# אפיון טכני - מערכת דיווח תקלות

## 1. מבוא

מערכת דיווח תקלות נועדה לאפשר למשתמשים לדווח על תקלות וגם לבצע בקשות הרשאות ולתת הצעת יעול עבור המערכותבצורה יעילה, תוך ניהול גמיש של סוגי התקלות והמודולים, שמירת הנתונים למעקב וניתוח, והפקת דוחות בצורה נוחה. המערכת תאפשר שימוש בסביבה גמישה כך שניתן יהיה להחליף שרתים ולבצע התאמות ללא שינוי קוד משמעותי.

## 2. מטרות המערכת

• דיווח תקלות בקלות ובמהירות

• ניהול דינמי של סוגי התקלות והמודולים באמצעות קובץ קונפיגורציה

• שמירת הנתונים למעקב וניתוח במסד נתונים או קובץ

• אפשרות ייצוא נתונים לקובץ Excel

• תמיכה בהחלפת שרתים בקלות באמצעות קונפיגורציה דינמית

• הקטנת זמני התגובה ושיפור תהליכי הטיפול בתקלות

## 3. מבנה כללי של המערכת

### 3.1. ממשק משתמש (Frontend)

ממשק המשתמש יתבסס על טכנולוגיות HTML, CSS, JavaScript (Vanilla/React) ויאפשר:

• טופס דיווח תקלה הכולל:

1. **שם מדווח** - שדה חופשי לקבלת טקסט מכל סוג.
2. **תאריך דיווח** - **לא יוצג בטופס**, יתווסף אוטומטית בקובץ ה-Excel.
3. **שם המערכת** - רשימה נפתחת עם אפשרות לבחור בין 10 מערכות (מערכת 1 עד מערכת 10).
4. **סיבת הפנייה** - רשימה נפתחת עם 3 אפשרויות:
   * **תקלה** - יפתח שדה טקסט חופשי עם כותרת: "תיאור שחזור התקלה".
   * **בקשת הרשאות** - יפתח שדה טקסט חופשי עם כותרת: "הכנס בקשה מפורטת לקבלת הרשאה".
   * **הצעת ייעול** - יפתח שדה טקסט חופשי עם כותרת: "הכנס את הצעת הייעול בצורה מפורטת".
5. **מודול מערכת** - הרשימה הנפתחת תשתנה בהתאם למערכת שנבחרה:
   * אם נבחרה מערכת 1: האפשרויות יהיו "מפה", "התראות", "מסננים".
   * אם נבחרה מערכת 2: האפשרויות יהיו "מפה 2", "התראות 2", "מסננים 2".
   * וכן הלאה לכל 10 המערכות.
6. **האם התקלה משתחררת בגרסה הבאה?** - שדה בחירה (כן/לא). יוצג **רק אם נבחרה תקלה**.
7. **תנהגות דינמית בטופס (Best Practices)**

* **הצגת שדות חופשיים בהתאם לסיבת הפנייה:**
  + אם נבחרה "תקלה" יוצג שדה "תיאור שחזור התקלה".
  + אם נבחרה "בקשת הרשאות" יוצג שדה "הכנס בקשה מפורטת לקבלת הרשאה".
  + אם נבחרה "הצעת ייעול" יוצג שדה "הכנס את הצעת הייעול בצורה מפורטת".
  + במידה והמשתמש משנה את סוג הפנייה - השדה החופשי יתעדכן בהתאם **ורק הערך הרלוונטי יישמר**.
* **סינון דינמי של מודול מערכת:**
  + הבחירות במודול יוצגו **בהתאם למערכת שנבחרה**.
  + ברירת מחדל - "הכל".

### 3.2. ממשק ניהול (Backend)

ממשק הניהול יתבסס על Node.js (Express.js) ויכלול:

• API לניהול קונפיגורציה דינמית (טעינת סוגי תקלות ומודולים מתוך קובץ JSON)

• API לשמירת דיווחי תקלות

• API להורדת קובץ Excel של הדיווחים

• אימות ובקרת גישה לממשק ניהול (בשלבים מתקדמים)

## 4. מבנה הנתונים

### 4.1. קובץ קונפיגורציה (config.json)

{  
 "apiBaseUrl": "https://defectreport.onrender.com",  
 "issueTypes": [  
 "תקלה קריטית",  
 "בעיה בהתחברות",  
 "תקלה בביצועים",  
 "חוסר אמינות",  
 "באגים אסתטיים",  
 "שגיאת מערכת"  
 ],  
 "modules": [  
 "דוגמה א",  
 "דוגמה ב",  
 "דוגמה ג",  
 "דוגמה ד",  
 "דוגמה ה"  
 ]  
}

## 5. API Backend

להלן רשימת הנתיבים הקיימים במערכת:

|  |  |  |
| --- | --- | --- |
| מתודה | נתיב | תיאור |
| GET | /config | שליפת קובץ הקונפיגורציה |
| POST | /submitBugReport | שמירת דיווח תקלה חדש |
| GET | /downloadExcel | הורדת כל הדיווחים בקובץ Excel |

## 6. שיקולים טכניים

• ניהול דינמי של קונפיגורציה דרך config.json

• תמיכה בדפדפנים מודרניים Chrome, Edge, Firefox, Safari

• אבטחת נתונים למניעת XSS ו-SQL Injection

• תמיכה בגישה מניידים באמצעות Responsive Design