**פרויקט גמר בריאקט – אפליקציית מתכונים**

אפיון מסד נתונים

**טבלת משתמשים - Users:**

קוד – מספרי, מפתח ראשי, אוטומטי

שם משפחה – מחרוזת, חובה

שם פרטי – מחרוזת, חובה

אימייל – מחרוזת, חובה

סיסמה – מחרוזת, חובה

**טבלת קטגוריות - Categories:**

קוד – מספרי, מפתח ראשי, אוטומטי

שם – מחרוזת, חובה

**טבלת רמות - Levels:**

קוד – מספרי, מפתח ראשי, אוטומטי

שם – מחרוזת, חובה

**טבלת רכיבים - Ingredients:**

קוד – מספרי, מפתח ראשי, אוטומטי

שם – מחרוזת, חובה

**טבלת מתכונים - Recipes:**

קוד – מספרי, מפתח ראשי, אוטומטי

שם – מחרוזת, חובה

תמונה – מחרוזת – שם התמונה

הוראות הכנה - מחרוזת

קוד משתמש – מספרי, חובה, מפתח זר לטבלת משתמשים – מייצג את עורך המתכון

קוד קטגוריה – מספרי, חובה, מפתח זר לטבלת קטגוריות

קוד רמה – מספרי, חובה, מפתח זר לטבלת רמות

הערה – מחרוזת, לא חובה, נימה אישית של העורך

**טבלת רכיבים למתכון - IngredientsToRecipes:**

קוד – מספרי, מפתח ראשי, אוטומטי

קוד מתכון – מספרי, חובה, מפתח זר לטבלת מתכונים

קוד רכיב – מספרי, חובה, מפתח זר לטבלת רכיבים

כמות – מחרוזת, לא חובה

**טבלת תגובות למתכון - CommentsToRecipes:**

קוד – מספרי, מפתח ראשי, אוטומטי

קוד מתכון – מספרי, חובה, מפתח זר לטבלת מתכונים

קוד משתמש – מספרי, חובה, מפתח זר לטבלת משתמשים - מייצג את המשתמש שהגיב

תגובה – מחרוזת, לא חובה

אפיון צד שרת – פונקציונאליות

**משתמשים:**

1. הוספת משתמש חדש

סוג קריאת שרת: post

ניתוב: **https://localhost:7130/api/User**

מקבלת משתמש חדש ב body

מחזירה את המשתמש החדש שנוצר

1. שליפת משתמש לפי אימייל וסיסמה

סוג קריאת שרת: get

ניתוב: [**https://localhost:7130/api/User/{email}/{password}**](https://localhost:7130/api/User/%7bemail%7d/%7bpassword%7d)

מקבלת בניתוב אימייל וסיסמה

מחזירה את המשתמש המתאים – אם לא נמצא - x.data=""

אם - (x.data!="")המשתמש נמצא

**קטגוריות:**

1. הוספת קטגוריה

סוג קריאת שרת: post

ניתוב: [**https://localhost:7130/api/Category**](https://localhost:7130/api/Category)

מקבלת קטגוריה חדשה ב body

מחזירה את רשימת הקטגוריות המעודכנת

1. שליפת כל הקטגוריות

סוג קריאת שרת: get

ניתוב:  **https://localhost:7130/api/Category**

אינה מקבלת פרמטרים

מחזירה את רשימת הקטגוריות

**רמות:**

1. הוספת רמה

סוג קריאת שרת: post

ניתוב: [**https://localhost:7130/api/Level**](https://localhost:7130/api/Level)

מקבלת רמה חדשה ב body

מחזירה את רשימת הרמות המעודכנת

1. שליפת כל הרמות

סוג קריאת שרת: get

ניתוב:  **https://localhost:7130/api/Level**

אינה מקבלת פרמטרים

מחזירה את רשימת הרמות

**רכיבים:**

1. הוספת רכיב

סוג קריאת שרת: post

ניתוב:

**<https://localhost:7130/api/Ingredient>**

מקבלת רכיב חדש ב body

מחזירה את רשימת הרכיבים המעודכנת

1. שליפת כל הרכיבים

סוג קריאת שרת: get

ניתוב:  **https://localhost:7130/api/Ingredient**

אינה מקבלת פרמטרים

מחזירה את רשימת הרכיבים

**מתכונים:**

1. הוספת מתכון

סוג קריאת שרת: post

ניתוב: [**https://localhost:7130/api/Recipe**](https://localhost:7130/api/Recipe)

מקבלת מתכון חדש ב body

מחזירה את קוד המתכון שנוסף

1. שליפת כל המתכונים

סוג קריאת שרת: get

ניתוב:  **https://localhost:7130/api/Recipe**

אינה מקבלת פרמטרים

מחזירה את רשימת המתכונים

**רכיבים למתכון: - אין צורך להשתמש בפונקציות אלו!**

1. הוספת רכיבים למתכון

סוג קריאת שרת: post

ניתוב: **https://localhost:7130/api/IngredientsToRecipe**

מקבלת רשימת רכיבים למתכון ב body

מחזירה true / false

1. שליפת רכיבים לפי קוד מתכון

סוג קריאת שרת: get

ניתוב:  **https://localhost:7130/api/ IngredientsToRecipe/{recipeId}**

מקבלת קוד מתכון בניתוב

מחזירה את רשימת הרכיבים לאותו מתכון

**תגובות למתכון:**

1. הוספת תגובה למתכון

סוג קריאת שרת: post

ניתוב: [**https://localhost:7130/api/CommentsToRecipe**](https://localhost:7130/api/CommentsToRecipe)

מקבלת תגובה חדשה ב body

מחזירה true / false

1. שליפת תגובות לפי קוד מתכון

סוג קריאת שרת: get

ניתוב:  **https://localhost:7130/api/CommentsToRecipe/{recipeId}**

מקבלת קוד מתכון בניתוב

מחזירה את רשימת התגובות לאותו מתכון

אפיון מחלקות לצד לקוח:

**מחלקות User, Category, Level, Ingredient** – ללא שינוי

**מחלקת IngredientsToRecipe:**

קוד – מספרי, מפתח ראשי, אוטומטי

קוד מתכון – מספרי, חובה, מפתח זר לטבלת מתכונים

קוד רכיב – מספרי, חובה, מפתח זר לטבלת רכיבים

* **IngredientName - שם רכיב** – מחרוזת, כברירת מחדל null, מתמלא אוטומטית לפי קוד הרכיב

כמות – מחרוזת, לא חובה

**מחלקת Recipe:**

קוד – מספרי, מפתח ראשי, אוטומטי

שם – מחרוזת, חובה

תמונה – מחרוזת

הוראות הכנה - מחרוזת

קוד משתמש – מספרי, חובה, מפתח זר לטבלת משתמשים – מייצג את עורך המתכון

* **UserName - שם משתמש** – מחרוזת, כברירת מחדל null, מתמלא אוטומטית בשם פרטי ושם משפחה של העורך

קוד קטגוריה – מספרי, חובה, מפתח זר לטבלת קטגוריות

* **CategoryName - שם קטגוריה** – מחרוזת, כברירת מחדל null, מתמלא אוטומטית לפי קוד הקטגוריה

קוד רמה – מספרי, חובה, מפתח זר לטבלת רמות

* **LevelName - שם רמה** – מחרוזת, כברירת מחדל null, מתמלא אוטומטית לפי קוד הרמה

הערה – מחרוזת, לא חובה, נימה אישית של העורך

* **IngredientsToRecipe -רכיבים למתכון** – מערך מסוג רכיב למתכון, כברירת מחדל = מערך ריק, מתמלא אוטומטית ע"י שליפת כל הרכיבים למתכון זה

**מחלקת CommentsToRecipe:**

קוד – מספרי, מפתח ראשי, אוטומטי

קוד מתכון – מספרי, חובה, מפתח זר לטבלת מתכונים

קוד משתמש – מספרי, חובה, מפתח זר לטבלת משתמשים - מייצג את המשתמש שהגיב

* **UserName - שם משתמש** - מחרוזת, כברירת מחדל null, מתמלא אוטומטית בשם פרטי ושם משפחה של המגיב

תגובה – מחרוזת, לא חובה

המשמעות של כל השינויים הללו:

1. בעת שליפה נקבל את המאפיינים הללו מלאים בערכים המתאימים אוטומטית, כך שלא נצטרך לשלוף מחדש לפי קוד.
2. בעת הוספה - נוכל להתעלם ממאפיינים אלו, כיון שאין לנו צורך בשמירה שלהם במסד הנתונים, אלא רק לחסוך בקוד.

הערה חשובה: בעת הוספת מתכון כן חובה לשלוח רשימת רכיבים, ע"מ להוסיף את הרכיבים הללו למסד הנתונים בטבלה המיועדת.

הערה לגבי כל המפתחות הראשיים: אין צורך וא"א לשלוח אותם בעת הוספה – הם מתמלאים אוטומטית ע"י המערכת של מסד הנתונים.

אפיון צד לקוח:

יש ליצור אפליקציה שתכיל את הפונקציונאליות הבאה:

1. אפשרויות הניתוב המוצגות בכל עת (nav):

דף הבית, הרשמה, כניסה, כל המתכונים

שם הלקוח המחובר יהיה מוצג בnav אא"כ המשתמש אנונימי

1. בדף הבית נציג פרסומות, לוגו, תמונות, וכדו'
2. בהרשמה נציג טופס הרשמה. יש לבדוק את תקינות האימייל. אין אפשרות לרשום לקוחות בעלי אימייל וסיסמה זהים!

(שימי לב, אין לכן אכיפה בצד שרת – יש לבדוק שהלקוח אכן אינו קיים לפני הרישום)

1. בכניסה – נציג טופס המכיל אימייל וסיסמה – יש לוודא שהלקוח קיים במערכת.

אם אינו קיים – נעביר לטופס ההרשמה. (תקינות אימייל)

1. אחרי כניסה והרשמה ננתב בחזרה לדף הבית. (שימי לב ששם הלקוח יופיע ב nav)
2. כל המתכונים – בדף זה נשלוף את כל המתכונים מהמאגר ונציג עבור כל מתכון: שם מתכון, תמונה, ושם העורך. כל מתכון יהיה ממוסגר בצבע המתאים לקטגוריה.
3. הקומפוננטה תאפשר סינון המתכונים לפי קטגוריה או רמה. (יש להציג למשתמש את כל הקטגוריות / את כל הרמות ולאפשר בחירה)
4. בלחיצה על מתכון – ננתב לקומפוננטה שתציג את כל פרטי המתכון. נאפשר הוספת תגובה – כמובן רק ללקוח מחובר.
5. תהיה אפשרות לראות את כל התגובות על המתכון הנבחר – הקומפוננטה תתוסף על פרטי המתכון (ולא ניתוב לקומפוננטה אחרת ,אפשרי ניתוב כילד...)
6. כאשר הלקוח מחובר (אחרי כניסה או הרשמה), תתוסף אפשרות ניתוב לאזור האישי.
7. אזור אישי יאפשר ניתוב פנימי (כילדים) לכל המתכונים שלי ולהוספת מתכון.
8. כל המתכונים שלי – שליפת כל המתכונים השייכים לאותו משתמש.
9. הוספת מתכון – טופס הוספה הכולל הוספת כל מאפייני המתכון. קטגוריה ורמה – בחירה מתוך הרשימה הקיימת. לצורך הוספת רכיבים, נציג את כל הרכיבים הקיימים במאגר ונאפשר בחירת רכיבים. כמו"כ נאפשר הוספת רכיב חדש שלא קיים עדיין. עבור כל רכיב שנבחר נאפשר הוספת כמות.

ממשק מנהל:

1. יש לשמור ברידקס את פרטי המנהל (אימייל וסיסמה). בכל כניסה למערכת, יש לבדוק לפני הפעלת הפונקציה מהשרת האם הפרטים תואמים לשל המנהל. אם כן נכתוב ב nav במקום שם הלקוח "מנהל" ונציג אפשרויות נוספות לניתוב: קטגוריות ורמות.
2. קטגוריות – תציג את כל רשימת הקטגוריות ותאפשר הוספת קטגוריה חדשה.
3. רמות – תציג את כל הרמות ותאפשר הוספת רמה.

דרישות כלליות:

* יש לעצב את האפליקציה באופן נעים לעין. ינתן ציון גם על עיצוב.
* יש להשתמש בנושאים השונים שנלמדו בשיעורים.   
  (style, form, useState, useRef, useEffect, routing, redux, axios…)
* יש לתת שמות משמעותיים למשתנים, להקפיד על קוד ברור, מובן ונקי, ולהבין את הקוד.

**הרבה הרבה הצלחה והנאה!!!**