**פרויקט בהנדסת תוכנה**



שם בית הספר: תיכון שקמה דרכא

שם התלמיד: איתי זיידל

ת.ז. התלמיד: 328441241

שם המנחה: אלכס ברוייטמן

שם החלופה: תכנות אסינכרוני

תאריך הגשה: 14.06.2024

תוכן עניינים

[**מבוא** 3](#_Toc169254016)

[**הרקע לפרויקט:** 3](#_Toc169254017)

[**מטרות המערכת:** 3](#_Toc169254018)

[**תיאור המערכת:** 3](#_Toc169254019)

[**גבולות המערכת של כל לקוח:** 3](#_Toc169254020)

[**סביבת פיתוח:** 3](#_Toc169254021)

[**שפות התכנות:** 3](#_Toc169254022)

[**שכבות:** 4](#_Toc169254023)

[**פלטפורמות של הלקוח:** 4](#_Toc169254024)

[**אתגרים מרכזיים:** 4](#_Toc169254025)

[**חידושים:** 4](#_Toc169254026)

[**ניתוח מערכת** 5](#_Toc169254027)

[**מכתב יזום** 5](#_Toc169254028)

[**תיאור הארגון:** 5](#_Toc169254029)

[**הבעיות בארגון:** 5](#_Toc169254030)

[**מטרות:** 5](#_Toc169254031)

[**תיחום מערכת המידע:** 6](#_Toc169254032)

[**הנחות עתידיות:** 6](#_Toc169254033)

[**דיאגרמת ERD** 6](#_Toc169254034)

[**Use Case** 7](#_Toc169254035)

[**DFD** 8](#_Toc169254036)

[**Prototypes** 9](#_Toc169254037)

[**תרשים )DSD קשרי גומלין(** 12](#_Toc169254038)

[**טבלאות עיצוב ראשיות** 12](#_Toc169254039)

[**צד שרת** 14](#_Toc169254040)

[**המחלקות המטפלות בנתונים** 16](#_Toc169254041)

[**שכבת שירותי שרת** 17](#_Toc169254042)

[**שכבת BusinessLogic לוגיקה עסקית** 18](#_Toc169254043)

[**צד לקוח** 20](#_Toc169254044)

[**מדריך למשתמש** 24](#_Toc169254045)

[**רפלקציה** 25](#_Toc169254046)

# **מבוא**

Gym Online Club

## **הרקע לפרויקט:**

פרויקט זה הוא פרויקט העוסק במכירה של מוצרים וציוד הקשורים לחיי הכושר.

קהל היעד הוא בעצם כל אדם הרוצה ומעוניין לעסוק בפעילות ספורטיבית. הפרויקט יציע לאדם פתרונות רכישה של ציוד ותוספי תזונה המותאמים לאותו אדם ולמטרתו בעולם הכושר.

הסיבות לבחירת הנושא הן בעיקר החיבור האישי שלי לעולם הכושר והתזונה אשר מעניין אותי מאוד וצברתי במהלך השנים הרבה מידע עליו שיוכל לסייע לי בפרויקט זה. בנוסף אני חושב שעולם הכושר בזמן האחרון צובר תאוצה רבה ברחבי העולם ואני חושב שזה נושא שהרבה אנשים יתעניינו בו.

## **מטרות המערכת:**

מטרות המערכת הן בכלליות לתת למשתמש גישה למידע שנוגע לעולם הכושר. המערכת תיתן למשתמש גישה לחנות הכושר האינטרנטית בה הוא יוכל לקנות מצרכים / מכשירים שיעזרו לו להתקדם לעבר המטרה שלו.

## **תיאור המערכת:**

המערכת תשתמש במאגרים שונים שיספקו למשתמש מענה אישי לבקשות שלו. המערכת תהפוך את כל עניין האימונים והתזונה לנוחים יותר למשתמש ותעזור לו להגיע למטרה שלו בצורה הכי טובה ונוחה שאפשר. על מנת לממש זאת, המערכת תעסוק במכירה של מכשירי כושר ומוצרים הקשורים לעולם הכושר למען קידום התהליך של המשתמש.

## **גבולות המערכת של כל לקוח:**

* המערכת לא תתעסק בתשלומים
* המשתמש המזדמן לא יתעסק בעריכה אישית
* המשתמש המזדמן לא יוכל לרכוש ציוד
* המנהל לא יתעסק בנתונים של המשתמשים

## **סביבת פיתוח:**

סביבת הפיתוח שאני משתמש בה לטובת הפרויקט שלי היא visual studio.

## **שפות התכנות:**

בפרויקט שלי אני משתמש בשפות התכנות הבאות: c#, html, JavaScript.

## **שכבות:**

1. שכבת תצוגה - (Presentation Layer)שכבת התצוגה אחראית על קלט ופלט. שכבה זאת יחידה שמתקשרת עם משתמש. לשכבה זאת אחריות לאימות הנתונים –

בדיקת ערכי הקלט שמכניס משתמש.

1. שכבת הלוגיקה (Business Logic Layer) שכבה זו שולטת בפונקציונליות של היישום. באחריותה: קבלת החלטות, ביצוע חישובים, עיבוד ערכי הקלט,

איסוף נתונים לפלט.

1. שכבת גישה לנתונים (Data Access Layer) - שכבה זו מעבירה נתונים בין יישום ומקור נתונים.
2. שכבת הנתונים - (Data Layer)שכבה זו מורכבת משרת בסיס נתונים אחד או יותר. מכאן המידע נשלף ומאוחסן. באחריותה: לשמור, לעבד, לארגן, למיין, לסננן, לבצע סיכומים.

## **פלטפורמות של הלקוח:**

מנהל – המנהל הוא עובד Windows ולכן המנהל יעבוד דרך מחשב. פלטפורמה זו תהיה הכי רלוונטית ונוחה עבורו.

משתמש רשום – המשתמש הרשום ישתמש בפלטפורמת ה web -. פלטפורמה זו תקנה למשתמש את הגישה הנוחה ביותר להסתכל על המוצרים שמציעה לו החנות האינטרנטית. פלטפורמה זו היא המתאימה והנוחה ביותר ולכן בחרתי בה.

משתמש מזדמן – המשתמש המזדמן גם כן ישתמש ב web. המשתמש המזדמן יוכל רק לעיין במידע המופיע במערכת ובחנות שהמערכת מציעה. פלטפורמה זו תהיה הכי נוחה עבורו.

## **אתגרים מרכזיים:**

בעיה בתקשורת בין מודולים: נתקלתי בקושי בתקשורת בין חלקים שונים של המערכת.

בעיות ביצועים: חלק מהפעולות לקחו זמן רב מדי.

ניהול גרסאות: היה קושי בניהול הגרסאות השונות של הקוד.

דרכי הפתרון:

תקשורת בין מודולים: השתמשתי ב-API ברור ומסודר כדי לאפשר תקשורת חלקה בין מודולים.

ביצועים: שיפרתי את הקוד כדי לשפר את הביצועים.

# **ניתוח מערכת**

## **מכתב יזום**

### **תיאור הארגון:**

הסבר קצר –

הארגון הוא חדר כושר אינטרנטי הנקרא "Gym Online Club", ארגון זה נותן מענה לכל מיני בקשות שונות הקשורות בעולם הכושר. הארגון יכלול מכשירי כושר למכירה אשר ייתן מענה למשתמשים לפי המטרות שלהם.

מטרות המערכת –

- לשרת בצורה הטובה ביותר את המשתמש על מנת שיוכל להתקדם כמה שיותר בעולם הכושר.

- מכירה של מכשירי כושר שונים.

היקף פעילות –

היקף הפעילות של הארגון יהיה 24/7. מבנה הארגון יהיה אני (admin) שאטפל בלקוחות המעוניינים ובהזמנות שהם מבצעים.

מבנה הארגון –

הארגון בנוי משלושה סוגי משתמשים: מנהל מערכת, משתמש רשום ומשתמש מזדמן. מטרת המנהל היא לדאוג לתפעול הארגון, יצירה והוספה של מוצרים חדשים וטיפול בלקוחות. משתמש רשום יכול לרכוש מוצרים. משתמש מזדמן יכול לצפות במוצרים שניתן לרכוש אך לא יכול לרכוש אותם.

### **הבעיות בארגון:**

לקהילת הכושר יש מספר בעיות:

* לא קיים קטלוג של מכשירי כושר מספיק נוח לרכישת מכשירי כושר שונים, ואין דרך מספיק נוחה להציג למשתמש את המוצרים שהארגון מציע.
* אין רישום של משתמשים רשומים למערכת ולמערכת אין מידע על האנשים הרשומים אליה.
* אין מקום שאוסף את המידע ומנגיש למשתמש את המידע הרלוונטי עבורו.

### **מטרות:**

למערכת המידע העתידית של חדר הכושר האינטרנטי יהיו המטרות והיעדים הבאים:

מכירת ציוד כושר:

* הצגה של מוצרים קיימים
* חיפוש נוח של מוצרים שונים
* מכירה של ציוד

ניהול המערכת:

* קבלת מידע על משתמשים רשומים
* ניהול רשימה של מוצרים
* מיון קטגוריות של מוצרים
* מיון חברות של המוצרים

### **תיחום מערכת המידע:**

מערכת המידע של חדר הכושר האינטרנטי תטפל ב:

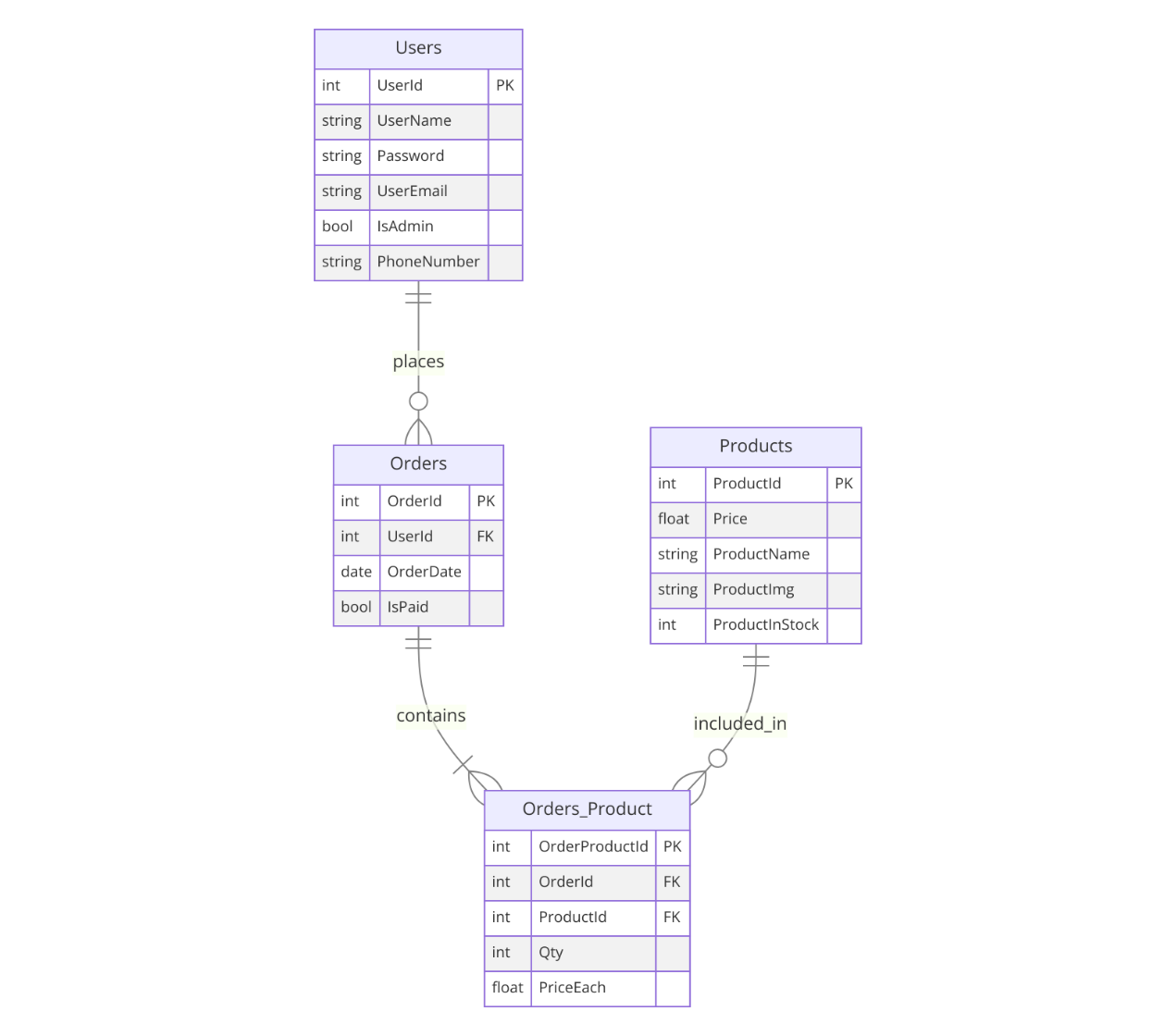
* מכירה של ציוד כושר וכל מיני מכשירים למיניהם

המערכת לא תטפל במעקב אחר ההתקדמות של המשתמש.

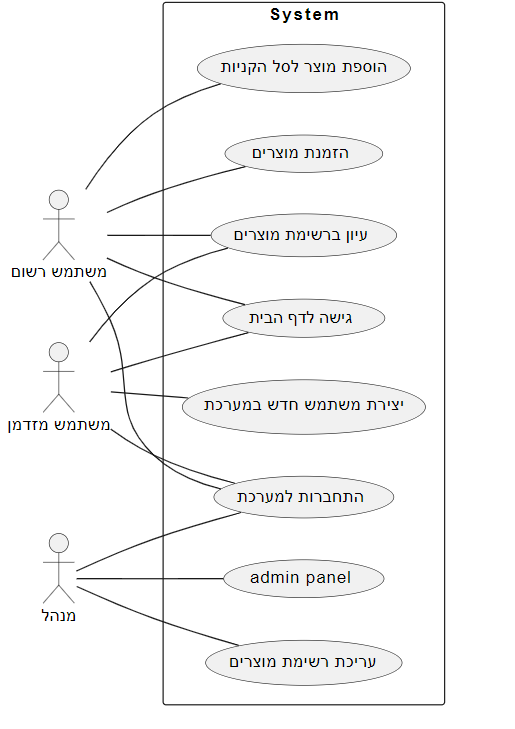
### **הנחות עתידיות:**

* המערכת תעזור לחדרי כושר שונים לתת עוד שירותים מגוונים ללקוחותיהם
* קהל המערכת העתידי יתרחב לאנשים העוסקים בספורט ספציפי ולא לכושר בכללי

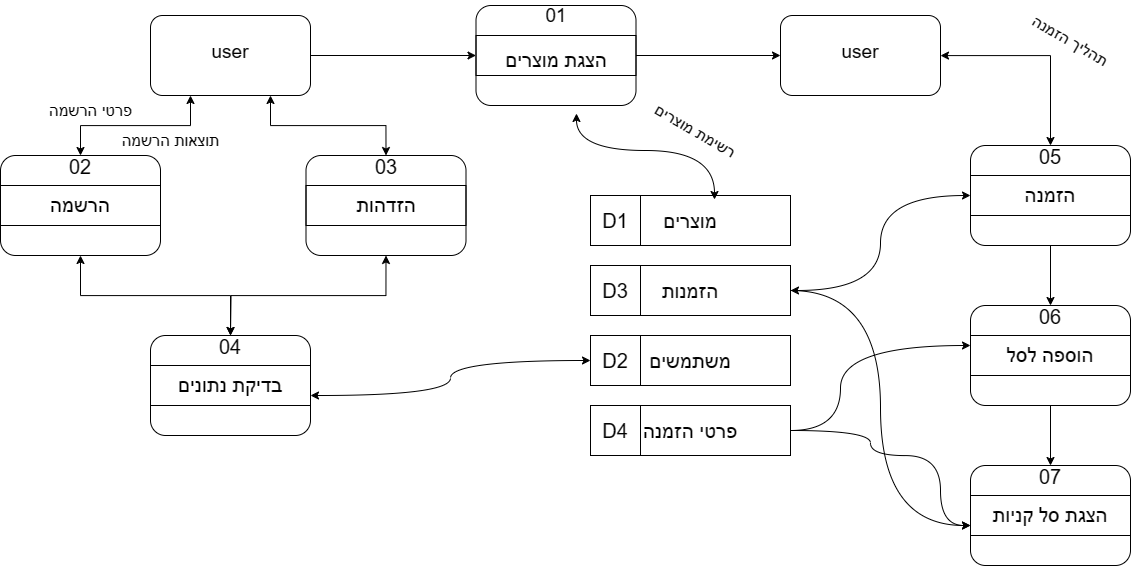
## **דיאגרמת ERD**

****

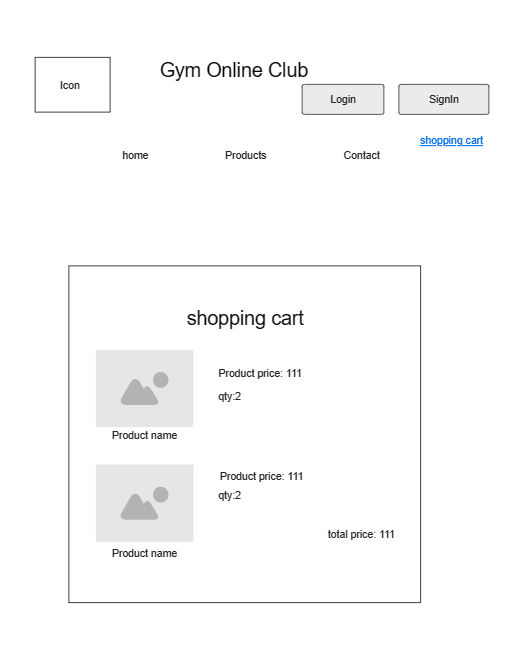
## **Use Case**

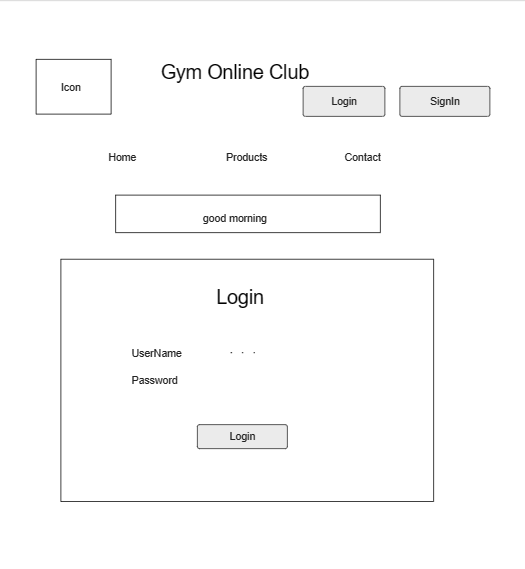
****

## **DFD**



# **Prototypes**

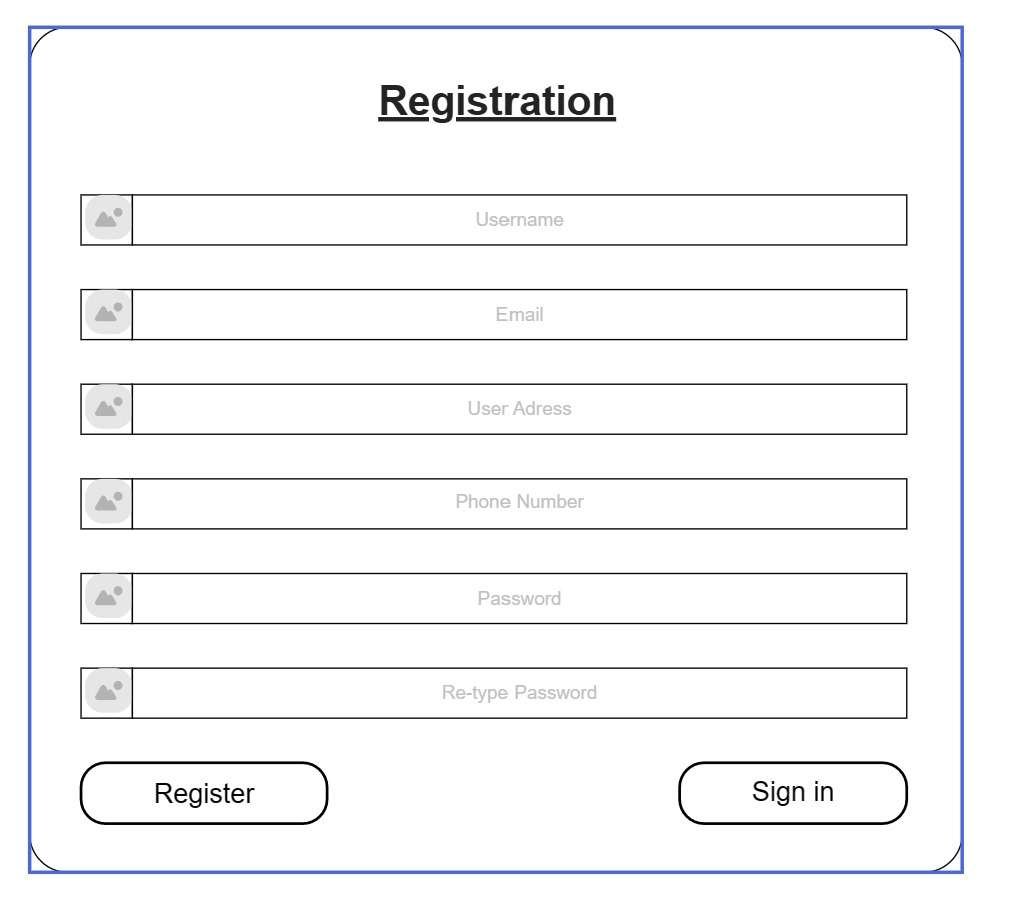
****

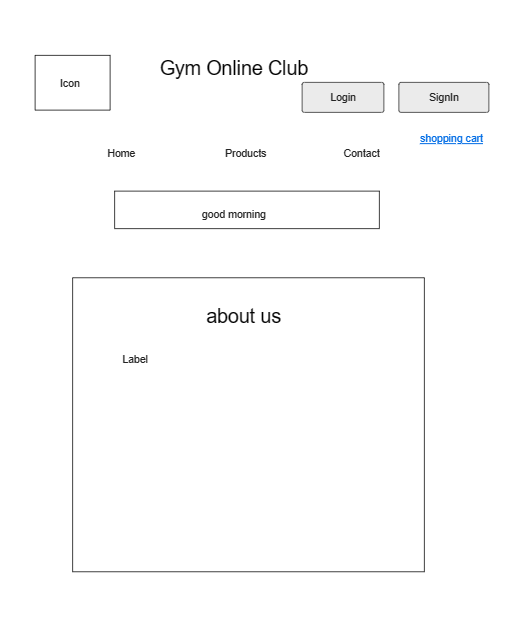
****

**A screenshot of a web page

Description automatically generated**

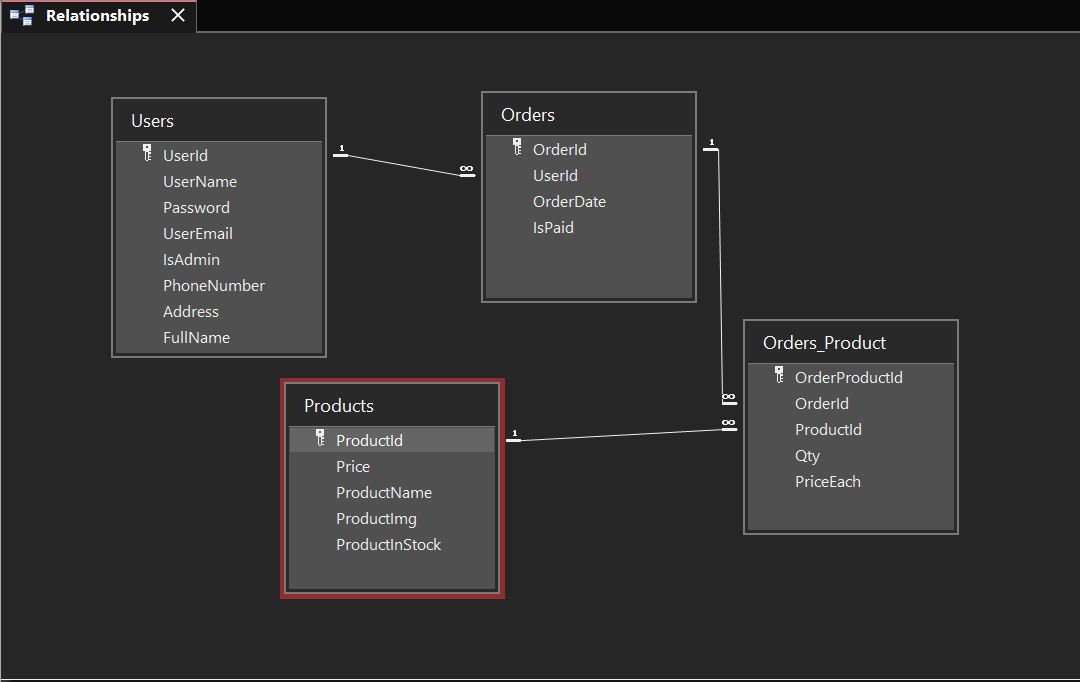


****

****

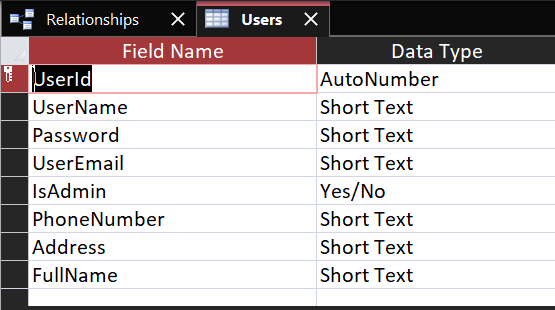
**בסיס הנתונים Data Base**

# **תרשים )DSD קשרי גומלין(**

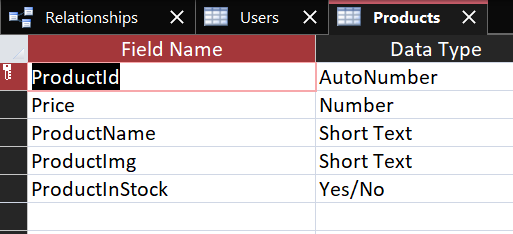
****

# **טבלאות עיצוב ראשיות**

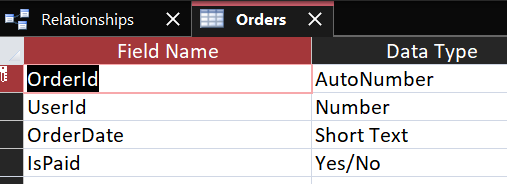
**Users**

****

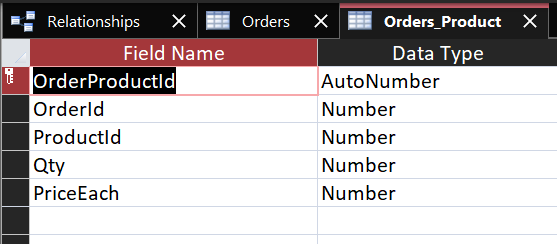
**Products**

****

**Orders**

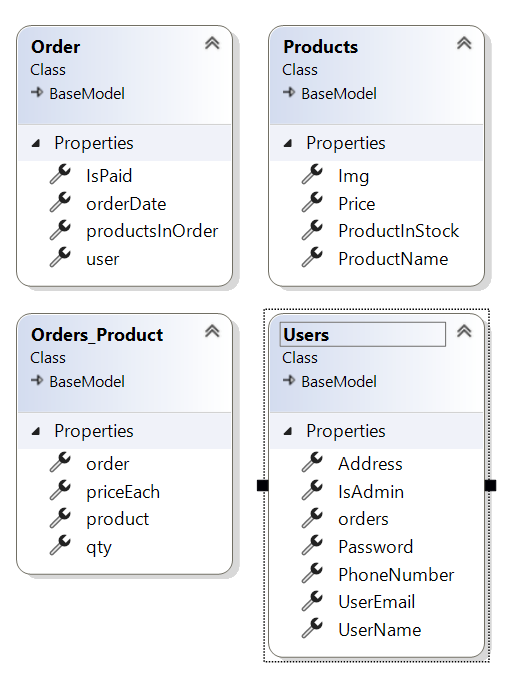
****

**Orders\_Product**

****

# **צד שרת**

**תרשים UML**

****

**קוד של מחלקות המודלים**

**Users**

public class Users: BaseModel

{

public string UserName { get; set; }

public string Password { get; set; }

public string UserEmail { get; set; }

public string PhoneNumber { get; set; }

public string Address { get; set; }

//public string FullName { get; set; }

public bool IsAdmin { get; set; }

public List<Order> orders { get; set; }

}

**Products**

public class Products: BaseModel

{

public string ProductName { get; set; }

public int Price { get; set; }

public string Img { get; set; }

public bool ProductInStock { get; set; }

}

**Order**

public class Order: BaseModel

{

public Users user { get; set; }

public string orderDate { get; set; }

public bool IsPaid { get; set; }

public List<Orders\_Product> productsInOrder { get; set; }

}

**Orders\_Products**

public class Orders\_Product: BaseModel

{

public Order order { get; set; }

public Products product { get; set; }

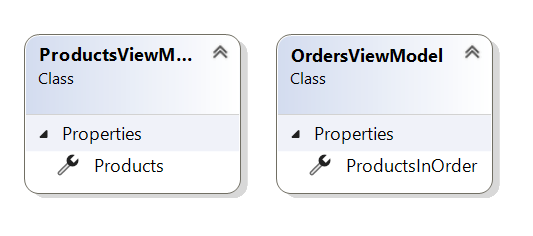
public int qty { get; set; }

public int priceEach { get; set; }

}

# **המחלקות המטפלות בנתונים**

**תרשים UML**

****

**הקוד של מחלקות ה view models**

**OrdersViewModel**

public class OrdersViewModel

{

public List<Orders\_Product> ProductsInOrder { get; set; }

}

**ProductsViewModel**

public class ProductsViewModel

{

public List<Products> Products { get; set; }

}

**הליך אסינכרוני**

תכנות אסינכרוני (Asynchronous Programming) הוא גישה לפיתוח תוכנה שמאפשרת ביצוע פעולות במקביל לפעולות אחרות מבלי לחכות להשלמתן לפני המשך ביצוע קוד אחר. זה חשוב במיוחד במקרים שבהם יש צורך לבצע פעולות שעשויות לקחת זמן רב, כמו קריאה או כתיבה לקבצים, גישה לרשת, או ביצוע חישובים מורכבים.

להלן דוגמה:

public async Task<T> HttpGetAsync()

{

{

this.requestMessage.Method = HttpMethod.Get;

string uri = $"http://{this.server}/api/{this.controller}/{this.method}";

if (this.requestParameters.Count() > 0)

{

uri += "/?";

foreach (KeyValuePair<string, string> parameter in this.requestParameters)

{

uri += $"{parameter.Key}={parameter.Value}&";

}

}

this.requestMessage.RequestUri = new Uri(uri);

this.responseMessage = await this.client.SendAsync(this.requestMessage);

if (this.responseMessage.IsSuccessStatusCode == true)

{

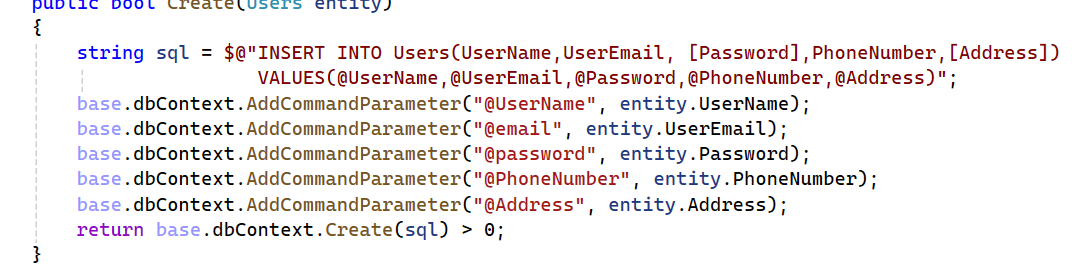
return await this.responseMessage.Content.ReadAsAsync<T>();

}

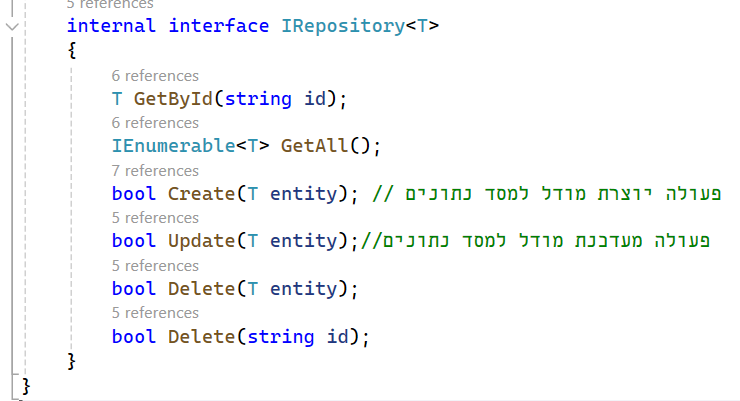
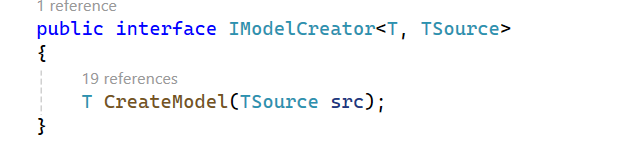
return default(T);

}

**הגנה מ sql injection**

השתמשנו בפרמטרים ב repository על מנת שהלקוח לא יוכל לגשת למסד נתונים ולתכונות שלו. להלן דוגמה: 

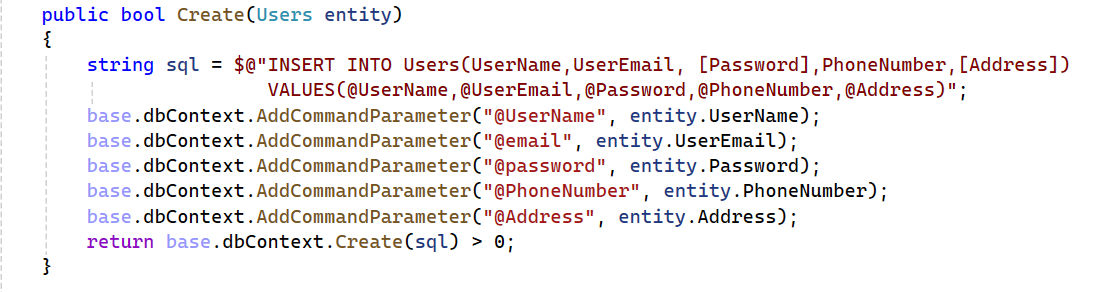
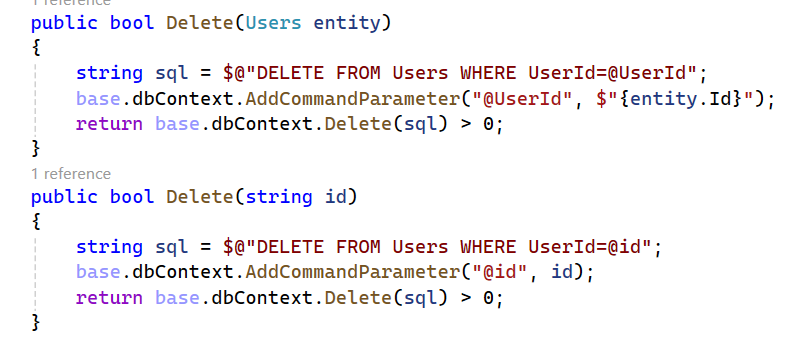
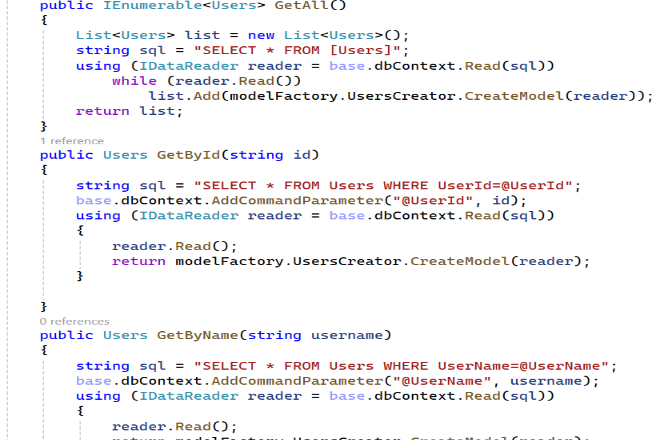
# **שכבת שירותי שרת**

******A screen shot of a computer code

Description automatically generatedInterface**

**מימוש**

****הדוגמאות האלה הן דוגמאות למימוש ב repository של users. בכל שאר הריפוזיטורים ישנן פעולות דומות

****

# **שכבת BusinessLogic לוגיקה עסקית**

ה-controllers המוצגים בתמונות הם חלק ממערכת ASP.NET MVC (Model-View-Controller), המהווה תבנית עיצוב לארגון קוד באפליקציות ווב. כל controller אחראי על ניהול הבקשות והתגובות בהתאם לנתיב ה-URL שמתקבל מהמשתמש, ועל אינטראקציה עם מודלים ותצוגות.

ProductsController

תיאור כללי:

ה-ProductsController מטפל בבקשות שקשורות למוצרים. הוא כולל את הפעולות הבאות: A screen shot of a computer program

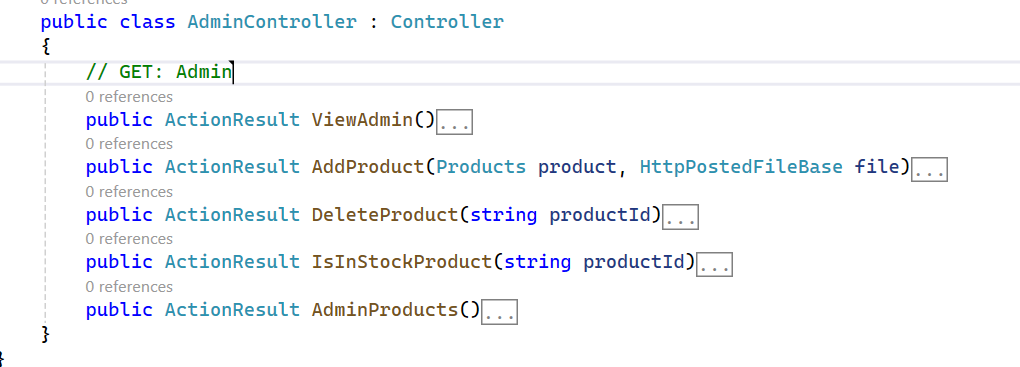
Description automatically generated

ViewProducts: פעולה שמחזירה את רשימת כל המוצרים לצפייה.

Get Product: פעולה שמחזירה את פרטי מוצר ספציפי לפי מזהה המוצר (productId).

AdminController

תיאור כללי:

ה-AdminController מטפל בבקשות ניהול מוצרים, ומתאים למשתמשי אדמין בלבד. הוא כולל את הפעולות הבאות: 

ViewAdmin: פעולה שמציגה את דף האדמין.

AddProduct: פעולה להוספת מוצר חדש, מקבלת אובייקט מוצר וקובץ (כמו תמונה).

DeleteProduct: פעולה למחיקת מוצר לפי מזהה המוצר (productId).

IsInStockProduct: פעולה לבדיקת זמינות מוצר במלאי לפי מזהה המוצר (productId).

AdminProducts: פעולה שמציגה את כל המוצרים המנוהלים על ידי האדמין.

UserController

תיאור כללי:

ה-UserController מטפל בבקשות שקשורות למשתמשים. הוא כולל את הפעולות הבאות: A screenshot of a computer program

Description automatically generated

ViewSignIn: פעולה שמציגה את דף ההתחברות.

AddUser: פעולה להוספת משתמש חדש, מקבלת אובייקט משתמש.

ViewLogIn: פעולה שמציגה את דף ההתחברות.

LogIn: פעולה שמבצעת התחברות של משתמש לפי שם משתמש וסיסמה. מסומנת כ-[HttpPost] כדי לציין שהיא מבוצעת כבקשת POST.

LogOut: פעולה שמבצעת יציאה מהמערכת.

סיכום כללי

כל אחד מה-controllers מתמקד בתחום אחר של המערכת:

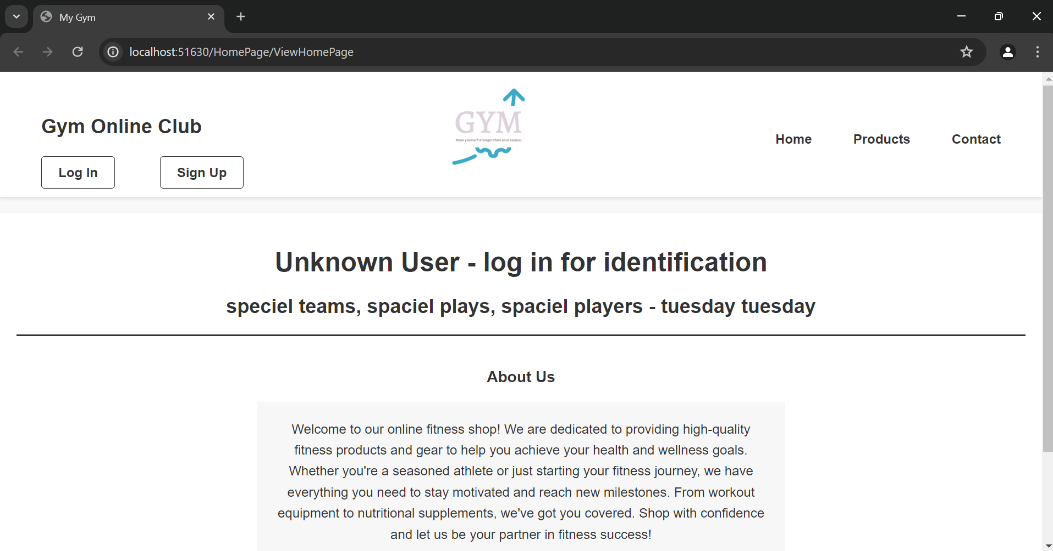
ProductsController: מנהל את המוצרים.

AdminController: מנהל פעולות אדמיניסטרטיביות על מוצרים.

UserController: מנהל את ההתחברות וההרשמה של משתמשים.

הקונטרולרים מכילים פעולות שונות (actions) שמטפלות בבקשות HTTP, ומחזירות תוצאות המתאימות לבקשות אלו.

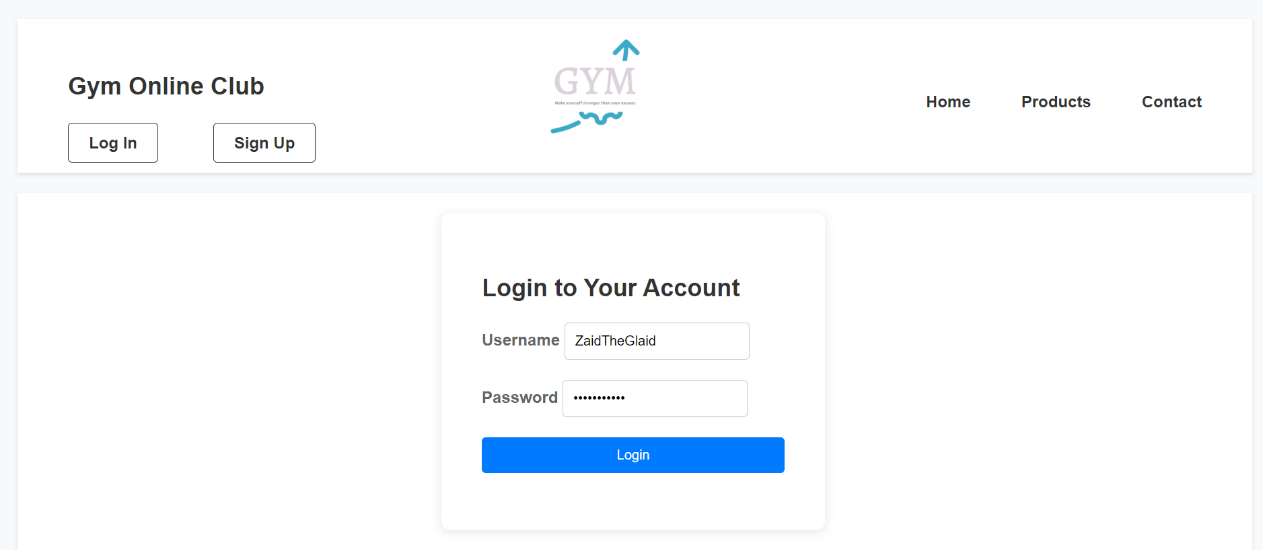
# **צד לקוח**

דף ראשי (Home Page)

תיאור: הדף הראשי של האפליקציה שמכיל קישורים לדפים אחרים.

משתמשים: כל המשתמשים.

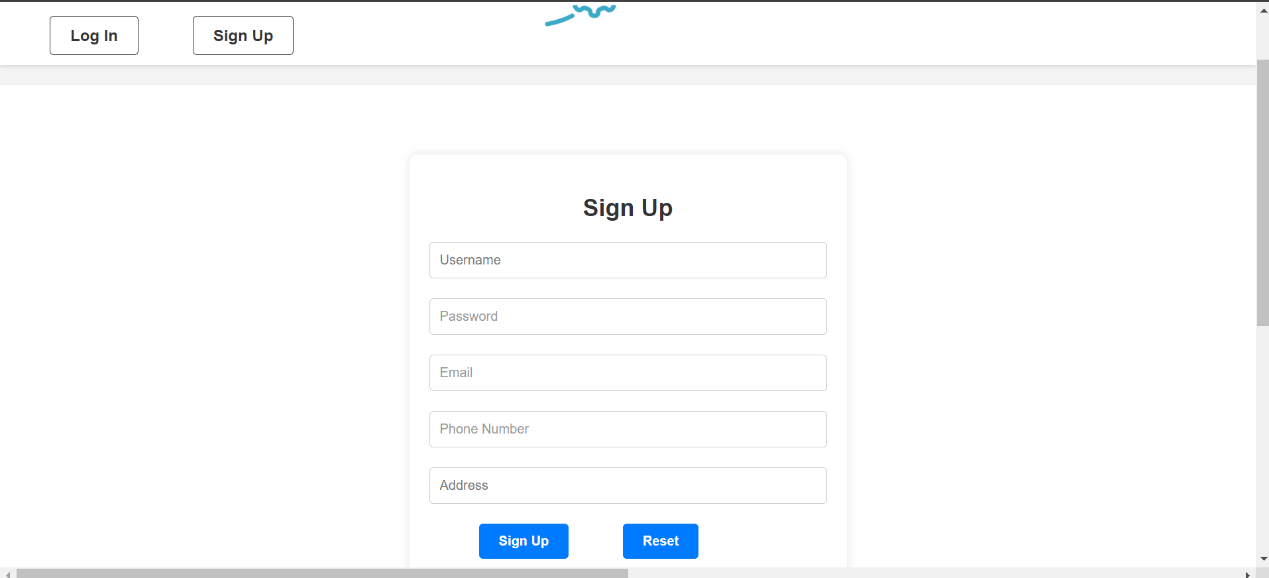
אלמנטים: תפריט ניווט, ברוכים הבאים, קישורים לדפים אחרים כמו מוצרים, התחברות, הרשמה וכו'.

דף התחברות (Login Page)

תיאור: דף שמאפשר למשתמשים להתחבר למערכת.

משתמשים: כל המשתמשים.

אלמנטים: שדות קלט לשם משתמש וסיסמה, כפתור התחברות.

דף הרשמה (Sign-Up Page)

תיאור: דף שמאפשר למשתמשים חדשים להירשם למערכת.

משתמשים: משתמשים לא רשומים.

אלמנטים: שדות קלט לשם משתמש, סיסמה, אימות סיסמה, דוא"ל, כפתור הרשמה ועוד.

דף מוצרים (Products Page)

A screenshot of a gym

Description automatically generated

תיאור: דף שמציג את רשימת המוצרים.

משתמשים: כל המשתמשים.

אלמנטים: רשימת מוצרים.

דף מוצר ספציפי (Product Details Page)

A screenshot of a gym

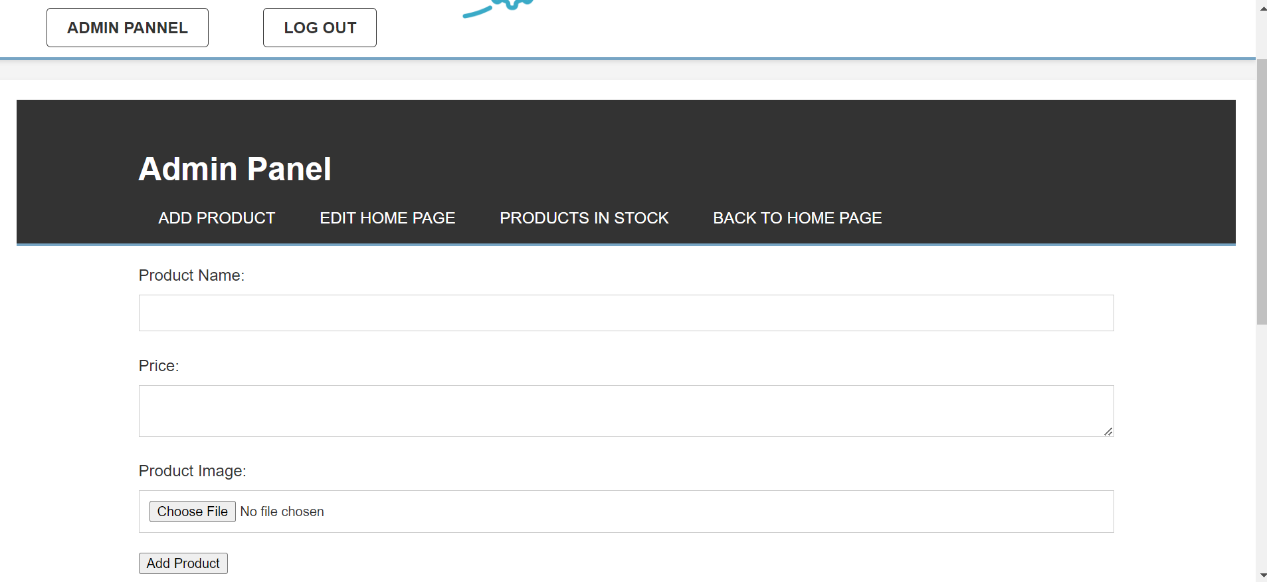
Description automatically generated

תיאור: דף שמציג פרטים על מוצר ספציפי.

משתמשים: כל המשתמשים.

אלמנטים: פרטי מוצר, תמונה, תיאור, כפתור הוספה לעגלה.

דף אדמין (Admin Page)



תיאור: דף ניהול שמאפשר לאדמינים לנהל מוצרים.

משתמשים: אדמינים בלבד.

אלמנטים: כפתורים לעדכון זמינות מוצר.

דף contact

A screenshot of a contact us

Description automatically generated

דף שמאפשר לכאורה לשלוח בקשה למערכת למטרת יצירת קשר

סיכום

הממשקים הללו מאפשרים תמיכה בשני סוגי משתמשים עיקריים: משתמשים רגילים ואדמינים. כל ממשק נבנה בצורה נוחה וקריאה, תוך שימוש בבדיקת תקינות קלט ואירועים מתאימים. תכנון ויישום הממשקים נעשו כדי להבטיח חווית משתמש טובה ויעילה.

# **מדריך למשתמש**

מדריך משתמש - משתמש מזדמן ומשתמש רשום

משתמש מזדמן:

יצירת משתמש חדש:

נווט לדף ההרשמה על ידי לחיצה על הקישור "הרשמה" בתפריט הניווט.

מלא את הפרטים הנדרשים.

לחץ על כפתור "הרשמה".

לאחר ההרשמה המוצלחת, תוכל להתחבר באמצעות שם המשתמש והסיסמה שהזנת.

עיון בדף הבית:

נווט לדף הבית על ידי לחיצה על הקישור "דף הבית" בתפריט הניווט.

בדף הבית תמצא מידע כללי על האתר וקישורים לדפים נוספים.

עיון בדף יצירת קשר (Contact):

נווט לדף יצירת הקשר על ידי לחיצה על הקישור "צור קשר" בתפריט הניווט.

בדף זה תמצא טופס ליצירת קשר עם האתר, כולל שדות להזנת שם, דוא"ל והודעה.

עיון בדף המוצרים (Products):

נווט לדף המוצרים על ידי לחיצה על הקישור "מוצרים" בתפריט הניווט.

בדף זה תוכל לעיין ברשימת המוצרים הקיימים באתר, כולל תמונות, תיאורים ומחירים.

משתמש רשום:

התחברות למערכת:

נווט לדף ההתחברות על ידי לחיצה על הקישור "התחברות" בתפריט הניווט.

מלא את שם המשתמש והסיסמה שלך ולחץ על כפתור "התחבר".

לאחר התחברות מוצלחת תוכל לגשת לכל הפונקציות של משתמש רשום.

הוספת מוצר לסל הקניות:

נווט לדף המוצרים על ידי לחיצה על הקישור "מוצרים" בתפריט הניווט.

בחר את המוצר הרצוי על ידי לחיצה על כפתור "הוסף לסל" הממוקם ליד המוצר.

תוכל לצפות בסל הקניות שלך ולעדכן את כמות המוצרים או להסיר מוצרים מהסל.

למשתמש רשום ישנן גם כל התכונות של משתמש מזדמן

מדריך משתמש – אדמין:

התחברות למערכת:

נווט לדף ההתחברות על ידי לחיצה על הקישור "התחברות" בתפריט הניווט.

מלא את שם המשתמש והסיסמה שלך ולחץ על כפתור "התחבר".

לאחר התחברות מוצלחת תוכל לגשת לכל הפונקציות של משתמש רשום ואדמין.

גישה לפאנל האדמין (Admin Panel):

לאחר ההתחברות, יופיע קישור נוסף בתפריט הניווט בשם "פאנל אדמין".

לחץ על הקישור כדי לנווט לפאנל האדמין.

עדכון זמינות מוצרים:

בפאנל האדמין תוכל לראות רשימת מוצרים קיימים.

בדף העריכה תוכל לעדכן את זמינות המוצר.

תכונות נוספות לאדמין

לאדמין יש את כל האפשרויות גישה של משתמש רשום, כולל עיון בדף הבית, בדף יצירת קשר ובדף המוצרים.

המדריכים הללו נועדו להבטיח שכל משתמש יוכל להשתמש באתר בצורה הנוחה והפשוטה ביותר, תוך מתן גישה לפונקציות המתאימות לסוג המשתמש שלו.

# **רפלקציה**

**תיאור התהליך**

בתחילת הפרויקט, הגדרתי מטרות ויעדים ברורים על פי דרישות הפרויקט. לאחר מכן, ערכתי מחקר ראשוני על הטכנולוגיות והכלים הרלוונטיים. הפרויקט התחיל בשלב התכנון, שבו יצרתי מסמך תכנון מפורט, כולל דיאגרמות UML ואפיון מסכים. לאחר מכן, התחלתי בפיתוח המודולים העיקריים של הפרויקט, כל מודול בנפרד תוך בדיקה ואימות כל שלב.

**הצלחות**

תכנון מסודר: התכנון המוקדם והדיאגרמות עזרו להבין את הפרויקט לעומק ולמנוע בעיות בהמשך.

עמידה בלוחות זמנים: הפרויקט הושלם בזמן תוך עמידה ביעדים ובתקנים שנקבעו מראש.

**אתגרים**

בעיה בתקשורת בין מודולים: נתקלתי בקושי בתקשורת בין חלקים שונים של המערכת.

בעיות ביצועים: חלק מהפעולות לקחו זמן רב מדי.

ניהול גרסאות: היה קושי בניהול הגרסאות השונות של הקוד.

דרכי הפתרון:

תקשורת בין מודולים: השתמשתי ב-API ברור ומסודר כדי לאפשר תקשורת חלקה בין מודולים.

ביצועים: שיפרתי את הקוד כדי לשפר את הביצועים.

**תהליך למידה**

למידה שהתקיימה:

שפות תכנות חדשות: למדתי להשתמש בשפות ובספריות חדשות שהיו נדרשות לפרויקט.

עקרונות תכנון: העמקתי בעקרונות תכנון תוכנה.

טכנולוגיות חדשות: נחשפתי לטכנולוגיות וכלים חדשים.

**כלים שנלקחו להמשך**

תכנון מסודר: כלים לתכנון ואפיון מערכת יעזרו לי בתכנון פרויקטים עתידיים.

תכנות אסינכרוני: הבנתי את היתרונות של תכנות אסינכרוני ואיישם אותו בפרויקטים עתידיים.

**תובנות מהתהליך:**

שיתוף מידע ולמידת עמיתים

שיתוף מידע ולמידת עמיתים היו חשובים מאוד. שיחות ותמיכה מחברי הצוות עזרו להתגבר על קשיים ולמצוא פתרונות יעילים.

שיפור פרויקט בראייה לאחור

במבט לאחור, הייתי מבצע בדיקות אוטומטיות בשלבים מוקדמים יותר של הפיתוח כדי למנוע בעיות שהיו קשות לזיהוי בשלב מאוחר יותר.

**שיפור עם משאבים נוספים:**

במידה והיו לי משאבים נוספים, הייתי משקיע יותר בבדיקות של ביצועים. בנוסף, הייתי שוקל שימוש בטכנולוגיות ענן כדי לשפר את המערכת.

**שאלות חקר עצמי**

אילו חלקים בפרויקט היו המאתגרים ביותר וכיצד התמודדת איתם?

האם היו דרכים נוספות שבהן יכולת לשפר את ביצועי המערכת?

כיצד התמודדת עם ניהול הזמן בפרויקט והאם היית משנה את הדרך בה ניהלת את הזמן?

מה היית עושה אחרת בתכנון ובפיתוח של הפרויקט?

סיכום

תהליך העבודה על הפרויקט היה מלמד ומאתגר. התמודדתי עם אתגרים שונים, למדתי כלים וטכנולוגיות חדשות, והבנתי את חשיבות העבודה בצוות ושיתוף מידע. במבט לאחור, ישנם דברים שהייתי עושה אחרת, אך התהליך כולו היה חוויה מלמדת ומעשירה.