



תיק משימה – גלריה גרסה v1.0.1

רקע ותכולת עבודה לגרסה

גרסה 1.0.0 של האפליקציה "גלריה" הייתה הצלחה אדירה והיא הפכה להיות אחת מהאפליקציות המובילות בתחום ניהול תמונות ואלבומים. לאור ההצלחה החליטה הנהלת 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/ן 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 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 בצורה משותפת 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 משותפת](#), התחלקו לזוגות, ובצעו את המשימות הרשומות במסמך ביחד עם חבר/ת הצוות שלכם. שימו לב – צריך להגיש את המשימה הזו.
3. **משימת תשתית:** המדריך/ה צריך/ה לפתוח פרויקט חדש שעליו נעבוד ב-3 השבועות הקרובות. המשימה הראשונה היא ליצור ענף פיתוח והעלאת קבצי גרסה v1.0.0 ל-GitLab. היעזרו ב-[נספח א'](#) כדי להבין מה מעלים ואיך נכון לעשות את זה.

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.

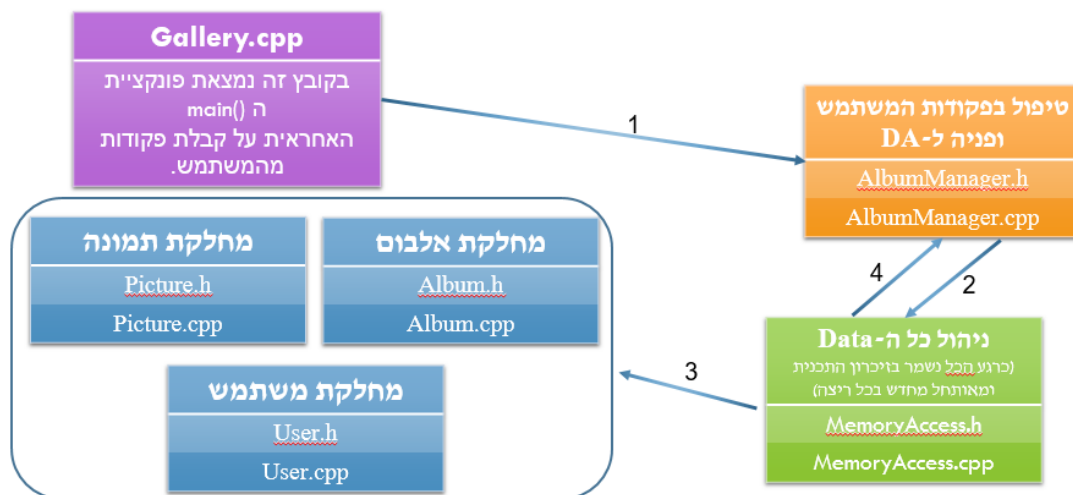
6. משימת פיתוח: פיצ'ר הדפסת פרטי מערכת במסך פתיחה. פרוט עבודה – [בנספח ב'](#). וודאו לסגור כל **issue** בסיום.

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

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 – יצירת ענף פיתוח והעלאת קבצי הבסיס

במשימה זו נבקש מהמדריך/ה שלנו ליצור **Repository** לפרויקט הגלריה. ולאחר מכן נעבוד עליו ממש כפי שעובדים בתעשיית ההייטק על פרויקטי תוכנה. נתחיל מלהוסיף את גרסה v1.0.0 (גרסה הבסיס), ואז נוסיף פיצ'ר חדש ונתקן באגים שמשתמשים דיווחו עליהם. כל זאת, בעזרת שימוש נכון ב-Git, כמו שלמדנו במשימות הלימוד

חלק א' – פנייה למדריך/ה לצורך יצירת ה-repo

לפני שמתחילים לעבוד, נפנה למדריך/ה שלנו ונבקש ממנו או ממנה ליצור פרויקט חדש עבור הגלריה. לפרויקט אמור להיות שם בפורמט הבא
Gallery_[First Name]_[Last Name]_[Year]

לדוגמא, נניח שקוראים לנו ישראל ישראלי ואנחנו ב-2022, שם הפרויקט שיווצר יהיה:

Gallery_Israel_Israeli_2022

אנחנו ניתוסף לפרויקט עם הרשאות מפתח/ת (Developer), ולמדריך תהיה הרשאה גבוהה יותר – Maintainer/Owner. בנוסף, נוכל לשים לב שאל הפרויקט התווסף גם המשתמש של קורס עקרונות.

זה מאוד דומה לאיך שעובדים בתעשייה או בצבא, בדרך כלל אנחנו נשתלב בפרויקט קיים ולא נקבל את ההרשאות הכי גבוהות. אחרי שאנחנו נכתוב קוד ראש הצוות שלנו יצטרך לאשר את הקוד שהעלינו ל-GIT.

חלק ב' – יצירת תשתית פיתוח

אחרי שהפרויקט נוצר סוף סוף אפשר לעבוד...
בחלק הזה נכין את תשתית הפיתוח שעל פיה נעבוד לאורך הפרויקט. אנחנו ניצור ענף פיתוח (Develop), ונעלה אליו את קבצי הבסיס של הפרויקט (Gallery V1.0.0).
אחר כך נפנה אל המדריך/ה כדי להפוך את הענף למוגן, ואנחנו נעבוד על תתי ענפים שיוצאים מענף הפיתוח.
כל פיצ'ר חדש, תיקון באג או patch שלנו ייכתב בענף חדש שיוצא מענף הפיתוח, וכשנסיים לעבוד נשלב את הענף החדש לענף הפיתוח. היחיד שיוכל לאשר את המיזוג בין הענפים יהיה המדריך או המדריכה.


בצעו את השלבים הבאים:

- 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.
- כעת עשו **git clone** מה-Repo אל תוך תיקייה על המחשב המקומי.

3. ב-repo המקומי - צרו branch בשם **Develop** ועברו אליו.
4. קוד הבסיס של פרויקט הגלריה כבר מוכן ב-drive בגרסה **V1.0.0**, ניתן לגשת אליו מה-NEO. הורידו אותו והעתיקו את הקבצים אל ה-repo המקומי שיצרתם.
5. שמרו את השינויים שעשיתם בעזרת `add & commit`. בתיאור של ה-commit תוכלו לרשום: **"Initial commit with version 1.0.0"**.
6. דחפו אל ה-Repo המרוחק את התוכן של ה-Repo המקומי (כפי שלמדנו במסמך "לימוד עצמי 2 – עבודה משותפת").
7. פנו למדריך/ה בפעם השנייה, הפעם כדי להפוך את ענף הפיתוח למוגן.


הנה דוגמא ל-repo שלנו בסוף שלב ב':




Develop gallery_israel_israeli_2022 / + History Find file Web IDE ↓ ↓ Clone



Added Gallery base files (V1.0.0)

Israel Israeli authored 3 hours ago

835fa41e 

Name	Last commit	Last update
 Gallery	Added Gallery base files (V1.0.0)	3 hours ago
 .gitignore	Added .gitignore	3 hours ago
 README.md	Initial commit	4 hours ago



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. שימו ♥, את הענף יש ליצור מתוך ענף הפיתוח (ולא מתוך הענף הראשי).
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 ל-repo באתר GitLab ובצעו **Merge Request** מה-branch החדש אל תוך הענף **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.



בפרויקט גלריה ישנם מספר באגים שדיווחו עליהם המשתמשים. תיקון הבאגים הינה הזדמנות מעולה להכיר את הפרויקט טוב יותר להמשך 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 והו 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 – עבודה משותפת.



סיימתם/ן? כל הכבוד! מזמנים להריץ את הפקודה הבאה:

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