# פרויקט גמר – סייבר ב'

* Implement Secure Code Scanner.
* Run Popular Static Coed Scanning tools ()
* Create a report on a server per project
* Each project will have a report
* Each report will show for each case what to fix
* Use LLM to recommend code fixes (such as. ChatGPT)

## GUI

* Use Streamlit (or NodeJS or CSS)

General views:

Project 1

Project 2

Project 3

…

|  |  |  |
| --- | --- | --- |
| fix | Bug | Tool |
| Fix suggested by LLM | Detected bug + Code | Detecting tool |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

מבנה העבודה:

1. מימוש כלים לניתוח סטטי, מתחילים מ-2. (Bandit & Pylint)

כל כלי לניתוח סטטי יממש את הממשק interface.StaticCodeAnylazer כך שיהיה ניתן לאגד את הכלים השונים.

1. מימוש UI בסיסי
2. לוודא שהפלט מכל כלי לניתוח קוד סטטי תקין ועובד.
3. יצירת קישור ל-LLM (chat.gpt) ע"י יצירת ./LLM/LLM.py אשר מקבל כקלט שם כלי לניתוח קוד סטטי ומחזיר תגובה לתיקון.

יש לשמור את התגובה לתיקון בקבצים ./LLM/Project\_name/Tool\_name

בשביל לשמור על שמות הקבצים ויכולת לזהותם. חשבנו על 2 אפשרויות:

1. שימוש במסד נתונים אשר מקדיש לכל פרויקט X וקובץ Y תעודת זהות ID כך ששם הקובץ המכיל את תוצאות הסריקה יהיה ID.txt
2. הפתרון השני ששם הקובץ המכיל את תוצאות הסריקה יהיה X-Y.txt. (לאחר החלפת כל התווים הלא חוקיים בשמות קבצים כגון '/')

הפתרון השני יותר פשוט ולכן נשתמש בו לבינתיים.

1. ניצור מערכת הלוקחת את כל הקבצים בתיקייה הראשית של הפרויקט ומוסיפה את כל הקבצים לפרויקט.

(יש לשים לב, כרגע המערכת סורקת רק קבצים עם סיומת .py)

1. הוספת פונקציונליות סריקת הקבצים דרך האתר.
2. הוספת דרך להצגת תוצאות הסריקה לכל קובץ.
3. לממש ignore