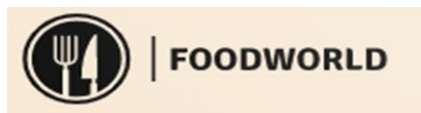


פרויקט WEB – חלק 3



FOODWORLD



מגישים:

318714508 ,305159980 ,203289665

מספר קבוצה- 19

תכנון האתר

****תכנון ובניית הקוד נעשה לפי הדרך הנוחה ביותר בשבילנו לעבוד על הקוד ועל כל אחת מהתיקיות שהוא כולל. מיינו בכוונה תחילה את כל קבצי ה-CSS, קבצי ה-TEMPLATES וקבצי ה-JS כל סוג של קובץ בתקיה שלו (ולא לפי צורת ה-Skeleton) מתוך נוחות שלנו לעבוד על הקוד וכל חלקיו. הקוד מורץ מתוך קובץ app.py ראשי שכולל את כל Routes הרלוונטיים לנו (ולא עפ"י מיון של כל דף באתר ל Directory נפרד עם הקבצים המתאימים לו בכדי שנוכל לעבוד במקביל על כל חלקי הפרויקט.**

קוד האתר מחולק ל-4 תיקיות מרכזיות:

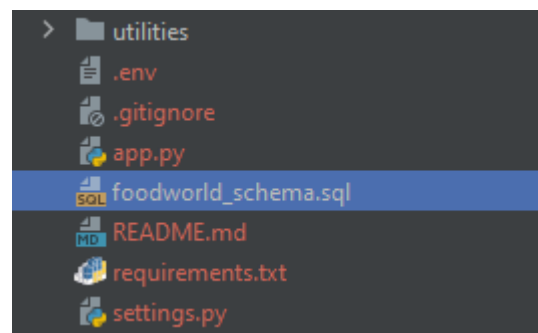
- (1) תיקיית static:
בתיקיה זו ישנה תת תיקיית CSS שבה מרוכזים כל קבצי ה-CSS במטרה שיהיו מסודרים במקום אחד. בנוסף ישנה תת תיקיית JS שבה מרוכזים כל קבצי ה-JAVASCRIPT שלנו, וכן תתי תיקיות לאייקונים והתמונות שהשתמשנו באתר.
- (2) תיקיית templates:
בתיקיה זו מרוכזים כל קבצי ה-html לכל אחד מהעמודים שלנו, במקום אחד.
- (3) תיקיית models:
בתיקיה זו נמצאות המחלקות המתווכות למחלקת בסיס הנתונים DBManager.
- (4) תיקיית utilities:
בתיקיה זו נמצאת מחלקת DBManager.
- (5) app.py ראשי – דרכו מריצים את כל ה-ROUTES במערכת
- (6) foodworld_schema.sql המכיל את כל מסד ה-DB

יצירת בסיס נתונים

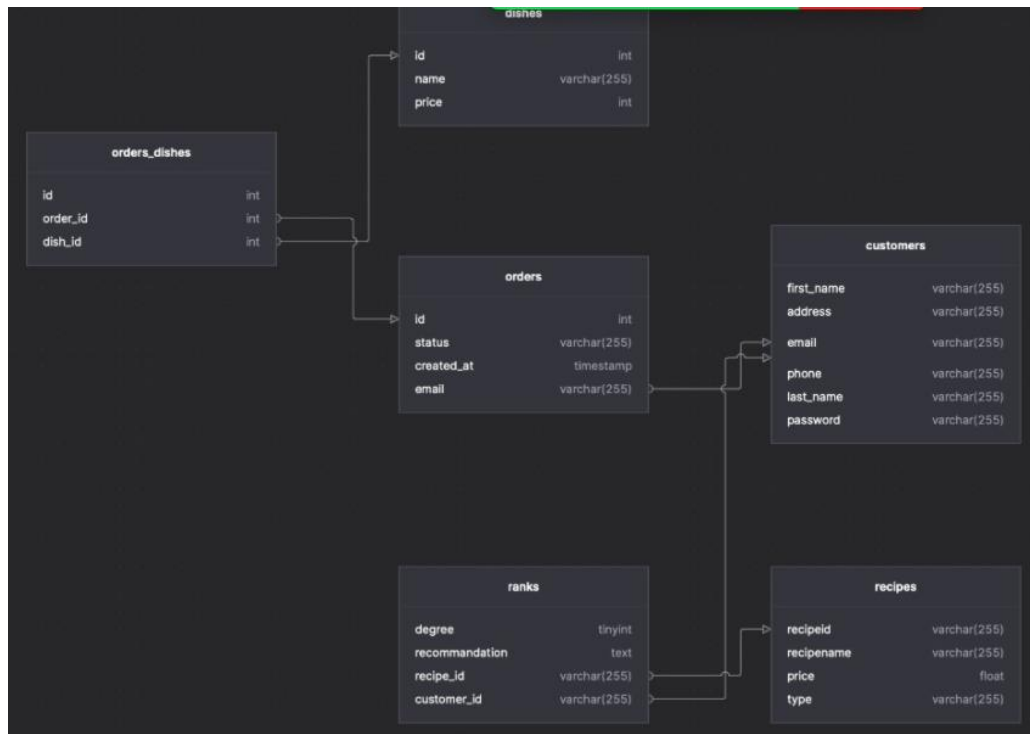
ניתן לגשת לבסיס הנתונים מתוך קובץ foodworld_schema.sql אשר מצורף בקובץ הפרויקט:

root = USER

Amber1958 = PASSWORD



בסיס הנתונים שלנו מכיל מספר טבלאות בהתאם לתרשים ה-ERD מחלק א' של הפרויקט, וטבלאות נוספות שהקמנו לפי צרכים שהתווספו בחלק ב' של הפרויקט:



1. טבלת משתמשים - Customers:

הלקוחות שלנו הם משתמשים שמבצעים רישום לאתר, ובהתאם הם יהיו יכולים לבצע הזמנות של מאכלים מהאתר, ולכתוב המלצות/דיס-המלצות על מתכונים קיימים באתר. המפתח הראשי- אימייל המשתמש (לכל משתמש אימייל ייחודי משלו).

```

CREATE TABLE `customers` (
  `first_name` varchar(255) DEFAULT NULL,
  `address` varchar(255) DEFAULT NULL,
  `email` varchar(255) NOT NULL,
  `phone` varchar(255) DEFAULT NULL,
  `last_name` varchar(255) DEFAULT NULL,
  `password` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci DEFAULT NULL,
  PRIMARY KEY (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
  
```

2. טבלת מנות - Dishes:

באתר שלנו ניתן להזמין מנות מהקטלוג תחת לשונית Gift Cards. לכל מנה באתר יש שם ומחיר. מכיוון שעלויות להיות מנות שונות בעלות שם דומה, המפתח הראשי של טבלה זו יהיה אינדקס מספרי שיינתן לכל מנה.

```

DROP TABLE IF EXISTS `dishes`;
CREATE TABLE `dishes` (
  `id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(255) DEFAULT NULL,
  `price` int NOT NULL,
  `image` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
  
```

3. טבלת מנות משתמש-Customer dishes:

כאשר משתמש מבצע הזמנה של מנה או מספר מנות מסוימות באתר, ההזמנות יהיו מאוחסנות בטבלה זו.
המפתח הראשי בטבלה זו יהיה אינדקס מספרי שיינתן לכל הזמנה.
המפתחות הזרים בטבלה זו יהיו אימייל המשתמש ואינדקס המנה המוזמנת, שכן משתמש יכול להזמין אותה מנה מספר פעמים.
שדה נוסף שיש בטבלה הוא 'חותמת זמן' על מנת שתהיה אפשרות לראות מתי בוצעה הזמנה מסוימת.

```
CREATE TABLE `customers_dishes` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `email` varchar(255) DEFAULT NULL,  
  `dish_id` int DEFAULT NULL,  
  `created_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP,  
  `status` varchar(255) NOT NULL,  
  PRIMARY KEY (`id`),  
  KEY `dish_id` (`dish_id`),  
  KEY `email` (`email`),  
  CONSTRAINT `customers_dishes_ibfk_1` FOREIGN KEY (`dish_id`) REFERENCES `dishes` (`id`),  
  CONSTRAINT `customers_dishes_ibfk_2` FOREIGN KEY (`email`) REFERENCES `customers` (`email`)  
) ENGINE=InnoDB AUTO_INCREMENT=38 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

4. טבלת מתכונים-Recipes:

מעבר להזמנת מנות מהקטלוג, באתר שלנו ניתן לראות המלצות למתכונים שונים על פי סוגי מדינות.
המפתח הראשי בטבלה יהיה אינדקס מספרי- recipe_id, ולא recipe_name, שכן כמו שהוסבר בטבלת Dishes, המטרה למנוע בלבול בין מתכונים שונים בעלי שם דומה.

```
DROP TABLE IF EXISTS `recipes`;  
CREATE TABLE `recipes` (  
  `recipeid` varchar(255) NOT NULL,  
  `recipename` varchar(255) DEFAULT NULL,  
  `price` float DEFAULT NULL,  
  `type` varchar(255) DEFAULT NULL,  
  PRIMARY KEY (`recipeid`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

5. טבלת Ranks:

באתר שלנו תהיה אפשרות למשתמשים לדרג ולתת ציון מספרי למתכונים באתר ובנוסף לכתוב ולהשאיר תגובה/המלצה על מתכון מסוים.
המפתח הראשי של הטבלה יורכב מ-2 שדות: האימייל של המשתמש, וה-id של המתכון.
בהתאם, 2 שדות אלו ישמשו כמפתחות זרים לשדה email בטבלת Customers, ולשדה recipe_id בטבלת Recipes.
כל משתמש יוכל לדרג ולכתוב לכמה מתכונים, ובהתאם כל מתכון יוכל להיות מדורג ולקבל תגובות ממספר שונה של משתמשים.

```

DROP TABLE IF EXISTS `ranks`;
CREATE TABLE `ranks` (
  `degree` tinyint NOT NULL,
  `recommandation` text,
  `recipe_id` varchar(255) NOT NULL,
  `customer_id` varchar(255) NOT NULL,
  PRIMARY KEY (`customer_id`,`recipe_id`),
  KEY `fk_recipeid` (`recipe_id`),
  CONSTRAINT `fk_customerid` FOREIGN KEY (`customer_id`) REFERENCES `customers` (`email`),
  CONSTRAINT `fk_recipeid` FOREIGN KEY (`recipe_id`) REFERENCES `recipes` (`recipeid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

יצירת מחלקות תומכות

Class dishes

במחלקה זו קיימות השאילתות המבוצעות על טבלאות המנות וההזמנות של המשתמשים באתר.

```

class Dishes:
    @staticmethod
    def get_dishes():
        # Get all dishes from database
        return dbManager.fetch("SELECT * FROM dishes")

```

דוגמא לפונקציה מהמחלקה:

```

@app.route('/create_user_dish', methods=['POST'])
def create_user_dish():
    args = request.args
    dish_id = args.get('dish_id')
    email = session.get('email')
    Dishes.create_user_dish(email, dish_id)
    data = {
        'dish_id': dish_id,
        'status': "success"
    }
    return jsonify(data)

```

Class Auth

בטבלה זו קיימות השאילתות שמבוצעות על טבלאות המשתמשים שנרשמים או קיימים באתר.

```

@app.route('/registration', methods=["GET", "POST"])
def registration_Page():
    if request.method == "POST":
        email = request.form.get('email')
        password = request.form.get('password')
        first_name = request.form.get('fname')
        last_name = request.form.get('lname')

        if len(email) >= 1 and len(password) >= 1 and len(first_name) >= 1:
            is_user_saved = Auth.register(email, password, first_name, last_name)

```

שאליות SQL

באתר שלנו הגדרנו מספר שאליות SQL שבאמצעותן מתבצעת התנהלות האתר אל מול בסיס הנתונים:

Insert customer registration

כשמשתמש חדש מבצע הרשמה לאתר, שאלית זו מוסיפה את הפרטים שלו לטבלת הלקוחות בבסיס הנתונים.

```

INSERT INTO customers (email, password, first_name, last_name) "
VALUES ('{email}', '{password}', '{first_name}', '{last_name}')"

```

Customer login

שאלית המאפשרת לקבל את פרטי המשתמשים הרשומים לאתר מבסיס הנתונים ולבדוק האם לקוח מסוים קיים/נרשם למערכת.

```

SELECT * FROM customers WHERE email = %s AND password = %s;

```

Get dishes

שאלית המאפשרת לנו לקבל את כל המנות הקיימות בקטלוג האתר, מהטבלה הרלוונטית בבסיס הנתונים. **בנוסף, כל שינוי של שם המנה, מחיר וכד' יתבצע דרך בסיס הנתונים ויתעדכן ישירות באתר.**

```

SELECT * FROM dishes

```

Create customer dish

שאלית שמוסיפה פרטים אודות המנה שלקוח הזמין, לטבלה הרלוונטית בבסיס הנתונים.

```

INSERT INTO customers_dishes (email, dish_id, status) "
VALUES ('{session.get('email')}', '{dish_id}', 'pending')"

```

Delete customer dish

כאשר לקוח מוחק מנה שהוא בחר כחלק מההזמנה שלו, השאלית תמחק את המנה מפרטי ההזמנה של הלקוח, בטבלה הרלוונטית בבסיס הנתונים.

```
DELETE FROM customers_dishes WHERE email = '{session.get('email')}' AND id = '{dish_id}'
```

Update customer dish

שאלתה שמעדכנת את סטטוס ההזמנה, בהתאם למצב שלה ("בתהליך עבודה" או "הזמנה נשלחה") בטבלה הרלוונטית בבסיס הנתונים.

```
UPDATE customers_dishes SET status = '{status}' "
WHERE email = '{session.get('email')}' AND id = '{dish_id}' "
```

Get customer dish

שאלתה המאפשרת לנו לקבל את כל הזמנות המנות שהלקוח ביצע באתר, כאשר ההזמנות ממוינות לפי תאריך היצירה העדכני ביותר שלהן.

```
SELECT customers_dishes.id, name, price, image, status, created_at FROM dishes "
JOIN customers_dishes ON customers_dishes.dish_id = dishes.id "
AND customers_dishes.email = '{session.get('email')}' ORDER BY created_at DESC "
```

תוכן דינמי

עמוד Recommendations

באתר שלנו ניתן להוסיף המלצה/ביקורת על מתכון מסוים, וההמלצה תתעדכן ותופיע בעמוד זה (תחת הלשונית Recommendations).

* בכדי להגיע לעמוד נתינת המלצה יש לגשת ללשונית FOODWORLD – ולבחור את המדינה שממנה הוא מעוניין לקבל מתכון (נדרש ללחוץ על שם המדינה בכדי להיכנס לעמוד המתכונים)



Pc



כאשר אין עדיין תגובות/המלצות על מתכונים, העמוד יופיע בצורה ריקה:

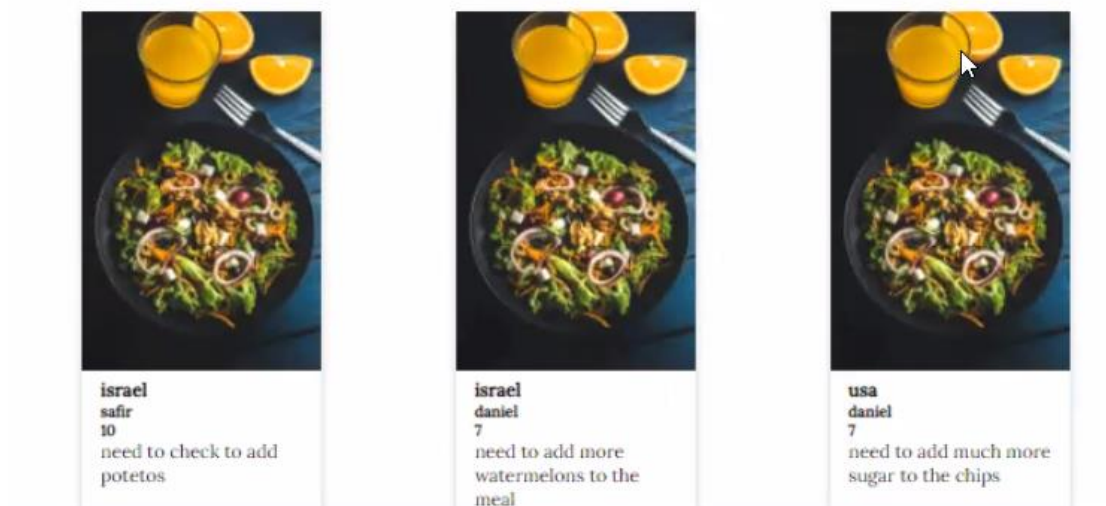


Top Recommendations

Food World © 2021 | [Policy](#) | [Customer Service](#)

כדי להוסיף המלצה, נבצע זאת דרך העמוד של המתכון הייעודי. בתמונה הבאה ניתן לראות שליחה של המלצה, כאשר דירגנו מנה מסוימת ממדינה מסוימת ונתנו לה דירוג 7. כמו כן ניתן לראות שלאחר מילוי ההמלצה, תופיע חלונית שמאשרת את ביצוע ההמלצה (חשוב לציין כי שם המשתמש מופיע בלשונית NAME ללא צורך ההזנה של המשתמש):

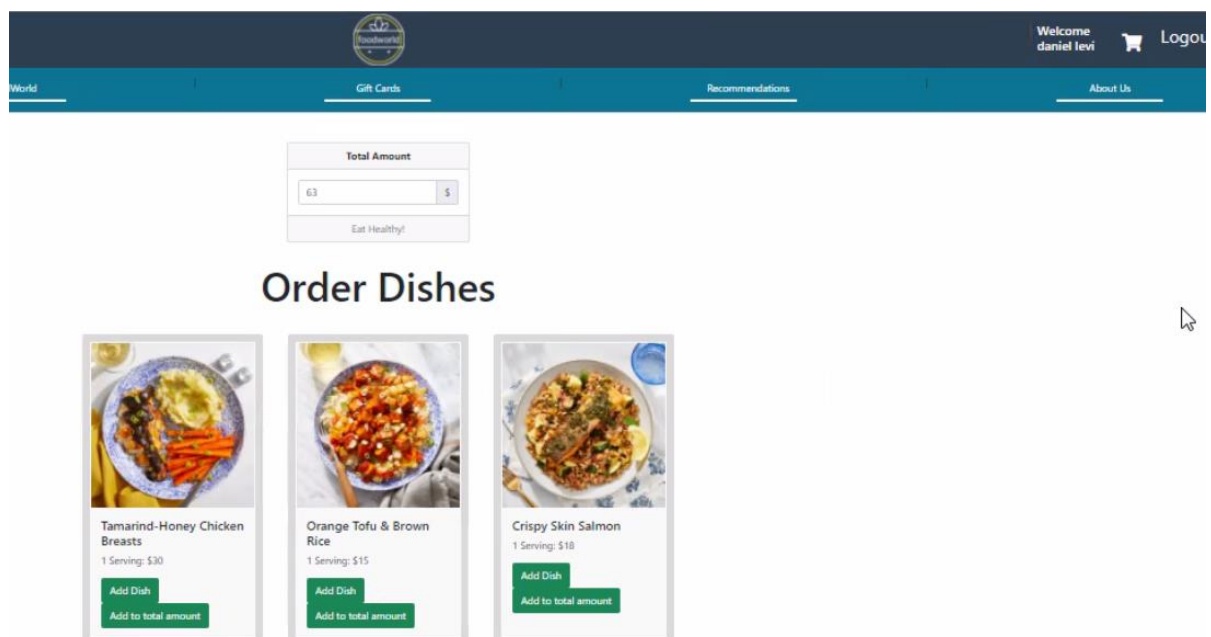
לאחר מכן ההמלצה תופיע בעמוד ההמלצות לצד תגובות/המלצות על מתכונים שניתנו בעבר (בצד ימין במקרה שלנו תופיע התגובה הנ"ל שמילאנו):



עמוד זה מתנהל אל מול בסיס הנתונים שלנו, כאשר כל הוספה או מחיקה של תגובה אודות מתכון מסוים, תתעדכן בהתאם בטבלה שבבסיס הנתונים.

עמוד Orders

למשתמשים הרשומים באתר שלנו תהיה האפשרות לבצע הזמנות של מנה/מספר מנות מסוימות מקטלוג המנות שלנו תחת לשונית Gift Cards.



כאשר משתמש לוחץ על כפתור "הוסף" (Add dish), המנה תתווסף לעמוד ההזמנות של המשתמש שניתן להגיע אליה באמצעות לחיצה על אייקון "עגלה" בבאנר העליון מצד ימין באתר (עיגול אדום ליד העגלה מסמן על כך שיש מוצרים בדף ההזמנות), וכן תקפוץ הודעה שתודיע למשתמש על הוספת הפריט לעגלה. לחיצה על כפתור "Add to total amount" תסייע למשתמש לחשב את סך המחיר של המנות אותן הוא רוצה להזמין.



בעמוד ההזמנות ניתן לראות את ההזמנות הנוכחיות של הלקוח ואת היסטוריית ההזמנות שלו, מסודרות מההזמנה העדכנית ביותר, כאשר ניתן לבצע מספר פעולות בעמוד זה:

- ניתן למחוק/לבטל הזמנה (כפתור Remove), והיא תוסר מעמוד ההזמנות.
 - ניתן להוציא את ההזמנה לפועל (כפתור Order Now), והסטטוס של ההזמנה ישתנה בהתאם.
 - ניתן להוסיף הערה להזמנה (למשל-הסרה של רכיב מסוים מהמנה).
- בתמונה למטה ניתן לראות את סך ההזמנות של משתמש מסוים, כאשר לחיצה על כפתור Order Now תשנה את סטטוס ההזמנה ל-"נשלחה".

daniel levi Orders

| | | |
|--------|--|-----------|
| Remove | Crispy Skin Salmon price: 18 Created at: 2022-02-22 15:23:22 Comment: <input type="text" value="insert comment"/> status: pending | Order Now |
| Remove | Orange Tofu & Brown Rice price: 15 Created at: 2022-02-22 15:23:19 Comment: <input type="text" value="insert comment"/> status: pending | Order Now |
| Remove | Tamarind-Honey Chicken Breasts price: 30 Created at: 2022-02-22 15:22:44 Comment: <input type="text" value="insert comment"/> status: pending | Order Now |

לאחר שלוחצים על כפתור Order Now, הכפתור ישתנה ל-Cancel Now, כלומר יהיה אפשר להפסיק את תהליך הזמנת המנה במידה והמשתמש יתחרט.



daniel levi Orders

| | | |
|---------------------------------------|--|---|
| <input type="button" value="Remove"/> | Tamarind-Honey Chicken Breasts price: 30 Created at: 2022-02-22 16:19:02 Comment: <input type="text" value="insert comment"/> status: delivered | <input type="button" value="Cancel Now"/> |
| <input type="button" value="Remove"/> | Orange Tofu & Brown Rice price: 15 Created at: 2022-02-22 16:11:39 Comment: <input type="text" value="insert comment"/> status: pending | <input type="button" value="Order Now"/> |
| <input type="button" value="Remove"/> | Orange Tofu & Brown Rice price: 15 Created at: 2022-02-22 16:07:12 Comment: <input type="text" value="insert comment"/> status: pending | <input type="button" value="Order Now"/> |

לחיצה על כפתור Remove תסיר את ההזמנה מעמוד ההזמנות של המשתמש.



עמוד זה מתנהל אל מול בסיס הנתונים שלנו, כאשר כל הוספה, עדכון או מחיקה של הזמנה אודות מנה מסוימת, תתעדכן בהתאם בטבלה שבבסיס הנתונים.

בתמונה למטה ניתן לראות את בסיס הנתונים שלנו שמכיל רשומות של הזמנות שונות.

| WHERE | | | | | | ORDER BY | |
|-------|----|-------------------------|---------|---------------------|-----------|----------|--|
| | id | email | dish_id | created_at | status | | |
| 1 | 14 | daniel@com | 3 | 2022-02-20 19:28:32 | pending | | |
| 2 | 35 | tomere@gmail.com | 2 | 2022-02-20 20:28:27 | delivered | | |
| 3 | 36 | tomere@gmail.com | 3 | 2022-02-20 20:28:28 | pending | | |
| 4 | 37 | tomere@gmail.com | 6 | 2022-02-20 20:28:30 | delivered | | |
| 5 | 45 | safirxpik@hotmail.com | 2 | 2022-02-22 14:36:23 | delivered | | |
| 6 | 47 | safirxpik@hotmail.com | 2 | 2022-02-22 14:37:02 | delivered | | |
| 7 | 49 | bibis@post.bgu.ac.il | 1 | 2022-02-22 14:41:31 | pending | | |
| 8 | 50 | bibis@post.bgu.ac.il | 2 | 2022-02-22 14:41:33 | pending | | |
| 9 | 51 | bibis@post.bgu.ac.il | 2 | 2022-02-22 14:41:34 | pending | | |
| 10 | 64 | daniel_levi@hotmail.com | 2 | 2022-02-22 16:07:12 | pending | | |
| 11 | 66 | daniel_levi@hotmail.com | 2 | 2022-02-22 16:11:39 | pending | | |
| 12 | 69 | daniel_levi@hotmail.com | 1 | 2022-02-22 16:19:02 | pending | | |

וכאשר לצורך הדוגמא המשתמש לוחץ על כפתור Remove ליד אחת המנות שהוא הזמין, בהתאם רשומת ההזמנה של אותה המנה תוסר מבסיס הנתונים:

| WHERE | | | | | | ORDER BY | |
|-------|----|-------------------------|---------|---------------------|-----------|----------|--|
| | id | email | dish_id | created_at | status | | |
| 1 | 14 | daniel@com | 3 | 2022-02-20 19:28:32 | pending | | |
| 2 | 35 | tomere@gmail.com | 2 | 2022-02-20 20:28:27 | delivered | | |
| 3 | 36 | tomere@gmail.com | 3 | 2022-02-20 20:28:28 | pending | | |
| 4 | 37 | tomere@gmail.com | 6 | 2022-02-20 20:28:30 | delivered | | |
| 5 | 45 | safirxpik@hotmail.com | 2 | 2022-02-22 14:36:23 | delivered | | |
| 6 | 47 | safirxpik@hotmail.com | 2 | 2022-02-22 14:37:02 | delivered | | |
| 7 | 49 | bibis@post.bgu.ac.il | 1 | 2022-02-22 14:41:31 | pending | | |
| 8 | 50 | bibis@post.bgu.ac.il | 2 | 2022-02-22 14:41:33 | pending | | |
| 9 | 51 | bibis@post.bgu.ac.il | 2 | 2022-02-22 14:41:34 | pending | | |
| 10 | 64 | daniel_levi@hotmail.com | 2 | 2022-02-22 16:07:12 | pending | | |
| 11 | 66 | daniel_levi@hotmail.com | 2 | 2022-02-22 16:11:39 | pending | | |

מימוש טפסים

טופס Login

על מנת להיכנס לאתר, כל משתמש מחויב להירשם אליו. כאשר משתמש רשום במערכת, הוא יזין את פרטיו ותבצע בדיקה אל מול טבלת המשתמשים בסיס הנתונים על מנת לאמת את הפרטים, ולבדוק את קיומו של המשתמש. במידה והמשתמש קיים, הוא יורשה להיכנס ובנוסף יופיע השם שלו בראש האתר כסימן המעיד על כך שהוא מחובר.



לחיצה על כפתור Logout תנתק את המשתמש מהמערכת, ותיאלץ אותו להיכנס מחדש. לשים לב שכאשר משתמש מתחבר למערכת, בחלק של השארת תגובה על מתכון מסוים, השם שלו יופיע אוטומטית בטופס התגובה (כפי שאפשר לראות בהסבר על עמוד Recommendations).

טופס Registration

כאשר משתמש לא קיים בבסיס הנתונים, הוא לא יורשה להיכנס לאתר והוא יתבקש לבצע רישום אליו. בעת הרישום מתבצע בדיקת ולידציה של הנתונים (למשל שהאימייל שנרשם אכן לא קיים במערכת, ובמידה וכן תוקפץ הערה על כך למשתמש). לאחר סיום ההרשמה, פרטי המשתמש החדש יוכנסו לטבלה הרלוונטית בבסיס הנתונים, על מנת לאפשר התחברות עתידית לאתר ללא צורך להירשם מחדש.

Email already exists

Create Your Account

Create Your Account

First Name*

Last Name*

Email*

Password*

Confirm Password*

Signup Now

If you have account, Please [Login Now](#)