

הסבר על הפונקציות הנוספות

1. הוספת כפתור "מצא את חברת התעופה המאחרת ביותר" – לחיצה על הכפתור תמצא את חברת התעופה אשר לה המספר הגדול של טיסות בהן הזמן המתוכנן שונה מהזמן בפועל.

```
2. function findMostDelayedAirline(flights) {
3.     let delaysCount = {};
4.     flights.forEach(flight => {
5.         if (flight.scheduledTime !== flight.actualTime) {
6.             if (!delaysCount[flight.operatorLong]) {
7.                 delaysCount[flight.operatorLong] = 1;
8.             } else {
9.                 delaysCount[flight.operatorLong]++;
10.            }
11.        }
12.    });
13.
14.    let maxDelayCount = 0;
15.    let mostDelayedAirline = '';
16.    for (let airline in delaysCount) {
17.        if (delaysCount[airline] > maxDelayCount) {
18.            maxDelayCount = delaysCount[airline];
19.            mostDelayedAirline = airline;
20.        }
21.    }
22.
23.    if (mostDelayedAirline !== '') {
24.        alert(`The most delayed airline is ${mostDelayedAirline}
25.        with ${maxDelayCount} delayed flights.`);
26.    } else {
27.        alert('No delayed flights found.');
```

```
28.    }
29.    document.getElementById('find-most-
30.    delayed').addEventListener('click', function () {
31.        findMostDelayedAirline(jsonFlights);
    });
```

אנחנו מאתחלים משתנה בשם `delaysCount` כדי לאחסן את ספירת הטיסות המאחרות עבור כל חברת תעופה. זה חוזר על כל טיסה במערך הטיסות. עבור כל טיסה, הוא בודק אם השעה המתוכננת (`scheduledTime`) שונה מהשעה בפועל (`actualTime`). אם יש הבדל, אנגדי ל את ספירת העיכובים עבור חברת התעופה המתאימה ב- `delaysCount`.

מציאת חברת התעופה המאחרת ביותר: לאחר ספירת העיכובים עבור כל חברות התעופה, היא חוזרת דרך המשתנה `delaysCount` כדי למצוא את חברת התעופה עם המספר הגבוה ביותר של טיסות עיכובות אחר השני. הוא מעדכן את המשתנה `mostDelayedAirline` בשם של חברת תעופה זו. ובסוף, הוא מציג הודעת התראה עם שם חברת התעופה המאחרת ביותר ומספר הטיסות המעוכבות שיש לה. אם אין טיסות שמאחרות, הוא נותן הודעה שלא נמצאו טיסות עיכוב.

הצגת גרף התפלגות טיסות – ע"י שימוש בספריה להצגת גרפים (למשל plotly (להציג גרף פאי של סוגי הטיסות) type) ו/או של התפלגות הסטטוסים

```
// Function to create a pie chart
function createPieChart(data, title) {
  const labels = Object.keys(data);
  const values = Object.values(data);

  const layout = {
    title: title,
    showlegend: true
  };

  const trace = {
    type: 'pie',
    labels: labels,
    values: values
  };

  Plotly.newPlot('flight-distribution-chart', [trace], layout);
}

// Call the functions to generate and display pie charts
function displayFlightDistributionCharts(flights) {
  const typeDistribution = generateFlightDistributionData(flights,
'type');
  const statusDistribution = generateFlightDistributionData(flights,
'status');

  createPieChart(typeDistribution, 'Flight Type Distribution');
  createPieChart(statusDistribution, 'Flight Status Distribution');
}

// Call this function after populating the table to display the charts
displayFlightDistributionCharts(jsonFlights);
});
```

פונקציית createPieChart:

מקבלת שני פרמטרים: נתונים אובייקט המכיל זוגות עבור הפאי וכותרת (מחרוזת המייצגת את הכותרת של הפאי).
בהתחלה מחלץ ערכים מאובייקט הנתונים.
מגדיר את הפריסה עבור הפאי, כולל הכותרת, יוצרים אובייקט מעקב עם הסוג מוגדר כ-'pie' ומקצה לו תוויות וערכים.
משתמש בשיטת newPlot של Plotly כדי ליצור את הפאי באלמנט ה-HTML עם 'תרשים-תפוצה-טיסה'.

פונקציית displayFlightDistributionCharts:

מקבלת מערך של נתוני טיסה (טיסות).
תהליך:

קוראת לפונקציה generFlightDistributionData עם פרמטרים שונים ('סוג' ו'סטטוס') כדי ליצור נתונים להפצת סוגי הטיסה והפצת מצב הטיסה.
מעביר את הנתונים שנוצרו יחד עם כותרות מתאימות לפונקציית createPieChart כדי ליצור ולהציג פאי: מצב טיסה. השתמשנו בספריית Plotly.js.