**מדריך למשתמש - User Manual:**

**מערכת ODSL-Jigsaw לייצור מסמכים אוטומטי:**

1. **מבוא**

מערכת ODSL-Jigsaw היא כלי חדשני לייצור אוטומטי של מסמכים, מצגות וגיליונות אלקטרוניים. המערכת משלבת שפות ייעודיות למשימה (DSLs) עם טכניקות סינתזת תוכניות, ומאפשרת יצירת תוכן מורכב באמצעות הוראות בשפה טבעית.

1. **דרישות מערכת**

* Python 3.8 או גרסה מתקדמת יותר.
* ספריות: python-docx, openpyxl, python-pptx, matplotlib .

1. **התקנה והגדרה**
2. התקן Python מהאתר הרשמי python.org.
3. התקן את הספריות הנדרשות .



1. הורד את קוד המקור של המערכת מהמאגר המסופק.
2. **מבנה המערכת:**

המערכת מורכבת ממספר מודולים עיקריים:

**MockNLToODSL:** ממיר קלט בשפה טבעית לפקודות ODSL.

**ODSLParser:** מפרק פקודות ODSL לאובייקטי פקודה.

**EnhancedExecutionEngine:** מבצע את הפקודות ויוצר את המסמכים.

**DocumentGenerator:** אחראי על יצירת הקבצים הפיזיים.

**QualityChecker:** בודק את איכות המסמכים שנוצרו.



**הסבר על הקשרים:**

1. **: User Input -> MockNLToODSL** קלט בשפה טבעית מועבר להמרה לפקודות ODSL.
2. **: MockNLToODSL -> ODSLParser** פקודות ODSL מועברות לפירוק לאובייקטי פקודה.
3. **: ODSLParser -> EnhancedExecutionEngine** אובייקטי פקודה מועברים לביצוע.
4. **EnhancedExecutionEngine <-> DocumentGenerator :** העברת הוראות ליצירת מסמכים וקבלת עדכונים על התקדמות.
5. **EnhancedExecutionEngine <-> QualityChecker:** בדיקת איכות המסמכים והעברת משוב לשיפור.
6. **DocumentGenerator -> Generated Documents :** יצירת הקבצים הפיזיים הסופיים.

* זה מייצג את זרימת המידע והתהליכים במערכת, מהקלט הראשוני ועד ליצירת המסמכים הסופיים.

1. **פורמט הקלטים:**

המערכת מקבלת קלט בשפה טבעית (עברית) המתאר את המסמך הרצוי. הקלט צריך לכלול:

* סוג המסמך (מסמך, מצגת, או גיליון אלקטרוני).
* שם המסמך.
* תוכן המסמך (כגון כותרות, פסקאות, שקופיות).
* אלמנטים נוספים (כגון תרשימים, טבלאות, תמונות).

**דוגמאות לקלטים תקינים:**

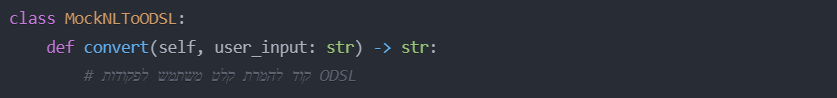
"צור מסמך 'דוח שנתי' עם תוכן 'סיכום הישגים' ותרשים 'עוגה' וטבלה 'נתונים כספיים'"

"הכן גיליון 'תקציב 2024' עם נוסחה 'SUM(A1:A10)' ועיצוב 'מקצועי' ותרשים 'קווי'"

"הכן מצגת 'תכנית עסקית' עם שקופית 'חזון החברה' ושקופית 'יעדים' ותמונה 'לוגו' ותרשים 'עמודות'"

1. **פונקציות עיקריות:**

6.1 MockNLToODSL



תפקיד: ממיר את הקלט בשפה טבעית לסדרת פקודות ODSL.

**ODSLParser 6.2**

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

תפקיד: מפרק את פקודות ה-ODSL לאובייקטים שהמערכת יכולה לעבוד איתם.

**EnhancedExecutionEngine 6.3**

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

תפקיד: מבצע את הפקודות ויוצר את המסמכים הווירטואליים.

**DocumentGenerator 6.4**

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

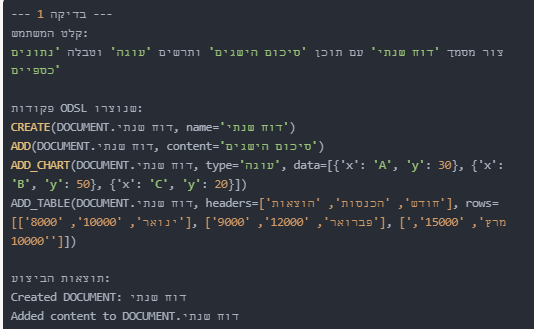
תפקיד: יוצר את הקבצים הפיזיים בפורמטים המתאימים.

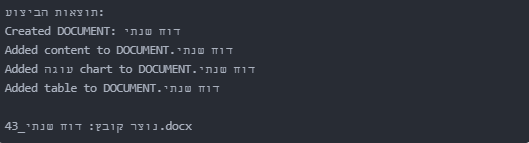
1. **תהליך הביצוע**
2. המשתמש מזין קלט בשפה טבעית
3. MockNLToODSL ממיר את הקלט לפקודות ODSL
4. ODSLParser מפרק את הפקודות לאובייקטים
5. EnhancedExecutionEngine מבצע את הפקודות
6. DocumentGenerator יוצר את הקבצים הפיזיים
7. QualityChecker בודק את איכות המסמכים שנוצרו
8. **פורמט הפלטים**

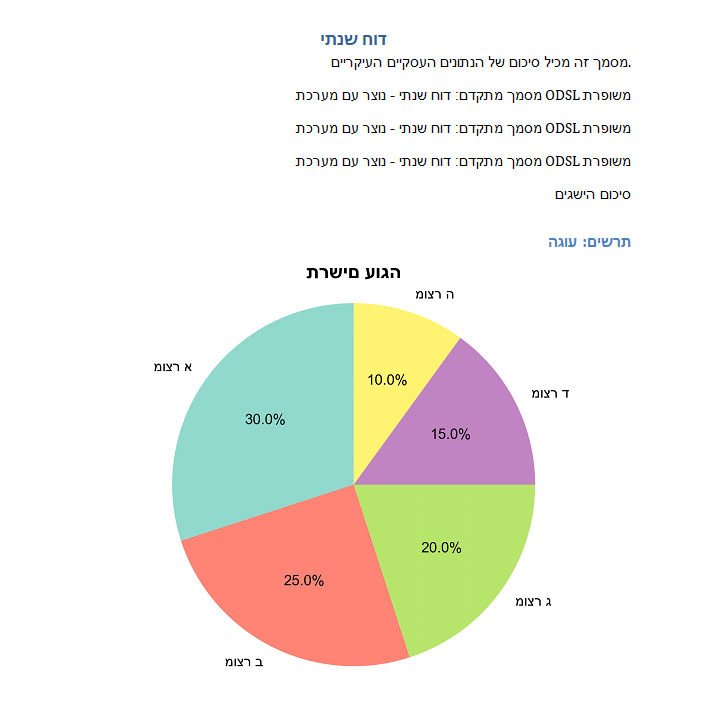
המערכת מייצרת קבצים פיזיים בפורמטים הבאים:

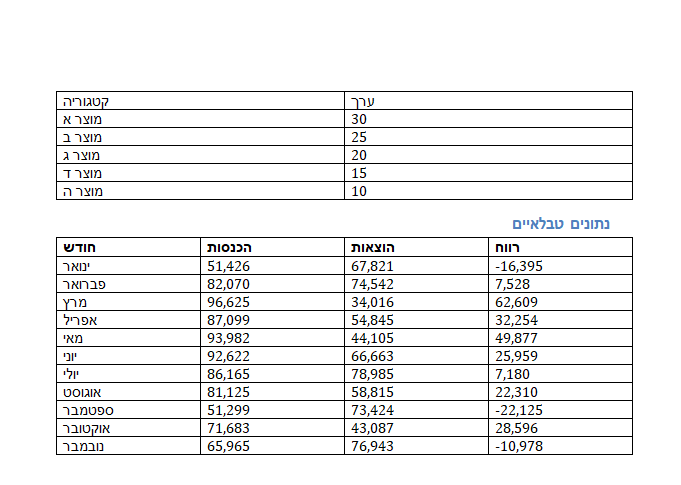
* מסמכים: קבצי .docx
* גיליונות אלקטרוניים: קבצי .xlsx
* מצגות: קבצי .pptx

**דוגמא לפלט:**

****

****





1. **טיפול בשגיאות**

* קלט לא תקין: המערכת תנסה לפרש את הקלט ככל האפשר ותדווח על חלקים שלא הצליחה לפרש
* בעיות ביצירת קבצים: יוצג לוג שגיאה עם פרטי הבעיה

1. **מגבלות ידועות**

* תמיכה בשפה העברית בלבד
* מוגבל לסוגי תרשימים בסיסיים (עוגה, קווי, עמודות)
* אין תמיכה בעיצוב מתקדם של מסמכים
* הוספת תמונות מוגבלת לתיאור בלבד, ללא הוספה בפועל

1. **טיפים לשימוש יעיל**

* השתמש בשפה ברורה ומדויקת בתיאור הבקשות
* בדוק את הקבצים שנוצרו לאחר כל פעולה
* התחל עם בקשות פשוטות והתקדם לבקשות מורכבות יותר

1. **נספח: דוגמא מפורטת**

קלט:



פלט:

