

**במידה והמשימה מתבצעת מרחוק, חובה לבצע את המשימה עם מצלמה פתוחה.**

**חל איסור מוחלט לדבר ולהיעזר בחברכם לכיתה או בכל גורם אחר.**

**חל איסור מוחלט ליצור קשר דרך האינטרנט בין חבריכם לכיתה או מול כל גורם אחר.**

## Angular-NodeJs-MongoDBMySQL Task V1



**הוראות:**

1. ליצור ביצוע המשימה יש ברשותכם שלוש וחצי שעות.
2. לאחר סיום המשימה, הבדיקה תתבצע בצורה פרונטאלית עבור כל תלמיד בנפרד. יש להעלות את המשימה לאתר הקורס (מגה סקול) בלשונית מטלות.
3. חומר עזר: אינטרנט.
4. **חל איסור מוחלט לדבר ולהיעזר בחברכם לכיתה או כל "חבר טלפוני".**
5. **חל איסור מוחלט ליצור קשר דרך האינטרנט בין חבריכם לכיתה או מול כל גורם אחר.**
6. המשימה מהווה מבחן!
7. התייחסו למשימה כחלק מראיון עבודה.
8. זכרו, אין רק מבחן התוצאה משקף את היכולת והציון שלכם אלא גם הדרך, צורת הכתיבה, best practice ודר וארגון בקוד.



**משימה:**

במשימה זו עליכם לבנות אפליקציה (SPA) לניהול פעולות של חשבונות בנק. כל פעולה המתבצעת על חשבון בנק תופיע כ – "מסמך" document ב "אוסף", collection ,

סוגי הפעולות האפשריות:

1. הפקדה
2. משיכה
3. לקיחת הלוואה

מבנה האפליקציה:

צד לקוח: Angular

צד שרת: Node

בסיס נתונים: MongoDB

1. בנו בסיס נתונים המכיל את סכמה בשם: BankAccounts ו אוסף בשם AccountOperations
    - א. אוסף – AccountOperations יהיה בנוי באופן הבא:
      - a. objectId מזהה
      - b. accountNumber מספר חשבון
      - c. type סוג פעולה (משיכה, הפקדה, הלוואה)
      - d. שאר השדות יהיו בהתאם:
        - i. משיכה – סכום, תאריך
        - ii. הפקדה – סכום, תאריך
        - iii. הלוואה – סכום, ריבית, כמות תשלומים, תאריך מתן הלוואה
  2. בנו API ע"י שימוש ב NodeJS Express , API ה יחשוף את המידע הבא:
    - א. כלל הפעולות לחשבון בנק – GET מחזיר את כל הפעולות של חשבון ספציפי
    - ב. הוספת פעולה – POST מוסיף פעולה ( חשבון בנק, שדות הפעולה הרלוונטית)
  3. בנו אפליקציית צד-לקוח – Angular שתשמש לביצוע הפעולות וצפייה בנתונים.
  4. הרכיבו את האפליקציה – צד לקוח קומפוננטות
    - א. הצגת נתונים - הדף הראשי (default) תציג input + button כאשר בהכנסת חשבון בנק מתאים, יופיעו כל הפעולות שבוצעו בחשבון הרלוונטי ( עדיפות תצוגה לכרטיסיות)
    - ב. ביצוע פעולות על חשבון – route נוסף – בדף זה ניתן לבצע כל אחד משלושת הפעולות המצוינות (משיכה, הפקדה, הלוואה)
      - הקומפוננטה תכיל את הרכיבים הבאים:
        - i. Input למספר חשבון
        - ii. DropDownList בחירת סוג הפעולה
        - iii. UI דינמי המציג את השדות הרלוונטיים:
          1. משיכה – input סכום, כפתור שמירה
          2. הפקדה – input סכום, כפתור שמירה
          3. הלוואה – input סכום, input תשלומים, input ריבית, כפתור שמירה
- שימו לב שה UI הדינמי מתבטא רק בתוספת של 2 שדות במקרה של בחירת הלוואה על פני משיכה או הפקדה.  
דוגמא למבנה הנשלח לשרת במקרה של שמירת פעולת הפקדה:
- ```
{ accountNumber:"889273", type:"deposit", amount: 100 }
```

#### טיפים:



1. התחילו מ- DB.
2. התקדמו לAPI, חברו אותו לDB וחשפו את הבקשות, בדקו זאת בעזרת PostMan\RestAdvanced.
3. לבסוף כתבו את ה UI.
4. צד לקוח - חשוב לכתוב קודם את הקומפוננטות routes + מימוש של router במודול הראשי.
5. לאחר מכן כתבו Service וממשו בו את הבקשות הרלוונטיות
6. שמרו את החשבון בדף יצירת פעולות ב service

**בהצלחה!!**