**שם התלמידה: ליליאן לייקין**

**תאריך: 11.7.2025**

**מרצה: קובי בידני**

**תוכנית בדיקות:**

**מסמך תכנון מערכת – Red Flag: Detecting Hidden Dangers in Teen Relationships**

# תוכנית בדיקות מערכת SafeGuard

## מערכת זיהוי מוקדם לבטיחות תלמידים

## 1. מטרות הבדיקות

### 1.1 מטרה ראשית

לוודא שמערכת SafeGuard עובדת כצפוי, מזהה בעיות ובאגים, ומבצעת בביצועים אופטימליים.

### 1.2 מטרות משניות

* **אמינות**: וידוא שהמערכת פועלת ביציבות גבוהה
* **אבטחה**: הגנה על מידע רגיש של תלמידים ומשתמשים
* **ביצועים**: זמני תגובה מהירים ויכולת טיפול בעומסים
* **שימושיות**: ממשק משתמש אינטואיטיבי וידידותי
* **תאימות**: תמיכה בדפדפנים שונים ומכשירים מובייל

## 2. סוגי בדיקות מפורטים

### 2.1 בדיקות פונקציונליות (Functional Testing)

#### 2.1.1 בדיקות תכונות עיקריות

**זיהוי וניתוח סיכונים:**

* בדיקת דיוק אלגוריתם זיהוי סיכונים
* וידוא שמירת היסטוריית הערכות
* בדיקת עדכון אוטומטי של דירוגי סיכון
* בדיקת טיפול בנתונים חסרים או שגויים

**ניהול התראות:**

* שליחת התראות לגורמים הרלוונטיים
* בדיקת עדיפויות התראות (דחוף, רגיל, נמוך)
* וידוא הגעת התראות במספר ערוצים (מייל, SMS, אפליקציה)
* בדיקת מנגנון אישור קבלת התראות

**ניהול משתמשים:**

* רישום משתמשים חדשים
* אימות זהות ובטיחות סיסמאות
* הגדרת הרשאות לפי תפקיד
* מחיקת/השבתת משתמשים

#### 2.1.2 בדיקות תהליכי עבודה

**תהליך דיווח על תלמיד:**

1. מורה מזין נתונים על תלמיד
2. מערכת מאמתת נתונים
3. חישוב רמת סיכון
4. שליחת התראה אם נדרש
5. עדכון מאגר הנתונים

**תהליך התערבות:**

1. זיהוי תלמיד בסיכון
2. יצירת תכנית התערבות
3. הקצאת גורמים מטפלים
4. מעקב אחר התקדמות
5. סיכום ודיווח

#### 2.1.3 בדיקות אינטגרציה

**חיבור בין מודולים:**

* תקשורת בין דשבורד מורים למנוע הניתוח
* סנכרון נתונים בין מאגרי הנתונים השונים
* אינטגרציה עם מערכות חיצוניות (מערכת בית הספר)

**API Testing:**

* בדיקת כל נקודות הקצה (endpoints)
* וידוא פורמט JSON נכון
* בדיקת קודי תגובה HTTP
* טיפול בשגיאות ובחריגים

### 2.2 בדיקות ביצועים (Performance Testing)

#### 2.2.1 בדיקות מהירות (Speed Testing)

**זמני תגובה:**

* טעינת דף הבית: מקסימום 2 שניות
* חיפוש תלמיד: מקסימום 1 שנייה
* יצירת דוח: מקסימום 5 שניות
* עדכון נתונים: מקסימום 3 שניות

**אופטימיזציה:**

* דחיסת קבצים (CSS, JS, תמונות)
* מטמון דפים סטטיים
* אופטימיזציה של שאילתות מסד נתונים
* CDN לתכנים סטטיים

#### 2.2.2 בדיקות עומס (Load Testing)

**עומס רגיל:**

* 100 משתמשים פעילים במקביל
* 1000 פעולות בדקה
* זמני תגובה קבועים
* שיעור שגיאות מתחת ל-1%

**עומס שיא:**

* 500 משתמשים במקביל
* 5000 פעולות בדקה
* זמני תגובה עד 5 שניות
* שיעור שגיאות מתחת ל-3%

#### 2.2.3 בדיקות לחץ (Stress Testing)

**עומס קיצוני:**

* 1000+ משתמשים במקביל
* זיהוי נקודת קריסה
* בדיקת התאוששות המערכת
* וידוא שלמות נתונים תחת לחץ

**תרחישי קיצון:**

* שליחת אלפי התראות במקביל
* יצירת מאות דוחות בו-זמנית
* עדכון מסיבי של נתונים
* כשל בשרת מסד נתונים

#### 2.2.4 בדיקות סקלביליות

**הרחבת המערכת:**

* הוספת שרתים נוספים
* מנגנון load balancing
* חלוקת עומס בין מסדי נתונים
* תכנון לגידול עתידי

### 2.3 בדיקות משתמש (User Testing)

#### 2.3.1 בדיקות שימושיות (Usability Testing)

**נגישות הממשק:**

* פשטות ניווט במערכת
* בהירות הוראות ומידע
* קלות למידה למשתמשים חדשים
* יעילות ביצוע משימות

**חוויית משתמש:**

* עיצוב אינטואיטיבי
* זמני תגובה מתקבלים על הדעת
* הודעות שגיאה ברורות
* עזרה והדרכה מובנית

#### 2.3.2 בדיקות קבלה (Acceptance Testing)

**בדיקות עם משתמשים אמיתיים:**

* מורים מבתי ספר שונים
* הורים ברמות טכנולוגיות שונות
* עובדים סוציאליים
* מנהלי בתי ספר

**תרחישי שימוש אמיתיים:**

* דיווח על תלמיד בסיכון
* מעקב אחר התקדמות
* יצירת דוחות תקופתיים
* תקשורת בין הגורמים השונים

#### 2.3.3 בדיקות נגישות (Accessibility Testing)

**תמיכה בקבוצות מיוחדות:**

* תמיכה בקוראי מסך
* ניווט במקלדת בלבד
* ניגודיות צבעים מתאימה
* גדלי פונט משתנים

**תקני נגישות:**

* עמידה בתקן WCAG 2.1 רמה AA
* תמיכה בטכנולוגיות מסייעות
* הוראות קוליות
* תמיכה בשפות שונות

#### 2.3.4 בדיקות אמינות (Reliability Testing)

**יציבות המערכת:**

* פעולה רציפה למשך 24 שעות
* התאוששות מכשלים
* שמירת נתונים במצבי חירום
* מנגנוני גיבוי אוטומטיים

**אמינות הנתונים:**

* דיוק המידע המוצג
* עקביות בין מסכים שונים
* מניעת אובדן נתונים
* סנכרון נתונים בין משתמשים

### 2.4 מתודולוגיות בדיקה

#### 2.4.1 בדיקות חיוביות (Positive Testing)

* בדיקת פעולות עם נתונים תקינים
* וידוא שהמערכת מבצעת את הפעולה הנדרשת
* מדידת זמני תגובה תקינים

#### 2.4.2 בדיקות שליליות (Negative Testing)

* הזנת נתונים שגויים או חסרים
* בדיקת טיפול בשגיאות
* וידוא הודעות שגיאה מתאימות

#### 2.4.3 בדיקות גבול (Boundary Testing)

* בדיקת ערכים מינימליים ומקסימליים
* טיפול בקלטים חריגים
* בדיקת גבולות זמן וזיכרון

### 2.5 כלי בדיקה ייעודיים

#### 2.5.1 בדיקות פונקציונליות

* **Selenium WebDriver** - אוטומציה דפדפן
* **Cypress** - בדיקות end-to-end
* **Jest** - בדיקות יחידה JavaScript
* **Postman** - בדיקות API

#### 2.5.2 בדיקות ביצועים

* **JMeter** - בדיקות עומס
* **LoadRunner** - בדיקות ביצועים מתקדמות
* **k6** - בדיקות עומס מודרניות
* **GTmetrix** - ניתוח מהירות אתר

#### 2.5.3 בדיקות משתמש

* **Hotjar** - מעקב אחר התנהגות משתמשים
* **UserTesting** - בדיקות עם משתמשים אמיתיים
* **Maze** - בדיקות שימושיות
* **Axe** - בדיקות נגישות

## 3. אסטרטגיית הבדיקות

### 2.1 סוגי בדיקות

* **בדיקות יחידה (Unit Tests)** - 40%
* **בדיקות אינטגרציה (Integration Tests)** - 30%
* **בדיקות מערכת (System Tests)** - 20%
* **בדיקות קבלה (Acceptance Tests)** - 10%

### 2.2 רמות בדיקה

1. **רמת רכיב (Component Level)**
2. **רמת שירות (Service Level)**
3. **רמת מערכת (System Level)**
4. **רמת קצה לקצה (End-to-End)**

## 3. מטריקות והצלחה

### 3.1 מטריקות ביצועים

* **זמן תגובה**: מקסימום 2 שניות לפעולות רגילות
* **זמן טעינה**: מקסימום 3 שניות לעמוד ראשי
* **זמינות**: 99.9% זמינות מערכת
* **עמידות עומס**: 1000 משתמשים במקביל

### 3.2 מטריקות איכות

* **כיסוי קוד**: מינימום 85%
* **שיעור הצלחה**: 98% עבור תרחישי השימוש העיקריים
* **מספר באגים**: מקסימום 5 באגים קריטיים בייצור

## 4. תכנון בדיקות לפי שכבות

### 4.1 שכבת הממשק (Frontend Layer)

#### 4.1.1 Teacher Dashboard

**בדיקות פונקציונליות:**

* התחברות מורים למערכת
* הצגת רשימת תלמידים
* יצירת ועדכון אזהרות
* סינון וחיפוש תלמידים
* הורדת דוחות

**בדיקות UI/UX:**

* עיצוב רספונסיבי
* נגישות (WCAG 2.1)
* תמיכה בדפדפנים שונים
* מהירות טעינה

#### 4.1.2 Parent Portal

**בדיקות פונקציונליות:**

* התחברות הורים
* הצגת מידע על הילד
* קבלת התראות
* **תקשורת עם מורים**

#### 4.1.3 Social Worker Interface

**בדיקות פונקציונליות:**

* גישה למקרים מורכבים
* יצירת תכניות התערבות
* מעקב אחר התקדמות
* תיעוד פגישות

#### 4.1.4 Admin Dashboard

**בדיקות פונקציונליות:**

* ניהול משתמשים
* הגדרות מערכת
* דוחות ניהול
* ביקורת פעילות

### 4.2 שער API (API Gateway)

#### 4.2.1 Authentication

**בדיקות אבטחה:**

* אימות נתונים חזק
* הגנה מפני התקפות brute force
* ניהול sessions
* logout אוטומטי

#### 4.2.2 Authorization

**בדיקות הרשאות:**

* בדיקת הרשאות לפי תפקיד
* מניעת גישה לא מורשית
* הגנה על נתונים רגישים

#### 4.2.3 Rate Limiting

**בדיקות עומס:**

* הגבלת בקשות לשנייה
* טיפול בעומס יתר
* הגנה מפני DoS

### 4.3 שכבת השירותים (Microservices)

#### 4.3.1 Risk Assessment

**בדיקות אלגוריתם:**

* דיוק זיהוי סיכונים
* זמני תגובה
* טיפול בנתונים חסרים
* עדכון דירוגים

#### 4.3.2 Behavior Analysis

**בדיקות AI/ML:**

* דיוק ניתוח התנהגות
* למידה מנתונים חדשים
* זיהוי דפוסים חריגים
* מניעת false positives

#### 4.3.3 Alert Management

**בדיקות התראות:**

* שליחת התראות בזמן אמת
* פרטי קבלה
* סינון התראות
* היסטוריית התראות

#### 4.3.4 Report Generation

**בדיקות דוחות:**

* יצירת דוחות מדויקים
* פורמטים שונים (PDF, Excel)
* מהירות יצירה
* איכות הנתונים

### 4.4 שכבת הנתונים (Data Layer)

#### 4.4.1 PostgreSQL

**בדיקות בסיס נתונים:**

* שלמות נתונים
* ביצועי שאילתות
* גיבוי ושחזור
* אבטחת נתונים

#### 4.4.2 Redis

**בדיקות מטמון:**

* מהירות אחזור נתונים
* תקפות מטמון
* ניהול זיכרון
* התאוששות מכשלים

#### 4.4.3 MongoDB

**בדיקות נתונים לא מובנים:**

* שמירת לוגים
* שאילתות מורכבות
* אינדקסים
* ביצועי כתיבה

## 5. תרחישי בדיקה קריטיים

### 5.1 תרחיש זיהוי סיכון

1. מורה מזין נתונים על תלמיד
2. מערכת מנתחת את הנתונים
3. זיהוי רמת סיכון גבוהה
4. שליחת התראה אוטומטית
5. עדכון דשבורד מורה והורה

### 5.2 תרחיש עומס מערכת

1. 1000 משתמשים נכנסים במקביל
2. ביצוע פעולות מרובות
3. בדיקת יציבות המערכת
4. מדידת זמני תגובה
5. וידוא שמירת נתונים

### 5.3 תרחיש כשל טכני

1. סימולציה של כשל בשרת
2. הפעלת מנגנוני התאוששות
3. בדיקת גיבוי נתונים
4. מדידת זמן השבתה
5. וידוא המשכיות השירות

## 6. כלים וסביבת הבדיקות

### 6.1 כלי בדיקה אוטומטיים

* **Frontend**: Cypress, Jest, React Testing Library
* **Backend**: Jest, Supertest, Mocha
* **API**: Postman, Newman, Rest Assured
* **Database**: pgTAP, MongoDB Test Framework
* **Load Testing**: JMeter, k6
* **Security**: OWASP ZAP, SonarQube

### 6.2 סביבות בדיקה

* **Development**: סביבת פיתוח מקומית
* **Testing**: סביבת בדיקות מבודדת
* **Staging**: סביבת הכנה לייצור
* **Production**: סביבת ייצור (בדיקות מוגבלות)

## 7. לוח זמנים ותפקידים

### 7.1 שלב 1: הכנה (שבוע 1)

* הכנת תוכנית בדיקות מפורטת
* הקמת סביבות בדיקה
* הכנת נתוני בדיקה

### 7.2 שלב 2: בדיקות יחידה (שבועות 2-3)

* בדיקת כל רכיב בנפרד
* מדידת כיסוי קוד
* תיקון באגים ראשוניים

### 7.3 שלב 3: בדיקות אינטגרציה (שבועות 4-5)

* בדיקת חיבור בין רכיבים
* תקשורת API
* זרימת נתונים

### 7.4 שלב 4: בדיקות מערכת (שבוע 6)

* בדיקות קצה לקצה
* בדיקות עומס וביצועים
* בדיקות אבטחה

### 7.5 שלב 5: בדיקות קבלה (שבוע 7)

* בדיקות עם משתמשים אמיתיים
* וידוא דרישות עסקיות
* אישור לייצור

## 8. מדדי איכות והצלחה

### 8.1 מדדי ביצועים

* זמן תגובה ממוצע: < 2 שניות
* זמן טעינה עמוד: < 3 שניות
* זמינות מערכת: > 99.9%
* קיבולת משתמשים: 1000 במקביל

### 8.2 מדדי איכות

* כיסוי קוד: > 85%
* שיעור הצלחה בדיקות: > 95%
* מספר באגים קריטיים: < 5
* שביעות רצון משתמשים: > 4.5/5

### 8.3 מדדי אבטחה

* אין פרצות אבטחה קריטיות
* הצפנת נתונים: 100%
* אימות משתמשים: 2FA
* גישה מבוקרת: RBAC

## 9. תיעוד וניהול באגים

### 9.1 תיעוד בדיקות

* מקרי בדיקה מפורטים
* תוצאות בדיקות
* לוגי מערכת
* צילומי מסך

### 9.2 ניהול באגים

* דירוג חומרה: קריטי, גבוה, בינוני, נמוך
* מעקב אחר תיקונים
* בדיקות רגרסיה
* אימות תיקונים

### 9.3 דוחות מסכמים

* דוח בדיקות יומי
* דוח בדיקות שבועי
* דוח מסכם פרויקט
* המלצות לשיפור

## 10. הכשרה וחילוץ לקחים

### 10.1 הכשרת צוות

* הכשרה בכלי בדיקה
* סדנאות בדיקות אבטחה
* הכשרה בבדיקות ביצועים

### 10.2 חילוץ לקחים

* סיכום תהליך הבדיקות
* זיהוי נקודות שיפור
* עדכון תהליכי עבודה
* הכנה לפרויקטים עתידיים

## 11. מקרי בדיקה מפורטים (Test Cases)

### מקרה בדיקה #001: התחברות מורה למערכת

**תיאור הבדיקה:** בדיקת תהליך התחברות מורה עם נתונים תקינים **דרישות מוקדמות:** משתמש רשום במערכת עם הרשאות מורה **שלבי הבדיקה:**

1. פתיחת דף התחברות
2. הזנת שם משתמש: "teacher001"
3. הזנת סיסמה: "Password123!"
4. לחיצה על כפתור "התחבר"

**תוצאה צפויה:**

* התחברות מצליחה
* הפניה לדשבורד מורים
* הצגת הודעת ברכה "שלום מורה רחל"
* טעינה תוך 2 שניות

**תוצאה בפועל:**

* התחברות הצליחה
* הפניה לדשבורד מורים
* הצגת הודעת ברכה
* זמן טעינה: 1.8 שניות

**תוצאה:** ✅ **עבר** **הערות:** הבדיקה עברה בהצלחה, זמן התגובה מעולה

### מקרה בדיקה #002: זיהוי תלמיד בסיכון גבוה

**תיאור הבדיקה:** בדיקת אלגוריתם זיהוי סיכון כאשר תלמיד מקבל ציון סיכון גבוה **דרישות מוקדמות:** מורה מחובר למערכת, יש תלמיד ברשימה **שלבי הבדיקה:**

1. כניסה לעמוד "הערכת סיכון"
2. בחירת תלמיד "יוסי כהן"
3. הזנת פרמטרים: היעדרויות (15), ציונים נמוכים (3), התנהגות חריגה (כן)
4. לחיצה על "חשב סיכון"

**תוצאה צפויה:**

* רמת סיכון: "גבוה" (מעל 75%)
* הצגת התראה אדומה
* שליחת הודעה אוטומטית למנהל ולהורים
* שמירת הנתונים במערכת

**תוצאה בפועל:**

* רמת סיכון: 82% - "גבוה"
* הצגת התראה אדומה
* שליחת הודעה למנהל בלבד (לא להורים)
* שמירת הנתונים במערכת

**תוצאה:** ❌ **נכשל חלקית** **הערות:** הודעה להורים לא נשלחה - צריך לבדוק הגדרות שליחת הודעות

### מקרה בדיקה #003: יצירת דוח תקופתי

**תיאור הבדיקה:** בדיקת יצירת דוח תלמידים בסיכון עבור החודש האחרון **דרישות מוקדמות:** מורה מחובר, יש נתונים מהחודש האחרון **שלבי הבדיקה:**

1. כניסה לעמוד "דוחות"
2. בחירת סוג דוח: "תלמידים בסיכון"
3. הגדרת תאריך: "01/12/2024 - 31/12/2024"
4. בחירת פורמט: PDF
5. לחיצה על "צור דוח"

**תוצאה צפויה:**

* יצירת דוח תוך 10 שניות
* הורדת קובץ PDF
* הדוח מכיל: רשימת תלמידים, רמות סיכון, הממלצות
* פורמט מסודר וקריא

**תוצאה בפועל:**

* יצירת דוח תוך 12 שניות
* הורדת קובץ PDF
* הדוח מכיל את כל הנתונים הנדרשים
* פורמט מסודר

**תוצאה:** ✅ **עבר** **הערות:** זמן יצירה מעט מעל הצפוי אך מקובל, איכות הדוח גבוהה

### מקרה בדיקה #004: שליחת התראה דחופה

**תיאור הבדיקה:** בדיקת שליחת התראה דחופה למספר נמענים **דרישות מוקדמות:** מקרה דחוף זוהה במערכת **שלבי הבדיקה:**

1. זיהוי מצב דחוף (סיכון מעל 90%)
2. הפעלת מנגנון התראה דחופה
3. בחירת נמענים: מנהל, עו"ס, הורים
4. שליחת ההתראה

**תוצאה צפויה:**

* שליחת התראה תוך 30 שניות
* הגעת הודעה לכל הנמענים
* תיעוד השליחה במערכת
* אישור קבלה מהנמענים

**תוצאה בפועל:**

* שליחת התראה תוך 45 שניות
* הגעת הודעה למנהל ולעו"ס
* הודעה להורים נכשלה (מספר טלפון שגוי)
* תיעוד השליחה במערכת

**תוצאה:** ❌ **נכשל חלקית** **הערות:** זמן תגובה איטי, צריך לבדוק מנגנון עדכון פרטי קשר

### מקרה בדיקה #005: חיפוש תלמיד במערכת

**תיאור הבדיקה:** בדיקת פונקציונליות חיפוש תלמיד לפי שם **דרישות מוקדמות:** מורה מחובר, יש תלמידים במערכת **שלבי הבדיקה:**

1. כניסה לדף "רשימת תלמידים"
2. הזנת שם בשדה החיפוש: "מיכל"
3. לחיצה על "חפש"
4. בחינת התוצאות

**תוצאה צפויה:**

* הצגת כל התלמידים ששמם מכיל "מיכל"
* תוצאות תוך שנייה אחת
* סינון אוטומטי של הרשימה
* הדגשת המילה המחופשת

**תוצאה בפועל:**

* הצגת 3 תלמידים עם השם "מיכל"
* תוצאות תוך 0.8 שניות
* סינון אוטומטי פועל
* הדגשת המילה המחופשת

**תוצאה:** ✅ **עבר** **הערות:** פונקציונליות עובדת מעולה, מהירות טובה

### מקרה בדיקה #006: עדכון פרטי תלמיד

**תיאור הבדיקה:** בדיקת עדכון פרטי תלמיד קיים במערכת **דרישות מוקדמות:** מורה מחובר, תלמיד קיים במערכת **שלבי הבדיקה:**

1. חיפוש תלמיד "דוד לוי"
2. כניסה לעמוד עריכת פרטים
3. עדכון מספר טלפון הורים: 054-1234567
4. שמירת השינויים

**תוצאה צפויה:**

* עדכון המספר במערכת
* הצגת הודעת אישור
* שמירת השינוי בהיסטוריה
* עדכון מיידי בכל המקומות הרלוונטיים

**תוצאה בפועל:**

* עדכון המספר במערכת
* הצגת הודעת אישור
* שמירת השינוי בהיסטוריה
* עדכון מיידי בכל המקומות

**תוצאה:** ✅ **עבר** **הערות:** פונקציונליות עובדת כמצופה, ממשק ידידותי

### מקרה בדיקה #007: בדיקת גישה לא מורשית

**תיאור הבדיקה:** בדיקת חסימת גישה למשתמש ללא הרשאות מתאימות **דרישות מוקדמות:** משתמש עם הרשאות הורה נמצא במערכת **שלבי הבדיקה:**

1. התחברות כהורה
2. ניסיון גישה לדף "ניהול מערכת"
3. הזנת URL ישירות בדפדפן
4. בחינת התגובה

**תוצאה צפויה:**

* חסימת הגישה לדף
* הצגת הודעת שגיאה "אין הרשאה"
* הפניה לדף הבית
* רישום הניסיון בלוג האבטחה

**תוצאה בפועל:**

* חסימת הגישה לדף
* הצגת הודעת שגיאה "אין הרשאה"
* הפניה לדף הבית
* רישום הניסיון בלוג האבטחה

**תוצאה:** ✅ **עבר** **הערות:** מנגנון אבטחה פועל כמתוכנן, הגנה יעילה

### מקרה בדיקה #008: בדיקת עומס - 100 משתמשים

**תיאור הבדיקה:** בדיקת ביצועי המערכת עם 100 משתמשים פעילים במקביל **דרישות מוקדמות:** מערכת פעילה, כלי בדיקת עומס מוכן **שלבי הבדיקה:**

1. הפעלת בדיקת עומס עם 100 משתמשים וירטואליים
2. כל משתמש מבצע: התחברות, חיפוש, עדכון, התנתקות
3. מדידת זמני תגובה
4. בחינת יציבות המערכת

**תוצאה צפויה:**

* זמני תגובה מתחת ל-3 שניות
* אחוז שגיאות מתחת ל-1%
* יציבות המערכת לאורך כל הבדיקה
* צריכת זיכרון יציבה

**תוצאה בפועל:**

* זמני תגובה: 2.1 שניות בממוצע
* אחוז שגיאות: 0.5%
* המערכת יציבה לאורך כל הבדיקה
* צריכת זיכרון יציבה

**תוצאה:** ✅ **עבר** **הערות:** המערכת עמדה בעומס בהצלחה, ביצועים טובים

### מקרה בדיקה #009: גיבוי ושחזור נתונים

**תיאור הבדיקה:** בדיקת תהליך גיבוי אוטומטי ושחזור נתונים **דרישות מוקדמות:** מערכת גיבוי מוגדרת, נתונים לבדיקה **שלבי הבדיקה:**

1. הפעלת גיבוי אוטומטי
2. מחיקת נתונים מסוימים (בסביבת בדיקה)
3. שחזור נתונים מהגיבוי
4. בדיקת שלמות הנתונים

**תוצאה צפויה:**

* גיבוי מוצלח תוך 5 דקות
* שחזור מדויק של כל הנתונים
* אין אובדן מידע
* המערכת זמינה למשתמשים

**תוצאה בפועל:**

* גיבוי הושלם תוך 7 דקות
* שחזור מדויק של 98% מהנתונים
* אובדן מינורי של 2% מהנתונים
* המערכת זמינה למשתמשים

**תוצאה:** ❌ **נכשל חלקית** **הערות:** זמן גיבוי ארוך מהצפוי, יש אובדן מינורי של נתונים - דורש בדיקה נוספת

### מקרה בדיקה #010: נגישות לקוראי מסך

**תיאור הבדיקה:** בדיקת נגישות האתר לאנשים עם לקויות ראייה **דרישות מוקדמות:** מותקן קורא מסך NVDA **שלבי הבדיקה:**

1. הפעלת קורא מסך NVDA
2. ניווט באתר באמצעות מקלדת בלבד
3. בדיקת קריאת תוכן הדפים
4. מילוי טופס והגשה

**תוצאה צפויה:**

* קריאה נכונה של כל הכותרות
* הודעות מובנות לכל הפעולות
* אפשרות ניווט מלא במקלדת
* מילוי טפסים בקלות

**תוצאה בפועל:**

* קריאה נכונה של רוב הכותרות
* הודעות מובנות לרוב הפעולות
* ניווט במקלדת עובד
* קושי במילוי טופס עדכון פרטים

**תוצאה:** ❌ **נכשל חלקית** **הערות:** רוב הנגישות עובדת, אך יש בעיות בטופס עדכון פרטים - דורש תיקון

## 12. סיכום תוצאות הבדיקות

### 12.1 סטטיסטיקות כלליות

* **בדיקות שעברו:** 6/10 (60%)
* **בדיקות שנכשלו חלקית:** 4/10 (40%)
* **בדיקות שנכשלו לחלוטין:** 0/10 (0%)

### 12.2 נושאים לטיפול

1. **שליחת הודעות להורים** - מקרים #002, #004
2. **זמני תגובה** - מקרה #004
3. **גיבוי נתונים** - מקרה #009
4. **נגישות** - מקרה #010

### 12.3 המלצות

* תיקון מנגנון שליחת הודעות
* אופטימיזציה של מנגנון גיבוי
* שיפור נגישות טפסים
* בדיקות נוספות לאחר תיקונים

**הערות חשובות:**

* כל בדיקה תתועד במערכת ניהול הבדיקות
* בדיקות אבטחה יבוצעו בהתאם לתקנות הגנת הפרטיות
* מיוחד לסביבת בי"ס - התאמה לשעות עבודה ולוח השנה
* תיאום עם מנהל המערכת בבית הספר לביצוע בדיקות

# דוח בעיות וסיכונים - מערכת SafeGuard

## מעקב אחר בעיות שזוהו בבדיקות

## 1. סיכום כללי

### 1.1 סטטיסטיקות בעיות

* **בעיות קריטיות:** 1
* **בעיות בעדיפות גבוהה:** 2
* **בעיות בעדיפות בינונית:** 1
* **בעיות בעדיפות נמוכה:** 0
* **סה"כ בעיות:** 4

### 1.2 סטטוס תיקונים

* **בעיות תוקנו:** 2/4 (50%)
* **בעיות בתהליך תיקון:** 1/4 (25%)
* **בעיות ממתינות לתיקון:** 1/4 (25%)

## 2. בעיות קריטיות

### 🔴 בעיה #001: כשל בגיבוי והשחזור נתונים

**מספר תקלה:** SAFE-001  
**זוהתה במקרה בדיקה:** #009  
**תאריך זיהוי:** 15/01/2025

#### תיאור הבעיה

תהליך הגיבוי האוטומטי אורך זמן רב מהצפוי (7 דקות במקום 5), ובנוסף נמצא אובדן של 2% מהנתונים בתהליך השחזור. הבעיה התגלתה בעת בדיקת מנגנון הגיבוי וההתאוששות.

#### פרטים טכניים

* **מיקום הבעיה:** מודול Data Backup & Recovery
* **קבצים מושפעים:** backup\_manager.py, data\_recovery.py
* **מסד נתונים:** PostgreSQL, MongoDB
* **סוג הנתונים החסרים: רשומות פעילות משתמשים מהשעה האחרונה**

#### השפעה וחומרה

* **רמת חומרה:** 🔴 קריטית
* **השפעה עסקית:** עלולה להוביל לאובדן מידע על תלמידים בסיכון
* **משתמשים מושפעים:** כל משתמשי המערכת (פוטנציאל)
* **זמן השבתה:** אין השבתה, אך סיכון לאובדן נתונים

#### תיקון שבוצע

**תאריך תיקון:** 18/01/2025  
**אחראי:** צוות הפיתוח הראשי

**שלבי התיקון:**

1. **אבחון השורש:** זוהה bottleneck בשאילתות מסד הנתונים
2. **אופטימיזציה:** הוספת אינדקסים חדשים לטבלאות הגיבוי
3. **שיפור אלגוריתם:** שינוי מנגנון הגיבוי לבצע backup incremental
4. **בדיקת שלמות:** הוספת checksum לכל batch של נתונים
5. **תיקון race condition:** פתרון בעיה בסנכרון threads

**קבצים שתוקנו:**

* backup\_manager.py - אופטימיזציה של שאילתות
* data\_recovery.py - שיפור אלגוריתם השחזור
* database\_config.py - הוספת אינדקסים חדשים
* backup\_scheduler.py - שיפור תזמון הגיבוי

#### בדיקת התיקון

**תאריך בדיקה:** 19/01/2025  
**תוצאות לאחר התיקון:**

* זמן גיבוי: 4.2 דקות ✅
* אובדן נתונים: 0% ✅
* שלמות נתונים: 100% ✅
* יציבות התהליך: ✅

#### סיכונים נותרים

* **סיכון נמוך:** יש להמשיך לנטר ביצועי הגיבוי
* **המלצה:** הגדרת מוניטורינג אוטומטי לזמני גיבוי
* **מעקב:** בדיקה חודשית של תהליך הגיבוי

## 3. בעיות בעדיפות גבוהה

### 🟠 בעיה #002: כשל בשליחת הודעות להורים

**מספר תקלה:** SAFE-002  
**זוהתה במקרי בדיקה:** #002, #004  
**תאריך זיהוי:** 14/01/2025

#### תיאור הבעיה

מערכת לא שולחת הודעות להורים כאשר זוהה תלמיד בסיכון גבוה. הבעיה התגלתה כהול בשני תרחישים נפרדים - הודעות אוטומטיות והודעות דחופות.

#### פרטים טכניים

* **מיקום הבעיה:** מודול Alert Management
* **קבצים מושפעים:** alert\_sender.py, parent\_notification.py
* **שירותים חיצוניים:** SMS Gateway, Email Service
* **סוג הכשל:** connection timeout, invalid phone numbers

#### השפעה וחומרה

* **רמת חומרה:** 🟠 גבוהה
* **השפעה עסקית:** הורים לא מקבלים התראות חשובות על ילדיהם
* **משתמשים מושפעים:** כל המשפחות במערכת
* **השפעה על בטיחות:** תלמידים בסיכון עלולים לא לקבל עזרה בזמן

#### תיקון שבוצע

**תאריך תיקון:** 16/01/2025  
**אחראי:** צוות הפיתוח הראשי

**שלבי התיקון:**

1. **אבחון:** זוהו פרטי קשר לא מעודכנים במסד הנתונים
2. **וולידציה:** הוספת בדיקת תקינות מספרי טלפון ומייל
3. **retry mechanism:** הוספת מנגנון חזרה על נסיונות כושלים
4. **logging:** שיפור מנגנון הלוגים למעקב אחר שליחות
5. **fallback:** הוספת שליחה דרך ערוץ חלופי בכשל

**קבצים שתוקנו:**

* alert\_sender.py - הוספת מנגנון retry
* parent\_notification.py - שיפור וולידציה
* contact\_validator.py - קובץ חדש לבדיקת תקינות
* notification\_config.py - הגדרות חדשות

#### בדיקת התיקון

**תאריך בדיקה:** 17/01/2025  
**תוצאות לאחר התיקון:**

* שליחת הודעות למנהל: 100% ✅
* שליחת הודעות לעו"ס: 100% ✅
* שליחת הודעות להורים: 95% ✅
* זמן שליחה: 25 שניות ✅

#### סיכונים נותרים

* **סיכון נמוך:** 5% מההודעות עדיין נכשלות בגלל נתונים שגויים
* **המלצה:** תהליך עדכון תקופתי של פרטי קשר
* **מעקב:** בדיקה יומית של סטטוס שליחת הודעות

### 🟠 בעיה #003: בעיות נגישות לקוראי מסך

**מספר תקלה:** SAFE-003  
**זוהתה במקרה בדיקה:** #010  
**תאריך זיהוי:** 15/01/2025

#### תיאור הבעיה

קוראי מסך מתקשים לקרוא חלקים מהאתר, במיוחד בטופס עדכון פרטי תלמידים. הבעיה מונעת מאנשים עם לקויות ראייה לעשות שימוש מלא במערכת.

#### פרטים טכניים

* **מיקום הבעיה:** Frontend - Teacher Dashboard
* **קבצים מושפעים:** student\_form.jsx, form\_styles.css
* **בעיות ספציפיות:** חסרים labels, בעיות מבנה HTML
* **דפדפנים מושפעים:** כל הדפדפנים עם קוראי מסך

#### השפעה וחומרה

* **רמת חומרה:** 🟠 גבוהה
* **השפעה עסקית:** אפליה של משתמשים עם מוגבלות
* **משתמשים מושפעים:** מורים ועובדים עם לקויות ראייה
* **השפעה משפטית:** אי עמידה בתקני נגישות

#### תיקון בתהליך

**תאריך תחילת תיקון:** 17/01/2025  
**אחראי:** צוות Frontend  
**סטטוס:** 🟡 בתהליך (80% הושלם)

**שלבי התיקון המתוכננים:**

1. **הוספת labels:** הוספת תוויות לכל שדות הטופס ✅
2. **שיפור מבנה HTML:** שינוי לsemantic HTML ✅
3. **ARIA attributes:** הוספת תכונות נגישות ✅
4. **keyboard navigation:** שיפור ניווט במקלדת ✅
5. **בדיקה עם NVDA:** בדיקה חוזרת עם קורא מסך 🟡

**קבצים בתיקון:**

* student\_form.jsx - הוספת נגישות
* form\_styles.css - שיפור עיצוב נגיש
* accessibility\_utils.js - יוטיליטי נגישות חדש

#### בדיקת התיקון (חלקית)

**תאריך בדיקה:** 20/01/2025  
**תוצאות עד כה:**

* קריאת כותרות: 100% ✅
* הבנת הודעות: 90% ✅
* ניווט במקלדת: 95% ✅
* מילוי טופס: 75% 🟡 (עדיין בעבודה)

#### סיכונים נותרים

* **סיכון בינוני:** טופס עדכון פרטים עדיין לא נגיש מספיק
* **המלצה:** השלמת התיקון וביצוע בדיקות נוספות
* **מעקב:** בדיקה עם משתמשים עם מוגבלות ראייה

## 4. בעיות בעדיפות בינונית

### 🟡 בעיה #004: זמן תגובה איטי בהתראות דחופות

**מספר תקלה:** SAFE-004  
**זוהתה במקרה בדיקה:** #004  
**תאריך זיהוי:** 14/01/2025

#### תיאור הבעיה

זמן תגובה בשליחת התראות דחופות ארוך מהצפוי - 45 שניות במקום 30 שניות. הבעיה עלולה לדחות התערבות קריטית.

#### פרטים טכניים

* **מיקום הבעיה:** Alert Processing Engine
* **קבצים מושפעים:** urgent\_alert\_processor.py
* **בעיית ביצועים:** איטיות בשאילתות מסד נתונים
* **תלות חיצונית:** SMS Gateway איטי

#### השפעה וחומרה

* **רמת חומרה:** 🟡 בינונית
* **השפעה עסקית:** עיכוב בהתערבות במקרי חירום
* **משתמשים מושפעים:** גורמים מטפלים בחירום
* **השפעה על בטיחות:** עיכוב בתגובה לסיכון

#### תיקון ממתין

**תאריך תחילת תיקון:** טרם נקבע  
**אחראי:** צוות Backend  
**סטטוס:** 🔴 ממתין לתיקון

**שלבי התיקון המתוכננים:**

1. **אופטימיזציה:** שיפור שאילתות מסד נתונים
2. **caching:** הוספת מטמון לנתונים נפוצים
3. **async processing:** עיבוד אסינכרוני של התראות
4. **provider alternatives:** הוספת ספק SMS חלופי

**קבצים לתיקון:**

* urgent\_alert\_processor.py - אופטימיזציה
* alert\_cache.py - מטמון חדש
* sms\_provider\_manager.py - ניהול ספקים

#### סיכונים נותרים

* **סיכון גבוה:** בינתיים התראות דחופות עדיין איטיות
* **המלצה:** קביעת עדיפות גבוהה לתיקון
* **מעקב:** מעקב יומי אחר זמני תגובה

## 5. מטריקות ומעקב

### 5.1 מטריקות איכות נוכחיות

|  |  |  |  |
| --- | --- | --- | --- |
| מטריקה | יעד | מצב נוכחי | סטטוס |
| זמן תגובה ממוצע | < 2 שניות | 2.1 שניות | 🟡 |
| זמינות מערכת | > 99.9% | 99.7% | 🟡 |
| שיעור הצלחה בדיקות | > 95% | 90% | 🟡 |
| מספר באגים קריטיים | < 5 | 1 | ✅ |
| כיסוי קוד | > 85% | 87% | ✅ |

### 5.2 מעקב אחר תיקונים

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| בעיה | תאריך זיהוי | תאריך יעד | סטטוס | התקדמות |
| גיבוי נתונים | 15/01/2025 | 19/01/2025 | ✅ תוקן | 100% |
| הודעות להורים | 14/01/2025 | 17/01/2025 | ✅ תוקן | 100% |
| נגישות | 15/01/2025 | 22/01/2025 | 🟡 בתהליך | 80% |
| זמן תגובה | 14/01/2025 | 25/01/2025 | 🔴 ממתין | 0% |

## 6. תוכנית פעולה

### 6.1 פעולות דחופות (השבוע הקרוב)

1. **השלמת תיקון נגישות** - עדיפות גבוהה
   * מסיים: צוות Frontend
   * יעד: 22/01/2025
2. **תחילת תיקון זמני תגובה** - עדיפות בינונית
   * מסיים: צוות Backend
   * יעד: 25/01/2025

### 6.2 פעולות מניעה

1. **מוניטורינג משופר:**
   * התקנת מערכת ניטור זמן אמת
   * התראות אוטומטיות על חריגות
2. **בדיקות רגרסיה:**
   * הרצה אוטומטית של בדיקות לאחר כל שינוי
   * בדיקות יומיות של פונקציות קריטיות
3. **תהליך code review:**
   * בדיקת קוד כפולה לפני פריסה
   * דגש על ביצועים ואבטחה

### 6.3 שיפורים לטווח ארוך

1. **הרחבת כיסוי בדיקות:**
   * הוספת בדיקות אוטומטיות
   * בדיקות עומס קבועות
2. **שיפור תהליכי CI/CD:**
   * פריסה אוטומטית לסביבת staging
   * בדיקות אוטומטיות בפיפליין

## 7. הוראות חירום

### 7.1 במקרה של בעיה קריטית חדשה

1. **הפסקת השירות:** אם נדרש לבטיחות הנתונים
2. **הודעה למשתמשים:** עדכון אוטומטי על הסטטוס
3. **הפעלת תוכנית התאוששות:** גיבוי ושחזור
4. **מעקב צמוד:** עדכונים כל 30 דקות

### 7.2 אנשי קשר לחירום

* **מוביל פיתוח:** יוסי כהן - 054-1234567
* **מנהל מערכת:** שרה לוי - 054-7654321
* **מנהל פרויקט:** דוד ישראל - 054-9876543

### 7.3 תקשורת עם בתי הספר

* **הודעה מיידית:** לכל המנהלים המשתמשים
* **עדכון שעתי:** עד לפתרון הבעיה
* **דוח מסכם:** תוך 24 שעות מפתרון

## 8. לקחים והמלצות

### 8.1 לקחים שנלמדו

1. **חשיבות בדיקות גיבוי:** בדיקות רגילות לא מספיקות
2. **נגישות מההתחלה:** קל יותר לבנות נגיש מלתקן
3. **מוניטורינג בזמן אמת:** הכרחי לזיהוי מוקדם
4. **תיעוד מדויק:** חוסך זמן בתיקונים

### 8.2 המלצות לעתיד

1. **השקעה בכלי מוניטורינג:** מערכת מקצועית
2. **הכשרת צוות:** בנושאי נגישות ואבטחה
3. **בדיקות עם משתמשים:** מוקדם בתהליך הפיתוח
4. **תיעוד מסודר:** של כל בעיה ופתרון

### 8.3 מדדי הצלחה לעתיד

* **זמן תגובה לבעיות:** פחות מ-4 שעות
* **שיעור בעיות חוזרות:** פחות מ-5%
* **שביעות רצון משתמשים:** מעל 4.5/5
* **זמינות מערכת:** מעל 99.9%

**עדכון אחרון:** 21/01/2025  
**אחראי על הדוח:** מנהל הפרויקט  
**בדיקה הבאה:** 28/01/2025

**הערות:**

* דוח זה מתעדכן שבועית
* כל בעיה חדשה תתווסף לטבלה
* תיקונים יאושרו רק לאחר בדיקה מלאה
* חשוב לתאם עם בתי הספר לפני שינויים גדולים

# סיכום מסקנות וההמלצות - מערכת SafeGuard

## ניתוח תוצאות בדיקות ותוכנית לעתיד

## 1. סיכום כללי של הבדיקות

### 1.1 סטטיסטיקות מסכמות

|  |  |  |  |
| --- | --- | --- | --- |
| מדד | תוצאה | יעד | הערכה |
| **אחוז הצלחה כללי** | 75% | 90% | 🟡 דורש שיפור |
| **בעיות קריטיות** | 1 | 0 | 🟡 תוקן אך דורש מעקב |
| **בעיות גבוהות** | 2 | מקסימום 1 | 🟠 דורש תשומת לב |
| **זמן תגובה ממוצע** | 2.1 שניות | 2.0 שניות | 🟡 קרוב ליעד |
| **כיסוי בדיקות** | 87% | 85% | ✅ מעל היעד |
| **זמינות מערכת** | 99.7% | 99.9% | 🟡 מתחת ליעד |

### 1.2 התפלגות בעיות לפי רכיבים

Frontend (UI/UX): 25% (1 בעיה)

Backend Services: 50% (2 בעיות)

Data Layer: 25% (1 בעיה)

## 2. מה עובד טוב ✅

### 2.1 חוזקות טכניות

#### ****אבטחה ובקרת גישה****

* מנגנון הרשאות פועל מעולה (100% הצלחה)
* חסימת גישה לא מורשית יעילה
* הצפנת נתונים ברמה גבוהה
* מנגנון 2FA פועל ללא תקלות

#### ****פונקציונליות עיקרית****

* זיהוי סיכונים: דיוק של 95%+
* מנגנון חיפוש: מהיר ומדויק (0.8 שניות)
* יצירת דוחות: איכות גבוהה ופורמט מסודר
* עדכון נתונים: מהיר ואמין

#### ****ביצועים תחת עומס****

* עמידה ב-100 משתמשים במקביל ללא בעיות
* צריכת זיכרון יציבה
* אחוז שגיאות נמוך (0.5%)
* התאוששות מהירה מעומסים

### 2.2 חוזקות במשתמש והממשק

#### ****קלות שימוש****

* ממשק אינטואיטיבי למורים
* זמני למידה קצרים למשתמשים חדשים
* ניווט ברור ולוגי
* הודעות שגיאה מובנות

#### ****תמיכה טכנית****

* תיעוד מקיף ומעודכן
* מערכת עזרה מובנית
* תמיכה בדפדפנים מרכזיים
* גיבוי אוטומטי (לאחר התיקון)

### 2.3 חוזקות ארכיטקטוניות

#### ****מודולריות****

* ארכיטקטורת microservices יציבה
* הפרדה נכונה בין שכבות
* קלות תחזוקה ועדכון
* יכולת הרחבה (scalability)

#### ****איכות קוד****

* כיסוי בדיקות גבוה (87%)
* קוד מסודר ומתועד
* עמידה בסטנדרטים טכניים
* מעט technical debt

## 3. מה צריך לשפר 🔧

### 3.1 נושאים טכניים לשיפור

#### ****ביצועים ותגובתיות****

**בעיות מזוהות:**

* זמני תגובה לא אחידים במערכת
* התראות דחופות איטיות (45 שניות)
* עומס מסד נתונים בשעות שיא
* יעילות גיבוי נתונים

**פעולות נדרשות:**

* אופטימיזציה של שאילתות מסד נתונים
* הוספת מטמון (caching) חכם
* load balancing משופר
* CDN לתכנים סטטיים

#### ****אמינות ותקשורת****

**בעיות מזוהות:**

* שליחת הודעות להורים לא יציבה
* תלות בשירותי צד שלישי
* מנגנון התאוששות לא מושלם
* עדכון פרטי קשר לא אוטומטי

**פעולות נדרשות:**

* מנגנון retry מתקדם
* הוספת ספקי שירות חלופיים
* תהליך עדכון קשרים אוטומטי
* מוניטורינג בזמן אמת

### 3.2 חוויית משתמש ונגישות

#### ****נגישות****

**בעיות מזוהות:**

* תמיכה חלקית בקוראי מסך
* חסרים labels בטפסים
* ניווט במקלדת לא מושלם
* ניגודיות צבעים בחלק מהמסכים

**פעולות נדרשות:**

* עמידה מלאה בתקן WCAG 2.1 AA
* הוספת ARIA attributes
* שיפור keyboard navigation
* בדיקות עם משתמשים עם מוגבלויות

#### ****ממשק משתמש****

**בעיות מזוהות:**

* טפסים מורכבים מדי
* חסרות קיצורי דרך
* לא רספונסיבי במכשירים מסוימים
* הודעות שגיאה לא תמיד ברורות

**פעולות נדרשות:**

* פישוט טפסים מורכבים
* הוספת keyboard shortcuts
* שיפור responsive design
* הוספת הודעות עזרה

### 3.3 תהליכים ותחזוקה

#### ****מוניטורינג ובקרה****

**בעיות מזוהות:**

* מוניטורינג לא מספיק מקיף
* התראות מוקדמות חסרות
* לוגים לא מסודרים מספיק
* מעקב ביצועים ידני

**פעולות נדרשות:**

* מערכת מוניטורינג מתקדמת
* dashboards בזמן אמת
* אוטומציה של התראות
* BI למעקב אחר מגמות

## 4. המלצות לעתיד 🚀

### 4.1 שיפורים לטווח קצר (1-3 חודשים)

#### ****עדיפות גבוהה****

1. **השלמת תיקון נגישות**
   * השלמת עמידה ב-WCAG 2.1
   * בדיקות עם משתמשים עם מוגבלויות
   * הדרכת צוות הפיתוח
2. **אופטימיזציה של ביצועים**
   * שיפור זמני תגובה להתראות דחופות
   * אופטימיזציה של מסד הנתונים
   * הוספת CDN
3. **שיפור מנגנון התראות**
   * ספקי SMS חלופיים
   * וולידציה טובה יותר של פרטי קשר
   * מנגנון retry מתקדם

#### ****עדיפות בינונית****

1. **מערכת מוניטורינג משופרת**
   * התקנת Grafana + Prometheus
   * dashboards מותאמים אישית
   * התראות אוטומטיות
2. **שיפור תהליכי CI/CD**
   * בדיקות אוטומטיות בפיפליין
   * deployment אוטומטי לסביבת staging
   * rollback אוטומטי במקרה של בעיה

### 4.2 פיתוחים לטווח בינוני (3-6 חודשים)

#### ****תכונות חדשות****

1. **מודול ניתוח מתקדם**
   * בינה מלאכותית לחיזוי סיכונים
   * ניתוח מגמות ודפוסים
   * המלצות אוטומטיות לפעולות
2. **אפליקציה לניידים**
   * אפליקציה ייעודית להורים
   * push notifications
   * ממשק מותאם לצגים קטנים
3. **אינטגרציות נוספות**
   * חיבור למערכות בית ספר קיימות
   * API פתוח לצדדים שלישיים
   * ייצוא נתונים למערכות BI

#### ****שיפור תשתיות****

1. **ארכיטקטורת cloud מתקדמת**
   * מעבר לcontainers (Docker/Kubernetes)
   * auto-scaling בזמן אמת
   * multi-region deployment
2. **אבטחה מתקדמת**
   * Zero Trust Architecture
   * encryption מתקדם
   * audit trail מקיף

### 4.3 חזון לטווח ארוך (6-12 חודשים)

#### ****טכנולוגיות מתקדמות****

1. **בינה מלאכותית וחכמה**
   * מודל ML לחיזוי סיכונים מתקדם
   * עיבוד שפה טבעית להודעות
   * computer vision לניתוח התנהגות
2. **פלטפורמה רחבה**
   * מערכת כוללת לניהול בית ספר
   * מודולים נוספים (נוכחות, ציונים)
   * marketplace של תוספות
3. **אנליטיקה מתקדמת**
   * big data analytics
   * machine learning ops
   * predictive analytics

## 5. רשימת כלים לבדיקה 🛠️

### 5.1 תוכנות בדיקה אוטומטיות

#### ****בדיקות Frontend****

|  |  |  |  |
| --- | --- | --- | --- |
| כלי | תיאור | רישיון | התקנה |
| **Cypress** | E2E testing מודרני | Open Source | npm install cypress |
| **Selenium WebDriver** | אוטומציה דפדפן קלאסית | Open Source | pip install selenium |
| **Playwright** | בדיקות cross-browser | Open Source | npm install playwright |
| **Jest + React Testing Library** | בדיקות יחידה React | Open Source | npm install jest @testing-library/react |
| **Storybook** | בדיקות רכיבי UI | Open Source | npx storybook init |

#### ****בדיקות Backend****

|  |  |  |  |
| --- | --- | --- | --- |
| כלי | תיאור | רישיון | התקנה |
| **Postman** | בדיקות API ידניות | Freemium | הורדה מהאתר |
| **Newman** | הרצת Postman בCI | Open Source | npm install newman |
| **Jest/Mocha** | בדיקות יחידה Node.js | Open Source | npm install jest |
| **Supertest** | בדיקות API ב-Node.js | Open Source | npm install supertest |
| **Artillery** | בדיקות עומס API | Open Source | npm install artillery |

#### ****בדיקות מסד נתונים****

|  |  |  |  |
| --- | --- | --- | --- |
| כלי | תיאור | רישיון | התקנה |
| **pgTAP** | בדיקות PostgreSQL | Open Source | דרך PostgreSQL |
| **MongoDB Test Framework** | בדיקות MongoDB | Open Source | דרך MongoDB |
| **Database Rider** | בדיקות DB ב-Java | Open Source | Maven/Gradle |
| **Flyway** | בדיקות migration | Commercial | pip install flyway |

### 5.2 כלי בדיקות ביצועים

#### ****עומס ולחץ****

|  |  |  |  |
| --- | --- | --- | --- |
| כלי | תיאור | רישיון | מומלץ לשימוש |
| **Apache JMeter** | בדיקות עומס מקיפות | Open Source | בדיקות עומס כלליות |
| **k6** | בדיקות עומס מודרניות | Open Source | בדיקות API ומערכות מודרניות |
| **LoadRunner** | בדיקות עומס מתקדמות | Commercial | בדיקות enterprise מורכבות |
| **Gatling** | בדיקות עומס בעלות ביצועים גבוהים | Open Source | בדיקות עומס כבד |
| **Artillery** | בדיקות עומס פשוטות | Open Source | בדיקות מהירות |

#### ****ניתוח ביצועים****

|  |  |  |  |
| --- | --- | --- | --- |
| כלי | תיאור | רישיון | שימוש |
| **GTmetrix** | ניתוח מהירות אתר | Freemium | אופטימיזציה Frontend |
| **Lighthouse** | audit ביצועים Chrome | Open Source | בדיקות PWA ונגישות |
| **WebPageTest** | בדיקות ביצועים מפורטות | Open Source | ניתוח מעמיק |
| **New Relic** | APM מקיף | Commercial | מוניטורינג production |

### 5.3 כלי בדיקות אבטחה

#### ****סריקת אבטחה****

|  |  |  |  |
| --- | --- | --- | --- |
| כלי | תיאור | רישיון | רמת מומחיות |
| **OWASP ZAP** | סריקת אבטחה אוטומטית | Open Source | מתחיל-בינוני |
| **Burp Suite** | בדיקות אבטחה מתקדמות | Freemium | מתקדם |
| **Nessus** | סריקת פגיעויות | Commercial | מתקדם |
| **SonarQube** | ניתוח איכות ואבטחת קוד | Open Source | מתחיל-מתקדם |
| **Snyk** | סריקת תלויות לאבטחה | Freemium | מתחיל |

#### ****ניתוח קוד****

|  |  |  |  |
| --- | --- | --- | --- |
| כלי | תיאור | רישיון | שפות נתמכות |
| **ESLint** | ניתוח JavaScript/TypeScript | Open Source | JS/TS |
| **Pylint** | ניתוח Python | Open Source | Python |
| **CodeQL** | ניתוח אבטחה מתקדם | Free for OSS | רב-שפות |
| **Checkmarx** | SAST מסחרי | Commercial | רב-שפות |

### 5.4 כלי בדיקות נגישות

#### ****בדיקות אוטומטיות****

|  |  |  |  |
| --- | --- | --- | --- |
| כלי | תיאור | רישיון | אינטגרציה |
| **axe-core** | בדיקות נגישות אוטומטיות | Open Source | דפדפן/CI |
| **WAVE** | Web Accessibility Evaluation | Free | דפדפן |
| **Pa11y** | בדיקות נגישות בCI | Open Source | Command line |
| **Lighthouse** | כולל בדיקות נגישות | Open Source | Chrome DevTools |

#### ****כלי ידניים****

|  |  |  |  |
| --- | --- | --- | --- |
| כלי | תיאור | רישיון | שימוש |
| **NVDA** | קורא מסך חינמי | Open Source | בדיקות ידניות |
| **JAWS** | קורא מסך מקצועי | Commercial | בדיקות מקצועיות |
| **Colour Contrast Analyser** | בדיקת ניגודיות | Free | בדיקת צבעים |
| **HeadingsMap** | ניתוח מבנה כותרות | Free | הרחבת דפדפן |

### 5.5 דפדפנים לבדיקה

#### ****דפדפנים עיקריים****

|  |  |  |  |
| --- | --- | --- | --- |
| דפדפן | גרסאות לבדיקה | שימוש נפוץ | עדיפות |
| **Google Chrome** | 3 גרסאות אחרונות | 65% | גבוהה מאוד |
| **Mozilla Firefox** | 3 גרסאות אחרונות | 15% | גבוהה |
| **Safari** | 2 גרסאות אחרונות | 12% | בינונית-גבוהה |
| **Microsoft Edge** | 2 גרסאות אחרונות | 5% | בינונית |
| **Opera** | גרסה אחרונה | 2% | נמוכה |

#### ****דפדפנים במובייל****

|  |  |  |  |
| --- | --- | --- | --- |
| דפדפן | פלטפורמה | שימוש | עדיפות |
| **Chrome Mobile** | Android | 45% | גבוהה מאוד |
| **Safari Mobile** | iOS | 35% | גבוהה מאוד |
| **Samsung Internet** | Android | 10% | בינונית |
| **Firefox Mobile** | Android/iOS | 5% | נמוכה |

### 5.6 מכשירים לבדיקה

#### ****מחשבים נייחים****

|  |  |  |  |
| --- | --- | --- | --- |
| רזולוציה | תיאור | שימוש | עדיפות |
| **1920x1080** | Full HD סטנדרטי | 40% | גבוהה מאוד |
| **1366x768** | HD נפוץ בלפטופים | 25% | גבוהה |
| **2560x1440** | QHD מסכים גדולים | 15% | בינונית |
| **3840x2160** | 4K מסכים מתקדמים | 10% | נמוכה |

#### ****מכשירים ניידים****

|  |  |  |  |
| --- | --- | --- | --- |
| מכשיר | רזולוציה | שימוש | עדיפות |
| **iPhone 12/13/14** | 390x844 | 20% | גבוהה מאוד |
| **Samsung Galaxy S21-S23** | 360x800 | 18% | גבוהה |
| **iPad** | 768x1024 | 12% | גבוהה |
| **Android Generic** | 360x640 | 25% | גבוהה |

#### ****רזולוציות נוספות לבדיקה****

Desktop:

- 1280x720 (HD)

- 1440x900 (MacBook Air ישן)

- 1920x1200 (WUXGA)

Mobile:

- 320x568 (iPhone 5)

- 375x667 (iPhone 6/7/8)

- 414x896 (iPhone XR)

- 360x740 (Android רגיל)

### 5.7 כלי מוניטורינג והקשה

#### ****מוניטורינג ביצועים****

|  |  |  |  |
| --- | --- | --- | --- |
| כלי | תיאור | רישיון | שימוש |
| **Grafana** | dashboards ויזואליים | Open Source | תצוגת מטריקות |
| **Prometheus** | איסוף מטריקות | Open Source | מוניטורינג |
| **New Relic** | APM מקיף | Commercial | מוניטורינג production |
| **DataDog** | מוניטורינג ו-logs | Commercial | חברות גדולות |
| **Pingdom** | מוניטורינג זמינות | Commercial | uptime monitoring |

#### ****ניתוח לוגים****

|  |  |  |  |
| --- | --- | --- | --- |
| כלי | תיאור | רישיון | מקרי שימוש |
| **ELK Stack** | Elasticsearch+Logstash+Kibana | Open Source | ניתוח לוגים מקיף |
| **Fluentd** | איסוף וניתוב לוגים | Open Source | ETL ללוגים |
| **Splunk** | פלטפורמת logs מתקדמת | Commercial | enterprise logging |
| **Graylog** | ניהול לוגים מרכזי | Open Source | ניהול לוגים |

### 5.8 כלי עזר נוספים

#### ****תוכנות עזר לבדיקה****

|  |  |  |  |
| --- | --- | --- | --- |
| כלי | תיאור | רישיון | שימוש |
| **Wireshark** | ניתוח תעבורת רשת | Open Source | בדיקות רשת |
| **Fiddler** | web debugging proxy | Free | בדיקת HTTP requests |
| **Charles Proxy** | web debugging proxy | Commercial | בדיקת mobile apps |
| **Insomnia** | REST client | Freemium | בדיקות API |

#### ****כלי אוטומציה****

|  |  |  |  |
| --- | --- | --- | --- |
| כלי | תיאור | רישיון | שימוש |
| **GitHub Actions** | CI/CD פלטפורמה | Free/Commercial | אוטומציה |
| **Jenkins** | CI/CD server | Open Source | אוטומציה מקומית |
| **Docker** | containerization | Open Source | בדיקות בסביבות מבודדות |
| **Vagrant** | Virtual machines | Open Source | סביבות פיתוח |

## 6. תוכנית יישום ההמלצות

### 6.1 לוח זמנים מפורט

#### ****שבוע 1-2: הכנות****

* רכישת כלים מסחריים נדרשים
* התקנת כלי הבדיקה החדשים
* הכשרת צוות הבדיקות
* הגדרת סביבות בדיקה מחודשות

#### ****שבוע 3-4: תיקונים דחופים****

* השלמת תיקון נגישות
* אופטימיזציה של זמני תגובה
* שיפור מנגנון התראות
* בדיקות רגרסיה

#### ****שבוע 5-8: שיפורים מובנים****

* הטמעת מערכת מוניטורינג
* שיפור תהליכי CI/CD
* אוטומציה של בדיקות נוספות
* הדרכת משתמשים

#### ****חודש 3-6: פיתוחים עתידיים****

* פיתוח תכונות חדשות
* שיפור ארכיטקטורה
* הרחבת יכולות המערכת
* אינטגרציות חדשות

### 6.2 תקציב משוער

#### ****כלים ורישיונות (שנתי)****

LoadRunner Professional: $15,000

New Relic Pro: $8,000

Burp Suite Professional: $3,000

מכשירי בדיקה (tablets, phones): $5,000

Training וקורסים: $10,000

סה"כ: $41,000

#### ****משאבי אנוש****

מהנדס QA senior נוסף: $80,000

מתמחה בנגישות: $45,000

DevOps engineer: $90,000

סה"כ: $215,000

### 6.3 מדדי הצלחה

#### ****מדדים טכניים****

* זמן תגובה ממוצע < 1.5 שניות
* זמינות מערכת > 99.95%
* אחוז הצלחה בדיקות > 98%
* כיסוי קוד > 90%

#### ****מדדי משתמש****

* שביעות רצון משתמשים > 4.7/5
* זמן למידה < 30 דקות
* שיעור אימוץ > 95%
* תלונות טכניות < 5 לחודש

## 7. סיכום ומסקנות

### 7.1 הישגים עיקריים

מערכת SafeGuard מראה **פוטנציאל גבוה** עם בסיס טכני חזק. הצלחנו לזהות ולתקן בעיות קריטיות, והמערכת יציבה בתנאים רגילים.

### 7.2 אתגרים מרכזיים

העיקרי שצריך התמקדות הוא **שיפור חוויית המשתמש** ו**אמינות התקשורת**. נושא הנגישות דורש השקעה נוספת.

### 7.3 ההמלצה הכללית

**המערכת ראויה להמשך הפיתוח והטמעה**, בתנאי שמשקיעים במשאבים הנדרשים לתיקון הבעיות שזוהו.

### 7.4 צעדים הבאים

1. **תיקון בעיות גבוהות** בעדיפות ראשונה
2. **הטמעת כלי מוניטורינג** מתקדמים
3. **הרחבת צוות הבדיקות** עם מומחי נגישות
4. **תכנון פיתוחים עתידיים** בהתאם לצרכים שזוהו

**תאריך סיכום:** 12/07/2025  
**מכין הדוח:** צוות בדיקות SafeGuard  
**אישור:** מנהל הפרויקט