**דרישות לפרויקט גמר בקורס נושאים מתקדמים באפליקציות 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.

**קבוצות:** ניתן לעבוד בקבוצות של 3- 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.

שימו לב, ללא קשר לאיזה הגנה אתם ניגשים, הגשת הפרויקט היא 24 שעות לפני ההגנה הראשונה. כלומר, גם אם אתם ניגשים להגנה השניה, אתם צריכים להגיש את הפרויקט 24 שעות לפני ההגנה הראשונה, ולא ניתן לבצע שינויים לאחר מכן.

**דרישות:**

1. המערכת תתוכנן ותמומש באופן הבא:
   1. תשתית המערכת תבוסס Node.JS בצד השרת
   2. שימוש ב Express.JS web framework בצד השרת
   3. אחסון ושליפת הנתונים מתוך Mongo DB ו firebase
   4. צד לקוח מבוסס React וגם MVW
2. טכנולוגיות צד הלקוח הנ"ל ישתמשו ב HTML5 וב bootstrap ו react design
3. האפליקציה תתבסס על עקרונות ה Single Page Application – משמע תתבצע טעינה של הדף הראשי פעם אחת בלבד וכל העדכונים הנוספים יעבדו על בסיס AJAX ו/או WebSockets
4. 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), עבור לפחות 3 סוגים שונים של אובייקטים.
5. המערכת תכיל לפחות 3 סוגי מסמכים ב MongoDB אשר קשורים ביניהם באמצעות reference
6. המערכת תתמוך בלפחות 2 שאילתות חיפוש המאפשרות ללקוח להגדיר פרמטרים (לפחות 3 פרמטרים) לחיפוש באמצעות ממשק המשתמש (כמו למשל חיפוש טלויזיה בזאפ על פי: גודל מסך, רזולוציה, משקל וכד')
7. המערכת תתמוך בלפחות שאילתה אחת המבצעת פעולת Group By ב MongoDB
8. המערכת תתמוך בלפחות שאילתה אחת המבצעת פעולת Map-Reduce ב MongoDB
9. המערכת תתמוך בניהול משתמשים באמצעות ה firebase
10. על המערכת להכיל לפחות מרכיב אחד המממש Two-Way-Binding בין השרת לבין ה Browser תוך שימוש ב WebSockets כך שגם השרת יכול ליזום עדכונים מול ה Browser. למשל במסך ה Admin יוצג שדה באתר המראה את מספר הלקוחות המחוברים באותו רגע (כל משתמש חדש שמתווסף יגרום לשרת לדווח על כך לכל המשתמשים שהם Admin המחוברים באמצעות WebSockets ללא שימוש ב Polling, וכל אחד מה Admins יוכל לבחור לעדכן את הנתון הזה, שיגרום לעדכון מיידי הן בשרת והן אצל שאר ה Admin-ים המחוברים באותו הרגע. דוגמא אחרת – אתר חדשות שמראה Live הודעות שהגיעו באותו הרגע ללא צורך ב Refresh/polling מה Browser, וה Admin יכול גם לערוך חדשות קיימות מה שיגרום עדכון מיידי בשרת, ב DB ואצל יתר הלקוחות שמחוברים כרגע לאתר). את רכיב זה יש לממש כ Custom Binding ב Angular.
11. על המערכת להכיל לפחותCustom Directive אחד המציג נתונים בגרף תוך שימוש בחבילה D3.js
12. על המערכת להכיל לפחות 2 סוגי hooks וגם costume hook
13. על המערכת להתבסס על נתונים שנשלפו מאתרים/מערכות אמיתיות באמצעות Scraping שייכתבו לתוך MongoDB – המכיל ביום ההגשה/הגנה לפחות 100 אובייקטים שנוצרו אוטומאטית כתוצאה מתהליך ה Scraping (ולא ידנית. ניתן להוסיף ידנית בנוסף ל 100 האלה)
14. חובה להשתמש בטכנולוגיות שנלמדו בקורס (ובהם בלבד).
15. הדרישות שתוארו לעיל הן דרישות המינימום עבור הפרוייקט. עבור ציון גבוה יש להגיש פרויקט מלא **המציג מערכת שלמה ומתפקדת (מעשית ולוגית) מקצה לקצה**, אשר מכיל מעבר לדרישות המינימום. למשל, במקרה של חנות, יש לממש את כל התרחישים הסטנדרטים שהיינו מצפים מחנות קלאסית , כמו למשל: מנוע חיפוש, עומד מוצר, עגלת קניות וכו'.

**בהצלחה!**