

חלק ג' - קבוצה 23

איל ארקוש עמית פדאלי

על הפרויקט:

בפרויקט שלנו בחרנו להקים חנות נעליים אינטרנטית מקוונת אשר בה ניתן ליצור משתמש ולהתחבר לאתר, לחפש ולהוסיף מוצרים לעגלת הקניות, לבצע רכישה ואף לראות את ההזמנות שבוצעו לכל משתמש.

סדר פעולות:

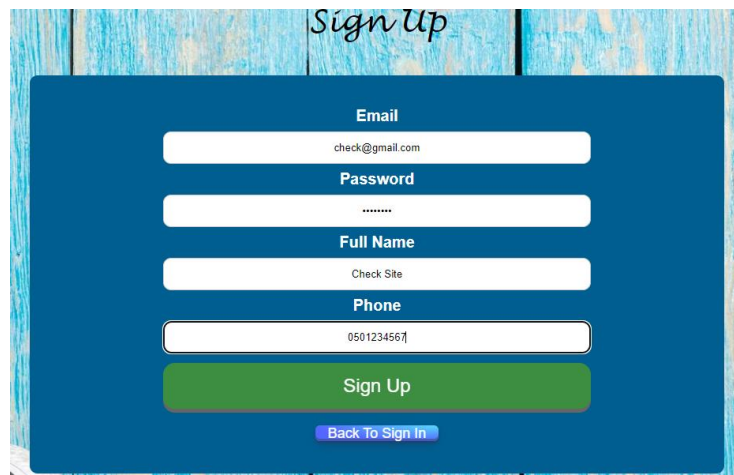
על מנת להקים את טבלאות הנתונים השונות באתר ולהוסיף להן מידע יש להיכנס לכתובת createDB/ולעקוב אחרי השלבים הממוספרים. (בסוף העבודה ניתן למחוק את הטבלאות של בסיס הנתונים באמצעות שלב 4).

Please click on the links below in order to CRUD the DB tables of this site.

1. [Click on the link to Create the DB tables](#)
2. [Click on the link Insert data to Users table](#)
3. [Click on the link Insert data to Products table](#)
4. [Click on the link to Drop the DB tables](#)

תהליך רישום משתמש:

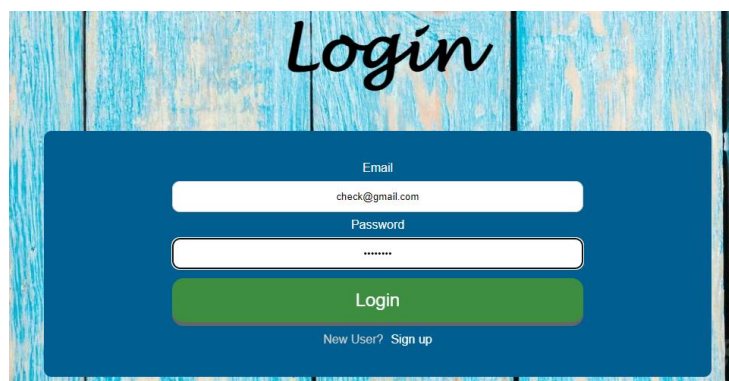
לאחר הקמת הטבלאות, יש לעבור לעמוד הבית "\". ולהירשם כמשתמש חדש לאתר בקישור:



The image shows a 'Sign Up' form with a blue background and a light blue border. The form contains the following fields and buttons:

- Email:** Input field with the text 'check@gmail.com'.
- Password:** Input field with masked characters '*****'.
- Full Name:** Input field with the text 'Check Site'.
- Phone:** Input field with the text '0501234567'.
- Sign Up:** A large green button.
- Back To Sign In:** A small blue button.

לאחר הרשמה, ניתן להזין את פרטי המשתמש ולהתחבר לאתר.



The image shows a 'Login' form with a blue background and a light blue border. The form contains the following fields and buttons:

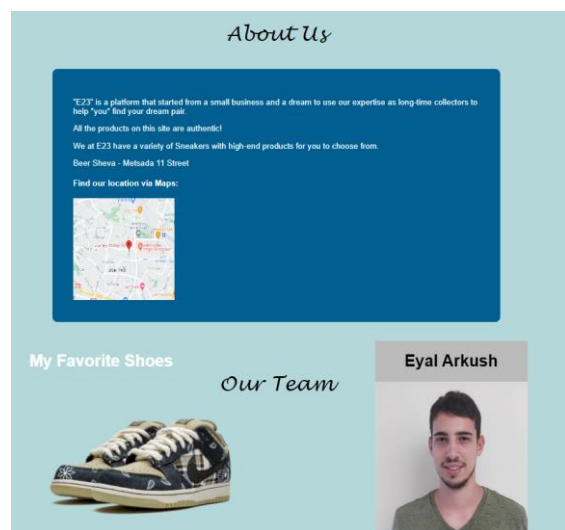
- Email:** Input field with the text 'check@gmail.com'.
- Password:** Input field with masked characters '*****'.
- Login:** A large green button.
- New User? Sign up:** A small blue button.

לאחר התחברות מועברים לעמוד הבית באתר:

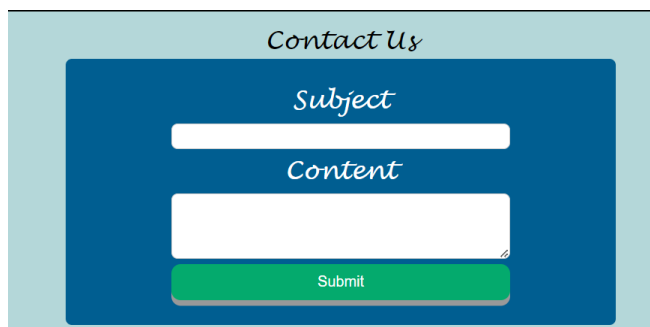


בעמוד הבית אפשרות לניווט באתר באמצעות Footer, Nav bar לעמודים השונים באתר. הפונקציות הפועלות בדף זה הן הצגת שם המשתמש המחובר באמצעות קוקיז (בכל עמודי האתר), פונקציה המראה תמונה של שמש/ירח בהתאם לשעה ביום, מצגת תמונות אוטומטית ואפשרות התנקות מהמשתמש המחובר לאתר.

הפונקציות המשמשות את עמוד זה נמצאות בקובץ home.js: Showslides1(),hidePhoto() :aboutUs: דף עם מידע אודות החברה שלנו.

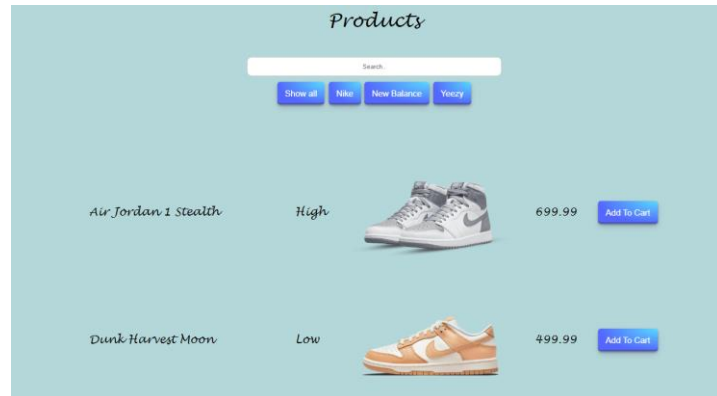


ContactUs: דף יצירת קשר אשר מכיל קישור ליצירת מייל שישלח אלינו.

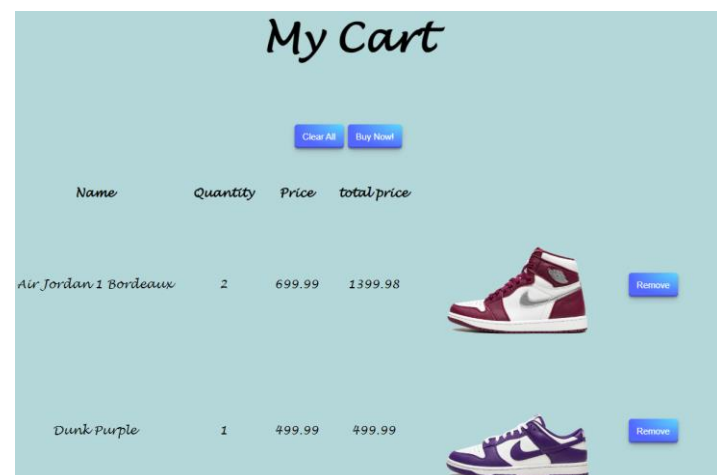


תהליך רכישה:

על מנת לבצע רכישה באתר יש להיכנס לדף פרודקטס. בדף זה ניתן לראות את מגוון הנעליים במלאי, קיימת אפשרות לסינון המוצרים לפי סוג המותג בלחיצה על הכפתורים השונים ואף לחפש מוצר מסוים לפי שם. בנוסף, ניתן ללחוץ על כפתור "הוספה" וכך מתווסף לעגלת המוצר.

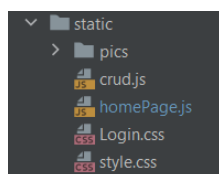
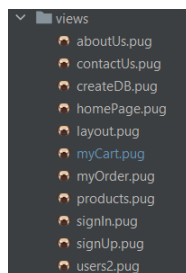


לאחר הוספת המוצרים, ניתן לעבור לעגלת הקניות בעמוד "מיקארט" ולצפות בפרטי ההזמנה הנוכחית. ניתן לראות את מספר היחידות מכל מוצר, מחיר הקניה של יחידה בודדת ומחיר סופי של המוצר. ניתן להסיר פריטים באמצעות כפתור "רימוב", להסיר את כל המוצרים מהעגלה באמצעות קליקאול ואף לקנות את המוצרים באמצעות מקש "בינאו".



בעת רכישת המוצרים מעבר לדף ההזמנות עם פרטי ההזמנות שבוצעו בעבר ע"י המשתמש.

Orders List		
Order	date	totalPrice
1	2023-07-04	1899.97



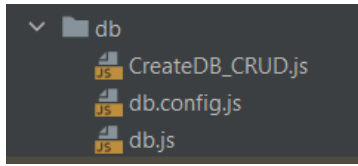
מבנה תיקיות והסבר לוגי:

תיקיית express server -

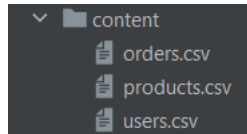
מכילה את קובץ השרת app.js, קובץ readme ואת חבילות הפאקג'. ובנוסף את התיקיות הבאות:

תיקיית views: תיקיית קבצי הפאג השונים באתר.

תיקיית static: תיקייה המכילה את הקבצים הסטטיים באתר - פונקציות JS ו-CSS. ב-crud.js אפליקציות CRUD לניהול האתר. ב-homePage.js פונקציות המופעלות בדף הבית. Login.css דף עיצוב רק לעמודי הרשמה והתחברות. style.css דף עיצוב לכל שאר האתר.



תיקית db: המכילה את קובץ בסיס הנתונים, קונפיגורציה ואת פונקציות ה-CRUD באתר. createDB_CRUD מכיל פונקציות ליצירת טבלאות ומחיקתן מהאתר. Db.config.js מכיל קונפיגורציה בסיס הנתונים. db.js מכיל את בסיס הנתונים והחיבור שלו.



תיקית content: המכילה את קבצי ה-csv באתר ממנה קוראים מידע שנכנס לטבלאות השונות בבסיס הנתונים.

app.js: האפליקציה שמכילה את הרוטים, ניתן להיכנס לעמודים השונים באתר ולהפעיל פונקציות שונות באתר ישירות מהכתובת. (קריאת צילומי מסך משמאל לימין)

```
//Link to CRUD the Data Base
Eyal Arkush
app.get('/createDB',(req,res)=>{
  res.render( view: 'createDB');
});

Eyal Arkush
app.get('/',(req,res)=>{
  res.render( view: 'signIn');
});

Eyal Arkush
app.get('/aboutUs',(req,res)=>{
  const loggedInUserName = req.cookies.name;
  res.render( view: 'aboutUs', options: { loggedInUserName });
});

Eyal Arkush
app.get('/contactUs',(req,res)=>{
  const loggedInUserName = req.cookies.name;
  res.render( view: 'contactUs', options: { loggedInUserName });
});
```

```
app.get('/homePage',(req,res)=>{
  const loggedInUserName = req.cookies.name;
  res.render( view: 'homePage', options: { loggedInUserName });
});

Eyal Arkush
app.get('/signIn',(req,res)=>{
  res.render( view: 'signIn');
});

Eyal Arkush
app.get('/signUp',(req,res)=>{
  res.render( view: 'signUp');
});

Eyal Arkush
app.get('/signOut',(req,res)=>{
  res.clearCookie( name: 'id');
  res.sendFile(path.join(__dirname,"views/signIn.html"));
});

app.get('/myOrder', crud.displayUserOrders)
app.get('/products', crud.displayProducts)
app.get('/myCart', crud.displayCart)
```

```
//drop tables in DB
app.get('/dropTables', CreateDB_CRUD.dropTables);

//All CRUD and functions in the site - Links for direct access
app.post( path: '/UserSignUp', crud.createNewUser);
app.post( path: '/signInUser', crud.userLogin);
app.get('/insertDataToUsers', crud.InsertDataToUsers);
app.get('/insertDataToOrders', crud.InsertDataToOrders);
app.get('/insertDataToProducts', crud.InsertDataToProducts);
app.get('/addToCart/:productId', crud.addToCart);
app.get('/addOrder', crud.addOrder);
app.get('/users2', crud.displayUsers)
app.get('/products55', crud.searchProduct)
app.get('/Nike', crud.filterNike)
app.get('/NewBalance', crud.filterNewBalance)
app.get('/Yeezy', crud.filterYeezy)
app.get('/All', crud.showAll)
app.get('/clearCart', crud.clearCart)
app.get('/removeFromCart/:cartId', crud.removeFromCart)

Eyal Arkush
app.listen(port, hostname,()=>{
  console.log("Server is running on port:",port);
});
```

```
const express = require('express');
const app = express();
const path = require('path');
const crud = require("../express_server/static/crud");
const port = 3000;
const bodyParser = require("body-parser");
app.use(bodyParser.json());
const cookieParser = require("cookie-parser");
app.use(cookieParser());
app.use(express.static(path.join(__dirname,"static")));
app.use(bodyParser.urlencoded({extended:true}));
app.set('views',path.join(__dirname,'views'));
app.set('view engine', 'pug');
const sql = require("../express_server/db/db");
const CreateDB_CRUD = require('./db/CreateDB_CRUD');
const csv = require('csvtojson');
```

אתחולים בדף app.js:

בסיס הנתונים:

פונקציות CRUD להקמת

- createTables() פונקציה ליצירת הטבלאות המשומשות באתר (users,orders,products,card). פירוט הטבלאות בצילומי המסך.
- dropTables() פונקציית למחיקת הטבלאות המשומשות באתר (users,orders,products,card).

```
CREATE TABLE IF NOT EXISTS products (
  productId VARCHAR(255) NOT NULL,
  name VARCHAR(255),
  fit VARCHAR(255),
  color1 VARCHAR(255),
  color2 VARCHAR(255),
  image1 VARCHAR(255),
  image2 VARCHAR(255),
  price decimal(10, 2),
  brand VARCHAR(255),
  PRIMARY KEY (productId),
  INDEX idx_productId (productId)
```

```
CREATE TABLE IF NOT EXISTS cart (
  cartId INT NOT NULL AUTO_INCREMENT,
  userEmail VARCHAR(255),
  productId VARCHAR(255),
  quantity INT DEFAULT 1,
  PRIMARY KEY (cartId),
  FOREIGN KEY (userEmail) REFERENCES users(email),
  FOREIGN KEY (productId) REFERENCES products(productId),
  INDEX fk_userEmail_idx (userEmail)
```

```
CREATE TABLE IF NOT EXISTS orders (
  orderId int NOT NULL AUTO_INCREMENT,
  date DATE,
  totalPrice decimal(10, 2),
  userEmail varchar(255),
  PRIMARY KEY (orderId, userEmail),
  FOREIGN KEY (userEmail) REFERENCES users(email) ON DELETE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS users (
  email varchar(255) NOT NULL,
  psw int(8),
  name VARCHAR(255),
  phone VARCHAR(255),
  PRIMARY KEY (email)
```

פונקציות CRUD לשימוש פונקציונלי באתר:

- **createNewUser()** – פונקציה ליצירת משתמש חדש באתר באמצעות שאילתה והכנסת המידע לטבלת המשתמשים. יש בדיקה האם המשתמש קיים כבר באתר לפי אימייל והכנסת המשתמש לבסיס הנתונים במידה ולא קיים. מתבצעת שמירת קוקיז על המשתמש לשימוש באתר.
- **userLogin()** – פונקציה להתחברות משתמש רשום באתר בדיקת התחברות באמצעות שאילתה הבודקת תאימות אימייל וסיסמה ושמירת קוקיז של פרטים אלה לשימוש באתר.
- **InsertDataToUsers(), InsertDataProducts(), InsertDataToOrders()** – פונקציות להכנסת מידע לתוך הטבלאות באמצעות קריאה מקובץ CSV ועדכון של בסיס הנתונים.
- **addToCart()** – פונקציה להוספת פריטים לעגלת הקניות של המשתמש (כפתור הוספה לעגלה בדף המוצרים). ווידוא ייחודיות לעגלה כדי ליצור עגלה אינדיבידואלית לכל משתמש באתר באמצעות בדיקה בשאילתה. במידה והפריט קיים בעגלה תתבצע רק הוספה של פריט נוסף לכמות ממנו quantity. עדכון בסיס הנתונים.
- **searchProduct()** – חיפוש מוצר בעמוד המוצרים לפי שם או חלק מהשם שלו. לאחר סינון יופיעו כל המוצרים המתאימים בדף המוצרים.
- **filterNike(), filterNewBalance(), filterYeezy()** – פונקציות לסינון מוצרים לפי שם המותג. בלחיצת כפתור בעמוד המוצרים יופיעו בעמוד רק המוצרים המתאימים למותג בהתאם לשם המותג.
- **showAll()** – בלחיצת כפתור בעמוד המוצרים ניתן להראות את כל המוצרים באתר (שימוש לאחר סינון לפי מותג או חיפוש לפי שם כאשר רוצים לחזור לראות את כל המוצרים). באמצעות שאילתה הבוחרת את כל המוצרים מטבלת המוצרים בבסיס הנתונים.
- **removeFromCart()** – פונקציה להסרת פריט מעגלת הקניות (כפתור remove בעמוד עגלת הקניות). ייחודי לכל פריט שהתווסף לעגלת הקניות. במקרה של פריט ממנו יש כמה יחידות אז רק הכמות תרד ביחידה אחת בעת לחיצה על הכפתור. עדכון בסיס הנתונים.
- **clearCart()** – פונקציה המנקה את כל הפריטים מעגלת הקניות בלחיצה על כפתור אחד. הכפתור ממוקם בעמוד עגלת הקניות.
- **addOrder()** – פונקציה המוסיפה הזמנה חדשה לבסיס הנתונים ואף מציגה אותו בדף ההזמנות שבוצעו. מומשה בכפתור buy now בעמוד עגלת הקניות.
- **displayCart(), displayUserOrders()** – פונקציות להצגת הזמנות, עגלת הקניות עבור כל משתמש באמצעות שאילתה שבודקת מטבלאות אלה היכן האימייל תואם לאימייל של

המשתמש ולאחר מכן מציגה לו את עגלת הקניות ואת ההזמנות, שמומש בעמודים של האתר.

- `displayProducts()` – פונקציה להצגת כל המוצרים באתר בעמוד המוצרים. באמצעות שאילתה הבוחרת את כל השורות מטבלת המוצרים
- `displayUsers()` – פונקציה להצגת כל המשתשים באתר, על מנת לבצע בדיקות ושימוש פנימי. פועלת באמצעות שאילתה הבוחרת את כל השורות מטבלת המשתמשים.