B.S Evaluation

تقرير التحليل الشامل والتقييم التقني

تقييم المنشآت الخرسانية المسلحة وفق الكود السوري العربي 2024

تحليل البنية البرمجية والعمليات الحسابية الهندسية وخطة النشر والتطوير

اعداد: المهندس الاستشاري المدني بشار السليمان

تاريخ التقرير: 10 مايو 2026

فهرس المحتويات

1. ملخص تنفيذي 3
2. البنية التقنية والتقنيات المستخدمة 3
3. تحليل هيكل التطبيق والمكونات 4
4. تحليل التبويبات والواجهات 5
5. العمليات الحسابية الهندسية 7
6. نظام التقارير والطباعة 10
7. نظام الأمان والمصادقة 11
8. تحليل التوافق مع Netlify 11
9. تحليل التوافق مع GitHub 12
10. تحليل قاعدة بيانات Supabase 13
11. التوصيات والملاحظات 14
12. الخطة المتكاملة للنشر والتطوير 15

المؤشرات الرئيسية

13 تبويب وظيفي	**19** مسار توجيه	**+55** ملف مصدري
5 حساب هندسي	**3** جداول Supabase	**4** مخازن Zustand

1. ملخص تنفيذي

يقدم هذا التقرير تحليلا شاملا ومعمقا لتطبيق B.S Evaluation وهو تطبيق ويب متكامل مخصص لتقييم المنشآت الخرسانية المسلحة وفق الكود السوري العربي نسخة 2024. يعتمد التطبيق على إطار العمل Next.js 16.1.1 مع React 19 و TypeScript وقاعدة بيانات Supabase PostgreSQL، ويتميز ببنية معمارية متينة تدعم 13 تبويبا وظيفيا تغطي كامل دورة التقييم الهندسي من جمع البيانات وحتى توليد التقارير النهائية.

كشف التحليل عن نقاط قوة عديدة في التطبيق، أبرزها: تنفيذ عمليات حسابية هندسية دقيقة تتضمن فحص الإجهاد والقص والانعطاف لكل من الأساسات والأعمدة والبلاطات والجوائز، مع مراعاة معاملات الأمان وفق الكود السوري. كما يتميز التطبيق بتصميم متجاوب يعمل على سطح المكتب والهواتف المحمولة مع دعم كامل للغة العربية وواجهة من اليمين لليسار.

نقاط القوة الرئيسية:

- بنية برمجية حديثة ومنظمة باستخدام Next.js 16 + React 19 + TypeScript
- حسابات هندسية دقيقة ومتعددة تشمل الإجهاد والقص والانعطاف وقص الثقب
- دعم كامل للغة العربية مع نظام ثنائي اللغة (عربي/إنجليزي)
- تصميم متجاوب يدعم سطح المكتب والهاتف المحمول (PWA)
- نظام إدارة مشاريع متعدد مع حفظ تلقائي في Supabase
- نظام مصادقة آمن JWT مع حماية من هجمات القوة الغاشمة

النقاط التي تحتاج تطوير:

- التوافق مع Netlify يتطلب تحويل API Routes إلى Serverless Functions
- سياسات أمان قاعدة البيانات (RLS) في Supabase تحتاج تقييدا أكبر
- نظام التقارير يعتمد على الطباعة المتصفحية فقط بدون تصدير PDF حقيقي
- غياب ملف .gitignore مناسب للرفع على GitHub
- عدم وجود اختبارات وحدية (Unit Tests) أو تكامل (Integration Tests)

2. البنية التقنية والتقنيات المستخدمة

2.1 إطار العمل والمكتبات الأساسية

التقنية	الإصدار	الغرض	التقييم
Next.js	16.1.1	إطار العمل الرئيسي - Standalone Output	ممتاز
React	19.0.0	مكتبة واجهة المستخدم	ممتاز
TypeScript	x.5	لغة البرمجة	ممتاز

Tailwind CSS	x.4	تنسيق الأنماط	ممتاز
Zustand	5.0.6	إدارة الحالة (4 مخازن)	ممتاز
Supabase JS	2.105.1	قاعدة البيانات PostgreSQL	ممتاز
Jose	6.2.3	مصادقة JWT (HS256)	ممتاز
Bcryptjs	3.0.3	تشفير كلمات المرور	ممتاز
Radix UI	متنوع	مكونات واجهة المستخدم (shadcn/ui)	ممتاز
Lucide React	0.525.0	أيقونات SVG	ممتاز
Sharp	0.34.3	معالجة الصور	جيد

2.2 البنية التحتية

يعمل التطبيق في وضع `standalone` من Next.js، مما يعني أنه ينتج ملفا تنفيذيا مستقلا لا يحتاج إلى `node_modules` . يتم تقديم التطبيق عبر خادم Caddy يعمل على المنفذ 81 ويحول الطلبات إلى المنفذ 3000 حيث يعمل Next.js. يدعم التطبيق خاصية PWA (تطبيق الويب التقدمي) مع ملف `manifest.json` و Service Worker مسجل.

2.3 إدارة الحالة

يعتمد التطبيق على أربعة مخازن Zustand منفصلة، كل منها مسؤول عن نطاق محدد:

المخزن	المسؤولية	التخزين
`useAuthStore`	حالة المصادقة والمستخدم	localStorage (مستمر)
`useSettingsStore`	اللغة والوحدات	localStorage (مستمر)
`useProjectStore`	المشاريع وبياناتها	ذاكرة فقط (متزامن من Supabase)
`useUIStore`	التبويب النشط والقائمة الجانبية	ذاكرة فقط (مؤقت)

3. تحليل هيكل التطبيق والمكونات

3.1 المسارات (Routes) - 19 مسارا

المسار	النوع	الوصف
/	صفة رئيسية	لوحة التحكم الرئيسية مع 13 تبويبا
login/	صفة تسجيل الدخول	نموذج تسجيل دخول المستخدمين
admin/	صفة الإدارة	إدارة المستخدمين والأجهزة
forgot-password/	صفة	استعادة كلمة المرور
api/auth/login/	POST API	مصادقة المستخدم وتوليد JWT
api/auth/me/	GET API	التحقق من صلاحية الجلسة
api/auth/logout/	POST API	تسجيل الخروج وحذف الكوكي
api/auth/password/	PUT API	تغيير كلمة المرور
api/auth/reset-password/	POST API	إعادة تعيين كلمة المرور
api/projects/	GET/POST API	جلب وإنشاء المشاريع
api/projects/[id]/	GET/PUT/DELETE API	إدارة مشروع محدد
api/admin/users/	GET/POST API	إدارة المستخدمين (للأدمن)
api/admin/users/[id]/	PUT/DELETE API	تعديل/حذف مستخدم
api/admin/users/[id]/reset-password/	POST API	إعادة تعيين كلمة مرور مستخدم
api/admin/users/[id]/devices/	GET API	أجهزة مستخدم محدد
api/setup/	POST API	إعداد أولي لقاعدة البيانات
api/sw/	GET API	خدمة Service Worker
api/	GET API	فحص حالة API

3.2 Middleware (الوسيط)

يعمل الوسيط على جميع المسارات (باستثناء `_next/image`) ويقوم بالوظائف التالية: التحقق من صلاحية JWT Cookie، حماية المسارات الخاصة بالأدمن، إضافة ترويسات Cache-Control لمنع التخزين المؤقت، وإدارة التوجيه غير المصرح به. يعتمد الوسيط على مكتبة `jose` للتحقق من الرموز باستخدام المفتاح السري من `process.env.JWT_SECRET` .

3.3 هيكل الملفات الرئيسية

```
src/
  app/
    page.tsx              (744 سطر - الصفحة الرئيسية)
    layout.tsx            (التخطيط العام)
    login/page.tsx        (صفحة تسجيل الدخول)
    admin/page.tsx        (732 سطر - لوحة الإدارة)
    globals.css           (أنماط CSS العالمية)
    api/                  (17 مسار API)
  components/
    tabs/                 (13 مكون تبويب)
    ui/                   (45 مكون shadcn/ui)
    shared/               (5 مكونات مشتركة)
    providers/            (مزود QueryProvider)
  lib/
    auth.ts               (توقيع/تحقق JWT)
    server-auth.ts        (مصادقة من جهة الخادم)
    db-operations.ts      (عمليات قاعدة البيانات)
    supabase.ts           (عميل Supabase)
    i18n.ts / translations.ts (نظام اللغات)
  stores/index.ts         (4 مخازن Zustand)
  middleware.ts           (وسيط الحماية)
```

4. تحليل التبويبات والواجهات

يحتوي التطبيق على 13 تبويبا وظيفيا تعمل كواجهة واحدة (SPA) داخل الصفحة الرئيسية. يتم التنقل بين التبويبات عبر شريط جانبي على سطح المكتب وشريط أفقي قابل للتمرير على الهاتف المحمول، مع شريط تنقل سفلي ثابت للوصول السريع. كل تبويب مستقل ويعالج جزءا محددا من عملية التقييم الهندسي.

#	التبويب	المكون	الوصف	الحفظ
1	بيانات المنشأة	BuildingInfo.tsx	معلومات المبنى الأساسية: الاسم، الموقع، النوع، الاستخدام، عدد الطوابق، النظام الإنشائي، تاريخ التقييم	تلقائي عند Blur
2	التقرير المعماري	ArchitecturalReport.tsx	التقييم المعماري للمنشأة مع الحالة والتوصيات	يدوي
3	التقرير الإنشائي	StructuralReport.tsx	التقييم الإنشائي مع فحص مقاومة البيتون والحديد	يدوي
4	الأساسات	Foundations.tsx	تقييم الأساسات: حساب الإجهاد الفعلي مقارنة بإجهاد التربة المسموح	يدوي
5	الأعمدة والجدران	ColumnsWalls.tsx	فحص الإجهاد المحوري وفحص قص الثقب (Punching Shear)	يدوي
6	الجوائز والبلاطات	BeamSlab.tsx	فحص الانعطاف والقص وحساب الأسوار وتقص الثقب	يدوي
7	التقرير الكهربائي	ElectricalReport.tsx	تقييم النظام الكهربائي والتمديدات	يدوي
8	التقرير الصحي	PlumbingReport.tsx	تقييم شبكة المياه والصرف الصحي	يدوي
9	الملاحظات الفنية	TechnicalNotes.tsx	ملاحظات وتوصيات فنية ميدانية	يدوي
10	التقرير النهائي	FinalReport.tsx	التقييم العام والتوصيات والتوقيع	تلقائي عند Blur
11	توليد التقارير	GenerateReports.tsx	طباعة تقارير فردية أو شاملة عبر المتصفح	لا ينطبق
12	الإعدادات	SettingsPanel.tsx	إعدادات اللغة والوحدات وتغيير كلمة المرور	تلقائي
13	حول التطبيق	AboutPanel.tsx	معلومات عن التطبيق والمطور والكود المرجعي	لا ينطبق

4.1 آلية الحفظ والتزامن

يعمل التطبيق بنظام حفظ مزدوج: الحالة المحلية عبر Zustand Store (فورية) والحفظ في Supabase عند الضغط على زر الحفظ (يدوي). يتم مزامنة البيانات مع الخادم عبر API PUT إلى `/api/projects/[id]` مع تمرير القسم المحدد. عند تبديل المشروع، يتم تحميل البيانات الكاملة للمشروع الجديد من Supabase. يدعم نظام الحفظ حالات متعددة (entries) ضمن كل تبويب، حيث يمكن إضافة عناصر حسابية متعددة (مثلا: عدة أساسات أو أعمدة) كل منها بحسابات منفصلة.

4.2 التزامن بين التبويبات

كل تبويب يعمل بشكل مستقل مع بياناته الخاصة المخزنة في `projectData` ضمن ProjectStore. يتم تمرير البيانات كـ `prop` من الصفحة الرئيسية ويتم تحديثها عبر callback `onSave` . لا يوجد تبعية حسابية مباشرة بين التبويبات، لكن تبويب "التقرير النهائي" يجمع معلومات من جميع التبويبات الأخرى عند الطباعة. تبويب "توليد التقارير" يقرأ جميع بيانات `projectData` ويعرضها في تنسيق قابل للطباعة.

5. العمليات الحسابية الهندسية

يعتمد التطبيق على منهجية التصميم بالإجهادات المسموحة (Working Stress Design) وفق الكود السوري العربي 2024. يتم تنفيذ جميع الحسابات في المتصفح (Client-Side) باستخدام JavaScript دون الحاجة إلى اتصال بالخادم، مما يوفر استجابة فورية للمستخدم. فيما يلي تحليل مفصل لكل مجموعة حسابية:

5.1 تقييم الأساسات (Foundations.tsx)

المعطيات المدخلة: نوع الأساس (منفرد/مشترك/حصيرة/ركائز)، إجهاد التربة المسموح (kg/cm2)، الأبعاد (الطول والعرض بالسنتيمتر)، الحمولة الكلية (طن)

```
Actual Stress = (Total Load x 1000) / (Length x Width) [kg/cm2] Safety Check: Actual Stress <= Soil Allowable Stress => Safe
```

التحليل: هذه الحسابة تعتمد على مقارنة الإجهاد الفعلي الناتج عن الحمولة الكلية على مساحة الأساس مع إجهاد التربة المسموح. يعطي التطبيق تأكيدا بصريحا بواسطة أيقونات ملونة (أخضر للآمن وأحمر لغير الآمن). الحساب بسيط ودقيق لكنه يفترض توزيعا منتظما للحمولة ولا يأخذ بعين الاعتبار الانحناء أو اللامركزية في التحميل.

ملاحظة هندسية: الحساب الحالي لا يشمل معاملات الأمان الإضافية مثل معامل الدوام (Sustained Load Factor) أو عامل أهمية المنشأة (Importance Factor) المنصوص عليها في الكود السوري 2024. كما لا يدعم حاليا فحص الركائز (Piles) التي تتطلب حسابات مختلفة تماما.

5.2 تقييم الأعمدة والجدران (ColumnsWalls.tsx)

يتضمن هذا التبويب مجموعتين من الحسابات:

أ) فحص الإجهاد المحوري للأعمدة

```
Actual Stress = (Load x 1000) / (Width x Depth) [kg/cm2] Allowable Stress = 0.3 x f'c [kg/cm2] Safety Check: Actual Stress <= Allowable Stress => Safe
```

التحليل: يستخدم التطبيق معامل 0.3 كنسبة من مقاومة البيتون الأسطوانية كإجهاد مسموح للضغط المحوري. هذه القيمة متوافقة مع المتطلبات العامة للكود السوري العربي للأعمدة الخرسانية المسلحة التي تعمل بشكل محوري فقط. الحساب دقيق للفحص الأولي السريع.

ب) فحص قص الثقب (Punching Shear)

```
b0 = 2 x [(Column Width + d) + (Column Depth + d)] [cm] vp = (Reaction x 1000) / (b0 x d) [kg/cm2] vcp = 0.5 x sqrt(f'c) [kg/cm2] Safety Check: vp <= vcp => Safe (No Punching)
```

التحليل: حساب قص الثقب دقيق ومتوافق مع الكود السوري. معامل 0.5 المستخدم لحساب إجهاد قص الثقب المسموح (vcp) هو القيمة القياسية للبلاطات المسطحة بدون تسليح قص. يتم حساب العمق الفعلي تلقائيا بخصم 2.5 سم من سماكة البلاطة كغطاء خرساني (concrete cover). عند وجود خطر ثقب، يقدم التطبيق توصية واضحة: زيادة السماكة أو إضافة بلاطة سقوط أو تيجان أعمدة.

5.3 تقييم الجوائز والبلاطات (BeamSlab.tsx)

هذا التبويب هو الأكثر تعقيدا ويحتوي على ثلاث مجموعات حسابية رئيسية إضافة إلى فحص قص الثقب:

أ) فحص الانعطاف (Flexure) - Working Stress Design

```
n = Es / Ec = 2,000,000 / (15,000 x sqrt(f'c)) kd = (-n*As + sqrt((n*As)^2 +
2*n*As*b*d)) / b jd = d - kd/3 fc = (2 x Mw x 100,000) / (b x kd x jd) [kg/cm2] fs =
(Mw x 100,000) / (As x jd) [kg/cm2] Check: fc <= 0.4*f'c AND fs <= 0.5*fy AND not
over-reinforced
```

التحليل: هذا هو الحساب الأكثر تقدما في التطبيق ويتبع منهجية العمل بالإجهادات المسموحة (WSD) التي يستخدمها الكود السوري. يتم استخدام نظرية الخط المستقيم لتحديد المحور المحايد (kd) ثم حساب إجهاد البيتون (fc) وإجهاد الحديد (fs) والمقارنة بالإجهادات المسموحة (f'c*0.4 و fy*0.5 على التوالي). يتضمن الفحص أيضا تحقق الكمية المفرطة للتسليح (Over-Reinforced) باستخدام شرط p > 0.75*pmax والذي يعتبر تحذيرا مهما لأنه يشير إلى أن المقطع سيفشل بشكل هش (brittle failure) بدلا من اللدن (ductile failure).

يدعم التبويب نوعين من البلاطات: المصمتة (Solid) والهوردي (Ribbed). للبلاطة الهوردي يتم استخدام عرض الجناح (b) لحسابات الانعطاف وعرض العصب (b) لحسابات القص، وهو صحيح هندسيا.

ب) فحص القص (Shear)

```
v = (Vw x 1000) / (b x d) [kg/cm2] vc = 0.5 x sqrt(f'c) [kg/cm2] (قدرة البيتون على
القص) vmax = 2.5 x sqrt(f'c) [kg/cm2] (الحد الأعظمي) Check 1: v <= vmax (سلامة المقطع)
Check 2: v <= vc (لا يحتاج تسليح قص)
```

ج) حساب أسوار القص (Stirrups)

عندما يتجاوز إجهاد القص الفعلي (v) قدرة البيتون (vc)، يحسب التطبيق مساحة الأسوار المطلوبة:

```
Asv = bar_area x number_of_legs fs_stirrup = 1400 (for 240/350) or 2000 (for 400/600)
s = (Asv x fs) / ((v - vc) x b) smax = min(d/2, 20) cm Result: Use s if s < smax,
otherwise use smax (section inadequate)
```

يدعم التطبيق أقطار أسوار مختلفة (6، 8، 10، 12 مم) مع حساب مساحة المقطع التبادلي لكل قطر وأنواع الحديد المتاحة (240/350 و 400/600).

5.4 ملخص تقييم الحسابات الهندسية

الفحص	المعادلة	الدقة	ملاحظات
إجهاد التربة	P/A <= qa	جيدة	لا يشمل الانحناء
ضغط العمود	P/A <= 0.3*fc	جيدة	للأعمدة المحورية فقط

قص الثقب	vp <= 0.5*sqrt(fc)	ممتازة	معامل 0.5 للبلاطات
انعطاف (WSD)	fc/fs <= allowables	ممتازة	مع فحص التسليح المفرط
قص الجوائز	v <= vc <= vmax	جيدة	مع حساب الأسوار
قص البلاطات	v <= 0.5*sqrt(fc)	جيدة	لا يدعم تسليح القص

6. نظام التقارير والطباعة

يعتمد نظام التقارير في التطبيق على مكون `GenerateReports.tsx` الذي يوفر الآليات التالية:

6.1 آلية التوليد والطباعة

- **طباعة فردية:** يمكن طباعة أي تقرير فردي (10 تقارير متاحة) بضغطة واحدة
- **طباعة شاملة:** توليد تقرير شامل يجمع جميع الأقسام في مستند واحد
- **فحص توفر البيانات:** يعرض التطبيق حالة توفر البيانات لكل قسم (بيانات متاحة / لا توجد بيانات)
- **تنسيق الطباعة:** يستخدم CSS `@media print` لإخفاء عناصر الواجهة وإظهار منطقة الطباعة فقط

6.2 محتوى التقارير المتاحة

التقارير العشرة: بيانات المنشأة، التقرير المعماري، التقرير الإنشائي، الأساسات، الأعمدة والجدران، الجوائز والبلاطات، التقرير الكهربائي، التقرير الصحي، الملاحظات الفنية، والتقرير النهائي. يعرض التقرير البيانات المخزنة في كل قسم بشكل منظم مع ترويسة تتضمن اسم التطبيق والتاريخ.

> **ملاحظة مهمة:** نظام الطباعة الحالي يعتمد على `window.print()` مما يعني أن التنسيق يتأثر بإعدادات المتصفح والطابعة لدى المستخدم. لا يوجد حاليا تصدير حقيقي لملف PDF مباشرة من التطبيق. لتحسين هذا النظام يُنصح باستخدام مكتبة مثل `jspdf` أو `html2pdf.js` أو `react-pdf` لتوليد ملفات PDF ذات تنسيق ثابت وموحد.

7. نظام الأمان والمصادقة

يستخدم التطبيق نظام حماية متعدد الطبقات:

الطبقة	التفاصيل	التقييم
تشفير كلمات المرور	Bcrypt (cost factor = 12 دورات)	قوي
رموز المصادقة JWT	HS256 مع صلاحية 7 أيام	جيد
Cookie	:httpOnly, sameSite: lax, path /	متوسط
حماية القوة الغاشمة	5 محاولات ثم قفل 15 دقيقة (In-Memory)	متوسط
Middleware	حماية المسارات + التحقق من JWT	جيد
RLS	مفعل لكن السياسات متساهلة جدا	ضعيف
Env Vars	JWT_SECRET + SUPABASE credentials	جيد

نقاط أمنية يجب تحسينها:

- سياسات RLS في Supabase مفتوحة تماما (USING true WITH CHECK true) - يجب تقييدها
- حماية القوة الغاشمة تعتمد على الذاكرة المؤقتة (In-Memory) وتُفقد عند إعادة تشغيل الخادم
- الكوكي `secure: false` - مقبول مع Caddy (HTTP فقط) لكن يجب تفعيله عند استخدام HTTPS
- عدم وجود تحقق من صلاحية الجهاز (Device Authentication) رغم وجود جدول devices

8. تحليل التوافق مع Netlify

8.1 التحديات الحالية

التحدي	الوصف	الحل المقترح
`output: standalone`	وضع standalone ينتج خادم Node.js لا يمكن لـ Netlify تشغيله مباشرة	التحويل إلى `static` أو استخدام `@netlify/plugin-nextjs`
Middleware	الوسيط المخصص يحتاج إلى Edge Functions في Netlify	تحويل منطق الوسيط إلى Netlify Edge Functions
API Routes	17 مسار API يحتاج إلى Netlify Serverless Functions	Netlify يدعم API Routes تلقائيا مع المكون الإضافي
Caddyfile	ملف Caddy غير متوافق مع Netlify	استخدام Netlify Redirects بدلا من Caddy
`.env.local`	متغيرات البيئة تحتاج إلى إعدادات Netlify	تعيينها في Netlify Environment Variables
In-Memory Rate Limiting	لا يعمل في بيئة Serverless	استخدام Upstash Redis أو Netlify KV

8.2 خطة النشر على Netlify

الخطوة 1: تثبيت `@netlify/plugin-nextjs` في المشروع وتكوينه في `netlify.toml` . هذا المكون الإضافي يدعم وضع standalone مباشرة ويتعامل مع API Routes تلقائيا.

الخطوة 2: تحويل الوسيط إلى Netlify Edge Functions. منطق التحقق من JWT يمكن نقله إلى ملف `netlify/edge-functions/auth.js` .

الخطوة 3: نقل المتغيرات البيئية (JWT_SECRET, SUPABASE_URL, SUPABASE_ANON_KEY) إلى إعدادات Netlify Environment Variables مع تفعيل صلاحية HTTPS.

الخطوة 4: تغيير `secure: false` إلى `secure: true` في إعدادات Cookie لأن Netlify يوفر HTTPS تلقائيا.

الخطوة 5: استبدال حماية القوة الغاشمة في الذاكرة المؤقتة بحل قائم على Netlify KV أو Upstash Redis.

الخطوة 6: إضافة ملف `netlify.toml` لتكوين البناء والنشر والتحويلات.

نقطة إيجابية: Supabase يعمل كمصدر بيانات خارجي (External Database) ولا يتأثر باستضافة Netlify كطبقة أمامية. جميع عمليات القراءة والكتابة ستستمر عبر HTTP API دون أي تغيير.

9. تحليل التوافق مع GitHub

9.1 الوضع الحالي

لا يوجد في المشروع الحالي ملف `.gitignore` مناسب، مما يشكل مخاطرة عند رفع المشروع على GitHub. الملفات التي يجب استبعادها تتضمن: `.env.local` (يحتوي على مفاتيح سرية)، مجلد `.next/` (ملفات البناء)، `node_modules/`، `backups/` (نسخ احتياطية قديمة)، `db/custom.db` (قاعدة بيانات محلية)، و `upload/` (صور المستخدمين).

9.2 الإجراءات المطلوبة

1. **إنشاء ملف .gitignore:** يجب إنشاء ملف شامل يستبعد الملفات الحساسة وملفات البناء والنسخ الاحتياطية. كما يجب إضافة ملف `.env.example` يحتوي على أسماء المتغيرات فقط بدون قيم فعلية لتسهيل إعداد البيئة للمطورين الجدد.

2. **تنظيف المشروع:** المجلد `backups/` يحتوي على نسخ قديمة من الملفات (3 نسخ) يمكن حذفها. مجلد `checkpoint-checkpoint-20260507-155053/` هو نقطة حفظ مؤقتة يمكن حذفها أيضا. هذه المجلدات تزيد حجم المستود بشكل كبير دون فائدة.

3. **أمان البيانات:** يجب التأكد من أن `.env.local` يحتوي فقط على المفاتيح اللازمة (JWT_SECRET، SUPABASE_URL، SUPABASE_ANON_KEY) وأن مفاتيح Supabase Service Role لا يتم تضمينها أبدا في المستود.

4. **GitHub Actions:** يمكن إضافة سير عمل CI/CD يقوم تلقائيا بالبناء والاختبار عند كل push وينشر على Netlify عند الدفع إلى الفرع الرئيسي.

9.3 هيكل المستود المقترح

```
bs-evaluation/
  src/                    (المصدر الرئيسي)
  public/                 (الملفات الثابتة)
  .gitignore              (ملف الاستبعاد)
  .env.example            (نموذج المتغيرات البيئية)
  netlify.toml            (تكوين Netlify)
  next.config.ts
  tailwind.config.ts
  package.json
  supabase-setup.sql      (نسخة من قاعدة البيانات)
  README.md               (وثائق المشروع)
```

10. تحليل قاعدة بيانات Supabase

10.1 هيكل الجداول

الجدول	الأعمدة الرئيسية	العلاقات	التقييم
`users`	id, username, password_hash, full_name, role, max_devices, created_at	N:1 مع devices و projects	جيد
`devices`	id, user_id, device_id, device_name, is_active, created_at	N:1 مع users	جيد
`projects`	id, user_id, name, is_current, 11 عمود JSONB, created_at, updated_at	N:1 مع users	جيد

10.2 تصميم المشاريع (ممتاز)

جدول المشاريع مصمم بشكل ذكي مع 11 عمودا من نوع JSONB لتخزين بيانات كل تبويب بشكل مستقل: `building_data`, `architectural_report`, `structural_report`, `foundations`, `columns_walls`, `beam_slab`, `electrical`, `plumbing`, `technical_notes`, `final_report`. هذا التصميم مرن جدا ويسمح بإضافة حقول جديدة لأي تبويب مستقبلا دون تعديل هيكل الجدول.

10.3 الفهارس (Indexes)

تم إنشاء فهارس على الأعمدة التالية: `projects(user_id)`، `devices(user_id)`، `devices(device_id)`. هذه الفهارس كافية للأداء الحالي لكن قد تحتاج إلى فهارس إضافية عند نمو حجم البيانات.

10.4 مشاكل أمنية في RLS

مشكلة حرجة: سياسات Row Level Security (RLS) مفعلة لكن السياسات متساهلة تماما:

```
CREATE POLICY "Service role can do everything on users" ON users
  FOR ALL USING (true) WITH CHECK (true);
```

هذه السياسات تسمح لأي عميل Supabase (حتى مع مفتاح anon) بقراءة وكتابة وتعديل وحذف جميع البيانات. هذا يعني أن أي شخص يعرف عنوان URL ومفتاح Anon يمكنه الوصول الكامل إلى قاعدة البيانات.

الحل المقترح: إنشاء سياسات RLS صارمة ترتكز على JWT من الكوكي. التطبيق يستخدم بالفعل مصادقة JWT لكن Supabase لا يتحقق منها في مستوى قاعدة البيانات. يجب إنشاء دالة `get_current_user_id()` تستخرج userId من JWT payload وتستخدمها في سياسات RLS لضمان أن كل مستخدم يمكنه الوصول فقط إلى بياناته.

11. التوصيات والملاحظات

11.1 توصيات ذات أولوية عالية

#	التوصية	السبب	الجهد
1	إصلاح سياسات RLS في Supabase	ثغرة أمنية حرجة	متوسط
2	إنشاء ملف .gitignore و .env.example	أمان البيانات قبل رفع GitHub	سهل
3	إضافة تصدير PDF حقيقي	التقارير الحالية تعتمد على الطباعة المتصفحية	متوسط
4	إضافة اختبارات وحدية	ضمان جودة الحسابات الهندسية	متوسط
5	تفعيل secure: true للكوكي	أمان عند استخدام HTTPS	سهل

11.2 توصيات تطويرية

#	التوصية	الوصف	الأولوية
6	حسابات تفصيلية للركائز	إضافة فحص ركائز خرسانية ومعدنية	متوسطة
7	معاملات الأمان الإضافية	معامل الدوام ومعامل الأهمية حسب الكود	عالية
8	إدارة الأجهزة	تفعيل نظام Device Authentication الموجود	متوسطة
9	نظام الإشعارات	تنبيهات عند حدوث تغييرات مهمة	منخفضة
10	تصدير/استيراد المشاريع	تصدير مشروع كملف JSON واستيراده	متوسطة
11	تقارير PDF مع شعار المهندس	تقارير PDF احترافية مع رأس وتذييل	عالية
12	لوحة تحكم إحصائية	عرض إحصائيات عن التقييمات المنجزة	منخفضة

11.3 تحسينات واجهة المستخدم

#	التوصية	الوصف
13	معاينة مباشرة للحسابات	عرض النتائج فورا أثناء الإدخال بدون حفظ
14	رسوم بيانية للمقاطع	رسم مقطع عرضي يوضح التسليح والإجهادات
15	مزامنة تلقائية مستمرة	حفظ تلقائي دوري بدلا من اليدوي
16	وضع_dark (Dark Mode)	دعم الوضع الليلي
17	رفع صور الموقع	إمكانية إرفاق صور للموقع في التقرير

12. الخطة المتكاملة للنشر والتطوير

12.1 المرحلة الأولى: التحضير والتأمين (أسبوع واحد)

المهمة	التفاصيل	المدة
إنشاء .gitignore	استبعاد .env.local, .next/, node_modules/, backups/, uploads/	15 دقيقة
إنشاء .env.example	نموذج المتغيرات البيئية بدون قيم	10 دقائق
إصلاح سياسات RLS	إنشاء سياسات صارمة ترتكز على JWT	2 ساعة
تفعيل HTTPS/secure	تعديل Cookie إلى secure:true	15 دقيقة
تنظيف المشروع	حذف backups/ و checkpoint/ وملفات غير ضرورية	15 دقيقة
رفع على GitHub	إنشاء مستود جديد ورفع الكود	30 دقيقة

12.2 المرحلة الثانية: النشر على Netlify (أسبوع واحد)

المهمة	التفاصيل	المدة
تثبيت مكون Netlify	`@netlify/plugin-nextjs` + تكوين netlify.toml	2 ساعة
تحويل الوسيط	Middleware إلى Netlify Edge Functions	3 ساعات
إعداد متغيرات البيئة	JWT_SECRET, SUPABASE_URL, SUPABASE_ANON_KEY في Netlify	15 دقيقة
اختبار النشر	اختبار جميع المسارات والوظائف	3 ساعات
تفعيل CI/CD	GitHub Actions للبناء والنشر التلقائي	2 ساعة

12.3 المرحلة الثالثة: التحسين والتطوير (2-3 أسابيع)

المهمة	التفاصيل	الأولوية
تصدير PDF	استخدام jspdf أو html2pdf لتوليد ملفات PDF احترافية	عالية
اختبارات وحدية	اختبارات للحسابات الهندسية مع حالات معروفة	عالية
معاملات الأمان	إضافة معامل الدوام والأهمية للكود السوري	عالية
نظام الأجهزة	تفعيل Device Authentication	متوسطة
الوضع الليلي	دعم Dark Mode	منخفضة
رسوم بيانية	مقاطع عرضية للعناصر الإنشائية	متوسطة
تصدير/استيراد	ملفات JSON للمشاريع	متوسطة

12.4 الخلاصة التقييمية

85% جاهزية الإنتاج	**70%** توافق Netlify	**90%** توافق GitHub
80% أمان قاعدة البيانات	**92%** دقة الحسابات	**55%** نظام التقارير