

## MagitHub – ex03

**מגישים :** דן בלדב – 205684277 – [danbeladev@gmail.com](mailto:danbeladev@gmail.com)

נווה פרג'ון – 312494206 – [navefarj@gmail.com](mailto:navefarj@gmail.com)

URL: localhost:8080/MagitHubWar

### **מבנה ועיצוב:**

התוכנית שלנו מורכבת משני מודולים עיקריים:

**MagitWebApp** – מודול זה אחראי בעצם על ממשק המשתמש ( web client ). ומכיל בתוכו את כל דפי ה HTML, JAVA SCRIPT, ו CSS.

בנוסף במודול זה נמצאים כל הסרבליטים הנדרשים על מנת לתפעל את האפליקציה. ( צד השרת – העושה שימוש במודול "מנוע המערכת" ).

**MagitEngine** - זהו מנוע המערכת. כל הלוגיקה של הפקודות האפשריות Magit טמונה פה. חלק זה פאסיבי לחלוטין ואינו מתקשר באופן ישיר עם המשתמש, ולא יוזם התקשרות עם המשתמש.

בנוסף עושה המערכת שימוש בשני ספריות חיצוניות :

Gson - שבעזרתו אנו מייצרים קבצי json מאובייקטים של השפה java

AncestorFinderJar - שבעזרתו אנו מחשבים את האב הקדמון המשותף הנמוך ביותר בפעולות המיזוג.

### **מחלקות עיקריות:**

Repository - זוהי המחלקה המכילה בחובה את כל המידע עבור רפוזיטורי בודד: בלובים קיימים, קומיטים, בראנצ'ים, מיקום רפוזיטורי - נתיב, מיקום הרפוזיטורי שעוקב אחריו וכולי.

RepositoryManger - מחלקה זו מקבלת מופע של משתמש במערכת ובעצם אחראית לניהול כל הרפוזיטורים שברשותו.

PullRequestManager - מחלקה חדשה שלא הייתה בתרגילים קודמים, מחלקה זה אחראית לניהול כל ה PR עבור משתמש כלשהו במערכת ה M.A.GitHub.

NotificationManager – מחלקה חדשה הייעודית לתרגיל 3, אחראית להצגת התראות למשתמש.

וכמובן שיש מחלקות נוספות במנוע המערכת כמו: Branch, MagitFile, XMLChecker, Commit וכו'.

## **יצירת repository:**

לאחר שנרשם כמשתמש חדש במערכת, או נכנס לחשבון הקיים, בידי המשתמש קיימות שתי דרכים ליצירת repository:

1. טעינת קובץ XML: במסך מספר 2, מיד לאחר שנכנס למערכת, ילחץ על כפתור "Choose file", תחת הכותרת Add new repository, יבחר את קובץ ה XML המייצג את הרפוזיטורי המבוקש, ולאחר מכן ילחץ "Upload File".  
באם הקובץ ימצא תקין, יעלה הרפוזיטורי ויופיע למשתמש ברשימת הרפוזיטורים שלו, אחרת- תעדכן המערכת מה היה פסול בקובץ ה XML.

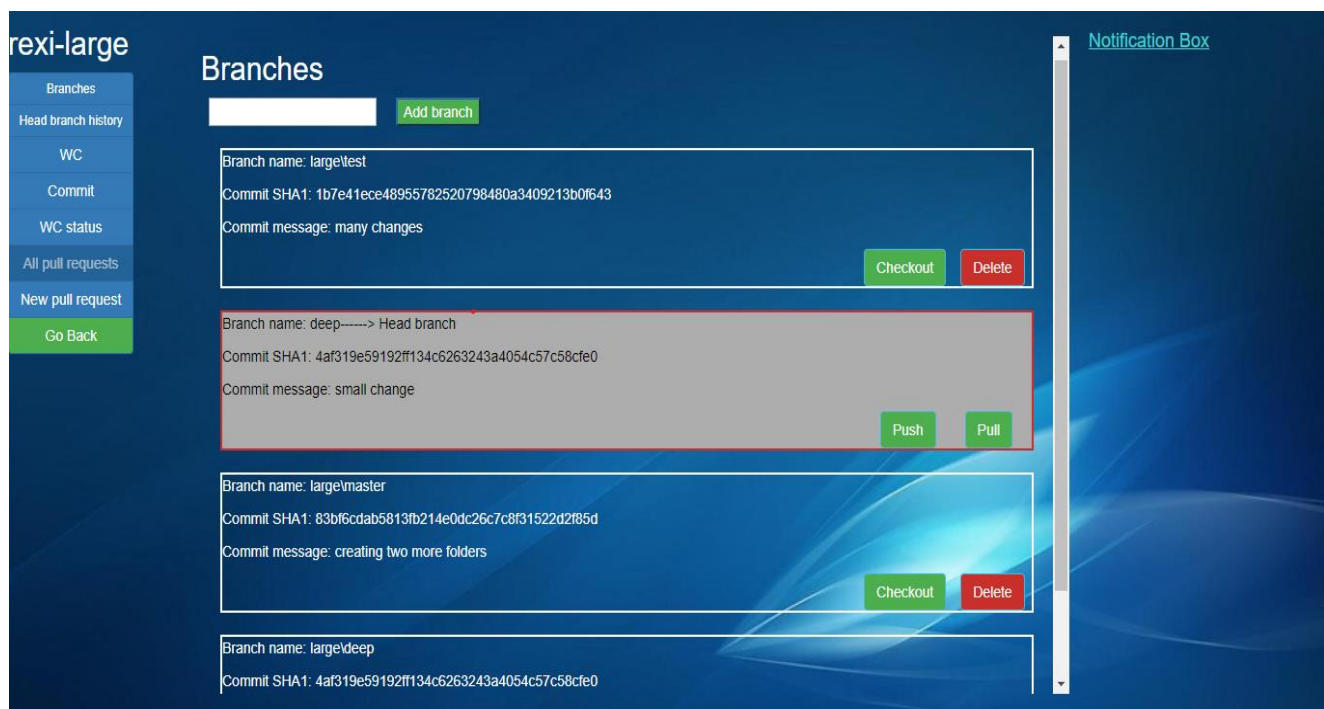
2. ביצוע Fork ממשתמש אחר: במסך מספר 2, מיד לאחר שנכנס למערכת, ישנה בצד ימין רשימת כל המשתמשים ב MAGitHub. ימצא המשתמש הנוכחי ברשימה זו את המשתמש ממנו הוא רוצה לבצע fork, ויקיש עליו.  
לאחר מכן יפתח מסך עם רשימת כל הרפוזיטורים של אותו משתמש נבחר. יבחר המשתמש את הרפוזיטורי בו הוא חפץ, ילחץ על כפתור fork שקיים על ידו. וזהו. הרפוזיטורי יופיע ברשימת הרפוזיטורים של המשתמש הנוכחי.

## ביצוע Push, Checkout, Pull ויצירת Branch חדש (לוקאלי או RTB) :

על מנת לבצע את הפעולות הנ"ל, במסך 3, לאחר בחירת רפוזיטורי לניהול, המשתמש יבחר בלשונית הפעולות משמאל Branches. לאחר מכן יראה על המסך את רשימת כל הבראנצ'ים הקיימים במערכת. במידה והבראנץ' איננו Head Branch יראה המשתמש ליד בראנץ' זה כפתור Checkout.

פעולות Push ו-Pull רלוונטיות רק עבור רפוזיטורים שנוצרו כתוצאה מפעולות Fork, לכן על מנת לבצע פעולות אלה, על משתמש להיכנס לרפוזיטורי מפורקק, לאחר מכן לבצע Checkout ל-RTB כלשהו. לאחר מכן יראה ליד אותו RTB כפתור של Pull או Push על מנת לדלות מידע חדש מהרפוזיטורי המקורי עבור ה-RTB המוצבע או לדחוף מידע חדש ללרפוזיטורי המקורי עבור ה-RTB המוצבע:

כפי שניתן לראות ה-Branch האקטיבי, Deep, אשר הינו RTB מצוייד בשני כפתורים Push ו-Pull.

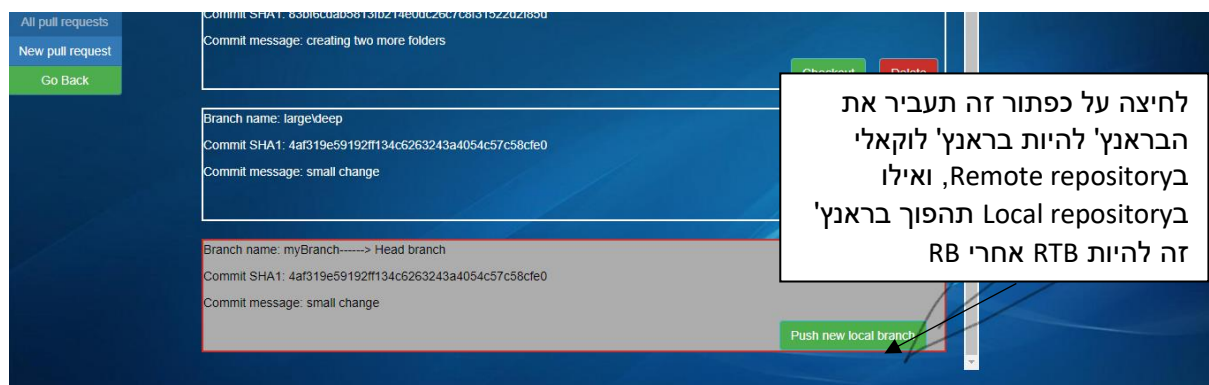


על מנת לבצע Push new local branch שזוהי הדרישה החדשה של התרגיל השלישי, על המשתמש ליצור בראנץ' לוקאלי חדש בתיבת הטקסט למעלה, לאחר לחיצה על הלשונית Branches, ואז ללחוץ על Add branch.

לאחר מכן, יופיע הBranch ברשימת הBranchים של המשתמש. במידה ויבצע Checkout לBranch זה אזי יהיה לידו כפתור של Push new local branch:branch



כאשר ימצא לנכון, לאחר שעבד על בראנץ' זה (או שלא), יוכל המשתמש, לאחר ביצוע Checkout לBranch זה, לבצע Push new local branch:



יצירת RTB תתבצע באמצעות ביצוע Checkout לRB, לאחר ביצוע פעולה זו המערכת תודיע כי לא ניתן לבצע Checkout לRB אך ניתן ליצר עבורו RTB. במידה וישיב המשתמש להודעה זו כי אכן ברצונו ליצר RTB חדש, יוצר במערכת RTB אותו ניצן יהיה לראות נוסף ברשימת הBranchים ויהיה אפשר לבצע אליו Checkout.

## ביצוע Pull request:

משתמש אשר יצר רפוזיטורי מפורקק, יוכל לבצע Pull request. תהליך זה כולל דחיפת בראנץ' לוקאלי חדש לרפוזיטורי המקור. לאחר מכן לחיצה על כפתור New pull request. החלון הקופץ הבא יפתח:

Branch name: large/myBranch  
Commit SHA1: 167c960bf4465227d554dd1ddc25f21e4eb584c8

**New pull request**

Target branch:  
myBranch

Base branch:  
deep

Message  
hey! please accept this request!

Send

Branch name: myBranch-----> Head branch  
Commit SHA1: 167c960bf4465227d554dd1ddc25f21e4eb584c8  
Commit message: amiram

בTarget יבחר השתמש את הבראנץ' אותו הוא רוצה לדחוף, בBase יבחר את הבראנץ' בRemote repository שממנו יצא הTarget. בראנץ' זה יבצע מיזוג עם בראנץ' הTarget ומשתמש הרפוזיטורי המקורי יקבל את בקשת הPull request. בנוסף יוכל לשלוח Message על מהות ה Pull request ובסיום כל הנ"ל ילחץ על Send. המשתמש בRemote Repository יקבל התראה כי מחכה לו בקשת Pull request חדשה. המשתמש יכנס ללשונית All pull requests שם יראה את Pull request החדש, ואת כל היתר הקודמים

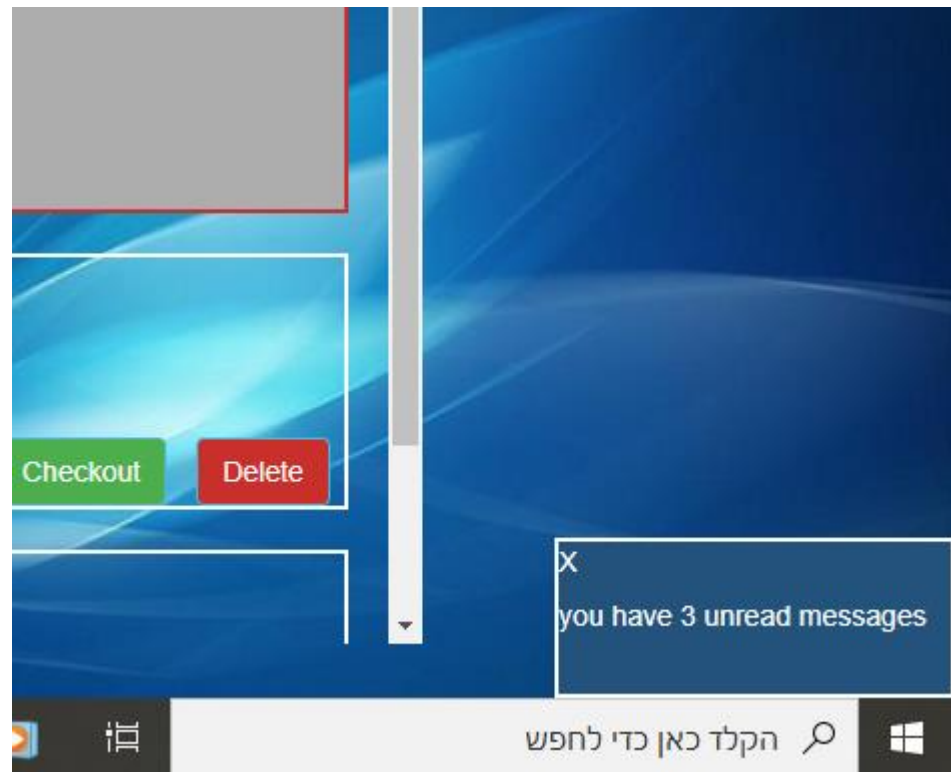
| base | target   | message                          | sender | sent at                  | request status |                  |
|------|----------|----------------------------------|--------|--------------------------|----------------|------------------|
| deep | myBranch | hey! please accept this request! | nisim  | Oct 28, 2019 11:36:23 PM | OPEN           | Accept Deny Exam |
| deep | momo     | regal                            | nisim  | Oct 28, 2019 11:37:13 PM | DENIED         |                  |

כפי שניתן לראות אכן מופיעה הבקשה האחרונה, היא איננה נענתה ללא נענתה לכן יש לצידה את הכפתורים Accept- על מנת לקבל הבקשה, Deny- על מנת לדחות את הבקשה Exam- על מנת לבחון את הבראנץ' החדש שנדחף. לחיצה Exam תוביל להצגת הבראנץ', לחיצה על קומיט מקומיטי הבראנץ' תפתח את קובצי הקומיט, לחיצה על כל קובץ בקומיט תראה את תוכנו. כמו כן קיימת אפשרות לאגרגציה המפורטת בתחתית המסמך תחת הכותרת "בונוסים שמומשו".

בנוסף, דחיה או קבלה של Pull request תלווה בעדכון של שולח ה Pull request. באם הפעולה שנבחרה בנוגע ל Pull request הייתה דחיה, אזי שאז יתבקש דוחה הבקשה להסביר לשולח הבקשה מדוע לא קיבל אותה.

## :Notifications

מערכת הנוטיפיקציות שלנו תוכננה קצת אחרת מדרישות התרגיל. שמנו לב רק לקראת ההגשה שאין אנחנו בדיוק עומדים בדרישה של ניקוי הנוטיפיקציות בעת התנתקות, ולאחר מכן בהתחברות יוותרו רק הנוטיפיקציות אותן טרם ראה המשתמש. אולם, לדעתנו מערכת הנוטיפיקציות שלנו מהווה הגדלת ראש. כל נוטיפיקציה שמקבל משתמש (משום שהעתיקו ממנו רפוזיטורי, שלחו לו Pull request, דחו/קיבלו Pull request ששלח וכו') תקפיץ חלון מצד ימין למטה אשר יתריע על קבלת הודעה חדשה שטרם נקראה. במידה ויש מספר רב של הודעות שלא נקראו החלון יציג במדויק את מספר ההודעות שלא נקראו. למשל, נניח כי למשתמש מסויים יש 3 הודעות במערכת אשר שתיים מהן לא נקראו, כעת מקבל המשתמש הודעה חדשה, כלומר סך הכל יהיו לו 4 הודעות אשר 3 מהן הן הודעות שלא נקראו. לכן יראה את החלון הקופץ הבא מצד ימין למטה עת יקבל את ההודעה הרביעית:

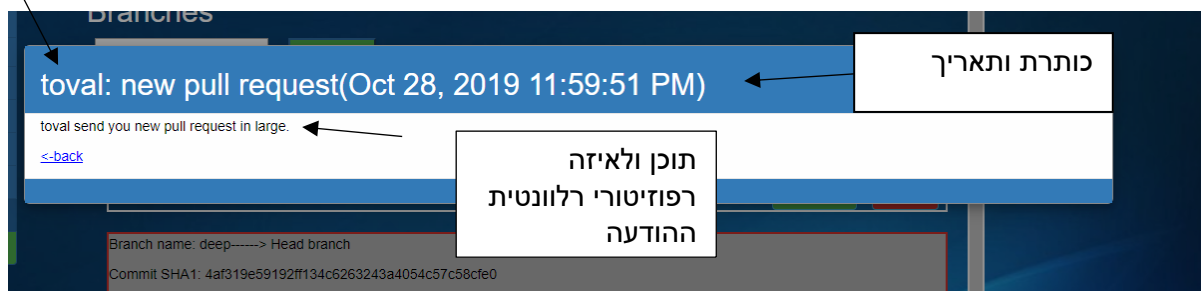


כעת, כאשר יפתח את תיבת הנוטיפיקציות (מצד ימין למעלה) יראה את ההודעות שקיבל כך שאלו הצבועות בכחול הינן ההודעות שטרם נקראו (כלומר המשתמש לא פתח אותן) ואלו שאינן צבועות אלהו הודעות שנקראו משכבר:



מוען

לחיצה על הודעה כלשהי תפתח אותה ותראה את תוכנה:

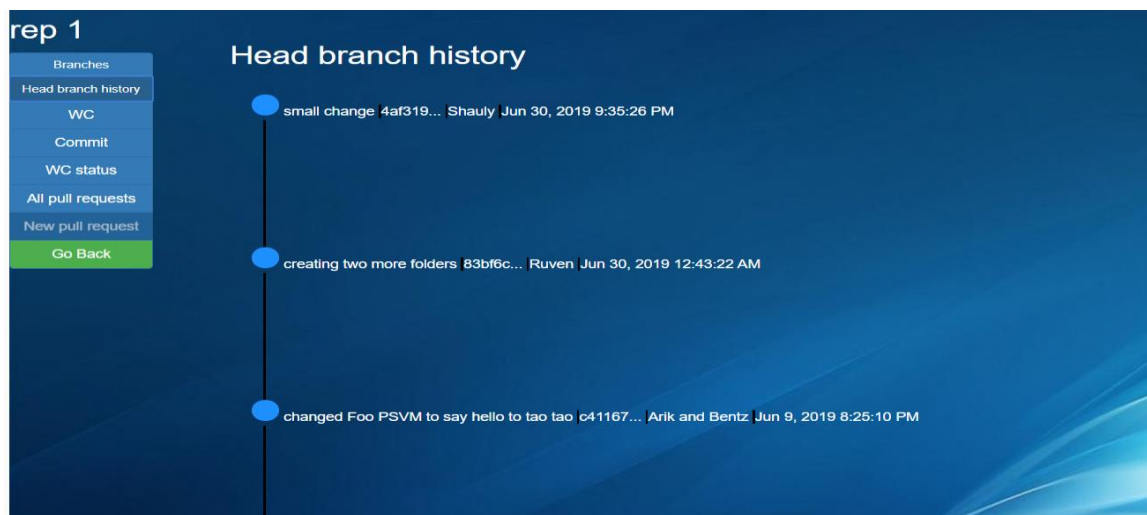


אם ההודעה היא הודעה שבטרם הפתיחה הייתה הודעה שסומנה שטרם נקראה אזי אין היא תסומן בכחול יותר, ולא תחשב להודעה שטרם נקראה. בMagithub כל ההודעות נשמרות, ובעצם תיבת הנוטיפקציות מתנהלת כמו מייל, עם התראות, כותרת לכל הודעה, תוכן, מוען, נמען והיכולת לדעת האם הודעה נקראה או לא נקראה, ומהו מספר ההודעות שלא נקראו.

הנוטיפקציות יקפצו, וניתן יהיה לפתוח את תיבת ההודעות בעמוד 2 וגם בעמוד מס' 3.

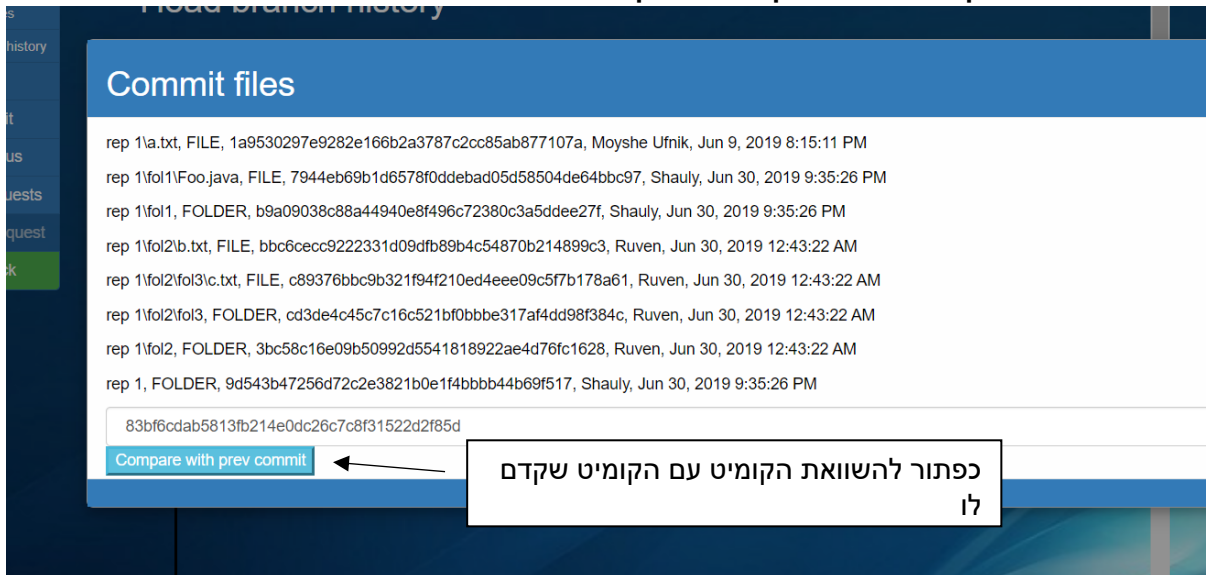
## היסטוריית HeadBranch:

המשתמש יוכל לראות את היסטוריית ההד בראנץ' על ידי לחיצה על הכפתור בלשונית משמאל Head branch history. יפתח עבורו החלון הבא:





לחיצה על כל קומיט תאפשר למשתמש לראות את תוכן הקומיט, כלומר לראות את כל קבצי, עבור כל קובץ יוכל המשתמש לראות את תוכנו, וגם להשוות את הקומיט עם הקומיט שקדם לו:



Commit files

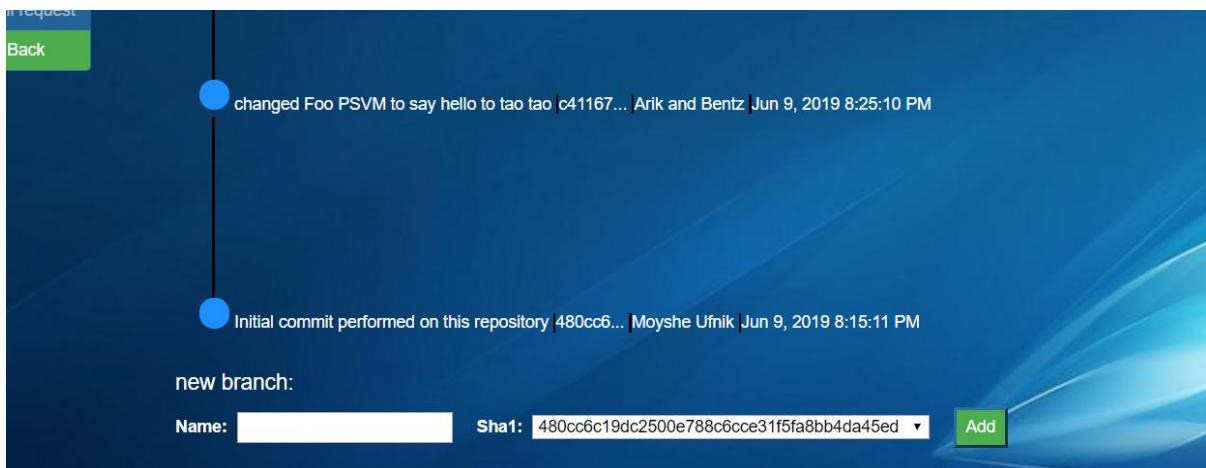
- rep 1\1a.txt, FILE, 1a9530297e9282e166b2a3787c2cc85ab877107a, Moyshe Ufnik, Jun 9, 2019 8:15:11 PM
- rep 1\fol1\Foo.java, FILE, 7944eb69b1d6578f0ddebada05d58504de64bbc97, Shauly, Jun 30, 2019 9:35:26 PM
- rep 1\fol1, FOLDER, b9a09038c88a44940e8f496c72380c3a5dde27f, Shauly, Jun 30, 2019 9:35:26 PM
- rep 1\fol2\1b.txt, FILE, bbc6cecc9222331d09dfb89b4c54870b214899c3, Ruven, Jun 30, 2019 12:43:22 AM
- rep 1\fol2\fol3\c.txt, FILE, c89376bbc9b321f94f210ed4eee09c5f7b178a61, Ruven, Jun 30, 2019 12:43:22 AM
- rep 1\fol2\fol3, FOLDER, cd3de4c45c7c16c521bf0bbbe317af4dd98f384c, Ruven, Jun 30, 2019 12:43:22 AM
- rep 1\fol2, FOLDER, 3bc58c16e09b50992d5541818922ae4d76fc1628, Ruven, Jun 30, 2019 12:43:22 AM
- rep 1, FOLDER, 9d543b47256d72c2e3821b0e1f4bbbb44b69f517, Shauly, Jun 30, 2019 9:35:26 PM

83bf6cdab5813fb214e0dc26c7c8f31522d2f85d

Compare with prev commit

כפתור להשוואת הקומיט עם הקומיט שקדם לו

כמו כן, בתחתית עמוד היסטוריית הHeadBranch תתאפשר למשתמש הוספה של בראנץ' חדש על אחד מקומיטי הבראנץ':



Back

- changed Foo PSVM to say hello to tao tao c41167... Arik and Bentz Jun 9, 2019 8:25:10 PM
- Initial commit performed on this repository / 480cc6... Moyshe Ufnik Jun 9, 2019 8:15:11 PM

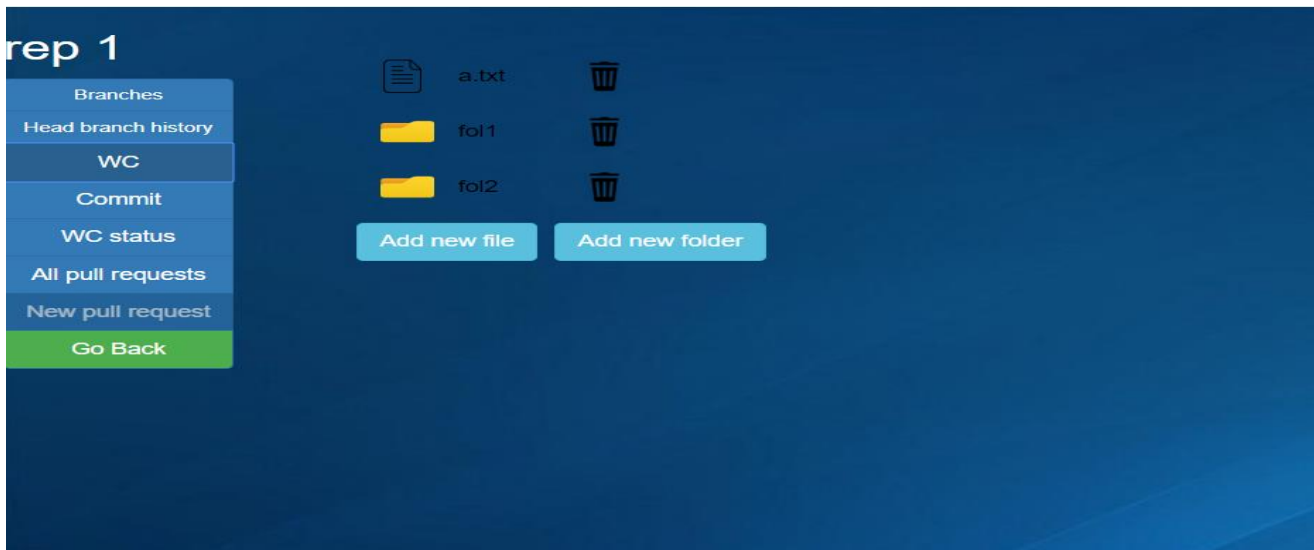
new branch:

Name:  Sha1: 480cc6c19dc2500e788c6cce31f5fa8bb4da45ed

## בונוסים שמומשו:

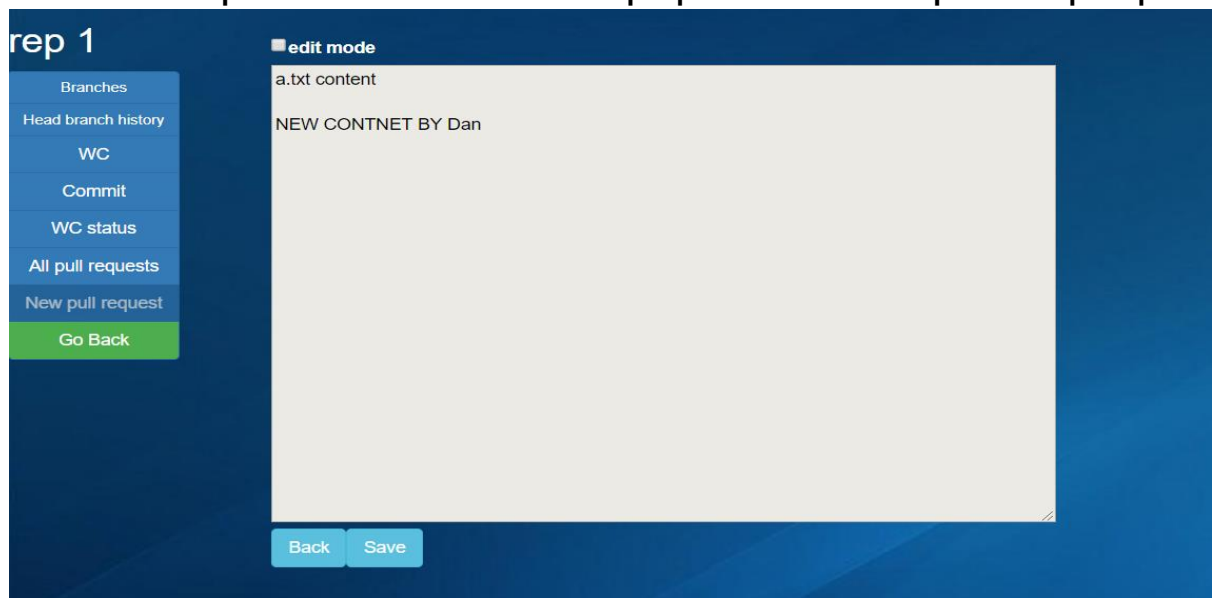
1. תצוגת מבנה רפוזיטורי בצורה מסודרת ונורמלית, בצורת עץ כך שניתן לערוך, למחוק ולהוסיף קובץ חדש – בלחיצה על כפתור WC תופיע ה root folder של הרפוזיטורי ותכולתה. משם אפשר להיכנס לכל תיקייה ולראות

את תכולתה או לקובץ ולראות את תוכנו (על מנת להיכנס יש ללחוץ על שם הקובץ). ניתן כמובן למחוק, להוסיף או לערוך קבצים.



ניתן מכאן למחוק קובץ ולאז תיקייה, להוסיף קובץ ולאז תיקייה.

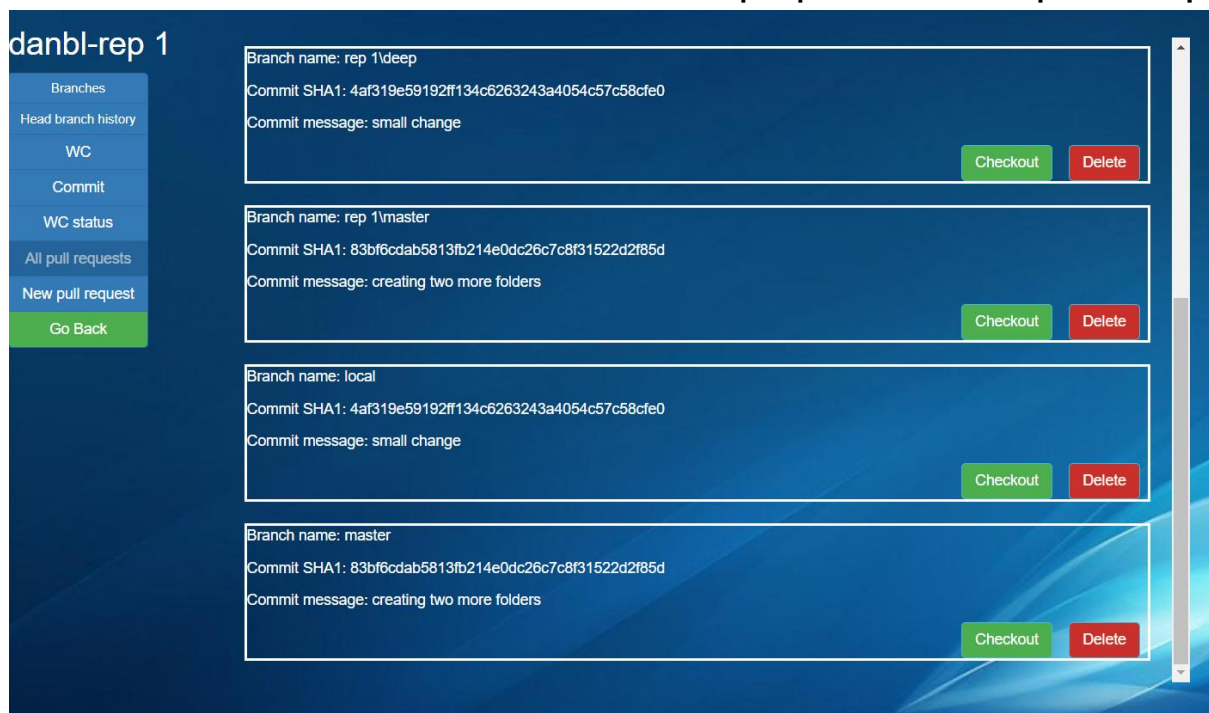
ולאחר לחיצה על קובץ ייפתח תוכנו ומשם ניתן לערוך אותו ע"י לחיצת על הצ'ק בוקס ולבסוף לשמור את הקובץ החדש ולחזור חזרה לעץ.



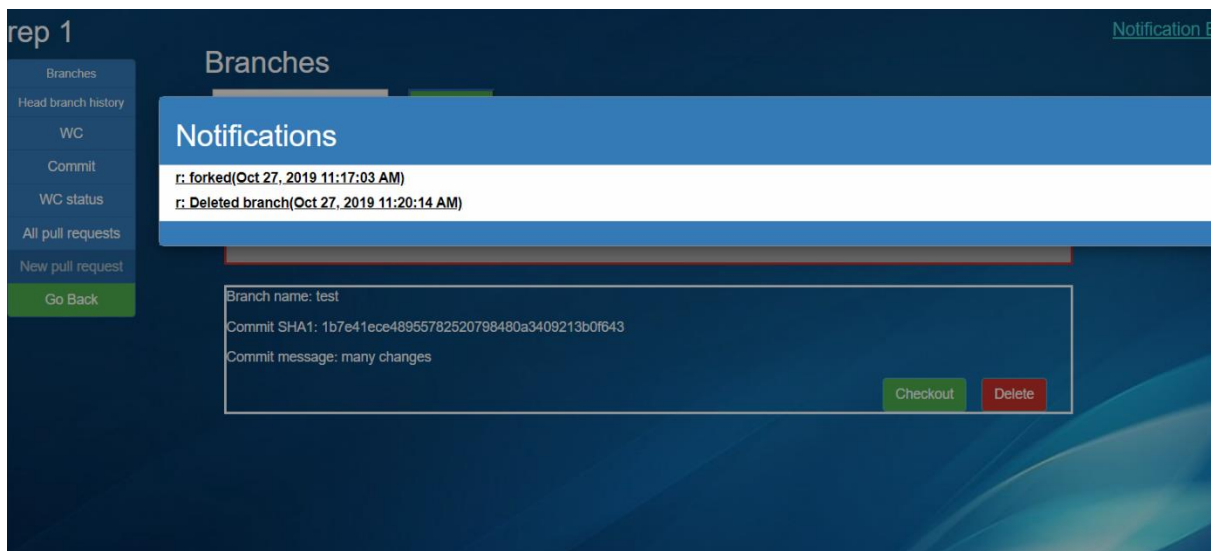
2. מחיקת בראנץ' – בלחיצה על כפתור Branches יופיעו כל הבראנצ'ים של הרפוזיטורי, ומלבד ה HEAD Branch ישנה אפשרות למחוק בראנצ'ים ע"י כפתור delete הנמצא בכל בלוק של בראנץ'. עבור remote branches

תיזרק שגיאה כי לא ניתן למחוק בראנצ'ים מסוג זה, עבור local branches יימחק רק הבראנץ' הנמצא ברפוזיטורי שלי ועבור rtb יימחקו אצלי ה RTB, RB ואצל RR יימחק כמובן הבראנץ' הלוקלי וכמובן שיקבל גם התראה על כך.

כאן ניתן לשים לב כי master הינו RTB ולכן ע"י לחיצה על delete הוא יימחק וגם rep1/master יימחק וכמובן שהבראנץ' master בRR יימחק גם כן. הבראנץ' "local" יימחק רק ברפוזיטורי הנוכחי.

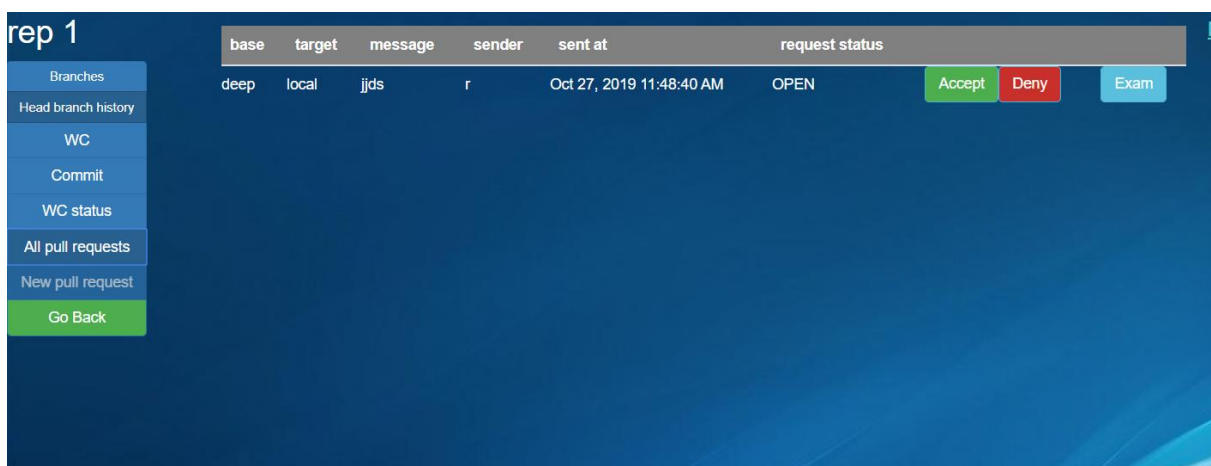


וכך ב RR תראה התראה על מחיקת הבראנץ' וכמובן שאותו בראנץ' כבר לא יופיע ברשימת הבראנצ'ים.

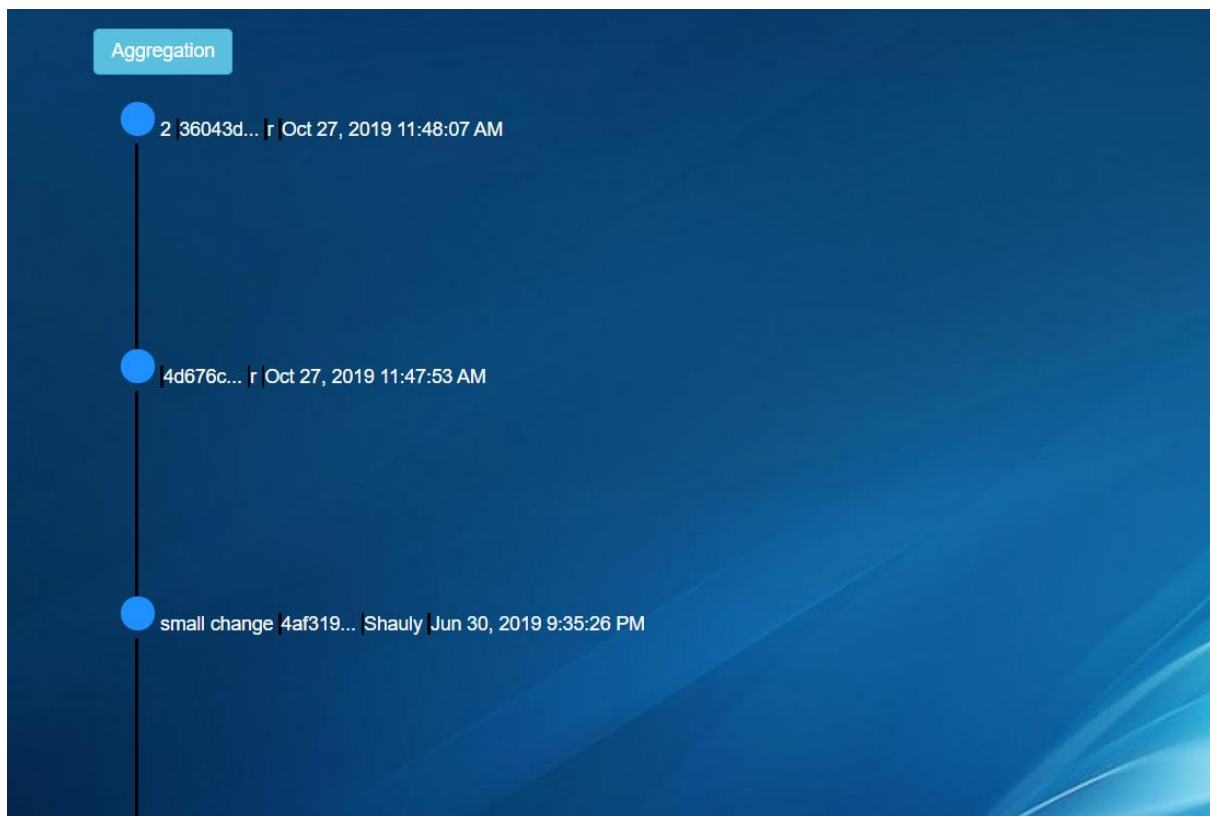


3. אגריגציה של סך השינויים ב-PR – עבור רפוזיטורי שפירקו אותו יהיה ניתן לראות את כל בקשות ה-PR ע"י לחיצה על כפתור "all pull requests" ועבור PR פתוחים יהיה ניתן לבחון את השינויים ע"י לחיצה של כפתור "exam" ולראות את השינויים עבור כל קומיט וקומיט. אך במעלה הדף יהיה ניתן ללחוץ על כפתור "aggregation" שהוא מראה את סך השינויים למיניהם בין ברנאץ' המטרה לברנאץ' הבסיס.

לאחר יצירת PR ב-LR יוכל משתמש ה-RR להיכנס לכל בקשות ה-PR ששלחו לו ( הוא גם מקבל התראה על כך) וכך זה יופיע לו:



לאחר לחיצה על כפתור "exam" יופיע לו שרשרת הקומיטים ויוכל ללחוץ על כל קומיט וקומיט ובעצם לראות את הקבצים שבו. וכך זה יראה לאחר לחיצה על "exam":



לחיצה על כל קומיט תציג את קבצי הקומיט וניתן להשוות שינויים אל מול כל קומיט אחר ע"י בחירה בקומבו בוקס ובפרט עבור הקומיט המוצבע ע"י ה base. ולאחר לחיצה על aggregation יפתח חלון ובו סך השינויים של כל הדלתאות מבראנץ' המטרה לבראנץ' הבסיס.

### Two commits comparison

added:

- none

changed:

- rep 1

deleted:

- rep 1\1a.txt
- rep 1\fol1\Foo.java
- rep 1\fol1