**פרויקט בסייבר**

**שם: אילן ישוק**

**ת"ז: 326102126**

**תאריך: 15 ביוני, 2022**

**כיתה: י"ב 6**

**מנחה: ברכה עסיס**

**חלופה: תכנון ותכנות מערכות מידע**

**שם הפרויקט: Safe Cloud**

**בי"ס: מקיף אמי"ת באר שבע**

**תוכן עניינים**

**מבוא לספר פרויקט**

**תיאור תכולת הפרויקט**

מסמך זה מאפיין, מפרט ומציג את התוצר הסופי אותו אני מגיש במסגרת פרויקט הגמר במגמת הנדסת תוכנה בכיתה י"ב. בנוסף לתוצר וליכולות שלו, הוא מציג אתגרים, דרכי התמודדות עמם, הסברים לדברים עקיפים בנושא ועוד מגוון רב של תכנים. מסמך זה מהווה את כל מה שצריך לדעת על הפרויקט, ויותר מכך.

**מוטיבציה לפרויקט**

כולם מכירים את גוגל דרייב. זו אחת מהתוכנות המשומשות בעולם. כל כך משומשת, עד שהיא שינתה את הדרך שבה אנו מעבירים קבצים ממחשב למחשב. אמנם גוגל דרייב מצויין בכל מידותיו, אבל אני תמיד רציתי משהו פשוט יותר, מהיר יותר, שניתן לבצע פעולות כמו הוספת/מחיקת/הורדת קבצים בלחיצה אחת.

אני לא מתיימר להיות גוגל דרייב או איי-קלאוד הבא ולא מנסה להתחרות בעשרות אלפי אנשים, אבל אני כן מציג דרך נוספת למשתמש, פשוטה ומהירה לשמירת קבצים בענן, בצורה הכי מאובטחת שיש (אם ירצה בזאת).

**הפרויקט**

הפרויקט עצמו הוא ענן, ששומר את הקבצים שלך במאגר נתונים הרחק ממך, ובצורה מאובטחת אם תבחר בכך, בשביל שתוכל להשתמש בקבצים האלו במקומות אחרים, במחשבים אחרים ובזמנים אחרים. אני ניסיתי בכוונה לעשות את הממשק הגרפי פשוט ומהיר ככל הניתן, אני לא אוהב להתעסק עם כל הפונקציונליות (שלדעתי מיותרת) בהרבה מהעננים שיש כיום ולכן ניסיתי לצמצם ככל הניתן במישור הזה.

בשביל להשתמש בענן אתה צריך משתמש. לאחר הרשמה שכוללת אימות דו-שלבי עם האימייל, אתה תגיע לאתר שיכיל את הקבצים שלך. אתה תוכל להוסיף קבצים ע"י גרירתם למסך, למחוק ולהוריד בלחיצה. ניתן להוסיף כמות רבה של קבצים בגרירה אחת. בנוסף, אתה יכול לבחור להוסיף ולהוריד קבצים בעזרת מפתח מיוחד. על המפתח אפרט בהמשך, אבל אציין שבעת הכנסת המפתח, המשתמש מוסיף עוד שכבת הגנה – קריטית וטובה מאוד – לקבציו המועברים באינטרנט אל הענן. חוץ מעמוד הקבצים, יש עמוד פרופיל שמציג את פרטיך ונותן לך להתנתק מהמשתמש אם תרצה ועמוד הסבר על הפונקציונליות של התוכנה.

השרת שלי זוכר משתמשים נכנסים ויוצאים, בצורה בטוחה, אצלו בדיסק. תודות לכך, משתמש לא צריך להכנס כל פעם מחדש לאתר, ולמלא את פרטיו. האתר זוכר אותו למשך זמן של יום אחד (שיש שיגידו קצר ויש שיגידו ארוך, אני מאמין שיום אחד זה כמות זמן טובה לנוחות וטובה לאבטחה). כמובן שהשרת לא יודע לא את סיסמת המשתמש, ואם המשתמש יבחר – גם לא את קבצי המשתמש.

**אתגרים שפגשתי במהלך הפרויקט**

פגשתי מס' לא מבוטל של אתגרים בזמן עשיית הפרויקט והנה כמה מהם:

- **אין זמן**. כיתה י"ב עמוסה גם ככה, בגרויות מפיזיקה, מתמטיקה, תנ"ך ולכן היה צריך לנהל את הזמן בצורה יעילה, יחד עם הצבת יעדים לסיום חלקים בעבודה.

- **ממשק גרפי**. כמעט ואף פעם לא עבדתי עם ממשק גרפי כמו שעבדתי בעבודה זו. הממשק הוא אתר אינטרנט, ולכן יצא לי לעבוד עם javascript הרבה, הרבה יותר ממה שידעתי או הכרתי. למדתי מהבסיס ביותר - כיצד javascript רץ ועד מטרותיו והדברים הקטנים והנוחים שהופכים את השפה הזו לכל כך משומשת בעולם.

- **תכנות אסינכרוני** - הרבה בעיות נוצרו עקב זה שjavascript היא שפה שעובדת רק בצורה אסינכרונית. כלומר, היה צריך לשנות את התפיסה שלי - שכל שורה שאני כותב, אני צריך להוסיף לה תכנותית, "לחכות" שהשורה תסתיים ורק אז לשלוח את זה. זה כולל שימוש בPromiseים, async וכו'.

- **אבטחה**. ניסיתי לשים דגש על האבטחה משתי סיבות. ראשית, זהו פרויקט בסייבר, כלומר צריך לגעת באבטחת מידע ורשתות. שנית, בן אדם שירצה לשלוח את הקבצים האלו, ירצה ברוב המקרים, שהם יישארו רק שלו ואף אחד, כולל אני (או מנהל מאגר הנתונים) לא ידע מה יש שם. האתגר הוא - כיצד לעשות את זה? מאיפה מתחילים בכלל?

- **הקלאסי**. לשבת על בעיה כמה שעות, שאין לך שמץ של מושג למה משהו לא עובד בכלל. אתן דוגמה על מה אני מדבר. כשאני מוסיף משתמש למאגר נתונים, אני עושה גיבוב של חלק מהנתונים שלו, בצורה כזו שיהיה קשה מאוד (כל כך שאפשר להגיד בלתי אפשרי) לחזור לנתוניו המקוריים (מהמחרוזת המגובבת). הבעיה שלי הייתה שמשום מה זה לא היה קבוע. בסופו של דבר, אחרי המון בדיקות הגעתי לפתרון והוא להוסיף סוגריים ל datetime.timestamp משום שזו פעולה. נשמע טיפשי נכון? אבל אז קפצה בעיה אחרת. עדיין הגיבוב לא היה קבוע. הפעם גיליתי, שאני שומר תאריך יצירת משתמש מדויק - עד לשניות במאגר הנתונים אבל כשהייתי עושה את הגיבוב (לשם השוואה לאחר הכנסת משתמש נכנס), התאריך היה מדויק לרמת המיקרו-שניות, ולכן בעת הוצאה ממאגר הנתונים וגיבוב המידע, לא הייתי מקבל שווי בין נתונים למרות שהיו שווים. בקיצור.... הקלאסי. וזה קרה פעמים רבות, אבל זה התהליך לתוצר הסופי וכמות הלמידה שניתן להוציא מנסיונות כאלו היא ללא הגבלה.

**קהל יעד**

קהל היעד של הפרויקט הוא בערך כל מי שמשתמש היום באינטרנט - כלומר כל אחד שרוצה את הקבצים שלו ביותר ממקום אחד. אנשים שעובדים עם חומר וצריכים אותו מסודר - רופאים, מורים, הייטק, פקידים, תלמידים ועוד...

**פיתוחים עתידיים**

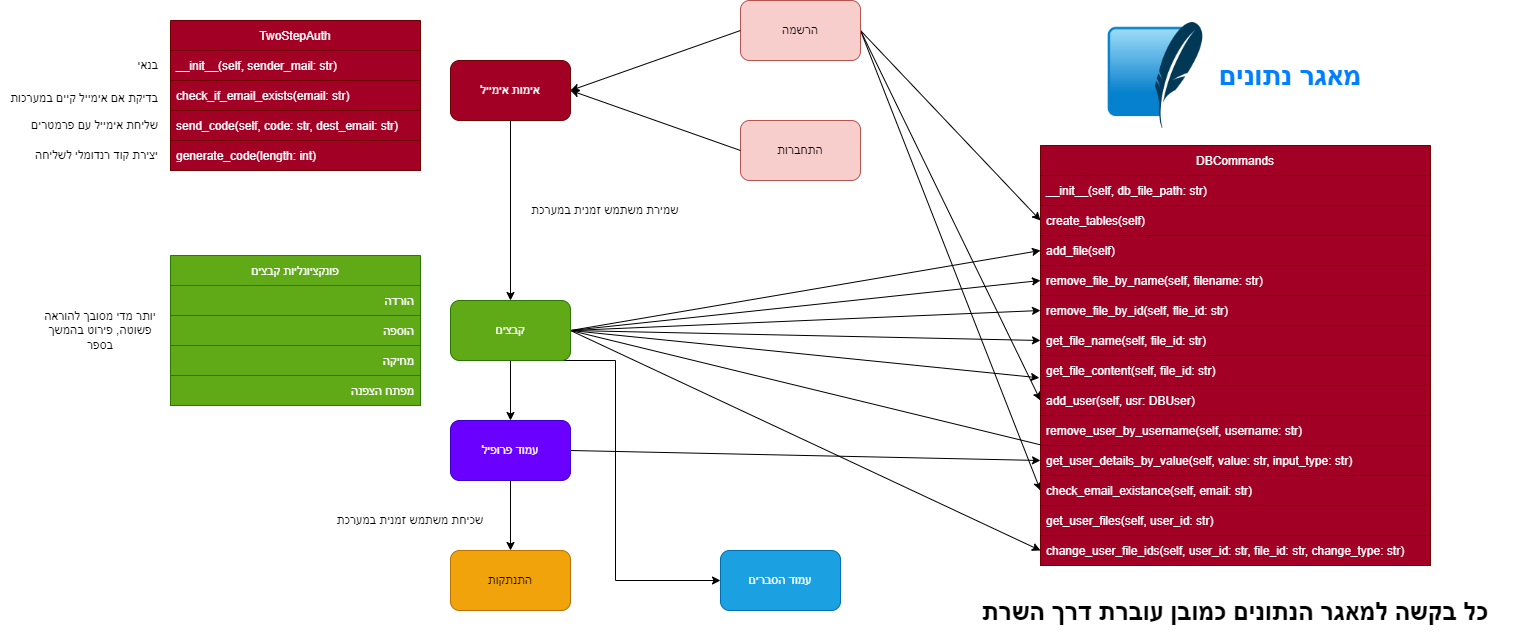
כמעט כל מה שתכננתי לפרויקט, בוצע. לכן, אין הרבה פערים שנוצרו אבל יש דברים שהיה נחמד להוסיף:

- שיתוף קבצים בין כמה אנשים, כלומר קישור קובץ מסוים ליותר מבן אדם אחד

- יצירת תיקיות שיכילו קבצים (כמובן לא באופן "זריקת תיקייה"... כי בדקתי... לא כל כך אפשרי)

**ארכיטקטורת הפרויקט**

ניתן לייצג את ארכיטקטורת הפרויקט באמצעות התרשים הבא:



ניתן לחלק את הפרויקט לכמה מישורי עבודה שונים:

- **ממשק גרפי**

- תהליך הרשמות משתמש חדש

- תהליך התחברות של משתמש קיים

- ניהול קבצים (הוספה/מחיקה/הורדה)

- **מאגר נתונים**:

- טבלת משתמשים

- טבלת קבצים

- **אבטחה**:

- אימות משתמש דו-שלבי

- הצפנת קבצים במאגר הנתונים

- גיבוב פרטיו של המשתמש ליצירת מחרוזת מזהה ייחודית

**בפירוט רק בסופו של דבר אגיע לממשק הגרפי משום שהוא המחבר של כל המישורים ולמען הבנתו יש להבין את שאר הפרויקט תחילה**

ובכן, נתחיל בהכי פשוט:

**מאגר נתונים**

**ממשק גרפי**

הענן מיוצג ע"י אתר. האתר מיוצר עם flask בשפת פייתון ומעוצב בשפות javascript, css, html. ישנה גם פונקציונליות רבה שנכתבה בjavascript, משום שזו הדרך היחידה לעדכן אתר בלייב (ללא רענון). הרי אף אחד לא רוצה לגרור כמה קבצים, שכל אחד ירענן את המסך. בדפים הבאים אני אציג את המסכים שהמשתמש יכול לפגוש ובעיקר