

תאריך: _____

לכבוד

יחידת

הפרויקטים מה"ט

הצעה לפרויקט גמר

* יש להדפיס את כל הנתונים הנדרשים

א. פרטי הסטודנטים

שם הסטודנט	ת.ז. 9 ספרות	כתובת	טלפון נייד	תאריך סיום הלימודים

שם המכללה _____ סמל המכללה: _____

מסלול ההכשרה: הנדסאים/טכנאים מוסמכים (מחק את המיותר).

מגמת לימוד: _____ מקום ביצוע

הפרויקט: _____

ב. פרטי המנחה האישי

שם המנחה *	כתובת	טלפון נייד	תואר	מקום עבודה/תפקיד

* עבור מנחה אישי חדש יש לצרף קורות חיים, ניסיון מקצועי ותעודות השכלה לאישור מה"ט.

חתימת הסטודנט _____
חתימת המנחה האישי _____
חתימת הגורם המקצועי מטעם מה"ט _____

1. שם הפרויקט

TechMaster

2. רקע

2.1. תיאור ורקע כללי

האתר TechMaster מהווה הכנה מושלמת למחפשי עבודה בתחום ההיטק, האתר נותן תמונת מצב על שליטת המועמד בשפות שונות ועל הרמה הטכנולוגית שלו וזאת באמצעות מבחנים מותאמי רמת קושי ותחומי התנסות, האתר מרחיב את ידיעות המועמד ע"י כתבות ומשלים את המושגים החסרים לו בתחום בו הוא מתמקצע על מנת לגשת לראיון עבודה עם מוכנות גבוהה ביותר

2.2. מטרות המערכת

- 1) להכין את המועמד לקראת ראיון עבודה בטכנולוגיה/ שפה בה הוא מחפש
- 2) לאפשר למועמד לבחון את רמת הידע שלו ביחס לרמת הידע בשוק
- 3) לכוון את המועמד לחיפוש עבודה בשפה שהוא הכי שולט

3. סקירת מצב קיים בשוק, אילו בעיות קימות

אתרים המציגים שחזורים מראיונות עבודה יש למכביר (כדוגמת www.jobhunt.co.il www.theworker.co.il) אך הם לא באים לתת מענה להכנה בתחומים ספציפיים אלא לשתף על תהליכי ראיון בחברות ידועות, כמו כן השאלות מופיעות ללא פתרונות וכמובן גם לא מחולקים לפי רמת ניסיון

4. מה הפרויקט אמור לחדש או לשפר

בפרויקט זה מעבר לשאלות המותאמות אישית לתחומי ההתמקצעות ורמת הידע – יש חידוש משמעותי בהשלמת חומר ופערים קטנים של המשתמש בתחום ההתמקצעות, האתר יציג בסיום המבדק את התשובות הנכונות בתוספת רקע והסבר על מהות השאלה, בנוסף, כדי להכין את המועמד באמת, לא יוצגו השאלות כטקסט אלא כראיון דמה עם זמן מוקצב לכל שאלה כאשר בסיומו יקבל במשתמש ציון מדויק של רמת הידע שלו

5. דרישות מערכת ופונקציונאליות

5.1. דרישות מערכת

דרישות מערכת, סביבת הטמעה ושימוש.
המערכת תעבור קומפילציה והפצה בסביבת visual studio עם התקנת .Net.
Framework גירסה 4 ומעלה והיא אמורה לרוץ בסביבת שרת אשר מריץ IIS Express לקבלת בקשות לתצוגת דפי אינטרנט.
המשתמש יוכל להריץ את האתר בכל מכשיר אשר מותקן עליו דפדפן אינטרנט.

5.2. שרידות ביצועים התמודדות עם עומסים:

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

גם עומס על שרת ה-SQL 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. משתמש יוכל לבחור את השפות בהם הוא מעוניין להתמקצע.
- 5.3.2. המערכת תשלף רנדומלית שאלות ברמת הקושי הרצויה.
- 5.3.3. המערכת תבחן את המשתמש בשיעור זמן מוקצב ותציג לו את הציון.
- 5.3.4. המערכת תציג למשתמש את תשובות המבחן ולצידן קטעי לימוד או קישור לכתבה על מהות השאלה.
- 5.3.5. המערכת תשמור ציוני משתמשים לפי תחומי מבחן ותציג למשתמש באמצעות גרפים את רמת הידע שלו בהשוואה לציונים בשוק.
- 5.3.6. המערכת תודיע למשתמש באיזה תחום הוא צבר ניקוד גבוה יותר ותמליץ לו לעבוד בתחום זה, התפלגות ציוניו בתחומים בהם נבדק יוצגו בגרף עוגה.
- 5.3.7. המערכת תפנה את המשתמש למבחן נוסף באותה רמת קושי או למבחן ברמת קושי אחרת בהתאם להצלחתו במבחן נוכחי.
- 5.3.8. המשתמש יוכל לגשת מכל מסך לקריאת חומר לימודי.
- 5.3.9. שאלות, תרגילים, שפות וחומר לימודי ישמרו במאגר דינמי ויהיו זמינים תמיד לעדכון והוספה.
- 5.3.10. המערכת תבדוק את הרשאת מנהל האתר לפני ביצוע שינויים.

6. בעיות צפויות במהלך הפיתוח ופתרונות תפעוליות, טכנולוגיות, עומס ועוד:

- 6.1. תיאור הבעיות- הללו כפועל יוצא של דרישות המשתמש מהתוכנה.
בעיה 1: היכן נשמור את מאגר השאלות למבחנים וכיצד נבטיח את רלוונטיות האתר גם כשיקומו טכנולוגיות חדשות?
בעיה 2: כמה מבחנים נעשה כשמשתמש ירצה לבדוק התאמה ל- 2 או יותר שפות תכנות?
- 6.2. פתרונות אפשריים.

לבעיה 1: פתרון 1: נשמור את השאלות במסד נתונים הניתן לעדכון ונשלף אותם עבור כל מבחן בנפרד

פתרון 2: נשמור את השאלות במסד נתונים סטטי ברמת הקליינט וכך נאפשר מעבר מהיר ממבחן למבחן.

פתרון 3: נשמור את השאלות במסד נתונים הניתן לעדכון ועם בחירת התחום

נטען את כל השאלות הרלוונטיות ברמות השונות

לבעיה 2: פתרון 1: מבחן של 10 שאלות לכל נושא

פתרון 2: מבחן של 10 שאלות לאחד מהנושאים בצורה רנדומלית

פתרון 3: חלוקת 10 שאלות באחוזים שווים לפי מספר הנושאים

הפתרון הנבחר עבור כל אחת מהבעיות:

לבעיה 1: נשתמש בפתרון 3 וכך נאפשר למשתמש לעבור בקלות משלב לשלב ולבצע

מבחנים נוספים מבלי לוותר על עדכניות השאלות באתר

לבעיה 2: נשתמש בפתרון 3 כדי לתת למשתמש מבחן המקיף את כל תחומי ההתמקצעות מלי להעמיס עליו מבחנים רבים

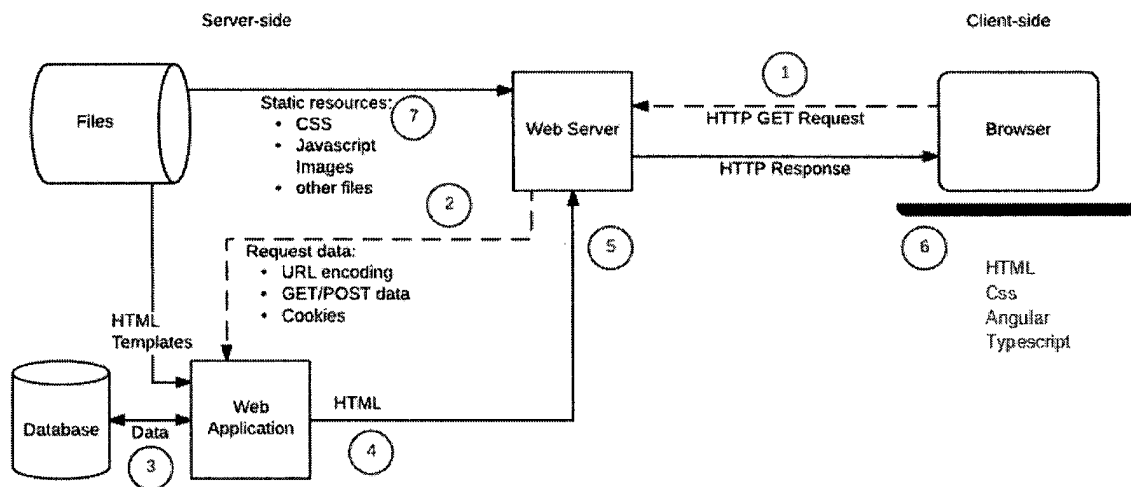
7. פתרון טכנולוגי נבחר:

7.1 טופולוגית הפתרון

המערכת מורכבת משרת IIS המריץ את האתר בסביבת ה-server,

מסד נתונים - של sql-server

ממשק משתמש בצד הלקוח: דפדפן אינטרנט כלשהו: internet explorer/ chrome על מחשב מקומי או על מכשיר ios/ android



7.2 טכנולוגיות בשימוש.

אנגולר - תשתית תוכנה ליישומי רשת בקוד פתוח המתוחזקת ע"י גוגל המשמשת בעיקר לפיתוח של אתרים דינמיים ומותאמת במיוחד לשימוש מקוון

Entity Framework - טכנולוגית גישה למסד נתונים בסביבת .NET. המאפשרת גישה נוחה ל DB

MVC - הפרדת תפקידים בישום כדי להפחית את התלות בין ממשק המשתמש לשאר חלקי התוכנה.

7.3 שפות הפיתוח:

שפות הפיתוח בצד השרת:

.net #C - שפת פיתוח קוד שרת חזקה במשק בגלל העקרונות הפשוטים

המאפשרים לנו לנהל מערכת חכמה והגיונית ומצד שני עשירה ביכולות מתקדמות

#C הוא תנאי הכרחי לשימוש במגוון טכנולוגיות, מתוכן **SignalR**.

SignalR - פלטפורמת תוכנה בצד השרת בפיתוח מיקרוסופט המתוכננת לכתיבת

אפליקציות אינטרנט סקילביליות, במקרה שלנו שירות אתר אינטרנט.

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.

Angular6 - שפת פיתוח צד לקוח המוסיפה אבסטרקציות מעולם פיתוח Object Oriented, הכתיבה מתבצעת ע"י כתיבת HTML שבתוכו משולבים תוספות של אנגולר, 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 עיצוב ועוד. זוהי שפת הקוד הבסיסית לסביבת ה-web עבור בניית אתרים. HTML5 כוללת כלים חדשים שבאו להקל על עבודת המפתחים.

Bootstrap היא סביבת עבודה בקוד פתוח לצד לקוח, שמכילה אוסף של כלים ליצירת אפליקציות Web ואתרים.

7.4. תיאור הארכיטקטורה הנבחרת- הסבר בכמה מילים מדוע.

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

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

7.5. 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 (**Data Access Layer - DAL**) - שכבה זו מורכבת ממקור נתונים – מסד הנתונים שלנו, וממערכת תוכנה entity framework אשר תפקידה לקרוא את המידע הנדרש למערכת, לשמור את העדכונים, ולהוסיף מידע חדש או למחוק פרטי מידע קיימים.

7.6. סביבת השרת

לצורך הפרוייקט נשתמש בשרת מקומי – IISExpress, המסופק עם סביבת העבודה של visual studio. אם האתר ירכש על ידי לקוח, נעלה אותו לשרת ארוח כלשהו או – Microsoft azure

7.7. ממשק המשתמש/לקוח – GUI.

שכבת ה GUI מורכבת מדפי HTML שמוצגים למשתמש דרך הדפדפן.

8. שימוש במבני נתונים וארגון קבצים

8.1. מבני הנתונים.

שפות: קוד, שם שפה, שם תת שפה, מבוא
שאלות: קוד שאלה, טקסט שאלה, שם תת שפה, רמת קושי, קוד תשובה נכונה, חומר לימודי
תשובות: קוד תשובה, קוד שאלה, טקסט
ציונים: קוד ציון, קוד שפה, ציון
מנהלים: קוד מנהל, שם, סמא, מייל

8.2. שיטת האיחסון

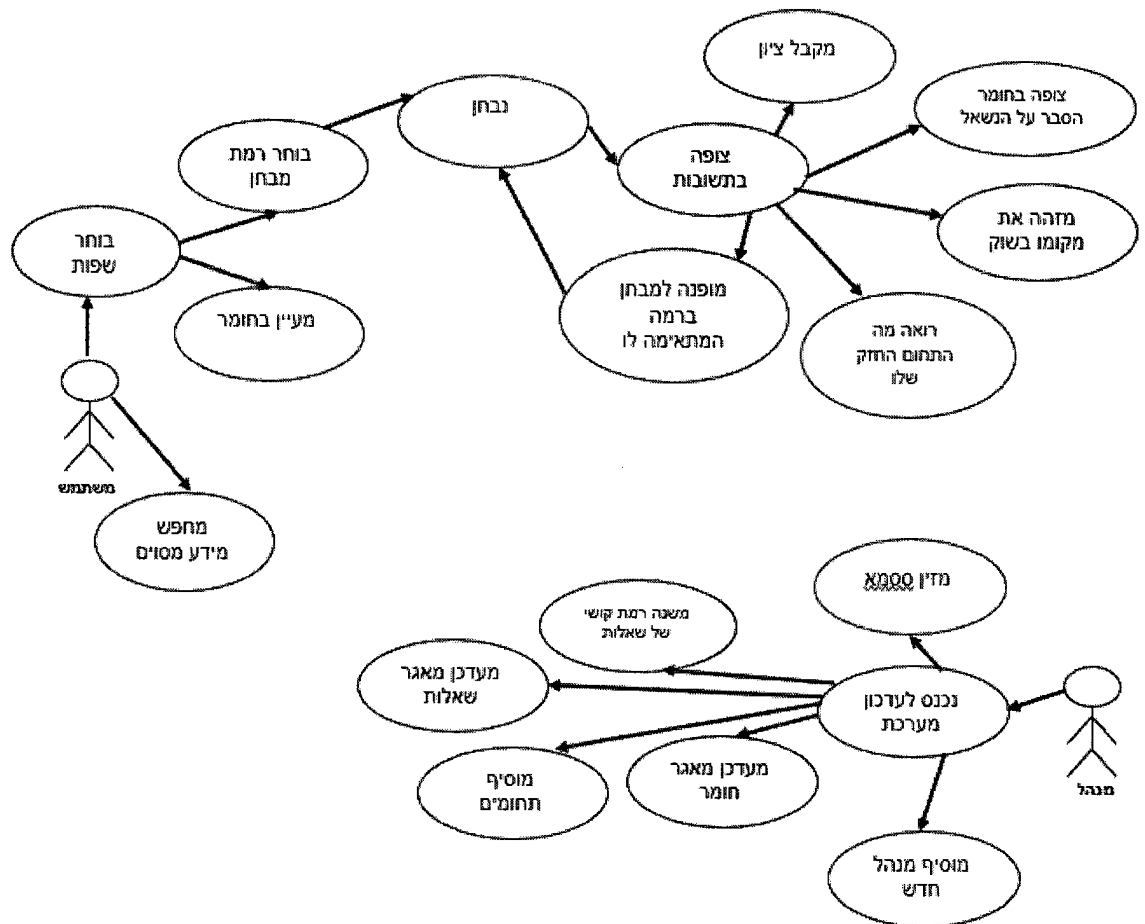
שיטת האחסון היא ע"י Sql Server-מסד נתונים. הגישה למסד הנתונים היא ע"י Entities.

8.3. נפילה לקריסה/תמיכה

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

9. תרשימי מערכת מרכזיים

Use Case



10. תיאור המרכיב האלגוריתמי – חישובי

10.1. איזה בעיה בא לפתור, איך יפתור?

הבעיה: חישוב ציון המשתמש על מבחניו

הפתרון: חישוב הנקודות במבחן יתבצע עבור כל תחום בנפרד בחלוקה לאחוזים לפי רמת השאלות

הבעיה: צריך להציע למשתמש מאגר שאלות מתחלף בשביל להציע עניין גם בכניסות הבאות לאתר

הפתרון: שלפת שאלות רנדומלית עם שמירת קוד שאלה בסשן ובדיקה אם זו שאלה חדשה

10.2. איסוף מידע וניתוחים סטטיסטיים (אנליטיקות)

- המערכת תציג למשתמש באמצעות היטוגרמה את מיקום הציון לעומת ציונים שהתקבלו עבור אותו תחום- איסוף המידע יתבצע בסיום כל מבחן וישמר בטבלת ציונים עבור כל תחום.
- המערכת תציג למשתמש באמצעות גרף עוגה ניתוח סטטי של רמת הידע שלו בתחומים שונים כדי שילמד באיזה תחום מומלץ לו לעבוד.
- המערכת תציג למשתמש באמצעות bar-chart את רמת הביקוש במשק לכל תחום- וזאת ע"י מנית המבחנים עבור כל תחום בטבלת ציונים.

11. תיאור/התייחסות לנושאי אבטחת מידע

יש לדאוג לאבטחת מאגרי מידע משינוי/ איבוד חומר הנגרם על ידי גורמים זדוניים, ולהגן על שרת ה- sql server מפני- sql injection.

נא ציין מס' מקרים ותגובות להם ניתן מענה אבטחתי.

- במקרה שמשתמש מנסה לגשת למאגרי המידע, המערכת תבקש שם משתמש וססמא על מנת לבדוק הרשאת מנהל, במידה והשם או הסיסמא אינם תואמים מידע קיים ברשימת מנהלי האתר, המערכת תציג לו הודעת שגיאה ולא תאפשר כניסה.
- במידה וירצו בעתיד להוסיף מנהל חדש או לאפשר גישה לאנשים נוספים, קוד ישלח לכתובת מייל של מנהל קיים ולא תותר הוספת מנהל בלעדי הקוד.
- הסיסמאות תהינה מוסתרות.

12. משאבים הנדרשים לפרויקט:

12.1. מספר שעות המוקדש לפרויקט, חלוקת עבודה בין חברי הצוות

מספר השעות המוקדשות לפרויקט הוא 700. חלוקה בין חברי הצוות הוא 350 שעות לכל אחת.

12.2. ציוד נדרש

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

12.3. תוכנות נדרשות

Visual Studio , SQL Server, iis express, דפדפנים

12.4. ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט

גרסאות 4, 5, 6 של אנגולר

אנימציות וגרפים באנגולר

12.5. ספרות ומקורות מידע

<https://angular.io/api/>

<https://angular.io/tutorial/toh-pt0>

<http://learning-programing.com/basic/architecture>

13. תכנית עבודה ושילבים למימוש הפרויקט

- (1 ייזום רעיון, ניתוח מערכת, הכרת השטח, אפיון מסד נתונים
- (2 תכנון UX-UI , עיצוב גריד, זרימת מסכים
- (3 תוכן האתר- איסוף ובדיקת חומר מקצועי, הכרת השוק, חקירה ומיון
- (4 כתיבת לוגיקה עסקית

(5) כתיבת ממשק משתמש ועיצוב

(6) בדיקות QA

(7) בדיקות קונספטואליות

(8) התקנה והטמעה

14. תכנון הבדיקות שיבוצעו

Full flow

מספר בדיקה	מס' דרישה במסמך אפיון	מקרי הבדיקה	ידנית אוטומטית	חשיבות	הערות
1	5.3.1 5.3.2	בדיקת התאמת השאלות לבחירת המשתמש בתחום, תת תחום ורמת קושי, כולל בדיקת בחירת כמה שדות וכמה תת תחומים	ידנית	גבוהה	
2	5.3.3	בדיקת ציון המבחן	ידנית	גבוהה	
3	5.3.5 5.3.6	בדיקת נכונות הגרפים	ידנית	גבוהה	
4	5.3.3	בדיקת הטיימר ותחילתו כל פעם מ-0	ידנית	בינונית	
5	5.3.10	בדיקת הרשאה לעדכון שאלות ושפות	ידנית	גבוהה	
6	5.3.8	בדיקת מעבר בין סקשנים בזמן מבחן	ידנית	בינונית	
7	-	בדיקת התאמת מראה האתר לרזולוציית דסקטופ ואפליקציה	ידנית	בינונית	
8	5.3.10	בדיקת הוספת מנהל ע"י שליחת סמא למייל	ידנית	נמוכה	

15. בקרת גרסאות

נשמור צורה ב GIT בעזרת מנגנון ה source-control הבא עם visual- code לאורך
התהליך, כך נוכל:

- לשלוט בפיתוחים ניסיוניים שונים, העלאת פיצ'רים שונים או סילוקם במקרים שישבשו את זרימת
המערכת,
- לעבוד על מחשבים מתחלפים
- לעבוד במקביל ולהכניס שינויים מ-2 מחשבים בו זמנית
- למנוע אובדן קוד

מתי נשמור את הגרסאות בשרת?

170

בסיום כתיבת קטע קוד ובדיקתו מול מדדים שונים (יותר ממדד אחד)
במקרה ונזדקק זו לקוד של זו כאשר הוא עדיין בסטטוס פיתוח נעלה לפיצ'ר זמני
נקפיד שבשמירת הגרסאות נתאר באופן כללי במה הוא שונה מהגרסה הקודמת. יתר על כן, נקפיד מאוד
שלקבצי הקוד יהיו שמות משמעותיים, כדי שהקוד יהיה ברור וקריא.

חתימת המנחה האישי
חתימת הסטודנט

ג. הערות ראש המגמה במכללה

ד. אישור ראש המגמה

שם: _____ חתימה: _____
תאריך: _____

ה. הערות הגורם המקצועי מטעם מה"ט

ו. אישור הגורם המקצועי מטעם מה"ט

שם: _____ חתימה: _____ תאריך: _____