משימה: אפליקציית Todo עם Context ו-LocalStorage

מבוא:

במשימה הקודמת, בניתם אפליקציית Todo בסיסית עם React, בה השתמשתם ב- `useState` ו- `useEffect` לניהול המשימות, ושמרתם את המידע ב- `localStorage` .

במשימה זו, נעמיק את ההבנה ב-**Context API** של React ונעביר את ניהול הסטייט של המשימות (todos), כולל העדכונים ל- `localStorage` , ל-Context.

דרישות המשימה:

1. **ניהול סטייט עם Context:**
   - צרו Context חדש לניהול המשימות (לדוגמה: `TodosContext`).
   - כל הפונקציונליות הקשורה לניהול המשימות (הוספה, עדכון, מחיקה) תועבר ל-Context.
   - ה-Context יכיל את הסטייט של המשימות וישתמש ב- `localStorage` כדי לשמור ולהביא את הנתונים.
2. **עדכון הקומפוננטות:**
   - עדכנו את הקומפוננטות הקיימות כך שישתמשו ב-Context במקום ב- `useState` .
   - השתמשו ב- `useContext` כדי לגשת לסטייט ולפונקציות מה-Context.
3. **שימוש ב-localStorage בתוך ה-Context:**
   - בתוך ה-Context, השתמשו ב- `useEffect` כדי לטעון את המשימות מ- `localStorage` כאשר האפליקציה נטענת.
   - עדכנו את ה- `localStorage` בכל פעם שהמשימות משתנות.
4. **פונקציונליות האפליקציה:**
   - **הוספת משימה:** המשתמש יכול להוסיף משימה חדשה לרשימה.
   - **שינוי מצב המשימה:** לחיצה על המשימה או על האייקון משנה את מצבה מ"לא הושלמה" ל"הושלמה" ולהפך.
   - **מחיקת משימה:** לחיצה על כפתור המחיקה מסירה את המשימה מהרשימה.
   - **שימוש בישויות (Entities):** הציגו אייקונים של "צ'ק" או "קרוס" ליד המשימות, בהתאם למצבם.
5. **שמירה על המידע לאחר רענון הדף:**
   - ודאו שהמשימות נשמרות ונטענות מ- `localStorage` , כך שהמידע לא יאבד אם המשתמש יעזוב או ירענן את הדף.

בונוס: שימוש ב-MockAPI במקום ב-localStorage

למעוניינים לאתגר את עצמם, ניתן במקום להשתמש ב- `localStorage` , להתחבר ל-MockAPI ולשמור את הנתונים שם.

- **שלבים:**
  - **יצירת חשבון ו-Endpoint:**
    - צרו חשבון ב-MockAPI וצרו Endpoint חדש עבור המשימות.
  - **שימוש ב-Fetch או Axios:**
    - השתמשו ב- `fetch` או ב-Axios כדי לבצע בקשות **GET**, **POST**, **PUT** ו-**DELETE** ל-API.
  - **עדכון ה-Context:**
    - עדכנו את ה-Context כך שישתמש ב-API במקום ב- `localStorage` לניהול המשימות.

הערות נוספות:

- **שימוש ב-JSON.stringify וב-JSON.parse:**
  - בעת שמירה ושליפה של הנתונים מ- `localStorage` , השתמשו ב- `JSON.stringify` ו- `JSON.parse` .
- **ארגון הקוד:**
  - שמרו על מבנה קוד נקי ומסודר.
  - הפרידו לקומפוננטות לפי הצורך, כגון קומפוננטת טופס להוספת משימות, קומפוננטת רשימת משימות וכו'.
- **שימוש ב-Hooks:**
  - השתמשו ב- `useEffect` כדי לסנכרן בין הסטייט ל- `localStorage` או ל-API.
  - זכרו שב-Context, ניתן להעביר לא רק את הסטייט עצמו, אלא גם פונקציות שמשנות אותו.
- **ניהול מצבים אסינכרוניים:**
  - אם אתם משתמשים ב-API, זכרו שהבקשות הן אסינכרוניות, וטפלו בהן בהתאם עם `async/await` או `then/catch` .

טיפ:

- **שימוש ב-Context** מאפשר לכם לגשת לסטייט ולפונקציות שמשנות אותו מכל קומפוננטה באפליקציה, מבלי להעביר props בצורה ידנית.

דוגמא לאפליקציה

סרטון הדגמה

בהצלחה!