

# Github

## ◀ עבודה עם visual studio

- ◆ פתיחת ריפו ופרוייקט
- ◆ שמירת שינויים ל GITHUB
- ◆ לקיחת שינויים מ GITHUB
- ◆ הוספת משתמש לפרויקט

## ◀ עבודה עם האתר של GitHub

- ◆ פתיחת פרויקט
- ◆ הוספת משתמש לפרויקט

## ◀ עבודה עם visual studio code

- ◆ עבודה על הפרויקט בצורה מקומית
- ◆ העלאת קבצים וסנכרון
- ◆ ניהול נכון של קובץ README

## ◀ טיפים וכלים

אורחיי בנאים וניר חן



# מינוחים

Repo - מקום שנשמור בו את הקוד. יש אחד מקומי על המחשב ויש אחר אתר של GITHUB

Commit – הכרה של GIT בשינוי האחרון שעשינו

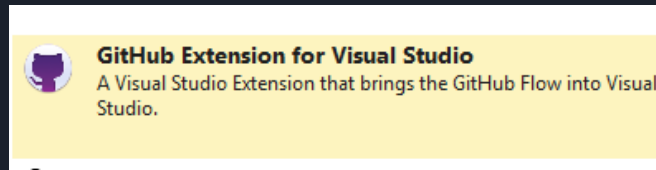
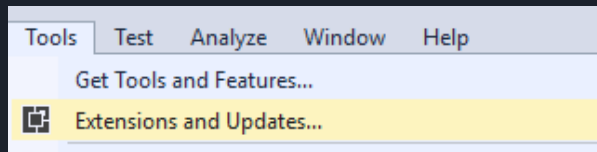
Stage – הצהרה על קובץ כמועמד לCommit

Push – העלאה של קובץ מהריפו המקומי לריפו של GITHUB

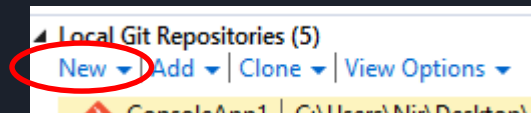
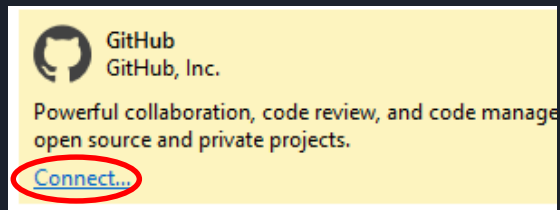
Sync – סנכרון מול הריפו של GITHUB לריפו המקומי, הכולל Pull ואז Push

# עבודה בעזרת visual studio התחברות

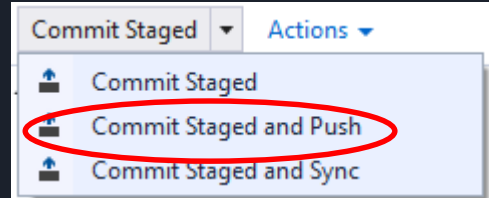
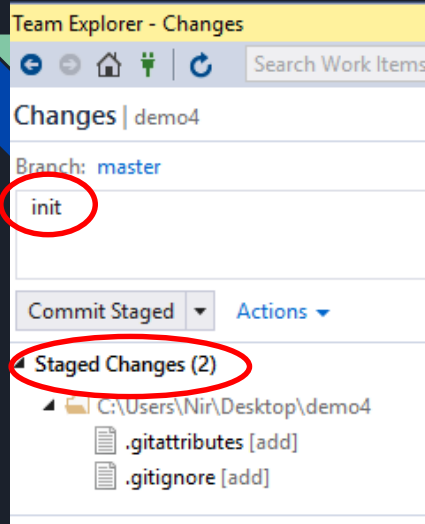
נתקין את התוסף של GitHub ל VS



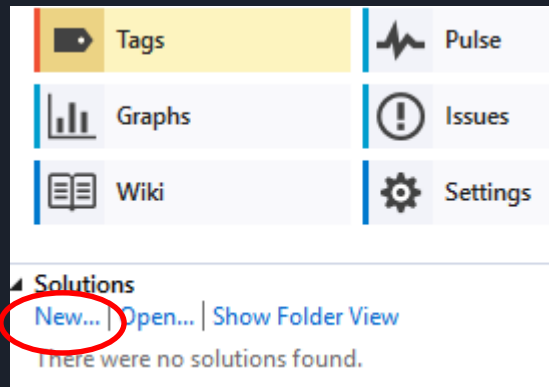
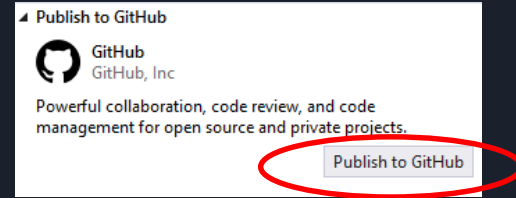
נתחבר ונייצר ריפו חדש(עדין ריק)



# עבודה בעזרת visual studio יצירת ריפו ופרוייקט

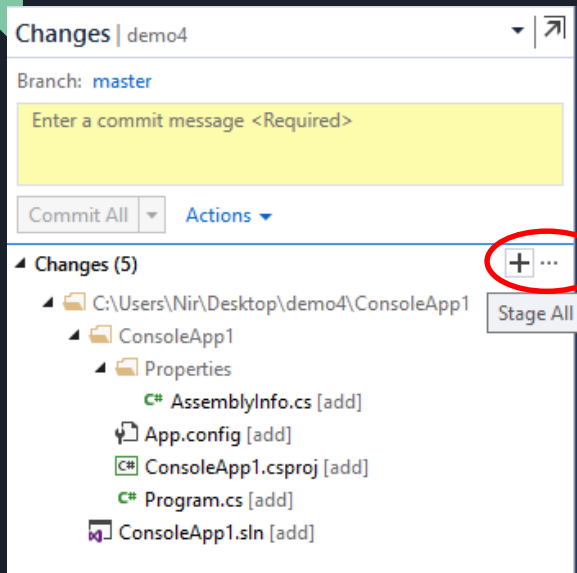


נכנס לריפו ותחת Changes ניתן שם לCommit למשל 'init'  
ועכשיו כל מה שב 'Staged Changes' בלחיצה על  
'Commit Staged and Push' ישמר גם בריפו הלוקאלי וגם  
יעבור לGit



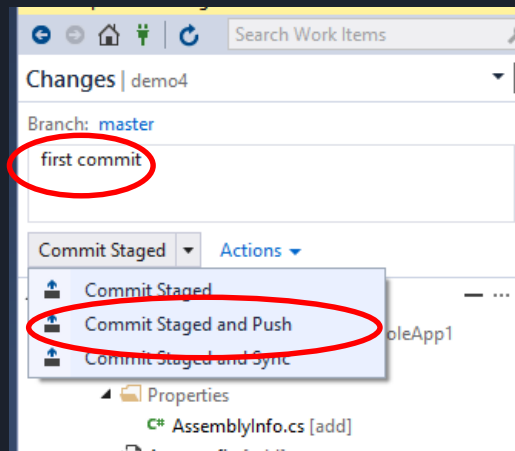
נניצר בריפו הזה (demo4) Solution חדש מכל סוג שהוא

# עבודה בעזרת visual studio העלאת הפרוייקט בפעם הראשונה



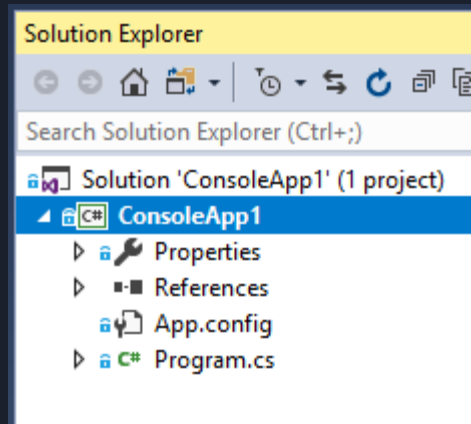
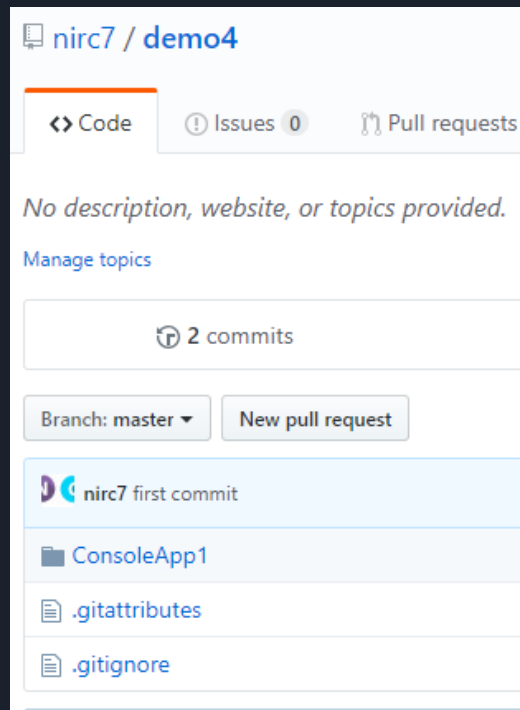
לאחר יצירת הפרוייקט נעבור ל CHANGES ונעביר את כל הקבצים למצב STAGE

עכשיו שוב ניתן לתת שם ל COMMIT ולבצע COMMIT AND PUSH



# עבודה בעזרת visual studio אתר GitHub

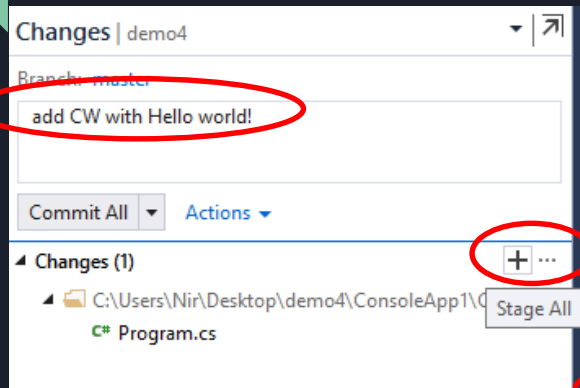
עכשיו נוכל לראות את הפרוייקט שלנו גם באתר של  
GITHUB



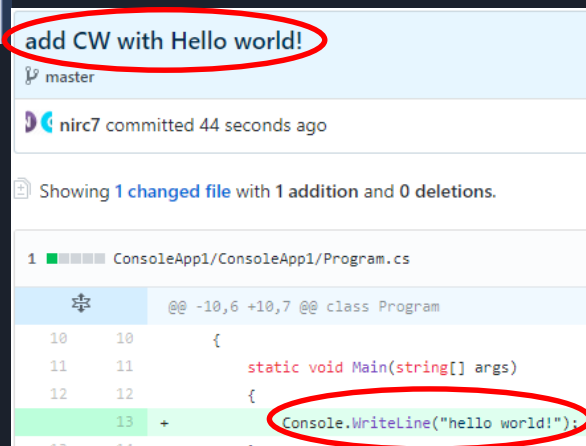
גם ב SOLUTION EXPLORER  
ניתן לראות שהקבצים נמצאים  
תחת SOURCE CONTROL

# עבודה בעזרת visual studio

## שמירת שינויים של הקוד שלנו בריפו של GitHub



כמו קודם תחת CHANGES ניתן שם ל COMMIT נעביר  
את השינויים לCHANGES STAGE ונבצע COMMIT  
AND PUSH



נרפרש באתר ונוכל גם שם לראות  
את השינויים החדשים בקוד

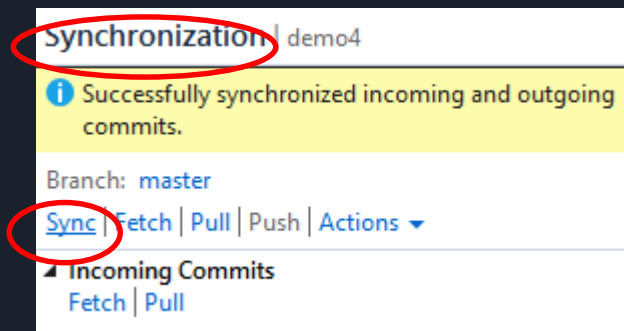
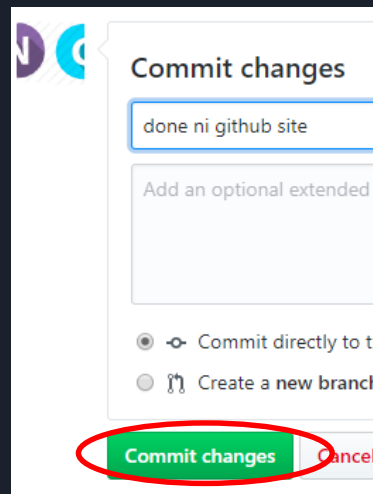
# עבודה בעזרת visual studio

## סינכרון עם שינויים שנעשו במקום אחר (משתמש אחר)



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ConsoleApp1
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13             Console.WriteLine("Welcome to GitHub");
14         }
15     }
16 }
```

למשל שינויים דרך אתר של GITHUB

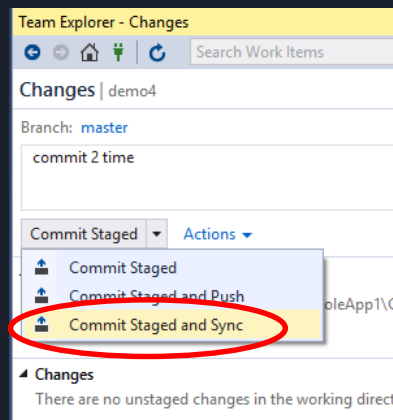
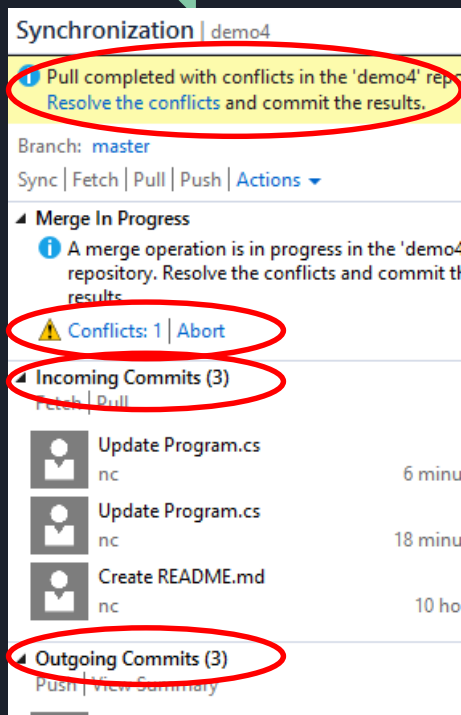


פשוט נלך ל SYNCHRONIZATION ונלחץ על Sync ☺



# עבודה בעזרת visual studio

## סינכרון עם שינויים שלנו ושנעשו במקום אחר (משתמש אחר)

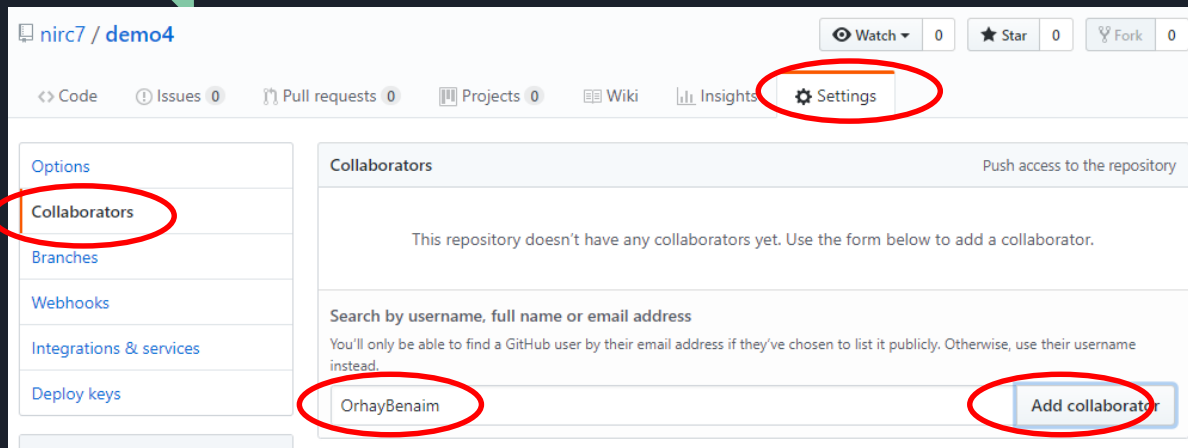


נבחר ב COMMIT STAGED AND SYNC

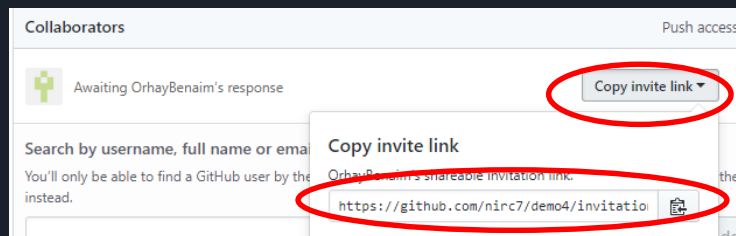
```
<<<<<<< HEAD
      Console.WriteLine("save from local seoncd time!");
=====
      Console.WriteLine("save from github second time");
>>>>>>> aa51f836c6d74fd9026f4c16a19a55d1b50393be
      }
```

במקרה יש קונפליקטים נוכל לראות אותם ולהחליט האם להשאר עם העותק הלוכלי או לקחת את העותק מהמשתמש השני

# עבודה בעזרת visual studio צירוף של מתכנתים נוספים לפרוקייט



פשוט להזמין ע"י שם המשתמש שלו בGITHUB

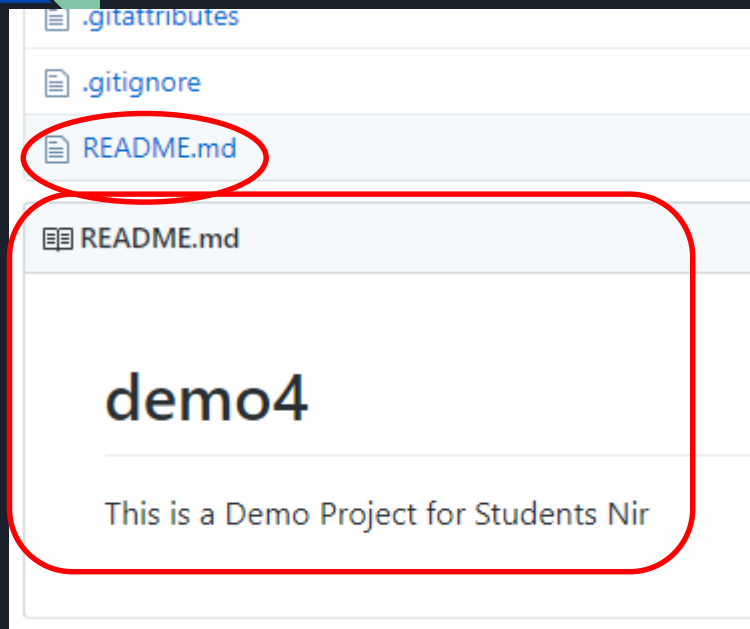


או ע"י שליחת הלינק במייל...

# עבודה בעזרת visual studio

## קובץ README

חובה לערוך את הקובץ הזה הכדי שיהיה אפשר להבין בקלות על מה הפרוייקט



# Github

## עבודה עם visual studio ◀

- ◆ פתיחת ריפו ופרוייקט
- ◆ שמירת שינויים ל GITHUB
- ◆ לקיחת שינויים מ GITHUB
- ◆ הוספת משתמש לפרויקט

## עבודה עם האתר של GitHub ◀

- ◆ פתיחת פרויקט
- ◆ הוספת משתמש לפרויקט

## עבודה עם visual studio code ◀

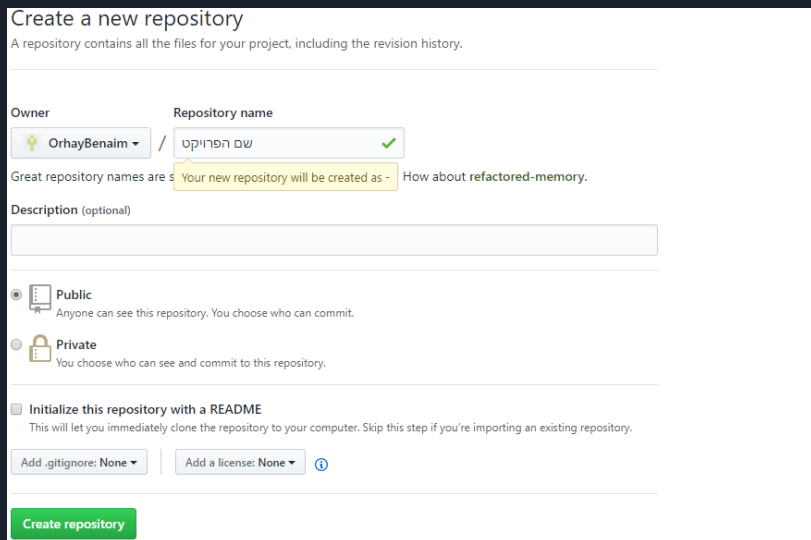
- ◆ עבודה על הפרויקט בצורה מקומית
- ◆ העלאת קבצים וסנכרון
- ◆ ניהול נכון של קובץ README

## טיפים וכלים ◀

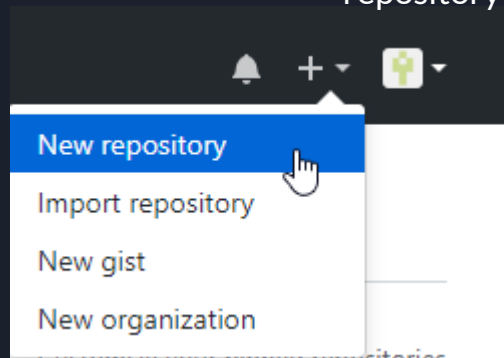
# פתיחת פרויקט חדש

ניתן לפתוח פרויקט חדש על ידי לחיצה על פלוס ו repository

לאחר מכן נמלא את שם הפרויקט ונלחץ create repository



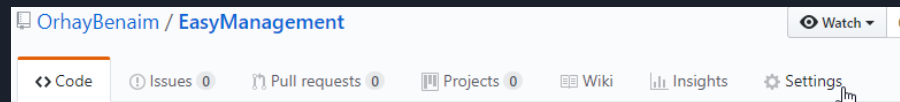
The screenshot shows the GitHub 'Create a new repository' page. At the top, it says 'Create a new repository' and 'A repository contains all the files for your project, including the revision history.' Below this, there are two main sections: 'Owner' and 'Repository name'. The 'Owner' section has a dropdown menu with 'OrhayBenaim' selected. The 'Repository name' section has a text input field with 'שם הפרויקט' (Project name) and a green checkmark. Below these, there is a note: 'Great repository names are s' and a yellow box that says 'Your new repository will be created as -'. To the right of this box is a link 'How about refactored-memory.' Below this, there is a 'Description (optional)' text input field. Further down, there are two radio button options: 'Public' (selected) and 'Private'. The 'Public' option has a subtext: 'Anyone can see this repository. You choose who can commit.' The 'Private' option has a subtext: 'You choose who can see and commit to this repository.' Below these, there is a checkbox labeled 'Initialize this repository with a README' and a subtext: 'This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.' At the bottom, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None'. A green 'Create repository' button is at the very bottom.



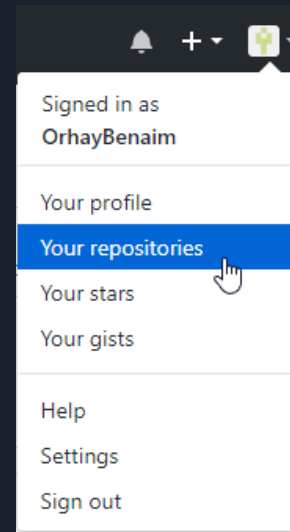
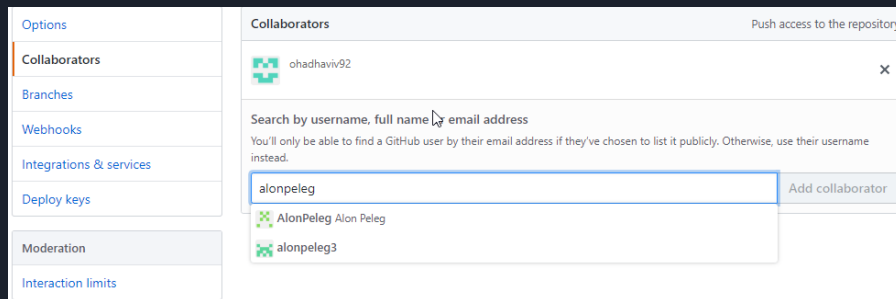
האופציה לפרטי עולה כסף\*

# הוספת משתמשים

לאחר שפתחנו את הריפו שלנו נוכל להיכנס ולשנות הגדרות



לאחר כניסה להגדרות נוכל להיכנס לשותפים collaborators ולהוסיף שותפים לפרויקט



משתמש שלא מופיע בשותפים עדיין יכול לצפות בפרויקט אך לא יכול לעשות שינויים בלי אישור, כל שינוי שיעשה יתבקש אישור מבעל הפרויקט

# Github

## ◀ עבודה עם visual studio

- ◆ פתיחת ריפו ופרוייקט
- ◆ שמירת שינויים ל GITHUB
- ◆ לקיחת שינויים מ GITHUB
- ◆ הוספת משתמש לפרויקט

## ◀ עבודה עם האתר של GitHub

- ◆ פתיחת פרויקט
- ◆ הוספת משתמש לפרויקט

## ◀ עבודה עם visual studio code

- ◆ עבודה על הפרויקט בצורה מקומית
- ◆ העלאת קבצים וסנכרון
- ◆ ניהול נכון של קובץ README

## ◀ טיפים וכלים

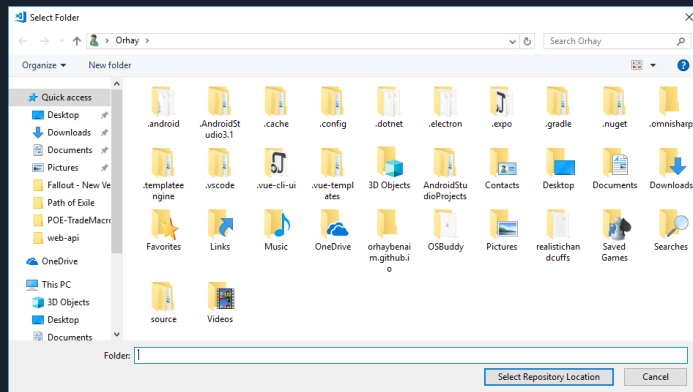
אורחיי בנאים וניר חן

# עבודה בעזרת visual studio code

מזין את כתובת הפרויקט שלנו

```
https://github.com/OrhayBenaim/EasyManagement  
Repository URL (Press 'Enter' to confirm or 'Escape' to cancel)
```

נבחר איפה לשמור את הפרויקט. לא צריך להכין תיקיה, גיט יפתח תיקיה בשם הפרויקט



נלחץ F1 ונחפש git ונבחר באופציה של Git: Clone

```
>git  
Git: Clone recently used  
Git: Initialize Repository  
Git: Open Repository other commands  
Git: Show Git Output  
GitLens: Open Settings  
GitLens: Welcome  
View: Show GitLens
```



# Commit

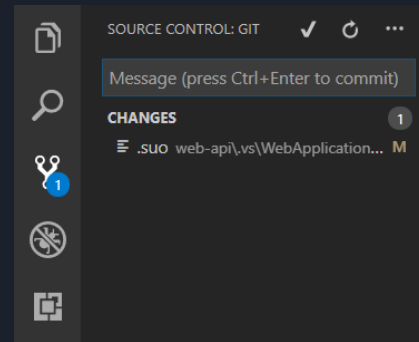
לחיצה על ה ✓ תפעיל את פקודת ה commit.

פה ניתן שם לשינוי שאנו מבצעים

Commit message

Please provide a commit message (Press 'Enter' to confirm or 'Escape' to cancel)

כאשר נעשו שינויים מקומיים יופיע עיגול כחול עם כמות הקבצים שהשתנו



כל פעולת commit תשמור את השינויים בגיט המקומי  
במחשב.

# העלאת קבצים וסנכרון

בכדי לקבל את הגרסה העדכנית שבשרת הגיט  
נשתמש בפקודה pull

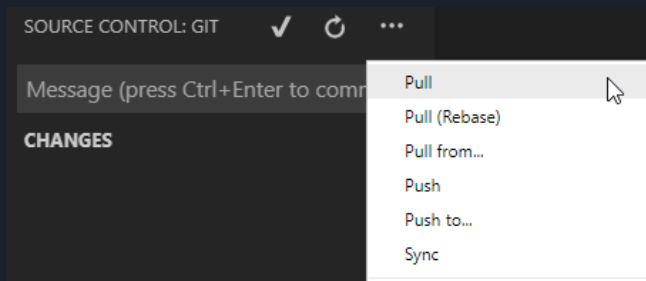
בכדי להעלות את השינוי שהתבצעו נשתמש  
בפקודה push



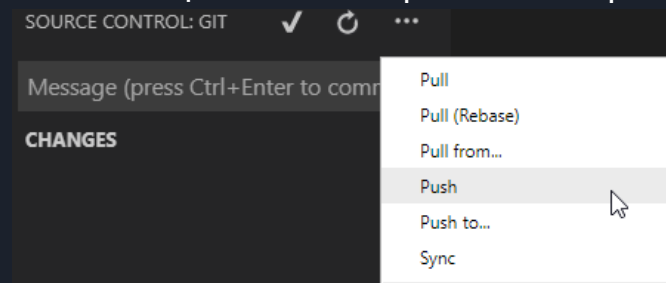
visual studio code סנכרון ל source control

לפני ביצוע הפקודה יש לבדוק שאין שינויים שמחכים לשמירה.

אם יש יופיע הודעת שגיאה שצריך לטפל בקבצים לפני ביצוע  
הפקודה  
נלחץ על שלושת הנקודות ונבצע pull



נלחץ על שלושת הנקודות ונבצע push



אם נרצה לבצע פקודת pull ולאחר מכן push נוכל  
להשתמש בפקודה sync

# מיזוג שינויים

כאשר שני משתמשים עובדים על אותו הקובץ ויש התנגשויות בקובץ ברגע שנעשה pull או sync יופיע לנו האופציה איזה שינויים להשאיר

```
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<<<< HEAD (Current Change)
| | | | | drawerLabel: 'חדש בית',
=====
| | | | | drawerLabel: 'home',
>>>>>>> 3e974710ced40c160ae50d90be4d265defd0c311 (Incoming Change)
```

בצבע ירוק השינויים המקומיים שלנו, בצבע כחול השינויים שמגיעים מהשרת

נוכל לבחור האם להשאיר את השינוי שלנו או האם לקבל את השינוי החדש או האם לקבל את שני השינויים.

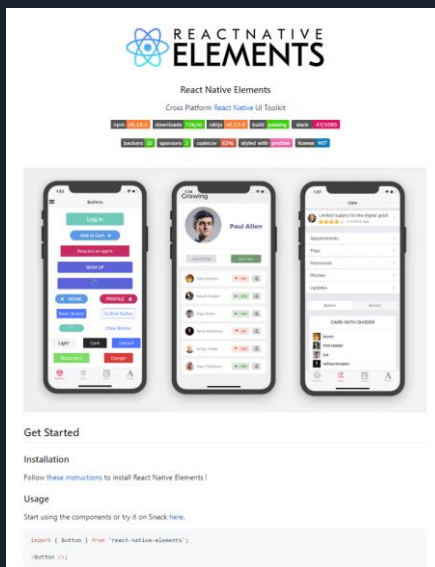
אם נרצה עוד מידע נוכל לבחור בהשוואת השינויים ששם נראה גם את ההודעה של אותו השינוי (הודעת ה commit)

# README

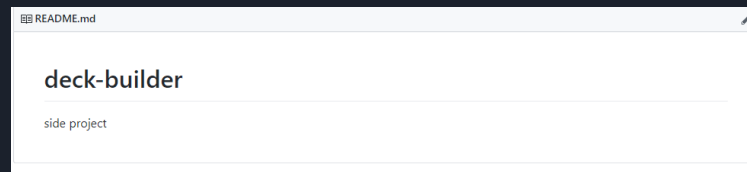
קובץ ה readme זה הקובץ המוצג בכניסה לפרויקט.

בקובץ זה נציג תיאור על הפרויקט, איך להתקין את התכנים שהפרויקט זקוק לו בכדי לעבוד, שלבי התקנה ועוד.

דוגמא טובה ל readme



דוגמא לא טובה ל readme



# טיפים וכלים

## כלים

## טיפים

- חלוקת עבודה בין חברים הקבוצה
- שימו לב שאתם עובדים על הגרסה העדכנית
- תיאור הגיוני של commit

לכל משתמש ניתן האופציה לפתיחת אתר אחד  
חינם

בכדי להשתמש בזה נצטרך לפתוח פרויקט בשם

username.github.io

את username ייש להחליף בשם המשתמש

**שימו לב שקובץ האתר נקרא index.html  
והוא נמצא בתיקייה הראשית של הפרויקט**

דוגמא <https://orhaybenaim.github.io/>

git <https://git-scm.com/downloads>

git desktop <https://desktop.github.com/>

git kraken <https://www.gitkraken.com/git-client>

gitlens <https://gitlens.amod.io/>

<https://marketplace.visualstudio.com/items?itemName=eamodio.gitlens>

[s](https://marketplace.visualstudio.com/items?itemName=eamodio.gitlens)

טמפלט לאתר לריכוז מידע אישי על הפרויקטים ועל עצמי

<https://github.com/OrhayBenaim/OrhayBenaim.github.io>