיר</strong> ₪${order.items.price}</li>
      <li><strong>חנות</strong> ${order.items.store_name}</li>
      <li><strong>לינק</strong> <a href="${order.items.original_url}">לחנות</a></li>
    </ul>
    <h2>🚚 פרטי משלוח</h2>
    <ul>
      <li><strong>שם</strong> ${order.shipping_name}</li>
      <li><strong>כתובת</strong> ${order.shipping_address}, ${order.shipping_city}</li>
      <li><strong>טלפון</strong> ${order.shipping_phone}</li>
    </ul>
    <h2>💰 תשלום</h2>
    <ul>
      <li><strong>שולם</strong> ₪${order.amount_paid}</li>
      <li><strong>קבלה</strong> <a href="${docUrl}">צפיה</a></li>
    </ul>
    <p><a href="${Deno.env.get('APP_URL')}/admin"
style="background:#954535;color:white;padding:10px 20px;text-decoration:none;border-
radius:5px;">פתח בדשבורד</a></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 (חילוף 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 'שילב';
  if (url.includes('motsesim.co.il')) return 'מוצעים';
  if (url.includes('baby-star.co.il')) return 'בייבי-סטאר';
  if (url.includes('super-pharm.co.il')) return 'סופר-פארם';
  if (url.includes('ksp.co.il')) return 'KSP';
  if (url.includes('bug.co.il')) return 'באג';
  if (url.includes('ikea.co.il')) return 'IKEA';
  return 'אחר';
}

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 יום: Setup + Auth

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

✓ Deliverable: משתמש יכול להירשם ולהתחבר

2 יום: Profile + Registry + Items (Manual)

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

✓ Deliverable: אפשר ליצור רשימה ולהוסיף פריטים ידנית

3 יום: URL Scraper

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

✓ Deliverable: עובד URL → Auto-fill הדבקת

4 יום: Guest View

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

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

✓ **Deliverable:** אורחים רואים רשימה ומתחילים קנייה

5 יום: Green Invoice Payment

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

✓ **Deliverable:** תשלום עובד מקצה לקצה

6 יום: 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	webhook-חבר שליחה ב	Emails sending
6-7	בדוק קבלת כל 3 אימיילים	Emails received
7-8	Admin Dashboard - list בנה	Admin list
8-9	status הוסף כפתורי	Status buttons
9-10	הוסף שליחת אימייל משלוח	Shipped email works

✓ **Deliverable:** Admin Dashboard + כל האימיילים עובדים

7 יום: Sharing + Polish

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

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

✓ **Deliverable:** מלוטש UI + שיתוף עובד

8 יום: Test + Deploy 🚀

שעה	משימה	פלט
1-2	15 test cases כתוב	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

9.1 הכנת הפרויקט

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

export default defineConfig({
  plugins: [react()],
  base: '/reshimat-tinok/', // ה 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=xxx
RESEND_API_KEY=xxx
ADMIN_EMAIL=your@email.com
APP_URL=https://username.github.io/reshimat-tinok
```

9.4 MVP - Custom Domain לאחר

משלך domain עובד, תוכל לחבר MVP-אחרי שה

1. domain קנה (reshimat-tinok.co.il למשל)
2. CNAME record הגדר username.github.io שמצביע ל
3. domain-עם ה public לתיקיית CNAME הוסף קובץ
4. Supabase secrets ב-APP_URL עדכן את
5. Green Invoice ב-successUrl/failureUrl עדכן את

10. בדיקות QA

10.1 Test Cases

#	תרחיש	תוצאה צפויה
1	הרשמה עם אימייל	חשבון נוצר
2	Google התחברות	עובר לדשבורד
3	השלמת פרופיל	נתונים נשמרים
4	יצירת רשימה	slug רשימה נוצרת עם
5	שילב URL הדבקת	מוצר נטען
6	מוצצים URL הדבקת	מוצר נטען
7	הוספה ידנית	פריט נשמר
8	עריכת פריט	שינויים נשמרים
9	מחיקת פריט	פריט נמחק
10	צפייה כאורח	רשימה מוצגת
11	פילטרים	מסננים עובדים
12	התחלת רכישה	מודל נפתח
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	Admin אימייל	מגיע תוך דקה
18	WhatsApp שיתוף	תקין Preview
19	Mobile responsive	נראה טוב
20	RTL	הכל בכיוון נכון

10.2 Security Checklist

- פועל על כל הטבלאות RLS
- של אחרים orders-אי אפשר לגשת ל
- API keys ב client-לא חשופים
- (signature: אופציונלי) מאומת Webhook
- בכל השדות Input validation

11-MVP לאחר ה.

11.1 Stabilization: שבוע 2

- feedback לפי bugs תקן
- custom domain חבר
- analytics (Plausible) הוסף
- error tracking (Sentry) הוסף

11.2 שיפורים: 1 חודש

- scrapers הוסף לחנויות נוספות
- Admin Dashboard-שפר את ה
- Welcome הוסף אימייל אוטומטי
- הוסף חיפוש רשימה

11.3 פיצ'רים חדשים: 2 חודש

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

11.4 Scale: 3 חודש

- אם יש 100+ הזמנות - שקול אוטומציה
- Premium tier (אופציונלי)
- שיווק בקבוצות פייסבוק
- שותפויות עם בלוגריות

Quick Reference: נוסף

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. חנויות נתמכות

חנות	Domain	עדיפות
שילב	shilav.co.il	P0
מוצצים	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

gear = ציוד (עגלות, כיסאות בטיחות)

clothing = הלבשה

feeding = האכלה (בקבוקים, משאבות)

health = בריאות (מדי חום, מוניטורים)

nursery = חדר תינוק (עריסה, שידה)

other = אחר

— סוף המסמך —
!בהצלחה 🍼

Green Invoice עם MVP-ימים ל 8