

## פרויקט סיום קורס

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

תכללה הפרויקט:

1. התכונות:
  - .1 הרשמה ולוגאין של משתמש לפי שם משתמש וויסמה
  - .2 אפשרות הרשמה ע"י ספק חיצוני (facebook או google)
  - .3 יש להשתמש ב- JWT כולל Refresh token כפי שנלמד בכיתה (בנוסף לשפק חיצוני).
  - .4 האפליקציה צריכה לזכור משתמש שהתחבר.
  - .5 לאפשר למשתמש גם להתנתק
2. מסך פרטי משתמש:
  - .2.1 מציג את פרטי המשתמש כולל תמונה
  - .2.2 הצגת היפותים של אותו משתמש
  - .2.3 עבור המשתמש המחבר תהיה אפשרות עריכה (של התמונה ושל שם המשתמש, אימיל וויסמה אין צורך לעדכן).
3. הצגת תכנים:

האפליקציה תציג תכנים אשר נלקחו ממוקור חיצוני - REST, יש להשתמש באחד מקורות ה- AI הקיימים: גמיינி, ...chatGPT ... המודיע צריך להיות רלוונטי לאפליקציה.

יש לשימוש לב להתמכחות מול ה- API כך שלא תהיה חריגה במספר הבקשות!
4. שיתוף תכנים:
  - משתמש יוכל להעלות מידע לאפליקציה הכלול:
    - טקסט
    - תמונה
  - משתמשים אחרים יכולים 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 בגליליה עם paging (טעינה מדורגת, לא דפים נפרדים)
  - תמונה פרופיל ושם משתמש לא נחשב לשיתוף תמונה/טקסט
5. יש להוסיף אפשרות למשתמשים להגיב על התכנים של משתמשים אחרים, התגובה יוצגו במסך נפרד ולא במסך הראשי, כאשר במסך הראשי יופיע מספר התגובות שנכתבו על אותו תוכן.
6. יש לאפשר למשתמש לסמן פוטטים נבחרים (like) וכמוון 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 כדי שנדרשتم לבצע בתרגילים. חלוקת הקוד בין חברי הקבוצה צריכה להיות סבירה!
8. יש לתעד את כל ה- API באמצעות swagger

- .9. יש לכתוב `test-it-out` עבור כל ה- `api` של השירות (פרט לשימוש ב- API החיצוני).
- .10. יש להשתמש ב- `typescript` בלבד
- .11. תמונות יש לשמר בשרת שלכם, לא ב- DB ולא בשירות חיצוני
- .12. יש להשתמש במונגו מקומי (בשרת של המכללה) בלבד (לא אטלוס או שירותים עננים אחרים).
- .13. יש להעלות את האפליקציה לשרת של המכללה שיהיה נגיש מכל מקום דרך URL.
- .13.1. הפעלת האפליקציה עם `https` (גם `http`- `backend` וגם `http`- `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 השירות ולא בשירותי ענן כאשר ההתחברות עם מונגו מוגנת ע"י שם משתמש וסיסמה.
- .13.5. עליהם לדאוג שהכינוי לאפליקציה תבוצע דרך הדומיין שקיבלתם מדורון - ככלומר אני מקליד את שם הדומיין (ללא פורט) והאפליקציה נפתחת אצלி במחשב ללא VPN.
- .13.6. אסור בכל מכירה להשתמש בשירותי `hosting` חיצוניים
- .13.7. בהתאם להתקדמות בחומר יש להשתמש ב- `Docker` (במידה ונלמד בקורס)
- .14. עיצוב, יש לדאוג לעיצוב המרכיבים, שימוש בצבעים, רקעים, ניצול שטח המסך... עיצוב לא טוב יגרור הורדת ציון.
- .15. נוספת תacula לקובוצה גדולה (4-3 מפתחים):  
הוסף `socket.io` לציט בין משתמשים (לימוד עצמי).  
הדרישות מהצט שני מושגים יוכלו לנחל שיחה דרך האפליקציה.  
צריך לשמור את הצט ב- DB כך שגם משתמש סגור את האפליקציה ונכנס מחדש הצט ישמר.  
בחירה משתמש לצט תבוצע מתוך האפליקציה - מתוך מסך פרטיו משתמש.
- .16. יש לעבור עם גיט בשיטת `branch and pull request` כפי שהוזג בគיטה.
- .17. יש להשתמש בכלים שנלמדו בគיטה כמו בדיקת קלט בטפסים וכו...

#### אוף הגשה:

- יש להציג את הפרויקט למודל יום לפני ההגנה.
- רק סטודנט אחד מהקובוצה מגיש
- מגישים קובץ עם הפרטים הבאים:
  - מס' תז של כל הסטודנטים בפרויקט
  - קישור לריפורוטה בגייט (`swagger`)
  - קישור לסרטון קצר שמרת את הפרויקט עובד, כולל `jest` ו- `swagger`