

## פרויקט סיום קורס

יש לפתח צד שרת (node.js) וצד לקוח (אפליקצית React) אפשר לבצע את הפרויקט בקבוצות של בין סטודנט אחד ל-4, תכולת הפרויקט וההקפדה בהגנה תלויה בגודל הקבוצה.

תכולה הפרויקט:

1. התחברות:
  - 1.1 הרשמה ולוגאין של משתמש לפי שם משתמש וסיסמה
  - 1.2 אפשרות הרשמה ע"י ספק חיצוני (google או facebook)
  - 1.3 יש להשתמש ב-JWT כולל Refresh token כפי שנלמד בכיתה (בנוסף לספק חיצוני).
  - 1.4 האפליקציה צריכה לזכור משתמש שהתחבר.
  - 1.5 לאפשר למשתמש גם להתנתק
2. מסך פרטי משתמש:
  - 2.1 מציג את פרטי המשתמש כולל תמונה
  - 2.2 הצגת הפוסטים של אותו משתמש
  - 2.3 עבור המשתמש המחובר תהיה אפשרות עריכה (של התמונה ושל שם המשתמש, אימייל וסיסמה אין צורך לעדכן).
3. הצגת תכנים:

האפליקציה תציג תכנים אשר נלקחו ממקור חיצוני - REST, יש להשתמש באחד ממקורות ה-AI הזמינים: גמיני, chatGPT... המידע צריך להיות רלוונטי לאפליקציה.

יש לשים לב להתממשקות מול ה-API כך שלא תהיה חריגה במספר הבקשות!
4. שיתוף תכנים:
  - משתמש יוכל להעלות מידע לאפליקציה הכולל:
    - טקסט
    - תמונה
  - משתמשים אחרים יוכלו לראות את המידע שהועלה
  - המשתמש שהעלה יוכל לעדכן את המידע שהוא פרסם גם את התמונה וגם את הטקסט
  - המשתמש שהעלה יוכל למחוק את המידע שהוא פרסם
  - כל משתמש יוכל לראות את התכנים שהוא העלה (מסך נפרד או סינון)
  - הצגת התכנים בגלילה עם paging (טעינה מדורגת, לא דפים נפרדים)
  - תמונת פרופיל ושם משתמש לא נחשב לשיתוף תמונה/טקסט
5. יש להוסיף אפשרות למשתמשים להגיב על התכנים של משתמשים אחרים, התגובות יוצגו במסך נפרד ולא במסך הראשי, כאשר במסך הראשי יופיע מספר התגובות שנכתבו על אותו תוכן.
6. יש לאפשר למשתמש לסמן פוסטים נבחרים (like) וכמובן להציג את הסימון... יש להתייחס למימוש מתאים ויעיל ב-DB.
7. חובה לעבוד עם גיט, כל משתמש עם הפרופיל שלו, לעשות קומיטים קטנים. לעבוד עם pull request כפי שנדרשתם לבצע בתרגילים. חלוקת הקוד בין חברי הקבוצה צריכה להיות סבירה!
8. יש לתעד את כל ה-API באמצעות swagger

9. יש לכתוב unit-test עבור כל ה-API של השרת (פרט לשימוש ב-API החיצוני).
10. יש להשתמש ב-typescript בלבד
11. תמונות יש לשמור בשרת שלכם, לא ב-DB ולא בשרות חיצוני
12. יש להשתמש במונגו מקומי (בשרת של המכללה) בלבד (לא אטלס או שירותי ענן אחרים).
13. יש להעלות את האפליקציה לשרת של המכללה שיהיה נגיש מכל מקום דרך URL.
  - 13.1 הפעלת האפליקציה עם https (גם ה-backend וגם ה-frontend)
  - 13.2 האפליקציה צריכה לעבוד ברקע גם אם הטרמינל שלכם סגור - יש להשתמש ב-PM2
  - 13.3 האפליקציה חייבת לרוץ במוד production  
node\_env = production
  - 13.4 יש להשתמש במונגו על השרת ולא בשרותי ענן כאשר ההתחברות עם מונגו מוגנת ע"י שם משתמש וסיסמה.
  - 13.5 עליכם לדאוג שהכניסה לאפליקציה תתבצע דרך הדומיין שקיבלתם מדורון - כלומר אני מקליד את שם הדומיין (ללא פורט) והאפליקציה נפתחת אצלי במחשב ללא VPN.
  - 13.6 **אסור בכל מקרה להשתמש בשרותי hosting חיצוניים**
  - 13.7 **בהתאם להתקדמות בחומר יש להשתמש ב-Docker (במידה ונלמד בקורס)**
14. עיצוב, יש לדאוג לעיצוב המסכים, שימוש בצבעים, רקעים, ניצול שטח המסך... עיצוב לא טוב יגרור הורדת ציון.
15. תוספת תכולה לקבוצה גדולה (3-4 מפתחים):
  - הוספת socket.io לצ'ט בין משתמשים (לימוד עצמי).
  - הדרישות מהצט ששני משתמשים יוכלו לנהל שיחה דרך האפליקציה.
  - וצריך לשמור את הצט ב-DB כך שאם משתמש סגר את האפליקציה ונכנס מחדש הצט ישמר.
  - בחירת משתמש לצט תתבצע מתוך האפליקציה - מתוך מסך פרטי משתמש.
16. יש לעבור עם גיט בשיטת branch and pull request כפי שהוצג בכיתה.
17. יש להשתמש בכלים שנלמדו בכיתה כמו בדיקת קלט בטפסים וכו...

אופן הגשה:

- יש להגיש את הפרויקט למודל יום לפני ההגנה.
- רק סטודנט אחד מהקבוצה מגיש
- מגישים קובץ עם הפרטים הבאים:
  - מס תז של כל הסטודנטים בפרויקט
  - קישור לריפו פתוח בגיט (public)
  - קישור לסרטון קצר שמראה את הפרויקט עובד, כולל swagger ו-jest