

דרישות פרויקט גמר בקורס פיתוח אפליקציות אינטרנטיות

כללי: עליכם לממש חנות אינטרנטית בהתאם לדרישות ותוך שילוב הטכנולוגיות המופיעות בסעיף הדרישות.

דרישות מנהליות:

1. ניתן לעבוד בקבוצות של 3 עד 5 סטודנטים בלבד (לא יותר ולא פחות). קבוצות של 5 סטודנטים נדרשות להציג פרויקט מושקע יותר מאשר קבוצות של 3-4 סטודנטים. 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, חובה על כל הסטודנטים להשתמש, בזמן פיתוח הפרויקט, במערכת לניהול גרסאות (**Git**) ובמועד ההגנה להציג את **ההתנהלות (וההשקעה)** של **כל אחד** מחברי הקבוצה. חובה לעבוד דרך הגיט, כאשר כל אחד מחברי הקבוצה עובד בנפרד מהמחשב שלו ומחשבון הגיט שלו (ולא ששניים או יותר מחברי הקבוצה עובדים על אותו מחשב או דרך אותו חשבון גיט). חובה לעבוד בגיט באופן נכון, כלומר, לבצע, `commits, pulls, pushes`, `merges` וכו'.
2. **לא ניתן לשלב בקבוצה סטודנטים אשר רשומים לקורס אצל מרצים שונים. כל חברי הקבוצה חייבים להיות רשומים כולם לאותו המרצה.** באחריותכם לבדוק ולוודא לאיזה מרצה אתם רשומים ולהרכיב קבוצות בהתאם. שימו לב שבמודל כל הסטודנטים רואים את אותו האתר (לא משנה לאיזה מרצה הם רשומים), אבל כל סטודנט רשום רק לאחד מהמרצים ויכול להיבחן אך ורק אצלו. **לא יתקבלו בקשות לערבב קבוצות, ובפרט כתוצאה מטעות ואי ההבנה לאיזה מרצה אתם שייכים.**
3. מועדי ההגנות מפורטים בלוח הבחינות באתר המכללה.
4. כל סטודנט רשאי להבחן בהגנה אחת בלבד. בפרט, לא ניתן לגשת שוב או לתקן או לשפר 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 אליה אתם ניגשים. תיבת ההגשה תיסגר בשעה 23:59 ביום שלפני ההגנה.
7. סטודנט שלא הגיש במועד את הפרויקט – ירדו לו 20 נקודות מהציון הסופי בקורס.
8. יש להגיש את הקוד של פרויקט הקורס (לבצע Zip לקוד הפרויקט) בתיבה שנפתחה לשם כך במודל, **וכן יש להוסיף הערה להגשה במודל המכילה קישור לגיט של הפרויקט.** יש לוודא שהגיט פתוח (מבחינת גישה צפיה) לכל מי שיש לו גישה לקישור. לכן, וודאו בזמן הפיתוח שאינכם מעלים לגיט שום דבר המכיל מידע רגיש, כמו סיסמאות, API-keys, טוקנים וכו'.
9. במעמד ההגנה, יש להפעיל את המערכת בגירסת הקוד שהועלתה למודל בלבד. אין לבצע שינויים לגרסה זו.
10. בהגנה יבדק הפרויקט, כאשר המרצה מתפעל ובודק את המערכת, את התכולה, והאם יש נפילות / באגים. המרצה יכול לשאול את כל הסטודנטים שאלות (לקבוצה ככלל ולסטודנטים ספציפיים בפרט) כשקוד המערכת פתוח בפניו, ובפרט, שאלות אודות הקוד והמימוש. מעמד ההגנה מהווה רק חלק מהציון הקשור למראה הכללי של המערכת, נפילות/באגים, שליטת הסטודנטים בקוד וחלוקת העבודה בקבוצה והעבודה כצוות. חלק נוסף במתן הציון מבצע המרצה לאחר המפגש עם הקבוצה, כאשר הוא בודק את הפרויקט בכלל ואת קוד המערכת בפרט באופן עצמאי על מנת לתת ציון סופי – על כן, לא ניתן לבשר על ציון סופי במהלך מפגש ההגנה.
11. עליכם לוודא שניתן להדגים את המערכת במעבדה שנקבעה טרם מועד ההגנה. מנסינונו – 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. **עליכם לוודא שהמערכת רצה על כל המחשבים האישיים שלכם, ובנוסף, שהיא מוכנה להצגה במידת הצורך במהלך ההגנה.**

12. חובה להשתמש בטכנולוגיות, שפות וארכיטקטורות שנלמדות בקורס (ובהן בלבד) ובאופן שבו הן נלמדות – ולא ניתן להשתמש בחלופות אחרות במקומן. סעיף זה רלוונטי במיוחד עבור סטודנטים הבוחרים לעבוד על הפרויקט שלא במקביל לקצב העברת החומר בכיתה. לא תוכלו להשתמש בטכנולוגיות שלא נלמדו בכיתה או שלא אושרו לשימוש. למשל, אין להשתמש ב React או Angular.
13. חלק מדרישות הפרויקט הינן דרישות במסגרת למידה עצמית. כלומר, בעזרת החומר הנלמד בכיתה ניתן לכם כלים, ובעזרתם עליכם יהיה ללמוד עצמאית כיצד לבצע דרישות אלו.
14. הפרויקט והקוד בו חייבים להיות פרי עבודה אישי של הקבוצה שלכם, ולהיכתב על ידכם. במהלך בדיקת הפרויקט יהיו בשימוש כלים ידניים ואוטומטיים לזיהוי קוד שאינו פרי עבודה מקורית של חברי הקבוצה.
15. ספציפית, את מימוש קוד הלקוח מבחינת נראות (HTML, CSS, Bootstrap) חובה לממש בעצמכם ואסור להשתמש בתבניות מוכנות או קטעי קוד מוכנים או לקחת מאתרים קיימים.
16. 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:

17. תשתית המערכת תבוס Node.JS בצד השרת בשימוש ב Express.
18. אחסון ושליפת הנתונים מתוך MongoDB.
19. המערכת תתוכנן ותמומש באמצעות MVC. תהיה הפרדה ברורה בין ה View, ה Controller וה Model.
20. על המערכת לתמוך בלפחות 3 מודלים שונים. (למשל: מודל אחד על המוצרים שנמכרים בחנות, מודל נוסף על הלקוחות ומודל על הספקים).
21. 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). שימו לב, **לא** מספיק שהקוד יכיל את המימוש, אלא האפליקציה צריכה לאפשר למשתמש לבצע את **כל אחת** מהפעולות הללו על כל אחד מהמודלים.
22. המערכת תתמוך בלפחות 2 שאליות חיפוש המאפשרות ללקוח להגדיר פרמטרים (לפחות 3 פרמטרים) לחיפוש באמצעות ממשק המשתמש. (לדוגמא, כמו בזאפ: חיפוש טלויזיה על פי גודל מסך, רזולוציה, ושם היצרן). סעיף זה הוא **בנוסף** לדרישת החיפוש של סעיף 21.
23. המערכת תתמוך בלפחות שאלית אחת המבצעת פעולת GroupBy ב MongoDB.
24. המערכת תכיל ממשק מנהל עם הרשאת גישה באמצעות שם משתמש וסיסמא. למנהל יהיו יכולות עריכה וחיפוש מורחבות לעומת משתמש רגיל. יש להקפיד על מנגנון הרשאת הגישה כך שעמודים ופונקציונאליות ניהול יהיו נגישים רק למי שיש לו הרשאות. בפרט, כל משתמש יכול לראות רק את המידע הפרטי שלו ולא של משתמש אחר. למשל: רק מנהל יכול להוסיף מוצרים לחנות, כל משתמש יכול לראות רק את היסטוריית ההזמנות שלו וכו'.
25. חובה על המערכת להכיל עגלת קניות ולאפשר למשתמשים לבצע הזמנות. כל משתמש יוכל לראות את היסטוריית ההזמנות שלו באיזור האישי שלו באתר (כל הזמנת עבר מציגה את רשימת הפריטים + מחיר). מנהל החנות יכול לראות ולנהל את כל ההזמנות של הלקוחות על ידי ממשק המנהל (כולל לראות את כל פרטי ההזמנה והמוצרים).
26. טרם ההגנה, יש להזין לחנות מספיק מידע בכדי שהפרויקט ידמה חנות אמיתית. כלומר, צריך להזין מוצרים, לבצע הזמנות וכו'.
27. **חובה** לטפל במקרי קצה, שגיאות, ולידציות וכו' – הן בצד השרת והן בצד הלקוח, ובפרט, להימנע מקריסה של השרת, גם בעת הזנה של קלט חריג והתנהגות לא צפויה של המשתמש.
28. ב Views של המערכת יעשה שימוש נרחב ביכולות ספרייט Jquery, כולל שימוש ב Ajax מתוך Jquery, ובפרט, ביצוע קריאות Ajax אל השרת שלכם. לגבי כל קטע קוד בצד הלקוח יש לשקול מימוש ב Jquery.
29. המערכת תתמוך ב HTML5 ותעשה שימוש ביכולות הבאות:

- .i Video
- .ii Canvas
- .iii Aside, footer, header, nav, section
- 30. המערכת תתמוך ב CSS 3 ותעשה שימוש ביכולות הבאות:
 - .i Text-shadow
 - .ii Transition
 - .iii Multiple-columns
 - .iv Font-face
 - .v Border-radius
- 31. חובה להכיל רכיב במערכת אשר מעביר את הנתונים בין השרת ללקוח באמצעות WebSockets / Socket.io
- 32. כמו כן:
 - .i המערכת תציג נתונים סטטיסטיים בלפחות 2 גרפים (לדוגמא – ממוצע סכום הרכישות המצטבר לכל חודש) באמצעות החבילה: D3.JS. על הנתונים המוצגים להגיע מה DB ולהציג מידע דינמי בהתאם למידע הקיים ב DB ברגע נתון.
 - .ii למערכת יהיה שימוש ב Web Service אחד לפחות (למשל עדכונים מהבורסה, טמפרטורות, חדשות וכד' – כמובן לפי ה context של האתר הספציפי). שימו לב: יש "לצרוך" את ה web service ממש, כלומר, לשלוח אליו נתונים ולקבל נתונים בחזרה ולהציג את הנתונים. לא מספיק להטמיע iframe או קוד בסיסי. עליכם לכתוב בעצמכם את הקוד שפונה לservice ומטפל בתשובה ומציג אותה למסך. אינכם יכולים להשתמש בקוד מוכן או ספרייה.
 - .iii באחד מדפי המערכת תוצג מפה מבוססת Maps Google/Maps Bing 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'. הטמעה של iframe או רק כפתורי לייק/שיתוף/פרסום או קוד שפשוט פותח דפים, חלונות או ממשקים של Twitter/Facebook וכדו' אינה נחשבת.

בהצלחה!