



# SMARTTRACK

מחלקת הנדסת תוכנה במכללה האקדמית להנדסה ע"ש סמי שמעון  
- אפריל 2022 -

מגישות: הודיה תורג'מן והילה פינקר | מנחה: חבצלת כהן



## תוכן עניינים

3	תקציר הפרויקט
4	סקר ספרות
4	הקדמה
4	למה לתכנן מסלול טיול מראש?
4	הקשיים בתכנון טיול
5	איך אתם מתכננים את החופשה הבאה שלכם?
6	מערכת המלצות
7	תכנון מסלולים יומיים - אלגוריתם k means clustering
8	סקר שוק
9	סיכום
10	מסמך אפיון - SRS
10	הקדמה
10	מטרה
10	קהל יעד
10	היקף המוצר
11	תיאור כללי
11	נקודת מבט המוצר
11	פונקציות המוצר
11	משתמשים ולקוחות
12	סביבת הפעלה
12	אילוצי עיצוב ויישום
12	ממשקים
12	ממשק מערכת
12	ממשק תוכנה
13	ממשק תקשורת
13	ממשק חומרה
13	ממשק משתמש
13	מאפייני המערכת:
13	דרישות חומרה
13	דרישות התקנה
13	דרישות-צד לקוח (האפליקציה)
17	דרישות - צד שרת
18	דרישות לא פונקציונליות
19	תהליך הפיתוח ואתגרים
19	תהליך עבודה
20	קשיים ואתגרים
22	אלגוריתם מערכת המלצות
22	הקדמה
22	מימוש האלגוריתם

	בס"ד
24	דוגמת הרצה
26	אלגוריתם תכנון טיול
26	הקדמה
26	אלגוריתם same size k-means clustering
26	מימוש האלגוריתם
27	מוקאפ
32	בסיס נתונים
33	פיתוח עתידי
34	ביבליוגרפיה

## תקציר הפרויקט

בפרויקט זה פיתחנו אפליקציה המשמשת לתכנון טיולים ומבוססת על מערכת המלצות ועל העדפות המשתמש לבניית טיול, המורכב מנקודות עניין מומלצות למשתמש. האפליקציה עוזרת למשתמש לצמצם את זמן תכנון הטיול, פוטרת אותו מבדיקת מיקומים של אטרקציות וחישובי מרחק ביניהן, מתכננת מסלול אופטימלי לפי מספר ימי הטיול ומציעה מסלולים יומיים המורכבים מאתרים קרובים ככל האפשר אחד לשני על מנת לחסוך בזמן הנסיעה.

בהרשמה המשתמש מזין פרטים אישיים ואת העדפותיו לגבי סוגי האטרקציות. לצורך תכנון טיול, המשתמש מזין את יעד הטיול ומספר הימים. האפליקציה מציגה למשתמש על ידי מערכת המלצה היברידית את רשימת האתרים המומלצים עבורו לפי הפרופיל האישי ולפי משתמשים דומים בהסתמך על דירוגים קודמים של המשתמש. המשתמש בוחר מתוך הרשימה את אתרי התיירות המועדפים עליו, כ 3 אתרים לכל יום, לאחר מכן האפליקציה מחלקת את הרשימה למספר ימי הטיול באופן אופטימלי לפי אלגוריתם `same size k means clustering` כך שהאתרים יהיו קרובים אחד לשני מה שמאפשר חיסכון בזמן ומניעת נסיעות מיותרות בין קצוות העיר. בסיום כל יום המשתמש ידרג את האתרים שביקר בהם לצורך שיפור מערך ההמלצות לטיולים עתידיים של המשתמש.

האפליקציה Smartrack אידיאלית עבור אנשים שרוצים לתכנן את הטיול במהירות ובקלות. היא עוזרת בהפחתת עומס העבודה על התכנון, מקלה על קבלת ההחלטות ומארגנת עבור המשתמש את הטיול החוויתי הבא.

## 2. סקר ספרות

### הקדמה

בשנים האחרונות עם שיפור רמת החיים, תהליכי גלובליזציה והתקדמות טכנולוגיית המידע, התיירות צמחה במהירות והפכה לאופנת חיים חדשה בעלת משמעות כלכלית, חברתית ותרבותית המהווה אחד התחומים הרווחיים של הכלכלה העולמית. על פי ארגון התיירות העולמי של UNWTO, המספר הכולל של טיולי תיירות בינלאומיים וההכנסה מתיירות בינלאומית גדלים כל הזמן. כאשר בשנת 2019 מספר התיירים חצה את רף ה-1.5 מיליארד [1]. השימוש בטכנולוגיות חכמות כגון אתרי אינטרנט הקשורים לטיולים, מדיה חברתית וסמארטפונים בתכנון נסיעות הולך וגדל והמונח תיירות חכמה הפך למונח נפוץ לתיאור השימוש בטכנולוגיות שונות לנסיעות [2]. עבור אנשים רבים, התיירות מספקת הזדמנות לבצע פעילויות פנאי, לחקור תרבויות חדשות, להכיר אנשים ומקומות, לבלות, לנוח ולהתנסות בחוויות מרגשות. בעבר, מטיילים תכננו טיולים באמצעים מסורתיים לאיסוף מידע כגון מדריכי טיולים, ספרים, חברים וסוכני נסיעות אך בשנים האחרונות תיירים מסתמכים במידה רבה על פלטפורמות מקוונות לתכנון טיולים [3] ומטיילים פונים יותר ויותר למערך מגוון של טכנולוגיות וערוצים כדי לקבל מידע מדויק, עשיר, מקיף ומותאם אישית משלב התכנון של הנסיעה ועד לחוויה בזמן אמת כאשר המכשירים הניידים משמשים כפלטפורמה ראשית לגישה למידע [2].

### למה לתכנן מסלול טיול מראש?

- ◀ כדי להגיע למקומות הכי מעניינים ויפים בהתאם להעדפות- לא כל דבר מעניין את כולם.
- יש שרוצים לראות רק טבע ונופים, ויש שרוצים תרבות ואוכל ומוזיקה ובילויים כל לילה, יש שמחפשים רק קניות ויש שרק רוצים להסתובב ברחובות ולספוג את התרבות המקומית.
- ◀ כדי לא לבזבז זמן במהלך הטיול- הטיול עולה המון כסף והזמן בדרך כלל מוגבל, אי אפשר לבקר בכל אתר תיירותי ביעד הטיול בשל אילוצי זמן.
- ◀
- ◀ כדי ליהנות יותר- מטרת העל של הטיול, בעצם... על ידי תכנון מראש פנויים להתמקד בחוויות עצמן בכל רגע בטיול ואין התעסקות בתכנון הטיול [4]

תהליך התכנון לוקח זמן ומצריך איסוף מידע וסינון שלו [4]. זהו שלב חשוב וברוב המקרים אף ארוך מהטיול עצמו. תכנון מוקדם יעשה את ההבדל בין ניצול נכון של הימים לבזבזם, בין טיול מאוזן תקציבית לטיול עם הוצאות שלא תכננתם והכי חשוב – ההבדל בין הנאה מהטיול לבין אכזבה מחוסר התאמה לציפיות.

### הקשיים בתכנון טיול

תיירים המבקרים בעיר במהלך טיול עם זמן מוגבל, מוצאים שזה בלתי אפשרי לבקר בכל נקודות העניין הקיימות באזור, לכן עליהם לבחור מספר נקודות עניין שהם מחשיבים מעניינות ומסקרנות עבורם. הבעיה היא סינון כמויות המידע התיירותי שיש באינטרנט. התיירות היא ענף עתיר מידע, רשימת האפשרויות שמציעים באתרי תיירות ובמנועי החיפוש עלולה להיות אינסופית ותוצאות רלוונטיות בדרך כלל טובעות ב"רעש" מידע שמונע, או לפחות מאט את תהליך הבחירה. הכמות

העצומה של מידע תיירותי מקוון הפכה את הכנת הטיול למאתגרת מאוד [4] ולכן, כדי לסייע לתיירים בתכנון טיול ולעזור להם למצוא את המידע שהם מחפשים, יש צורך במערכת המלצה אישית שתסכן את כמויות המידע ותציע למטייל אתרי תיירות המתאימים לו [5], [6]. לאחר הרכבת רשימת נקודות עניין שבה המטייל ירצה לבקר, יש צורך בתכנון מפורט של לוחות זמנים וחלוקה של האתרים לימים מסודרים עם התחשבות בשעות הפתיחה והסגירה של כל אתר ואתר על מנת למנוע עוגמת נפש [4].

אתגר נוסף ועיקרי הוא תכנון מסלול יומי אופטימלי בהתבסס על המרחק בין האתרים המתוכננים לאותו יום. תיירים מבקרים עיר לפרק זמן קצוב ולכן רוצים להספיק כמה שיותר מקומות בכמה שפחות זמן. ניצול זמן יעיל הוא הכרחי בתכנון טיול והעדיפות היא לבקר אתרים הקרובים אחד לשני בכל יום על מנת למנוע זמן נסיעה מיותר. כדי לתכנן זאת יש צורך לזהות אילו אטרקציות קרובות אחת לשניה ולהרכיב תוכנית טיול המצמצמת עד כמה שאפשר נסיעה לאורך ולרוחב העיר, מקצה אחד לשני, ביום מסוים [4].

### הפתרון-האפליקציה Smartrack

מטרת הפרויקט היא פיתוח אפליקציה נוחה וידידותית למשתמש לתכנון מסלול טיול אופטימלי ליעד מסוים על פי המלצות והעדפות המשתמש.

### איך אתם מתכננים את החופשה הבאה שלכם?

דמיינו לעצמכם את התרחיש הבא:

אתם רוצים לטוס לחופשה מהנה ומפנקת בקיץ, ליעד בו לא ביקרתם מעולם. ניצבות בפניכם מספר אפשרויות:

אפשרות 1: טיול מאורגן. טיולים מאורגנים הם פתרון נהדר למי מאיתנו שאין לו זמן או עניין בתכנון כל פרט ופרט בטיול בכוחות עצמו. הרשמה לטיול מאורגן היא דרך חסכונית בזמן להכיר ולטייל עם אנשים בעלי תחומי עניין משותפים ומונעת צורך בארגון ותכנון. מצד שני- טיולים מאורגנים בדרך כלל יקרים יותר משמעותית, אתה תלוי בלוח זמנים קבוע מראש ובאנשים אחרים ולא תמיד מבקר באתרים שמעניינים אותך [4].

אפשרות 2: לא לתכנן... אפשר לטוס לחו"ל בלי תכנון, ו-'לזרום' בשטח עם ספר הדרכה למטיילים, או לשאול את הפקידים במלון מה קורה בסביבה, מהן ההמלצות שלהם והאם יש לבית המלון קופונים לאטרקציות. זה מתאים לחלקנו אבל לא לכל יעד, וזה גם מעלה את הסיכוי לבזבז הזמן היקר שלכם בחו"ל.

אפשרות 3: "עשה זאת בעצמך": רושמים את כל המקומות בהם רוצים לבקר -> מעלים על מפה -> רואים מה סדר הטיול ההגיוני -> בודקים מרחקים מיעד ליעד וכן את נקודת ההתחלה ונקודת

הסיום -> בניית שלד יומי לטיול. אחרי שבונים שלד הגיוני לטיול, הגיע הזמן להיכנס לפרטים הקטנים של המסלול: מה יש לעשות בכל מקום? כמה זמן להקציב לכל פעילות/אטרקציה? מהם ימי הפעילות ושעות הפעילות באתר? מה עושים בדרך? ועוד... את המידע שנאסף מגבשים למסלול יומי מפורט הלוקח בחשבון הערכת זמנים של ביקור בכל אתר או זמן נסיעה. לתכנן מסלול טיול עצמאית זה כיף ונותן סיפוק, אבל גם לוקח הרבה זמן...יש צורך בשימת לב לפרטים הקטנים ובחשיבה רבה [4].

## אפשרות 4: האפליקציה Smartrack

## מערכת המלצות

מערכות המלצה הן כלי סינון מידע שניתן להשתמש בו על מנת למצוא פריטי עניין של משתמש. מערכות אלו נפוצות כיום בתחומים רבים ומבוססות על שיטות שונות. הן מסננות את המידע הרב שיש באינטרנט ומציגות מידע רלוונטי למשתמש וכך חוסכות למשתמש זמן. המשימה החשובה ביותר של מערכת המלצות היא להעריך את תחומי העניין של המשתמש לגבי פריטים במערכת. לשם כך, דעות המשתמשים נרכשות במפורש או במשתמע כדי למצוא פריטים רלוונטיים. בנוסף, מערכות אלו יכולות להשתמש במשאבי מידע נוספים כגון נתונים דמוגרפיים, היסטורית גלישה ומידע חברתי [5].

## סוגי מערכות ההמלצה:

- מערכות המלצה מבוססות ידע (knowledge-based) ממליצות על פריטים בהתבסס על ההעדפות והצרכים שהמשתמש מספק.
- מערכות המלצה מבוססות תוכן (content-based) ממליצות על פריטים על סמך היסטוריית הדירוג של המשתמש. הן משתמשות בתכונות פריטים כדי להמליץ על פריטים אחרים הדומים למה שהמשתמש אוהב, על סמך פעולות קודמות של המשתמש או משוב מפורש. המערכת לא צריכה מידע על משתמשים אחרים אך מכיון שעובדת רק על סמך תחומי העניין הקיימים של המשתמש, יש לה יכולת מוגבלת להרחיב את תחומי העניין של המשתמשים.
- מערכות המלצה מבוססות שיתופיות (collaborative-filtering) ממליצות על פריטים פופולריים בקרב משתמשים אחרים בעלי תחומי עניין דומים. גישה זו ממליצה על הפריטים הרלוונטיים למשתמשים בהתבסס על דעותיהם של המשתמשים האחרים שנמצאים בקבוצת השכנים שלהם. קבוצת שכנים זו נוצרת על סמך ערכי הדמיון בין משתמשים ולאחר מכן בחירת המשתמשים בעלי ערכי הדמיון הגבוהים יותר עם המשתמש. המערכת יכולה לעזור למשתמשים לגלות תחומי עניין חדשים על סמך משתמשים דומים אך יש לה את בעיית cold start-המלצה על פריט חדש או עבור משתמש חדש שאין עליהם מספיק מידע. בעיה זו מובילה להפחתת ביצועים של המערכת. כאשר משתמש חדש מצטרף למערכת יש לו פרופיל דירוג ריק או קטן מאוד ולכן המערכת לא יכולה לחשב במדויק את ערכי הדמיון בין משתמש זה למשתמשים אחרים. יתרה מכך, זיהוי מערך השכנים של המשתמשים הוא תהליך קשה המוביל להפחתת אמינות ההמלצות [5].
- מערכות ממליצים היברידיים מכוונות לשלב אלמנטים משתי טכניקות המלצות או יותר כדי לצמצם את החולשות של כל אחת מהן ולהגיע לתוצאות טובות ומדויקות יותר [6].

### שימוש בפרויקט

לאחרונה מערכות המלצות מותאמות אישית צברו עניין גם בתחום התיירות כדי לסייע לנוסעים עם תוכניות נסיעה. ישנן שיטות רבות להערכת אתרי התיירות המועדפים על מטיילים וניתן לסווג שיטות אלו לשתי קבוצות: האחת מבוססת על התנהגות נסיעות בעבר של המשתמשים (גישה מבוססת תוכן), והשנייה מבוססת על התנהגות נסיעה של קבוצת מטיילים בעלי מאפיינים דומים למטייל (גישה סינון שיתופי).

בחרנו להשתמש במערכת היברידית עם סינון שיתופי המבוססת על הרחבת פרופיל המשתמש כדי להקל על בעיית ההתחלה הקרה.

בבנה רשימת אתרי תיירות מומלצים למשתמש על סמך Collaborative filtering. מכיון שאין לנו ידע על משתמשים חדשים ואתרי תיירות חדשים, ברישום משתמש חדש נציג שאלון ובו נשאל את המשתמש לגבי העדפותיו ופרטיו האישיים ונשתמש בידע הזה על מנת להמליץ לו על אתרי תיירות [5].

נבקש מהמשתמש לדרג את האתרים שביקר בהם ונשתמש בדירוג זה כדי לעדכן באופן מרומז את העדפות המשתמש. מכיון שאנחנו רוצים להימנע מהטיות ורמאות בהמלצות, לא נכריח את המשתמשים לדרג את האתרים כדי לא לקבל מידע מוטעה [6].

### תכנון מסלולים יומיים - אלגוריתם k means clustering

המשתמש בוחר מספר מוגבל של אתרי תיירות שהוא רוצה לבקר בהם מתוך רשימת ההמלצות, לאחר מכן המערכת תשבץ מקומות הקרובים זה לזה באותו יום כדי לסייע בהפחתת זמן הנסיעה בין האתרים וזאת על ידי שימוש באלגוריתם same size k-means clustering. האלגוריתם יארגן עבור התיירים את מסלול הטיול שישבץ את המקומות הקרובים ביותר על סמך המיקום הגיאוגרפי שלהם, מקסימום 3 אתרים בכל יום ויחסוך להם זמן הן בתכנון מסלול עבור כל יום והן במעבר בפועל ממקום למקום.

k-means clustering הוא אחד האלגוריתמים הנפוצים ביותר בכריית נתונים והוא פשוט יחסית ליישום ומתבסס בכל הנוגע לחלוקת מערך נתונים למספר אשכולות,  $k$ , כמפורט על ידי המשתמש ובמקרה שלנו- מספר ימי הטיול. K-means משתמש במרחק אוקלידי כדי למדוד את המרחק בין נקודות הנתונים במערך נתונים. מרחק אוקלידי נמדד על ידי חישוב המרחק בקו הישר בין שתי נקודות. אלגוריתם ה-k-means מתכוון למזער את המרחק האוקלידי בין נקודות הנתונים לבין נציג האשכולות הקרוב ביותר שלהן (centroid) ומבטיח שכל נקודת נתונים שייכת לאשכול אחד בדיוק ובמקרה שלנו- כל אתר תיירותי ישובץ ביום אחד בלבד יחד עם אתרים קרובים [4]. אתגר שעלה במהלך החשיבה על אלגוריתם k means clustering הוא חלוקת האתרים לימים בצורה הגיונית כך שהמשתמש יוכל להספיק לבקר בכל יום בכל האתרים שתוכננו ליום זה.

יום בטיול הוא אינו 24 שעות רצופות של פעילות. הוא כולל גם שינה, התארגנות, אכילה והפוגה קצרה בין האתרים. האלגוריתם בצורתו המקורית מחלק את האתרים על סמך מיקום בלבד ועלול להוביל לחלוקה לא הוגנת של אתרים בין הימים.

כדי להתגבר על כך, נשתמש בגרסה מתקדמת יותר של האלגוריתם, same size k-means clustering שמחלק את האשכולות לגודל שווה. האתרים יחולקו בצורה שווה בין מרכזי הכובד על ידי הגדרת הגודל המקסימלי של כל קבוצה כך שבחלוקה ל clustering הגודל של כל קבוצה ילקח בחשבון ביחד עם חישוב המרחקים בין הנקודות [7].



## סקר שוק

שוק האפליקציות בתחום התיירות הוא גדול ורחב ומתפתח עם השנים. ישנן קטגוריות שונות של אפליקציות אפליקציות להשוואת מחירים כגון skyscanner, trivago אפליקציות להזמנת מקומות לינה כגון booking, airBNb אפליקציות מידע על אתרי תיירות כגון tripAdvisor, yelp אפליקציות לתכנון מסלול כגון visitAcity, trip hobo אפליקציות למזג אוויר, תרגום, תחבורה ציבורית, המרת מטבע ועוד ועוד.

## מוצרים דומים בשוק:

במהלך סקירת שוק האפליקציות בתחום התיירות מצאנו שישנן אפליקציות לתכנון מסלולים בדומה למוצר שלנו והבולטות מביניהן:

- visit a city  
אפליקציה ליצירת מדריך טיולים אישי. מציגה מדריכים מוכנים עבור רשימה מוגבלת של יעדים. את המדריכים ניתן להתאים, לערוך ולשנות. עבור כל אטרקציה מוצגים פרטים כגון כתובת, טלפון, שעות פתיחה, ביקורות ודירוג אך המסלולים והאתרים המוצעים לא מותאמים אישית, אין מערכת המלצות אישית לכל משתמש אלא מספר מדריכים מוכנים מראש אותם אפשר לשנות.
- wanderlog  
אפליקציה לתכנון טיול. ישנם מדריכים מוכנים של טיולים ויש אפשרות לתכנן מסלול לבד. מציגים מפה של היעד ואת תכנון הטיול- עבור כל יום ניתן לבחור אתרים והם מוצגים בצורה בה רואים את המרחק והזמן מאתר לאתר. ניתן לשתף בתכנון מטיילים נוספים אך התכנון עצמו ושיבוץ האתרים בימים נעשה על ידי המשתמש כך שהוא עלול להיות לא אופטימלי.
- route perfect  
מציג עבור יעד מסוים ומספר ימים מסוים מינימלי מסלולי טיול מוכנים על סמך העדפות שונות ומדריכי טיולים. המסלולים מתאימים יותר עבור טיולים ארוכים למדינה שבה רוצים לבקר במספר ערים ולא להתמקד רק בעיר אחת.
- trip hobo  
מתחילים בשאלון בו שואלים מה היעד, מה התאריכים, שעות נחיתה והמראה, סוג טיול- עם מי נוסעים, גילאים ואז מוצע תכנון טיול ובו בכל יום משובצים בין 2 ל 4 אתרי תיירות פופולריים הניתנים לשינוי. אבל אין מפה שמסמנת כל מקום והתכנון לא מבוסס על המלצות משתמש.
- tripAdvisor  
האפליקציה נותנת אפשרות לסמן "אהבתי" על אתרים ולקבץ אותם בתיקיות לפי יעד הטיול. בתכנון טיול יש אפשרות לבחור מספר ימים/ תאריכים ואז לשבץ את האתרים בימים אלו. מראה מפה בה האתרים מסומנים כדי להקל על התכנון. הבעיה- ניתן להוסיף אתר מסוים ליותר מיום אחד. אין תכנון של הטיול אלא המשתמש מתכנן בעצמו.

ההבדל בין smartrack לבין רוב האפליקציות הוא שהן מציעות מסלולים מוכנים מראש עבור כל יום. אם רוצים לשנות אטרקציה מוצעת לחלופה אחרת, לא תמיד יתחשבו במרחק. כך יכול להיווצר מצב בו יהיו 2 אתרים קרובים שבשל התכנון מראש ישובצו ב 2 ימים שונים. כמו כן לשנות משהו קיים דורש יותר יצירתיות כי כשרואים מסלול מוכן מראש קשה יותר לשנות אותו. בפרויקט שלנו לעומת זאת המשתמש בוחר בעצמו את המקומות בהם הוא רוצה לבקר מתוך מקומות המומלצים לו אישית על סמך הפרופיל שלו וטיוליו הקודמים ואז מתוכנן מסלול אופטימלי.

### סיכום

המערכת Smartrack מתכננת מסלולים לכל יום בטיול ליעד מסוים על פי העדפות והמלצות למשתמש בצורה אופטימלית עד כמה שניתן. ישנם שני שלבים עיקריים במערכת:

- מערכת המלצות לבחירת נקודות עניין - לאחר השוואה בין גישות שונות להמלצת נסיעות שניתן להשתמש בהן כדי לפתור את בעיית עומס המידע בענף התיירות, המערכת שלנו תשתמש במערכת המלצה היברידית המבוססת על דמיון בין משתמשים ועל פרופיל המשתמש כדי להציע רשימה של אטרקציות. בניית מערכת המלצה היברידית יכולה להוות אתגר.
- מערכת בניית מסלול עבור כל יום - המשתמש יבחר את האטרקציות לטיול שנוצר ועל סמך בחירתו המערכת תחלק את האתרים למספר ימי הטיול ותקצה מקומות קרובים זה לזה באותו יום כדי לחסוך בנסיעות מיותרות וזמן. חישוב המסלול הוא על ידי אלגוריתם same size k means clustering.

### 3. מסמך אפיון - SRS

#### 3.1. הקדמה

##### 3.1.1. מטרה

מטרת מסמך זה היא להציג תיאור מפורט של הדרישות לבקשה לתכנון טיול שתציע למשתמש מסלולי טיול ביעד מסוים בהתאם לפרק הזמן המוקצב לאותו יעד.

האפליקציה תמליץ למשתמש על אטרקציות תיירותיות ביעד הרצוי בהתאם להעדפותיו ובהתבסס על דמיון למשתמשים אחרים במערכת. היא תתכן מסלולי טיול יומיים תוך התחשבות במיקום האתרים והעדפות המשתמש כדי לחסוך לו זמן תכנון ולעזור לו לבנות טיול אישי אופטימלי. המסמך יסביר את המטרות, התכונות, ממשקי המערכת והאילוצים.

##### 3.1.2. קהל יעד

המסמך מיועד לחברות הצוות שיפתחו את הפרויקט על מנת לספק להם סקירה מפורטת של מטרות הפרויקט ודרישותיו וכן לסגל האקדמי על מנת שיוכלו לעיין בתהליך פיתוח האפליקציה ולוודא את תפקוד המערכת.

##### 3.1.3. היקף המוצר

האפליקציה עוזרת למשתמשים לתכנן את הטיול שלהם. לאחר הזנת היעד ומספר ימי הטיול, המערכת תמליץ למשתמש על רשימת אטרקציות תיירותיות אופציונליות המותאמות להעדפותיו ואהובות על משתמשים דומים לו על סמך נסיעותיו הקודמות. לאחר בחירת מספר מוגבל של אתרים הוא יציע את המסלול האופטימלי לכל יום. האפליקציה תעניק חווית משתמש מותאמת אישית כדי לתת לכל משתמש תחושה שהמסלולים הותאמו במיוחד עבורו. האפליקציה תתקשר עם אתרים חיצוניים לצורך הבאת מידע על אתרי תיירות ומסלולי נסיעה על מנת להקל על מאגרי המידע הפנימיים של המערכת.

## 3.2 תיאור כללי

## 3.2.1 נקודת מבט המוצר

Smartrack היא אפליקציה לתכנון טיול. כל משתמש שנרשם למערכת ימלא פרופיל אישי בו יגדיר העדפות לקטגוריות אתרי התיירות לביקור. בעת תכנון טיול חדש המשתמש יזין את היעד ואת מספר ימי הטיול, האפליקציה תשתמש בגוגל מפות ותייבא מידע על אתרי תיירות ביעד הרצוי בעזרת מערכת המלצות היברידית המשלבת מידע על משתמש ומשתמשים דומים. המערכת תמליץ על אתרים על ידי ניתוח העדפות המשתמש לאתרי תיירות והדירוגים שנתן לאתרים בטיולים הקודמים שלו. הוא יאתר משתמשים דומים - משתמשים שהדירוגים וההעדפות שלהם דומים לדירוגי והעדפות המשתמש ותמליץ על אתרי תיירות שהם העדיפו. ההמלצות יוצגו כרשימה עם מידע על כל אתר תיירות והמשתמש יבחר מספר מוגבל של אתרים בהתאם למספר ימי הטיול. לאחר מכן, האפליקציה תחשב בעזרת אלגוריתם בינה מלאכותית "אותו גודל k-means clustering" חלוקה שווה של האתרים לימי הטיול כך שלכל יום יוצג מסלול אופטימלי הכולל אתרים הקרובים ביותר אחד לשני. מטרת האפליקציה היא לסנן את עומס המידע בענף התיירות עבור המשתמש באופן פרטני, לחשוף אותו לאתרי תיירות מומלצים עבורו, לחסוך לו זמן תכנון מסלולים לכל יום ולמזער את מספר האפליקציות או השירותים השונים שיצטרך לחיפוש מידע לתכנון. האפליקציה מאחדת עבור המשתמש הכל במקום אחד בצורה נוחה.

## 3.2.2 פונקציות המוצר

- הרשמה
- התחברות
- עריכת פרופיל אישי
- תכנון טיול
- לוח זמנים של טיולי המשתמש
- דירוג אתר תיירות

## 3.2.3 משתמשים ולקוחות

קבוצת המשתמשים של המערכת הם לקוחות שרוצים לתכנן לעצמם טיול מהנה. המשתמש צריך לבצע התחברות לאפליקציה כדי לשמור את המידע עליו, את הטיולים ודירוג האטרקציות התיירותיות שלו. כל הפונקציונליות באפליקציה תהיה זמינה לאחר רישום המשתמש ומילוי הפרטים הנדרשים.



### 3.2.4 סביבת הפעלה

האפליקציה תתמוך בגרסת אנדרואיד 7 ומעלה.

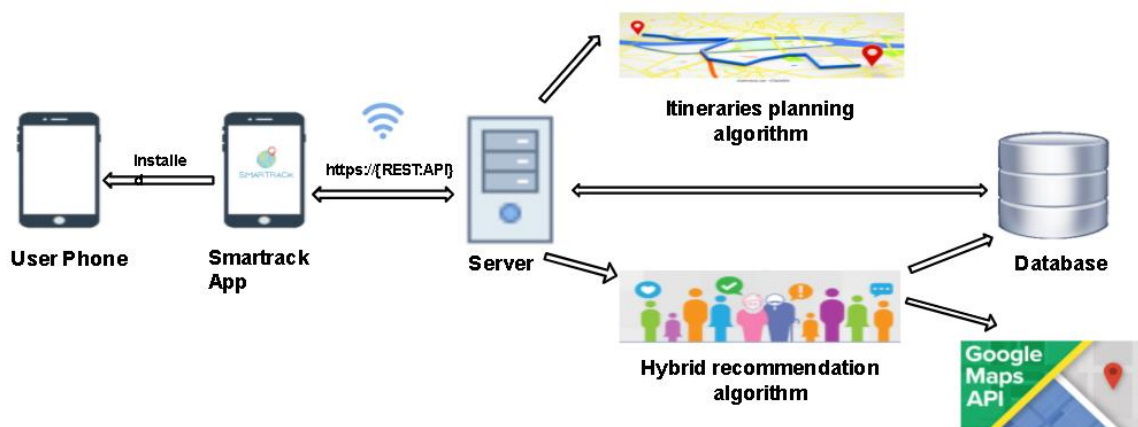
### 3.2.5 אילוצי עיצוב ויישום

- שפת התכנות תהיה Java.
- חיבור יציב לאינטרנט למכשיר שמשתמשים לצורך תכנון טיול , עדכון פרופיל, ודירוג אתרי תיירות .
- קיבולת ענן גדולה ומאובטח כדי לאחסן נתונים.
- תלות ב google map api מבחינת אמינות המידע וזמינות המידע.
- נפח זיכרון במכשיר על מנת לשמור את טיולי המשתמש במכשיר האישי שלו כדי לאפשר יכולת צפייה בהיסטורית טיולים גם ללא חיבור לאינטרנט.

## 3.3 ממשקים

### 3.3.1 ממשק מערכת

- התוכנה רצה על טלפון חכם אנדרואיד בה פונקציות עיקריות צורכות חיבור יציב לאינטרנט .
  - בסיס נתונים mongoDB עבור רשימת משתמשים ונתונים עליהם.
- ארכיטקטורה:



### 3.3.2 ממשק תוכנה

- צד שרת נכתב בשפת JavaScript בסביבת Visual Studio Code.
- צד לקוח נכתב בשפת JAVA בסביבת Android Studio.
- MongoDB - בסיס נתונים לאחסון מידע על המשתמשים והיסטורית הטיולים שלהם.
- ספריות בצד שרת:
  - mongoose - תקשורת עם מסד הנתונים MongoDB.
  - Express - תקשורת פרוטוקול HTTP עם צד לקוח והמסד נתונים.
  - parser-body - מאפשרת ל Express לקבל ולפענח פרמטרים בפורמט JSON

js-recommender - מערכת המלצות בשיטת content collaborative filtering  
kmeans-same-size - חלוקת אתרים ל k קבוצות בצורה אופטימלית

### 3.3.3 ממשק תקשורת

- פרוטוקול HTTPS/HTTP לחיבור למסד הנתונים והאפליקציה.
- ממשק API Rest וספריות HttpURLConnection ו URL לתקשורת בין השרת לאפליקציה.

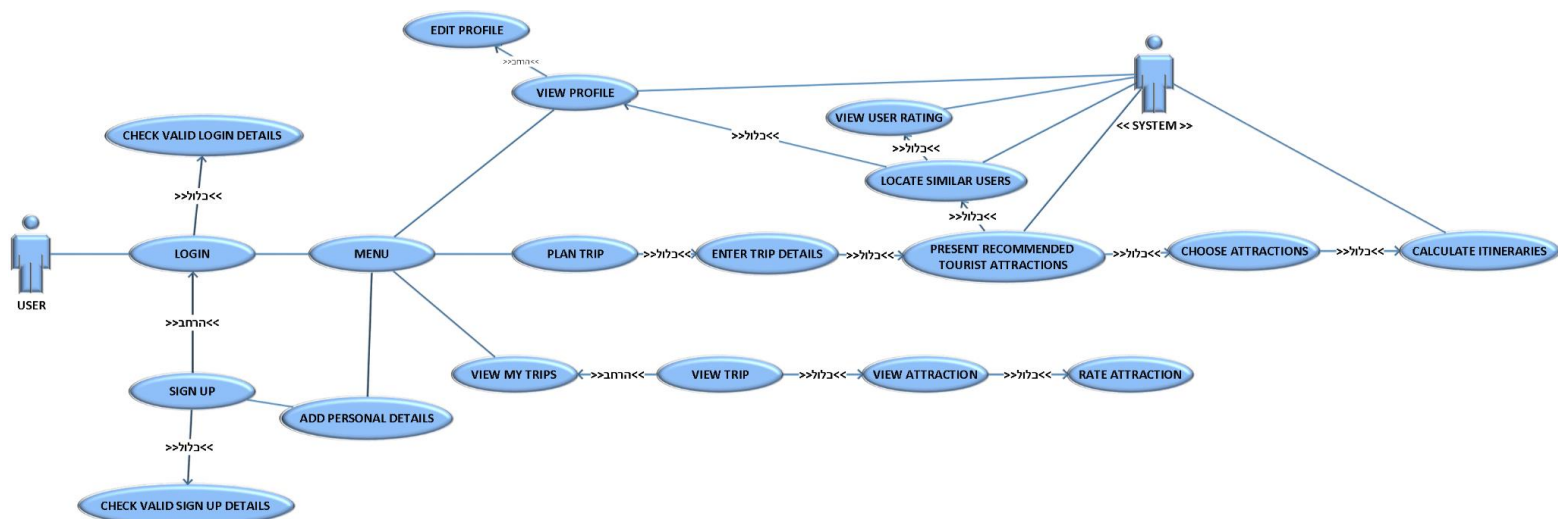
### 3.3.4 ממשק חומרה

האפליקציה תפעל בכל מכשיר אנדרואיד בגרסה 7 ומעלה.

### 3.3.5 ממשק משתמש

- הממשק יהיה נוח ונעים לשימוש.
- יהיה תפריט שבו התכונות העיקריות יוצגו.
- הודעות שגיאה יציגו תיאור שגיאה למשתמש במידת הצורך.

## USE CASE



## 3.4 מאפייני המערכת:

### 3.4.1 דרישות חומרה

- טלפון חכם מחובר לאינטרנט אנדרואיד בגרסה 7 (צד לקוח).
- מערכת הפעלה: ווינדוס 10, 64 סיביות. מעבד: ארכיטקטורת אינטל או AMD,
- ארבע ליבות או יותר. זיכרון: GB8 של RAM או יותר (צד שרת).

### 3.4.2 דרישות התקנה

- על המחשב שמריץ את השרת להיות מחובר לאינטרנט.
- על הטלפון המריץ את האפליקציה להיות מחובר לאינטרנט.
- יש לוודא שקיים חיבור תמידי לחשמל עבור השרת

### 3.4.3 דרישות-צד לקוח (האפליקציה)

#### 3.4.3.1 הרשמה

#### תיאור

כמנהל מוצר אני רוצה שלאפליקציה תהיה דרישת רישום על מנת לשמור פרטים רלוונטיים על המשתמשים ולהשתמש במידע זה.

עדיפות: גבוהה

דרישות פונקציונליות

- 3.4.3.1.1 המשתמש ימלא את השדות הבאים: דואר אלקטרוני, סיסמה, אימות סיסמה.
- 3.4.3.1.1.1 הקלט עבור השדות 'סיסמה' ו'אמת סיסמה' צריך להיות שווה.
- 3.4.3.1.1.2 האימייל חייב להכיל '@'
- 3.4.3.1.1.3 האורך המינימלי לסיסמה הוא 6 תווים.
- 3.4.3.1.1.4 במידת הצורך, יוצגו הודעות שגיאה מתאימות
- 3.4.3.1.1.5 לחיצה על כפתור הרישום תוסיף את המשתמש כמשתמש רשום במסד הנתונים של המערכת.
- 3.4.3.1.2 המשתמש החדש ימלא פרטים אישיים אודותיו.
- 3.4.3.1.2.1 שדה השם הוא חובה.
- 3.4.3.1.2.2 תוצג רשימה של קטגוריות מועדפות של אטרקציות תיירותיות. יש לבחור לפחות קטגוריה אחת.
- 3.4.3.1.2.3 במידת הצורך, יוצגו הודעות שגיאה מתאימות
- 3.4.3.1.2.4 לחיצה על כפתור האישור תוסיף את המשתמש במסד הנתונים.
- 3.4.3.1.2.5 המשתמש יכול לעבור למסך הכניסה למקרה שהמשתמש כבר רשום למערכת.
- 3.4.3.1.2.6 לאחר השלמת תהליך ההרשמה, יוצג המסך הראשי של האפליקציה.
- 3.4.3.1.2.7 במקרה ותהליך הכניסה נכשל, תוצג הודעת שגיאה למשתמש.

3.4.3.2 התחברות

תיאור

כמשתמש אני רוצה להיכנס לחשבון שלי כדי לראות את הטיולים שתכננתי.

עדיפות: גבוהה

דרישות פונקציונליות

- 3.4.3.2.1 המשתמש ימלא את השדות הבאים: דואר אלקטרוני, סיסמה.
- 3.4.3.2.1.1 לחיצה על כפתור הכניסה תכניס את המשתמש לחשבון שלו במערכת.
- 3.4.3.2.2 כניסה אוטומטית - אם המשתמש כבר נכנס לאפליקציה ממכשיר מסוים, הוא לא יצטרך להזין שוב את פרטי הכניסה אלא יועבר אוטומטית למסך הראשי של האפליקציה.
- 3.4.3.2.3 המשתמש יכול לעבור למסך ההרשמה במקרה שהמשתמש אינו רשום למערכת.

3.4.3.2.4. למשתמש תהיה אפשרות לשנות את הסיסמה שלו במקרה שישכח אותה.

3.4.3.2.5. לאחר השלמת תהליך הכניסה, יוצג המסך הראשי של האפליקציה.

3.4.3.2.5.1. במקרה ותהליך הכניסה נכשל, תוצג הודעת שגיאה למשתמש.

3.4.3.3. תכנן טיול

#### תיאור

כמפתח אני רוצה של אפליקציה תהיה אפשרות למשתמש לתכנן טיול חדש כדי ליישם את מטרת האפליקציה.

#### עדיפות גבוהה

#### דרישות פונקציונליות

##### ● מסך תכנון טיול:

3.4.3.3.1. המשתמש יכול לבחור את יעד הטיול ממנוע חיפוש של מפת גוגל שיציג רשימה של מקומות חזויים לבחירה בהתאם לקלט מהמשתמש.

3.4.3.3.1.1. שדה רישום זה הוא חובה.

3.4.3.3.2. המשתמש מחליט על מספר ימי הטיול.

3.4.3.3.2.1. שדה זה הוא חובה.

3.4.3.3.2.2. מספר הימים המינימלי הוא 1.

3.4.3.3.2.3. מספר הימים המרבי הוא 40.

3.4.3.3.3. לחיצה על כפתור "אישור" תבדוק את שדה הקלט ותעביר את המשתמש למסך חדש.

3.4.3.3.3.1. במידת הצורך, יוצגו הודעות שגיאה מתאימות.

##### ● מסך רשימת אטרקציות תיירותיות ביעד:

3.4.3.3.4. למשתמש תוצג רשימה לבחירת אתרי תיירות ביעד הטיול.

3.4.3.3.4.1. עבור כל אתר תיירות יוצגו: שם, תמונה, דירוג וכפתור הוספה.

3.4.3.3.4.1.1. לחיצה על אתר תיירות תעביר את המשתמש

למסך "פרטי אתר תיירות".

3.4.3.3.4.1.2. כפתור 'הוסף' יעודכן ל"הסר" אם המשתמש

ילחץ עליו

3.4.3.3.4.2. המספר המינימלי של אתרים הוא מספר ימי הטיול.

3.4.3.3.4.3. מספר אתרים מקסימלי הוא 3 אתרים לכל יום בטיול.

3.4.3.3.5. הרשימה תורכב מנקודות עניין מקטגוריות שונות. הרשימה

תיבנה על פי מערכת המלצות ומידע שיש למשתמש על

העדפותיו ונסיעות קודמות.

3.4.3.3.6. לחיצה על כפתור "תכנן טיול" תבדוק את מספר האתרים

הנבחרים. במקרה שהמספר אינו חוקי תוצג הודעה בהתאם. אם



כן, המערכת תחשב את המסלול האופטימלי עבור האתרים הנבחרים ותעביר את המשתמש למסך הטיול שנוצר.

● מסך פרטי אתר תיירות:

3.4.3.3.7. עבור כל אתר תיירות יוצגו הפרטים הבאים: שם, כתובת, מספר טלפון, שעות פתיחה, דירוג ותמונה וכפתור הוספה לרשימת האתרים הנבחרים לטיול.

3.4.3.3.8. כפתור 'הוסף' יעודכן ל"הסר" אם המשתמש ילחץ עליו.

● מסך לוח זמנים של טיול

3.4.3.3.9. עבור כל טיול יוצגו הפרטים הבאים: שם, מספר ימים, יעד ורשימת ימים

3.4.3.3.10. רשימת הימים תהיה אורך מספר ימי הטיול.

3.4.3.3.11. לחיצה על יום ספציפי תעביר את המשתמש למסך "מקומות ביום מסוים".

● מקומות במסך יום ספציפי

3.4.3.3.12. מסך זה יציג עבור כל יום את אתרי התיירות שתוכננו לאותו יום.

3.4.3.3.13. עבור כל אתר יוצגו הפרטים הבאים: שם, תמונה, דירוג

3.4.3.3.14. לחיצה על אתר תיירות מסוים תוביל למסך פרטי אתר תיירות

3.4.3.4. היסטוריית לוח זמנים של טיולי המשתמש

תיאור

כמשתמש הייתי רוצה לראות את הטיולים שתכננתי כדי לדרג ולדעת מה התוכניות שלי.

עדיפות: גבוהה

דרישות פונקציונליות

● מסך טיולים שלי

3.4.3.4.1. מסך זה יציג למשתמש רשימה של כל הנסיעות העתידיות שתכנן.

3.4.3.4.2. עבור כל טיול יוצג השם הטיול מספר ימי טיול יעד הטיול ותאריך יצירת הטיול.

3.4.3.4.3. לחיצה על נסיעה מסוימת תעביר את המשתמש ל"מסך לוח זמנים של טיול".

3.4.3.4.4. רשימת היסטוריית הטיולים תהיה ממוינת בסדר עולה כך שהטיול העדכני ביותר בראש הרשימה.

3.4.3.5. עריכת פרופיל משתמש

תיאור

כמשתמש הייתי רוצה שתהיה לי אפשרות לערוך את הפרטים האישיים שלי כך שיהיו מעודכנים.

עדיפות: בינונית

דרישות פונקציונליות

3.4.3.5.1. פרטי המשתמש יוצגו והמשתמש יכול לשנותם בהתאם לדרישות שהוזכרו לעיל.

- 3.4.3.5.2. המשתמש יוכל לבטל את העריכה.
- 3.4.3.5.3. המשתמש יכול לערוך את הפרופיל רק כאשר מחובר לרשת האינטרנט.
- 3.4.3.6. דירוג אטרקציה תיירותית
- תיאור
- כמפתח אני רוצה שהמשתמש ידרג את אתרי התיירות בהם היה על מנת לאסוף מידע על העדפותיו למערכת המלצות אתרי תיירות.
- עדיפות: גבוהה
- דרישות פונקציונליות
- 3.4.3.6.1. המשתמש יכול לדרג את אתרי תיירות שביקר רק כאשר יש לו חיבור לרשת האינטרנט.
- 3.4.3.6.2. דירוג הטיול יהיה נעה בין 1 ל 5
- 3.4.3.6.2.1. הדירוג יהיה על ידי כוכבים שבלחיצה עליהם יצבעו כך שהמשתמש יכול לצפות בדירוג
- 3.4.3.6.3. המערכת לא תחייב את המשתמש לדרג ותשים לו ערך דיפולטי 0 המעיד על אי דירוג האתר.
- 3.4.4. דרישות - צד שרת
- 3.4.4.1. תקשורת עם דאטה בייס חיצוני
- למערכת תהיה אפשרות לתקשר עם דאטה בייס חיצוני עבור הפעולות: הכנסה, שליפה ועדכון .
- 3.4.4.1.1. הכנסה - כוללת הוספת אובייקט שלם ל collection המתאים.
- 3.4.4.1.2. שליפה - כוללת שליפה של אובייקט ספציפי על ידי פילטר, ושליפת שדות מרובים.
- 3.4.4.1.3. עדכון - כוללת עדכון של שדות ספציפיים באובייקט ספציפי ועדכון מרובה.
- 3.4.4.1.4. המערכת תדע לטפל בתגובות המתקבלות מהדטא בייס לאחר כל פעולה.
- 3.4.4.1.5. בדאטה בייס ישמרו הנתונים הבאים: טיולי משתמש, משתמשים, אתרי תיירות ודירוגים.
- 3.4.4.2. תקשורת עם צד לקוח
- 3.4.4.2.1. למערכת תהיה אפשרות לתקשר עם הצד לקוח על ידי API Rest וחיבור HTTP.
- 3.4.4.2.2. המערכת תדע לבצע מול הצד לקוח את הפעולות: קבלת נתונים ושליחת נתונים.
- 3.4.4.3. אלגוריתם תכנון מסלול טיול
- 3.4.4.3.1. האלגוריתם מקבל מצד לקוח רשימת מיקומים של אתרי תיירות ומספר ימי טיול מהלקוח ומחזיר רשימה של יום בטיול לכל אתר תיירות בטיול .
- 3.4.4.3.2. האלגוריתם משתמש בספריית "kmeans-same-size" לצורך חלוקת האתרים ל k ימים בצורה אופטימלית.

#### 3.4.4.4 אלגוריתם המלצות טיולים

- 3.4.4.4.1 האלגוריתם מקבל את המייל של המשתמש ויעד הטיול ומחזיר רשימת אתרי תיירות מומלצים למשתמש ביעד הטיול.
  - 3.4.4.4.2 האלגוריתם ימליץ למשתמש טיולים ביעד שבו משתמשים דומים למשתמש ביקרו שדירוגם מעל 3.
  - 3.4.4.4.3 אלגוריתם ישתמש בסיפריית "js-recommender" לצורך מציאת אתרי תיירות מומלצים למשתמש.
  - 3.4.4.4.4 במידה והאלגוריתם לא ימצא אתרי תיירות למשתמש הוא יחזיר false
  - 3.4.4.5 הפעלת המערכת
- הפעלת המערכת תתבצע באופן ידני ולאחר מכן תפעל בצורה אוטומטית ללא התערבות.

#### 3.4.5 דרישות לא פונקציונליות

- 3.4.5.1 דרישות ביצועים
  - המערכת צריכה להיות זמינה 24/7.
  - היישום צריך להיות עמיד בעומס.
- 3.4.5.2 דרישות אבטחה
  - המשתמשים יידרשו להיכנס למערכת כדי לגשת לפונקציות השונות באפליקציה.
  - המידע המקושר לאתר יהיה מוגן ומאובטח.
  - המערכת צריכה להיות מוגנת מפני הפרות של פרטיות הנתונים.
  - המערכת צריכה להיות מוגנת מפני התחברות לא חוקית של משתמשים
- 3.4.5.3 דרישות איכות
  - זמינות- האפליקציה תמיד תהיה זמינה כך שמשתמשים יוכלו לבצע בה פעולות שונות בזמנים הנוחים להם.
  - גמישות- האפליקציה תהיה גמישה כך שיהיה קל להוסיף פונקציונליות ותכונות אם נרצה.
  - שימושיות- האפליקציה תהיה שמישה ותעמוד בדרישות.
  - ידידותיות - האפליקציה צריכה להיות ידידותית למשתמש וקלה לשימוש, כך שהמשתמשים יוכלו להשתמש בה ללא בעיות.
  - עמידות- האפליקציה צריכה להיות עמידה לעומס כך שהמערכת תמשיך לתפקד גם כשיש עומס ולחץ על האפליקציה.
- 3.4.5.4 דרישות עסקיות
  - משתמשים לא יוכלו לגשת למידע האישי של משתמשים אחרים.
  - האפליקציה לא תאלץ את המשתמשים לדרג את אתרי התיירות שבהם הם היו כדי לא לפגוע באמינות מערכת ההמלצות, התלויה בדירוגי משתמשים.

## 4. תהליך הפיתוח ואתגרים

### 4.1 תהליך עבודה

בשלב הראשון: קריאת מאמרים אקדמיים על חשיבות האפליקציה והצריכה שלה כיום בעידן התיירות החכמה (smart tourism),

האם תשפר את חיי האנושיות ואיך היא תתרום לקהילה. קראנו בעיקר מה אנשים מחפשים כאשר הם מתכננים טיול ליעד חדש שלא ביקרו בו מעולם ולכן יש חשיבות לבניית אפליקציית Smartrack מכיון שהיא חלק מחווית משתמש בטיול.

בשלב השני: התחלנו עם חיפוש אחר מקורות מידע על אתרי תיירות ברחבי העולם, חיפשנו מידע אמין בעלות סבירה ואכן באפליקציה השתמשנו במאגר הנתונים של google place.

בשלב השלישי: בינינו את המוקאפ של האפליקציה, מה היא תכיל, איך היא תראה, על מנת לקבל תמונה כללית בגדול על הפרויקט.

בשלב הרביעי: כתיבת מסמך הדרישות של צד השרת והלקוח.

בשלב החמישי: בצד הלקוח -האפליקציה, התחלנו לבנות את המסכים של ההרשמה, התחברות והתנתקות מהאפליקציה.

בשלב השישי: בצד השרת בינינו מערכת המלצות המבוססת תחילה על מידע המשתמש על סמך הפרופיל שלו.

בשלב השביעי: בניית אלגוריתם לתכנון הטיול Same-size k-Means כדי לבצע חלוקה אופטימלית של אתרי התיירות בימי הטיול, כך ש K הוא מספר ימי הטיול ובכל יום בטיול במקסימום יהיה שלוש אתרי תיירות.

בשלב השמיני: בניית מערכת הדירוגים באפליקציה והתחלנו למלא את בסיס הנתונים שלנו בדירוג משתמשים.

בשלב הסופי: בניית מערכת ההמלצות היברידית שמשלבת בין עדיפויות המשתמש לבין אתרי תיירות ביעד הטיול שמשתמשים דומים למשתמש אהבו

במהלך פיתוח התוכנה הקפדנו על שמירה של עקרונות פיתוח תוכנה כמו מודולריות - הפרדה בין צד שרת ללקוח, הפרדה של לוגיקות שונות בתוך המערכת בצורה המאפשרת החלפה של חלקים או הוספה של חלקים שלא הוגדרו מראש, השתדלנו ליצור כמה שפחות תלות בין הפונקציות השונות.



## 4.2 קשיים ואתגרים

### 4.2.1

מציאת מידע על אתרי תיירות בכל עיר ברחבי העולם

אתגר מרכזי בפרויקט הוא מציאת מאגר נתונים אמין על אתרי תיירות ברחבי העולם.

תחילה חשבנו לקחת את המידע מאתר **tripadvisor** - אתר בו מוצגים אתרי תיירות ביעד מסוים בעולם על ידי Web scraping שזה תהליך אוטומטי המיושם באמצעות בוט או סורק אינטרנט. זוהי צורת העתקה שבה נתונים ספציפיים נאספים ומועתקים מהאינטרנט, בדרך כלל למסד נתונים מקומי או גיליון אלקטרוני מרכזי, לצורך אחזור או ניתוח מאוחר יותר. חסרונות שמצאנו בכך:

- עלויות יקרות- תשלום לאתרים שהם סורקי דפי אינטרנט ומחזירים את המידע שנמצא באותו הדף.
- ביצועים- זמן העברת המידע עובר בין כמה שרתיים ולכן הבאת המידע על היעד איטית יותר וזה גורע מהביצועים של האפליקציה.
- לאחר חיפוש ולמידה עצמית מעמיקה הבנו שכל אתרים וגם tripadvisor לוקחים את המידע הבסיסי שלהם מ **google place** מכיון שהמידע שלו אמין ומשפרים את הידע שלהם על אותם אתרי תיירות ומרחיבים אותו על ידי פידבק משתמשים ואנשי פיתוח.

ל google place יש ממשק מאוד נוח למפתח. היינו צריכות ללמוד אותו ולהבין אותו.

חסרון:

- עלויות כספיות במידה ועוברים את מספר הגישות והקריאות לשרתי google place יתרון:
- ביצועים. מכיון שאנו ניגשים בעצמנו למידע ולא עוברים דרך צד שלישי ולכן הבאת המידע הרבה יותר מהירה.

### 4.2.2 הקמת השרת

אתגר מהותי היה בחירת שפה לצד השרת בה אנו בונים פונקציות עיקריות במערכת ומתקשרים עם בסיס נתונים של MongoDB. תחילה כתבנו את השרת בשפת python מכיון שזו שפה קלה לכתיבה וקיימות בה הרבה פונקציות וספריות שיכלו לעזור לנו לבניית אלגוריתמים המרכזיים במערכת. הקושי היה חיבור שרת לבסיס נתונים ולשרתי של heroku. כדי להתמודד עם בעיה זו התחלנו בחיפושים אחר אלגוריתמים מוכנים שיכולים להתאים למערכת שלנו שכבר מומשו על ידי אחרים בשפת Javascript שאיתה יש לנו יותר ניסיון בכתיבת שרת ועבודה מול בסיס נתונים ואכן מצאנו ספריות שיכולות לעזור לבניית האלגוריתם במערכת.

## 4.2.3.

## בניית מערכת המלצות

בניית מערכת המלצות היתה אתגר, תחילה ניסינו לממש אלגוריתם של מערכת המלצות המבוססת על collaborative-filtering שנלקח ממאמר, המימוש היה מסובך, הבנת האלגוריתם הייתה קשה, מה שלקח הרבה זמן. לאחר החלטה לכתוב את צד השרת ב Javascript מצאנו ספריה מוכנה "js-recommender" של מערכת המלצות מבוססת שיתופית שיכולה לעזור לנו בבניית האלגוריתם ואכן התאמנו אותה למערכת שלנו.

## 4.2.4.

## בניית אלגוריתם לתכנון מסלול טיולים

האתגר בבניית אלגוריתם לחלוקה שווה של אתרי תיירות במספר ימי הטיול בצורה אופטימלית: תחילה חשבנו לקחת קוד מוכן של ספריית Elki. התחלנו בהעתקת הקוד מהגיט שלהם אך גילינו שהם השתמשו בספריות לא עדכניות ולתקן זאת היה נורא מסובך. לכן החלטנו לממש בעצמנו את האלגוריתם של Same-size k-Means לפי מאמר. זה היה מסובך אבל לבסוף הצלחנו. לאחר ששינינו את השרת לשפת Javascript מצאנו ספריה 'kmeans-same-size' שמממשת את אלגוריתם זה והשתמשנו בה.

## אלגוריתם מערכת המלצות

### הקדמה

מערכות המלצה הן מערכות שמשמשות על מנת לסנן כמויות גדולות של מידע באמצעות הפצת תהליך של סינון בקרב קבוצה גדולה של אנשים. המערכות מספקות המלצות למשתמשים על פריטים שעשויים לעניין אותם. ההמלצות עוזרות למשתמשים לאתר פריטים רלוונטיים. הנחת היסוד של גישת מערכות ההמלצה היא כי אלו שהסכימו בעבר נוטים להסכים שוב בעתיד, עם זאת יש לשים לב כי ההמלצות ניתנות לפי בחירות המשתמש היחיד אותו זיהתה המערכת והצליבה עם מידע קודם שהיה עליו ועם מידע אחר שהיה לה ממשתמשים שונים בעלי בחירות דומות. במערכת Smartrack איסוף הנתונים על המשתמש הוא הדירוגים שלו באתרי תיירות שביקר בהם. המערכת ממליצה למשתמש אתרי תיירות שמשתמשים דומים אהבו (ז"א דירגו מעל דירוג 3) ביעד הטיול.

מערכת ההמלצה משווה את הנתונים שנאספו לנתונים של אנשים אחרים ומשתמשת בשני עקרונות בסיסיים המעורבים ביישום:

1. בחוכמת ההמונים ובחוק המספרים הגדולים שניהם מראים כי ככל שהקהילות גדלות, אך לא רק אלא גם מגוונות, ועצמאיות לדוגמה, כך הקהילה יכולה לקבל החלטות טובות יותר מאשר קומץ העורכים, ולכן ככל שהקהילה תהיה גדולה יותר ההחלטות שלה עשויות להיות טובות יותר.
2. העיקרון השני מציע כי בכל קהילה גדולה, עם מספיק נתונים על המשתתפים בודדים, על איך משתתפים בודדים מתואמים או משתפים פעולה אחד עם השני, אנחנו יכולים לעשות תחזיות לגבי מה שהמשתמשים האלה יאהבו בעתיד על סמך הטעמים שלהם בעבר, כלומר לפתח מנועי המלצה מסוננים בשיתוף פעולה. זה, כמובן, מסתמך על העובדה כי האינטרסים של אנשים, העדפות אידאולוגיות אינם משתנים באופן דרסטי מדי לאורך זמן, ובהתאם לחשב את רשימת הפריטים מומלצים עבור המשתמש על בסיס עקרונות אלו.

מערכת Smartrack היא מערכת המלצות היברידית שמשלבת מערכת מבוססת שיתופיות (collaborative-filtering) לבין מערכת המבוססת על מידע של פרופיל המשתמש כדי להקל על בעיית ההתחלה הקרה.

### מימוש האלגוריתם

כדי להתמודד עם בעיית התחלה קרה במערכת המלצות הפתרון שלנו הוא מערכת המלצות היברידית ולכן בהרשמה לאפליקציה המשתמש יבחר לפחות קטגוריה אחת של אתר תיירות מהרשימה הבאה:

1. amusement park
2. aquarium
3. art gallery
4. bar
5. casino
6. museum

7. night club
8. park
9. shopping mall
10. spa
11. tourist attraction
12. zoo
13. bowling alley
14. cafe
15. church
16. city hall
17. library
18. mosque

כאשר משתמש מתכן מסלול טיולים ליעד מסוים אנו פונים לשרת ושולחים לו את מייל המשתמש ויעד הטיול.

### **בצד שרת**

בשלב הראשון אנו ממליצות למשתמש על אתרי תיירות ביעד הטיול שמשתמשים דומים לו אהבו (כלומר שדירגו את אתר התיירות מעל דירוג 3).

אלגוריתם ההמלצות משתמש בספריה מוכנה "js-recommender" המבוססת על ידי שיטת Collaborative Filtering. קישור לספריה: <https://www.npmjs.com/package/js-recommender>

בנינו טבלה כך ש:

עמודות - אלו אתרי תיירות שהמשתמש שלנו ביקר בהם וגם אתרי תיירות ביעד הטיול שנמצאים בבסיס הנתונים שלנו.

שורות - משתמשי המערכת שביקרו באתרי התיירות שנמצאים בעמודות הטבלה.

תוכן הטבלה - דירוג אתרי תיירות של המשתמש.

הפעלנו את האלגוריתם ההמלצות של ספריית "js-recommender" על הטבלה שלנו והאלגוריתם החזיר לנו מהו הדירוג הצפוי של המשתמש עבור כל יעד שלא ביקר בו. אנחנו לקחנו רק את אתרי התיירות ביעד הטיול שהדירוג המשווער שלהם גדול מ 3.

בשלב השני הבאנו מבסיס הנתונים של google place בעזרת שימוש ב google place api את כל הטיולים שנמצאים ביעד לפי העדפות המשתמש במרחק של עד 30 ק"מ מהיעד. המימוש נעשה על ידי שימוש ב Nearby Search HTTP request.

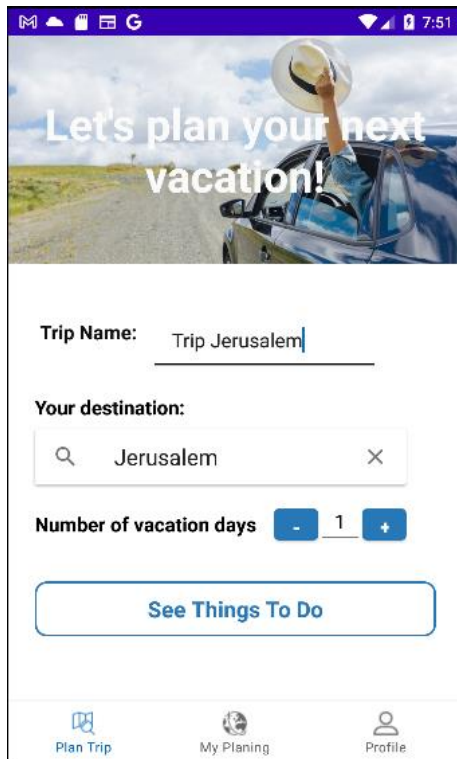
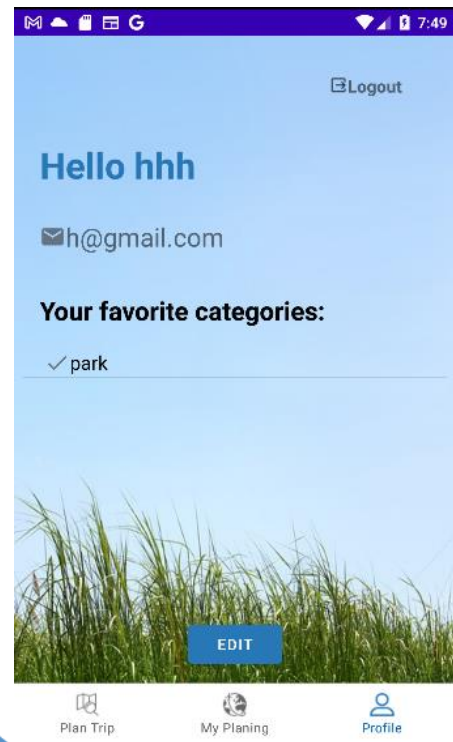
בסופו של התהליך השרת יחזיר לאפליקצית Smartrack רשימת אתרי תיירות המומלצים למשתמש ביעד. האפליקציה תציג רשימה של אתרי תיירות ביעד כך שאתר התיירות שהתקבל ממערכת ההמלצות, סומן בתג "מומלץ" שמסמל שאת אתר זה אהבו משתמשים הדומים למשתמש. במידה ואתר התיירות נלקח מ google place על סמך העדפות המשמש הוא לא סומן בתג.



## דוגמת הרצה

פרופיל משתמש: [h@gmail.com](mailto:h@gmail.com)

תכנון טיול ל Jerusalem:

בצד השרת:

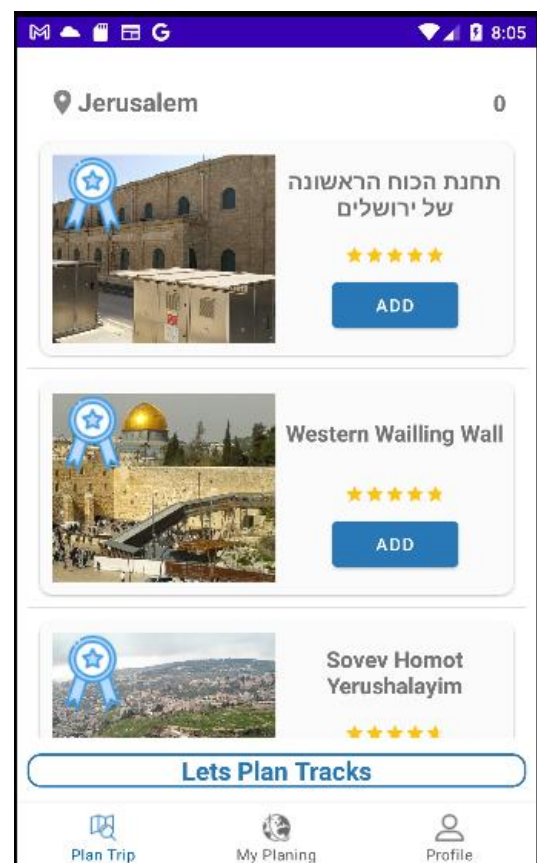
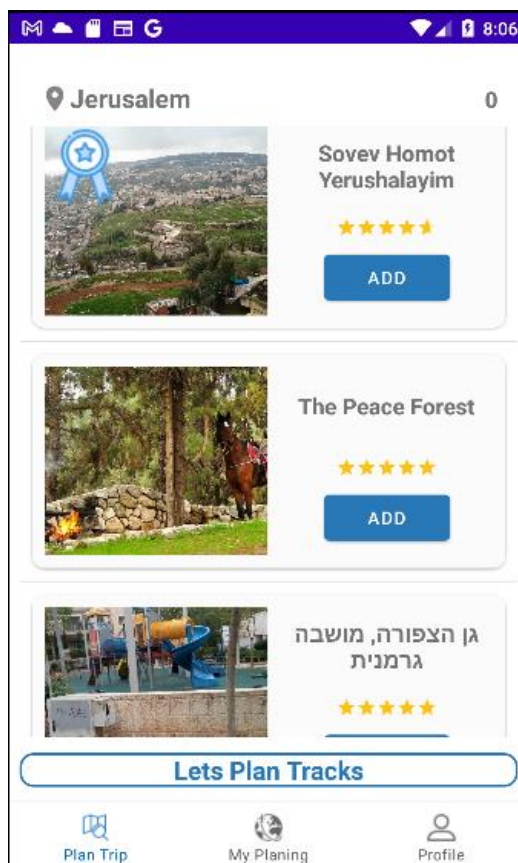
השרת מקבל את מייל המשתמש [h@gmail.com](mailto:h@gmail.com) ויעד הטיול Jerusalem. בניית הטבלה והפעלת אלגוריתם המלצות אתרים ביעד שמשתמשים דומים אהבו ועדיין המשתמש לא ביקר בהם.

	אתרי תיירות שהמשתמש דירג														אתרי תיירות ב Jerusalem								
אתרים/משתמשים	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
a@gmail.com	4																						
b@gmail.com	5													3	2	5	4	5	3	4			
c@gmail.com	3																						
e@gmail.com		3	5	5																			
f@gmail.com		2	5				5	5									2	2					
g@gmail.com							5	5	0								0						
<b>h@gmail.com</b>	<b>5</b>	<b>5</b>	<b>4</b>	<b>4</b>	<b>2</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>2</b>	<b>1</b>	<b>3</b>	<b>2</b>	<b>5</b>	<b>1</b>	<b>✓</b>				<b>✓</b>		<b>✓</b>		
i@gmail.com	4	5	5	5		0							5	5									
j@gmail.com		5	5	4	4	0	5	5	2	1		2	5	5							3	5	0
k@gmail.com	0												0	0									

המספר הסידורי של האתרים הוא לפי מ"ז של האתרים על פי הטבלה הבא:

שם האתר תיירות	מספר מזהה אתר תיירות	מספר סידורי
	ChIJadgVFQIAhURb8q8Zfo47qY	1
	ChIJaelghY5LHRUR9x8gG8vfcVWs	2
	ChIJst84WKFLHRURwJdABZ8g7wl	3
	ChIJP9vz3ZxLHRURb7lr4P8pMwU	4
	ChIJH7T057hMHRURzPhw7G0PQYs	5
	ChIJwSVKVpGjAhURH8U5q4Eyc1c	6
	ChIJasIZuLApAxURQ0a8QhPbBPE	7
	ChIJ085LraTXAhURNvgFz825tYw	8
	ChIJgxPoiwLXAhURx0s8Kh0Q8hQ	9
	ChIJd6YThsrXAhURllo2GQhX0Uw	10
	ChIJgyPYHNwpAxURo84Yz8yasOM	11
	ChIJYXHtci3WAhURDkQC8oP7DI8	12
	ChIJOyFGqHSjAhURQMGGxJj2tAQ	13
	ChIJF08X18qjAhUR5k6C5gthRSY	14
	ChIJCWst09cpAxURcPL674BYEvA	15
תחנת הכוח הראשונה של ירושלים	ChIJ4X7aO4QpAxURd2lfPJHCO1M	16
	ChIJgwDqNNbXAhUR8SmYGst1JRY	17
	ChIJr3y1_UnWAhURBQdqtU9fUyg	18
	ChIJP2ulpNEpAxUR5grH1dVfKmq	19
Western Wall	ChIJJZOa_0PXAhURfk6JMiza4sl	20
	ChIJ50Ad1igoAxURbK0eb7wAvXY	21
Sovev Homot Yerushalayim	ChIJUzUSqzQoAxURwQzHtWd5Szc	22
	ChIJTyUSRdApAxUR0GEBXmTrLHA	23

הרשימה של אתרי תיירות ביעד שתוצג בצד הלקוח:



## 6. אלגוריתם תכנון טיול

### הקדמה

מטרת האלגוריתם היא חלוקת אתרי תיירות למספר ימי הטיול, המקומות הקרובים זה לזה יהיו באותו היום וזאת כדי לסייע בהפחתת זמן הנסיעה בין האתרים. זה ממומש על ידי האלגוריתם same size k-means clustering. האלגוריתם יארגן עבור התיירים את מסלול הטיול שישבץ את המקומות הקרובים ביותר על סמך המיקום הגיאוגרפי שלהם, מקסימום 3 אתרים בכל יום ויחסוך להם זמן הן בתכנון מסלול עבור כל יום והן במעבר בפועל ממקום למקום.

### אלגוריתם same size k-means clustering

אלגוריתם k-מרכזים (k-means) הוא שיטה פופולרית עבור ניתוח אשכולות (Clustering) בכריית נתונים. מטרתו לחלק את התצפיות ל-k אשכולות לפי מרכזי כובד (k-means). כל תצפית שייכת לאחד מ"מרכזי הכובד". על ידי בחירה נכונה של מרכזי כובד ניתן לאתר את הקבוצות השונות. נדרשות תצפיות רבות על מנת להשתמש במודל ותוספת של תצפיות חדשות עשויה לחייב חישוב חוזר. מדובר באלגוריתם היוריסטי שמשתמשים בו כדי לבצע חישובים המובילים להתכנסות לפתרון מקומי (שאינו בהכרח הפתרון הטוב ביותר).

זהו מודל סטטיסטי שאינו מתבסס על ידע מוקדם על הנתונים אלא רק על תצפיות בפועל. שיטה זו דומה לאלגוריתם EM המניח התפלגות הדוגמאות מכמה התפלגויות גאוסיות.

מודל זה נוטה למצוא אשכולות בעלי מרחבי מידה הניתנים להשוואה. בעוד שמנגנון ציפייה מביא למקסום ומאפשר לאשכולות להיות בעלי צורות שונות. מודל זה לא ליניארי, ולכן הוא מאפשר להתמודד עם כל סוגי ההתפלגויות האפשריות.

האלגוריתם עובד בהינתן קבוצה של תצפיות  $(x_1, \dots, x_n)$  כאשר כל תצפית היא וקטור ממשי היכול להיות בעל מספר ממדים. מטרת המודל היא לחלק את n התצפיות לא אשכולות, על מנת למזער את סכום המרחקים בין התצפיות בתוך האשכול.

האלגוריתם מאפשר להתמודד עם נתונים ללא התפלגות מוגדרת או ללא סיווג (קלסיפיקציה) וגם עם מצבים שבהם בתצפית מסוימת חסרות חלק מהמידות של המשתנים.

האלגוריתם מאפשר לבצע למידה בלתי מונחית (Unsupervised learning) של סיווגים שונים של הנתונים.

אלגוריתם same size k-means clustering מתבסס על אלגוריתם k-means רק שבו גודל כל האשכולות שווה.

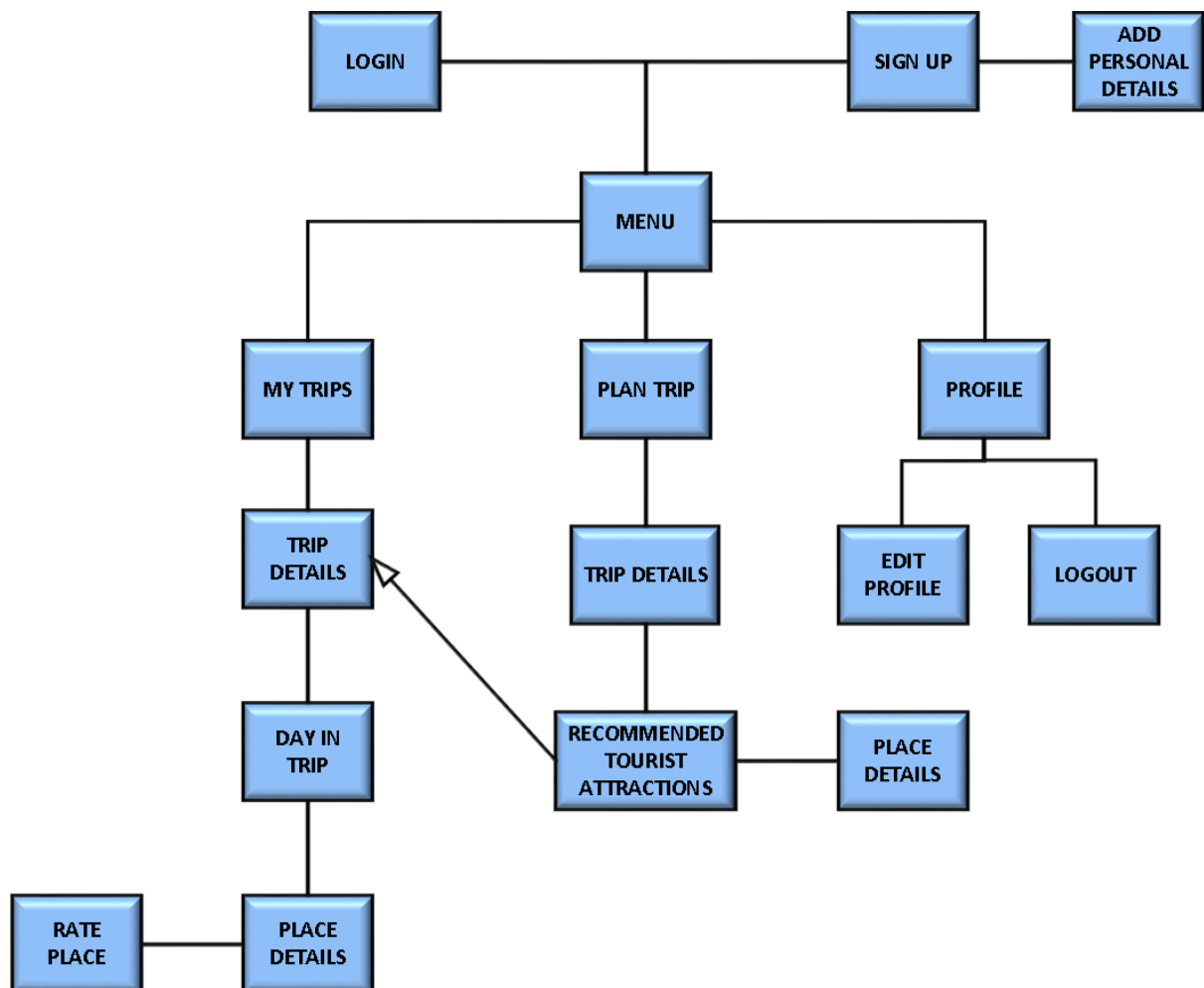
### מימוש האלגוריתם

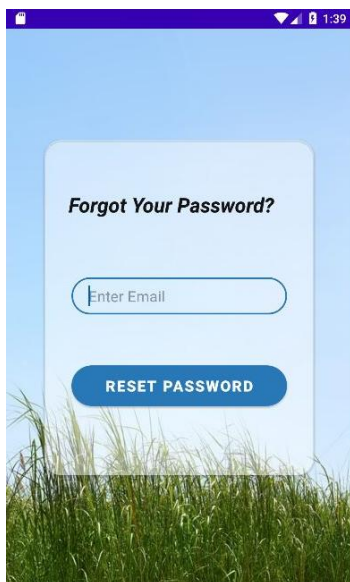
צד השרת קיבל את רשימת אתרי התיירות וחילק אותם למספר ימי הטיול על ידי שימוש בספריית 'kmeans-same-size'

קישור לספרייה: <https://www.npmjs.com/package/kmeans-same-size>

האלגוריתם חילק את אתרי התיירות ל K אשכולות שווים, K הוא מספר ימי הטיול.

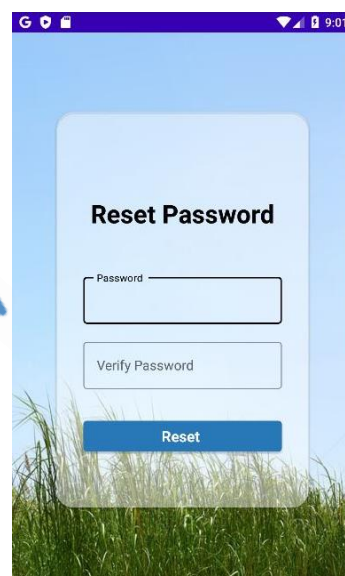
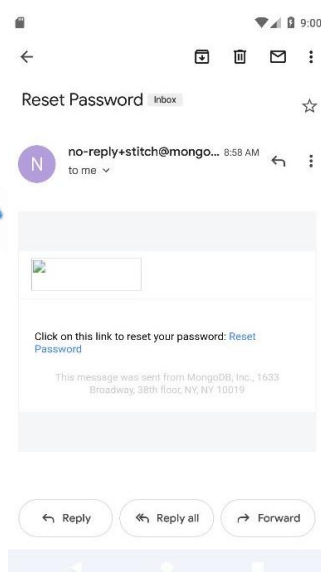
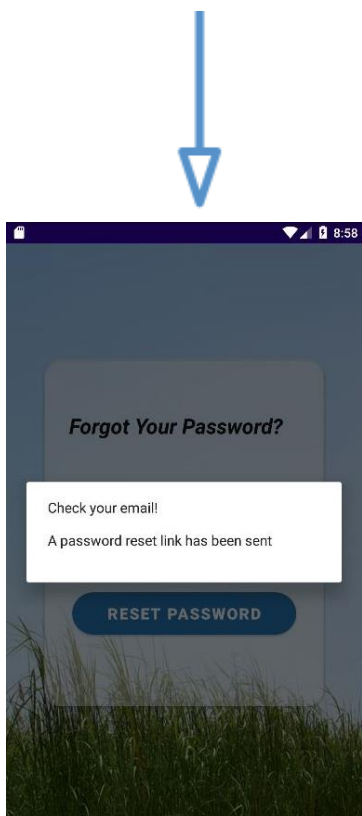
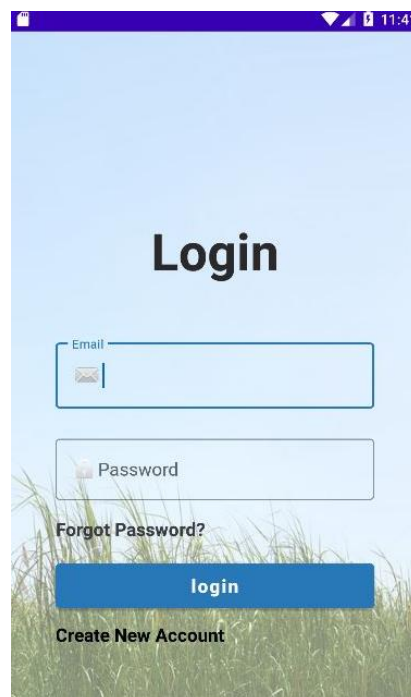
השרת החזיר לאפליקציה Smartrack עבור כל אתר תיירות יום בטיול שהוא משובץ.





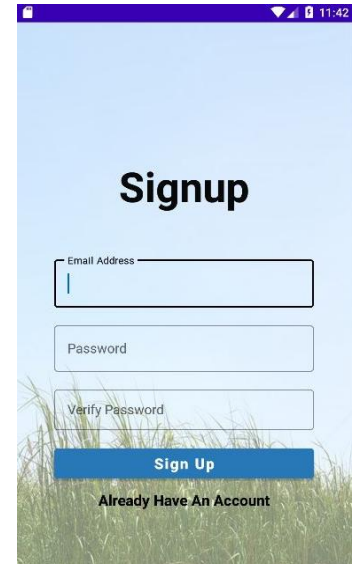
**מסך התחברות:**  
משתמשים רשומים יתחברו לחשבונם  
על ידי הכנסת מייל וסיסמה.  
משתמשים חדשים יוכלו לעבור למסך  
הרשמה.

**מסך שכחתי סיסמא:**  
המשתמש יכול לאפס  
את סיסמתו על ידי קבלת  
קישור לאיפוס סיסמא  
במייל.



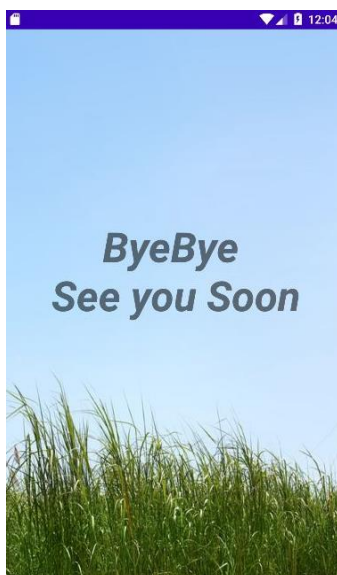


**מסך הרשמה:**  
משתמשים חדשים  
נרשמים למערכת



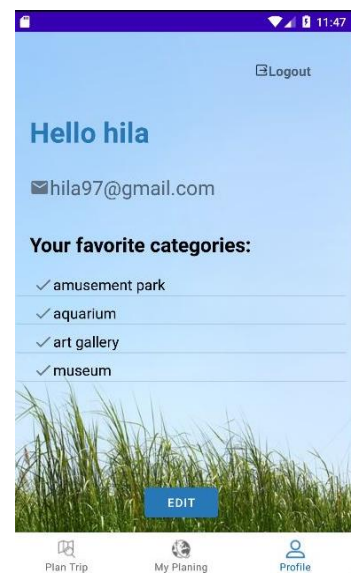
### מסך פרטים אישיים:

משתמשים חדשים יתבקשו למלא פרטים אישיים עליהם כולל בחירת קטגוריות מועדפות של אתרי תיירות. לאחר לחיצה על "ok" המשתמש ינותב למסך פרופיל ויוצג לו תפריט תחתון קבוע לניתוב באפליקציה



### מסך התנתקות:

לאחר לחיצה על כפתור logout בפרופיל תוצג הודעה למשתמש והוא ינותב בחזרה למסך login



### מסך פרופיל משתמש:

מציג את פרטי המשתמש. מכיל כפתור edit לעריכת הפרטים וכפתור logout להתנתקות מהפרופיל



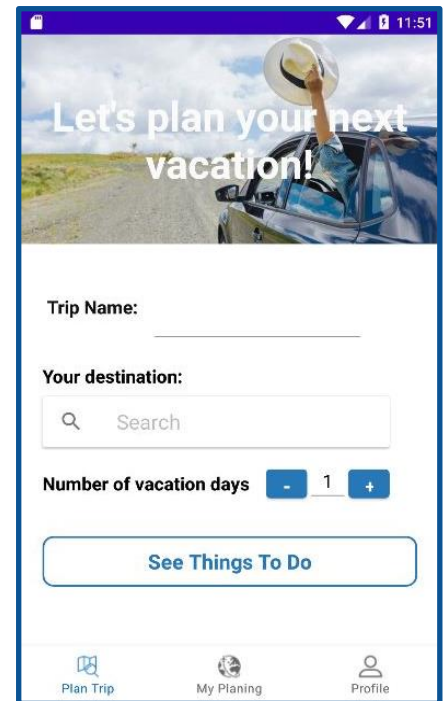
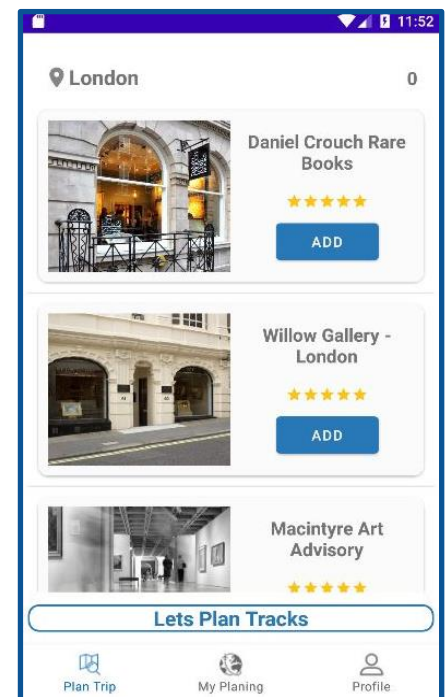
### מסך עריכת פרופיל:

המשתמש יכול לערוך את פרטי הפרופיל שלו.



### מסך תכנון טיול:

המשתמש ימלא את פרטי הטיול ויבחר את יעד הטיול ואת מספר הימים .  
see things to do על כפתור  
המשתמש יעבור למסך הצגת מקומות ביעד הטיול.

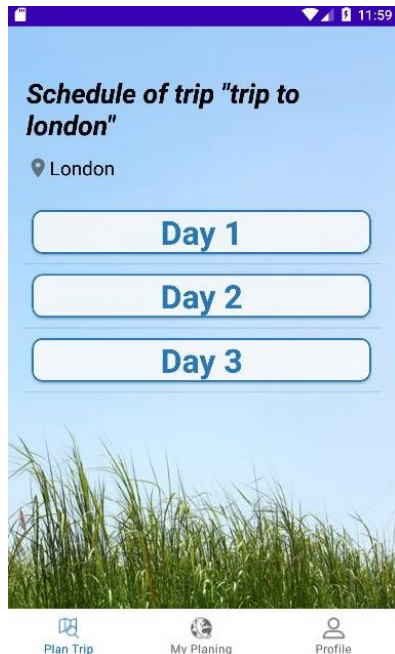
### מסך פרטי אתר תיירות:

במסך זה יוצגו פרטי אתר התיירות. לחיצה על הלינק תעביר את המשתמש לאתר האינטרנט. לחיצה על מספר הטלפון תעביר את המשתמש למסך החיוג של המכשיר. לחיצה על כפתור add תוסיף את האתר לרשימת האתרים שנבחרו.

### מסך בחירת אתרי תיירות:

במסך זה תוצג רשימה של אתרי תיירות ביעד הטיול לפי מערכת ההמלצות. המשתמש יבחר מספר מוגבל של אתרים. בלחיצה על כפתור let's plan tracks המערכת תיצור טיול עם מסלולים והמשתמש יעבור למסך פרטי הטיול.





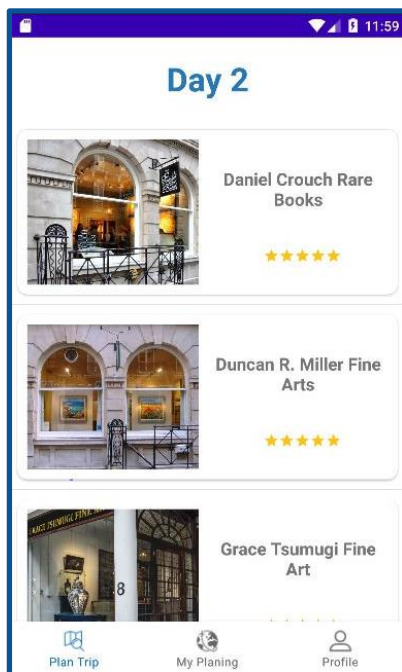
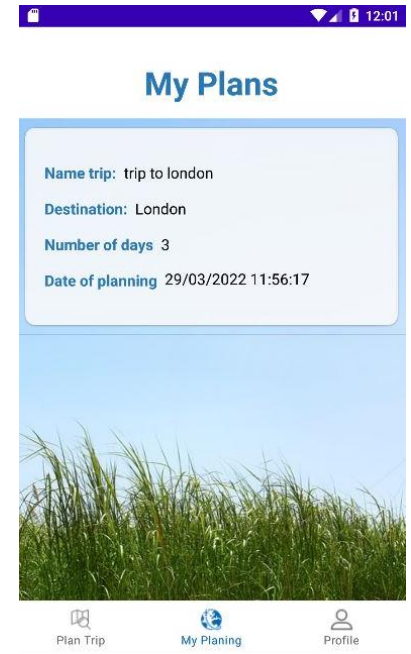
### מסך הטיולים שלי:

במסך זה תוצג רשימה של הטיולים שהמשתמש יצר עם פרטי הטיול. לחיצה על טיול תוביל למסך פרטי טיול.



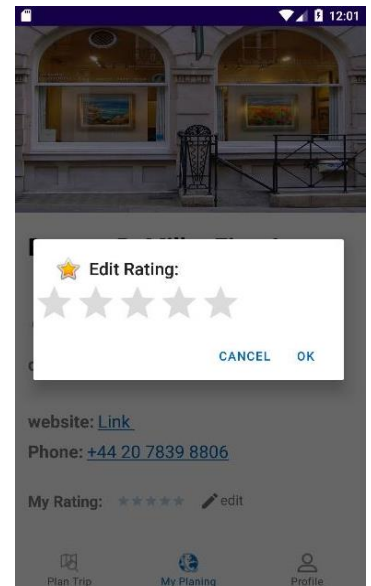
### מסך פרטי טיול:

במסך זה יוצגו פרטי הטיול ורשימת ימי הטיול. לחיצה על יום תוביל למסך פרטי יום בטיול.



### מסך פרטי יום בטיול:

במסך זה תוצג עבור כל יום רשימת אתרי התיירות המתוכננים ליום זה. לחיצה על אתר תיירות תעביר למסך פרטי אתר תיירות.



### מסך פרטי אתר תיירות:

במסך זה יוצגו פרטי אתר התיירות. בתחתית המסך יוצג הדירוג שהמשתמש דירג את האתר עם אפשרות עריכה.

## בסיס נתונים

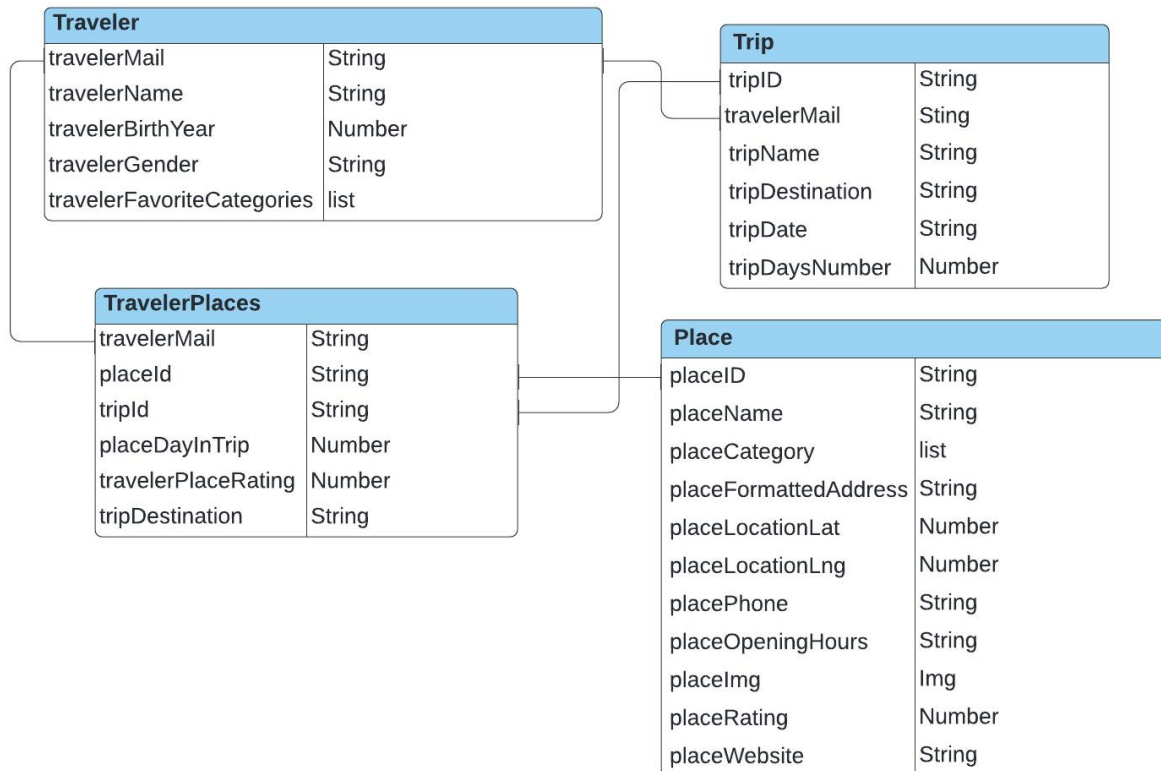
החלטנו לשמור את הנתונים שלנו בבסיס הנתונים MongoDB.  
 התלבטנו לגבי 3 בסיסי נתונים: MongoDB, Firebase, MySQL

	MongoDB	Firebase	MySQL
SQL	noSQL	noSQL	SQL
Performance	MongoDB provides high performance with high traffic apps	Firebase has inferior performance than MongoDB	slower than MongoDB while dealing with large databases
Security	more Secure than Firebase	Firebase is not as Secure as MongoDB	more secure than MongoDB
Applications	MongoDB is best suitable for large-scale applications	Firebase is ideal for small-scale applications	better for read-mostly applications
data representation	Json	Json	tables and rows
Supported programming languages	java javaScript	java javaScript	SQL

החלטנו לבחור ב MongoDB כי ניתן לאחסן בו כמויות גדולות של מידע ומהירות הביצועים שלו גבוהה יחסית למסדי הנתונים האחרים.

יצרנו את ה collections הבאים:

- **Traveler** - מכיל את פרטי המשתמש כולל קטגוריות אתרי תיירות האהובות על המשתמש
- **Trip** - מכיל את פרטי הטיול עבור כל משתמש וכן את המזהה של המשתמש שיצר את הטיול.
- **Place** - מכיל את פרטי אתר התיירות כולל שם, כתובת, מיקום וכו'.
- **TravelerPlaces** - מכיל את אתרי התיירות שמשתמש בחר להוסיף לטיוליו. עבור כל אתר ישמר באיזה טיול הוא התווסף ושל איזה משתמש, והדירוג של המשתמש במידה והמשתמש דירג את האתר.



## 9. פיתוח עתידי

- מסלול אופטימלי תוך התחשבות בשעות פתיחת וזמן שהות באתרים ותכנון מסלול יומי לפי שעות.
- תכנון מסלול לפי תקציב המשתמש.
- מערכת המלצות שתמליץ על אתרים לפי גיל המשתמש.
- המלצות של אתרי תיירות ברדיוס מסוים על סמך מיקום המלון ולפי הגבלת המשתמש.
- הוספת המלצות על מסעדות בקרבת מקום האטרקציות עבור יום מסוים.
- הצגת מסלול מדויק מאטרקציה לאטרקציה על סמך google map.
- הצגת מפה של הטיול וסימון אתרי התיירות המתוכננים כולל צביעה לפי חלוקתם לימי הטיול.

- [1] unwto, "INTERNATIONAL TOURISM GROWTH CONTINUES TO OUTPACE THE GLOBAL ECONOMY," 20 1 2020. [Online]. Available: <https://www.unwto.org/international-tourism-growth-continues-to-outpace-the-economy>.
- [2] C. D. Huang, J. Goo, K. Nam and C. W. Yoo, "Smart tourism technologies in travel planning: The role of exploration and exploitation," *Information \& Management*, vol. 54, no. 6, pp. 757--770, 2017.
- [3] M. Stavarakis, P. Koutsabasis, D. Gavalas, V. Damianos and Spyros, "TouristHub: user experience and interaction design for supporting tourist trip planning," in *2020 IEEE 10th International Conference on Intelligent Systems (IS)*, IEEE, 2020, pp. 370-379.
- [4] S. L. M. Belaidan, L. Y. Yee, N. A. Abd Rahman and K. S. Harun, "Implementing k-means clustering algorithm in collaborative trip advisory and planning system," *Periodicals of Engineering and Natural Sciences*, vol. 7, no. 2, pp. 723-740, 2019.
- [5] F. Tahmasebi, M. Meghdadi, S. Ahmadian and K. Valiallahi, "A hybrid recommendation system based on profile expansion technique to alleviate cold start problem," *Multimedia Tools and Applications*, vol. 80, no. 2, pp. 2339-2354, 2021.
- [6] H. Alrasheed, A. Alzeer, A. Alhowimel and A. Althyabi, "A multi-level tourism destination recommender system," *Procedia Computer Science*, vol. 170, pp. 333-340, 2020.
- [7] M. I. Malinen and P. Franti, "Balanced k-means for clustering," in *Joint iapr international workshops on statistical techniques in pattern recognition (spr) and structural and syntactic pattern recognition (sspr)*, Springer, 2014, pp. 32-41.