

# :CODE REVIEW

הקובץ שנבחר: REWARDS מפרויקט Family Nest.

הנה כמה הצעות לשיפור הקוד שלך:

קישור לגיטהאב:

<https://github.com/11sdz/FamilyNest/blob/master/src/screens/RewardsScreen.jsx#L45>

מספר קומיט:

יש כמה דברים שהייתי עושה כדי לייעל את הקוד:

1. **קריאות ומבנה קוד:** השימוש ב-useEffect מצוין, עם זאת כדאי לשפר את קריאות הקוד על ידי הוספה של תיאורים מדויקים יותר והתחום של כל פונקציה יהיה עבור מטרה אחת בלבד. החלקים שקשורים לניהול מודלים (כמו רכישת תגמולים או יעד), די גדולים ומסובכים, כדאי לחלק אותם לפונקציות קטנות יותר כדי להקל על קריאת הקוד.
2. **עבודה עם redux ונתונים:** בשורה הזו

```
;(const user = useSelector((state) => state.user.user
```

כדאי להוסיף אימות שהמשתמש לא ריק לפני שמבצעים עליו פעולות. בנוסף, הdispatch בסוף כל עדכון נתונים מעולה, אך כדאי לוודא שניתן לעדכן רק את המידע החיוני ולא בהכרח את כל המידע מחדש. בשורת uploadUserData בקוד, יש לוודא שהפונקציה מקבלת את הנתונים הנכונים ולא גורמת לעדכון יתר או כפילות.

3. **ניהול אנימציות ותגובות:** השימוש ב-LottieView מצוין להוספת אנימציות. כדאי לוודא שכל האנימציות פועלות באופן אחיד לפי הצורך (למשל, שלא יהיה מצב שבו אנימציה לא מתחילה או לא מסתיימת).

השימוש ב-useRef עבור האנימציות (כמו animationRewardRef) מצוין, אך חשוב להוסיף תנאים שמוודאים שהרפרנס קיים לפני הפעלת האנימציה.

4. **שימוש ב-Toast:** השימוש ב-Toast מוצג במקום נכון, אך כדאי לוודא שנפלטים הודעות שגיאה בצורה ברורה כמו: "לא מספיק נקודות".
5. **תחום הבעיה והפונקציות:** כדאי לבדוק את הפעולות שמבצע כל useEffect, במיוחד כשיש הרבה משתנים תלויים. לדוגמה בשורה הבאה:

```
} <= ())useEffect
```

```
;(("setFilteredRewards(rewardsList.filter((item) => item.status === "ACTIVE
```

```
},{[rewardsList]},{
```

ברגע ששורה זו מתעדכנת, כל פעם ש-rewardsList משתנה תופיע התוצאה החדשה ב-filteredRewards, אבל אם יש הרבה פעולות עם תלות הדדית כדאי לבחון אם ניתן לאחד חלק מהן.

6. **ניהול יעדים:** חלק מהקוד עוסק בעדכון יעדים עבור המשתמש. חשוב לוודא שאין אפשרות לעדכון כפול של יעד לא פעיל או יעד שנקבע כבר.
- יש לשפר את הלוגיקה של חישוב המשימות שהושלמו, במיוחד בנוגע להמרת תאריכים בצורה נכונה:
- ```
;(const deadlineDate = new Date(user.target?.deadline
```
7. **שיפור קריאות הקוד:** חשוב לשפר את שמות המשתנים כך שהם יתארו טוב יותר את מטרת השימוש שלהם.
8. **צמצום פניות לרשת:** במקום לעדכן את המידע בצורה חוזרת ונשנית אחרי כל שינוי (כמו בעדכון תגמול או יעד), כדאי לשקול לאחד את קריאות הרשת כך שניתן יהיה לשלוח את הנתונים החדשים בצורה מרוכזת, ולא לבצע עדכון אחרי כל שינוי קטן.