

1. כשל ה"ניסוח העקיף" (Social Engineering)

- **הכשל:** המשתמש ביקש "שהתיקייה יתהיה ריקה ונקיה לגמרי". המודל המקורי חיפש מילות מפתח כמו "מחק" (Delete) ולא זיהה את הסכנה שבניסוח העקיף, מה שהוביל לביצוע פעולה הרסנית.
- **מה למדנו:** אסור להסתמך על רשימת "מילים אסורות". המודל חייב לבצע **ניתוח כוונה (Intent Analysis)** ולבחון מה תהיה התוצאה הסופית של הפקודה לפני שהוא מייצר אותה.

2. כשל "הנתיב הארוך" (Long Path Error)

- **הכשל:** פקודת החיפוש הרקורסיבי (`Recurse-`) קרסה כשהיא נתקלה בתיקיות פיתוח עמוקות (כמו `node_modules`), מה שגרם להצפת הטרמינל בשגיאות אדומות ועצירת התהליך.
- **מה למדנו:** בסביבת Windows, חיפוש עמוק חייב לכלול טיפול בשגיאות מובנה. למדנו שחובה להוסיף את הפרמטר `ErrorAction SilentlyContinue` כדי שהפקודה תדלג על מכשולים טכניים ותמשיך ליעד שלה.

3. כשל ה-Python Syntax (Escape Sequences)

- **הכשל:** הטרמינל הציג שגיאת `'\D SyntaxWarning: invalid escape sequence'`. זה קרה כי Python ניסה לפרש לוכסנים בנתיבי קבצים (כמו `Documents\`) כתווי בקרה.
- **מה למדנו:** כשמגדירים פרומפטים המכילים נתיבי Windows, חובה להשתמש ב-`Raw Strings` (הוספת האות `r` לפני הגרשיים) כדי למנוע מ-Python לשבש את הפקודות.

4. כשל "השם הנעלם" (Missing Extensions)

- **הכשל:** בשינוי שם הקובץ "קורות חיים 1", המודל הסיר את סיומת ה-`.pdf`. כתוצאה מכך, הקובץ הפך לבלתי מזוהה על ידי המערכת.
- **מה למדנו:** המודל צריך להיות מודע לסוג הקובץ. למדנו שיש להנחות אותו לשמור על `Parameter Integrity` ולוודא שהסיומת נשמרת גם לאחר שינוי השם.

5. כשל "הזהירות המופרזת" (False Positives)

- **הכשל:** המודל סיווג שאלות מידע תמימות כמו "מה השעה?" או "כמה קבצים יש לי?" כבקשות מסוכנות והחזיר הודעת שגיאה.
- **מה למדנו:** יש להגדיר בבירור למודל מהן `Read-Only Queries`. למדנו שפעולות קריאה וספירה הן בטוחות לחלוטין וצריך להפריד אותן בבירור מפעולות כתיבה או מחיקה.