



תיק פרויקט בהנדסת תוכנה - תוכנן ותוכנות מערכות



בה"ס אליאנס חיפה
אדם | חברה | סביבה

שם תלמיד: יונתן לוי

תעודת זהות: 214719247

מורה מנהה: ולרי קרייפוקו

תאריך הגשה: 11/5/2022

תוכן עניינים

2	תוכן עניינים
5	מבוא
5	תיאור תוכנת הספר
5	הרקע לפרויקט
5	תהליך המחבר
5	אתגרים מרכזיים
7	משתמשים ותפקידם באתר
7	סרטוני מדריך למשתמש
7	רישימת Use Case
8	דיאגרמת Use Case
9	בסיס הנתונים
9	סכמה והסבר כללי
10	טבלאות
15	רישימת Stored Procedures
16	קובץ לשימרת נתונים
17	מדריך למשתמש
17	תרשימים Screen Flow Diagram
18	דף האתר
42	ארQUITקטורת הפרויקט
42	הציג הפתרון המוצע והסיבות לבחירתו
42	ארQUITקטורת הפתרון המוצע (Top-Down Level Design)
45	מחלקות הפרויקט
46	תיאור המחלקות
58	מבנה נתונים בהם נעשה שימוש
59	זרימת מידע וקשרים בין היחידות השונות
60	ארQUITקטורת רשות
61	שירותי רשות שהפרויקט מספק
63	שירותי רשות שבסימוש האתר
64	טכנולוגיות בפרויקט
65	רפלקציה
66	ביבליוגרפיה

67.....	מדריך למפתח
67.....	HomePage.aspx
69.....	Gallery.aspx
72.....	Registration.aspx
78.....	LogIn.aspx
82.....	ClimbingShop.aspx
89.....	ShoppingCart.aspx
101.....	EditDataPage.aspx
107.....	ShoeCalcPage.aspx
110.....	WeatherPage.aspx
118.....	UserList.aspx
121.....	ManageCardPage.aspx
125.....	PastPurchasePage.aspx
129.....	ClimbingGroupPage.aspx
139.....	GroupCreationPage.aspx
144.....	MeetingCreationPage.aspx
150.....	GroupManagementPage.aspx
163.....	AdminPage.aspx
171.....	SupplyManagement.aspx
181.....	AddNewItem.aspx
185.....	SalesReportPage.aspx
192.....	NewCreditCardPage.aspx
195.....	MemberManagementPage.aspx
202.....	LogOut.aspx
203.....	Web.config
206.....	packages.config
209.....	MyAdoHelper.cs
225.....	SaleStatisticsHelper.cs
228.....	main.css
301.....	(שירות רשות) ShoeCalculatorService.asmx
306.....	EmailXML.xml
307.....	(שירות רשות) Web.config
308.....	(שירות רשות) packages.config

309.....	גופחים.....
309.....	הוראות להרצת האתר
309.....	פרטי משתמשים

מבוא

תיאור תכולת הספר

תיק הפרויקט מכיל את תיאור הפרויקט שבניתו, שבאנגלית נקרא "The Climbing Web", ובעברית - "רשות הטיפוס".

התיק מפרט בהרחבה על מבנה הפרויקט, שלל המרכיבים ממנו הוא בנו, ובכלים בהם השתמשתי בעת בנייתו. התיק מכיל את הפרקים הבאים: מבוא, מבנה/ארQUITטורה הפרויקט, מדריך למשתמש, בסיס הנתונים, מדריך למפתח, רפלקציה, ביבליוגרפיה, ונספחים.

הרקע לפרויקט

את הפרויקט התחלתי לבנות בכיתה י', במסגרת לימודי הנדסת תוכנה. תהליך הבניה נמשך כ-3 שנים, בהם למדתי ופיתחתי בו-זמנית את הפרויקט שלי, כך שאני לומד ומישם את מה שאני לומד בזמן אמיתי.

במהלך פיתוח האתר, השתמשתי באינספור מקורות חיצוניים באינטרנט, דרכם למדתי כיצד לפתח את האתר שלי בצורה הטובה ביותר ולפתור בעיות שקדם לא הצלחת להתמודד איתן. לדעתי הפרויקט חיזק אצלי את יכולת המחקר באופן שימושי ומהיר, וכיוום מיומנויות המחקר שלי נמצאות ברמה גבוהה מאוד שמאפשרת לי למצוא פתרונות לא רק עבור הפרויקט, אלא עברו כל בעיה שצאה מולו.

תהליך המחקר

לפני ובמהלך העבודה, חקרתי באינטרנט על אתרים אחרים שעוסקים בטיפוס, ולקחת מהם השראה לבנייה הפרויקט ולמרכיבים שלו. למדתי על איך שאתר אמרור להירות ולתפקוד, ומה הקשיים בבניית אתרים שאחרים נתקלו בהם במהלך בניית האתרים שלהם.

עשיתי שימוש נרחב בפורוםים בהם אנשים הדיעו לגבי בעיות שונות בהן נתקلت我自己, והוציאתי מכך תועלת רבה ולא פעם בזכות הפתרונות שהוצעו בפורוםים אלו הצלחת ליישם מרכיב חשוב בפרויקט שלי. בנוסף, צפיתי סרטונים רבים המסבירים נושאים שונים הקשורים לפרויקט, ובעזרת סרטונים אלו קיבלתי תמונה מוקדמת של מה שעלי' לבצע בזמן הקרוב בעבודה שלי על הפרויקט.

אתגרים מרכזיים

הבעיה אותה התמודדתי

הבעיה אותה התמודדתי בפרויקט שלי היא שכרגע כמעט אין אתר אינטרנט שמתעסקים בטיפוס מעבר לחנות או סרטוני טיפוס. אין מספיק אתרים שմשלבים יותר מאחד מהמרכיבים הללו ולכן לקחת עלי עצמי ליצור אתר שמספק מידע, קבוצות ומפגשי טיפוס, ובנוסף לכך גם חנות לנעל טיפוס.

הסיבות לבחירת הנושא

הנושא שבחרתי לבסס עליו את הפרויקט הוא טיפוס, מכיוון שאינו אישית מטפס ולכן הספורט קרוב אליו אישית, ומכיון שלפי דעתך אין מודעות רבה לענף הספורט וליפוי שבו ולכן אני בחרתי ליצור אתר שיכל להאריך עליו ולענות לצרכים של מטפס מתחילה.

מוטיבציה לעבודה

את המוטיבציה שלי לעבודה שאבתי במספר מקורות, אך בעיקר מצאתי מוטיבציה לעבודה בכך שהיא עוסקת בנושא שאין מתעניין בו ומאוד אהוב לעסוק בו ולחזור אותו. בנוסף ההגשות שהיו לנו במהלך השנה הציבו לי מטרות לאתר שהmarker על המוטיבציה שלי לעמוד במטרות אלו ולעבד על הפרויקט.

על איזה צורך הפרויקט עונה? איזה פתרון הפרויקט זהה בא לתת?

הפרויקט עונה על הצורך של כל מטפס מתחילה לקבל מסגרת כלשהי בעת העיסוק שלו בספורט. כל מטפס רוצה ולדעתו צריך להצטרף לקבוצת טיפוס מקצועית ולהתאמן באופן רציני. בנוסף, לכל מטפס יש את הסקרנות והרצון ללמידה על הספורט ועל האנשים שעוסקים בו, וכן האתר שלי גם מספק מידע על טיפוס כך שהගולשים בו יכולים להימנע מרקע על הספורט ולהכיר אותו. יתר על כן, האתר שלי כולל חנות לנעלי טיפוס, מכיוון שהמוצר הבסיסי וכמעט היחיד שמטפס מתחילה צריך הוא נעל טיפוס.

פתרונות לבעה

כפי שתכתבתי לעיל, הפרויקט שלי מספק מספר שירותים חשובים ונוחים לשימוש עבור המשתמשים. בפרויקט שלי יצאי מtower ההנחה שהמשתמשים הם מטפסים מתחילה שלא יודעים כמעט כלום על עולם הטיפוס והשימוש באתר אינטרנט, ולכן הקפדי שהאתר יהיה מעוצב באופן ידידותי למשתמש וגם למנהל. האתר מאפשר למשתמשים להירשם לקבוצות טיפוס, לקנות נעלי טיפוס, לצפות במצג אויר (לאלה שמתכוונים טיפוס בטבע), ועוד. כל השירותים הללו מעדו להקל על מטפסים להיכנס למסגרת של עולם הטיפוס ולהתאקלם לשימוש באתר שעוסק בטיפוס.

משתמשים ותפקידם באתר

סרטוני מדריך למשתמש

מדריך לאורח – <https://youtu.be/wFgvV7Wj5Ac>

מדריך למשתמש הרגיל – <https://youtu.be/X334dqCLgYM>

מדריך למאן – <https://youtu.be/CUCi7-F7R4k>

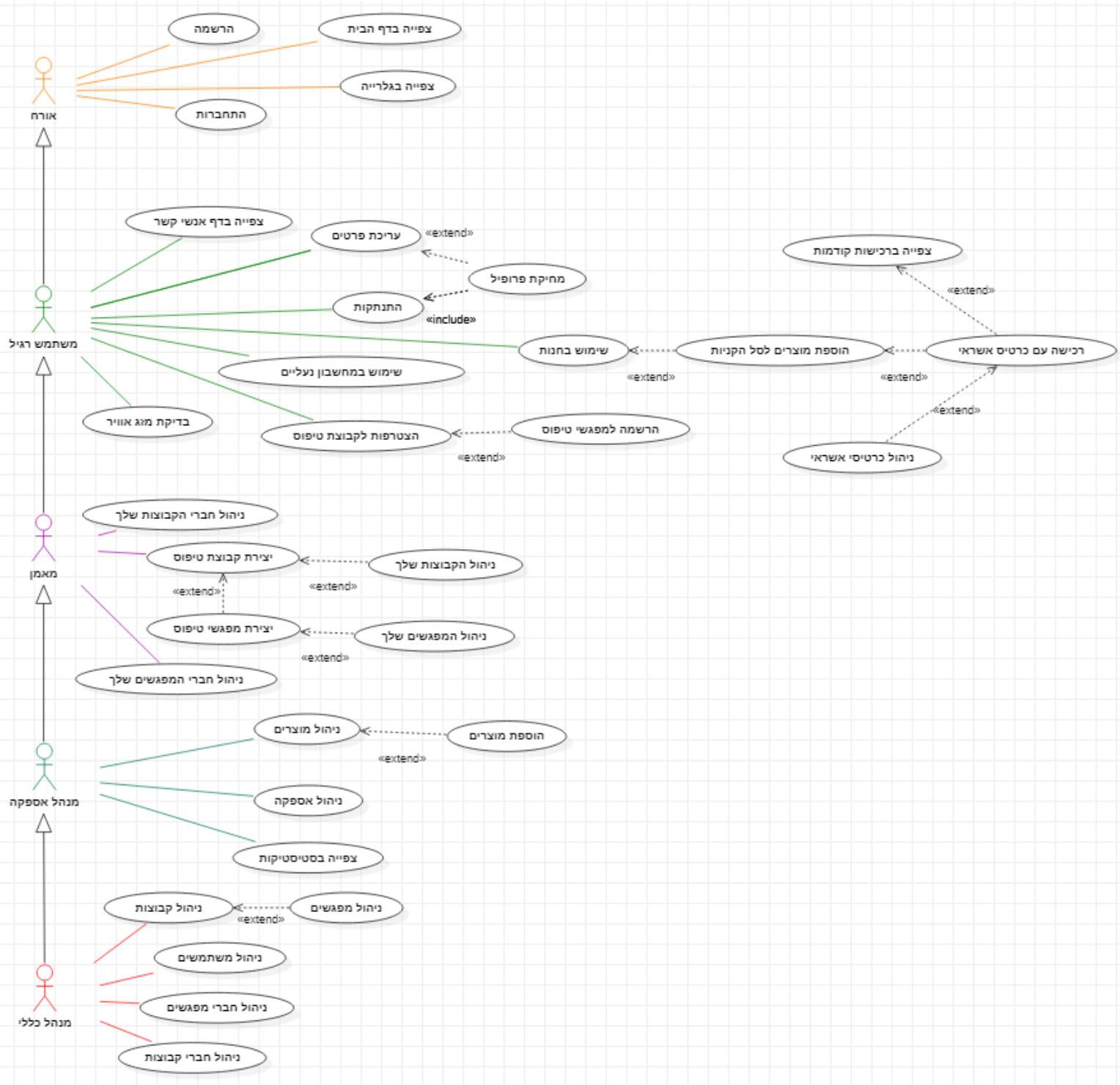
מדריך למנהל האספקה – <https://youtu.be/OJJJDBZ3DMA>

מדריך למנהל הכללי – <https://youtu.be/ns58IvpVg>

רישימת Use Case

- משתמש אורח - רשאי לצפות בדף הבית ובגלאיה. יכול להתחבר ולהירשם.
- משתמש רשום רגיל – רשאי לעשות את כל מה שהמשתמש האורח יכול. בנוסף, רשאי להשתמש בחנות הטיפוס, לנוהל סל קניות, לבצע תשלום סופי, לנוהל כרטיסי אשראי (רק את שלו), לנוהל את החשבון שלו, להשתמש במחשבון הנעלאים, לצפות ברישימת אנשי הקשר, לצפות ולשחרר קבלות מקניות ישנות, לצפות בקבוצות הטיפוס הקיימות, להציג/לעוזוב קבוצת טיפוס, ולהירשם/לבטל הגעה לאימונים.
- מאן – רשאי לעשות את כל מה שהמשתמש הרשום הרגיל יכול. בנוסף, רשאי ליצור קבוצות טיפוס, ליצור אימונים לקבוצות שלו, לנוהל את הקבוצות והאימונים שלו.
- מנהל אספקה – רשאי לעשות את כל מה שהמאן יכול לעשות. בנוסף, רשאי לנוהל את האספקה באתר ולצפות בסטייטיקות המכירות באתר.
- מנהל כללי – רשאי לעשות את כל מה שמנהל האספקה יכול לעשות. בנוסף, רשאי לנוהל משתמשים, לנוהל את כלל קבוצות הטיפוס והאימונים שלהם.

Diagram Use Case

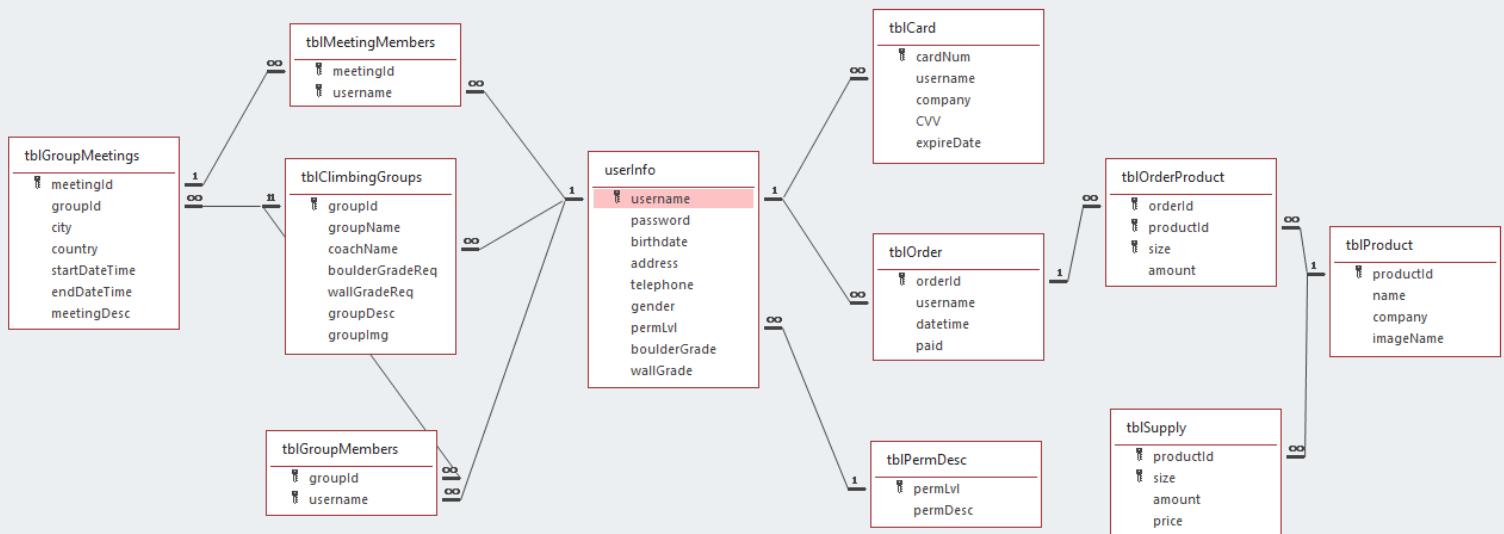


בסיס הנתונים

סכמה והסבר כללי

בסיס הנתונים בפרויקט שלי הוא מוצר Microsoft SQL Server גרסה 13.00.4001. בסיס הנתונים שלי יש כ-11 טבלאות, כלן מחוברות דרך מפתח זר עם לפחות טבלה אחרת אחת. טבלאות בסיס הנתונים שלי מוגדרות עד לרמת נרמול NF4 (כולל), ובננו לשם שמרית נתונים עבור המשתמשים, רמות הרשות באתר, קבוצות ופיגישות טיפוס, והזמנות ומוצרים מהחנות.

להלן סכמת טבלאות בסיס הנתונים:



טבלאות

כפי שציינתי, בסיס הנתונים שלי יש כ-11 טבלאות שונות. להלן צילומי המבנים של הטבלאות:

טבלה userInfo

Name	Data Type	Allow Nulls	Default
username	nvarchar(50)	<input type="checkbox"/>	
password	nvarchar(50)	<input type="checkbox"/>	
birthdate	date	<input type="checkbox"/>	
address	nvarchar(50)	<input type="checkbox"/>	
email	nvarchar(254)	<input type="checkbox"/>	
gender	nvarchar(50)	<input type="checkbox"/>	
permLvl	int	<input type="checkbox"/>	((0))
boulderGrade	nvarchar(50)	<input type="checkbox"/>	
wallGrade	nvarchar(50)	<input type="checkbox"/>	
city	nvarchar(50)	<input type="checkbox"/>	('Haifa')
		<input type="checkbox"/>	

Design
T-SQL

```

1 CREATE TABLE [dbo].[userInfo] (
2     [username] NVARCHAR (50) NOT NULL,
3     [password] NVARCHAR (50) NOT NULL,
4     [birthdate] DATE NOT NULL,
5     [address] NVARCHAR (50) NOT NULL,
6     [email] NVARCHAR (254) NOT NULL,
7     [gender] NVARCHAR (50) NOT NULL,
8     [permLvl] INT DEFAULT ((0)) NOT NULL,
9     [boulderGrade] NVARCHAR (50) NOT NULL,
10    [wallGrade] NVARCHAR (50) NOT NULL,
11    [city] NVARCHAR (50) DEFAULT ('Haifa') NOT NULL,
12    PRIMARY KEY CLUSTERED ([username] ASC),
13    CONSTRAINT [FK_userInfo_tblPermDesc] FOREIGN KEY ([permLvl]) REFERENCES [dbo].[tblPermDesc] ([permLvl]) ON UPDATE CASCADE
14 );

```

▲ Keys (1)
 <unnamed> (Primary Key, Clustered: username)

Check Constraints (0)

Indexes (0)

Foreign Keys (1)
 FK_userInfo_tblPermDesc (permLvl)

Triggers (0)

טבלה tblPermDesc

	Name	Data Type	Allow Nulls	Default	
permLvl	permLvl	int	<input type="checkbox"/>		
permDesc	permDesc	nvarchar(200)	<input type="checkbox"/>		<input type="checkbox"/>

Keys (1)

<unnamed> (Primary Key, Clustered: permLvl)

Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Design T-SQL

```

1 CREATE TABLE [dbo].[tblPermDesc] (
2     [permLvl] INT NOT NULL,
3     [permDesc] NVARCHAR (200) NOT NULL,
4     PRIMARY KEY CLUSTERED ([permLvl] ASC)
5 );

```

טבלה tblOrder

	Name	Data Type	Allow Nulls	Default	
orderId	orderId	int	<input type="checkbox"/>		
username	username	nvarchar(50)	<input type="checkbox"/>		
datetime	datetime	datetime	<input type="checkbox"/>		
paid	paid	float	<input type="checkbox"/>	((0))	<input type="checkbox"/>

Keys (1)

<unnamed> (Primary Key, Clustered: orderId)

Check Constraints (0)
Indexes (0)
Foreign Keys (1)

FK_tblOrder_userInfo (username)

Triggers (0)

Design T-SQL

```

1 CREATE TABLE [dbo].[tblOrder] (
2     [orderId] INT IDENTITY (1, 1) NOT NULL,
3     [username] NVARCHAR (50) NOT NULL,
4     [datetime] DATETIME NOT NULL,
5     [paid] FLOAT (53) DEFAULT ((0)) NOT NULL,
6     PRIMARY KEY CLUSTERED ([orderId] ASC),
7     CONSTRAINT [FK_tblOrder_userInfo] FOREIGN KEY ([username]) REFERENCES [dbo].[userInfo] ([username]) ON DELETE CASCADE ON UPDATE CASCADE
8 );

```

טבלה tblOrderProduct

	Name	Data Type	Allow Nulls	Default	
orderId	orderId	int	<input type="checkbox"/>		
productId	productId	int	<input type="checkbox"/>		
size	size	int	<input type="checkbox"/>		
amount	amount	int	<input type="checkbox"/>		<input type="checkbox"/>

Keys (1)

<unnamed> (Primary Key, Clustered: orderId, productId, size)

Check Constraints (0)
Indexes (0)
Foreign Keys (2)

FK_tblOrderProduct_tblProduct (productId)

FK_tblOrderProduct_tblOrder (orderId)

Triggers (0)

Design T-SQL

```

1 CREATE TABLE [dbo].[tblOrderProduct] (
2     [orderId] INT NOT NULL,
3     [productId] INT NOT NULL,
4     [size] INT NOT NULL,
5     [amount] INT NOT NULL,
6     PRIMARY KEY CLUSTERED ([orderId] ASC, [productId] ASC, [size] ASC),
7     CONSTRAINT [FK_tblOrderProduct_tblProduct] FOREIGN KEY ([productId]) REFERENCES [dbo].[tblProduct] ([productId]) ON DELETE CASCADE ON UPDATE CASCADE,
8     CONSTRAINT [FK_tblOrderProduct_tblOrder] FOREIGN KEY ([orderId]) REFERENCES [dbo].[tblOrder] ([orderId]) ON DELETE CASCADE ON UPDATE CASCADE
9 );

```

tblProduct טבלה

Name	Data Type	Allow Nulls	Default	
productId	int	<input type="checkbox"/>		
name	nvarchar(100)	<input type="checkbox"/>		
company	nvarchar(100)	<input type="checkbox"/>		
imageName	nvarchar(200)	<input type="checkbox"/>	('not_found.png')	<input type="checkbox"/>

Keys (1)

<unnamed> (Primary Key, Clustered: productId)

Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Design T-SQL

```

1 CREATE TABLE [dbo].[tblProduct] (
2     [productId] INT          IDENTITY (1, 1) NOT NULL,
3     [name]      NVARCHAR (100) NOT NULL,
4     [company]   NVARCHAR (100) NOT NULL,
5     [imageName] NVARCHAR (200) DEFAULT ('not_found.png') NOT NULL,
6     PRIMARY KEY CLUSTERED ([productId] ASC)
7 );

```

tblSupply טבלה

Name	Data Type	Allow Nulls	Default	
productId	int	<input type="checkbox"/>		
size	int	<input type="checkbox"/>		
amount	int	<input type="checkbox"/>		
price	float	<input type="checkbox"/>		<input type="checkbox"/>

Keys (1)

<unnamed> (Primary Key, Clustered: size, productId)

Check Constraints (0)
Indexes (0)
Foreign Keys (1)

FK_tblSupply_tblProduct (productId)

Triggers (0)

Design T-SQL

```

1 CREATE TABLE [dbo].[tblSupply] (
2     [productId] INT      NOT NULL,
3     [size]       INT      NOT NULL,
4     [amount]    INT      NOT NULL,
5     [price]     FLOAT (53) NOT NULL,
6     PRIMARY KEY CLUSTERED ([size] ASC, [productId] ASC),
7     CONSTRAINT [FK_tblSupply_tblProduct] FOREIGN KEY ([productId]) REFERENCES [dbo].[tblProduct] ([productId]) ON DELETE CASCADE ON UPDATE CASCADE
8 );

```

tblCard טבלה

Name	Data Type	Allow Nulls	Default	
cardNum	nvarchar(100)	<input type="checkbox"/>		
username	nvarchar(50)	<input type="checkbox"/>		
company	nvarchar(50)	<input type="checkbox"/>		
CVV	nvarchar(3)	<input type="checkbox"/>		
expireDate	date	<input type="checkbox"/>		

Keys (1)

<unnamed> (Primary Key, Clustered: cardNum, username)

Check Constraints (0)
Indexes (0)
Foreign Keys (1)

FK_tblCard_userInfo (username)

Triggers (0)

Design T-SQL

```

1 CREATE TABLE [dbo].[tblCard] (
2     [cardNum]   NVARCHAR (100) NOT NULL,
3     [username]  NVARCHAR (50) NOT NULL,
4     [company]   NVARCHAR (50) NOT NULL,
5     [CVV]        NVARCHAR (3) NOT NULL,
6     [expireDate] DATE      NOT NULL,
7     PRIMARY KEY CLUSTERED ([cardNum] ASC, [username] ASC),
8     CONSTRAINT [FK_tblCard_userInfo] FOREIGN KEY ([username]) REFERENCES [dbo].[userInfo] ([username]) ON DELETE CASCADE ON UPDATE CASCADE
9 );

```

טבלה tblClimbingGroups

Name	Data Type	Allow Nulls	Default	
groupId	int	<input type="checkbox"/>		
coachName	nvarchar(50)	<input type="checkbox"/>		
boulderGradeReq	nvarchar(50)	<input type="checkbox"/>		
wallGradeReq	nvarchar(50)	<input type="checkbox"/>		
groupDesc	nvarchar(300)	<input type="checkbox"/>		
groupImg	nvarchar(200)	<input type="checkbox"/>	('not_found.png')	
groupName	nvarchar(50)	<input type="checkbox"/>		
		<input type="checkbox"/>		

Keys (1)
 <unnamed> (Primary Key, Clustered: groupId)
Check Constraints (0)
Indexes (0)
Foreign Keys (1)
 FK_tblClimbingGroups_userInfo (username)
Triggers (0)

Design		T-SQL		
1	CREATE TABLE [dbo].[tblClimbingGroups] ([groupId]	INT	IDENTITY (1, 1) NOT NULL,
2		[coachName]	NVARCHAR (50)	NOT NULL,
3		[boulderGradeReq]	NVARCHAR (50)	NOT NULL,
4		[wallGradeReq]	NVARCHAR (50)	NOT NULL,
5		[groupDesc]	NVARCHAR (300)	NOT NULL,
6		[groupImg]	NVARCHAR (200)	DEFAULT ('not_found.png') NOT NULL,
7		[groupName]	NVARCHAR (50)	NOT NULL,
8				PRIMARY KEY CLUSTERED ([groupId] ASC),
9				CONSTRAINT [FK_tblClimbingGroups_userInfo] FOREIGN KEY ([coachName]) REFERENCES [dbo].[userInfo] ([username]) ON DELETE CASCADE ON UPDATE CASCADE
10);

טבלה tblGroupMeetings

Name	Data Type	Allow Nulls	Default	
meetingId	int	<input type="checkbox"/>		
groupId	int	<input type="checkbox"/>		
city	nvarchar(90)	<input type="checkbox"/>		
country	nvarchar(60)	<input type="checkbox"/>		
startDateTime	datetime	<input type="checkbox"/>		
endDateTime	datetime	<input type="checkbox"/>		
meetingDesc	nvarchar(300)	<input type="checkbox"/>		
		<input type="checkbox"/>		

Keys (1)
 PK_tblGroupMeetings (Primary Key, Clustered: meetingId)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Design		T-SQL		
1	CREATE TABLE [dbo].[tblGroupMeetings] ([meetingId]	INT	IDENTITY (1, 1) NOT NULL,
2		[groupId]	INT	NOT NULL,
3		[city]	NVARCHAR (90)	NOT NULL,
4		[country]	NVARCHAR (60)	NOT NULL,
5		[startDateTime]	DATETIME	NOT NULL,
6		[endDateTime]	DATETIME	NOT NULL,
7		[meetingDesc]	NVARCHAR (300)	NOT NULL,
8				CONSTRAINT [PK_tblGroupMeetings] PRIMARY KEY CLUSTERED ([meetingId] ASC)
9);

טבלה tblGroupMembers

Name	Data Type	Allow Nulls	Default	
groupId	int	<input type="checkbox"/>		
username	nvarchar(50)	<input type="checkbox"/>		<input type="checkbox"/>

Keys (1)

<unnamed> (Primary Key, Clustered: groupId, username)

Check Constraints (0)
Indexes (0)
Foreign Keys (2)

FK_tblGroupMembers_tblClimbingGroups (groupId)

FK_tblGroupMembers_userInfo (username)

Triggers (0)
Design ↑ T-SQL

```

1 CREATE TABLE [dbo].[tblGroupMembers] (
2     [groupId] INT NOT NULL,
3     [username] NVARCHAR (50) NOT NULL,
4     PRIMARY KEY CLUSTERED ([groupId] ASC, [username] ASC),
5     CONSTRAINT [FK_tblGroupMembers_tblClimbingGroups] FOREIGN KEY ([groupId]) REFERENCES [dbo].[tblClimbingGroups] ([groupId]),
6     CONSTRAINT [FK_tblGroupMembers_userInfo] FOREIGN KEY ([username]) REFERENCES [dbo].[userInfo] ([username])
7 );

```

טבלה tblMeetingMembers

Name	Data Type	Allow Nulls	Default	
meetingId	int	<input type="checkbox"/>		
username	nvarchar(50)	<input type="checkbox"/>		<input type="checkbox"/>

Keys (1)

<unnamed> (Primary Key, Clustered: meetingId, username)

Check Constraints (0)
Indexes (0)
Foreign Keys (2)

FK_tblMeetingMembers_tblGroupMeetings (meetingId)

FK_tblMeetingMembers_userInfo (username)

Triggers (0)
Design ↑ T-SQL

```

1 CREATE TABLE [dbo].[tblMeetingMembers] (
2     [meetingId] INT NOT NULL,
3     [username] NVARCHAR (50) NOT NULL,
4     PRIMARY KEY CLUSTERED ([meetingId] ASC, [username] ASC),
5     CONSTRAINT [FK_tblMeetingMembers_tblGroupMeetings] FOREIGN KEY ([meetingId]) REFERENCES [dbo].[tblGroupMeetings] ([meetingId]),
6     CONSTRAINT [FK_tblMeetingMembers_userInfo] FOREIGN KEY ([username]) REFERENCES [dbo].[userInfo] ([username])
7 );

```

רשימת Stored Procedures

בבסיס הנתונים שלי נעשה שימוש נרחב ב-SP (Stored Procedures), מכיוון שפעולות אלו מאפשרות נוחות בעבודה מול הטבלאות ו מבטיחות טרנסקציה בטוחה עם מסד הנתונים, שמתרכשת בתוך מסד הנתונים עצמו. הזרה המוגדרת מראש של ה-SP עוזר במניע התקפות SQL Injection, וגם מצמצמות שאינן עצמן. להלן רשימת ה-SP בהן נעשה שימוש בפרויקט:

- ▲ Stored Procedures
 - ▷ deleteUser
 - ▷ getCardAutocomplete
 - ▷ getCompleteOrderProductTbl
 - ▷ getOrderProductTbl
 - ▷ getOrderTbl
 - ▷ getOrderTblByUser
 - ▷ getProductTbl
 - ▷ getSupplyTbl
 - ▷ getUserAndPasswordInfo
 - ▷ getUserInfoTbl
 - ▷ getUsernameInfo
 - ▷ getUserPermLvl
 - ▷ insertNewCard
 - ▷ insertNewProduct
 - ▷ insertNewUser
 - ▷ sortByParam
 - ▷ updateAmount
 - ▷ updateUser
 - ▷ updateUserWithGender

קובץ לשמרות נתונים

בשירות הרשות שהפרויקט שלו מספק, נעשה שימוש בקובץ XML על מנת לשמר באופן זמני את המידע שנשלח יחד עם אימייל. קובץ ה-XML מצורף לאימייל ולאחר מכן נשאר בפרויקט עם המידע שהועבר אליו עד שמו עבר אליו מידע עבור הודעה חדשה הדורס את המידע הקודם הקודם מכיוון שכבר אין בו צורך. להלן דוגמה לתכני קובץ ה-XML, XML.xml, EmailXML.xml, לאחר שליחת קבלה בפורמט XML באימייל ללקוח:

```
<OrderReceipt>
  <BasicOrderInfo>
    <username>valery_admin</username>
    <date>17/4/2022</date>
    <totalPayment>14893.7</totalPayment>
  </BasicOrderInfo>
  <ProductDetails>
    <size>37</size>
    <amount>9</amount>
    <name>Tomoa Narasaki Pro</name>
  </ProductDetails>
  <ProductDetails>
    <size>40</size>
    <amount>9</amount>
    <name>Solutions Comp Men's</name>
  </ProductDetails>
  <ProductDetails>
    <size>41</size>
    <amount>12</amount>
    <name>Solutions Comp Men's</name>
  </ProductDetails>
  <ProductDetails>
    <size>46</size>
    <amount>2</amount>
    <name>Shark 2.0</name>
  </ProductDetails>
</OrderReceipt>
```

מדריך למשתמש

תרשים Flow Diagram

אורח בלבד - אדום
אורח ומעלה - כתום
משתמש רשום וגיל ומעלה - צהוב
מאמן ומעלה - ירוק
מנהל אספקה ומעלה - כחול
מנהל כללי - סגול
חריג - ללא צבע

מקרה חריג - רק
אורח או מנהל
כללי

כניסה לאתר

דף הבית

דף התמחות

గליה

עריכת
פרופיל

דף הרשמה/הוספה
משתמש

תחזית מזג
אוויר

מחשבון
נעליים

חנות

ניהול כרטיסי
אשראי

הוספה כרטיס
אשראי

סל קניות

דף אנשי
קשר

דף קבוצות
ଓایمونিস

צפייה בביבlio
 קודמות

דף יצירתי
אימוניות

דף יצירתי
קבוצות

דף הוסף
מוצר

דף ניהול
קבוצות

דף ניהול
אספקה

דף ניהול
קבוצות או
מפגשים

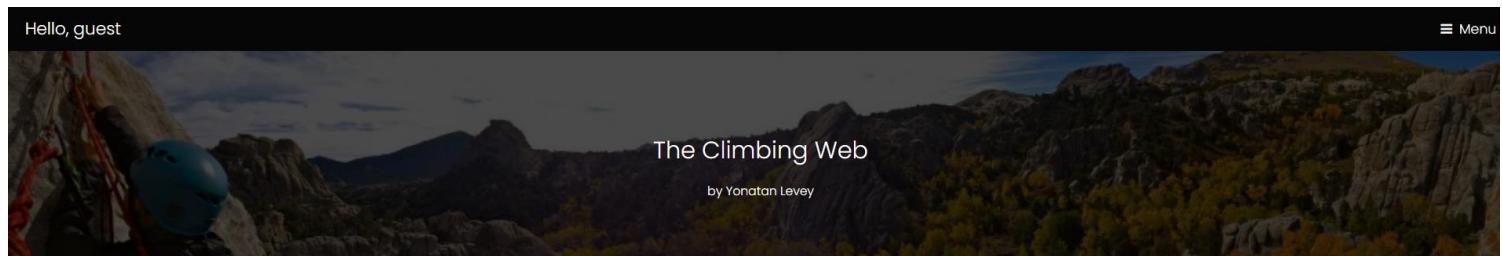
דף ניהול
משתמשים

עבור מנהל כללי - ניהול
כל הקבוצות. עבור
מאmins/מנהל אספקה
- ניהול הקבוצות שלהם -
בלבד.
כרגע עם חברי קבוצות
ומפגשים

דף הבית

דף הבית

דף הבית הוא דף הבסיס של האתר. הוא הדף הראשון שמשתמש רואה כשהוא מתחבר, והוא נגיש לכל סוגי המשתמשים ואורחיהם. בדף הבית ניתן לקרוא אודוטות האתר, ולהשתמש בתפריט.



Hello, guest ≡ Menu

The Climbing Web
by Yonatan Levey

About The Site

This site is all about climbing. Climbing is an extremely popular sport and has gotten even more popular lately. Here you can learn all about it.



In my website you can read about different climbing styles, lots of the current and former professional climbers, and climbing spots all over the world.

Climbing can be done as both a hobby and a career and has lots of different levels.

My goal in this website is to let you learn more about climbing and all its different aspects, and maybe get you into it if you don't climb yet.

📞 Support: 055-886-1427

© Yonatan Levey. All rights reserved.

галריה

галריה יכולם לגשת כל משתמשי האתר ואורחים. הгалריה מציגה ארבעה מטפסים מפורטים, וביניהם אלכס צזנוב, אחד המטפסים הטובים ביותר בישראל. הгалריה מאפשרת גישה לדפי המדיה החברתית של מטפסים אלו ומספקת פסקת מידע על כל אחד מהם.

The screenshot shows the homepage of The Climbing Web. At the top, there's a large banner featuring a climber on a wall with mountains in the background, with the text "Hello, guest" and "Menu". Below the banner, the title "The Climbing Web" and subtitle "by Yonatan Levey" are displayed. The main content area is titled "Gallery" and contains four sections, each featuring a photo of a climber and their name:

- Adam Ondra**: A Czech professional rock climber, specializing in lead climbing and bouldering. Adam Ondra is one of the most iconic rock climbers of all time. He is renowned for sending the world's first and only 9c, which he named Silence.
- Alex Megos**: Alexander Megos is a German rock climber. He was the first climber to on-sight (climb without prior practice or advice) a route graded 9a (5.14d). He has climbed multiple notable hard routes and boulder problems, including three 9b+ (5.15c) routes (Perfecto Mundo, Bibliographie and King Capella).
- Alex Khazanov**: Alex Khazanov is a professional climber in the bouldering industry, and won a gold medal in bouldering climbing in the 2018 World Cup round and two bronze medals in bouldering at the European Youth Championships in 2013 and 2014. As of January 2022 he is ranked 53rd in the world in bouldering, and is a member of the Israel national team in climbing.
- Chris Sharma**: Chris Omprakash Sharma is an American rock climber and considered one of the greatest in the history of the sport. In 2001, Sharma became the world's first climber to redpoint a consensus 9a+ (5.15a) route, Realization/Biographie, ushering in what was called a "technical revolution" in sport climbing. In 2008, Sharma redpointed the first 9b (5.15b) route, Jumbo love, and at 2013 became only the second to climb a 9b+ (5.15c) route La Dura Dura.

Each climber's section includes social media links for Instagram, YouTube, and Facebook.

דף הרשמה

דף זה נועד עבור אורחים בלבד. הדף מאפשר לאורח ליצור משתמש חדש, ולקבוע לו שם, סיסמה, עיר מגורים, כתובת,מין, תאריך לידה, אימייל, רמת טיפוס בבלדר, ורמת טיפוס בהובלה.

להלן תנאי הולידציה:

- שם משתמש – חייב להיות אלפאנומי, בין 5 ל-50 תווים, ולא תפוס
- סיסמה – חייבת להכיל לפחות אות גדולה אחת, אות קטנה אחת, מספר אחד, ולהיות בין 5 ל-50 תווים
- כתובת – חייבת להיות מילה אחת לפחות מלאה במספר, בין 5 ל-50 תווים
- תאריך לידה – המשתמש חייב להיות בין גיל 8 עד 120
- אימייל – חייב להיות לפי פורתט האימייל הסטנדרטי (abc.xyz@abc.xyz), עד 254 תווים
- רמת טיפוס בבלדר – חייבת להיות לפי הפורתט הסטנדרטי (V0-V17)
- רמת טיפוס בהובלה – חייבת להיות לפי הפורתט הזרפת (5A-9C+)

Hello, guest

The Climbing Web
by Yonatan Levey

Registration Form:

Username:

Password:

City:

Address:

Gender:

Male

Female

Birthdate:

Email:

Best Bouldering Grade (v-Scale):

Best Sport Climbing Grade (French, capital letters):

SIGN UP

Support: 055-886-1427

© Yonatan Levey. All rights reserved.

דף התחברות

דף זה נועד עבור אורחים בלבד. הדף מאפשר לאורח להזין פרטי משתמש ולנסות להתחבר. במידה וההתחברות מצליחה – הוא מועבר לדף הבית, הפעם כמשתמש רשום. יש לאורח גם אפשרות לבקש לשמור את פרטי ההתחברות או לא לשמור אותם. שמירת פרטי המשתמש נעשית דרך פעולות בקוד מאחור שיעשו שימוש ב-Cookies.

Hello, guest

The Climbing Web
by Yonatan Levey

Log In Form:

Username:

Password:

Remember me?

LOG IN

Support: 055-886-1427

© Yonatan Levey. All rights reserved.

דף החנות

האתר שלי מספק למשתמשים הרשומים שלו חנות טיפוס פונקציונאלית ונוחה לשימוש. החנות מאפשרת למשתמש ליקנות נעלים טיפוס שונות בגודלים שונים, וمعدכנת את תగית המחיר בהתאם לגודל ומספר הנעלים שהזין המשתמש. המשתמש יוכל רק להזמין נעלים שנמצאות במלאי, ולכן נעלים שאין להם שום גודל במלאי לא ניתן לרכישה כלל, ונעלים שחילק מוגדלין במלאי אפשרות רק לקנות את גודלים אלו. בעת סיום בחירת מוצר, ניתן לבחור להוסיף לסל או לבטל את הבחירה בו.

The screenshot shows the homepage of 'The Climbing Web' with a banner featuring a climber and a scenic mountain view. Below the banner, the title 'Our Climbing Shop' is displayed, along with a 'VIEW SHOPPING CART' button. The shopping cart section contains three items:

Image	Product Name	Product Company	Size	Amount	Price
	Scarpa Boostic	Scarpa	Select A Size	Select An Amount	-
	Tomoa Narasaki Pro	Unparallel	Select A Size	Select An Amount	-
	Solutions Comp Men's	La Sportiva	Select A Size	Select An Amount	-

Each item row includes an 'EDIT' button at the bottom.

סל הנקניות

בסל הנקניות המשמש יכול לצפות ב מוצרים שהוסיף לסל הנקניות, לעורר את הכמות של כל מוצר בגודל כלשהו שהוא הוסיף לסל, או למחוק אותו מהסל. בנוסף, המשמש יכול לראות כמה מהגודל הספציפי של המוצר נמצא במלאי, ולפיכך לדעת כמה הוא יכול להזמין. באותו הדף המשמש יכול לבצע את הרכישה הסופית, אך אם ינסה לעבור לדף הרכישה הסופית כשהוא מנסה להזמין יותר מוצר מאשר במלאי, הוא יקבל הודעה אזירה על כך ולא יעבור לתשלום. אם ליקוח מנסה להיכנס לסל הנקניות כשהוא ריק, הוא יعبر לדף החנות יחד עם הודעה שמצינית כי סל הנקניות שלו ריק.

בתשלום הסופי, המשמש יכול לבחור להשלים פרטים מכרטיים אשר אי שהוא שמר עבור עצמוו, או להשתמש בכרטיס חדש ובחירה אם לשומר אותו או לא. אם התשלום מבוצע בהצלחה, המשמש יקבל קבלה לרכישה מייל.

סל הנקניות:

Editing	Deletion	Size	Amount	Product Name	Price	In Stock
EDIT ORDER	DELETE	39	2	Scarpa Boostic	819.98	7
EDIT ORDER	DELETE	37	1	Tomoa Narasaki Pro	458.99	9
EDIT ORDER	DELETE	41	1	Solutions Comp Men's	475.99	17

Support: 055-886-1427
© Yonatan Levey. All rights reserved.

טופס תשלום:

תנאיolidציה בטופס התשלום –

- מספר כרטיס אשראי – מספרי באורך 16 ספרות ומאושר על ידי שירות הרשות החיצוני LuhnChecker כתקין
- ספרות בגב הkartis – 3 ספרות
- תאריך תפוגה – חייב להיות תקין (אחרי חודש הרכישה לפחות)

Hello, valery_admin. 1 users are logged in.

The Climbing Web
by Yonatan Levey

Shopping Cart

Total Price: 1754.96

GO BACK TO SHOPPING CART

Use A Saved Credit Card:

Card Number (No Spaces):

USE CARD

CVV:

Expiration Date:

Save card info?

PURCHASE

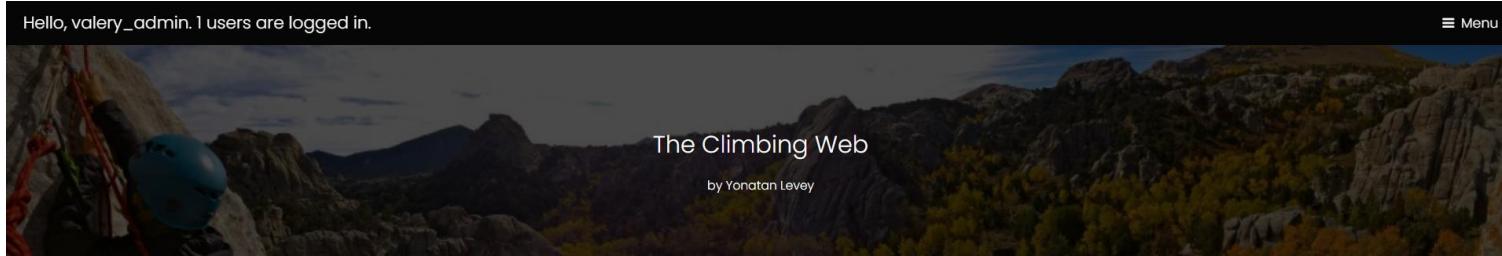
Support: 055-886-1427

© Yonatan Levey. All rights reserved.

דף ערך פרטי משתמש

בדף זה משתמש יכול לעורר את פרטיו, כਮון בתנאי שיעמדו בתנאי הולידציה המקוריים בהירשותו לאתר. המשתמש יכול גם לבחור למחוק את עצמו לצמיתות. אם יבחר למחוק את עצמו, המשתמש ימחק מטבלת המשתמשים ויתפרק.

Hello, valery_admin. 1 users are logged in. ≡ Menu



The Climbing Web
by Yonatan Levey

User Update Form:

Username:

Password:

City:

Address:

Gender: Male Female

Birthdate: □

Telephone:

Best Bouldering Grade (V-Scale):

Best Sport Climbing Grade (French, capital letters):

UPDATE USER DELETE USER

מחשבון נעליים

בדף זה יכול המשתמש לחשב את גודל נעלי הטיפוס המתאימים עבורו מספר חברות ספציפיות ולפי העדפה אישית. אלגוריתם החישוב נמצא בשירות הרשות שיצרת והמחשבון קורא לפועלה בשירות הרשות המחזירה את הגודל המומלץ לאחר חישוב.

The screenshot shows the homepage of The Climbing Web. At the top, there is a banner featuring a climber's helmet and a scenic mountain landscape. Below the banner, the site's name "The Climbing Web" and the author's name "by Yonatan Levey" are displayed. A dark navigation bar at the very top contains the text "Hello, valery_admin. 1 users are logged in." and a "Menu" icon. The main content area is titled "Shoe Size Calculator". It includes three input fields: "Street shoe size:" (empty), "Company:" (set to "La Sportiva"), and "Fit:" (set to "Loose"). A large "CALCULATE" button is centered below these fields. At the bottom of the page, there is a black footer bar with a support phone number "Support: 055-886-1427" and a copyright notice "© Yonatan Levey. All rights reserved."

Hello, valery_admin. 1 users are logged in. ≡ Menu

The Climbing Web
by Yonatan Levey

Shoe Size Calculator

Street shoe size:

Company:

Fit:

CALCULATE

Support: 055-886-1427
© Yonatan Levey. All rights reserved.

תחזית מזג אוויר

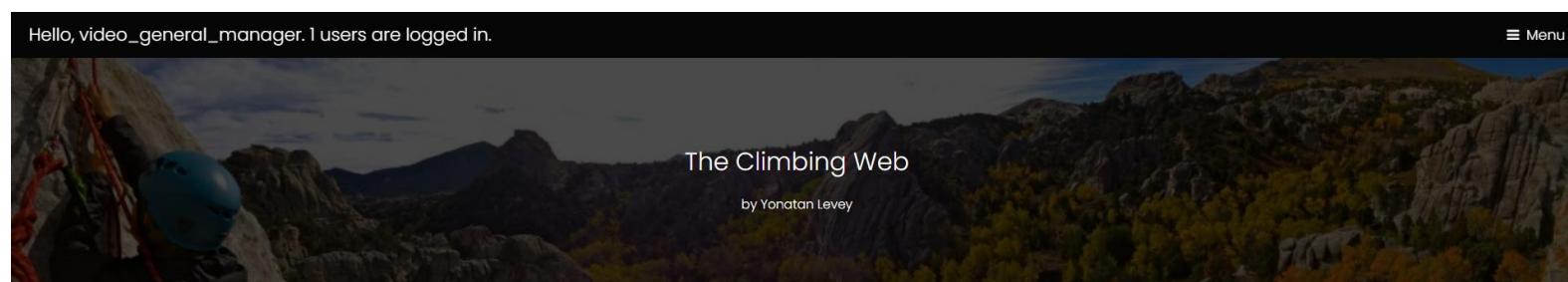
בדף זה יכול המשתמש לצפות בתחזית מזג האוויר לשבוע הקרוב בכל עיר שיבחר. במידה והמשתמש עוד לא נכנס העיר, יוצג מזג האוויר בהתאם למיקום כתובת-HD שלו, ואם זה לא הצליח, יוצג מזג האוויר בעיר מגוריו. בקדם לכך יש בדיקה של העיר והמדינה שהוזנו, על מנת לבדוק שהמקום אכן קיים. לאחר מכן, נקראו שירות החיצוני שמוריד את תחזית מזג האוויר בפורמט OpenWeatherMap. המידע עובר דה-סריאליזציה ומוכנס לאובייקטים, ואז מעובד ומודג בדף. יתר על כן, אם קיימות התראות מיוחדות לגבי מזג האוויר הן יוצגו מעל מזג האוויר בראשית Alerts.

The screenshot shows the homepage of The Climbing Web. At the top, there's a banner with a climber and the text "Hello, valery_admin. 1 users are logged in." and a "Menu" button. Below the banner, the site title "The Climbing Web" and subtitle "by Yonatan Levey" are displayed. A main heading "Always check the weather before going climbing outdoors!" is followed by input fields for "City:" and "Country:", and a "GET WEATHER" button. The page then displays a weather forecast for Haifa, Israel, from Monday, 25/04/2022, to Monday, 02/05/2022. Each day's forecast includes a weather icon, date, condition, temperature range, highest temperature, lowest temperature, and humidity. The forecast is presented in two rows of four columns each.

Monday, 25/04/2022	Tuesday, 26/04/2022	Wednesday, 27/04/2022	Thursday, 28/04/2022
Clear Sky Temperature: 22.43 Highest Temperature: 23.34 Lowest Temperature: 18.07 Humidity: 64%	Clear Sky Temperature: 21.25 Highest Temperature: 21.37 Lowest Temperature: 18.2 Humidity: 60%	Broken Clouds Temperature: 20.63 Highest Temperature: 20.81 Lowest Temperature: 17.41 Humidity: 57%	Clear Sky Temperature: 20.64 Highest Temperature: 20.64 Lowest Temperature: 15.93 Humidity: 56%
Friday, 29/04/2022	Saturday, 30/04/2022	Sunday, 01/05/2022	Monday, 02/05/2022
Clear Sky Temperature: 20.91 Highest Temperature: 21.03 Lowest Temperature: 15.97 Humidity: 47%	Clear Sky Temperature: 21.43 Highest Temperature: 21.64 Lowest Temperature: 17.48 Humidity: 60%	Clear Sky Temperature: 20.41 Highest Temperature: 20.41 Lowest Temperature: 16.56 Humidity: 72%	Clear Sky Temperature: 20.68 Highest Temperature: 20.68 Lowest Temperature: 17.13 Humidity: 77%

דף אנשי קשר

דף זה מציג את פרטי המשתמשים הבסיסיים של משתמשי האתר והאימיילים שלהם, כך שאם מישחו רצוח ליצור קשר עם משתמש אחר הוא יוכל להשתמש באימייל שלו ולעשות זאת. ניתן לעורר חיפוש עבור משתמש מסוים כפי שמוצג בתמונה, בה חיפשתי משתמשים בשם מכיל את האות "z". במידה ולא נמצאו משתמשים, תוצג הודעה מתאימה. על הקלט של שם המשתמש לענות על תנאי הולידציה לשם משתמש שהציגו קודם לכן.



Hello, video_general_manager. 1 users are logged in. ≡ Menu

The Climbing Web

by Yonatan Levey

Contact List:

Get in touch with our users!

User Search:

SEARCH **RESET SEARCH**

Username: TaliGendel Email: danceteam12345@gmail.com Birth date: 2/1/2007 Gender: Female Best Bouldering Grade: V2 Best Wall Grade: 7B	Username: test_coach Email: example@gmail.com Birth date: 10/26/2004 Gender: Male Best Bouldering Grade: V0 Best Wall Grade: 5A	Username: test_supply_manager Email: example@gmail.com Birth date: 10/26/2004 Gender: Male Best Bouldering Grade: V0 Best Wall Grade: 5A	Username: testerDude45 Email: what@yahoo.com Birth date: 3/3/2003 Gender: Male Best Bouldering Grade: V4 Best Wall Grade: 9A	Username: testGuy1 Email: yessir@gmail.com Birth date: 2/3/2001 Gender: Female Best Bouldering Grade: VII Best Wall Grade: 8C+
Username: test_basic Email: example@gmail.com Birth date: 10/26/2004 Gender: Male Best Bouldering Grade: V0 Best Wall Grade: 5A	Username: test_general_manager Email: example@gmail.com Birth date: 10/26/2004 Gender: Male Best Bouldering Grade: V0 Best Wall Grade: 5A	Username: testDude2 Email: test.tester@yahoo.com Birth date: 1/2/2001 Gender: Male Best Bouldering Grade: V2 Best Wall Grade: 5B+		

ניהול כרטיסי אשראי

בדף זה יכול משתמש לעורוך את כרטיסי האשראי השמורים שלו, כמפורט בתנאי שיעמודו בתנאי הולידציה לכרטיסי אשראי. יתר על כן, המשתמש יכול למחוק כרטיס אשראי מהכרטיסים השמורים שלו במידה והוא כבר לא רוצה שישמר באתר. המשתמש יכול גם להוסיף כרטיס אשראי חדש.

The screenshot shows a user interface for managing credit cards. At the top, a banner features a climbing scene with the text "Hello, valery_admin. 1 users are logged in." and a "Menu" button. Below the banner, the website's name "The Climbing Web" and the author's name "by Yonatan Levey" are displayed. A large, semi-transparent background image of a mountainous landscape is visible. The main content area is titled "Credit Card Page" and contains a "ADD NEW CARD" button. Below this, there is a table with columns for "Editing", "Deletion", "Card Number", "Company", "CVV", and "Expiration Date". Under the "Card Number" column, the card details are listed: 5216442233476632, MasterCard, 980, and 9/1/2025. To the left of the card number are "EDIT CARD" and "DELETE" buttons. At the bottom of the page, there is a support phone number (055-886-1427) and a copyright notice: "© Yonatan Levey. All rights reserved."

صفיה ברכישות קודמות

בדף זה משתמש יכול לראות את הרכישות הקודמות שביצע באתר ואת פרטייה. בנוסף, הוא יכול לבקש שתישלח לו שוב הקבלה לרכישה כלשהי.

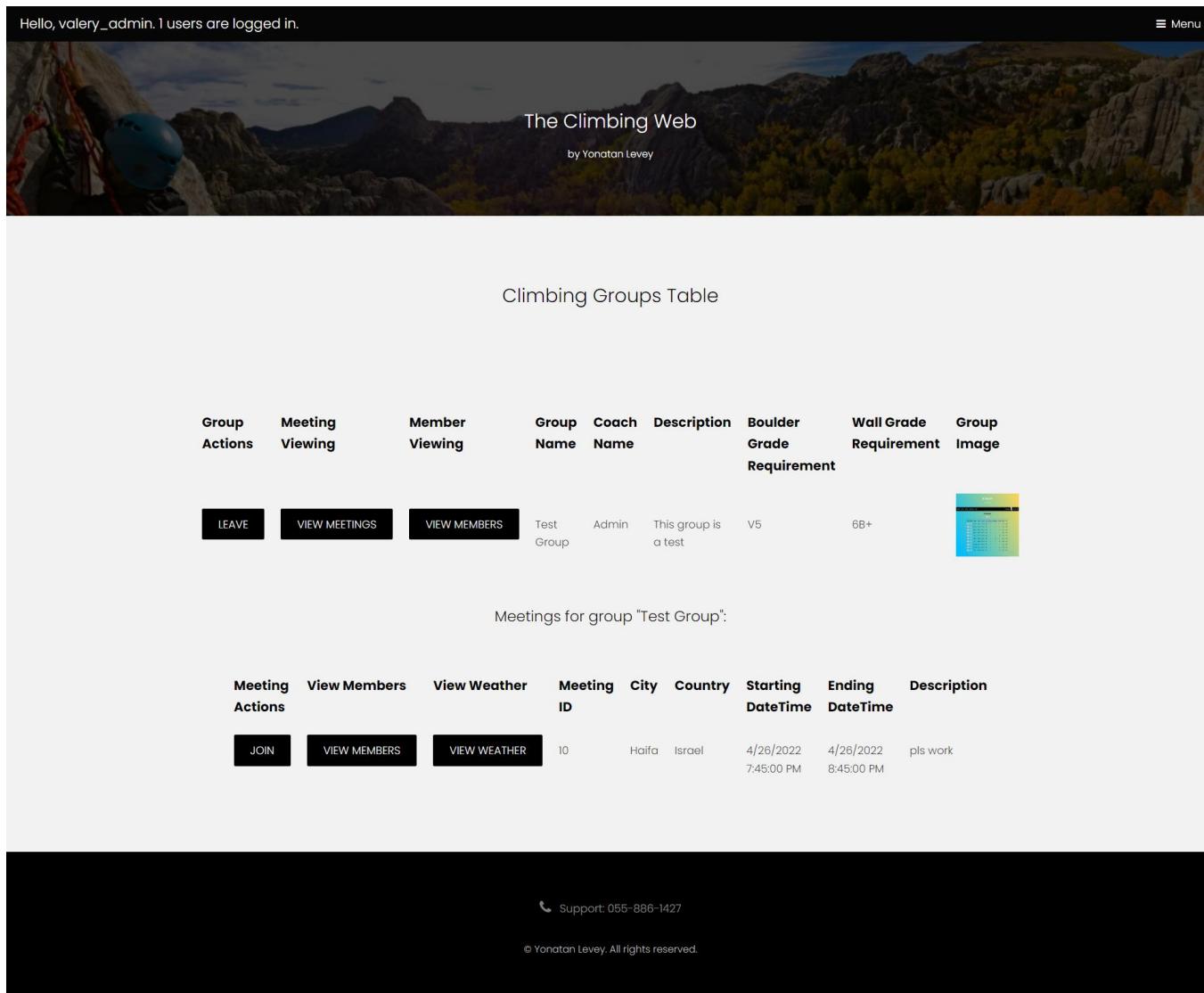
The screenshot shows the 'Past Orders' section of the website. At the top, there's a banner with a climber's helmet and a scenic mountain view. Below the banner, the page title 'Past Orders' is centered. A table lists three orders:

Details	Order ID	Order Date	Total Paid
RESEND RECEIPT VIEW DETAILS	1	10/10/2019 12:00:00 AM	1059.99
RESEND RECEIPT VIEW DETAILS	8	12/2/2020 3:57:44 PM	455.99
RESEND RECEIPT VIEW DETAILS	9	12/2/2021 7:18:31 PM	222.99

Below the table, there are page navigation links: 1, 2, 3. At the bottom of the page, there's a support phone number and a copyright notice.

דף קבוצות ואמוני טיפוס

בדף זה משתמש יכול לצפות בכל קבוצות הטיפוס הקיימות באתר, ולהציגו לקבוצה שמתאימה לו והוא עומד בדרישות רמת הטיפוס שלה. משתמש יכול להיות חבר בקבוצה אחת בלבד, אך יכול תמיד לעזוב אותה ולהציגו לאחרת אם יחשך לו. משתמש רשום ל千古ה כלשהי יכול לצפות בל"ז האימונים שלה, ולהירשם או לבטל הגעה לאימונים. מאמן של קבוצה כלשהו לא יכול להירשם ל千古ה של עצמו.



Hello, valery_admin. 1 users are logged in. ☰ Menu

The Climbing Web
by Yonatan Levey

Climbing Groups Table

Group Actions	Meeting Viewing	Member Viewing	Group Name	Coach Name	Description	Boulder Grade	Wall Grade Requirement	Group Image
LEAVE	VIEW MEETINGS	VIEW MEMBERS	Test Group	Admin	This group is a test	V5	6B+	

Meetings for group "Test Group":

Meeting Actions	View Members	View Weather	Meeting ID	City	Country	Starting DateTime	Ending DateTime	Description
JOIN	VIEW MEMBERS	VIEW WEATHER	10	Haifa	Israel	4/26/2022 7:45:00 PM	4/26/2022 8:45:00 PM	pls work

Support: 055-886-1427
© Yonatan Levey. All rights reserved.

דף יצירת קבוצות טיפוס

בדף זה יכולים מאמנים ליצור קבוצות טיפוס חדשות, עם שם, תמונה, תיאור, ודרישות רמת טיפוס לבחירתם. מנהלים כללים יכולים ליצור קבוצה בשם מאמנים, ולכן מופיע בצלם הבא שדה "שם מאמן", שמוופיע רק למנהלים כללים ולא למאמנים רגילים.

להלן תנאי הולידציה של הטופס –

- שם קבוצה – אלפabetי, בין 2 ל-50 תווים
- שם מאמן (עבור מנהל כללי) – אלפאנומרי, בין 5 ל-50 תווים
- תיאור – אלפabetי עם כמה תווים מיוחדים כגון נקודה, פסיק, ועוד. בין 2 ל-300 תווים
- דרישות רמת טיפוס – בפורמטים לרמות טיפוס שהציגו קודם לכך
- תמונה –קובץ jpg או png

Hello, valery_admin. 1 users are logged in. ≡ Menu

The Climbing Web
by Yonatan Levey

New Group Form

Group Name:

Coach Name:

Group Description:

Bouldering Grade Requirement (V-Scale):

Sport Climbing Grade Requirement (French, capital letters):

Group Image: Choose File No file chosen

CREATE GROUP

 Support: 055-886-1427

© Yonatan Levey. All rights reserved.

דף יצירת מפגשי/אימוני טיפוס

בדף זה יכולים מאמנים ליצור מפגשים עבור הקבוצות שלהם. מנהלים כלליים יכולים ליצור מפגשים עבור כל קבוצה שיבחרו. תנאי יצירת מפגש הוא שלמהן הקבוצה אין כבר מפגש אחר בשעות שנבחרה. אם יש לו, תופיע הודעה אזהרה מתאימה.

להלן תנאי הולידציה של הטופס –

- שם קבוצה – אלפabetי, בין 2 ל-50 תווים
- תיאור – אלפabetי עם כמה תווים מיוחדים כגון נקודה, פסיק, ועוד. בין 2 ל-300 תווים
- תאריכים וזמנים – זמן סיום לפחות שעה אחרי זמן התחלתה, שני הזמןים אחריו הזמן הנוכחי
- עיר ומדינה – עיר ומדינה אמיתיים (נבדק בקדם לאחר מכן)

Hello, valery_admin. 1 users are logged in. ≡ Menu

The Climbing Web
by Yonatan Levey

New Meeting Form

Group Name:

Meeting Description:

City:

Country:

Meeting Start: dd/mm/yyyy --:--

Meeting Ending: dd/mm/yyyy --:--

CREATE MEETING

Support: 055-886-1427

© Yonatan Levey. All rights reserved.

ניהול קבוצות ומפגשי טיפוס
 דף זה מאפשר למנהל הכלליים לנהל את קבוצות הטיפוס הקיימות – לעורר אותן, למחוק אותן, או לעורר ולמחוק את פגישות הקבוצות.

Hello, valery_admin. 1 users are logged in. ≡ Menu



The Climbing Web
by Yonatan Levey

Group Management Table

[ADD NEW GROUP](#)

Editing	Deleting	Meeting Viewing	Group ID	Group Name	Coach Name	Group Description	Boulder Grade	Wall Grade Requirement	Image
EDIT	DELETE	VIEW MEETINGS	1	Da Admin Bois	valery_admin	We are the best!	V9	7C	
EDIT	DELETE	VIEW MEETINGS	2	Bouldering For Beginners	valery_admin	If you are a beginner in bouldering, feel free to join and learn how to begin your climb to success!	V0	5A	
EDIT	DELETE	VIEW MEETINGS	4	Test Group	Admin	This group is a test	V5	6B+	

[ADD NEW MEETING](#)

📞 Support: 055-886-1427
 © Yonatan Levey. All rights reserved.

ניהול משתמשים

דף זה מאפשר למנהל הכללי לנהל את משתמשי האתר. כמובן, מנהלים אלו יכולים לצפות בפרטים המלאים של כל המשתמש, לעורוך את פרטייהם, למחוק אותם, או ליצור משתמשים חדשים. העריכה ויצירה של משתמשים נעשית בהתאם לolidzity הרשמה שהוזגה קודם לכך. בנוסף, המנהלים יכולים לבקש לקבל מייל קובץ XML המכיל את טבלת נתונים המשתמשים. קובץ זה נשלח אליהם מייל דרך שירות שיצרתי שמאפשר זאת.

The screenshot shows the 'Admin Page' of 'The Climbing Web'. At the top, there's a banner with a climbing scene and the text 'Hello, valery_admin. 1 users are logged in.' and a 'Menu' button. Below the banner, the page title 'The Climbing Web' and author 'by Yonatan Levey' are displayed. The main area is titled 'Admin Page' and contains two buttons: 'CREATE NEW USER' and 'OPEN XML'. A table lists 10 user entries with columns: Editing, Deletion, Username, Password, Email, Address, Gender, Permission Level, Boulder Grade, Wall Grade, Birthdate, and City. Each row has 'EDIT USER' and 'DELETE' buttons. The users listed are: Admin, blopiblop, Bobol, bongo34, dofusdooo, fghfghfh, gghfg, goob43, GoobooGobo698, and guy8. At the bottom, there are navigation links 1, 2, and 3.

Editing	Deletion	Username	Password	Email	Address	Gender	Permission Level	Boulder Grade	Wall Grade	Birthdate	City
EDIT USER	DELETE	Admin	Admin12345	admin@gmail.com	Yigagalal Alon 37	Male	Coach	V17	9C	10/26/2004	Haifa
EDIT USER	DELETE	blopiblop	GlopDiBlop6	blopdislop@yahoo.com	Bapi Bopo 1	Female	Normal User	V3	8C+	2/8/1978	Bnei Brak
EDIT USER	DELETE	Bobol	Gogo343	fdgdg@gmail.com	Bro Code 19	Male	Normal User	V8	5C	7/12/2001	Beer Sheva
EDIT USER	DELETE	bongo34	fdfdghJJgg4	bogno@yahoo.com	Nosd 98	Male	Coach	V2	5C	8/18/1988	Jerusalem
EDIT USER	DELETE	dofusdooo	goofyGoober444	sdfsd232f@gmail.com	Gobber Street 300	Male	General Manager	V2	6B+	5/28/2012	Netanya
EDIT USER	DELETE	fghfghfh	hhJJ666gfh	ysl3728@gmail.com	Ghlobo Gabon 33	Female	Normal User	V5	5B	4/4/2009	Jerusalem
EDIT USER	DELETE	gghfg	fgh55tGG	mfsdf.sal@gmail.com	Boior 34	Female	Supply Manager	V7	6B	4/3/1992	Netanya
EDIT USER	DELETE	goob43	Pass23	yosef22@gmail.com	Somewhere 21	Female	Normal User	V5	5C+	3/2/1982	Tel Aviv
EDIT USER	DELETE	GoobooGobo698	7667JJJ77	junior233e@yahoo.com	Yigal Alon 37	Male	Normal User	V3	7B	2/6/2013	Tel Aviv
EDIT USER	DELETE	guy8	5656HHHgg	junior233e3@yahoo.com	Borohov 2	Male	Normal User	V12	7C	3/5/1991	Haifa

דף ניהול אספקה

דף זה מאפשר למנהל האספקה ולמנהל הכלליים לצפות ב מוצרים שהאתר מציע ולשלוט באספקה שלהם. המנהלים יכולים להוסיף גודל חדש בכמות ומחר לבחרתם מוצר קיימ, לעורק מחיר ואספקה של מוצר קיימ, או ליצור מוצר חדש לגמרא.

Hello, valery_admin. 1 users are logged in. ≡ Menu



The Climbing Web
by Yonatan Levey

Supply Management Table

[ADD NEW ITEM](#) | [VIEW SALES REPORT](#)

Editing	Supply Viewing	Product ID	Product Name	Company	Image
EDIT	VIEW SUPPLY	1	Scarpa Boostic	Scarpa	
EDIT	VIEW SUPPLY	2	Tomoa Narasaki Pro	Unparallel	
EDIT	VIEW SUPPLY	3	Solutions Comp Men's	La Sportiva	
		1		2	

Supply Table For Product "Tomoa Narasaki Pro"

Controls	Size	Price	Amount
EDIT RESET SUPPLY	36	400	0
EDIT RESET SUPPLY	37	458.99	9
EDIT RESET SUPPLY	38	479.72	77
ADD			

📞 Support: 055-886-1427

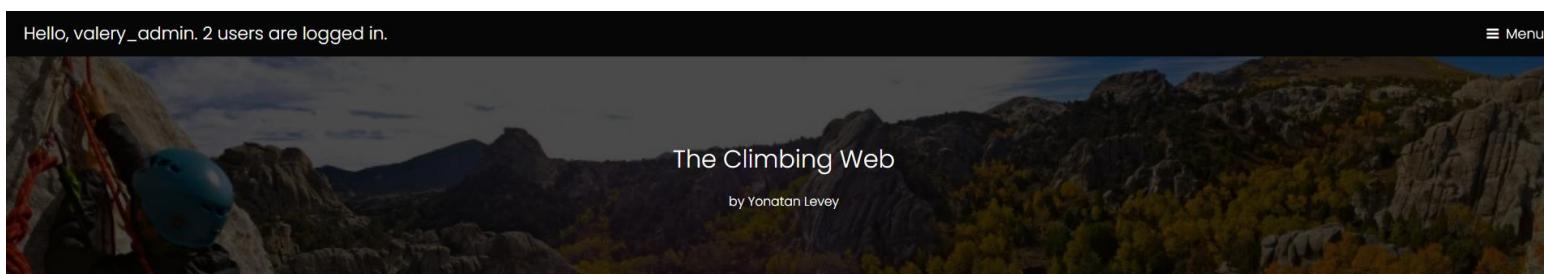
© Yonatan Levey. All rights reserved.

דף הוסף מוצר

בדף זה יכולים מנהלי האספה להוסיף מוצרים חדשים לאתר. בטופס המנהל יוצר מוצר חדש ומוסיף אותו לאספה בגודל ומחריר ראשוניים, ומשם ניתן להוסיף ולערוך גדלים נוספים של המוצר במידת הצורך. לא ניתן להוסיף מוצר עם שם של מוצר קיימ.

להלן תנאי הולידציה של הטופס –

- שם המוצר – אלפabetי, בין 2 ל-100 תווים
- חברת המוצר – אלפabetית, בין 2 ל-100 תווים
- גודל המוצר – מספר שלם, בין 30 ל-55
- מחיר – מספר דצימלי, בין 0.01 ל-9999.99
- תמונה מוצר – קובץ מסוג jpg או png



Hello, valery_admin. 2 users are logged in.

The Climbing Web
by Yonatan Levey

Menu

New Product Form

Product Name:

Company:

Size:

Price:

Product Image: Choose File No file chosen

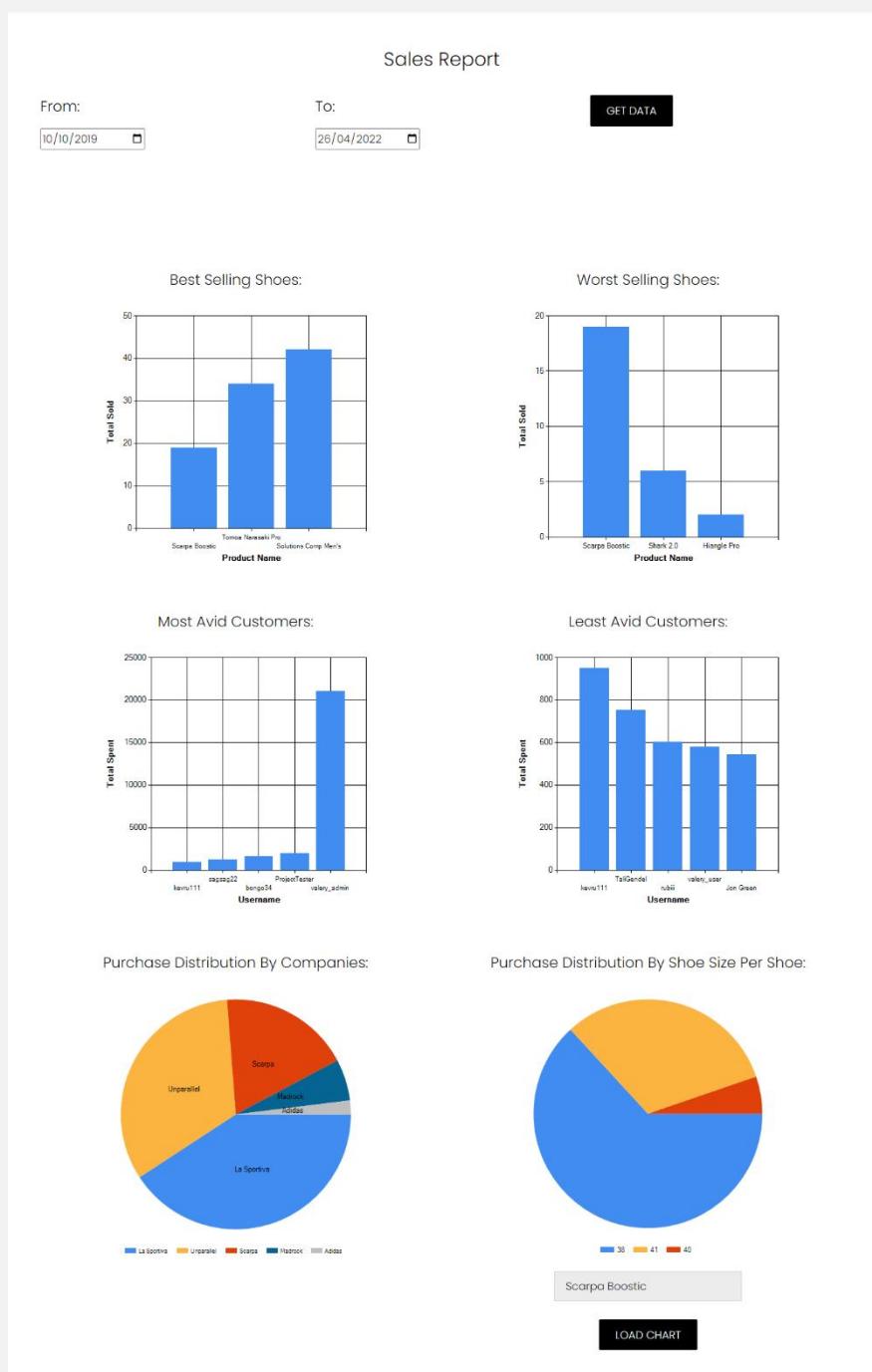
ADD ITEM

📞 Support: 055-886-1427

© Yonatan Levey. All rights reserved.

דף סטטיסטיות

בדף זה יכולים לצפות מנהלי האספקה בסטטיסטיות שונות המוגעות לחנות האתר בטוויה תאריכים לבחירתם. הסטטיסטיות שמצאותן הן – המוצרים הנמכרים ביותר, המוצרים היכי פחות נמכרים, הליקוחות שקנו בהci הרבה כ温和, הליקוחות שקנו בהci פחות כ温和, חלוקת מחירות לפי חברות, וחלוקת מחירות של מוצר ספציפי לפי גודלים. במידה ואין מידע לטוויה תאריכים מסוימים, מוצגת הודעה מתאימה.



דף הוסף כרטיס אשראי

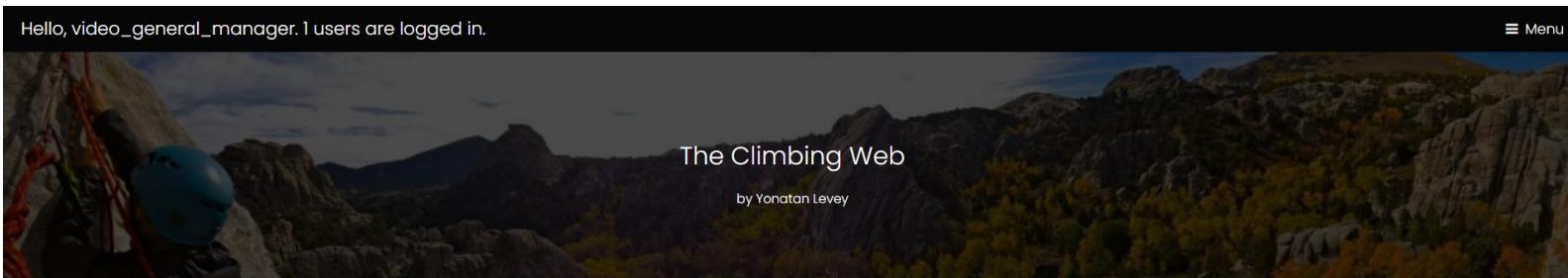
דף זה מאפשר למשתמש להוסיף כרטיס אשראי לצרטיס האשראי שהשמורים שלו באתר על מנת שיוכל להשתמש בו בקניות הבאות שלו. תנאי הולידציה לפרטי כרטיס האשראי הם כפויים לכך נטען קודם בדף סל הקניות בתשלום הסופי.

The screenshot shows a user interface for adding a new credit card. At the top, there's a header bar with the text "Hello, valery_admin. 1 users are logged in." and a "Menu" button. Below the header is a banner image of a climber on a rock face with mountains in the background. The main content area has a title "The Climbing Web" and a subtitle "by Yonatan Levey". A form titled "New Credit Card:" contains fields for "Card Number (No Spaces)" (with a placeholder input field), "CVV" (with a placeholder input field), and "Expiration Date" (with a placeholder input field). A "dd/mm/yyyy" placeholder is visible in the expiration date field. A "CREDIT CARD" icon is shown next to the expiration date field. A large "ADD CARD" button is centered below the form fields. At the bottom of the page, there's a footer with a support phone number "Support: 055-886-1427" and a copyright notice "© Yonatan Levey. All rights reserved."

דף ניהול חברים קבוצות/מפגשים

בדף זה יכולים מאמני האתר ומנהלי האתר לנהל את חברים קבוצות/מפגשי הטיפוס. מאמן ומנהל אספה יכולים לנוהל ארק ו록 את הקבוצות שלהם, אך מנהל כללי יכול לנוהל את חברי הקבוצות/מפגשים עבור כלל קבוצות הטיפוס באתר. המאמן/מנהל רשאי להויסף ולהסריר משתמשים מקבוצות/מפגשי הטיפוס שלו בהתאם לרצונו.

Hello, video_general_manager. 1 users are logged in. ≡ Menu



The Climbing Web
by Yonatan Levey

Member Management Page

Select A Type:

Group/Meeting ID:

LOAD MEMBERS

Member Data For Group 2

Controls	Username	Boulder Grade	Wall Grade
REMOVE	bongo34	V2	5C
REMOVE	valery_user	V5	6B
ADD		-	-

📞 Support: 055-886-1427
© Yonatan Levey. All rights reserved.

ארכיטקטורת הפרויקט

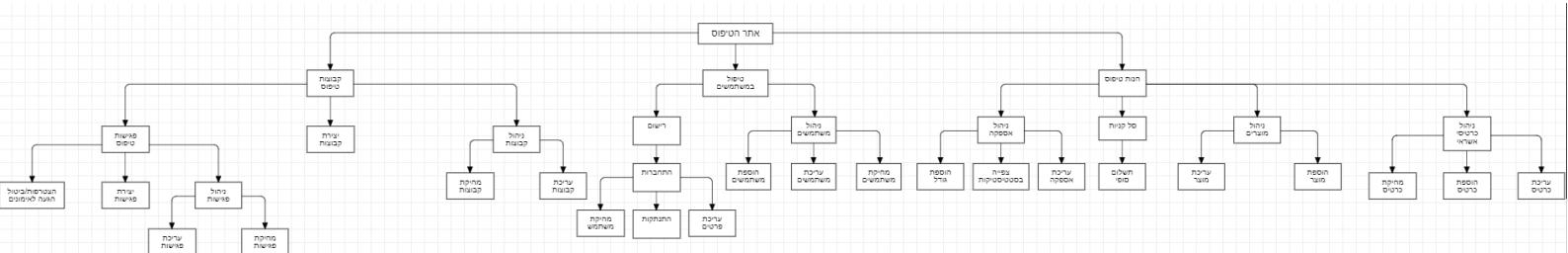
הציג הפתרון המוצע והסבירות לבחירתו

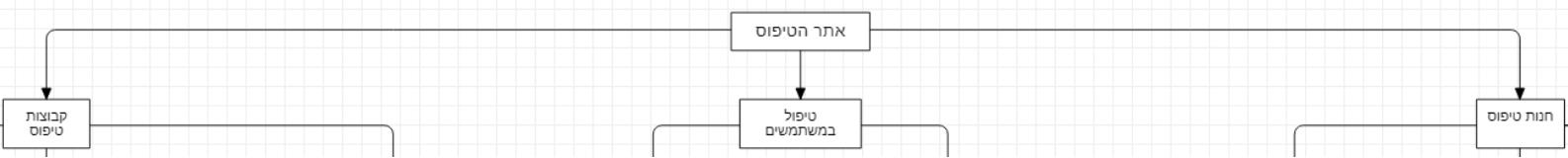
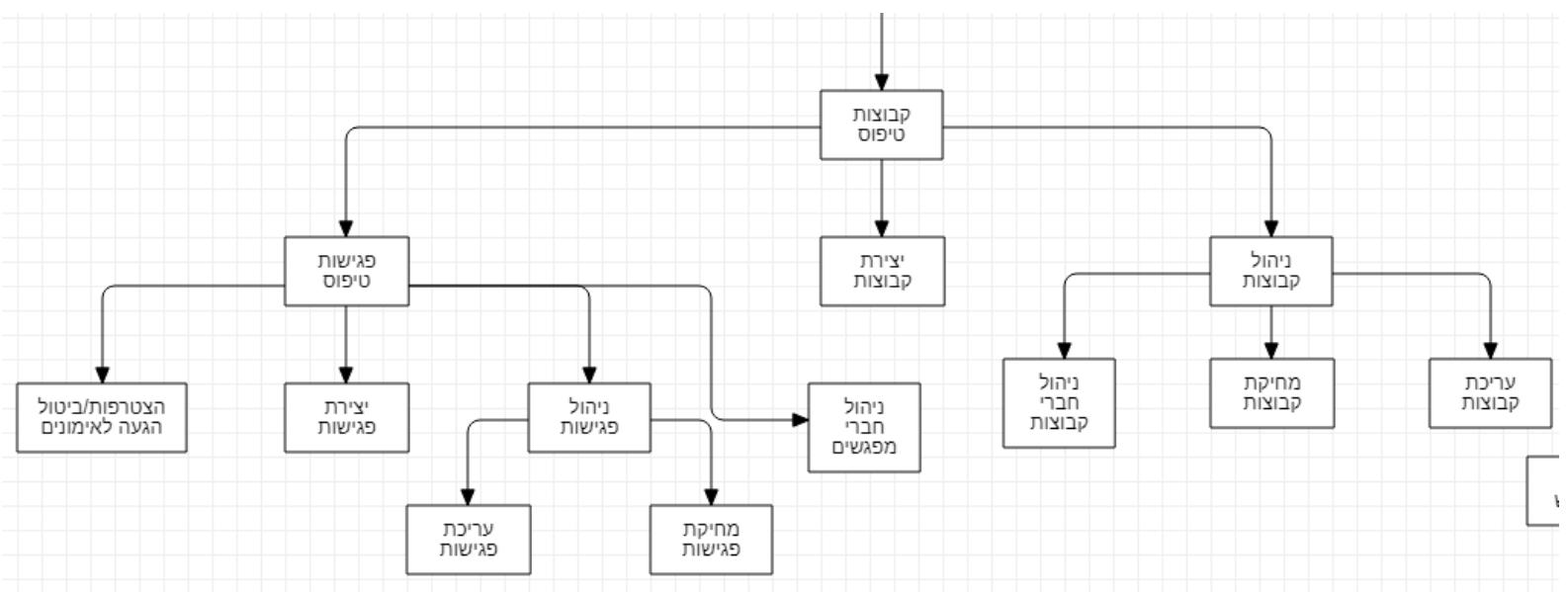
הפטרון שיצרת הוא של אטר שכלל את כל מה שמטפס ציר במקומות אחד, במקומות בכמה מקומות נפרדים ומנותקים. האטר של מספק את הקשר בין מטפסים לבין מאמנים שעוזרו להם להתקדם בטיפוס, ובנוסף לכך גם עונה על הצורך שלהם בקניית געל טיפוס, מותר הידיעה שככל שרמת המטפס מתתקדמת גם צורכי הנעלאים שלו משתנות בהתאם. האטר של הוא מוכן לשימוש וידידותי למשתמש, כדי לעודד שימוש בו ורחבה של הפלטפורמה של מאמנים-מטפסים בו.

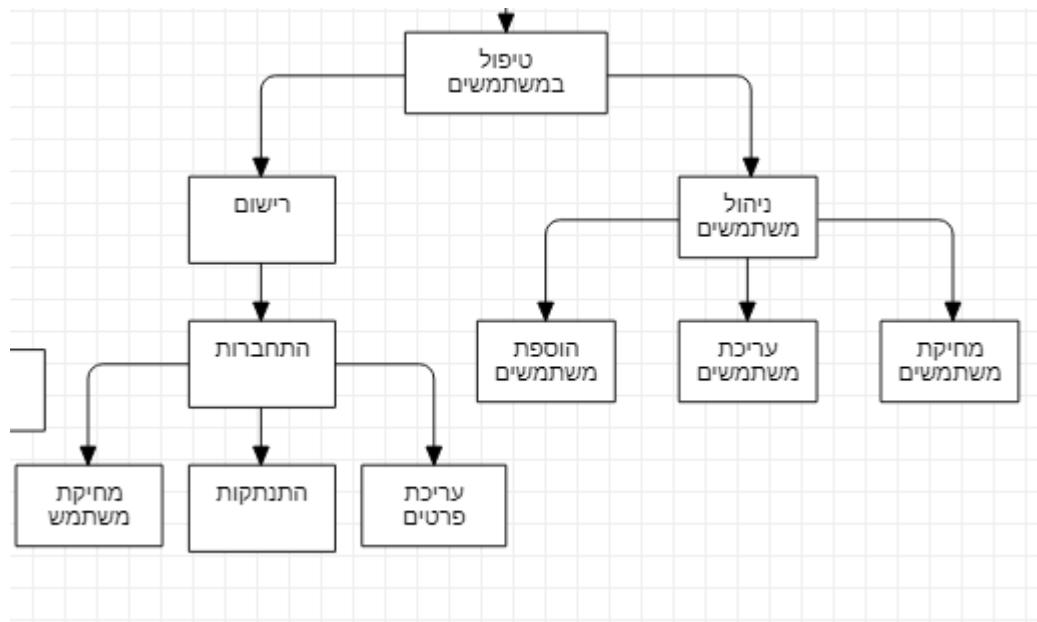
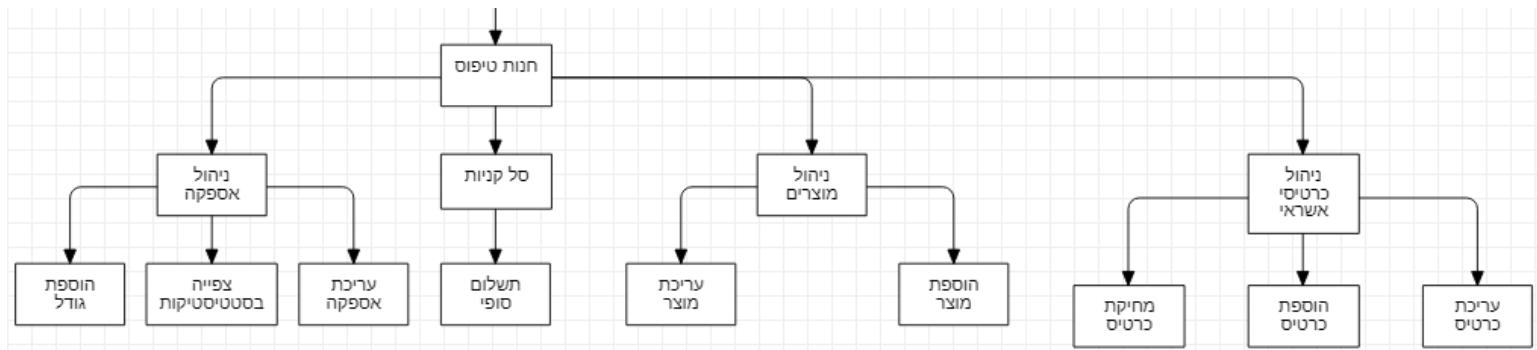
ארכיטקטורת הפתרון המוצע (Top-Down Level Design)

התרשים גדול מאוד ולכן התמונה עלולה לא להיות ברורה, אך אחלק אותו בהמשך לחלקים שונים למען נוחות הקורא.

התרשים המלא:

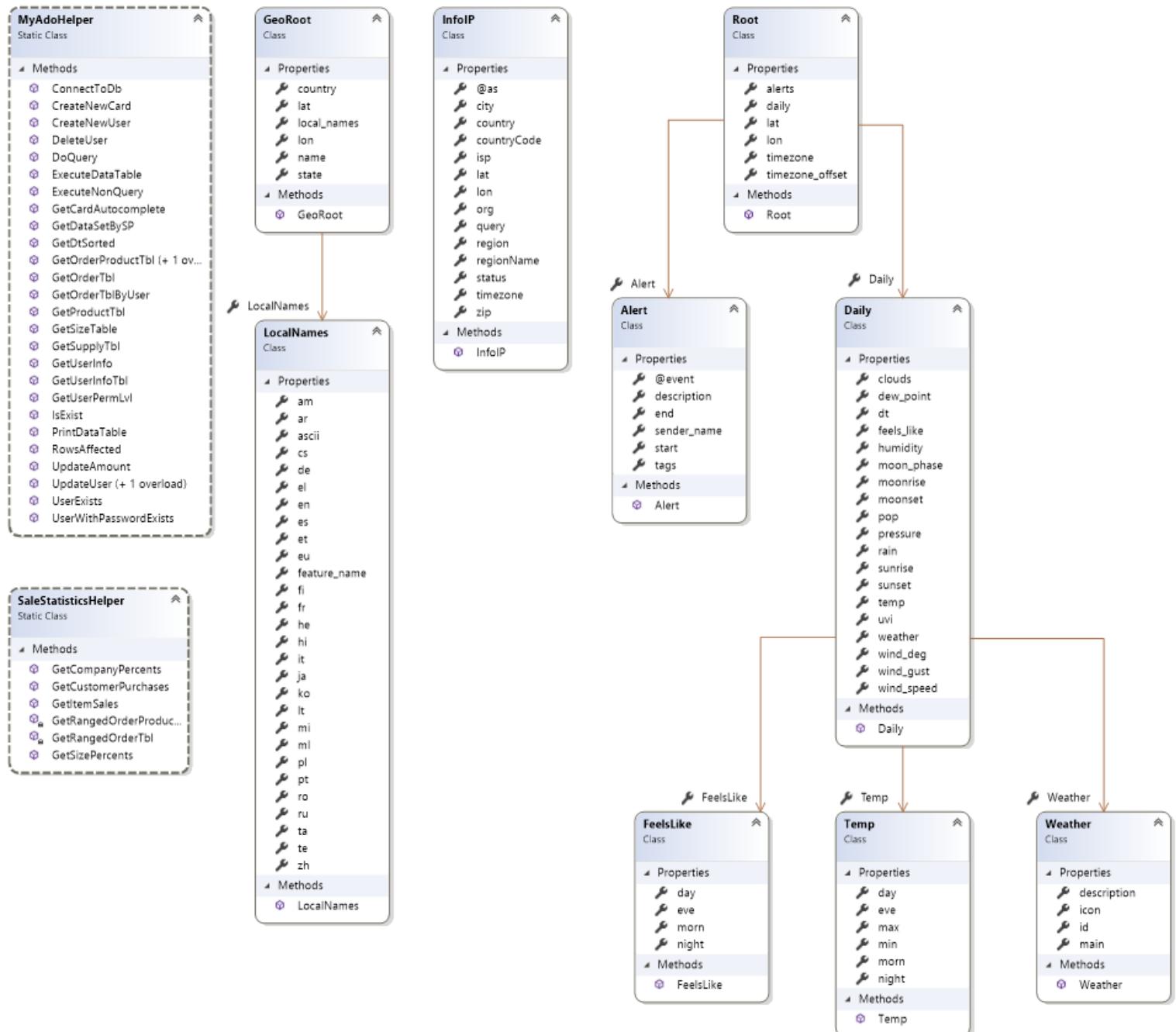


שכבות 1+2:**הרחבת קבוצות טיפוס:**

הרחבת טיפול במשתמשים:**הרחבת חנות טיפול:**

מחלקות הפרויקט

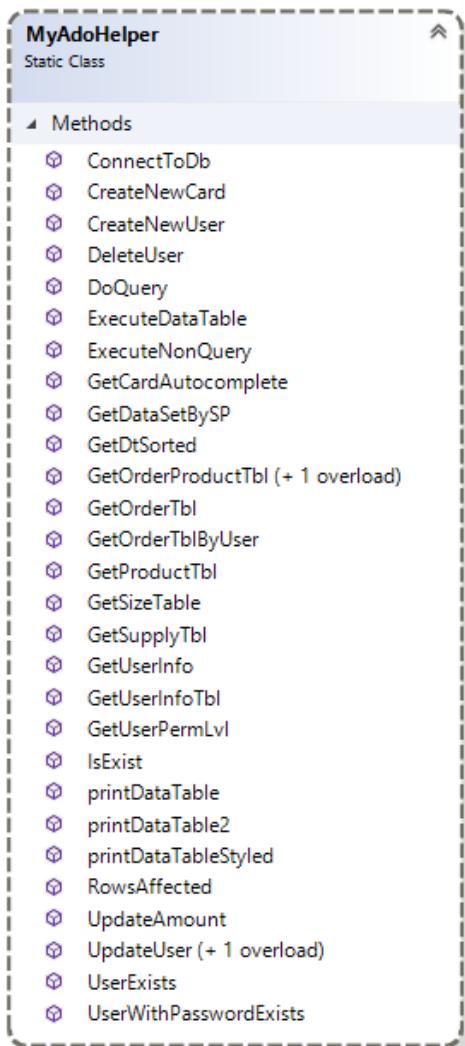
Class Diagram



תיאור המחלקות

מחלקה MyAdoHelper

מחלקה זו הינה מחלקה סטטית שמטרתה לאפשר לבדוק יעילות, נוחה, ובטוחה מול מסד הנתונים. בעזרתה חלק מהפעולות ניתן לשופק שאילתת SQL שתבוצע במסד הנתונים, ובפעולות שאבי הוסף מינית לבצע פעולות שונות מול מסד הנתונים תוך שימוש בטרנסזקציות על מנת לוודא ביצוע תקין של השאלתה במסד הנתונים ולמנוע מתקפות/תקלות חמורות.



```

public static class MyAdoHelper{
    // מקבל שם של קובץ מסד נתונים ומבצע המחברות אליו
    public static SqlConnection ConnectToDb(string fileName){...}

    // הפעולה מקבלת שם מסד נתונים ומחזירת מדיה/הוספה/עדכו ומבצעת את הפקודה על המסד הפידי
    public static void DoQuery(string fileName, string sql){...}

    // הפעולה מקבלת מסלול אל מסד הנתונים ושאיילמת המדכו, מבצעת עדכו במסד הנתונים
    public static int RowsAffected(string fileName, string sql){...}

    // הפעולה מקבלת שם קובץ ומשפט בירית נתון ומחזירה אם הנזוני קיימים ושקר אזהה
    public static bool IsExist(string fileName, string sql){...}

    // הפעולה מקבלת שאילמה ומחזירה את טבלת הנתונים שמוחדרת מסד הנתונים בעקבות ביצוע השאילחה
    public static DataTable ExecuteDataTable(string fileName, string sql){...}

    // מפעיל שאילמה שלא מחייב מישר כלשהו
    public static void ExecuteNonQuery(string fileName, string sql){...}

    // מודיע טבלת נתונים מוכנה להודפה מסד הנתונים
    public static string printDataTable(string fileName, string sql){...}

    // מקבלת שם משתמש ובודקת אם הוא קיים בטבלת המשתמשים
    public static int UserExists(string username){...}

    // יוצרת שורה חדשה לטבלת משתמשים ומחזירה אם הצליחה או לא
    public static bool CreateNewUser(string username, string password, string city, string gender, string birthdate, string boulderGrade, string wallGrade, string bio){...}

    // מוסיפה שורה חדשה לטבלת כרטיסי האשראי ומחזירה אם הצליחה או לא
    public static bool CreateNewCard(string cardNum, string username, string company, string cvv, string expireDate){...}

    // בודקת אם קיים משתמש עם הסיסמה הנתונה ומחזירה אם כן או לא
    public static int UserWithPasswordExists(string username, string password){...}

    // מודיע אם רמת הרשות של משתמש מסוים לפי קלט שם המשתמש
    public static int GetUserPermLvl(string username){...}

    // מודיע את טבלת המשתמשים
    public static DataTable GetUserInfoTbl(){...}

    // מקבלת קלט שם משתמש וומרחת אותו לטבלת המשתמשים
    public static bool DeleteUser(string username){...}

    // מקבלת קלט פרטי משתמש חדשים ומודכנת את נתוני המשתמש הנוכחי
    public static bool UpdateUser(string username, string password, string email, string address, string boulderGrade, string wallGrade, string birthDate){...}

    // קבלת קלט פרטי משתמש חדשים ומגדכנת את נתוני המשתמש הנוכחי, אך ללא שינוי הרשאות הקיימת
    public static bool UpdateUser(string username, string password, string email, string address, string boulderGrade, string wallGrade, string birthDate){...}

    // מודיעה את טבלת המשתמשים ממויינת בסדר עליה/ירוד על פי העומدة שבקלט
    public static DataTable GetDtSorted(string sortParam, string sortDirection){...}

    // מודיעה את השורה מטבלת המשתמשים של משתמש מסוים לפי קלט
    public static DataTable GetUserInfo(string username){...}

    // מודיעה את השלמה של כרטיסי האשראי של המשתמש בקלט
    public static DataTable GetCardAutocomplete(string username, string cardNum){...}

    // מודיעה את טבלת המוצרים על המסך המוזרים
    public static DataTable GetProductTbl(){...}

    // מודיעה את טבלת האספקה
    public static DataTable GetSupplyTbl(){...}

    // מודיעה את טבלת פרטי הזמן של הזמנה לפי מסה הזמן
    public static DataTable GetOrderProductTbl(int orderId){...}

    // מודיעה את טבלת הזמן
    public static DataTable GetOrderTbl(){...}

    // מודיעה את טבלת פרטי הזמן
    public static DataTable GetOrderProductTbl(){...}

    // מודיעה את טבלת מוצר מוצר של הזמנה לפי מסה הזמן
    public static DataTable GetOrderProductTbl(){...}

    // מודיעת אם מוצר מוצר מסוים בטבלת האספקה
    public static bool UpdateAmount(int orderId, int productId, double size, int amount){...}

    // מודיעה את טבלת הזמינות של מוצר מסוים
    public static DataTable GetOrderTblByUser(string username){...}

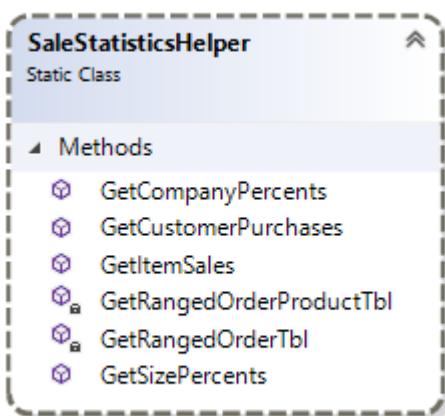
    // מודיעה דאטסט לפי ביצוע פטולה שモירה שמה נזון בקלט
    public static DataSet GetDataSetBySP(string sp){...}

    // מודיער את טבלת הגודלים של מוצר מסוים בטבלת האספקה
    public static DataTable GetSizeTable(int productId){...}
}

```

מחלקה SaleStatisticsHelper

מחלקה זו הינה גם היא מחלוקת סטטיטית, שנועדה לאפשר עבודה עילית בניתוח הסטטיסטיות של מכירות החנות באתר. המחלוקת מאפשרת לקבל נתונים מדויקים על מגוון משתנים בנסיבות האתר בטוויה תאריכים מסוימים, שמאפשר בהמשך ליצור תצוגה מוחה וモובנת של הנתונים עבור המנהל שצופה בהם.



```

public static class SaleStatisticsHelper
{
    // הפעולה מקבלת טווח תאריכים ומזהירה את טבלת החזנות בין התאריכים הנוכחיים להלה
    private static DataTable GetRangedOrderTbl(DateTime startDate, DateTime endDate)...

    // הפעולה מקבלת טווח תאריכים ומזהירה את טבלת פרטיה החזנות בין התאריכים הנוכחיים להלה
    private static DataTable GetRangedOrderProductTbl(DateTime startDate, DateTime endDate)...

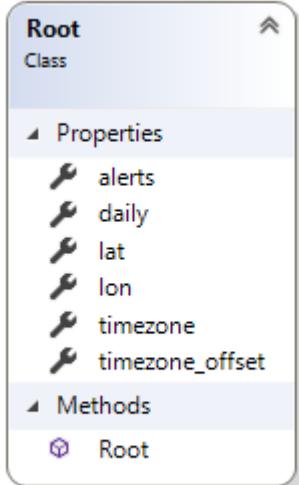
    // המחלוקת מודירה את מספר המוצרים וכי או וכי פחות נמכרים בטווח התאריכים הנוכחיים
    public static Dictionary<string, int> GetItemSales(int amount, bool best, DateTime startDate, DateTime endDate)...

    // הפעולה מודירה את מספר המכירות של מוצרים מהחברות השונות בטווח התאריכים הנוכחיים
    public static Dictionary<string, int> GetCompanyPercents(DateTime startDate, DateTime endDate)...

    // הפעולה מודירה את מספר המכירות של כל גודל של מוצר מסוים בטווח התאריכים הנוכחיים
    public static Dictionary<int, int> GetSizePercents(int productId, DateTime startDate, DateTime endDate)...

    // הפעולה מודירה את מספר הלקוחות שרכשו באתר וכי הרבה או וכי פחות כפף בטווח התאריכים הנוכחיים
    public static Dictionary<string, double> GetCustomerPurchases(int amount, bool best, DateTime startDate, DateTime endDate)...
}
  
```

מחלקה Root



מחלקה זו מכילה מידע שנוצר מדה-סריאליזציה של מידע שהתקבל משירות רשות בפורמט Json, ולכן למחלקה עצמה אין פעולות מלבד הפעולה הבונה, אלא נעשה בה השימוש ב-properties. מחלקה זו היא "מחלקת האם" של המידע שהתקבל ומכילה משתנים שהם אובייקטים ממחלקות אחרות הקשורות למידע זה.

```

public class Root
{
    public double lat { get; set; }
    public double lon { get; set; }
    public string timezone { get; set; }
    public int timezone_offset { get; set; }
    public List<Daily> daily { get; set; }
    public List<Alert> alerts { get; set; }

    //בנינה ברירת מחדל
    public Root()...
}
  
```

מחלקה Daily

מחלקה זו מכילה מידע שנוצר מדה-סרייאלייזציה של מידע שהתקבל משירות רשות השת בפורמט JSON, ולכן למחלקה עצמה אין פעולות מלבד הפעולה הבונה, אלא נעשה בה השימוש Properties. מחלקה זו מכילה את המידע לכל יום של תחזית מזג האוויר.

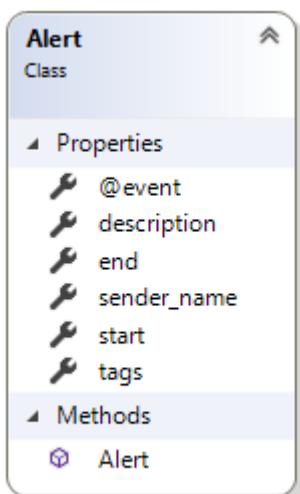


```

public class Daily
{
    public int dt { get; set; }
    public int sunrise { get; set; }
    public int sunset { get; set; }
    public int moonrise { get; set; }
    public int moonset { get; set; }
    public double moon_phase { get; set; }
    public Temp temp { get; set; }
    public FeelsLike feels_like { get; set; }
    public int pressure { get; set; }
    public int humidity { get; set; }
    public double dew_point { get; set; }
    public double wind_speed { get; set; }
    public int wind_deg { get; set; }
    public double wind_gust { get; set; }
    public List<Weather> weather { get; set; }
    public int clouds { get; set; }
    public double pop { get; set; }
    public double rain { get; set; }
    public double uvi { get; set; }

    //טבלה בונה ברירת מחדל
    public Daily()...
}

```



מחלקה Alert

מחלקה זו מכילה מידע שנותר מדה-סריאליזציה של מידע שהתקבל משירות רשות בפורמט JSON, ולכן למחלקה עצמה אין פעולות מלבד הפעולה הבונה, אלא נעשה בה השימוש Properties. מחלוקת זו מכילה את המידע לכל יומם של הודעות חשובות לגבי מזג האוויר.

```

public class Alert
{
    public string sender_name { get; set; }
    public string @event { get; set; }
    public int start { get; set; }
    public int end { get; set; }
    public string description { get; set; }
    public List<string> tags { get; set; }

    //נוולות בונה בירית מזוודה
    public Alert()...
}
  
```

Weather
Class

- ◀ Properties
 - 🔑 description
 - 🔑 icon
 - 🔑 id
 - 🔑 main
- ◀ Methods
 - ⌚ Weather

מחלקה Weather

מחלקה זו מכילה מידע שנוצר מדה-סרייאלייזציה של מידע שהתקבל ממשירות רשת בפורמט Json, ולכן למחלקה עצמה אין פעולות מלבד הפעולה הבונה, אלא נעשה בה השימוש ב-properties. מחלקה זו מכילה את המידע לכל יום של תיאור וアイיקון מזג האוויר.

```
public class Weather
{
    public int id { get; set; }
    public string main { get; set; }
    public string description { get; set; }
    public string icon { get; set; }

    // פטולה בונה ברירתם חזק!
    public Weather()...
}
```

מחלקה Temp

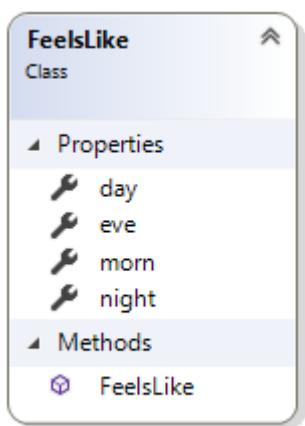
מחלקה זו מכילה מידע שנוצר מדה-סרייאלייזציה של מידע שהתקבל משירות רשות בפורמט JSON, ולכן למחלקה עצמה אין פעולות מלבד הפעולה הבונה, אלא נעשה בה השימוש Properties. מחלקה זו מכילה את המידע לכל יום של הטמפרטורות במהלך היום.

```

public class Temp
{
    public double day { get; set; }
    public double min { get; set; }
    public double max { get; set; }
    public double night { get; set; }
    public double eve { get; set; }
    public double morn { get; set; }

    // Constructor
    public Temp()...
}

```



מחלקה FeelsLike

מחלקה זו מכילה מידע שנוצר מדה-סריאליזציה של מידע שהתקבל ממשירות רשות בפורמט Json, ולכן למחלקה עצמה אין פעולות מלבד הפעולה הבונה, אלא נעשה בה השימוש ב-properties. מחלקה זו מכילה את איך שהטמפרטורות מרגישות במהלך היום.

```
public class FeelsLike
{
    public double day { get; set; }
    public double night { get; set; }
    public double eve { get; set; }
    public double morn { get; set; }

    // Constructor
    public FeelsLike()...
}
```

פעולה בונה בירית מזוין//

public FeelsLike()...

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

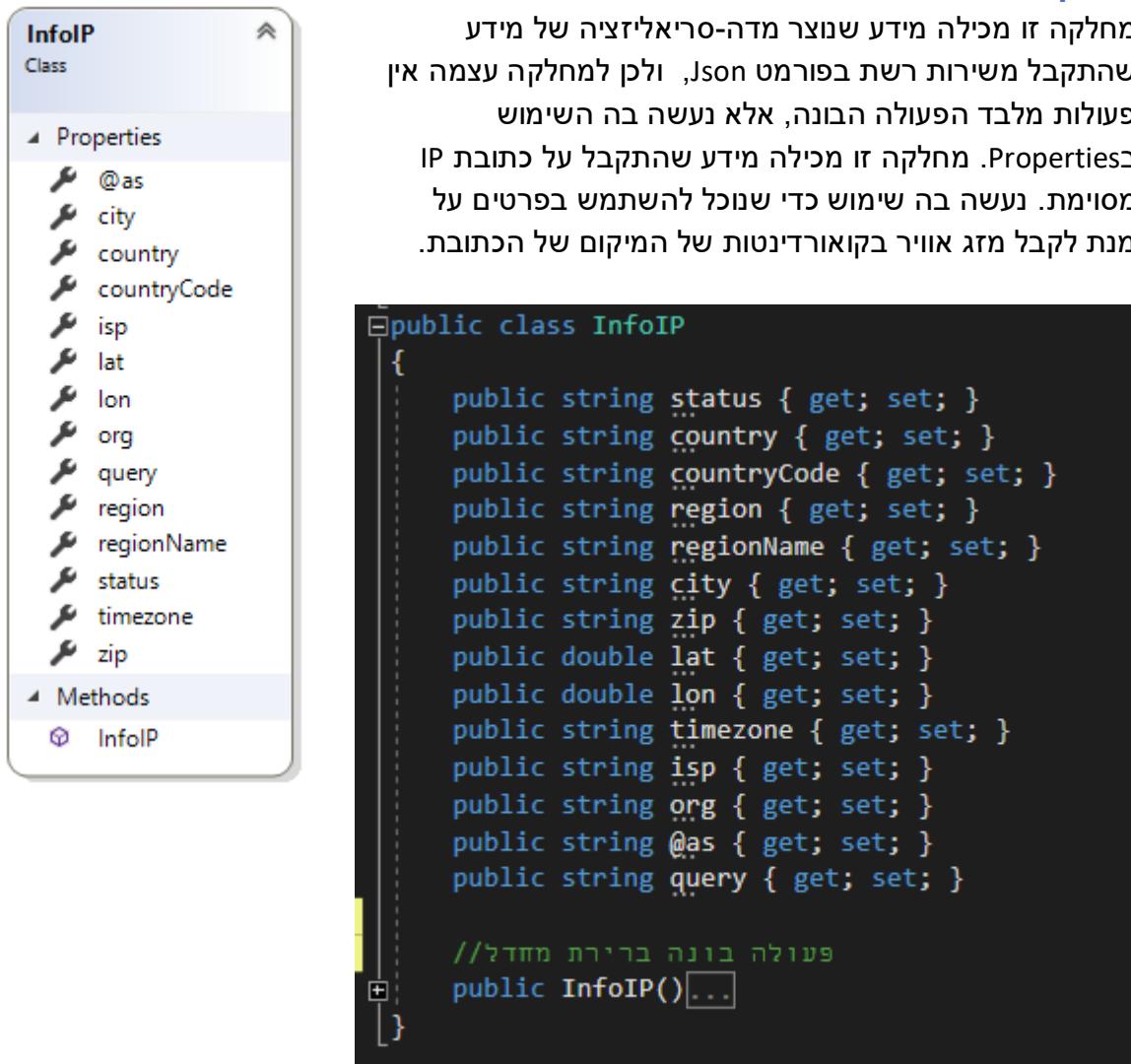
}

}

}

מחלקה InfoIP

מחלקה זו מכילה מידע שנוצר מדה-סריאליזציה של מידע שהתקבל משירות רשת בפורמט JSON, ולכן למחלקה עצמה אין פעולות מלבד הפעולה הבונה, אלא נעשה בה השימוש Properties. מחלקה זו מכילה מידע שהתקבל על כתובת IP מסוימת. נעשה בה שימוש כדי שנוכל להשתמש בפרטים על מנת לקבל מידע א�ויר בקואורדינטות של המיקום של הכתובת.



The screenshot shows a UML class diagram on the left and its corresponding C# code on the right. The class 'InfoIP' has properties: @as, city, country, countryCode, isp, lat, lon, org, query, region, regionName, status, timezone, and zip. It also has a constructor 'InfoIP()'.

```

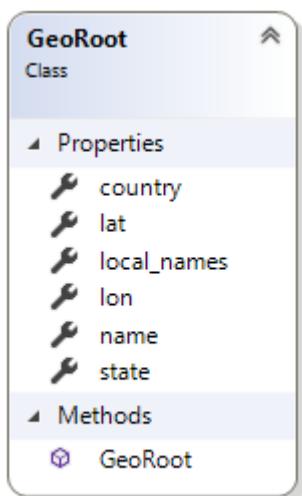
public class InfoIP
{
    public string status { get; set; }
    public string country { get; set; }
    public string countryCode { get; set; }
    public string region { get; set; }
    public string regionName { get; set; }
    public string city { get; set; }
    public string zip { get; set; }
    public double lat { get; set; }
    public double lon { get; set; }
    public string timezone { get; set; }
    public string isp { get; set; }
    public string org { get; set; }
    public string @as { get; set; }
    public string query { get; set; }

    // פונקיה בונה ברייתן חזק
    public InfoIP()...
}

```

מחלקה GeoRoot

מחלקה זו מכילה מידע שנוצר מדה-סריאליזציה של מידע שהתקבל ממשירות רשת בפורמט Json, ולכן למחלקה עצמה אין פעולות מלבד הפעולה הבונה, אלא נעשה בה השימוש באמצעות Properties. מחלקה זו מכילה מידע שהתקבל על עיר ומדינה מסוימת. נעשה בה שימוש כדי שטכני להשתמש בפרטים על מנת לקבל מזג אויר בקואורדינטות של המיקום.



```

public class GeoRoot
{
    public string name { get; set; }
    public LocalNames local_names { get; set; }
    public double lat { get; set; }
    public double lon { get; set; }
    public string country { get; set; }
    public string state { get; set; }

    // Constructor
    public GeoRoot()...
}
  
```

מחלקה LocalNames

מחלקה זו מכילה מידע שנותר מדה-סריאליזציה של מידע שהתקבל משירות רשת בפורמט JSON, ולכן למחלקה עצמה אין פעולות מלבד הפעולה הבונה, אלא נעשה בה השימוש Properties. מחלקה זו נועדה לשמר את השמות המוקומיים של עיר מסוימת במגוון שפות שונות.



```

public class LocalNames
{
    public string fi { get; set; }
    public string pl { get; set; }
    public string te { get; set; }
    public string ru { get; set; }
    public string mi { get; set; }
    public string fr { get; set; }
    public string feature_name { get; set; }
    public string hi { get; set; }
    public string eu { get; set; }
    public string ml { get; set; }
    public string ro { get; set; }
    public string et { get; set; }
    public string ta { get; set; }
    public string es { get; set; }
    public string he { get; set; }
    public string it { get; set; }
    public string en { get; set; }
    public string pt { get; set; }
    public string am { get; set; }
    public string ko { get; set; }
    public string ja { get; set; }
    public string ascii { get; set; }
    public string ar { get; set; }
    public string cs { get; set; }
    public string el { get; set; }
    public string zh { get; set; }
    public string de { get; set; }
    public string lt { get; set; }

    // פונקיה בונה ברירת מחדל
    public LocalNames()
    {
        //
        // TODO: Add constructor logic here
        //
    }
}
  
```

מבנה נתונים בהם נעשה שימוש

בפרויקט נעשה שימוש במבנה נתונים הבאים:

- רשימה - List
- מילון - Dictionary
- מערך – Array

מבנה נתונים מהווים חלק קרייטי מהתכנות בפרויקט ואפשרו לי לעבודה נוחה וგמישה עם נתונים שונים.

List

רשימה הינה אוסף גנרי שמאפשר שמירה ושליפה של אובייקטים לפי אינדקס. היתרון המשמעותי בשימוש במבנה נתונים זה מעלה מערך רגיל הוא האורך הדינמי שלו והטוחות הכלילית בשימוש בו (למשל, כדי להכניס איבר באמצעות הערך להציג את כל האיברים האחרים כמו במערך). החיסרון הוא שבגלל שלחוב רשימות ממושכות כרך שכלל חוליה מצבעה על זו שאחריה, שליפת נתונים מרשימה יותר איטית מאשר מערך. דוגמה לשימוש במבנה נתונים זה הוא בקוד דף סל הקניות, בו שמרתי גDAL נעלאים שאין מספיק מהן במלאי ושמות של מנהלים ברשומות.

בailo מקרים עדיף להשתמש ברשימה ולא במערך? במקרים שאינם יודעים מראש את גודל הרשימה, ובמקרים שנוצרך לבצע שליפות והכנסות מידע מקומיים משתמשים ברשימה. במקרים שאנו מבצעים בקוד גישות רבות למיקומים ספציפיים, עדיף להשתמש במערך על מנת לשמר על ייעילות הקוד.

Dictionary

מילון (Dictionary) הינו אוסף של זוגות מפתח-ערך (key-value), ולרוב אפשר ערך יחיד לכל מפתח. יתרונותיו העיקריים הינם הכנסה ושליפה יעילים של מידע במבנה נתונים. המפתחות והערכים יכולים להיות מטיפוסים שונים (למשל, המפתח מסוג `to` והערך מסוג `.string`).

השתמשתי במילון בפעולות שיצרתי במחלקה `SaleStatisticsHelper` על מנת להחזיר עבור כל שם מוצר/חברה/לקוח מידע רלוונטי עבורו בזוגות.

Array

מערך (Array) הוא טיפוס נתונים המכיל מספר משתנים/אובייקטים מסווג קבוע שנשמרים בזיכרון באופן רציף, כלומר אחד אחרי השני. כאשר מגדרים משתנה מסווג מערך אמן נדרשים לציין את גודל המערך (מספר האיברים השמורים בו), את סוג המשתנים/אובייקטים, ואת שם המערך. היתרון בשימוש במערך הוא שבגלל הצורה שהוא נשמר בזיכרון, הגישה אליו יעלה מאד ובמקרים של מערכים גדולים מאוד חוסכת בזמן רב (לעומת השימוש במבנה נתונים כגון רשימה). החסרון הוא שהמערך הוא בעל גודל שנקבע מראש ולא גמיש מאד, ולכן הוא משמש לשימרת נתונים מכמות ידועה מראש ולשם מטרה ספציפית.

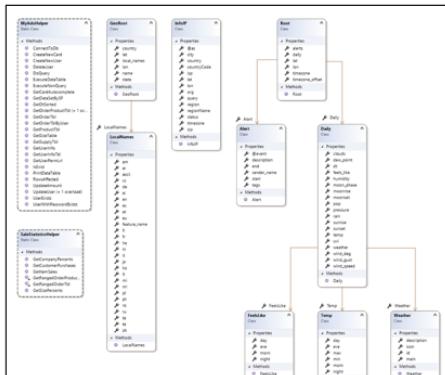
בפרויקט שלי אחד השימושים שעשייתי במערך הוא עבור שימרת הנתונים של צירי ה-X וה-Y של הגרפים בדף הסטטיסטיות.

דרימת מידע וקשרים בין היחידות השונות

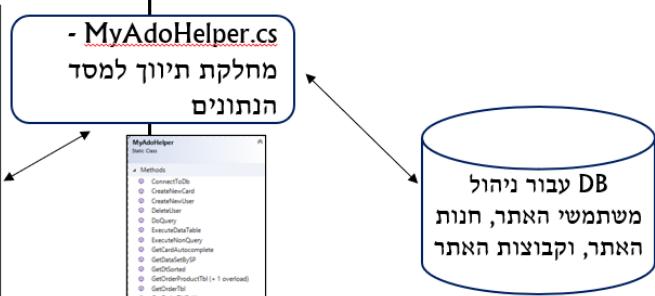
Presentation tier



Application tier



Data tier



בנוסף, מחלקות קוד
אחרוי של קבצי ASP

אפליקציית WS שפיתחתי

2 שירותים SOAP בפרויקט
נפרד שפיתחתי עבור שליחת
מיילים וחישוב נתונים על ידי
אלגוריתם

- EmailService.cs
מחלקה לוגינה
לשילוח אימיילים

קובץ XML עבור
שמירת מידע שיישלה
באימיילים

ארQUITקטורת רשת

תיאור פרוטוקולי התקשורת

באופן נעשה שימוש בפרוטוקולי התקשורת הבאים:

- HTTP
- SOAP
- SMTP

האתר עצמו מבוסס על תקשורת HTTP, ולמרות שכרגע האתר למעשה רץ על המחשב הפרטี้ שלו נמצוא השירות, הוא תיאורתי ועושה שימוש ב프וטוקול HTTP.

שירות הרשות שיצרת בפרויקט הוא שירות רשות SOAP (Simple Object Access Protocol), וכן השירות בין לבין השירות נעשית תוך שימוש בפרוטוקול זה.

מערכת שליחת המיילים שיישמשי באתר של משתמש ב-SMTP (Simple Mail Transfer Protocol).

אזורים הדורשים אבטחה

בקורת גישה לאתר - באתר ישן רמות הרשות שונות, שמטרתן להבטיח כי משתמש בעל רמת הרשות מסוימת לא יגש לתוכנים רגילים שלא נועד לו והוא עלול לנצל לרעה. אך, בסיסד הנתונים נשמרים כ-4 מפתחות לרמות הרשות, והם:

- 0 – רמת הגישה הבסיסית ביותר, ניתנת לכל משתמש חדש שנרשם. משתמש זה יכול רק לבצע פעולות טנגעות לעצמו ולא להשפיע על האתר באופן מהותי.
- 1 – רמת הגישה שנדרשת לאמנים. אלה משתמשים שהמנחים סומכים עליהם יותר, אך גם באמנים לא מקבלים גישה לתוכנים רגילים באתר והבדל היחיד בין ובין משתמשים רגילים הוא שהם יכולים לנהל אימונים.
- 2 – רמת הגישה שנדרשת למנהל אספקה באתר. מנהלי אספקה באתר רשאים להשפיע על האתר דרך ניהול המוצרים והאספקה באתר.
- 3 – רמת הגישה שנדרשת למנהל כללים באתר. מנהלים אלו רשאים לגשת לכל תוכני האתר ולנהל משתמשים אחרים.
-

חשבונות משתמשים - כדי שצוין לעיל, רק מנהל כללי יכול לנוהל את חשבונות המשתמשים (למחוק, ליצור, ולערוך). כל משתמש יכול לנוהל את החשבון הפרטี้ שלו (להתחבר, להתנתק, למחוק, ולערוך), אך לא מעבר לכך.

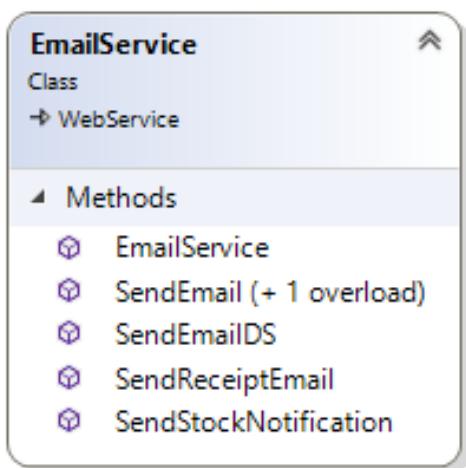
מאגרי מידע – לאף משתמש באתר אין מגע ישיר עם מאגרי המידע. בין שכבת הליקוי לשכבות המידע מפרידה שכבת הלוגיקה, שמנועת מהמעגנים לבצע בסיסד הנתונים פעולות שהן מעבר לאלו שניתנות להם כתלות בהרשאת הגישה שלהם.

שירותי רשת שהפרויקט מספק

שליחת אימיילים – שירות רשת מסוג SOAP

שירות רשת זה שיצרתי מאפשר דרך שליחת מיילים דרך חשבון מייל שיצרתי עבור האתר. שירות זה מאפשר לשלוח אימייל ולקבוע בו את התוכן, כותרת, נמענים, וקבצים שיש לצרף. השתמשתי בשירות זה בשבייל שליחת קבוצות ללקוחות באתר, ושליחת התראות ומידע על האתר לאדמינים.

בשירות זה עשית שימוש בספריית MailKit, שמספקת כלים לשילוח מיילים דרך פרוטוקול SMTP.



```

public class EmailService : System.Web.Services.WebService
{
    public EmailService()...

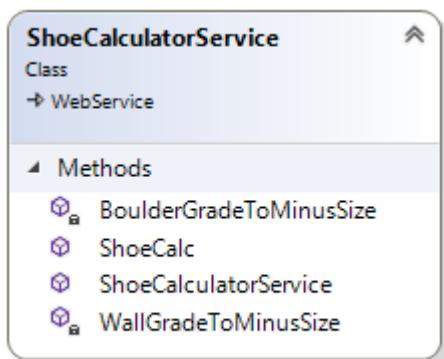
    [WebMethod]
    [Description("Item out of stock notifier")]
    public void SendStockNotification(string username, List<string> adminEmailList, List<string> adminNameList, List<string> outOfStockItemList, List<int> outOfStockSizeList)...
        מקבל רשימה של שמות מנהליים וסוצרים שמחמם מנהט לבנות ומודיעו אותו שאנו מספיק מה מוצרים באספנות וצריך לעדכן אותו

    [WebMethod]
    [Description("Email sender")]
    public string SendEmail(string content, string subject, string target)...
        שולח אימייל עם התוכן, נושא, ונתן שהתקבלו במייל

    [WebMethod]
    [Description("Receipt Sender")]
    public string SendReceiptEmail(DataSet infoDs, string target, string username)...
        שולח אימייל אליו נדרש לתקן עם פרטי החומרה

    [WebMethod]
    [Description("Send dataset through email")]
    public string SendEmailDS(string subject, string username, string target, string content, DataSet ds)...
        XML שולח אימייל אליו נדרש דataset בפורמט

    [WebMethod]
    [Description("Send email with attachments")]
    public string SendEmail(string content, string subject, string target, string[] @attachments)...
        שולח אימייל יחד עם הקבצים שהתקבלו מתוך תיבות ובילט
}
  
```



מחשבון נעליים – שירות רשת מסוג SOAP

שירות רשת זה שיצרתי מספק פונקציה שמקבלת פרטיהם על משתמש ועל ההעדפות שלו לגבי נעלי טיפוס, ומחשב את גודל נעלי הטיפוס האידיאלי עבור משתמש זה.

```

public class ShoeCalculatorService : System.Web.Services.WebService
{
    // פעולה בונה בירית מזוין//
    public ShoeCalculatorService()...

    // מקבל נתונים על משתמש והעדפות לנעל, ולפיכך מחשב גודל נעל טיפוס עבור המפתח//
    [WebMethod(Description = "Shoe size calculator")]
    public int ShoeCalc(int normalSize, string company, string shoeFit, string boulderGrade, string wallGrade)...

    // ממיר מרמת טיפוס לכמות הורדה בנעל//
    private int BoulderGradeToMinusSize(string boulderGrade)...

    // ממיר מרמת טיפוס לכמות הורדה בנעל//
    private int WallGradeToMinusSize(string wallGrade)...
}

```

שירותי רשות שימוש האתר

בדיקות כרטיסי אשראי – שירות רשות מסוג SOAP

מקור - <https://ws.cdyne.com/creditcardverify/luhnchecker.asmx?WSDL>

שירותות הרשות כולן פועלות מרכזית אחת, CheckCC, המתקבלת מספר כרטיס אשראי ומחזירה אובייקט עם שני המאפיינים הבאים:

- – שומר את חברת כרטיס האשראי. במידה והחברה לא ידועה ערכו "Unknown"
- – שומר ערך בולייני המבטא אם מספר כרטיס האשראי תקין או לא

דוגמה לישום השירות בקוד:

```
OnlineServiceReference.LuhnCheckerSoapClient client = new OnlineServiceReference.LuhnCheckerSoapClient();

bool cardValid = client.CheckCC(cardNum).CardValid;
string company = client.CheckCC(cardNum).CardType;

if (cardValid == false || company.Equals("Unknown"))
{
    errorMsg = "The credit card you entered is invalid!";
    return;
}
```

שירות רשות חיצוני – מזג אוויר ופרטים על מיקום

פרויקט שלי עשייתי שימוש בשירות הרשות שספק האתר

<https://openweathermap.org>

שירות זה מאפשר, באמצעות WebClient יחיד עם מפתח API שמקבלים בעת ההרשמה, להוריד מידע על מזג אוויר ומיקום בפורמט JSON.

השתמשתי בשירות הרשות על מנת למצוא את הקואורדינטות של עיר במדינה מסוימת, ותוך שימוש בערכי הקואורדינטות קיבל את תחזית מזג האוויר לשבוע הקרוב במיקום מסוים.

דוגמה לישום השירות בקוד:

```
url = string.Format("http://api.openweathermap.org/geo/1.0/direct?q={0}&appid={1}", query, weatherApiKey);

string json = client.DownloadString(url);
GeoRoot geoRoot = JsonConvert.DeserializeObject<GeoRoot>(json.Substring(1, json.Length - 2));
```

מציאת פרטי מיקום לפי כתובת IP – שירות רשות מסוג REST API

בנוסף למציאת הפרטים על מיקום לפי שם העיר והמדינה, השתמשתי ב프וטוקול החינמי שאיפילו לא מצירף הרשמה מהאתר <https://ip-api.com>. שירות זה מאפשר לספק לכתובת IP בלבד ולקבל מידע על המיקום של כתובת ה-IP בפורמט JSON.

דוגמה לישום השירות בקוד:

```
string locationUrl = string.Format("http://ip-api.com/json/{0}", userIP);
string json = client.DownloadString(locationUrl);
InfoIP locInfo = JsonConvert.DeserializeObject<InfoIP>(json);
```

טכנולוגיות בפרויקט

SQL Server – זהה מערכת לניהול בסיסי נתונים שפותחה על ידי מיקרוסופט. הפקציה העיקרית של מוצר תכונה זה היא אחסון ואחסור נתונים בהתאם לצרכים של מרכיבים אחרים שימושיים בנזונים. בפרויקט שלי ישתי את מסד הנתונים שלי בSQL Server ובעודו מאוחסנים רוב טבלאות הנתונים עבור האתר שלי.

XML – XML, בדומה ל-HTML, היא שפת תגיות, אך בשונה מ-HTML אין בה תגיוט מוגדרות מראש. במקרים זאת, המשתמש עצמו מגדר את התגיוט בהתאם לצרכים שלו. תכונה זו של השפה מקנה לה גמישות כמעט בלתי מוגבלת ושימושיות רבה באחסון נתונים. מכיוון של XML קיים פורמט סטנדרטי, ניתן לשתף אותו בקלות בין פלטפורמות שונות. באתר שלי השתמשתי ב-XML בשירות הרשות שיצרתי כדי לשמר נתונים מטבלאות ולשלוח אותם כקובץ XML.

ADO.NET – זהה קבוצה רחבה של מחלקות המכילות שיטות גישה לנזונים בעברית מפתחים בסביבת .NET Framework. (הסבירה בה אני מתכוון בפרויקט זה). השתמשתי בקבוצה זו במחלקה MyAdoHelper על מנת לגשת לנזונים מסד הנתונים שלי דרך הכלים המובנים בה.

ASP.NET – זהה מסגרת ליישומי אינטרנט בצד השירות המיועדת לפיתוח אתרים עם דפי אינטרנט דינמיים בשלוב נתונים מבוססי נתונים, וטיפול במידע שנשלח מהמשתמשים. תפקידו של ASP.NET הוא לנתח את המידע וליצור פلت בהתאם.

HTML – זהה שפת תגיות המיועדת לבניית תצוגה ועיצוב של דפי אינטרנט. זו שפת התגיוט הנפוצה ביותר בעולם, בשל היותה נוחה מאוד לשימוש וקללה ללימוד. החלק הקדמי של דפי ASP מתרגם ל-HTML, כך שלמעשה דפי האתר שלי מוצגים אצל הלוקוטCDFI HTML.

CSS – CSS הינו פורמט לעיצוב דפי אינטרנט. באמצעות CSS ניתן לקבוע את עיצובה של תגים ב-HTML, ובאופן כללי כמעט כל שפת תגיות לבניית אתרי אינטרנט. באתר שלי נעשה שימוש ב-CSS מקור חיצוני.

JavaScript – זהה שפת תכנות דינמית מונחית-עצמים ששימושה המרכזי הוא להרחב את יכולות שפת התגיות הבסיסית HTML ולאפשר יצירה של דפי אינטרנט דינמיים ומתחכמים יותר. קוד ה-JS משולב בדף HTML מבוצע על ידי הדפדפן של הלוקוט. בפרויקט שלי שילבתי JS בחנות הטיפוס, על מנת להציג את המחיר של מוצר באופן דינמי בהתאם לגודל ולכמות שנבחרו.

C# – זהה שפת תכנות כללנית, מודרנית, ובסיסת על תכונות מונחה-עצמים, בדומה ל-JavaScript. היתרונות בשימוש בשפה זו רבים, וביניהם העבודה שזו שפה קלה מאוד ללמידה ולתכנות, ומכוון שהקהילה משתמשת בה גדולת כל כך, שפותחים לשפה כמיות אדרות של כלים והרחבות. בפרויקט שלי צד-השירות תוכנת כולל ב-C#.

רפלקציה

העבדה על הפרויקט הייתה עבורי מתוגרת אך גם מעניינת ומספקת לאורך כל הדרך. הנהיתי מאד לבנות את האתר שלי ולפתח אותו לפי רצוני, ולראות איך הוא מפותח ומתקדם ככל שאני ממשיר לעובד עליו. השקעתו באתר שלי אין-ספור שעות, ואני יכול עתה להגיד שאני מאוד מרוצה מההתוצר הסופי.

מהעבדה קיבלתי כלים מגוונים ושימושיים. למדתי איך לבצע מחקר לגבי בעיה שאני נתקל בה באופן יעיל ונכון, כך שאוכל למצוא את מקור הבעיה ופתרון שלה באינטרנט במהירות ובקלות. למדתי איך לתכנן פרויקט, לראות את מה שאני רוצה לעשות ולהחלק אותו למשימות קטנות יותר שתתחברו בסוף למרכיב חדש.

במהלך בניית הפרויקט התמודדתי עם אתגרים רבים ומשונים – מאות באגים, שלפעמים לקחו דקה לתקן ולפעמים כמה שעות. במהלך העבודה נהיתי יותר ויותר מהיר ויעיל בתיקון הבעיה הקטנות האלה, ואני חשב שככל הטעויות שגרמו לבאגים שיפור וחידדו את יכולות התכונות שלי באופן משמעותי. לבסוף, נהניתי צריך לחשב על המבנה של האתר ואיך כל מרכיב יתחבר עם האחרים ולתכן את האתר בקדניות, כך שההתוצר הסופי שיתקבל יהיה מסודר ונוח עבור המשתמש והמנהל. תהליך זהלקח ממני לעיתים יותר זמן מאשרי, אך אני מרוצה שעשיתי אותו והתוצר הסופי לדעתו מבטא את זה.

המסקנות שלי הן שעבדה צזו יש לקחת ברצינות ולוודוד אליה קשה, אך גם להינות מהתהליך וללמוד ממנו. פרויקט באורך כזה הוא דבר שהולך ומפתחת ומשתנה לאורך הזמן, ולדעתי זה דבר טוב, מכיוון שהשינוי התמידי מאפשר לתלמיד למדוד תחומים רבים ומגוונים ולצבור בהם ניסיון חשוב מאוד שאין לזלزل בערכו שלו.

לו הייתי מתחילה היום, הייתי משקיע יותר בתכנון הראשוני של הפרויקט וגם מציב לעצמי מסגרת שלמה ומפורטת של האתר. לאחר מכן, הייתי מפתח את האתר תוך המסגרת הזאת אך גם מרשה לעצמי לצאת מהמסגרת ולהוציא עד תתי-מסגרות שהוں עשוות את האתר יותר מעניין. למרות זאת, אני עדין מרוצה מהתכןון שעשיתי ומההתוצר הסופי.

שאלת חקר עצמית שהייתי שואל את עצמי היא מה אני חשב לגבי הנושא, השפונות והטכנולוגיות שלמדנו. לפידעת, למדנו דברים מאוד מעניינים וגם מתוגרים, עליהם מתבسطים הרבה אתרים אינטרנט ובעזרתם נוכל להמשיך ולהעמיק בהמשך בנושא פיתוח אתרים ופרויקטים בכלל. למרות זאת, לפידעת, הינו יכולים למדוד נושאים יותר עדכניים וROLוונטיים לשוק הנוכחי, למשל פיתוח אפליקציות, רובוטיקה, או בניית האתר תוך שימוש בטכנולוגיות יותר עדכניות כגון React, Angular, ועוד.

ביבליוגרפיה

מקורות מרכזים בהם השתמשתי לפיתוח הפרויקט

קרייפוקט, ו' (2019-2022). מציגות בנושאי הנדסת תוכנה, Moodle.

אתר StackOverflow - <https://stackoverflow.com> - StackOverflow

אתר MSDN Forums - <https://social.msdn.microsoft.com/Forums/en-US/home> - MSDN Forums

אתר YouTube - <https://www.youtube.com> - YouTube

מדריך למפתח

HomePage.aspx

מיקום: ComputersProjectNew/HomePage.aspx

מיקום קוד מאוחר: ComputersProjectNew/HomePage.aspx.cs

קוד: aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="HomePage.aspx.cs" Inherits="HomePage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>Home Page</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2">
        <div class="inner">
            <div class="box">
                <div class="content">
                    <header class="align-center">
                        <h2>About The Site</h2>
                    </header>
                    <p>This site is all about climbing. Climbing is an
extremely popular sport and has gotten even more popular lately. Here you can
learn all about it.</p>
                    
                    
                    
                    <p>In my website you can read about different climbing
styles, lots of the current and former professional climbers, and climbing
spots all over the world.</p>
                    <p>Climbing can be done as both a hobby and a career
and has lots of different levels. </p>
                    <p>My goal in this website is to let you learn more
about climbing and all its different aspects, and maybe get you into it if you
don't climb yet.</p>
                </div>
            </div>
        </div>
    </section>
</asp:Content>
```

:C# טיפ

```
using System;
using System.Data;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class HomePage : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
}
```

Gallery.aspx

מיקום: ComputersProjectNew/Gallery.aspx

מיקום קוד מאחור: ComputersProjectNew/Gallery.aspx.cs

קוד aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Gallery.aspx.cs" Inherits="Gallery" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>Gallery</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section id="three" class="wrapper style2">
        <div class="inner">
            <header class="align-center">
                <h2>Gallery</h2>
            </header>
            <div class="gallery">
                <div class="box">
                    <div class="image fit">
                        <a></a>
                    </div>
                <div class="content">
                    <header class="align-center">
                        <h2>Adam Ondra</h2>
                        <h4>A Czech professional rock climber,
specializing in lead climbing and bouldering, Adam Ondra is one of the most
iconic rock climbers of all time. He is renowned for sending the worlds first
and only 9c, which he named Silence.</h4>
                        <br />
                        <br />
                    </header>
                </div>
                <footer class="align-center">
                    <ul class="actions">
                        <li><a class="button icon fa-instagram"
href="https://www.instagram.com/adam.ondra/" target="_blank">Instagram</a></li>
                        <li><a class="button icon fa-youtube"
href="https://www.youtube.com/c/AdamOndra" target="_blank">YouTube</a></li>
                        <li><a class="button icon fa-facebook"
href="https://www.facebook.com/adamondraofficial"
target="_blank">Facebook</a></li>
                    </ul>
                </footer>
            </div>
            <div class="box">
                <div class="image fit">
                    <a></a>
                </div>
            <div class="content">
                <header class="align-center">
                    <h2>Alex Megos</h2>
                    <h4>Alexander Megos is a German rock
climber. He was the first climber to on-sight (climb without prior practice or

```

advice) a route graded 9a (5.14d). He has climbed multiple notable hard routes and boulder problems, including three 9b+ (5.15c) routes (Perfecto Mundo, Bibliographie and King Capella)

```

        </header>
        </div>
        <footer class="align-center">
            <ul class="actions">
                <li><a class="button icon fa-instagram"
href="https://www.instagram.com/alexandermegos/"
target="_blank">Instagram</a></li>
                <li><a class="button icon fa-youtube"
href="https://www.youtube.com/channel/UCL3K1-rgMHFdGvGFU2Z_Enw"
target="_blank">YouTube</a></li>
                <li><a class="button icon fa-facebook"
href="https://www.facebook.com/AlexanderMegosSportler/"
target="_blank">Facebook</a></li>
            </ul>
        </footer>
        </div>
        <div class="box">
            <div class="image fit">
                <a></a>
            </div>
        </div>
        <div class="content">
            <header class="align-center">
                <h2>Alex Khazanov</h2>
                <h4>Alex Khazanov is a professional climber
in the bouldering climbing industry, and won a gold medal in bouldering
climbing in the 2018 World Cup round and two bronze medals in bouldering at the
European Youth Championships in 2013 and 2014. As of January 2022 he is ranked
53rd in the world in bouldering, and is a member of the Israel national team in
climbing.</h4>
                <br />
                <br />
            </header>
        </div>
        <footer class="align-center">
            <ul class="actions">
                <li><a class="button icon fa-instagram"
href="https://www.instagram.com/alexkhazanov/"
target="_blank">Instagram</a></li>
                <li><a class="button icon fa-youtube"
href="https://www.youtube.com/user/alexkhazanov"
target="_blank">YouTube</a></li>
                <li><a class="button icon fa-facebook"
href="https://www.facebook.com/alexkhazanovclimber/"
target="_blank">Facebook</a></li>
            </ul>
        </footer>
        </div>
        <div class="box">
            <div class="image fit">
                <a></a>
            </div>
        </div>
        <div class="content">
            <header class="align-center">
                <h2>Chris Sharma</h2>
                <h4>Chris Omprakash Sharma is an American
rock climber and considered one of the greatest in the history of the sport. In
2001, Sharma became the world's first climber to redpoint a consensus 9a+
            </header>
        </div>
    
```

(5.15a) route, Realization/Biographie, ushering in what was called a "technical revolution" in sport climbing. In 2008, Sharma redpointed the first 9b (5.15b) route, Jumbo love, and at 2013 became only the second to climb a 9b+ (5.15c) route La Dura Dura.</h4>

```

        </header>
    </div>
    <footer class="align-center">
        <ul class="actions">
            <li><a class="button icon fa-instagram"
href="https://www.instagram.com/chris_sharma/"
target="_blank">Instagram</a></li>
            <li><a class="button icon fa-youtube"
href="https://www.youtube.com/c/SharmaChannel" target="_blank">YouTube</a></li>
            <li><a class="button icon fa-facebook"
href="https://www.facebook.com/sharmaclimbing"
target="_blank">Facebook</a></li>
        </ul>
    </footer>
    </div>
</div>
</section>
</asp:Content>

```

:C# טיפ

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Gallery : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
}

```

Registration.aspx

מיקום: ComputersProjectNew/Registration.aspx

מיקום קוד מאחור: ComputersProjectNew/Registration.aspx.cs

קוד aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Registration.aspx.cs" Inherits="Registration"
%>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <%
        if(Session["permLvl"] == null)
        {
            Response.Write("<title>Registration Page</title>");
        }
        else
        {
            Response.Write("<title>User Creation Page</title>");
        }
    %>
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2">
        <div class="inner">
            <div class="box">
                <div class="content">
                    <header class="align-center">
                        <h2>Registration Form:</h2>
                    </header>
                    <form id="form1" runat="server" method="post">
                        <div class="row uniform">
                            <div class="6u 12u$(small)">
                                <h3>Username:</h3>
                            </div>
                            <div class="6u$ 12u$(small)">
                                <asp:TextBox id="UsernameTextBox"
runat="server"></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="UsernameTextBox" Display="None" ID="RequiredFieldValidator1"
runat="server" ErrorMessage="Username is
required"></asp:RequiredFieldValidator>
                                <asp:RegularExpressionValidator
ValidationExpression="^\w{5,50}$" ControlToValidate="UsernameTextBox"
Display="None" ID="RegularExpressionValidator1" runat="server"
ErrorMessage="Username must be between 5 and 50 characters
long"></asp:RegularExpressionValidator>
                            </div>
                        </div>
                        <div class="6u 12u$(small)">
                            <h3>Password:</h3>
                        </div>
                        <div class="6u$ 12u$(small)">
                            <asp:TextBox id="PasswordTextBox"
runat="server" TextMode="Password"></asp:TextBox>
                        </div>
                    </form>
                </div>
            </div>
        </section>
    </div>
</asp:Content>

```

```

        <asp:RequiredFieldValidator
ControlToValidate="PasswordTextBox" Display="None" ID="RequiredFieldValidator2"
runat="server" ErrorMessage="Password is
required"></asp:RequiredFieldValidator>
        <asp:RegularExpressionValidator
ValidationExpression="^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{5,50}$"
ControlToValidate="PasswordTextBox" Display="None"
ID="RegularExpressionValidator2" runat="server" ErrorMessage="Password must be
between 5 and 50 characters long, and contain numbers and uppercase and
lowercase letters"></asp:RegularExpressionValidator>
    </div>

        <div class="6u 12u$(small)">
            <h3>City:</h3>
        </div>
        <div class="6u$ 12u$(small)">
            <div class="select-wrapper">
                <asp:DropDownList ID="CityDDL"
runat="server">
                    <asp:ListItem Value="Haifa">Haifa</asp:ListItem>
                    <asp:ListItem Value="Tel Aviv">Tel
Aviv</asp:ListItem>
                    <asp:ListItem Value="Jerusalem">Jerusalem</asp:ListItem>
                    <asp:ListItem Value="Ashdod">Ashdod</asp:ListItem>
                    <asp:ListItem Value="Rishon
LeZiyon">Rishon LeZiyon</asp:ListItem>
                    <asp:ListItem Value="Petah
Tikva">Petah Tikva</asp:ListItem>
                    <asp:ListItem Value="Beer
Sheva">Beer Sheva</asp:ListItem>
                    <asp:ListItem Value="Netanya">Netanya</asp:ListItem>
                    <asp:ListItem Value="Holon">Holon</asp:ListItem>
                    <asp:ListItem Value="Bnei
Brak">Bnei Brak</asp:ListItem>
                    <asp:ListItem Value="Rehovot">Rehovot</asp:ListItem>
                    <asp:ListItem Value="Bat
Yam">Bat
Yam</asp:ListItem>
                </asp:DropDownList>
            </div>
            <asp:RequiredFieldValidator
ControlToValidate="CityDDL" Display="None" ID="RequiredFieldValidator3"
runat="server" ErrorMessage="City is required"></asp:RequiredFieldValidator>
        </div>

        <div class="6u 12u$(small)">
            <h3>Address:</h3>
        </div>
        <div class="6u$ 12u$(small)">
            <asp:TextBox id="AddressTextBox"
runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator
ControlToValidate="AddressTextBox" Display="None" ID="RequiredFieldValidator4"
runat="server" ErrorMessage="Address is required"></asp:RequiredFieldValidator>
            <asp:RegularExpressionValidator
ValidationExpression="^([A-Z][a-z]*\s)*[1-9][0-9]*$"
ControlToValidate="AddressTextBox" Display="None">

```

```
ID="RegularExpressionValidator3" runat="server" ErrorMessage="Wrong address
syntax"></asp:RegularExpressionValidator>
</div>

<div class="6u 12u$(small)">
    <h3>Gender:</h3>
</div>
<div class="6u$ 12u$(small)">
    <asp:RadioButtonList ID="GenderRBL"
runat="server">
        <asp:ListItem
Value="male">Male</asp:ListItem>
        <asp:ListItem
Value="female">Female</asp:ListItem>
    </asp:RadioButtonList>
    <asp:RequiredFieldValidator
ControlToValidate="GenderRBL" Display="None" ID="RequiredFieldValidator9"
runat="server" ErrorMessage="Gender is required"></asp:RequiredFieldValidator>
</div>

<div class="6u 12u$(small)">
    <h3>Birthdate:</h3>
</div>
<div class="6u$ 12u$(small)">
    <asp:TextBox id="BirthdateTextBox"
runat="server" TextMode="Date"></asp:TextBox>
    <asp:RequiredFieldValidator
ControlToValidate="BirthdateTextBox" Display="None"
ID="RequiredFieldValidator8" runat="server" ErrorMessage="Birthdate is
required"></asp:RequiredFieldValidator>
    <asp:RangeValidator
id="BirthDateRangeValidator" runat="server" Type="Date"
ControlToValidate="BirthdateTextBox" ErrorMessage="Invalid age, out of range"
Display="None"/>
</div>

<div class="6u 12u$(small)">
    <h3>Email:</h3>
</div>
<div class="6u$ 12u$(small)">
    <asp:TextBox id="EmailTextBox"
runat="server"></asp:TextBox>
    <asp:RequiredFieldValidator
ControlToValidate="EmailTextBox" Display="None" ID="RequiredFieldValidator5"
runat="server" ErrorMessage="Email is required"></asp:RequiredFieldValidator>
    <asp:RegularExpressionValidator
ValidationExpression="^[\w-\.]+@[([\w-]+\.)+[\w-]{2,4}$"
ControlToValidate="EmailTextBox" Display="None"
ID="RegularExpressionValidator4" runat="server" ErrorMessage="Wrong email
syntax"></asp:RegularExpressionValidator>
</div>

<div class="6u 12u$(small)">
    <h3>Best Boulderding
Grade (V-Scale):</h3>
</div>
<div class="6u$ 12u$(small)">
    <asp:TextBox id="BoulderGradeTextBox"
runat="server"></asp:TextBox>
    <asp:RequiredFieldValidator
ControlToValidate="BoulderGradeTextBox" Display="None"
```

```
ID="RequiredFieldValidator6" runat="server" ErrorMessage="Boulder grade is
required">></asp:RequiredFieldValidator>
    <asp:RegularExpressionValidator
ValidationExpression="^V([0-9]|1[0-7])$"
ControlToValidate="BoulderGradeTextBox" Display="None"
ID="RegularExpressionValidator5" runat="server" ErrorMessage="Wrong boulder
grade syntax. Must be V0 to V17"></asp:RegularExpressionValidator>
</div>

    <div class="6u 12u$(small)">
        <h3>Best Sport
Climbing Grade (French, capital letters):</h3>
    </div>
    <div class="6u$ 12u$(small)">
        <asp:TextBox id="WallGradeTextBox"
runat="server"></asp:TextBox>
        <asp:RequiredFieldValidator
ControlToValidate="WallGradeTextBox" Display="None"
ID="RequiredFieldValidator7" runat="server" ErrorMessage="Wall grade is
required">></asp:RequiredFieldValidator>
        <asp:RegularExpressionValidator
ValidationExpression="^[5-9][A-C][+]{0,1}$"
ControlToValidate="WallGradeTextBox" Display="None"
ID="RegularExpressionValidator6" runat="server" ErrorMessage="Wrong wall grade
syntax. Must be 5A to 9C (+ is optional)"></asp:RegularExpressionValidator>
    </div>
    <div class="12u$"></div>

    <div class="12u$"><asp:ValidationSummary
runat="server" ShowMessageBox="true" ShowSummary="false" /></div>

    <div class="12u$">
        <div class="align-center">
            <asp:Button runat="server"
ID="SubmitDataButton" OnClick="Submit_Click"/>
        </div>
    </div>

    <div class="12u$"><div class="align-center"><h3
class="red"><%= registerMsg %></h3></div></div>
    </div>
    </form>
</div>
</div>
</section>
</asp:Content>
```

.C# קוד

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Registration : System.Web.UI.Page
{
    //public string rvMin = DateTime.Today.AddYears(-
    120).ToString("yyyy/MM/dd"),
    //rvMax = DateTime.Today.AddYears(-8).ToString("yyyy/MM/dd");
    public string registerMsg; //הරשמה הודיעת את שומר
    public string buttonMsg; //הארשמה בכפתור הטקסט את שומר
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["username"] == null)
        {
            buttonMsg = "Sign Up";
        }
        else
        {
            buttonMsg = "Create User";
        }
        SubmitDataButton.Text = buttonMsg;

        BirthDateRangeValidator.MinimumValue = DateTime.Today.AddYears(-
        120).ToString("yyyy/MM/dd");
        BirthDateRangeValidator.MaximumValue = DateTime.Today.AddYears(-
        8).ToString("yyyy/MM/dd");
    }
    protected void Submit_Click(object sender, EventArgs e)
    {

        string username = Request.Form[UsernameTextBox.UniqueID]; //ערך את שומר מהטופס השלוף המשתמש שם
        string password = Request.Form[PasswordTextBox.UniqueID]; //ערך את שומר מהטופס השלוף הסיסמה
        string city = Request.Form[CityDDL.UniqueID]; //השלוף העיר ערך את שומר מהטופס מהטופס
        string gender = Request.Form[GenderRBL.UniqueID]; //השלוף המין את שומר מהטופס מהטופס
        string birthdate = Request.Form[BirthdateTextBox.UniqueID]; //את שומר מהטופס הילדה תאריך
        string boulderGrade = Request.Form[BoulderGradeTextBox.UniqueID]; //שומר משתמש של הבoulder רמת את
        string wallGrade = Request.Form[WallGradeTextBox.UniqueID]; //את שומר המשתמש של הקיר רמת
        string email = Request.Form[EmailTextBox.UniqueID]; //טלפון מסטר את שומר מהטופס השלוף מהטופס
        string address = Request.Form[AddressTextBox.UniqueID]; //הכתובת את שומר מהטופס מהטופס

        int usernameTaken = MyAdoHelper.UserExists(username);

        if (usernameTaken == 1)
        {
            //תפוס כן המשתמש שם אם

```

```
registerMsg = "Sorry! The username you entered is taken"; //כתובן  
לטבלה ערבים מוכנס ולא תפוס המשמש שם הودעה  
}  
if (usernameTaken == 2)  
{  
    //נזהרים מסד עם הטרנווקציה בעת בעיה חלה אם  
    registerMsg = "Oops! There appears to have been an error. Please  
    כך אחר שוב ולנסות הטרנווקציה בעת תקללה שהלה הודעה כותב//  
}  
if (usernameTaken == 0)  
{  
    bool isUserEntered = MyAdoHelper.CreateNewUser(username, password,  
city, gender, birthdate, boulderGrade, wallGrade, email, address);  
  
    if (isUserEntered)  
    {  
        if (Session["username"] != null)  
        {  
            Response.Redirect("AdminPage.aspx");  
        }  
        else  
        {  
            Response.Redirect("LogIn.aspx");  
        }  
    }  
    else  
    {  
        registerMsg = "Oops! There appears to have been an error.  
        כך אחר שוב ולנסות הטרנווקציה בעת תקללה שהלה הודעה כותב//  
    }  
}
```

LogIn.aspx

מיקום: ComputersProjectNew/LogIn.aspx

מיקום קוד מאחור: ComputersProjectNew/LogIn.aspx.cs

קוד aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="LogIn.aspx.cs" Inherits="LogIn" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>Log In Page</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2">
        <div class="inner">
            <div class="box">
                <div class="content">
                    <header class="align-center">
                        <h2>Log In Form:</h2>
                    </header>
                    <form id="form1" runat="server" method="post">
                        <div class="row uniform">
                            <div class="6u 12u$(small)">
                                <h3>Username:</h3>
                                </div>
                            <div class="6u$ 12u$(small)">
                                <asp:TextBox id="UsernameTextBox"
runat="server"></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="UsernameTextBox" Display="None" ID="RequiredFieldValidator1"
runat="server" ErrorMessage="Username is
required"></asp:RequiredFieldValidator>
                                <asp:RegularExpressionValidator
ValidationExpression="^\w{5,50}$" ControlToValidate="UsernameTextBox"
Display="None" ID="RegularExpressionValidator1" runat="server"
ErrorMessage="Username must be between 5 and 50 characters
long"></asp:RegularExpressionValidator>
                            </div>
                            <div class="6u 12u$(small)">
                                <h3>Password:</h3>
                                </div>
                            <div class="6u$ 12u$(small)">
                                <asp:TextBox id="PasswordTextBox"
runat="server"></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="PasswordTextBox" Display="None" ID="RequiredFieldValidator2"
runat="server" ErrorMessage="Password is
required"></asp:RequiredFieldValidator>
                                <asp:RegularExpressionValidator
ValidationExpression="^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{5,50}$"
ControlToValidate="PasswordTextBox" Display="None"
ID="RegularExpressionValidator2" runat="server" ErrorMessage="Password must be
between 5 and 50 characters long, and contain numbers and uppercase and
lowercase letters"></asp:RegularExpressionValidator>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </section>
    </div>
</asp:Content>
```

```

        <div class="12u$">
            <asp:CheckBox runat="server"
ID="RememberMeBox" Text="Remember me?"
OnCheckedChanged="RememberMeBox_CheckedChanged"/>
        </div>

        <div class="12u$"></div>

        <div class="12u$"><div class="align-
center"><asp:Button runat="server" ID="submitData" OnClick="Submit_Click"
Text="Log In"/></div></div>

        <div class="12u$"><div class="align-center"><h3
class="red"><%= logInMsg %></h3></div></div>
        </form>
        </div>
    </div>
</section>
</asp:Content>

```

C# קוד:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class LogIn : System.Web.UI.Page
{
    public string logInMsg, //הזהירות מתאימה הودעה שומר
        usernameValue;
    public bool isChecked = false;
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Request.Cookies["userinfo"] != null)
        {
            HttpCookie info = Request.Cookies["userinfo"];
            usernameValue = info["username"];
            UsernameTextBox.Text = usernameValue;
            PasswordTextBox.Text = info["password"];
            RememberMeBox.Checked = true;
            isChecked = true;
        }
        PasswordTextBox.Attributes["type"] = "password";
    }

    protected void Submit_Click(object sender, EventArgs e)
    {
        string username = Request.Form[UsernameTextBox.UniqueID]; //את שולף
        מהテופס המשמש שם
        string password = Request.Form[PasswordTextBox.UniqueID]; //את שולף
        מהטופס הסיסמה
        bool rememberMe = isChecked; //הערך את שולף את
        int userExists = MyAdoHelper.UserExists(username);
    }
}

```

```

if (userExists == 0) // אם לא אונט
{
    לא און
    logInMsg = "Sorry! The username you entered does not exist";
    בהתאם הודעה מדפס קיים לא המשתמש שם אם //
}
if (userExists == 2)
{
    נתונים מסד עם הטרנווקציה בעיה חלה אם //
    logInMsg = "Oops! There appears to have been an error. Please
    כך אחר שוב ולנסות הטרנווקציה בעיה תקללה הודעה כותב //";
}

if (userExists == 1)
{
    int passwordMatch =
MyAdoHelper.UserWithPasswordExists(username, password);

    אם (passwordMatch == 0) // אם הסיסמה לא מתחילה
    {
        לא און
        logInMsg = "Sorry! The password you entered is incorrect";
        בהתאם הודעה מדפס המשתמש לשם מתחילה לא הסיסמה אם //
    }
    if (passwordMatch == 2)
    {
        נתונים מסד עם הטרנווקציה בעיה חלה אם //
        logInMsg = "Oops! There appears to have been an error.
        Please try again later.";// כרך אחר שוב ולנסות הטרנווקציה בעיה תקללה הודעה כותב //
    }

    if (passwordMatch == 1)
    {
        מתחילה הסיסמה אם //

        int userPermLvl = MyAdoHelper.GetUserPermLvl(username);

        אם (userPermLvl == -1)
        {
            נתונים מסד עם הטרנווקציה בעיה חלה אם //
            logInMsg = "Oops! There appears to have been an error.
            Please try again later.";// כרך אחר שוב ולנסות הטרנווקציה בעיה תקללה הודעה כותב //
        }
        else
        {
            Session["username"] = username; // יוצר סשן
            Session["userPermLvl"] = userPermLvl; // להרשאת סשן יוצר
            המשתמש לשם סשן יוצר
            להרשאת סשן יוצר המשתמש לשם סשן יוצר
            הגישה הרשות את בו ושומר המשתמש של הכניסה
        }
    }
}

Application.Lock();
if (Application["userNum"] == null)
    Application["userNum"] = 0;
Application["userNum"] = (int)Application["userNum"] +
1;
Application.UnLock();

if (!rememberMe)
{
    אם (username.Equals(usernameValue))
    {
        Response.Cookies["userinfo"].Value = null;
    }
}

```

```
Response.Cookies["userinfo"].Expires =
DateTime.Now.AddYears(-1);
}
}
else
{
    HttpCookie rememberUser = new
HttpCookie("userinfo");
rememberUser.Expires = DateTime.Now.AddMonths(6);
rememberUser["username"] = username;
rememberUser["password"] = password;

Response.Cookies.Add(rememberUser);
}
Response.Redirect("HomePage.aspx"); //דף המשתמש את מוחזיר

הבית
}
}
}
}

protected void RememberMeBox_CheckedChanged(object sender, EventArgs e)
{
    isChecked = !isChecked;
}
}
```

ClimbingShop.aspx

מיקום: ComputersProjectNew/ClimbingShop.aspx

מיקום קוד מאחור: ComputersProjectNew/ClimbingShop.aspx.cs

קוד aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="ClimbingShop.aspx.cs" Inherits="ClimbingShop"
%>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>Climbing Shop</title>
    <script type="text/javascript">
        function SizeChanged(obj) {
            var objId = obj.id;
            var elementNum = objId.substr(objId.length - 1);
            var amountTxtObj =
document.getElementById('ContentPlaceHolder1_ShopDataList_AmountTxtEdit_' + elementNum);
            var amountTxtValue = amountTxtObj.value;
            var priceTxtObj =
document.getElementById('ContentPlaceHolder1_ShopDataList_PriceTxtEdit_' + elementNum);
            var selectedOption = obj.options[obj.selectedIndex];
            var price = selectedOption.attributes.sizeprice.value;
            priceTxtObj.value = price * amountTxtValue;
        }
        function AmountChanged(obj) {
            var objId = obj.id;
            var objValue = obj.value;
            var elementNum = objId.substr(objId.length - 1);
            var sizeDdlObj =
document.getElementById('ContentPlaceHolder1_ShopDataList_SizeDDL_' + elementNum);
            var selectedOption = sizeDdlObj.options[sizeDdlObj.selectedIndex];
            var price = selectedOption.attributes.sizeprice.value;
            var priceTxtObj =
document.getElementById('ContentPlaceHolder1_ShopDataList_PriceTxtEdit_' + elementNum);
            priceTxtObj.value = price * objValue;
        }
    </script>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2">
        <div class="inner">
            <div class="box">
                <div class="content">
                    <header class="align-center">
                        <h2>Our Climbing Shop</h2>
                    </header>
                    <div>
                        <form id="form1" runat="server">
                            <div class="align-center">
                                <asp:Button ID="ShoppingCartButton"
runat="server" OnClick="ShoppingCartButton_Click" Text="View Shopping Cart" />
                                <br />
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </section>
</asp:Content>
```

```

        <br />
        <asp:Label ID="ErrorLbl" runat="server"
Visible="false"></asp:Label>
    </div>
    <asp:DataList ID="ShopDataList" runat="server"
RepeatDirection="Horizontal" RepeatLayout="Table" RepeatColumns="3"
DataKeyField="productId" OnEditCommand="Edit_Command"
OnCancelCommand="Cancel_Command" OnUpdateCommand="Buy_Command"
OnItemDataBound="ShopDataList_ItemDataBound">
        <ItemStyle BackColor="LightYellow"
BorderWidth="5px" BorderColor="White"/>
        <AlternatingItemStyle BackColor="LightBlue">
    />
        <ItemTemplate>
            <asp:TextBox ID="ProductIdTxt"
Visible="false" runat="server" ReadOnly="true" Text='<%# Eval("productId") %>'></asp:TextBox>
            <asp:Image ID="ImageBox" runat="server"
ImageUrl='<%# "~/productImages/" + Eval("imageName") %>' Height="250"
Width="250"/>
            <br />
            <h3>Product Name:&nbsp</h3>
            <asp:TextBox ID="ProductNameTxt"
runat="server" Text='<%# Eval("name") %>' ReadOnly="true"></asp:TextBox>
            <br />
            <h3>Product Company:&nbsp</h3>
            <asp:TextBox ID="CompanyTxt"
runat="server" Text='<%# Eval("company") %>' ReadOnly="true"></asp:TextBox>
            <br />
            <h3>Size:&nbsp</h3>
            <asp:TextBox ID="SizeTxt"
runat="server" Text='Select A Size' ReadOnly="true"></asp:TextBox>
            <br />
            <h3>Amount:&nbsp</h3>
            <asp:TextBox ID="AmountTxt"
runat="server" Text='Select An Amount' ReadOnly="true"></asp:TextBox>
            <br />
            <h3>Price:&nbsp</h3>
            <asp:TextBox ID="PriceTxt"
runat="server" Text="-" ReadOnly="true"></asp:TextBox>
            <br />
            <div class="align-center">
                <asp:Button ID="EditButton"
runat="server" CommandName="Edit" Text="Edit" CssClass="button shopbutton"
CausesValidation="false" />
            </div>
        </ItemTemplate>
        <EditItemTemplate>
            <asp:TextBox ID="ProductIdTxt"
Visible="false" runat="server" ReadOnly="true" Text='<%# Eval("productId") %>'></asp:TextBox>
            <asp:Image ID="ImageBox" runat="server"
ImageUrl='<%# "~/productImages/" + Eval("imageName") %>' Height="250"
Width="250"/>
            <br />
            <h3>Product Name:&nbsp</h3>
            <asp:TextBox ID="ProductNameTxtEdit"
runat="server" Text='<%# Eval("name") %>' ReadOnly="true"></asp:TextBox>
            <br />
            <h3>Product Company:&nbsp</h3>
            <asp:TextBox ID="CompanyTxtEdit"
runat="server" Text='<%# Eval("company") %>' ReadOnly="true"></asp:TextBox>

```

```

        <br />
        <h3>Size:&nbsp;</h3>
        <asp:DropDownList ID="SizeDDL"
runat="server"></asp:DropDownList>
        <asp:RequiredFieldValidator
ID="SizeDDLRequiredFieldValidator" runat="server" ControlToValidate="SizeDDL"
ErrorMessage="Please select a size!"
Display="None"></asp:RequiredFieldValidator>
        <asp:RangeValidator
ID="SizeDDLRageValidator" runat="server" ControlToValidate="SizeDDL"
Type="Integer" MinimumValue="30" MaximumValue="55" ErrorMessage="Please select
a valid size!" Display="None"></asp:RangeValidator>
        <br />
        <h3>Amount:&nbsp;</h3>
        <asp:TextBox ID="AmountTxtEdit"
runat="server" Text='1'></asp:TextBox>
        <asp:RequiredFieldValidator
ControlToValidate="AmountTxtEdit" Display="None" ID="RequiredFieldValidator2"
runat="server" ErrorMessage="Amount is required"></asp:RequiredFieldValidator>
        <asp:RegularExpressionValidator
ValidationExpression="^([1-9]|10)$" ControlToValidate="AmountTxtEdit"
Display="None" ID="RegularExpressionValidator1" runat="server"
ErrorMessage="Invalid amount! Must be between 1 and
10"></asp:RegularExpressionValidator>
        <br />
        <h3>Price:&nbsp;</h3>
        <asp:TextBox ID="PriceTxtEdit"
runat="server" Text="-" ReadOnly="true"></asp:TextBox>
        <br />
        <div class="align-center">
            <asp:Button ID="BuyButton"
runat="server" CommandName="Update" Text="Add To Cart" CssClass="button
shopbutton" CausesValidation="true"/>
        </div>
        <br />
        <div class="align-center">
            <asp:Button ID="CancelButton"
runat="server" CommandName="Cancel" Text="Cancel" CssClass="button shopbutton"
CausesValidation="false"/>
        </div>
        <asp:ValidationSummary
ID="SummaryValidator" runat="server" ShowMessageBox="true" ShowSummary="false"
/>
        </EditItemTemplate>
    </asp:DataList>
</form>
</div>
</div>
</div>
</div>
</div>

</section>
</asp:Content>

```

#C קוד

```

using System;
using System.Data;
using System.Data.SqlClient;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;

public partial class ClimbingShop : System.Web.UI.Page
{
    public string priceStr = "";
    string[] priceArr;
    public string ddl = "";
    public DataTable productDt,
        supplyDt; // על המוחזרים הנתונים את שומר
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["username"] == null)// בדק אם המשתמש רשום
            Response.Redirect("NoAccessPage.aspx"); // כניסה האין לדף המשתמש את מעביר

        if(Session["cartStatus"] != null)
        {
            switch (Session["cartStatus"].ToString())
            {
                case "empty":
                {
                    ErrorLbl.Text = "<h3 class='red'>Your cart is
empty!</h3>";
                    ErrorLbl.Visible = true;
                    Session["cartStatus"] = null;
                    break;
                }
                case "bought":
                {
                    ErrorLbl.Text = "<h3 class='green'>Thank you for your
purchase! A receipt has been sent to your email!</h3>";
                    ErrorLbl.Visible = true;
                    Session["cartStatus"] = null;
                    break;
                }
                case "error":
                {
                    ErrorLbl.Text = "<h3 class='red'>There seems to have
been an error! Maybe the admins changed something</h3>";
                    ErrorLbl.Visible = true;
                    Session["cartStatus"] = null;
                    break;
                }
            }
        }

        if (!IsPostBack)
        {
            GetData();
        }
    }
}

```

```

public void GetData()
{
    productDt = MyAdoHelper.GetProductTbl();
    supplyDt = MyAdoHelper.GetSupplyTbl();

    if (productDt == null || supplyDt == null)
        Response.Redirect("NoAccessPage.aspx");
    ShopDataList.DataSource = productDt;
    ShopDataList.DataBind();
}

protected void GenerateSizeDDL(int productId)
{
    ddl = "";

    DataTable supplyTbl = MyAdoHelper.GetSupplyTbl().Select("productId=" +
productId).CopyToDataTable();

    foreach (DataRow dtRow in supplyTbl.Rows)
    {
        ddl += "<option value='" + dtRow["size"].ToString() + "'>" +
dtRow["size"].ToString() + "</option>";
    }
}

protected void ShoppingCartButton_Click(object sender, EventArgs e)
{
    if (MyAdoHelper.IsExist("MyDatabase.mdf", "SELECT * FROM tblOrder WHERE
username=' " + Session["username"].ToString() + "' AND paid='0'"))
    {
        Response.Redirect("ShoppingCart.aspx");
    }
    else
    {
        ErrorLbl.Text = "<h3 class='red'>Your cart is empty!</h3>";
        ErrorLbl.Visible = true;
    }
}

protected void Edit_Command(object sender, DataListCommandEventArgs e)
{
    ShopDataList.EditItemIndex = e.Item.ItemIndex;

    GetData();
}

protected void Cancel_Command(object source, DataListCommandEventArgs e)
{
    ShopDataList.EditItemIndex = -1;
    GetData();
}

protected void Buy_Command(object source, DataListCommandEventArgs e)
{
    int productId =
int.Parse(((TextBox)e.Item.FindControl("ProductIdTxt")).Text),
        amount =
int.Parse(((TextBox)e.Item.FindControl("AmountTxtEdit")).Text);
        int size =
int.Parse(((DropDownList)e.Item.FindControl("SizeDDL")).SelectedValue);
}

```

```

        if(size == -1)
    {
        Session["cartStatus"] = "error";
        Response.Redirect("ClimbingShop.aspx");
    }

    if(!MyAdoHelper.IsExist("MyDatabase.mdf", "SELECT * FROM tblSupply
WHERE productId=''" + productId + "' AND size=''" + size + "'"))
    {
        Session["cartStatus"] = "error";
        Response.Redirect("ClimbingShop.aspx");
    }

    if (MyAdoHelper.IsExist("MyDatabase.mdf", "SELECT * FROM tblOrder WHERE
username=''" + Session["username"].ToString() + "' AND paid='0'"))
    {
        DataTable orderTbl = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf",
"SELECT * FROM tblOrder WHERE username=''" + Session["username"].ToString() + "'"
AND paid='0'");
        int orderId = int.Parse(orderTbl.Rows[0][0].ToString());
        if (MyAdoHelper.IsExist("MyDatabase.mdf", "SELECT * FROM
tblOrderProduct WHERE orderId=''" + orderId + "' AND productId=''" + productId +
"' AND size=''" + size + "'"));
        {
            MyAdoHelper.ExecuteNonQuery("MyDatabase.mdf", "UPDATE
tblOrderProduct SET amount=amount+" + amount + " WHERE orderId=''" + orderId +
"' AND productId=''" + productId + "' AND size=''" + size + "'");
        }
        else
        {
            MyAdoHelper.ExecuteNonQuery("MyDatabase.mdf", "INSERT INTO
tblOrderProduct(orderId, productId, size, amount) VALUES (''" + orderId + "', '"
+ productId + "', '" + size + "', '" + amount + "')");
        }
    }
    else
    {
        DateTime myDateTime = DateTime.UtcNow;
        string sqlFormattedDate = myDateTime.ToString("yyyy-MM-dd
HH:mm:ss.fff");
        MyAdoHelper.ExecuteNonQuery("MyDatabase.mdf", "INSERT INTO tblOrder
(username, datetime, paid) VALUES (''" + Session["username"].ToString() + "', '"
+ sqlFormattedDate + "', '0')");
        DataTable orderTbl = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf",
"SELECT * FROM tblOrder WHERE username=''" + Session["username"].ToString() + "'"
AND paid='0'");
        int newOrderId = int.Parse(orderTbl.Rows[0][0].ToString());
        MyAdoHelper.ExecuteNonQuery("MyDatabase.mdf", "INSERT INTO
tblOrderProduct (orderId, productId, size, amount) VALUES (''" + newOrderId +
"', '" + productId + "', '" + size + "', '" + amount + "')");
    }
    ShopDataList.EditItemIndex = -1;
    GetData();
}

protected void ShopDataList_ItemDataBound(object sender,
DataListItemEventArgs e)
{
    if(e.Item.ItemType == ListItemType.Item || e.Item.ItemType ==
ListItemType.AlternatingItem)
    {
}

```

```

        int productId =
int.Parse(((TextBox)e.Item.FindControl("ProductIdTxt")).Text);
        Button editButton = (Button)e.Item.FindControl("EditButton");

        DataTable supplyTbl =
MyAdoHelper.GetSupplyTbl().Select("productId=" + productId).CopyToDataTable();

        if(supplyTbl.Select("amount>0").Count() < 1)
{
            editButton.Enabled = false;
            editButton.Text = "Out Of Stock";
}
        if (e.Item.ItemType == ListItemType.EditItem)
{
            int productId =
int.Parse(((TextBox)e.Item.FindControl("ProductIdTxt")).Text);
            DropDownList sizeDDL = (DropDownList)e.Item.FindControl("SizeDDL");
            TextBox amountTextBox =
(TextBox)e.Item.FindControl("AmountTxtEdit");

            amountTextBox.Attributes["onchange"] = "AmountChanged(this);";

            DataTable supplyTbl =
MyAdoHelper.GetSupplyTbl().Select("productId=" + productId).CopyToDataTable();

            ListItem item = new ListItem("Select A Size", "-1");
            item.Attributes["disabled"] = "disabled";
            sizeDDL.Items.Add(item);

            foreach (DataRow dataRow in supplyTbl.Rows)
{
                item = new ListItem(dataRow["size"].ToString());
                item.Attributes.Add("sizePrice", dataRow["price"].ToString());
                if(int.Parse(dataRow["amount"].ToString()) < 1)
{
                    item.Attributes["disabled"] = "disabled";
                    item.Text = dataRow["size"].ToString() + " - Out Of Stock";
                    item.Value = "-1";
}
                sizeDDL.Items.Add(item);
}
            sizeDDL.Attributes["onchange"] = "SizeChanged(this);";

            sizeDDL.SelectedIndex = 0;
}
}
}
    
```

ShoppingCart.aspx

מיקום: ComputersProjectNew/ShoppingCart.aspx

מיקום קוד מאחור: ComputersProjectNew/ShoppingCart.aspx.cs

קוד aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="ShoppingCart.aspx.cs" Inherits="ShoppingCart"
%>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>Shopping Cart</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2" id="three">
        <div class="inner">
            <header class="align-center">
                <h2>Shopping Cart</h2>
                <br />
                <asp:Label ID="TotalPriceLbl"
runat="server"></asp:Label>
                <br />
            </header>
        </div>
        <form id="CartForm" runat="server">
            <div class="align-center">
                <asp:Button ID="PaymentButton" runat="server" Text="Go To
Final Checkout" OnClick="PaymentButton_Click" />
                <asp:Button ID="CancelPaymentButton" runat="server"
Text="Go Back To Shopping Cart" OnClick="CancelPaymentButton_Click"
Visible="false" />
                <br />
                <br />
                <asp:Label ID="CartErrorLbl" runat="server"
Visible="false"></asp:Label>
            </div>
            <br />
            <br />
            <asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="False"
OnSelectedIndexChanged="GridView1_SelectedIndexChanged"
HorizontalAlign="Center" AllowPaging="True"
OnRowDeleting="GridView1_RowDeleting"
OnRowCancelingEdit="GridView1_RowCancelingEdit"
OnRowEditing="GridView1_RowEditing" OnRowUpdating="GridView1_RowUpdating"
OnPageIndexChanging="GridView1_PageIndexChanging">
                <PagerStyle HorizontalAlign="Center"/>
                <Columns>
                    <asp:CommandField ButtonType="Button"
ShowEditButton="true" HeaderText="Editing" CancelText="Cancel Edit"
EditText="Edit Order" UpdateText="Confirm Edit" ItemStyle-Width="300"/>
                    <asp:ButtonField ButtonType="Button" Text="Delete"
CommandName="Delete" HeaderText="Deletion"/>
                    <asp:BoundField DataField="productId" ReadOnly="true"
HeaderText="Product ID" ItemStyle-CssClass="hiddencol" HeaderStyle-
CssClass="hiddencol"/>
                </Columns>
            </asp:GridView>
        </form>
    </section>
</div>
```

```

        <asp:BoundField DataField="size" ReadOnly="true"
HeaderText="Size"/>
        <asp:TemplateField HeaderText="Amount">
            <EditItemTemplate>
                <asp:TextBox ID="AmountTxt" runat="server"
Text='<%# Eval("amount") %>'></asp:TextBox>
                <asp:RequiredFieldValidator
ControlToValidate="AmountTxt" Display="None" ID="RequiredFieldValidator1"
runat="server" ErrorMessage="Amount is required"></asp:RequiredFieldValidator>
                <asp:RegularExpressionValidator
ValidationExpression="^([1-9]|10)$" ControlToValidate="AmountTxt"
Display="None" ID="RegularExpressionValidator1" runat="server"
ErrorMessage="Invalid amount! Must be between 1 and
10"></asp:RegularExpressionValidator>
            </EditItemTemplate>
            <ItemTemplate>
                <asp:Label ID="AmountLbl" runat="server"
Text='<%# Bind("amount") %>'></asp:Label>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:BoundField DataField="name" ReadOnly="true"
HeaderText="Product Name"/>
        <asp:BoundField DataField="price" ReadOnly="true"
HeaderText="Price"/>
        <asp:BoundField DataField="stock" ReadOnly="true"
HeaderText="In Stock" />
    </Columns>
</asp:GridView>

<div class="inner" id="PaymentDiv" runat="server"
visible="false">
    <div class="box">
        <div class="content">
            <div class="row uniform">
                <div class="4u 12u$(small)">
                    <h3
id="SavedCardHeader" runat="server">Use A Saved Credit Card:</h3>
                    </div>
                <div class="4u 12u$(small)">
                    <asp:DropDownList ID="SavedCardDDL"
runat="server"
OnSelectedIndexChanged="SavedCardDDL_SelectedIndexChanged"></asp:DropDownList>
                    </div>
                <div class="4u 12u$(small)">
                    <asp:Button ID="SavedCardButton"
runat="server" Text="Use Card" OnClick="SavedCardButton_Click"/>
                    </div>
                <div class="4u 12u$(small)">
                    <h3>Card Number (No
Spaces):</h3>
                    </div>
                <div class="4u 12u$(small)">
                    <asp:TextBox id="CardNumTextBox"
runat="server"></asp:TextBox>
                    <asp:RequiredFieldValidator
ValidationGroup="PurchaseValidationGroup" ControlToValidate="CardNumTextBox"
Display="None" ID="RequiredFieldValidator1" runat="server" ErrorMessage="Card
number is required"></asp:RequiredFieldValidator>
                    <asp:RegularExpressionValidator
ValidationGroup="PurchaseValidationGroup" ValidationExpression="^[0-9]{16}$"
ControlToValidate="CardNumTextBox" Display="None">
                </div>
            </div>
        </div>
    </div>

```

```
ID="RegularExpressionValidator1" runat="server" ErrorMessage="Card number must  
be all digits and 16 characters long">></asp:RegularExpressionValidator>  
        </div>  
        <div class="4u 12u$(small)"></div>  
            <div class="4u 12u$(small)">  
                <h3>CVV:</h3>  
            </div>  
            <div class="4u 12u$(small)">  
                <asp:TextBox id="CVVTextBox"  
runat="server"></asp:TextBox>  
                    <asp:RequiredFieldValidator  
ValidationGroup="PurchaseValidationGroup" ControlToValidate="CVVTextBox"  
Display="None" ID="RequiredFieldValidator4" runat="server" ErrorMessage="CVV is  
required"></asp:RequiredFieldValidator>  
                        <asp:RegularExpressionValidator  
ValidationGroup="PurchaseValidationGroup" ValidationExpression="^[0-9]{3}$"  
ControlToValidate="CVVTextBox" Display="None" ID="RegularExpressionValidator3"  
runat="server" ErrorMessage="CVV must be 3 non-0  
digits"></asp:RegularExpressionValidator>  
                    </div>  
                    <div class="4u 12u$(small)"></div>  
                        <div class="4u 12u$(small)">  
                            <h3>Expiration Date:</h3>  
                        </div>  
                        <div class="4u 12u$(small)">  
                            <asp:TextBox id="ExpireDateTextBox"  
runat="server" TextMode="Date"></asp:TextBox>  
                                <asp:RequiredFieldValidator  
ValidationGroup="PurchaseValidationGroup" ControlToValidate="ExpireDateTextBox"  
Display="None" ID="RequiredFieldValidator8" runat="server"  
ErrorMessage="Expiration date is required"></asp:RequiredFieldValidator>  
                                <asp:RangeValidator  
ValidationGroup="PurchaseValidationGroup" id="ExpireDateRangeValidator"  
runat="server" Type="Date" ControlToValidate="ExpireDateTextBox"  
MinimumValue="" MaximumValue="" ErrorMessage="Invalid expiration date, out of  
range" Display="None"/>  
                            </div>  
                            <div class="4u 12u$(small)"></div>  
                                <div class="12u$">  
                                    <asp:CheckBox runat="server"  
ID="SaveCardBox" Text="Save card info?"/>  
                                </div>  
                                <div class="12u$"><asp:ValidationSummary  
runat="server" ShowMessageBox="true" ShowSummary="false"  
ValidationGroup="PurchaseValidationGroup"/></div>  
                                <asp:Label ID="Label1"  
runat="server"></asp:Label>  
                                <div class="12u$">  
                                    <div class="align-center">  
                                        <asp:Button runat="server"  
ID="FinalPurchaseButton" OnClick="FinalPurchaseButton_Click" Text="Purchase"  
ValidationGroup="PurchaseValidationGroup"/>  
                                    </div>  
                                </div>  
                                <div class="12u$">
```

```

                <div class="align-center">
                    <asp:Label ID="PurchaseErrorLbl"
runat="server" Visible="false"></asp:Label>
                </div>
            </div>
        </div>
    </div>
    <asp:ValidationSummary ID="SummaryValidator" runat="server"
ShowMessageBox="true" ShowSummary="false" />
</form>
</section>
</asp:Content>

```

:C# קוד

```

using System;
using System.Data;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Services;
using System.Text;
using System.Web.Script;
using System.Web.Script.Services;

public partial class ShoppingCart : System.Web.UI.Page
{
    public DataTable orderProductTbl; // שומר הנתונים את שומר
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["username"] == null)// בודק אם המשתמש רשום
            Response.Redirect("NoAccessPage.aspx"); // ננישת המשתמש אל מועדון האחנון ידי על המוחזרים

        if (!IsPostBack)
        {
            GetData();
        }
    }

    public void GetData()
    {
        List<string> outOfStockItemList = new List<string>();
        List<int> outOfStockSizeList = new List<int>();

        ExpireDateRangeValidator.MinimumValue =
DateTime.Today.AddMonths(1).ToString("yyyy/MM/dd");
        ExpireDateRangeValidator.MaximumValue =
DateTime.Today.AddYears(15).ToString("yyyy/MM/dd");

        ViewState["outOfStock"] = false;
        string currentUser = Session["username"].ToString();
        DataTable orderTbl = MyAdoHelper.GetOrderTblByUser(currentUser);
        DataTable supplyTbl = MyAdoHelper.GetSupplyTbl();
        DataTable productTbl = MyAdoHelper.GetProductTbl();
    }
}

```

```

        int orderId;

        if(orderTbl.Rows.Count >= 1)
        {
            orderId = int.Parse(orderTbl.Rows[0][0].ToString());
            orderProductTbl = MyAdoHelper.GetOrderProductTbl(orderId);
        }
        else
        {
            Session["cartStatus"] = "empty";
            Response.Redirect("ClimbingShop.aspx");
        }

        DataColumn productNameColumn = orderProductTbl.Columns.Add("name",
typeof(string));
        DataColumn priceColumn = orderProductTbl.Columns.Add("price",
typeof(float));
        DataColumn stockColumn = orderProductTbl.Columns.Add("stock",
typeof(int));

        float totalPrice = 0;

        foreach (DataRow dataRow in orderProductTbl.Rows)
        {
            int drProductId = int.Parse(dataRow["productId"].ToString());
            int drProductSize = int.Parse(dataRow["size"].ToString());
            int drAmount = int.Parse(dataRow["amount"].ToString());

            dataRow["name"] = productTbl.Rows[drProductId - 1][1].ToString();

            DataRow[] supplyDrArr = supplyTbl.Select("productId=" + drProductId
+ " AND size=" + drProductSize);
            foreach(DataRow supplyDataRow in supplyDrArr)
            {
                float price = float.Parse(supplyDataRow["price"].ToString()) *
drAmount;
                int stock = int.Parse(supplyDataRow["amount"].ToString());

                dataRow["price"] = price;
                dataRow["stock"] = stock;
                if (stock < drAmount)
                {
                    ViewState["outOfStock"] = true;
                    outOfStockItemList.Add(dataRow["name"].ToString());
                    outOfStockSizeList.Add(drProductSize);
                }
                totalPrice += price;
            }
        }

        TotalPriceLbl.Text = "<h3> Total Price: " + totalPrice.ToString("0.00")
+ "</h3>";
        ViewState["totalPrice"] = totalPrice;

        GridView1.DataSource = orderProductTbl;
        GridView1.DataBind();

        CartErrorLbl.Visible = false;

        if (outOfStockItemList.Count > 0)
        {
            ViewState["outOfStockItemList"] = outOfStockItemList;
        }
    }
}

```

```

        ViewState["outOfStockSizeList"] = outOfStockSizeList;
    }
}

protected void NotifyStockAdmins(List<string> outOfStockItemList, List<int>
outOfStockSizeList)
{
    List<string> adminEmailList = new List<string>(), adminNameList = new
List<string>();

    DataTable supplyAdminTbl =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM userInfo WHERE
permLvl>1");

    foreach(DataRow dataRow in supplyAdminTbl.Rows)
    {
        adminEmailList.Add(dataRow["email"].ToString());
        adminNameList.Add(dataRow["username"].ToString());
    }

    MyServiceReference.ShoeCalculatorServiceSoapClient client = new
MyServiceReference.ShoeCalculatorServiceSoapClient();
    client.SendStockNotification(Session["username"].ToString(),
adminEmailList.ToArray(), adminNameList.ToArray(),
outOfStockItemList.ToArray(), outOfStockSizeList.ToArray());
}

protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
{

}

protected void GridView1_RowCancelingEdit(object sender,
GridViewCancelEventArgs e)
{
    GridView1.EditIndex = -1;
    GetData();
}

protected void GridView1_RowEditing(object sender, GridViewEditEventArgs e)
{
    GridView1.EditIndex = e.NewEditIndex;
    GetData();
}

protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs
e)
{
    string currentUser = Session["username"].ToString();
    DataTable orderTbl = MyAdoHelper.GetOrderTblByUser(currentUser);

    if(orderTbl.Rows.Count < 1)
    {
        Session["cartStatus"] = "error";
        Response.Redirect("ClimbingShop.aspx");
    }

    DataTable supplyTbl = MyAdoHelper.GetSupplyTbl();

    int orderId = int.Parse(orderTbl.Rows[0][0].ToString());

    GridViewRow DeletedRow = GridView1.Rows[e.RowIndex];
}

```

```

        if (GridView1.Rows.Count > 1)
    {
        int size = int.Parse(DeletedRow.Cells[3].Text);
        int productId = int.Parse(DeletedRow.Cells[2].Text);
        if(MyAdoHelper.IsExist("MyDatabase.mdf", "SELECT * FROM
tblOrderProduct WHERE orderId='" + orderId + "' AND productId='" + productId +
"' AND size='" + size + "')")
            MyAdoHelper.DoQuery("MyDatabase.mdf", "DELETE FROM
tblOrderProduct WHERE orderId='" + orderId + "' AND productId='" + productId +
"' AND size='" + size + "'");
        }
        else
        {
            MyAdoHelper.DoQuery("MyDatabase.mdf", "DELETE FROM tblOrder WHERE
orderId='" + orderId + "'");
            Session["cartStatus"] = "empty";
            Response.Redirect("ClimbingShop.aspx");
        }
        GridView1.EditIndex = -1;
        GetData();
        Response.Redirect("ShoppingCart.aspx");
    }

    protected void GridView1_PageIndexChanged(object sender,
GridViewEventArgs e)
{
    GridView1PageIndex = e.NewPageIndex;
    GetData();
}

protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs
e)
{
    string currentUser = Session["username"].ToString();
    DataTable orderTbl = MyAdoHelper.GetOrderTblByUser(currentUser);

    if (orderTbl.Rows.Count < 1)
    {
        Session["cartStatus"] = "error";
        Response.Redirect("ClimbingShop.aspx");
    }

    DataTable supplyTbl = MyAdoHelper.GetSupplyTbl();

    int orderId = int.Parse(orderTbl.Rows[0][0].ToString());

    GridViewRow changedRow = GridView1.Rows[e.RowIndex];
    int productId = int.Parse(changedRow.Cells[2].Text);
    int size = int.Parse(changedRow.Cells[3].Text);
    int amount =
int.Parse(((TextBox)changedRow.Cells[4].FindControl("AmountTxt")).Text);
    if (MyAdoHelper.IsExist("MyDatabase.mdf", "SELECT * FROM
tblOrderProduct WHERE orderId='" + orderId + "' AND productId='" + productId +
"' AND size='" + size + "')")
        MyAdoHelper.UpdateAmount(orderId, productId, size, amount);
    else
    {
        Session["cartStatus"] = "error";
        Response.Redirect("ClimbingShop.aspx");
    }
    GridView1.EditIndex = -1;
    GetData();
}

```

```

}

protected void PaymentButton_Click(object sender, EventArgs e)
{
    if (bool.Parse(ViewState["outOfStock"].ToString()))
    {
        NotifyStockAdmins((List<string>)ViewState["outOfStockItemList"],
(List<int>)ViewState["outOfStockSizeList"]);
        GetData();
        CartErrorLbl.Text = "<h3 class='red'>You are trying to order more
of an item than is in stock!</h3>";
        CartErrorLbl.Visible = true;
        return;
    }

    string currentUser = Session["username"].ToString();

    DataTable orderTbl = MyAdoHelper.GetOrderTblByUser(currentUser);

    if (orderTbl.Rows.Count < 1)
    {
        Session["cartStatus"] = "empty";
        Response.Redirect("ClimbingShop.aspx");
    }

    DataTable supplyTbl = MyAdoHelper.GetSupplyTbl();
    int orderId = int.Parse(orderTbl.Rows[0][0].ToString());
    DataTable orderProductTbl = MyAdoHelper.GetOrderProductTbl(orderId);

    ExpireDateRangeValidator.MinimumValue =
DateTime.Today.AddMonths(1).ToString("yyyy/MM/dd");
    ExpireDateRangeValidator.MaximumValue =
DateTime.Today.AddYears(15).ToString("yyyy/MM/dd");

    DataTable cardTbl = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf",
"SELECT * FROM tblCard WHERE username='" + currentUser + "'");

    if(cardTbl.Rows.Count < 1)
    {
        SavedCardDDL.Visible = false;
        SavedCardHeader.Visible = false;
    }
    else
    {
        foreach(DataRow dataRow in cardTbl.Rows)
        {
            SavedCardDDL.Items.Add(dataRow["cardNum"].ToString());
        }
    }

    PaymentButton.Visible = false;
    CancelPaymentButton.Visible = true;
    PurchaseErrorLbl.Visible = false;
    GridView1.Visible = false;
    PaymentDiv.Visible = true;
}

protected void GetReceiptXML(int orderId, string username)
{
    DataTable orderTbl = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf",
"SELECT * FROM tblOrder WHERE orderId='" + orderId + "'");
}

```

```

        DataTable orderProductTbl =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblOrderProduct
WHERE orderId=''' + orderId + '''");
        DataTable userInfoTbl = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf",
"SELECT * FROM userInfo WHERE username=''' + username + '''");
        DataTable productTbl = MyAdoHelper.GetProductTbl();

        orderTbl.TableName = "BasicOrderInfo";
        orderProductTbl.TableName = "ProductDetails";
        productTbl.TableName = "ProductTable";

        string email = userInfoTbl.Rows[0]["email"].ToString();

        MyServiceReference.ShoeCalculatorServiceSoapClient client = new
MyServiceReference.ShoeCalculatorServiceSoapClient();

        DataSet infoDs = new DataSet();

        infoDs.Tables.Add(orderTbl);
        infoDs.Tables.Add(orderProductTbl);
        infoDs.Tables.Add(productTbl);

        client.SendReceiptEmail(infoDs, email, username);
    }

protected void FinalPurchaseButton_Click(object sender, EventArgs e)
{
    try
    {
        PurchaseErrorLbl.Visible = false;

        string currentUser = Session["username"].ToString();

        DataTable orderTbl = MyAdoHelper.GetOrderTblByUser(currentUser);

        if (orderTbl.Rows.Count < 1)
        {
            Session["cartStatus"] = "empty";
            Response.Redirect("ClimbingShop.aspx");
        }

        DataTable supplyTbl = MyAdoHelper.GetSupplyTbl();
        int orderId = int.Parse(orderTbl.Rows[0][0].ToString());
        DataTable orderProductTbl =
MyAdoHelper.GetOrderProductTbl(orderId);

        float newTotalPrice = 0, totalPrice =
float.Parse(ViewState["totalPrice"].ToString());

        foreach (DataRow dataRow in orderProductTbl.Rows)
        {
            int drProductId = int.Parse(dataRow["productId"].ToString());
            int drProductSize = int.Parse(dataRow["size"].ToString());
            int drAmount = int.Parse(dataRow["amount"].ToString());

            DataRow[] priceDrArr = supplyTbl.Select("productId=" +
drProductId + " AND size=" + drProductSize);
            foreach (DataRow priceDataRow in priceDrArr)
            {
                newTotalPrice +=
float.Parse(priceDataRow["price"].ToString()) * drAmount;
            }
        }
    }
}

```

```

        }

        if (totalPrice != newTotalPrice)
        {
            GetData();
            CartErrorLbl.Text = "<h3 class='red'>Something in your shopping
cart was changed! We suggest you check it and try again</h3>";
            CartErrorLbl.Visible = true;
            GridView1.Visible = true;
            PaymentDiv.Visible = false;
            CancelPaymentButton.Visible = false;
            PaymentButton.Visible = true;
            return;
        }

        string cardNum = Request.Form[CardNumTextBox.UniqueID];
        string cvv = Request.Form[CVVTextBox.UniqueID];
        string expireDate = Request.Form[ExpireDateTextBox.UniqueID];
        bool rememberCard = SaveCardBox.Checked;

        OnlineServiceReference.LuhnCheckerSoapClient client = new
        OnlineServiceReference.LuhnCheckerSoapClient();

        bool cardValid = client.CheckCC(cardNum).CardValid;
        string company = client.CheckCC(cardNum).CardType;

        if (cardValid == false || company.Equals("Unknown"))
        {
            PurchaseErrorLbl.Text = "<h3 class='red'>The credit card you
entered is invalid!</h3>";
            PurchaseErrorLbl.Visible = true;
            return;
        }

        DateTime expireDateNormalized = DateTime.Parse(expireDate);
        expireDateNormalized = new DateTime(expireDateNormalized.Year,
        expireDateNormalized.Month, 1);

        DateTime myDateTime = DateTime.UtcNow;
        string sqlFormattedDate = myDateTime.ToString("yyyy-MM-dd
HH:mm:ss.fff");

        expireDate = expireDateNormalized.ToString("yyyy-MM-dd");

        if (rememberCard)
        {
            if(MyAdoHelper.IsExist("MyDatabase.mdf", "SELECT * FROM tblCard
WHERE cardNum='" + cardNum + "' AND username='" + currentUser + "')")
            {
                DataTable cardDt =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblCard WHERE
cardNum='" + cardNum + "' AND username='" + currentUser + "'");
                if (!(cardDt.Rows[0]["CVV"].ToString().Equals(cvv) &&
((DateTime)cardDt.Rows[0]["expireDate"]).ToString("yyyy-MM-
dd").Equals(expireDate)))
                    MyAdoHelper.DoQuery("MyDatabase.mdf", "UPDATE tblCard
SET CVV='" + cvv + "', expireDate='" + expireDate + "' WHERE cardNum='" +
cardNum + "' AND username='" + currentUser + "'");
            }
            else
            {
        }
    }
}

```

```
        MyAdoHelper.CreateNewCard(cardNum, currentUser, company,
cvv, expireDate);
    }
}

MyAdoHelper.DoQuery("MyDatabase.mdf", "UPDATE tblOrder SET paid='"
+ totalPrice + ', datetime=' + sqlFormattedDate + ' WHERE orderId="'
+ orderId);

GetReceiptXML(orderId, currentUser);

Session["cartStatus"] = "bought";

Response.Redirect("ClimbingShop.aspx");
}
catch
{
    PurchaseErrorLbl.Text = "<h3 class='red'>There seems to have been
an error! Try again later</h3>";
    PurchaseErrorLbl.Visible = true;
}
}

protected void CancelPaymentButton_Click(object sender, EventArgs e)
{
    GridView1.Visible = true;
    PaymentDiv.Visible = false;
    CancelPaymentButton.Visible = false;
    PaymentButton.Visible = true;
    GetData();
}

[ScriptMethod()]
[WebMethod(Description = "Get Card Data")]
public List<string> GetCreditCards(string cardNum)
{
    List<string> cardList = new List<string>();

    try
    {
        DataTable cardTbl =
MyAdoHelper.GetCardAutocomplete(Session["username"].ToString(), cardNum);
        foreach (DataRow dataRow in cardTbl.Rows)
        {
            cardList.Add(dataRow["cardNum"].ToString());
        }

        return cardList;
    }
    catch
    {
        return cardList;
    }
}

protected void SavedCardDDL_SelectedIndexChanged(object sender, EventArgs e)
{
}

protected void SavedCardButton_Click(object sender, EventArgs e)
```

```
{  
    string cardNum = Request.Form[SavedCardDDL.UniqueID];  
  
    DataTable cardTbl = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf",  
"SELECT * FROM tblCard WHERE username=" + Session["username"].ToString() + "  
AND cardNum=" + cardNum + "");  
  
    if (cardTbl.Rows.Count < 1)  
    {  
        SavedCardDDL.Items.Remove(cardNum);  
        return;  
    }  
  
    CardNumTextBox.Text = cardNum;  
    CVVTextBox.Text = cardTbl.Rows[0]["cvv"].ToString();  
    ExpireDateTextBox.Text =  
((DateTime)cardTbl.Rows[0]["expireDate"]).ToString("yyyy-MM-dd");  
}  
}
```

EditDataPage.aspx

מיקום: ComputersProjectNew/EditDataPage.aspx

מיקום קוד מאחור: ComputersProjectNew/EditDataPage.aspx.cs

קוד aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="EditDataPage.aspx.cs" Inherits="EditDataPage"
%>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>Edit Data Page</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2">
        <div class="inner">
            <div class="box">
                <div class="content">
                    <header class="align-center">
                        <h2>User Update Form:</h2>
                    </header>
                    <form id="form1" runat="server" method="post">
                        <div class="row uniform">
                            <div class="6u 12u$(small)">
                                <h3>Username:</h3>
                            </div>
                            <div class="6u$ 12u$(small)">
                                <asp:TextBox id="UsernameTextBox"
runat="server" ReadOnly="true"></asp:TextBox>
                            </div>
                            <div class="6u 12u$(small)">
                                <h3>Password:</h3>
                            </div>
                            <div class="6u$ 12u$(small)">
                                <asp:TextBox id="PasswordTextBox"
runat="server"></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="PasswordTextBox" Display="None" ID="RequiredFieldValidator2"
runat="server" ErrorMessage="Password is
required"></asp:RequiredFieldValidator>
                                <asp:RegularExpressionValidator
ValidationExpression="^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{5,50}$"
ControlToValidate="PasswordTextBox" Display="None"
ID="RegularExpressionValidator2" runat="server" ErrorMessage="Password must be
between 5 and 50 characters long, and contain numbers and uppercase and
lowercase letters"></asp:RegularExpressionValidator>
                            </div>
                            <div class="6u 12u$(small)">
                                <h3>City:</h3>
                            </div>
                            <div class="6u$ 12u$(small)">
                                <div class="select-wrapper">
                                    <asp:DropDownList ID="CityDDL"
runat="server">
                                </div>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </section>
    </div>
</asp:Content>
```

```

        <asp:ListItem Value="Haifa">
Selected="True">Haifa</asp:ListItem>
        <asp:ListItem Value="Tel Aviv">Tel
Aviv</asp:ListItem>
        <asp:ListItem>
Value="Jerusalem">Jerusalem</asp:ListItem>
        <asp:ListItem>
Value="Ashdod">Ashdod</asp:ListItem>
        <asp:ListItem Value="Rishon
LeZiyon">Rishon LeZiyon</asp:ListItem>
        <asp:ListItem Value="Petah
Tikva">Petah Tikva</asp:ListItem>
        <asp:ListItem Value="Beer
Sheva">Beer Sheva</asp:ListItem>
        <asp:ListItem Value="Netanya">Netanya</asp:ListItem>
        <asp:ListItem Value="Holon">Holon</asp:ListItem>
        <asp:ListItem Value="Bnei
Brak">Bnei Brak</asp:ListItem>
        <asp:ListItem Value="Rehovot">Rehovot</asp:ListItem>
        <asp:ListItem Value="Bat Yam">Bat
Yam</asp:ListItem>
        </asp:DropDownList>
    </div>
    <asp:RequiredFieldValidator
ControlToValidate="CityDDL" Display="None" ID="RequiredFieldValidator3"
runat="server" ErrorMessage="City is required"></asp:RequiredFieldValidator>
    </div>

    <div class="6u 12u$(small)">
        <h3>Address:</h3>
    </div>
    <div class="6u$ 12u$(small)">
        <asp:TextBox id="AddressTextBox"
runat="server"></asp:TextBox>
        <asp:RequiredFieldValidator
ControlToValidate="AddressTextBox" Display="None" ID="RequiredFieldValidator4"
runat="server" ErrorMessage="Address is required"></asp:RequiredFieldValidator>
        <asp:RegularExpressionValidator
ValidationExpression="^([A-Z][a-z]*\s)*[1-9][0-9]*$"
ControlToValidate="AddressTextBox" Display="None"
ID="RegularExpressionValidator3" runat="server" ErrorMessage="Wrong address
syntax"></asp:RegularExpressionValidator>
    </div>

    <div class="6u 12u$(small)">
        <h3>Gender:</h3>
    </div>
    <div class="6u$ 12u$(small)">
        <asp:RadioButtonList ID="GenderRBL"
runat="server">
            <asp:ListItem
Value="male">Male</asp:ListItem>
            <asp:ListItem
Value="female">Female</asp:ListItem>
        </asp:RadioButtonList>
        <asp:RequiredFieldValidator
ControlToValidate="GenderRBL" Display="None" ID="RequiredFieldValidator9"
runat="server" ErrorMessage="Gender is required"></asp:RequiredFieldValidator>
    </div>

```

```

<div class="6u 12u$(small)">
    <h3>Birthdate:</h3>
</div>
<div class="6u$ 12u$(small)">
    <asp:TextBox id="BirthdateTextBox"
runat="server" TextMode="Date"></asp:TextBox>
        <asp:RequiredFieldValidator
ControlToValidate="BirthdateTextBox" Display="None"
ID="RequiredFieldValidator8" runat="server" ErrorMessage="Birthdate is
required"></asp:RequiredFieldValidator>
        <asp:RangeValidator
id="BirthDateRangeValidator" runat="server" Type="Date"
ControlToValidate="BirthdateTextBox" ErrorMessage="Invalid age, out of range"
Display="None"/>
    </div>

    <div class="6u 12u$(small)">
        <h3>Telephone:</h3>
    </div>
    <div class="6u$ 12u$(small)">
        <asp:TextBox id="EmailTextBox"
runat="server"></asp:TextBox>
        <asp:RequiredFieldValidator
ControlToValidate="EmailTextBox" Display="None" ID="RequiredFieldValidator5"
runat="server" ErrorMessage="Email is required"></asp:RequiredFieldValidator>
        <asp:RegularExpressionValidator
ValidationExpression="^[\w-\.]+@[\\w-]+\.\w+[\\w-]{2,4}$"
ControlToValidate="EmailTextBox" Display="None"
ID="RegularExpressionValidator4" runat="server" ErrorMessage="Wrong email
syntax"></asp:RegularExpressionValidator>
    </div>

    <div class="6u 12u$(small)">
        <h3>Best Boulderding
Grade (V-Scale):</h3>
    </div>
    <div class="6u$ 12u$(small)">
        <asp:TextBox id="BoulderGradeTextBox"
runat="server"></asp:TextBox>
        <asp:RequiredFieldValidator
ControlToValidate="BoulderGradeTextBox" Display="None"
ID="RequiredFieldValidator6" runat="server" ErrorMessage="Boulder grade is
required"></asp:RequiredFieldValidator>
        <asp:RegularExpressionValidator
ValidationExpression="^V([0-9]|1[0-7])$"
ControlToValidate="BoulderGradeTextBox" Display="None"
ID="RegularExpressionValidator5" runat="server" ErrorMessage="Wrong boulder
grade syntax. Must be V0 to V17"></asp:RegularExpressionValidator>
    </div>

    <div class="6u 12u$(small)">
        <h3>Best Sport
Climbing Grade (French, capital letters):</h3>
    </div>
    <div class="6u$ 12u$(small)">
        <asp:TextBox id="WallGradeTextBox"
runat="server"></asp:TextBox>
        <asp:RequiredFieldValidator
ControlToValidate="WallGradeTextBox" Display="None"
ID="RequiredFieldValidator7" runat="server" ErrorMessage="Wall grade is
required"></asp:RequiredFieldValidator>

```

```
        <asp:RegularExpressionValidator  
ValidationExpression="^([5-9][A-C][+]{0,1})$"  
ControlToValidate="WallGradeTextBox" Display="None"  
ID="RegularExpressionValidator6" runat="server" ErrorMessage="Wrong wall grade  
syntax. Must be 5A to 9C (+ is optional)"></asp:RegularExpressionValidator>  
        </div>  
        <div class="12u$"></div>  
  
        <div class="12u$"><asp:ValidationSummary  
runat="server" ShowMessageBox="true" ShowSummary="false" /></div>  
  
        <div class="12u$">  
            <div class="align-center">  
                <asp:Button runat="server"  
ID="SubmitDataButton" OnClick="SubmitUpdate_Click" Text="Update User"/>  
            </div>  
        </div>  
  
        <div class="12u$">  
            <div class="align-center">  
                <asp:Button runat="server"  
ID="DeleteUserButton" OnClick="Delete_Click" Text="Delete User"/>  
            </div>  
        </div>  
  
        <div class="12u$"><div class="align-center"><h3  
class="red"><%= updateDataMsg %></h3></div></div>  
            </div>  
            </form>  
        </div>  
    </div>  
    </div>  
</section>  
</asp:Content>
```

#C קוד

```

using System;
using System.Data;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class EditDataPage : System.Web.UI.Page
{
    public string updateDataMsg = "";
    protected void Page_Load(object sender, EventArgs e)
    {
        string username = Session["username"].ToString();

        DataTable dataTable = MyAdoHelper.GetUserInfo(username); //tblת את שומר
        המשמש ערכיו

        UsernameTextBox.Text = username;
        PasswordTextBox.Text = dataTable.Rows[0]["password"].ToString(); //ממיר
        אותו ושומר לסתירינג המשתמש ססמת את
        CityDDL.Text = dataTable.Rows[0]["city"].ToString(); //המשמש עיר את ממיר
        אותו ושומר לסתירינג
        EmailTextBox.Text = dataTable.Rows[0]["email"].ToString(); //מספר את ממיר
        אותו ושומר לסתירינג המשתמש טלפון
        GenderRBL.Text = dataTable.Rows[0]["gender"].ToString(); //מין את ממיר
        אותו ושומר לסתירינג המשתמש
        BoulderGradeTextBox.Text =
        dataTable.Rows[0]["boulderGrade"].ToString(); //
        WallGradeTextBox.Text = dataTable.Rows[0]["wallGrade"].ToString(); //
        AddressTextBox.Text = dataTable.Rows[0]["address"].ToString(); //

        DateTime birthdate =
        DateTime.Parse(dataTable.Rows[0]["birthdate"].ToString());
        BirthdateTextBox.Text = birthdate.ToString("yyyy-MM-dd");

        BirthDateRangeValidator.MinimumValue = DateTime.Today.AddYears(-
        120).ToString("yyyy/MM/dd");
        BirthDateRangeValidator.MaximumValue = DateTime.Today.AddYears(-
        8).ToString("yyyy/MM/dd");

        PasswordTextBox.Attributes["type"] = "password";
    }

    protected void SubmitUpdate_Click(object sender, EventArgs e)
    {
        string username = Session["username"].ToString();

        string newPassword = Request.Form[PasswordTextBox.UniqueID]; //את שולף
        מהטופס המשתמש של החדש הסיסמה
        string newCity = Request.Form[CityDDL.UniqueID]; //של החדש העיר את שולף
        מהטופס המשתמש
        string newEmail = Request.Form[EmailTextBox.UniqueID]; //מספר את שולף
        מהטופס המשתמש של החדש הטלפון
        string newGender = Request.Form[GenderRBL.UniqueID]; //של החדש המין את שולף
        מהטופס המשתמש
        string newBirthdate = Request.Form[BirthdateTextBox.UniqueID]; //את שולף
        מהטופס המשתמש של החדש הלידה תאריך
        string newBoulderGrade = Request.Form[BoulderGradeTextBox.UniqueID]; //
    }
}

```

```
string newWallGrade = Request.Form[WallGradeTextBox.UniqueID]; //  
string newAddress = Request.Form[AddressTextBox.UniqueID];  
  
bool successUpdate = MyAdoHelper.UpdateUser(username, newPassword,  
newEmail, newAddress, newBoulderGrade, newWallGrade, newBirthdate, newCity,  
newGender);  
  
if (!successUpdate)  
    updateDataMsg = "Oops! There appears to have been an error. Please  
try again later.";// כתוב באנגלית ובלשון העברית  
else  
    Response.Redirect("HomePage.aspx"); // כתוב באנגלית ובלשון העברית  
}  
  
protected void Delete_Click(object sender, EventArgs e)  
{  
    string username = Session["username"].ToString();  
  
    bool successDelete = MyAdoHelper.DeleteUser(username);  
    if (!successDelete)  
        updateDataMsg = "Oops! There appears to have been an error. Please  
try again later.";// כתוב באנגלית ובלשון העברית  
    else  
    {  
        Session.Abandon(); // כתוב באנגלית ובלשון העברית  
        Application.Lock();  
        Application["userNum"] = (int)Application["userNum"] - 1;  
        Application.UnLock();  
        Response.Redirect("HomePage.aspx"); // כתוב באנגלית ובלשון העברית  
    }  
}
```

ShoeCalcPage.aspx

מיקום: ComputersProjectNew/ShoeCalcPage.aspx

מיקום קוד מאחור: ComputersProjectNew/ShoeCalcPage.aspx.cs

קוד aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="ShoeCalcPage.aspx.cs" Inherits="ShoeCalcPage"
%>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>Shoe Size Calculator</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2">
        <div class="inner">
            <div class="box">
                <div class="content">
                    <header class="align-center">
                        <h2>Shoe Size Calculator</h2>
                    </header>
                    <br />
                    <div class="12u$"><div class="align-center"><h3 <%
messageHidden %>Your suggested size is: <%= size %></h3></div></div>
                    <br />
                    <form id="form1" runat="server" method="post">
                        <div class="row uniform">
                            <div class="6u 12u$(small)">
                                <h3>Street shoe
size:</h3>
                                </div>
                            <div class="6u$ 12u$(small)">
                                <asp:TextBox ID="NormalSizeTextBox"
runat="server"></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="NormalSizeTextBox" Display="None"
ID="RequiredFieldValidator1" runat="server" ErrorMessage="Street shoe size is
required" ValidationGroup="ShoeCalcValidate"></asp:RequiredFieldValidator>
                                <asp:RegularExpressionValidator
ValidationExpression="^([3-5][0-9])|(60)$"
ControlToValidate="NormalSizeTextBox" Display="None"
ID="RegularExpressionValidator1" runat="server" ErrorMessage="Shoe size must be
an integer between 30 and 60!">
                                </asp:RegularExpressionValidator>
                            </div>
                            <div class="6u 12u$(small)">
                                <h3>Company:</h3>
                            </div>
                            <div class="6u$ 12u$(small)">
                                <asp:DropDownList ID="CompanyNameDDL"
runat="server">
                                    <asp:ListItem Value="La Sportiva"
Selected="True">La Sportiva</asp:ListItem>
                                    <asp:ListItem
Value="Scarpa">Scarpa</asp:ListItem>
                                </asp:DropDownList>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </section>
</div>

```

```
        <asp:ListItem  
Value="Unparallel">Unparallel</asp:ListItem>  
        <asp:ListItem  
Value="Madrock">Madrock</asp:ListItem>  
        <asp:ListItem  
Value="Adidas">Adidas</asp:ListItem>  
    </asp:DropDownList>  
    <asp:RequiredFieldValidator  
ControlToValidate="CompanyNameDDL" Display="None" ID="RequiredFieldValidator2"  
runat="server" ErrorMessage="Company name is required"  
ValidationGroup="ShoeCalcValidate"></asp:RequiredFieldValidator>  
    </div>  
  
    <div class="6u 12u$(small)">  
        <h3>Fit:</h3>  
    </div>  
    <div class="6u$ 12u$(small)">  
        <asp:DropDownList ID="FitTypeDDL"  
runat="server">  
            <asp:ListItem Value="Loose"  
Selected="True">Loose</asp:ListItem>  
            <asp:ListItem  
Value="Normal">Normal</asp:ListItem>  
            <asp:ListItem  
Value="Tight">Tight</asp:ListItem>  
        </asp:DropDownList>  
        <asp:RequiredFieldValidator  
ControlToValidate="FitTypeDDL" Display="None" ID="RequiredFieldValidator3"  
runat="server" ErrorMessage="Fit type is required"  
ValidationGroup="ShoeCalcValidate"></asp:RequiredFieldValidator>  
    </div>  
  
    <div class="12u$"><div class="align-  
center"><asp:Button runat="server" ID="submitData" OnClick="Submit_Click"  
Text="Calculate" CausesValidation="true"  
ValidationGroup="ShoeCalcValidate"/></div></div>  
  
    <div class="12u$"><asp:ValidationSummary  
runat="server" ShowMessageBox="true" ShowSummary="false"  
ValidationGroup="ShoeCalcValidate"/></div>  
    </div>  
    </form>  
    </div>  
    </div>  
    </section>  
</asp:Content>
```

:C# טיפ

```
using System;
using System.Data;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class ShoeCalcPage : System.Web.UI.Page
{
    public int size = 1;
    public string messageHidden = "hidden='hidden'";
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void Submit_Click(object sender, EventArgs e)
    {
        MyServiceReference.ShoeCalculatorServiceSoapClient client = new
MyServiceReference.ShoeCalculatorServiceSoapClient();

        int normalSize = int.Parse(Request.Form[NormalSizeTextBox.UniqueID]);
        string company = Request.Form[CompanyNameDDL.UniqueID];
        string fit = Request.Form[FitTypeDDL.UniqueID];

        DataTable dt = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT
boulderGrade, wallGrade FROM userInfo WHERE username=" +
Session["username"].ToString() + "'");

        string boulderGrade = dt.Rows[0][0].ToString();
        string wallGrade = dt.Rows[0][1].ToString();

        size = client.ShoeCalc(normalSize, company, fit, boulderGrade,
wallGrade);
        messageHidden = "";
    }
}
```

WeatherPage.aspx

מיקום: ComputersProjectNew/WeatherPage.aspx

מיקום קוד מאחור: ComputersProjectNew/WeatherPage.aspx.cs

קוד aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="WeatherPage.aspx.cs" Inherits="WeatherPage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2">
        <div class="inner">
            <div class="box">
                <div class="content">
                    <header class="align-center">
                        <h2>Always check the weather before going climbing
outdoors!</h2>
                    </header>

                    <form id="form1" runat="server" method="post">
                        <div class="row uniform">
                            <div class="6u 12u$(small)">
                                <h3>City:</h3>
                                </div>
                            <div class="6u$ 12u$(small)">
                                <asp:TextBox id="CityTextBox"
runat="server"></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="CityTextBox" Display="None" ID="RequiredFieldValidator1"
runat="server" ErrorMessage="City is required"></asp:RequiredFieldValidator>
                                <asp:RegularExpressionValidator
ValidationExpression="^[a-zA-Z\d]+(?: [A-Za-z\d]{2,})*$"
ControlToValidate="CityTextBox" Display="None" ID="RegularExpressionValidator1"
runat="server" ErrorMessage="City name must be one or more
words"></asp:RegularExpressionValidator>
                            </div>

                            <div class="6u 12u$(small)">
                                <h3>Country:</h3>
                                </div>
                            <div class="6u$ 12u$(small)">
                                <asp:TextBox id="CountryTextBox"
runat="server"></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="CountryTextBox" Display="None" ID="RequiredFieldValidator2"
runat="server" ErrorMessage="Country is required"></asp:RequiredFieldValidator>
                                <asp:RegularExpressionValidator
ValidationExpression="^[a-zA-Z\d]+(?: [A-Za-z\d]{2,})*$"
ControlToValidate="CountryTextBox" Display="None"
ID="RegularExpressionValidator2" runat="server" ErrorMessage="Country name must
be one or more words"></asp:RegularExpressionValidator>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </section>
    </div>
</body>
</html>

```

```

        <div class="align-center">
            <asp:Button runat="server"
ID="weatherButton" OnClick="GetWeather" Text="Get Weather"/>
        </div>
    </div>

    <div class="12u$">
        <div class="align-center">
            <asp:Image ID="IconImg" runat="server"
Height="100" Width="100" Visible="false"/>
        </div>
    </div>

    <div class="12u$">
        <div class="align-center">
            <asp:Label ID="EnterCityLbl"
runat="server" Text="<h3>Enter a city name!</h3>"></asp:Label>
        </div>
    </div>

    <div class="12u$">
        <div class="align-center">
            <asp:Label id="CityCountryLbl"
runat="server" Visible="false"></asp:Label>
        </div>
    </div>

    <div class="12u$">
        <div class="align-center">
            <asp:Label ID="AlertLbl" runat="server"
Visible="false"></asp:Label>
        </div>
    </div>

        <div class="12u$">
            <asp:DataList ID="AlertDataList"
runat="server" RepeatDirection="Horizontal" RepeatLayout="Table"
RepeatColumns="3" DataKeyField="start" Width="400">
                <ItemStyle BackColor="HotPink"
BorderWidth="20" BorderColor="White" Width="400px"/>
                <AlternatingItemStyle
BackColor="PaleVioletRed" Width="400px"/>
                <ItemTemplate>
                    <asp:Image ID="AlertImg"
runat="server" ImageUrl="~/weatherIcons/newAlert.png" Width="400"
Height="400"/>
                    <asp:Label ID="SenderLbl"
runat="server" Text='<%# "<h4>Sender: " + Eval("sender") +
"/>%>'></asp:Label>
                    <asp:Label ID="EventLbl"
runat="server" Text='<%# "<h4>Event: " + Eval("event") +
"/>%>'></asp:Label>
                    <asp:Label ID="AlertStartLbl"
runat="server" Text='<%# "<h4>Starts: " + Eval("start") +
"/>%>'></asp:Label>
                    <asp:Label ID="AlertEndLbl"
runat="server" Text='<%# "<h4>Ends: " + Eval("end") + "</h4>%>'></asp:Label>
                    <asp:Label ID="DescLbl"
runat="server" Text='<%# "<h4>Description: " + Eval("desc") +
"/>%>'></asp:Label>
                </ItemTemplate>
            </asp:DataList>
        </div>
    </div>

```

```
</div>

<div class="12u$">
    <asp:DataList ID="WeatherDataList"
runat="server" RepeatDirection="Horizontal" RepeatLayout="Table"
RepeatColumns="4" DataKeyField="date">
        <ItemStyle BackColor="LightBlue"
BorderWidth="20" BorderColor="White"/>
        <AlternatingItemStyle
BackColor="Lavender" />
        <ItemTemplate>
            <asp:Image ID="ImageBox"
runat="server" ImageUrl='<%# "~/weatherIcons/" + Eval("icon") + ".png" %>' Height="250" Width="250"/>
            <asp:Label ID="DayDateLbl"
runat="server" Text='<%# "<h3>" + Eval("day") + ", " + Eval("date") + "</h3>" %>'></asp:Label>
            <asp:Label ID="WeatherStatusLbl"
runat="server" Text='<%# "<h3>" + Eval("weatherStatus") +
"</h3>"%>'></asp:Label>
            <asp:Label ID="TempLbl"
runat="server" Text='<%# "<h3>Temperature: " + Eval("temp") +
"</h3>"%>'></asp:Label>
            <asp:Label ID="MaxTempLbl"
runat="server" Text='<%# "<h3>Highest Temperature: " + Eval("maxTemp") +
"</h3>"%>'></asp:Label>
            <asp:Label ID="MinTempLbl"
runat="server" Text='<%# "<h3>Lowest Temperature: " + Eval("minTemp") +
"</h3>"%>'></asp:Label>
            <asp:Label ID="HumidityLbl"
runat="server" Text='<%# "<h3>Humidity: " + Eval("humidity") +
"%</h3>"%>'></asp:Label>
        </ItemTemplate>
    </asp:DataList>
</div>
<div class="12u$"><asp:ValidationSummary
runat="server" ShowMessageBox="true" ShowSummary="false" /></div>
</div>
</form>
</div>
</div>
</div>
</section>
</asp:Content>
```

:C# טיפ

```

using System;
using System.Data;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.Web.Script.Serialization;
using Newtonsoft.Json;
using System.Globalization;
using Twilio;
using Twilio.Rest.Api.V2010.Account;
using Vonage;
using Vonage.Request;
using System.Text.RegularExpressions;
using Nager;
using Nager.Country;
using Nager.Country.CountryInfos;

public partial class WeatherPage : System.Web.UI.Page
{
    public delegate string URLDeleg();
    public string weatherCountry, weatherCity;
    public const string badIPStr = "BadIP", badCountryStr = "BadCountry",
weatherApiKey = "43df790eef2340a0e861fed4cba26f37";
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["username"] == null)
            Response.Redirect("NoAccessPage.aspx");

        if(Session["city"] != null && Session["country"] != null)
        {
            GetWeather(ExternalWeather(Session["city"].ToString(),
Session["country"].ToString()));
            CityTextBox.Text = Session["city"].ToString();
            CountryTextBox.Text = Session["country"].ToString();
            Session["city"] = null;
            Session["country"] = null;
        }

        GetWeather(LocalWeather());
    }

    protected void GetWeather(object sender, EventArgs e)
    {
        GetWeather(ExternalWeather());
    }

    protected string LocalWeather()
    {
        string userIP = GetIPAddress();

        Regex reg = new Regex(@"^((25[0-5]|2[0-4][0-9]| [01]?[0-9][0-
9]?)\.){3}(25[0-5]|2[0-4][0-9]| [01]?[0-9][0-9]?)$");
        if (userIP == "127.0.0.1" || !reg.IsMatch(userIP))
    }
}

```

```

        /* Get Server IP
        string externalIpString = new
WebClient().DownloadString("http://icanhazip.com").Replace("\\r\\n",
"").Replace("\\n", "").Trim();
        var externalIp = IPAddress.Parse(externalIpString);

        userIP = externalIp.ToString();*/

        DataTable cityTbl = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf",
"SELECT city FROM userInfo WHERE username='" + Session["username"] + "'");
        string userCity = cityTbl.Rows[0]["city"].ToString();

        weatherCity = userCity;
        weatherCountry = "Israel";
        return ExternalWeather(userCity, "Israel");
    }

    using (WebClient client = new WebClient())
    {
        string locationUrl = string.Format("http://ip-api.com/json/{0}",
userIP);
        string json = client.DownloadString(locationUrl);
        InfoIP locInfo = JsonConvert.DeserializeObject<InfoIP>(json);
        double lat = locInfo.lat, lon = locInfo.lon;
        weatherCity = locInfo.city;
        weatherCountry = locInfo.country;
        string url =
string.Format("https://api.openweathermap.org/data/2.5/onecall?lat={0}&lon={1}&
exclude=current,minutely,hourly,alerts&units=metric&appid={2}", lat, lon,
weatherApiKey);
        return url;
    }
}

protected string ExternalWeather()
{
    weatherCity =
System.Threading.Thread.CurrentThread.CurrentCulture.TextInfo.ToTitleCase(CityT
extBox.Text.ToLower());
    weatherCountry =
System.Threading.Thread.CurrentThread.CurrentCulture.TextInfo.ToTitleCase(Count
yTextBox.Text.ToLower());

    return ExternalWeather(weatherCity, weatherCountry);
}

protected string ExternalWeather(string city, string country)
{
    string url;

    using (WebClient client = new WebClient())
    {
        try
        {
            ICountryProvider countryProvider = new CountryProvider();
            var countryInfo = countryProvider.GetCountryByName(country);

            string query = weatherCity + ",0," +
countryInfo.Alpha2Code.ToString();

```

```

        url =
string.Format("http://api.openweathermap.org/geo/1.0/direct?q={0}&appid={1}",
query, weatherApiKey);

        string json = client.DownloadString(url);
GeoRoot geoRoot =
JsonConvert.DeserializeObject<GeoRoot>(json.Substring(1, json.Length - 2));
        double lat = geoRoot.lat, lon = geoRoot.lon;
        url =
string.Format("https://api.openweathermap.org/data/2.5/onecall?lat={0}&lon={1}&
exclude=current,minutely,hourly&lang=en&units=metric&appid={2}", lat, lon,
weatherApiKey);
    }
    catch
{
    url = badCountryStr;
EnterCityLbl.Text = "<h3>Bad city/country name!</h3>";
EnterCityLbl.Visible = true;
CityCountryLbl.Visible = false;
WeatherDataList.Visible = false;
AlertDataList.Visible = false;
}
}

return url;
}

protected void GetWeather(string url)
{
    if (url.Equals(badIPStr) || url.Equals(badCountryStr))
        return;
    using (WebClient client = new WebClient())
    {
        try
{
        string json = client.DownloadString(url);
Root weatherRoot = JsonConvert.DeserializeObject<Root>(json);
DataTable weatherTbl = new DataTable();
DataTable alertTbl = new DataTable();

weatherTbl.Columns.Add("icon");
weatherTbl.Columns.Add("day");
weatherTbl.Columns.Add("date");
weatherTbl.Columns.Add("weatherStatus");
weatherTbl.Columns.Add("temp");
weatherTbl.Columns.Add("minTemp");
weatherTbl.Columns.Add("maxTemp");
weatherTbl.Columns.Add("humidity");

alertTbl.Columns.Add("sender");
alertTbl.Columns.Add("event");
alertTbl.Columns.Add("start");
alertTbl.Columns.Add("end");
alertTbl.Columns.Add("desc");

CityCountryLbl.Text = "<h3>Weather for " + weatherCity + ", " +
weatherCountry + "</h3>";

CityCountryLbl.Visible = true;
EnterCityLbl.Visible = false;

for (int i = 0; i < weatherRoot.daily.Count; i++)

```

```

{
    DataRow newRow = weatherTbl.NewRow();

    newRow[ "icon" ] = weatherRoot.daily[i].weather[0].icon;
    DateTime weatherDt =
UnixTimeStampToDateTIme(weatherRoot.daily[i].dt);
    newRow[ "day" ] = weatherDt.DayOfWeek;
    newRow[ "date" ] = weatherDt.ToString( "dd/MM/yyyy",
CultureInfo.InvariantCulture);
    newRow[ "weatherStatus" ] =
System.Threading.Thread.CurrentThread.CurrentCulture.TextInfo.ToTitleCase(weatherRoot.daily[i].weather[0].description);
    newRow[ "temp" ] = weatherRoot.daily[i].temp.day;
    newRow[ "minTemp" ] = weatherRoot.daily[i].temp.min;
    newRow[ "maxTemp" ] = weatherRoot.daily[i].temp.max;
    newRow[ "humidity" ] = weatherRoot.daily[i].humidity;

    weatherTbl.Rows.Add(newRow);
}

WeatherDataList.DataSource = weatherTbl;
WeatherDataList.DataBind();

WeatherDataList.Visible = true;
CityCountryLbl.Visible = true;

if (weatherRoot.alerts != null && weatherRoot.alerts.Count > 0)
{
    AlertLbl.Text = "<h2 class='red'>Alerts:</h2>";
    AlertLbl.Visible = true;
    for (int i = 0; i < weatherRoot.alerts.Count; i++)
    {
        DataRow newRow = alertTbl.NewRow();

        newRow[ "sender" ] = weatherRoot.alerts[i].sender_name;
        DateTime unixToDT =
UnixTimeStampToDateTIme(weatherRoot.alerts[i].start);
        newRow[ "start" ] = unixToDT.DayOfWeek + ", " +
unixToDT.ToString( "dd/MM/yyyy", CultureInfo.InvariantCulture);
        unixToDT =
UnixTimeStampToDateTIme(weatherRoot.alerts[i].end);
        newRow[ "end" ] = unixToDT.DayOfWeek + ", " +
unixToDT.ToString( "dd/MM/yyyy", CultureInfo.InvariantCulture);
        newRow[ "event" ] = weatherRoot.alerts[i].@event;
        newRow[ "desc" ] = weatherRoot.alerts[i].description;

        alertTbl.Rows.Add(newRow);
    }
}
else
{
    AlertLbl.Visible = false;
}

AlertDataList.DataSource = alertTbl;
AlertDataList.DataBind();

AlertDataList.Visible = true;
}

catch (Exception exp)
{

```

```
        EnterCityLbl.Text = "<h3>Bad city name!</h3>";
        WeatherDataList.Visible = false;
        AlertLbl.Visible = false;
        CityCountryLbl.Visible = false;
        AlertDataList.Visible = false;
        EnterCityLbl.Visible = true;
        return;
    }
}

protected string GetIPAddress()
{
    string ipaddress;
    ipaddress = Request.ServerVariables["HTTP_X_FORWARDED_FOR"];
    if (ipaddress == "" || ipaddress == null)
        ipaddress = Request.ServerVariables["REMOTE_ADDR"];
    return ipaddress;
}

public static DateTime UnixTimeStampToDateTime(double unixTimeStamp)
{
    // Unix timestamp is seconds past epoch
    DateTime dateTime = new DateTime(1970, 1, 1, 0, 0, 0, 0,
    DateTimeKind.Utc);
    dateTime = dateTime.AddSeconds(unixTimeStamp).ToLocalTime();
    return dateTime;
}

//https://json2csharp.com/ - very nice tool
}
```

UserList.aspx

מיקום: ComputersProjectNew/UserList.aspx

מיקום קוד מאחור: ComputersProjectNew/UserList.aspx.cs

קוד aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="UserList.aspx.cs" Inherits="UserList" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>User List</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2" id="three">
        <div class="inner">
            <header class="align-center">
                <h2>Contact List:</h2>
                <h4>Get in touch with our users!</h4>
            </header>
        </div>
        <form id="form1" runat="server">
            <div class="inner">
                <div class="box" style="background:#f2f2f2 !important">
                    <div class="content">
                        <div class="row uniform">
                            <div class="3u 12u$(small)"></div>
                            <div class="3u 12u$(small)">
                                <h3>User Search:</h3>
                            </div>
                            <div class="3u 12u$(small)">
                                <asp:TextBox id="UsernameTextBox"
runat="server"></asp:TextBox>
                                    <asp:RequiredFieldValidator
ControlToValidate="UsernameTextBox" Display="None" ID="RequiredFieldValidator1"
runat="server" ErrorMessage="Username is
required"></asp:RequiredFieldValidator>
                                    <asp:RegularExpressionValidator
ValidationExpression="^\w{1,50}$" ControlToValidate="UsernameTextBox"
Display="None" ID="RegularExpressionValidator1" runat="server"
ErrorMessage="Username must be between 1 and 50 characters
long"></asp:RegularExpressionValidator>
                            </div>
                            <div class="3u$ 12u$(small)"></div>
                            <div class="12u$">
                                <div class="align-center">
                                    <asp:Button ID="SearchButton"
runat="server" Text="Search" OnClick="SearchButton_Click"
CausesValidation="true"/>
                                    <asp:Button ID="ResetButton"
runat="server" Text="Reset Search" OnClick="ResetButton_Click"
CausesValidation="false"/>
                                </div>
                            </div>
                        </div>
                    <asp:ValidationSummary ID="ValidationSummary1"
runat="server" ShowMessageBox="true" ShowSummary="false" />
                </div>
            </form>
        </div>
    </section>
</div>
</body>
</html>
```

```
<div class="12u$">
    <div class="align-center">
        <asp:Label ID="ErrorLbl" runat="server"
Visible="false"></asp:Label>
    </div>
</div>
<br />
<br />
<asp:DataList ID="DataList1" runat="server"
RepeatDirection="Vertical" RepeatLayout="Table" CellSpacing="10"
DataKeyField="username" RepeatColumns="5">
    <ItemStyle BackColor="LightYellow" BorderWidth="50"
BorderColor="White"/>
    <ItemTemplate>
        <span><strong>Username: </strong></span>
        <asp:Label ID="Label1" runat="server" Text='<%#
Eval("username") %>'></asp:Label>
        <br />
        <span><strong>Email: </strong></span>
        <asp:Label ID="Label6" runat="server" Text='<%#
Eval("email") %>'></asp:Label>
        <br />
        <span><strong>Birth date: </strong></span>
        <asp:Label ID="Label2" runat="server" Text='<%#
Eval("birthdate", "{0:d}") %>'></asp:Label>
        <br />
        <span><strong>Gender: </strong></span>
        <asp:Label ID="Label3" runat="server" Text='<%#
Eval("gender").ToString().Substring(0,1).ToUpper() +
Eval("gender").ToString().Substring(1) %>'></asp:Label>
        <br />
        <span><strong>Best Bouldering Grade: </strong></span>
        <asp:Label ID="Label4" runat="server" Text='<%#
Eval("boulderGrade") %>'></asp:Label>
        <br />
        <span><strong>Best Wall Grade: </strong></span>
        <asp:Label ID="Label5" runat="server" Text='<%#
Eval("wallGrade") %>'></asp:Label>
        <br />
    </ItemTemplate>
</asp:DataList>
</form>
</section>
</asp:Content>
```

:C# קוד

```
using System;
using System.Data;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class UserList : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["username"] == null)//רשותם רשום
            Response.Redirect("NoAccessPage.aspx"); //כניתה מעביר את המשתמש לדף האין כניסה

        if (!IsPostBack)
        {
            GetUsers(MyAdoHelper.GetUserInfoTbl());
        }
    }

    public void GetUsers(DataTable userInfoTbl)
    {
        if (userInfoTbl.Rows.Count < 1)
        {
            DataList1.Visible = false;
            ErrorLbl.Text = "<h3 class='red'>No Data Found!</h3>";
            ErrorLbl.Visible = true;
            return;
        }
        DataList1.DataSource = userInfoTbl;
        DataList1.DataBind();
        DataList1.Visible = true;
        ErrorLbl.Visible = false;
    }

    protected void SearchButton_Click(object sender, EventArgs e)
    {
        string username = Request.Form[UsernameTextBox.UniqueID];
        GetUsers(MyAdoHelper.GetUsernameAutocomplete(username));
    }

    protected void ResetButton_Click(object sender, EventArgs e)
    {
        UsernameTextBox.Text = "";
        GetUsers(MyAdoHelper.GetUserInfoTbl());
    }
}
```

ManageCardPage.aspx

מיקום: ComputersProjectNew/ManageCardPage.aspx
מיקום קוד מאחור: ComputersProjectNew/ManageCardPage.aspx.cs
קוד aspx:

```

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="ManageCardPage.aspx.cs"
Inherits="ManageCardPage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>Credit Card Page</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2" id="three">
        <div class="inner">
            <header class="align-center">
                <h2>Credit Card Page</h2>
            </header>
            <div class="align-center">
                <input type="button" value="Add New Card"
onclick="location.href='NewCreditCardPage.aspx'" />
            </div>
        </div>
        <form id="form1" runat="server">
            <br />
            <br />
            <asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="False"
OnSelectedIndexChanged="GridView1_SelectedIndexChanged"
HorizontalAlign="Center" AllowPaging="True"
OnRowDeleting="GridView1_RowDeleting"
OnRowCancelingEdit="GridView1_RowCancelingEdit"
OnRowEditing="GridView1_RowEditing" OnRowUpdating="GridView1_RowUpdating"
OnPageIndexChanging="GridView1_PageIndexChanging" PageSize="5"
OnRowDataBound="GridView1_RowDataBound">
                <PagerStyle HorizontalAlign="Center"/>
                <Columns>
                    <asp:CommandField ButtonType="Button"
ShowEditButton="true" HeaderText="Editing" CancelText="Cancel Edit"
>Edit Text="Edit Card" UpdateText="Confirm Edit"/>
                    <asp:ButtonField ButtonType="Button"
Text="Delete" CommandName="Delete" HeaderText="Deletion" />
                    <asp:BoundField DataField="cardNum"
ReadOnly="true" HeaderText="Card Number"/>
                    <asp:BoundField DataField="company"
ReadOnly="true" HeaderText="Company"/>
                    <asp:TemplateField HeaderText="CVV"
SortExpression="telephone">
                        <EditItemTemplate>
                            <asp:TextBox ID="CVVTextBox"
runat="server" Text='<%# Bind("CVV") %>'></asp:TextBox>
                            <asp:RequiredFieldValidator
ControlToValidate="CVVTextBox" Display="None" ID="RequiredFieldValidator5"
runat="server" ErrorMessage="CVV is required"></asp:RequiredFieldValidator>
                            <asp:RegularExpressionValidator
ValidationExpression="^[0-9]{3}$" ControlToValidate="CVVTextBox" Display="None"

```

```
ID="RegularExpressionValidator4" runat="server" ErrorMessage="CVV must be a
number of length 3"></asp:RegularExpressionValidator>
        </EditItemTemplate>
        <ItemTemplate>
            <asp:Label ID="CVVLabel" runat="server">
Text='<%# Bind("CVV") %>'</asp:Label>
                </ItemTemplate>
            </asp:TemplateField>

            <asp:TemplateField HeaderText="Expiration
Date">
                <EditItemTemplate>
                    <asp:TextBox ID="ExpireDateTextBox"
runat="server" Text='<%# Bind("expireDate", "{0:yyyy-MM-dd}") %>'>
TextMode="Date"></asp:TextBox>
                    <asp:RequiredFieldValidator
ControlToValidate="ExpireDateTextBox" Display="None"
ID="RequiredFieldValidator8" runat="server" ErrorMessage="Birthdate is
required"></asp:RequiredFieldValidator>
                    <asp:RangeValidator
id="ExpireDateRangeValidator" runat="server" Type="Date"
ControlToValidate="ExpireDateTextBox" ErrorMessage="Invalid credit card
expiration date" Display="None" MinimumValue="<%# rvMin %>" MaximumValue="<%#
rvMax %>"/>
                </EditItemTemplate>
                <ItemTemplate>
                    <asp:Label ID="ExpireDateLabel"
runat="server" Text='<%# Bind("expireDate", "{0:d}") %>'></asp:Label>
                </ItemTemplate>
            </asp:TemplateField>

        </Columns>
    </asp:GridView>
    <asp:ValidationSummary ID="SummaryValidator"
runat="server" ShowMessageBox="true" ShowSummary="false" />
</form>
</section>
</asp:Content>
```

:C# קורס

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;

public partial class ManageCardPage : System.Web.UI.Page
{
    public string rvMin = DateTime.Today.AddMonths(1).ToString("yyyy/MM/dd"),
        rvMax = DateTime.Today.AddYears(15).ToString("yyyy/MM/dd");
    public string errorMsg = "";
    DataTable dt; // המוחזרים הנתונים את שומרן
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["username"] == null)// בודק אם משתמש רשום
            Response.Redirect("NoAccessPage.aspx"); // כניסה לאין לדף המשתמש את מעביר

        if (!IsPostBack)
        {
            getUsers();
        }
    }

    public void getUsers()
    {
        dt = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM
tblCard WHERE username=" + Session["username"].ToString() + "");
        if (dt == null)
            Response.Redirect("NoAccessPage.aspx");
        GridView1.DataSource = dt;
        GridView1.DataBind();
    }

    protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
    {
    }

    protected void GridView1_RowCancelingEdit(object sender,
GridViewCancelEventArgs e)
    {
        GridView1.EditIndex = -1;
        getUsers();
    }

    protected void GridView1_RowEditing(object sender, GridViewEditEventArgs e)
    {
        GridView1.EditIndex = e.NewEditIndex;
        getUsers();
    }

    protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs
e)
    {
        GridViewRow DeletedRow = GridView1.Rows[e.RowIndex];
        string cardNum = DeletedRow.Cells[2].Text;
    }
}

```

```
//Delete card
MyAdoHelper.DoQuery("MyDatabase.mdf", "DELETE FROM tblCard WHERE
cardNum=' " + cardNum + " '");

GridView1.EditIndex = -1;
getUsers();
}

protected void GridView1_PageIndexChanging(object sender,
GridViewEventArgs e)
{
    GridView1PageIndex = e.NewPageIndex;
    getUsers();
}

protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs
e)
{
    GridViewRow changedRow = GridView1.Rows[e.RowIndex];
    string cardNum = changedRow.Cells[2].Text;
    string company =
((TextBox)changedRow.Cells[3].FindControl("CompanyTextBox")).Text;
    string cvv =
((TextBox)changedRow.Cells[4].FindControl("CVVTextBox")).Text;
    string expireDate =
((TextBox)changedRow.Cells[5].FindControl("ExpireDateTextBox")).Text;

    DateTime expireDateNormalized = DateTime.Parse(expireDate);
    expireDateNormalized = new DateTime(expireDateNormalized.Year,
expireDateNormalized.Month, 1);

    expireDate = expireDateNormalized.ToString("yyyy-MM-dd");

    //Update card
    MyAdoHelper.DoQuery("MyDatabase.mdf", "UPDATE tblCard SET company=' " +
company + " ', CVV=' " + cvv + " ', expireDate=' " + expireDate + " ' WHERE
cardNum=' " + cardNum + " '");

    GridView1.EditIndex = -1;
    getUsers();
}

protected void GridView1_RowDataBound(object sender, GridViewRowEventArgs
e)
{
}
```

PastPurchasePage.aspx

מיקום: ComputersProjectNew/PastPurchasePage.aspx

מיקום קוד מאחור: ComputersProjectNew/PastPurchasePage.aspx.cs

קוד aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="PastPurchasePage.aspx.cs"
Inherits="PastPurchasePage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>Supply Management</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2" id="three">
        <div class="inner">
            <header class="align-center">
                <h2>Past Orders</h2>
                <br />
            </header>
        </div>
        <form id="form1" runat="server">
            <div class="align-center">
                <br />
                <br />
                <asp:GridView ID="OrderGridView" runat="server"
                    AutoGenerateColumns="False" HorizontalAlign="Center" AllowPaging="True"
                    OnSelectedIndexChanged="OrderGridView_SelectedIndexChanged"
                    OnPageIndexChanging="OrderGridView_PageIndexChanging" PageSize="3"
                    OnRowCommand="OrderGridView_RowCommand">
                    <PagerStyle HorizontalAlign="Center"/>
                    <Columns>
                        <asp:ButtonField ButtonType="Button" Text="Resend
                    Receipt" CommandName="ResendReceipt"/>
                        <asp:ButtonField ButtonType="Button" Text="View
                    Details" CommandName="Select" HeaderText="Details"/>
                        <asp:BoundField DataField="orderId" ReadOnly="true"
                    HeaderText="Order ID"/>
                        <asp:BoundField DataField="datetime" ReadOnly="true"
                    HeaderText="Order Date"/>
                        <asp:BoundField DataField="paid" ReadOnly="true"
                    HeaderText="Total Paid"/>
                    </Columns>
                </asp:GridView>
                <br />
                <div class="align-center">
                    <asp:Label ID="OrderDescLbl" runat="server"
Visible="false"></asp:Label>
                </div>
                <br />
                <asp:GridView ID="OrderProductGridView" runat="server"
                    AutoGenerateColumns="False" HorizontalAlign="Center" AllowPaging="True"
                    OnSelectedIndexChanged="OrderGridView_SelectedIndexChanged"
                    OnPageIndexChanging="OrderProductGridView_PageIndexChanging">
                    <PagerStyle HorizontalAlign="Center"/>
                </asp:GridView>
            </div>
        </form>
    </div>
</section>

```

```

        <Columns>
            <asp:BoundField DataField="productId" ReadOnly="true"
HeaderText="Product ID" ItemStyle-CssClass="hiddencol" HeaderStyle-
CssClass="hiddencol"/>
            <asp:BoundField DataField="productName" ReadOnly="true"
HeaderText="Product Name"/>
            <asp:BoundField DataField="size" ReadOnly="true"
HeaderText="Size"/>
            <asp:BoundField DataField="amount" ReadOnly="true"
HeaderText="Amount"/>
        </Columns>
    </asp:GridView>
</form>
</section>
</asp:Content>

```

:C# קוד

```

using System;
using System.Data;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class PastPurchasePage : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["username"] == null)// בודק אם המשתמש רשום
            Response.Redirect("NoAccessPage.aspx"); // מפנה לصفحة כניסה

        GetOrderGridviewData();
    }

    protected void GetOrderGridviewData()
    {
        DataTable orderDt = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf",
"SELECT * FROM tblOrder WHERE username='" + Session["username"].ToString() + "' AND paid>0");

        OrderGridView.DataSource = orderDt;
        OrderGridView.DataBind();
    }

    protected void OrderGridView_SelectedIndexChanged(object sender, EventArgs e)
    {
        GetOrderProductGridviewData();
    }

    protected void GetOrderProductGridviewData()
    {
        int orderId = int.Parse(OrderGridView.SelectedRow.Cells[2].Text);
    }
}

```

```

DataTable orderProductDt = MyAdoHelper.GetOrderProductTbl(orderId);
DataTable productDt = MyAdoHelper.GetProductTbl();

orderProductDt.Columns.Add("productName");

foreach (DataRow dataRow in orderProductDt.Rows)
{
    int drProductId = int.Parse(dataRow["productId"].ToString());

    dataRow["productName"] = productDt.Rows[drProductId -
1][1].ToString();
}

OrderDescLbl.Text = "<h3>Details of order #" + orderId + ":</h3>";
OrderDescLbl.Visible = true;

OrderProductGridView.DataSource = orderProductDt;
OrderProductGridView.DataBind();
}

protected void OrderGridView_PageIndexChanging(object sender,
GridViewEventArgs e)
{
    OrderGridViewPageIndex = e.NewPageIndex;
    GetOrderGridviewData();
}

protected void OrderProductGridView_PageIndexChanging(object sender,
GridViewEventArgs e)
{
    OrderProductGridViewPageIndex = e.NewPageIndex;
    GetOrderProductGridviewData();
}

protected void GetReceiptXML(int orderId, string username)
{
    DataTable orderTbl = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf",
"SELECT * FROM tblOrder WHERE orderId='" + orderId + "'");
    DataTable orderProductTbl =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblOrderProduct
WHERE orderId='" + orderId + "'");
    DataTable userInfoTbl = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf",
"SELECT * FROM userInfo WHERE username='" + username + "'");
    DataTable productTbl = MyAdoHelper.GetProductTbl();

    orderTbl.TableName = "BasicOrderInfo";
    orderProductTbl.TableName = "ProductDetails";
    productTbl.TableName = "ProductTable";

    string email = userInfoTbl.Rows[0]["email"].ToString();

    DataSet infoDs = new DataSet();

    infoDs.Tables.Add(orderTbl);
    infoDs.Tables.Add(orderProductTbl);
    infoDs.Tables.Add(productTbl);

    MyServiceReference.ShoeCalculatorServiceSoapClient client = new
MyServiceReference.ShoeCalculatorServiceSoapClient();

    client.SendReceiptEmail(infoDs, email, username);
}

```

```
protected void OrderGridView_RowCommand(object sender,
GridViewCommandEventArgs e)
{
    if (e.CommandName.Equals("ResendReceipt"))
    {
        int index = int.Parse(e.CommandArgument.ToString());

        GridViewRow selectedRow = OrderGridView.Rows[index];

        int orderId = int.Parse(selectedRow.Cells[2].Text);

        GetReceiptXML(orderId, Session["username"].ToString());
    }
}
```

ClimbingGroupPage.aspx

מיקום: ComputersProjectNew/ClimbingGroupPage.aspx

מיקום קוד מאחור: ComputersProjectNew/ClimbingGroupPage.aspx.cs

קוד aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="ClimbingGroupPage.aspx.cs"
Inherits="ClimbingGroupPage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>Climbing Group Page</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2" id="three">
        <div class="inner">
            <header class="align-center">
                <h2>Climbing Groups Table</h2>
                <br />
            </header>
        </div>
        <form id="form1" runat="server">
            <div class="align-center">
                <br />
                <asp:Label ID="ErrorLbl" runat="server"
Visible="false"></asp:Label>
                <br />
            </div>
            <br />
            <br />
            <br />
            <asp:GridView ID="GroupGridView" runat="server" Width="100%" 
AutoGenerateColumns="False" HorizontalAlign="Center" AllowPaging="True"
PageSize="3" OnSelectedIndexChanged="GroupGridView_SelectedIndexChanged"
OnPageIndexChanging="GroupGridView_PageIndexChanging"
OnRowCommand="GroupGridView_RowCommand">
                <PagerStyle HorizontalAlign="Center"/>
                <Columns>
                    <asp:TemplateField HeaderText="Group Actions">
                        <ItemTemplate>
                            <asp:Button ID="GroupActionButton"
runat="server" Text="Join" CommandName="Join" CommandArgument='<%#
((GridViewRow)Container).RowIndex %>'>
                        </ItemTemplate>
                    </asp:TemplateField>
                    <asp:ButtonField ButtonType="Button" Text="View
Meetings" CommandName="Select" HeaderText="Meeting Viewing"/>
                    <asp:ButtonField ButtonType="Button" Text="View
Members" CommandName="ViewMembers" HeaderText="Member Viewing"/>
                    <asp:BoundField DataField="groupId" ReadOnly="true"
HeaderText="Group ID"/>
                    <asp:BoundField DataField="groupName" ReadOnly="true"
HeaderText="Group Name"/>
                    <asp:BoundField DataField="coachName" ReadOnly="true"
HeaderText="Coach Name"/>
                    <asp:BoundField DataField="groupDesc" ReadOnly="true"
HeaderText="Description"/>
                </Columns>
            </asp:GridView>
        </form>
    </div>
</section>
```

```

        <asp:BoundField DataField="boulderGradeReq"
ReadOnly="true" HeaderText="Boulder Grade Requirement"/>
        <asp:BoundField DataField="wallGradeReq"
ReadOnly="true" HeaderText="Wall Grade Requirement"/>
        <asp:ImageField DataImageUrlField="groupImg"
DataImageUrlFormatString="~/groupImages/{0}" HeaderText="Group Image"
ControlStyle-Height="100" ControlStyle-Width="100" ControlStyle-
CssClass="gv"></asp:ImageField>
    </Columns>
</asp:GridView>
<br />
<div class="align-center">
    <asp:Label ID="MeetingsLbl" runat="server"
Visible="false"></asp:Label>
    </div>
    <br />
    <br />
    <asp:GridView ID="GroupMembersGridView" runat="server"
AutoGenerateColumns="false" HorizontalAlign="Center" AllowPaging="true"
PageSize="5" OnPageIndexChanging="GroupMembersGridView_PageIndexChanging">
        <PagerStyle HorizontalAlign="Center"/>
        <Columns>
            <asp:BoundField DataField="username"
HeaderText="Username" SortExpression="username"/>
            <asp:BoundField DataField="boulderGrade"
HeaderText="Bouldering Level" SortExpression="boulderGrade"/>
            <asp:BoundField DataField="wallGrade" HeaderText="Sport
Climbing Level" SortExpression="wallGrade"/>
        </Columns>
    </asp:GridView>
    <asp:GridView ID="MeetingsGridView" runat="server"
AutoGenerateColumns="False" HorizontalAlign="Center" AllowPaging="True"
OnPageIndexChanging="MeetingsGridView_PageIndexChanging"
OnRowCommand="MeetingsGridView_RowCommand">
        <PagerStyle HorizontalAlign="Center"/>
        <Columns>
            <asp:TemplateField HeaderText="Meeting Actions">
                <ItemTemplate>
                    <asp:Button ID="MeetingActionButton"
runat="server" Text="Join" CommandName="Join" CommandArgument='<%#
((GridViewRow)Container).RowIndex %>' />
                </ItemTemplate>
            </asp:TemplateField>
            <asp:ButtonField ButtonType="Button" Text="View
Members" CommandName="Members" HeaderText="View Members"/>
            <asp:ButtonField ButtonType="Button" Text="View
Weather" CommandName="Weather" HeaderText="View Weather"/>
            <asp:BoundField DataField="groupId" ReadOnly="true"
HeaderText="Group ID" ItemStyle-CssClass="hiddencol" HeaderStyle-
CssClass="hiddencol" FooterStyle-CssClass="hiddencol"/>
            <asp:BoundField DataField="meetingId" ReadOnly="true"
HeaderText="Meeting ID"/>
            <asp:BoundField DataField="city" ReadOnly="true"
HeaderText="City"/>
            <asp:BoundField DataField="country" ReadOnly="true"
HeaderText="Country"/>
            <asp:BoundField DataField="startTime"
ReadOnly="true" HeaderText="Starting DateTime"/>
            <asp:BoundField DataField="endTime" ReadOnly="true"
HeaderText="Ending DateTime"/>
            <asp:BoundField DataField="meetingDesc" ReadOnly="true"
HeaderText="Description"/>
        </Columns>
    </asp:GridView>

```

```

        </Columns>
    </asp:GridView>
    <br />
    <br />
    <asp:GridView ID="MeetingMembersGridView" runat="server"
AutoGenerateColumns="false" HorizontalAlign="Center" AllowPaging="true"
PageSize="5" OnPageIndexChanging="MeetingMembersGridView_PageIndexChanging">
        <PagerStyle HorizontalAlign="Center"/>
        <Columns>
            <asp:BoundField DataField="username"
HeaderText="Username" SortExpression="username"/>
            <asp:BoundField DataField="boulderGrade"
HeaderText="Bouldering Level" SortExpression="boulderGrade"/>
            <asp:BoundField DataField="wallGrade" HeaderText="Sport
Climbing Level" SortExpression="wallGrade"/>
        </Columns>
    </asp:GridView>
</form>
</section>
</asp:Content>

```

:C# טיפ

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;

public partial class ClimbingGroupPage : System.Web.UI.Page
{
    public const int CoachReq = 1;
    public int permLvl, selectedGroupId = -1, selectedMeetingId = -1;
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["username"] == null)//רשותם רשום
            RedirectTo("NoAccessPage.aspx"); //כניתה

        permLvl = int.Parse(Session["userPermLvl"].ToString()); //מקבל את הרשותת
        //הגישה של המשתמש מהesson

        if (!IsPostBack)
        {
            GetGroupData();
        }
    }

    protected void GetGroupData()
    {
        DataTable climbingGroupsTbl =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM
tblClimbingGroups");

```

```

        DataTable groupMembersTbl =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblGroupMembers
WHERE username='" + Session["username"].ToString() + "'");
        int groupId = -1;

        if(climbingGroupsTbl.Rows.Count < 1)
{
    ErrorLbl.Text = "<h3 class='red'>No groups found!</h3>";
    ErrorLbl.Visible = true;
    GroupGridView.Visible = false;
    MeetingsGridView.Visible = false;
    MeetingMembersGridView.Visible = false;
    GroupMembersGridView.Visible = false;
    return;
}

        if (permLvl >= CoachReq)
{
    if(climbingGroupsTbl.Select("coachName<>'"
Session["username"].ToString() + "'").Count() < 1)
    {
        ErrorLbl.Text = "<h3 class='red'>No groups found!</h3>";
        ErrorLbl.Visible = true;
        GroupGridView.Visible = false;
        MeetingsGridView.Visible = false;
        MeetingMembersGridView.Visible = false;
        GroupMembersGridView.Visible = false;
        return;
    }
    else
        climbingGroupsTbl = climbingGroupsTbl.Select("coachName<>'"
Session["username"].ToString() + "'").CopyToDataTable();
}
}

        if (groupMembersTbl.Rows.Count > 0)
{
    groupId = int.Parse(groupMembersTbl.Rows[0]["groupId"].ToString());
    DataRow memberRow = climbingGroupsTbl.Select("groupId=" +
groupId)[0];
    object[] itemArray = memberRow.ItemArray;
    climbingGroupsTbl.Rows.Remove(memberRow);
    DataRow newRow = climbingGroupsTbl.NewRow();
    newRow.ItemArray = itemArray;
    climbingGroupsTbl.Rows.InsertAt(newRow, 0);
}

        GroupGridView.DataSource = climbingGroupsTbl;
        GroupGridView.DataBind();
        GroupGridView.Visible = true;

        if(groupId != -1)
{
    Button joinButton =
GroupGridView.Rows[0].FindControl("GroupActionButton") as Button;
    joinButton.CommandName = "Leave";
    joinButton.Text = "Leave";
}

        ErrorLbl.Visible = false;
        MeetingsLbl.Visible = false;
        MeetingsGridView.Visible = false;
        GroupMembersGridView.Visible = false;

```

```

        MeetingMembersGridView.Visible = false;
    }

    protected void GetMeetingData()
    {
        int groupId = int.Parse(GroupGridView.SelectedRow.Cells[3].Text);

        DataTable groupMembersTbl =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblGroupMembers
WHERE groupId=" + groupId + " AND username='" + Session["username"].ToString()
+ "'");

        if (groupMembersTbl.Rows.Count < 1)
        {
            ErrorLbl.Text = "<h3 class='red'>You are not a member of this
group!</h3>";
            ErrorLbl.Visible = true;
            MeetingsLbl.Visible = false;
            MeetingsGridView.Visible = false;
            GroupMembersGridView.Visible = false;
            MeetingMembersGridView.Visible = false;
            return;
        }

        string groupName = GroupGridView.SelectedRow.Cells[4].Text;

        DataTable meetingsTbl = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf",
"SELECT * FROM tblGroupMeetings WHERE groupId=" + groupId + " AND
startDateTime>" + DateTime.Now.AddHours(1).ToString("yyyy-MM-dd HH:mm:ss.fff")
+ "'");

        DataTable meetingMembersTbl =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblMeetingMembers
WHERE username='" + Session["username"].ToString() + "'");

        meetingsTbl.DefaultView.Sort = "startDateTime";
        meetingsTbl = meetingsTbl.DefaultView.ToTable();

        if(meetingsTbl.Rows.Count < 1)
        {
            ErrorLbl.Text = "<h3 class='red'>No meetings for this group
yet!</h3>";
            ErrorLbl.Visible = true;
            MeetingsLbl.Visible = false;
            MeetingsGridView.Visible = false;
            MeetingMembersGridView.Visible = false;
            return;
        }

        MeetingsLbl.Text = "<h3>Meetings for group \"" + groupName +
"\":</h3>";
        MeetingsLbl.Visible = true;

        MeetingsGridView.DataSource = meetingsTbl;
        MeetingsGridView.DataBind();

        foreach(GridViewRow gridViewRow in MeetingsGridView.Rows)
        {
            int meetingId = int.Parse(gridViewRow.Cells[4].Text);
            Button joinButton = gridViewRow.FindControl("MeetingActionButton")
as Button;

```

```

        if(meetingMembersTbl.Select("meetingId=" + meetingId).Count() > 0)
        {
            joinButton.CommandName = "Leave";
            joinButton.Text = "Leave";
        }
    }

    MeetingsGridView.Visible = true;
    ErrorLbl.Visible = false;
    GroupMembersGridView.Visible = false;
    MeetingMembersGridView.Visible = false;
}

protected void GroupGridView_SelectedIndexChanged(object sender, EventArgs e)
{
    GetMeetingData();
}

protected void GroupGridView_PageIndexChanging(object sender, GridViewEventArgs e)
{
    GroupGridView.PageIndex = e.NewPageIndex;
    GetGroupData();
}

protected void GroupGridView_RowCommand(object sender, GridViewCommandEventArgs e)
{
    if(!(e.CommandName.ToLower().Equals("viewmembers") ||
e.CommandName.ToLower().Equals("join") ||
e.CommandName.ToLower().Equals("leave"))){
        return;
    }

    int index = int.Parse(e.CommandArgument.ToString()) %
GroupGridView.PageSize;
    int groupId = int.Parse(GroupGridView.Rows[index].Cells[3].Text);

    selectedGroupId = groupId;

    if (e.CommandName.ToLower().Equals("viewmembers"))
    {
        GetGroupMembersData();
    }
    if (e.CommandName.ToLower().Equals("join"))
    {
        DataTable groupMembersTbl =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblGroupMembers
WHERE username='" + Session["username"].ToString() + "'");

        if(groupMembersTbl.Rows.Count > 0)
        {
            ErrorLbl.Text = "<h3 class='red'>You may only be in one group
at a time!</h3>";
            ErrorLbl.Visible = true;
            return;
        }

        DataTable userInfoTbl =
MyAdoHelper.GetUserInfo(Session["username"].ToString());
    }
}

```

```

        int boulderGradeNum =
int.Parse(userInfoTbl.Rows[0]["boulderGrade"].ToString().Substring(1));
        string wallGrade = userInfoTbl.Rows[0]["wallGrade"].ToString();

        int boulderGradeReqNum =
int.Parse((GroupGridView.Rows[index].Cells[7].Text).Substring(1));
        string wallGradeReq = (GroupGridView.Rows[index].Cells[8].Text);

        if(boulderGradeNum < boulderGradeReqNum ||
wallGrade.CompareTo(wallGradeReq) == -1)
{
    ErrorLbl.Text = "<h3 class='red'>Sorry, you don't have the
required climbing grades for this group</h3>";
    ErrorLbl.Visible = true;
    return;
}

MyAdoHelper.DoQuery("MyDatabase.mdf", "INSERT INTO tblGroupMembers
VALUES (" + groupId + ", '" + Session["username"].ToString() + "')");

GroupGridView.PageIndex = 0;
GetGroupData();
}
if(e.CommandName.ToLower().Equals("leave"))
{
    MyAdoHelper.DoQuery("MyDatabase.mdf", "DELETE FROM tblGroupMembers
WHERE groupId=" + groupId + " AND username='" + Session["username"].ToString()
+ "'");

DataTable groupMeetingsTbl =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblGroupMeetings
WHERE groupId=" + groupId);

if(groupMeetingsTbl.Rows.Count > 0)
{
    string meetingIdList = "(" +
groupMeetingsTbl.Rows[0]["meetingId"].ToString();

    for (int i = 1; i < groupMeetingsTbl.Rows.Count; i++)
    {
        meetingIdList += ", " +
groupMeetingsTbl.Rows[i]["meetingId"].ToString();
    }

    meetingIdList += ")";
}

MyAdoHelper.DoQuery("MyDatabase.mdf", "DELETE FROM
tblMeetingMembers WHERE meetingId IN " + meetingIdList + " AND username='"
+ Session["username"].ToString() + "'");
}

GetGroupData();
}

protected void MeetingsGridView_RowCommand(object sender,
GridViewEventArgs e)
{
    if (!(e.CommandName.ToLower().Equals("viewmembers") ||
e.CommandName.ToLower().Equals("join") ||
e.CommandName.ToLower().Equals("leave") ||
e.CommandName.ToLower().Equals("weather")))
}

```

```

    {
        return;
    }
    int index = int.Parse(e.CommandArgument.ToString()) %
MeetingsGridView.PageSize;
    int groupId = int.Parse(MeetingsGridView.Rows[index].Cells[3].Text);
    int meetingId = int.Parse(MeetingsGridView.Rows[index].Cells[4].Text);

    selectedMeetingId = meetingId;

    if (e.CommandName.ToLower().Equals("viewmembers"))
    {
        GetMeetingMembersData();
    }
    if (e.CommandName.ToLower().Equals("join"))
    {
        MyAdoHelper.DoQuery("MyDatabase.mdf", "INSERT INTO
tblMeetingMembers VALUES (" + meetingId + ", '" +
Session["username"].ToString() + "')");

        GetMeetingData();
    }
    if (e.CommandName.ToLower().Equals("leave"))
    {
        MyAdoHelper.DoQuery("MyDatabase.mdf", "DELETE FROM
tblMeetingMembers WHERE meetingId=" + meetingId + " AND username=' " +
Session["username"].ToString() + "'");

        GetMeetingData();
    }
    if (e.CommandName.ToLower().Equals("weather"))
    {
        Session["city"] = MeetingsGridView.Rows[index].Cells[5].Text;
        Session["country"] = MeetingsGridView.Rows[index].Cells[6].Text;

        Response.Redirect("WeatherPage.aspx");
    }
}

protected void MeetingsGridView_PageIndexChanging(object sender,
GridViewEventArgs e)
{
    MeetingsGridViewPageIndex = e.NewPageIndex;
    GetMeetingData();
}

protected void GroupMembersGridView_PageIndexChanging(object sender,
GridViewEventArgs e)
{
    GroupMembersGridViewPageIndex = e.NewPageIndex;
    GetGroupMembersData();
}

protected void MeetingMembersGridView_PageIndexChanging(object sender,
GridViewEventArgs e)
{
    MeetingMembersGridViewPageIndex = e.NewPageIndex;
    GetMeetingMembersData();
}

protected void GetGroupMembersData()
{

```

```

        if(selectedGroupId == -1)
        {
            ErrorLbl.Text = "<h3 class='red'>There seems to have been an
error!</h3>";
            ErrorLbl.Visible = true;
            MeetingsLbl.Visible = false;
            GroupMembersGridView.Visible = false;
            MeetingMembersGridView.Visible = false;
            return;
        }

        DataTable groupMembersTbl =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblGroupMembers
WHERE groupId=" + selectedGroupId);
        DataTable groupTbl = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf",
"SELECT * FROM tblClimbingGroups WHERE groupId=" + selectedGroupId);

        string groupName = groupTbl.Rows[0][ "groupName"].ToString();

        if(groupMembersTbl.Rows.Count < 1)
        {
            ErrorLbl.Text = "<h3 class='red'>This group has no members!</h3>";
            ErrorLbl.Visible = true;
            MeetingsLbl.Visible = false;
            GroupMembersGridView.Visible = false;
            MeetingMembersGridView.Visible = false;
            return;
        }

        DataTable userInfoTbl = MyAdoHelper.GetUserInfoTbl();

        DataColumn boulderGradeColumn =
groupMembersTbl.Columns.Add("boulderGrade", typeof(string));
        DataColumn wallGradeColumn = groupMembersTbl.Columns.Add("wallGrade",
typeof(string));

        foreach(DataRow dataRow in groupMembersTbl.Rows)
        {
            string username = dataRow[ "username"].ToString();

            dataRow[ "boulderGrade"] = userInfoTbl.Select("username='"
+ username + "'").First()["boulderGrade"];
            dataRow[ "wallGrade"] = userInfoTbl.Select("username='"
+ username + "'").First()["wallGrade"];
        }

        GroupMembersGridView.DataSource = groupMembersTbl;
        GroupMembersGridView.DataBind();
        GroupMembersGridView.Visible = true;

        MeetingsLbl.Text = "<h3>Members of group \\" + groupName + "\\</h3>";
        MeetingsLbl.Visible = true;
        MeetingsGridView.Visible = false;
        MeetingMembersGridView.Visible = false;
        ErrorLbl.Visible = false;
    }

protected void GetMeetingMembersData()
{
    if (selectedMeetingId == -1)
    {

```

```
        ErrorLbl.Text = "<h3 class='red'>There seems to have been an  
error!</h3>";  
        ErrorLbl.Visible = true;  
        MeetingsLbl.Visible = false;  
        GroupMembersGridView.Visible = false;  
        MeetingMembersGridView.Visible = false;  
        return;  
    }  
  
    DataTable meetingMembersTbl =  
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblMeetingMembers  
WHERE groupId=" + selectedGroupId);  
  
    if (meetingMembersTbl.Rows.Count < 1)  
    {  
        ErrorLbl.Text = "<h3 class='red'>This meeting has no  
members!</h3>";  
        ErrorLbl.Visible = true;  
        return;  
    }  
  
    DataTable userInfoTbl = MyAdoHelper.GetUserInfoTbl();  
  
    DataColumn boulderGradeColumn =  
meetingMembersTbl.Columns.Add("boulderGrade", typeof(string));  
    DataColumn wallGradeColumn = meetingMembersTbl.Columns.Add("wallGrade",  
typeof(string));  
  
    foreach (DataRow dataRow in meetingMembersTbl.Rows)  
    {  
        string username = dataRow["username"].ToString();  
  
        dataRow["boulderGrade"] = userInfoTbl.Select("username=' " +  
username + "'").First()["boulderGrade"];  
        dataRow["wallGrade"] = userInfoTbl.Select("username=' " + username +  
"'").First()["wallGrade"];  
    }  
  
    MeetingMembersGridView.DataSource = meetingMembersTbl;  
    MeetingMembersGridView.DataBind();  
    MeetingMembersGridView.Visible = true;  
    ErrorLbl.Visible = false;  
}  
}
```

GroupCreationPage.aspx

מיקום: ComputersProjectNew/GroupCreationPage.aspx

מיקום קוד מאחור: ComputersProjectNew/GroupCreationPage.aspx.cs

קוד aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="GroupCreationPage.aspx.cs"
Inherits="GroupCreationPage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>Climbing Group Creation Page</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2">
        <div class="inner">
            <div class="box">
                <div class="content">
                    <header class="align-center">
                        <h2>New Group Form</h2>
                    </header>
                    <form id="form1" runat="server" method="post">
                        <div class="row uniform">
                            <div class="6u 12u$(small)">
                                <h3>Group Name:</h3>
                            </div>
                            <div class="6u$ 12u$(small)">
                                <asp:TextBox id="GroupNameTextBox"
runat="server"></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="GroupNameTextBox" Display="None"
ID="RequiredFieldValidator1" runat="server" ErrorMessage="Group name is
required"></asp:RequiredFieldValidator>
                                <asp:RegularExpressionValidator
ValidationExpression="^[a-zA-Z][a-zA-Z ]{0,48}[a-zA-Z]$"
ControlToValidate="GroupNameTextBox" Display="None"
ID="RegularExpressionValidator1" runat="server" ErrorMessage="Group name must
be up to 50 characters long!"></asp:RegularExpressionValidator>
                            </div>
                            <div class="6u 12u$(small)">
                                <h3 id="CoachNameHeader" runat="server" visible="false">
                                    <h3>Coach Name:</h3>
                                </div>
                                <div class="6u$ 12u$(small)">
                                    <asp:TextBox id="CoachNameTextBox"
runat="server" Visible="false"></asp:TextBox>
                                    <asp:RequiredFieldValidator
ControlToValidate="CoachNameTextBox" Display="None"
ID="CoachNameRequiredFieldValidator" runat="server" ErrorMessage="Coach name is
required" Enabled="false"></asp:RequiredFieldValidator>
                                    <asp:RegularExpressionValidator
ValidationExpression="^\w{5,50}$" ControlToValidate="CoachNameTextBox"
Display="None" ID="CoachNameRegularExpressionValidator" runat="server">
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </section>
    </div>
</asp:Content>

```

```

ErrorMessage="Coach name must be between 5 and 50 characters long"
Enabled="false">></asp:RegularExpressionValidator>
                                </div>

                <div class="6u 12u$(small)">
                    <h3>Group
Description:</h3>
                                </div>
                <div class="6u$ 12u$(small)">
                    <asp:TextBox id="DescTextBox"
runat="server"></asp:TextBox>
                    <asp:RequiredFieldValidator
ControlToValidate="DescTextBox" Display="None" ID="RequiredFieldValidator2"
runat="server" ErrorMessage="Description is
required"></asp:RequiredFieldValidator>
                    <asp:RegularExpressionValidator
ValidationExpression="^a-zA-Z0-9_.!-:[a-zA-Z0-9_.!-:]{0,298}[a-zA-Z0-
9_.!-:]$" ControlToValidate="DescTextBox" Display="None"
ID="RegularExpressionValidator2" runat="server" ErrorMessage="Description must
be up to 300 characters long, and not contain and special
characters!"></asp:RegularExpressionValidator>
                                </div>

                <div class="6u 12u$(small)">
                    <h3>Bouldering Grade
Requirement (V-Scale):</h3>
                                </div>
                <div class="6u$ 12u$(small)">
                    <asp:TextBox id="BoulderGradeTextBox"
runat="server"></asp:TextBox>
                    <asp:RequiredFieldValidator
ControlToValidate="BoulderGradeTextBox" Display="None"
ID="RequiredFieldValidator6" runat="server" ErrorMessage="Boulder grade is
required"></asp:RequiredFieldValidator>
                    <asp:RegularExpressionValidator
ValidationExpression="^V([0-9]|1[0-7])$"
ControlToValidate="BoulderGradeTextBox" Display="None"
ID="RegularExpressionValidator5" runat="server" ErrorMessage="Wrong boulder
grade syntax. Must be V0 to V17"></asp:RegularExpressionValidator>
                                </div>

                <div class="6u 12u$(small)">
                    <h3>Sport Climbing
Grade Requirement (French, capital letters):</h3>
                                </div>
                <div class="6u$ 12u$(small)">
                    <asp:TextBox id="WallGradeTextBox"
runat="server"></asp:TextBox>
                    <asp:RequiredFieldValidator
ControlToValidate="WallGradeTextBox" Display="None"
ID="RequiredFieldValidator7" runat="server" ErrorMessage="Wall grade is
required"></asp:RequiredFieldValidator>
                    <asp:RegularExpressionValidator
ValidationExpression="^5-9[A-C][+]{0,1}$"
ControlToValidate="WallGradeTextBox" Display="None"
ID="RegularExpressionValidator6" runat="server" ErrorMessage="Wrong wall grade
syntax. Must be 5A to 9C (+ is optional)"></asp:RegularExpressionValidator>
                                </div>

                <div class="6u 12u$(small)">
                    <h3>Group Image:</h3>
                </div>

```

```
<div class="6u$ 12u$(small)">
    <asp:FileUpload ID="ImageUploader"
runat="server" AllowMultiple="false"/>
        <asp:RequiredFieldValidator
ID="RequiredFieldValidator5" runat="server" ControlToValidate="ImageUploader"
Display="None" ErrorMessage="Group image is
required"></asp:RequiredFieldValidator>
        <asp:RegularExpressionValidator
ID="RegularExpressionValidator3" runat="server"
ControlToValidate="ImageUploader" ValidationExpression="^(([a-zA-
Z]:)|(\{\2\}\w+)\$?)(\w[\w].*)(.jpe?g|.JPE?G|.png|.PNG)$" ErrorMessage="You
may only upload .jpg and .png files"
Display="None"></asp:RegularExpressionValidator>
    </div>

    <div class="12u$"><asp:ValidationSummary
runat="server" ShowMessageBox="true" ShowSummary="false" /></div>

    <div class="12u$">
        <div class="align-center">
            <asp:Button runat="server"
ID="SubmitDataButton" OnClick="Submit_Click" Text="Create Group"/>
        </div>
    </div>

    <div class="12u$">
        <div class="align-center">
            <asp:Label ID="ResultLbl"
runat="server" Visible="false"></asp:Label>
        </div>
    </div>
    </form>
</div>
</div>
</section>
</asp:Content>
```

C# קורס

```

using System;
using System.Data;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class GroupCreationPage : System.Web.UI.Page
{
    int permLvl;
    public const int PermReq = 1, AdminReq = 3;
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["username"] == null)// בודק אם המשתמש נכנס האן לדף המשתמש את מעביר
            Response.Redirect("NoAccessPage.aspx"); // כנראה את מתקבל מהSESSION של המשתמש

        permLvl = int.Parse(Session["userPermLvl"].ToString()); // הרשות את מקובל מהSESSION של המשתמש

        if (permLvl < PermReq) // המשתמש הרשות את בודק
            Response.Redirect("NoAccessPage.aspx"); // כנראה האן לדף המשתמש את מעביר

        if(permLvl == AdminReq)
        {
            CoachNameHeader.Visible = true;
            CoachNameRegularExpressionValidator.Enabled = true;
            CoachNameRequiredFieldValidator.Enabled = true;
            CoachNameTextBox.Visible = true;
            CoachNameHeaderDiv.Visible = true;
            CoachNameTextBoxDiv.Visible = true;
        }
    }

    protected void Submit_Click(object sender, EventArgs e)
    {
        string groupName = Request.Form[GroupNameTextBox.UniqueID];
        string groupDesc = Request.Form[DescTextBox.UniqueID];
        string boulderGradeReq = Request.Form[BoulderGradeTextBox.UniqueID];
        string wallGradeReq = Request.Form[WallGradeTextBox.UniqueID];

        if (!MyAdoHelper.IsExist("MyDatabase.mdf", "SELECT * FROM
tblClimbingGroups WHERE groupName='" + groupName + "'"))
        {
            if(permLvl == AdminReq)
            {
                string coachName = Request.Form[CoachNameTextBox.UniqueID];

                if(!MyAdoHelper.IsExist("MyDatabase.mdf", "SELECT * FROM
userInfo WHERE username='" + coachName + "' AND permLvl>0"))
                {
                    ResultLbl.Text = "<h3 class='red'>This coach does not exist
or is not a coach!</h3>";
                    ResultLbl.Visible = true;
                    return;
                }
                else
                {

```

```

        MyAdoHelper.DoQuery("MyDatabase.mdf", "INSERT INTO
tblClimbingGroups VALUES ('" + coachName + "', '" + boulderGradeReq + "', '" +
wallGradeReq + "', '" + groupDesc + "', 'not_found.png', '" + groupName +
"')");
    }
}
else
{
    MyAdoHelper.DoQuery("MyDatabase.mdf", "INSERT INTO
tblClimbingGroups VALUES ('" + Session["username"].ToString() + "', '" +
boulderGradeReq + "', '" + wallGradeReq + "', '" + groupDesc + ',
'not_found.png', '" + groupName + "')");
}
else
{
    ResultLbl.Text = "<h3 class='red'>A group with this name already
exists!</h3>";
    ResultLbl.Visible = true;
    return;
}

DataTable climbingGroupsDt =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblClimbingGroups
WHERE groupName='" + groupName + "'");
int groupId =
int.Parse(climbingGroupsDt.Rows[0]["groupId"].ToString());

if (ImageUploader.HasFile)
{
    string ext = Path.GetExtension(ImageUploader.PostedFile.FileName);
    ImageUploader.SaveAs(Server.MapPath("~/groupImages/group" + groupId +
+ ext.ToLower()));
    MyAdoHelper.DoQuery("MyDatabase.mdf", "UPDATE tblClimbingGroups SET
groupImg='group" + groupId + ext.ToLower() + "' WHERE groupId=" + groupId);
}

ResultLbl.Text = "<h3 class='green'>Group successfully created!</h3>";
ResultLbl.Visible = true;
}
}

```

MeetingCreationPage.aspx

מיקום: ComputersProjectNew/MeetingCreationPage.aspx

מיקום קוד מאחור: ComputersProjectNew/MeetingCreationPage.aspx.cs

קוד aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="MeetingCreationPage.aspx.cs"
Inherits="MeetingCreationPage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>Meeting Creation Page</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2">
        <div class="inner">
            <div class="box">
                <div class="content">
                    <header class="align-center">
                        <h2>New Meeting Form</h2>
                    </header>
                    <form id="form1" runat="server" method="post">
                        <div class="row uniform">
                            <div class="6u 12u$(small)">
                                <h3>Group Name:</h3>
                                </div>
                            <div class="6u$ 12u$(small)">
                                <asp:TextBox id="GroupNameTextBox"
runat="server"></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="GroupNameTextBox" Display="None"
ID="RequiredFieldValidator1" runat="server" ErrorMessage="Group name is
required"></asp:RequiredFieldValidator>
                                <asp:RegularExpressionValidator
ValidationExpression="^[a-zA-Z][a-zA-Z ]{0,48}[a-zA-Z]$"
ControlToValidate="GroupNameTextBox" Display="None"
ID="RegularExpressionValidator1" runat="server" ErrorMessage="Group name must
be up to 50 characters long!"></asp:RegularExpressionValidator>
                            </div>
                            <div class="6u 12u$(small)">
                                <h3>Meeting
Description:</h3>
                                </div>
                            <div class="6u$ 12u$(small)">
                                <asp:TextBox id="DescTextBox"
runat="server"></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="DescTextBox" Display="None" ID="RequiredFieldValidator2"
runat="server" ErrorMessage="Description is
required"></asp:RequiredFieldValidator>
                                <asp:RegularExpressionValidator
ValidationExpression="^[a-zA-Z0-9_.!-:][a-zA-Z0-9 _.,!-:]{0,298}[a-zA-Z0-
9_.!-:]$" ControlToValidate="DescTextBox" Display="None"
ID="RegularExpressionValidator2" runat="server" ErrorMessage="Description must
be up to 300 characters long, and not contain and special
characters!"></asp:RegularExpressionValidator>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </section>
    </div>
</asp:Content>

```

```

        </div>

        <div class="6u 12u$(small)">
            <h3>City:</h3>
        </div>
        <div class="6u$ 12u$(small)">
            <asp:TextBox id="CityTextBox"
runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator
ControlToValidate="CityTextBox" Display="None" ID="RequiredFieldValidator3"
runat="server" ErrorMessage="City is required"></asp:RequiredFieldValidator>
            <asp:RegularExpressionValidator
ValidationExpression="^[a-zA-Z\d]+(?: [A-Za-z\d]{2,})*$"
ControlToValidate="CityTextBox" Display="None" ID="RegularExpressionValidator4"
runat="server" ErrorMessage="City name must be one or more
words"></asp:RegularExpressionValidator>
        </div>

        <div class="6u 12u$(small)">
            <h3>Country:</h3>
        </div>
        <div class="6u$ 12u$(small)">
            <asp:TextBox id="CountryTextBox"
runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator
ControlToValidate="CountryTextBox" Display="None" ID="RequiredFieldValidator4"
runat="server" ErrorMessage="Country is required"></asp:RequiredFieldValidator>
            <asp:RegularExpressionValidator
ValidationExpression="^[a-zA-Z\d]+(?: [A-Za-z\d]{2,})*$"
ControlToValidate="CountryTextBox" Display="None"
ID="RegularExpressionValidator5" runat="server" ErrorMessage="Country name must
be one or more words"></asp:RegularExpressionValidator>
        </div>

        <div class="6u 12u$(small)">
            <h3>Meeting
Start:</h3>
        </div>
        <div class="6u$ 12u$(small)">
            <asp:TextBox id="StartTimeDateTimeTextBox"
runat="server" TextMode="DateTimeLocal"></asp:TextBox>
            <asp:RequiredFieldValidator
ControlToValidate="StartTimeDateTimeTextBox" Display="None"
ID="RequiredFieldValidator5" runat="server" ErrorMessage="Meeting start is
required"></asp:RequiredFieldValidator>
        </div>

        <div class="6u 12u$(small)">
            <h3>Meeting
Ending:</h3>
        </div>
        <div class="6u$ 12u$(small)">
            <asp:TextBox id="EndTimeDateTimeTextBox"
runat="server" TextMode="DateTimeLocal"></asp:TextBox>
            <asp:RequiredFieldValidator
ControlToValidate="EndTimeDateTimeTextBox" Display="None"
ID="RequiredFieldValidator6" runat="server" ErrorMessage="Meeting end is
required"></asp:RequiredFieldValidator>
        </div>

        <div class="12u$"><asp:ValidationSummary
runat="server" ShowMessageBox="true" ShowSummary="false" /></div>

```

```

        <div class="12u$">
            <div class="align-center">
                <asp:Button runat="server"
ID="SubmitDataButton" OnClick="Submit_Click" Text="Create Meeting"/>
                </div>
            </div>

        <div class="12u$">
            <div class="align-center">
                <asp:Label ID="ResultLbl"
runat="server" Visible="false"></asp:Label>
                </div>
            </div>
        </form>
    </div>
</div>
</section>
</asp:Content>

```

C# קוד

```

using System;
using System.Data;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.Web.Script.Serialization;
using Newtonsoft.Json;
using System.Globalization;
using Nager;
using Nager.Country;
using Nager.Country.CountryInfos;

public partial class MeetingCreationPage : System.Web.UI.Page
{
    public const int PermReq = 1, AdminReq = 3;
    public const string weatherApiKey = "43df790eef2340a0e861fed4cba26f37";
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["username"] == null)// בודק אם המשתמש נכנס האין לדף המשתמש את מעביר
            Response.Redirect("NoAccessPage.aspx"); //הגישה הרשות את מקבל מהשנץ המשתמש של הגישה הרשות

        int permLvl = int.Parse(Session["userPermLvl"].ToString()); //המשתמש של הגישה הרשות את בודק
        if (permLvl < PermReq) //המשתמש של הגישה הרשות את בודק
            Response.Redirect("NoAccessPage.aspx"); //הגישה האין לדף המשתמש את מעביר
    }

    protected void Submit_Click(object sender, EventArgs e)
    {
        string groupName = Request.Form[GroupNameTextBox.UniqueID];
        string meetingDesc = Request.Form[DescTextBox.UniqueID];

```

```

        string city =
System.Threading.Thread.CurrentThread.CurrentCulture.TextInfo.ToTitleCase(Request.Form[CityTextBox.UniqueID].ToLower());
        string country =
System.Threading.Thread.CurrentThread.CurrentCulture.TextInfo.ToTitleCase(Request.Form[CountryTextBox.UniqueID].ToLower());
        DateTime startTime =
DateTime.Parse(Request.Form[StartTimeTextBox.UniqueID]);
        DateTime endTime =
DateTime.Parse(Request.Form[EndTimeTextBox.UniqueID]);

        if (startTime < DateTime.Now || endTime < DateTime.Now ||
startTime > DateTime.Now.AddDays(30) || endTime >
DateTime.Now.AddDays(30))
{
    ResultLbl.Text = "<h3 class='red'>The starting and ending datetimes
have to be within 30 days of now!</h3>";
    ResultLbl.Visible = true;
    return;
}

if(startTime > endTime.AddHours(-1))
{
    ResultLbl.Text = "<h3 class='red'>The starting datetime must be
before the ending datetime, and the meeting must be at least one hour
long!</h3>";
    ResultLbl.Visible = true;
    return;
}

DataTable climbingGroupDt =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblClimbingGroups
WHERE groupName=' " + groupName + "'");

if(climbingGroupDt.Rows.Count < 1)
{
    ResultLbl.Text = "<h3 class='red'>There is no such group! Please be
accurate when entering the group name!</h3>";
    ResultLbl.Visible = true;
    return;
}

int permLvl = int.Parse(Session["userPermLvl"].ToString());

string coachName = climbingGroupDt.Rows[0]["coachName"].ToString();

if (!coachName.Equals(Session["username"].ToString()) && permLvl <
AdminReq)
{
    ResultLbl.Text = "<h3 class='red'>You are not the coach of this
group!</h3>";
    ResultLbl.Visible = true;
    return;
}

DataTable coachGroupsDt =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblClimbingGroups
WHERE coachName=' " + coachName + "'");

string groupIdList = "(" + coachGroupsDt.Rows[0]["groupId"].ToString();

for(int i = 1; i < coachGroupsDt.Rows.Count; i++)

```

```

{
    groupIdList += ", " + coachGroupsDt.Rows[i]["groupId"].ToString();
}

groupIdList += ")";

DataTable groupMeetingsDt =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblGroupMeetings
WHERE groupId IN " + groupIdList);

if(groupMeetingsDt.Rows.Count > 0)
{
    foreach(DataRow dataRow in groupMeetingsDt.Rows)
    {
        if(IsDateTimeBetween((DateTime)dataRow["startDateTime"],
(DateTime)dataRow["endDateTime"], startTime) ||
IsDateTimeBetween((DateTime)dataRow["startDateTime"],
(DateTime)dataRow["endDateTime"], endTime))
        {
            if(!coachName.Equals(Session["username"].ToString()))
                ResultLbl.Text = "<h3 class='red'>The coach for this
group already has a meeting during this time period!</h3>";
            else
                ResultLbl.Text = "<h3 class='red'>You already have a
meeting during this time period!</h3>";
            ResultLbl.Visible = true;
            return;
        }
    }
}

using (WebClient client = new WebClient())
{
    try
    {
        ICountryProvider countryProvider = new CountryProvider();
        var countryInfo = countryProvider.GetCountryByName(country);

        string query = city + ",0," +
countryInfo.Alpha2Code.ToString();

        string url =
string.Format("http://api.openweathermap.org/geo/1.0/direct?q={0}&appid={1}",
query, weatherApiKey);

        string json = client.DownloadString(url);
        GeoRoot geoRoot =
JsonConvert.DeserializeObject<GeoRoot>(json.Substring(1, json.Length - 2));
    }
    catch
    {
        ResultLbl.Text = "<h3 class='red'>You entered a bad
city/country name!</h3>";
        ResultLbl.Visible = true;
        return;
    }
}

MyAdoHelper.DoQuery("MyDatabase.mdf", "INSERT INTO tblGroupMeetings
(groupId, city, country, startTime, endTime, meetingDesc) VALUES (" +
climbingGroupDt.Rows[0]["groupId"] + ", '" + city + "', '" + country + "', '" +

```

```
startDateTime.ToString("yyyy-MM-dd HH:mm:ss.fff") + ",'" +  
endDateTime.ToString("yyyy-MM-dd HH:mm:ss.fff") + ",'" + meetingDesc + "')");  
  
        ResultLbl.Text = "<h3 class='green'>Meeting successfully  
created!</h3>";  
        ResultLbl.Visible = true;  
    }  
  
    protected bool IsDateTimeBetween(DateTime min, DateTime max, DateTime  
value)  
    {  
        if (value >= min && value <= max)  
            return true;  
        return false;  
    }  
}
```

GroupManagementPage.aspx

מיקום: ComputersProjectNew/GroupManagementPage.aspx

מיקום קוד מאחור: ComputersProjectNew/GroupManagementPage.aspx.cs

קוד aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="GroupManagementPage.aspx.cs"
Inherits="GroupManagementPage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>Climbing Group Management Page</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2" id="three">
        <div class="inner">
            <header class="align-center">
                <h2>Group Management Table</h2>
                <br />
            </header>
        </div>
        <form id="form1" runat="server">
            <div class="align-center">
                <asp:Button ID="NewGroupButton" runat="server" Text="Add
New Group" OnClick="NewGroupButton_Click"/>
                <br />
                <br />
                <asp:Label ID="ErrorLbl" runat="server"
Visible="false"></asp:Label>
                <br />
            </div>
            <br />
            <br />
            <br />
            <asp:GridView ID="GroupGridView" runat="server" Width="100%" 
AutoGenerateColumns="False" HorizontalAlign="Center" AllowPaging="True"
PageSize="3" OnRowEditing="GroupGridView_RowEditing"
OnRowCancelingEdit="GroupGridView_RowCancelingEdit"
OnSelectedIndexChanged="GroupGridView_SelectedIndexChanged"
OnRowUpdating="GroupGridView_RowUpdating"
OnPageIndexChanging="GroupGridView_PageIndexChanging"
OnRowCommand="GroupGridView_RowCommand"
OnRowDeleting="GroupGridView_RowDeleting">
                <PagerStyle HorizontalAlign="Center"/>
                <Columns>
                    <asp:CommandField ButtonType="Button"
ShowEditButton="true" HeaderText="Editing" CancelText="Cancel Edit"
EditText="Edit" UpdateText="Confirm Edit" ItemStyle-Width="300"/>
                    <asp:ButtonField ButtonType="Button" Text="Delete"
CommandName="Delete" HeaderText="Deleting"/>
                    <asp:ButtonField ButtonType="Button" Text="View
Meetings" CommandName="Select" HeaderText="Meeting Viewing"/>
                    <asp:BoundField DataField="groupId" ReadOnly="true"
HeaderText="Group ID"/>
                    <asp:TemplateField HeaderText="Group Name">
                        <EditItemTemplate>
                            <asp:TextBox ID="GroupNameTextBox"
runat="server" Text='<%# Eval("groupName") %>'></asp:TextBox>
                        </EditItemTemplate>
                    </asp:TemplateField>
                </Columns>
            </asp:GridView>
        </form>
    </div>
</section>

```

```

        <asp:RequiredFieldValidator
ControlToValidate="GroupNameTextBox" Display="None"
ID="RequiredFieldValidator1" runat="server" ErrorMessage="Group name is
required"></asp:RequiredFieldValidator>
        <asp:RegularExpressionValidator
ValidationExpression="^a-zA-Z[a-zA-Z]{0,48}a-zA-Z$"
ControlToValidate="GroupNameTextBox" Display="None"
ID="RegularExpressionValidator1" runat="server" ErrorMessage="Group name must
be up to 50 characters long!"></asp:RegularExpressionValidator>
        </EditItemTemplate>
        <ItemTemplate>
            <asp:Label ID="GroupNameLbl" runat="server"
Text='<%# Bind("groupName") %>'></asp:Label>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Coach Name">
            <EditItemTemplate>
                <asp:TextBox id="CoachNameTextBox"
runat="server" Text='<%# Eval("coachName") %>'></asp:TextBox>
                <asp:RequiredFieldValidator
ControlToValidate="CoachNameTextBox" Display="None"
ID="CoachNameRequiredFieldValidator" runat="server" ErrorMessage="Coach name is
required"></asp:RequiredFieldValidator>
                <asp:RegularExpressionValidator
ValidationExpression="^\w{5,50}$" ControlToValidate="CoachNameTextBox"
Display="None" ID="CoachNameRegularExpressionValidator" runat="server"
ErrorMessage="Coach name must be between 5 and 50 characters
long"></asp:RegularExpressionValidator>
            </EditItemTemplate>
            <ItemTemplate>
                <asp:Label ID="CoachNameLbl" runat="server"
Text='<%# Bind("coachName") %>'></asp:Label>
                </ItemTemplate>
            </asp:TemplateField>
            <asp:TemplateField HeaderText="Group Description">
                <EditItemTemplate>
                    <asp:TextBox id="DescTextBox" runat="server"
Text='<%# Eval("groupDesc") %>'></asp:TextBox>
                    <asp:RequiredFieldValidator
ControlToValidate="DescTextBox" Display="None" ID="RequiredFieldValidator2"
runat="server" ErrorMessage="Description is
required"></asp:RequiredFieldValidator>
                    <asp:RegularExpressionValidator
ValidationExpression="^a-zA-Z0-9_.!-:[a-zA-Z0-9_.!-:]{0,298}[a-zA-Z0-
9_.!-:]$" ControlToValidate="DescTextBox" Display="None"
ID="RegularExpressionValidator2" runat="server" ErrorMessage="Description must
be up to 300 characters long, and not contain and special
characters!"></asp:RegularExpressionValidator>
                </EditItemTemplate>
                <ItemTemplate>
                    <asp:Label ID="DescLbl" runat="server"
Text='<%# Bind("groupDesc") %>'></asp:Label>
                    </ItemTemplate>
                </asp:TemplateField>
                <asp:TemplateField HeaderText="Boulder Grade
Requirement">
                    <EditItemTemplate>
                        <asp:TextBox id="BoulderGradeTextBox"
runat="server" Text='<%# Eval("boulderGradeReq") %>'></asp:TextBox>
                        <asp:RequiredFieldValidator
ControlToValidate="BoulderGradeTextBox" Display="None">

```

```

ID="RequiredFieldValidator6" runat="server" ErrorMessage="Boulder grade is
required">></asp:RequiredFieldValidator>
    <asp:RegularExpressionValidator
ValidationExpression="^([V]([0-9]|1[0-7]))$"
ControlToValidate="BoulderGradeTextBox" Display="None"
ID="RegularExpressionValidator5" runat="server" ErrorMessage="Wrong boulder
grade syntax. Must be V0 to V17">></asp:RegularExpressionValidator>
    </EditItemTemplate>
    <ItemTemplate>
        <asp:Label ID="BoulderGradeLbl" runat="server"
Text='<%# Bind("boulderGradeReq") %>'></asp:Label>
        </ItemTemplate>
    </asp:TemplateField>
    <asp:TemplateField HeaderText="Wall Grade Requirement">
        <EditItemTemplate>
            <asp:TextBox id="WallGradeTextBox"
runat="server" Text='<%# Eval("wallGradeReq") %>'></asp:TextBox>
            <asp:RequiredFieldValidator
ControlToValidate="WallGradeTextBox" Display="None"
ID="RequiredFieldValidator7" runat="server" ErrorMessage="Wall grade is
required">></asp:RequiredFieldValidator>
            <asp:RegularExpressionValidator
ValidationExpression="^([5-9][A-C][+]{0,1})$"
ControlToValidate="WallGradeTextBox" Display="None"
ID="RegularExpressionValidator6" runat="server" ErrorMessage="Wrong wall grade
syntax. Must be 5A to 9C (+ is optional)">></asp:RegularExpressionValidator>
            </EditItemTemplate>
            <ItemTemplate>
                <asp:Label ID="WallGradeLbl" runat="server"
Text='<%# Bind("wallGradeReq") %>'></asp:Label>
                </ItemTemplate>
            </asp:TemplateField>
            <asp:TemplateField HeaderText="Image">
                <EditItemTemplate>
                    <asp:FileUpload ID="ImageUploader"
runat="server" AllowMultiple="false"/>
                    <asp:RegularExpressionValidator
ID="RegularExpressionValidator3" runat="server"
ControlToValidate="ImageUploader" ValidationExpression="^(([a-zA-
Z:]|(\w{2}\w+)$?)(\w[\w.]*)(.jpg|.JPG|.png|.PNG|.jpeg|.JPEG)$"
ErrorMessage="You may only upload .jpg and .png files"
Display="None">></asp:RegularExpressionValidator>
                    <asp:Button CssClass="button small"
ID="UploadButton" runat="server" Text="Upload" CommandName="Upload"/>
                </EditItemTemplate>
                <ItemTemplate>
                    <asp:Image ID="ImageBox" runat="server"
ImageUrl='<%# "~/groupImages/" + Eval("groupImg") %>' Height="100" Width="100"
CssClass="gv"/>
                </ItemTemplate>
            </asp:TemplateField>
        </Columns>
    </asp:GridView>
    <br />
    <div class="align-center">
        <asp:Label ID="MeetingsLbl" runat="server"
Visible="false">></asp:Label>
    </div>
    <br />
    <div class="align-center">
        <asp:Button ID="NewMeetingButton" runat="server" Text="Add
New Meeting" OnClick="NewMeetingButton_Click"/>
    </div>

```

```

        </div>
        <br />
        <asp:GridView ID="MeetingsGridView" runat="server"
AutoGenerateColumns="False" HorizontalAlign="Center" AllowPaging="True"
OnRowDeleting="MeetingsGridView_RowDeleting"
OnRowCancelingEdit="MeetingsGridView_RowCancelingEdit"
OnRowEditing="MeetingsGridView_RowEditing"
OnRowUpdating="MeetingsGridView_RowUpdating"
OnPageIndexChanging="MeetingsGridView_PageIndexChanging"
OnRowCommand="MeetingsGridView_RowCommand">
            <PagerStyle HorizontalAlign="Center"/>
            <Columns>
                <asp:CommandField ButtonType="Button"
ShowEditButton="true" HeaderText="Editing" CancelText="Cancel Edit"
EditText="Edit" UpdateText="Confirm Edit" ItemStyle-Width="300"/>
                    <asp:ButtonField ButtonType="Button" Text="Delete"
CommandName="Delete" HeaderText="Deleting"/>
                    <asp:BoundField DataField="groupId" ReadOnly="true"
HeaderText="Group ID" ItemStyle-CssClass="hiddencol" HeaderStyle-
CssClass="hiddencol" FooterStyle-CssClass="hiddencol"/>
                    <asp:BoundField DataField="meetingId" ReadOnly="true"
HeaderText="Meeting ID"/>
                    <asp:TemplateField HeaderText="City">
                        <EditItemTemplate>
                            <asp:TextBox id="CityTextBox" runat="server"
Text='<%# Eval("city") %>'></asp:TextBox>
                            <asp:RequiredFieldValidator
ControlToValidate="CityTextBox" Display="None" ID="RequiredFieldValidator3"
runat="server" ErrorMessage="City is required"></asp:RequiredFieldValidator>
                            <asp:RegularExpressionValidator
ValidationExpression="^a-zA-Z\d+(: [A-Za-z\d]{2,})*$"
ControlToValidate="CityTextBox" Display="None" ID="RegularExpressionValidator4"
runat="server" ErrorMessage="City name must be one or more
words"></asp:RegularExpressionValidator>
                        </EditItemTemplate>
                        <ItemTemplate>
                            <asp:Label ID="CityLbl" runat="server"
Text='<%# Bind("city") %>'></asp:Label>
                        </ItemTemplate>
                    </asp:TemplateField>
                    <asp:TemplateField HeaderText="Country">
                        <EditItemTemplate>
                            <asp:TextBox id="CountryTextBox" runat="server"
Text='<%# Eval("country") %>'></asp:TextBox>
                            <asp:RequiredFieldValidator
ControlToValidate="CountryTextBox" Display="None" ID="RequiredFieldValidator4"
runat="server" ErrorMessage="Country is required"></asp:RequiredFieldValidator>
                            <asp:RegularExpressionValidator
ValidationExpression="^a-zA-Z\d+(: [A-Za-z\d]{2,})*$"
ControlToValidate="CountryTextBox" Display="None"
ID="RegularExpressionValidator5" runat="server" ErrorMessage="Country name must
be one or more words"></asp:RegularExpressionValidator>
                        </EditItemTemplate>
                        <ItemTemplate>
                            <asp:Label ID="CountryLbl" runat="server"
Text='<%# Bind("country") %>'></asp:Label>
                        </ItemTemplate>
                    </asp:TemplateField>
                    <asp:TemplateField HeaderText="Starting Date-Time">
                        <EditItemTemplate>

```

```

        <asp:TextBox id="StartTimeTextBox"
runat="server" TextMode="DateTimeLocal" Text='<%#
((DateTime)Eval("startDateTime")).ToString("yyyy-MM-ddTHH:mm")
%>'></asp:TextBox>
        <asp:RequiredFieldValidator
ControlToValidate="StartTimeTextBox" Display="None"
ID="RequiredFieldValidator5" runat="server" ErrorMessage="Meeting start is
required"></asp:RequiredFieldValidator>
        </EditItemTemplate>
        <ItemTemplate>
            <asp:Label ID="StartTimeGradeLbl"
runat="server" Text='<%# Bind("startDateTime") %>'></asp:Label>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Ending Date-Time">
            <EditItemTemplate>
                <asp:TextBox id="EndTimeTextBox"
runat="server" TextMode="DateTimeLocal" Text='<%#
((DateTime)Eval("endDateTime")).ToString("yyyy-MM-ddTHH:mm") %>'></asp:TextBox>
                <asp:RequiredFieldValidator
ControlToValidate="EndTimeTextBox" Display="None"
ID="RequiredFieldValidator6" runat="server" ErrorMessage="Meeting end is
required"></asp:RequiredFieldValidator>
            </EditItemTemplate>
            <ItemTemplate>
                <asp:Label ID="EndTimeLbl" runat="server"
Text='<%# Bind("endDateTime") %>'></asp:Label>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Meeting Description">
            <EditItemTemplate>
                <asp:TextBox id="DescTextBox" runat="server"
Text='<%# Eval("meetingDesc") %>'></asp:TextBox>
                <asp:RequiredFieldValidator
ControlToValidate="DescTextBox" Display="None" ID="RequiredFieldValidator2"
runat="server" ErrorMessage="Description is
required"></asp:RequiredFieldValidator>
                <asp:RegularExpressionValidator
ValidationExpression="^([a-zA-Z0-9_,!-:][a-zA-Z0-9 _,!-:]{0,298}[a-zA-Z0-
9_,!-:]*$" ControlToValidate="DescTextBox" Display="None"
ID="RegularExpressionValidator2" runat="server" ErrorMessage="Description must
be up to 300 characters long, and not contain and special
characters!"></asp:RegularExpressionValidator>
            </EditItemTemplate>
            <ItemTemplate>
                <asp:Label ID="DescLbl" runat="server"
Text='<%# Bind("meetingDesc") %>'></asp:Label>
            </ItemTemplate>
        </asp:TemplateField>
    </Columns>
</asp:GridView>
<asp:ValidationSummary ID="ValidationSummary1" runat="server"
ShowMessageBox="true" ShowSummary="false"/>
</form>
</section>
</asp:Content>

```

C# קורס

```

using System;
using System.Data;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.Web.Script.Serialization;
using Newtonsoft.Json;
using System.Globalization;
using Nager;
using Nager.Country;
using Nager.Country.CountryInfos;
using System.IO;

public partial class GroupManagementPage : System.Web.UI.Page
{
    public const string weatherApiKey = "43df790eef2340a0e861fed4cba26f37";
    public const int AdminReq = 3, PermReq = 1;
    public int permLvl;
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["username"] == null) // בודק אם המשתמש נכנס האן לדף המשתמש את מעבירו
            Response.Redirect("NoAccessPage.aspx"); // כנראה הוא לא מילא את הדרישות

        permLvl = int.Parse(Session["userPermLvl"].ToString()); // מתקבלת מהSESSION
        // הרשות את מקובל

        if (permLvl < PermReq) // בודק אם המשתמש נכנס האן לדף המשתמש את מעבירו
            Response.Redirect("NoAccessPage.aspx"); // כנראה הוא לא מילא את הדרישות

        if (!IsPostBack)
        {
            GetGroupData();
        }
    }

    protected void GetGroupData()
    {
        DataTable climbingGroupsTbl =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM
tblClimbingGroups");

        if (permLvl < AdminReq)
            climbingGroupsTbl = climbingGroupsTbl.Select("coachName=" + Session["username"].ToString() + "").CopyToDataTable();

        GroupGridView.DataSource = climbingGroupsTbl;
        GroupGridView.DataBind();

        ErrorLbl.Visible = false;
    }

    protected void GetMeetingData()
    {
        int groupId = int.Parse(GroupGridView.SelectedRow.Cells[3].Text);
        string groupName =
((Label)GroupGridView.SelectedRow.FindControl("GroupNameLbl")).Text;
    }
}

```

```

        DataTable meetingsTbl = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf",
"SELECT * FROM tblGroupMeetings");

        if(meetingsTbl.Select("groupId=" + groupId).Count() < 1)
        {
            ErrorLbl.Text = "<h3 class='red'>No meetings have been created for
group \\" + groupName + "\\ yet!</h3>";
            ErrorLbl.Visible = true;
            MeetingsGridView.Visible = false;
            MeetingsLbl.Visible = false;
            return;
        }

        meetingsTbl = meetingsTbl.Select("groupId=" +
groupId).CopyToDataTable();

        MeetingsLbl.Text = "<h3>Meetings Table For Group \\'" + groupName +
"\\"</h3>";
        MeetingsLbl.Visible = true;

        ErrorLbl.Visible = false;

        MeetingsGridView.Visible = true;
        MeetingsGridView.DataSource = meetingsTbl;
        MeetingsGridView.DataBind();
    }

    protected void NewGroupButton_Click(object sender, EventArgs e)
    {
        Response.Redirect("GroupCreationPage.aspx");
    }

    protected void GroupGridView_RowEditing(object sender,
GridViewEventArgs e)
    {
        GroupGridView.EditIndex = e.NewEditIndex;
        GetGroupData();

        if (permLvl < AdminReq)
        {
            TextBox coachNameTextBox =
(TextBox)GroupGridView.Rows[e.NewEditIndex].FindControl("CoachNameTextBox");
            coachNameTextBox.ReadOnly = true;
        }
    }

    protected void GroupGridView_RowCancelingEdit(object sender,
GridViewCancelEventArgs e)
    {
        GroupGridView.EditIndex = -1;
        GetGroupData();
    }

    protected void GroupGridView_SelectedIndexChanged(object sender, EventArgs
e)
    {
        GetMeetingData();
    }

    protected void GroupGridView_RowUpdating(object sender,
GridViewUpdateEventArgs e)

```

```

{
    GridViewRow changedRow = GroupGridView.Rows[e.RowIndex];

    int groupId = int.Parse(changedRow.Cells[3].Text);
    string groupName =
((TextBox)changedRow.FindControl("GroupNameTextBox")).Text;
    string coachName =
((TextBox)changedRow.FindControl("CoachNameTextBox")).Text;
    string desc = ((TextBox)changedRow.FindControl("DescTextBox")).Text;
    string boulderGradeReq =
((TextBox)changedRow.FindControl("BoulderGradeTextBox")).Text;
    string wallGradeReq =
((TextBox)changedRow.FindControl("WallGradeTextBox")).Text;

    if (MyAdoHelper.IsExist("MyDatabase.mdf", "SELECT * FROM
tblClimbingGroups WHERE groupName=' " + groupName + "' AND groupId!=" +
groupId))
    {
        ErrorLbl.Text = "<h3 class='red'>A group with this name already
exists!</h3>";
        ErrorLbl.Visible = true;
        return;
    }

    if (!MyAdoHelper.IsExist("MyDatabase.mdf", "SELECT * FROM userInfo
WHERE username=' " + coachName + "' AND permLvl>0"))
    {
        ErrorLbl.Text = "<h3 class='red'>This coach does not exist or is
not a coach!</h3>";
        ErrorLbl.Visible = true;
        return;
    }

    MyAdoHelper.DoQuery("MyDatabase.mdf", "UPDATE tblClimbingGroups SET
groupName=' " + groupName + "' , coachName=' " + coachName + "' ,
boulderGradeReq=' " + boulderGradeReq + "' , wallGradeReq=' " + wallGradeReq + "' ,
groupDesc=' " + desc + "' WHERE groupId=" + groupId);

    GroupGridView.EditIndex = -1;
    GetGroupData();
}

protected void GroupGridView_PageIndexChanged(object sender,
GridViewEventArgs e)
{
    GroupGridViewPageIndex = e.NewPageIndex;
    GetGroupData();
}

protected void GroupGridView_RowCommand(object sender,
GridViewCommandEventArgs e)
{
    Button button = e.CommandSource as Button;
    if (e.CommandName.ToLower().Equals("upload"))
    {
        FileUpload fileUpload =
(FileUpload)button.FindControl("ImageUploader");

        if (fileUpload.HasFile)
        {
            UploadFile(fileUpload);
        }
    }
}

```

```

        }

    protected void UploadFile(FileUpload fileUpload)
    {
        if (GroupGridView.EditIndex == -1)
            return;

        GridViewRow changedRow = GroupGridView.Rows[GroupGridView.EditIndex];

        int groupId = int.Parse(changedRow.Cells[3].Text);

        DataTable groupTbl = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf",
"SELECT * FROM tblClimbingGroups");
        DataRow groupRow = groupTbl.Select("groupId=" + groupId)[0];

        if (groupRow["groupImg"].ToString() != "not_found.png" &&
File.Exists(Server.MapPath("~/groupImages/" +
groupRow["groupImg"].ToString())))
        {
            File.Delete(Server.MapPath("~/groupImages/" +
groupRow["groupImg"].ToString()));
        }
        string ext = Path.GetExtension(fileUpload.PostedFile.FileName);
        fileUpload.SaveAs(Server.MapPath("~/groupImages/group" + groupId +
ext.ToLower()));
        MyAdoHelper.DoQuery("MyDatabase.mdf", "UPDATE tblClimbingGroups SET
groupImg='group" + groupId + ext.ToLower() + "' WHERE groupId=" + groupId);

        GroupGridView.EditIndex = -1;
        GetGroupData();
    }

    protected void GroupGridView_RowDeleting(object sender,
GridViewDeleteEventArgs e)
{
    GridViewRow deletedRow = GroupGridView.Rows[e.RowIndex];

    int groupId = int.Parse(deletedRow.Cells[3].Text);

    MyAdoHelper.DoQuery("MyDatabase.mdf", "DELETE FROM tblClimbingGroups
WHERE groupId=" + groupId);
    MyAdoHelper.DoQuery("MyDatabase.mdf", "DELETE FROM tblGroupMembers
WHERE groupId=" + groupId);

    DataTable groupMeetingsTbl =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblGroupMeetings
WHERE groupId=" + groupId);

    if (groupMeetingsTbl.Rows.Count > 0)
    {
        string meetingIdList = "(" +
groupMeetingsTbl.Rows[0]["meetingId"].ToString();

        for (int i = 1; i < groupMeetingsTbl.Rows.Count; i++)
        {
            meetingIdList += ", " +
groupMeetingsTbl.Rows[i]["meetingId"].ToString();
        }

        meetingIdList += ")";
    }
}

```

```

        MyAdoHelper.DoQuery("MyDatabase.mdf", "DELETE FROM
tblMeetingMembers WHERE meetingId IN " + meetingIdList);
    }

    MyAdoHelper.DoQuery("MyDatabase.mdf", "DELETE FROM tblGroupMeetings
WHERE groupId=" + groupId);

    MeetingsGridView.Visible = false;
    MeetingsLbl.Visible = false;

    GroupGridView.EditIndex = -1;
    GetGroupData();
}

protected void MeetingsGridView_RowDeleting(object sender,
GridViewDeleteEventArgs e)
{
    GridViewRow deletedRow = MeetingsGridView.Rows[e.RowIndex];

    int meetingId = int.Parse(deletedRow.Cells[3].Text);

    MyAdoHelper.DoQuery("MyDatabase.mdf", "DELETE FROM tblGroupMeetings
WHERE meetingId=" + meetingId);
    MyAdoHelper.DoQuery("MyDatabase.mdf", "DELETE FROM tblMeetingMembers
WHERE meetingId=" + meetingId);

    GetMeetingData();
}

protected void MeetingsGridView_RowCancelingEdit(object sender,
GridViewCancelEventArgs e)
{
    MeetingsGridView.EditIndex = -1;
    GetMeetingData();
}

protected void MeetingsGridView_RowEditing(object sender,
GridViewEditEventArgs e)
{
    MeetingsGridView.EditIndex = e.NewEditIndex;
    GetMeetingData();
}

protected void MeetingsGridView_RowUpdating(object sender,
GridViewUpdateEventArgs e)
{
    GridViewRow changedRow = MeetingsGridView.Rows[e.RowIndex];

    int groupId = int.Parse(changedRow.Cells[2].Text);
    int meetingId = int.Parse(changedRow.Cells[3].Text);
    string city =
System.Threading.Thread.CurrentThread.CurrentCulture.TextInfo.ToTitleCase(((TextBox)changedRow.FindControl("CityTextBox")).Text.ToLower());
    string country =
System.Threading.Thread.CurrentThread.CurrentCulture.TextInfo.ToTitleCase(((TextBox)changedRow.FindControl("CountryTextBox")).Text.ToLower());
    string desc = ((TextBox)changedRow.FindControl("DescTextBox")).Text;
    DateTime startTime =
DateTime.Parse(((TextBox)changedRow.FindControl("StartTimeTextBox")).Text);
    DateTime endTime =
DateTime.Parse(((TextBox)changedRow.FindControl("EndTimeTextBox")).Text);
}

```

```

        if (startDateTime < DateTime.Now || endDateTime < DateTime.Now ||
startDateTime > DateTime.Now.AddDays(30) || endDateTime >
DateTime.Now.AddDays(30))
    {
        ErrorLbl.Text = "<h3 class='red'>The starting and ending datetimes
have to be within 30 days of now!</h3>";
        ErrorLbl.Visible = true;
        return;
    }

    if (startDateTime > endDateTime.AddHours(-1))
    {
        ErrorLbl.Text = "<h3 class='red'>The starting datetime must be
before the ending datetime, and the meeting must be at least one hour
long!</h3>";
        ErrorLbl.Visible = true;
        return;
    }

    DataTable groupCoachDt = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf",
"SELECT * FROM tblClimbingGroups WHERE groupId=" + groupId);

    string coachName = groupCoachDt.Rows[0]["coachName"].ToString();

    DataTable coachGroupsDt =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblClimbingGroups
WHERE coachName='" + coachName + "'");

    string groupIdList = "(" + coachGroupsDt.Rows[0]["groupId"].ToString();

    for (int i = 1; i < coachGroupsDt.Rows.Count; i++)
    {
        groupIdList += ", " + coachGroupsDt.Rows[i]["groupId"].ToString();
    }

    groupIdList += ")";

    DataTable groupMeetingsDt =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblGroupMeetings
WHERE groupId IN " + groupIdList);

    DataRow meetingRow = groupMeetingsDt.Select("meetingId=" +
meetingId)[0];

    groupMeetingsDt.Rows.Remove(meetingRow);

    if (groupMeetingsDt.Rows.Count > 0)
    {
        foreach (DataRow dataRow in groupMeetingsDt.Rows)
        {
            if (IsDateTimeBetween((DateTime)dataRow["startDateTime"],
(DateTime)dataRow["endDateTime"], startDateTime) ||
IsDateTimeBetween((DateTime)dataRow["startDateTime"],
(DateTime)dataRow["endDateTime"], endDateTime))
            {
                ErrorLbl.Text = "<h3 class='red'>This group's coach already
has a meeting during this time period!</h3>";
                ErrorLbl.Visible = true;
                return;
            }
        }
    }
}

```

```
using (WebClient client = new WebClient())
{
    try
    {
        ICountryProvider countryProvider = new CountryProvider();
        var countryInfo = countryProvider.GetCountryByName(country);

        string query = city + ",0," +
countryInfo.Alpha2Code.ToString();

        string url =
string.Format("http://api.openweathermap.org/geo/1.0/direct?q={0}&appid={1}",
query, weatherApiKey);

        string json = client.DownloadString(url);
        GeoRoot geoRoot =
JsonConvert.DeserializeObject<GeoRoot>(json.Substring(1, json.Length - 2));
    }
    catch
    {
        ErrorLbl.Text = "<h3 class='red'>You entered a bad city/country
name!</h3>";
        ErrorLbl.Visible = true;
        return;
    }
}

MyAdoHelper.DoQuery("MyDatabase.mdf", "UPDATE tblGroupMeetings SET
city='" + city + "', country='" + country + "', startTime='" +
dateTime.ToString("yyyy-MM-dd HH:mm:ss.fff") + "', endTime='" +
dateTime.ToString("yyyy-MM-dd HH:mm:ss.fff") + "', meetingDesc='" + desc +
"' WHERE meetingId=" + meetingId);

MeetingsGridView.EditIndex = -1;
GetMeetingData();
}

protected bool IsDateTimeBetween(DateTime min, DateTime max, DateTime
value)
{
    if (value >= min && value <= max)
        return true;
    return false;
}

protected void MeetingsGridView_PageIndexChanging(object sender,
GridViewEventArgs e)
{
    MeetingsGridViewPageIndex = e.NewPageIndex;
    GetMeetingData();
}

protected void MeetingsGridView_RowCommand(object sender,
GridViewCommandEventArgs e)
{
}

protected void NewMeetingButton_Click(object sender, EventArgs e)
{
    Response.Redirect("MeetingCreationPage.aspx");
```

}
}

AdminPage.aspx

מיקום: ComputersProjectNew/AdminPage.aspx

מיקום קוד מאחור: ComputersProjectNew/AdminPage.aspx.cs

קוד aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="AdminPage.aspx.cs" Inherits="AdminPage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>Admin Page</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2" id="three">
        <div class="inner">
            <header class="align-center">
                <h2>Admin Page</h2>
            </header>
            <div class="align-center">
                <input type="button" value="Create New User"
onclick="location.href='Registration.aspx'" />
            </div>
        </div>
        <form id="form1" runat="server">
            <br />
            <div class="align-center">
                <asp:Button ID="XMLButton" runat="server"
Text="Open XML" OnClick="GetXML" OnClientClick="form1.target = '_blank';"/>
            </div>
            <br />
            <br />
                <asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="False"
OnSelectedIndexChanged="GridView1_SelectedIndexChanged"
HorizontalAlign="Center" AllowSorting="False" AllowPaging="True"
OnRowDeleting="GridView1_RowDeleting"
OnRowCancelingEdit="GridView1_RowCancelingEdit"
OnRowEditing="GridView1_RowEditing" OnRowUpdating="GridView1_RowUpdating"
OnPageIndexChanging="GridView1_PageIndexChanging" OnSorting="GridView1_Sorting"
PageSize="10">
                    <PagerStyle HorizontalAlign="Center"/>
                    <Columns>
                        <asp:CommandFieldButtonType="Button"
ShowEditButton="true" HeaderText="Editing" CancelText="Cancel Edit"
EditText="Edit User" UpdateText="Confirm Edit"/>
                        <asp:ButtonFieldButtonType="Button"
Text="Delete" CommandName="Delete" HeaderText="Deletion" />
                        <asp:BoundField DataField="username"
ReadOnly="true" HeaderText="Username"/>
                        <asp:TemplateField HeaderText="Password"
SortExpression="password">
                            <EditItemTemplate>
                                <asp:TextBox ID="PasswordTextBox"
runat="server" Text='<%# Bind("password") %>'></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="PasswordTextBox" Display="None" ID="RequiredFieldValidator2" />
                            </EditItemTemplate>
                        </asp:TemplateField>
                    </Columns>
                </asp:GridView>
            </form>
        </section>
    </div>
</body>
</html>
```

```

runat="server" ErrorMessage="Password is
required">></asp:RequiredFieldValidator>
                                <asp:RegularExpressionValidator
ValidationExpression="^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{5,50}$"
ControlToValidate="PasswordTextBox" Display="None"
ID="RegularExpressionValidator2" runat="server" ErrorMessage="Password must be
between 5 and 50 characters long, and contain numbers and uppercase and
lowercase letters"></asp:RegularExpressionValidator>
                                </EditItemTemplate>
                                <ItemTemplate>
                                    <asp:Label ID="PasswordLabel">
runat="server" Text='<%# Bind("password") %>'></asp:Label>
                                    </ItemTemplate>
                                </asp:TemplateField>
                                <asp:TemplateField HeaderText="Email">
SortExpression="email">
                                <EditItemTemplate>
                                    <asp:TextBox ID="EmailTextBox">
runat="server" Text='<%# Bind("email") %>'></asp:TextBox>
                                    <asp:RequiredFieldValidator
ControlToValidate="EmailTextBox" Display="None" ID="RequiredFieldValidator5"
runat="server" ErrorMessage="Email is required"></asp:RequiredFieldValidator>
                                    <asp:RegularExpressionValidator
ValidationExpression="^[\w-\.]+@[^\w-]+\.\w{2,4}$"
ControlToValidate="EmailTextBox" Display="None"
ID="RegularExpressionValidator4" runat="server" ErrorMessage="Wrong email
syntax"></asp:RegularExpressionValidator>
                                </EditItemTemplate>
                                <ItemTemplate>
                                    <asp:Label ID="EmailLabel">
runat="server" Text='<%# Bind("email") %>'></asp:Label>
                                    </ItemTemplate>
                                </asp:TemplateField>
                                <asp:TemplateField HeaderText="Address">
SortExpression="address">
                                <EditItemTemplate>
                                    <asp:TextBox ID="AddressTextBox">
runat="server" Text='<%# Bind("address") %>'></asp:TextBox>
                                    <asp:RequiredFieldValidator
ControlToValidate="AddressTextBox" Display="None" ID="RequiredFieldValidator4"
runat="server" ErrorMessage="Address is required"></asp:RequiredFieldValidator>
                                    <asp:RegularExpressionValidator
ValidationExpression="^([A-Z][a-z]*\s)*[1-9][0-9]*$"
ControlToValidate="AddressTextBox" Display="None"
ID="RegularExpressionValidator3" runat="server" ErrorMessage="Wrong address
syntax"></asp:RegularExpressionValidator>
                                </EditItemTemplate>
                                <ItemTemplate>
                                    <asp:Label ID="AddressLabel">
runat="server" Text='<%# Bind("address") %>'></asp:Label>
                                    </ItemTemplate>
                                </asp:TemplateField>
                                <asp:TemplateField HeaderText="Gender">
SortExpression="gender">
                                <EditItemTemplate>
                                    <asp:RadioButtonList ID="GenderRBL">
runat="server">
                                    <asp:ListItem
Value="male">Male</asp:ListItem>
                                    <asp:ListItem
Value="female">Female</asp:ListItem>
                                </asp:RadioButtonList>

```

```

        <asp:RequiredFieldValidator
ControlToValidate="GenderRBL" Display="None" ID="RequiredFieldValidator9"
runat="server" ErrorMessage="Gender is required"></asp:RequiredFieldValidator>
        </EditItemTemplate>
        <ItemTemplate>
            <asp:Label ID="GenderLbl"
runat="server" Text='<%# Eval("gender").ToString().Substring(0, 1).ToUpper() +
Eval("gender").ToString().Substring(1) %>'></asp:Label>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Permission
Level" SortExpression="permLvl">
            <EditItemTemplate>
                <asp:DropDownList ID="PermLvlDDL"
runat="server">
                    <asp:ListItem Value="0">Normal
User</asp:ListItem>
                    <asp:ListItem
Value="1">Coach</asp:ListItem>
                    <asp:ListItem Value="2">Supply
Manager</asp:ListItem>
                    <asp:ListItem Value="3">General
Manager</asp:ListItem>
                </asp:DropDownList>
                <asp:RequiredFieldValidator
ControlToValidate="PermLvlDDL" Display="None" ID="RequiredFieldValidator10"
runat="server" ErrorMessage="Permission level is
required"></asp:RequiredFieldValidator>
            </EditItemTemplate>
            <ItemTemplate>
                <asp:Label ID="PermLvlLbl"
runat="server" Text='<%# PermLvlToStr(Eval("permLvl").ToString()) %>'></asp:Label>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Boulder Grade"
SortExpression="boulderGrade">
            <EditItemTemplate>
                <asp:TextBox ID="BoulderGradeTextBox"
runat="server" Text='<%# Bind("boulderGrade") %>'></asp:TextBox>
                <asp:RequiredFieldValidator
ControlToValidate="BoulderGradeTextBox" Display="None"
ID="RequiredFieldValidator6" runat="server" ErrorMessage="Boulder grade is
required"></asp:RequiredFieldValidator>
                <asp:RegularExpressionValidator
ValidationExpression="^V([0-9]|1[0-7])$"
ControlToValidate="BoulderGradeTextBox" Display="None"
ID="RegularExpressionValidator5" runat="server" ErrorMessage="Wrong boulder
grade syntax. Must be V0 to V17"></asp:RegularExpressionValidator>
            </EditItemTemplate>
            <ItemTemplate>
                <asp:Label ID="BoulderGradeLabel"
runat="server" Text='<%# Bind("boulderGrade") %>'></asp:Label>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Wall Grade"
SortExpression="wallGrade">
            <EditItemTemplate>
                <asp:TextBox ID="WallGradeTextBox"
runat="server" Text='<%# Bind("wallGrade") %>'></asp:TextBox>

```

```

        <asp:RequiredFieldValidator
ControlToValidate="WallGradeTextBox" Display="None"
ID="RequiredFieldValidator7" runat="server" ErrorMessage="Wall grade is
required"></asp:RequiredFieldValidator>
        <asp:RegularExpressionValidator
ValidationExpression="^([5-9][A-C][+]{0,1})$"
ControlToValidate="WallGradeTextBox" Display="None"
ID="RegularExpressionValidator6" runat="server" ErrorMessage="Wrong wall grade
syntax. Must be 5A to 9C (+ is optional)"></asp:RegularExpressionValidator>
    </EditItemTemplate>
    <ItemTemplate>
        <asp:Label ID="WallGradeLabel"
runat="server" Text='<%# Bind("wallGrade") %>'></asp:Label>
    </ItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Birthdate">
    <EditItemTemplate>
        <asp:TextBox ID="BirthdateTextBox"
runat="server" Text='<%# Bind("birthdate", "{0:yyyy-MM-dd}") %>'>
        <asp:RequiredFieldValidator
ControlToValidate="BirthdateTextBox" Display="None"
ID="RequiredFieldValidator8" runat="server" ErrorMessage="Birthdate is
required"></asp:RequiredFieldValidator>
        <asp:RangeValidator
id="BirthDateRangeValidator" runat="server" Type="Date"
ControlToValidate="BirthdateTextBox" MinimumValue="<%# rvMin %>"
MaximumValue="<%# rvMax %>" ErrorMessage="Invalid age, out of range"
Display="None"/>
    </EditItemTemplate>
    <ItemTemplate>
        <asp:Label ID="BirthdateLabel"
runat="server" Text='<%# Bind("birthdate", "{0:d}") %>'></asp:Label>
    </ItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="City"
SortExpression="city">
    <EditItemTemplate>
        <asp:DropDownList ID="CityDDL"
runat="server" Height="16px" SelectedValue='<%# Bind("city") %>' Width="106px">
            <asp:ListItem>Haifa</asp:ListItem>
            <asp:ListItem>Tel
Aviv</asp:ListItem>
            <asp:ListItem>Jerusalem</asp:ListItem>
                <asp:ListItem>Ashdod</asp:ListItem>
                <asp:ListItem>Rishon
LeZiyon</asp:ListItem>
                <asp:ListItem>Petah
Tikva</asp:ListItem>
                <asp:ListItem>Beer
Sheva</asp:ListItem>
            <asp:ListItem>Netanya</asp:ListItem>
                <asp:ListItem>Holon</asp:ListItem>
                <asp:ListItem>Bnei
Brak</asp:ListItem>
            <asp:ListItem>Rehovot</asp:ListItem>
                <asp:ListItem>Bat
Yam</asp:ListItem>
        </asp:DropDownList>
    </EditItemTemplate>

```

```

        <asp:RequiredFieldValidator
ControlToValidate="CityDDL" Display="None" ID="RequiredFieldValidator3"
runat="server" ErrorMessage="City is required"></asp:RequiredFieldValidator>
        </EditItemTemplate>
        <ItemTemplate>
            <asp:Label ID="CityLabel"
runat="server" Text='<%# Bind("city") %>'></asp:Label>
            </ItemTemplate>
        </asp:TemplateField>
    </Columns>
</asp:GridView>
<asp:ValidationSummary ID="SummaryValidator"
runat="server" ShowMessageBox="true" ShowSummary="false" />
</form>
</section>
</asp:Content>

```

C# קוד:

```

using System;
using System.Data;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Globalization;

public partial class AdminPage : System.Web.UI.Page
{
    public string rvMin = DateTime.Today.AddYears(-120).ToString("yyyy/MM/dd"),
               rvMax = DateTime.Today.AddYears(-8).ToString("yyyy/MM/dd");
    public const int AdminReq = 3;
    public DataTable dt; //ןן המוחזרים הנתונים את שומר
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["username"] == null)//רשות המשתמש אם בודק
            Response.Redirect("NoAccessPage.aspx"); //כניתה אין לדף המשתמש את מעביר

        int permLvl = int.Parse(Session["userPermLvl"].ToString()); //את מקבל מהסשן המשתמש של הגישה הרשות

        if (permLvl < AdminReq)//המשתמש של הגישה הרשות את בודק
            Response.Redirect("NoAccessPage.aspx"); //כניתה אין לדף המשתמש את מעביר

        if (!IsPostBack)
        {
            foreach (GridViewRow r in GridView1.Rows)
            {
                r.Visible = r.Cells[2].Text != Session["username"].ToString();
            }
            getUsers();
        }
    }

    public void getUsers()
    {
        dt = MyAdoHelper.GetUserTbl();
    }
}

```

```
    if (dt == null)
        Response.Redirect("NoAccessPage.aspx");
    GridView1.DataSource = dt;
    GridView1.DataBind();
}

protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
{

}

protected void GridView1_RowCancelingEdit(object sender,
GridViewCancelEventArgs e)
{
    GridView1.EditIndex = -1;
    getUsers();
}

protected void GridView1_RowEditing(object sender, GridViewEditEventArgs e)
{
    GridView1.EditIndex = e.NewEditIndex;

    getUsers();

    DropDownList permLvlDDL =
(DropDownList)GridView1.Rows[e.NewEditIndex].FindControl("PermLvlDDL");
    RadioButtonList genderRBL =
(RadioButtonList)GridView1.Rows[e.NewEditIndex].FindControl("GenderRBL");

    string username = GridView1.Rows[e.NewEditIndex].Cells[2].Text;

    DataTable userInfoDt = MyAdoHelper.GetUserInfo(username);

    permLvlDDL.SelectedValue = userInfoDt.Rows[0]["permLvl"].ToString();
    genderRBL.SelectedValue = userInfoDt.Rows[0]["gender"].ToString();
}

protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    GridViewRow DeletedRow = GridView1.Rows[e.RowIndex];
    string username = DeletedRow.Cells[2].Text;
    bool userDeleted = MyAdoHelper.DeleteUser(username);
    if (!userDeleted)
        Response.Redirect("NoAccessPage.aspx");
    GridView1.EditIndex = -1;
    getUsers();
}

protected void GridView1_PageIndexChanging(object sender,
GridViewPageEventArgs e)
{
    GridView1PageIndex = e.NewPageIndex;
    getUsers();
}

protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    GridViewRow changedRow = GridView1.Rows[e.RowIndex];
    string username = changedRow.Cells[2].Text;
```

```

        string password =
((TextBox)changedRow.FindControl("PasswordTextBox")).Text;
        string email = ((TextBox)changedRow.FindControl("EmailTextBox")).Text;
        string address =
((TextBox)changedRow.FindControl("AddressTextBox")).Text;
        string boulderGrade =
((TextBox)changedRow.FindControl("BoulderGradeTextBox")).Text;
        string wallGrade =
((TextBox)changedRow.FindControl("WallGradeTextBox")).Text;
        string birthdate =
((TextBox)changedRow.FindControl("BirthdateTextBox")).Text;
        string city = ((DropDownList)changedRow.FindControl("CityDDL")).Text;
        string permLvl =
((DropDownList)changedRow.FindControl("PermLvlDDL")).Text;
        string gender =
((RadioButtonList)changedRow.FindControl("GenderRBL")).Text;
        bool success = MyAdoHelper.UpdateUser(username, password, email,
address, boulderGrade, wallGrade, birthdate, city, permLvl, gender);
        if (!success)
            Response.Redirect("NoAccessPage.aspx");
        GridView1.EditIndex = -1;
        getUsers();
    }

protected void GridView1_Sorting(object sender, GridViewSortEventArgs e)
{
    string column = e.SortExpression;
    if (Session[column] == null)
    {
        Session[column] = 0;
    }
    if ((int)Session[column] % 2 == 0)
    {
        dt = MyAdoHelper.GetDtSorted(column, "ASC");
    }
    else
    {
        dt = MyAdoHelper.GetDtSorted(column, "DESC");
    }
    if(dt == null)
    {
        Session[column] = null;
        Response.Redirect("NoAccessPage.aspx");
    }
    else
    {
        Session[column] = (int)Session[column] + 1;
        GridView1.DataSource = dt;
        GridView1.DataBind();
    }
}

protected void GetXML(object sender, EventArgs e)
{
    string currentUser = Session["username"].ToString();
    DataTable userDt = MyAdoHelper.GetUserInfo(currentUser);

    string email = userDt.Rows[0]["email"].ToString();
    string content = "Hi, " + currentUser + "! We attached the info you
requested in XML format to this email so you can download it and save it.
Thanks for caring for our website :)\n\nBest regards,\nThe Climbing Web Team";
}

```

```
DataSet ds = MyAdoHelper.GetDataSetBySP("getUserInfoTbl");
DataTable dt = ds.Tables[0];

dt.TableName = "User";
ds.DataSetName = "USERS";

ds.WriteXml(Server.MapPath(@"UserInfoFile.xml"));

MyServiceReference.ShoeCalculatorServiceSoapClient client = new
MyServiceReference.ShoeCalculatorServiceSoapClient();

client.SendEmailDS("User Info Request", currentUser, email, content,
ds);
}

protected string PermLvlToStr(string permLvl)
{
    switch (permLvl)
    {
        case "0":
            return "Normal User";
        case "1":
            return "Coach";
        case "2":
            return "Supply Manager";
        case "3":
            return "General Manager";
        default:
            return "Error";
    }
}
}
```

SupplyManagement.aspx

מיקום: ComputersProjectNew/SupplyManagement.aspx

מיקום קוד מאחור: ComputersProjectNew SupplyManagement.aspx.cs

קוד aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="SupplyManagement.aspx.cs"
Inherits="SupplyManagement" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>Supply Management</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2" id="three">
        <div class="inner">
            <header class="align-center">
                <h2>Supply Management Table</h2>
                <br />
            </header>
        </div>
        <form id="form1" runat="server">
            <div class="align-center">
                <asp:Button ID="NewItemButton" runat="server" Text="Add New
Item" OnClick="NewItemButton_Click" />
                <asp:Button ID="SalesReportPage" runat="server" Text="View
Sales Report" OnClick="SalesReportPage_Click"/>
                <br />
                <asp:Label ID="ErrorLbl" runat="server"
Visible="false"></asp:Label>
                <br />
            </div>
            <br />
            <br />
            <br />
            <asp:GridView ID="ProductGridView" runat="server" Width="100%">
                <AutoGenerateColumns=False HorizontalAlign="Center" AllowPaging="True"
PageSize="3" OnRowEditing="ProductGridView_RowEditing"
OnRowCancelingEdit="ProductGridView_RowCancelingEdit"
OnSelectedIndexChanged="ProductGridView_SelectedIndexChanged"
OnRowUpdating="ProductGridView_RowUpdating"
OnPageIndexChanging="ProductGridView_PageIndexChanging"
OnRowCommand="ProductGridView_RowCommand">
                    <PagerStyle HorizontalAlign="Center"/>
                    <Columns>
                        <asp:CommandField ButtonType="Button"
ShowEditButton="true" HeaderText="Editing" CancelText="Cancel Edit"
EditText="Edit" UpdateText="Confirm Edit" ItemStyle-Width="300"/>
                        <asp:ButtonField ButtonType="Button" Text="View Supply"
CommandName="Select" HeaderText="Supply Viewing"/>
                        <asp:BoundField DataField="productId" ReadOnly="true"
HeaderText="Product ID"/>
                        <asp:TemplateField HeaderText="Product Name">
                            <EditItemTemplate>
                                <asp:TextBox ID="ProductNameTextBox"
runat="server" Text='<%# Eval("name") %>'></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="ProductNameTextBox" Display="None" />
                            </EditItemTemplate>
                        </asp:TemplateField>
                    </Columns>
                </asp:GridView>
            </div>
        </form>
    </div>
</section>

```

```

ID="RequiredFieldValidator1" runat="server" ErrorMessage="Product name is
required"></></asp:RequiredFieldValidator>
          <asp:RegularExpressionValidator
ValidationExpression="^([a-zA-Z][a-zA-Z ]{0,98}[a-zA-Z]$"
ControlToValidate="ProductNameTextBox" Display="None"
ID="RegularExpressionValidator1" runat="server" ErrorMessage="Product name must
be up to 100 characters long!"></asp:RegularExpressionValidator>
          </EditItemTemplate>
          <ItemTemplate>
            <asp:Label ID="ProductNameLbl" runat="server"
Text='<%# Bind("name") %>'></asp:Label>
            </ItemTemplate>
          </asp:TemplateField>
          <asp:TemplateField HeaderText="Company">
            <EditItemTemplate>
              <asp:TextBox ID="CompanyTextBox" runat="server"
Text='<%# Eval("company") %>'></asp:TextBox>
              <asp:RequiredFieldValidator
ControlToValidate="CompanyTextBox" Display="None" ID="RequiredFieldValidator2"
runat="server" ErrorMessage="Company is required"></asp:RequiredFieldValidator>
              <asp:RegularExpressionValidator
ValidationExpression="^([a-zA-Z][a-zA-Z ]{0,98}[a-zA-Z]$"
ControlToValidate="CompanyTextBox" Display="None"
ID="RegularExpressionValidator2" runat="server" ErrorMessage="Company must be
up to 100 characters long!"></asp:RegularExpressionValidator>
            </EditItemTemplate>
            <ItemTemplate>
              <asp:Label ID="CompanyLbl" runat="server"
Text='<%# Bind("company") %>'></asp:Label>
              </ItemTemplate>
            </asp:TemplateField>
            <asp:TemplateField HeaderText="Image">
              <EditItemTemplate>
                <asp:FileUpload ID="ImageUploader"
runat="server" AllowMultiple="false"/>
                <asp:RegularExpressionValidator
ID="RegularExpressionValidator3" runat="server"
ControlToValidate="ImageUploader" ValidationExpression="^(([a-zA-
Z:]|(\w{2}\w+)$?)(\w[\w.*])(.jpg|.JPG|.png|.PNG|.jpeg|.JPEG)$"
ErrorMessage="You may only upload .jpg and .png files"
Display="None"></asp:RegularExpressionValidator>
                <asp:Button CssClass="button small"
ID="UploadButton" runat="server" Text="Upload" CommandName="Upload"/>
              </EditItemTemplate>
              <ItemTemplate>
                <asp:Image ID="ImageBox" runat="server"
ImageUrl='<%# "~/productImages/" + Eval("imageName") %>' Height="100"
Width="100" CssClass="gv"/>
              </ItemTemplate>
            </asp:TemplateField>
          </Columns>
        </asp:GridView>

<br />
<div class="align-center">
  <asp:Label ID="SupplyLbl" runat="server"
Visible="false"></asp:Label>
</div>
<br />
  <asp:GridView ID="SupplyGridView" runat="server"
AutoGenerateColumns="False"
OnSelectedIndexChanged="SupplyGridView_SelectedIndexChanged"

```

```
HorizontalAlign="Center" AllowPaging="True"
OnRowDeleting="SupplyGridView_RowDeleting"
OnRowCancelingEdit="SupplyGridView_RowCancelingEdit"
OnRowEditing="SupplyGridView_RowEditing"
OnRowUpdating="SupplyGridView_RowUpdating"
OnPageIndexChanging="SupplyGridView_PageIndexChanging"
OnRowCommand="SupplyGridView_RowCommand">
    <PagerStyle HorizontalAlign="Center"/>
    <FooterStyle HorizontalAlign="Left"/>
    <Columns>
        <asp:TemplateField HeaderText="Controls">
            <ItemTemplate>
                <asp:Button ID="EditButton" runat="server"
CommandName="Edit" Text="Edit"/>
                <asp:Button ID="DeleteButton" runat="server"
CommandName="Delete" Text="Reset Supply"/>
            </ItemTemplate>
            <EditItemTemplate>
                <asp:Button ID="CancelEditButton"
runat="server" CommandName="Cancel" Text="Cancel Edit"/>
                <asp:Button ID="UpdateButton" runat="server"
CommandName="Update" Text="Update" ValidationGroup="EditingValidationGroup"/>
            </EditItemTemplate>
            <FooterTemplate>
                <asp:Button ID="AddNewButton" runat="server"
CommandName="AddNew" Text="Add" ValidationGroup="AddNewValidationGroup"/>
            </FooterTemplate>
        </asp:TemplateField>
        <asp:BoundField DataField="productId" ReadOnly="true"
HeaderText="Product ID" ItemStyle-CssClass="hiddencol" HeaderStyle-
CssClass="hiddencol" FooterStyle-CssClass="hiddencol"/>
        <asp:TemplateField HeaderText="Size">
            <ItemTemplate>
                <asp:Label ID="SizeLbl" runat="server"
Text='<%# Bind("size") %>'></asp:Label>
            </ItemTemplate>
            <FooterTemplate>
                <asp:TextBox ID="NewSizeTxt"
CssClass="widetext" runat="server"></asp:TextBox>
                <asp:RequiredFieldValidator
ControlToValidate="NewSizeTxt" ValidationGroup="AddNewValidationGroup"
Display="None" ID="RequiredFieldValidator4" runat="server" ErrorMessage="CVV is
required"></asp:RequiredFieldValidator>
                <asp:RangeValidator id="RangeValidator2"
runat="server" Type="Integer" ControlToValidate="NewSizeTxt"
ValidationGroup="AddNewValidationGroup" MinimumValue="30" MaximumValue="55"
ErrorMessage="Invalid size, out of range (must be between 30 and 55)"
Display="None"/>
            </FooterTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Price">
            <EditItemTemplate>
                <asp:TextBox ID="PriceTxt" CssClass="widetext"
runat="server" Text='<%# Eval("price") %>'></asp:TextBox>
                <asp:RequiredFieldValidator
ControlToValidate="PriceTxt" ValidationGroup="EditingValidationGroup"
Display="None" ID="RequiredFieldValidator1" runat="server" ErrorMessage="Price
is required"></asp:RequiredFieldValidator>
                <asp:RangeValidator
ControlToValidate="PriceTxt" ValidationGroup="EditingValidationGroup"
Display="None" ID="RangeValidator1" runat="server" Type="Double"
```

```

MinimumValue="0.01" MaximumValue="9999.99" ErrorMessage="Invalid price! Price
must be a decimal value between 0.01 and 9999.99">></asp:RangeValidator>
        </EditItemTemplate>
        <ItemTemplate>
            <asp:Label ID="PriceLbl" runat="server"
Text='<%# Bind("price") %>'></asp:Label>
        </ItemTemplate>
        <FooterTemplate>
            <asp:TextBox ID="NewPriceTxt"
CssClass="widetext" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator
ControlToValidate="NewPriceTxt" ValidationGroup="EditingValidationGroup"
Display="None" ID="RequiredFieldValidator3" runat="server"
ErrorMessage="Expiration date is required"></asp:RequiredFieldValidator>
            <asp:RangeValidator id="RangeValidator5"
runat="server" Type="Double" ControlToValidate="NewPriceTxt"
ValidationGroup="EditingValidationGroup" MinimumValue="0.01"
MaximumValue="9999.99" ErrorMessage="Invalid price! Price must be a decimal
value between 0.01 and 9999.99" Display="None"/>
        </FooterTemplate>
    </asp:TemplateField>
    <asp:TemplateField HeaderText="Amount">
        <EditItemTemplate>
            <asp:TextBox ID="AmountTxt" CssClass="widetext"
runat="server" Text='<%# Eval("amount") %>'></asp:TextBox>
            <asp:RequiredFieldValidator
ControlToValidate="AmountTxt" ValidationGroup="EditingValidationGroup"
Display="None" ID="RequiredFieldValidator2" runat="server" ErrorMessage="Amount
is required"></asp:RequiredFieldValidator>
            <asp:RangeValidator
ControlToValidate="AmountTxt" ValidationGroup="EditingValidationGroup"
Display="None" ID="RangeValidator2" runat="server" Type="Integer"
MinimumValue="0" MaximumValue="9999" ErrorMessage="Invalid amount! Amount must
be an integer value between 0 and 9999"></asp:RangeValidator>
        </EditItemTemplate>
        <ItemTemplate>
            <asp:Label ID="AmountLbl" runat="server"
Text='<%# Bind("amount") %>'></asp:Label>
        </ItemTemplate>
        <FooterTemplate>
            <asp:TextBox ID="NewAmountTxt"
CssClass="widetext" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator
ControlToValidate="NewAmountTxt" ValidationGroup="AddNewValidationGroup"
Display="None" ID="RequiredFieldValidator2" runat="server" ErrorMessage="Amount
is required"></asp:RequiredFieldValidator>
            <asp:RangeValidator
ControlToValidate="NewAmountTxt" ValidationGroup="AddNewValidationGroup"
Display="None" ID="RangeValidator4" runat="server" Type="Integer"
MinimumValue="0" MaximumValue="9999" ErrorMessage="Invalid amount! Amount must
be an integer value between 0 and 9999"></asp:RangeValidator>
        </FooterTemplate>
    </asp:TemplateField>
</Columns>
</asp:GridView>
<asp:ValidationSummary ID="ValidationSummary1" runat="server"
ShowMessageBox="true" ShowSummary="false"
ValidationGroup="EditingValidationGroup"/>
<asp:ValidationSummary ID="ValidationSummary2" runat="server"
ShowMessageBox="true" ShowSummary="false"
ValidationGroup="AddNewValidationGroup"/>
</form>

```

```

        </section>
</asp:Content>
```

:C# טוֹפַּה

```

using System;
using System.Data;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class SupplyManagement : System.Web.UI.Page
{
    public const int AdminReq = 2;

    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["username"] == null)// בודק אם המשתמש נכנס לאין לדף המשתמש את מעביר את מקלט
            Response.Redirect("NoAccessPage.aspx"); // כנראה אין לו הרשות

        int permLvl = int.Parse(Session["userPermLvl"].ToString()); // מהסמן המשתמש של הגישה הרשות

        if (permLvl < AdminReq) // בודק אם המשתמש לא אינן לאין לדף המשתמש את מעביר את מקלט
            Response.Redirect("NoAccessPage.aspx"); // כנראה אין לו הרשות

        if (!IsPostBack)
        {
            GetProductData();
        }
    }

    protected void GetProductData()
    {
        DataTable productTbl = MyAdoHelper.GetProductTbl();

        ProductGridView.DataSource = productTbl;
        ProductGridView.DataBind();
    }

    protected void GetSupplyData()
    {
        int productId = int.Parse(ProductGridView.SelectedRow.Cells[2].Text);
        string productName =
        ((Label)ProductGridView.SelectedRow.FindControl("ProductNameLbl")).Text;

        DataTable supplyTbl = MyAdoHelper.GetSupplyTbl().Select("productId=" +
productId).CopyToDataTable();

        SupplyLbl.Text = "<h3>Supply Table For Product \" " + productName +
" \"</h3>";
        SupplyLbl.Visible = true;

        SupplyGridView.ShowFooter = true;
        SupplyGridView.DataSource = supplyTbl;
```

```
        SupplyGridView.DataBind();
    }

    protected void SupplyGridView_SelectedIndexChanged(object sender, EventArgs e)
    {
    }

    protected void SupplyGridView_RowCancelingEdit(object sender,
GridViewCancelEventArgs e)
{
    SupplyGridView.EditIndex = -1;
    GetSupplyData();
}

protected void SupplyGridView_RowEditing(object sender,
GridViewEditEventArgs e)
{
    SupplyGridView.EditIndex = e.NewEditIndex;
    GetSupplyData();
}

protected void SupplyGridView_RowDeleting(object sender,
GridViewDeleteEventArgs e)
{
    GridViewRow DeletedRow = SupplyGridView.Rows[e.RowIndex];

    int productId = int.Parse(DeletedRow.Cells[1].Text);
    int size = int.Parse(DeletedRow.Cells[2].Text);

    MyAdoHelper.DoQuery("MyDatabase.mdf", "UPDATE tblSupply SET amount=0
WHERE productId='"
+ productId + "' AND size='"
+ size + "'");

    SupplyGridView.EditIndex = -1;
    GetSupplyData();
}

protected void SupplyGridView_PageIndexChanged(object sender,
GridViewPageEventArgs e)
{
    SupplyGridViewPageIndex = e.NewPageIndex;
    GetSupplyData();
}

protected void SupplyGridView_RowUpdating(object sender,
GridViewUpdateEventArgs e)
{
    GridViewRow changedRow = SupplyGridView.Rows[e.RowIndex];
    int productId = int.Parse(changedRow.Cells[1].Text);
    int size = int.Parse(((Label)changedRow.FindControl("SizeLbl")).Text);
    double price =
double.Parse(((TextBox)changedRow.FindControl("PriceTxt")).Text);
    int amount =
int.Parse(((TextBox)changedRow.FindControl("AmountTxt")).Text);

    MyAdoHelper.DoQuery("MyDatabase.mdf", "UPDATE tblSupply SET price='"
+ price + "', amount='"
+ amount + "' WHERE productId='"
+ productId + "' AND
size='"
+ size + "'");

    SupplyGridView.EditIndex = -1;
    GetSupplyData();
}
```

```
}

protected void NewItemButton_Click(object sender, EventArgs e)
{
    Response.Redirect("AddNewItem.aspx");
}

protected void GetXML(object sender, EventArgs e)
{
    string currentUser = Session["username"].ToString();
    DataTable userDt = MyAdoHelper.GetUserInfo(currentUser);

    string email = userDt.Rows[0]["email"].ToString();
    string content = "Hi, " + currentUser + "! We attached the info you
requested in XML format to this email so you can download it and save it.
Thanks for caring for our website :)\n\nBest regards,\nThe Climbing Web Team";

    DataTable supplyTbl = MyAdoHelper.GetSupplyTbl();
    DataSet ds = new DataSet();

    supplyTbl.TableName = "Item";
    ds.DataSetName = "SUPPLY";

    MyServiceReference.ShoeCalculatorServiceSoapClient client = new
MyServiceReference.ShoeCalculatorServiceSoapClient();

    client.SendEmailDS("Supply Info Request", currentUser, email, content,
ds);
}

protected void SalesReportPage_Click(object sender, EventArgs e)
{
    Response.Redirect("SalesReportPage.aspx");
}

protected void ProductGridView_RowEditing(object sender,
GridViewEventArgs e)
{
    ProductGridView.EditIndex = e.NewEditIndex;
    GetProductData();
}

protected void ProductGridView_RowCancelingEdit(object sender,
GridViewCancelEventArgs e)
{
    ProductGridView.EditIndex = -1;
    GetProductData();
}

protected void ProductGridView_SelectedIndexChanged(object sender,
EventArgs e)
{
    GetSupplyData();
}

protected void ProductGridView_RowUpdating(object sender,
GridViewUpdateEventArgs e)
{
    GridViewRow changedRow = ProductGridView.Rows[e.RowIndex];

    int productId = int.Parse(changedRow.Cells[2].Text);
```

```

        string productName =
((TextBox)changedRow.Cells[3].FindControl("ProductNameTextBox")).Text;
        string company =
((TextBox)changedRow.Cells[4].FindControl("CompanyTextBox")).Text;

        DataTable productTbl = MyAdoHelper.GetProductTbl();
        DataRow productRow = productTbl.Select("productId=" + productId)[0];

        if(!productTbl.Select("productId=" +
productId)[0]["name"].Equals(productName))
{
    if(productTbl.Select("name=' " + productName + " '").Count() > 0)
    {
        ErrorLbl.Text = "<h3 class='red'>A product with the name you
tried to use already exists!</h3>";
        ErrorLbl.Visible = true;
        return;
    }
}

MyAdoHelper.DoQuery("MyDatabase.mdf", "UPDATE tblProduct SET name=' " +
productName + " ', company=' " + company + " ' WHERE productId=" + productId);

        ProductGridView.EditIndex = -1;
        GetProductData();
    }

    protected void ProductGridView_PageIndexChanging(object sender,
GridViewEventArgs e)
{
    ProductGridViewPageIndex = e.NewPageIndex;
    GetProductData();
}

    protected void ProductGridView_RowCommand(object sender,
GridViewCommandEvent e)
{
    Button button = e.CommandSource as Button;
    if (e.CommandName.ToLower().Equals("upload"))
    {
        FileUpload fileUpload =
(FileUpload)button.FindControl("ImageUploader");

        if (fileUpload.HasFile)
        {
            UploadFile(fileUpload);
        }
    }
}

    protected void UploadFile(FileUpload fileUpload)
{
    if (ProductGridView.EditIndex == -1)
        return;

    GridViewRow changedRow =
ProductGridView.Rows[ProductGridView.EditIndex];

    int productId = int.Parse(changedRow.Cells[2].Text);

    DataTable productTbl = MyAdoHelper.GetProductTbl();
    DataRow productRow = productTbl.Select("productId=" + productId)[0];
}

```

```

        if (productRow["imageName"].ToString() != "not_found.png" &&
File.Exists(Server.MapPath("~/productImages/" +
productRow["imageName"].ToString())))
{
    File.Delete(Server.MapPath("~/productImages/" +
productRow["imageName"].ToString()));
}
string ext = Path.GetExtension(fileUpload.PostedFile.FileName);
fileUpload.SaveAs(Server.MapPath("~/productImages/product" + productId
+ ext.ToLower()));
MyAdoHelper.DoQuery("MyDatabase.mdf", "UPDATE tblProduct SET
imageName='product" + productId + ext.ToLower() + "' WHERE productId=" +
productId);

        ProductGridView.EditIndex = -1;
        GetProductData();
    }

protected void SupplyGridView_RowCommand(object sender,
GridViewEventArgs e)
{
    if (e.CommandName.ToLower().Equals("addnew"))
    {
        int productId = int.Parse(SupplyGridView.Rows[0].Cells[1].Text);
        int size =
int.Parse(((TextBox)SupplyGridView.FooterRow.FindControl("NewSizeTxt")).Text);
        double price =
double.Parse(((TextBox)SupplyGridView.FooterRow.FindControl("NewPriceTxt")).Text);
        int amount =
int.Parse(((TextBox)SupplyGridView.FooterRow.FindControl("NewAmountTxt")).Text);
;

        foreach(GridViewRow row in SupplyGridView.Rows)
        {
            if
(((Label)row.FindControl("SizeLbl")).Text.Equals(size.ToString()))
            {
                ErrorLbl.Text = "<h3 class='red'>You tried to add a size
which already exists!</h3>";
                ErrorLbl.Visible = true;
                return;
            }
        }

        try
        {
            MyAdoHelper.DoQuery("MyDatabase.mdf", "INSERT INTO tblSupply
VALUES (" + productId + ", " + size + ", " + amount + ", " + price + ")");
        }
        catch
        {
            ErrorLbl.Text = "<h3 class='red'>There seems to have been an
error adding a new size!</h3>";
            ErrorLbl.Visible = true;
            return;
        }

        GetSupplyData();
    }
}

```

}

AddNewItem.aspx

מיקום: ComputersProjectNew/AddNewItem.aspx

מיקום קוד מאחור: ComputersProjectNew/AddNewItem.aspx.cs

קוד aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="AddNewItem.aspx.cs" Inherits="AddNewItem" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>New Product Page</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2">
        <div class="inner">
            <div class="box">
                <div class="content">
                    <header class="align-center">
                        <h2>New Product Form</h2>
                    </header>
                    <form id="form1" runat="server" method="post">
                        <div class="row uniform">
                            <div class="6u 12u$(small)">
                                <h3>Product
Name:</h3>
                                </div>
                            <div class="6u$ 12u$(small)">
                                <asp:TextBox id="ProductNameTextBox"
runat="server"></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="ProductNameTextBox" Display="None"
ID="RequiredFieldValidator1" runat="server" ErrorMessage="Product name is
required"></asp:RequiredFieldValidator>
                                <asp:RegularExpressionValidator
ValidationExpression="^[a-zA-Z][a-zA-Z ]{0,98}[a-zA-Z]$"
ControlToValidate="ProductNameTextBox" Display="None"
ID="RegularExpressionValidator1" runat="server" ErrorMessage="Product name must
be up to 100 characters long!"></asp:RegularExpressionValidator>
                            </div>
                            <div class="6u 12u$(small)">
                                <h3>Company:</h3>
                            </div>
                            <div class="6u$ 12u$(small)">
                                <asp:TextBox id="CompanyTextBox"
runat="server"></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="CompanyTextBox" Display="None" ID="RequiredFieldValidator2"
runat="server" ErrorMessage="Company is required"></asp:RequiredFieldValidator>
                                <asp:RegularExpressionValidator
ValidationExpression="^[a-zA-Z][a-zA-Z ]{0,98}[a-zA-Z]$"
ControlToValidate="CompanyTextBox" Display="None"
ID="RegularExpressionValidator2" runat="server" ErrorMessage="Company must be
up to 100 characters long!"></asp:RegularExpressionValidator>
                            </div>
                            <div class="6u 12u$(small)">
```

```

        <h3>Size:</h3>
        </div>
        <div class="6u$ 12u$(small)">
            <asp:TextBox id="SizeTextBox"
runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator
ControlToValidate="SizeTextBox" Display="None" ID="RequiredFieldValidator4"
runat="server" ErrorMessage="CVV is required"></asp:RequiredFieldValidator>
            <asp:RangeValidator id="RangeValidator3"
runat="server" Type="Integer" ControlToValidate="SizeTextBox" MinimumValue="30"
MaximumValue="55" ErrorMessage="Invalid size, out of range (must be between 30
and 55)" Display="None"/>
        </div>

        <div class="6u 12u$(small)">
            <h3>Price:</h3>
        </div>
        <div class="6u$ 12u$(small)">
            <asp:TextBox id="PriceTextBox"
runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator
ControlToValidate="PriceTextBox" Display="None" ID="RequiredFieldValidator3"
runat="server" ErrorMessage="Expiration date is
required"></asp:RequiredFieldValidator>
            <asp:RangeValidator id="RangeValidator2"
runat="server" Type="Double" ControlToValidate="PriceTextBox"
MinimumValue="0.01" MaximumValue="9999.99" ErrorMessage="Invalid price! Price
must be a decimal value between 0.01 and 9999.99" Display="None"/>
        </div>

        <div class="6u 12u$(small)">
            <h3>Product Image:</h3>
        </div>
        <div class="6u$ 12u$(small)">
            <asp:FileUpload ID="ImageUploader"
runat="server" AllowMultiple="false"/>
            <asp:RequiredFieldValidator
ID="RequiredFieldValidator5" runat="server" ControlToValidate="ImageUploader"
Display="None" ErrorMessage="Product image is
required"></asp:RequiredFieldValidator>
            <asp:RegularExpressionValidator
ID="RegularExpressionValidator3" runat="server"
ControlToValidate="ImageUploader" ValidationExpression="^(([a-zA-
Z:]|(\{\2\}\w+)|\$?)(\w[\w.]*)(.jpe?g|.JPE?G|.png|.PNG)$" ErrorMessage="You
may only upload .jpg and .png files"
Display="None"></asp:RegularExpressionValidator>
        </div>

        <div class="12u$"><asp:ValidationSummary
runat="server" ShowMessageBox="true" ShowSummary="false" /></div>

        <div class="12u$">
            <div class="align-center">
                <asp:Button runat="server"
ID="SubmitDataButton" OnClick="Submit_Click" Text="Add Item"/>
            </div>
        </div>

        <div class="12u$">
            <div class="align-center">
                <asp:Label ID="ResultLbl"
runat="server" Visible="false"></asp:Label>

```

```

                </div>
            </div>
        </form>
    </div>
</div>
</section>
</asp:Content>

```

קוד:C#

```

using System;
using System.Data;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class AddNewItem : System.Web.UI.Page
{
    public const int AdminReq = 2;

    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["username"] == null) // בודק אם המשתמש נכנס האן לדף המשתמש את מעביר
            Response.Redirect("NoAccessPage.aspx"); // כנisa haan l'daf hemshtmesh at mevir

        int permLvl = int.Parse(Session["userPermLvl"].ToString()); // מתקבל מהסשן המשתמש של הגישה הרשותה

        if (permLvl < AdminReq) // המستخدم של הגישה הרשות את בודק
            Response.Redirect("NoAccessPage.aspx"); // כנisa haan l'daf hemshtmesh at mevir
    }

    protected void Submit_Click(object sender, EventArgs e)
    {
        string productName = Request.Form[ProductNameTextBox.UniqueID];
        string company = Request.Form[CompanyTextBox.UniqueID];
        string size = Request.Form[SizeTextBox.UniqueID];
        string price = Request.Form[PriceTextBox.UniqueID];

        if (!MyAdoHelper.IsExist("MyDatabase.mdf", "SELECT * FROM tblProduct
WHERE name= '" + productName + "'"))
        {
            MyAdoHelper.DoQuery("MyDatabase.mdf", "INSERT INTO tblProduct
(name, company, imageName) VALUES ('" + productName + "', '" + company + ',
'not_found.png')");

        }
        else
        {
            ResultLbl.Text = "<h3 class='red'>A product with this name already
exists!</h3>";
            ResultLbl.Visible = true;
            return;
        }
    }
}

```

```
    }

    DataTable productDt = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf",
"SELECT * FROM tblProduct WHERE name='" + productName + "'");
    int productId = int.Parse(productDt.Rows[0]["productId"].ToString());

    MyAdoHelper.DoQuery("MyDatabase.mdf", "INSERT INTO tblSupply VALUES (" +
+ productId + ", " + size + ", 0, " + price + ")");

    if (ImageUploader.HasFile)
    {
        string ext = Path.GetExtension(ImageUploader.PostedFile.FileName);
        ImageUploader.SaveAs(Server.MapPath("~/productImages/product" +
productId + ext.ToLower()));
        MyAdoHelper.DoQuery("MyDatabase.mdf", "UPDATE tblProduct SET
imageName='product" + productId + ext.ToLower() + "' WHERE productId=" +
productId);
    }

    ResultLbl.Text = "<h3 class='green'>Item successfully added!</h3>";
    ResultLbl.Visible = true;
}
}
```

SalesReportPage.aspx

מיקום: ComputersProjectNew/SalesReportPage.aspx

מיקום קוד מאחור: ComputersProjectNew/SalesReportPage.aspx.cs

קוד aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="SalesReportPage.aspx.cs"
Inherits="SalesReportPage" %>

<%@ Register Assembly="System.Web.DataVisualization, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35"
Namespace="System.Web.UI.DataVisualization.Charting" TagPrefix="asp" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>Sales Report Page</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2">
        <div class="inner">
            <div class="box">
                <div class="content">
                    <header class="align-center">
                        <h2>Sales Report</h2>
                    </header>
                    <form id="ReportForm" runat="server">
                        <div class="row uniform">
                            <div class="4u 12u$(small)">
                                <h3>From: </h3>
                                <asp:TextBox ID="StartDateTextBox"
runat="server" TextMode="Date"></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="StartDateTextBox" Display="None"
ID="RequiredFieldValidator1" runat="server" ErrorMessage="Start date is
required"></asp:RequiredFieldValidator>
                                <asp:RangeValidator
id="StartDateRangeValidator" runat="server" Type="Date"
ControlToValidate="StartDateTextBox" ErrorMessage="Invalid start date, out of
range" Display="None"/>
                            </div>
                            <div class="4u 12u$(small)">
                                <h3>To: </h3>
                                <asp:TextBox ID="EndDateTextBox"
runat="server" TextMode="Date"></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="EndDateTextBox" Display="None" ID="RequiredFieldValidator2"
runat="server" ErrorMessage="End date is
required"></asp:RequiredFieldValidator>
                                <asp:RangeValidator
id="EndDateRangeValidator" runat="server" Type="Date"
ControlToValidate="EndDateTextBox" ErrorMessage="Invalid end date, out of
range" Display="None"/>
                            </div>
                            <div class="4u$ 12u$(small)">
                                <asp:Button ID="GetStatsButton"
runat="server" OnClick="GetStatsButton_Click" Text="Get Data"/>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </section>
</div>
</body>
</html>
```

```
</div>

<div class="12u$"><asp:ValidationSummary
runat="server" ShowMessageBox="true" ShowSummary="false" /></div>

<div class="12u$">
    <div class="align-center">
        <asp:Label ID="ErrorLbl" runat="server"
Visible="false"></asp:Label>
    </div>
</div>

<br />
<br />
<br />
<br />

<div class="6u 12u$(small)">
    <div class="align-center">
        <h3>Best Selling Shoes: </h3>
        <asp:Chart ID="BestShoeChart"
runat="server" Width="450px" Height="450px">
            <Legends>
                <asp:Legend Alignment="Center"
Docking="Bottom" IsTextAutoFit="True" Name="Default" LegendStyle="Row" />
            </Legends>
            <Series>
                <asp:Series Name="Default"
IsVisibleInLegend="false"></asp:Series>
            </Series>
            <ChartAreas>
                <asp:ChartArea
Name="ChartArea5" BorderWidth="0"></asp:ChartArea>
            </ChartAreas>
            </asp:Chart>
        </div>
    </div>

    <div class="6u 12u$(small)">
        <div class="align-center">
            <h3>Worst Selling Shoes: </h3>
            <asp:Chart ID="WorstShoeChart"
runat="server" Width="450px" Height="450px">
                <Legends>
                    <asp:Legend Alignment="Center"
Docking="Bottom" IsTextAutoFit="True" Name="Default" LegendStyle="Row" />
                </Legends>
                <Series>
                    <asp:Series Name="Default"
IsVisibleInLegend="false"></asp:Series>
                </Series>
                <ChartAreas>
                    <asp:ChartArea
Name="ChartArea6" BorderWidth="0"></asp:ChartArea>
                </ChartAreas>
                </asp:Chart>
            </div>
        </div>

        <br />
        <br />
        <br />
```

```
<br />

<div class="6u 12u$(small)">
    <div class="align-center">
        <h3>Most Avid Customers: </h3>
        <asp:Chart ID="BestCustomerChart"
runat="server" Width="450px" Height="450px">
            <Legends>
                <asp:Legend Alignment="Center" />
            </Legends>
            <Series>
                <asp:Series Name="Default"
IsVisibleInLegend="false"></asp:Series>
            </Series>
            <ChartAreas>
                <asp:ChartArea
Name="ChartArea3" BorderWidth="0"></asp:ChartArea>
            </ChartAreas>
        </asp:Chart>
    </div>
</div>

<div class="6u 12u$(small)">
    <div class="align-center">
        <h3>Least Avid Customers: </h3>
        <asp:Chart ID="WorstCustomerChart"
runat="server" Width="450px" Height="450px">
            <Legends>
                <asp:Legend Alignment="Center" />
            </Legends>
            <Series>
                <asp:Series Name="Default"
IsVisibleInLegend="false"></asp:Series>
            </Series>
            <ChartAreas>
                <asp:ChartArea
Name="ChartArea4" BorderWidth="0"></asp:ChartArea>
            </ChartAreas>
        </asp:Chart>
    </div>
</div>

<br />
<br />
<br />
<br />

<div class="6u 12u$(small)">
    <div class="align-center">
        <h3>Purchase Distribution By Companies:
</h3>
        <asp:Chart ID="CompanySalesChart"
runat="server" Width="450px" Height="450px">
            <Legends>
                <asp:Legend Alignment="Center" />
            </Legends>
            <Series>
                <asp:Series
Name="Series1"></asp:Series>
            </Series>
        </asp:Chart>
    </div>
</div>
```

```
</Series>
<ChartAreas>
    <asp:ChartArea
Name="ChartArea1" BorderWidth="0"></asp:ChartArea>
        </ChartAreas>
    </asp:Chart>
</div>
</div>

<div class="6u$ 12u$(small)">
    <div class="align-center">
        <h3>Purchase Distribution By Shoe Size
Per Shoe: </h3>
        <asp:Chart ID="ShoeSizeChart"
runat="server" Width="450px" Height="450px">
            <Legends>
                <asp:Legend Alignment="Center"
Docking="Bottom" IsTextAutoFit="False" Name="Default" LegendStyle="Row" />
            </Legends>
            <Series>
                <asp:Series
Name="Series2"></asp:Series>
            </Series>
            <ChartAreas>
                <asp:ChartArea
Name="ChartArea2" BorderWidth="0"></asp:ChartArea>
                    </ChartAreas>
                </asp:Chart>
                <div class="12u$">
                    <div class="align-center">
                        <asp:DropDownList
ID="ProductDDL" runat="server" Width="300px"></asp:DropDownList>
                        <br />
                        <asp:Button
ID="ProductSelectButton" runat="server" Text="Load Chart"
OnClick="ProductSelectButton_Click"/>
                    </div>
                    </div>
                </div>
            </form>
        </div>
    </div>
</section>
</asp:Content>
```

:C# קוד

```

using System;
using System.Data;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.DataVisualization;
using System.Web.UI.DataVisualization.Charting;
using System.Drawing;

public partial class SalesReportPage : System.Web.UI.Page
{
    public const int AdminReq = 2;
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["username"] == null)// בודק אם המשתמש נכנס האין לדף המשתמש את מעביר
            Response.Redirect("NoAccessPage.aspx"); // כנראה לא יתאפשר לטעות

        int permLvl = int.Parse(Session["userPermLvl"].ToString()); // את מקבל מהסשן המשתמש של הגישה הרשות

        if (permLvl < AdminReq) // המשתמש הרשות את בודק
            Response.Redirect("NoAccessPage.aspx"); // כנראה לא יתאפשר לטעות

        DataTable orderTbl = MyAdoHelper.GetOrderTbl();
        orderTbl.DefaultView.Sort = "datetime";
        orderTbl = orderTbl.DefaultView.ToTable();
        DateTime startDate = (DateTime)orderTbl.Rows[0]["datetime"];
        DateTime endDate = DateTime.Today;
        ViewState["startDate"] = startDate;
        ViewState["endDate"] = endDate;
        StartDateTextBox.Text = startDate.ToString("yyyy-MM-dd");
        EndDateTextBox.Text = endDate.ToString("yyyy-MM-dd");
        StartDateRangeValidator.MinimumValue =
            startDate.ToString("yyyy/MM/dd");
        StartDateRangeValidator.MaximumValue = endDate.ToString("yyyy/MM/dd");
        EndDateRangeValidator.MinimumValue = startDate.ToString("yyyy/MM/dd");
        EndDateRangeValidator.MaximumValue = endDate.ToString("yyyy/MM/dd");

        if (!IsPostBack)
        {
            BindAllCharts(startDate, endDate);
            BindProductDropDownList();
        }
    }

    protected void GetStatsButton_Click(object sender, EventArgs e)
    {
        DateTime startDate =
DateTime.Parse(Request.Form[StartDateTextBox.UniqueID]);
        DateTime endDate =
DateTime.Parse(Request.Form[EndDateTextBox.UniqueID]);
        if(startDate >= endDate)
        {
            ErrorLbl.Text = "<h3 class='red'>The starting date needs to be before the ending date!</h3>";
            ErrorLbl.Visible = true;
            return;
        }
    }
}

```

```

        }

        ViewState[ "startDate" ] = startDate;
        ViewState[ "endDate" ] = endDate;
        BindAllCharts(startDate, endDate);
    }

    protected void BindAllCharts(DateTime startDate, DateTime endDate)
    {
        DataTable productTbl = MyAdoHelper.GetProductTbl();

        Dictionary<int, int> sizeDict =
SaleStatisticsHelper.GetSizePercents(int.Parse(productTbl.Rows[0][ "productId" ].ToString()), startDate, endDate);
        Dictionary<string, int> companyDict =
SaleStatisticsHelper.GetCompanyPercents(startDate, endDate);
        Dictionary<string, double> bestCustomerDict =
SaleStatisticsHelper.GetCustomerPurchases(5, true, startDate, endDate);
        Dictionary<string, double> worstCustomerDict =
SaleStatisticsHelper.GetCustomerPurchases(5, false, startDate, endDate);
        Dictionary<string, int> bestItemDict =
SaleStatisticsHelper.GetItemSales(3, true, startDate, endDate);
        Dictionary<string, int> worstItemDict =
SaleStatisticsHelper.GetItemSales(3, false, startDate, endDate);

        BindChart(ref ShoeSizeChart, sizeDict, SeriesChartType.Pie, false, "", "");
        BindChart(ref CompanySalesChart, companyDict, SeriesChartType.Pie, false, "", "");
        BindChart(ref BestCustomerChart, bestCustomerDict,
SeriesChartType.Column, true, "Username", "Total Spent");
        BindChart(ref WorstCustomerChart, worstCustomerDict,
SeriesChartType.Column, true, "Username", "Total Spent");
        BindChart(ref BestShoeChart, bestItemDict, SeriesChartType.Column,
true, "Product Name", "Total Sold");
        BindChart(ref WorstShoeChart, worstItemDict, SeriesChartType.Column,
true, "Product Name", "Total Sold");
    }

    protected void BindChart<TKey, TValue>(ref Chart chart, Dictionary<TKey,
TValue> dataDict, SeriesChartType chartType, bool reverse, string titleX,
string titleY)
    {
        if(dataDict.Count < 1)
        {
            TextAnnotation errorAnnotation = new TextAnnotation();
            errorAnnotation.Text = "No data for this period!";
            errorAnnotation.X = 18;
            errorAnnotation.Y = 30;
            errorAnnotation.Font = new Font("Sans Serif", 18, FontStyle.Bold);
            errorAnnotation.ForeColor = Color.Red;
            chart.Annotations.Add(errorAnnotation);
            chart.Series[0].IsVisibleInLegend = false;
            chart.ChartAreas[0].Visible = false;
            return;
        }

        TKey[] arrX = dataDict.Keys.ToArray();
        TValue[] arrY = dataDict.Values.ToArray();

        if (reverse)
        {
    
```

```
        Array.Reverse(arrX);
        Array.Reverse(arrY);
    }

    chart.ChartAreas[0].AxisX.Title = titleX;
    chart.ChartAreas[0].AxisY.Title = titleY;
    chart.ChartAreas[0].AxisX.TitleFont = new Font("Sans Serif", 10,
FontStyle.Bold);
    chart.ChartAreas[0].AxisY.TitleFont = new Font("Sans Serif", 10,
FontStyle.Bold);
    chart.Series[0].Points.DataBindXY(arrX, arrY);
    chart.Series[0].BorderWidth = 10;
    chart.Series[0].ChartType = chartType;
}

protected void BindProductDropDownList()
{
    DataTable productTbl = MyAdoHelper.GetProductTbl();

    foreach(DataRow dataRow in productTbl.Rows)
    {
        ListItem item = new ListItem(dataRow["name"].ToString(),
dataRow["productId"].ToString());
        ProductDDL.Items.Add(item);
    }
}

protected void ProductSelectButton_Click(object sender, EventArgs e)
{
    int productId = int.Parse(Request.Form[ProductDDL.UniqueID]);

    BindAllCharts((DateTime)ViewState["startDate"],
(DateTime)ViewState["endDate"]);

    Dictionary<int, int> sizeDict =
SaleStatisticsHelper.GetSizePercents(productId,
(DateTime)ViewState["startDate"], (DateTime)ViewState["endDate"]);

    BindChart(ref ShoeSizeChart, sizeDict, SeriesChartType.Pie, false, "", "");
}
}
```

NewCreditCardPage.aspx

מיקום: ComputersProjectNew/NewCreditCardPage.aspx

מיקום קוד מאחור: ComputersProjectNew/NewCreditCardPage.aspx.cs

קוד aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="NewCreditCardPage.aspx.cs"
Inherits="NewCreditCardPage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>New Credit Card Page</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2">
        <div class="inner">
            <div class="box">
                <div class="content">
                    <header class="align-center">
                        <h2>New Credit Card:</h2>
                    </header>
                    <form id="form1" runat="server" method="post">
                        <div class="row uniform">
                            <div class="6u 12u$(small)">
                                <h3>Card Number (No
Spaces):</h3>
                                </div>
                            <div class="6u$ 12u$(small)">
                                <asp:TextBox id="CardNumTextBox"
runat="server"></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="CardNumTextBox" Display="None" ID="RequiredFieldValidator1"
runat="server" ErrorMessage="Card number is
required"></asp:RequiredFieldValidator>
                                <asp:RegularExpressionValidator
ValidationExpression="^[0-9]{16}$" ControlToValidate="CardNumTextBox"
Display="None" ID="RegularExpressionValidator1" runat="server"
ErrorMessage="Card number must be all digits and 16 characters
long"></asp:RegularExpressionValidator>
                                </div>
                            <div class="6u 12u$(small)">
                                <h3>CVV:</h3>
                            </div>
                            <div class="6u$ 12u$(small)">
                                <asp:TextBox id="CVVTextBox"
runat="server"></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="CVVTextBox" Display="None" ID="RequiredFieldValidator4"
runat="server" ErrorMessage="CVV is required"></asp:RequiredFieldValidator>
                                <asp:RegularExpressionValidator
ValidationExpression="^[0-9]{3}$" ControlToValidate="CVVTextBox" Display="None"
ID="RegularExpressionValidator3" runat="server" ErrorMessage="CVV must be 3
non-0 digits"></asp:RegularExpressionValidator>
                            </div>
                            <div class="6u 12u$(small)">
                                <h3>Expiration Date:</h3>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </section>
    </div>
</asp:Content>

```

```

        </div>
        <div class="6u$ 12u$(small)">
            <asp:TextBox id="ExpireDateTextBox"
runat="server" TextMode="Date"></asp:TextBox>
            <asp:RequiredFieldValidator
ControlToValidate="ExpireDateTextBox" Display="None"
ID="RequiredFieldValidator8" runat="server" ErrorMessage="Expiration date is
required"></asp:RequiredFieldValidator>
            <asp:RangeValidator
id="ExpireDateRangeValidator" runat="server" Type="Date"
ControlToValidate="ExpireDateTextBox" MinimumValue="" MaximumValue=""
ErrorMessage="Invalid expiration date, out of range" Display="None"/>
        </div>

        <div class="12u$"><asp:ValidationSummary
runat="server" ShowMessageBox="true" ShowSummary="false" /></div>

        <div class="12u$">
            <div class="align-center">
                <asp:Button runat="server"
ID="SubmitDataButton" OnClick="Submit_Click" Text="Add Card"/>
            </div>
        </div>

        <div class="12u$"><div class="align-center"><h3
class="red"><%= errorMsg %></h3></div></div>
            </div>
        </form>
    </div>
</div>
</section>
</asp:Content>

```

:C# טיפ

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class NewCreditCardPage : System.Web.UI.Page
{
    public string errorMsg;
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["username"] == null)
            Response.Redirect("NoAccessPage.aspx");

        ExpireDateRangeValidator.MinimumValue =
DateTime.Today.AddMonths(1).ToString("yyyy/MM/dd");
        ExpireDateRangeValidator.MaximumValue =
DateTime.Today.AddYears(15).ToString("yyyy/MM/dd");
    }
    protected void Submit_Click(object sender, EventArgs e)
    {
        string cardNum = Request.Form[CardNumTextBox.UniqueID];

```

```
string cvv = Request.Form[CVVTextBox.UniqueID];
string expireDate = Request.Form[ExpireDateTextBox.UniqueID];

OnlineServiceReference.LuhnCheckerSoapClient client = new
OnlineServiceReference.LuhnCheckerSoapClient();

bool cardValid = client.CheckCC(cardNum).CardValid;
string company = client.CheckCC(cardNum).CardType;

if (cardValid == false || company.Equals("Unknown"))
{
    errorMsg = "The credit card you entered is invalid!";

    return;
}

DateTime expireDateNormalized = DateTime.Parse(expireDate);
expireDateNormalized = new DateTime(expireDateNormalized.Year,
expireDateNormalized.Month, 1);

expireDate = expireDateNormalized.ToString("yyyy-MM-dd");

if (MyAdoHelper.IsExist("MyDatabase.mdf", "SELECT * FROM tblCard WHERE
cardNum=" + cardNum + ""))
{
    errorMsg = "This card number is taken!";
}
else
{
    bool success = MyAdoHelper.CreateNewCard(cardNum,
Session["username"].ToString(), company, cvv, expireDate);
    if (success)
        Response.Redirect("ManageCardPage.aspx");
    else
        errorMsg = "There appears to have been an error!";
}
}
```

MemberManagementPage.aspx

מיקום: ComputersProjectNew/MemberManagementPage.aspx

מיקום קוד מאחור: ComputersProjectNew/MemberManagementPage.aspx.cs

קוד aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="MemberManagementPage.aspx.cs"
Inherits="MemberManagementPage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <title>Member Management Page</title>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <section class="wrapper style2" id="three">
        <div class="inner">
            <header class="align-center">
                <h2>Member Management Page</h2>
            </header>
            <div class="align-center">
                <asp:Label ID="ErrorLbl" runat="server"
Visible="false"></asp:Label>
            </div>
        </div>
        <form id="form1" runat="server">
            <div class="inner">
                <div class="box" style="background:#f2f2f2 !important">
                    <div class="content">
                        <div class="row uniform">
                            <div class="4u 12u$(small)">
                                <h3>Select A Type:</h3>
                            </div>
                            <div class="4u 12u$(small)">
                                <asp:DropDownList ID="TypeDDL" runat="server"
CssClass="align-center">
                                    <asp:ListItem Text="Meeting"
Value="meeting"></asp:ListItem>
                                    <asp:ListItem Text="Group"
Value="group"></asp:ListItem>
                                </asp:DropDownList>
                            </div>
                            <div class="4u$ 12u$(small)"></div>
                            <div class="4u 12u$(small)">
                                <h3>Group/Meeting ID:</h3>
                            </div>
                            <div class="4u 12u$(small)">
                                <asp:TextBox ID="IDTextBox" runat="server"
TextMode="Number"></asp:TextBox>
                                <asp:RequiredFieldValidator
ControlToValidate="IDTextBox" Display="None" ID="RequiredFieldValidator1"
runat="server" ValidationGroup="LoadMembersValidationGroup"
ErrorMessage="Group/meeting ID is required"></asp:RequiredFieldValidator>
                                <asp:RangeValidator ID="RangeValidator1"
runat="server" ControlToValidate="IDTextBox" Type="Integer" MinimumValue="1"
MaximumValue="2147483646" ValidationGroup="LoadMembersValidationGroup"
ErrorMessage="Please enter a valid group/meeting ID"></asp:RangeValidator>
                            </div>
                        </div>
                    </div>
                </div>
            </form>
        </div>
    </section>
</div>
```

```

        </div>
        <div class="4u$ 12u$(small)"></div>

        <div class="align-center">
            <asp:Button ID="TypeButton" runat="server"
Text="Load Members" OnClick="TypeButton_Click"
ValidationGroup="LoadMembersValidationGroup"/>
        </div>
    </div>
</div>
<br />
<br />
<br />

<asp:GridView ID="MembersGridView" runat="server"
AutoGenerateColumns="false" HorizontalAlign="Center" AllowPaging="true"
PageSize="5" OnPageIndexChanging="MembersGridView_PageIndexChanging"
OnRowDeleting="MembersGridView_RowDeleting"
OnRowCommand="MembersGridView_RowCommand" ShowFooter="true" CaptionAlign="Top">
    <PagerStyle HorizontalAlign="Center"/>

    <Columns>
        <asp:TemplateField HeaderText="Controls">
            <ItemTemplate>
                <asp:Button ID="RemoveButton" runat="server"
Text="Remove" CommandName="Delete"/>
            </ItemTemplate>
            <FooterTemplate>
                <asp:Button ID="AddNewButton" runat="server"
Text="Add" CommandName="AddNew" ValidationGroup="AddNewMemberValidationGroup"/>
            </FooterTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Username">
            <ItemTemplate>
                <asp:Label ID="UsernameLbl" runat="server"
Text='<%# Bind("username") %>'></asp:Label>
            </ItemTemplate>
            <FooterTemplate>
                <asp:TextBox ID="UsernameTextBox"
runat="server"></asp:TextBox>
                <asp:RequiredFieldValidator
ControlToValidate="UsernameTextBox" Display="None" ID="RequiredFieldValidator1"
runat="server" ValidationGroup="AddNewMemberValidationGroup"
ErrorMessage="Username is required"></asp:RequiredFieldValidator>
                <asp:RegularExpressionValidator
ValidationExpression="^\w{5,50}$" ControlToValidate="UsernameTextBox"
Display="None" ID="RegularExpressionValidator1" runat="server"
ValidationGroup="AddNewMemberValidationGroup" ErrorMessage="Username must be
between 5 and 50 characters long"></asp:RegularExpressionValidator>
            </FooterTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Boulder Grade">
            <ItemTemplate>
                <asp:Label ID="BoulderGradeLbl" runat="server"
Text='<%# Bind("boulderGrade") %>'></asp:Label>
            </ItemTemplate>
            <FooterTemplate>
                <asp:Label ID="BoulderGradeLbl" runat="server"
Text="-"></asp:Label>
            </FooterTemplate>
        </asp:TemplateField>
    </Columns>

```

```

        </asp:TemplateField>
        <asp:TemplateField HeaderText="Wall Grade">
            <ItemTemplate>
                <asp:Label ID="WallGradeLbl" runat="server"
Text='<%# Bind("wallGrade") %>'></asp:Label>
            </ItemTemplate>
            <FooterTemplate>
                <asp:Label ID="WallGradeLbl" runat="server"
Text="-"></asp:Label>
            </FooterTemplate>
        </asp:TemplateField>
    </Columns>
</asp:GridView>
<asp:ValidationSummary ID="ValidationSummary1" runat="server"
ShowMessageBox="true" ShowSummary="false"
ValidationGroup="AddNewMemberValidationGroup"/>
<asp:ValidationSummary ID="ValidationSummary2" runat="server"
ShowMessageBox="true" ShowSummary="false"
ValidationGroup="LoadMembersValidationGroup"/>
</form>
</section>
</asp:Content>

```

C# קוד:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Globalization;

public partial class MemberManagementPage : System.Web.UI.Page
{
    const int AdminReq = 3, CoachReq = 1;

    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["username"] == null)// בודק אם המשתמש רשום
            Response.Redirect("NoAccessPage.aspx"); // מעביר את המשתמש לדף האין כניסה

        int permLvl = int.Parse(Session["userPermLvl"].ToString()); // מקבל את הרשותת הגישה של המשתמש מהבסיס
        if (permLvl < CoachReq) // בודק את הרשותת הגישה של המשתמש
            Response.Redirect("NoAccessPage.aspx"); // מעביר את המשתמש לדף האין כניסה
    }

    protected void MembersGridView_PageIndexChanging(object sender,
GridViewEventArgs e)
    {
        MembersGridViewPageIndex = e.NewPageIndex;
        GetMembersData();
    }

    protected void MembersGridView_RowCommand(object sender,
GridViewEventArgs e)

```

```

{
    if (e.CommandName.ToLower().Equals("addnew"))
    {
        string type = ViewState["type"].ToString();
        string id = ViewState["id"].ToString();

        string username =
((TextBox)MembersGridView.FooterRow.FindControl("UsernameTextBox")).Text;

        DataTable userInfoTbl = MyAdoHelper.GetUserInfo(username);

        if(userInfoTbl.Rows.Count < 1)
        {
            ErrorLbl.Text = "<h3 class='red'>This user does not
exist!</h3>";
            ErrorLbl.Visible = true;
            return;
        }

        if (type.Equals("group"))
        {
            MyAdoHelper.DoQuery("MyDatabase.mdf", "INSERT INTO
tblGroupMembers VALUES (" + id + ", '" + username + "')");
        }
        else
        {
            MyAdoHelper.DoQuery("MyDatabase.mdf", "INSERT INTO
tblMeetingMembers VALUES (" + id + ", '" + username + "')");
        }

        GetMembersData();
    }
}

protected void MembersGridView_RowDeleting(object sender,
GridViewDeleteEventArgs e)
{
    GridViewRow DeletedRow = MembersGridView.Rows[e.RowIndex];

    string username = ((Label)DeletedRow.FindControl("UsernameLbl")).Text;

    string type = ViewState["type"].ToString();
    string id = ViewState["id"].ToString();

    if (type.Equals("group"))
    {
        MyAdoHelper.DoQuery("MyDatabase.mdf", "DELETE FROM tblGroupMembers
WHERE groupId=" + id + " AND username='" + username + "'");
    }
    else
    {
        MyAdoHelper.DoQuery("MyDatabase.mdf", "DELETE FROM
tblMeetingMembers WHERE meetingId=" + id + " AND username='" + username + "'");
    }

    GetMembersData();
}

protected void TypeButton_Click(object sender, EventArgs e)
{
    string type = Request.Form[TypeDDL.UniqueID];
    string id = Request.Form[IDTextBox.UniqueID];
}

```

```
if (type.Equals("meeting"))
{
    ViewState["type"] = "meeting";
    ViewState["id"] = id;
    GetMembersData();
    return;
}
if (type.Equals("group"))
{
    ViewState["type"] = "group";
    ViewState["id"] = id;
    GetMembersData();
    return;
}

ErrorLbl.Text = "<h3 class='red'>There seems to have been an
error!</h3>";
ErrorLbl.Visible = true;
MembersGridView.Visible = false;
}

protected void GetMembersData()
{
    string username = Session["username"].ToString();
    int permLvl = int.Parse(Session["userPermLvl"].ToString());

    string type = ViewState["type"].ToString();
    string id = ViewState["id"].ToString();

    if (!(type.Equals("group") || type.Equals("meeting")))
    {
        ErrorLbl.Text = "<h3 class='red'>There seems to have been an
error!</h3>";
        ErrorLbl.Visible = true;
        MembersGridView.Visible = false;
        return;
    }

    DataTable membersTbl;

    if (type.Equals("group"))
    {
        DataTable groupTbl = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf",
"SELECT * FROM tblClimbingGroups WHERE groupId=" + id);
        if (groupTbl.Rows.Count < 1)
        {
            ErrorLbl.Text = "<h3 class='red'>This group does not
exist!</h3>";
            ErrorLbl.Visible = true;
            MembersGridView.Visible = false;
            return;
        }
        if (!groupTbl.Rows[0][ "coachName"].ToString().Equals(username) &&
permLvl < AdminReq)
        {
            ErrorLbl.Text = "<h3 class='red'>You are not the coach of this
group!</h3>";
            ErrorLbl.Visible = true;
            MembersGridView.Visible = false;
            return;
        }
    }
}
```

```

        membersTbl = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT
* FROM tblGroupMembers WHERE groupId=" + id);
    }
    else
    {
        DataTable meetingTbl =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblGroupMeetings
WHERE meetingId=" + id);
        if (meetingTbl.Rows.Count < 1)
        {
            ErrorLbl.Text = "<h3 class='red'>This meeting does not
exist!</h3>";
            ErrorLbl.Visible = true;
            MembersGridView.Visible = false;
            return;
        }
        if (permLvl < AdminReq)
        {
            string groupId = meetingTbl.Rows[0][ "groupId" ].ToString();
            DataTable groupTbl =
MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT * FROM tblClimbingGroups
WHERE groupId=" + id);
            if (!groupTbl.Rows[0][ "coachName" ].ToString().Equals(username))
            {
                ErrorLbl.Text = "<h3 class='red'>You are not the coach of
the group to whom this meeting belongs!</h3>";
                ErrorLbl.Visible = true;
                MembersGridView.Visible = false;
                return;
            }
            membersTbl = MyAdoHelper.ExecuteDataTable("MyDatabase.mdf", "SELECT
* FROM tblMeetingMembers WHERE meetingId=" + id);
        }
    }

    DataColumn boulderGradeColumn = membersTbl.Columns.Add("boulderGrade",
typeof(string));
    DataColumn wallGradeColumn = membersTbl.Columns.Add("wallGrade",
typeof(string));

    if (membersTbl.Rows.Count > 0)
    {
        DataTable userInfoTbl = MyAdoHelper.GetUserInfoTbl();

        foreach (DataRow dataRow in membersTbl.Rows)
        {
            string dataRowUsername = dataRow[ "username" ].ToString();

            dataRow[ "boulderGrade" ] = userInfoTbl.Select("username='"
+ dataRowUsername + "'").First()["boulderGrade"];
            dataRow[ "wallGrade" ] = userInfoTbl.Select("username='"
+ dataRowUsername + "'").First()["wallGrade"];
        }

        MembersGridView.DataSource = membersTbl;
        MembersGridView.DataBind();
    }
    else
    {
        DataRow dummyRow = membersTbl.NewRow();
        dummyRow[ "username" ] = "No Data";
        dummyRow[ "boulderGrade" ] = "No Data";
    }
}

```

```
        dummyRow[ "wallGrade" ] = "No Data";
        membersTbl.Rows.Add(dummyRow);

        MembersGridView.DataSource = membersTbl;
        MembersGridView.DataBind();

        Button dummyRowButton =
(Button)MembersGridView.Rows[0].FindControl( "RemoveButton" );
        dummyRowButton.Enabled = false;
    }

    MembersGridView.Visible = true;
    MembersGridView.ShowFooter = true;
    MembersGridView.Caption = string.Format("<h3>Member Data For {0}<br/>{1}</h3><br/>", CultureInfo.CurrentCulture.TextInfo.ToTitleCase(type), id);

    ErrorLbl.Visible = false;
}
}
```

LogOut.aspx

מיקום: ComputersProjectNew/LogOut.aspx

מיקום קוד מאחור: ComputersProjectNew/LogOut.aspx.cs

קוד aspx:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="LogOut.aspx.cs" Inherits="LogOut" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
</asp:Content>
```

קוד C#:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class LogOut : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        בפעולה הSSIONים כל את מס'ם //Session.Abandon();

        Application.Lock();
        Application["userNum"] = (int)Application["userNum"] - 1;
        Application.UnLock();

        Response.Redirect("HomePage.aspx"); //הבית לדף המשתמש את מהזיר
    }
}
```

Web.config

מיקוד: ComputerProjectNew/Web.config

:XML קוד

```

<?xml version="1.0"?>
<!--
    For more information on how to configure your ASP.NET application, please
    visit
    https://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
    <connectionStrings>
        <add name="ConnectionString" connectionString="Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\MyDatabase.mdf;I
ntegrated Security=True" providerName="System.Data.SqlClient"/>
        <add name="constr" providerName="System.Data.SqlClient"
connectionString="Data Source=localhost;Initial Catalog=databasename;Integrated
Security=False;User Id=userid;Password=password;MultipleActiveResultSets=True"
/>
    </connectionStrings>
    <appSettings>
        <add key="ValidationSettings:UnobtrusiveValidationMode" value="None"/>
        <add key="ChartImageHandler" value="storage=file;timeout=20;" />
    </appSettings>
    <system.net>
        <mailSettings>
            <smtp deliveryMethod="Network">
                <network host="smtp.gmail.com" port="587" userName="email"
password="password" enableSsl="true"/>
            </smtp>
        </mailSettings>
    </system.net>
    <system.web>
        <httpHandlers>
            <add path="ChartImg.axd" verb="GET,HEAD,POST"
type="System.Web.UI.DataVisualization.Charting.ChartHttpHandler,
System.Web.DataVisualization, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35"
                validate="false" />
        </httpHandlers>
        <globalization enableClientBasedCulture="false" culture="en-EN"
uiCulture="en"/>
        <compilation debug="true" targetFramework="4.6.1">
            <assemblies>
                <add assembly="System.Windows.Forms, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
                <add assembly="System.Net, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B03F5F7F11D50A3A"/>
                <add assembly="System.Net.Http, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B03F5F7F11D50A3A"/>
                <add assembly="System.AppContext, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B03F5F7F11D50A3A"/>
                <add assembly="System.ComponentModel.Composition, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=B77A5C561934E089"/>
                <add assembly="System.IO.Compression, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
                <add assembly="System.IO.Compression.FileSystem, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=B77A5C561934E089"/>
            </assemblies>
        </compilation>
    </system.web>
</configuration>

```

```
<add assembly="System.Numerics, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
<add assembly="System.Web.DataVisualization, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
</assemblies>
</compilation>
<httpRuntime targetFramework="4.6.1"/>
<sessionState timeout="30"/>
<pages enableViewState="true">
<controls>
<add tagPrefix="asp"
namespace="System.Web.UI.DataVisualization.Charting"
assembly="System.Web.DataVisualization, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" />
</controls>
</pages>
</system.web>
<system.webServer>
<handlers>
<add name="ChartImg" verb="*" path="ChartImg.axd"
type="System.Web.UI.DataVisualization.Charting.ChartHttpHandler,
System.Web.DataVisualization, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" />
</handlers>
<validation validateIntegratedModeConfiguration="false" />
</system.webServer>
<system.codedom>
<compilers>
<compiler language="c#;cs;csharp" extension=".cs"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:default /nowarn:1659;1699;1701"/>
<compiler language="vb;vbs;visualbasic;vbscript" extension=".vb"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:default /nowarn:41008
#define:_MYTYPE="Web" /optionInfer+/">
</compilers>
</system.codedom>
<system.serviceModel>
<bindings>
<basicHttpBinding>
<binding name="ShoeCalculatorServiceSoap">
<binding name="LuhnCheckerSoap">
<security mode="Transport"/>
</binding>
<binding name="LuhnCheckerSoap1"/>
</basicHttpBinding>
<customBinding>
<binding name="LuhnCheckerSoap12">
<textMessageEncoding messageVersion="Soap12"/>
<httpsTransport/>
</binding>
</customBinding>
</bindings>
<client>
<endpoint address="http://localhost:51156/ShoeCalculatorService.asmx"
binding="basicHttpBinding" bindingConfiguration="ShoeCalculatorServiceSoap"
contract="MyServiceReference.ShoeCalculatorServiceSoap"
name="ShoeCalculatorServiceSoap"/>
```

```
    <endpoint
address="https://ws.cdyne.com/creditcardverify/luhnchecker.asmx"
binding="basicHttpBinding" bindingConfiguration="LuhnCheckerSoap"
contract="OnlineServiceReference.LuhnCheckerSoap" name="LuhnCheckerSoap"/>
</client>
</system.serviceModel>
<runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
        <dependentAssembly>
            <assemblyIdentity name="Newtonsoft.Json"
publicKeyToken="30ad4fe6b2a6aeed" culture="neutral"/>
            <bindingRedirect oldVersion="0.0.0.0-13.0.0.0" newVersion="13.0.0.0"/>
        </dependentAssembly>
        <dependentAssembly>
            <assemblyIdentity name="System.Net.Http"
publicKeyToken="b03f5f7f11d50a3a" culture="neutral"/>
            <bindingRedirect oldVersion="0.0.0.0-4.1.1.3" newVersion="4.1.1.3"/>
        </dependentAssembly>
    </assemblyBinding>
</runtime>
</configuration>
```

packages.config

מיקוד XML: computersProjectNew/packages.config

קוד XML

```

<?xml version="1.0" encoding="utf-8"?>
<packages>
    <package id="jose-jwt" version="2.3.0" targetFramework="net461" />
    <package id="Json.Net" version="1.0.33" targetFramework="net461" />
    <package id="Microsoft.CodeDom.Providers.DotNetCompilerPlatform" version="2.0.0" targetFramework="net461" />
    <package id="Microsoft.Extensions.Configuration" version="1.1.2" targetFramework="net461" />
    <package id="Microsoft.Extensions.Configuration.Abstractions" version="1.1.2" targetFramework="net461" />
    <package id="Microsoft.Extensions.Configuration.FileExtensions" version="1.1.2" targetFramework="net461" />
    <package id="Microsoft.Extensions.Configuration.Json" version="1.1.2" targetFramework="net461" />
    <package id="Microsoft.Extensions.DependencyInjection.Abstractions" version="1.1.1" targetFramework="net461" />
    <package id="Microsoft.Extensions.FileProviders.Abstractions" version="1.1.1" targetFramework="net461" />
    <package id="Microsoft.Extensions.FileProviders.Physical" version="1.1.1" targetFramework="net461" />
    <package id="Microsoft.Extensions.FileSystemGlobbing" version="1.1.1" targetFramework="net461" />
    <package id="Microsoft.Extensions.Logging" version="1.1.2" targetFramework="net461" />
    <package id="Microsoft.Extensions.Logging.Abstractions" version="1.1.2" targetFramework="net461" />
    <package id="Microsoft.Extensions.Primitives" version="1.1.1" targetFramework="net461" />
    <package id="Microsoft.IdentityModel.Logging" version="1.1.2" targetFramework="net461" />
    <package id="Microsoft.IdentityModel.Tokens" version="5.1.2" targetFramework="net461" />
    <package id="Microsoft.NETCore.Platforms" version="1.1.0" targetFramework="net461" />
    <package id="Microsoft.Win32.Primitives" version="4.3.0" targetFramework="net461" />
    <package id="Nager.Country" version="3.1.1" targetFramework="net461" />
    <package id="NETStandard.Library" version="1.6.1" targetFramework="net461" />
    <package id="Newtonsoft.Json" version="13.0.1" targetFramework="net461" />
    <package id="System.AppContext" version="4.3.0" targetFramework="net461" />
    <package id="System.Collections" version="4.3.0" targetFramework="net461" />
    <package id="System.Collections.Concurrent" version="4.3.0" targetFramework="net461" />
    <package id="System.Collections.Immutable" version="1.2.0" targetFramework="net461" />
    <package id="System.ComponentModel" version="4.3.0" targetFramework="net461" />
    <package id="System.ComponentModel.Annotations" version="4.7.0" targetFramework="net461" />
    <package id="System.ComponentModel.Composition" version="4.7.0" targetFramework="net461" />
    <package id="System.Console" version="4.3.0" targetFramework="net461" />
    <package id="System.Diagnostics.Debug" version="4.3.0" targetFramework="net461" />

```

```
<package id="System.Diagnostics.Tools" version="4.3.0"
targetFramework="net461" />
<package id="System.Diagnostics.Tracing" version="4.3.0"
targetFramework="net461" />
<package id="System.Globalization" version="4.3.0" targetFramework="net461"
/>
<package id="System.Globalization.Calendars" version="4.3.0"
targetFramework="net461" />
<package id="System.IdentityModel.Tokens.Jwt" version="5.1.2"
targetFramework="net461" />
<package id="System.IO" version="4.3.0" targetFramework="net461" />
<package id="System.IO.Compression" version="4.3.0" targetFramework="net461"
/>
<package id="System.IO.Compression.ZipFile" version="4.3.0"
targetFramework="net461" />
<package id="System.IO.FileSystem" version="4.3.0" targetFramework="net461"
/>
<package id="System.IO.FileSystem.Primitives" version="4.3.0"
targetFramework="net461" />
<package id="System.Linq" version="4.3.0" targetFramework="net461" />
<package id="System.Linq.Expressions" version="4.3.0"
targetFramework="net461" />
<package id="System.Net.Http" version="4.3.4" targetFramework="net461" />
<package id="System.Net.Primitives" version="4.3.0" targetFramework="net461"
/>
<package id="System.Net.Sockets" version="4.3.0" targetFramework="net461" />
<package id="System.ObjectModel" version="4.3.0" targetFramework="net461" />
<package id="System.Reflection" version="4.3.0" targetFramework="net461" />
<package id="System.Reflection.Extensions" version="4.3.0"
targetFramework="net461" />
<package id="System.Reflection.Primitives" version="4.3.0"
targetFramework="net461" />
<package id="System.Resources.ResourceManager" version="4.3.0"
targetFramework="net461" />
<package id="System.Runtime" version="4.3.0" targetFramework="net461" />
<package id="System.Runtime.CompilerServices.Unsafe" version="4.3.0"
targetFramework="net461" />
<package id="System.Runtime.Extensions" version="4.3.0"
targetFramework="net461" />
<package id="System.Runtime.Handles" version="4.3.0" targetFramework="net461"
/>
<package id="System.Runtime.InteropServices" version="4.3.0"
targetFramework="net461" />
<package id="System.Runtime.InteropServices.RuntimeInformation"
version="4.3.0" targetFramework="net461" />
<package id="System.Runtime.Numerics" version="4.3.0"
targetFramework="net461" />
<package id="System.Security.Cryptography.Algorithms" version="4.3.0"
targetFramework="net461" />
<package id="System.Security.Cryptography.Encoding" version="4.3.0"
targetFramework="net461" />
<package id="System.Security.Cryptography.Primitives" version="4.3.0"
targetFramework="net461" />
<package id="System.Security.Cryptography.X509Certificates" version="4.3.0"
targetFramework="net461" />
<package id="System.Text.Encoding" version="4.3.0" targetFramework="net461"
/>
<package id="System.Text.Encoding.Extensions" version="4.3.0"
targetFramework="net461" />
<package id="System.Text.RegularExpressions" version="4.3.0"
targetFramework="net461" />
<package id="System.Threading" version="4.3.0" targetFramework="net461" />
```

```
<package id="System.Threading.Tasks" version="4.3.0" targetFramework="net461">
/>
<package id="System.Threading.Timer" version="4.3.0" targetFramework="net461">
/>
<package id="System.Xml.ReaderWriter" version="4.3.0"
targetFramework="net461" />
<package id="System.Xml.XDocument" version="4.3.0" targetFramework="net461">
/>
<package id="Twilio" version="5.70.1" targetFramework="net461" />
<package id="Vonage" version="5.9.5" targetFramework="net461" />
</packages>
```

MyAdoHelper.cs

מיקום: ComputersProjectNew/App_Code/MyAdoHelper.cs

קוד:C#

```

using System;
using System.Data;
using System.Configuration;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;

/// <summary>
/// Summary description for MyAdoHelper
/// פועלות זור לשימוש במסד נתונים מסווג
/// SQL SERVER
/// App_Data המסד ממוקם בתקיה
/// </summary>

public static class MyAdoHelper{

    // מקבל שם של קובץ מסד נתונים ומבצע התחברות אליו/
    public static SqlConnection ConnectToDb(string fileName){
        string path = HttpContext.Current.Server.MapPath("App_Data/" +
fileName); // מסדר את מיקום מסד הנתונים מהוורש ועד התקיה בה ממוקם המסד
        string connString = @"Data Source = (LocalDB)\MSSQLLocalDB;
AttachDbFilename = "+path+"; Integrated Security = True";

        SqlConnection conn = new SqlConnection(connString);
        return conn;
    }

    // הפעולה מקבלת שם מסד נתונים ומחרוזת מהיקה/ הוספה/ עדכו ומבצעת את הפקודה על המסד
    public static void DoQuery(string fileName, string sql){
        SqlConnection conn = ConnectToDb(fileName);
        conn.Open();
        SqlCommand com = new SqlCommand(sql, conn);
        com.ExecuteNonQuery();
        com.Dispose();
        conn.Close();
    }

    // הפעולה מקבלת מסלול אל מסד הנתונים ושאיילת העדכו, מבצעת עדכו במסד הנתונים
    public static int RowsAffected(string fileName, string sql){
        SqlConnection conn = ConnectToDb(fileName);
        conn.Open();
        SqlCommand com = new SqlCommand(sql, conn);
        int rowsA = com.ExecuteNonQuery();
        conn.Close();
        return rowsA;
    }
}

```

```

הפעולה מקבלת שם קובץ ומשפט בחירות נתון ומזהירה אם את הנתונים קיימים ושרר אחריה/
public static bool IsExist(string fileName, string sql){
    SqlConnection conn = ConnectToDb(fileName);
    conn.Open();
    SqlCommand com = new SqlCommand(sql, conn);
    SqlDataReader data = com.ExecuteReader();
    bool found;
    found=(bool) data.Read(); // הערך קיים
    if(data.Read()) // אם יש נתונים נוספים לקריאה יושם אמת אחרית שקר - הערך קיים
        found=false;
    data.Close();
    return found;
}

הפעולה מקבלת שאלה ומזהירה את טבלת הנתונים שモוחזרת מסך הנתונים ביצוע השאלה/
public static DataTable ExecuteDataTable(string fileName, string sql){
    SqlConnection conn = ConnectToDb(fileName);
    conn.Open();
    SqlDataAdapter tableAdapter = new SqlDataAdapter(sql,conn);
    DataTable dt = new DataTable();
    tableAdapter.Fill(dt);
    return dt;
}

מפעיל שאלה שלא מזהירה מידע כלשהו/
public static void ExecuteNonQuery(string fileName, string sql){
    SqlConnection conn = ConnectToDb(fileName);
    conn.Open();
    SqlCommand command = new SqlCommand(sql, conn);
    command.ExecuteNonQuery();
    conn.Close();
}

מזכיר טבלת נתונים מוכנה להדפסה מסך הנתונים/
public static string printDataTable(string fileName, string sql){
    DataTable dt = ExecuteDataTable(fileName, sql);
    string printStr = "<table border='1'>";

    foreach (DataRow row in dt.Rows){
        printStr += "<tr>";
        foreach (object myItemArray in row.ItemArray){
            printStr += "<td>" + myItemArray.ToString() + "</td>";
        }
        printStr += "</tr>";
    }
    printStr += "</table>";
    return printStr;
}

מקבלת שם משתמש ובודקת אם הוא קיים בטבלת המשתמשים/
public static int UserExists(string username)
{
    int result;
    bool found;

    SqlConnection cn = ConnectToDb("MyDatabase.mdf");
    SqlCommand cmd = new SqlCommand("getUsernameInfo", cn);

    cmd.CommandType = CommandType.StoredProcedure;

    cmd.Parameters.AddWithValue("@username", username);

    SqlTransaction trans = null;
}

```

```

try
{
    cn.Open();

    trans = cn.BeginTransaction();
    cmd.Transaction = trans;

    SqlDataReader data = cmd.ExecuteReader();
    found = data.Read(); // אם יש נתונים לקריאה יושם אמת אחרת שקר - הערך קיים
    if (found)
        result = 1;
    else
        result = 0;

    trans.Commit();
}
catch(Exception exp)
{
    trans.Rollback();
    result = 2;
}
finally
{
    cn.Close();
}

return result;
}

// יצירת משתמש חדש ומוחירה אם הפעולה הצליחה או לא
public static bool CreateNewUser(string username, string password, string
city, string gender, string birthdate, string boulderGrade, string wallGrade,
string email, string address)
{
    bool success = true;

    SqlConnection cn = ConnectToDb("MyDatabase.mdf");
    SqlCommand cmd = new SqlCommand("insertNewUser", cn);

    cmd.CommandType = CommandType.StoredProcedure;

    cmd.Parameters.AddWithValue("@username", username);
    cmd.Parameters.AddWithValue("@password", password);
    cmd.Parameters.AddWithValue("@city", city);
    cmd.Parameters.AddWithValue("@gender", gender);
    cmd.Parameters.AddWithValue("@birthdate", birthdate);
    cmd.Parameters.AddWithValue("@boulderGrade", boulderGrade);
    cmd.Parameters.AddWithValue("@wallGrade", wallGrade);
    cmd.Parameters.AddWithValue("@email", email);
    cmd.Parameters.AddWithValue("@address", address);

    SqlTransaction trans = null;

    try
    {
        cn.Open();

        trans = cn.BeginTransaction();
        cmd.Transaction = trans;

```

```

        cmd.ExecuteNonQuery();

        trans.Commit();
    }
    catch(Exception exp)
    {
        trans.Rollback();
        success = false;
    }
    finally
    {
        cn.Close();
    }

    return success;
}

// מוסיף שורה חדשה לטבלה כרטיסי האשראי ומחזירה אם הפעולה הצליחה או לא
public static bool CreateNewCard(string cardNum, string username, string
company, string cvv, string expireDate)
{
    bool success = true;

    SqlConnection cn = ConnectToDb("MyDatabase.mdf");
    SqlCommand cmd = new SqlCommand("insertNewCard", cn);

    cmd.CommandType = CommandType.StoredProcedure;

    cmd.Parameters.AddWithValue("@cardNum", cardNum);
    cmd.Parameters.AddWithValue("@username", username);
    cmd.Parameters.AddWithValue("@company", company);
    cmd.Parameters.AddWithValue("@CVV", cvv);
    cmd.Parameters.AddWithValue("@expireDate", expireDate);

    SqlTransaction trans = null;

    try
    {
        cn.Open();

        trans = cn.BeginTransaction();
        cmd.Transaction = trans;

        cmd.ExecuteNonQuery();

        trans.Commit();
    }
    catch (Exception exp)
    {
        trans.Rollback();
        success = false;
    }
    finally
    {
        cn.Close();
    }

    return success;
}

// בודקת אם קיים משתמש עם הסיסמה הנתונה ומחזירה אם כן או לא
public static int UserWithPasswordExists(string username, string password)

```

```

{
    int result;
    bool found;

    SqlConnection cn = ConnectToDb("MyDatabase.mdf");
    SqlCommand cmd = new SqlCommand("getUserAndPasswordInfo", cn);

    cmd.CommandType = CommandType.StoredProcedure;

    cmd.Parameters.AddWithValue("@username", username);
    cmd.Parameters.AddWithValue("@password", password);

    SqlTransaction trans = null;
    try
    {
        cn.Open();

        trans = cn.BeginTransaction();
        cmd.Transaction = trans;

        SqlDataReader data = cmd.ExecuteReader();
        found = data.Read(); // אם יש נתונים לקריאה יושם אמת אחרת שקר - הערך קיים

        data.Close();
        if (found)
            result = 1;
        else
            result = 0;

        trans.Commit();
    }
    catch (Exception exp)
    {
        trans.Rollback();
        result = 2;
    }
    finally
    {
        cn.Close();
    }

    return result;
}

// מוחזר את רמת ההרשאה של משתמש מסוים לפי קלט שם משתמש
public static int GetUserPermLvl(string username)
{
    int result = -1;

    SqlConnection cn = ConnectToDb("MyDatabase.mdf");
    SqlCommand cmd = new SqlCommand("getUserPermLvl", cn);

    cmd.CommandType = CommandType.StoredProcedure;

    cmd.Parameters.AddWithValue("@username", username);

    SqlTransaction trans = null;

    try
    {
        cn.Open();

```

```

        trans = cn.BeginTransaction();
        cmd.Transaction = trans;

        SqlDataReader data = cmd.ExecuteReader();
        while (data.Read())
            result = int.Parse(data["permLvl"].ToString());
        data.Close();
        trans.Commit();
    }
    catch(Exception exp)
    {
        trans.Rollback();
        result = -1;
    }
    finally
    {
        cn.Close();
    }

    return result;
}

//מחזירה את טבלת המשתמשים
public static DataTable GetUserInfoTbl()
{
    SqlConnection cn = ConnectToDb("MyDatabase.mdf");

    SqlCommand cmd = new SqlCommand("getUserInfoTbl", cn);
    cmd.CommandType = CommandType.StoredProcedure;

    SqlDataAdapter da = new SqlDataAdapter();
    DataTable dt = new DataTable();

    da.SelectCommand = cmd;

    SqlTransaction trans = null;

    try
    {
        cn.Open();

        trans = cn.BeginTransaction();
        cmd.Transaction = trans;

        da.Fill(dt);

        trans.Commit();
    }
    catch(Exception exp)
    {
        trans.Rollback();
        dt = null;
    }
    finally
    {
        cn.Close();
    }

    return dt;
}
}

מקבלת כקלט שם משתמש ומויקת אותו מטבלת המשתמשים/

```

```

public static bool DeleteUser(string username)
{
    bool success = true;

    SqlConnection cn = ConnectToDb("MyDatabase.mdf");
    SqlCommand cmd = new SqlCommand("deleteUser", cn);

    cmd.CommandType = CommandType.StoredProcedure;

    cmd.Parameters.AddWithValue("@username", username);

    SqlTransaction trans = null;

    try
    {
        cn.Open();

        trans = cn.BeginTransaction();
        cmd.Transaction = trans;

        cmd.ExecuteNonQuery();

        trans.Commit();
    }
    catch (Exception exp)
    {
        trans.Rollback();
        success = false;
    }
    finally
    {
        cn.Close();
    }

    return success;
}

// מקבלת קלט פרטי משתמש חדש ומעדכנת את נתונים המשתמש הנוכחי
public static bool UpdateUser(string username, string password, string
email, string address, string boulderGrade, string wallGrade, string birthdate,
string city, string permLvl, string gender)
{
    bool success = true;

    SqlConnection cn = ConnectToDb("MyDatabase.mdf");
    SqlCommand cmd = new SqlCommand("updateUser", cn);

    cmd.CommandType = CommandType.StoredProcedure;

    cmd.Parameters.AddWithValue("@username", username);
    cmd.Parameters.AddWithValue("@password", password);
    cmd.Parameters.AddWithValue("@email", email);
    cmd.Parameters.AddWithValue("@address", address);
    cmd.Parameters.AddWithValue("@boulderGrade", boulderGrade);
    cmd.Parameters.AddWithValue("@wallGrade", wallGrade);
    cmd.Parameters.AddWithValue("@birthdate", birthdate);
    cmd.Parameters.AddWithValue("@city", city);
    cmd.Parameters.AddWithValue("@permLvl", permLvl);
    cmd.Parameters.AddWithValue("@gender", gender);

    SqlTransaction trans = null;
}

```

```

try
{
    cn.Open();

    trans = cn.BeginTransaction();
    cmd.Transaction = trans;

    cmd.ExecuteNonQuery();

    trans.Commit();
}
catch (Exception exp)
{
    trans.Rollback();
    success = false;
}
finally
{
    cn.Close();
}

return success;
}

// מקבלת קולט פרטי משתמש חדשים ומעדכנת את נתונים המשתמש הנוכחי, אך ללא שינוי הרשאות הגישת/
public static bool UpdateUser(string username, string password, string
email, string address, string boulderGrade, string wallGrade, string birthdate,
string city, string gender)
{
    bool success = true;

    SqlConnection cn = ConnectToDb("MyDatabase.mdf");
    SqlCommand cmd = new SqlCommand("updateUserWithGender", cn);

    cmd.CommandType = CommandType.StoredProcedure;

    cmd.Parameters.AddWithValue("@username", username);
    cmd.Parameters.AddWithValue("@password", password);
    cmd.Parameters.AddWithValue("@email", email);
    cmd.Parameters.AddWithValue("@address", address);
    cmd.Parameters.AddWithValue("@boulderGrade", boulderGrade);
    cmd.Parameters.AddWithValue("@wallGrade", wallGrade);
    cmd.Parameters.AddWithValue("@birthdate", birthdate);
    cmd.Parameters.AddWithValue("@city", city);
    cmd.Parameters.AddWithValue("@gender", gender);

    SqlTransaction trans = null;

    try
    {
        cn.Open();

        trans = cn.BeginTransaction();
        cmd.Transaction = trans;

        cmd.ExecuteNonQuery();

        trans.Commit();
    }
    catch (Exception exp)
    {
        trans.Rollback();
    }
}

```

```

        success = false;
    }
    finally
    {
        cn.Close();
    }

    return success;
}

//מחזירה את טבלת המשתמשים ממוינית בסדר עולה/ירוד על פי העמודה שבקלט
public static DataTable GetDtSorted(string sortParam, string sortDirection)
{
    SqlConnection cn = ConnectToDb("MyDatabase.mdf");

    SqlCommand cmd = new SqlCommand("sortByParam", cn);
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("@sortParam", sortParam);
    cmd.Parameters.AddWithValue("@sortDirection", sortDirection);

    SqlDataAdapter da = new SqlDataAdapter();
    DataTable dt = new DataTable();

    da.SelectCommand = cmd;

    SqlTransaction trans = null;

    try
    {
        cn.Open();

        trans = cn.BeginTransaction();
        cmd.Transaction = trans;

        da.Fill(dt);

        trans.Commit();
    }
    catch (Exception exp)
    {
        trans.Rollback();
        dt = null;
    }
    finally
    {
        cn.Close();
    }

    return dt;
}

//מחזירה את השורה מטבלת המשתמשים של משתמש מסוים לפי הקלט
public static DataTable GetUserInfo(string username)
{
    SqlConnection cn = ConnectToDb("MyDatabase.mdf");

    SqlCommand cmd = new SqlCommand("getUsernameInfo", cn);
    cmd.CommandType = CommandType.StoredProcedure;

    cmd.Parameters.AddWithValue("@username", username);

    SqlDataAdapter da = new SqlDataAdapter();
}

```

```

DataTable dt = new DataTable();

da.SelectCommand = cmd;

SqlTransaction trans = null;

try
{
    cn.Open();

    trans = cn.BeginTransaction();
    cmd.Transaction = trans;

    da.Fill(dt);

    trans.Commit();
}
catch (Exception exp)
{
    trans.Rollback();
    dt = null;
}
finally
{
    cn.Close();
}

return dt;
}

// מהוירה השלמה של כרטיסי האשראי של המשתמש בטלפון
public static DataTable GetCardAutocomplete(string username, string
cardNum)
{
    SqlConnection cn = ConnectToDb("MyDatabase.mdf");

    SqlCommand cmd = new SqlCommand("getCardAutocomplete", cn);
    cmd.CommandType = CommandType.StoredProcedure;

    cmd.Parameters.AddWithValue("@username", username);
    cmd.Parameters.AddWithValue("@cardNum", cardNum);

    SqlDataAdapter da = new SqlDataAdapter();
    DataTable dt = new DataTable();

    da.SelectCommand = cmd;

    SqlTransaction trans = null;

    try
    {
        cn.Open();

        trans = cn.BeginTransaction();
        cmd.Transaction = trans;

        da.Fill(dt);

        trans.Commit();
    }
    catch (Exception exp)
    {

```

```

        trans.Rollback();
        dt = null;
    }
    finally
    {
        cn.Close();
    }

    return dt;
}

//מחזירה את טבלת המידע על המוצרים
public static DataTable GetProductTbl()
{
    SqlConnection cn = ConnectToDb("MyDatabase.mdf");

    SqlCommand cmd = new SqlCommand("getProductTbl", cn);
    cmd.CommandType = CommandType.StoredProcedure;

    SqlDataAdapter da = new SqlDataAdapter();
    DataTable dt = new DataTable();

    da.SelectCommand = cmd;
    SqlTransaction trans = null;

    try
    {
        cn.Open();

        trans = cn.BeginTransaction();
        cmd.Transaction = trans;

        da.Fill(dt);

        trans.Commit();
    }
    catch (Exception exp)
    {
        trans.Rollback();
        dt = null;
    }
    finally
    {
        cn.Close();
    }

    return dt;
}

//מחזירה את טבלת האספקה
public static DataTable GetSupplyTbl()
{
    SqlConnection cn = ConnectToDb("MyDatabase.mdf");

    SqlCommand cmd = new SqlCommand("getSupplyTbl", cn);
    cmd.CommandType = CommandType.StoredProcedure;

    SqlDataAdapter da = new SqlDataAdapter();
    DataTable dt = new DataTable();

    da.SelectCommand = cmd;
}

```

```

SqlTransaction trans = null;

try
{
    cn.Open();

    trans = cn.BeginTransaction();
    cmd.Transaction = trans;

    da.Fill(dt);

    trans.Commit();
}
catch (Exception exp)
{
    trans.Rollback();
    dt = null;
}
finally
{
    cn.Close();
}

return dt;
}

//מחזירה את טבלת פרטי ההזמנה מסוימת לפי מפתח ההזמנה
public static DataTable GetOrderProductTbl(int orderId)
{
    SqlConnection cn = ConnectToDb("MyDatabase.mdf");

    SqlCommand cmd = new SqlCommand("getOrderProductTbl", cn);
    cmd.CommandType = CommandType.StoredProcedure;

    cmd.Parameters.AddWithValue("@orderId", orderId);

    SqlDataAdapter da = new SqlDataAdapter();
    DataTable dt = new DataTable();

    da.SelectCommand = cmd;

    SqlTransaction trans = null;

    try
    {
        cn.Open();

        trans = cn.BeginTransaction();
        cmd.Transaction = trans;

        da.Fill(dt);

        trans.Commit();
    }
    catch (Exception exp)
    {
        trans.Rollback();
        dt = null;
    }
    finally
    {

```

```

        cn.Close();
    }

    return dt;
}

//מחזירה את טבלת הזמנות
public static DataTable GetOrderTbl()
{
    SqlConnection cn = ConnectToDb("MyDatabase.mdf");

    SqlCommand cmd = new SqlCommand("getOrderTbl", cn);
    cmd.CommandType = CommandType.StoredProcedure;

    SqlDataAdapter da = new SqlDataAdapter();
    DataTable dt = new DataTable();

    da.SelectCommand = cmd;

    SqlTransaction trans = null;

    try
    {
        cn.Open();

        trans = cn.BeginTransaction();
        cmd.Transaction = trans;

        da.Fill(dt);

        trans.Commit();
    }
    catch (Exception exp)
    {
        trans.Rollback();
        dt = null;
    }
    finally
    {
        cn.Close();
    }

    return dt;
}

//מחזירה את טבלת פרטי הזמנה
public static DataTable GetOrderProductTbl()
{
    SqlConnection cn = ConnectToDb("MyDatabase.mdf");

    SqlCommand cmd = new SqlCommand("getCompleteOrderProductTbl", cn);
    cmd.CommandType = CommandType.StoredProcedure;

    SqlDataAdapter da = new SqlDataAdapter();
    DataTable dt = new DataTable();

    da.SelectCommand = cmd;

    SqlTransaction trans = null;

    try
    {

```

```

        cn.Open();

        trans = cn.BeginTransaction();
        cmd.Transaction = trans;

        da.Fill(dt);

        trans.Commit();
    }
    catch (Exception exp)
    {
        trans.Rollback();
        dt = null;
    }
    finally
    {
        cn.Close();
    }

    return dt;
}

//עדכנת את כמות מוצר מסוים בטבלת האספקה
public static bool UpdateAmount(int orderId, int productId, double size,
int amount)
{
    //Return to bool later on
    bool success = true;

    SqlConnection cn = ConnectToDb("MyDatabase.mdf");
    SqlCommand cmd = new SqlCommand("updateAmount", cn);

    cmd.CommandType = CommandType.StoredProcedure;

    cmd.Parameters.AddWithValue("@orderId", orderId);
    cmd.Parameters.AddWithValue("@productId", productId);
    cmd.Parameters.AddWithValue("@size", size);
    cmd.Parameters.AddWithValue("@amount", amount);

    SqlTransaction trans = null;

    try
    {
        cn.Open();

        trans = cn.BeginTransaction();
        cmd.Transaction = trans;

        cmd.ExecuteNonQuery();

        trans.Commit();
    }
    catch (Exception exp)
    {
        trans.Rollback();
        success = false;
    }
    finally
    {
        cn.Close();
    }
}

```

```

        return success;
    }


public static DataTable GetOrderTblByUser(string username)
{
    SqlConnection cn = ConnectToDb("MyDatabase.mdf");

    SqlCommand cmd = new SqlCommand("getOrderTblByUser", cn);
    cmd.CommandType = CommandType.StoredProcedure;

    cmd.Parameters.AddWithValue("@username", username);

    SqlDataAdapter da = new SqlDataAdapter();
    DataTable dt = new DataTable();

    da.SelectCommand = cmd;

    SqlTransaction trans = null;

    try
    {
        cn.Open();

        trans = cn.BeginTransaction();
        cmd.Transaction = trans;

        da.Fill(dt);

        trans.Commit();
    }
    catch (Exception exp)
    {
        trans.Rollback();
        dt = null;
    }
    finally
    {
        cn.Close();
    }

    return dt;
}


public static DataSet GetDataSetBySP(string sp)
{
    SqlConnection cn = ConnectToDb("MyDatabase.mdf");

    SqlCommand cmd = new SqlCommand(sp, cn);
    cmd.CommandType = CommandType.StoredProcedure;

    SqlDataAdapter da = new SqlDataAdapter();
    DataSet ds = new DataSet();

    da.SelectCommand = cmd;

    SqlTransaction trans = null;

    try
    {
        cn.Open();

```

```
        trans = cn.BeginTransaction();
        cmd.Transaction = trans;

        da.Fill(ds);

        trans.Commit();
    }
    catch (Exception exp)
    {
        trans.Rollback();
        ds = null;
    }
    finally
    {
        cn.Close();
    }

    return ds;
}

//מחזר את טבלת הגודלים של מוצר מסוים מטבלת האספקה
public static DataTable GetSizeTable(int productId)
{
    SqlConnection cn = ConnectToDb("MyDatabase.mdf");

    SqlCommand cmd = new SqlCommand("getSizeFromSupply", cn);
    cmd.CommandType = CommandType.StoredProcedure;

    cmd.Parameters.AddWithValue("@productId", productId);

    SqlDataAdapter da = new SqlDataAdapter();
    DataTable dt = new DataTable();

    da.SelectCommand = cmd;

    SqlTransaction trans = null;

    try
    {
        cn.Open();

        trans = cn.BeginTransaction();
        cmd.Transaction = trans;

        da.Fill(dt);

        trans.Commit();
    }
    catch (Exception exp)
    {
        trans.Rollback();
        dt = null;
    }
    finally
    {
        cn.Close();
    }

    return dt;
}
}
```

SaleStatisticsHelper.cs

מיקום: ComputersProjectNew/App_Code/SaleStatisticsHelper.cs

קוד:C#

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data;

/// <summary>
/// Summary description for SaleStatistic
/// </summary>
public static class SaleStatisticsHelper
{
    // פעולה מקבלת טווח תאריכים ומחזירה את טבלת ההזמנות בין התאריכים האלה
    private static DataTable GetRangedOrderTbl(DateTime startDate, DateTime
endDate)
    {
        DataTable orderTbl = MyAdoHelper.GetOrderTbl();
        DataTable tempRangedOrderTbl =
orderTbl.Select("paid>0").CopyToDataTable(), rangedOrderTbl = orderTbl.Clone();
        foreach(DataRow dataRow in tempRangedOrderTbl.Rows)
        {
            if (!((DateTime)dataRow[ "datetime" ] < startDate ||
(DateTime)dataRow[ "datetime" ] > endDate))
                rangedOrderTbl.Rows.Add(dataRow.ItemArray);
        }
        return rangedOrderTbl;
    }

    // פעולה מקבלת טווח תאריכים ומחזירה את טבלת פרטי ההזמנה בין התאריכים האלה
    private static DataTable GetRangedOrderProductTbl(DateTime startDate,
DateTime endDate)
    {
        DataTable rangedOrderTbl = GetRangedOrderTbl(startDate, endDate),
orderProductTbl = MyAdoHelper.GetOrderProductTbl(), orderProductTblCopy =
orderProductTbl.Copy(), rangedOrderProductTbl = orderProductTbl.Clone();

        foreach(DataRow dataRow in orderProductTblCopy.Rows)
        {
            if(rangedOrderTbl.Select("orderId=" +
dataRow[ "orderId" ].ToString()).Count() > 0)
            {
                rangedOrderProductTbl.Rows.Add(dataRow.ItemArray);
            }
        }
        return rangedOrderProductTbl;
    }

    // המחלקה מחזירה את מספר המוצרים הכל או חמי פחות נמכרים בטווח התאריכים הנתון
    public static Dictionary<string, int> GetItemSales(int amount, bool best,
DateTime startDate, DateTime endDate)
    {
        Dictionary<string, int> salesDict = new Dictionary<string, int>(),
finalDict = new Dictionary<string, int>();

```

```

DataTable orderProductRangedTbl = GetRangedOrderProductTbl(startDate,
endDate), productTbl = MyAdoHelper.GetProductTbl();

var results = orderProductRangedTbl.AsEnumerable()
    .GroupBy(x => new { productid = x.Field<int>("productid") })
    .Select(x => new { productid = x.Key.productid, total = x.Sum(y
=> y.Field<int>("amount")) })
    .OrderByDescending(x => x.total)
    .ToList();

if (!best)
    results.Reverse();

for(int i = 0; i < amount && i < results.Count; i++)
{
    salesDict.Add(productTbl.Select("productId=" +
results[i].productid)[0]["name"].ToString(), results[i].total);
}

return salesDict;
}

// הפעולה מוחירה את מספר המכירות של מוצרים מהחברות השונות בטווח התאריכים הנתון
public static Dictionary<string, int> GetCompanyPercents(DateTime
startDate, DateTime endDate)
{
    Dictionary<string, int> salesDict = new Dictionary<string, int>();

    DataTable orderProductRangedTbl = GetRangedOrderProductTbl(startDate,
endDate), productTbl = MyAdoHelper.GetProductTbl();
    orderProductRangedTbl.Columns.Add("company");

    foreach(DataRow dataRow in orderProductRangedTbl.Rows)
    {
        dataRow["company"] = (productTbl.Select("productId=" +
dataRow["productId"]))[0]["company"];
    }

    var results = orderProductRangedTbl.AsEnumerable()
        .GroupBy(x => new { company = x.Field<string>("company") })
        .Select(x => new { company = x.Key.company, total = x.Sum(y =>
y.Field<int>("amount")) })
        .OrderByDescending(x => x.total)
        .ToList();

    foreach(var item in results)
    {
        salesDict.Add(item.company, item.total);
    }

    return salesDict;
}

// הפעולה מוחירה את גודל כל מוצר במסום בטווח התאריכים הנתון
public static Dictionary<int, int> GetSizePercents(int productId, DateTime
startDate, DateTime endDate)
{
    Dictionary<int, int> percentMap = new Dictionary<int, int>();

    DataTable orderProductRangedTbl = GetRangedOrderProductTbl(startDate,
endDate);

```

```

    var results = orderProductRangedTbl.Select("productId=" +
productId).AsEnumerable()
    .GroupBy(x => new { productid = x.Field<int>"productid"), size =
x.Field<int>"size" })
    .Select(x => new { productid = x.Key.productid, size =
x.Key.size, total = x.Sum(y => y.Field<int>"amount") })
    .OrderByDescending(x => x.total)
    .ToList();

    foreach(var item in results)
    {
        percentMap.Add(item.size, item.total);
    }
    return percentMap;
}

// הפעולה מהוירה את מספר הלקוחות שרכשו באתר כי פחות או שווה התאריכים הנתון
public static Dictionary<string, double> GetCustomerPurchases(int amount,
bool best, DateTime startDate, DateTime endDate)
{
    Dictionary<string, double> purchaseDict = new Dictionary<string,
double>();

    DataTable newOrderTbl = GetRangedOrderTbl(startDate, endDate);

    var results = newOrderTbl.AsEnumerable()
        .GroupBy(x => new { username = x.Field<string>"username" })
        .Select(x => new { username = x.Key.username, total = x.Sum(y
=> y.Field<double>"paid" ) })
        .OrderByDescending(x => x.total)
        .ToList();

    if (!best)
        results.Reverse();

    for (int i = 0; i < amount && i < results.Count; i++)
    {
        purchaseDict.Add(results[i].username, results[i].total);
    }

    return purchaseDict;
}
}

```

main.cssCSS קוד: [חיקום](#)

```

@charset "UTF-8";
@import url(font-awesome.min.css);
@import url("https://fonts.googleapis.com/css?family=Poppins:300,400,500,600,700");

/*
   Hielo by TEMPLATED
   templated.co @templatedco
   Released for free under the Creative Commons Attribution 3.0 license
   (templated.co/license)
*/

/* Reset */

.wrapper.style2 input[type="text"].widetext{
    width: 300px !important;
}

.test {
    display: block !important;
    margin: 0 auto !important;
}

.wrapper.style2 select{
    justify-content: center !important;
    align-content: center !important;
    display: block;
    margin: 0 auto;
}

.hiddencol {
    display: none;
}

.wrapper.style2 p.red {
    color: red;
    text-align: center;
}

.wrapper.style2 h1.red, h2.red, h3.red, h4.red, h5.red, h6.red {
    color: red !important;
}

.wrapper.style2 p.green {
    color: forestgreen;
    text-align: center;
}

.wrapper.style2 h1.green, h2.green, h3.green, h4.green, h5.green, h6.green {
    color: forestgreen !important;
}

img.gv {
    margin: 0;
}

```

```
padding: 0;
border: 0;
font-size: 100%;
font: inherit;
vertical-align: middle !important;
}

html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6,
p, blockquote, pre, a, abbr, acronym, address, big, cite, code, del, dfn, em,
img, ins, kbd, q, s, samp, small, strike, strong, sub, sup, tt, var, b, u, i,
center, dl, dt, dd, ol, ul, li, fieldset, form, label, legend, table, caption,
tbody, tfoot, thead, tr, th, td, article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup, menu, nav, output, ruby, section,
summary, time, mark, audio, video {
    margin: 0;
    padding: 0;
    border: 0;
    font-size: 100%;
    font: inherit;
    vertical-align: baseline;
}

article, aside, details, figcaption, figure, footer, header, hgroup,
menu, nav, section {
    display: block;
}

body {
    line-height: 1;
}

ol, ul {
    list-style: none;
}

blockquote, q {
    quotes: none;
}

blockquote:before, blockquote:after, q:before, q:after {
    content: '';
    content: none;
}

table {
    border-collapse: collapse;
    border-spacing: 0;
}

body {
    -webkit-text-size-adjust: none;
}

/* Box Model */

*, *:before, *:after {
    -moz-box-sizing: border-box;
    -webkit-box-sizing: border-box;
    box-sizing: border-box;
}

/* Containers */
```

```
.container {  
    margin-left: auto;  
    margin-right: auto;  
}  
  
.container.\31 25\25 {  
    width: 100%;  
    max-width: 100rem;  
    min-width: 80rem;  
}  
  
.container.\37 5\25 {  
    width: 60rem;  
}  
  
.container.\35 0\25 {  
    width: 40rem;  
}  
  
.container.\32 5\25 {  
    width: 20rem;  
}  
  
.container {  
    width: 80rem;  
}  
  
@media screen and (max-width: 1680px) {  
  
    .container.\31 25\25 {  
        width: 100%;  
        max-width: 100rem;  
        min-width: 80rem;  
    }  
  
    .container.\37 5\25 {  
        width: 60rem;  
    }  
  
    .container.\35 0\25 {  
        width: 40rem;  
    }  
  
    .container.\32 5\25 {  
        width: 20rem;  
    }  
  
    .container {  
        width: 80rem;  
    }  
  
}  
  
@media screen and (max-width: 1280px) {  
  
    .container.\31 25\25 {  
        width: 100%;  
        max-width: 81.25rem;  
        min-width: 65rem;  
    }  
  
    .container.\37 5\25 {
```

```
        width: 48.75rem;
    }

.container.\35 0\25 {
    width: 32.5rem;
}

.container.\32 5\25 {
    width: 16.25rem;
}

.container {
    width: 65rem;
}

}

@media screen and (max-width: 980px) {

    .container.\31 25\25 {
        width: 100%;
        max-width: 112.5%;
        min-width: 90%;
    }

    .container.\37 5\25 {
        width: 67.5%;
    }

    .container.\35 0\25 {
        width: 45%;
    }

    .container.\32 5\25 {
        width: 22.5%;
    }

    .container {
        width: 90%;
    }

}

@media screen and (max-width: 736px) {

    .container.\31 25\25 {
        width: 100%;
        max-width: 112.5%;
        min-width: 90%;
    }

    .container.\37 5\25 {
        width: 67.5%;
    }

    .container.\35 0\25 {
        width: 45%;
    }

    .container.\32 5\25 {
        width: 22.5%;
    }

}
```

```
.container {  
    width: 90% !important;  
}  
  
}  
  
@media screen and (max-width: 480px) {  
  
    .container.\31 25\25 {  
        width: 100%;  
        max-width: 112.5%;  
        min-width: 90%;  
    }  
  
    .container.\37 5\25 {  
        width: 67.5%;  
    }  
  
    .container.\35 0\25 {  
        width: 45%;  
    }  
  
    .container.\32 5\25 {  
        width: 22.5%;  
    }  
  
    .container {  
        width: 90% !important;  
    }  
  
}  
  
/* Grid */  
  
.row {  
    border-bottom: solid 1px transparent;  
    -moz-box-sizing: border-box;  
    -webkit-box-sizing: border-box;  
    box-sizing: border-box;  
}  
  
.row > * {  
    float: left;  
    -moz-box-sizing: border-box;  
    -webkit-box-sizing: border-box;  
    box-sizing: border-box;  
}  
  
.row:after, .row:before {  
    content: '';  
    display: block;  
    clear: both;  
    height: 0;  
}  
  
.row.uniform > * > :first-child {  
    margin-top: 0;  
}  
  
.row.uniform > * > :last-child {  
    margin-bottom: 0;
```

```
}

.row.\30\25 > * {
    padding: 0 0 0 0rem;
}

.row.\30\25 {
    margin: 0 0 -1px 0rem;
}

.row.uniform.\30\25 > * {
    padding: 0rem 0 0 0rem;
}

.row.uniform.\30\25 {
    margin: 0rem 0 -1px 0rem;
}

.row > * {
    padding: 0 0 0 2rem;
}

.row {
    margin: 0 0 -1px -2rem;
}

.row.uniform > * {
    padding: 2rem 0 0 2rem;
}

.row.uniform {
    margin: -2rem 0 -1px -2rem;
}

.row.\32\00\25 > * {
    padding: 0 0 0 4rem;
}

.row.\32\00\25 {
    margin: 0 0 -1px -4rem;
}

.row.uniform.\32\00\25 > * {
    padding: 4rem 0 0 4rem;
}

.row.uniform.\32\00\25 {
    margin: -4rem 0 -1px -4rem;
}

.row.\31\50\25 > * {
    padding: 0 0 0 3rem;
}

.row.\31\50\25 {
    margin: 0 0 -1px -3rem;
}

.row.uniform.\31\50\25 > * {
    padding: 3rem 0 0 3rem;
}
```

```
.row.uniform.\31_50\25 {  
    margin: -3rem 0 -1px -3rem;  
}  
  
.row.\35_0\25 > * {  
    padding: 0 0 0 1rem;  
}  
  
.row.\35_0\25 {  
    margin: 0 0 -1px -1rem;  
}  
  
.row.uniform.\35_0\25 > * {  
    padding: 1rem 0 0 1rem;  
}  
  
.row.uniform.\35_0\25 {  
    margin: -1rem 0 -1px -1rem;  
}  
  
.row.\32_5\25 > * {  
    padding: 0 0 0 0.5rem;  
}  
  
.row.\32_5\25 {  
    margin: 0 0 -1px -0.5rem;  
}  
  
.row.uniform.\32_5\25 > * {  
    padding: 0.5rem 0 0 0.5rem;  
}  
  
.row.uniform.\32_5\25 {  
    margin: -0.5rem 0 -1px -0.5rem;  
}  
  
.\\31_2u, .\\31_2u\\24 {  
    width: 100%;  
    clear: none;  
    margin-left: 0;  
}  
  
.\\31_1u, .\\31_1u\\24 {  
    width: 91.6666666667%;  
    clear: none;  
    margin-left: 0;  
}  
  
.\\31_0u, .\\31_0u\\24 {  
    width: 83.3333333333%;  
    clear: none;  
    margin-left: 0;  
}  
  
.\\39_u, .\\39_u\\24 {  
    width: 75%;  
    clear: none;  
    margin-left: 0;  
}  
  
.\\38_u, .\\38_u\\24 {  
    width: 66.6666666667%;
```

```
        clear: none;
        margin-left: 0;
    }

.\37 u, .\37 u\24 {
    width: 58.3333333333%;
    clear: none;
    margin-left: 0;
}

.\36 u, .\36 u\24 {
    width: 50%;
    clear: none;
    margin-left: 0;
}

.\35 u, .\35 u\24 {
    width: 41.6666666667%;
    clear: none;
    margin-left: 0;
}

.\34 u, .\34 u\24 {
    width: 33.3333333333%;
    clear: none;
    margin-left: 0;
}

.\33 u, .\33 u\24 {
    width: 25%;
    clear: none;
    margin-left: 0;
}

.\32 u, .\32 u\24 {
    width: 16.6666666667%;
    clear: none;
    margin-left: 0;
}

.\31 u, .\31 u\24 {
    width: 8.3333333333%;
    clear: none;
    margin-left: 0;
}

.\31 2u\24 + *,
.\31 1u\24 + *,
.\31 0u\24 + *,
.\39 u\24 + *,
.\38 u\24 + *,
.\37 u\24 + *,
.\36 u\24 + *,
.\35 u\24 + *,
.\34 u\24 + *,
.\33 u\24 + *,
.\32 u\24 + *,
.\31 u\24 + * {
    clear: left;
}

.\-11u {
```

```
        margin-left: 91.66667%;  
    }  
  
.\\-10u {  
    margin-left: 83.33333%;  
}  
  
.\\-9u {  
    margin-left: 75%;  
}  
  
.\\-8u {  
    margin-left: 66.66667%;  
}  
  
.\\-7u {  
    margin-left: 58.33333%;  
}  
  
.\\-6u {  
    margin-left: 50%;  
}  
  
.\\-5u {  
    margin-left: 41.66667%;  
}  
  
.\\-4u {  
    margin-left: 33.33333%;  
}  
  
.\\-3u {  
    margin-left: 25%;  
}  
  
.\\-2u {  
    margin-left: 16.66667%;  
}  
  
.\\-1u {  
    margin-left: 8.33333%;  
}  
  
@media screen and (max-width: 1680px) {  
  
.row > * {  
    padding: 0 0 0 2rem;  
}  
  
.row {  
    margin: 0 0 -1px -2rem;  
}  
  
.row.uniform > * {  
    padding: 2rem 0 0 2rem;  
}  
  
.row.uniform {  
    margin: -2rem 0 -1px -2rem;  
}  
  
.row.\\32 00\\25 > * {
```

```
        padding: 0 0 0 4rem;
    }

.row.\32_00\25 {
    margin: 0 0 -1px -4rem;
}

.row.uniform.\32_00\25 > * {
    padding: 4rem 0 0 4rem;
}

.row.uniform.\32_00\25 {
    margin: -4rem 0 -1px -4rem;
}

.row.\31_50\25 > * {
    padding: 0 0 0 3rem;
}

.row.\31_50\25 {
    margin: 0 0 -1px -3rem;
}

.row.uniform.\31_50\25 > * {
    padding: 3rem 0 0 3rem;
}

.row.uniform.\31_50\25 {
    margin: -3rem 0 -1px -3rem;
}

.row.\35_0\25 > * {
    padding: 0 0 0 1rem;
}

.row.\35_0\25 {
    margin: 0 0 -1px -1rem;
}

.row.uniform.\35_0\25 > * {
    padding: 1rem 0 0 1rem;
}

.row.uniform.\35_0\25 {
    margin: -1rem 0 -1px -1rem;
}

.row.\32_5\25 > * {
    padding: 0 0 0 0.5rem;
}

.row.\32_5\25 {
    margin: 0 0 -1px -0.5rem;
}

.row.uniform.\32_5\25 > * {
    padding: 0.5rem 0 0 0.5rem;
}

.row.uniform.\32_5\25 {
    margin: -0.5rem 0 -1px -0.5rem;
}
```

```
.\31 2u\28xlarge\29, .\31 2u\24\28xlarge\29 {
    width: 100%;
    clear: none;
    margin-left: 0;
}

.\31 1u\28xlarge\29, .\31 1u\24\28xlarge\29 {
    width: 91.6666666667%;
    clear: none;
    margin-left: 0;
}

.\31 0u\28xlarge\29, .\31 0u\24\28xlarge\29 {
    width: 83.3333333333%;
    clear: none;
    margin-left: 0;
}

.\39 u\28xlarge\29, .\39 u\24\28xlarge\29 {
    width: 75%;
    clear: none;
    margin-left: 0;
}

.\38 u\28xlarge\29, .\38 u\24\28xlarge\29 {
    width: 66.6666666667%;
    clear: none;
    margin-left: 0;
}

.\37 u\28xlarge\29, .\37 u\24\28xlarge\29 {
    width: 58.3333333333%;
    clear: none;
    margin-left: 0;
}

.\36 u\28xlarge\29, .\36 u\24\28xlarge\29 {
    width: 50%;
    clear: none;
    margin-left: 0;
}

.\35 u\28xlarge\29, .\35 u\24\28xlarge\29 {
    width: 41.6666666667%;
    clear: none;
    margin-left: 0;
}

.\34 u\28xlarge\29, .\34 u\24\28xlarge\29 {
    width: 33.3333333333%;
    clear: none;
    margin-left: 0;
}

.\33 u\28xlarge\29, .\33 u\24\28xlarge\29 {
    width: 25%;
    clear: none;
    margin-left: 0;
}

.\32 u\28xlarge\29, .\32 u\24\28xlarge\29 {
```

```
        width: 16.6666666667%;  
        clear: none;  
        margin-left: 0;  
    }  
  
.31 u\28xlarge\29, .31 u\24\28xlarge\29 {  
    width: 8.3333333333%;  
    clear: none;  
    margin-left: 0;  
}  
  
.31 2u\24\28xlarge\29 + *,  
.31 1u\24\28xlarge\29 + *,  
.31 0u\24\28xlarge\29 + *,  
.39 u\24\28xlarge\29 + *,  
.38 u\24\28xlarge\29 + *,  
.37 u\24\28xlarge\29 + *,  
.36 u\24\28xlarge\29 + *,  
.35 u\24\28xlarge\29 + *,  
.34 u\24\28xlarge\29 + *,  
.33 u\24\28xlarge\29 + *,  
.32 u\24\28xlarge\29 + *,  
.31 u\24\28xlarge\29 + * {  
    clear: left;  
}  
  
.11u\28xlarge\29 {  
    margin-left: 91.66667%;  
}  
  
.10u\28xlarge\29 {  
    margin-left: 83.33333%;  
}  
  
.9u\28xlarge\29 {  
    margin-left: 75%;  
}  
  
.8u\28xlarge\29 {  
    margin-left: 66.66667%;  
}  
  
.7u\28xlarge\29 {  
    margin-left: 58.33333%;  
}  
  
.6u\28xlarge\29 {  
    margin-left: 50%;  
}  
  
.5u\28xlarge\29 {  
    margin-left: 41.66667%;  
}  
  
.4u\28xlarge\29 {  
    margin-left: 33.33333%;  
}  
  
.3u\28xlarge\29 {  
    margin-left: 25%;  
}
```

```
.\-\-2u\28xlarge\29 {
    margin-left: 16.66667%;
}

.\-\-1u\28xlarge\29 {
    margin-left: 8.33333%;
}

}

@media screen and (max-width: 1280px) {

    .row > * {
        padding: 0 0 0 1.5rem;
    }

    .row {
        margin: 0 0 -1px -1.5rem;
    }

    .row.uniform > * {
        padding: 1.5rem 0 0 1.5rem;
    }

    .row.uniform {
        margin: -1.5rem 0 -1px -1.5rem;
    }

    .row.\32_00\25 > * {
        padding: 0 0 0 3rem;
    }

    .row.\32_00\25 {
        margin: 0 0 -1px -3rem;
    }

    .row.uniform.\32_00\25 > * {
        padding: 3rem 0 0 3rem;
    }

    .row.uniform.\32_00\25 {
        margin: -3rem 0 -1px -3rem;
    }

    .row.\31_50\25 > * {
        padding: 0 0 0 2.25rem;
    }

    .row.\31_50\25 {
        margin: 0 0 -1px -2.25rem;
    }

    .row.uniform.\31_50\25 > * {
        padding: 2.25rem 0 0 2.25rem;
    }

    .row.uniform.\31_50\25 {
        margin: -2.25rem 0 -1px -2.25rem;
    }

    .row.\35_0\25 > * {
        padding: 0 0 0 0.75rem;
    }
```

```
}

.row.\35 0\25 {
    margin: 0 0 -1px -0.75rem;
}

.row.uniform.\35 0\25 > * {
    padding: 0.75rem 0 0 0.75rem;
}

.row.uniform.\35 0\25 {
    margin: -0.75rem 0 -1px -0.75rem;
}

.row.\32 5\25 > * {
    padding: 0 0 0 0.375rem;
}

.row.\32 5\25 {
    margin: 0 0 -1px -0.375rem;
}

.row.uniform.\32 5\25 > * {
    padding: 0.375rem 0 0 0.375rem;
}

.row.uniform.\32 5\25 {
    margin: -0.375rem 0 -1px -0.375rem;
}

.\31 2u\28large\29, .\31 2u\24\28large\29 {
    width: 100%;
    clear: none;
    margin-left: 0;
}

.\31 1u\28large\29, .\31 1u\24\28large\29 {
    width: 91.6666666667%;
    clear: none;
    margin-left: 0;
}

.\31 0u\28large\29, .\31 0u\24\28large\29 {
    width: 83.3333333333%;
    clear: none;
    margin-left: 0;
}

.\39 u\28large\29, .\39 u\24\28large\29 {
    width: 75%;
    clear: none;
    margin-left: 0;
}

.\38 u\28large\29, .\38 u\24\28large\29 {
    width: 66.6666666667%;
    clear: none;
    margin-left: 0;
}

.\37 u\28large\29, .\37 u\24\28large\29 {
    width: 58.3333333333%;
```

```
        clear: none;
        margin-left: 0;
    }

.\36 u\28large\29, .\36 u\24\28large\29 {
    width: 50%;
    clear: none;
    margin-left: 0;
}

.\35 u\28large\29, .\35 u\24\28large\29 {
    width: 41.6666666667%;
    clear: none;
    margin-left: 0;
}

.\34 u\28large\29, .\34 u\24\28large\29 {
    width: 33.3333333333%;
    clear: none;
    margin-left: 0;
}

.\33 u\28large\29, .\33 u\24\28large\29 {
    width: 25%;
    clear: none;
    margin-left: 0;
}

.\32 u\28large\29, .\32 u\24\28large\29 {
    width: 16.6666666667%;
    clear: none;
    margin-left: 0;
}

.\31 u\28large\29, .\31 u\24\28large\29 {
    width: 8.3333333333%;
    clear: none;
    margin-left: 0;
}

.\31 2u\24\28large\29 + *,
.\31 1u\24\28large\29 + *,
.\31 0u\24\28large\29 + *,
.\39 u\24\28large\29 + *,
.\38 u\24\28large\29 + *,
.\37 u\24\28large\29 + *,
.\36 u\24\28large\29 + *,
.\35 u\24\28large\29 + *,
.\34 u\24\28large\29 + *,
.\33 u\24\28large\29 + *,
.\32 u\24\28large\29 + *,
.\31 u\24\28large\29 + * {
    clear: left;
}

.-11u\28large\29 {
    margin-left: 91.66667%;
}

.-10u\28large\29 {
    margin-left: 83.33333%;
}
```

```
.\-\-9u\28large\29 {
    margin-left: 75%;
}

.\-\-8u\28large\29 {
    margin-left: 66.66667%;
}

.\-\-7u\28large\29 {
    margin-left: 58.33333%;
}

.\-\-6u\28large\29 {
    margin-left: 50%;
}

.\-\-5u\28large\29 {
    margin-left: 41.66667%;
}

.\-\-4u\28large\29 {
    margin-left: 33.33333%;
}

.\-\-3u\28large\29 {
    margin-left: 25%;
}

.\-\-2u\28large\29 {
    margin-left: 16.66667%;
}

.\-\-1u\28large\29 {
    margin-left: 8.33333%;
}

}

@media screen and (max-width: 980px) {

    .row > * {
        padding: 0 0 0 1.5rem;
    }

    .row {
        margin: 0 0 -1px -1.5rem;
    }

    .row.uniform > * {
        padding: 1.5rem 0 0 1.5rem;
    }

    .row.uniform {
        margin: -1.5rem 0 -1px -1.5rem;
    }

    .row.\32 00\25 > * {
        padding: 0 0 0 3rem;
    }

    .row.\32 00\25 {
```

```
        margin: 0 0 -1px -3rem;
    }

    .row.uniform.\32 00\25 > * {
        padding: 3rem 0 0 3rem;
    }

    .row.uniform.\32 00\25 {
        margin: -3rem 0 -1px -3rem;
    }

    .row.\31 50\25 > * {
        padding: 0 0 0 2.25rem;
    }

    .row.\31 50\25 {
        margin: 0 0 -1px -2.25rem;
    }

    .row.uniform.\31 50\25 > * {
        padding: 2.25rem 0 0 2.25rem;
    }

    .row.uniform.\31 50\25 {
        margin: -2.25rem 0 -1px -2.25rem;
    }

    .row.uniform {
        margin: -1.5rem 0 -1px -1.5rem;
    }

    .row.\32 00\25 > * {
        padding: 0 0 0 3rem;
    }

    .row.\32 00\25 {
        margin: 0 0 -1px -3rem;
    }

    .row.uniform.\32 00\25 > * {
        padding: 3rem 0 0 3rem;
    }

    .row.uniform.\32 00\25 {
        margin: -3rem 0 -1px -3rem;
    }

    .row.\31 50\25 > * {
        padding: 0 0 0 2.25rem;
    }

    .row.\31 50\25 {
        margin: 0 0 -1px -2.25rem;
    }

    .row.uniform.\31 50\25 > * {
        padding: 2.25rem 0 0 2.25rem;
    }

    .row.uniform.\31 50\25 {
        margin: -2.25rem 0 -1px -2.25rem;
    }
```

```
.row.\35 0\25 > * {
    padding: 0 0 0 0.75rem;
}

.row.\35 0\25 {
    margin: 0 0 -1px -0.75rem;
}

.row.uniform.\35 0\25 > * {
    padding: 0.75rem 0 0 0.75rem;
}

.row.uniform.\35 0\25 {
    margin: -0.75rem 0 -1px -0.75rem;
}

.row.\32 5\25 > * {
    padding: 0 0 0 0.375rem;
}

.row.\32 5\25 {
    margin: 0 0 -1px -0.375rem;
}

.row.uniform.\32 5\25 > * {
    padding: 0.375rem 0 0 0.375rem;
}

.row.uniform.\32 5\25 {
    margin: -0.375rem 0 -1px -0.375rem;
}

.\31 2u\28large\29, .\31 2u\24\28large\29 {
    width: 100%;
    clear: none;
    margin-left: 0;
}

.\31 1u\28large\29, .\31 1u\24\28large\29 {
    width: 91.6666666667%;
    clear: none;
    margin-left: 0;
}

.\31 0u\28large\29, .\31 0u\24\28large\29 {
    width: 83.3333333333%;
    clear: none;
    margin-left: 0;
}

.\39 u\28large\29, .\39 u\24\28large\29 {
    width: 75%;
    clear: none;
    margin-left: 0;
}

.\38 u\28large\29, .\38 u\24\28large\29 {
    width: 66.6666666667%;
    clear: none;
    margin-left: 0;
}
```

```
.\37 u\28large\29, .\37 u\24\28large\29 {
    width: 58.3333333333%;
    clear: none;
    margin-left: 0;
}

.\36 u\28large\29, .\36 u\24\28large\29 {
    width: 50%;
    clear: none;
    margin-left: 0;
}

.\35 u\28large\29, .\35 u\24\28large\29 {
    width: 41.6666666667%;
    clear: none;
    margin-left: 0;
}

.\34 u\28large\29, .\34 u\24\28large\29 {
    width: 33.3333333333%;
    clear: none;
    margin-left: 0;
}

.\33 u\28large\29, .\33 u\24\28large\29 {
    width: 25%;
    clear: none;
    margin-left: 0;
}

.\32 u\28large\29, .\32 u\24\28large\29 {
    width: 16.6666666667%;
    clear: none;
    margin-left: 0;
}

.\31 u\28large\29, .\31 u\24\28large\29 {
    width: 8.3333333333%;
    clear: none;
    margin-left: 0;
}

.\31 2u\24\28large\29 + *,
.\31 1u\24\28large\29 + *,
.\31 0u\24\28large\29 + *,
.\39 u\24\28large\29 + *,
.\38 u\24\28large\29 + *,
.\37 u\24\28large\29 + *,
.\36 u\24\28large\29 + *,
.\35 u\24\28large\29 + *,
.\34 u\24\28large\29 + *,
.\33 u\24\28large\29 + *,
.\32 u\24\28large\29 + *,
.\31 u\24\28large\29 + * {
    clear: left;
}

.-11u\28large\29 {
    margin-left: 91.66667%;
```

```
.\-\-10u\28large\29 {
    margin-left: 83.33333%;
}

.\-\-9u\28large\29 {
    margin-left: 75%;
}

.\-\-8u\28large\29 {
    margin-left: 66.66667%;
}

.\-\-7u\28large\29 {
    margin-left: 58.33333%;
}

.\-\-6u\28large\29 {
    margin-left: 50%;
}

.\-\-5u\28large\29 {
    margin-left: 41.66667%;
}

.\-\-4u\28large\29 {
    margin-left: 33.33333%;
}

.\-\-3u\28large\29 {
    margin-left: 25%;
}

.\-\-2u\28large\29 {
    margin-left: 16.66667%;
}

.\-\-1u\28large\29 {
    margin-left: 8.33333%;
}

}

@media screen and (max-width: 980px) {

    .row > * {
        padding: 0 0 0 1.5rem;
    }

    .row {
        margin: 0 0 -1px -1.5rem;
    }

    .row.uniform > * {
        padding: 1.5rem 0 0 1.5rem;
    }

    .row.uniform {
        margin: -1.5rem 0 -1px -1.5rem;
    }

    .row.\32 00\25 > * {
        padding: 0 0 0 3rem;
    }
}
```

```
}

.row.\32_00\25 {
    margin: 0 0 -1px -3rem;
}

.row.uniform.\32_00\25 > * {
    padding: 3rem 0 0 3rem;
}

.row.uniform.\32_00\25 {
    margin: -3rem 0 -1px -3rem;
}

.row.\31_50\25 > * {
    padding: 0 0 0 2.25rem;
}

.row.\31_50\25 {
    margin: 0 0 -1px -2.25rem;
}

.row.uniform.\31_50\25 > * {
    padding: 2.25rem 0 0 2.25rem;
}

.row.uniform.\31_50\25 {
    margin: -2.25rem 0 -1px -2.25rem;
}

.row.\35_0\25 > * {
    padding: 0 0 0 0.625rem;
}

.row.\35_0\25 {
    margin: 0 0 -1px -0.625rem;
}

.row.uniform.\35_0\25 > * {
    padding: 0.625rem 0 0 0.625rem;
}

.row.uniform.\35_0\25 {
    margin: -0.625rem 0 -1px -0.625rem;
}

.row.\32_5\25 > * {
    padding: 0 0 0 0.3125rem;
}

.row.\32_5\25 {
    margin: 0 0 -1px -0.3125rem;
}

.row.uniform.\32_5\25 > * {
    padding: 0.3125rem 0 0 0.3125rem;
}

.row.uniform.\32_5\25 {
    margin: -0.3125rem 0 -1px -0.3125rem;
}
```

```
.\31 2u\28small\29, .\31 2u\24\28small\29 {
    width: 100%;
    clear: none;
    margin-left: 0;
}

.\31 1u\28small\29, .\31 1u\24\28small\29 {
    width: 91.6666666667%;
    clear: none;
    margin-left: 0;
}

.\31 0u\28small\29, .\31 0u\24\28small\29 {
    width: 83.3333333333%;
    clear: none;
    margin-left: 0;
}

.\39 u\28small\29, .\39 u\24\28small\29 {
    width: 75%;
    clear: none;
    margin-left: 0;
}

.\38 u\28small\29, .\38 u\24\28small\29 {
    width: 66.6666666667%;
    clear: none;
    margin-left: 0;
}

.\37 u\28small\29, .\37 u\24\28small\29 {
    width: 58.3333333333%;
    clear: none;
    margin-left: 0;
}

.\36 u\28small\29, .\36 u\24\28small\29 {
    width: 50%;
    clear: none;
    margin-left: 0;
}

.\35 u\28small\29, .\35 u\24\28small\29 {
    width: 41.6666666667%;
    clear: none;
    margin-left: 0;
}

.\34 u\28small\29, .\34 u\24\28small\29 {
    width: 33.3333333333%;
    clear: none;
    margin-left: 0;
}

.\33 u\28small\29, .\33 u\24\28small\29 {
    width: 25%;
    clear: none;
    margin-left: 0;
}

.\32 u\28small\29, .\32 u\24\28small\29 {
    width: 16.6666666667%;
```

```
        clear: none;
        margin-left: 0;
    }

.\31 u\28small\29, .\31 u\24\28small\29 {
    width: 8.3333333333%;
    clear: none;
    margin-left: 0;
}

.\31 2u\24\28small\29 + *,
.\31 1u\24\28small\29 + *,
.\31 0u\24\28small\29 + *,
.\39 u\24\28small\29 + *,
.\38 u\24\28small\29 + *,
.\37 u\24\28small\29 + *,
.\36 u\24\28small\29 + *,
.\35 u\24\28small\29 + *,
.\34 u\24\28small\29 + *,
.\33 u\24\28small\29 + *,
.\32 u\24\28small\29 + *,
.\31 u\24\28small\29 + * {
    clear: left;
}

.\-11u\28small\29 {
    margin-left: 91.66667%;
}

.\-10u\28small\29 {
    margin-left: 83.33333%;
}

.\-9u\28small\29 {
    margin-left: 75%;
}

.\-8u\28small\29 {
    margin-left: 66.66667%;
}

.\-7u\28small\29 {
    margin-left: 58.33333%;
}

.\-6u\28small\29 {
    margin-left: 50%;
}

.\-5u\28small\29 {
    margin-left: 41.66667%;
}

.\-4u\28small\29 {
    margin-left: 33.33333%;
}

.\-3u\28small\29 {
    margin-left: 25%;
}

.\-2u\28small\29 {
```

```
        margin-left: 16.66667%;  
    }  
  
.\\-1u\\28small\\29 {  
    margin-left: 8.33333%;  
}  
  
}  
  
@media screen and (max-width: 480px) {  
  
.row > * {  
    padding: 0 0 0 1.25rem;  
}  
  
.row {  
    margin: 0 0 -1px -1.25rem;  
}  
  
.row.uniform > * {  
    padding: 1.25rem 0 0 1.25rem;  
}  
  
.row.uniform {  
    margin: -1.25rem 0 -1px -1.25rem;  
}  
  
.row.\\32_00\\25 > * {  
    padding: 0 0 0 2.5rem;  
}  
  
.row.\\32_00\\25 {  
    margin: 0 0 -1px -2.5rem;  
}  
  
.row.uniform.\\32_00\\25 > * {  
    padding: 2.5rem 0 0 2.5rem;  
}  
  
.row.uniform.\\32_00\\25 {  
    margin: -2.5rem 0 -1px -2.5rem;  
}  
  
.row.\\31_50\\25 > * {  
    padding: 0 0 0 1.875rem;  
}  
  
.row.\\31_50\\25 {  
    margin: 0 0 -1px -1.875rem;  
}  
  
.row.uniform.\\31_50\\25 > * {  
    padding: 1.875rem 0 0 1.875rem;  
}  
  
.row.uniform.\\31_50\\25 {  
    margin: -1.875rem 0 -1px -1.875rem;  
}  
  
.row.\\35_0\\25 > * {  
    padding: 0 0 0 0.625rem;  
}
```

```
.row.\35 0\25 {
    margin: 0 0 -1px -0.625rem;
}

.row.uniform.\35 0\25 > * {
    padding: 0.625rem 0 0 0.625rem;
}

.row.uniform.\35 0\25 {
    margin: -0.625rem 0 -1px -0.625rem;
}

.row.\32 5\25 > * {
    padding: 0 0 0 0.3125rem;
}

.row.\32 5\25 {
    margin: 0 0 -1px -0.3125rem;
}

.row.uniform.\32 5\25 > * {
    padding: 0.3125rem 0 0 0.3125rem;
}

.row.uniform.\32 5\25 {
    margin: -0.3125rem 0 -1px -0.3125rem;
}

.\31 2u\28xsmall\29, .\31 2u\24\28xsmall\29 {
    width: 100%;
    clear: none;
    margin-left: 0;
}

.\31 1u\28xsmall\29, .\31 1u\24\28xsmall\29 {
    width: 91.6666666667%;
    clear: none;
    margin-left: 0;
}

.\31 0u\28xsmall\29, .\31 0u\24\28xsmall\29 {
    width: 83.3333333333%;
    clear: none;
    margin-left: 0;
}

.\39 u\28xsmall\29, .\39 u\24\28xsmall\29 {
    width: 75%;
    clear: none;
    margin-left: 0;
}

.\38 u\28xsmall\29, .\38 u\24\28xsmall\29 {
    width: 66.6666666667%;
    clear: none;
    margin-left: 0;
}

.\37 u\28xsmall\29, .\37 u\24\28xsmall\29 {
    width: 58.3333333333%;
    clear: none;
```

```
        margin-left: 0;
    }

.\36 u\28xsmall\29, .\36 u\24\28xsmall\29 {
    width: 50%;
    clear: none;
    margin-left: 0;
}

.\35 u\28xsmall\29, .\35 u\24\28xsmall\29 {
    width: 41.6666666667%;
    clear: none;
    margin-left: 0;
}

.\34 u\28xsmall\29, .\34 u\24\28xsmall\29 {
    width: 33.3333333333%;
    clear: none;
    margin-left: 0;
}

.\33 u\28xsmall\29, .\33 u\24\28xsmall\29 {
    width: 25%;
    clear: none;
    margin-left: 0;
}

.\32 u\28xsmall\29, .\32 u\24\28xsmall\29 {
    width: 16.6666666667%;
    clear: none;
    margin-left: 0;
}

.\31 u\28xsmall\29, .\31 u\24\28xsmall\29 {
    width: 8.3333333333%;
    clear: none;
    margin-left: 0;
}

.\31 2u\24\28xsmall\29 + *,
.\31 1u\24\28xsmall\29 + *,
.\31 0u\24\28xsmall\29 + *,
.\39 u\24\28xsmall\29 + *,
.\38 u\24\28xsmall\29 + *,
.\37 u\24\28xsmall\29 + *,
.\36 u\24\28xsmall\29 + *,
.\35 u\24\28xsmall\29 + *,
.\34 u\24\28xsmall\29 + *,
.\33 u\24\28xsmall\29 + *,
.\32 u\24\28xsmall\29 + *,
.\31 u\24\28xsmall\29 + * {
    clear: left;
}

.\-11u\28xsmall\29 {
    margin-left: 91.66667%;
}

.\-10u\28xsmall\29 {
    margin-left: 83.33333%;
```

```
.\-\-9u\28xsmall\29 {  
    margin-left: 75%;  
}  
  
.\\-8u\28xsmall\29 {  
    margin-left: 66.66667%;  
}  
  
.\\-7u\28xsmall\29 {  
    margin-left: 58.33333%;  
}  
  
.\\-6u\28xsmall\29 {  
    margin-left: 50%;  
}  
  
.\\-5u\28xsmall\29 {  
    margin-left: 41.66667%;  
}  
  
.\\-4u\28xsmall\29 {  
    margin-left: 33.33333%;  
}  
  
.\\-3u\28xsmall\29 {  
    margin-left: 25%;  
}  
  
.\\-2u\28xsmall\29 {  
    margin-left: 16.66667%;  
}  
  
.\\-1u\28xsmall\29 {  
    margin-left: 8.33333%;  
}  
  
}  
  
/* Basic */  
  
@-ms-viewport {  
    width: device-width;  
}  
  
body {  
    -ms-overflow-style: scrollbar;  
}  
  
@media screen and (max-width: 480px) {  
  
    html, body {  
        min-width: 320px;  
    }  
  
}  
  
body {  
    background: #fff;  
}  
  
body.is-loading *, body.is-loading *:before, body.is-loading  
*:after {
```

```
        -moz-animation: none !important;
        -webkit-animation: none !important;
        -ms-animation: none !important;
        animation: none !important;
        -moz-transition: none !important;
        -webkit-transition: none !important;
        -ms-transition: none !important;
        transition: none !important;
    }

/* Type */

html {
    font-size: 13pt;
}

@media screen and (max-width: 1680px) {

    html {
        font-size: 11pt;
    }
}

@media screen and (max-width: 1280px) {

    html {
        font-size: 11pt;
    }
}

@media screen and (max-width: 980px) {

    html {
        font-size: 12pt;
    }
}

@media screen and (max-width: 736px) {

    html {
        font-size: 12pt;
    }
}

@media screen and (max-width: 480px) {

    html {
        font-size: 12pt;
    }
}

body {
    background-color: #fff;
    color: #444;
}

body, input, select, textarea {
```

```
font-family: "Poppins", sans-serif;
font-weight: 300;
font-size: 1rem;
line-height: 1.65;
}

a {
    text-decoration: underline;
}

a:hover {
    text-decoration: none;
}

strong, b {
    font-weight: 600;
}

em, i {
    font-style: italic;
}

p {
    margin: 0 0 2rem 0;
}

p.special {
    text-transform: uppercase;
    font-size: .75rem;
    font-weight: 300;
    margin: 0 0 .5rem 0;
    padding: 0 0 1rem 0;
    letter-spacing: .25rem;
}

p.special:after {
    content: '';
    position: absolute;
    margin: auto;
    right: 0;
    bottom: 0;
    left: 0;
    width: 50%;
    height: 1px;
    background-color: rgba(0, 0, 0, 0.125);
}

h1, h2, h3, h4, h5, h6 {
    font-weight: 300;
    line-height: 1.5;
    margin: 0 0 1rem 0;
}

h1 a, h2 a, h3 a, h4 a, h5 a, h6 a {
    color: inherit;
    text-decoration: none;
}
h1{
    font-size: 2.25rem;
}

h2 {
```

```
        font-size: 1.75rem;
    }

h3 {
    font-size: 1.35rem;
}

h4 {
    font-size: 1.1rem;
}

h5 {
    font-size: 0.9rem;
}

h6 {
    font-size: 0.7rem;
}

sub {
    font-size: 0.8rem;
    position: relative;
    top: 0.5rem;
}

sup {
    font-size: 0.8rem;
    position: relative;
    top: -0.5rem;
}

blockquote {
    border-left: solid 4px;
    font-style: italic;
    margin: 0 0 2rem 0;
    padding: 0.5rem 0 0.5rem 2rem;
}

code {
    border-radius: 2px;
    border: solid 1px;
    font-family: "Courier New", monospace;
    font-size: 0.9rem;
    margin: 0 0.25rem;
    padding: 0.25rem 0.65rem;
}

pre {
    -webkit-overflow-scrolling: touch;
    font-family: "Courier New", monospace;
    font-size: 0.9rem;
    margin: 0 0 2rem 0;
}

pre code {
    display: block;
    line-height: 1.75;
    padding: 1rem 1.5rem;
    overflow-x: auto;
}

hr {
```

```
border: 0;
border-bottom: solid 1px;
margin: 2rem 0;
}

hr.major {
    margin: 3rem 0;
}

.align-left {
    text-align: left;
}

.align-center {
    text-align: center;
}

.align-right {
    text-align: right;
}

input, select, textarea {
    color: #555;
}

a {
    color: #8a4680;
}

strong, b {
    color: #555;
}

h1, h2, h3, h4, h5, h6 {
    color: #555;
}

blockquote {
    border-left-color: rgba(144, 144, 144, 0.25);
}

code {
    background: rgba(144, 144, 144, 0.075);
    border-color: rgba(144, 144, 144, 0.25);
}

hr {
    border-bottom-color: rgba(144, 144, 144, 0.25);
}

/* Box */

.box {
    margin-bottom: 2rem;
    background: #FFF;
}

.box .image.fit {
    margin: 0;
    border-radius: 0;
}
```

```
.box .image.fit img {
    border-radius: 0;
}

.box header h2 {
    margin-bottom: 2rem;
}

.box header p {
    text-transform: uppercase;
    font-size: .75rem;
    font-weight: 300;
    margin: 0 0 .25rem 0;
    padding: 0 0 .75rem 0;
    letter-spacing: .25rem;
}

.box header p:after {
    content: '';
    position: absolute;
    margin: auto;
    right: 0;
    bottom: 0;
    left: 0;
    width: 50%;
    height: 1px;
    background-color: rgba(0, 0, 0, 0.125);
}

.box .content {
    padding: 3rem;
}

.box > :last-child,
.box > :last-child > :last-child,
.box > :last-child > :last-child > :last-child {
    margin-bottom: 0;
}

.box.alt {
    border: 0;
    border-radius: 0;
    padding: 0;
}

@media screen and (max-width: 736px) {

    .box .content {
        padding: 2rem;
    }
}

.box {
    border-color: rgba(144, 144, 144, 0.25);
}

/* Button */

input[type="submit"],
input[type="reset"],
input[type="button"],
```

```
button,
.button {
    -moz-appearance: none;
    -webkit-appearance: none;
    -ms-appearance: none;
    appearance: none;
    -moz-transition: background-color 0.2s ease-in-out, color 0.2s
ease-in-out;
    -webkit-transition: background-color 0.2s ease-in-out, color 0.2s
ease-in-out;
    -ms-transition: background-color 0.2s ease-in-out, color 0.2s
ease-in-out;
    transition: background-color 0.2s ease-in-out, color 0.2s ease-
in-out;
    border-radius: 2px;
    border: 0;
    cursor: pointer;
    display: inline-block;
    font-weight: 300;
    height: 2.85rem;
    line-height: 2.95rem;
    padding: 0 1.5rem;
    text-align: center;
    text-decoration: none;
    text-transform: uppercase;
    white-space: nowrap;
}

input[type="submit"].icon,
input[type="reset"].icon,
input[type="button"].icon,
button.icon,
.button.icon {
    padding-left: 1.35rem;
}

input[type="submit"].icon:before,
input[type="reset"].icon:before,
input[type="button"].icon:before,
button.icon:before,
.button.icon:before {
    margin-right: 0.5rem;
}

input[type="submit"].fit,
input[type="reset"].fit,
input[type="button"].fit,
button.fit,
.button.fit {
    display: block;
    margin: 0 0 1rem 0;
    width: 100%;
}

input[type="submit"].small,
input[type="reset"].small,
input[type="button"].small,
button.small,
.button.small {
    font-size: 0.8rem;
}
```

```
input[type="submit"].big,
input[type="reset"].big,
input[type="button"].big,
button.big,
.button.big {
    font-size: 1.35rem;
}

input[type="submit"].disabled, input[type="submit"]:disabled,
input[type="reset"].disabled,
input[type="reset"]:disabled,
input[type="button"].disabled,
input[type="button"]:disabled,
button.disabled,
button:disabled,
.button.disabled,
.button:disabled {
    -moz-pointer-events: none;
    -webkit-pointer-events: none;
    -ms-pointer-events: none;
    pointer-events: none;
    opacity: 0.25;
}

@media screen and (max-width: 480px) {

    input[type="submit"],
    input[type="reset"],
    input[type="button"],
    button,
    .button {
        padding: 0;
        width: 100%;
    }
}

input[type="submit"],
input[type="reset"],
input[type="button"],
button,
.button {
    background-color: #f2f2f2;
    color: #000 !important;
}

input[type="submit"]:hover,
input[type="reset"]:hover,
input[type="button"]:hover,
button:hover,
.button:hover {
    background-color: white;
}

input[type="submit"]:active,
input[type="reset"]:active,
input[type="button"]:active,
button:active,
.button:active {
    background-color: #e5e5e5;
}
```

```
input[type="submit"].alt,
input[type="reset"].alt,
input[type="button"].alt,
button.alt,
.button.alt {
    background-color: transparent;
    box-shadow: inset 0 0 0 2px rgba(144, 144, 144, 0.25);
    color: #555 !important;
}

input[type="submit"].alt:hover,
input[type="reset"].alt:hover,
input[type="button"].alt:hover,
button.alt:hover,
.button.alt:hover {
    background-color: rgba(144, 144, 144, 0.075);
}

input[type="submit"].alt:active,
input[type="reset"].alt:active,
input[type="button"].alt:active,
button.alt:active,
.button.alt:active {
    background-color: rgba(144, 144, 144, 0.2);
}

input[type="submit"].alt.icon:before,
input[type="reset"].alt.icon:before,
input[type="button"].alt.icon:before,
button.alt.icon:before,
.button.alt.icon:before {
    color: #bbb;
}

input[type="submit"].special,
input[type="reset"].special,
input[type="button"].special,
button.special,
.button.special {
    background-color: #8a4680;
    color: #ffffff !important;
}

input[type="submit"].special:hover,
input[type="reset"].special:hover,
input[type="button"].special:hover,
button.special:hover,
.button.special:hover {
    background-color: #9b4f90;
}

input[type="submit"].special:active,
input[type="reset"].special:active,
input[type="button"].special:active,
button.special:active,
.button.special:active {
    background-color: #793d70;
}

/* Form */

form {
```

```
        margin: 0 0 2rem 0;
    }

label {
    display: block;
    font-size: 0.9rem;
    font-weight: 700;
    margin: 0 0 1rem 0;
}

input[type="text"],
input[type="password"],
input[type="email"],
select,
textarea {
    -moz-appearance: none;
    -webkit-appearance: none;
    -ms-appearance: none;
    appearance: none;
    border-radius: 2px;
    border: none;
    border: solid 1px;
    color: inherit;
    display: block;
    outline: 0;
    padding: 0 1rem;
    text-decoration: none;
    width: 100%;
}

input[type="text"]:invalid,
input[type="password"]:invalid,
input[type="email"]:invalid,
select:invalid,
textarea:invalid {
    box-shadow: none;
}

.select-wrapper {
    text-decoration: none;
    display: block;
    position: relative;
}

.select-wrapper:before {
    -moz-osx-font-smoothing: grayscale;
    -webkit-font-smoothing: antialiased;
    font-family: FontAwesome;
    font-style: normal;
    font-weight: normal;
    text-transform: none !important;
}

.select-wrapper:before {
    content: '\f078';
    display: block;
    height: 2.75rem;
    line-height: 2.75rem;
    pointer-events: none;
    position: absolute;
    right: 0;
    text-align: center;
```

```
        top: 0;
        width: 2.75rem;
    }

.select-wrapper select::-ms-expand {
    display: none;
}

input[type="text"],
input[type="password"],
input[type="email"],
select {
    height: 2.75rem !important;
}

textarea {
    padding: 0.75rem 1rem;
}

input[type="checkbox"],
input[type="radio"] {
    -moz-appearance: none;
    -webkit-appearance: none;
    -ms-appearance: none;
    appearance: none;
    display: block;
    float: left;
    margin-right: -2rem;
    opacity: 0;
    width: 1rem;
    z-index: -1;
}

input[type="checkbox"] + label,
input[type="radio"] + label {
    text-decoration: none;
    cursor: pointer;
    display: inline-block;
    font-size: 1rem;
    font-weight: 400;
    padding-left: 2.4rem;
    padding-right: 0.75rem;
    position: relative;
}

input[type="checkbox"] + label:before,
input[type="radio"] + label:before {
    -moz-osx-font-smoothing: grayscale;
    -webkit-font-smoothing: antialiased;
    font-family: FontAwesome;
    font-style: normal;
    font-weight: normal;
    text-transform: none !important;
}

input[type="checkbox"] + label:before,
input[type="radio"] + label:before {
    border-radius: 2px;
    border: solid 1px;
    content: '';
    display: inline-block;
    height: 1.65rem;
```

```
        left: 0;
        line-height: 1.58125rem;
        position: absolute;
        text-align: center;
        top: 0;
        width: 1.65rem;
    }

input[type="checkbox"]:checked + label:before,
input[type="radio"]:checked + label:before {
    content: '\f00c';
}

input[type="checkbox"] + label:before {
    border-radius: 2px;
}

input[type="radio"] + label:before {
    border-radius: 100%;
}

::webkit-input-placeholder {
    opacity: 1.0;
}

:-moz-placeholder {
    opacity: 1.0;
}

:-moz-placeholder {
    opacity: 1.0;
}

:-ms-placeholder {
    opacity: 1.0;
}

.formerize-placeholder {
    opacity: 1.0;
}

label {
    color: #555;
}

input[type="text"],
input[type="password"],
input[type="email"],
select,
textarea {
    background: rgba(144, 144, 144, 0.075);
    border-color: rgba(144, 144, 144, 0.25);
}

input[type="text"]:focus,
input[type="password"]:focus,
input[type="email"]:focus,
select:focus,
textarea:focus {
    border-color: #8a4680;
    box-shadow: 0 0 0 1px #8a4680;
}
```

```
.select-wrapper:before {
    color: rgba(144, 144, 144, 0.25);
}

input[type="checkbox"] + label,
input[type="radio"] + label {
    color: #444 !important;
}

input[type="checkbox"] + label:before,
input[type="radio"] + label:before {
    background: rgba(144, 144, 144, 0.075);
    border-color: rgba(144, 144, 144, 0.25);
}

input[type="checkbox"]:checked + label:before,
input[type="radio"]:checked + label:before {
    background-color: #8a4680;
    border-color: #8a4680;
    color: #ffffff;
}

input[type="checkbox"]:focus + label:before,
input[type="radio"]:focus + label:before {
    border-color: #8a4680;
    box-shadow: 0 0 0 1px #8a4680;
}

::webkit-input-placeholder {
    color: #bbb !important;
}

:-moz-placeholder {
    color: #bbb !important;
}

:-moz-placeholder {
    color: #bbb !important;
}

:-ms-placeholder {
    color: #bbb !important;
}

.formerize-placeholder {
    color: #bbb !important;
}

/* Icon */

.icon {
    text-decoration: none;
    border-bottom: none;
    position: relative;
}

.icon:before {
    -moz-osx-font-smoothing: grayscale;
    -webkit-font-smoothing: antialiased;
    font-family: FontAwesome;
    font-style: normal;
}
```

```
        font-weight: normal;
        text-transform: none !important;
    }

.icon > .label {
    display: none;
}

/* Image */

.image {
    border-radius: 2px;
    border: 0;
    display: inline-block;
    position: relative;
}

.image img {
    border-radius: 2px;
    display: block;
}

.image.left, .image.right {
    max-width: 40%;
}

.image.left img, .image.right img {
    width: 100%;
}

.image.left {
    float: left;
    margin: 0 1.5rem 1rem 0;
    top: 0.25rem;
}

.image.right {
    float: right;
    margin: 0 0 1rem 1.5rem;
    top: 0.25rem;
}

.image.fit {
    display: block;
    margin: 0 0 2rem 0;
    width: 100%;
}

.image.fit img {
    width: 100%;
}

.image.main {
    display: block;
    margin: 0 0 3rem 0;
    width: 100%;
}

.image.main img {
    width: 100%;
}
```

```
/* List */

ol {
    list-style: decimal;
    margin: 0 0 2rem 0;
    padding-left: 1.25rem;
}

ol li {
    padding-left: 0.25rem;
}

ul {
    list-style: disc;
    margin: 0 0 2rem 0;
    padding-left: 1rem;
}

ul li {
    padding-left: 0.5rem;
}

ul.alt {
    list-style: none;
    padding-left: 0;
}

ul.alt li {
    border-top: solid 1px;
    padding: 0.5rem 0;
}

ul.alt li:first-child {
    border-top: 0;
    padding-top: 0;
}

ul.icons {
    cursor: default;
    list-style: none;
    padding-left: 0;
}

ul.icons li {
    display: inline-block;
    padding: 0 1rem 0 0;
}

ul.icons li:last-child {
    padding-right: 0;
}

ul.icons li .icon:before {
    font-size: 1.3rem;
}

ul.actions {
    cursor: default;
    list-style: none;
    padding-left: 0;
}
```

```
ul.actions li {
    display: inline-block;
    padding: 0 1rem 0 0;
    vertical-align: middle;
}

ul.actions li:last-child {
    padding-right: 0;
}

ul.actions.small li {
    padding: 0 0.5rem 0 0;
}

ul.actions.vertical li {
    display: block;
    padding: 1rem 0 0 0;
}

ul.actions.vertical li:first-child {
    padding-top: 0;
}

ul.actions.vertical li > * {
    margin-bottom: 0;
}

ul.actions.vertical.small li {
    padding: 0.5rem 0 0 0;
}

ul.actions.vertical.small li:first-child {
    padding-top: 0;
}

ul.actions.fit {
    display: table;
    margin-left: -1rem;
    padding: 0;
    table-layout: fixed;
    width: calc(100% + 1rem);
}

ul.actions.fit li {
    display: table-cell;
    padding: 0 0 0 1rem;
}

ul.actions.fit li > * {
    margin-bottom: 0;
}

ul.actions.fit.small {
    margin-left: -0.5rem;
    width: calc(100% + 0.5rem);
}

ul.actions.fit.small li {
    padding: 0 0 0 0.5rem;
}

@media screen and (max-width: 480px) {
```

```
        ul.actions {
            margin: 0 0 2rem 0;
        }

        ul.actions li {
            padding: 1rem 0 0 0;
            display: block;
            text-align: center;
            width: 100%;
        }

        ul.actions li:first-child {
            padding-top: 0;
        }

        ul.actions li > * {
            width: 100%;
            margin: 0 !important;
        }

        ul.actions li > *.icon:before {
            margin-left: -2rem;
        }

        ul.actions.small li {
            padding: 0.5rem 0 0 0;
        }

        ul.actions.small li:first-child {
            padding-top: 0;
        }

    }

dl {
    margin: 0 0 2rem 0;
}

dl dt {
    display: block;
    font-weight: 700;
    margin: 0 0 1rem 0;
}

dl dd {
    margin-left: 2rem;
}

ul.alt li {
    border-top-color: rgba(144, 144, 144, 0.25);
}

/* Section/Article */

section.special, article.special {
    text-align: center;
}

header p {
    position: relative;
    margin: 0 0 1.5rem 0;
}
```

```
}

header h2 + p {
    font-size: 1.25rem;
    margin-top: -1rem;
}

header h3 + p {
    font-size: 1.1rem;
    margin-top: -0.8rem;
}

header h4 + p,
header h5 + p,
header h6 + p {
    font-size: 0.9rem;
    margin-top: -0.6rem;
}

header p {
    color: #bbb;
}

/* Table */

.table-wrapper {
    -webkit-overflow-scrolling: touch;
    overflow-x: auto;
}

table {
    margin: 0 0 2rem 0;
    width: 100% !important;
    max-width: 400px;
    margin-left: auto;
    margin-right: auto;
}

    table tbody tr {
        border: solid 1px;
        border-left: 0;
        border-right: 0;
    }

    table td {
        padding: 0.75rem 0.75rem;
    }

    table th {
        font-size: 1.2rem;
        font-weight: 700;
        padding: 0 0.75rem 0.75rem 0.75rem;
        text-align: left;
    }

    table thead {
        border-bottom: solid 2px;
    }

    table tfoot {
        border-top: solid 2px;
    }
}
```

```
table.alt {  
    border-collapse: separate;  
}  
  
table.alt tbody tr td {  
    border: solid 1px;  
    border-left-width: 0;  
    border-top-width: 0;  
}  
  
table.alt tbody tr td:first-child {  
    border-left-width: 1px;  
}  
  
table.alt tbody tr:first-child td {  
    border-top-width: 1px;  
}  
  
table.alt thead {  
    border-bottom: 0;  
}  
  
table.alt tfoot {  
    border-top: 0;  
}  
  
table tbody tr {  
    border-color: rgba(144, 144, 144, 0.25);  
}  
  
table tbody tr:nth-child(2n + 1) {  
    background-color: rgba(144, 144, 144, 0.075);  
}  
  
table th {  
    color: #555;  
}  
  
table thead {  
    border-bottom-color: rgba(144, 144, 144, 0.25);  
}  
  
table tfoot {  
    border-top-color: rgba(144, 144, 144, 0.25);  
}  
  
table.alt tbody tr td {  
    border-color: rgba(144, 144, 144, 0.25);  
}  
  
/* Wrapper */  
  
.wrapper {  
    padding: 6rem 0 4rem 0 ;  
    position: relative;  
    margin: 0;  
}  
  
.wrapper > .inner {  
    margin: 0 auto;
```

```
        width: 80rem;
    }

.wrapper.style1 {
    background-color: #8a4680;
    color: #e2d1df;
}

.wrapper.style1 input, .wrapper.style1 select,
.wrapper.style1 textarea {
    color: #ffffff;
}

.wrapper.style1 a {
    color: #ffffff;
}

.wrapper.style1 strong, .wrapper.style1 b {
    color: #ffffff;
}

.wrapper.style1 h1, .wrapper.style1 h2, .wrapper.style1
h3, .wrapper.style1 h4, .wrapper.style1 h5, .wrapper.style1 h6 {
    color: #ffffff;
}

.wrapper.style1 blockquote {
    border-left-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style1 code {
    background: rgba(255, 255, 255, 0.075);
    border-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style1 hr {
    border-bottom-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style1 .box {
    border-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style1 input[type="submit"],
.wrapper.style1 input[type="reset"],
.wrapper.style1 input[type="button"],
.wrapper.style1 button,
.wrapper.style1 .button {
    background-color: #f2f2f2;
    color: #000 !important;
}

.wrapper.style1 input[type="submit"]:hover,
.wrapper.style1 input[type="reset"]:hover,
.wrapper.style1 input[type="button"]:hover,
.wrapper.style1 button:hover,
.wrapper.style1 .button:hover {
    background-color: white;
}

.wrapper.style1 input[type="submit"]:active,
.wrapper.style1 input[type="reset"]:active,
```

```
.wrapper.style1 input[type="button"]:active,
.wrapper.style1 button:active,
.wrapper.style1 .button:active {
    background-color: #e5e5e5;
}

.wrapper.style1 input[type="submit"].alt,
.wrapper.style1 input[type="reset"].alt,
.wrapper.style1 input[type="button"].alt,
.wrapper.style1 button.alt,
.wrapper.style1 .button.alt {
    background-color: transparent;
    box-shadow: inset 0 0 0 2px rgba(255, 255,
255, 0.25);
    color: #ffffff !important;
}

input[type="submit"].alt:hover,
input[type="reset"].alt:hover,
input[type="button"].alt:hover,
0.075);

input[type="submit"].alt:active,
input[type="reset"].alt:active,
input[type="button"].alt:active,
0.2);
}

.input[type="submit"].alt.icon:before,
.input[type="reset"].alt.icon:before,
.input[type="button"].alt.icon:before,
.wrapper.style1 button.alt.icon:before,
.wrapper.style1 .button.alt.icon:before {
    color: #d0b5cc;
}

.wrapper.style1 input[type="submit"].special,
.wrapper.style1 input[type="reset"].special,
.wrapper.style1 input[type="button"].special,
.wrapper.style1 button.special,
.wrapper.style1 .button.special {
    background-color: #ffffff;
    color: #8a4680 !important;
}

.wrapper.style1 label {
```

```
        color: #ffffff;
    }

.wrapper.style1 input[type="text"],
.wrapper.style1 input[type="password"],
.wrapper.style1 input[type="email"],
.wrapper.style1 select,
.wrapper.style1 textarea {
    background: rgba(255, 255, 255, 0.075);
    border-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style1 input[type="text"]:focus,
.wrapper.style1 input[type="password"]:focus,
.wrapper.style1 input[type="email"]:focus,
.wrapper.style1 select:focus,
.wrapper.style1 textarea:focus {
    border-color: #ffffff;
    box-shadow: 0 0 0 1px #ffffff;
}

.wrapper.style1 .select-wrapper:before {
    color: rgba(255, 255, 255, 0.25);
}

.wrapper.style1 input[type="checkbox"] + label,
.wrapper.style1 input[type="radio"] + label {
    color: #e2d1df;
}

.wrapper.style1 input[type="checkbox"] +
label:before,
.wrapper.style1 input[type="radio"] + label:before
{
    background: rgba(255, 255, 255, 0.075);
    border-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style1 input[type="checkbox"]:checked +
label:before,
.wrapper.style1 input[type="radio"]:checked + label:before
{
    background-color: #ffffff;
    border-color: #ffffff;
    color: #8a4680;
}

.wrapper.style1 input[type="checkbox"]:focus +
label:before,
.wrapper.style1 input[type="radio"]:focus + label:before {
    border-color: #ffffff;
    box-shadow: 0 0 0 1px #ffffff;
}

.wrapper.style1 ::-webkit-input-placeholder {
    color: #d0b5cc !important;
}

.wrapper.style1 :-moz-placeholder {
    color: #d0b5cc !important;
}
```

```
.wrapper.style1 ::-moz-placeholder {
    color: #d0b5cc !important;
}

.wrapper.style1 :-ms-input-placeholder {
    color: #d0b5cc !important;
}

.wrapper.style1 .formerize-placeholder {
    color: #d0b5cc !important;
}

.wrapper.style1 ul.alt li {
    border-top-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style1 header p {
    color: #d0b5cc;
}

.wrapper.style1 table tbody tr {
    border-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style1 table tbody tr:nth-child(2n + 1) {
    background-color: rgba(255, 255, 255, 0.075);
}

.wrapper.style1 table th {
    color: #ffffff;
}

.wrapper.style1 table thead {
    border-bottom-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style1 table tfoot {
    border-top-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style1 table.alt tbody tr td {
    border-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style2 {
    background-color: #f2f2f2;
    color: #444;
    min-height: 100%;
}

.wrapper.style2 input, .wrapper.style2 select,
.wrapper.style2 textarea {
    color: #000;
}

.wrapper.style2 a {
    color: #8a4680;
}

.wrapper.style2 strong, .wrapper.style2 b {
    color: #000;
}
```

```
.wrapper.style2 h1, .wrapper.style2 h2, .wrapper.style2  
h3, .wrapper.style2 h4, .wrapper.style2 h5, .wrapper.style2 h6 {  
    color: #000;  
}  
  
.wrapper.style2 blockquote {  
    border-left-color: rgba(0, 0, 0, 0.15);  
}  
  
.wrapper.style2 code {  
    background: rgba(0, 0, 0, 0.075);  
    border-color: rgba(0, 0, 0, 0.15);  
}  
  
.wrapper.style2 hr {  
    border-bottom-color: rgba(0, 0, 0, 0.15);  
}  
  
.wrapper.style2 .box {  
    border-color: rgba(0, 0, 0, 0.15);  
}  
  
.wrapper.style2 input[type="submit"],  
.wrapper.style2 input[type="reset"],  
.wrapper.style2 input[type="button"],  
.wrapper.style2 button,  
.wrapper.style2 .button {  
    background-color: #000;  
    color: #f2f2f2 !important;  
}  
  
.wrapper.style2 input[type="submit"].alt,  
.wrapper.style2 input[type="reset"].alt,  
.wrapper.style2 input[type="button"].alt,  
.wrapper.style2 button.alt,  
.wrapper.style2 .button.alt {  
    background-color: transparent;  
    box-shadow: inset 0 0 0 2px rgba(0, 0, 0,  
0.15);  
    color: #000 !important;  
}  
  
input[type="submit"].alt:hover,  
input[type="reset"].alt:hover,  
input[type="button"].alt:hover,  
0.075);  
  
.wrapper.style2  
.wrapper.style2  
.wrapper.style2  
.wrapper.style2 button.alt:hover,  
.wrapper.style2 .button.alt:hover {  
    background-color: rgba(0, 0, 0,  
0.075);  
}  
  
.wrapper.style2  
.wrapper.style2  
.wrapper.style2  
.wrapper.style2 button.alt:active,
```

```
.wrapper.style2 .button.alt:active {
    background-color: rgba(0, 0, 0, 0.2);
}

.wrapper.style2
input[type="submit"].alt.icon:before,
.wrapper.style2
input[type="reset"].alt.icon:before,
.wrapper.style2
input[type="button"].alt.icon:before,
    .wrapper.style2 button.alt.icon:before,
    .wrapper.style2 .button.alt.icon:before {
        color: #999999;
    }

.wrapper.style2 input[type="submit"].special,
.wrapper.style2 input[type="reset"].special,
.wrapper.style2 input[type="button"].special,
.wrapper.style2 button.special,
.wrapper.style2 .button.special {
    background-color: #8a4680;
    color: #ffffff !important;
}

.wrapper.style2
input[type="submit"].special:hover,
.wrapper.style2
input[type="reset"].special:hover,
.wrapper.style2
input[type="button"].special:hover,
    .wrapper.style2 button.special:hover,
    .wrapper.style2 .button.special:hover {
        background-color: #9b4f90;
    }

.wrapper.style2
input[type="submit"].special:active,
.wrapper.style2
input[type="reset"].special:active,
.wrapper.style2
input[type="button"].special:active,
    .wrapper.style2 button.special:active,
    .wrapper.style2 .button.special:active {
        background-color: #793d70;
    }

.wrapper.style2 input[type="submit"].shopbutton,
.wrapper.style2 input[type="reset"].shopbutton,
.wrapper.style2 input[type="button"].shopbutton,
.wrapper.style2 button.shopbutton,
.wrapper.style2 .button.shopbutton {
    background-color: gray;
    color: #000 !important;
}

.wrapper.style2 label {
    color: #000;
}

.wrapper.style2 input[type="text"],
.wrapper.style2 input[type="password"],
.wrapper.style2 input[type="email"],
```

```
.wrapper.style2 select,
.wrapper.style2 textarea {
    background: rgba(0, 0, 0, 0.075);
    border-color: rgba(0, 0, 0, 0.15);
height: 2.75rem;
}

.wrapper.style2 input[type="text"]:focus,
.wrapper.style2 input[type="password"]:focus,
.wrapper.style2 input[type="email"]:focus,
.wrapper.style2 select:focus,
.wrapper.style2 textarea:focus {
    border-color: #8a4680;
    box-shadow: 0 0 0 1px #8a4680;
}

.wrapper.style2 .select-wrapper:before {
    color: rgba(0, 0, 0, 0.15);
}

.wrapper.style2 input[type="checkbox"] + label,
.wrapper.style2 input[type="radio"] + label {
    color: #a6a6a6;
}

.wrapper.style2 input[type="checkbox"] +
.wrapper.style2 input[type="radio"] + label:before
{
    background: rgba(0, 0, 0, 0.075);
    border-color: rgba(0, 0, 0, 0.15);
}

.wrapper.style2 input[type="checkbox"]:checked +
.wrapper.style2 input[type="radio"]:checked + label:before
{
    background-color: #8a4680;
    border-color: #8a4680;
    color: #ffffff;
}

.wrapper.style2 input[type="checkbox"]:focus +
.wrapper.style2 input[type="radio"]:focus + label:before {
    border-color: #8a4680;
    box-shadow: 0 0 0 1px #8a4680;
}

.wrapper.style2 ::-webkit-input-placeholder {
    color: #999999 !important;
}

.wrapper.style2 ::-moz-placeholder {
    color: #999999 !important;
}

.wrapper.style2 ::-moz-placeholder {
    color: #999999 !important;
}

.wrapper.style2 :-ms-input-placeholder {
```

```
        color: #999999 !important;
    }

.wrapper.style2 .formerize-placeholder {
    color: #999999 !important;
}

.wrapper.style2 ul.alt li {
    border-top-color: rgba(0, 0, 0, 0.15);
}

.wrapper.style2 header p {
    color: #999999;
}

.wrapper.style2 table tbody tr {
    border-color: transparent;
}

.wrapper.style2 table tbody tr:nth-child(2n + 1) {
    background-color: transparent;
}

.wrapper.style2 table th {
    color: #000;
}

.wrapper.style2 table thead {
    border-bottom-color: rgba(0, 0, 0, 0.15);
}

.wrapper.style2 table tfoot {
    border-top-color: rgba(0, 0, 0, 0.15);
}

.wrapper.style2 table.alt tbody tr td {
    border-color: rgba(0, 0, 0, 0.15);
}

.wrapper.style3 {
    background-color: #000;
    color: #bfbfbf;
    background-image: url(..../images1/headerBG.jpg);
background-size: 100%;
    background-attachment: fixed;
    background-position: top;
    position: relative;
min-height: 100%;
}

.wrapper.style3 input, .wrapper.style3 select,
.wrapper.style3 textarea {
    color: #ffffff;
}

.wrapper.style3 a {
    color: #8a4680;
}

.wrapper.style3 strong, .wrapper.style3 b {
    color: #ffffff;
}
```

```
.wrapper.style3 h1, .wrapper.style3 h2, .wrapper.style3  
h3, .wrapper.style3 h4, .wrapper.style3 h5, .wrapper.style3 h6 {  
    color: #ffffff;  
}  
  
.wrapper.style3 blockquote {  
    border-left-color: rgba(255, 255, 255, 0.25);  
}  
  
.wrapper.style3 code {  
    background: rgba(255, 255, 255, 0.075);  
    border-color: rgba(255, 255, 255, 0.25);  
}  
  
.wrapper.style3 hr {  
    border-bottom-color: rgba(255, 255, 255, 0.25);  
}  
  
.wrapper.style3 .box {  
    border-color: rgba(255, 255, 255, 0.25);  
}  
  
.wrapper.style3 input[type="submit"],  
.wrapper.style3 input[type="reset"],  
.wrapper.style3 input[type="button"],  
.wrapper.style3 button,  
.wrapper.style3 .button {  
    background-color: #f2f2f2;  
    color: #000 !important;  
}  
  
.wrapper.style3 input[type="submit"]:hover,  
.wrapper.style3 input[type="reset"]:hover,  
.wrapper.style3 input[type="button"]:hover,  
.wrapper.style3 button:hover,  
.wrapper.style3 .button:hover {  
    background-color: white;  
}  
  
.wrapper.style3 input[type="submit"]:active,  
.wrapper.style3 input[type="reset"]:active,  
.wrapper.style3 input[type="button"]:active,  
.wrapper.style3 button:active,  
.wrapper.style3 .button:active {  
    background-color: #e5e5e5;  
}  
  
.wrapper.style3 input[type="submit"].alt,  
.wrapper.style3 input[type="reset"].alt,  
.wrapper.style3 input[type="button"].alt,  
.wrapper.style3 button.alt,  
.wrapper.style3 .button.alt {  
    background-color: transparent;  
    box-shadow: inset 0 0 0 2px rgba(255, 255,  
255, 0.25);  
    color: #ffffff !important;  
}  
  
.wrapper.style3  
input[type="submit"].alt:hover,
```

```
.wrapper.style3  
input[type="reset"].alt:hover,  
input[type="button"].alt:hover,  
  
0.075);  
  
.wrapper.style3  
input[type="submit"].alt:active,  
input[type="reset"].alt:active,  
input[type="button"].alt:active,  
  
0.2);  
  
.wrapper.style3  
input[type="submit"].alt.icon:before,  
input[type="reset"].alt.icon:before,  
input[type="button"].alt.icon:before,  
  
.wrapper.style3 button.alt.icon:before,  
.wrapper.style3 .button.alt.icon:before {  
    color: #999999;  
}  
  
.wrapper.style3 input[type="submit"].special,  
.wrapper.style3 input[type="reset"].special,  
.wrapper.style3 input[type="button"].special,  
.wrapper.style3 button.special,  
.wrapper.style3 .button.special {  
    background-color: #8a4680;  
    color: #ffffff !important;  
}  
  
.wrapper.style3  
input[type="submit"].special:hover,  
input[type="reset"].special:hover,  
input[type="button"].special:hover,  
  
.wrapper.style3 button.special:hover,  
.wrapper.style3 .button.special:hover {  
    background-color: #9b4f90;  
}  
  
.wrapper.style3  
input[type="submit"].special:active,  
input[type="reset"].special:active,  
input[type="button"].special:active,  
  
.wrapper.style3 button.special:active,  
.wrapper.style3 .button.special:active {  
    background-color: #793d70;
```

```

        }

.wrapper.style3 label {
    color: #ffffff;
}

.wrapper.style3 input[type="text"],
.wrapper.style3 input[type="password"],
.wrapper.style3 input[type="email"],
.wrapper.style3 select,
.wrapper.style3 textarea {
    background: rgba(255, 255, 255, 0.075);
    border-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style3 input[type="text"]:focus,
.wrapper.style3 input[type="password"]:focus,
.wrapper.style3 input[type="email"]:focus,
.wrapper.style3 select:focus,
.wrapper.style3 textarea:focus {
    border-color: #8a4680;
    box-shadow: 0 0 0 1px #8a4680;
}

.wrapper.style3 .select-wrapper:before {
    color: rgba(255, 255, 255, 0.25);
}

.wrapper.style3 input[type="checkbox"] + label,
.wrapper.style3 input[type="radio"] + label {
    color: #bfbfbf;
}

.wrapper.style3 input[type="checkbox"] +
.wrapper.style3 input[type="radio"] + label:before {
    background: rgba(255, 255, 255, 0.075);
    border-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style3 input[type="checkbox"]:checked +
.wrapper.style3 input[type="radio"]:checked + label:before {
    background-color: #8a4680;
    border-color: #8a4680;
    color: #ffffff;
}

.wrapper.style3 input[type="checkbox"]:focus +
.wrapper.style3 input[type="radio"]:focus + label:before {
    border-color: #8a4680;
    box-shadow: 0 0 0 1px #8a4680;
}

.wrapper.style3 ::-webkit-input-placeholder {
    color: #999999 !important;
}

.wrapper.style3 :-moz-placeholder {
}

```

```
        color: #999999 !important;
    }

.wrapper.style3 ::-moz-placeholder {
    color: #999999 !important;
}

.wrapper.style3 :-ms-input-placeholder {
    color: #999999 !important;
}

.wrapper.style3 .formerize-placeholder {
    color: #999999 !important;
}

.wrapper.style3 ul.alt li {
    border-top-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style3 header p {
    color: #999999;
}

.wrapper.style3 table tbody tr {
    border-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style3 table tbody tr:nth-child(2n + 1) {
    background-color: rgba(255, 255, 255, 0.075);
}

.wrapper.style3 table th {
    color: #ffffff;
}

.wrapper.style3 table thead {
    border-bottom-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style3 table tfoot {
    border-top-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style3 table.alt tbody tr td {
    border-color: rgba(255, 255, 255, 0.25);
}

.wrapper.style3 .inner {
    position: relative;
    z-index: 2;
}

.wrapper.style3 p {
    text-transform: uppercase;
    font-size: .75rem;
    font-weight: 300;
    margin: 0 0 .5rem 0;
    padding: 0 0 1rem 0;
    letter-spacing: .25rem;
}

.wrapper.style3 p:after {
```

```
        content: '';
        position: absolute;
        margin: auto;
        right: 0;
        bottom: 0;
        left: 0;
        width: 50%;
        height: 1px;
        background-color: rgba(255, 255, 255, 0.5);
    }

.wrapper.style3:before {
    -moz-transition: opacity 3s ease;
    -webkit-transition: opacity 3s ease;
    -ms-transition: opacity 3s ease;
    transition: opacity 3s ease;
    -moz-transition-delay: 1.25s;
    -webkit-transition-delay: 1.25s;
    -ms-transition-delay: 1.25s;
    transition-delay: 1.25s;
    content: '';
    display: block;
    background-color: rgba(0, 0, 0, 0.95);
    height: 100%;
    left: 0;
    opacity: 0.75;
    position: absolute;
    top: 0;
    width: 100%;
    z-index: 1;
}

@media screen and (max-width: 1280px) {

    .wrapper > .inner {
        width: 75rem;
    }
}

@media screen and (max-width: 980px) {

    .wrapper > .inner {
        width: 90%;
    }

    .wrapper.style3 {
        background-attachment: scroll;
    }
}

@media screen and (max-width: 736px) {

    .wrapper {
        padding: 3rem 0 1rem 0 ;
    }
}

/* Flexgrid */
```

```
.grid-style {  
    width: 100%;  
    margin: 0 0 2.5rem 0;  
    display: -moz-flex;  
    display: -webkit-flex;  
    display: -ms-flex;  
    display: flex;  
    -moz-flex-wrap: wrap;  
    -webkit-flex-wrap: wrap;  
    -ms-flex-wrap: wrap;  
    flex-wrap: wrap;  
    -moz-align-items: stretch;  
    -webkit-align-items: stretch;  
    -ms-align-items: stretch;  
    align-items: stretch;  
}  
  
.grid-style > * {  
    -moz-flex-shrink: 1;  
    -webkit-flex-shrink: 1;  
    -ms-flex-shrink: 1;  
    flex-shrink: 1;  
    -moz-flex-grow: 0;  
    -webkit-flex-grow: 0;  
    -ms-flex-grow: 0;  
    flex-grow: 0;  
}  
  
.grid-style > * {  
    width: 50%;  
}  
  
.grid-style > * {  
    padding: 2rem;  
    width: calc(50% + 2rem);  
}  
  
.grid-style > :nth-child(-n + 2) {  
    padding-top: 0;  
}  
  
.grid-style > :nth-last-child(-n + 2) {  
    padding-bottom: 0;  
}  
  
.grid-style > :nth-child(2n + 1) {  
    padding-left: 0;  
}  
  
.grid-style > :nth-child(2n) {  
    padding-right: 0;  
}  
  
.grid-style > :nth-child(2n + 1),  
.grid-style > :nth-child(2n) {  
    width: calc(50% + 0rem);  
}  
  
.grid-style .box {  
    margin: 0;  
}
```

```
@media screen and (max-width: 980px) {  
  
    .grid-style > * {  
        width: 100%;  
    }  
  
    .grid-style > * {  
        padding: 1rem;  
        width: calc(50% + 1rem);  
    }  
  
    .grid-style > :nth-child(-n + 2) {  
        padding-top: 1rem;  
    }  
  
    .grid-style > :nth-last-child(-n + 2) {  
        padding-bottom: 1rem;  
    }  
  
    .grid-style > :nth-child(2n + 1) {  
        padding-left: 1rem;  
    }  
  
    .grid-style > :nth-child(2n) {  
        padding-right: 1rem;  
    }  
  
    .grid-style > :nth-child(2n + 1),  
    .grid-style > :nth-child(2n) {  
        padding: 1rem;  
        width: calc(100% + 2rem);  
    }  
  
    .grid-style > * {  
        padding: 1rem;  
        width: calc(100% + 2rem);  
    }  
  
    .grid-style > :nth-child(-n + 1) {  
        padding-top: 0;  
    }  
  
    .grid-style > :nth-last-child(-n + 1) {  
        padding-bottom: 0;  
    }  
  
    .grid-style > :nth-child(1n + 1) {  
        padding-left: 0;  
    }  
  
    .grid-style > :nth-child(1n) {  
        padding-right: 0;  
    }  
  
    .grid-style > :nth-child(1n + 1),  
    .grid-style > :nth-child(1n) {  
        width: calc(100% + 1rem);  
    }  
  
}  
  
.gallery {
```

```
width: 100%;  
margin: 2.5rem 0 2.5rem 0;  
display: -moz-flex;  
display: -webkit-flex;  
display: -ms-flex;  
display: flex;  
-moz-flex-wrap: wrap;  
-webkit-flex-wrap: wrap;  
-ms-flex-wrap: wrap;  
flex-wrap: wrap;  
-moz-align-items: stretch;  
-webkit-align-items: stretch;  
-ms-align-items: stretch;  
align-items: stretch;  
}  
  
.gallery > * {  
    -moz-flex-shrink: 1;  
    -webkit-flex-shrink: 1;  
    -ms-flex-shrink: 1;  
    flex-shrink: 1;  
    -moz-flex-grow: 0;  
    -webkit-flex-grow: 0;  
    -ms-flex-grow: 0;  
    flex-grow: 0;  
}  
  
.gallery > * {  
    width: 50%;  
}  
  
.gallery > * {  
    padding: 2rem;  
    width: calc(50% + 2rem);  
}  
  
.gallery > :nth-child(-n + 2) {  
    padding-top: 0;  
}  
  
.gallery > :nth-last-child(-n + 2) {  
    padding-bottom: 0;  
}  
  
.gallery > :nth-child(2n + 1) {  
    padding-left: 0;  
}  
  
.gallery > :nth-child(2n) {  
    padding-right: 0;  
}  
  
.gallery > :nth-child(2n + 1),  
.gallery > :nth-child(2n) {  
    width: calc(50% + 0rem);  
}  
  
.gallery .image {  
    background: #FFF;  
    padding: 1rem;  
}
```

```
.gallery img{
    height: 300px;
}

.gallery .image.fit {
    margin: 0;
}

@media screen and (max-width: 980px) {

    .gallery > * {
        width: 100%;
    }

    .gallery > * {
        padding: 1rem;
        width: calc(50% + 1rem);
    }

    .gallery > :nth-child(-n + 2) {
        padding-top: 1rem;
    }

    .gallery > :nth-last-child(-n + 2) {
        padding-bottom: 1rem;
    }

    .gallery > :nth-child(2n + 1) {
        padding-left: 1rem;
    }

    .gallery > :nth-child(2n) {
        padding-right: 1rem;
    }

    .gallery > :nth-child(2n + 1),
    .gallery > :nth-child(2n) {
        padding: 1rem;
        width: calc(100% + 2rem);
    }

    .gallery > * {
        padding: 1rem;
        width: calc(100% + 2rem);
    }

    .gallery > :nth-child(-n + 1) {
        padding-top: 0;
    }

    .gallery > :nth-last-child(-n + 1) {
        padding-bottom: 0;
    }

    .gallery > :nth-child(1n + 1) {
        padding-left: 0;
    }

    .gallery > :nth-child(1n) {
        padding-right: 0;
    }
}
```

```
.gallery > :nth-child(1n + 1),
.gallery > :nth-child(1n) {
    width: calc(100% + 1rem);
}

/*
 * Header */
body.subpage {
    padding-top: 3.25em;
}

@-moz-keyframes reveal-header {
    0% {
        top: -4em;
        opacity: 0;
    }

    100% {
        top: 0;
        opacity: 1;
    }
}

@-webkit-keyframes reveal-header {
    0% {
        top: -4em;
        opacity: 0;
    }

    100% {
        top: 0;
        opacity: 1;
    }
}

@-ms-keyframes reveal-header {
    0% {
        top: -4em;
        opacity: 0;
    }

    100% {
        top: 0;
        opacity: 1;
    }
}

@keyframes reveal-header {
    0% {
        top: -4em;
        opacity: 0;
    }

    100% {
        top: 0;
        opacity: 1;
    }
}

#header {
```

```
background: rgba(0, 0, 0, 0.975);
color: #a6a6a6;
cursor: default;
height: 3.25em;
left: 0;
line-height: 3.25em;
position: fixed;
text-align: right;
top: 0;
width: 100%;
z-index: 10001;
}

#header > .logo {
    display: inline-block;
    height: inherit;
    left: 1.25em;
    line-height: inherit;
    margin: 0;
    padding: 0;
    position: absolute;
    top: 0;
}

#header > .logo a {
    font-size: 1.25em;
    color: #FFF;
    text-decoration: none;
}

#header > .logo a:hover {
    color: white;
}

#header > .logo span {
    font-weight: 400;
    font-size: .8em;
    color: rgba(255, 255, 255, 0.65);
}

#header > a {
    -moz-transition: color 0.2s ease-in-out;
    -webkit-transition: color 0.2s ease-in-out;
    -ms-transition: color 0.2s ease-in-out;
    transition: color 0.2s ease-in-out;
    display: inline-block;
    padding: 0 0.75em;
    color: inherit;
    text-decoration: none;
    color: #FFF;
}

#header > a:hover {
    color: #f2f2f2;
}

#header > a[href="#menu"] {
    text-decoration: none;
    -webkit-tap-highlight-color: transparent;
}

#header > a[href="#menu"]:before {
```

```
        content: "\u2606";
        -moz-osx-font-smoothing: grayscale;
        -webkit-font-smoothing: antialiased;
        font-family: FontAwesome;
        font-style: normal;
        font-weight: normal;
        text-transform: none !important;
    }

#header > a[href="#menu"]:before {
    margin: 0 0.5em 0 0;
}

#header > a + a[href="#menu"]:last-child {
    border-left: solid 1px rgba(0, 0, 0, 0.15);
    padding-left: 1.25em;
    margin-left: 0.5em;
}

#header > a:last-child {
    padding-right: 1.25em;
}

@media screen and (max-width: 736px) {

    #header > a {
        padding: 0 0.5em;
    }

    #header > a + a[href="#menu"]:last-child {
        padding-left: 1em;
        margin-left: 0.25em;
    }

    #header > a:last-child {
        padding-right: 1em;
    }

}

#header.reveal {
    -moz-animation: reveal-header 0.5s ease;
    -webkit-animation: reveal-header 0.5s ease;
    -ms-animation: reveal-header 0.5s ease;
    animation: reveal-header 0.5s ease;
}

#header.alt {
    -moz-animation: none;
    -webkit-animation: none;
    -ms-animation: none;
    animation: none;
    background-color: transparent;
    box-shadow: none;
    overflow: hidden;
    position: absolute;
    top: 1.5em;
}

#header.alt h1 {
    left: 2.5em;
}
```

```
#header.alt nav {
    right: 2.5em;
}

@media screen and (max-width: 980px) {

    body.subpage {
        padding-top: 44px;
    }

    #header {
        height: 44px;
        line-height: 44px;
    }

    #header > h1 {
        left: 1em;
    }

    #header > h1 a {
        font-size: 1em;
    }
}

@media screen and (max-width: 480px) {

    #header {
        min-width: 320px;
    }
}

/* Menu */

#menu {
    -moz-transform: translateX(20rem);
    -webkit-transform: translateX(20rem);
    -ms-transform: translateX(20rem);
    transform: translateX(20rem);
    -moz-transition: -moz-transform 0.5s ease, box-shadow 0.5s ease,
visibility 0.5s;
    -webkit-transition: -webkit-transform 0.5s ease, box-shadow 0.5s
ease, visibility 0.5s;
    -ms-transition: -ms-transform 0.5s ease, box-shadow 0.5s ease,
visibility 0.5s;
    transition: transform 0.5s ease, box-shadow 0.5s ease, visibility
0.5s;
    -webkit-overflow-scrolling: touch;
    background: rgba(0, 0, 0, 0.95);
    box-shadow: none;
    color: #000;
    height: 100%;
    max-width: 80%;
    overflow-y: auto;
    padding: 3rem 2rem;
    position: fixed;
    right: 0;
    top: 0;
    visibility: hidden;
    width: 20rem;
```

```
        z-index: 10002;
    }

    #menu > ul {
        margin: 0 0 1rem 0;
    }

    #menu > ul.links {
        list-style: none;
        padding: 0;
    }

    #menu > ul.links > li {
        padding: 0;
    }

    #menu > ul.links > li > a:not(.button) {
        border: 0;
        border-top: solid 1px rgba(255, 255,
255, 0.125);
        color: rgba(255, 255, 255, 0.5);
        display: block;
        line-height: 3.5rem;
        text-decoration: none;
        text-transform: uppercase;
    }

    #menu > ul.links > li >
a:not(.button):hover {
        color: #FFF;
    }

    #menu > ul.links > li > .button {
        display: block;
        margin: 0.5rem 0 0 0;
    }

    #menu > ul.links > li:first-child >
a:not(.button) {
        border-top: 0 !important;
    }

    #menu .close {
        text-decoration: none;
        -moz-transition: color 0.2s ease-in-out;
        -webkit-transition: color 0.2s ease-in-out;
        -ms-transition: color 0.2s ease-in-out;
        transition: color 0.2s ease-in-out;
        -webkit-tap-highlight-color: transparent;
        border: 0;
        color: #999999;
        cursor: pointer;
        display: block;
        height: 3.25rem;
        line-height: 3.25rem;
        padding-right: 1.25rem;
        position: absolute;
        right: 0;
        text-align: right;
        top: 0;
        vertical-align: middle;
        width: 7rem;
```

```
}

#menu .close:before {
    -moz-osx-font-smoothing: grayscale;
    -webkit-font-smoothing: antialiased;
    font-family: FontAwesome;
    font-style: normal;
    font-weight: normal;
    text-transform: none !important;
}

#menu .close:before {
    content: '\f00d';
    font-size: 1.25rem;
}

#menu .close:hover {
    color: #000;
}

@media screen and (max-width: 736px) {

    #menu .close {
        height: 4rem;
        line-height: 4rem;
    }

}

#menu.visible {
    -moz-transform: translateX(0);
    -webkit-transform: translateX(0);
    -ms-transform: translateX(0);
    transform: translateX(0);
    box-shadow: 0 0 1.5rem 0 rgba(0, 0, 0, 0.2);
    visibility: visible;
}

@media screen and (max-width: 736px) {

    #menu {
        padding: 2.5rem 1.75rem;
    }

}

/* Banner */

.banner {
    background-color: #8a4680;
    color: #e2d1df;
    padding: 8em 0;
    position: relative;
}

.banner input, .banner select, .banner textarea {
    color: #ffffff;
}

.banner a {
    color: #ffffff;
}
```

```
.banner strong, .banner b {  
    color: #ffffff;  
}  
  
.banner h1, .banner h2, .banner h3, .banner h4, .banner h5,  
.banner h6 {  
    color: #ffffff;  
}  
  
.banner blockquote {  
    border-left-color: rgba(255, 255, 255, 0.25);  
}  
  
.banner code {  
    background: rgba(255, 255, 255, 0.075);  
    border-color: rgba(255, 255, 255, 0.25);  
}  
  
.banner hr {  
    border-bottom-color: rgba(255, 255, 255, 0.25);  
}  
  
.banner.full {  
    padding: 0;  
    min-height: 100vh;  
    height: 100vh !important;  
}  
  
.banner.half {  
    padding: 0;  
    min-height: 50vh;  
    height: 50vh !important;  
}  
  
.banner:after {  
    -moz-pointer-events: none;  
    -webkit-pointer-events: none;  
    -ms-pointer-events: none;  
    pointer-events: none;  
    -moz-transition: opacity 1.5s ease-in-out, visibility  
1.5s;  
    -webkit-transition: opacity 1.5s ease-in-out, visibility  
1.5s;  
    -ms-transition: opacity 1.5s ease-in-out, visibility 1.5s;  
    transition: opacity 1.5s ease-in-out, visibility 1.5s;  
    background: #000000;  
    content: '';  
    display: block;  
    height: 100%;  
    left: 0;  
    opacity: 0;  
    position: absolute;  
    top: 0;  
    visibility: hidden;  
    width: 100%;  
    z-index: 2;  
}  
  
.banner .indicators {  
    bottom: 1.5em;  
    left: 0;
```

```
list-style: none;
margin: 0;
padding: 0;
position: absolute;
text-align: center;
width: 100%;
z-index: 2;
}

.banner .indicators li {
    cursor: pointer;
    display: inline-block;
    height: 2em;
    overflow: hidden;
    padding: 0;
    position: relative;
    text-indent: 2em;
    width: 2em;
}

.banner .indicators li:before {
    background: rgba(255, 255, 255, 0.35);
    border-radius: 100%;
    content: '';
    display: inline-block;
    height: 0.8em;
    left: 50%;
    margin: -0.4em 0 0 -0.4em;
    position: absolute;
    text-indent: 0;
    top: 50%;
    width: 0.8em;
}

.banner .indicators li.visible:before {
    background: #fff;
}

.banner > article {
    -moz-transition: opacity 1.5s ease, visibility 1.5s;
    -webkit-transition: opacity 1.5s ease, visibility 1.5s;
    -ms-transition: opacity 1.5s ease, visibility 1.5s;
    transition: opacity 1.5s ease, visibility 1.5s;
    background-attachment: fixed;
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;
    height: 100%;
    left: 0;
    opacity: 0;
    position: absolute;
    text-align: center;
    top: 0;
    visibility: hidden;
    width: 100%;
    z-index: 0;
}

.banner > article:before {
    content: '';
    display: inline-block;
    height: 100%;
```

```
        vertical-align: middle;
    }

.banner > article:after {
    content: '';
    display: block;
    width: 100%;
    height: 100%;
    position: absolute;
    top: 0;
    left: 0;
    background: #000;
    opacity: 0.35;
}

.banner > article .inner {
    position: relative;
    display: inline-block;
    vertical-align: middle;
    z-index: 1;
}

.banner > article .inner > :last-child {
    margin-bottom: 0;
}

.banner > article h2 {
    font-size: 7rem;
    margin-bottom: 0;
    color: #FFF;
    font-weight: 300;
}

.banner > article h2:after {
    display: none;
}

.banner > article p {
    color: rgba(255, 255, 255, 0.65);
    text-transform: uppercase;
    font-size: 1rem;
    font-weight: 300;
    margin: 0;
    padding-bottom: 1.75rem;
    letter-spacing: .25rem;
}

.banner > article p:after {
    content: '';
    position: absolute;
    margin: auto;
    right: 0;
    bottom: 0;
    left: 0;
    width: 50%;
    height: 1px;
    background-color: rgba(255, 255, 255, 0.65);
}

.banner > article a {
    color: #FFF;
    text-decoration: none;
```

```
}

.banner > article img {
    display: none;
}

.banner > article.visible {
    opacity: 1;
    visibility: visible;
}

.banner > article.top {
    z-index: 1;
}

.banner > article.instant {
    -moz-transition: none !important;
    -webkit-transition: none !important;
    -ms-transition: none !important;
    transition: none !important;
}

body.is-loading .banner:after {
    opacity: 1.0;
    visibility: visible;
}

@media screen and (max-width: 1280px) {

    .banner.full {
        padding: 0;
        min-height: 75vh;
        height: 75vh !important;
    }
}

@media screen and (max-width: 980px) {

    .banner.full {
        padding: 0;
        min-height: 50vh;
        height: 50vh !important;
    }

    .banner > article {
        background-attachment: scroll;
    }
}

@media screen and (max-width: 736px) {

    .banner > article .inner {
        width: 90%;
    }

    .banner > article p {
        margin-bottom: 1rem;
    }

    .banner > article h2 {
```

```
        font-size: 4em;
    }

}

body.is-mobile .banner > article {
    background-attachment: scroll;
}

/* Main */

#main {
    padding: 4rem 0 2rem 0 ;
}

@media screen and (max-width: 736px) {

    #main {
        padding: 3rem 0 1rem 0 ;
    }

}

/* Footer */

#footer {
    padding: 4rem 0 2rem 0 ;
    background: #000;
    text-align: center;
    position: relative;
    bottom: 0;
    width: 100%;
}

#footer a {
    color: rgba(255, 255, 255, 0.5);
}

#footer .copyright {
    color: #bbb;
    font-size: 0.9rem;
    margin: 0 0 2rem 0;
    padding: 0;
    text-align: center;
}

@media screen and (max-width: 736px) {

    #footer {
        padding: 3rem 0 1rem 0 ;
    }

}
```

(שירות רשת) ShoeCalculatorService.asmx

מיקום: ProjectWebServiceSOAP/ShoeCalculatorService.asmx

מיקום קוד מאחור: ProjectWebServiceSOAP/App_Code/ShoeCalculatorService.cs

קוד asmx

```
<%@ WebService Language="C#" CodeBehind("~/App_Code/ShoeCalculatorService.cs"
Class="ShoeCalculatorService" %>
```

קוד C#

```
using System;
using System.Data;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Net.Mail;
using System.Net;
using MailKit;
using MailKit.Net.Smtp;
using MimeKit;

/// <summary>
/// Summary description for ShoeCalculatorService
/// </summary>
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
// To allow this Web Service to be called from script, using ASP.NET AJAX,
uncomment the following line.
// [System.Web.Script.Services.ScriptService]
public class ShoeCalculatorService : System.Web.Services.WebService
{
    // פעולה בונה בירית מודול
    public ShoeCalculatorService()
    {

        //Uncomment the following line if using designed components
        //InitializeComponent();
    }

    מקבל נתונים על משתמש והעדפות לנעל, ולפיכך מחשב גודל נעל טיפוס עבור המשתמש
    [WebMethod	Description = "Shoe size calculator")]
    public int ShoeCalc(int normalSize, string company, string shoeFit, string
boulderGrade, string wallGrade)
    {
        int minusSize = (BoulderGradeToMinusSize(boulderGrade) +
WallGradeToMinusSize(wallGrade)) / 2;

        switch (shoeFit)
        {
            case "Loose":
                minusSize--;
                break;
            case "Normal":
```

```

        break;
    case "Tight":
        minusSize++;
        break;
    default:
        break;
    }

    switch (company)
    {
        case "La Sportiva":
            minusSize += 2;
            break;
        case "Scarpa":
        case "Unparallel":
        case "Madrock":
            minusSize++;
            break;
        case "Adidas":
            break;
        default:
            break;
    }

    return normalSize - minusSize;
}

// ממיר מרמת טיפוס לכמות הורדה בגעלה
private int BoulderGradeToMinusSize(string boulderGrade)
{
    int gradeNum = int.Parse(boulderGrade.Substring(1));
    return gradeNum / 6;
}

// ממיר מרמת טיפוס לכמות הורדה בגעלה
private int WallGradeToMinusSize(string wallGrade)
{
    int gradeNum = int.Parse(wallGrade[0].ToString()) * 2;
    return gradeNum / 6;
}

מקבל רשימה של שמות מנהלים ומוצרים שימושה לקנות ומידע אותו שאין מספיק מהמוצרים //  

באספקה וצריך לבדוק אותה
[WebMethod(Description = "Item out of stock notifier")]
public void SendStockNotification(string username, List<string>  

adminEmailList, List<string> adminNameList, List<string> outOfStockItemList,  

List<int> outOfStockSizeList)
{
    string outOfStockListStr = "";
    for(int i = 0; i < outOfStockItemList.Count; i++)
    {
        outOfStockListStr += string.Format("{0} - Size {1}\n",
outOfStockItemList[i], outOfStockSizeList[i]);
    }
    for(int i = 0; i < adminEmailList.Count; i++)
    {
        string content = string.Format("Hi, {0}. \nThe user {1} is trying to  

order the following products:\n{2}\nBut it seems that there aren't enough in  

stock, so he can't purchase them!\nPlease check up on this and restock if  

needed, otherwise we might lose some sales!\nBest regards,\nThe Climbing Web  

Team", adminNameList[i], username, outOfStockListStr);
        SendEmail(content, "Item Out Of Stock", adminEmailList[i]);
    }
}

```

```

        }

    //}

    //שלוח אימייל עם התוכן, נושא, וنمען שהתקבלו במייל
    [WebMethod(Description = "Email sender")]
    public string SendEmail(string content, string subject, string target)
    {
        return SendEmail(content, subject, target, new string[0]);
    }

    //שלוח אימייל קבלה ללקוח עם פרטי ההזמנה/
    [WebMethod(Description = "Receipt Sender")]
    public string SendReceiptEmail(DataSet infoDs, string target, string
username)
    {
        DataTable orderTbl = infoDs.Tables["BasicOrderInfo"].Copy();
        DataTable orderProductTbl = infoDs.Tables["ProductDetails"].Copy();
        DataTable productTbl = infoDs.Tables["ProductTable"].Copy();

        DataColumn productNameColumn = orderProductTbl.Columns.Add("name",
typeof(string));
        DataColumn buyDateColumn = orderTbl.Columns.Add("date",
typeof(string));

        DateTime buyDate =
DateTime.Parse(orderTbl.Rows[0]["datetime"].ToString());

        orderTbl.Rows[0]["date"] = buyDate.ToString("dd/M/yyyy");

        foreach (DataRow dataRow in orderProductTbl.Rows)
        {
            int drProductId = int.Parse(dataRow["productId"].ToString());

            dataRow["name"] = productTbl.Rows[drProductId -
1]["name"].ToString();
        }

        orderTbl.Columns.Add("totalPayment");
        orderTbl.Rows[0]["totalPayment"] = orderTbl.Rows[0]["paid"].ToString();

        orderTbl.Columns.Remove("datetime");
        orderTbl.Columns.Remove("orderId");
        orderTbl.Columns.Remove("paid");
        orderProductTbl.Columns.Remove("orderId");
        orderProductTbl.Columns.Remove("productId");

        DataSet ds = new DataSet();

        ds.Tables.Add(orderTbl);
        ds.Tables.Add(orderProductTbl);

        orderTbl.TableName = "BasicOrderInfo";
        orderProductTbl.TableName = "ProductDetails";
        ds.DataSetName = "OrderReceipt";

        ds.WriteXml(Server.MapPath(@"EmailXML.xml"));

        string[] @pathArr = { @"EmailXML.xml" };

        string content = "Hi, " + username + "! Thanks for ordering from our
website! We attached your receipt in XML format to this email so you can
download it and save it.\n\nBest regards,\nThe Climbing Web Team";
    }
}

```

```

        return SendEmail(content, "Your Receipt", target, @pathArr);
    }

    //שלוח אימייל אלין מצורף דאטאסט בפורט XML
    [WebMethod(Description ="Send dataset through email")]
    public string SendEmailDS(string subject, string username, string target,
    string content, DataSet ds)
    {
        ds.WriteXml(Server.MapPath(@"EmailXML.xml"));

        string[] @pathArr = { @"EmailXML.xml" };

        return SendEmail(content, subject, target, @pathArr);
    }

    //שלוח אימייל יחד עם הקבצים שהתקבלו שמותיהם בקובץ XML
    public string SendEmail(string content, string subject, string target,
    string[] @attachments)
    {
        string status;

        MimeMessage message = new MimeMessage();

        message.From.Add(new MailboxAddress("Climbing Web",
        "yonatansmailbot@gmail.com"));

        message.To.Add(MailboxAddress.Parse(target));

        message.Subject = subject;

        var builder = new BodyBuilder();

        builder.TextBody = content;

        foreach(string @path in @attachments)
        {
            builder.Attachments.Add(Server.MapPath(@path));
        }

        message.Body = builder.ToMessageBody();

        MailKit.Net.Smtp.SmtpClient smtpClient = new
        MailKit.Net.Smtp.SmtpClient();

        try
        {
            smtpClient.Connect("smtp.gmail.com", 465, true);
            smtpClient.Authenticate("yonatansmailbot@gmail.com", "mail4proj");
            smtpClient.Send(message);
            status = "Success";
        }
        catch (Exception exp)
        {
            Console.WriteLine(exp.Message);
            status = "Fail";
        }
        finally
        {
            smtpClient.Disconnect(true);
            smtpClient.Dispose();
        }
    }
}

```

```
    }  
    return status;  
}  
}
```

EmailXML.xml

מיקוד: ProjectWebServiceSOAP/EmailXML.xml

קוד XML

```
<?xml version="1.0" standalone="yes"?>
<OrderReceipt>
  <BasicOrderInfo>
    <username>valery_admin</username>
    <date>17/4/2022</date>
    <totalPayment>14893.7</totalPayment>
  </BasicOrderInfo>
  <ProductDetails>
    <size>37</size>
    <amount>9</amount>
    <name>Tomoa Narasaki Pro</name>
  </ProductDetails>
  <ProductDetails>
    <size>40</size>
    <amount>9</amount>
    <name>Solutions Comp Men's</name>
  </ProductDetails>
  <ProductDetails>
    <size>41</size>
    <amount>12</amount>
    <name>Solutions Comp Men's</name>
  </ProductDetails>
  <ProductDetails>
    <size>46</size>
    <amount>2</amount>
    <name>Shark 2.0</name>
  </ProductDetails>
</OrderReceipt>
```

(שירות רשת) Web.config

מיקום: ProjectWebServiceSOAP/Web.config

קוד XML

```
<?xml version="1.0"?>
<!--
  For more information on how to configure your ASP.NET application, please
  visit
  https://go.microsoft.com/fwlink/?LinkId=169433
  -->
<configuration>

  <system.web>
    <compilation debug="true" targetFramework="4.6.1">
      <assemblies>
        <add assembly="System.Net.Http, Version=4.0.0.0, Culture=neutral,
          PublicKeyToken=B03F5F7F11D50A3A"/>
        <add assembly="System.Security, Version=4.0.0.0, Culture=neutral,
          PublicKeyToken=B03F5F7F11D50A3A"/>
      </assemblies>
    </compilation>
    <httpRuntime targetFramework="4.6.1"/>
  </system.web>
  <system.codedom>
    <compilers>
      <compiler language="c#;cs;csharp" extension=".cs"
        type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider,
        Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.0.0,
        Culture=neutral, PublicKeyToken=31bf3856ad364e35" warningLevel="4"
        compilerOptions="/langversion:default /nowarn:1659;1699;1701"/>
      <compiler language="vb;vbs;visualbasic;vbscript" extension=".vb"
        type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider,
        Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.0.0,
        Culture=neutral, PublicKeyToken=31bf3856ad364e35" warningLevel="4"
        compilerOptions="/langversion:default /nowarn:41008
        /define:_MYTYPE="Web" /optionInfer+/">
    </compilers>
  </system.codedom>
</configuration>
```

(שירות רשת) packages.config

מיקום: ProjectWebServiceSOAP/packages.config

קוד XML

```
<?xml version="1.0" encoding="utf-8"?>
<packages>
    <package id="MailKit" version="3.1.1" targetFramework="net461" />
    <package id="Microsoft.CodeDom.Providers.DotNetCompilerPlatform"
version="2.0.0" targetFramework="net461" />
    <package id="MimeKit" version="3.1.1" targetFramework="net461" />
    <package id="Portable.BouncyCastle" version="1.9.0" targetFramework="net461"
/>
    <package id="System.Buffers" version="4.5.1" targetFramework="net461" />
</packages>
```

נספחים

הוראות להרצת האתר

1. יש לפתח את הפרויקט ולהיכנס לקוד של הדף HomePage.aspx.
2. לחוץ על הcptions Ctrl + F5
3. על מנת להתחבר, ניתן להזין פרטי משתמש עבור המשתמשים הבאים (או להירשם).

פרטי משתמשים

משתמש רגיל

שם משתמש: test_basic

סיסמה: Test123

אמאן

שם משתמש: test_coach

סיסמה: Test123

מנהל אספקה

שם משתמש: test_supply_manager

סיסמה: Test123

מנהל כללי

שם משתמש: test_general_manager

סיסמה: Test123