

# Fakebook

**עבודת גמר  
בתכנון ותכנות מערכות  
במסלול שירותי רשת אינטרנט  
שנת הלימודים**

שם התלמיד : עומרי קיריב

בית ספר : מקיף יב' המעיין

ת.ז : 319129326

שם המורה : חנה לוי

### הקדשה/תודה:

ברצוני להודות למורתי חנה לוי אשר ליוותה אותי במהלך כל הפרוייקט, תמיד כאשר נתקלתי בשאלה יכולתי לפנות אליה והיא עזרה לתקן את הבעיה. תודה לך חנה על העזרה, ההדרכה, פיתוח החשיבה היצירתית והשרשת בסיס יציב ויעיל לתכנות.

## תוכן עניינים

2	הקדשה/תודה:
3	תוכן עניינים
4	מבוא
4	מטרות האתר
4	קהל יעד
4	תיאור האתר
5	מפת האתר
6	מדריך האפליקציה למשתמש
14	מבנה בסיס הנתונים
14	טבלת Users – משתמשים
14	טבלת Messages – הודעות
15	טבלת Likes – לייקים
15	טבלת Friendships – חברויות קיימות
15	טבלת FriendRequests – בקשות חברות
15	טבלת Ads – פרסומות
16	תמונת/תמונות קשרי הגומלין מתוך מסד הנתונים ( DSD )
18	Users
20	Friendships
21	Messages
21	Likes
22	FriendRequest
23	Posts
26	CreditCard
26	DataService
27	שירותי רשת שהאפליקציה מספקת
27	מבנה בסיס הנתונים של שירותי הרשת
27	מבנה התיקיות
27	CreditCards
28	ReadMe

## מבוא

### **מטרות האתר**

מטרת האתר היא להוות פלטפורמה חברתית בה אנשים יכולים להוסיף אחרים כחברים ולתקשר עם אנשים.

### **קהל יעד**

האתר נועד לגילאי 16+, קהל היעד הוא כלל האנשים הגולשים באינטרנט המעוניינים להתחבר עם אנשים אחרים.

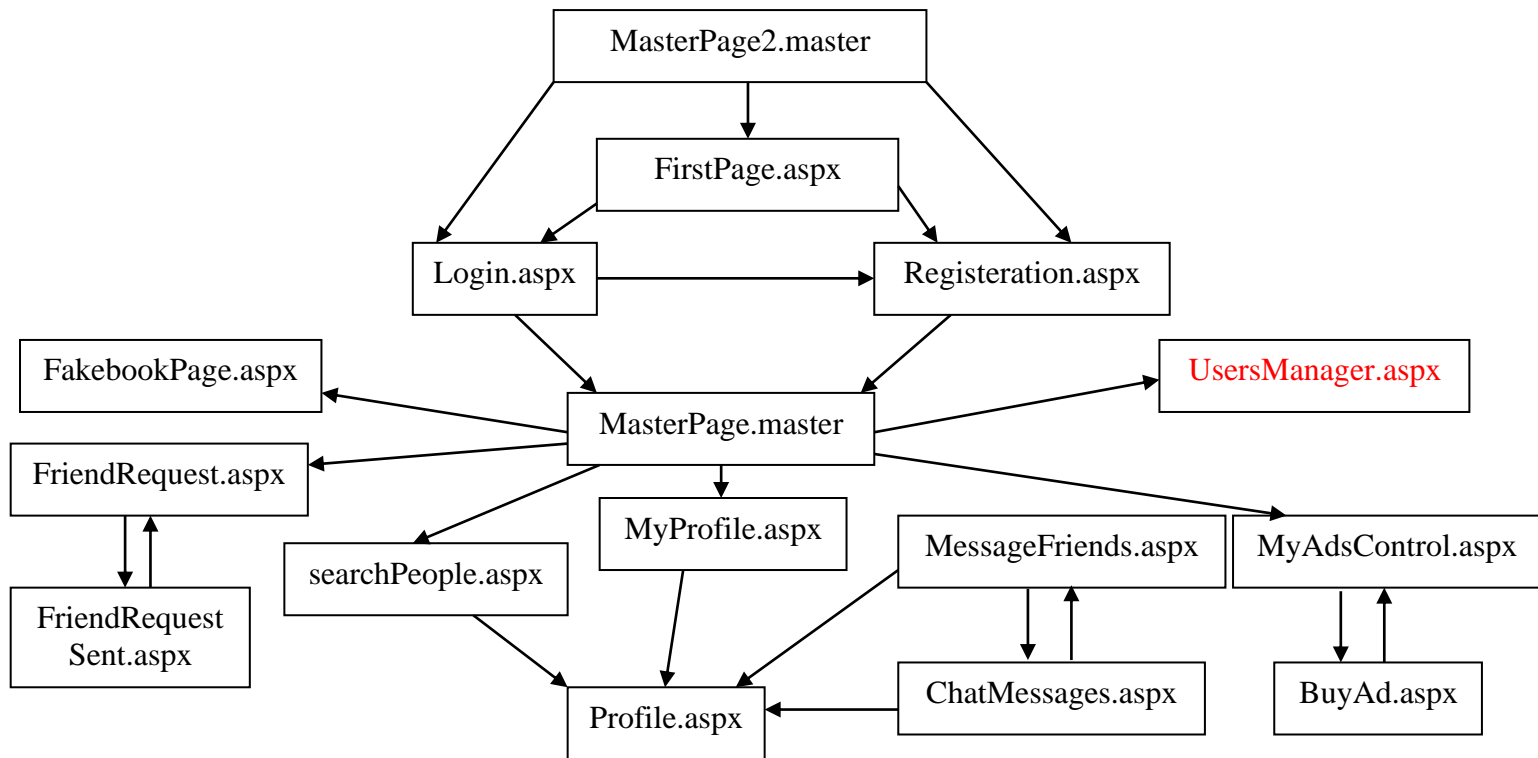
### **תיאור האתר**

האתר הוא העתק של פייסבוק, רשת חברתית בה כל אדם יכול להירשם, להתחבר ולהשתתף. באתר אתה וחברים שלך יכולים לפרסם פוסטים בקיר שלך ואנשים אחרים יכולים להגיב עליהם. ניתן לעשות לייק לפוסט ולהראות שאהבת אותו. באתר אתה יכול לחפש אנשים אחרים שרשומים באתר ולהזמין אותם להיות חברים שלך. יש באתר צ'אט (לא לייב) שניתן בו לכתוב הודעות לחבר ולראות הודעות ישנות. באתר ניתן "לקנות" פרסומות, כאשר בהזמנה של הפרסומת ניתנת האפשרות לבחור לאיזה אורך של זמן הפרסומת תפורסם (בחודשים), איזו חבילה של פרסומת (נמוכה, רגילה, גבוהה). למשתמש ישנה גישה לדף בו ניתן לערוך את הפרטים של הפרסומת. בדף הראשון (FakebookPage.aspx) ישנו מקום קבוע בו שתי פרסומות (לכל היותר) מוצגות בו זמנית, והפרסומות שם מתחלפות כל 10 שניות. בלחיצה על הפרסומת מדד כניסות לפרסומת גדל באחד ובעמוד של בקרת הפרסומת המשתמש יכול לגשת ולראות את מספר הכניסות בפרסומת.

## מפת האתר

דוגמא:

להלן מוצג תרשים הזרימה האפשרי באתר:



## מדריך האפליקציה למשתמש

משתמש מזדמן:

### FirstPage.aspx

זהו העמוד הראשון של הפרויקט, בדף זה יש מידע כללי בנוגע לאתר. מדף זה ניתן להגיע לדפי ההרשמה וההתחברות. אם המשתמש הגיע לדפים שתוכנם מיועד רק לגולשים מחוברים המשתמש יופנה לדף זה.

### Registration.aspx

בדף זה המשתמש נרשם לאתר. עליו למלא את הפרטים שלו: שם פרטי, שם משפחה, אימייל, מין, סיסמא ואימותה, תאריך לידה ומין. לאחר שהמשתמש נרשם הוא מועבר לעמוד הראשון של המשתמש הרשום.

### Login.aspx

המשתמש מתחבר, מזין אימייל וסיסמה אם הנתונים תקינים המשתמש מועבר לדף הראשון של המשתמש הרשום. אם המשתמש לא רשום ניתנת לגולש האפשרות לעבור לדף ההרשמה.

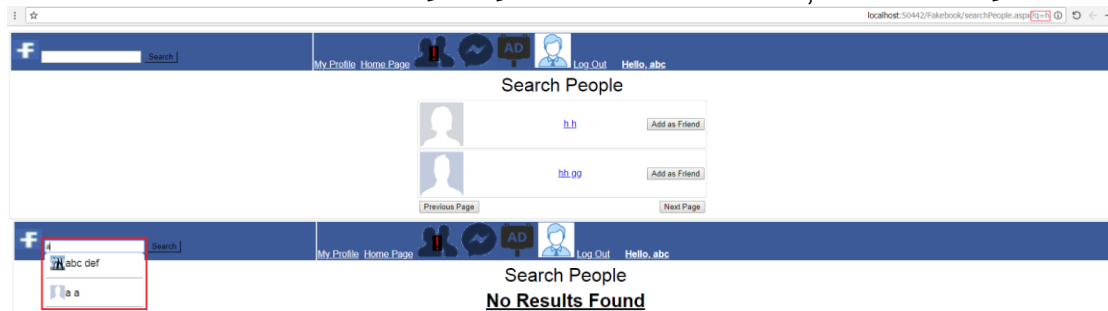
### FakebookPage.aspx

הדף שאליו מועברים הגולשים שנרשמו/התחברו. בעמוד זה יש "פרסומות" בצד, הפרסומות נלקחות מהדטה בייס שבו הוכנסו ערכים על ידי משתמשים ששילמו על הפרסומות. כל 10 שניות נבחרות שתי פרסומות (לכל היותר) באופן אקראי לפי, לכל פרסומת בטבלה יש רמת חבילה אשר מדורגת מ 1 עד 3 כאשר המספר מייצג את מספר ההופעות של הפרסומת ברשימה שממנה נבחר באקראי הפרסומת. כאשר לוחצים על התמונה של הפרסומת מועברים לקישור שמפרסם הפרסומת בחר ומדד הכניסות של הפרסומת גדל באחד.



### searchPeople.aspx

בעמוד זה המשתמש מופיעות תוצאות החיפוש של המשתמש בשורת חיפוש האנשים. כאשר ישנם תוצאות מופיע בפקד הדטה ליסט התוצאות המתאימות כאשר יש עמוד של 5 אנשים בעמוד לכל היותר, אם לא היה תוצאות יופיע הודעה שלא נמצאו אנשים.



פרט לחיפוש הידני כאשר מזינים שם ולוחצים על כפתור ה"search" התיבת טקסט מנסה להשלים למשתמש לפי מה שהוזן, וכאשר לוחצים על השורה המתאימה המשתמש מועבר לפרופיל של האדם שהוא מצא – המגבלה של ההשלמה האוטומטית היא 5 תוצאות.

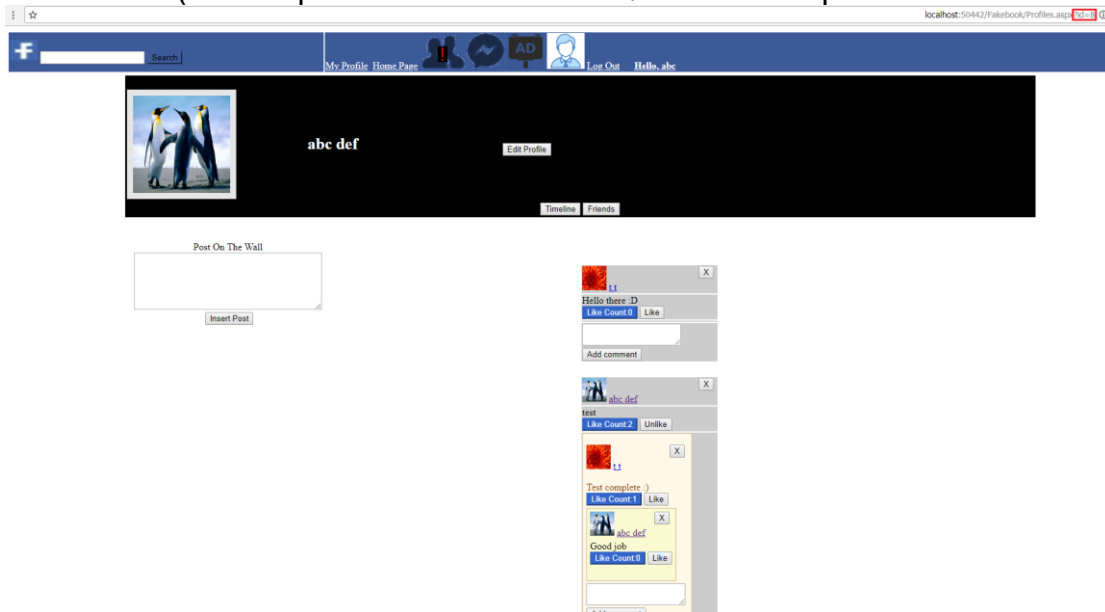
בחיפוש מופיעים כפתורים המתאימים למצב בין המשתמש הגולש לבין תוצאת החיפוש: אם תוצאת החיפוש הוא המשתמש הגולש עצמו יופיע כפתור "גלוש לפרופיל שלי" אם תוצאת החיפוש הוא חבר/ישנה בקשת חברות הכפתור שיופיע הוא "מחק חבר" שימחוק את החברות ביניהם, אם תוצאת החיפוש לא חבר ואין בקשת חברות בין השניים יופיע כפתור המאפשר לשלוח בקשת חברות למשתמש האחר.



## Profiles.aspx

עמוד הפרופילים – בעמוד זה מוצג הפרטים של משתמש ספציפי, המידע המוצג הוא בהתאם לאידי שמופיע בשורת הלינק. מוצגת תמונת פרופיל, שם פרטי ושם משפחה. אם הפרופיל המוצג הוא חבר של המשתמש הגולש או שהפרופיל הוא של המשתמש הגולש, ניתן לפרסם פוסט על הקיר. כאשר לוחצים על כפתור הטיימליין/הכניסה הבסיסית לפרופיל הדף המוצג הוא כמופיע בתמונה, כאשר ישנה אפשרות להכניס פוסט על הקיר ויש דטה ליסט שבו מופיעים כל הפוסטים שמופיעים על הקיר ולכל פוסט יופיעו התגובות שנכתבו על הפוסט וגם תיבת טקסט המאפשרת להגיב על הפוסט/תגובה.

אם המשתמש גולש בפרופיל שלו עצמו הוא יכול למחוק את כל הפוסטים (יש לו כפתור X שמוחק את הפוסט מהקיר) אחרת כפתור מחיקת הפוסט מופיע רק למפרסם התגובה(אם אתה מנהל כפתור מחיקת הפוסט מופיע תמיד – המנהל יכול למחוק כל פוסט)




על כל תגובה המשתמש יכול לעשות לייק, אם המשתמש כבר עשה לייק הכפתור ישתנה למחיקת הלייק. לכל תגובה יש מדד לייקים שהוא כפתור בו כתוב כמה לייקים יש, כאשר לוחצים על הכפתור נפתח פאנל שבו מציגים את המשתמשים ביצעו לייק לפוסט/תגובה, כאשר לוחצים על האיקס חוזרים לפאנל שמציג את הפוסטים.



בלחיצה על הכפתור "friends" יופיע דטה ליסט שבו מופיעים כל החברים של המשתמש שזהו הפרופיל שלו, יש עימוד של עד 5 חברים בעמוד. אם אין חברים ייכתב שאין חברים.



## Friend List

 tt

Friends Since 25/02/2018 14:28:47

[Previous Page](#) [Next Page](#)

## Friend List

No Friends Yet

אם הפרופיל הוא של המשתמש שמחובר אז יופיע כפתור "עריכה" שבלחיצה עליו יפתח למשתמש האפשרות לשנות את תמונת הפרופיל שלו, את השם פרטי את השם משפחה ואת הסיסמה.

Update your profile image

Update your info

	First Name	Last Name	Password	Email	Gender	Birthday
<a href="#">Update</a> <a href="#">Cancel</a>	abc	def	b	b	Male	23/04/1984

בעמוד יש כפתורים בהתאם למצב של המשתמש הגולש לבין המשתמש שלו שייך הפרופיל

This person is in your friendlist

[Add friend](#) [Edit Profile](#) [Delete Friend](#)

[Confirm Friendship request](#) [Chat With Me](#)

אם הפרופיל הוא לא חבר של המשתמש הגולש יופיע הכפתור "הוסף חבר" שישלח בקשת חברות.


אם הפרופיל הוא חבר של המשתמש הגולש יופיע הכפתור "שוחח עמי"(בלחיצה עליו הגולש יועבר לצאט עם החבר) "מחק חבר" ויופיע הלייבל בו כתוב שהאדם הזה כבר ברשימת חברים שלך.

אם הפרופיל הזה שלח למשתמש הגולש בקשת חברות יופיע הכפתור "אשר בקשת חברות".

## MessageFriends.aspx

Search Friend To Converse With

tt

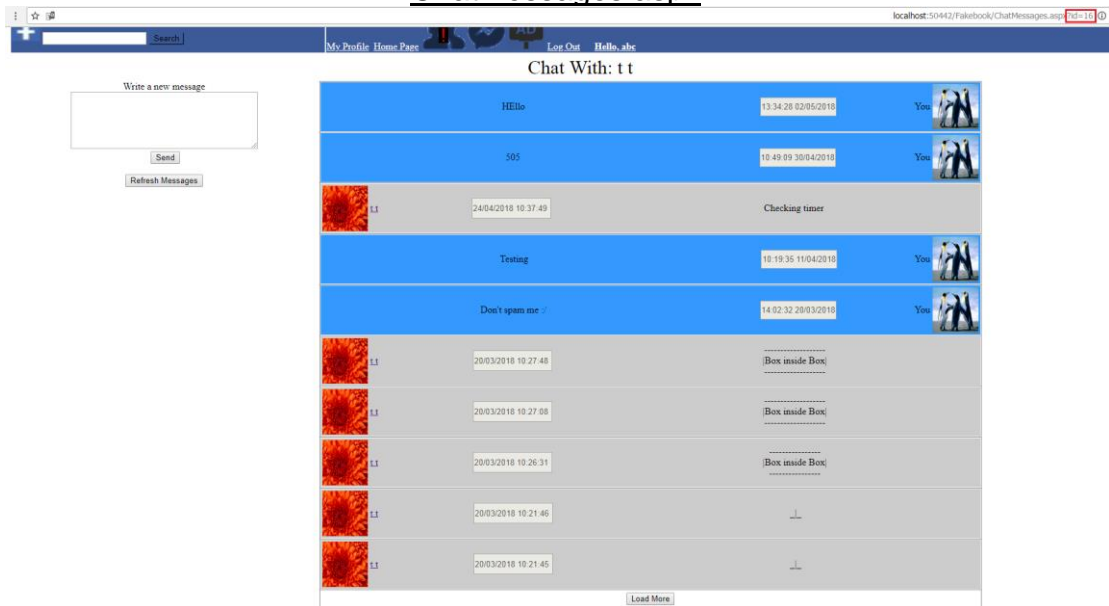
 tt

Private Message One Of Your Friends

	Friend	Friends Since	Last Message's date	Unread Messages?
<a href="#">Enter A Conversation</a>	Amir Kalinsky	19/05/2018 01:00:22	19/05/2018 01:02:40	<input checked="" type="checkbox"/>
<a href="#">Enter A Conversation</a>	tt	25/02/2018 14:28:47	19/05/2018 00:45:01	<input checked="" type="checkbox"/>
<a href="#">Enter A Conversation</a>	Nicole Safony	18/05/2018 22:32:37	You have yet to message this friend	<input type="checkbox"/>

בעמוד הזה ניתן לפנות לצאט עם החברים שלך, בצד ימין יש גריד ויו שמציג את כל החברים, ובצד שמאל ניתן להזין חלק מהשם ויושלם לך שם החברים שמתאימים לך ויהיה ניתן ללחוץ ואז מועברים לשיחה עם החבר. הטבלה מצד ימין ממיינת בסדר יורד מההודעה האחרונה שנשלחה בשיחה, לכן שיחות פעילות יותר יופיעו למעלה. נוסף לכך ישנה עמודה המציגה האם ישנן הודעות חדשות בשיחה הנוכחית(הודעות חדשות אלו הודעות שלא נקראו, הודעה מסומנת כנקראת ברגע שהגולש נכנס לשיחה). אם בין הגולש לחבר לא הייתה שיחה עדיין תכתב הודעה שאומרת כי עדיין לא נכתבו הודעות בין השניים.

## ChatMessages.aspx



כאשר הגולש לוחץ על כפתור שיחה עם חבר (דרך הפרופיל של החבר או דרך העמוד של צאט עם חברים), כאשר נכנסים לעמוד הדף בודק האם קיים חברים בין המשתמש הגולש לבין המשתמש שהאידי שלו מופיע בקישור. אם האידי לא חבר של המשתמש/ אותו אידי של המשתמש הגולש/ לא קיים אידי בכתובת קישור המשתמש יופנה לדף הבית של הפייקבוק.

הצ'אט הוא לא לייב, ניתן ללחוץ על "רענן הודעות", כאשר שולחים הודעה חדשה מתעדכן הדטה ליסט שמכיל את כל ההודעות. העימוד הוא עד 10 הודעות בעמוד, ניתן ללחוץ על כפתור "טען עוד" שיטען 10 הודעות ישנות יותר(לכל היותר).

## FriendsRequest.aspx



בעמוד הנ"ל מוצגים הבקשות חברות שהתקבלו למשתמש, הגולש יכול לאשר או לדחות את בקשת החברות. אם ישנה בקשת חברות למשתמש בתמונה של החברות בשורת של המאסטר פייג יופיע סימן קריאה. בעמוד ניתן להגיע לעמוד בקשות חברות שנלחו ולראות אילו בקשות חברות שלח המשתמש.

## FriendRequestSent.aspx

בעמוד הנ"ל מוצגים האנשים שלהם שלחת בקשת חברות, ניתנת האפשרות למחוק את בקשת החברות.



## MyAdsControl.aspx

בעמוד הזה יש למשתמש שליטה על הפרסומות שלו, מוצגים בטבלה המידע של הפרסומות.

My Published Ads

[Go to advertise an ad](#)

	Advertising Until	Advertiser Name	Website URL	Ad Image	Plan Level	Entries
<a href="#">Edit this ad</a>	23/05/2018	Google	<a href="https://www.google.com">https://www.google.com</a>		1	4
<a href="#">Edit this ad</a>	23/04/2019	Microsoft	<a href="https://www.microsoft.com">https://www.microsoft.com</a>		1	2
<a href="#">Edit this ad</a>	24/08/2018	The Real Facebook - TRF	<a href="https://www.facebook.com">https://www.facebook.com</a>		1	1
<a href="#">Edit this ad</a>	01/06/2018	Scratch	<a href="https://scratch.mit.edu/projects/editor?ip_bar=getStarted">https://scratch.mit.edu/projects/editor?ip_bar=getStarted</a>		1	2


ניתן ללחוץ על העריכה ואז מופיע הפאנל שבו ניתן לערוך את הפרסומות הספציפית כדוגמת

My Published Ads

[Go to advertise an ad](#)

Advertiser Name: Google

Advised Website URL: <https://www.google.com>

Advertisement Image: 

לא נבחר קובץ | בחירת קובץ

Cancel the image changing | Update

בדף הזה ניתן לעבור לדף לקניית פרסומות, שם משתמש מפרסם פרסומות חדשות.

## BuyAd.aspx

בדף זה ניתן להזמין פרסומות, המשתמש בוחר תחילה חבילת פרסום (נמוכה/רגילה/גבוהה)

### Advertise Ad

Low Plan

\* Lowest amount of shows per month  
\* Cheapest plan 10\$ per month

Choose

Normal Plan

\* Middle amount of shows per month  
\* Good price/shows ratio  
\* 20\$ per month

Choose

High Plan

\* Best shows per month  
\* High cost, 30\$ per month

Choose

לאחר מכן המשתמש מתבקש להזין את פרטי הפרסומת – חודשים לפרסום, שם מפרסם, אתר לקישור, תמונת פרסומת.

### Advertise Ad

Low Plan - 10\$ per month

Amount of time for advertising: Choose Amount of Months

Advertiser Name:

Advised Website URL:

Advertisement Image:

לא נבחר קובץ | בחירת קובץ

Proceed to check out

לבסוף המשתמש צריך להזין פרטי כרטיס אשראי, תקינות פרטי הכרטיס נבדקים בweb services אילו כל הפרטים תקינים וגם יש מספיק יתרה בכרטיס ההזמנה של הפרסומת מבוצעת והפרסומת נכנסת למאגר הפרסומות שיש בטבלה.

### Advertise Ad

Payment

Credit Card Type:

Enter your ID:

Credit Card Number:

Credit Card Cvv:

Expiration Date: Month  Year

Check Out

## מדריך האפליקציה למנהל

### UsersManager.aspx

בעמוד זה למנהל יש שליטה על כל המשתמשים ועל כל הפרסומות שמפורסמות באתר. המנהל יכול למחוק ולערוך כל משתמש, וגם למחוק ולערוך כל פרסומת. אם המשתמש מנהל יופיע קישור בשורה למעלה. (בתמונה מסומן באדום)

**Control users info**

	First Name	Last Name	Password	Email	Gender	Birthday	User ID	Profile Image	Is Manager
<a href="#">Edit/Delete</a>	abc	def	h	h	Male	23/04/1984 00:00:00	8		<a href="#">Edit</a>
<a href="#">Edit/Delete</a>	h	h	h	h	Female	01/01/2017 00:00:00	11		<a href="#">Edit</a>
<a href="#">Edit/Delete</a>	h	h	h	h	Male	16/03/1993 00:00:00	12		<a href="#">Edit</a>
<a href="#">Edit/Delete</a>	l	l	l	l	Male	17/10/1999 00:00:00	13		<a href="#">Edit</a>
<a href="#">Edit/Delete</a>	g	g	g	g	Female	13/01/2000 00:00:00	14		<a href="#">Edit</a>

1 2

**Control Ads**

	Last Date	Advertiser Name	Website URL	Ad Image	Ad ID	Pin	Level	Entries
<a href="#">Edit/Delete</a>	23/05/2018 00:00:00	Google	<a href="https://www.google.com">https://www.google.com</a>		3	1	4	
<a href="#">Edit/Delete</a>	23/04/2019 00:00:00	Microsoft	<a href="https://www.microsoft.com/">https://www.microsoft.com/</a>		5	1	2	
<a href="#">Edit/Delete</a>	24/06/2018 00:00:00	The Real Facebook - TRF	<a href="https://www.facebook.com/">https://www.facebook.com/</a>		6	1	1	
<a href="#">Edit/Delete</a>	01/06/2018 00:00:00	Scratch	<a href="https://scratch.mit.edu/projects/editor/?ip_hamper&amp;start=0">https://scratch.mit.edu/projects/editor/?ip_hamper&amp;start=0</a>		8	1	2	

למשתמש ספציפי שהוא לא מנהל ישנה האפשרות למחוק רק פוסטים שלו או פוסטים שפורסמו על הדף שלו אך למנהל האפשרות למחוק את כל הפוסטים של כל האנשים.



## מבנה בסיס הנתונים

### טבלת Users – משתמשים

dbo.Users: Tabl...TA\DATABASE.MDF) ✕			
	Column Name	Data Type	Allow Nulls
▶	FirstName	nvarchar(50)	<input type="checkbox"/>
	LastName	nvarchar(50)	<input type="checkbox"/>
	Password	nvarchar(50)	<input type="checkbox"/>
	Email	nvarchar(50)	<input type="checkbox"/>
	Gender	nvarchar(50)	<input type="checkbox"/>
	Birthday	date	<input type="checkbox"/>
🔑	UserID	int	<input type="checkbox"/>
	ProfileImage	nvarchar(MAX)	<input checked="" type="checkbox"/>
	IsManager	bit	<input type="checkbox"/>
			<input type="checkbox"/>

### טבלת Posts – פוסטים

dbo.Posts: Tabl...TA\DATABASE.MDF) ✕			
	Column Name	Data Type	Allow Nulls
▶	Contents	nvarchar(MAX)	<input type="checkbox"/>
	CommentUser	int	<input type="checkbox"/>
	CommentPost	int	<input type="checkbox"/>
	CommentPage	int	<input type="checkbox"/>
	Date	datetime	<input type="checkbox"/>
🔑	PostID	int	<input type="checkbox"/>
	PublisherID	int	<input type="checkbox"/>
			<input type="checkbox"/>

### טבלת Messages – הודעות

dbo.Messages: Ta...TA\DATABASE.MDF) ✕			
	Column Name	Data Type	Allow Nulls
	SenderUserID	int	<input type="checkbox"/>
	GettingUserID	int	<input type="checkbox"/>
	Date	datetime	<input type="checkbox"/>
	Contents	nvarchar(MAX)	<input type="checkbox"/>
🔑	MessageID	int	<input type="checkbox"/>
	IsNew	bit	<input type="checkbox"/>
▶			<input type="checkbox"/>

## טבלת Likes – לייקים

dbo.Likes: Table...TA\DATABASE.MDF) ✕			
	Column Name	Data Type	Allow Nulls
▶	LikeID	int	<input type="checkbox"/>
	PostID	int	<input type="checkbox"/>
	UserID	int	<input type="checkbox"/>
	Date	datetime	<input type="checkbox"/>
			<input type="checkbox"/>

## טבלת Friendships – חברויות קיימות

dbo.Friendships:...TA\DATABASE.MDF) ✕			
	Column Name	Data Type	Allow Nulls
▶	UserID	int	<input type="checkbox"/>
	FriendID	int	<input type="checkbox"/>
	Date	datetime	<input type="checkbox"/>
			<input type="checkbox"/>

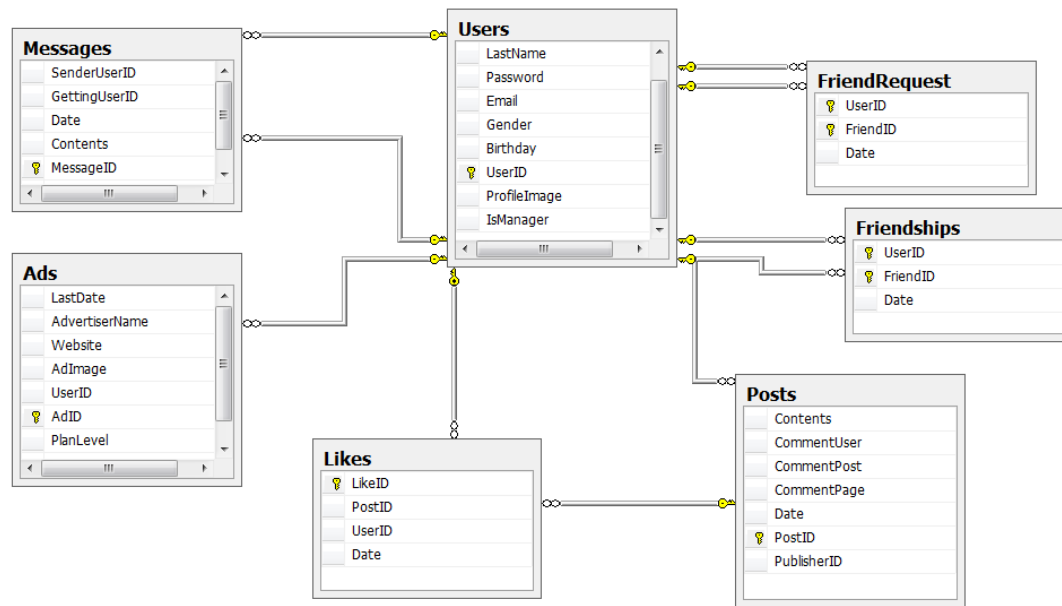
## טבלת FriendRequests – בקשות חברות

dbo.FriendReque...A\DATABASE.MDF) ✕			
	Column Name	Data Type	Allow Nulls
▶	UserID	int	<input type="checkbox"/>
	FriendID	int	<input type="checkbox"/>
	Date	date	<input type="checkbox"/>
			<input type="checkbox"/>

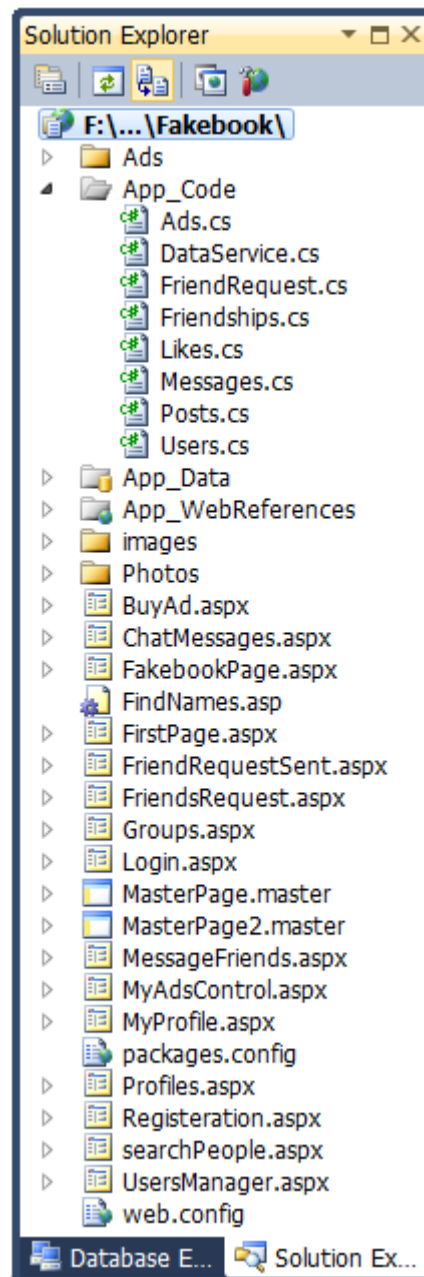
## טבלת Ads – פרסומות

dbo.Ads: Table(...TA\DATABASE.MDF) ✕			
	Column Name	Data Type	Allow Nulls
▶	LastDate	date	<input type="checkbox"/>
	AdvertiserName	nvarchar(50)	<input type="checkbox"/>
	Website	nvarchar(150)	<input type="checkbox"/>
	AdImage	nvarchar(50)	<input type="checkbox"/>
	UserID	int	<input type="checkbox"/>
	AdID	int	<input type="checkbox"/>
	PlanLevel	int	<input type="checkbox"/>
	Entries	int	<input type="checkbox"/>
			<input type="checkbox"/>

## תמונת/תמונות קשרי הגומלין מתוך מסד הנתונים ( DSD )







## המחלקות

### Users

```
public class Users
{
    private string firstName;//שם פרטי
    private string lastName;//שם משפחה
    private int userId;//הזיהוי של המשתמש
    private string email;//אימייל של המשתמש
    private string userPassword;//סיסמה של המשתמש
    private string gender;//מין של המשתמש
    private string profileImage;//תמונת הפרופיל
    private DateTime date;//תאריך הרשמה
    private bool isManager;האם מנהל

    public Users(string firstName, string lastName, string email, string userPassword, string gender, DateTime date, string profileImage = "", int userId = -1, bool isManager = false)
    {
        this.firstName = firstName;
        this.lastName = lastName;
        this.userId = userId;
        this.email = email;
        this.userPassword = userPassword;
        this.gender = gender;
        this.isManager = isManager;
        if (profileImage != "")
            this.profileImage = profileImage;
        else
        {
            if (gender == "Male")
                this.profileImage = "images/Male.gif";
            else
                this.profileImage = "images/Female.jpg";
        }
        this.date = date;
    }
    //פונקציה בונה
    public int UserID
    {
        get { return this.userId; }
    }
    //פונקציה נט של אידי
    public bool IsManager
    {
        get { return this.isManager; }
        set { this.isManager = value; }
    }
    //פונקציה נט וטס של האם מנהל
    public string FirstName
    {
        get { return this.firstName; }
    }
    //פונקציה נט של שם פרטי
    public string LastName
    {
        get { return this.lastName; }
    }
    //פונקציה נט של שם משפחה
    public string Gender
    {
        get { return this.gender; }
    }
    //פונקציה נט של מין
    public string ProfileImage
    {
        get { return this.profileImage; }
        set { this.profileImage = value; }
    }
    //פונקציה נט וטס של תמונת פרופיל
    public string Email
    {
        get { return this.email; }
    }
    //פונקציה נט של אימייל
    public static bool Exist(string email)
    {
        string strSql = string.Format("select * from Users where Email='{0}'", email);
        Object obj = DataService.ExecuteScalar(strSql);
        return obj != null;
    }
    //פונקציה המזירה האם האימייל קיים
    public static bool Exist(int userID)
    {
        string strSql = string.Format("select * from Users where userID='{0}'", userID);
        Object obj = DataService.ExecuteScalar(strSql);
        return obj != null;
    }
    //פונקציה המזירה האם המשתמש תקין/קיים
    public static bool Exist(string email, string password)
    {
        string strSql = string.Format("select * from Users where Email='{0}' AND Password='{1}'", email,password);
        Object obj = DataService.ExecuteScalar(strSql);
        return obj != null;
    }
    //פונקציה המזירה האם קיים משתמש באימייל והסיסמה - האם נכונה ההתאבדות תקינים
}
```

```
public static Users GetUser(string email, string password)
{
    string strSql = string.Format("select * from Users where Email='{0}' AND Password='{1}'", email, password);
    DataSet ds = DataService.GetDataSet(strSql, "User");
    if (ds == null || ds.Tables[0].Rows.Count == 0)
        return null;
    int userID = int.Parse(ds.Tables[0].Rows[0]["UserID"].ToString());
    string firstName = ds.Tables[0].Rows[0]["FirstName"].ToString();
    string lastName = ds.Tables[0].Rows[0]["LastName"].ToString();
    string gender = ds.Tables[0].Rows[0]["Gender"].ToString();
    string profileImage = ds.Tables[0].Rows[0]["ProfileImage"].ToString();
    DateTime date = DateTime.Parse(ds.Tables[0].Rows[0]["Birthday"].ToString());
    bool isManager = bool.Parse(ds.Tables[0].Rows[0]["IsManager"].ToString());
    return new Users(firstName, lastName, email, password, gender, date, profileImage, userID, isManager);
}
//הצגת כל המידע של כל המשתמשים
public static DataSet GetAllUsers()
{
    string strSql = string.Format("select * from Users");
    DataSet ds = DataService.GetDataSet(strSql, "User");
    if (ds == null || ds.Tables[0].Rows.Count == 0)
        return null;
}
//הצגת מידע על המשתמשים על פי שם המשתמש
public static DataSet GetUser(string name)
{
    string strSql = string.Format("select * from Users where (FirstName = '{0}' OR LastName Like '{0}%')", name);
    DataSet ds = DataService.GetDataSet(strSql, "User");
    if (ds == null || ds.Tables[0].Rows.Count == 0)
        return null;
}
//הצגת מידע על המשתמשים על פי ID
public static DataSet GetUser(int userID)
{
    string strSql = string.Format("select * from Users where UserID={0}", userID);
    DataSet ds = DataService.GetDataSet(strSql, "User");
    if (ds == null || ds.Tables[0].Rows.Count == 0)
        return null;
}
//הצגת מידע על המשתמשים על פי ID
public static Users GetUser(int userID)
{
    string strSql = string.Format("select * from Users where UserID={0}", userID);
    DataSet ds = DataService.GetDataSet(strSql, "User");
    if (ds == null || ds.Tables[0].Rows.Count == 0)
        return null;
    string email = ds.Tables[0].Rows[0]["Email"].ToString();
    string password = ds.Tables[0].Rows[0]["Password"].ToString();
    string firstName = ds.Tables[0].Rows[0]["FirstName"].ToString();
    string lastName = ds.Tables[0].Rows[0]["LastName"].ToString();
    string gender = ds.Tables[0].Rows[0]["Gender"].ToString();
    string profileImage = ds.Tables[0].Rows[0]["ProfileImage"].ToString();
    DateTime date = DateTime.Parse(ds.Tables[0].Rows[0]["Birthday"].ToString());
    return new Users(firstName, lastName, email, password, gender, date, profileImage, userID);
}
//הצגת מידע על המשתמשים על פי ID
public int Update()
{
    int rowsAffected;
    string strSql = string.Format("Update Users Set FirstName='{0}', LastName='{1}', Email='{2}', Password='{3}', Gender='{4}', ProfileImage='{5}' where UserID={6}", firstName, lastName, email, userPassword, gender, profileImage, userID);
    rowsAffected = (int)DataService.ExecuteNonQuery(strSql);
    return rowsAffected;
}
//הצגת מידע על המשתמשים
public static int Delete(int userID)
{
    int rowsAffected;
    string strSql = string.Format("Delete From Users where UserID={0}", userID);
    rowsAffected = (int)DataService.ExecuteNonQuery(strSql);
    return rowsAffected;
}
//הצגת מידע על המשתמשים
public static int Delete(int userID)
{
    int rowsAffected;
    string strSql = string.Format("Delete From Users where UserID={0}", userID);
    rowsAffected = (int)DataService.ExecuteNonQuery(strSql);
    return rowsAffected;
}
//הצגת מידע על המשתמשים
public int Insert()
{
    int rowsAffected;
    string strSql = "Insert into Users(FirstName,LastName,Password,Email,Gender,Birthday,ProfileImage, IsManager) Values ('" + firstName + "','" + lastName + "','" + userPassword + "','" + email + "','" + gender + "','" + date.ToString("MM/dd/yyyy") + "','" + profileImage + "','" + isManager + "')";
    rowsAffected = (int)DataService.ExecuteNonQuery(strSql);
    this.userID = GetUser(email, userPassword).UserID;
    return rowsAffected;
}
//הצגת מידע על המשתמשים
```

## Ads

```
public class Ads
{
    private DateTime lastDate;//תאריך האחרון
    private string advertisementName;//שם הפרסום
    private string websiteUrl;//כתובת האתר
    private string imagePath;//תמונת הפרסום
    private int userID;//מזהה המשתמש
    private int planLevel;//רמת הפרסום
    private int entries;//מספר הפרסומים

    //מטודה
    public Ads(DateTime lastDate, string advertisementName, string websiteUrl, string imagePath, int userID, int planLevel, int entries)
    {
        this.lastDate = lastDate;
        this.advertisementName = advertisementName;
        this.websiteUrl = websiteUrl;
        this.imagePath = imagePath;
        this.userID = userID;
        this.planLevel = planLevel;
        this.entries = entries;
    }

    public static DataSet GetAllAds()
    {
        string strSql = string.Format("select * from Ads");
        return DataService.GetDataSet(strSql, "Ads");
    }

    //הצגת מידע על הפרסומים
    public static int DeleteUser(int userID)
    {
        string strSql = string.Format("Delete from Ads where UserID={0}", userID);
        return DataService.ExecuteNonQuery(strSql);
    }

    //הצגת מידע על הפרסומים
    public static DataSet GetUserAds(int userID)
    {
        string strSql = string.Format("select * from Ads where UserID={0}", userID);
        return DataService.GetDataSet(strSql, "Ads");
    }

    //הצגת מידע על הפרסומים
    public static void Delete(int adID)
    {
        string strSql = string.Format("Delete from Ads where AdID={0}", adID);
        DataService.ExecuteNonQuery(strSql);
    }

    //הצגת מידע על הפרסומים
    public static int updated(int adID, string advertisementName, string website)
    {
        string strSql = string.Format("Update Ads Set AdvertisementName='{0}', Website='{1}' where AdID={2}", advertisementName, website, adID);
        int rowsAffected = (int)DataService.ExecuteNonQuery(strSql);
        return rowsAffected;
    }

    //הצגת מידע על הפרסומים
    private static int GetEntries(int adID)
    {
        string strSql = string.Format("select Entries from Ads where AdID={0}", adID);
        return int.Parse(DataService.ExecuteScalar(strSql).ToString());
    }

    //הצגת מידע על הפרסומים
    public static void AddEntry(int adID)
    {
        int entries = GetEntries(adID);
        string strSql = string.Format("Update Ads Set Entries={0} where AdID={1}", entries + 1, adID);
        DataService.ExecuteNonQuery(strSql);
    }

    //הצגת מידע על הפרסומים
    public static string GetLinkFromID(int adID)
    {
        string strSql = string.Format("select Website from Ads where AdID={0}", adID);
        return DataService.ExecuteScalar(strSql).ToString();
    }

    //הצגת מידע על הפרסומים
    public int Insert()
    {
        int rowsAffected;
        string strSql = string.Format("Insert into Ads(lastDate,advertisementName,website,AdImage,userID,PlanLevel,Entries) Values ('{0}','{1}','{2}','{3}','{4}','{5}','{6}')", lastDate.ToString("MM/dd/yyyy HH:mm:ss"), advertisementName, websiteUrl, imagePath, userID, planLevel, entries);
        rowsAffected = (int)DataService.ExecuteNonQuery(strSql);
        return rowsAffected;
    }

    //הצגת מידע על הפרסומים
}
```

# Friendships

```
public class Friendships
{
    private int userID; //א"י של משתמש
    private int friendID; //א"י של חבר
    private DateTime date; //תאריך חברים מאז תאריך

    public Friendships(int userID, int friendID, DateTime date)
    {
        this.userID = userID;
        this.friendID = friendID;
        this.date = date;
    }

    //פעולה בונה
    public static DataSet GetFriends(int userID)
    {
        string strSql = string.Format("select * from Friendships where (UserID='{0}']", userID);
        DataSet ds = DataService.GetDataSet(strSql, "Friendships");

        if (ds == null || ds.Tables[0].Rows.Count == 0)
            return null;

        return ds;
    }

    //פעולה המחזירה דטה שש של כל החברים של משתמש ספציפי
    //public static DataSet GetFriendsWithName(int userID)
    //{
    //    string strSql = string.Format("SELECT a.*, b.FirstName + ' ' + b.LastName AS FriendName FROM Friendships AS a INNER JOIN Users AS b ON a.FriendID = b.UserID WHERE (a.U
    //    DataSet ds = DataService.GetDataSet(strSql, "Friendships");

    //    if (ds == null || ds.Tables[0].Rows.Count == 0)
    //        return null;

    //    return ds;
    //}

    //פעולה המחזירה דטה שש של כל החברים של משתמש ספציפי ומוסיפה גם עמודה של שם של החבר - אינר ג'ויין עם הטבלה של משתמשים
    public static DataSet GetFriendsWithName(int userID)
    {
        string strSql = string.Format(@"SELECT Friendships.FriendID, Friendships.Date, (case when Messages.SenderUserID=Friendships.UserID then 'true' else 'false' end) As IsMe,
            Messages.IsNew, Messages.Date As LastMessageDate, Users.FirstName + ' ' + Users.LastName As FriendName,
            ROW_NUMBER() OVER (PARTITION BY Friendships.FriendID ORDER BY Messages.MessageID DESC) AS RowNumber FROM
            Friendships INNER JOIN Users ON Friendships.FriendID = Users.UserID Left JOIN Messages ON
            (Friendships.FriendID = Messages.SenderUserID And Friendships.UserID = Messages.GettingUserID) OR
            (Friendships.FriendID = Messages.GettingUserID And Friendships.UserID = Messages.SenderUserID)) WHERE
            (Friendships.UserID = '{0}') ORDER BY Messages.MessageID DESC", userID);

        DataSet ds = DataService.GetDataSet(strSql, "Friendships");

        if (ds == null || ds.Tables[0].Rows.Count == 0)
            return null;

        DataTable dt = ds.Tables[0].Select("RowNumber='1'").CopyToDataTable();
        DataSet ds2 = new DataSet();
        ds2.Tables.Add(dt);
        return ds2;
    }

    //פעולה המחזירה דטה שש של כל החברים של משתמש ספציפי ומוסיפה גם עמודה של שם של החבר - אינר ג'ויין עם הטבלה של משתמשים
    public static bool ExistFriendship(int userID, int friendID)
    {
        string strSql = string.Format("select Date from Friendships where (UserID='{0}' AND FriendID='{1}']", userID, friendID);
        Object obj = DataService.ExecuteScalar(strSql);
        return obj != null;
    }

    //פעולה המחזירה האם קיים חברות בין שני משתמשים ספציפיים
    public static void DeleteFriendship(int userID, int friendID)
    {
        string strSql = string.Format("Delete From Friendships Where UserID={0} And FriendID={1}", userID, friendID);
        DataService.ExecuteNonQuery(strSql);
        strSql = string.Format("Delete From Friendships Where UserID={1} And FriendID={0}", userID, friendID);
        DataService.ExecuteNonQuery(strSql);
    }

    //פעולה הוחזקה חברים בין שני חברים
    public static void DeletedUserFriendships(int userID)
    {
        string strSql = string.Format("Delete From Friendships Where UserID={0} OR FriendID={0}", userID);
        DataService.ExecuteNonQuery(strSql);
    }

    //מחזיקת כל החברויות של משתמש ספציפי
    public int Insert()
    {
        int rowsAffected;
        string strSql = string.Format("Insert into Friendships(UserID,FriendID,Date) Values ('{0}','{1}','{2}']", userID, friendID, date.ToString("MM/dd/yyyy HH:mm:ss"));
        rowsAffected = (int)DataService.ExecuteNonQuery(strSql);
        strSql = string.Format("Insert into Friendships(UserID,FriendID,Date) Values ('{0}','{1}','{2}']", friendID, userID, date.ToString("MM/dd/yyyy HH:mm:ss"));
        rowsAffected += (int)DataService.ExecuteNonQuery(strSql);
        return rowsAffected;
    }

    //הכנסת חברים לטבלה
}
```

## Messages

```
public class Messages
{
    אידי של המשתמש השולח את ההודעה
    private int senderUserID;
    אידי של המשתמש המקבל את ההודעה
    private int gettingUserID;
    תאריך שבו נשלחה ההודעה
    private DateTime date;
    תוכן של ההודעה
    private string contents;
    private bool isNew;

    public Messages(int senderUserID, int gettingUserID, DateTime date, string contents, bool isNew=true)
    {
        this.senderUserID = senderUserID;
        this.gettingUserID = gettingUserID;
        this.date = date;
        this.contents = contents;
        this.isNew = isNew;
    }
    //שורה בונה
    public static DataSet GetMessages(int user1, int user2)
    {
        string strSql = string.Format("select * from Messages where (SenderUserID='{0}' And GettingUserID='{1}') OR (SenderUserID='{1}' And GettingUserID='{0}') Order By MessageID Desc",
            user1, user2);
        DataSet ds = DataService.GetDataSet(strSql, "Messages");
        return ds;
    }
    //שורה המאשרת דטה שס המכיל את הסייה בין שני חברים
    public static void SetRead(int readingUserID, int sentUserID)
    {
        string strSql = string.Format("Update Messages Set IsNew='false' Where SenderUserID='{0}' And GettingUserID='{1}' And IsNew='True'", sentUserID, readingUserID);
        DataService.ExecuteNonQuery(strSql);
    }
    //שורה מחזיקה את התוצר של ההודעה לקורא
    public static bool IsUnreadMessages(int userID)
    {
        string strSql = string.Format("select * from Messages Where GettingUserID='{0}' And IsNew='True'", userID);
        object obj = DataService.ExecuteScalar(strSql);
        return obj != null;
    }
    public int Insert()
    {
        int rowsAffected;
        string strSql = string.Format("Insert into Messages(SenderUserID,GettingUserID,Date,Contents,IsNew) Values ('{0}','{1}','{2}','{3}','{4}')",
            senderUserID, gettingUserID, date.ToString("MM/dd/yyyy HH:mm:ss"), contents, isNew);
        rowsAffected = (int)DataService.ExecuteNonQuery(strSql);

        return rowsAffected;
    }
    //שורה המכניסה הודעה לסבלה
    public static void DeleteAllMessages(int userID)
    {
        string strSql = string.Format("Delete From Messages Where SenderUserID='{0}' OR GettingUserID='{0}'", userID);
        DataService.ExecuteNonQuery(strSql);
    }
    //שורה המוחקת את כל הסייה של משתמש ספציפי
}
```

## Likes

```
public class Likes
{
    אידי של הלייק
    private int likeID;
    אידי של הפוסט עליו נעשה לייק
    private int postID;
    אידי של המשתמש שביצע לייק
    private int userID;
    תאריך שבו נעשה הלייק
    private DateTime date;

    public Likes(int postID, int userID, DateTime date, int likeID = -1)
    {
        this.likeID = likeID;
        this.postID = postID;
        this.userID = userID;
        this.date = date;
    }
    //שורה בונה
    public int Insert()
    {
        int rowsAffected;
        string strSql = string.Format("Insert into Likes(PostID,UserID,Date) Values ('{0}','{1}','{2}')", postID, userID, date.ToString("MM/dd/yyyy HH:mm:ss"));
        rowsAffected = (int)DataService.ExecuteNonQuery(strSql);

        if (this.likeID == -1)
        {
            strSql = string.Format("Select LikeID from Likes Where (PostID='{0}' And UserID = '{1}' And Date = '{2}']", postID, userID, date.ToString("MM/dd/yyyy HH:mm:ss"));
            this.likeID = int.Parse(DataService.ExecuteScalar(strSql).ToString());
        }

        return rowsAffected;
    }
    //הכנסה של הלייק לסבלה
    public static bool LikeExist(int postID, int userID)
    {
        string strSql = string.Format("select * from Likes where PostID='{0}' And UserID='{1}'", postID, userID);
        DataSet ds = DataService.GetDataSet(strSql, "Likes");
        return ds != null;
    }
    //שורה המחזירה חכם המשתמש עשה לייק על פוסט ספציפי
    public static DataSet GetAllLikes(int postID)
    {
        string strSql = string.Format("select * from Likes where PostID='{0}'", postID);
        DataSet ds = DataService.GetDataSet(strSql, "Likes");
        return ds;
    }
    //שורה המחזירה דטה שס של כל הלייקים על פוסט ספציפי
    public static int LikesAmount(int postID)
    {
        string strSql = string.Format("select Count(PostID) from Likes where PostID='{0}'", postID);
        return int.Parse(DataService.ExecuteScalar(strSql).ToString());
    }
    //שורה המחזירה את מספר הלייקים על פוסט ספציפי
    public static void DeleteLike(int likeID = -1, int userID = -1, int postID = -1)
    {
        if (likeID != -1)
        {
            string strSql = string.Format("Delete From Likes Where LikeID='{0}'", likeID);
            DataService.ExecuteNonQuery(strSql);
        }
        else if (userID != -1 && postID != -1)
        {
            string strSql = string.Format("Delete From Likes Where userID='{0}' And postID='{1}'", userID, postID);
            DataService.ExecuteNonQuery(strSql);
        }
        else if (postID != -1)
        {
            string strSql = string.Format("Delete From Likes Where postID='{0}'", postID);
            DataService.ExecuteNonQuery(strSql);
        }
    }
    //שורה המוחקת לייק לפי האידי של הלייק או לפי המשתמש
    //שורה לייק והאידי של הפוסט או לפי אידי של פוסט ספציפי
    public static void DeleteLikeAllLikes(int userID)
    {
        string strSql = string.Format("Delete From Likes Where userID='{0}'", userID);
        DataService.ExecuteNonQuery(strSql);
    }
    //שורה המוחקת את כל הלייקים שעשה משתמש ספציפי
}
```

## FriendRequest

```
public class FriendRequest
{
    private int userID;//משתמש השולח הזמנה
    private int friendID;//משתמש המקבל הזמנה
    private DateTime date;//שליחת ההזמנה

    public FriendRequest(int userID, int friendID, DateTime date)
    {
        this.userID = userID;
        this.friendID = friendID;
        this.date = date;
    }

    //פעולה בונה
    public static DataSet GetPending(int gotInvite)
    {
        string strSql = string.Format("select * from FriendRequest where FriendID='{0}'", gotInvite);
        DataSet ds = DataService.GetDataSet(strSql, "FriendRequests");

        if (ds == null || ds.Tables[0].Rows.Count == 0)
            return null;
        return ds;
    }

    //פעולה המחזירה דטה סט של בקשות שנשלחו למשתמש ספציפי
    public static DataSet SentRequests(int sentID)
    {
        string strSql = string.Format("select * from FriendRequest where UserID='{0}'", sentID);
        DataSet ds = DataService.GetDataSet(strSql, "FriendRequests");

        if (ds == null || ds.Tables[0].Rows.Count == 0)
            return null;
        return ds;
    }

    //פעולה המחזירה האם קיים בקשת חברות בין שני משתמשים
    public static bool ExistFriendRequest(int sentInvite, int gotInvite)
    {
        string strSql = string.Format("select Date from FriendRequest where (UserID='{0}' AND FriendID='{1}']", sentInvite, gotInvite);
        object obj = DataService.Executescalar(strSql);
        strSql = string.Format("select Date from FriendRequest where (UserID='{1}' AND FriendID='{0}']", sentInvite, gotInvite);
        object obj2 = DataService.Executescalar(strSql);
        return obj != null || obj2 != null;
    }

    //פעולה המחזירה האם משתמש ספציפי קיבל הזמנת חברות ממשתמש ספציפי אחר
    public static bool GotFriendRequestFrom(int gotInvite, int sentInvite)
    {
        string strSql = string.Format("select Date from FriendRequest where (UserID='{0}' AND FriendID='{1}']", sentInvite, gotInvite);
        object obj = DataService.Executescalar(strSql);
        return obj != null;
    }

    //פעולה המחזירה האם משתמש ספציפי קיבל הזמנת חברות ממשתמש ספציפי אחר
    public static void AcceptFriendRequest(int sentInvite, int gotInvite)
    {
        FriendRequest.DeleteFriendRequest(sentInvite, gotInvite);
        new Friendships(sentInvite, gotInvite, DateTime.Now).Insert();
    }

    //פעולה המחזירה האם משתמש ספציפי קיבל הזמנת חברות ממשתמש ספציפי אחר
    public static void DeleteFriendRequest(int sentInvite, int gotInvite)
    {
        string strSql = string.Format("Delete From FriendRequest Where UserID={0} And FriendID={1}", sentInvite, gotInvite);
        DataService.ExecuteNonQuery(strSql);
    }

    //פעולה המחזירה האם משתמש ספציפי קיבל הזמנת חברות ממשתמש ספציפי אחר
    public static void DeletedUserFriendRequests(int userID)
    {
        string strSql = string.Format("Delete From FriendRequest Where UserID={0} OR FriendID={0}", userID);
        DataService.ExecuteNonQuery(strSql);
    }

    //מחזיקת כל ההזמנות חברות של משתמש
    public int Insert()
    {
        int rowsAffected;
        string strSql = string.Format("Insert into FriendRequest(UserID,FriendID,Date) values ('{0}','{1}','{2}']", userID, friendID, date.ToString("MM/dd/yyyy HH:mm:ss"));
        rowsAffected = (int)DataService.ExecuteNonQuery(strSql);
        return rowsAffected;
    }

    //הכנסת הזמנת בקשת חברות לטבלה
}
```

# Posts

```
public class Posts
{
    private int commentUser;//מיידי של מחברת הערה
    private int commentPost;//מיידי של הוספת הערה
    private int commentPage;//מיידי של העמוד
    private int publisherID;//מיידי של הוצאת הספרים
    private int postID;//מיידי של ההודעה
    private string contents;//תוכן ההודעה
    private DateTime date;//תאריך בו הודעה הוספה

    public Posts(int publisherID, string contents, DateTime date, int postID = -1, int commentUser = -1, int commentPost = -1, int commentPage = -1)
    {
        this.commentUser = commentUser;
        this.publisherID = publisherID;
        this.contents = contents;
        this.date = date;
        this.commentPost = commentPost;
        this.commentPage = commentPage;
        this.postID = postID;
    }

    //מציג את הpostID
    public int PostID
    {
        get { return this.postID; }
        set { this.postID = value; }
    }

    //מציג את תוכן ההודעה
    public string Contents
    {
        get { return this.contents; }
        set { this.contents = value; }
    }

    //מציג את תאריך ההודעה
    public int Insert()
    {
        int rowsAffected;
        string strSql = string.Format("Insert into Posts(Contents,CommentUser,CommentPost,CommentPage,Date,PublisherID) values ('{0}','{1}','{2}','{3}','{4}','{5}')", contents, commentUser, commentPost, commentPage, date.ToString("MM/dd/yyyy HH:mm:ss"), publisherID);
        rowsAffected = (int)DataService.ExecuteNonQuery(strSql);

        if (this.postID == -1)
        {
            strSql = string.Format("Select PostID from Posts where (Contents='{0}' And PublisherID = '{1}' And Date = '{2}')", contents, publisherID, date.ToString("MM/dd/yyyy HH:mm:ss"));
            this.postID = int.Parse(DataService.ExecuteScalar(strSql).ToString());
        }

        return rowsAffected;
    }

    //מציג את כל הposts
    public static DataSet GetAllPosts(int commentUser = -1, int commentPost = -1, int commentPage = -1, int publisherID = -1)
    {
        string strSql = "";
        if (publisherID != -1 && commentUser != -1)
            strSql = string.Format("select * from Posts where PublisherID='{0}' OR CommentUser='{1}'", publisherID, commentUser);
        else if (publisherID != -1)
            strSql = string.Format("select * from Posts where PublisherID='{0}'", publisherID);
        else if (commentUser != -1 && commentPost != -1)
            strSql = string.Format("select * from Posts where CommentUser='{0}' And CommentPost='{1}' And CommentPage='{2}' Order By PostID Desc", commentUser, commentPost, commentPage);
        else
            strSql = string.Format("select * from Posts where CommentUser='{0}' And CommentPost='{1}' And CommentPage='{2}'", commentUser, commentPost, commentPage);
        DataSet ds = DataService.GetDataSet(strSql, "Posts");
        return ds;
    }

    //מציג את כל הposts
    public static void DeletePost(int postID)
    {
        string strSql = string.Format("Delete from Posts where PostID={0}", postID);
        Likes.DeleteLike(-1, -1, postID);
        DataService.ExecuteNonQuery(strSql);
        DataSet ds = GetAllPosts(-1, postID);
        if (ds != null)
        {
            foreach (DataRow dr in ds.Tables[0].Rows)
            {
                DeletePost(int.Parse(dr["PostID"].ToString()));
            }
        }

        public static void DeleteAllUserPosts(int userID)
        {
            DataSet ds = GetAllPosts(userID, -1, -1, userID);
            if (ds != null)
            {
                foreach (DataRow dr in ds.Tables[0].Rows)
                {
                    DeletePost(int.Parse(dr["PostID"].ToString()));
                }
            }
        }

    }
}
```

## DataService

```
public class DataService
{
    static private string path; // מסלול
    static public string Path
    {
        set { path = value; }
        get { return path; }
    }
    // פעולת המאגירת / מעדכנת את המסלול
    static public string ConnectionString
    {
        get
        {
            return @"Data Source=.\SQLEXPRESS;AttachDbFilename=" + DataService.path + @";Integrated Security=True;User Instance=True";
        }
    }
    // פעולת המאגירת את Connection String
    static public object ExecuteScalar(string strSql)
    {
        object obj = null;
        SqlConnection connection = new SqlConnection(DataService.ConnectionString);
        SqlCommand command = new SqlCommand(strSql, connection);
        try
        {
            connection.Open();
            obj = command.ExecuteScalar();
        }
        catch (Exception ex)
        {
            return null;
        }
        finally
        {
            connection.Close();
        }
        return obj;
    }
    // פעולת בחירה שמחזירה ערך בודד
    static public int ExecuteNonQuery(string strSql)
    {
        int rowsAffected;
        SqlConnection connection = new SqlConnection(DataService.ConnectionString);
        SqlCommand command = new SqlCommand(strSql, connection);
        try
        {
            connection.Open();
            rowsAffected = command.ExecuteNonQuery();
        }
        catch (Exception ex)
        {
            return -1;
        }
        finally
        {
            connection.Close();
        }
        return rowsAffected;
    }
    // פעולת עדכון/הוספה/מחיקה
}
```



```

static public object ExecuteReader(string strSql)
{
    string output="";
    SqlConnection connection = new SqlConnection(DataService.ConnectionString);
    SqlDataReader dr = null;
    SqlCommand command = new SqlCommand(strSql, connection);
    try
    {
        connection.Open();
        dr = command.ExecuteReader();
        while (dr.Read())
        {
            output += dr["FirstName"].ToString();
        }
    }
    catch (Exception ex)
    {
        return null;
    }
    finally
    {
        connection.Close();
    }

    if (output == "")
        return null;
    return output;
}
//פעולה אחזור ליותר מנתון אחד
public static DataSet GetDataSet(string sql, string tblName)
{
    DataSet ds = new DataSet();
    SqlConnection connection = new SqlConnection(DataService.ConnectionString);
    SqlCommand cmd = new SqlCommand(sql, connection);
    SqlDataAdapter ad = new SqlDataAdapter(cmd);

    try
    {
        ad.Fill(ds, tblName);
    }
    catch (Exception ex)
    {
        return null;
    }
    finally
    {
    }

    if (ds.Tables[0].Rows.Count == 0)
        return null;
    return ds;
}
//פעולת אחזור של דטה סט
}

```

שרותי הרשת באתר

שירותי רשת נצרכים ממקור חיצוני המשמש כ"בנק".

באמצעותו ניתן לבדוק את פרטי כרטיס האשראי, את יתרת הכרטיס ולשלם.

<http://localhost:61116/FakebookWebService/WebService.aspx>

## CreditCard

```
public class CreditCards
{
    private string cardNumber; //ספר כרטיס אשראי
    private string type; //סוג כרטיס אשראי
    private string ownerId; //מזהה בעל הכרטיס אשראי
    private string expirationDate; //תאריך תפוגה
    private double balance; //יתרה
    private string cvv; //ספרות מאחורי הכרטיס

    public CreditCards(string cardNumber, string type, string ownerId, string expirationDate, double balance, string cvv)
    {
        this.cardNumber = cardNumber;
        this.type = type;
        this.ownerID = ownerId;
        this.expirationDate = expirationDate;
        this.balance = balance;
        this.cvv = cvv;
    }

    //מחיקה
    public static int Delete(int cardNumber, int ownerId)
    {
        string st = string.Format("Delete from CreditCards where (CardNumber='{0}') And OwnerID='{1}'", cardNumber, ownerId);
        int row = (int)DataService.ExecuteNonQuery(st);
        return row;
    }

    //הוספה
    public bool Insert()
    {
        string st = string.Format("Insert into CreditCards (CardNumber, Type, OwnerID, ExpirationDate, balance, cvv) values('{0}','{1}','{2}','{3}','{4}','{5}'", cardNumber, type, ownerId, expirationDate, balance, cvv);
        int row = (int)DataService.ExecuteNonQuery(st);
        if (row > 0)
            return true;
        return false;
    }

    //האם הכרטיס קיים
    public static bool Exist(string cardNumber, string ownerId, string type, int expirationMonth, int expirationYear, string cvv)
    {
        string expirationDate = "";
        if (expirationYear.ToString().length == 2)
            expirationDate = expirationMonth.ToString("00") + "/" + expirationYear.ToString("00");
        else
            expirationDate = expirationMonth.ToString("00") + "/" + expirationYear.ToString().Substring(2, 2);

        string st = string.Format("select * from CreditCards where (CardNumber = '{0}') AND (Type = '{1}') AND (OwnerID = '{2}') AND (ExpirationDate = '{3}') AND (cvv = '{4}')", cardNumber, type, ownerId, expirationDate, cvv);
        object obj = DataService.ExecuteScalar(st);
        if (obj == null)
            return false;
        return true;
    }

    //האם הכרטיס קיים עם סכום
    public static double getBalance(string ownerId, string cardNumber)
    {
        string strSql = string.Format("select Balance from CreditCards where (OwnerID='{0}') And CardNumber='{1}'", ownerId, cardNumber);
        object balance = DataService.ExecuteScalar(strSql);
        if (balance == null)
            return 0;
        return double.Parse(balance.ToString());
    }

    //האם יש יתרה
    public static bool update(string ownerId, string cardNumber, double amountToPay)
    {
        double amount = CreditCards.getBalance(ownerID, cardNumber);
        if (amount >= amountToPay)
        {
            string st = string.Format("update CreditCards set Balance='{0}' where (OwnerID='{1}') And CardNumber='{2}'", amount - amountToPay, ownerId, cardNumber);
            int row = (int)DataService.ExecuteNonQuery(st);
            if (row > 0)
                return true;
            return false;
        }
        return false;
    }

    //האם יש יתרה מספיק
    public static bool enoughMoney(string ownerId, string cardNumber, double amountToPay)
    {
        double amount = CreditCards.getBalance(ownerID, cardNumber);
        return amount >= amountToPay;
    }

    //האם יש יתרה מספיק
    public static bool enoughMoney(string ownerId, string cardNumber, double amountToPay)
    {
        double amount = CreditCards.getBalance(ownerID, cardNumber);
        return amount >= amountToPay;
    }
}
```

## DataService

```
public class DataService
{
    static private string path; //מסלול
    static public string Path
    {
        set { path = value; }
        get { return path; }
    }

    //מחברת / מחברת
    static public string ConnectionString
    {
        get
        {
            return @"Data Source=.\SQLEXPRESS;AttachDbFilename=" + DataService.path + @";Integrated Security=True;User Instance=True";
        }
    }

    //מחברת
    static public object ExecuteScalar(string strSql)
    {
        object obj = null;
        SqlConnection connection = new SqlConnection(DataService.ConnectionString);
        SqlCommand command = new SqlCommand(strSql, connection);
        try
        {
            connection.Open();
            obj = command.ExecuteScalar();
        }
        catch (Exception ex)
        {
            return null;
        }
        finally
        {
            connection.Close();
        }
        return obj;
    }

    //מחברת
    static public int ExecuteNonQuery(string strSql)
    {
        int rowsAffected;
        SqlConnection connection = new SqlConnection(DataService.ConnectionString);
        SqlCommand command = new SqlCommand(strSql, connection);
        try
        {
            connection.Open();
            rowsAffected = command.ExecuteNonQuery();
        }
        catch (Exception ex)
        {
            return -1;
        }
        finally
        {
            connection.Close();
        }
        return rowsAffected;
    }
}
```

## שירותי רשת שהאפליקציה מספקת

```
public class WebService : System.Web.Services.WebService {
    public WebService () {
        DataService.Path = Server.MapPath("~/App_Data/CreditCard.mdf");
    }

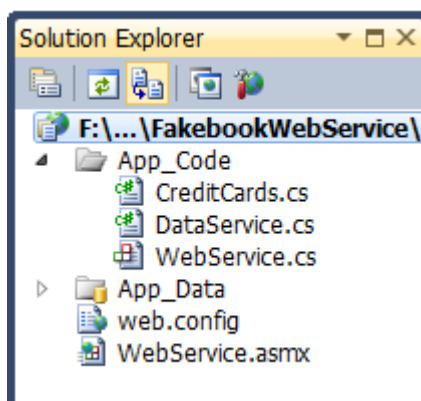
    [WebMethod]//פעולה הבדוקת תקינות של כרטיס אשראי
    public bool CheckValidCreditCardInfo(string cardNumber,string ownerID, string type, int expirationMonth, int expirationYear, string cvv)
    {
        return CreditCards.Exist(cardNumber, ownerID, type, expirationMonth, expirationYear, cvv);
    }

    [WebMethod]//פעולה הבדוקת האם לכרטיס אשראי יש יתרה מספקת בכדי לקנות
    public bool IsEnoughMoney(string ownerID, string cardNumber, double amountToPay)
    {
        return CreditCards.EnoughMoney(ownerID, cardNumber, amountToPay);
    }

    [WebMethod]//פעולה המשלמת סכום מסויים - פעולה הקונה
    public bool BuyItem(string ownerID, string cardNumber, double amountToPay)
    {
        return CreditCards.Update(ownerID, cardNumber, amountToPay);
    }
}
```

## מבנה בסיס הנתונים של שירותי הרשת

### מבנה התיקיות



תמונת הטבלאות מתוך מסד הנתונים

### CreditCards

טבלת כרטיסי האשראי המכילה מספר כרטיס אשראי, סוג כרטיס אשראי, ת.ז, תאריך התוקף, יתרה, 3 ספרות מאחורי הכרטיס.

dbo.CreditCards:...\CREDITCARD.MDF			
	Column Name	Data Type	Allow Nulls
▶	CardNumber	nvarchar(50)	<input type="checkbox"/>
	Type	nvarchar(50)	<input type="checkbox"/>
▶	OwnerID	nvarchar(50)	<input type="checkbox"/>
	ExpirationDate	nvarchar(50)	<input type="checkbox"/>
	Balance	real	<input type="checkbox"/>
	cvv	nvarchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

### **ReadMe**

יש להעתיק את התיקייה "פרויקט" למחשבך , לאחר מכן יש להיכנס לתוכנה  
"Microsoft Visual Web Developer".  
לחץ על "Open Website" וסמן את התיקייה של הפרויקט.  
כנס לדף "FirstPage.aspx" ולבסוף Control F5.

לצורך כניסה בתור משתמש (רגיל) רשום באתר יש למלא את הפרטים הבאים בLogin :  
שם משתמש : normal@gmail.com  
סיסמה : test123

לצורך כניסה בתור מנהל האתר יש למלא את הפרטים הבאים בLogin :  
שם משתמש : admin@gmail.com  
סיסמה : admin123