



תיק משימה – גלריה גרסה 1.0.1V

רקע ותכולת עבודה לגרסה

גרסה 1.0.0 של האפליקציה "גלריה" היתה הצלחה אדירה והיא הפכה להיות אחת מהאפליקציות המובילות בתחום ניהול תמונות ואלבומים. לאור ההצלחה החליטה החברה להמשיך ולהשקיע בפיתוח האפליקציה על מנת ליצור גרסאות המשך, כך שיענו לדרישות שמגיעות מהמשתמשים.

על מנת שנוכל לשחרר לשוק את גרסה 1.0.1 שתכיל שיפורי תשתית ותיקון באגים אנו צריכים את עזרתכם כך שנוכל לשחרר אותה למשתמשים במהירות האפשרית עם התוספות הבאות:

1. יצירת תשתית של ניהול גרסאות עבור הפרויקט.
 2. הוספת פיצ'ר הדפסת פרטי מערכת במסך פתיחה.
 3. תיקון באגים שעלו מהמשתמשים:
 - a. השלמת מידע חסר בפקודה List Album.
 - b. פקודת Remove User אינה עובדת בצורה תקינה.
 - c. השלמת מידע חסר בפקודה User Statistics.
- פרוט על הבאגים כפי שהגיעו מהמשתמשים תופיע בהמשך המסמך.

שלבי עבודה

1. **משימת לימוד עצמי 1:** קראו את משימת **לימוד עצמי 1 – מבוא לגיט**. זהו מסמך חזרה על החומר שכבר למדנו בסמסטר א', והוא נועד לעשות לכם קצת יותר סדר ולרענן את מה שכבר הספקנו לשכוח.
2. **משימת לימוד עצמי 2:** כדי שנוכל לעבוד על פרויקט בזוגות (כמו שנעשה בהמשך הסמסטר) נרצה ללמוד איך עובדים עם ענפים ב-Git. בנוסף, גם אם לא עובדים בזוגות, עדיין חשוב להבין איך לנהל את הפיתוח נכון ולחלק את הפיתוח לענפים. קראו את משימת **לימוד עצמי 2 – עבודה עם ענפים**, ובצעו את המשימות הרשומות במסמך. שימו לב – אין צורך להגיש את המשימה הזו, אבל מאוד מומלץ לעשות אותה כדי להבין איך להתנהל ב-GIT בפרויקט הגלריה.
3. **משימת תשתית:** העלאת הפרויקט ל-GitLab. היעזרו ב-**נספח א'** כדי להעלות את הפרויקט.
4. **משימת לימוד עצמי 3:** קראו את מסמך **לימוד עצמי 3 – ניהול משימות** ובצעו את המשימות שלו.

5. **משימת תשתית:** הוסיפו את כל המשימות (issues) שנפתור בגרסה הנוכחית לתוך ה-GitLab וקבעו יחד עם המדריך את תאריך הסיום עבורן:

- a. הדפסת פרטי מערכת במסך פתיחה.
- b. תיקון באג: האופציה 'רשימת אלבומים' מדפיסה את רשימת האלבומים עם הפרטים שלהם ללא 'זמן יצירה'.
- c. תיקון באג: באופציה של הדפסת סטטיסטיקה לא מודפס כמה אלבומים בבעלות המשתמש.

6. **משימת פיתוח:** פיצ'ר הדפסת פרטי מערכת במסך פתיחה. פרוט עבודה - [בנספח ב'](#). וודאו לסגור כל issue בסיום.

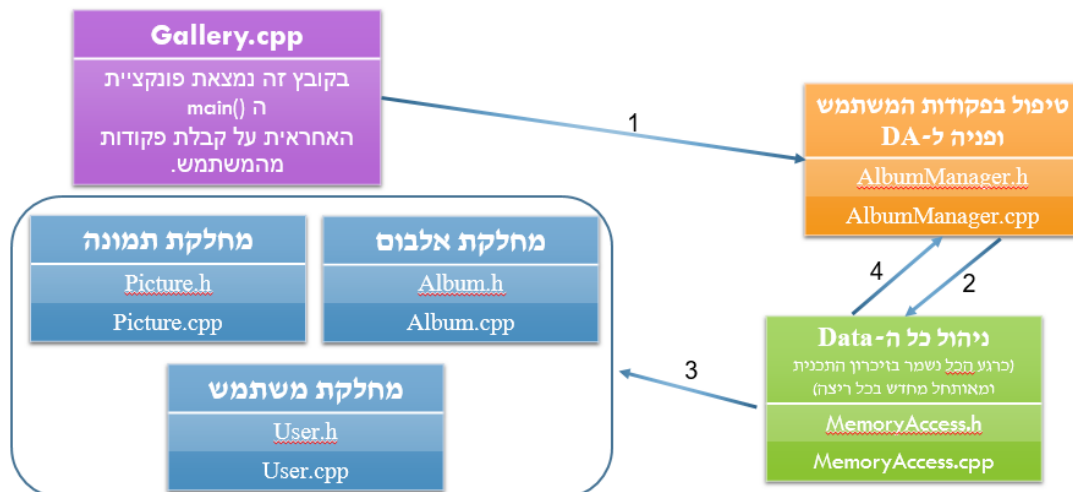
7. **משימת פיתוח:** תיקון באגים. פרוט עבודה - [בנספח ג'](#). וודאו לסגור את כל ה-issues בסיום.

שימו לב! מעתה ההגשות לא ייעשו דרך ה-NEO אלא כמו בעולם האמיתי הם יעשו דרך מעקב ה-commit-ים שלכם (למעט במקרים בהם יצוין אחרת).

ה- Design של הגרסה

בגרסה זו ה-**design** המקורי של הפרויקט לא משתנה אבל הוא חשוב מאוד להבנה על מנת לבצע את משימות הפיתוח הנדרשות.

עברו על קוד הפרויקט וודאו שאתם מבינים את הקשרים בין המחלקות כפי שמתואר בתרשים הבא:



נספחים

נספח א': משימת תשתית – העלאת הפרויקט ל-Git

החלק הבא רלבנטי רק אם המדריך/ה לא פתח/ה לנו פרויקט ב-GIT.

במשימה זו תיצרו **Repository** לפרויקט הגלריה ותעבדו עליו ממש כפי שעובדים בתעשיית ההייטק על פרויקטי תוכנה. לאחר יצירת ה-repository, תוסיפו פיצ'ר חדש ותתקנו באגים שמשתמשים דיווחו עליהם. כל זאת, בעזרת שימוש נכון ב-Git, כפי שלמדתם בשיעור וכפי שיוסבר לכם במטלה.

חלק א' – יצירת Group

צרו Group משותף לכם ולמדריך (דרך האתר Gitlab), קראו לו:

`[full_name]_advanced_programming`

חלק ב' – init

1. צרו Repository חדש פרטי באתר GitLab (תחת ה-Team שלכם עם המדריך) בשם `gallery`.
2. הוסיפו קובץ `.gitignore` ל-Repo בעזרת הכפתור `[+]` באתר GitLab. תוכלו להשתמש בקובץ הזה: <https://goo.gl/4gvKwR>. קובץ זה מגדיר את רשימת הקבצים שעל git להתעלם מהם, קבצים ש-Visual Studio יוצר שאינם חשובים לעבודה על הקוד של הפרויקט (למשל קבצי ה-build/debug, קבצים זמניים ועוד הרבה אחרים), כלומר לא לעקוב אחרי שינויים בהם ולא להעלות אותם אל ה-Repository המרוחק.
3. כעת עשו `git clone` ל-Repo שלכם אל תוך תיקייה על המחשב המקומי.
4. ב-repo המקומי - צרו branch בשם `develop` ועברו אליו.
5. קוד הבסיס של פרויקט הגלריה כבר מוכן לכם בגרסה `V1.0.0`, הורידו אותו והעתיקו את הקבצים אל ה-repo המקומי שיצרתם.
6. שמרו את השינויים שעשיתם בעזרת `add & commit`. בתיאור של ה-commit תוכלו לרשום: `"Initial commit with version 1.0.0"`.
7. דחפו אל ה-Repo המרוחק את התוכן של ה-Repo המקומי (כפי שלמדנו במסמך "לימוד עצמי 2 – עבודה משותפת").

[חזרו למעלה](#)

נספח ב': משימת פיתוח – פיצ'ר הדפסת פרטי מערכת במסך פתיחה

1. בחלק זה תוסיפו לפרויקט פיצ'ר חדש ופשוט, הדפסת הודעה למשתמש שמכילה את השם שלכם כמפתחים של הפרויקט + התאריך והשעה הנוכחיים.
2. צרו branch חדש (חשבו על שם משמעותי, הקפידו על הקונבנציות שלמדנו בתרגיל כיתה), לאחר היצירה עברו אליו.
3. הוסיפו הדפסה למסך של השם שלכם כמפתחים של הפרויקט + התאריך והשעה הנוכחיים בתחילת ריצת התכנית. תחילה עשו זאת בתוך ה-main. בצעו commit.
4. הפרידו את ההדפסה שהוספתם לפונקציה נפרדת. בצעו commit.
5. עדכנו את ה-branch ב-Repository המקומי ודחפו את השינויים אל ה-Repository המרוחק.
6. כנסו לrepo- באתר GitLab ובצעו Merge Request מה branch החדש אל תוך develop.
הערות:
 - a. אל תמחקו את ה-branch.
 - b. הוסיפו את המדריך/ה כ-Assignee (ע"מ לאפשר בדיקה של התרגיל ומתן פידבק).
 - c. במידה ואתם לא זוכרים כיצד לבצע, חזרו מסמך לימוד עצמי 2 – עבודה משותפת.

[חזרו למעלה](#)

בפרויקט גלריה ישנם מספר באגים שדיווחו עליהם המשתמשים. תיקון הבאגים הינה הזדמנות מעולה להכיר את הפרויקט טוב יותר ולהמשך העבודה על הגרסאות הבאות של הפרויקט.

צרו branch יחיד עבור כל הבאגים (חשבו על שם משמעותי, הקפידו על הקונבנציות שנלמדו בתרגיל כיתה) – שם ה-branch אמור להיות עם התחילית Bugfix/. בצעו commit לאחר תיקון של כל באג:

1. הפקודה List Album מדפיסה את רשימת האלבומים עם הפרטים שלהם. אולם המפתחים שכחו להוסיף הדפסת "Creation Time" עבור כל אלבום.

2. הפקודה Remove User אמורה להסיר משתמש מהמערכת. אולם היא לא פועלת כראוי (למשל, הריצו את הפקודה List Album וזהו התנהגות מוזרה).

הערה: ישנם מספר מקרי קצה – נסו לגלות את כולם.

3. הפקודה User Statistics מדפיסה סטטיסטיקות על משתמש. כרגע היא מדפיסה את המידע הבא:

- מספר האלבומים שקיימת בהם תמונה שהמשתמש מתיוג בה.
- מספר התמונות שהמשתמש מתיוג בהם.
- ממוצע התיוגים לאלבום.

הפקודה הייתה אמורה בנוסף להדפיס גם את מספר האלבומים שהמשתמש הוא הבעלים.

4. כנסו לrepo- באתר GitLab ובצעו Merge Request מה-branch החדש אל תוך develop.
הערות:

- אל תמחקו את ה-branch.
- הוסיפו את המדריך/ה-Assignee (ע"מ לאפשר בדיקה של התרגיל ומתן פידבק).
- במידה ואתם לא זוכרים כיצד לבצע, חזרו מסמך לימוד עצמי 2 – עם ענפים.

[חזרו למעלה](#)

סיימתם? כל הכבוד! מוזמנים להריץ את הפקודה הבאה:

```
git clone https://magshimimit@bitbucket.org/magshimimit/git-specialist.git
```