בס"ד

**CODE REVIEW:**

**הקבצים שנבחרו: Home, SignUpScreen מפרויקט FamilyNest.**

**קישור לגיטהאב:** [**https://github.com/11sdz/FamilyNest/blob/master/src/screens/Home.jsx**](https://github.com/11sdz/FamilyNest/blob/master/src/screens/Home.jsx)

**https://github.com/11sdz/FamilyNest/blob/master/src/screens/SignUpScreen.jsx**

על מנת לייעל את הקוד נוכל לבצע את הדברים הבאים:

1. אחריות ברכיבים: יש רכיבים שמערבבים בין מה שהם מציגים לבין הלוגיקה שהם עושים (לדוגמה SignUpButtonComponent). עדיף להפריד – רכיב אחד ידאג לתצוגה ורכיב אחר יתעסק עם הלוגיקה. ככה גם יותר ברור מה כל רכיב עושה וגם קל לתקן דברים בעתיד.

2. פיצול רכיבים ליחידות קטנות יותר: למשל בקובץ של SignUpScreen, יש המון דברים שקורים בתוך הרכיב המרכזי. עדיף לפצל את זה – רכיב אחד שינהל את כל הטופס, רכיב אחר לפרטים האישיים, ורכיב נוסף לניהול הסיסמה. הרבה יותר נקי ומסודר ככה.

3. יצירת חוויית משתמש חלקה יותר: כשיש פעולות כמו הרשמה או טעינת נתונים, שווה להראות למשתמש מה קורה – לדוגמה, להוסיף סמל טעינה או הודעה קטנה שמראה שהפעולה בתהליך. זה גם נראה יותר מקצועי וגם מונע בלבול.

4. שיפור ניהול נתונים באמצעות מערכי נתונים מורכבים: למשימות או פרופילים עם הרבה פרטים, שווה לעבוד עם מבנה נתונים מתוחכם יותר, כמו Map. זה יעשה סדר, ימנע כפילויות, וגם יגרום לדברים לעבוד יותר מהר.

5. התאמה לכל המסכים: במקום לעבוד עם גדלים קבועים בעיצוב, עדיף להשתמש באחוזים או ביחידות גמישות. ככה האפליקציה תיראה טוב בכל מסך, גדול או קטן.

6. לוגיקה ותצוגה: לוגיקות כבדות כמו פונקציות של הרשמה או סינון משימות עדיף להוציא לקבצים נפרדים. זה לא רק מסדר את הקוד, אלא גם מקל על בדיקות ושימוש חוזר באותן פונקציות.

7. מראה שמתאים גם לאורך וגם לרוחב: האפליקציה צריכה להיראות טוב גם כשהמסך מסתובב. כדאי להשתמש ב-useWindowDimensions כדי לוודא שהעיצוב מתאים לכל מצב.

8. עדיפות למשימות חשובות: במסך "Home" במשימות כרגע אין התחשבות במה יותר דחוף. כדאי להוסיף מאפיין של "עדיפות" ולתת למשימות הכי חשובות להופיע ראשונות.

9. מעקב אחרי בעיות ושגיאות: אין שום מערכת שעוקבת אחרי שגיאות. שווה להוסיף משהו כמו Sentry, שיעזור להבין איפה נפלו דברים ולפתור אותם בקלות.