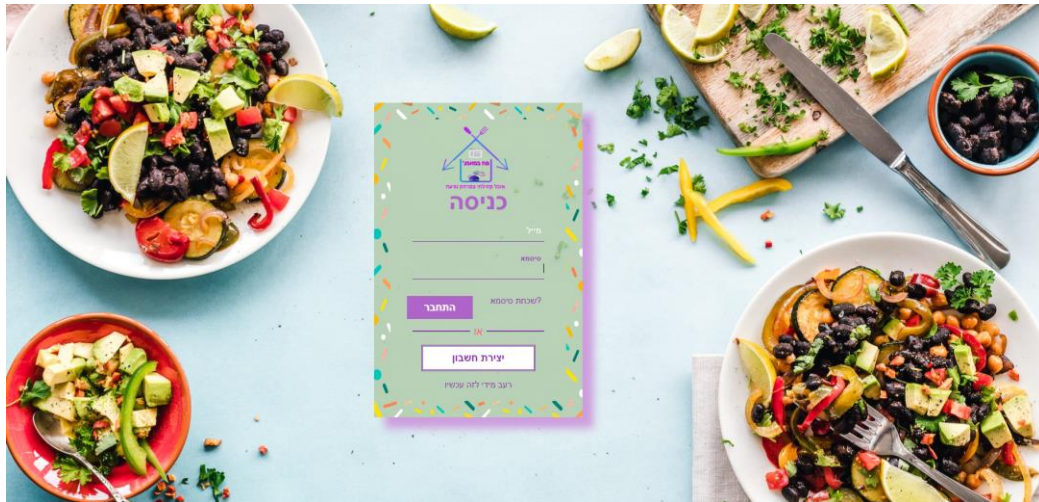




הגשה Web-2

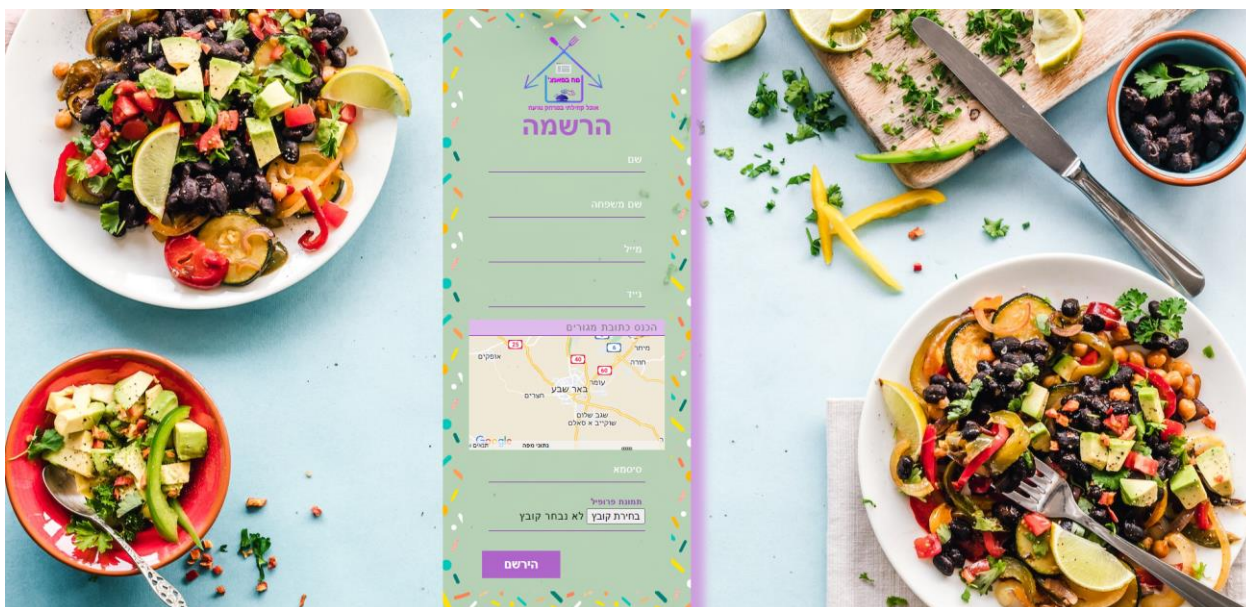
יצירת עמודים HTML:

1. עמוד LOGIN



בעמוד הכניסה, המשתמש נכנס לאתר במידה ורשום. במידה ושכח סיסמא, קופץ לו חלון בו מזיין את המייל שלו לריסוט הסיסמא. במידה ולמשתמש אין חשבון הוא יכול לבחור באחת מ2 האופציות: 1. להיכנס לעמוד הרישום 2. ללחוץ על "רעב מידי לזה עכשיו" ולהגיע ישירות לדף הבית. כל השדות הם required.

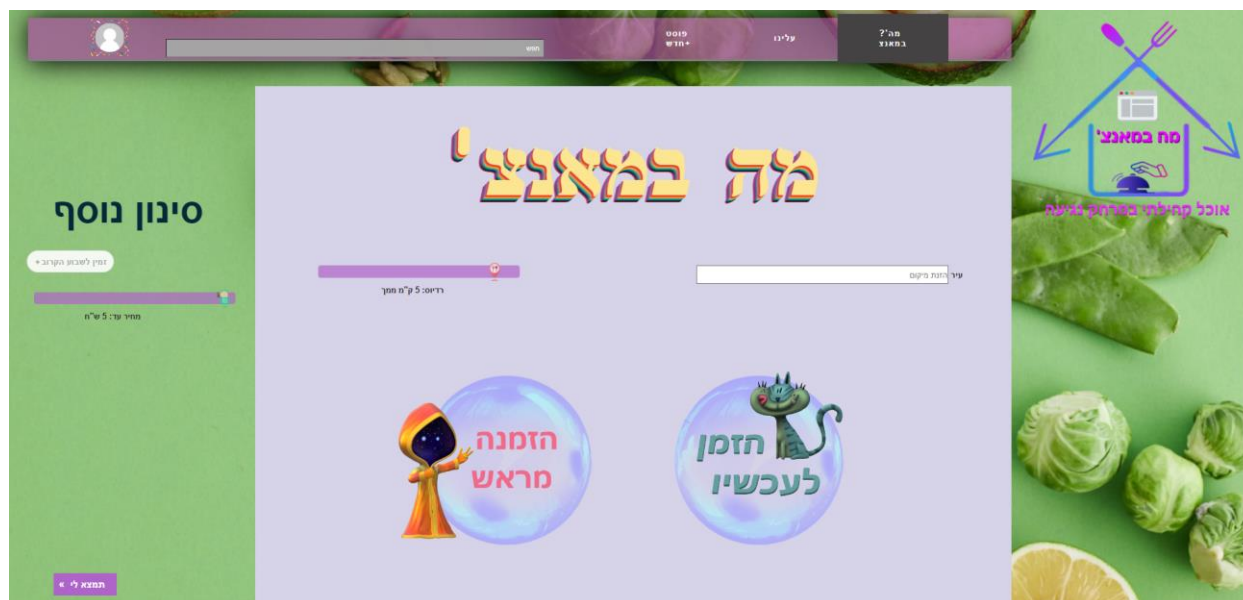
2. יצירת חשבון



טופס בו המשתמש ממלא את פרטיו: שם, שם משתמש, מייל, ניד, כתובת (דרך גוגל), סיסמא ומוסיף תמונת פרופיל במידה ומעוניין. כל השדות הם required, מלבד הכתובת ותמונת הפרופיל.

ניתן להתחיל להזין את כתובת הדירה וזה כבר יושלם לבד. וכך מבצע דקירה אשר תיחשב ככתובת משתמש.

3. עמוד כניסה



מימין ומשמאל יש מרחק שווה של 20vw, מצד ימין מופיע הלוגו של החברה ומשמאל מופיע סינון נוסף (לא חובה על מנת להמשיך, לעומת העיר והרדיוס), סינון המרחק הוא חובה מבחינתי כי על זה מתבסס האתר, מציאת "אוכל קהילתי במרחק נגיעה". מלמעלה ניתן לראות את ה-nav, דרכו ניתן לנווט באתר ואף כאשר

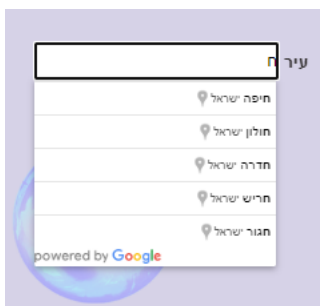
עומדים על תמונת הפרופיל נפתחות 2 אופציות נוספות של "להתחבר/להתנתק" והמשמעות היא לחזור לעמוד הכניסה או "להירשם" ובכך להישלח לעמוד יצירת החשבון. לאחר בחירת המיקום (ערים בישראל בלבד, בעזרת גוגל) ורדיוס אשר המינימאלי הוא 1 ק"מ, המשתמש יתחיל ללחוץ על הבועות עד שהעדפה שלו תסווג ותוצאות יופיעו בהתאם. לאחר כל לחיצה, העמוד ירד קצת בהתאם.

כלל הסיווגים הם אופציונאליים שכן סינון המחיר והרדיוס פעילים באופן דיפולטי. במידה ולא יהיה אישור למיקום, התוצאות יחזירו את כלל האופציות במגבלת המחיר המסונן.

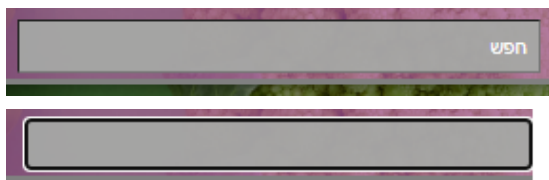
* הבועות אומנם בעלות מראה מעט ילדותי אך נעשו לבד.

אופציה לדוגמה המתקבל מלחיצה על:
הזמן לעכשיו/הזמנה מראש -> לא כשר -> מלוח .

סינון ערים בלבד.
בנוסף להוצאת API והוספת קוד של גוגל לJS.



```
function autocompleteCity(){
    var autocomcity = new google.maps.places.Autocomplete((document.getElementById( "citySearch")), {
        types:['(cities)'],componentRestrictions: {country: "il" } ,
    })
}
```



בשורה של החיפוש, ברגע בו עומדים עליו, המילה "חיפוש" נעלמת.

הסינון "זמין לשבוע הקרוב" עוצב דרך CSS במראה ובנוסף כאשר לוחצים עליו, ישנה אנימציה הפועלת מאחורה אשר משנה לו את הצורה הצבע ואת הקטנה בפנים מ'פלוס' ל- 'מינוס'.



הקוד לשינוי התגים
(כן/לא).

```
.boxTags li label {
    padding: 8px 24px;
    cursor: pointer;
}
.boxTags li label::before {
    display: inline-block;
    text-rendering: auto;
    font-weight: 900;
    font-size: 14px;
    padding: 3px;
    content: "\002b";
    transition: transform .3s ease-in-out;
}
.boxTags li input[type="checkbox"]:checked + label::before {
    content: "\221A";
    transform: rotate(-360deg);
    transition: transform .3s ease-in-out;
}
.boxTags li input[type="checkbox"]:checked + label {
    border: 2px solid #1dbcf8;
    background-color: #cfe9de;
    color: #fff;
    transition: all .2s;
}
.boxTags li input[type="checkbox"] {
    position: absolute;
    opacity: 0;
}
.boxTags li input[type="checkbox"]:focus + label {
    border: 2px solid #e9a1ff;
```



הסקלות הנעות של "רדיוס" ו"מחיר עד" מעוצבות בCSS כך שמעט מאופרות וכאשר עומדים עליהן נצבעות. בנוסף האייקון הזז הוא תמונה. בנוסף, עיצבתי דרך JS כך שכאשר מזיזים את הסקלה ה"רדיוס" או ה"מחיר" היה רשום למטה בהתאמה. HTML:

```
<div class="searchRadiusBar">
  <input type="range" min="1" max="50" value="5" class="slider" id="myRange">
  <label for="myRange"></label>
  <p class="radius"><span id="demo"></span> ק"מ ממך </p>
</div>
```

```
let slider = document.getElementById( elementId: "myRange");
let slider2 = document.getElementById( elementId: "myRange2");
let output = document.getElementById( elementId: "demo");
let output2 = document.getElementById( elementId: "demo2");

output.innerHTML = slider.value;
output2.innerHTML = slider2.value;

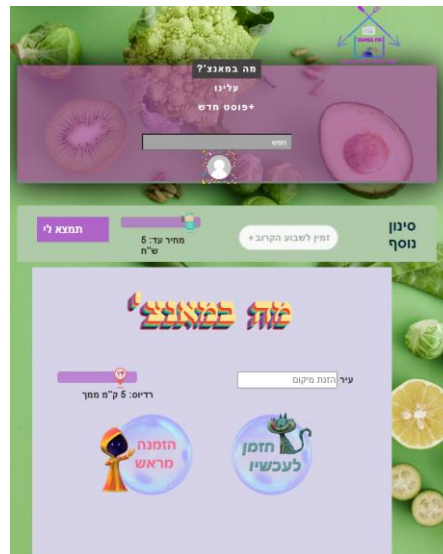
slider.oninput = function() {
  output.innerHTML = this.value;
}

slider2.oninput = function() {
  output2.innerHTML = this.value;
}
```

כאן ניתן לראות את הקוד ב js שמאפשר זאת.

```
.contentItem.h2 {
  grid-row: 1 ;
  display: flex;
  margin-top: 2vh;
  justify-content: center;
  font-family: 'Calligraphitti', cursive;
  font-size: 7vw;
  color: #fde092;
  text-shadow: -2px 3px #eb452b, -3px 2px #A54C9CB2, -3px 4px #eb452b, -4px 3px #A54C9CB2,
  -4px 5px #efa032, -5px 4px #A54C9CB2, -5px 6px #efa032, -6px 5px #A54C9CB2,
  -6px 7px #40b59b, -7px 6px #A54C9CB2, -7px 8px #40b59b, -8px 7px #A54C9CB2,
  -8px 9px #017e7f, -9px 8px #A54C9CB2, -9px 10px #017e7f, -10px 9px #A54C9CB2,
  -10px 11px #052939, -11px 10px #A54C9CB2, -11px 12px #052939, -12px 11px #A54C9CB2,
  -12px 13px #c11a2b, -13px 12px #A54C9CB2, -13px 14px #c11a2b, -14px 13px #A54C9CB2;
```

כמו כן, גם הכתב בכותרת עשוי משילוב 6 שכבות של אותיות בתוזה קלה.



מצב מסך קטן עם עמודה אחת.

```
.button, .submit-btn{
  display: inline-block;
  background-color: #af64c9;
  border: none;
  color: white;
  font-weight: bold;
  font-size: 1.2rem;
  padding: 10px;
  width: 120px;
  margin-right: 130px;
}

.submit-btn{
  transition: all 0.5s;
  cursor: pointer;
}

.submit-btn span{
  cursor: pointer;
  position: relative;
  transition: 0.5s;
}

.submit-btn span:after{
  content: '\00ab';
  position: absolute;
  opacity: 0;
  top: 0;
  left: -10px;
  transition: 0.5s;
}

.submit-btn:hover span{
  padding-left: 22px;
}

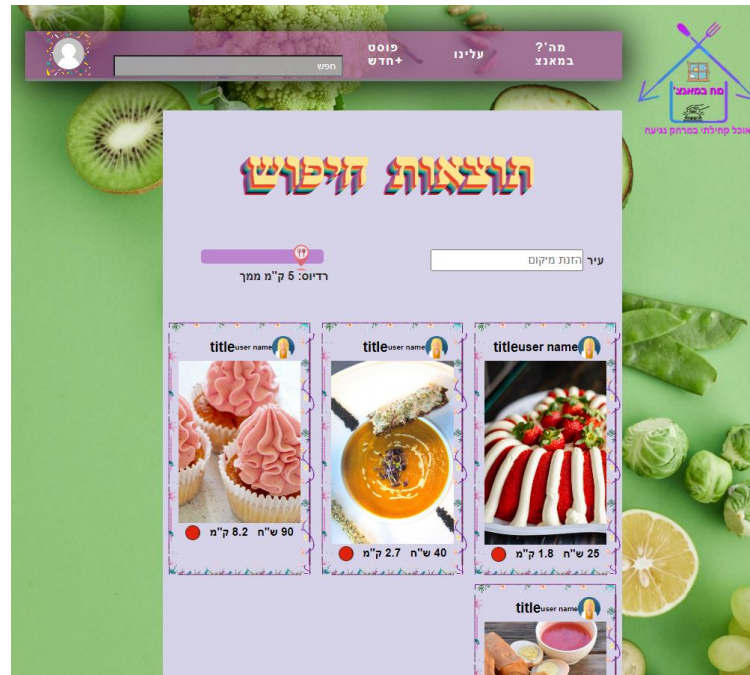
.submit-btn:hover span:after{
  opacity: 1;
  left: 0;
}
```

שינוי כפתור:

כפתור "התחבר", "שלח", "הירשם", "תמצא לי", "סינון נוסף", "פרסם" בוצע ע"י אותה סימולציה ב CSS.



4. עמוד לאחר חיפוש



כאן מתקבלות תוצאות הסינון,

ניתן להמשיך לסנן עוד או ללחוץ על אחת התמונות וכך בעצם להיכנס לתוך דף המשתמש.

לכלל העמודים נעשתה התאמה למסך קטן מ 850px.

5. יצירת פוסט חדש

בעמוד זה, המשתמש יכול ליצור לעצמו פוסט מאכל או פוסט אירוע (אירוע בו המאכלים שלו מכבדים כמובן).

מדובר בטופס בו המשתמש מזין פרטים רלוונטיים, כולל תיוגים שיעזרו ללקוחות למצוא אותם.

בנוסף, ניתן הסבר למשתמש כיצד להוסיף יומן גוגל לאתר ובכך לנהל את הבקשות המתקבלות בצורה נוחה, מהירה ומוכרת.

יומן זה ישמש אותו לכלל המאכלים, במידה והוסיף פעם אחת, לא יהיה צורך לעשות זאת שוב.



6. עמוד משתמש אליו נכנסים בעקבות חיפוש

בעת כניסה למשתמש מתרחב המידע הודות המאכל/ההופעה אותם פירסמו. ניתן כעת לראות בנוסף לתמונה, מרחק ומחיר גם הסבר מהמוכר, כולל זמינות למשלוחים, נייד המוכר ומיקום.

לוח שנה הוא פר מוכר, בהם יוכל לפתוח תאריכים בהם פנוי. היום ידע לשקלל את הזמן מראש אותו המוכר מבקש לקבל לפני ביצוע הזמנה על מנת להכינה ולקונים נראה רק תאריכים פנויים.

בדף זה הקונה יכול לבחור לשלם מאשראי ויעביר אותו למסלקה. לשלם בביט או במזומן מול המוכר. דבר היחיד אישור מקדים של המוכר.

מבחינת JS, תחילה רצייתי ליצור HTML נפרד לפוסט ואותו לשכפל לתוך HTML רלוונטי אך הבנתי כי ישנן דרכים רבות ליצור INCLUDE אך כולן כוללת תוספים אחרים לHTML ולכן כעת הכנסתי אותם פיזית, אך בהמשך אחרי יצירת הטופס של המשתמש לפוסט חדש, את אותו האובייקט אכניס לתבנית של 'פוסט'.

הקוד מייצר אובייקט ישירות משליחת טופס.

```
// חפיכת פוסט חדש לאובייקט
const formPost = document.getElementById('formToPost');
formPost.addEventListener('submit', callbackFunction);

function callbackFunction(event) {
  event.preventDefault();
  const myFormData = new FormData(event.target);
  const formDataObj = {};
  myFormData.forEach((value: FormDataEntryValue, key: string) => (formDataObj[key] = value));
  formDataObj.hashtag = myFormData.getAll('hashtag');
  console.log(formDataObj);

  return(formDataObj);
}
```

פונקציה לתזוזת הבועות:

נפתחות הבועות הנכונות בהתאם ללחיצה הקודמת שלך, וכרגע הבועות הראשונות, בעת לחיצה זוכית עליהם גם מחזירים את הבועות חזרה.

```
/* ***** Bubbles move ***** */
document.querySelectorAll( selectors: '.bubbleItem').forEach(myfunc);

function myfunc(item){
  item.clickCount = 0;
  let n = 0;
  item.addEventListener('click', function () {
    item.clickCount++;
    if(searchBTN.clickCount === 1){
      console.log("BREAK");
    }
    else if (item.clickCount === 2*n+1) { //זיזת ימ
      console.log(item);
      console.log(item.clickCount);
      if (item.id === ('meRosh'))||item.id === ('forNow')) bubopen1( vis vis = 1, item.id);
      n++;
    }else{
      if (item.id === ('meRosh') || item.id === ('forNow')) {
        bubopen1( vis vis = 0);
      }
    }
  })
}

function bubopen1(vis,id) {
  // if(id==='meRosh') {forNow.onclick=null} else{meRosh.onclick=null};

  if(vis===1) {
    Kosher.style.visibility = "visible";
    notKosher.style.visibility = "visible";
    scrollBy( options: {top: 250, behavior: 'smooth'});
  }else{
    Kosher.style.visibility = "hidden";
    notKosher.style.visibility = "hidden";
    scrollBy( options: {top: -250, behavior: 'smooth'});
  }
}
```

```
document.getElementById( elementId: 'Kosher').onclick=function () {bubopen3( n: 2)};
document.getElementById( elementId: 'notKosher').onclick=function () {bubopen3( n: 1)};

function bubopen3(n){
  //if (n===1) document.getElementById('Kosher').onclick=null;
  // if (n===2) document.getElementById('notKosher').onclick=null;
  let elements3=document.querySelectorAll( selectors: '#sweety${n},#salty${n}');
  elements3.forEach( callbackfn: elem=> elem.style.visibility = "visible");
  scrollBy( options: {top: 250, behavior: 'smooth'});
}

document.getElementById( elementId: 'sweety1').onclick=function () {bubopen4( n: 1)};
document.getElementById( elementId: 'sweety2').onclick=function () {bubopen4( n: 2)};
document.getElementById( elementId: 'salty1').onclick=function () {bubopen5( n: 1)};
document.getElementById( elementId: 'salty2').onclick=function () {bubopen5( n: 2)};

function bubopen4(n){
  document.getElementById( elementId: 'salty1').onclick=null;
  document.getElementById( elementId: 'salty2').onclick=null;
  let elements4=document.querySelectorAll( selectors: '#healthy${n},#mushhat${n}');
  elements4.forEach( callbackfn: elem=> elem.style.visibility = "visible");
  console.log(elements4);
  scrollBy( options: {top: 250, behavior: 'smooth'});
}

function bubopen5(n){
  document.getElementById( elementId: 'sweety1').onclick=null;
  document.getElementById( elementId: 'sweety2').onclick=null;
  let elements5=document.querySelectorAll( selectors: '#milk${n},#meat${n},#fish${n}');
  elements5.forEach( callbackfn: elem=> elem.style.visibility = "visible");
  console.log(elements5);
  scrollBy( options: {top: 400, behavior: 'smooth'});
}
```

```
const activePage= window.location.pathname;
const activeView= document.querySelectorAll( selectors: 'ul a').forEach ( callbackfn: link => {
  if(link.href.includes( `${activePage}` )){
    link.classList.add("active");
  }
});
// console.log(activeView);
```

פונקציה לnavBar:

:CSS

חילקתי את העבודה ל CSS 2, כאשר הראשון עוסק בעיקר בשני העמודים הראשונים של כניסה והרשמה והשני בשאר העמודים. אם כי היכולת של כפתור "שלח" לסוגיו נטמע בכמה עמודים אחרי הכניסה.

** בנוסף יש את כל הקוד הקשור למפות של גוגל, כתוב גם JS היכן מופיע וגם ב CSS מופיע ב- Home.css כמעט בסוף, פשוט מדובר בהרבה קוד.