

מטלה בנושא רינדור מותנה, Local Storage, עבודה עם ה-DOM ועיצוב בסיסי

הוראות:

עליכם לבנות אתר בעל שני עמודים עם התחברות, רינדור דינאמי של כרטיסים, ושימוש ב-Local Storage לשמירת מצב התחברות. האתר יכלול עיצוב רספונסיבי ושינוי בין מצב אור ומצב לילה.

דרישות הפיתוח:

1. עמוד התחברות:

- עמוד זה יכיל טופס עם שני שדות: אימייל וסיסמא.
- שמרו מערך סטטי של משתמשים כקובץ JavaScript:

```
const users = [  
  { email: "user1@example.com", password: "password1" },  
  { email: "user2@example.com", password: "password2" },  
  // הוסיפו עוד משתמשים כרצונכם  
];
```

- במידה והמשתמש מכניס פרטים לא נכונים, יופיע מודאל המודיע על כך למשך 15 שניות ולאחר מכן ייסגר אוטומטית.
- במידה וההתחברות מצליחה, יש לשמור ב-Local Storage מפתח בשם isLoggedIn עם הערך true.

2. ניווט ועיצוב:

- לאחר התחברות מוצלחת, יופיע נאבבר בחלק העליון של האתר עם שני כפתורים:
 - כפתור שמאלי: מחליף בין מצב "אור" למצב "לילה" (שינוי ערכת נושא).
 - כפתור ימני: "התנתקות" – מחזיר לעמוד ההתחברות ומנקה את המפתח isLoggedIn מה-Local Storage.
- עיצוב: על האתר לעבור בין ערכת נושא בהירה לכהה כאשר לוחצים על כפתור הערכת נושא.

3. עמוד הבית:

- מתחת לנאבבר יוצגו **כרטיסים דינאמיים** שמגיעים ממערך אובייקטים JavaScript:

```
const cards = [
  { title: "Card 1",
    description: "This is the first card",
    img: "img1.jpg",
    alt: "Image 1"
  },
  { title: "Card 2",
    description: "This is the second card",
    img: "img2.jpg",
    alt: "Image 2"
  },
  // הוסיפו עוד כרטיסים כרצונכם
];
```

◦ כל כרטיס יכלול:

- כותרת
- תיאור

▪ תמונה ו- alt לתמונה

◦ תצוגת הכרטיסים תשתנה בהתאם לערכת הנושא (חושך/אור).

4. שימוש ב-Local Storage:

◦ אם המפתח isLoggedIn שווה ל-true, יש להציג את דף הבית ואת הנאבבר במקום עמוד ההתחברות.

◦ במידה והמשתמש התנתק, מחזירים אותו לעמוד ההתחברות ומנקים את הנתונים מה-Local Storage.

5. קוד נקי:

◦ יש לשמור על עקרונות קוד נקי, כפי שנלמדו בשיעור:

- שמירת משתנים גלובליים בקונסטים בקוד הנקי
- פיצול הקוד לפונקציות קצרות ומובנות
- שימוש ב-createElement ו-appendChild ליצירת אלמנטים חדשים ולא ב-.innerHTML

הגשה:

1. יש להעלות את כל הקוד לריפוזיטורי ייעודי ב-GitHub ולוודא שיש קובץ index.html בתיקיית השורש.
2. יש ליצור GitHub Pages עבור הריפוזיטורי ולהגיש את קישור האתר.
3. הקפידו על עיצוב ברמה גבוהה, יש להציג את האתר בצורה מסודרת, שימוש ב-Flexbox הוא חובה.

דרישות עיצוב:

- האתר צריך להיראות מקצועי ואסתטי.
- יש להימנע מעיצוב זרוק – חשוב לשמור על רווחים אחידים ועיצוב קריא.
- שימו דגש על תצוגת הכרטיסים במצב כהה ובהיר.

בהצלחה!