

## דרישות לפרויקט גמר בקורס פיתוח אפליקציות אינטרנטיות

**כללי:** עליכם לממש מערכת אינטרנטית תוך שילוב הטכנולוגיות המופיעות בסעיף הדרישות. המערכת יכולה להיות חנות אינטרנטית, רשת חברתית, אך תוכלו להציע רעיון אחר כל עוד הוא תומך במסמך הדרישות.

### דרישות מנהלתיות:

1. ניתן לעבוד בקבוצות של 3 עד 4 סטודנטים בלבד. עם זאת, כל אחד מחברי הקבוצה **חייב להכיר את כל הקוד** שנכתב ועליו להיות מסוגל לענות על שאלות מכל חלקי הקוד במהלך ההגנה. כמו כן, על הסטודנטים להשתמש, בזמן פיתוח הפרויקט, במערכת לניהול גרסאות ([Git](#)) ובמועד ההגנה להציג את **ההתנהלות (וההשקעה)** של **כל אחד** מחברי הקבוצה.
2. **לא ניתן** לשלב בקבוצה סטודנטים אשר רשומים לקבוצות שונות של אותו המרצה.
3. מועדי ההגנות מפורטים בלוח הבחינות באתר המכללה.
4. כל סטודנט רשאי להבחן בהגנה אחת בלבד. בפרט, לא ניתן לגשת שוב לאחר שניגשתם כבר.
5. אישור לגשת למועד המיוחד יאושר אך ורק ע"י הממונה לענייני סטודנטים, בהצגת סיבת היעדרות משני המועדים ובצירוף אישורים וסימוכין רלוונטיים.
6. הפרויקט יוגש ע"י כל הסטודנטים עד 24 שעות לפני מועד ההגנה הראשונה. תיבת ההגשה תיסגר בשעה 23:59.
7. **גם סטודנטים שבחרו להיבחן במועדי ההגנה האחרים - מגישים את הפרויקט לפני המועד של ההגנה הראשונה.**
8. סטודנט שלא הגיש במועד את הפרויקט – יורדו לו 20 נקודות מהציון הסופי בקורס.
9. יש להגיש את הקוד של פרויקט הקורס (לבצע Zip לספריית הפרויקט) בתיבה שנפתחה לשם כך במודל וכן קישור לגיט של הפרויקט. יש לוודא שהגיט פתוח (מבחינת גישת צפיה) לכל מי שיש לו גישה לקישור.
10. במעמד ההגנה, יש להפעיל את המערכת בגירסת הקוד שהועלתה למודל בלבד. אין לבצע שינויים לגירסה זו.
11. ההגנה תתחלק ל 2 חלקים: בחלק הראשון תידרשו להדגים את המערכת. בחלק השני ישאל המרצה שאלות (לקבוצה ככלל ולסטודנטים ספציפיים בפרט) כשקוד המערכת פתוח בפניו. מעמד ההגנה מהווה רק חלק מהציון הקשור למראה הכללי של המערכת, נפילות/באגים, שליטת הסטודנטים וחלוקת העבודה בקבוצה. חלק נוסף במתן הציון מבצע המרצה לאחר המפגש עם הקבוצה, כאשר הוא בודק את קוד המערכת באופן עצמאי על מנת לתת ציון סופי – על כן, לא ניתן לבשר על ציון סופי במהלך מפגש ההגנה.
12. עליכם לוודא שניתן להדגים את המערכת במעבדה שנקבעה טרם מועד ההגנה. מנסיונו – עשויות להיות בעיות הרשאה/התקנת DB וכד', על כן וודאו זאת מספר ימים לפני ההגנה. ניתן להגיע להגנה עם לפטופ אישי, אך תקינות המחשב הפרטי והתוכנות בו – באחריות הסטודנט – על כן וודאו כי המערכת עובדת גם במחשב המעבדה למקרה של תקלה.
13. חובה להשתמש בטכנולוגיות, שפות וארכיטקטורות שנלמדות בקורס (ובהם בלבד) ובאופן שבו הן נלמדות – ולא ניתן להשתמש בחלופות אחרות במקומן. בפרט, צד השרת חייב להיות בטכנולוגיית asp.net MVC גירסה 5 ומעלה (**עדיף להשתמש ב .net core**). בשפת #C ובארכיטקטורת MVC וחובה לעבוד בשיטת code-first בלבד (כמו שראינו בשיעורים) ולא ניתן ליצור את מסד הנתונים בדרך אחרת (לא באמצעות SQL או כלים למיניהם).
14. דרישות המינימום עבור הפרויקט מפורטות בהמשך. עבור ציון גבוה יש להגיש פרויקט מלא המציג מערכת שלמה ומתפקדת (מעשית ולוגית), **אשר מכיל מעבר לדרישות המינימום.**

### דרישות טכניות:

15. המערכת תתוכנן ותמומש באמצעות MVC גירסה אשר מותקנת במעבדת המחשבים במכללה. תהיה הפרדה ברורה בין ה View, ה Controller וה Model.
16. על המערכת לתמוך בלפחות 3 מודלים שונים. (למשל בחנות: מודל אחד על המוצרים שנמכרים בחנות, מודל נוסף על הלקוחות ומודל על הספקים).

- חובה לממש את שלושת סוגי הקשרים (one-to-one, one-to-many, many-to-many)
17. על כל אחד מהמודלים, המערכת צריכה לתמוך ב Create (יצירת אובייקט חדש), Update (עדכון תוכן האובייקט – למשל עדכון פרטי המוצר הנמכר בחנות), Delete (מחיקת אובייקט – למשל מחיקת ספק), List (רשימת אובייקטים: למשל רשימת הלקוחות), Search (חיפוש אובייקט על פי קטגוריות שונות – למשל מצא את הלקוחות הגרים באיזור המרכז).
  18. הגישות לכלל נתוני המערכת ב DB יהיו באמצעות Entity Framework בלבד (אין לכתוב בקוד שאלות ב SQL אלא רק באמצעות פקודות Entity Framework ישירות או עם LINQ). **שימו לב**, כלל העבודה מול ה DB מתבצעת באמצעות Entity Framework ו/או LINQ, **כולל** שלב יצירת ה DB. אין לכתוב SQL ליצירת ה DB ו/או קשרים ואין לבנות טבלאות ו/או קשרים באמצעות ממשקי ניהול למיניהם.
  19. המערכת תתמוך בלפחות 2 שאלות חיפוש המאפשרות ללקוח להגדיר פרמטרים (לפחות 3 פרמטרים) לחיפוש באמצעות ממשק המשתמש. (לדוגמא, כמו בזאפ: חיפוש טלוויזיה על פי גודל מסך, רזולוציה, ושם היצרן). שימו לב, סעיף זה הינו בנוסף לחיפוש הנדרש בסעיף 17.
  20. המערכת תתמוך בלפחות 2 שאלות המבצעות Join בין 2 סוגי ישויות או יותר. מדובר על Join בקוד שלכם ממש ולא רק ש"מאחורי הקלעים" הקוד יתורגם לJoin.
  21. המערכת תתמוך בלפחות שאלות אחת המבצעות Group By. מדובר על Group by בקוד שלכם ממש ולא רק ש"מאחורי הקלעים" הקוד יתורגם ל Group by.
  22. המערכת תכיל ממשק מנהל עם הרשאת גישה באמצעות שם משתמש וסיסמא. למנהל יהיו יכולות עריכה/חיפוש מורחבות לעומת משתמש רגיל. יש להקפיד על מנגנון הרשאת הגישה כך שעמודים ופונקציונאליות ניהול יהיו נגישים רק למי שיש לו הרשאות.
  23. יש לטפל במקרי קצה, שגיאות, ולידציות וכו' – הן בצד השרת והן בצד הלקוח.
  24. ב Views של המערכת יעשה שימוש נרחב ביכולות ספריית JQuery, כולל שימוש ב Ajax מתוך JQuery. לגבי כל קטע קוד בצד הלקוח (Javascript) יש לשקול מימוש ב JQuery.
  25. המערכת תתמוך ב HTML5 ותעשה שימוש ביכולות הבאות:
    - i. Video
    - ii. Canvas
    - iii. Aside, footer, header, nav, section
  26. המערכת תתמוך ב CSS 3 ותעשה שימוש ביכולות הבאות:
    - i. Text-shadow
    - ii. Transition
    - iii. Multiple-columns
    - iv. Font-face
    - v. Border-radius
  27. את מימוש קוד הלקוח מבחינת נראות (HTML, CSS, Bootstrap) יש לממש בעצמכם ולא ניתן להשתמש בתבניות מוכנות. (התבנית המובנת ב visual studio לאחר יצירת פרויקט מותרת לשימוש – אך היא מהווה נק' פתיחה בלבד. היא אינה מספיקה בפני עצמה ויש להוסיף לה מימוש צד לקוח עיצובי ניכר).
  28. כמו כן:
    - i. המערכת תציג נתונים סטטיסטיים בלפחות 2 גרפים (לדוגמא – ממוצע סכום הרכישות המצטבר לכל חודש) באמצעות החבילה D3.JS : <http://d3js.org>
    - ii. למערכת יהיה שימוש ב Web Service אחד לפחות (למשל עדכונים מהבורסה, טמפרטורות, חדשות וכד' – כמובן לפי ה context של האתר הספציפי)
    - iii. באחד מדפי המערכת תוצג מפה מבוססת Google Maps/Bing Maps ובה מסומנות כתובות שנקראו מבסיס הנתונים (למשל רשימת סניפים של רשת החנויות)
    - iv. המערכת תתממשק ל Twitter API או Facebook API ותאפשר לקבל/לשדר נתונים בהתאם (לוגין לא נחשב) – למשל – עדכון דף ה Facebook של המערכת לגבי מוצרים חדשים, מספר הלקוחות הגולשים וכד'.

**בהצלחה!**