הצעת פרויקט

**אורי שניידר ת"ז - 308403047**



[1. שם הפרויקט: 3](#_heading=h.30j0zll)

[2. רקע: 3](#_heading=h.1fob9te)

[2.1 תיאור רקע כללי: 4](#_heading=h.3znysh7)

[2.2 מטרות המערכת: 4](#_heading=h.2et92p0)

[3. סקירת מצב קיים: 4](#_heading=h.tyjcwt)

[4. מה הפרויקט אמור לחדש: 4](#_heading=h.3dy6vkm)

[5. דרישות מערכת: 4](#_heading=h.1t3h5sf)

[5.1 דרישות מערכת 4](#_heading=h.4d34og8)

[5.2 דרישות פונקציונאליות 5](#_heading=h.2s8eyo1)

[6. בעיות צפויות במהלך הפיתוח ופתרונות: 5](#_heading=h.17dp8vu)

[6.1 תיאור הבעיות: 5](#_heading=h.3rdcrjn)

[6.2 פתרונות אפשריים: 6](#_heading=h.26in1rg)

[7. פתרון טכנולוגי נבחר: 6](#_heading=h.lnxbz9)

[7.1 טופולוגית הפתרון: 6](#_heading=h.35nkun2)

[7.2 טכנולוגיות בשימוש: 7](#_heading=h.1ksv4uv)

[7.3 שפות פיתוח: 7](#_heading=h.44sinio)

[7.4 ארכיטקטורה: 8](#_heading=h.2jxsxqh)

[7.5 חלוקה למודולים: 8](#_heading=h.z337ya)

[7.6 סביבת שרת: 8](#_heading=h.3j2qqm3)

[7.7 ממשק משתמש: 8](#_heading=h.1y810tw)

[7.8 ממשקים למערכות אחרות / API: 8](#_heading=h.4i7ojhp)

[7.9 שימוש בחבילות תוכנה: 8](#_heading=h.2xcytpi)

[8. שימוש במבני נתונים וארגון קבצים 9](#_heading=h.1ci93xb)

[8.1 מבני נתונים: 9](#_heading=h.3whwml4)

[8.2 אחסון נתונים: 9](#_heading=h.2bn6wsx)

[8.3 התאוששות וטרנזקציות: 10](#_heading=h.qsh70q)

[9. תרשימי מערכת מרכזיים. 10](#_heading=h.3as4poj)

[9.1 Use Case: ראה נספח א. 10](#_heading=h.1pxezwc)

[9.2 Sequence diagram: ראה נספח ב. 10](#_heading=h.49x2ik5)

[10. תיאור המרכיב האלגוריתמי – חישובי: 10](#_heading=h.2p2csry)

[10.1 איזה בעיה בא לפתור, איך יפתור? 10](#_heading=h.147n2zr)

[10.2 איסוף מידע וניתוחים סטטיסטיים (אנליטיקות): 10](#_heading=h.3o7alnk)

[11. תיאור/התייחסות לנושאי אבטחת מידע 11](#_heading=h.23ckvvd)

[12. משאבים הנדרשים לפרויקט: 11](#_heading=h.ihv636)

[12.1 מספר שעות המוקדשות לפרויקט: 11](#_heading=h.32hioqz)

[12.2 ציוד נדרש: 12](#_heading=h.1hmsyys)

[12.3 תוכנות נדרשות: 12](#_heading=h.41mghml)

[12.4 ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט: 12](#_heading=h.2grqrue)

[12.5 ספרות ומקורות מידע: 12](#_heading=h.vx1227)

[13. תכנית עבודה ושלבים למימוש הפרויקט: 13](#_heading=h.3fwokq0)

[14. תכנון הבדיקות שיבוצעו : 13](#_heading=h.1v1yuxt)

[14.1 בדיקות תהליכיות: 14](#_heading=h.4f1mdlm)

[14.2 בדיקות יחידה: 14](#_heading=h.2u6wntf)

[15. בקרת גרסאות (control version): 15](#_heading=h.19c6y18)

[נספחים 16](#_heading=h.3tbugp1)

[א.](#_heading=h.28h4qwu) Use Case 16

[ב.](#_heading=h.nmf14n) חיפוש נפטר - Sequence diagram 17

## שם הפרויקט:

"זיכרון עולם".

## 2. רקע:

### 2.1 תיאור רקע כללי:

האתר מאפשר למשפחות וחברים של נפטרים להנציח את זכרם על ידי זיכרונות ולימוד משניות לעילוי נשמתם.

### 2.2 מטרות המערכת:

לאפשר לימוד משניות לעילוי נשמות, לאפשר למשפחות וחברים להוסיף את פרטי הנפטרים תמונות וסיפורים עליהם , לאפשר מעקב אחר מספר המשניות שנלמדו לזכר כל נפטר.

## 3. סקירת מצב קיים:

קיימים כיום אתרי הנצחה מקוונים רבים, אך לא קיים פתרון ייעודי ללימוד משניות לעילוי נשמה באופן מסודר ומקוון.

## 4. מה הפרויקט אמור לחדש:

הפרויקט יאפשר לראשונה לימוד משניות לעילוי נשמה באופן מקוון,עם האפשרות לניהול נכון של הלימוד מה שימנע כפילויות של מסכתות עד לסיום ששת הסדרים , עם מעקב אחר מספר המשניות שנלמדו לזכר כל נפטר.

## 5. דרישות מערכת:

### 5.1 דרישות מערכת

סביבת הטמעה:

* המערכת תפעל בסביבת האינטרנט (web application) עם ממשק משתמש אינטואיטיבי וידידותי.
* המערכת תתמוך בדפדפנים נפוצים (Chrome, Firefox, Safari).

שרידות:

* הדאטה בייס יגובה בקבצי SQL.
* קוד המקור יגובה בgithub.

התמודדות עם עומסים:

* המערכת תוכל להתמודד עם עומסים של עד 100 משתמשים בו זמנית.

### 5.2 דרישות פונקציונאליות

המערכת תאפשר למשתמשים לחפש אחר דפי זיכרון של נפטרים אחרים, לפי שם, תאריך,ואפשרויות נוספות.

המערכת תאפשר למשתמשים להירשם ולהתחבר עם פרטים אישיים.

המערכת תאפשר למשתמשים להצטרף ללימוד משניות לעילוי נשמת נפטר, ולקבל מייל התראה על התקרבות יום הפטירה.

המערכת תאפשר למנהלים ליצור דף זיכרון לנפטר, ולהוסיף את שמו, ותאריך הפטירה, תמונות, סיפורים, וקישורים למשניות שנבחרו ללימוד לעילוי נשמתו.

## 6. בעיות צפויות במהלך הפיתוח ופתרונות:

### 6.1 תיאור הבעיות:

בעיה א: איך להבטיח שהמשניות שנלמדו לעילוי נשמה אינן חוזרות על עצמן ?

בעיה ב: איך להתמודד עם תוכן לא הולם שיכול להיתווסף על ידי המשתמשים לדפי הזיכרון?

בעיה ג: איך למנוע מורכבות למי שמקבל משניות כדי שלא יווצר חסם להתנדב לזה?

בעיה ד: איך לשמור תאריכים עבריים בSQL ?

בעיה ה: איך לבדוק עם זה שנה מעוברת וחודש מלא או חסר

### 6.2 פתרונות אפשריים:

פתרון לבעיה א: להשתמש במאגר נתונים של כל המשניות הקיימות, ולאפשר למשתמשים לבחור רק ממשניות שעוד לא נלמדו לזכר הנפטר.

פתרון לבעיה ב: לאפשר למי שהוסיף את הנפטר לנהל את העמוד ולהסיר זיכרונות שאינם מתאימים.

פתרון לבעיה ג: ליצור לו רישום רק עם מייל וסיסמא וללא פרטים נוספים שדורשים מורכבות מהמשתמש.

פתרון לבעיה ד:ישנם מספר אפשרויות כסטרינג בDate רגיל לאחר המרה או במספרים שמשמעותם יום חודש ושנה.

פתרון לבעיה ה: שימוש בספריות שמטפלות בזה.

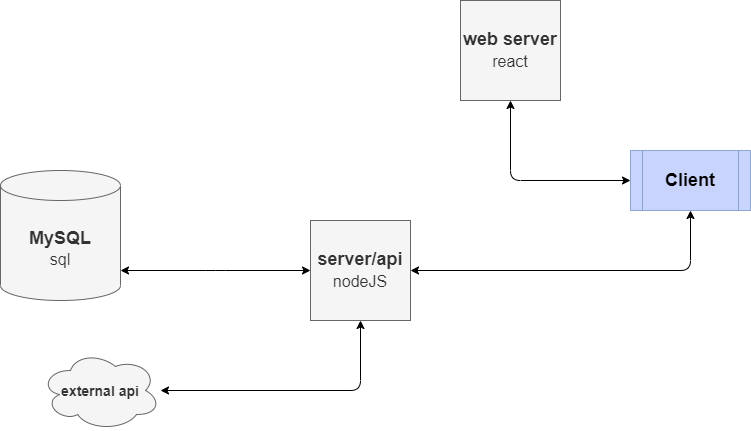
## 

## 7. פתרון טכנולוגי נבחר:

**7.1 טופולוגית הפתרון:**

המערכת תורכב מהרכיבים הבאים:

* שרת API - שרת ניהול הבקשות והנתונים של המערכת, ממומש ב-Node.js
* מסד נתונים SQL - מסד נתונים יחסי (MySQL) אשר יאחסן את כל המידע של המערכת.
* ממשק משתמש - הממשק של האפליקציה אליו ניגש המשתמש דרך הדפדפן, ימומש ב-React.



בשלב זה, המערכת תיושם בצורה לוקאלית על מחשב אישי עבור צורכי פיתוח ובדיקות התחלתיות.

בהמשך לאחר הפיתוח תיבדק האפשרות להעלות לענן לצורך סקלביליות וזמינות גבוהים יותר.

### 7.2 טכנולוגיות בשימוש:

* Node.js - מאפשר פיתוח שרת יעיל וקל לתחזוקה ב - JavaScript.
* React - ספריית UI פופולרית עם יכולות מתקדמות.
* MySQL - מסד נתונים יחסי ותואם תקן שנתמך היטב.
* axios - ספריית JavaScript לביצוע וניהול בקשות HTTP בצורה קלה וטובה.

### 7.3 שפות פיתוח:

* HTML - שפת סימון ליצירת מבנה ותוכן בדפי אינטרנט.
* CSS - לעיצוב ועימוד של HTML.
* JavaScript - שפת פיתוח פופולרית עם תמיכה נרחבת וקהילה גדולה.
* SQL - לגישה למסד הנתונים.

### 7.4 ארכיטקטורה:

ארכיטקטורת שרת-לקוח ארכיטקטורה נפוצה מאד המאפשרת חלוקה ברורה לרכיבים וסקלביליות קלה לתיחזוק ואבטחה

### 7.5 חלוקה למודולים:

* מודול ממשק משתמש
* מודול שרת
* מודול מסד נתונים
* מודול לוגיקת עסקית

### 7.6 סביבת שרת:

בתחילה שרת מקומי לצורכי פיתוח. בהמשך תיבדק האפשרות להעלות לענן.

### 7.7 ממשק משתמש:

ממשק דינאמי ואינטראקטיבי בטכנולוגיית React.

### 7.8 ממשקים למערכות אחרות / API:

*sefaria APIs* API ללימוד משניות.

### 7.9 שימוש בחבילות תוכנה:

* express - מסגרת עבודה ל-NodeJS
* mysql2 - ממשק MySQL
* @hebcal/core - עבודה עם תאריכים עבריים
* @mui/joy - רכיבי UI
* axios - בקשות HTTP
* cookie-parser - פענוח קובצי Cookie
* cors - אפשרות בקשות ממקורות שונים
* dotenv - טעינת משתני סביבה
* hebcal - לוח שנה עברי
* joi - ולידציה
* react - ספריית ממשק משתמש

## 8. שימוש במבני נתונים וארגון קבצים

### 8.1 מבני נתונים:

המערכת תשתמש במסד נתונים יחסי MySQL שיכיל בין היתר את הטבלאות הבאות:

* נפטרים - טבלה לאחסון פרטי הנפטרים.
* מנהלים - טבלה לאחסון פרטי מנהלי המערכת שיכולים להוסיף נפטרים.
* משתמשים - טבלה לאחסון פרטי המשתמשים הרשומים.
* סיפורים - טבלה לאחסון הזיכרונות והסיפורים שהתווספו על כל נפטר.
* התחייבויות - טבלת קישור בין המשתמשים לנפטרים שהם לומדים לזכרם, כוללת פרטי ההתחייבות כמו מסכת ותאריך התחייבות.

טבלאות נוספות לפי הצורך.

### 8.2 אחסון נתונים:

* נתוני המערכת יאוחסנו במסד נתונים יחסי MySQL.
* קבצי הגיבוי של מסד הנתונים ישמרו כקבצי SQL.
* קוד המערכת יאוחסן במערכת גרסאות GIT ובפרט בGitHub.

### 8.3 התאוששות וטרנזקציות:

* גיבויים תקופתיים של מסד הנתונים יאפשרו שחזור מנפילה\קריסה.
* עבודה מול GitHub תאפשר גרסאות ושחזור קוד.
* מסד הנתונים MySQL תומך בטרנזקציות ומנגנוני התאוששות.

## 9. תרשימי מערכת מרכזיים.

### 9.1 Use Case: ראה נספח א.

### 9.2 Sequence diagram: ראה נספח ב.

## 10. תיאור המרכיב האלגוריתמי – חישובי:

### 10.1 איזה בעיה בא לפתור, איך יפתור?

1. איך לאחסן תאריך עברי בSQL ? ע"י חלוקה לשנים חודשים וימים לחוד.

2. איך להציג למשתמש את כל השנים העבריות לבחירה ? לולאה שעוברת על כל המספרים עד השנה העברית הנוכחית ופונקציה שמקבלת מספר והופכת אותו לשנה עברית מתאימה.

3.אלגוריתם לסינון הנפטרים שיצגו ע"פ אפשרויות נבחרים.

4.איך להביא את כל הנפטרים שהיארצייט (תאריך פטירה ע"פ הלוח העברי) שלהם היום ?

דבר מורכב מאד (היארצייט משתנה ע"פ שנים חסרות מלאות וע"פ חודשים מלאים חסרים)

בניית אלגוריתם לחישוב

### 10.2 איסוף מידע וניתוחים סטטיסטיים (אנליטיקות):

המערכת תאסוף מידע סטטיסטי על:

* סך המשניות שלמדו לכל נפטר.
* סך הנפטרים במערכת.
* סך המשתמשים הרשומים.
* סך המנהלים הרשומים במערכת.

## 11. תיאור/התייחסות לנושאי אבטחת מידע

אזורים הדורשים הגנה:

* כל פניה לשרת למידע על משתמש - הפניה תטופל רק לאחר אימות על המשתמש.
* כל פניה לשרת לפעולת ניהול - הפניה תטופל רק לאחר אימות שהפונה הינו מנהל ע"י שם משתמש וסיסמא כראוי.
* סיפורים על נפטרים עלולים להכיל תוכן פוגעני או לא הולם - תינתן אפשרות למנהלים להסיר סיפור שלא הולם
* הגנה מפני איומים נפוצים:
* התקפת SQL Injection - קלט משתמש יותאם ויטופל לפני ביצוע שאילתות.(Parameterization).
* מניעת דליפת סיסמאות ומידע רגיש - הסיסמאות והמידע הרגיש לא ישמרו ישירות בקוד אלא במשתני סביבה (process.env) והתעלמות של גיט מהם (gitignore).
* הסיסמאות יעברו הצפנה לפני שמירתם בדאטה בייס.

## 12. משאבים הנדרשים לפרויקט:

### 12.1 מספר שעות המוקדשות לפרויקט:

כשלוש מאות שעות המתחלקות בין פיתוח צד שרת צד לקוח דאטה בייס בדיקות והגנה.

### 12.2 ציוד נדרש:

מחשבים בלימודים ובבית עם חיבור לרשת.

### 12.3 תוכנות נדרשות:

* visual studio code
* MySQL Workbench
* Node.js
* Postman
* Git
* GitHub Desktop

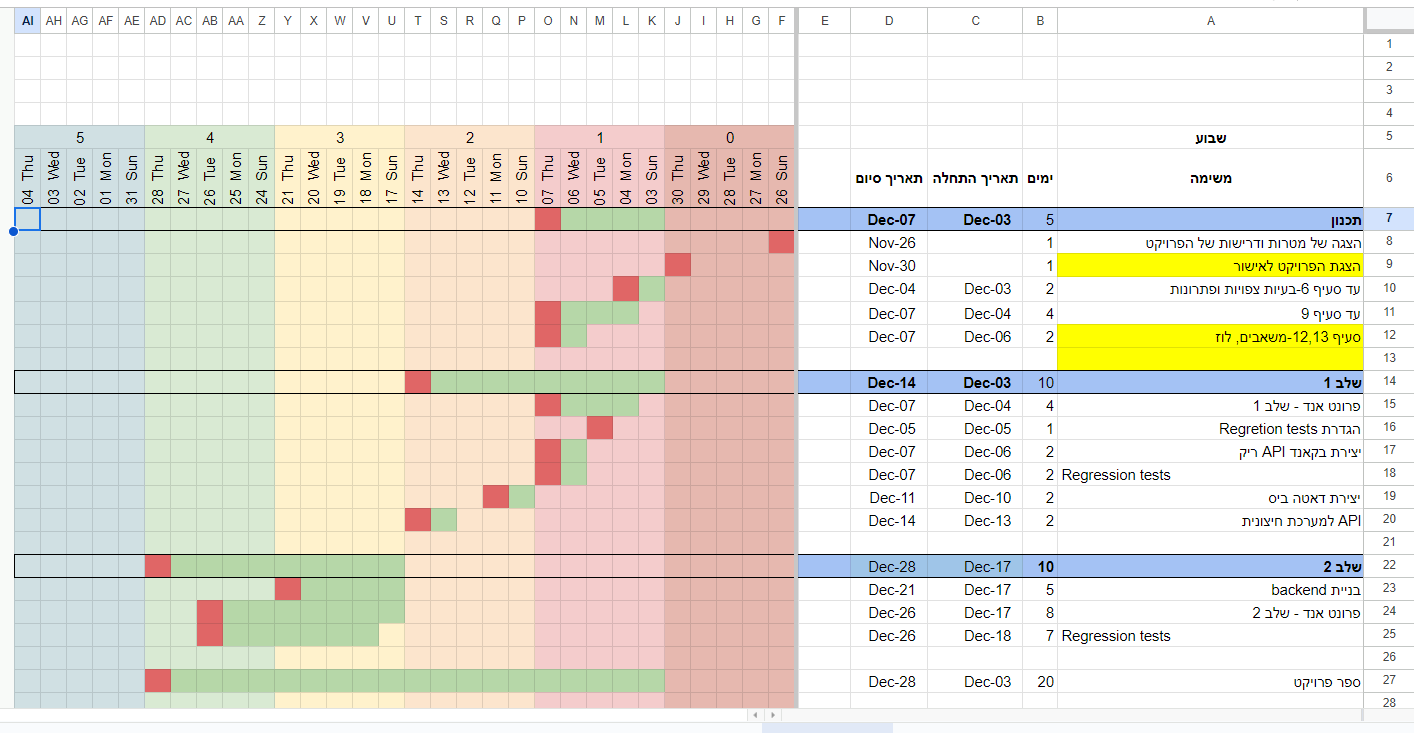
### 12.4 ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט:

* Material UI
* api sefaria library of Jewish texts
* npm dotenv
* npm @hebcal/core

### 12.5 ספרות ומקורות מידע:

* [**https://react.dev/**](https://react.dev/)
* [**https://www.npmjs.com/package/@hebcal/core?activeTab=readme**](https://www.npmjs.com/package/@hebcal/core?activeTab=readme)
* [**https://www.npmjs.com/package/@sefaria/search**](https://www.npmjs.com/package/@sefaria/search)
* [**https://mui.com/joy-ui/getting-started/**](https://mui.com/joy-ui/getting-started/)
* [**https://www.youtube.com/@WebDevSimplified**](https://www.youtube.com/@WebDevSimplified)
* [**https://www.w3schools.com/**](https://www.w3schools.com/)
* [**https://claude.ai/chats**](https://claude.ai/chats)

## 13. תכנית עבודה ושלבים למימוש הפרויקט:



## 14. תכנון הבדיקות שיבוצעו :

### 14.1 בדיקות תהליכיות:

| מספר | תיאור תהליך | תוצאה צפויה |
| --- | --- | --- |
| 1 | הרשמה למערכת | המשתמש נרשם בהצלחה |
| 2 | התחברות למערכת | המשתמש מתחבר בהצלחה |
| 3 | הוספת נפטר חדש | הנפטר נוסף בהצלחה למערכת |
| 4 | חיפוש נפטר | החיפוש מניב תוצאות רלוונטיות |
| 5 | צפייה בדף נפטר | דף הנפטר מוצג כראוי |
| 6 | הרשמה כמנהל | המנהל נרשם בהצלחה עם הרשאות מתאימות |

### 14.2 בדיקות יחידה:

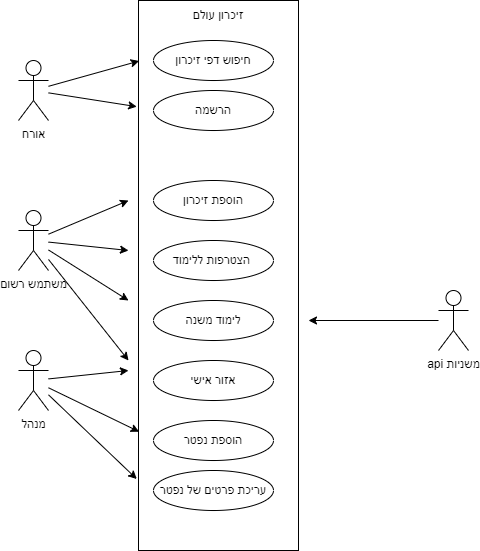
| מודול | תיאור בדיקה | תוצאה צפויה |
| --- | --- | --- |
| הרשמה | הזנת מייל לא תקין | הודעת שגיאה |
| התחברות | הזנת סיסמה לא נכונה | הודעת שגיאה |
| חיפוש | חיפוש עם קלט שגוי | ללא תוצאות |
| הוספת נפטר | שדות חובה לא הוזנו | הודעת שגיאה |
| צפייה בדף | גישה לדף לא קיים | הודעת שגיאה |

## 15. בקרת גרסאות (control version):

נעשה שימוש ב- git לבקרת גרסאות של הקוד.

# נספחים

### Use Case



### חיפוש נפטר - Sequence diagram

