תוכן עניינים

[מטרת השלב המקדים:](#_gjdgxs) **2**

[הנחיות כלליות](#_30j0zll) **2**

[המטלה – קווים כלליים](#_1fob9te) **2**

[הוראות פרטניות](#_3znysh7) **3**

[פתיחת חשבון ב-github.com](#_2et92p0) 3

[פתיחת מאגר חדש](#_tyjcwt) 5

[שיתוף בן הזוג במאגר](#_1t3h5sf) 6

[יצירת אסימון גישה אישי](#_fm5v43uest65) 8

[יצירת עותק מקומי של המאגר](#_ydfd42xkxqm2) 11

[יצירת פתרון בתוך המאגר המקומי](#_17dp8vu) 13

[יצירת פרויקט](#_3rdcrjn) 13

[שימוש ב-Snippets](#_26in1rg) 14

[כתיבת קוד קצר](#_lnxbz9) 14

[העברת הקוד לפונקציה באמצעות Refactor](#_35nkun2) 15

[עדכון וסנכרון המאגרים](#_1ksv4uv) 17

[אישור השינויים (ע"י השותף הראשון)](#_2jxsxqh) 17

[עדכון המאגר הראשי (ע"י השותף הראשון)](#_3j2qqm3) 18

[עדכון המאגר המקומי אצל השותף השני (ע"י השותף השני)](#_1y810tw) 19

[עדכון וסנכרון המאגר של השותף השני (ע"י השותף השני)](#_2xcytpi) 20

[סנכרון המאגר של השותף הראשון (ע"י השותף הראשון)](#_1ci93xb) 20

[פעולות נוספות לתרגול עבודה בזוגות](#_3whwml4) 21

[עבודה באותו קובץ, בשורות שונות](#_2bn6wsx) 21

[עבודה באותן שורות](#_qsh70q) 22

[הגשה](#_3as4poj) 24

[יצירת תג (Tag)](#_1pxezwc) 24

*נ.ב.1 המסמך נכתב בלשון רבים או בלשון זכר אך מתייחס לסטודנטים ולסטודנטיות כאחד.*

*נ.ב.ב בחלק מהדוגמאות בילומי המסך עשויה להופיע שנה 5782 – יש להתעלם ולהקפיד על השנה הנכונה 5783.*

# מטרת השלב המקדים:

מטרת השלב המקדים הינה הכרות ראשונית עם:

* שפת **C#**
* סביבת הפיתוח **Visual Studio 2022**
* מערכת ניהול גרסאות **git**
* שירות ענן של הארחת מערכת ניהול גרסאות מבוזרת **github.com**

רמת השלב – קל מאד

# הנחיות כלליות

* ההגשה בזוגות בלבד (**חשוב**: הקפידו כבר בשלב המקדים לעבוד עם שותף שאתו תבצעו את יתר הפרויקט בקורס.
* עבודה על השלב יכולה להיעשות רק לאחר ששני השותפים לזוג ביצעו את הנחיות התקנה\עדכון של Visual Studio 2022, גם הסטודנטים שחושבים שהתכנה הזו כבר מותקנת אצלם – חובה לבצע את העדכון ע"פ קובץ ההנחיות שבאתר הקורס.

**נ.ב. אירעו כבר מקרים כאשר סטודנטים נתקעו באמצע השלב בגלל אי ביצוע העדכון**.

* הגדרת ההרשאה של המאגר שייפתח ב-**github.com** בעבור השלב (ויתר השלבים של הפרויקט) חייבת להיות כ**"פרטית**", החזקת המאגר עם הרשאה ציבורית אפילו לפרק זמן קצר תמנע קבלת ציון לשלב ובכך גם לקורס.
* יש לסנכרן את המאגרים המקומיים במחשבים של שני השותפים לזוג עם המאגר ב-**github.com**.
* יש להוסיף את המרצה ואת הבודק (שמות משתמש ב-**github.com** יינתנו ע"י המרצה) כשותפים לפרויקט ב-**github.com** כבר בתחילת העבודה.
* אסור לבצע שינויים בפרויקט של השלב ב-**github.com** לאחר הגשתו. שינויים כאלה ימנעו קבלת ציון לשלב.
* כל השלבים יבוצעו כפרויקטים באותו פתרון (**Solution**) ובאותו מאגר. שם ה-**Solution** והמאגר יהיה זהה והפורמט שלו יהיה: **dotNet5783\_ xxxx\_yyyy**, כאשר:
* **dotNet5783** – שמו המקוצר של הקורס ומספר השנה העברית בספרות לועזיות
* **xxxx** ו-**yyyy** – 4 ספרות אחרונות של תעודת הזהות של שני הסטודנטים
* שם הפרויקט עבור השלב יהיה **Stage0**.

**נ.ב.** אי הקפדה על פורמטים אלה תמנע קבלת ציון לשלב או לכל השלבים בהתאם.

* הגשת השלבים תתבצע ע"י הזנת הקישור לתגית במאגר (**Tag**) ב-**github.com** שנוצר ע"י הסטודנטים במאגר המקומי עבור ההגשה ונדחף למאגר המשותף בענן.

# המטלה – קווים כלליים

1. יצירת חשבון אישי ב-**github.com** ע"י שני הסטודנטים
2. יצירת מאגר **פרטי** עם שם לפי הפורמט ב-**github.com**
3. שיתוף המאגר עם השותף, עם המרצה ועם הבודק
4. יצירת **Solution** ב-**Visual Studio 2022** עם פרויקט עבור שלב 0 בשמות לפי הפורמט, וחיבורו למאגר – במחשבים של שני הסטודנטים
5. סנכרון ה-**Solution** של שני השותפים לזוג עם המאגר ב-**github.com**
6. עבודה על קובץ **README.md**:
   1. הקובץ מגיע עם מספר שורות.
   2. שני השותפים לזוג מוסיפים \ עורכים שורות שונות ומסנכרנים ביניהם.
   3. שני השותפים לזוג עורכים באותה שורה ואחד ממזג.
   4. שני השותפים לזוג עורכים באותה שורה, והשני ממזג.
7. הגשה

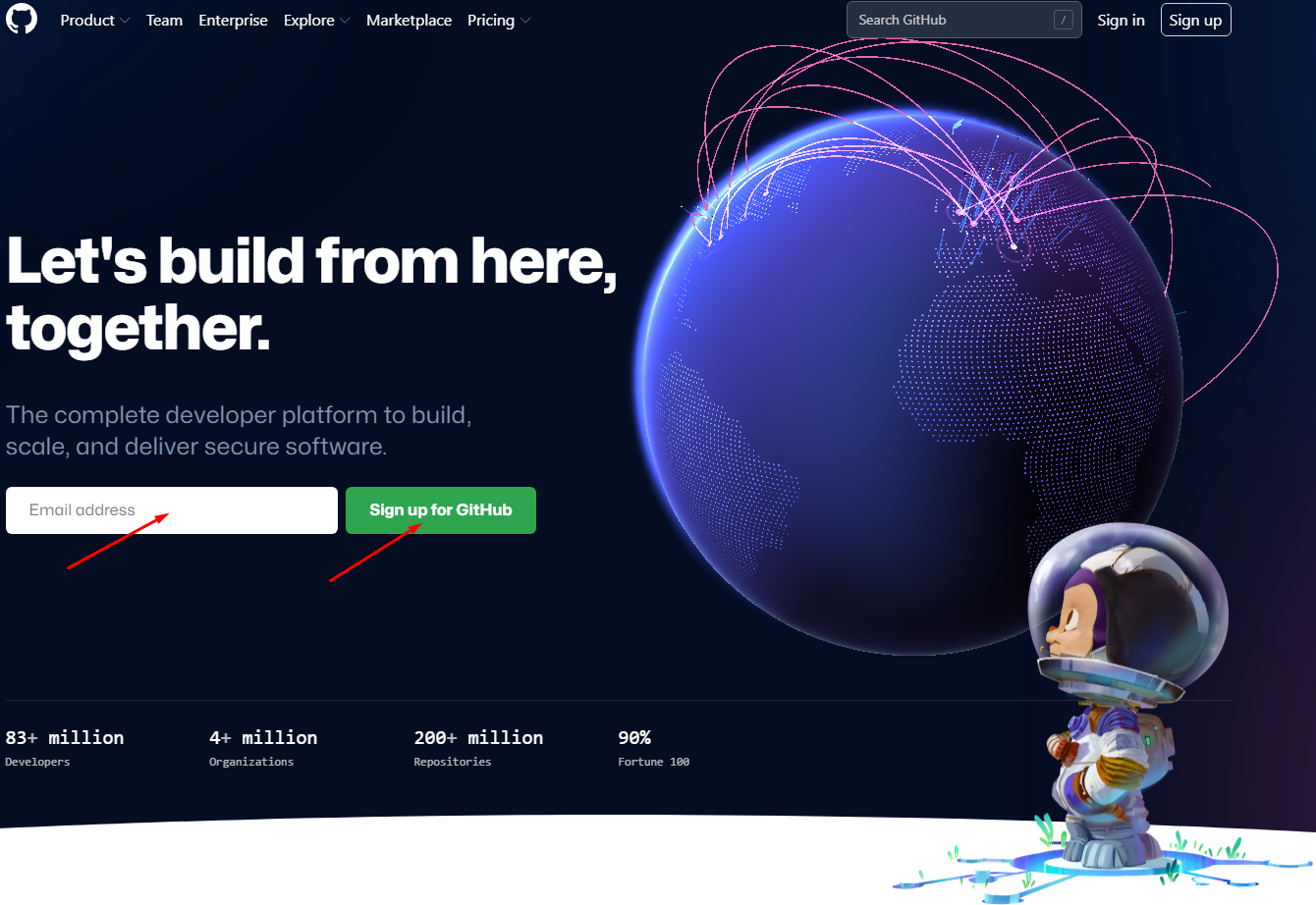
שימו לב: עבור כל שלב, יסומן במרקר צהוב מי מהשותפים לזוג אמור לבצע אותו.

ההוראות כתובות בלשון רבים והן פונות לזכר או נקבה כאחד.

# הוראות פרטניות

## פתיחת חשבון ב-**github.com**

פרק זה יתבצע בנפרד על ידי **כל אחד** משני השותפים לזוג

1. אם יש לכם כבר חשבון ב-**github.com**, ניתן לדלג לפרק הבא.
2. היכנסו לאתר בכתובת [**https://github.com**](https://github.com/)
3. דף הבית של האתר נראה כך (נכון לאלול תשפ"ב):
4. הכניסו את כתובת דוא"ל בדף הנ"ל (אם תשתמשו בכתובת דוא"ל שלכם מהמכון – עם הדומיין **g.jct.ac.il**, אפשר לקבל הטבות לסטודנטים) ותלחצו על הכפתור הירוק **Sign up for GitHub** 

Graphical user interface, text

Description automatically generatedבמסך הבא לחצו על הכפתור Continue, ובחרו סיסמא לפי ההוראות, עוד פעם Continue, בחרו שם המשתמש שלכם בשירות **github.com** ולחצו עוד פעם Continue. בשאלה הבאה תקלידו "y" אם תרצו לקבל עדכונים ההודעות מהאתר לדוא"ל שלכם או "" אם לאו, ועוד פעם Continue.

1. בהמשך, צריך לפתור פאזל (נכון לתשרי תשפ"ב), וללחוץ על כפתור Create account.
2. Graphical user interface, text, application, website

   Description automatically generatedלאחר מכן תקבלו הודעה לדוא"ל עם קוד אמות – תקלידו את הקוד שהתקבל במסך הבא.
3. בשלב הבא, מוסרים קצת מידע על הידע שלכם והמטרות לפתיחת החשבון. בדרך כלל תרצו לבחור\לסמן את Graphical user interface, text, application

   Description automatically generatedהמשבצות הבאות: Diagram

   Description automatically generated

ולבסוף תלחצו על הכפתור  **Apply for your GitHub student benefits**

A screenshot of a computer

Description automatically generated with medium confidence

1. Diagram

   Description automatically generatedבזמנכם החופשי ולפי רצונכם תוכלו לסייר בדפי **github.com** עבור סטודנטים (**education.github.com**) ולהשתמש בכלים המוצעים שם. לא כל כך רלוונטי לפרויקט שלנו. בינתיים חזרו לאתר הראשי **github.com** אבל הפעם לחצו על **Sign in**:
2. Graphical user interface, application, website

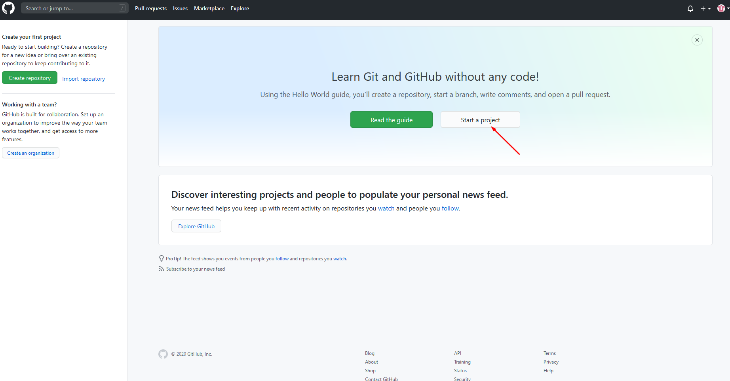
   Description automatically generatedהקלידו את שם המשתמש והסיסמה שבחרתם קודם:
3. מגיעים לדף שבו מציעים ליצור מאגר (**Repository**) חדש. נמשיך לפרק הבא.

## פתיחת מאגר חדש

פרק זה יתבצע רק על ידי **אחד** מהשותפים לזוג - בפרק 3 משתפים את המאגר עם השותף השני

**הבהרה**: במסמך זה השתמשנו במילה "מאגר" שמכונה גם ריפו, ריפוזיטורי, **Repo**, **Repository**.

על מנת לחסוך עבודה חוזרת על השלב הזה, המדריך הזה מיועד ליצירת מאגר בודד לכל הקורס, כאשר המאגר יכיל את כל השלבים. כמובן, **לאחר סיום** הקורס וקבלת ציון, ניתן לפתוח מאגר נוסף ולהעלות אליו רק את הפרויקט הסופי, להפוך אותו לציבורי, ולהוסיף קישור בקו"ח בחלק שמתאר את הפרויקט, או פשוט להפוך את הרשאת המאגר לציבורית.

1. תיכנסו לאתר בכתובת [**https://github.com**](https://github.com) עם החשבון שפתחתם בפרק 1,(במידה ואתם כבר באתר – תוכלו לדלג לסעיף 3)
2. הדף הראשי אחרי שנכנסים נראה כך: 
3. לחצו על כפתור  **Start a project**  או  **Create Repository** . אפשר גם ללחוץ על הכפתור הירוק ולקרוא את המדריך הבסיסי.
4. בחלון הבא תיצרו את המאגר הראשון. לפני היצירה הוא נראה ככה: פרטים שצריך לשים לב אליהם: Graphical user interface, text, application, email

   Description automatically generated
   1. הקלידו את שם המאגר לפי הדרישות כנ"ל.
   2. תיאור (**Description**) – רצוי לרשום, אפשר לדלג.
   3. המאגר **חייב** להיות מוגדר **פרטי (Private)**. במקרה של העתקה ממאגר ציבורי, גם בעל המאגר הציבורי ייחשב כאשם (לא עכברא גנב אלא חורא גנב...).
   4. סמנו את התיבה ליד "**Add a README file**".
   5. סמנו את התיבה ליד "**Add .gitignore**". בתיבת תפריט נפתחת שתופיע לאחר הסימון בחרו ב-**Visual Studio**. בדרך הזו, גיט לא ישמור במאגר את כל הקבצים שהפנימיים של VS שהוא מייצר לצורך הניהול הפנימי של הפרויקט, אלא רק את קבצי הקוד הנצרכים.
   6. **אל** תסמנו את התיבה ליד **Choose a license** (בחר רישיון).
5. לחצו על הכפתור הירוק  **Create repository** .

## שיתוף בן הזוג במאגר

פרק זה יתבצע על ידי **אותו** הסטודנט שביצע את פרק 2

אחרי פרק 2, יש לנו מאגר פרטי, ואנחנו רוצים לעבוד עליו עם עוד מישהו. **לצורך העניין נניח שהוא כבר פתח לעצמו חשבון ב-github.com** (לפי ההוראות בפרק 1).

1. כנסו למאגר באחת משתי הדרכים:
   1. כנסו לעמוד הראשי של האתר, בצד שמאל יש רשימת הקישורים למאגרים שלך. לחצו על הקישור המתאים.
   2. הקלידו בשורת הכתובת של הדפדפן את הכתובת בצורה הבאה:

**https://github.com/***username***/***repo-name* כאשר נרשם שם חשבונך במקום *username* ושם המאגר במקום *repo-name*.

1. בחרו בכרטיסיית **Settings** בסרגל הכרטיסיות.
2. Graphical user interface, application, email, website

   Description automatically generatedGraphical user interface, text, application, email

   Description automatically generatedבחרו "**Manage Access**" בצד שמאל, ואז לחצו על הכפתור הירוק  **Invite a collaborator**  (הזמן משת"פ):
3. Graphical user interface, application, Word

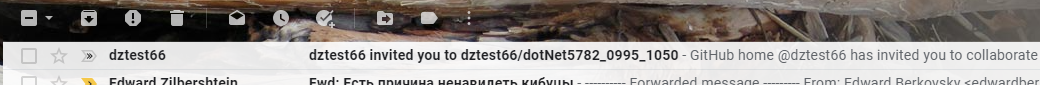
   Description automatically generatedGraphical user interface, application

   Description automatically generatedרשמו את שם המשתמש או השם או מספר טלפון של שותפכם לזוג (ובהמשך גם של המרצה, ושל הבודק), ולחצו על הכפתור הירוק  **Add** *user* **to this repository**  .
4. בשלב הזה, **github.com** אמורים לשלוח קישור של ההזמנה לדוא"ל של המשתמש שהוזמן.

Graphical user interface, application

Description automatically generated

1. Graphical user interface, application

   Description automatically generatedהשותף השני יפתח את הודעת ההזמנה בדוא"ל שלו, ילחץ על הקישור בהזמנה, ויאשר את ההזמנה. 

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application

Description automatically generatedלאחר האישור, בדף ניהול הגישה של השותף הראשון יופיע (לאחר רענון הדף):

1. במקרה של תקלה בקבלת הדוא"ל - עד שהמשתמש שהוזמן יאשר את ההזמנה, ההזמנה מחכה, וניתן להעתיק את "קישור לאישור ההזמנה", ולשלוח אותה עצמאית בכל אמצעי אחר למשתמש שהוזמן.
2. מהרגע שההזמנה אושרה, שני השותפים לזוג יכולים לעבוד מול המאגר ישירות. תדאגו לוודא ששותפכם מאשר את ההזמנה ואם לאו, זה סימן שייתכן והוא לא מעניין להשתתף בקורס וכדאי לכם לבדוק אפשרות להחליפו. אם ההזמנה אינה מאושרת תוך זמן סביר, ההזמנה יכולה להפוך ללא תקינה (בלחיצת אישור ההזמנה מקבלים תשובה שההזמנה לא נמצאה). במצב הזה יהיה צורך שהמזמין ימחק את ההזמנה הישנה ויזמין את השותף מחדש.
3. כעת, הזמינו באותה הדרך את המרצה ואת הבודק להיות שותפים למאגר שלכם.

**סיכום ההוראות עד לשלב זה**: יש לנו שני חשבונות ב-**github.com**, מאגר בודד עם תוכן בסיסי (קובץ **README.md** והגדרות שמתאימות לעבודה עם **VS**), משני החשבונות יש גישה למאגר (לא משנה מי מהשותפים מחזיק את המאגר, מלבד היכולת "לזרוק" את השני על ידי ביטול ההזמנה).

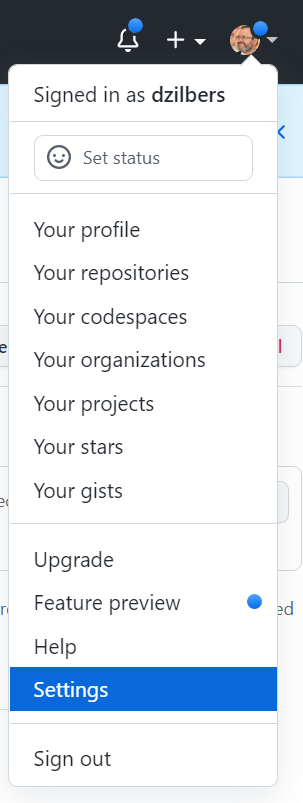
על מנת לאפשר בדיקה של השלבים, יש לשתף גם את מרצה וגם את בודק הקבוצה בה אתם לומדים.

## יצירת אסימון גישה אישי

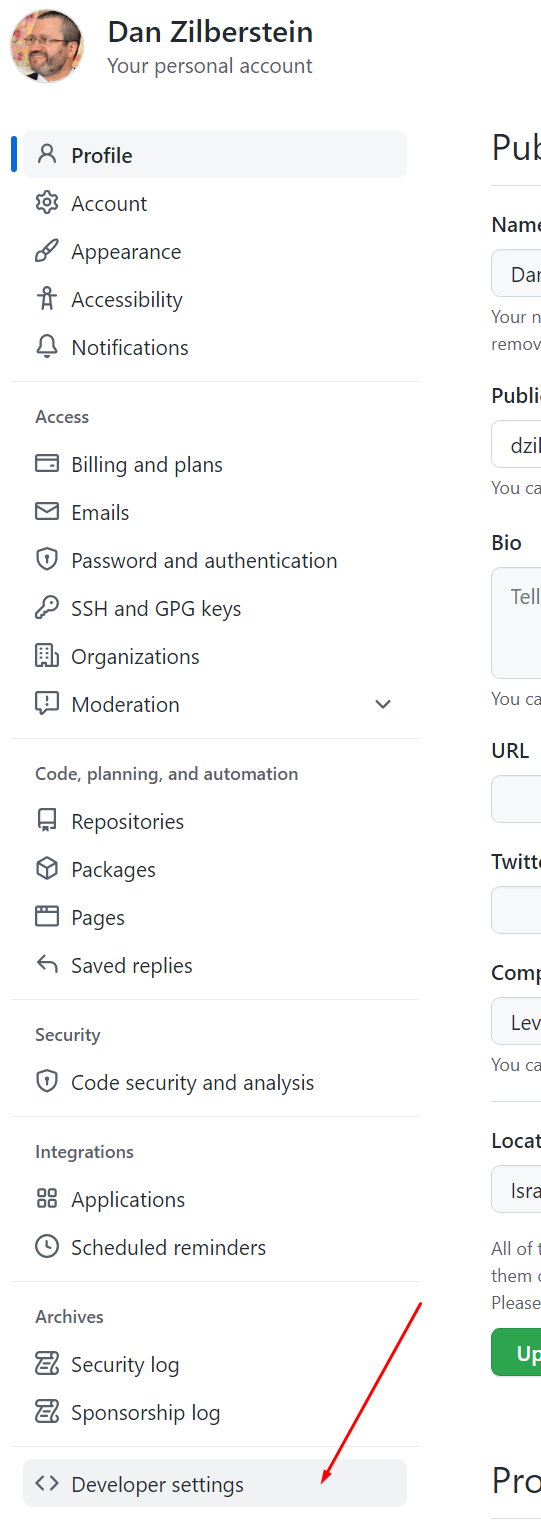
פרק זה יתבצע על ידי **שני** השותפים.

מיקרוסופט הקשיחו את האבטחה בשירותי הענן של github.com. התחברות בעזרת https עבור פקודות git למאגר מרוחק, הסיסמה הרגילה של המשתמש שעבדה עד סוף 2021 - לא עובדת יותר. מעתה, על מנת לבצע חיבור של המאגר המקומי (או סביבת פיתוח, כמו ויז'ואל סטודיו) למאגר המרוחק ב-github.com, במקום סיסמא הרגילה תצטרכו להשתמש באסימון אישי (personal token). שימו לב שניתן לראות ולעשות העתק (על מנת לעשות בהמשך הדבק) של האסימון האישי - רק פעם אחת. לא תוכלו לשחזר אותו - אלא ליצור אותו מחדש או להוסיף עוד אסימון חדש. כל סטודנט יצטרך ליצור אסימון אישי בחשבון שלו ולשמור אותו בצד (שליחת הודעה (לעצמו\ה) הכוללת את האסימון בדוא"ל או בדרך אחרת.

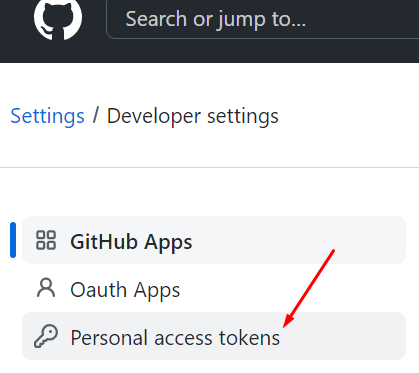
1. כנסו לחשבון שלכם ב-github.com.
2. לחצו על תמונתכם בפינה העליונה בצדו הימני של דף האתר ובתפריט שנפחת לבחור "Settings":



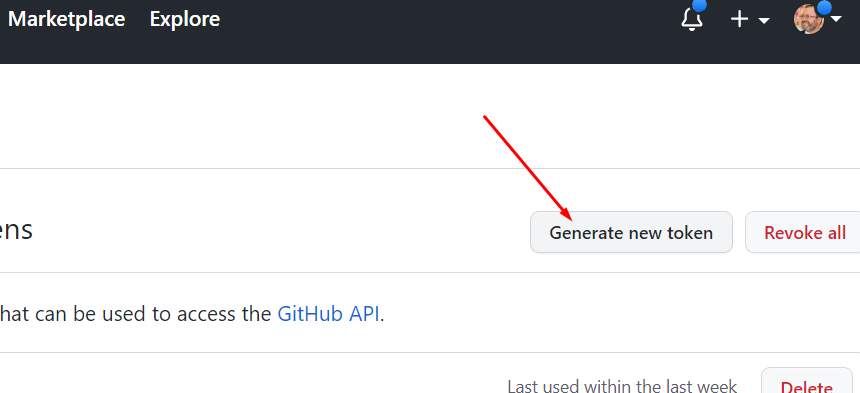
1. בדף שייפתח, רדו (גלגלו) למטה עד שתראו בסוף הרשימה של אפשרויות בצדו השמאלי של הדף "**Developer settings<>**" ולבחור את האפשרות הזו (ללחוץ עליה):



1. בדף שייפתח, בחרו בצדו השמאלי את האפשרות "Personal access tokens":

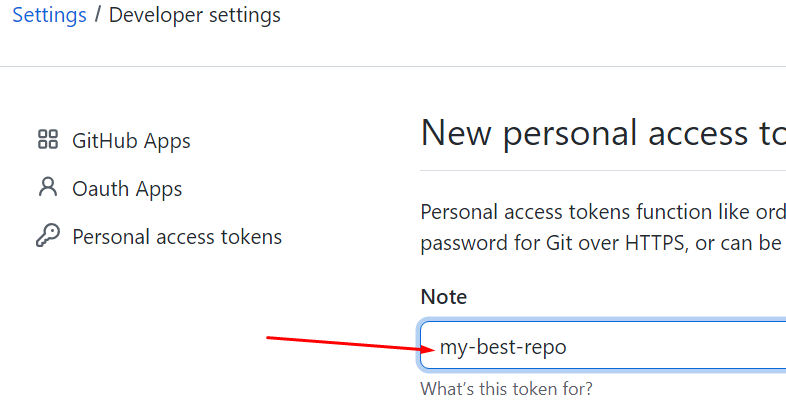


1. בדף שייפתח, ללחוץ על הכפתור "**Generate new token**":

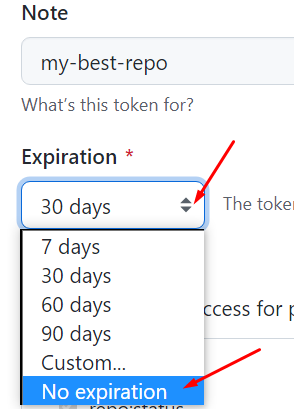


1. בדף שייפתח:

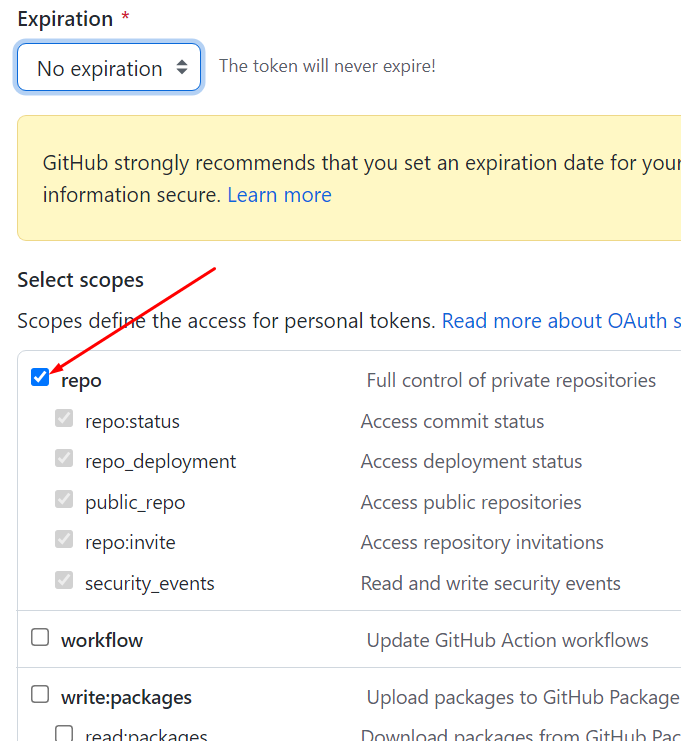
* רשמו הערה (הכי טוב - שם סביבת הפיתוח או שם המאגר שלכם) בתיבה שמתחת ל"Notes":



* שנו את ה-"**Expiration**" כך שיספיק עד לאחר הסמסטר או בכלל שנו ל-"No expiration":

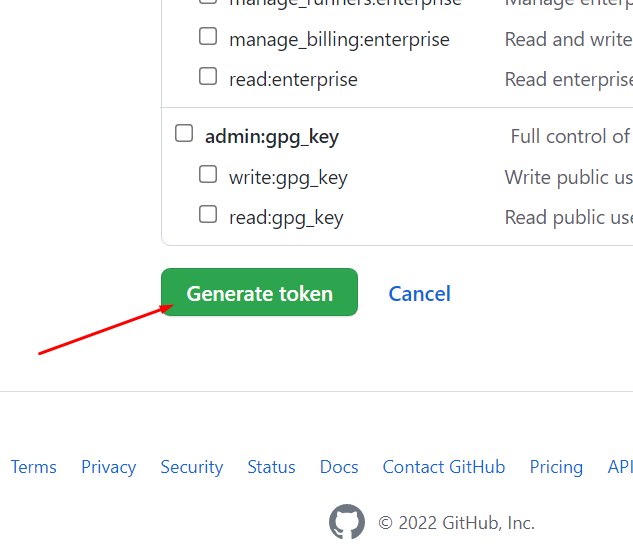


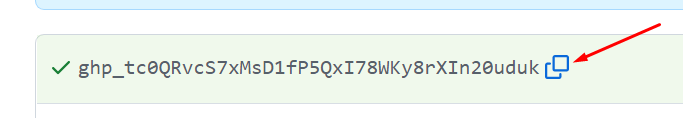
* סמנו "**V**" בתיבת סימון ליד "repo" (ברגע הסימון יסומנו גם אוטומטית כמה תיבות סימון המוזחות שמתחתיו), ואם אתם מתכוונים לעבוד עם כלים נוספים שדורשים הרשאות נוספות (במקרה שאתם מכירים אותם) - סמנו גם בתיבות הסימון המתאימות:



**נ.ב.** מומלץ לסמן גם את התיבות הבאות: **project**, **user**, **notifications**, **gist**, **workflow**

* רדו (גלגלו) למטה ולחצו על כפתור ירוק "**Generate token**":



* בדף שייפתח תראו את האסימון האישי שלכם בשורה עם רקע ירקרק, סימן "**V**" משמאלו וסימן של שני ריבועים אחד מעל השני לימינו (למשל:). הטקסט הוא בעצם האסימון האישי שאתם צריכים. כרגע לחצו על סימן שני הריבועים (או לחלופין - תסמנו עם העכבר את הטקסט של האסימון ותעשו פעולת "העתק" כהרגלכם - רק **שימו לב** לא לסמן רווחים מיותרים מסביב לטקסט).
* הדביקו את האסימון לתוך הודעה לעצמכם בדוא"ל או בואטסאפ או בטלגרם, או לחילופין בקובץ שאתם שומרים את הנתונים שלכם לשימוש מאוחר יותר. משם "תקחו" את האסימון בהמשך כשתזדקקו לו.

## 

## יצירת עותק מקומי של המאגר

פרק זה יתבצע בנפרד **(ובמקביל)** על ידי **כל אחד** מהשותפים לזוג

המטרה של השלב הזה, היא ליצור בתוך המחשב של הסטודנט מאגר מקומי, שהתוכן שלו יהיה זהה לתוכן המאגר ב-**github.com**, כאשר מפעם לפעם נדאג לבצע עדכון של התוכן בין המאגרים.

בהמשך עבור כל שלב בהתאם, נדאג **כל פעם שאנחנו יוצרים פרויקט ליצור אותו באותו הפתרון (Solution)** שנפתח כעת ונקשור אותו למאגר שפתחנו קודם. לכן כשנעדכן את המאגר מול **github.com** – **נמשוך** (**pull**) מ- **github.com** את השינויים שהשותף שלנו לזוג עשה, **ונדחוף** (**push**) ל-**github.com** את השינויים שעשינו.

המדריך כאן נכתב עבור **VS2022**, אחרי שאותחלו בו הגדרות ברירת מחדל עבור **#C**.

**שים לב שיש להשתמש בגרסה האחרונה ביותר שהותקנה לפי ההנחיות באתר הקורס**, וייתכן שלצורך כך תצטרך לעדכן את גרסת **VS** שיש בידך. להנחיות להתקנה ועדכון ראה במסמך ההנחיות שנמצא באתר הקורס.

1. היכנסו בדפדפן לאתר **github.com** ופתח את המאגר הרצוי.
2. Graphical user interface, text, application, email

   Description automatically generatedלחצו על הכפתור הירוק  **Code v** :

Graphical user interface, text, application, chat or text message, email

Description automatically generated

1. תופיע כתובת, אפשר לסמן אותה ולהעתיק, או פשוט ללחוץ על הכפתור "העתק":

הכתובת בדרך כלל תהיה בפורמט **https://github.com/***username***/***repo-name***.git**.

Graphical user interface, application, Word

Description automatically generated**עכשיו נעבור לויז'ואל סטודיו**.

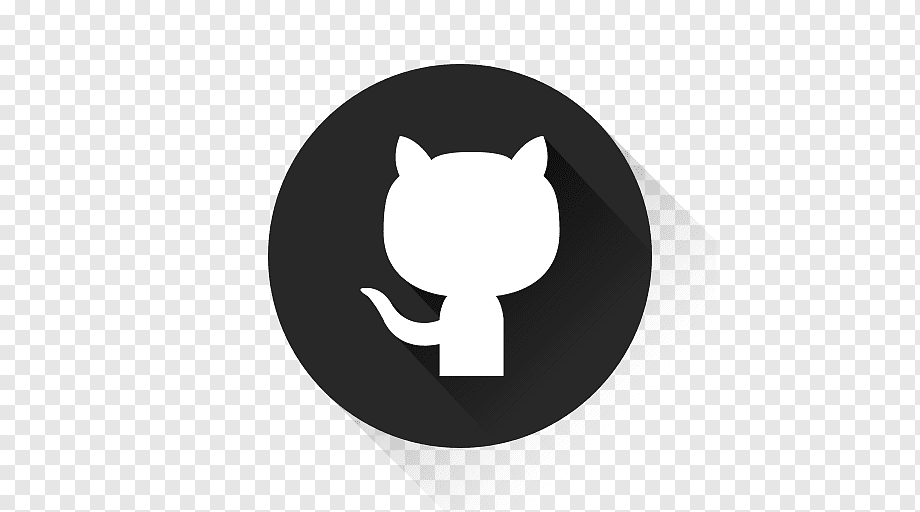
1. פתחו (הריצו) את תכנת ה-**Visual Studio**.
2. בחלון הפתיחה, ישנן מספר אפשרויות. בחרו באפשרות לשבט (להעתיק) מאגר קיים, בצד ימין למעלה:

1. בחלון הבא, כתבו את הכתובת המאגר, ואיפה במחשב אנחנו רוצים את העותק שלו.

**שימו לב**: הכתובת ב-**local path** צריכה להיות תיקייה שעוד לא קיימת – כדאי שתהיה במקום שאתם מרכזים את כל חומרי הקורס שלכם. כדאי שלא תהיה עברית בנתיב (**path**) של התיקיה. שם החלק האחרון של תיקיית המאגר יהיה כשם המאגר. את הקישור למאגר נעתיק ע"י הדבק לאחר שעשינו העתק בשלב הקודם.

**נ.ב.1** תיקיית המאגר **צריכה להיות** תחת תת-תיקיה **Sources/Repos/** שבתיקיית המשתמש – יחד עם שם החשבון ב-**github.com**. **בדרך כלל הנתיב המלא לתיקיה יהיה כדלקמן (שם המאגר רק פעם אחת בנתיב):**

**C:\Users\***user-name***\Source\Repos\***account-name***\dotNet5783\_***xxxx***\_***yyyy*

**נ.ב.2** ניתן ללחוץ על הכפתור  **GitHub** למטה ולבחור ישירות את מאגר המקור (תצטרכו לבצע כניסה לחשבון מתוך VS), לאחר בחירת המאגר לחצו על הכפתור Clone : Graphical user interface, text, application, email

Description automatically generated

1. אם לא בחרתם לפתוח חלון בחירת מאגרים מחשבון רשום) בנתיב (**Path**) תבדקו שמופיע שם החשבון שבו נפתח המאגר ב-**github.com** והוסיפוהו אם חסר (תוסיפו גם סלאש הפוך "\") ולאחר מכן לחצו על הכפתור Clone :Graphical user interface, text, application, email

   Description automatically generated

**נ.ב.** בפעם הראשונה תצטרכו לתת שם משתמש וסיסמא לאתר github.com. **עבור הסיסמא השתמשו ב**[**אסימון האישי**](#_fm5v43uest65) שיצרתם קודם.

1. Graphical user interface, text, application, email

   Description automatically generatedייפתח חלון של **VS**, אבל בגלל שהמאגר לא מכיל פתרון של **VS**, הוא יהיה כמעט ריק.
2. בחר לעדכן את השם ואת כתובת הדוא"ל שלכם עבור רישום ב-**commit** בהמשך:
3. Graphical user interface, application

   Description automatically generatedGraphical user interface, text, application, email

   Description automatically generatedבפעם הראשונה גם מומלץ לעבור על הגדרות **git** של **VS** ולעדכן איפה שצריך (תפריט **Tools** ושם **Options**, בחלון שנפתח תרחיבו בצד שמאל את **Source Control**). למשל:
4. תעברו ללשונית **Solution Explorer** ותעברו לתצוגת תיקייה (**Folder View**), תראו את התוכן הבא. שימו לב על שני הקבצים המופיעים שם – **README.md** ו-**.gitignore** שנבנו בגיטהאב כשבנינו את המאגר שם.
5. **Graphical user interface, text, application, email

   Description automatically generatedבחרו בתפריט הראשית File => Close folder כדי לסגור את התיקיה ולחזור למסך הפתיחה של VS.** Graphical user interface, text, application, email

   Description automatically generatedGraphical user interface, text, application, email

   Description automatically generated

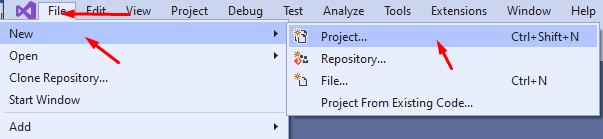
## יצירת פתרון בתוך המאגר המקומי

פרק זה יבוצע על ידי **אחד (בלבד)** מהשותפים לזוג (נקרא לו **השותף הראשון**) במחשבו האישי, רצוי שבזמן ביצוע הפרק שני השותפים יהיו מחוברים ב-zoom או יישבו ביחחד כאשר השותף הראשון משתף את המסך ונותן לשותפו זמן להכיל את הנעשה.

חובה שלשני השותפים לזוג תהיה אותה גירסה של **Visual Studio**.

### יצירת פרויקט

צרו פרויקט חדש (**Create a new Project** או בתפריט **File=>New=>Project…**)

Graphical user interface, application

Description automatically generatedובחרו בניית פרויקט מסוג מסוג **Console Application**:

**Graphical user interface, application, email

Description automatically generatedנ.ב.** בשוםפנים ואופן **לא** לבחור סוג **Console Application (.NET Framework).**

קנפגו את הפרויקט כדלקמן, **לפי הסדר כפי שמופיע כאן – קודם כל שם הפרויקט ורק לאחר מכן שם הפתרון!**

1. שם הפתרון (**Solution**) יהיה **בדיוק** כשם המאגר: **dotNet5783\_xxxx\_yyyy** (אפשר להעתיק משם תיקיית המאגר בסייר הקבצים, למשל)
2. שם הפרויקט יהיה S**tage0**
3. המיקום (**Location**) בתיקיית אב של המאגר –לעדכן בשדה! (ניתן לעבור לתיקיה בלחיצה על הכפתור "**...**".
4. **חשוב!!** **אין** לסמן בתיבת סימון **Place solution and project in the same directory**!

**Graphical user interface, application, email

Description automatically generatedשימו לב שתיקיית המאגר המקומי המקורית היא גם תהיה תיקיית הפתרון ופרויקט ימוקם בתוך תיקיית המאגר!**

**נ.ב.1** בכל מקרה וודאו ששם התיקיה ב-**Location** יהיה שם תיקיית האב שכוללת בה את תיקיית המאגר המקומי שיצרתם בפרק 4 (הפרק הקודם). אי הקפדה על הכלל הזה תיצור בעיה שתדרוש מחיקה וביצוע הפרק מחדש.

לחצו על כפתור **Next**. בחלון הבא וודאו ש-**Target Framework** מוגדר כ-**(NET 6.0 (Current.** ולחצו על כפתור **Create**.

**נ.ב.2** אם קיבלתם הודעת שגיאה לאחר לחיצה על הכפתור **Create** (כן – גם למיקרוסופט יש באגים) – סגרו את **Visual Studio** והפעילו אותו מחדש תוך שאתם בוחרים באפשרות **Create a new project** בחלון הראשון שנפתח (החלון הזה מודגם בתחילת הפרק הזה).

### שימוש ב-**Snippets**

בתוכנית הראשית (שנפתחה אוטומטית) ,הקלידו **cw**Graphical user interface, text, application

Description automatically generated

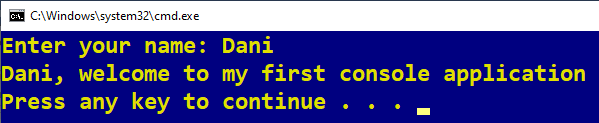
ציור של דף חתוך (משמאל לשורה הראשונה בתפריט שנפתח) אומר לנו שמדובר ב-**snippet**[[1]](#footnote-1) ויש כאן השלמה אוטומטית לאחר לחיצה על **Tab** (יש לצאת מתפריט הקשר ואז ללחוץ **Tab** או פשוט ללחוץ פעמיים **Tab**). לאחר לחיצה על **Tab** נקבל את ההשלמה להדפסה למסך.

### כתיבת קוד קצר

נדפיס למסך את המחרוזת (אפשר למחוק את ה-"***Line***" מפקודת הפלט על מנת לבצע פלט באותה שורה ע"י הפעולה Console.Write(…)):

נקלוט מהמשתמש את שמו בעזרת הפעולה **Console.ReadLine()** ונדפיס את ההודעה הבאה בהתאם:

(כמובן, עלייך להגדיר משתנה, להדפיס באמצעות שימוש ב-**{0}** כך שבהדפסה יופיע השם שהמשתמש הכניס וכו', אנו מניחים שכבר למדת מעט תכנות...)

על מנת שהתוכנית לא תיסגר אלא רק לאחר שנלחץ על מקש כלשהו לסיום בסוף הפעולה הראשית אפשר להוסיף:

לחילופין, ניתן להריץ ע"י לחיצה למקשים **Ctrl-F5** (במקום רק **F5**). לאחר ההרצה ומילוי השם החלון יראה כך:

### העברת הקוד לפונקציה באמצעות **Refactor**

כעת נרצה שהקוד לא יהיה סתם בתוכנית הראשית אלא בתוך פונקציה נפרדת שנקרא לה מהתוכנית הראשית. נוכל להשתמש ב-refactor על מנת להעביר שורות בקוד לפונקציה נפרדת. נסמן את שלושת שורות הקוד שכתבנו(ללא **()Console.ReadKey**) ובתפריט העריכה נבחר תת-תפריט של **Refactor** ואז **Extract Method**: Graphical user interface, text, application

Description automatically generated

הקוד שסימננו יועבר לפונקציה בשם **NewMethod** – השם יסומן ברקע ייחודי וייפתח חלון קטן לשינוי שם: Graphical user interface, text, application, email

Description automatically generated

נשנה את השם המסומן בתוך הקוד ל-**WelcomeXXXX** (במקום **XXXX** כמובן, ארבע הספרות האחרונות של מספר ת"ז של השותף הראשון) ונלחץ על **Apply**. כעת הקוד יראה כך: Graphical user interface, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

שימו לב שנוספה לנו פונקציה למחלקה **Program** וקריאה לה מהתוכנית הראשית שבמחלקה **Program**. שימו לב לתיבת הבחירה למעלה שניתן לדלג באמצעותה בקלות לקוד של כל פונקציה במחלקה.

**שאלה: מדוע הפונקציה הוגדרה כ-static ?** (לא צריך לענות בהגשה אבל חשוב לדעת ולהבין את התשובה)

כעת, על מנת שבשעת שיתוף המאגר לא ניתקל בשלב הזה בבעיות בסנכרון בין שני השותפים, נשנה את הגדרת המחלקה למחלקה חלקית (בהמשך הקורס תלמדו עליה יותר לעומק). חשוב! הסינכרון יעבוד היטב גם במחלקה רגילה, ולא רק במחלקה חלקית. כך נעבוד ברוב המקרים בהמשך. אך כרגע לצורך הקלה על הלימוד נשתמש ברעיון החלקיות. לכן:

1. השותף הראשון יוסיף מילת מפתח **partial** לפני הגדרת המחלקה.
2. וכן יוסיף הגדרת פונקציה חלקית ללא מימוש בשם **WellcomeYYYY**, כאשר **YYYY** אלו ארבע הספרות האחרונות של מספר ת"ז של השותף השני.
3. וכן יפעיל את הפונקציה בתוכנית הראשית: A picture containing timeline

   Description automatically generated
4. בהמשך לכך השותף הראשון ישנה את שם המודול (הקובץ) ל-**ProgramXXXX.cs** (**XXXX** – ארבע הספרות האחרונות של השותף הראשון): Graphical user interface, application

   Description automatically generated
5. חשוב! בחלון הקופץ של אישור עדכון רוחבי יש לבחור באפשרות **No** (זה חשוב על מנת לא לשנות את שם המחלקה): Graphical user interface, application, Word

   Description automatically generated

## עדכון וסנכרון המאגרים

**אין להתחיל בביצוע הפרק הזה לפני שהשותף הראשון השלים את פרק 5.**

שלבים א – ב, ה של פרק זה יבוצע על ידי **השותף הראשון** במחשבו האישי,

שלבים ג – ד של פרק זה יבוצע על ידי **השותף השני** במחשבו האישי,

רצוי שבזמן ביצוע הפרק שני השותפים יהיו מחוברים ב-zoom כאשר השותפים משתפים את המסך לסירוגין.

סינכרון מורכב משני חלקים עיקריים: אישור השינויים שביצעתי לפרסום (commit), ועדכון מול המאגר הראשי.

סיכום קצר של הפעולות שאנו הולכים לבצע בשלב זה (ובכלל בכל עבודה משותפת) - כל פעם שמבצעים עדכון המאגר יש 3 שלבים בלשונית של **Git Changes**:

1. כתיבת הערה => **Commit All** [אישור השינויים שביצעתי במחשב שלי]
2. כפתור **pull** [קבלת השינויים שאחרים ביצעו מהמאגר הראשי למחשב האישי ומיזוגם עם השינוים שביצעתי במחשב האישי שלי]
3. כפתור **push** [עדכון השינויים במאגר הראשי ב **github.com**]
4. נ.ב. במקום שתי הפעולות ב. ו-ג. ניתן לבצע פעולה אחת ב-**VS** ע"י כפתור **Sync**

### אישור השינויים (ע"י השותף הראשון)

1. Graphical user interface, application

   Description automatically generatedאם חסרה לשונית **Git Changes**, בחרו מהתפריט הראשית **View => Git Changes**: Graphical user interface, application

   Description automatically generated
2. בפעם הראשונה, אם לא עדכנתם את הפרטים כמו שמופיע [בסעיף 9 בפרק 4](#49x2ik5), נצטרך למסור פרטים שיתארו אותנו ויופיעו על כל "**commit**" שנעשה (כלומר, כל שורת קוד או קובץ שנוסיף, יהיה רשום שאנחנו הוספנו אותו, עם הפרטים האלו).
3. בשלב הזה יופיע פירוט כל השינויים שנעשו בתיקיית העבודה המקומית (לעומת המאגר בשרת שכלל רק את התיקייה הראשית, אנחנו הוספנו את ה-**Solution** ואת הפרויקט עם הקוד כנ"ל) – ראה בתמונה לעיל.
4. בחלק של קלט טקסט (בתחילה מופיע שם באפור: Text

   Description automatically generated with medium confidence ) חובה לרשום הערה שתתאר את השינוי שעשינו, למשל: Graphical user interface, text, application, email

   Description automatically generated
5. בחלק עם השינויים (**Changes**) נלחץ על סימן ה"+" בצד ימין. בהמשך, כשתרצו לכלול בעדכון רק חלק מהשינויים שנעשו, תוכלו ללחוץ על סימן ה-"+" מימין לקבצים ספציפיים (ראו את החץ הקטן בתמונה). לאחר הפעולה יופיע חלק חדש של **Staged Changes** (שינויים שעלו "על הבמה" לעדכון במאגר גיט מקומי):Graphical user interface, text, application

   Description automatically generatedGraphical user interface, text, application

   Description automatically generated

בהמשך, אם טעיתם ותרצו "להוריד" קבצים מהבמה – חפשו סימני "-" בחלק של **Staged Changes**.

1. עכשיו על מנת לשלוח את השינויי שעלו על הבמה למאגר גיט המקומי נוכל ללחוץ על הכפתור **Commit Staged** שהחליף את **Commit All** שהיה שם קודם. לידיעתכם, אפשר היה לדלג על סעיף 5 וללחוץ ישירות על **Commit All** – אז כל השינויים היו "עולים על הבמה" והעדכון היה נשלח למאגר בלחיצה אחת. השלימו את הפעולה ע"י הלחיצה כפי שהוסבר. שימו לב שלמעלה הופיעה הודעה שהעדכון (**commit**) נבנה מקומית (בענף **main**) ועוד יופיעה שיש עדכון יוצא אחד (זאת אומרת יש עדכון במאגר מקומי שלא נשלח למאגר המרוחק – בגיטהאב): Graphical user interface, application

   Description automatically generated

**נ.ב.1**: אם כפתור העדכון "אפור" ולא ניתן ללחוץ עליו – זה אומר שאין שינויים שאפשר לעדכן אתם את המאגר המקומי. זה יכול לקרות גם אם באמת אין שינויים וגם אם בטעות המאגר שלכם לא מקושר לתיקיית העבודה של הפרויקט בגלל שטעיתם או דילגתם על משהו באחד השלבים הקודמים. במצב הזה אולי תרצו למחוק את התיקייה ולחזור לאחד השלבים הקודמים (לפי המצב שבו זיהיתם את הבעיה).

**נ.ב.2**: שימו לב שבכפתור העדכון ניתן לפתוח תפריט ולראות אפשרויות נוספות הכוללות "**… and Push**" ו-"**… and Sync**". מטרת הראשונה לבצע שליחה אוטומטית של העדכון במאגר המקומי למאגר המרוחק (מה שנעשה בנפרד בצעדים הבאים), והשנייה תכלול גם משיכה ומיזוג העדכונים החדשים מהמאגר המרוחק למאגר המקומי לפני ביצוע העדכון המקומי עם השינויים שלכם – נרחיב על זה בהמשך.

### עדכון המאגר הראשי (ע"י השותף הראשון)

1. ניתן לבצע פעולת **fetch** ע"י כפתור של חץ מקוקו יורד  על מנת לעדכן את המאגר המקומי לגבי עדכונים במאגר הרחוק שלא מופיעים במאגר המקומי. לחלופין יש לשאול את השותף האם שלח עדכונים לגיטהאב.
2. פעולת שליחת עדכונים מהמאגר המקומי למאגר המרוחק (בגיטהאב) תתבצע **רק** כאשר בחלון רואים ש**יש עדכונים יוצאים** ו**אין עדכונים נכנסים**. הפעולה תתבצע ע"י לחצי על כפתור **Push** (החץ העולה) כדלקמן: Graphical user interface, application

   Description automatically generated

יתכן ותיתקלו בבעיה בביצוע push, כאשר בהודעת השגיאה כתוב: "SSL certificate problem: Unable to get local issuer certificate". במקרה הזה יש לבצע בחלון git-bash או בחלון CMD (אם התקנתם גם את התכנה של גיט מהאתר git-scm.com) את הפקודה הבאה:Graphical user interface, application

Description automatically generated

C:\> **git config --global http.sslVerify false**

נ.ב. קישור להורדת התכנה של גיט: <https://git-scm.com/download/win> ושם לחיצה על אחד הקישורים הבאים (הריצו את קובץ ההתקנה לאחר השלמת ההורדה):



1. באזור הודעות גיט יופיעו ויתחלפו הפעולות הפנימיות המתבצעות ע"י ויז'ואל סטודיו בגיט לאחר השלמת פעולת השליחה (ה"דחיפה") תופיע הודעת הצלחה:
2. אם תקבלו הודעה על תקלה – כנראה שלא הקפדתם על אחד או יותר מהכללים שהזכרנו לעיל.
3. תפתחו את האתר של גיטהאב במאגר שלכם וראו שהוא התעדכן (אם היה פתוח – תעשו ריענון לחלון): Graphical user interface, text, application, email

   Description automatically generated
4. בתפריט Git של VS תוכלו למצוא אפשרויות **Manage Branches** ו-**View Branch History**. תפתחו אותן ותעיינו בלשוניות שנפתחות, תנסו להבין מה רואים.

### עדכון המאגר המקומי אצל השותף השני (ע"י השותף השני)

1. בשלב הזה נכנס לתמונה השותף השני במחשבו האישי. כמובן שהשותף השני כבר יצר מאגר מקומי ע"י השלמת ההנחיות ב[פרק 4](#2s8eyo1) **במלואן ובמדויק**. שימו לב שאם השותף השני ביצע את פרק 4 לאחר שהשותף הראשון השלים את דחיפת השינויים של שלב ב' בפרק הזה (כנ"ל), בתיקיית העבודה של המאגר כבר תופיע תיקיית ה-Solution. נתייחס למצב הזה בהמשך – אך חובה על שני השותפים לקרוא את הנחיות השלב הזה בעיון, כי כאן תיבנה ההבנה של פעולת משיכת העדכונים (**pull**).
2. השותף השני יריץ את ויז'ואל סטודיו, אך במקום יצירת פרויקט חדש יבקש לפתוח את תיקיית המאגר לפי אחת משני התמונות הבאות (תלוי באיזה מצב יהיה ה-**VS**): Graphical user interface, text, application

   Description automatically generatedGraphical user interface, application

   Description automatically generated
3. בחלון שייפתח סייר לתיקיית המאגר ופתח אותה (כפתור Select Folder): Graphical user interface, application

   Description automatically generated
4. אם [פרק 4](#2s8eyo1) בוצע ע"י השותף השני לפני נוספת פרויקט **VS** ע"י השותף הראשון, על השותף השני לבצע משיכת השינויים ע"י לחיצה על כפתור משיכת עדכונים (**pull**): Background pattern

   Description automatically generated with low confidence
5. עבור ללשונית **Solution Explorer** ובחר בלחיצה כפולה (**double-click**) את קובץ הניהול של **VS** עם סיומת **.sln**: Graphical user interface, text, application, email

   Description automatically generated
6. VS יחליף את הסביבה מהתיקייה של המאגר לפתרון ב-**#C** הזהה למאגר ב-**github.com** ולמאגר המקומי של השותף הראשון.
7. כעת, השותף השני יוסיף "קובץ נוסף" למחלקה, שבו הגדרה נוספת של המחלקה החלקית **Program** – ע"י הוספת מחלקה חדשה בשם **Program**: Graphical user interface, application

   Description automatically generatedGraphical user interface, application

   Description automatically generated
8. לאחר מכן ישנה השותף השני את שם המודול (הקובץ) החדש ע"י הוספת ארבעת הספרות האחרונות של מספר ת"ז שלו (**ProgramYYYY.cs**) – ללא עדכונים רוחביים (עם לחיצה על **No** בחלון הקופץ).
9. להגדרת המחלקה יוסיף השותף השני את מילת מפתח **partial** בדומה למה שעשה השותף הראשון, כמו כן יוסיף בתוך המחלקה את הפעולה **WellcomeYYYY**, הפעם עם קוד שבו יבצע פלט ו-\או קלט ברצונו.Graphical user interface, text, application, email

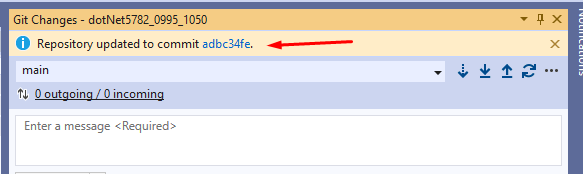
   Description automatically generated
10. לאחר שסיימתם את השלב הזה – החלון אמור להיראות כך:

### עדכון וסנכרון המאגר של השותף השני (ע"י השותף השני)

1. השותף השני יבצע גם הוא את עדכון השינויים במאגר המקומי ושליחת העדכון למאגר המרוחק כפי שמופיע בשלבים [א'](#44sinio) ו-[ב'](#z337ya) של הפרק הזה.

### סנכרון המאגר של השותף הראשון (ע"י השותף הראשון)

1. השותף הראשון יבצע משיכת העדכון של השותף השני ע"י לחיצה על כפתור **pull** בלשונית **Git Changes**. Background pattern

   Description automatically generated with low confidence
2. עקוב אחר ביצוע הפעולה בשורת המצב וראה את הודעת ההצלחה. 

סיכום: בשני הפרקים האחרונים, הדגמנו איך כל אחד מהשותפים לזוג יכול לעבוד במקביל, במחשב שלו על מודולים\קבצים שונים באותה תיקייה בתוך המאגר, ובסוף, מעבירים את העבודה של כל אחד מהשותפים לזוג למחשב של השני.

באופן כללי, כל פעם שתלמיד א' רוצה לעדכן את תלמיד ב' במה שהוא עשה, הסדר הוא:

1. כל אחד משני השותפים מעדכן במאגר המקומי שלו (**Commit**) את השינויים שביצע במחשב שלו.
2. שותף ראשון מבצע סנכרון מול המאגר באתר (מעביר את השינויים שאושרו על ידיו לאתר).
3. שותף שני מסנכרן מול המאגר באתר (מושך את העדכונים של השותף הראשון ודוחף את העדכונים של עצמו).
4. שותף ראשון מסנכרן עוד פעם מול המאגר באתר (מושך את העדכונים של השותף השני)
5. וחוזר חלילה סעיף 3 לסירוגין... כל פעולת סנכרון מושכת עדכונים מהמאגר בגיטהאב (**pull**) ואז דוחפת את השינויים מהמאגר המקומי לגיטהאב (**push**). שתי הפעולות **pull** + **push** אוגדו ב-VS ל"פעולה" אחת של **sync** שהזכרנו קודם. והכלל הלא כתוב הוא:

**"Always Pull Before Push"**

כעת נעבור להדגמת עבודה במקביל על אותו הקובץ.

## פעולות נוספות לתרגול עבודה בזוגות[[2]](#footnote-2)

בשלב זה אנחנו מתרגלים ביצוע שינויים במקביל, על ידי שני השותפים לזוג, **באותו הקובץ**. בסיום כל שלב, חובה לבצע סינכרון, ולוודא שכל אחד מהצדדים נשאר עם השינויים ששני הצדדים עשו.

תזכורת של הפעולות שאנו הולכים לבצע בשלב זה (ובכלל בכל עבודה משותפת) - כל פעם שמבצעים עדכון המאגר יש 3 שלבים בלשונית של **Git Changes**:

1. כתיבת הערה => **Commit All** [אישור השינויים שביצעתי במחשב שלי]
2. כפתור **pull** [קבלת השינויים שאחרים ביצעו מהמאגר הראשי למחשב האישי ומיזוגם עם השינוים שביצעתי במחשב האישי שלי]
3. כפתור **push** [עדכון השינויים במאגר הראשי ב **github.com**]
4. נ.ב. במקום שתי הפעולות ב. ו-ג. ניתן לבצע פעולה אחת ב-**VS** ע"י כפתור **Sync**

### עבודה באותו קובץ, בשורות שונות

במאגר שלנו, בתיקייה הראשית קיים קובץ שנקרא README.md. הקובץ הזה בפורמט [markdown](https://en.wikipedia.org/wiki/Markdown)  נוצר על ידי github כשיצרנו את המאגר. אנחנו נתרגל בחלק זה שינויים בקובץ (השינויים יכולים להתבצע גם ישירות ב-github – אך אנו רוצים לעבוד דווקא ב-VS). על מנת לפתוח את הקובץ בVS יש להוסיף קובץ קיים: Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

לאחר מכן בלשונית של Solution Explorer תופיע תיקיה וירטואלית (מדומה) בשם Solution items עם הקובץ: Graphical user interface, text, application

Description automatically generated

1. אחד מהשותפים לזוג יוסיף שורה לפני השורה הראשונה ויכתוב בה משהו כרצונו. לאחר מכן ישלח את השינוי לעדכון (Commit) במאגר המקומי של גיט וידחוף את העדכון (push) למאגר המרוחק בגיטהאב
2. השני מהשותפים לזוג ימשוך (pull) את העדכון מגיטהאב ויוסיף שורה אחרי כל השורות וכותב בה Hello World. לאחר מכן ישלח את השינוי לעדכון במאגר המקומי של גיט וידחוף את העדכון למאגר המרוחק בגיטהאב
3. השותף הראשון ימשוך את העדכון האחרון למאגר המקומי שלו.
4. וודאו ששני השותפים לזוג קיבלו את כל השורות שהיו קודם, בתוספת שתי השורות החדשות.

### עבודה באותן שורות

1. שני השותפים נכנסים לתפריט Git => Settings ובחלון שנפתח בוחרים בצד שמאל Source Control => Git Global Settings. בחלון הזה גוללים בצד ימין עד למטה ומוודאים שבכלים מוגדר עבור שני הכלים (Diff Tool ו-Merge Tool) שימוש בויז'ואל סטודיו (Use Visual Studio). Graphical user interface, application

   Description automatically generatedGraphical user interface, text, application

   Description automatically generated
2. אחד מהשותפים לזוג משנה את המילה Hello למילה אחרת כרצונו, השני משנה את המילה world.
3. אחד מהשותפים לזוג מבצע עדכון וסנכרון ראשון. Graphical user interface, application, Word

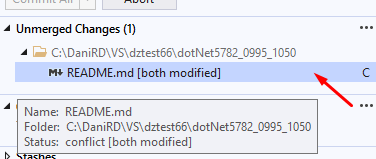
   Description automatically generatedGraphical user interface, application, Teams

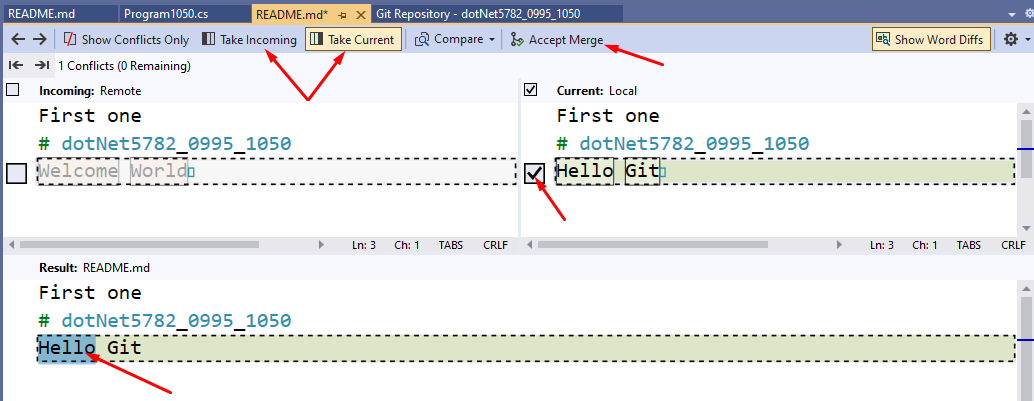
   Description automatically generated
4. השני מבצע עדכון וסנכרון, ומגלה שיש תקלה: Graphical user interface, text, application, email

   Description automatically generatedGraphical user interface, text, application

   Description automatically generated

כך נראים חלון הקוד וחלון ה-Git Changes. חלון הקוד מציג את הנקודה בה יש סתירה בין שתי הגרסאות (כמה חיצים בתחילת השורה), וחלון ה-Git Changes הנ"ל, מציג את העובדה שהסנכרון לא הושלם, אלא יש תקלות.

1. עשו לחיצה כפולה על הקובץ בחלק Unmerged Changes בחלון Git Changes או על הקישור Open Merge Editor בחלון המציג את ההתנגשות בקוד\בטקסט. ייפתח חלון ההשוואה עם כפתורי בחירת החלטות ואישור מיזוג השינויים. Graphical user interface, text, application, email

   Description automatically generated
2. בחלון ניתן לסמן שינויים מסוימים בצד שמאל (השינויים שהגיעו מהמאגר המרוחק) או בצד שמאל (השינויים בעדכון במאגר המקומי שעוד לא נשלח למאגר הרחוק) או לסמן בכפתורים למעלה לקחת את כל השינויים מהמאגר המרוחק (Take Incoming) או מהמאגר המקומי (Take Current) ולאחר מכן לבצע עריכה ישירה ידנית למטה בתוצאה של המיזוג. לבסוף יש ללחוץ על כפתור אישור המיזוג .
3. בחלון ה-Git Changes השינויים יעברו ל-Staged Changes. השלימו את תיאור העדכון (מומלץ להתחיל ממילה Merge על מנת לציין שמדובר בעדכון-מיזוג. לאחר שכל הקבצים עם התנגשויות (כרגע יש רק אחד כזה) יעברו על הבמה, לחצו על כפתור Commit Staged ולאחר מכן דחפו את העדכון-מיזוג למאגר המרוחק. Graphical user interface, text, application, email

   Description automatically generatedGraphical user interface, application

   Description automatically generated
4. עשו עוד ניסיונות של שינויים "לא מסונכרנים" באותו הקובץ (לא חייב להיות באותה שורה – העיקר שהשינויים באותו הקובץ) שיגרמו להתנגשויות (תתחלפו ב"תפקידים" של הראשון והשני) ובדרך שחקו עם האפשרויות השונות בחלון המיזוג. הניסיון שתרכשו יעזור לכם בהתמודדויות עם בעיות של חוסר תאום פעולה בין בני הזוג במהלך העבודה על שלבי הפרויקט של הקורס.

שימו לב שבעבודה על פתרון ההתנגשויות אין כלל ברור, לפעמים הגרסה הסופית תהיה שילוב של שני השינויים שיתווספו ידנית על ידי הממזג.

## הגשה

### יצירת תג (Tag)

לפני הגשה יש ליצור טאג במאגר.

1. בצעו את השינוי האחרון ועדכנו אותו עם כיתוב "Stage0 Final commit", בצעו סנכרון העדכון לאתר.
2. בויז'ואל סטודיו של השותף שביצע את העדכון האחרון הנ"ל לחצו על הקישור 0 outgoing / 0 incoming או בחרו בתפריט Git => Manage Branches Graphical user interface, text, application, Word

   Description automatically generatedGraphical user interface, application

   Description automatically generated
3. בחלון שייפתח וודאו שבצד שמאל מודגש הענף המקומי הראשי. לאחר מכן עם הסמן מעל העדכון האחרון לחצו על כפתור הימני בעכבר על מנת לפתוח תפריט הקשר. בתפריט שייפתח בחרו ב-Create Tag. Graphical user interface, text, application

   Description automatically generated
4. ייפתח חלון עם תצוגת השינויים של העדכון ובצד ימין חלק של פעולות על העדכון, כולל הוספת תאג. בשדה שכתוב בו בהתחלה Enter a tag name כתבו את שם התאג עבור ההגשה **stage0**. אין צורך להוסיף הודעת הסבר – כי העברתו לא בהכרח נתמכת בדחיפת התאג למאגר המרוחק. לחצו על הכפתור Create Tag. תראו שנוסף התאג בהיסטוריה של עדכונים. Graphical user interface, application

   Description automatically generatedText

   Description automatically generated with medium confidence
5. בחלון Git Changes לחצו על כפתור  ובתפריט שייפתח בחרו בדחיפת כל התאגים ובחרו בחיבור המרוחק origin. Graphical user interface, text, application, Word

   Description automatically generated
6. כנסו למאגר באתר גיטהאב (או רעננו את החלון אם כבר הייתם שם). שימו לב שבנוסף לכיתוב 1 branch נוסף גם כיתוב 1 tag. ב תפריט הענפים (עם הענף הראשי) בחרו בלשונית Tags. תראו שם את התאג שייצרתם. Graphical user interface, text, application, email

   Description automatically generated
7. אם תלחצו על הכיתוב 1 tag ייפתח חלון עם כל התאגים (כרגע רק אחד), לחצו בחלון הזה על שם התאג שייצרתם (stage0) ותראו יותר פרטים עליו. שימו לב שבגיטהאב מייצרים גרסת שחרור (release) עבור התאג – משהו ייחודי לאתר גיטהאב ולא קשור לגיט עצמו.
8. בתפריט שפתחתם בסעיף 6 לעיל לחצו על שם התאג. הכיתוב של הענף הראשי יתחלף לתאג.Graphical user interface, text, application, email

   Description automatically generated
9. בעמוד שמוצג בכתובת הדף יופיע משהו דמוי: **https://github.com/***account-name***/***repo-name***/tree/stage0**. A screenshot of a computer

   Description automatically generated with medium confidence
10. עליכם להעתיק את הכתובת (הקישור) הזו לטקסט בתיבת ההגשה. Graphical user interface, text, application

    Description automatically generated

**נ.ב. שימו לב – כל אחד מהשותפים לזוג חייב לבצע בנפרד את ההגשה לתיבת ההגשה של השלב!**

1. רשימה חלקית של עוד **snippet** שונים ניתן לראות בקישור הבא (ניתן גם להיכנס לעורך **snippet** ע"י לחיצה **Ctrl-K** ואז **Ctrl-B**):  
   [**https://docs.microsoft.com/en-us/visualstudio/ide/visual-csharp-code-snippets?view=vs-2022**](https://docs.microsoft.com/en-us/visualstudio/ide/visual-csharp-code-snippets?view=vs-2022) [↑](#footnote-ref-1)
2. ניתן לבצע את רוב החלק הזה על ידי אדם בודד. פשוט מבצעים צד אחד במחשב בתוכנת **VS**, וצד שני באתר של **github**. מתרגלים את המשיכה ודחיפה מתוך **VS**. החיסרון: המדריך לא כולל הדרכה של "איך לבצע שינויים בקוד באתר של **github**". מצד שני, זה די אינטואיטיבי. [↑](#footnote-ref-2)