בס"ד

הי שלום, טוב זה פרויקט שמדמה ניהול מוצרים בחנות, משהו די בסיסי, עשיתי את זה בטכנולוגיה של react בקורס האחרון שאותו למדתי.

ויש כאן פרויקט נוסף שבעצם מציג נתוני משתמש מהאתר.

1. בפרויקט הזה אני תרגלתי שימוש בredux זה בעצם כמו מרחב זיכרון משותף באפליקציה,

כדי לחסוך קריאות שרת, וחזרתיות בקוד, יוצרים כעין פינה ששם שומרים זיכרון נגיש,

יש קומפוננטות שמאזינות לredux בעזרת useselector הם יתרנדרו בכל פעם בו יהיה שינוי בredux בפרמטר ספציפי אותו הגדירו, אפשר גם להגדיר את הכל...., כדי בעצם להיות עדכניים,

ויש קומפוננטות שכותבות לredux בעזרת dispatch הן בעצם רק כותבות לו וברגע שנכתב לו משהו, הוא מודיע לכל המאזינים שלו להתעדכן.

זה בגדול, עכשיו נפרט.

נקפוץ רגע לקובץ הmain הקובץ הראשי של המערכת, יש שם כמה הגדרות חשובות.

שם אנחנו יוצרות מופע של הredux נקרא בעצם store שזו חנות הזיכרון שלנו. הstore שווה לappStore שזה הפונקציה הראשית של redux ונלך לראות שם מה היא בעצם מבצעת.

הפונקציה מקבלת שתי פרמטרים:

* פרמטר האתחול של redux נגיד במקרה שלנו זה מערך ריק, הפעולה לביצוע בזמן הקריאה לredux מורכבת מ- מחרוזת של שם הפעולה, ופרמטרים שרוצים לשלוח, בכל פעם בו אני ירצה לכתוב לredux אני אשלח את האובייקט הזה עם פרמטרים שונים.

נוכל לראות בפנים את הפעולות השונות שניתן לבצע על הredux הזה.

הוספת מוצר, מחיקת מוצר, ובבררת מחדל החזרת כל המוצרים הקיימים.

אני חוזרת לדף הבית הראשי של הפרויקט,

יש כאן שלושה קומפוננטות ואני קודם כל אפרט בקצרה מה כל אחת מהם עושה:

* Products- מציג את כל המוצרים הקיימים,
* Totalprice- מציג את המחיר הסופי של כל המוצרים,
* Addproduct- טופס להוספת מוצר חדש,

ונפרט בהתאמה:

* Products- בתחילת העמוד נוכל לראות כעין הצהרה, בעצם כאן הקומפוננטה נרשמת להאזנה לredux, בהחזרה נוכל לראות תנאי שבודק אם אין לי מוצרים במערך המוצרים (האורך שלו קטן מ0 ) הוא מציג לי הודעת למשתמש, אחרת הוא שולח אותי לקומפוננטת product עם הid של המוצר.
* Product- בעמוד נוכל לראות כמו בproducts האזנה לredux ובנוסף גם להרשם לרשום לredux – כיוון שרציתי לתת אופציה למחוק את המוצר ושביל זה אני אצטרך למחוק את זה מהredux.

לפני הכל אני מבצעת פלטור על כל מערך המוצרים לפי הid כדי שיהיה לי את המוצר הנוכחי, אותו אני מציגה ומוסיפה לו כפתור שבעת לחיצה עליו יתבצע מחיקה של המוצר- קריאה לredux עם שם פעולה- REMOVE ופרמטר id, הכל כבר יתבצע בתוך הappreducer.

* Totalprice- מאזין לredux, באם המערך ריק הוא מציג 0, אחרת הוא סוכם את כל המערך, כמובן אם לדוג' ימחק מוצר הקומפוננטה תתרנדר שוב והסכום יחושב מחדש.
* Addproduct- הוספת מוצר-

כאן ביצעתי כמו טריק קטן ליצור id ושהם לא יחזרו על עצמם, זה די פשטני ולא הכי יעיל, אבל זה טוב לפחות לבסיס, השתמשתי לזה בתכונות הבסיסיות של react שימוש בusestate- שימוש בפונקציית עזר זו מבטיח שהמשתנה מוגדר פעם אחת בתחילת הריצה על האפליקציה, לכן יצרתי משתנה id ובכל פעם שיצרתי מוצר לקחתי את הid הקודם והוספתי לו מספר, כך אני שומרת על הייחודיות של המזהה.

מציג את המוצר, יחד עם כפתור של הוספה, בפונקצית ההוספה יש קריאה לredux, ואיפוס של הטופס.

1. בדף הראשי homepage ניתן לראות שימוש בספריית router-dom, זו ספרייה בreact כדי לבצע ניווטים באתר,

כך שבעצם הURL למעלה ישתנה, המשתמש כביכול יעבור עמוד, אבל בפועל לפי המתודולוגיה של react הדף הראשי של האפליקציה לא יתרנדר שוב ושוב על כל מעבר קומפוננטה.

אפשר לראות שיש כאן מערך המכיל ניתובים, ולכל ניתוב את התתי ניתובים שלו.

בעבור כל ניתוב הוא מקפיץ עבורי קומפוננטה אחרת,

עכשיו נעבור על הקומפוננטות לפי הסדר,

בקצרה-

* Names- מציג את כל רשימת השמות מהאתר באופן כללי,
* Name- דואג לתצוגה של כל שם ושם בנפרד,

בעצם קומפוננטה names היא תקרא לפי כמות השמות הקיימים לname כל פעם.

* Posts- מציגה את הפוסטים הייחודים לאותו שם.
* Todos- מציגה את המשימות הייחודיות לאותו שם.

ונפרט-

* Names- למעלה רואים שימוש בכמה ספריות שונות בreact.

Usestate- יצירת משתנים.

Useeffect- הפעלת פונקציה באופן יזום ולא כל פעם שמתרנדרת הקומפוננטה.

Link- ליצירת קישורים

Axios- ספריה בreact המבצעת קריאות שרת.

בוא נקפוץ למה הקומפוננטה בעצם מחזירה לי-

מערך של השמות-

בעת טעינת העמוד- ([] מערך ריק בuseeffect) מתבצעת קריאת שרת להביא את כל שמות האנשים, תוכל לראות שאנחנו שולפות בעצם את כל הנתונים אבל בפועל מציגות רק את השמות של כל אדם,

את השם של האדם אנחנו מציגות בצורה של קישור- (אם אנחנו רוצות לראות לאיזה קומפוננטה הוא מוביל, ניגש לעמוד הבית ונסתכל במערך routes)

הקישור הזה מוביל אותנו לעמוד name, יחד עם הid היחודי של כל שם, כדי שנוכל לטפל בו,

הid כרגע שמור לנו בתוך הURL נצטרך לשלוף אותו משם.

נעבור הלאה

* Name- נוספו לנו כאן עוד שני פונקציות מערכת,

Useparams- משתמש לשליפת מידע מכתובת URL

Outlet- להצגת תוכן קומפוננטה מוכל בתוך הקומפוננטה הראשית.

הפונקציה מציגה נתוני הבן אדם ושלושה קישורים, לפוסטים, משימות, ודף הבית.

בעת טעינת העמוד, מתבצעת קריאת שרת להביא את תוכן האיש לפי id,

הקומפוננטות הבאות עושות בעצם אותו דבר-

* Posts, todos- מבצעות קריאות שרת להביא את כל הנתונים, הפונקציה מפלטרת לפי הid של המשתמש ומציגה על המסך.