פרוייקט מועדוני צלילה – Angualr + NodeJs

טכניקות ושפות:

1. מסד נתונים – MongoDb(Atlas)
2. צד שרת – NodeJs(Heroku):

* Express.Js
* Mongoose – להממשקות עם הMongoDb
* Joi – ולידצייה להרשמה והתחברות.
* Lodash – Pick לבחירת מאפיינים שיוצגו לאחר הרשמה
* Jwt – ליצירת טוקן המשמש לאוטנטיקציה.
* Bcrypt – להצפנת הסיסמא.
* Axios – בקשת http לשרת הslack.
* Slack – לקבלת התראות בתהליכי האתר + אישור מועדון צלילה.

1. צד לקוח – Angular(Netlify):

* שימוש בservices לבקשות http, לקומפוננטות משתמש וקומפוננטות מועדונים.
* שימוש בMaterial,bootstrap,fx-layout,css לעיצוב.
* Localstorage – לשמירת הטוקן והתאמת התצוגה למשתמש מחובר ואורח.
* חיפוש, סדר ופג'יניישן ע"י בקשות מתאימות לצד שרת ושליפת נתונים נכונה מהצד שרת.
* Toastr – לניהול לעיצוב התראות alert וחוויית משתמש נוחה יותר.
* האתר רספונסיבי בצורה מלאה גם לdesktop וגם לmobile.

קשיים ואתגרים:

1. התעסקות עם אוטנטיקציה שנבנתה בצד שרת ומימוש בצד לקוח, הבנה של שליחת http headers ושמירת הtoken בlocalstorage + בדיקת תוקף בכל בקשה הדורשת token.
2. טיפול בשגיאות המגיעות מצד שרת בצורה נכונה בין אם זה לתפוס את השגיאה עצמה ובין אם להתאים את השגיאה שמוצגת ללקוח על פי הerror שמגיע מצד שרת.
3. Pagination מלא שנעשה בצד שרת ומומש בצד לקוח על ידי החזרת אורך המערך של התוצאות המתקבלות מהחיפוש או הבקשה ובניית מערך חדש שיציג את מספר העמודים לאותו חיפוש או בקשה.

סיכום:

* חשוב לציין שזה פרוייקט ראשון שלי המשלב בין צד שרת לצד לקוח ברמה מלאה ששתיהם נבנו על ידי.
* למדתי לבד המון מהשיטות שעבדתי איתן והרווחתי מזה שהצלחתי להבין המון על ההבדים בין המניפולציות והשליפות נתונים ובכללי על עבודה של צד שרת מול צד לקוח.
* כל הפרוייקט נבנה תוך כדי עבודה עם git + github.
* הפרוייקט הועלה על ידי שימוש בשירותים חינמיים:
* MongoDb 🡪 Atlas
* Node.Js 🡪 Heroku
* Angular 🡪 Netlify
* לסיכום, עם כל התקלות, הקשיים והאתגרים שעברתי באופן פרטני אני חושב שלמדתי בעיקר איך להתמודד כשמגיע אתגר, מה לחפש, איך לעבוד ולהסתדר לבד.