

# משימה למועמד - פיתוח מערכת מטלות

יש לייצר רשימת מטלות המוצגות בטבלה. הגישה לרשימה תהיה מוגנת בסיסמה, והרשימה תהיה ניתנת לעריכה ע"פ הדרישות להלן:

- המטלות ותיאוריהן יישמרו בדאטה בייס מסוג MongoDB או mySQL או Postgres.
- הקוד של צד השרת ייכתב ב-Express או NestJS והפנייה אליו תיעשה באמצעות TypeORM (עבור MongoDB) או Sequelize (עבור mySQL/Postgres). צד השרת יממש REST API.
- הקוד ייכתב ב-TypeScript ולכל מתודה יהיה מוגדר היטב הטיפוס המוחזר על ידה.
- קוד צד הפרונט ייכתב ב-React בצורת **Functions**, בשימוש hooks כגון: `useState`, `useEffect`.
- לכל מטלה שתישמר ניתן יהיה לייצר יותר מתיאור יחיד, כך שתיווצר "היסטוריית תיאורים" פר מטלה.
- המטלות יוצגו ברשימה בסדר יורד על פי סדר הכנסתן (כלומר המטלה שהוכנס אחרונה תוצג ראשונה). הרשימה תציג 10 מטלות אחרונות ותכיל דפדוף (כלומר אם יש 70 מטלות אזי נציג 7 דפי דפדוף).
- ליד כל מטלה ברשימה יופיע כפתור "ערוך" המאפשר עריכה שם שם המטלה.
- לחיצה על שם המטלה עצמה יפתח Modal Popup המציג את רשימת התיאורים ("היסטוריית התיאורים") של אותה מטלה. הפופ-אפ יכיל גלילה, ככל שרשימת התיאורים תהיה ארוכה מאורך המסך.
- ליד כל תיאור בפופ התיאורים, יופיע משמאל סמל "פח-אשפה" המאפשר למחוק את התיאור מהמטלה.
- מעל רשימת התיאורים יופיע סמל "פלוס" המאפשר להוסיף תיאור שישוויך למטלה הנוכחית (אותו תיאור "יתווסף להיסטוריה" של המטלה), ובפינה למעלה יופיע סמל "X" המאפשר לסגור את הפופ-אפ.
- הגישה לטבלה / לעריכה תהיה מוגנת בשם וסיסמה **qboעים**: משתמש `lstech` וסיסמה `LStech123`.
- כאשר מישוה מזין את השם והסיסמה אזי הוא מורשה לגשת למשך שעתיים. כעבור שעתיים הוא נדרש להקלידם מחדש. יש לממש את המנגנון באמצעות `jwt-token` (או כל טוקן אחר המוגבל בזמן) שיועבר לצד ה-API בשיטת Bearer ב-Headers ויכיל טוקן.
- בסיום העבודה, על המועמד להעלות את קוד המקור (צד פרונט וצד בק) אל Github או Bitbucket ולתת לנו הרשאה על מנת שנוכל לקרוא את הקוד שנכתב.

זמן המשימה המקסימלי המוקצב למועמד הינו 5 שעות

