**חלוקה לאיטרציות (ספירנטים) – ImageToText:**

**איטרציה 1:**

**מטרה:** בניית אב טיפוס של שרת שמקבל תמונה ופועל לפי עיבוד תמונה, הוא מאפשר לערוך את התמונה, הוא בודק ומזהה את הצבע שחזר על עצמו הכי הרבה בתמונה, ומציג את התמונה לאחר העריכה ואת הצבע שחזר על עצמו הכי הרבה בתמונה.

**פיצ'רים באיטרציה 1:**

1. **קבלת תמונה לשרת** – השרת מקבל תמונה.

**2. עריכת התמונה: חיתוך התמונה -**  המשתמש יוכל לחתוך את התמונה, לבחור את הערך שהוא רוצה שיומר על ידי המכונה

**3. עריכת התמונה: סיבוב התמונה -** המשתמש יוכל לסובב את התמונה

**4. עריכת התמונה : הבהרת התמונה -**  המשתמש יוכל להבהיר את התמונה במידת הצורך.

**5. בדיקה וזיהוי הצבע שחוזר על עצמו הכי הרבה בתמונה –** השרת בודק מהו הצבע הכי נפוץ בתמונה.

**6. שליחת התמונה לשרת –** לאחר שהמשתמש ערך את התמונה ובחר את החלק אותו הוא רוצה להמיר, הוא שולח אותה לשרת שמכיל את המכונה.

**איטרציה 2 :**

**מטרה:** בניית אב טיפוס לאפליקציה מצד הקליינט עם GUI בסיסי שתומכת ויכולה לדבר עם השרת שכתבנו, להעביר לו תמונה (שניתן לערוך באפליקציה) , ולקבל ממנה את התמונה ערוכה. כמו כן, נחקור על מודל CNN וכיצד יש לכתוב את המודל לצורך למידת המכונה בספרינט הבא.

**פיצ'רים באיטרציה 2 :**

1. חקירת הספרייה react native.

2. התקנת סביבת העבודה והתקנת java script וnpm.

3. ביצוע דוגמה עם web application לצורך בדיקה וניסיון עם הקוד והאפליקציה.

4. בדיקה כיצד לבצע קישור של קובץ פייתון לreact native (מעין backend של הקוד בפייתון אל web application). התייעצות של שקד

5. בניית האתר עם ממשק gui דומה לקובץ הפייתון , ללא הפונקציות עובדות (רק כפתורים) , ניתן יהיה רק לפתוח את התמונה.

6. תקשורת של השרת עם הלקוח – שליחה לשרת וקבלת התמונה מצד השרת.

7. הוספת הbackend, פונקציות עריכת התמונה שלנו בתוך האפליקציה עצמה ולא בממשק gui של python.

**איטרציה 3:**

**מטרה:** לימוד המכונה את האותיות לפי הdataset, ובאמצעותה נשלח למכונה תמונה והיא תדע לזהות ולסווג את האות.

**פיצ'רים באיטרציה 3 :**

**1. חיפוש בתוך מאגר המידע של האותיות –** המכונה מחפשת במאגר המידע באיזה אות מדובר.

**2. זיהוי האותיות בתמונה שהתקבלה –** המכונה לומדת תבניות של אותיות בכך שעוברת על עשרות אלפי דוגמאות של אותיות, בונה לעצמה תבניות של אותיות ובכך מזהה את התו ומעביר את התו לשרת.

**3. סיווג האותיות בתמונה שהתקבלה –** לאחר חיפוש במאגר והבנה באיזה אות מדובר, היא מסווגת את האות (אומרת באיזה אות או תו מדובר).

**5. הפיכת הסיווג לפונט קריא –** האותיות שסווגו הופכות לפונט קריא שניתן להבין.

**איטרציה 4:**

**מטרה:** ממשיך עם עיבוד התמונה, מבצעים עוד מספר פיצ'רים לצורך זיהוי המכונה הלומדת את האותיות בתמונה , לבדוק את המכונה.

**פיצ'רים באיטרציה 4 :**

**1. דגימת התמונה כדי למצוא איזה גוון מבדיל בין הטקסט לרקע התמונה –** לאחר שהמשתמש שולח את התמונה למכונה, המכונה למעשה מבדילה בין הטקסט לבין רקע התמונה, שלב זה מבוצע בשביל שלב זיהוי האותיות בהמשך.

**2. זיהוי שורות הטקסט ויישור השורה כך שהאותיות יהיו אופקיות –** המכונה למעשה מזהה שורות טקסט ומיישרת אותם במידת הצורך , שלב זה פותר בעיות של דפים שסריקתם נעשתה בזווית, מה שמקשה על התוכנה לזהות את צורת האותיות בצורה נכונה.

**3. הפרדת השורות השלמות למילים על פי מרווחים –** המכונה מפרידה את השורות למילים לצורך זיהוי האותיות בהמשך.

**4. הפרדת השורות השלמות לאותיות על פי המילים –** המכונה מפרידה את המילים לאותיות לצורך זיהוי האותיות בהמשך.

**5. קבלת לינק לקריאה בפונט קריא –** לאחר סידור וסיווג הטקסט שהיה בתמונה, המערכת מעבירה את הטקסט שהומר לקובץ txt או word.

**איטרציה 5:**

**מטרה:** הוספת פיצ'רים לצד השרת ושדרוג האפליקציה והוספת עיצוב.

**פיצ'רים באיטרציה 5 :**

**1. תרגום הטקסט שזוהה בתמונה –** אם המשתמש בחר בתרגום הטקסט, הטקסט שהיה בתמונה בשפה האנגלית יתורגם לשפה העברית

**2. הרצת קוד שזוהה בתמונה –** אם המשתמש בחר בסריקת קוד, תמונת הקוד שצילם המשתמש בשפת python, מומרת לקוד וניתן להריץ אותה באמצעות המערכת.

**3. העברת קובץ הטקסט ל PDF-** אם המשתמש בחר בהעברת קובץ הטקסט לpdf, הטקסט שהומר ועבר לקובץ txt או word מועבר לקובץ pdf**.**