

רח' סעדיה גאון 15
 פתח תקוה 49463
 ת.ד. 103 מ"י. 49106
 טל. 03 9245315
 פקס. 03 9214408
 *
 סמל מוסד 490128

לכבוד

יחידת הפרויקטים מה"ט

הצעה לפרויקט גמר

פרטי הסטודנטים:

שם	ת.ז.	טלפון	כתובת	תאריך סיום לימודים
תמר צוברי	211540471	0533150572	שמעון בן שטח 13 אלעד	מאי 2021
תהילה רפאלי	207647702	0583291728	פינקל 2 פתח תקווה	מאי 2021

שם המכללה: סמינר בית יעקב פתח תקווה. **סמל המכללה:** 76053

מסלול ההכשרה: הנדסאים. **מגמת לימוד:** הנדסת תוכנה.

מקום ביצוע הפרויקט: בבית.

פרטי המנחה:

שם	מקום עבודה	טלפון	כתובת	תואר
שלומית זקש	סמינר ב"ע פ"ת	0533117151	סלומון 1 פ"ת	B.A

1. שם הפרויקט.

היהודי הנודד.

2. רקע.

2.1. תיאור ורקע כללי.

האתר מהווה כלי עזר בסיסי לתייר היהודי בכל מקום בו הוא נמצא בעולם. האתר כולל מידע על: בתי כנסיות, מסעדות כשרות, קהילות יהודיות וכו' ברחבי העולם. הוא נותן הצעות לטיולים על סמך זמני פתיחת אתרים וזמני תפילות ועוד, ובכך מקל על המטיילים לקבל שירותים ההולמים את צרכיהם הדתיים.

רבני קהילות - אף רחוקות ולא מוכרות, יכולים להציג את קהילתם באתר ולספר על הפעילות שהקהילה שלהם מארגנת, ובכך לחשוף אותה לעיני הציבור ואף להרחיבה. הם יכולים לתרום מניסיונם הרב בהתמודדות עם תנאי המקום ליהודים אחרים ולעזור להם.

האתר תורם לקשר שבין היהודי היחידי לבין הקהילות היהודיות ברחבי העולם וכן תורם לקשר ביניהן ולשיתופי פעולה.

2.2. מטרת המערכת.

1. לאפשר למשתמש לטייל ולהתאקלם בנוחות בכל מקום, תוך שמירה על אורח חייו.
2. לאפשר למשתמש לגשת למידע ולהתאים אותו לצרכיו.
3. לאפשר לרבנים בכל העולם לנהל את מערכת המידע של המערכת. ולהכניס נתונים חדשים בצורה מהירה ונוחה.
4. לאפשר לרבנים לעזור ליהודים ולתרום מניסיונם המקצועי בנוגע להתמודדות עם חיים בגולה.
5. לאפשר לקהילות יהודיות רחוקות להיחשף לציבור היהודי בעולם.
6. לאפשר גישה מכל מקום ובכל זמן למידע אודות כל קהילה יהודית ברחבי תבל.

3. סקירת המצב הקיים בשוק - אילו בעיות קיימות.

מערכת כזו לא קיימת. אין קשר בין הקהילות היהודיות ברחבי העולם. יהודי דתי הנוסע למקום כלשהוא אם לצורך עסקים או טיול וכדומה, אינו יודע אם ימצא שם את צרכיו הדתיים. וגם אם יודע, צריך לעבוד מאוד קשה כדי למצוא בית כנסת וכו'. לעיתים מגיע לאתר כלשהוא ומגלה שהאתרקציה אינה הולמת את אורח חייו ועוזב את המקום במפח נפש. אתר זה מונע את עוגמת הנפש הזו בכך שמאגד בתוכו את כל האתרים הראויים.

4. מה הפרויקט אמור לחדש או לשפר?

האתר חדשני ויחיד מסוגו חוסך מהמשתמשים עבודה רבה ולעיתים בלתי אפשרית של חיפוש אתרים ומקומות התואמים את אורח חייו.

5. דרישות מערכת ופונקציונאליות.

5.1. דרישות מערכת, סביבת הטמעה ושימוש.

המערכת תעבור קומפילציה והפצה בסביבת visual studio code . תשתית לשרת היא סביבת Express.js Framework לשפת Node.js המאפשרת קבלת בקשות לתצוגת דפי אינטרנט.

המשתמש יוכל להריץ את האתר בכל מכשיר אשר מותקן עליו דפדפן אינטרנט.

5.2. שרידות ביצועים והתמודדות עם עומסים.

צד השרת מריץ Express.js המסוגל להתמודד עם מספר קריאות של קריאות בו זמנית.

גם עומס על שרת ה-MySQL אינו צפוי בסדר גודל כזה של אתר מכיוון שהוא בנוי להתמודדות בהצלחה עם עומסים כבדים בהרבה.

5.3. דרישות פונקציונאליות.

רשימת דרישות המשתמש מהמערכת:

- 5.3.1. המערכת מאפשרת למשתמש להירשם למערכת ולזכות במרחב אישי ושמידת חיפוש.
- 5.3.2. המערכת מאפשרת למשתמש להיכנס למרחב האישי על ידי הכנסת כתובת מייל (שם משתמש) וסיסמא.
- 5.3.3. המערכת מאפשרת למשתמש לצפות בתוכן האתר.
- 5.3.4. המערכת מאפשרת למשתמש לבחור מדינה\עיר אשר ירצה לצפות בנתוניו.

- 5.3.5. המערכת מאפשרת למשתמש לצפות בשירותים לפי מסננים שבחר.
- 5.3.6. המערכת מאפשרת למשתמש לשמור במועדפים את השירותים שמצאו חן בעיניו.
- 5.3.7. המערכת מאפשרת למשתמש להזין דרישות לתכנון טיול ולצפות באפשרויות שניתנו לו.
- 5.3.8. המערכת מאפשרת למשתמש להדפיס נתונים שונים מהאתר.
- 5.3.9. המערכת מאפשרת למשתמש החדש גישה רק לאחר שאימתה את כתובת המייל.
- 5.3.10. המערכת מאפשרת למשתמש לקבל את תכנון המסלול שהוצע לו למייל האישי.
- 5.3.11. המערכת מאפשרת למנהל ולרבנים להכניס, למחוק ולעדכן נתונים.

6. בעיות צפויות במהלך הפיתוח, ופתרונות תפעוליות, טכנולוגיות, עומס וכו'.

6.1. תיאור הבעיות.

כפועל יוצא של דרישות המשתמש מהתוכנה:

- **בעיה 1 :** כיצד נמנע מצב בו משתמש מנסה לשנות נתונים באתר?
- **בעיה 2 :** כיצד נוודא שכתובת המייל שהכניס המשתמש החדש אכן נכונה ושייכת לו?

6.2. פתרונות אפשריים.

- **לבעיה 1 :**
 - א. כאשר מנסים לשנות נתונים המערכת תבקש לאשר את רמת הגישה.
 - ב. המערכת תציע למשתמש למלא טופס מיוחד בו יפרט את מהות השינוי וסיבתו. הטופס ישלח לצוות המנהלים ובמידה והפניה תמצא ראויה, הנתונים ישתנו.
- **לבעיה 2 :**
 - א. לשלוח מייל לכתובת. אם לא התקבלה הודעת שגיאה, הכתובת תקינה.
 - ב. לשלוח מייל לכתובת. המשתמש יצטרך לאשר את קבלת המייל.

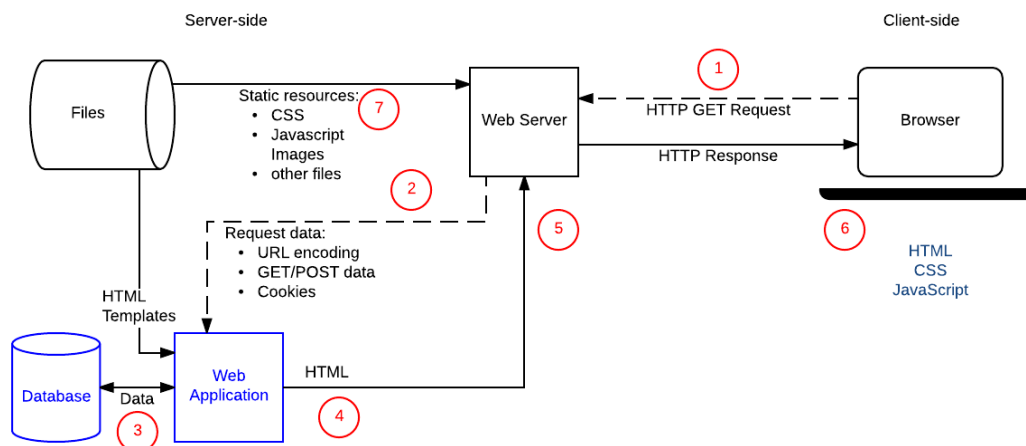
הפתרון הנבחר עבור כל אחת מהבעיות:

- **לבעיה 1 :** הפתרון הנבחר הוא פתרון א'. משום שאם נשתמש בפתרון השני, קשה מאוד לבעלי רמת גישה 2 – הרבנים, לטפל בנתונים שתחת אחריותם. כי על כל שנוי יצטרכו להגיש בקשה למנהלים, בזמן שהם אלו שבאמת מעודכנים בכל הפרטים שבאזור אחריותם ושולטים בפרטים. מה גם, שפתרון השני מהווה הכבדה על עבודת המנהלים, כי מסתבר ורוב הפניות לא ימצאו כרלוונטיות. אם למשתמש תהיה הערה\הארה חשובה מאוד, יוכל לצור קשר עם הרב הרלוונטי.
- **לבעיה 2 :** הפתרון הנבחר הוא פתרון ב'. הפתרון הראשון אולי מאשר שהכתובת תקינה אך לא מאמת את זהות בעליה, לכן הוספנו שהמשתמש יצטרך לאשר בעלות על כתובת זו. דרך נוחה, קלה ובטוחה, עם מאה אחוזי אמינות.

7. פתרון טכנולוגי נבחר.

7.1. טופולוגית הפתרון.

המערכת מורכבת משרת Express ל- Node.js המריץ את האתר בסביבת ה- Server, מסד נתונים - של MySQL ממשק משתמש בצד הלקוח: דפדפן אינטרנט כלשהו כמו chrome על מחשב שולחני או על מכשיר iOS/ android



טכנולוגיות בשימוש:

React.js: ספריית קוד פתוח מבית פייסבוק שמטרתה לאפשר פיתוח נוח ומהיר של אתר אינטרנט ואפליקציות.

MySQL: חבילה ל-Node המאפשרת את הגישה למסד הנתונים.

Web Api: טכנולוגיה המקשרת בין צד שרת - כלומר מסד נתונים לבין צד הלקוח - React.

7.2 שפות הפיתוח.

שפות הפיתוח בצד השרת:

Node.js - Express: Node.js הוא מפרש לשפת JavaScript שנכתב ב C++ ומבוסס על מנוע V8. הוא תוכנן לכתיבת אפליקציות רשת סקילאביליות כגון שרתי HTTP. Express זהו Framework שפותח ל-Node כדי לאפשר תקשורת ע"י בקשות HTTP.

שפות הפיתוח בצד הלקוח:

Typescript: שפת תכנות דינמית מונחית עצמים מבוססת מחלקות המותאמת לשילוב באתרי אינטרנט ורצה על ידי דפדפן האינטרנט בצד הלקוח. השפה מרחיבה את יכולות שפת התגיות הבסיסית HTML ומאפשרת בכך ליצור יישומי אינטרנט מתוחכמים יותר.

React: שפת פיתוח צד לקוח המוסיפה אבסטרקציות מעולם פיתוח Object Oriented, הכתיבה מתבצעת ע"י כתיבת JS ורנדור התצוגה ע"י בניית HTML דרך הקוד. יצירת Store ראשי כדי לשלוט על המידע של האפליקציה ועל שינויים שחלים בו. כתיבת Components כדי לשלוט על ה- Templates, כתיבת Services כדי לשלוט על הלוגיקה, ושילוב של Components ו- Services ליצירת Modules.

HTML5: היא שפת בניית אתרים לסימון טקסט, שפת תגיות, אשר נותנת הנחיות לדפדפנים בנוגע לאופן הצגתו של דף האינטרנט מבחינת טקסט, תמונות, טבלאות ועיצוב ועוד. זוהי שפת הקוד הבסיסית לסביבת ה-Web עבור בניית אתרים. HTML5 כוללת כלים חדשים שבאו להקל על עבודת המפתחים.

7.3 תיאור הארכיטקטורה הנבחרת.

הארכיטקטורה הנבחרת היא חלוקה ל - 3 שכבות: **Tier Architecture-3**.

בפרויקטים שמבוססים על Database מומלץ להשתמש בגישה של **מודל השכבות**, בארכיטקטורה זו קיימת הפרדה בין השכבות השונות בפרויקט.

היתרון העיקרי בחלוקה לשכבות הוא **תחזוקה קלה**, במיוחד במערכות גדולות, כך שניתן יחסית בקלות להחליף כל שכבה בלי לגעת בשכבות האחרות כאשר נרצה לעשות שינויים בתוכנית.

7.4 חלוקה לתכניות ומודולים.

1. שכבת תצוגה (Presentation Layer) - שכבת ממשק המשתמש (UI - User Interface).
שכבת התצוגה מדפי HTML באתר אינטרנט.
השכבה מתקשרת בין המשתמש לבין שאר השכבות של המערכת.

2. שכבת הלוגיקה העסקית (BL - Business Logic) - השכבה שאמונה על הלוגיקה של המערכת, עוסקת בעיבוד המידע, בחישובים שונים ושליחתו לשכבת התצוגה.
בשכבה זו נממש את הפונקציונאליות של המערכת.

3. שכבת הנתונים (DAL - Data Access Layer) - שכבה זו מורכבת ממקור נתונים – מסד הנתונים שלנו, ומספריית JS - MySQL אשר תפקידה לקרוא את המידע הנדרש למערכת, לשמור את העדכונים, ולהוסיף מידע חדש או למחוק פרטי מידע קיימים.

7.5 סביבת השרת.

לצורך הפרויקט נשתמש בשרת מקומי - Express.js, המסופק על ידי ה - npm.
אם האתר יירכש על ידי לקוח, נעלה אותו לשרת אירוח כלשהו .

7.6 ממשק המשתמש / לקוח - GUI.

שכבת ה GUI מורכבת מדפי HTML שמוצגים למשתמש דרך הדפדפן.

7.7 ממשקים למערכות אחרות / API

WebApi

7.8 שימוש בחבילות תוכנה.

JQuery, CSS, Express, MySQL

8 שימוש במבנה נתונים וארגון קבצים.

8.1 מבנה הנתונים.

הרשאה: קוד, רמת הרשאה.

משתמשים: קוד, שם משתמש, מדינה, שפה, מייל, סיסמא, הרשאה.

הנהלה: קוד, שם משתמש, מדינה, שפה, מייל, סיסמא, הרשאה, קהילה, טלפון.

זמנים: קוד, זמן התחלה, זמן סיום, שירות.

סוג שירות: קוד, שם שירות.

שירותים: קוד, שם, סוג שירות, טלפון, כתובת, עיר, מדינה.

מלונות: קוד, שם, סוג שירות, טלפון, כתובת, עיר, מדינה, מנהל, מספר כוכבים.

מסעדות: קוד, שם, סוג שירות, טלפון, כתובת, עיר, מדינה, הכשר.

אתרים: קוד, שם, סוג שירות, טלפון, כתובת, עיר, מדינה, סוג פעילות, גילאים.

סיורים: קוד, שם, סוג שירות, טלפון, כתובת, עיר, מדינה, מדריך, טלפון מדריך, מקום יציאה, משך זמן, מסלול, גילאים, תיאור.

בתי כנסת: קוד, שם, סוג שירות, טלפון, כתובת, עיר, מדינה, רב, נוסח, קהילה.

הרצאות: קוד, מרצה, נושא, תאריך, שעה, משך זמן, בית כנסת.

ימי זיכרון: קוד, תאריך, נושא, תיאור, בית קברות.

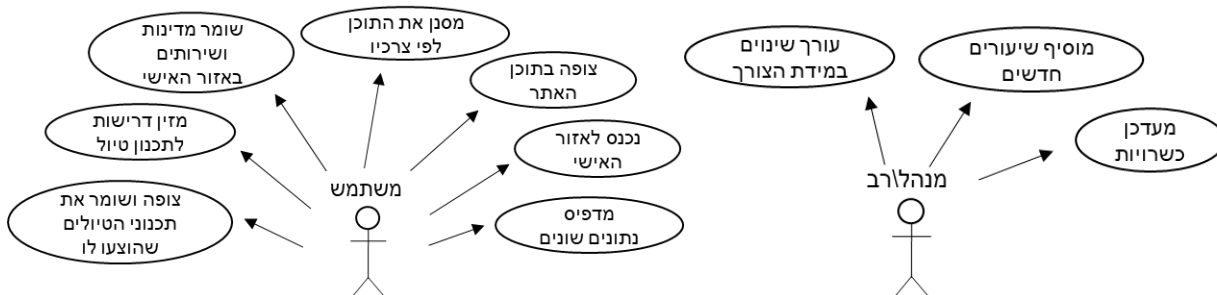
8.2 שיטת האחסון.

שיטת האחסון היא ע"י MySQL Server - מסד נתונים.
הגישה למסד הנתונים היא ע"י ספריית MySQL - JS.

8.3 נפילה \ קריסה \ תמיכה.

MySQL Server בנוי לתמיכה במס' גדול של קריאות בו זמניות ואין חשש לקריסה ברמה של הפרויקט.

9. תרשימי מערכת מרכזיים.



10. תיאור המרכיב האלגוריתמי \ חישובי.

10.1 איזה בעיה בא לפתור. איך יפתור?

הבעיה: כיצד נמצא למשתמש את המסלול המיטבי?
הפתרון: נכתוב פונקציה שתתחשב בכל זמני הפתיחה של כל האתרים והפרשי המעבר ביניהם. הפונקציה תחשב באמצעות אלגוריתם את המסלול הטוב ביותר והמנוצל ביותר מבחינת הספקים.

10.2 איסוף מידע ונתונים סטטיסטיים - אנליטיקות.

באתר יוצגו הנתונים הסטטיסטיים באחוזים שמשקפים למשתמשים את המקום הנצפה ביותר מכל שרות, לפי המדינה אותה בחרו.

11. תיאור התייחסות לנושא אבטחת המידע.

יש לדאוג לאבטחת השרת מעומסים מופרזים הנגרמים על ידי גורמים זדוניים, ואת שרת ה-MySQL מ-SQL Injection.

משתמש הגולש באתר מריץ אותה באופן אבסולוטי. קוד המשתמש ישלח לשרת ויצטרף לכל רשומה הנשמרת עבורו. לא יתכן מצב בו משתמש מריץ אפליקציה בתור משתמש אחר.

ציין מס' מקרים ותגובות להם ניתן מענה אבטחתי:

- כניסת משתמש על ידי אמיל וסיסמא, כך שלכל משתמש אזור אישי ופרטי שאינו חשוף למשתמשים האחרים.
- שליחת אישור הרשמה למייל כדי לוודא שהאמייל שהוזן אכן שייך לנרשם.
- רמת הרשאת גישה גבוהה יותר למנהל ולרבנים כדי שלא כל משתמש יוכל לשנות את הנתונים באתר.
- במקרה שבעת כניסת משתמש הסיסמא אינה תואמת לשם המשתמש שהקיש, המערכת תציג לו הודעת שגיאה ולא תאפשר כניסה.

12. משאבים הנדרשים לפרויקט.

12.1 מספר שעות המוקדש לפרויקט והחלוקה בין חברי הצוות.

מספר השעות המוקדשות לפרויקט הוא 700. החלוקה בין חברות הצוות היא 350 שעות לכל אחת.

12.2 ציוד נדרש.

CPU i5, RAM 8GB, HD SSD , חיבור אינטרנט.

12.3 תוכנות נדרשות.

Browsers, Visual Studio Code, Workbench

12.4 ידע חדש שנדרש לצורך ביצוע הפרויקט.

הרחבת הידע בשפת React ו-Node.js.

12.5 ספרות ומקורות מידע. אקספרס ומטריאל

<https://reactjs.org/>
<https://reactjs.org/tutorial/tutorial.html>
<https://stackoverflow.com/>
<https://github.com/>
<https://www.npmjs.com/>
<https://www.geeksforgeeks.org/>
<https://material-ui.com/>
<https://www.react.express/>

13. תכנית עבודה ושלבנים למימוש הפרויקט.

- 13.1.** ייזום רעיון, ניתוח מערכת, הכרת השטח ואפיון מסד נתונים - עד מאי 2020.
- 13.2.** תכנון UI-UX, עיצוב גריד, זרימת מסכים - עד יוני 2020.
- 13.3.** כתיבה לוגית עסקית - עד ספטמבר 2020.
- 13.4.** כתיבת ממשק משתמש ועיצוב - עד נובמבר 2020.
- 13.5.** בדיקות QA - עד דצמבר 2020.
- 13.6.** התקנה והטמעה - עד ינואר 2021.

14. תכנון הבדיקות שיבוצעו.

מספר בדיקה	מספר דרישה במסמך אפיון	מקרי הבדיקה	ידנית \ אוטומטית	חשיבות	הערות
1.	5.3.1	בדיקת יצירת משתמש חדש (האם קיים כבר במערכת)	ידנית	גבוהה	
2.	5.3.2	בדיקת התאמת סיסמא	ידנית	גבוהה	
3.	5.3.5	בדיקת מיון הנתונים לפי מסננים	ידנית	בינונית	
4.	5.3.9	בדיקת נכונות כתובת המייל	ידנית	גבוהה	
5.	5.3.11	בדיקת רמת הרשאת הגישה	ידנית	גבוהה	

15. בקרת גרסאות.

תישמרנה גרסאות במהלך הפיתוח (תלוי במידת ההתקדמות היומית), כל גרסה תיבדק בהשוואה לגרסה הקודמת. לכל גרסה נצרף תיאור המסביר את תוכן הגרסה וההבדלים מהגרסה הקודמת.

52

חתימת המנחה האישי

א. רפאלי ת. צוברי

תהילה רפאלי תמר צוברי

חתימת הסטודנט

הערות ראש המגמה במכללה:

אישור ראש המגמה:

שם: שולמית קמינסקי חתימה: שולמית קמינסקי תאריך: 7/11/2020

הערות הגורם המקצועי מטעם מה"ט:

אישור הגורם המקצועי מטעם מה"ט:

שם: חתימה: תאריך: