**דרישות לפרויקט גמר בקורס מעבדת פיתוח אפליקציות אינטרנטיות**

**ד"ר שי הורוביץ, ד"ר איגור רוכלין**

**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.

בנוסף, באפשרותכם להשתמש בתרגילים שהגשתם במהלך הסמסטר (הבלוג של שאולי) כבסיס על מנת לממש את הפרויקט.

**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 לקורס אצל **מרצים שונים**. כן ניתן לשלב בקבוצה סטודנטים אשר רשומים לקבוצות שונות של אותו המרצה.
3. טרם ההגנה, יש להגיש את הקוד של פרויקט הקורס (לבצע Zip לספריית הפרויקט) בתיבה שנפתחה לשם כך במודל. לא ניתן לבצע את ההגנה לפני שהגשתם את הקוד. מומלץ לבצע זאת בבוקר ההגנה על מנת שזו תהיה הגירסה האחרונה של הקוד שפיתחתם.
4. במעמד ההגנה, יש להפעיל את המערכת בגירסת הקוד שהועלתה למודל בלבד. אין לבצע שינויים לגירסה זו.
5. 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
6. עליכם לוודא שניתן להדגים את המערכת במעבדה שנקבעה טרם מועד ההגנה. מנסיוננו – 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.
7. מועד ההגנה יתפרסם בנפרד.

**דרישות טכניות:**

1. המערכת תתוכנן ותמומש באמצעות MVC גירסה אשר מותקנת במעבדת המחשבים במכללה. תהיה הפרדה ברורה בין ה View, ה Controller וה Model.
2. על המערכת לתמוך בלפחות 3 מודלים שונים. (למשל בחנות: מודל אחד על המוצרים שנמכרים בחנות, מודל נוסף על הלקוחות ומודל על הספקים). חובה שלפחות בין 2 מודלים מתוך ה 3 – יהיה קשר (אשר יפורש כמפתח משני ב DB)
3. 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).
4. הגישות לכלל נתוני המערכת ב DB יהיו באמצעות Entity Framework בלבד (שימו לב שלא יתאפשר לכתוב בקוד שאילתות ב SQL אלא רק באמצעות פקודות Entity Framework ישירות או עם LINQ).
5. המערכת תתמוך בלפחות 2 שאילתות חיפוש המאפשרות ללקוח להגדיר פרמטרים (לפחות 3 פרמטרים) לחיפוש באמצעות ממשק המשתמש. (לדוגמא, כמו בזאפ: חיפוש טלויזיה על פי גודל מסך, רזולוציה, ושם היצרן)
6. המערכת תתמוך בלפחות 2 שאילתות המבצעות Join בין 2 סוגי ישויות או יותר
7. המערכת תתמוך בלפחות שאילתה אחת המבצעת Group By
8. המערכת תכיל ממשק מנהל עם הרשאת גישה באמצעות שם משתמש וסיסמא. למנהל יהיו יכולות עריכה/ חיפוש מורחבות לעומת משתמש רגיל
9. 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
10. המערכת תתמוך ב HTML5 ותעשה שימוש ביכולות הבאות:
    1. Video
    2. Canvas
    3. Aside, footer, header, nav, section
11. המערכת תתמוך ב CSS 3 ותעשה שימוש ביכולות הבאות:
    1. Text-shadow
    2. Transition
    3. Multiple-columns
    4. Font-face
    5. Border-radius
12. כמו כן:
    1. המערכת תציע מוצר היכול לעניין את הלקוח על פי למידת סטטיסטיקה (היסטוריה) - לא מדובר ב query של DB. בדוגמא של "הבלוג של שאולי" המערכת תציע ללקוח post שיכול לעניין אותו.
    2. המערכת תציג נתונים סטסטיטיים בלפחות 2 גרפים (לדוגמא – ממוצע סכום הרכישות המצטבר לכל חודש) באמצעות החבילה D3.JS : <http://d3js.org/>
    3. למערכת יהיה שימוש ב Web Service אחד לפחות (למשל עדכונים מהבורסה, טמפרטורות, חדשות וכד' – כמובן לפי ה context של האתר הספציפי)
    4. באחד מדפי המערכת תוצג מפה מבוססת Google Maps/Bing Maps ובה מסומנות כתובות שנקראו מבסיס הנתונים (למשל רשימת סניפים של רשת החנויות)
    5. 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!**