**הצעה לפרויקט גמר** תאריך:19/10/2022

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **שם התלמיד** | **ת.ז.** | **כתובת** | **עיר** | **טלפון** | **תאריך סיום הלימודים** |
| דויטש אפרת | 213331572 | טללים 23 רמות א י | ירושלים | 0527159256 |  |
| כרמי מיכל | 213334675 | וולנשטיין 4 רמות ד' | ירושלים | 0583257199 |  |

1. **פרטי הסטודנטים:**

שם המכללה: סמינר מרכז בית יעקב

מגמה: הנדסת תוכנה

מסלול: הנדסאים

1. **פרטי המנחה האישי:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **שם המנחה** | **כתובת** | **טלפון נייד** | **תואר** | **מקום עבודה / תפקיד** |
| גב ורדית לוי |  |  |  |  |

1. **שם הפרויקט: אתר גמחים**
2. **רקע**
   1. **תיאור ורקע כללי**  
      הצורך היומיומי בשימוש בגמח"ים לסוגיהם הוא גדול ורחב מאד בכלל הציבורים. האתר שלנו מאפשר צפייה במגוון רחב של גמ"חים ומוצריהם לפי אזור מגורים וכן פלטפורמת ניהול הלוואות מתקדמת לבעלי גמ"חים.
   2. **מטרות המערכת**  
      1.לאפשר ללקוחות לאתר גמ"ח רצוי בקלות רבה. המערכת מחזירה פרטי גמח מיקומו ופרוט כללי על תכולת הגמ"ח.

2.לאפשר למנהלי גמח"ים לנהל את ההלוואות בגמ"ח שלהם בצורה מסודרת ומאורגנת ללא ניירת מיותרת.

1. **סקירת מצב קיים בשוק** נכון להיום אין בנמצא אתר כזה שמאפשר גישה לגמחים בכל הארץ ובכל הציבורים וכן ניהול מקצועי של ההלווואת בגמח"ים
2. **מה הפרויקט אמור לחדש או לשפר**הפרויקט יאפשר לכל אדם לחפש ולמצוא גמ"ח עפ"י דרישותיו וישפר את שיטת המעקב אחר הלואות לבעלי הגמחים שהיתה מקובלת עד כה.
3. **דרישות מערכת ופונקציונאליות**
   1. **דרישות מערכת**צד השרת   
      יועמד על שרת IIS המאפשר גלישה אנונימית.

MySql כולל מנגנוני התאוששות מקריסה.  
נפח דיסק פנוי של 30GB וזיכרון מומלץ 16GB.

צד הלקוח   
יוכל לפעול על מכשיר או מחשב אישי המריץ דפדפן google chrome (תיתכן תמיכה בדפדפנים נוספים), רוחב פס 100MBPS לפחות, ואפשרות קבלת התראות אודות מיילים חדשים על גבי המכשיר.

* 1. **דרישות פונקציונאליות**המערכת תאפשר לגולשים  
     הזדהות

חיפוש גמח עפ"י עיר מגורים וקטגוריה  
צפיה בכל הגמחים עפ"י קטגוריה מסוימת

צפיה בפרוט מוצרים כללי לגמח ספציפי

המערכת תאפשר למנהל  
צפיה ברשימת הגמחים שלו

הוספת גמח

נהוספת הלוואה לגמח קיים

עריכת הלוואה

דיווח על החזרה של הפריט

1. **בעיות צפויות במהלך הפיתוח ופתרונות**
   1. **תיאור הבעיות**
      1. מאחר שהאפליקציה נפתחת בדפדפן אינטרנט, כיצד נציג למשתמש התראות על עסקאות חדשות כאשר האפליקציה אינה פתוחה?
      2. כיצד נתמודד עם דיווחים כוזבים על עסקאות?
   2. **פתרונות אפשריים**
      1. נשלח למשתמש הודעות בדואר אלקטרוני (או באמצעי תקשורת מקובל אחר) והמכשיר שלו יציג התראה אוטומטית על המייל שהגיע.
      2. נאפשר פניות למנהל בדבר דיווחים כוזבים, והוא ימחק את הדיווח הכוזב ויגדיר למערכת שלא תציג להבא דיווחים מאת השולח המכזב.
2. **פתרון טכנולוגי נבחר**:
   1. **טופולוגית הפתרון**  
      המערכת תיושם ברשת האינטרנט, ותהיה מורכבת משרת המנהל מסד נתונים יחסי של SQL SERVER, וממשק משתמש בצד הלקוח. התקשורת ביניהם תתבצע באמצעות WEB API.
   2. **טכנולוגיות בשימוש**  
      c#   
      השרת יממש Web API כך שקליינטים יוכלו לפנות אליו ולקבל מידע. התשתית שנשתמש בה לצורך כך היא c#  
      Node.js היא סביבת ריצה לשפת JavaScript שנכתבה ב-C++ ומבוססת על מנוע V8. היא תוכננה לכתיבת אפליקציות רשת כגון שרתי HTTP.  
      Node.js מאפשרת לקוד JavaScript לרוץ ללא צורך בדפדפן, מה שמאפשר לה לשמש כשפת צד שרת.  
      React   
      React היא ספריית קוד פתוח בשפת JavaScript המשמשת לפיתוח ממשקי משתמש דינמיים, המותאמים לכל סוג מסך ולכל המאפיינים ושינויי המצב של המסך.
   3. **שפות הפיתוח**  
      השרת ייכתב בשפת JavaScript אשר בבסיסה היא שפה שיעודה לצד לקוח , אולם תשתית Node.js מאפשרת להשתמש בה וביתרונותיה גם בצד השרת.  
      חלק נוסף של צד השרת, האחראי על הבינה המלאכותית בפרויקט, ייכתב בשפת Python שהיא נוחה ושימושית מאד לצרכי חישוב ועיבוד.  
      צד הלקוח ייכתב גם הוא בשפת JavaScript (שעליה מבוססת React).  
      בחרנו בשפות אלו שהן פופולריות, שימושיות, ונוחות לפיתוח.
   4. **תיאור הארכיטקטורה הנבחרת**שרת SQL Server יאחסן את הנתונים.

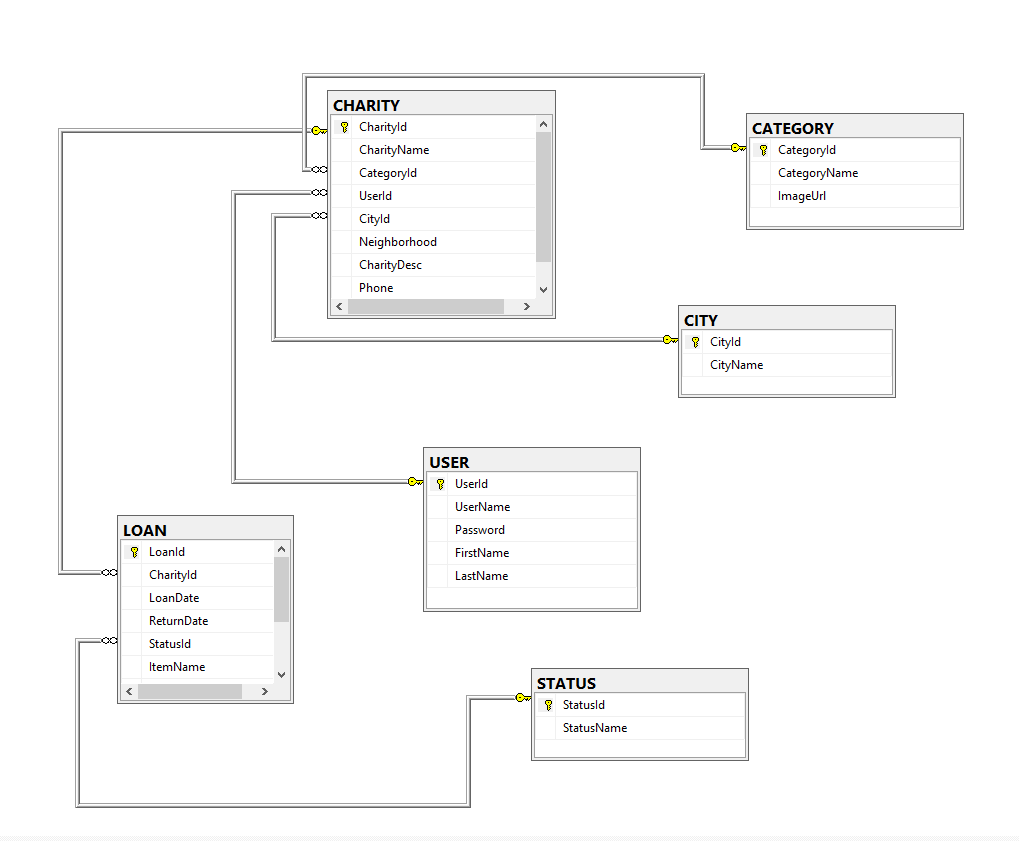
שכבת ניהול הנתונים – Data Layer - תהיה אחראית לשליפת הנתונים הנדרשים ולעדכונם.

השכבה הלוגית – Business Layer – תנהל את הלוגיקה הנדרשת.

תשתית Node.js תאפשר תקשורת בין השרת ללקוח.

ממשק צד הלקוח ב React יפנה לשרת Node.js לצורך קבלת המידע הנדרש או לצורך עדכונו ויציג את דפי האפליקציה למשתמש.

* 1. **חלוקה לתוכניות ומודולים**חלק א': מכלול תוכנה שרתי, שינהל את מרכיבי הניהול העסקי של המערכת.  
     חלק ב': מכלול הלקוח, שיחשוף ממשק משתמש מעל הweb
  2. **סביבת השרת**שרת IIS יארח את הAPI של צד השרת.
  3. **ממשק המשתמש –GUI**בטכנולוגיית React ניצור דפי אינטרנט שיוצגו על גבי דפדפן chrome (תיתכן תמיכה בדפדפנים נוספים).
  4. **ממשקים למערכות אחרות**לצורך זיהוי המיקום של המשתמש, על מנת להציע לו עסקאות במיקומים קרובים, נשתמש ב Distance Matrix API של Google(אופציונלי).
  5. **שימוש בחבילות תוכנה**במהלך הפיתוח נשתמש בNPM ובPIP להתקנת חבילות תוכנה נוספות באם תידרשנה.

1. **שימוש במבני נתונים וארגון קבצים:**
   1. הישויות העיקריות במערכת: גמ"חים, לקוחות, הלוואת, ערים, קטגוריות.  
      הנתונים יאוחסנו במסד נתונים יחסי של SqlServer
2. **תרשימי מערכת מרכזיים**
   1. **Use cases**כאן יש להציג תרשים
   2. **Sequence Diagram**כאן יש להציג תרשים
   3. **Data Flow Diagram**כאן יש להציג תרשים
3. **תיאור המרכיב האלגוריתמי – חישובי**
   1. הבעיה העיקרית שהפרויקט יפתור היא ריכוז הצעות מכירה משתלמות שיתקבלו מקהל המשתמשים, ושליחת התראות מתאימות ללקוחות על פי העדפותיהם האישיות.  
      על מנת לסנן עבור הלקוח את העסקאות המתאימות לו, נבדוק כל עסקה האם התעניין בה, ולאחר מכן נציג רק עסקאות המצויות בטווח מיקום סביר עבור הלקוח (אופציונלי).
   2. נתוני העסקאות ונושאי העניין של הלקוחות יישמרו בDB באופן שניתן יהיה להפיק מידע סטטיסטי לגביהם, אך הפרויקט אינו כולל הפקת מידע זה. (לגבי פרטיות הלקוח, יוצע לו להתיר או לאסור את שימוש המערכת בנתונים שלו לצורכי סטטיסטיקה.)
4. **אבטחת מידע**  
   בנושא אבטחת המידע נסתמך על המנגנונים האוטומטיים של שרת הIIS המארח. אין התייחסות מיוחדת נוספת לנושא אבטחת המידע מכיוון שהאפליקציה אינה עוסקת בתכנים רגישים.

לצורך אבטחת הנתונים הרגישים במערכת נבנה מערכת הרשאות, שתדאג לכך שהמידע הרגיש לא יוצג בפני משתמשים בלתי מורשים. כמו כן נסתמך על תשתית האבטחה של שרת IIS מאובטח.

1. **משאבים הנדרשים לפרויקט**
   1. היקף שעות העבודה כ700 שעות, שיחולקו בין המגישות בצורה מאוזנת.
   2. ציוד נדרש: מחשב PC עם זיכרון RAM של 8GB, תשתית רשת המאפשרת שליחת הודעות דואר אלקטרוני.
   3. תוכנות נדרשות:  
      PyCharm

Visual Studio Code  
MySql

NodeJS

* 1. ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט:  
     העמקת הידע וההכרות עם הטכנולוגיות שבשימוש הפרויקט, וכן שימוש ב Distance Matrix API של Google (אופציונלי).
  2. ספרות ומקורות מידע  
     https://reactjs.org/  
     https://nodejs.org/en/

https://stackoverflow.com

1. **תכנית עבודה ושלבים לביצוע הפרויקט**התכנון והפיתוח החל מחודש אלול תשפ"ב:
   1. תכנון פונקציונאליות המערכת (כולל ציורי מסכים) – כשלושה שבועות.
   2. יצירת תרשימי מערכת- כשבועיים.
   3. תכנון בסיס הנתונים- כשבועיים.
   4. כתיבת הצעת הפרויקט- כשבוע.
   5. פיתוח צד השרת- כעשרה שבועות.
   6. פיתוח צד הלקוח- כעשרה שבועות.
   7. כתיבת ספר הפרויקט- לאורך כל תהליך הפיתוח.
2. **תכנון הבדיקות שיבוצעו**
   1. בדיקות תהליכיות ברמת משתמש – Full Flow

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| מספר בדיקה | מהות הבדיקה | אופן ביצוע הבדיקה | תוצאה רצויה | תוצאה בפועל | עבר או נכשל |
| 1 | הזדהות | הזנת שם וסיסמה, והקשה על אישור | כניסה לדף הראשי של האפליקציה |  |  |
| 2 | הגדרת העדפות אישיות | הזנת העדפות לקוח והקשה על אישור | קבלת הודעה שהמידע נשמר |  |  |
| 3 | דיווח על דיל | הזנת פרטי הדיל ואישור | קבלת הודעה שהדיווח נקלט |  |  |
| 4 | דיווח על דיל תוך השמטת פרטים הכרחיים | הזנת פרטי דיל ללא שדה מחיר נוכחי ואישור | קבלת הודעה שחסר שדה מחיר |  |  |
| 5 | חיפוש אקטיבי של דילים | הזנת שדות לסינון  והקשת "הצג דילים" | קבלת רשימת דילים המתאימים לשדות הסינון |  |  |
| 6 | קבלת התראות על דילים רלוונטיים | דיווח מכוון על דיל העונה להעדפות לקוח ספציפי | הלקוח הספציפי יקבל הודעת מייל אודות הדיל |  |  |

* 1. בדיקות יחידה –Unit Test

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| מספר בדיקה | מהות הבדיקה | אופן ביצוע הבדיקה | תוצאה רצויה | תוצאה בפועל | עבר או נכשל |
| 1 | שליפת רשימת דילים לפי תנאי | קריאה לWEB-API לפונקציה GetDeals עם פרמטר סינון לפי שם מוצר | רשימת דילים הכוללים את שם המוצר שצוין |  |  |
| 2 | עדכון דיל חדש | שליחה לפונקציה AddDeal עם פרטי הדיל | הדיל החדש יעודכן בDB  התראות במייל ישלחו ללקוחות המעוניינים |  |  |
| 3 | בדיקה האם הדיל מתאים לאזור גאוגרפי מסוים | שליחת מספר דיל ומיקום גיאוגרפי לפונקציה המתאימה בWeb-API | True באם הדיל מתאים לאזור  False באם אינו מתאים |  |  |
| 4 | עדכון העדפות אישיות | קריאה לפונקציה UpdatePreferences עם הנתונים העדכניים | פרטי ההעדפות יעודכנו בDB |  |  |

1. **בקרת גרסאות**לצורך בקרת הגרסאות נשתמש במנגנון הפופולארי Git ובשירותי האחסון של GitHub.

###### \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

###### חתימת הסטודנט חתימת המנחה האישי

1. הערות ראש המגמה במכללה

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **אישור ראש המגמה**

## שם: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימה: \_\_\_\_\_\_\_\_\_\_\_\_ תאריך: \_\_\_\_\_\_\_\_\_\_\_\_\_

1. **הערות הגורם המקצועי מטעם מה"ט**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **אישור הגורם המקצועי מטעם מה"ט**

## שם: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימה: \_\_\_\_\_\_\_\_\_\_\_\_ תאריך: \_\_\_\_\_\_\_\_\_\_\_\_\_