**🖥️ מיני פאוור שלל בג׳אווה**

פרויקט זה הוא סימולציה קלת משקל של ממשק שורת פקודה הכתוב בג׳אווה.  
הוא מאפשר פעולות מערכת קבצים בסיסיות כגון צפייה, ניווט ויצירת קבצים וספריות באמצעות פקודות מוכרות (ls, cd, mkdir, touch, pwd, help, exit).

**1. 🎯 סקירת הפרויקט**

פרויקט המיני פאוור שלל מדגים מניפולציה מעשית של מערכת קבצים וטיפול בקלט משתמש בג׳אווה.  
יישום חינוכי זה עוזר להבין כיצד ממשקי שורת פקודה עובדים על ידי סימולציה של פונקציונליות שלל חיונית בסביבה פשוטה.

**2. 🚀 איך להדר ולהריץ**

**דרישות**

* ערכת פיתוח ג׳אווה (JDK) 8 ומעלה
* טרמינל/שורת פקודה או סביבת פיתוח (IntelliJ IDEA, Eclipse, VS Code)

**הוראות התקנה**

1. צור פרויקט ג׳אווה חדש בסביבת הפיתוח המועדפת עליך או ספרייה חדשה
2. הוסף את שני הקבצים הבאים לפרויקט שלך:
   * MiniShell.java — מחלקת היישום הראשית שמעבדת פקודות משתמש
   * ShellCommandHandler.java — מחלקת היישום המכילה את כל פעולות מערכת הקבצים

**הידור והרצה**

# נווט לספריית הפרויקט שלך

cd path/to/project

# הדר את שני קבצי הג׳אווה

javac MiniShell.java ShellCommandHandler.java

# הרץ את היישום

java MiniShell

**3. 💻 דוגמת הרצה**

כאשר תריץ את התוכנית, תראה שורת פקודה המציגה את המיקום הנוכחי שלך:

C:\Users\yourname > help

מציג את כל הפקודות הזמינות  
**פלט:**

פקודות שאתה יכול להשתמש בהן:

pwd = ראה תיקייה נוכחית

ls = רשימת קבצים

cd [folder] = שנה תיקייה

mkdir [name] = צור תיקייה חדשה

touch [name] = צור קובץ חדש

help = ראה הודעה זו

exit = צא מהתוכנית

C:\Users\yourname > mkdir projects

יוצר ספרייה חדשה בשם projects  
**פלט:** נוצרה תיקייה חדשה: projects

C:\Users\yourname > cd projects

מנווט אל ספריית projects  
**פלט:** אתה כעת נמצא ב: C:\Users\yourname\projects

C:\Users\yourname\projects > touch app.java

יוצר קובץ חדש בשם app.java  
**פלט:** נוצר קובץ חדש: app.java

C:\Users\yourname\projects > ls

מציג את כל הקבצים והספריות במיקום הנוכחי  
**פלט:** [קובץ] app.java

C:\Users\yourname\projects > cd ..

חוזר לספריית האב  
**פלט:** אתה כעת נמצא ב: C:\Users\yourname

C:\Users\yourname > exit

מסיים את היישום  
**פלט:** להתראות!

**4. 🔍 מבנה הקוד ופונקציונליות**

**מחלקה ראשית: MiniShell.java**

* יוצרת את לולאת עיבוד הפקודות
* מנתחת קלט משתמש לפקודות וארגומנטים
* מנתבת פקודות לשיטות הטיפול המתאימות
* מנהלת את מחזור החיים של היישום

// יישום מפתח - ניתוח פקודות

String command = "";

String argument = null;

int spaceIndex = line.indexOf(" ");

if (spaceIndex == -1) {

command = line;

} else {

command = line.substring(0, spaceIndex);

argument = line.substring(spaceIndex + 1);

}

**מחלקת מטפל: ShellCommandHandler.java**

מחלקה זו מנהלת את כל אינטראקציות מערכת הקבצים באמצעות השיטות הבאות:

| **שיטה** | **תיאור** |
| --- | --- |
| printWorkingDirectory() | מציג את הנתיב המוחלט של הספרייה הנוכחית |
| listDirectory() | מציג את כל הקבצים והתיקיות עם סמני סוג |
| changeDirectory(String folder) | מנווט לספריות שצוינו (כולל הורה עם ..) |
| makeDirectory(String name) | יוצר ספריות חדשות עם בדיקות אימות |
| createFile(String name) | יוצר קבצים ריקים עם טיפול בשגיאות |
| printHelp() | מציג את כל הפקודות הזמינות עם תיאורים |
| getCurrentDirectory() | מחזיר את מיקום הספרייה הנוכחית כאובייקט File |

**5. ✅ פקודות זמינות**

| **פקודה** | **תיאור** | **דוגמה** |
| --- | --- | --- |
| pwd | הדפס ספריית עבודה נוכחית | pwd |
| ls | רשימת קבצים וספריות | ls |
| cd [dir] | עבור לספרייה שצוינה | cd documents או cd .. |
| mkdir [name] | צור ספרייה חדשה | mkdir photos |
| touch [name] | צור קובץ ריק חדש | touch notes.txt |
| help | הצג מידע עזרה | help |
| exit | סיים את השלל | exit |

**6. 🛠️ טיפול בשגיאות**

היישום כולל אימות למצבים נפוצים:

* ארגומנטים חסרים (שימוש: mkdir [שם\_ספרייה])
* ניווט לספריות לא קיימות (לא ניתן למצוא את התיקייה: xyz)
* יצירת קבצים או ספריות כפולים (הקובץ הזה כבר קיים!)
* חריגות IO במהלך פעולות קבצים (שגיאה ביצירת קובץ!)
* פקודות לא חוקיות (פקודה לא ידועה. הקלד 'help' כדי לראות פקודות זמינות.)

**7. 📝 שיפורים עתידיים**

שיפורים פוטנציאליים לגרסאות עתידיות:

* יכולות עריכת תוכן של קבצים
* היסטוריית פקודות עם ניווט חצים למעלה/למטה
* פעולות העתקה, העברה ומחיקת קבצים
* תמיכה בתווים כלליים לבחירת קבצים
* השלמת Tab לשמות ספריות וקבצים
* פלט בקידוד צבע לסוגי קבצים שונים