**דרישה עסקית – מערכת חכמה להתאמת מתנדבים לפעילויות התנדבות**

**רקע:**

בעולם ההתנדבות קיימים פערים משמעותיים בין הצעות ההתנדבות הקיימות לבין זמינות, תחומי עניין, וכישורי המתנדבים בפועל. מתנדבים רבים מתקשים למצוא פעילות מתאימה, ומוסדות קהילתיים מתקשים לאייש משימות באופן אפקטיבי. מטרת המערכת היא לגשר על הפער הזה ולבצע התאמה חכמה ומדויקת בין מתנדבים לפעילויות התנדבות, תוך שיפור שביעות הרצון ויעילות המענה לצרכים בשטח.

**מטרה עסקית:**

לאפשר **התאמה מיטבית בין מתנדבים להזדמנויות התנדבות**, על בסיס פרופיל אישי, זמינות, תחומי עניין, כישורים, ומרחק גיאוגרפי, כך ש:

* המתנדב ירגיש מוערך ויעשה שימוש ביכולותיו במקום מתאים.
* הארגון המארח יקבל את האדם המתאים ביותר לצרכים הקיימים.
* יוגדל שיעור ההתמדה, המוטיבציה והתרומה בפועל.

**צרכים עסקיים עיקריים:**

1. **יצירת פרופיל מתנדב**
   * פרטים אישיים, כתובת, גיל
   * תחומי עניין (למשל: ילדים, קשישים, בעלי חיים, סביבה וכו’)
   * כישורים וניסיון רלוונטי (שפות, הדרכה, טכנולוגיה, עזרה ראשונה ועוד)
   * זמינות (ימים, שעות, משך מועדף)
   * העדפות (מרחק מקסימלי, סוג פעילות מועדפת, עבודה לבד/בקבוצה)
2. **הגדרת פרופיל פעילות התנדבות – עתידי – יבוצע באמצעות מילוי טבלאות ניהול פעילות התנדבות**
   * שם הארגון ופרטי קשר
   * תחום פעילות וסוג משימה
   * דרישות (כישורים חובה/רצוי, גיל מינימלי, שפה)
   * מועדים וזמני הפעילות
   * מיקום גיאוגרפי
3. **מנגנון התאמה חכם**
   * אלגוריתם שישקלל את כלל הנתונים ויציג הצעות מותאמות אישית
   * אפשרות לניקוד / דירוג רמת התאמה לכל פעילות
   * תמיכה בהתאמה חד-פעמית או קבועה
4. **עתידי- מערכת ניהול ובקרה**
   * עבור מתנדבים: ממשק לבחירה, אישור, שינוי או ביטול
   * עבור ארגונים: אפשרות לאישור מתנדבים, תיעוד היסטוריית התאמות
   * לוג פעילות לצורכי מדידה ושיפור

**מדדי הצלחה:**

* אחוז גידול במספר ההתאמות המוצלחות (התנדבות שהתקיימה בפועל)
* זמן ממוצע להתאמה מרגע ההרשמה
* שביעות רצון מתנדבים וארגונים (נמדדת בסקר תקופתי)
* אחוז שימור מתנדבים לחודש/רבעון שני

**שיקולים טכנולוגיים (רקע לאפיון עתידי):**

* שימוש ב-AI/ML לשיפור התאמות לאורך זמן - עתידי
* חוויית משתמש נגישה גם למשתמשים לא טכנולוגיים
* תמיכה במסכים ניידים, אינטגרציה עם WhatsApp/SMS – אינטגרציה ל-whatsapp בהמשך
* אבטחת מידע ועמידה ברגולציה – אבטחה בסיסית כך שיהיה ניתן ליישם במסגרת זמן מוגבלת של פרויקט גמר

מה שיושם בשלב זה ב-aws , יש להתחשב בכך בשלב הבא של האפיון במידה והטקסט אינו: כאן תשתלו את מה שהסקריפט החזיר – נקו את התוצאות שיהיו רק המשאבים הרלוונטיים. אלא טקסט של משאבים.

[ כאן תשתלו את מה שהסקריפט החזיר – נקו את התוצאות שיהיו רק המשאבים הרלוונטיים]

יש להציג תרחישי Use Case למערכת

**“מתנדבים חכם”**

**תרחישי Use Case עיקריים:**

**1.**

**רישום מתנדב חדש**

**Actor:** מתנדב

**Trigger:** רצון להצטרף לפעילות התנדבותית

**Steps:**

1. פתיחת טופס רישום באתר.
2. הזנת פרטים אישיים: שם, גיל, כתובת.
3. בחירת תחומי עניין.
4. סימון כישורים רלוונטיים.
5. הזנת זמינות ושעות מועדפות.
6. הגדרת העדפות נוספות: מרחק גיאוגרפי, סוג פעילות, עבודה לבד/בקבוצה.
7. שליחה ושמירה של הפרופיל למערכת.

**Outcome:** פרופיל מתנדב נשמר במערכת ונשלח לאלגוריתם ההתאמה.

**2.**

**רישום פעילות התנדבות – יתבצע בטבלה שצריך ליצור ב-dynamodb ללא יישום המפורט מטה**

====לא ליישום====

**Actor:** רכז/ת בארגון

**Trigger:** פתיחת משימה חדשה

**Steps:**

1. כניסה לפורטל הארגונים.
2. הזנת פרטי הפעילות: שם, מיקום, סוג, תאריכים.
3. הגדרת דרישות (כישורים, גיל, שפה).
4. בחירת זמן הפעילות וימי ההתנדבות האפשריים.
5. שליחה לאישור ולפרסום.

**Outcome:** פעילות נשמרת בבסיס הנתונים וממתינה להתאמות מתנדבים.

====סוף החלק לא ליישום====

**3.**

**התאמה חכמה בין מתנדבים לפעילויות**

**שם התרחיש:** התאמה חכמה של מתנדב לפעילות והתנדבות יזומה

**שחקנים ראשיים:** מתנדב

**שחקנים משניים:** אין

**תנאי מקדים:** המתנדב השלים רישום מלא למערכת, כולל פרופיל אישי מעודכן

**תוצאה נדרשת:** המתנדב מצטרף לפעילות רלוונטית בהתאם לפרופיל האישי שלו

**מהלך התרחיש:**

1. המתנדב ניגש למערכת באמצעות אתר האינטרנט.
2. לאחר אימות זהות, המתנדב מקבל רשימת פעילויות התנדבות הממויינות לפי רמת התאמה לפרופיל האישי שלו.
3. ההתאמה מבוססת על השוואה בין נתוני הפרופיל של המתנדב לבין מאפייני הפעילויות הזמינות, לרבות:
   * תחומי עניין
   * זמינות: ימים, שעות, טווח
   * מיקום גיאוגרפי מועדף
   * כישורים וניסיון רלוונטי
   * העדפות כגון אופי הפעילות: יחיד/קבוצתי
4. לכל פעילות ברשימה מוצגים פרטים רלוונטיים: שם הארגון, תחום הפעילות, תיאור המשימה, מועדים, מיקום פיזי, רמת התאמה.
5. המתנדב בוחר את אחת מהפעילויות ומאשר את הצטרפותו אליה באמצעות לחיצה על כפתור “הצטרפות” או פעולה דומה.
6. בקשת ההצטרפות מתועדת ומסומנת כ”ממתינה לאישור” מצד הארגון הרלוונטי.

**Postconditions:**

* הפעילות מסומנת כפעילות נבחרת בפרופיל המתנדב.
* רשומת ההתאמה מתווספת ללוג הפעילות לצורכי בקרה ומעקב.
* הארגון מקבל התראה עם פרטי הבקשה לצורך אישור/דחייה.
* עד לאישור סופי, הפעילות מסומנת כ”מוקצת זמנית” עבור אותו מתנדב.

**Exceptions:**

* במידה ולא קיימות כלל התאמות מתאימות, תוצג הודעה רלוונטית עם אפשרות לעדכן את פרטי הפרופיל.
* במידה והמתנדב בחר פעילות אך שונתה זמינותה או דרישותיה בזמן אמת – תופיע הודעה מתאימה והמתנדב יתבקש לבחור מחדש.

**הערות:**

* בשלב עתידי ניתן יהיה ליישם דירוג התאמות וסינון חכם לפי העדפות אישיות של המתנדב (למשל, הדגשת פעילויות עם השפעה קהילתית גבוהה).
* כלל הפעולות יתועדו לצורכי מדידה, שיפור איכות ההתאמה ומעקב שוטף.

**5.  עתידי – לא ליישום**

**אישור מתנדב על ידי הארגון**

**Actor:** רכז/ת בארגון

**Trigger:** קבלת בקשה להצטרפות מתנדב

**Steps:**

1. צפייה בפרופיל מתנדב.
2. אישור או דחיית הבקשה.
3. שליחת התראה למתנדב על סטטוס ההתאמה.

**Outcome:** התחייבות ההתנדבות מתבצעת, נרשמת היסטוריית התאמה.

**6.**

**עתידי – לא ליישום - ביטול/עדכון ע”י מתנדב או רכז**

**Actor:** מתנדב / רכז בארגון

**Trigger:** שינוי זמינות או ביטול פעילות

**Steps:**

1. כניסה לממשק אישי.
2. בחירה בפעילות.
3. עדכון או ביטול.
4. שליחת התראה לצד השני.

**Outcome:** ביטול ההתאמה, שחרור המשבצת לפעילות נוספת.

**7.**

**מדידת הצלחה ובקרה**

**Actor:** מנהל מערכת

**Trigger:** תקופה מוגדרת (חודש/רבעון)

**Steps:**

1. שליפת דוחות מערכת.
2. ניתוח מדדים:
   * כמות התאמות מוצלחות
   * זמן ממוצע להתאמה
   * אחוז שימור מתנדבים
   * שביעות רצון ממשובים

**Outcome:** קבלת המלצות לשיפור האלגוריתם וחוויית המשתמש.

אם תרצה, אפשר להרחיב את זה לתרשימי זרימה (Flow Diagrams), מסכים לדוגמה, או להמשיך לאפיון טכני/פונקציונלי מלא לכל תרחיש.

שלב הפיתוח

הנחיות: לוקחים רק תרחיש אחד בלבד ומזינים למערכת – פרטים נוספים באחריות שלכם לשבץ

יש לבצע את החלק של האפיון תוך שימוש בשירותי severless של aws , יש לכתוב תכנון ארכיטקטורה ולאחר אישור יש לבצע את הקוד ברכיבים השונים.

**תרחישי Use Case עיקריים:**

**רישום מתנדב חדש**

**Actor:** מתנדב

**Trigger:** רצון להצטרף לפעילות התנדבותית

**Steps:**

1. פתיחת טופס רישום באתר.
2. הזנת פרטים אישיים: שם, גיל, כתובת.
3. בחירת תחומי עניין.
4. סימון כישורים רלוונטיים.
5. הזנת זמינות ושעות מועדפות.
6. הגדרת העדפות נוספות: מרחק גיאוגרפי, סוג פעילות, עבודה לבד/בקבוצה.
7. שליחה ושמירה של הפרופיל למערכת.

**Outcome:** פרופיל מתנדב נשמר במערכת ונשלח לאלגוריתם ההתאמה.

=== התוצר שהתקבל והתיקונים

על בסיס הארכיטקטורה יש לבנות את ה-frontend

## ✅ ארכיטקטורת מערכת לרישום מתנדב חדש – AWS Serverless

### 🎯 מטרות התהליך:

* רישום מתנדבים חדשים בצורה מאובטחת
* אחסון פרופיל מתנדב בצורה נגישה לאלגוריתם ההתאמה
* שימוש מלא ב-AWS Serverless: ניהול תעבורה, פונקציות חכמות, ולא שמירת שרתים

## רכיבי הארכיטקטורה:

### 1.

### Frontend (טופס רישום):

* React / Vue / HTML + JS
* מתארח ב־**Amazon S3 + CloudFront** כ־static website
* משתמש ב־**Amazon Cognito** לאימות – יש ליצור user pool

### 2.

### API Gateway (HTTP API):

* Endpoint: POST /volunteer
* מקבל את טופס הרישום בפורמט JSON
* מחובר ל־Lambda (ראו להלן)

### 3.

### AWS Lambda – SaveVolunteerProfile

* בודק שהמשתמש מאומת – יש טוקן JWT
* מקבלת את הנתונים מה-API
* מבצעת ולידציה בסיסית
* ממירה את הנתונים למבנה סטנדרטי
* שומרת ל־DynamoDB
* שולחת את הנתונים ל־SNS Topic / EventBridge לאלגוריתם ההתאמה

### 4.

### Amazon DynamoDB – VolunteerProfiles

* טבלת מתנדבים עם שדות:
  + VolunteerId (Hash Key)
  + Name, Age, Address
  + Interests [list]
  + Skills [list]
  + Availability [object]
  + Preferences [object]
  + CreatedAt (timestamp)

### 5.

### Amazon SNS

* נשלחת הודעה עם VolunteerId לאחר רישום
* הטריגר יכול להפעיל שירות התאמה חכם בהמשך

### 6.

### CloudWatch Logs + Metrics

* ניתוח ביצועים, תקלות, ותיעוד