**הנחיות פרויקט גמר – קורס פיתוח Full-Stack**

כללי:

במסגרת פרויקט הגמר, נפתח חנות אינטרנטית לחברה מסויימת (נתון לבחירתכם).

דוגמאות: חנות מקוונת למכירת אופניים, פלטפורמה לשיתוף ספרים, חנות יד שניה, חנות לסופר אינטרנטי.

**השירות יכלול:**

* מוצרים הכוללים תמונה ותיאור (חובה), ומחיר במידת הצורך.
* קטגוריות משתמשים - כגון אורחים,משתמשים רשומים, עורכים, ומנהלים. עם הרשאות בהתאם.(בהתאם לצורך)
* עריכת מוצרים רק על ידי משתמשים מורשים לפי הרשאתם.
* ניהול משתמשים (רק ע"י מנהל האתר)
* ניהול הזמנות והיסטורית הזמנות
* אזור תמיכה – בו משתמשים יכולים לפתוח "טיקט" לתמיכה ולקבל תשובה מבעל האתר.

צד שרת של הפרויקט יתבצע בסביבת nodejs ו-express ב-JavaScript, הנתונים ישמרו בבסיס נתונים של MongoDB.

**דגשים:**

* האתר יפותח בתצורת SPA בלבד (חוץ ממסכי לוג-אין, אם נדרש)

צד לקוח יפותח בעזרתAngular על בסיס html5, css3 תוך שימוש בספריות לעיצוב כגון Bootstrap

האתר יהיה מעוצב ויכלול תפריט עליון, עיצוב בסיסי, הצגת מוצרים, פוטר ואפשרויות נוספות בהתאם לפרויקט.

כל אתר חייב לכלול את הדפים הבאים:

* + אודות החברה
  + קטלוג מוצרים
  + דף מוצר בודד
  + סל קניות
  + היסטורית הזמנות (ללקוח רשום)
  + צ'אט\לוג-אין
  + ממשק ניהול ויזואלי הדורש כניסה עם שם משתמש וסיסמה להצגת ההזמנות שבוצעו.
  + ניתן יהיה להוסיף/לערוך/למחוק מוצרים והזמנות.
  + פתיחת פניה לתמיכה

חלקים נוספים יקבעו על ידי מבצע הפרויקט בהתאם לתוכניות ובהתאם לאופי הפרויקט.

**ניהול משתמשים**:

* אימות המשתמש חייב להתבצע רק בצד שרת, ניתן לבצע זאת גם על ידי שימוש בכלים נוספים אשר למדנו כגון FireBase
* ניהול משתמשים יכלול יכולת למחוק, להוסיף ולשנות את תפקיד המשתמש או פרטיו האחרים (ע"י המנהל)
* המערכת תאפשר רישום עצמי של משתמשים מהקטגוריה הנמוכה, כל משתמש יוכל לעדכן את פרטיו המתאימים – כגון סיסמה, כתובת, טלפון וכו', לבקש לאפס את סיסמתו ("שכחתי את הסיסמה").

**שרות תמיכה:**

אזור התמיכה באתר הוא אזור ללקוחות רשומים, וכדי לפרסם בקשה לתמיכה יש צורך להרשם/להתחבר.

אזור זה יכלול פתיחה של קריאות שירות, כאשר לכל קריאת שירות יש סטטוס.

**דרישות כלליות**:

* מבנה הפרויקט צד שרת יהיה תואם לארכיטקטורה MVC
* מצב כניסה ונתוני גלישה אחרים הרלוונטים לגלישה (כולל סל קניות של גולש מזדמן) תנוהל ע"י session שפוקע תוך זמן סביר (למשל 15 דקות)
* השירות לא ישלח לכיוון צד הלקוח שום נתון ושום אלמנט אתר שלא תואם את הרשאות ואת הסמכויות של המשתמש – ההחלטה תתקבל בעזרת בקרה של השרת על בקשות AJAX.
* השרת יבדוק את כל הנתונים המתקבלים מהלקוח ולעולם לא יסמוך על הבדיקה בקוד צד לקוח

**דרישות לא פונקציונאליות**:

1. המערכת תתבסס על טכנולוגיות MEAN וחבילות הרחבה נוספות שתמצאו לנכון להוסיף.
2. יש לארגן פיזית את הקוד בעץ ספריות סטנדרטי, ניתן להשתמש בחבילת express-generator
3. מומלץ לעצב את התצוגה באמצעות BootStrap או לבחור תבנית מוכנה ולהתאימה לצרכים ( לדוגמא: <http://w3layouts.com/> ), בכל מקרה על המערכת להיות אסתטית ומקצועית.
4. השתדלו, במידת האפשר ובאופן סביר, להעדיף עיבודים בצד הלקוח על פני צד השרת, כדי לשפר את היכולת לשרת מקסימום בקשות ומשתמשים בו זמנית – אך יש לבדוק\לאמת נתונים בצד שרת לפני ביצוע כל פעולה.
5. **בכל פרויקט חובה להוסיף שימוש בלפחות שתי טכנולוגיות שלא נלמדו במסגרת הקורס – כגון שליחת מיילים עם Node , תשלום דרך paypal, וכדומה.**

**הנחיות נוספות:**

1. חובה לבצע איפיון מלא בטרם התחלת עבודה. דבר אשר יחסוך לכם בעיות בהמשך. (מצורף דוגמא)
2. את הפרויקט יש להגיש דרך העלאה לאתר GIT
3. מומלץ להעלות את הפרויקט גם לפרודקשן ולהציג לינק לאתר חי.
4. בהצגת הפרויקט הרשמית תצטרכו להציגו בשילוב מצגת הסוקרת את כל האפשרויות.
5. סביר מאוד שבמהלך ביצוע הפרויקט תצטרכו לחקור באינטרנט נושאים מסויימים/ללמוד נושאים חדשים.
6. **קחו את הפרויקט ברצינות, הוא גם מעולה להצגה בפני מעסיקים פוטנציאלים.**

**בהצלחה!**