****

****

**שלב 1 - יצירת הDatabase**

לצורך הפרויקט, ניצור את הסכמה הבאה )הערה: הוסף constraint null not עבור השדות הרלוונטיים(

Background pattern

Description automatically generated with low confidenceDiagram

Description automatically generated

ייעוד הטבלאות:

* טבלת **Airlines**  מכילה את רשימת חברות התעופה.

לכל חברת תעופה יש מפתח מזהה (Id), שם (Name), קוד המדינה של החברה (Country\_Id) ומס' משתמש (User\_Id).

לכל חברה במערכת יש רשימה של טיסות אשר היא מפרסמת בכדי שלקוחות יקנו כרטיסים עבור הטיסה (ראה טבלת Flights, וטבלת Tickets)  
לכל חברה יש שם משתמש וסיסמא

* טבלת **Flights**  מכילה את רשימת הטיסות.

בפרטי הטיסה קיימים הפרטים הבאים: חברת התעופה אליה היא שייכת (Airline\_Id), קוד מדינת המקור (Origin\_Country\_Id), קוד מדינת היעד (Destination\_Country\_Id), זמן המראה (Departure\_Time), זמן נחיתה (Landing\_Time) ומספר הכרטיסים שנשארו (Remaining\_Tickets).

* טבלת **Customers** מכילה את רשימת הלקוחות הקיימים במאגר.   
  לכל לקוח יש מפתח מזהה (Id), שם פרטי (First\_Name), שם משפחה (Last\_Name), כתובת (Address), מספר טלפון (Phone\_No) מספר כרטיס אשראי (Credit\_Card\_No)   
  לכל משתמש יש שם משתמש וסיסמא
* טבלת **Tickets** מכילה את הכרטיסים שנרכשו עבור הטיסות.   
  לכל כרטיס יש מפתח מזהה (Id), מספר טיסה (Flight\_Id) ומספר לקוח (Customer\_Id).

שים לב שאותו הלקוח אינו יכול לרכוש פעמיים כרטיס לאותה הטיסה, לכן השילוב של מספר הטיסה ומספר הלקוח הוא Unique.

* טבלת **Countries** מייצגת את כל המדינות הקיימות במערכת.   
  לכל מדינה יש מפתח מזהה (Id) ושם (Name).
* טבלת **Administrators** מכילה את רשימת מנהלי המערכת.   
  לצורך הפשטות נייצר מנהל אחד (עם סיסמא קבועה) ונזין אותו בטבלה

Diagram

Description automatically generated

**שלב 2 - כתיבת הפונקציות המעדכנות את בסיס הנתונים:**

עבור כל טבלה נייצר ב- JS את הפונקציות הבאות:

* החזרת כל הרשומות
* החזרת רשומה לפי ה- ID
* הוספת רשומה
* עדכון רשומה
* מחיקת רשומה
* \*אתגר: עבור טבלת מנהלים/ חברות-תעופה/ לקוחות נחזיר את הרשומה לפי שם\_משתמש

רעיונות לדוגמא:

getAllAirlineCompanies ()

getAirlineById ()

addAirline ()

updateAirline ()

removeAirline ()

\*bonus: getAirlineByUsername ()

getAllCustomers ()

getCustomerById ()

addCustomer ()

Icon

Description automatically generatedupdateCustomer ()

removeCustomer ()

\*bonus: getCustomerByUsername ()

getAllAdmins ()

getAdminById ()

addAdmin ()

updateAdmin ()

removeAadmin ()

\*bonus: getAdminByUsername ()

getAllFlights ()

getFlightById ()

addFlight ()

updateFlight ()

removeFlight ()

\*bonus: getFlightsByAirline ()

getAllTickets ()

getTicketById ()

addTicket ()

updateTicket ()

removeTicket ()

Icon

Description automatically generated\*bonus: getTicketsByUserID ()

\*\*double bonus: getTicketsByUserName ()

getAllCountries ()

getCountryById ()

addCountry ()

updateCountry ()

removeCountry ()

* כל הפונקציות יקבלו את הפרמטרים הדרושים בתוך אובייקט JSON יחיד.
* כל פונקציה שעורכת או מוחקת פריט מה-DB (משתמש, חברה, טיסה וכו') תבדוק לפני כן שהוא קיים. לא ייתכן שננסה למחוק או לשנות טיסה שלא קיימת, לדוגמא.
* באפשרותכם לחשוב על לוגיקה נוספת או חוקים שברצונכם להחיל על הפונקציות
* ניתן להוסיף עוד פונקציות לפי הצורך

**שלב 3 - שרת EXPRESS:**

נשתמש במודול EXPRESS כדי לחסוף את הפונקציות ב- REST API

לצורך הפרויקט ניצור 6 Controllers שונים, אחד עבור כל טבלה

**בניית ה-Controllers:**

1. Airline Controller - עבור חברת תעופה
2. Customer Controller - עבור לקוחות
3. Admin Controller - עבור מנהל המערכת
4. Ticket Controller - עבור מנהל המערכת
5. Country Controller - עבור מנהל המערכת
6. Flight Controller - עבור מנהל המערכת

כל קונטרולר יכיל פונקציות מסוג: ***GET PUT POST DELETE***

הפונקציות ב- controller יפעילו את הפונקציות שכתבנו **בסעיף הקודם**

**לדוגמא עבור AirlineController-**

שים לב שיש לתת נתיב נכון לכל פונקציה + סימון נכון של GET PUT POST DELETE

מומלץ לבדוק כל פונקציה באמצעות ה- POSTMAN

Path: "/airline" GET

will execute- getAllAirlineCompanies ()

Path: "/airline/{ id }" GET

will execute- getAirlineById ()

Path: "/airline/{ id }" POST

will execute- addAirline ()

Path: "/airline/{ id }" PUT

will execute- updateAirline ()

Path: "/airline/{ id }" DELETE

will execute- removeAirline ()

\*bonus: Path: "/airline/byusername/{ username }" GET

will execute- getAirlineByUsername ()

Icon

Description automatically generated

**שלב 4- בניית ה-Front End**

A picture containing aircraft, airplane

Description automatically generated

אנו נבנה את ה-"פנים" של האתר - דפי ה-Client שיציגו למשתמש את המידע הרלוונטי ויאפשרו לו לבצע פעולות שונות.

(התמונות המוצגות הן להמחשה בלבד)  
 **דף הבית**

זהו דף פשוט ביותר, ובו תוצג כרזה של "Welcome" (או כל טקסט אחר) שאותה יראה המשתמש כאשר הוא נכנס לאתר. באפשרותכם (לא חובה) להעניק לדף עיצוב ייחודי ומעניין, אנימציות ב-CSS וכו'

A plane flying over a sign

Description automatically generated with medium confidence

**דף טיסות**

בעמוד זה, יוכל המשתמש לצפות בכל הטיסות.

דוגמא להמחשה

Graphical user interface, table

Description automatically generated

כמובן שיש להשתמש בשירותי ה EXPRESS שכתבת בסעיף הקודם

**דף הרשמה ללקוח**

בדף זה ימלא הלקוח החדש את כל הפרטים הדרושים על מנת להירשם למערכת:

השדות שהלקוח יצטרך למלא הם:

1. שם פרטי
2. שם משפחה
3. כתובת
4. מספר טלפון
5. מספר כרטיס אשראי
6. שם משתמש
7. סיסמא
8. אימייל

הדף יבדוק שהמשתמש מילא את כל הפרטים לפני שהוא יאפשר לו ללחוץ על Submit.

Icon

Description automatically generated

**דף עדכון פרטי טיסות:**

בנה דף המשרת כממשק משתמש לעריכת הטיסות

בדף זה יהיה ניתן לצפות בפרטי טיסה, להוסיף טיסה, לעדכן טיסה ולימחוק טיסה

בחר בממשק לנוחיותך,

ניתן לעשות זאת בכמה חלקים שונים, לדוגמא- מסמך להוספת טיסה, מסמך למחיקה , מסמך לעדכון וכו'

(או בכמה דפים שונים)

או... **\*אתגר-** לעשות הכול בטבלה אחת ...

דוגמא להמחשה (טבלה):

Table

Description automatically generated

**\*אתגר:** מה דעתך לבקש מהמשתמש את שם\_משתמש וסיסמא בטרם הפעולה (כדי לדוגמא לבדוק שאותה החברת תעופה היא שאכן זאת המנסה למחוק את הטיסה השיכת לה), מומלץ sweet alert 😊

**\*אתגר: דף חיפוש טיסות**

Graphical user interface

Description automatically generated

הוסף אפשרות לחפש טיסות

ראשית, יוצגו למשתמש הפרמטרים הבאים לבחירה: מדינת מקור, מדינת יעד ותאריך.

לאחר שהמשתמש יבחר את פרטי הטיסה שהוא רוצה לחפש, הוא ילחץ על כפתור Search והדף יציג לפניו את הטיסות המתאימות לחיפוש שלו.

**\*אתגר: דף משתמש (לקוח)**

ראשית המשתמש יכניס שם\_משתמש וסיסמא ואז...

דף זה יתחלק למספר "אזורים" ובהם הלקוח יוכל לבצע את הפעולות הבאות:

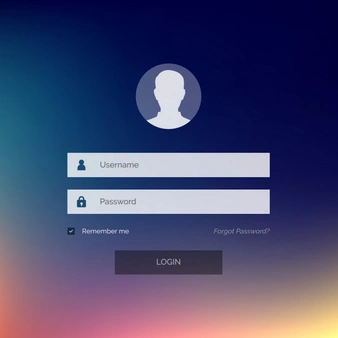
1. צפייה בפרטי הכרטיסים שהוא שרכש:  
   הלקוח יוכל לראות את פרטי הטיסה של הכרטיסים שרכש. הוא יוכל גם לבטל אותם אם ירצה.
2. עריכת הפרטים האישיים שלו:  
   ללקוח יוצג טופס עם שדות כל פרטי המשתמש שלו. הוא יוכל לשנות אותם לפי רצונו.



**\*אתגר דף לוגין (Login)**

בדף זה יוכלו המשתמשים באתר להתחבר באמצעות שם משתמש וסיסמא.

העמוד יהיה יכול להיות בנפרד או במשותף ללקוחות, חברות ומנהלים.



אם הלוגין הצליח, המשתמש יועבר לעמוד הבית.

\*אם יצרתם דפי משתמש (רשות) ניתן לשלוח את הדפדפן אל עמוד המשתמש המתאים כאשר הלוגין מצליח.

\*אם יצרתם דפי הרשמה (רשות) ניתן להוסיף קישורים לדפי ההרשמה עבור משתמשים לא רשומים.

**הערות:**

1. אין חובה לעצב את דפי האתר. ניתן להגיש את הפרויקט עם עיצוב בסיסי בלבד.
2. יש לטפל במקרים אפשריים של שגיאות ולהציג הודעה למשתמש אם משהו השתבש.
3. ניתן להוסיף פונקציות חדשות לצד השרת לפי הצורך.
4. מומלץ (לא חובה) להשתמש ב-Bootstrap.
5. מומלץ לפתוח את הגדרות ה-CORS בשרת.
6. כדאי להוסיף הערות בקוד על מנת לתעד את תפקידן של הפונקציות השונות.