

My Burgers

מגישים: אדם הדרי ורועי עזרי.

נושא: אתר ייחודי להזמנת המבורגר בהרכבה עצמית.

מכללה: מכללת אורט סינגאלובסקי תל אביב.

מחלקה: מחלקת תוכנה.

מרצה: שי אדרי .

תוכן עניינים

עמוד 3 - הצהרת סטודנטים.

עמוד 4 - פרטי הסטודנטים.

עמוד 5 - תאור הפרויקט ורקע כללי.

עמוד 6 - מטרות המערכת , סקירת מצב השוק.

עמוד 7 - חידוש ושיפור, בעיות צפויות במהלך הפיתוח.

עמוד 8 - תאור כללי של השימוש באתר.

עמודים 9-14 - צילומי מסך של כל דפי האתר.

עמוד 15 - הטכנולוגיות בהם השתמשנו.

עמוד 16 - Site map .

עמודים 17-24 - צילומי מסך של הקומפוננטות מVSCODE.

עמוד 25 - סיכום ומסקנות, פיתוחים עתידיים, תודות.

הצהרת הסטודנטים

החתומים מטה, מצהירים בזאת כי כל פרויקט הגמר המוגש בספר זה, הינו פרי עבודתנו בלבד, על בסיס הנחייתו של המנחה ותוך הסתמכות על מקורות הידע והמידע האחרים המצויים בביבליוגרפיה המובאת בסיום ספר זה. אנחנו מודעים לאחריות שהננו מקבלים על עצמנו ע"י חתימתנו על הצהרה זאת, שכל הנאמר בה הינה אמת ורק אמת.

חתימת התלמידים

אישור המנחה
הריני מאשר הגשת ספר פרויקט הגמר להערכה.

חתימת המנחה

חתימת הסטודנטים

חתימת המנחה האישי

המכון הממשלתי להכשרה בטכנולוגיה ובמדע יחידת הפרויקטים חוזר מנהל מה"ט 11-4-51 – נספח מס' 10

תאריך: 30.7.2023

לכבוד יחידת הפרויקטים מה"ט הצעה לפרויקט גמר
פרטי הסטודנטים

ת.ז	שם הסטודנט	כתובת	טלפון נייד	תאריך סיום הלימודים
305601114	רועי עזרי	רותר וולטר 4 חולון	0542412474	1.7.2023
043196864	אדם הדרי	אוסטושינסקי 14 כפר סבא	0506686863	1.7.2023

שם המכללה : אורט סינגאלובסקי

סמל המכללה:

מגמת לימוד: תוכנה / stack full

מקום ביצוע הפרויקט: מכללה א.

פרטי המנחה האישי

שם המנחה	כתובת	טלפון נייד	תואר	מקום עבודה/תפקיד
שי אדרי	תל אביב	0546333715	תואר ראשון	אורט סינגאלובסקי

חתימת הסטודנט: _____

חתימת המנחה האישי: _____

חתימת הגורם המקצועי מטעם מה"ט : _____

שם הפרויקט : *My Burgers*

תיאור ורקע כללי

המבורגר נחשב לאחד ממאכלי המזון המהיר המוכרים ביותר בעולם. ענף ההמבורגרים מגלגל בישראל על פי ההערכות, כ- 2-2.5 מיליארד שקל, עם יותר מ-600 עסקים הפועלים בתחום. בארה"ב לבד, נאכלים קרוב ל-50 ביליון המבורגרים בשנה! כן כן, קראתם נכון! אמריקאים אוכלים קרוב ל-50 ביליון המבורגרים בשנה. זה לא סוד שהמאכל פופולרי במיוחד באמריקה. אם היינו לוקחים את כמות ההמבורגרים הזו ומסדרים אותה בקו ישר, היא הייתה יכולה להקיף את כדור הארץ במשך כ-32 פעמים! אז אם רק בארה"ב אוכלים מספר כזה של המבורגרים, תתארו לכם כמה נאכלים בעולם כולו?!

בעקבות הפופולריות הרבה והערצה הרבה למאכל הנחשק החלטנו לפתח פלטפורמה שתהפוך את תהליך הזמנת ההמבורגר לחוויה בלתי נשכחת!

מדובר באתר להזמנת המבורגר בהרכבה עצמית על פי פרמטרים שנקבעים על ידי הלקוח/המזמין כגון הוספת בשר/סלט/חמוצים.

מטרות המערכת

1. לאפשר ללקוח להזמין המבורגר בהרכבה עצמית על פי בחירתו .
2. להעניק רעיונות חדשים ומגוונים להרכבת המבורגר בצורה רדנומלית , בדרך כזאת שתאפשר ללקוח לנסות טעמים ושילובים חדשים שלא העלה על דעתו.
3. לספק חווית הזמנה נוחה ונגישה.
4. לחסוך את הזמן לקופאי או נציג השירות במתן מענה והצעות הזמנה ללקוח .

סקירת מצב קיים בשוק, הבעיות הקיימות

ענף ההמבורגרים עובר בשנתיים האחרונות תקופה מאוד דינאמית מבחינה שיווקית ופרסומית. ים של תקציבים לא הגיוניים יוצא מחברות ההמבורגרים למדיה. שנת '99 היא אפילו עוד יותר מטורפת מ-98.

בבורגר ראנץ' מחפשים את כיווני הצמיחה העתידיים. זו אחת הסיבות לכך שבורגר ראנץ' בחרה להקים רשת נוספת, "פרצלס גורמה", מותג אמריקני. בארה"ב על כל 6,000 תושבים יש מסעדת המבורגרים.

הביקוש הרב להמבורגרים מביא איתו תחרות רבת עוצמה והרשתות והמסעדות נמצאות במרדף מתמיד אחר הלקוח . בנוסף, אתרי אינטרנט להזמנת מזון מהיר נגישים לכל דורש , דבר המגדיל את התחרות ומכריח את הרשתות והמסעדות לחשוב על דרכים חדשות ויצירתיות להבאת לקוחות ושימורם.

חידוש ושיפור

החידוש של האתר שלנו הוא הנוחות, היעילות, והיצירתיות שבהליך הזמנת ההמבורגר. עם פלטפורמה ייחודית כמו שלנו בעלת עיצוב חדשני ומזמין, ניתן למשוך לקוחות בקלות וגם לשמרם להזמנות הבאות. באתר שלנו הלקוח בוחר את הרכבת ההמבורגר לפי דרישתו ולא נאלץ להזמין רק על פי המנות המוצעות. בנוסף האתר קל ונוח לשימוש, ומאפשר הזמנה מהירה של המבורגר ללא תשואלים מיותרים והצעות הזמנה נוספות כנהוג באתרים אחרים.

בעיות צפויות במהלך הפיתוח ופתרונות(תפעול, טכנולוגיה, עומס ועוד)

תיאור הבעיה: מטרת האתר להיות אטרקטיבי כמה שיותר ונצרך מחקר רב לחקור אחר הסגנונות השונים והאפשריים למשיכת לקוחות.

פתרונות אפשריים: מחקר אחר סגנונות עיצוב ושימוש בספריות בעלות אפשריות רבות כגון tailwind.

תיאור כללי של השימוש באתר

User

המשתמש צריך להירשם לאתר ובכך יקבל גישה להזמנת המבורגר בהרכבה עצמית.

דרישות פונקציונאליות User

תהליך ההרשמה פשוט וקל וכולל אך ורק השמת כתובת המייל ובחירת סיסמא.

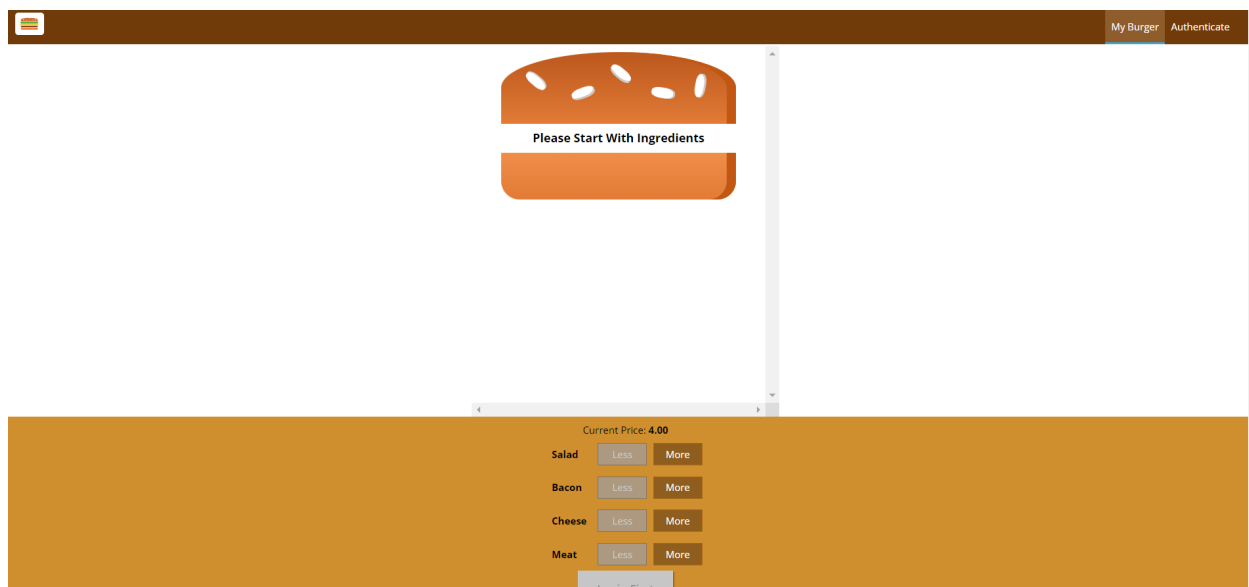
: Admin

ל ADMIN בדומה למערכת של user יהיה חובה להיות רשום למערכת בתור משתמש רגיל .

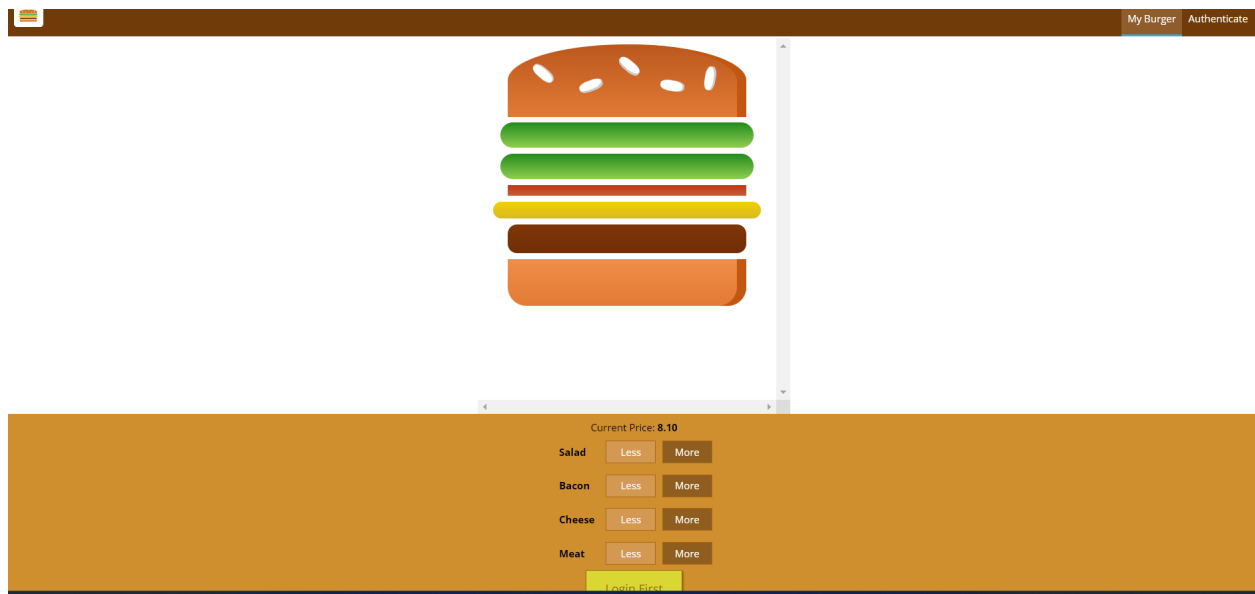
האדמין יוכל לצפות בכל המשתמשים הרשומים וכמו כן בהזמנות שביצעו.

צילומי מסך של כל דפי האתר

דף הבית



דף הבית לאחר הוספות להמבורגר



חלון הזמנה



Cancel

Payment

Fill Your Contact Data Information!

Your Name

Your Mail

Your Street

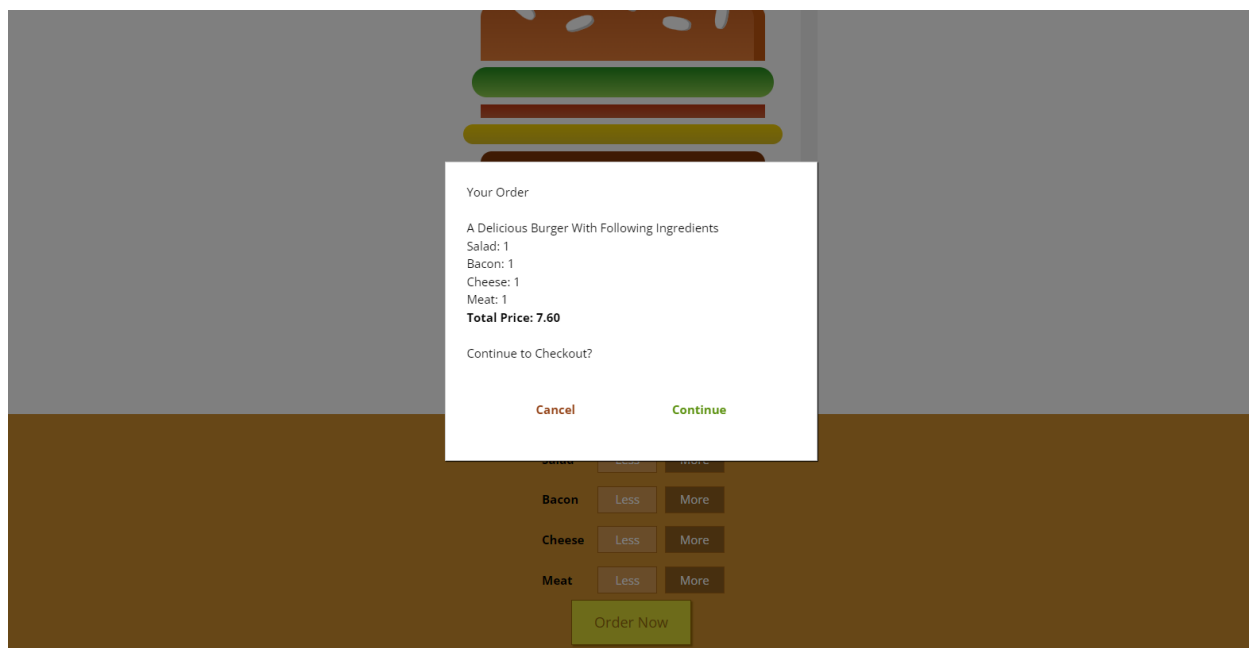
Your Country

Zip Code

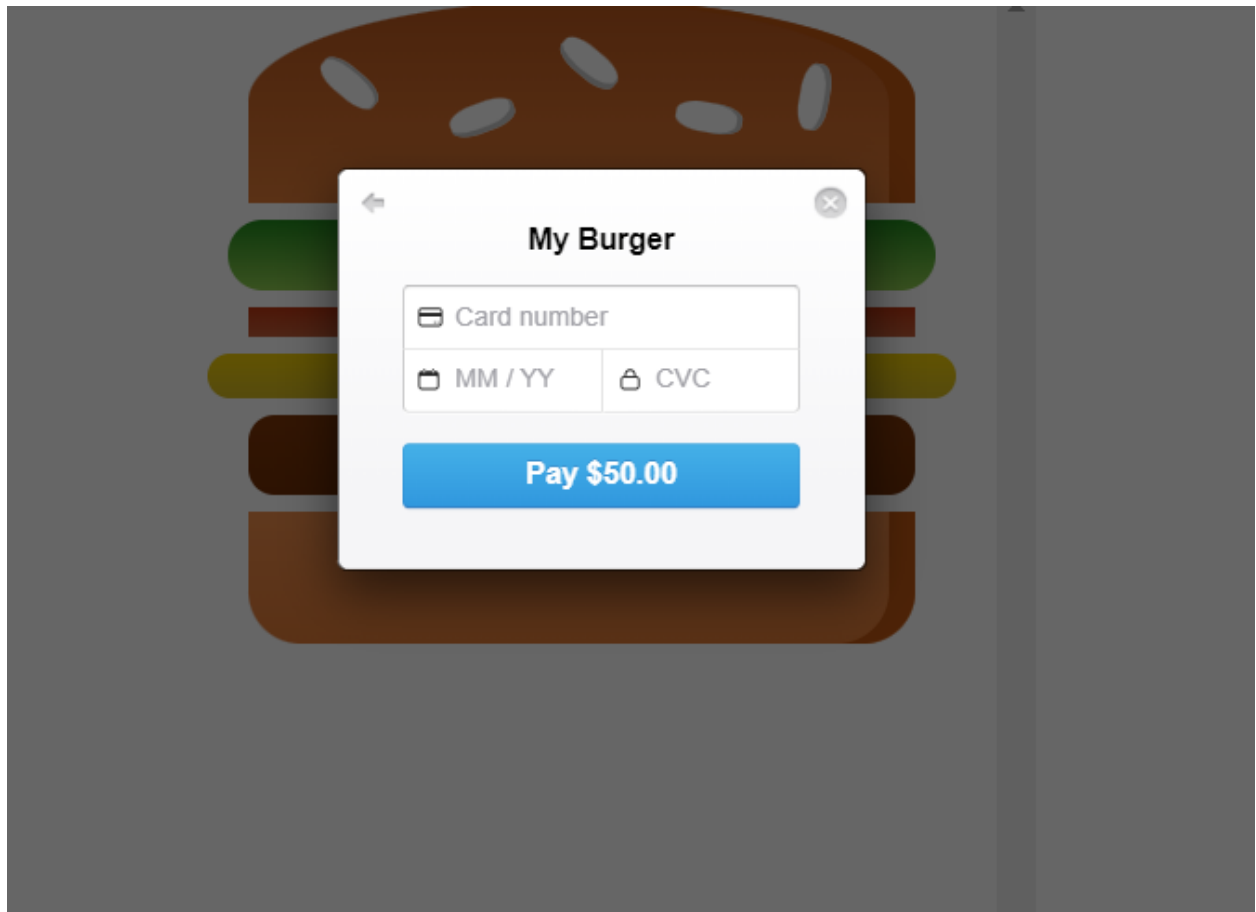
Fastest

ORDER

אישור הזמנה



חלון תשלום



הצעה רנדומלית

Generate,order,eat - our random menu is here.
It's simple, try it now!

I'M HUNGRY, GENERATE A BURGER!

Aloette Buger

Haute cuisine meets cravings for cheeseburgers.

ingredients:

6oz beef patty, served medium
griddled beaufort cheese
cheddar mayo
shredded lettuce
potato roll

Price: 5 USD

ORDER IT NOW

הטכנולוגיות בהן השתמשתי

React : React toastify , React stripe , React spinners

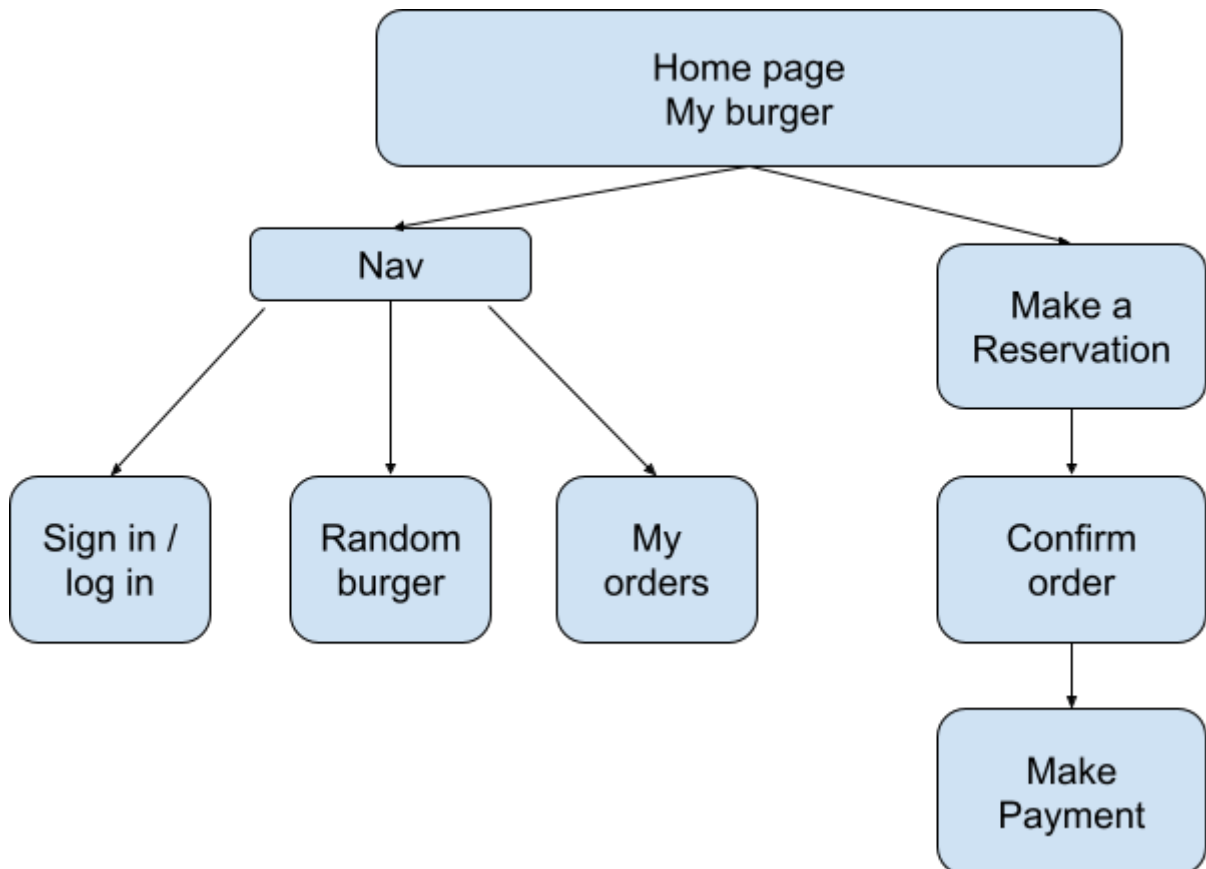
Redux

Tailwind

API : Stripe - לביצוע תשלום

להזמנת המבורגר רנדומלי - My burger api

Site Map



צילומי מסך של הקומפוננט מ VSCODE

BuildControls component

```
const BuildControls = (props) => {
  return (
    <div className={style.BuildControls}>
      <p>
        Current Price: <strong>{props.price.toFixed(2)}</strong>
      </p>
      {controls.map((Ctrl) => (
        <BuildControl
          key={Ctrl.label}
          label={Ctrl.label}
          added={() => props.addIngredient(Ctrl.type)}
          removed={() => props.removeIngredient(Ctrl.type)}
          disabled={props.disabled[Ctrl.type]}
        </>
      ))}
      <button
        className={style.OrderButton}
        disabled={!props.purchasable}
        onClick={props.ordered}
      >
        {props.isAuth ? "Order Now" : "Login First"}
      </button>
    </div>
  );
};

export default BuildControls;
```

BurgerIngredient component

```
const BurgerIngredient = props => {
  let ingredient = null;
  switch (props.type) {
    case "bread-bottom":
      ingredient = <div className={style.BreadBottom}></div>;
      break;
    case "bread-top":
      ingredient = (
        <div className={style.BreadTop}>
          <div className={style.Seeds1}></div>
          <div className={style.Seeds2}></div>
        </div>
      );
      break;
    case "meat":
      ingredient = <div className={style.Meat}></div>;
      break;
    case "cheese":
      ingredient = <div className={style.Cheese}></div>;
      break;
    case "bacon":
      ingredient = <div className={style.Bacon}></div>;
      break;
    case "salad":
      ingredient = <div className={style.Salad}></div>;
      break;
    default:
      ingredient = null;
  }
}
```

Ordersummary component

```
const OrderSummary = (props) => {
  const ingredients = Object.keys(props.ingredients).map((igKey) => {
    return (
      <li key={igKey}>
        <span style={{ textTransform: "capitalize" }}>{igKey}: </span>
        {props.ingredients[igKey]}
      </li>
    );
  });
  return (
    <Fragment>
      <h3 className="p-3">Your Order</h3>
      <div className="p-3">
        <p>A Delicious Burger With Following Ingredients</p>
        <ul>{ingredients}</ul>
        <p>
          <strong>Total Price: {props.price.toFixed(2)}</strong>
        </p>
      </div>
      <p className="p-3">Continue to Checkout?</p>
      <div className="flex justify-around w-[100%] p-10">
        <div>
          <Button btnType="Danger" clicked={props.cancelOrder}>
            Cancel
          </Button>
        </div>
        <div>
```

Burger component

```
const Burger = props => {
  let transformedIngredient = Object.keys(props.ingredient)
    .map(igKey => {
      return [...Array(props.ingredient[igKey]).map((_, i) => {
        return <BurgerIngredient key={igKey + i} type={igKey} />;
      })];
    })
    .reduce((arr, el) => {
      return arr.concat(el);
    }, []);
  if (transformedIngredient.length === 0) {
    transformedIngredient = <p>Please Start With Ingredients</p>;
  }
  return (
    <div className={style.Burger}>
      <BurgerIngredient type="bread-top" />
      {transformedIngredient}
      <BurgerIngredient type="bread-bottom" />
    </div>
  );
};

export default Burger;
```

NavigationItems component

```
const NavigationItems = (props) => {
  return (
    <ul className={styles.NavigationItems}>
      <NavigationItem link="/" exact>
        My Burger
      </NavigationItem>
      {props.isAuthenticated ? (
        <NavigationItem link="/random">Random Burger</NavigationItem>
      ) : null}
      {props.isAuthenticated ? (
        <NavigationItem link="/orders">Orders</NavigationItem>
      ) : null}
      {!props.isAuthenticated ? (
        <NavigationItem link="/auth">Authenticate</NavigationItem>
      ) : (
        <NavigationItem link="/logout">Logout</NavigationItem>
      )}
    </ul>
  );
};

export default NavigationItems;
```

Sidedrawer component

```
const SideDrawer = (props) => {
  let attachedClasses = [styles.SideDrawer, styles.Close];
  if (props.open) {
    attachedClasses = [styles.SideDrawer, styles.Open];
  }

  return (
    <Fragment>
      <Backdrop show={props.open} clicked={props.sideDrawerClosed} />
      <div className={attachedClasses.join(" ")}>
        <div className={styles.Logo}>
          <Logo />
        </div>

        <nav>
          <NavigationItems isAuth={props.isAuth} />
        </nav>
      </div>
    </Fragment>
  );
};

export default SideDrawer;
```

Toolbar component

```
const Toolbar = (props) => (  
  <header className={styles.Toolbar}>  
    <DrawerToggle clicked={props.DrawerToggleClick} />  
    <div className={styles.Logo}>  
      <Logo />  
    </div>  
    <nav className={styles.DesktopOnly}>  
      <NavigationItems isAuth={props.isAuth} />  
    </nav>  
  </header>  
>);  
  
export default Toolbar;
```

Order component

```
const Order = (props) => {
  const ingredients = [];
  for (let iName in props.ingredient) {
    ingredients.push({
      name: iName,
      amount: props.ingredient[iName],
    });
  }
  const outPutIngredients = ingredients.map((ig) => {
    return (
      <span
        style={{
          margin: "0 5px",
          padding: "5px",
          border: "1px solid #eee",
        }}
      >
        {ig.name} ({ig.amount})
      </span>
    );
  });
  return (
    <div className={styles.Order} key={props.keyItem}>
      <p>Ingredient: {outPutIngredients}</p>
      <p>
        Price: <strong>USD {props.price.toFixed(2)}</strong>
      </p>
    </div>
  );
};
```


סיכום ומסקנות

מטרתנו היתה לקחת רעיון נחמד וליישם אותו בצורה הטכנולוגית הטובה והנקייה ביותר , מה שהצריך אותנו לחקור אחר הדרכים הנכונות לכתיבת קוד נקי ומסודר. הליך הפיתוח היה מאד מרתק ומאתגר. נדרש מאיתנו לחקור אחר הדרכים להתחברות API חיצוניים ולדעת איך לעבוד איתם. ההליך דרש שעות עבודה רבות . למדנו שגם רעיון קטן ונראה לכאורה קל ליישום מצריך המון עבודה וידע מעמיק. הפרויקט הזה הוא אבן דרך ראשונה בתחילת דרכנו לשוק העבודה ובהחלט תרם לנו המון.

פיתוחים עתידיים

ישנם כמה רעיונות לקידום האתר:

1. לשפר את העיצוב ואת ההמבורגר המופיע בדף הבית.
2. לתת את האופציה ללקוח לכתוב ביקורת על הליך ההזמנה.
3. ליצור אפשרות חיבור עם גוגל.
4. להוסיף עגלת הזמנות כך שיהיה אפשר להזמין יותר המבורגרים.
5. לייצר כתובת.

תודות

תודה רבה למרצה היקר שלנו עופר שלי שליווה אותנו בכל הדרך ולימד אותנו את כל תחום הפולסטאק . תודה רבה לשי אדרי על כל העצות , התמיכה , והידע המקצועי שהעניק לנו . תודה רבה לכל צוות מחלקת תוכנה מכללת אורט סינגאלובסקי שנתנה לנו את כל הכלים להצליח.