

מטלה 4: ניווט באמצעות react-router

הנחיות כלליות

במטלה זו נבנה אתר מידע על דמויות ממלחמת הכוכבים, הכולל דף בית, דף אודות, רשימת דמויות, דף פרטים דינמי לכל דמות, וטיפול בכתובות שגויות.

הנחיות הגשה

- **אופן העבודה:** המטלה תבוצע בזוגות או בקבוצות הפרויקט שלכם.
- **אופן ההגשה:** יש לאגד את כל קבצי הקוד הרלוונטיים לפתרון המטלה לקובץ zip אחד, למעט תיקיית node_modules.
- **העלאה:**
 - יש להעלות את קובץ ה-zip למערכת הגשת המטלות. (רק אחד/אחת מחברי הצוות צריך/ה להגיש את הקובץ).
 - יש לכתוב בתיבת ההגשה קישור לגיטהאב של המטלה.

שלב 1: הכנת דפים וקומפוננטת Layout

לפני שניצור את הראטר, נכין את התשתית. בגישה החדשה, כדי שהתפריט יופיע בכל העמודים, עלינו ליצור קומפוננטה "עוטפת" (Layout).

1. **יצירת דפים בסיסיים:** צרו שתי קומפוננטות פשוטות (קבצים חדשים):

- Home.jsx: תכיל כותרת "ברוכים הבאים לאתר מלחמת הכוכבים" ופסקה קצרה.
- About.jsx: תכיל הסבר קצר על הפרויקט.

2. **יצירת קומפוננטת Layout:**

- צרו קובץ חדש בשם Layout.jsx.
- קומפוננטה זו תשמש כ"תבנית" לאתר.
- היא צריכה להחזיר div שבתוכו יופיעו שני דברים:

1. קומפוננטת ה-Navbar (שניצור בשלב הבא - כרגע אפשר לשים סתם

כותרת או להשאיר מקום).

2. קומפוננטת <Outlet/> (יש לייבא אותה מ-react-router). ה-Outlet הוא

המקום שבו יתחלף התוכן של העמודים (Home, About וכו') בהתאם לכתובת.

שלב 2: תפריט ניווט

צרו קומפוננטה בשם Navbar.jsx.

1. הקומפוננטה תכיל קישורים לדפים: "בית" (/), "אודות" (/about), ו"דמויות"

(/characters - שניצור בהמשך).

2. **חובה:** השתמשו בקומפוננטת <NavLink> ולא ב-<Link> רגיל.

3. הגדירו עיצוב כך שהקישור הפעיל יהיה מודגש. השתמשו ב-attribute style שמקבל

פונקציה:

`style={({ isActive }) => ({ color: isActive ? 'red' : 'black' })}`

4. **חיבור:** ייבאו את Navbar לתוך קובץ ה-Layout.jsx שיצרתם בשלב 1, ומקמו אותה מעל

ה-<Outlet/>.

שאלה 3: הגדרת הראוטר (router.jsx)

בשלב זה נגדיר את מפת האתר בקובץ חיצוני.

1. צרו קובץ חדש בשם router.jsx בתיקיית src.
2. השתמשו בפונקציה createBrowserRouter (יש לייבא מ-react-router).
3. צרו והמליצו (export) אובייקט router המכיל את הגדרת הנתיבים:
 - הנתיב הראשי (path: "/") יטען את קומפוננטת ה-Layout (בתור element).
 - בתוך הנתיב הראשי, הגדירו מערך children שיכיל את דפי המשנה:
 - נתיב ריק (index: true או path: "") שיוביל ל-Home.
 - נתיב path: "about" שיוביל ל-About.
4. חיבור ל-App.jsx
 - פתחו את קובץ App.jsx.
 - ייבאו את ה-router שיצרתם ואת הקומפוננטה RouterProvider.
 - החליפו את התוכן של App כך שיחזיר רק את:
<RouterProvider router={router}/>

שלב 4: רשימת דמויות ודף פרטים (Dynamic Routing)

כעת נוסיף את הלוגיקה של הדמויות לתוך הראוטר הקיים.

1. רשימת דמויות (Characters.jsx):

- ייבאו את מערך ה-characters מקובץ ה-data.js (מצורף למטלה זו ומופיע בסוף ההנחיות).
- הציגו רשימת שמות (), כאשר כל שם הוא <Link> לכתובת ./characters/[ID].
- הוסיפו את הנתיב לקובץ router.jsx בתוך מערך ה-children: נתיב path: "characters" שמוביל לקומפוננטה זו.

2. דף פרטים (CharacterDetails.jsx):

- צרו קומפוננטה שמציגה פרטי דמות בודדת.
- השתמשו ב-useParams כדי לחלץ את ה-id מהכתובת.

- מצאו את הדמות במערך (שימו לב להמרת המחרוזת למספר).
- בונוס: אם הצד הוא "Dark" הטקסט באדום, ואם "Light" בכחול.
- טיפול בשגיאה: אם ה-ID לא קיים, הציגו "דמות לא נמצאה".
- הוסיפו את הנתיב הדינמי ל-router.jsx בתוך ה-children:
נתיב "characters/:id" path: שמוביל לקומפוננטה זו.

שלב 5: דף 404 (Not Found)

1. צרו קומפוננטת NotFound.jsx עם הודעה מתאימה.
2. הוסיפו נתיב "Catch All" בקובץ router.jsx.
 - זהו אובייקט נוסף בתוך מערך ה-children, בסוף הרשימה.
 - הגדירו לו: path:"*" element-מתאים.

// src/data.js

```
export const characters = [  
  { id: 1, name: 'Luke Skywalker', role: 'Jedi', side: 'Light' },  
  { id: 2, name: 'Darth Vader', role: 'Sith Lord', side: 'Dark' },  
  { id: 3, name: 'Han Solo', role: 'Smuggler', side: 'Light' },  
  { id: 4, name: 'Yoda', role: 'Jedi Master', side: 'Light' },  
  { id: 5, name: 'Palpatine', role: 'Sith Emperor', side: 'Dark' }  
];
```