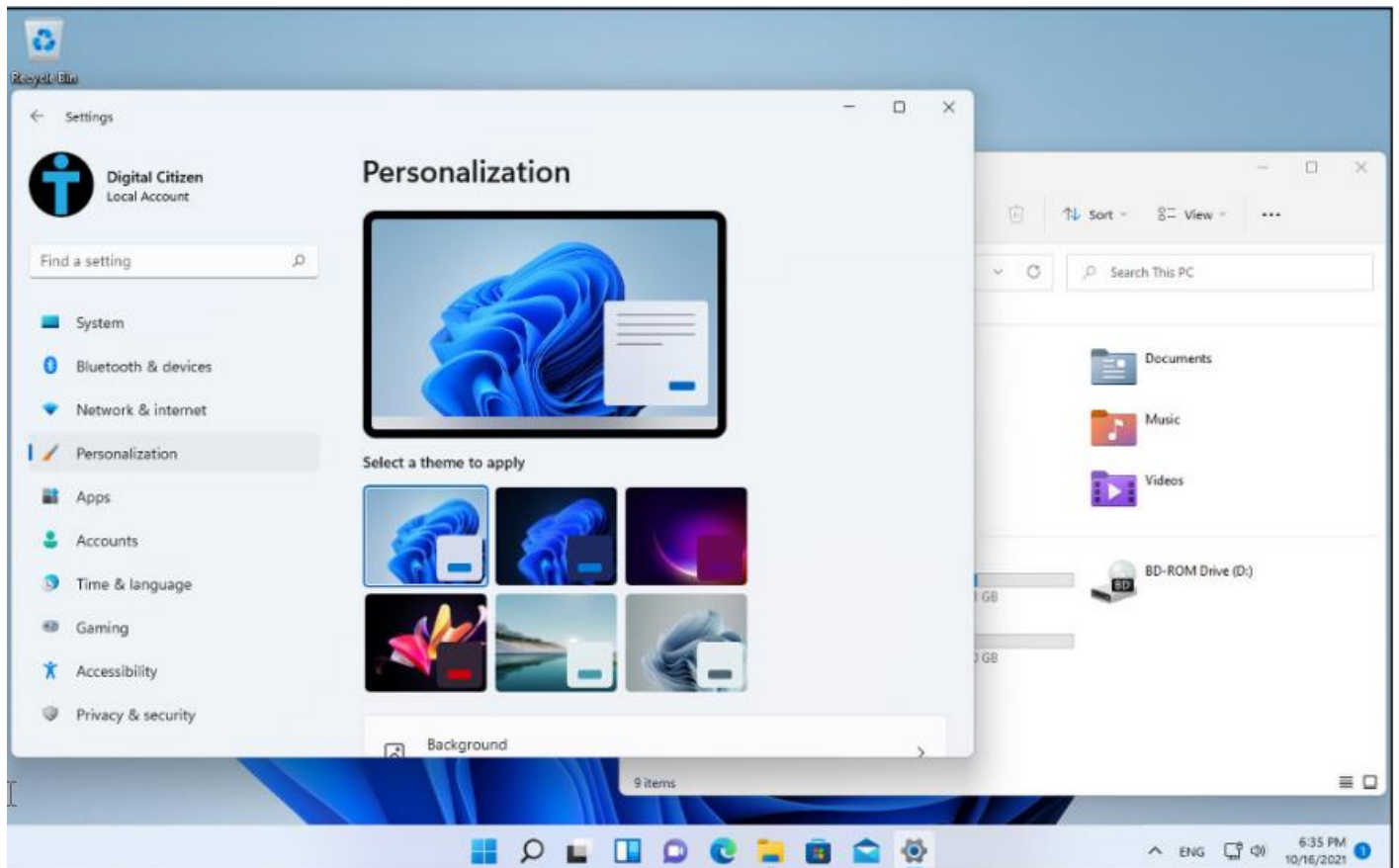


הכרות עם ווינדוס 11

למרות ההכרזות של מיקרוסופט על כך שווינדוס 10 תהיה מע' ההפעלה האחרונה למשתמשי קצה, בסוף יוני 2021 מיקרוסופט הציגה לנו מע' הפעלה חדשה ומלוטשת יותר. ווינדוס 11 עוצבה מחדש, כפתור ההתחל שונה, המראה יותר נקי וזורם. כמו כן בגירסה זו חזרו הווי'דג'טים – תוספי תוכנה הניתנים להתקנה על שולחן העבודה.

מיקרוסופט שילבה את אפליקציית TEAMS כחלק ממע' ההפעלה, ישנו שיפור בביצועי המערכת. במקום CMD ו POWERSHELL יש מקום אחד להריץ את כל הפקודות Windows Terminal, חנות האפליקציות עברה עדכון.

ישנה תמיכה בלינוקס וכעת גם באנדרואיד בקרוב נוכל להוריד אפליקציות אנדרואיד בחנות מיקרוסופט



בתמונה: כניסה להגדרות התאמה אישית בווינדוס 11 – נוכל להבחין בעיצוב השונה של שורת המשימות וכפתור התחל

ווינדוס 11 יכולה להריץ מגוון אפליקציות, אפליקציות מסורתיות (win32 api) ואפליקציות מודרניות (אפליקציות הניתנות להורדה בחנות)

אפליקציות מודרניות אילו שונות בהתנהגות מאפליקציות ישנות בכמה מאפיינים:

תוכנות מודרניות רצות בסביבה מוגנת ולא יכולות לשנות קבצי מע' והגדרות מע' הפעלה ולכן יותר בטוחות

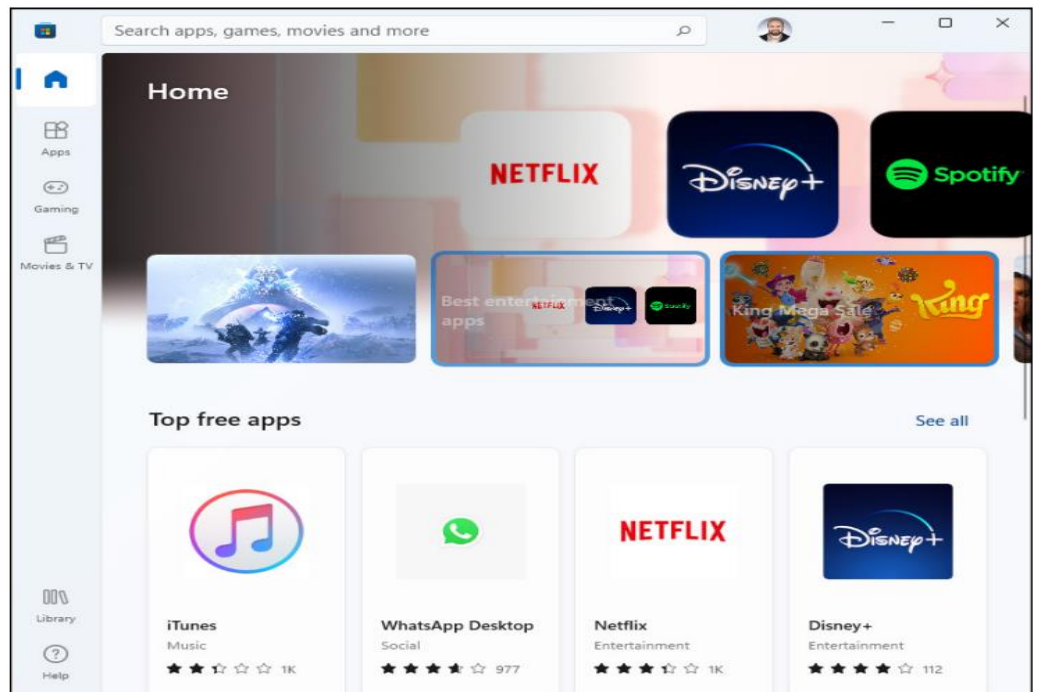
תוכנות מודרניות ניתנות לניהול בקלות וברוב המקרים אם תוכנה כזו נתקעת היא לא תגרום להשבתת המחשב

תוכנות אילו תוכננו לעבוד גם עם מסכי מגע בשונה מתוכנות מסורתיות שעובדות רק עם עכבר ומקלדת

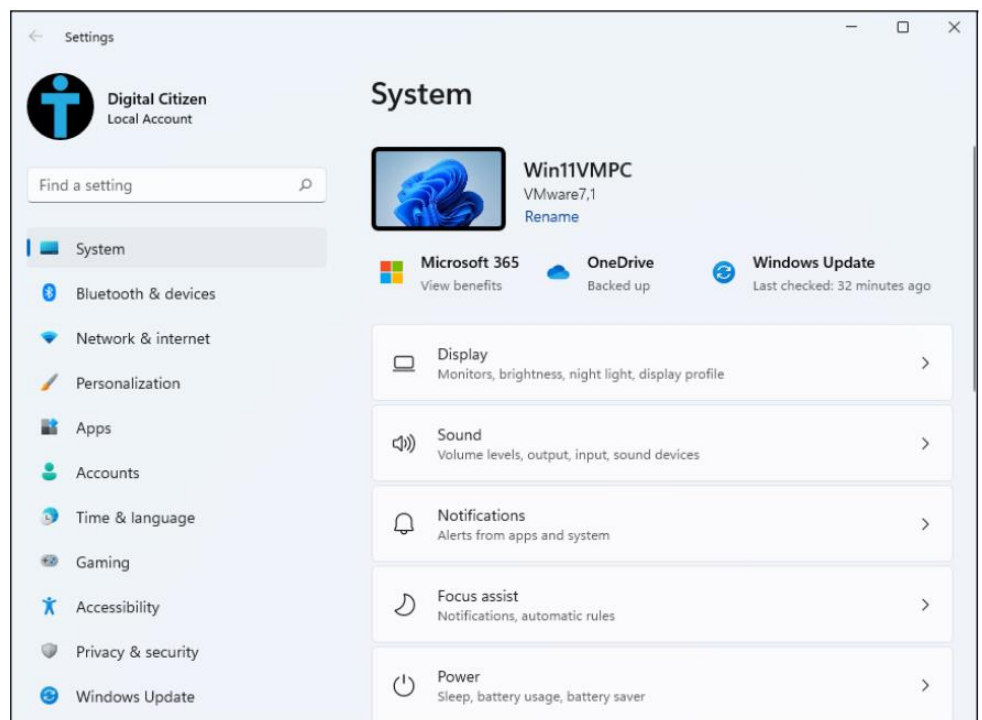
לא ניתן להריץ מספר מופעים של אפליקציה מודרנית במקביל

ההפצה שלהן מתבצעת רק דרך החנות של מיקרוסופט והן מתעדכנות בצורה אוטומטית, אפליקציות מסורתיות ניתן להוריד מכל מקום אך יש צורך לעדכן ידנית

בחנות האפליקציות של מיקרוסופט ניתן להוריד אפליקציות שולחניות (ישנות) וגם מודרניות ובקרוב גם אפליקציות אנדרואיד



מבחינה ויזואלית מייקרוסופט שינתה את העיצוב של כפתור ההתחל ושורת המשימות וההשראה מגיעה מעולמות המקינטוש, ישנן פחות אפשרויות להתאמה, לא נוכל לשנות את הגודל של כפתור התחל, אבל העיצוב נשמר נקי ומאורגן. שיפור משמעותי נוסף הוא מסך ההגדרות שהיא למעשה אפליקציה, כעת היא מאורגנת טוב יותר כאשר כל הקטגוריות מופיעות בצד. כניסה זריזה (Windows+I)



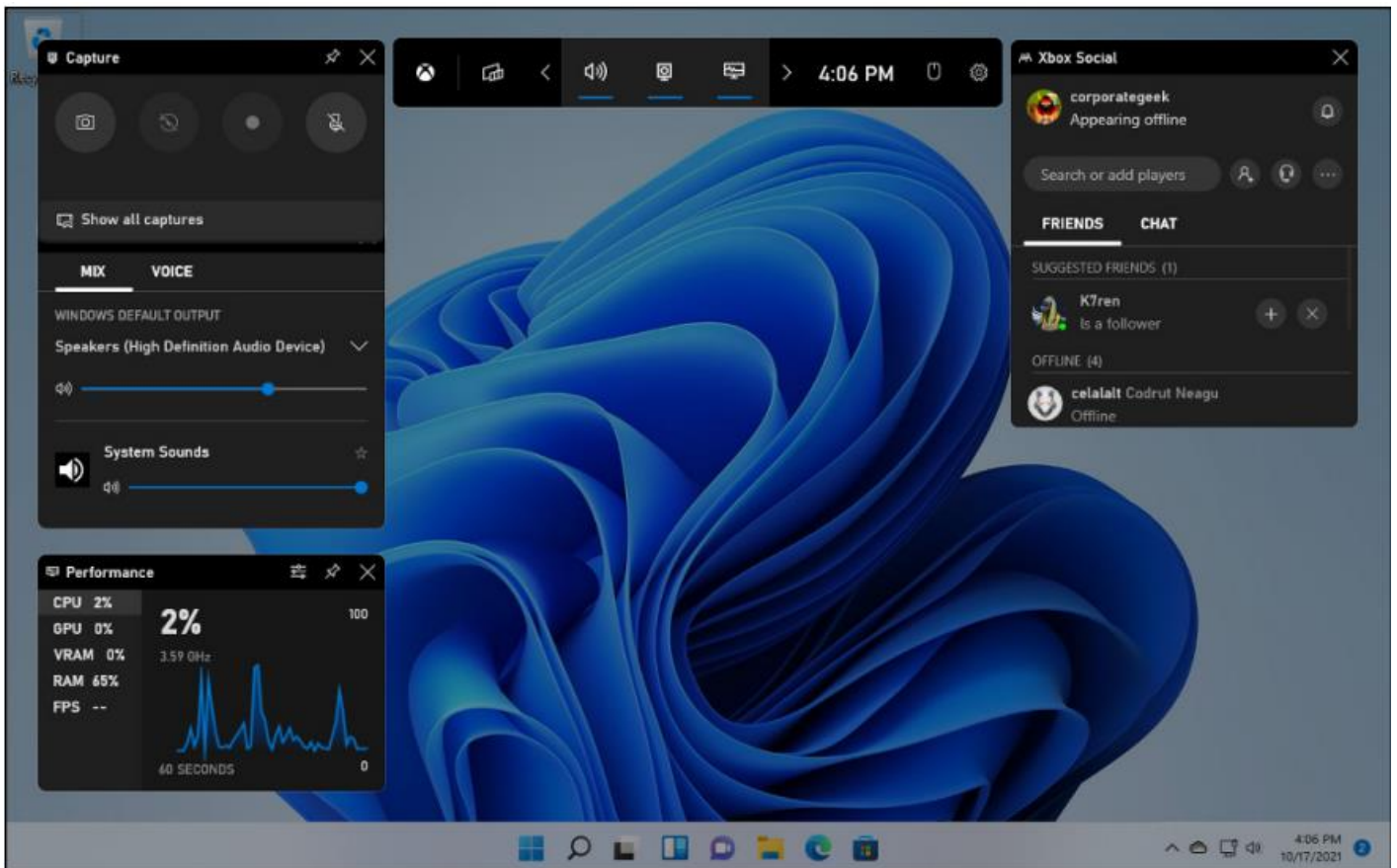
בנוסף המון הגדרות שהיו בלוח הבקרה נכנסו לתפריט המודרני, מה שמרמז על הכיוון לוותר על לוח הבקרה הישן דבר שכנראה יתבצע בגירסאות הבאות.

ווינדוס 11 מציגה שיפור בביצועים לעומת ווינדוס 10 ובנוסף המע' יודעת לתעדף אפליקציות ולתת משאבים לאפליקציות פעילות ולהעביר אפליקציות לעבודה ברקע ע"מ לחסוך משאבים.

מבחינת משחקים ווינדוס 11 תומכת ב-DIRECTX בגרסה הכי חדשה עם תמיכה ב-HDR (אם המסך תומך), כמו כן ישנה תמיכה בתכונה Direct Storage המאפשרת לעקוף את הפנייה למעבד בכל פעם שכרטיס מסך צריך מידע מכונן Nvme ובכך להאיץ את המהירות שבה המשחק יכול לעבוד, כרגע אין משחקים שתומכים בתכונה זו אבל הדבר ישתנה בעתיד.

בוינדוס 11 ישנו מצב משחק – ברגע שמע' ההפעלה מזהה שאנו מפעילים משחק , היא מפנה יותר משאבים אל האפליקציה ועוצרת במקביל עדכוני מערכת והתראות על מנת שלא יפריעו לחוויית המשחק.

תוספת לגיימרים היא Xbox game bar תכונה המאפשרת לצלם ולתעד משחקים הן בוידאו ובתמונות וכמו כן לשחק עם עוד שחקנים ולצ'אט איתם , הפעלה באמצעות Win+G



בתמונה : חלק מהיכולות של XBOX GAME BAR

עוד שיפור הוא דפדפן אדג' , הדפדפן מחליף את אקספלורר שיצא לגמלאות והוא מבוסס על מנוע כרומיום כמו גוגל כרום. ישנו שיפור בביצועים ובאבטחה.

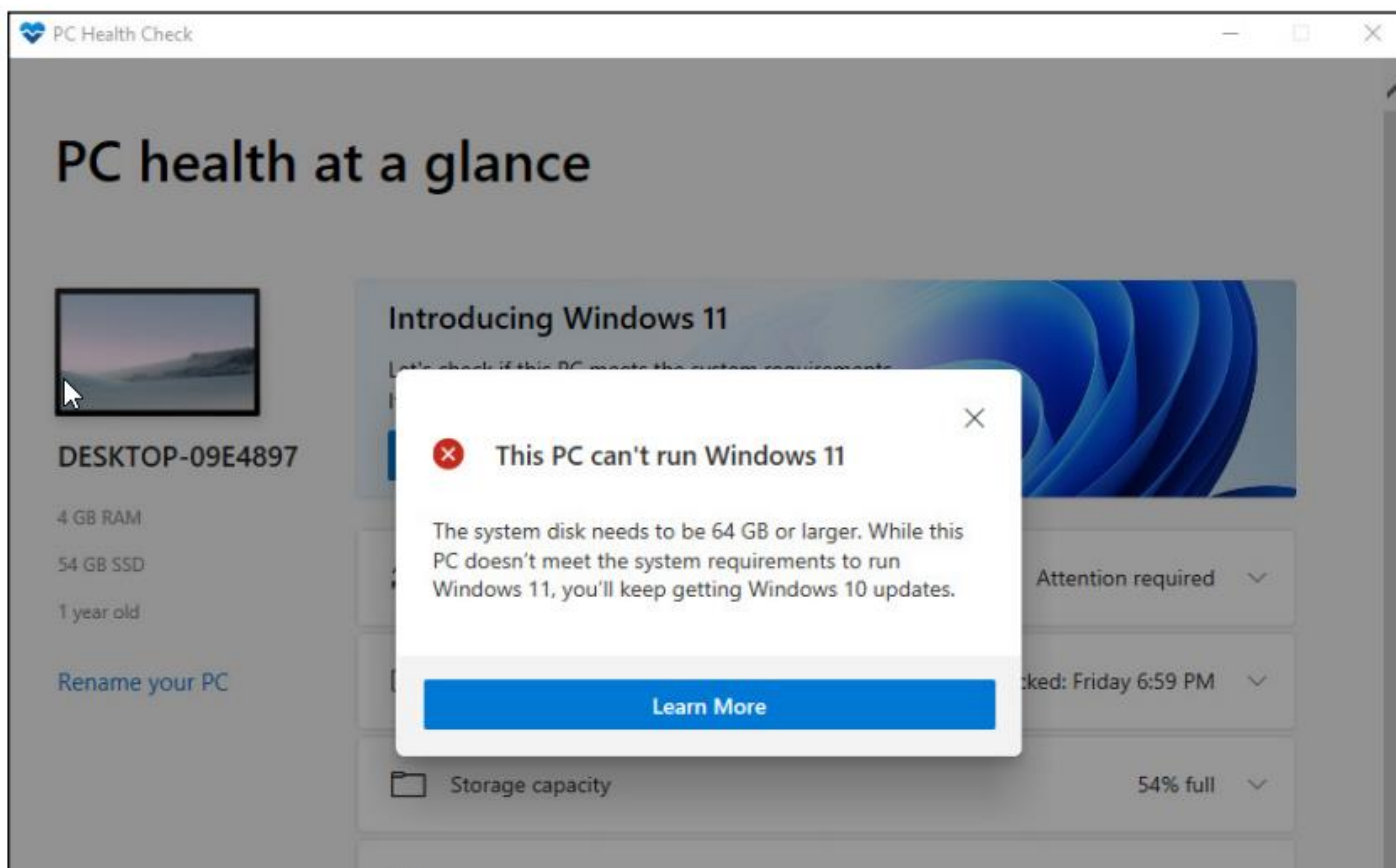
כמו כן בשונה מווינדוס 10 קורטנה היא לא חלק אינטגרלי ממע' ההפעלה ויש לנו יכולת לנטרל אותה לגמרי .

ווינדוס 11 מציגה שיפור משמעותי באבטחת מידע , מיקרוסופט החליטה שזוהי תהיה מע' ההפעלה המאובטחת ביותר ומאלצת אותנו לעבוד עם מעבד שתומך בשבב TPM ותמיכה ב-Secure Boot . שתי הטכנולוגיות הללו היו קיימות עוד בתקופת ווינדוס 8 אך כעת הן לא דרישה אלא חובה.

TPM הינו שבב המובנה בלוח האם של המחשב ומאפשר ליצור מפתח הצפנה הניתן לפענוח רק באמצעות השבב שיצר אותם. תוכנת הצפנה לדוגמת Bitlocker עושה שימוש ברכיב זה , היות ומפתח הפענוח הוא ייחודי לכל רכיב (למעשה לכל לוח...) תוכנות יכולות לוודא במהירות שהמחשב שרוצה לגשת למידע הינו המחשב שמורשה לכך.

Secured Boot הינו תהליך שנועד לוודא שקבצי האתחול של מע' ההפעלה לא עברו שינוי לא מורשה מהאתחול האחרון , כאשר מתגלה חריגה הקוד לא מבוצע והמע' תחזור לקבצים המקוריים ועל ידי כך יבוצע אתחול נקי ותקין.

למרות שלרוב המחשבים יש את היכולת הזו של TPM Secured Boot לעיתים נצטרך להפעיל אותך דרך ה-BIOS



בתמונה מחשב שהתכונה הזו לא מופעלת אצלו ולכן לא יכול לשדרג מווינדוס 10 אל 11.

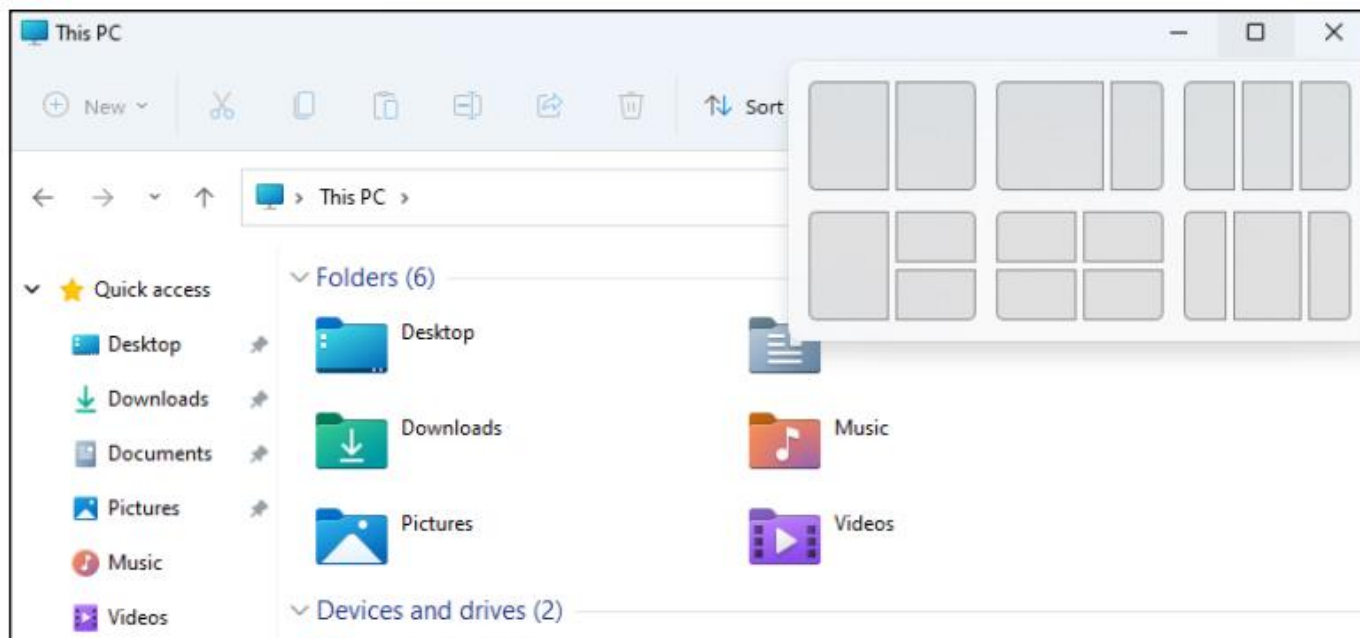
כחלק מהגדרות האבטחה ווינדוס 11 מצפינה את המידע בתור ברירת מחדל, חלק מהאפליקציות רצות בסביבות מבודדות על מנת לא לאפשר פגיעה במע' ההפעלה.

ישנה תמיכה בכניסה למע' באמצעי זיהוי ביומטרים או אפליקציות אימות.

ווינדוס תומכת בשולחנות עבודה וירטואליים בדומה לווינדוס 10, אך בצורה נקיה יותר וחלקה

קיצור דרך Win+Ctrl+D

כמו כן ניתן לעבוד עם יותר מחלון אחד וכעת לבצע הצמדות במגוון אפשרויות, בזמן שנצמיד אפליקציה לצד כלשהו נוכל לבחור את הפריסה (הצמדה של אפליקציה לצד המסך יכולה להתבצע עם כפתור Windows + אחד מחיצ' הכיוון)



בתמונה נוכל לראות את אפשרויות הפריסה של החלונות על גבי המסך

ווינדוס 11 מגיעה במגוון גרסאות, Windows 11 Home מתאימה למשתמש הביתי אך לא מכילה השתלטות מרחוק והצפנת BITLOCKER

גרסת PRO – מכילה יכולות וירטואליזציה, הצפנת BITLOCKER והשתלטות מרחוק וגם את היכולת להצטרף לדומיין.

ישנה גרסה Pro for Workstation המאפשרת תמיכה בעד 6TB זיכרון RAM ועד ארבעה מעבדים במקביל. גרסת ENTERPRISE - זמינה לארגונים גדולים שקונים רישוי מסוג VOLUME LICENSE מכילה את התכונות של PRO ובנוסף ניתנת לרכישה בתצורה שמעדכנת אחת לשנתיים (LTSC)

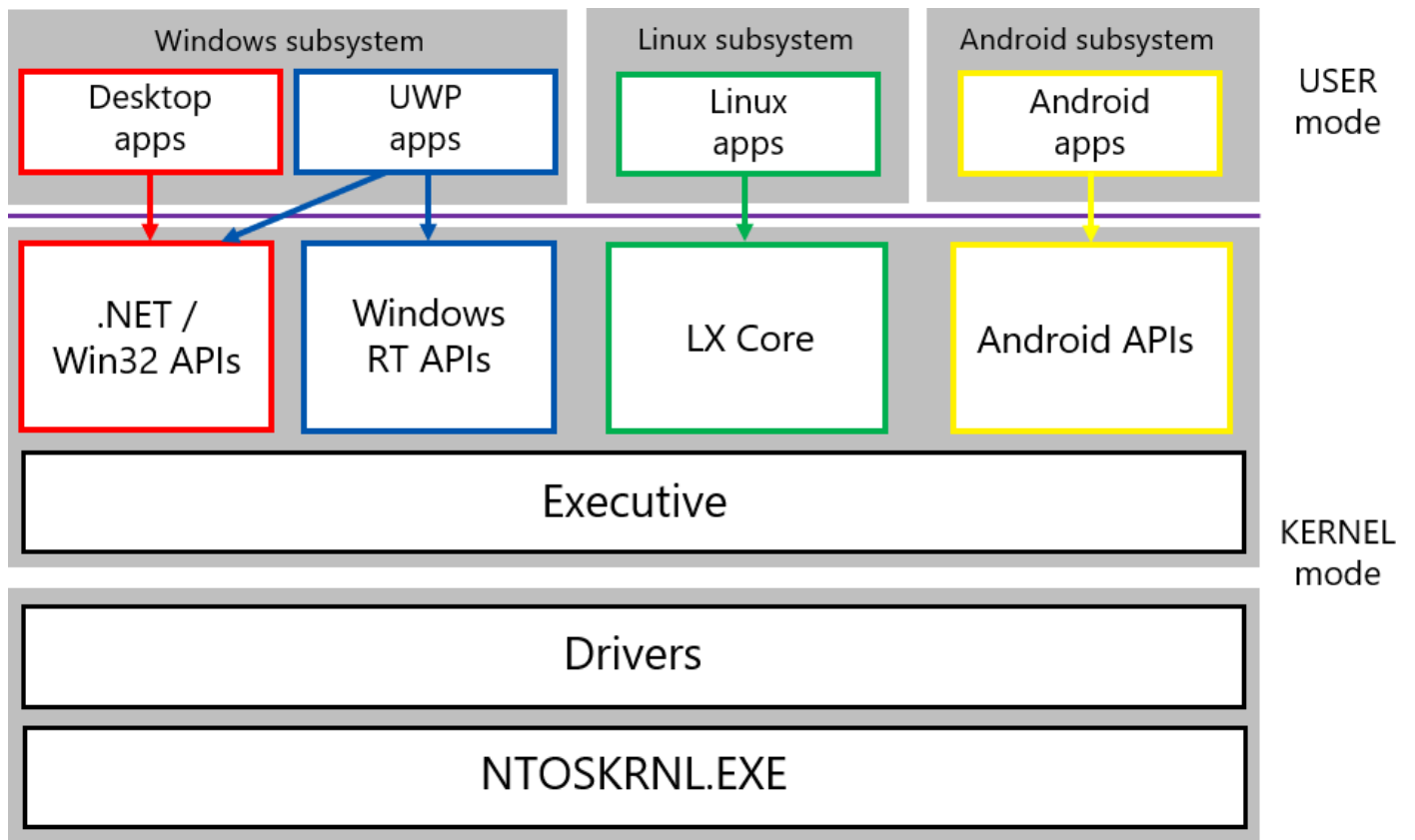
ישנן גרסאות למע' החינוך

כל הגרסאות מגיעות בתצורה של 64 ביט בלבד

קישור להשוואה בין ווינדוס 11 ל-10

<https://www.microsoft.com/en-us/windows/compare-windows-11-home-vs-pro-versions?tab=2>

ארכיטקטורה של ווינדוס 11 :



בתמונה אנו יכולים לראות את הרכיבים שעליהם נשענת מע' ההפעלה. בסביבת המשתמש נוכל לראות תמיכה באפליקציות שולחניות רגילות וגם ממשק תמיכה לאפליקציות מודרניות כמו כן שופרה התמיכה בסביבת לינוקס ונוספה תמיכה בסביבת אנדרואיד

ליבת המע' דומה מאוד לווינדוס 10 , נוספו שיפורי אבטחה , הממשק עבר מתיחת פנים , ישנה תמיכה בחיבור WIFI6 ועוד...

התקנה נקייה של ווינדוס 11

על מנת לבצע התקנה נקייה נצטרך מחשב שעומד בדרישות המינימום להתקנה שהן :

מעבד במהירות 1 ג'יגה או יותר בעל שתי ליבות או יותר התואם ל-64 ביט

זיכרון 4 ג'יגה מינימום

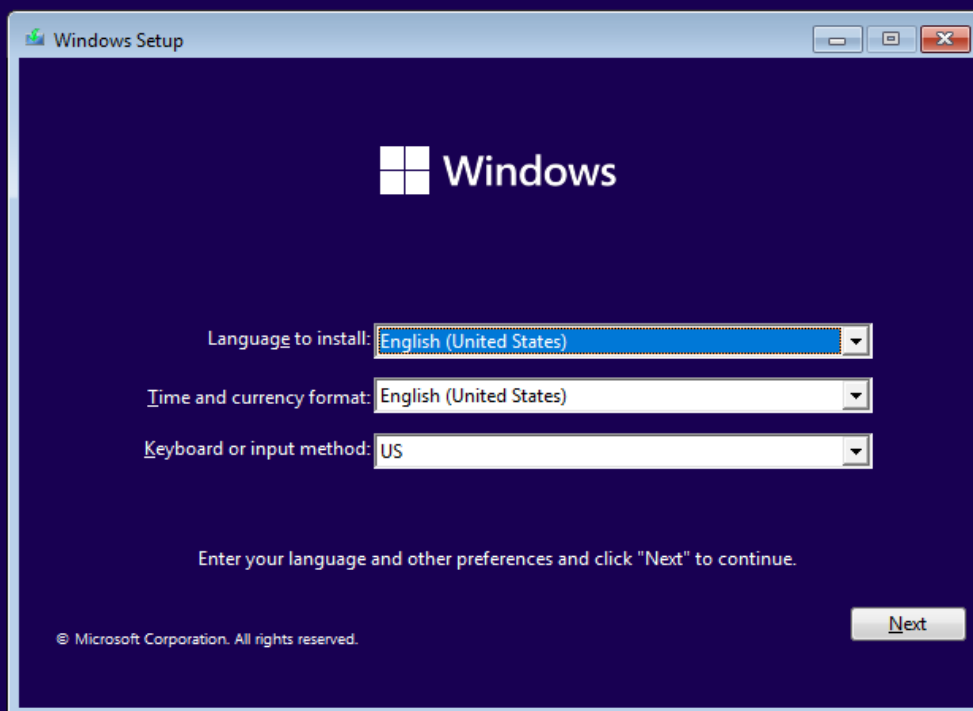
שטח אחסון של לפחות 64 ג'יגה

כרטיס מסך מתאים ל DIRECTX 12

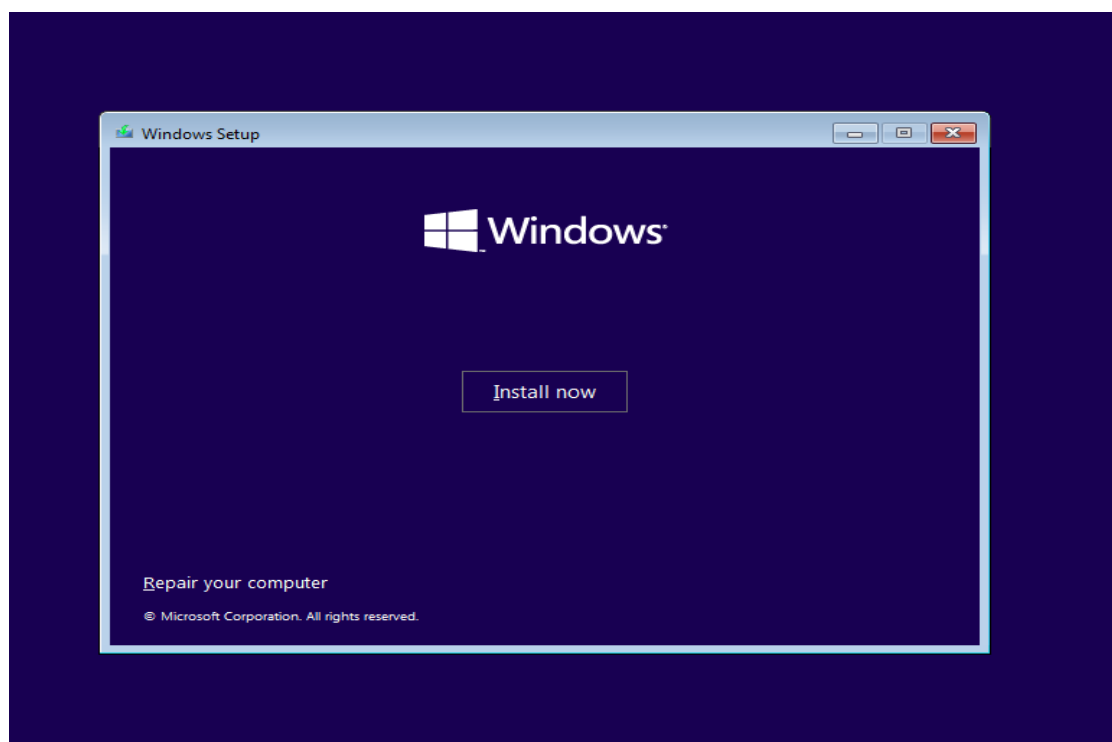
שבב TPM

בגירסת HOME נדרש חיבור אינטרנט על מנת להגדיר חשבון מייקרוסופט

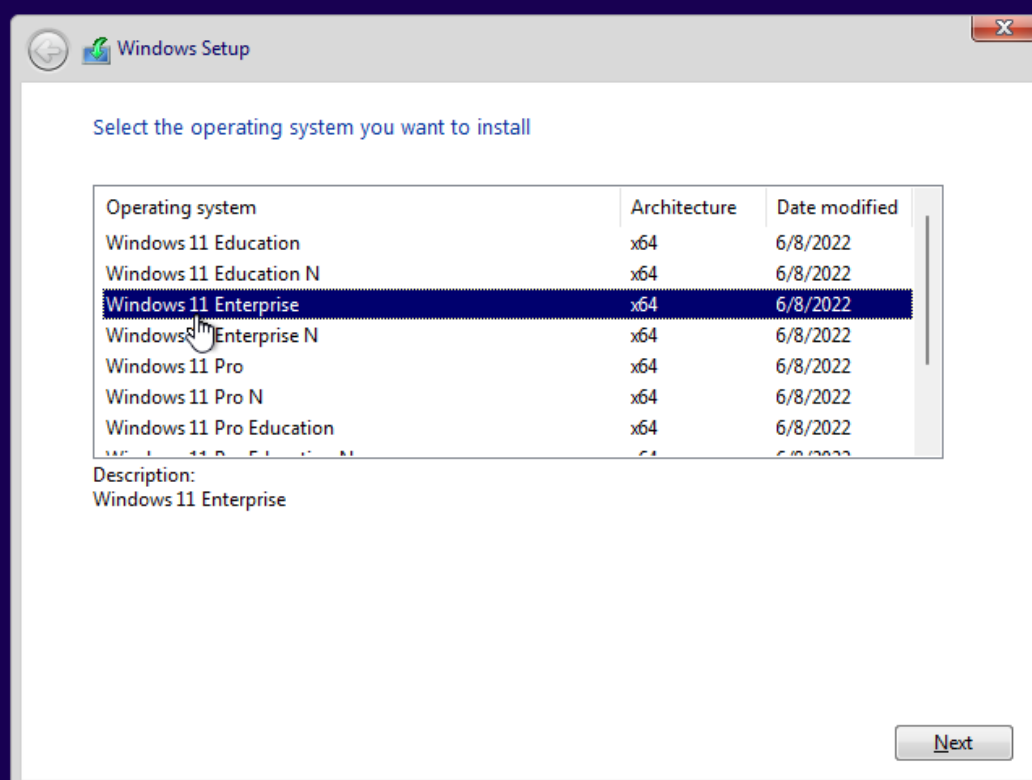
תהליך ההתקנה של ווינדוס 11 :



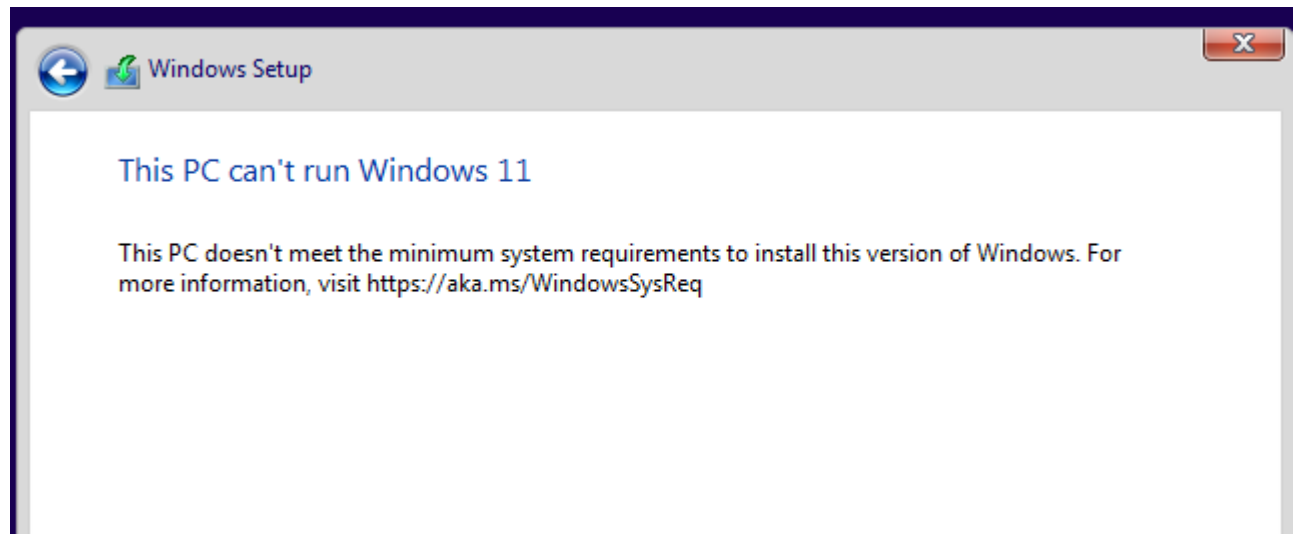
נבחר את שפת ההתקנה ונלחץ NEXT



נבחר את גירסת ההתקנה (במקרה זה Enterprise)



קיבלתי הודעה שהמחשב אינו מתאים להתקנה של ווינדוס (ההתקנה מבוצעת במכונה וירטואלית ואין TPM)



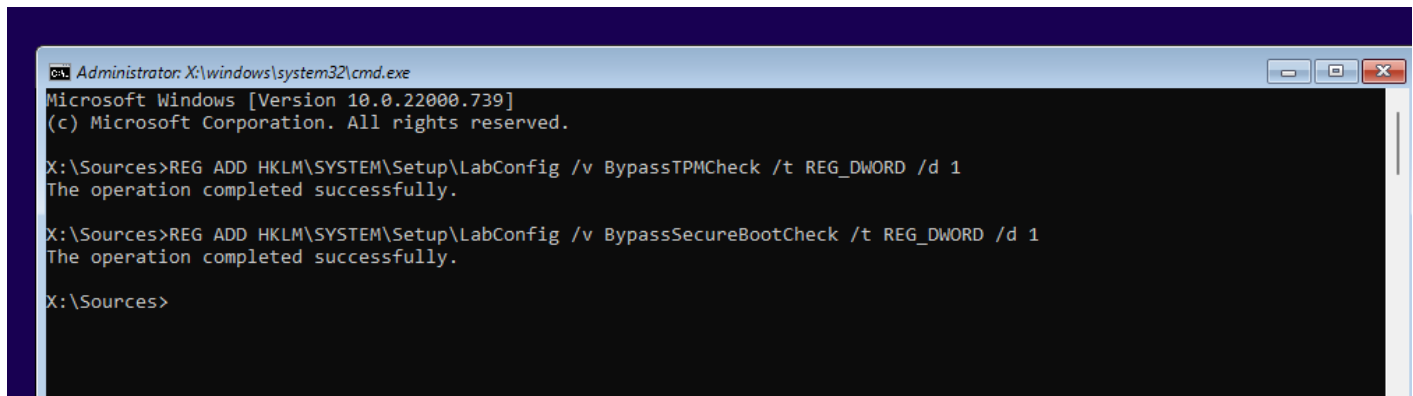
נבצע מעקף שייאפשר לנו לבצע התקנה גם ללא דרישות האבטחה המוגברות, פתרון זה יוכל לעזור לכם בהתקנה בסביבה וירטואלית

נלחץ על SHIFT+F10 וניכנס לממשק הפקודה

נקליד את הפקודות הבאות :

```
REG ADD HKLM\SYSTEM\Setup\LabConfig /v BypassTPMCheck /t REG_DWORD /d 1
```

```
REG ADD HKLM\SYSTEM\Setup\LabConfig /v BypassSecureBootCheck /t REG_DWORD /d 1
```



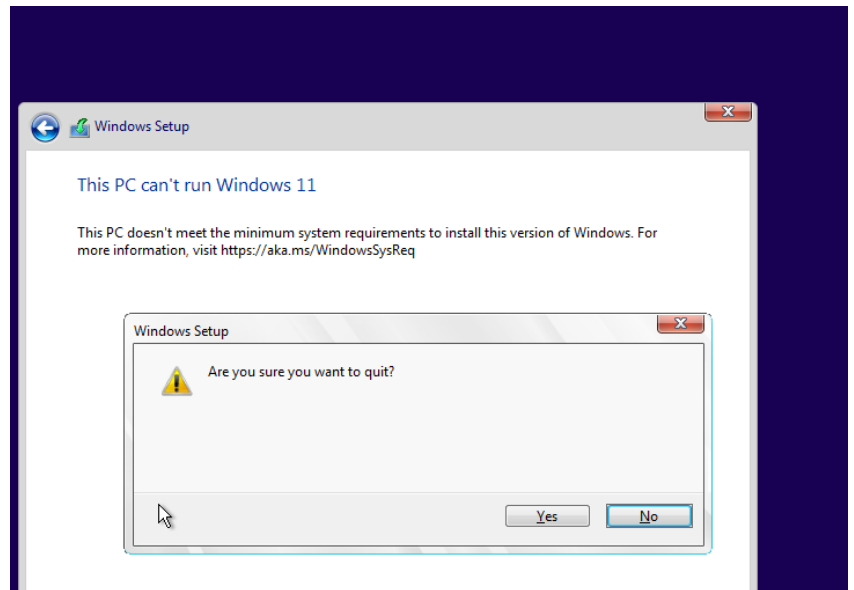
הפקודות יוסיפו שני ערכים לרגיסטרי שייאפשרו לנו להתקין במצב "מעבדה" ולעקוף את המגבלות

תוכלו לקבל מידע נוסף בקישור הבא :

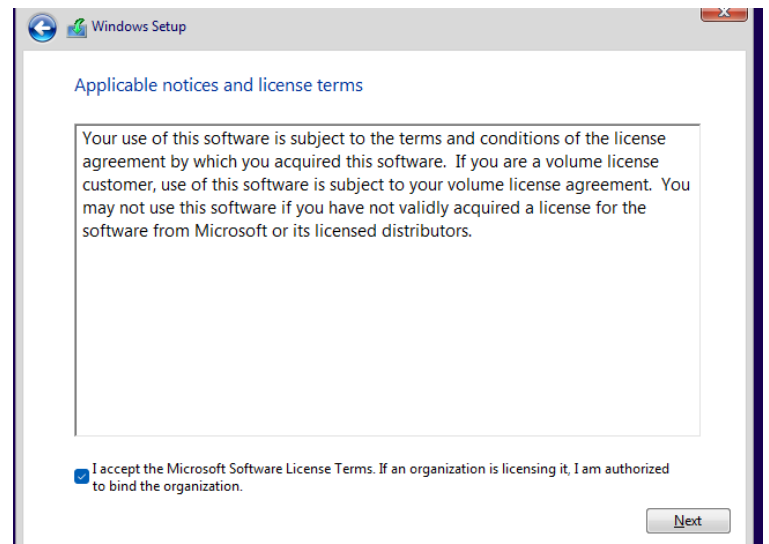
<https://www.techadvisor.com/article/744222/download-windows-11-unsupported-pc.html>

על הרגיסטרי נלמד בפרקים הבאים

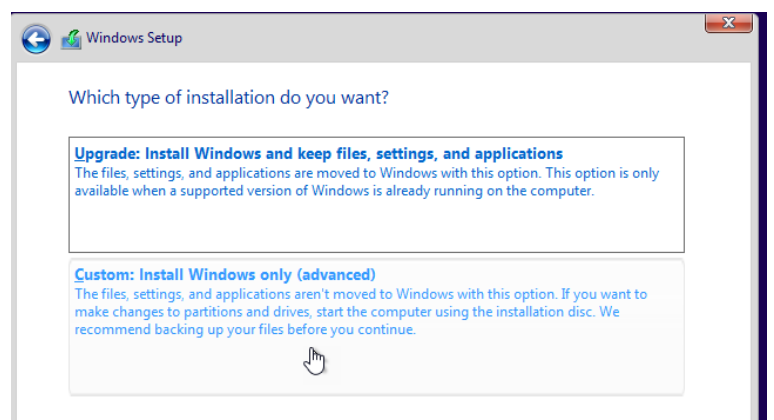
כעת נסגור את החלון שורת הפקודה , נסגור את החלון עם ההודעה ונחזור להתקנה מחדש



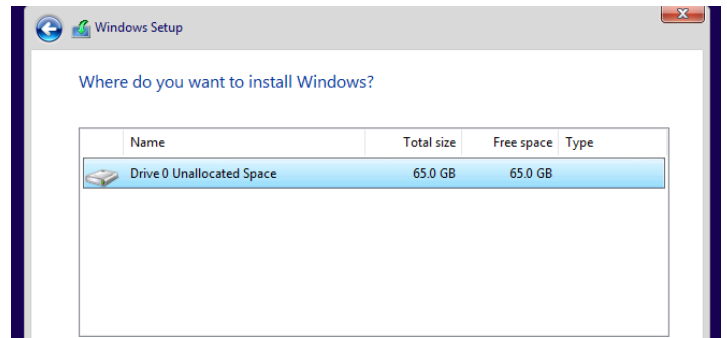
נקבל את תנאי הרישוי



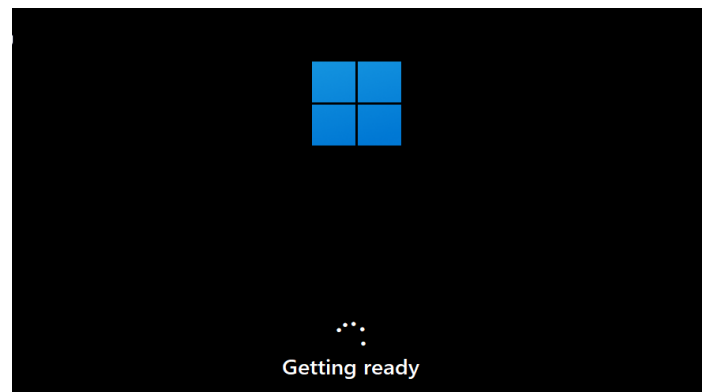
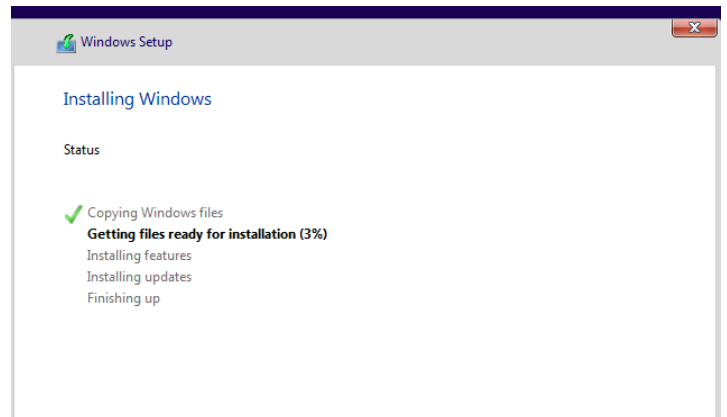
נבחר התקנה מסוג Custom התאמה אישית



נבחר את הדיסק עליו נתקין



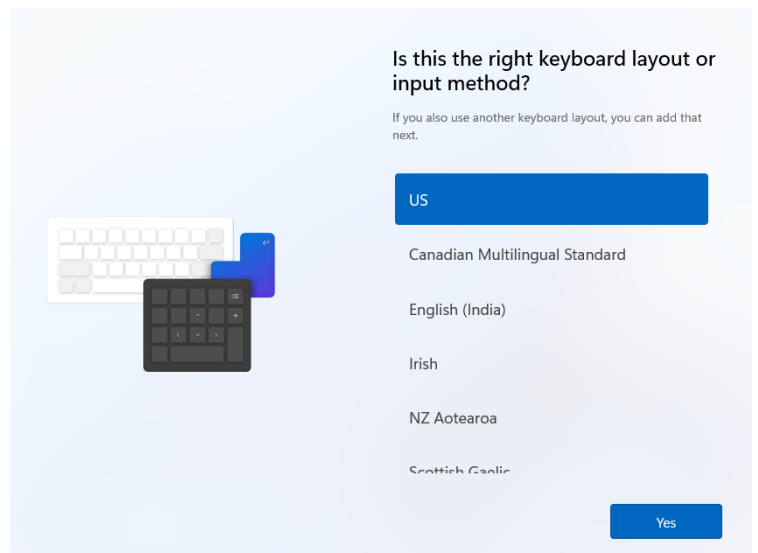
נמתין לסיום ההתקנה ואתחול מחדש



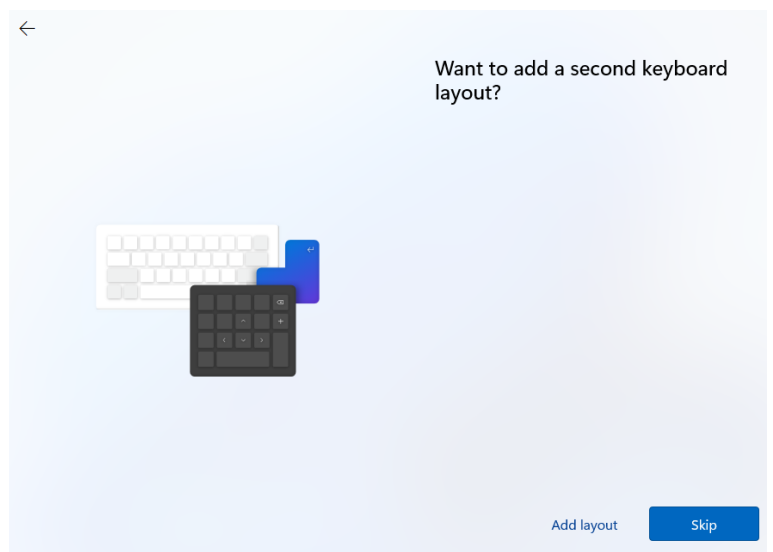
נבחר איזור



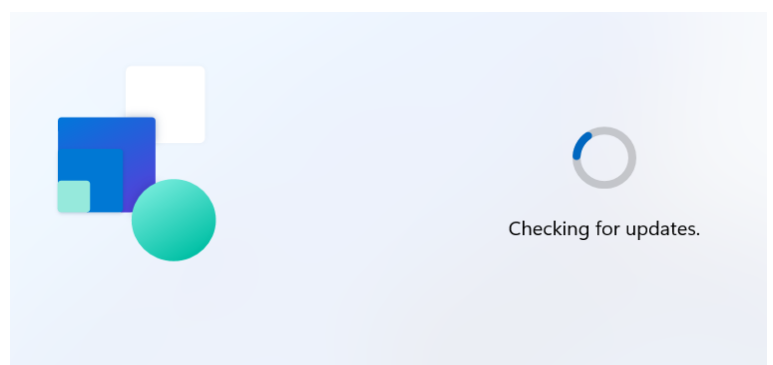
נבחר שפת מקלדת



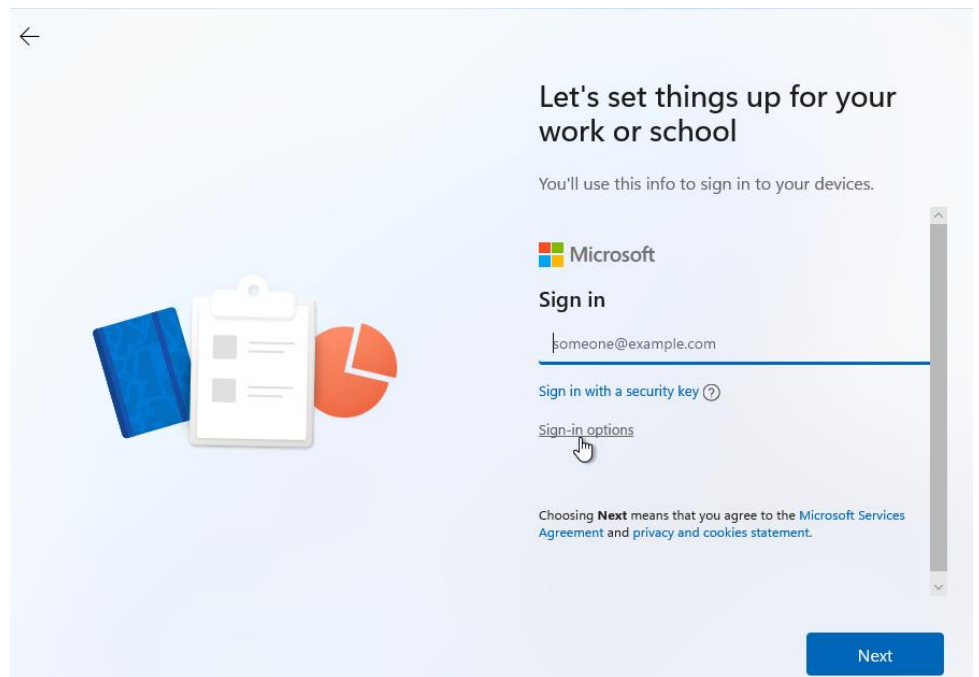
נבחר אם להוסיף עוד שפה למקלדת (ניתן להוסיף בהמשך)



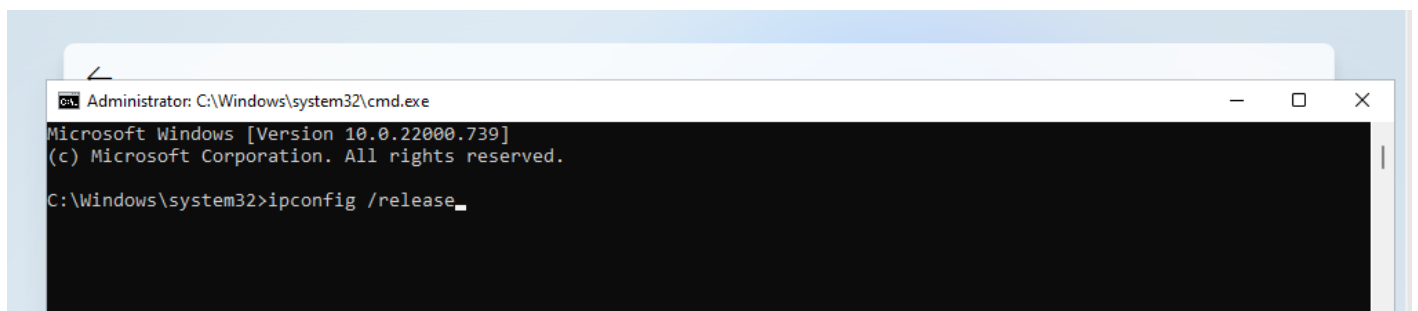
נמתין שהמע' תסיים לחפש עידכונים



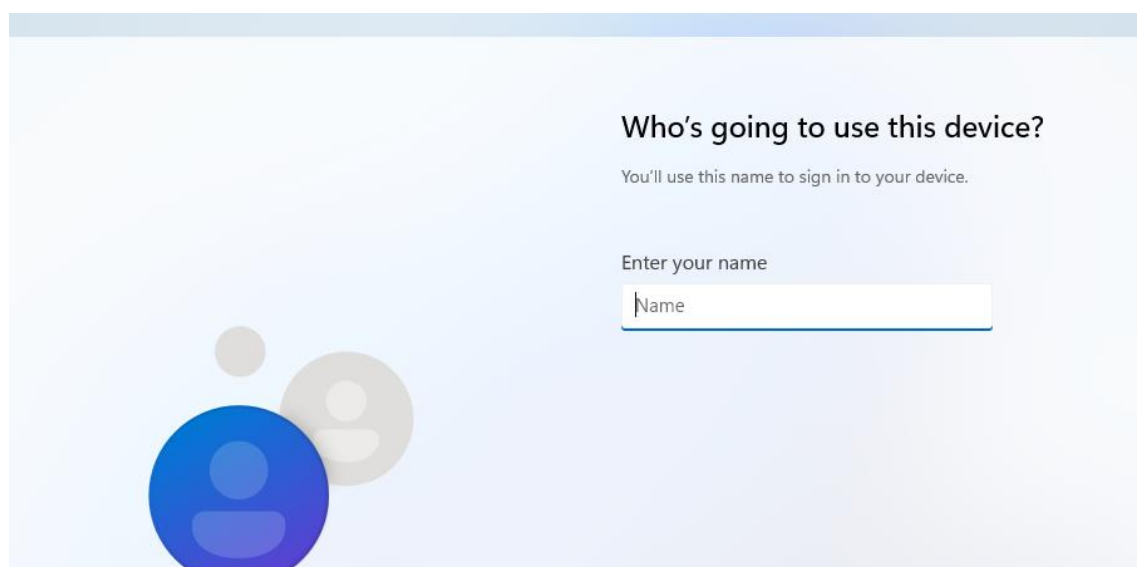
במסך הזה נבחר אפשרויות התחברות אחרות על מנת ליצור חשבון מקומי



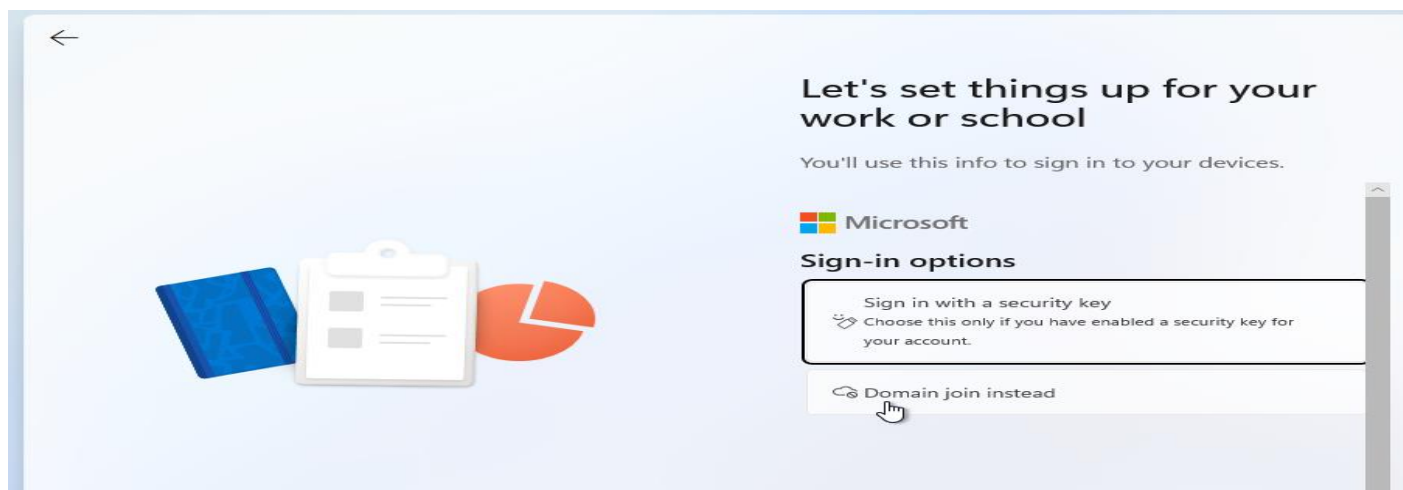
אם אתם מתקינים גרסת HOME ניתן לבצע מעקף ולנטרל את האינטרנט זמנית על ידי לחיצה על Shift+F10 ובשורת הפקודה לכתוב `ipconfig /release`. הפקודה תבטל את הגדרות ה-IP



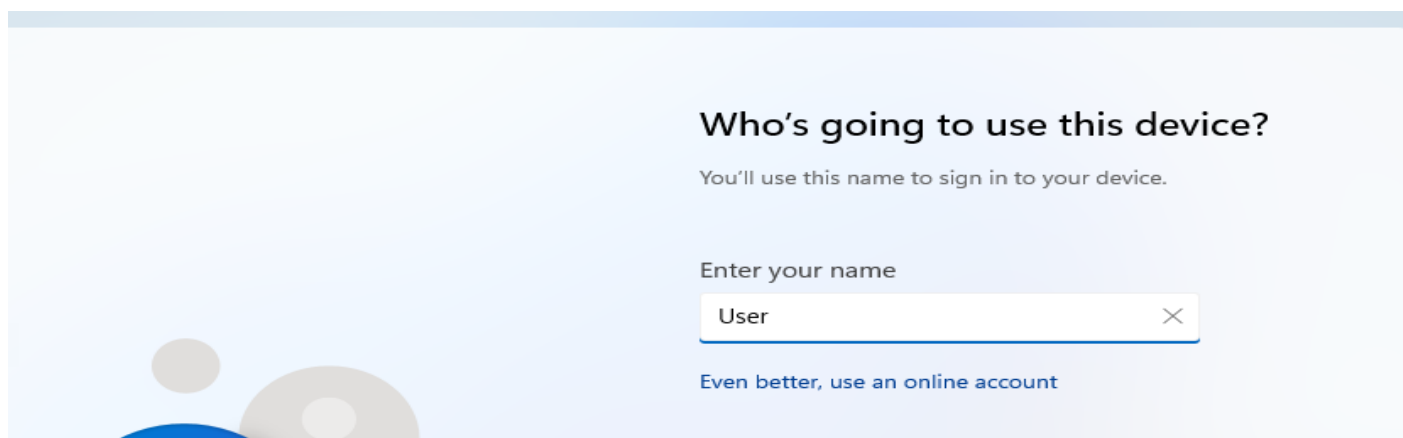
לחיצה על כפתור חזור תביא אותנו ישירות למסך הבא : נציין את שם המשתמש שיהיה האדמין המקומי



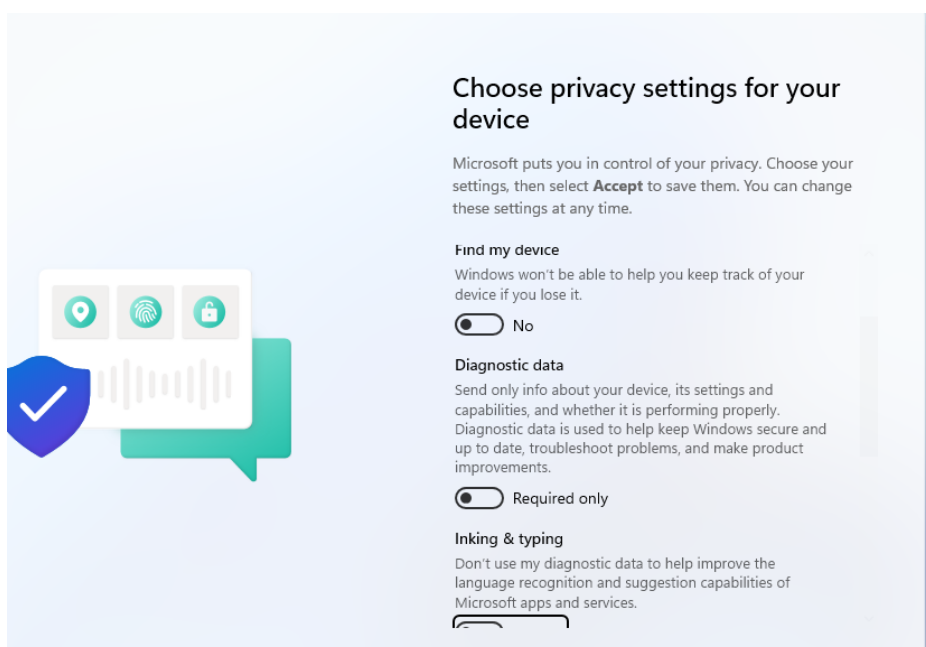
בגרסת ה-Enterprise נבחר להצטרף לדומיין ואז במסך הבא נבחר שם משתמש שיהיה האדמין המקומי



בחרתי בשם המשתמש USER ובמסך הסיסמה לחצתי אנטר (סיסמה ריקה)



בהגדרות הפרטיות נוכל לבחור לאפשר הכל או לנטרל את ההגדרות

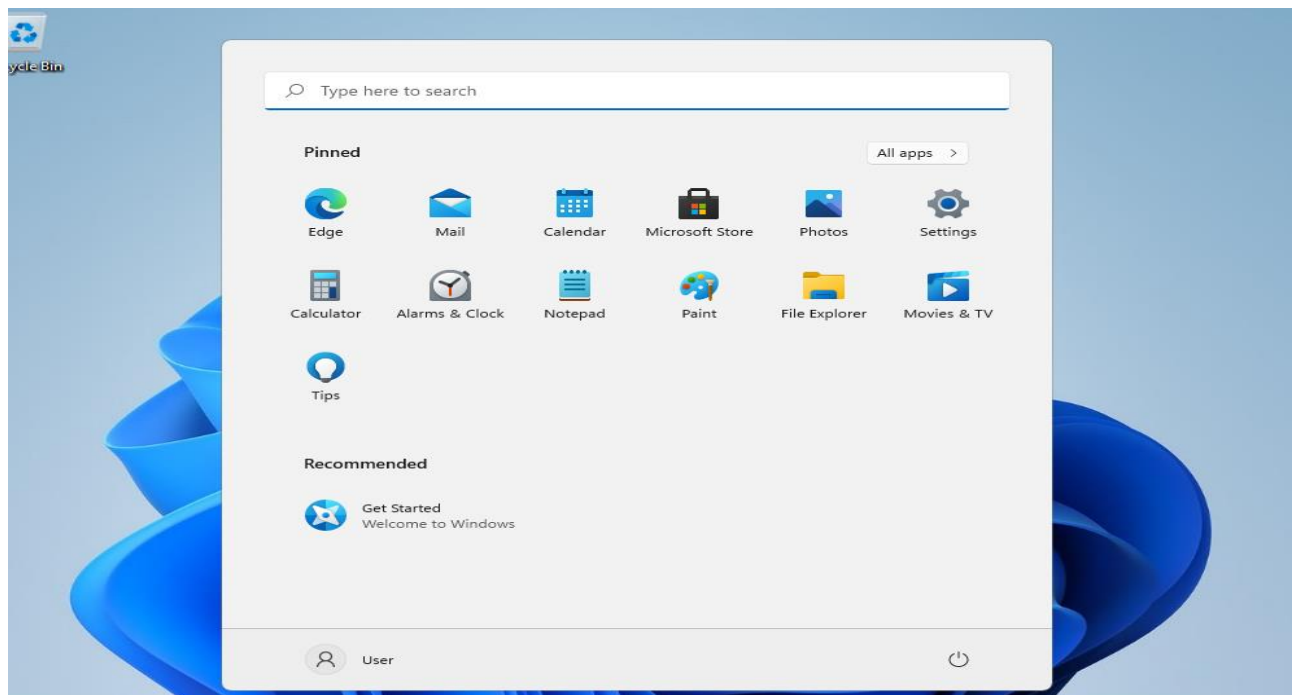


נמתין לכניסה אל שולחן העבודה

This might take a few minutes.

Don't turn off your PC

ניכנס אל שולחן העבודה



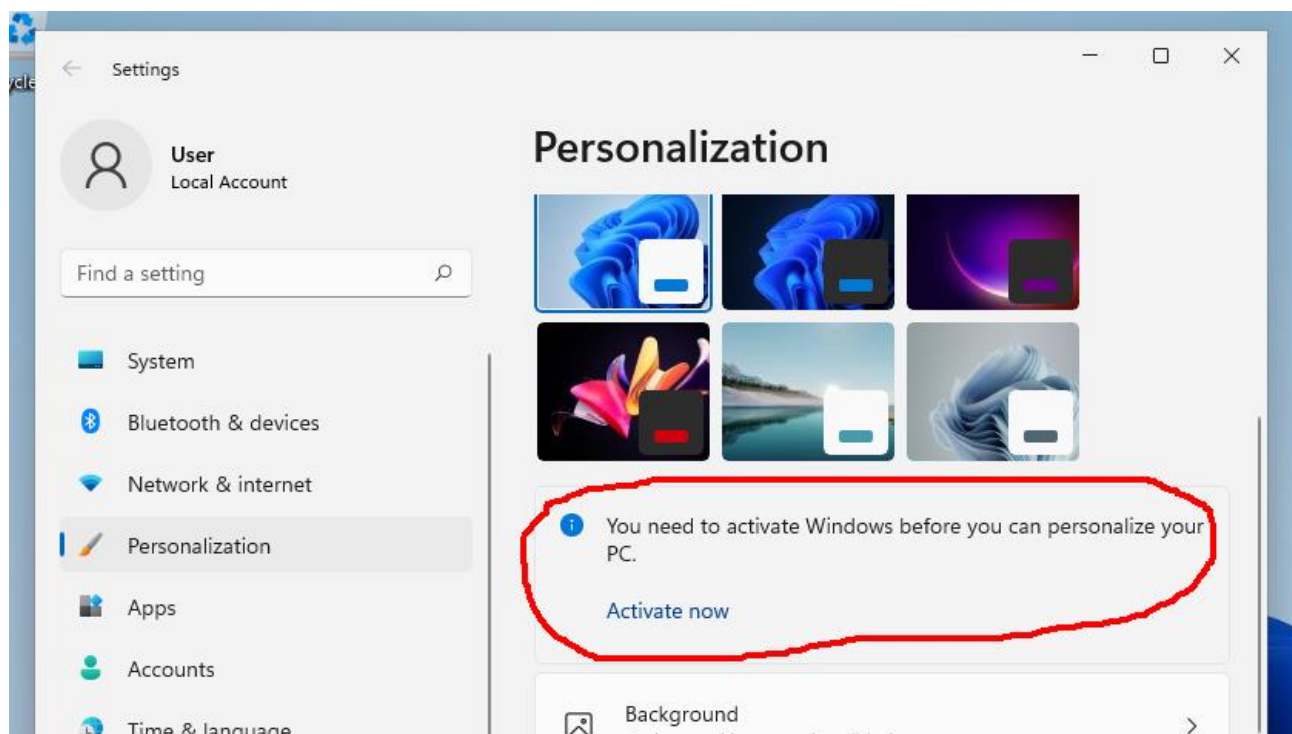
בשלב הבא נצרך לבצע אקטיבציה של ווינדוס אם נרצה את היכולת לעבוד עם מע' ההפעלה ולבצע התאמה אישית. הרשיון יכול להיות משלושה סוגים :

Retail – רכשנו את מע' ההפעלה ישירות ממיקרוסופט או מחנות והתקנו על המחשב שלנו

OEM – קיבלנו את מע' ההפעלה מותקנת במחשב הנייד או במחשב שקנינו מספק ציוד

Volume - רישוי ארגוני המבוצע בד"כ עם מפתח יחיד שמאפשר מספר רב של הפעלות , MAK KEY ובארגון עם מספר רב של מחשבים ישנו שרת המבצע את האקטיבציה והרישום של המחשבים , לשרת זה קוראים שרת KMS

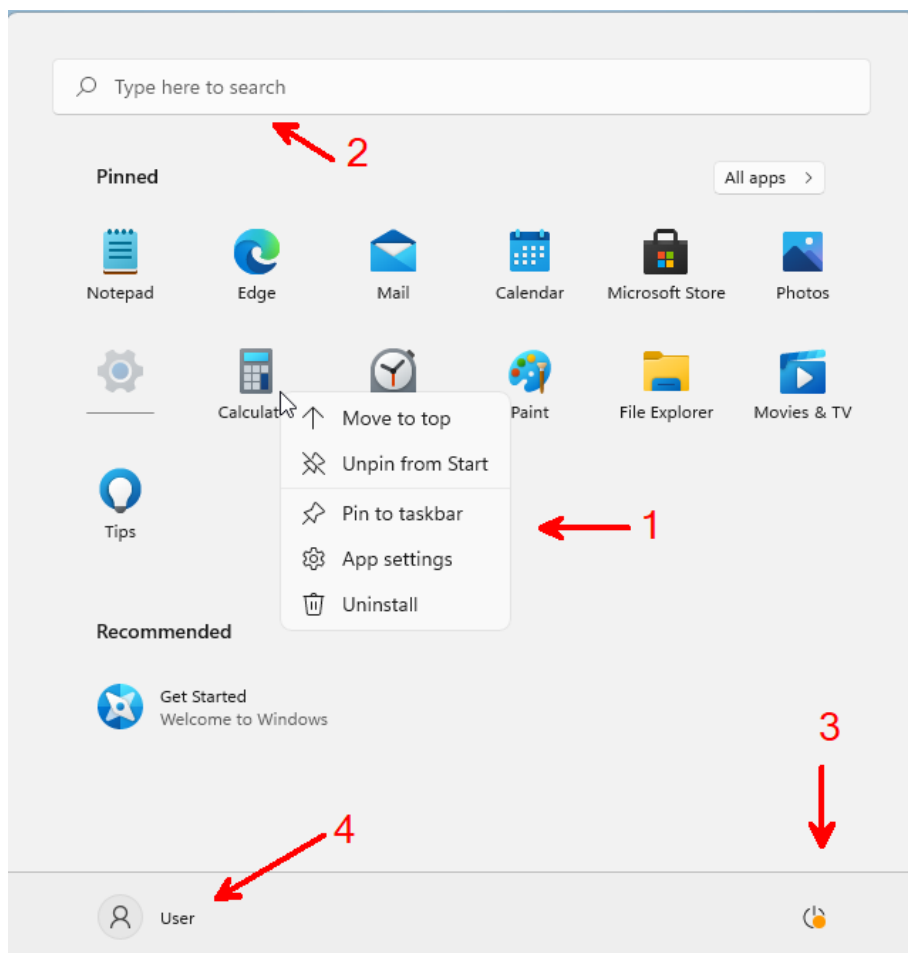
מחשב לא מאוקטב לא מאפשר לבצע התאמה אישית וכמו כן המע' תבצע כיבוי בזמנים רנדומליים ללא אזהרה בתום תקופת החסד של חודש



התאמה אישית של ממשק המשתמש

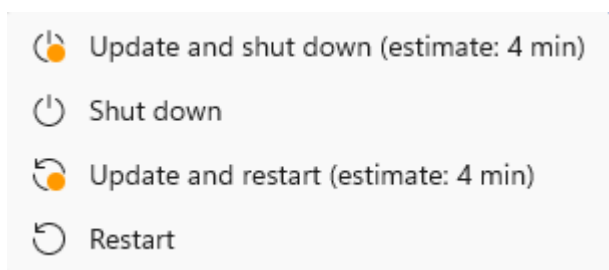
לאחר התקנה וביצוע אקטיבציה נרצה לבצע מס' התאמות אישיות לממשק המערכת.

נלחץ על כפתור התחל ונראה תפריט נקי יותר ומסודר יותר מאשר בווינדוס 10

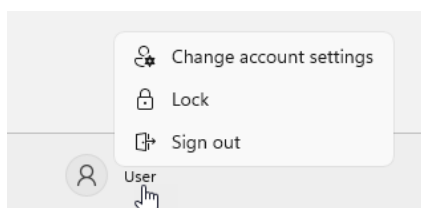


1. לחיצה על כפתור ימני בעכבר באחת האפליקציות תפתח לנו תפריט משנה שיאפשר לנו להסיר את האפליקציה מהרשימה, לגשת להגדרות שלה במחשב ואף להסיר אותה בצורה מיידיית

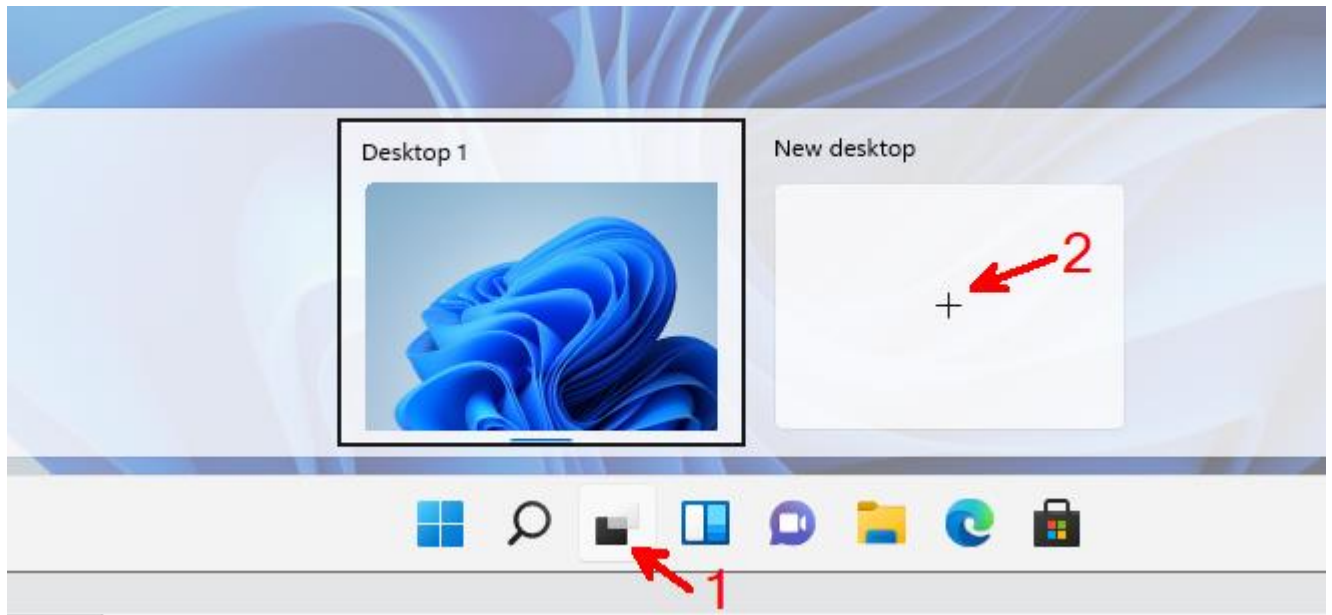
2. בשורת החיפוש נוכל לבצע חיפוש לאפליקציות, הגדרות ואף חיפוש אינטרנט



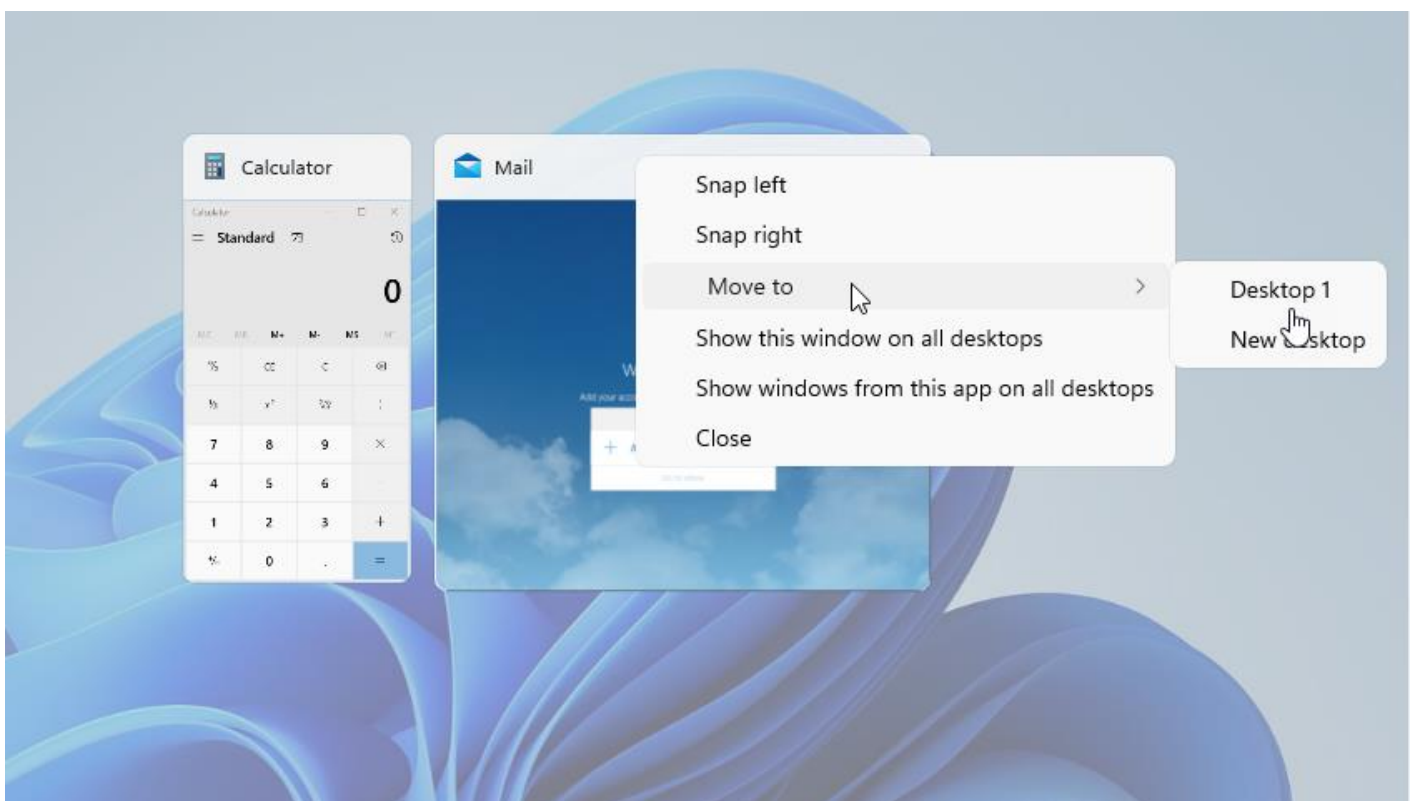
3. כפתור הכיבוי מאפשר לנו גישה מהירה לכיבוי המחשב, שימו לב שצבע הנקודה מעיד על כך שהמחשב צריך לקבל עדכונים



4. לחיצה על שם המשתמש שלנו תאפשר לנו לבצע נעילה או יציאה מהמע' במהירות



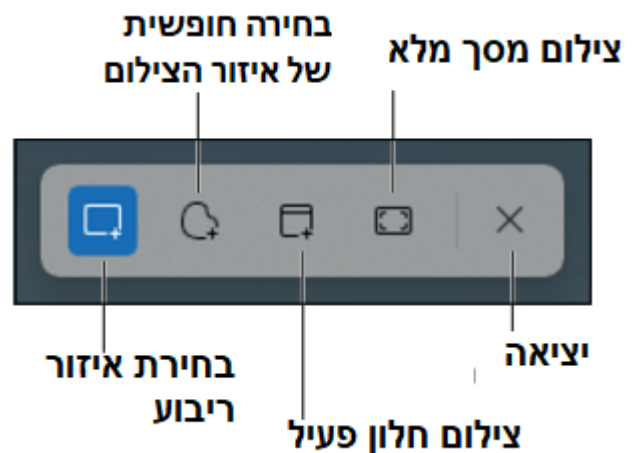
1 נלחץ על כפתור Task View (מסומן בתמונה) קיצור דרך Win+Tab, 2 נבחר לפתוח עוד שולחן עבודה
נוכל לפתוח בכל שולחן עבודה אפליקציה שונה ובכך לייעל את העבודה, נוכל אם נרצה להיכנס לתפריט ולהעביר כל אפליקציה לשולחן עבודה שנבחר.



ביצוע צילום מסך בווינדוס 11

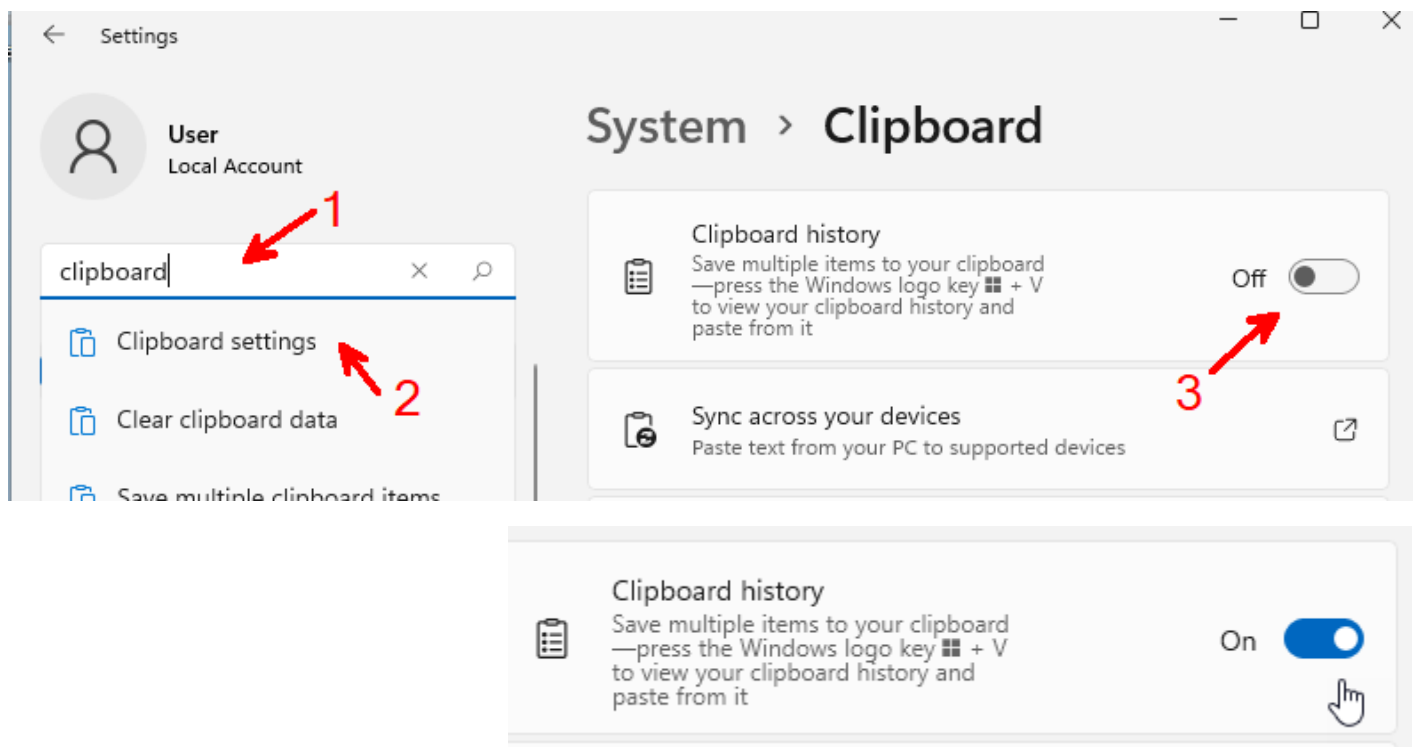
ווינדוס כוללת בתוכה יכולת לצילום מסך ללא צורך בתוכנה חיצונית, למעשה זהו שיפור של תוכנת לכידת המסך המוכרת מווינדוס 7.

כניסה לתוכנה Win+Shift+S - בראש המסך יופיע תפריט כמו בתמונה למטה

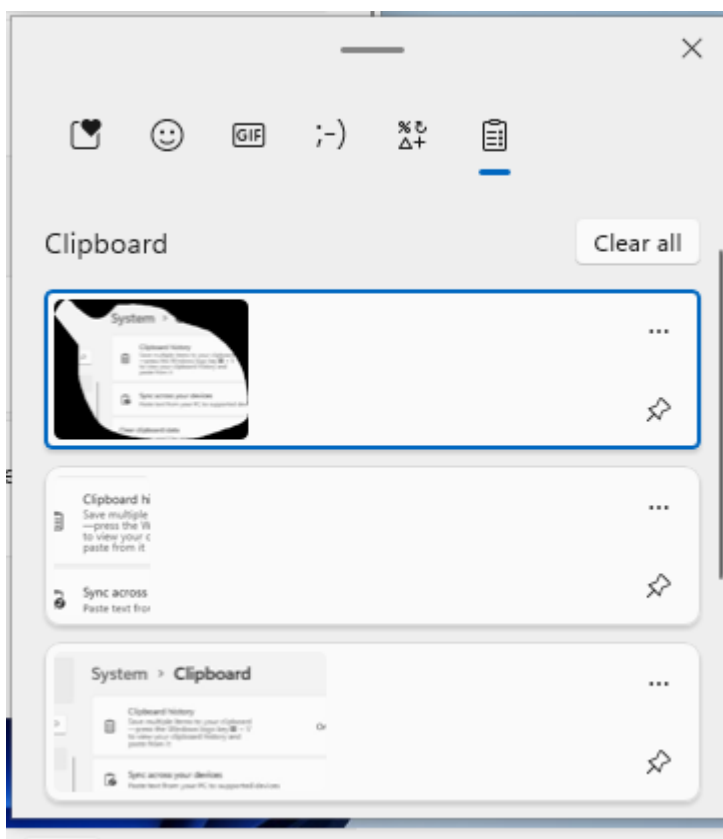


לאחר סימון ובחירה התמונה תועתק ישירות אל ה-Clipboard ונוכל להדביק אותה במגוון רב של תוכנות (וורד, אימיל, צייר וכו...)

מומלץ להפעיל עם תוכנה זו גם את התכונה ClipBoard History שמאפשרת לזכור מס' רב של צילומים והעתקות אל ה-Clipboard. תוכלו לבצע זאת ע"י כניסה אל ממשק ההגדרות ובחירה באפשרות הזו.



הדבר יאפשר לנו לבחור מס' רב של העתקות אל ה CLIPBOARD באמצעות כפתור Win+V



כמו כן נוכל לבחור להדביק גם GIF או סמיילים וכו....

כמה קיצורי מקלדת שימושיים :

Win+I - פותח את ההגדרות

Win+E - סייר הקבצים

Win+M - מוריד את כל האפליקציות הפעילות בשולחן העבודה למצב ממוזער

Win+X - תפריט הפעלה מקוצר (מאפשר גישה ישירה ל TERMINAL ועוד...)

CTRL+Shift+Esc – מנהל המשימות

Windows+Tab - קיצור דרך ל TASKVIEW

CTRL+X - חתוך... (בבחירת קבצים המשמעות היא העבר)

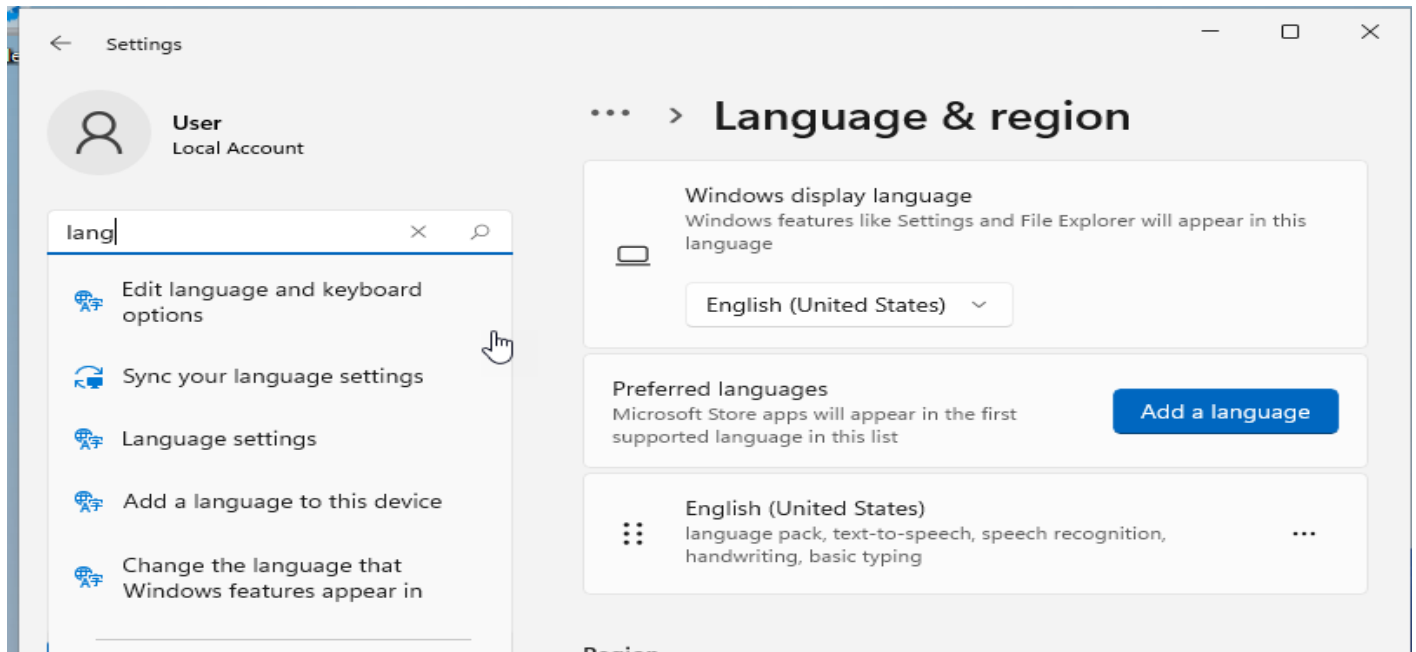
CTRL+V - הדבק

CTRL+C - הדבק

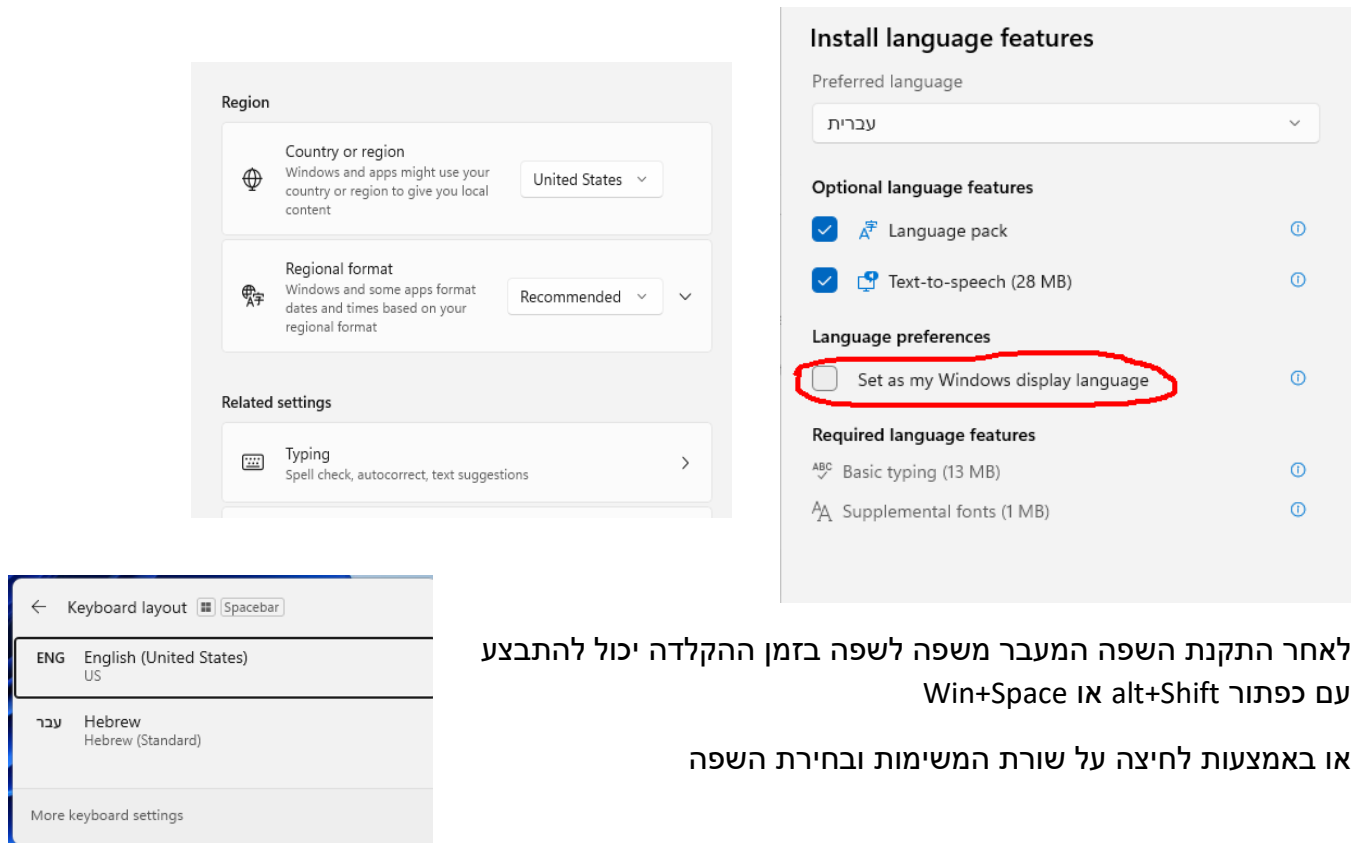
WIN+PrintScreen – צלם את כל המסך ושמור בתיקייה ScreenShots תחת התיקייה Pictures

הוספת תמיכת שפה לווינדוס 11 :

ווינדוס 11 מאפשרת לנו בקלות להוסיף שפת מקלדת או אפילו להחליף את שפת הממשק כולו
נוכל לבצע זאת באמצעות תפריט ההגדרות ובחירה באפשרויות שפה



בחרתי להוסיף שפה – שימו לב לאפשרות לבחירת שפת התצוגה (מצריך ביצוע יציאה וכניסה מחדש על מנת לעדכן). שימו לב שאם נרד בתפריט הבחירה נוכל לראות גם שניתן לשנות הגדרות איזוריות, דבר שישפיע על איזור הזמן, פורמט התאריך, המטבע המקומי ועוד...



לאחר התקנת השפה המעבר משפה לשפה בזמן ההקלדה יכול להתבצע
עם כפתור alt+Shift או Win+Space
או באמצעות לחיצה על שורת המשימות ובחירת השפה

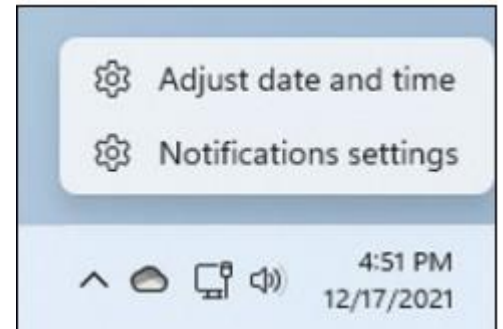
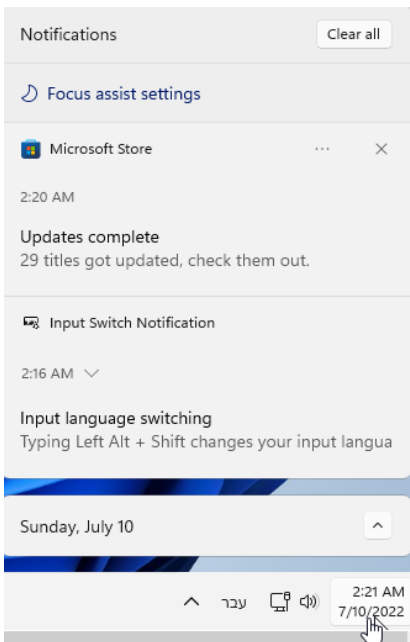
ניהול התראות בווינדוס 11 :

בשורת המשימות נלחץ על התאריך וניכנס לתפריט ההתראות

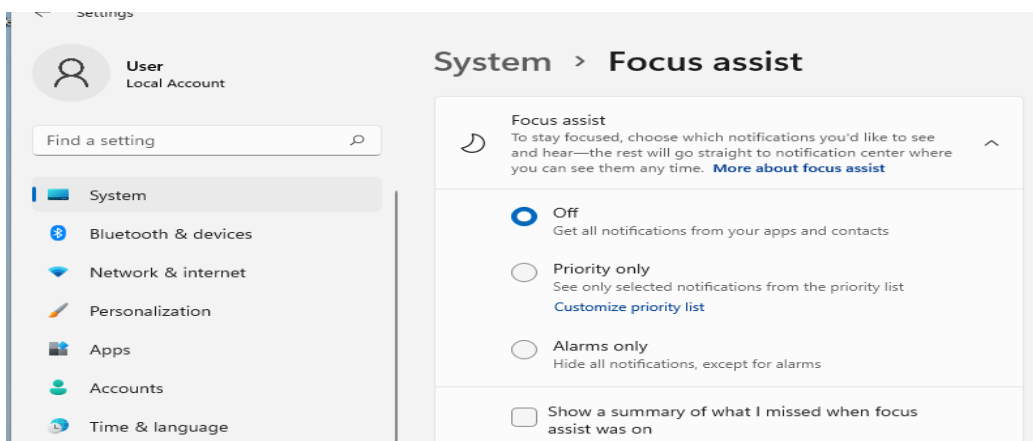
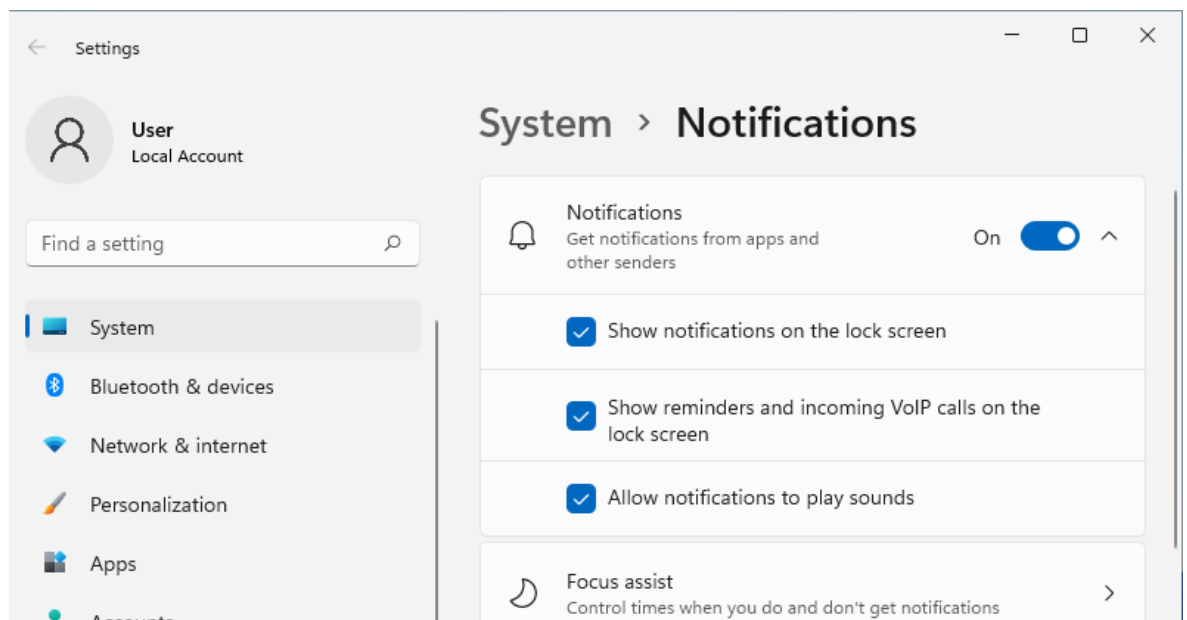
נוכל לעבור על ההתראות ואף למחוק אותן

אם נרצה לבטל התראות נוכל לעמוד על התאריך שוב והפעם לבחור כפתור ימני בעכבר

הדבר יביא אותנו להגדרות של של ההתראות



נוכל לבחור האם לקבל התראות והיכן

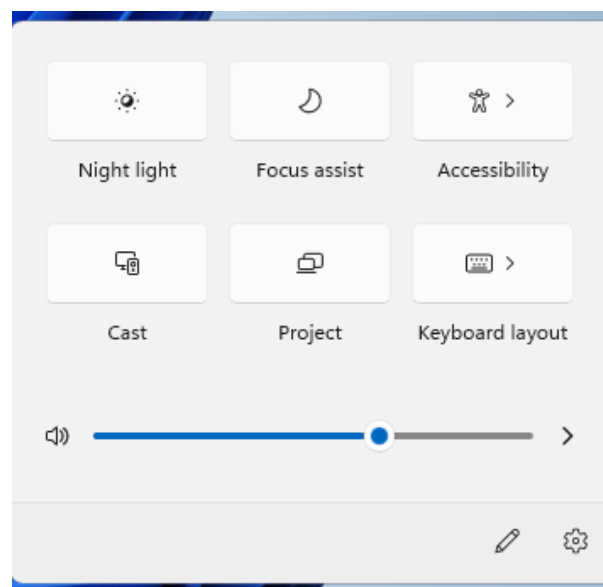


וכמו כן נוכל להגדיר מצב ממוקד שבו לא נקבל הודעות לפרק זמן מסויים או רק התראות חשובות בלבד...

נוכל גם לבחור את פרקי הזמן שבהם נרצה להיות ממוקדים ללא הפרעה.

שימוש בתפריט פעולות מהירות

לחיצה על סמל הרשת או הסוללה במידה ויש תביא אותנו לתפריט פעולות מהירות שם נוכל לבחור את הפעולות הבאות



שם הפעולה	מה מבוצע
Accessibility	יביא אותנו לתפריט כלי נגישות – שם נוכל לבחור להגדיל את המסך, להפעיל את היכולת להקריא את המידע מהמסך באנגלית ועוד כלי נגישות.
Airplane mode	מצב טיסה ינתק את כל התקשורת האינטרנטית כולל הגד' WIFI
Battery saver	מצב חיסכון סוללה יעמעם את בהירות המסך – לא פעיל כאשר מחוברים למקור מתח חשמלי
Bluetooth	יאפשר כיבוי והדלקה של התכונה
Brightness	שליטה על עוצמת בהירות המסך
Cast	יחפש התקנים התומכים ב Wireless Display על מנת לשדר להם את המסך והאודיו ישנה תמיכה מובנת בתקן Miracast
Focus assist	יכבה או יפעיל את האפשרות למיקוד
Keyboard layout	יבצע החלפה בין שפות המקלדת המותקנות
Project	יאפשר התחברות למקרן או למסך נוסף (קיצור דרך Win+P)
Rotation Lock	ינעל את המסך מביצוע סיבוב אוט' – שימושי כאשר עובדים עם טאבלטים
Volume	שליטה על עוצמת השמע

שדרוג או התקנה נקייה....

כאשר אנו נמצאים כבר עם מע' הפעלה קיימת נוכל לבחור לשדרג את המע' אל המע' החדשה. הדבר אומר לבצע את ההתקנה מתוך מע' ההפעלה הקיימת ולא לבצע אתחול עם דיסק ההתקנה.

ישנן שתי גישות לשדרוג הראשונה ביצוע שידרוג מלא : In-Place Upgrade כלומר ביצוע התקנה בזמן שמע' ההפעלה הקיימת פועלת . בטבלה נוכל לראות את מסלולי השדרוג

גירסה קיימת של ווינדוס	ווינדוס 11 HOME	ווינדוס 11 PRO	ווינדוס 11 ENT
ווינדוס 7			
ווינדוס 10 HOME	X	X	
ווינדוס 10 PRO		X	X
ווינדוס 10 ENT			X

בטבלה נוכל לראות שאין מסלול ישיר לשדרוג מווינדוס 7 אל ווינדוס 11

היתרונות של אפשרות זו הינן שמירה על כל ההגדרות כולל התוכנות המותקנות במחשב . חיסרון משמעותי הינו שאנו למעשה מחליפים את הממשק ונשארים עם המון הגדרות וקבצים ישנים ומיותרים שיכולים להאט את הביצועים שלנו. אנו נשקול אפשרות זו רק במידה ויש לנו תוכנה חשובה שאנו לא רוצים לוותר עליה ואין לנו את קבצי ההתקנה שלה.

אפשרות נוספת היא ביצוע הגירה – המשמעות היא שמירה על כל הגדרות המשתמש והמסמכים החשובים (ללא האפליקציות) וביצוע התקנה מחדש של מע' ההפעלה (יכול להיות באותו המחשב) ואז נמחק את המע' הישנה ונבצע התקנה נקייה או התקנה במחשב חדש לגמרי לדוגמא משתמש שעבר ממחשב ישן עם ווינדוס 7 למחשב חדש עם ווינדוס 11.

היתרון הוא שאנו מקבלים הזדמנות למעשה להתחיל בצורה נקייה ומה שיעבור הינו רק הגדרות המשתמש והקבצים החשובים לנו .

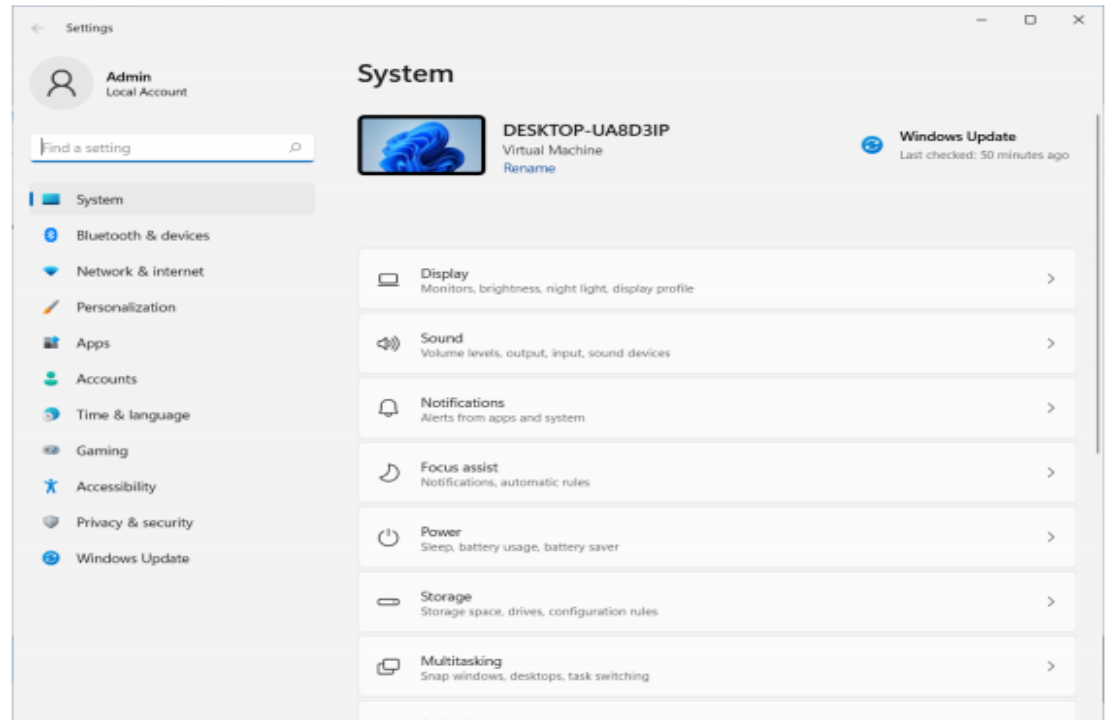
יש לזכור שהאפליקציות צריכות להיות מותקנות מחדש ולכן רצוי לברר אם האפליקציות יכולות לפעול בסביבת ווינדוס 11 לפני שנקבל את ההחלטה .

בארגונים ישנם כלים לביצוע אוטומציה של פעולות ההתקנה וההפצה של מע' ההפעלה הן ברמה הארגונית והן דרך הענן. בקורסי ניהול רשתות אנו מתמקדים גם בכלים אילו.

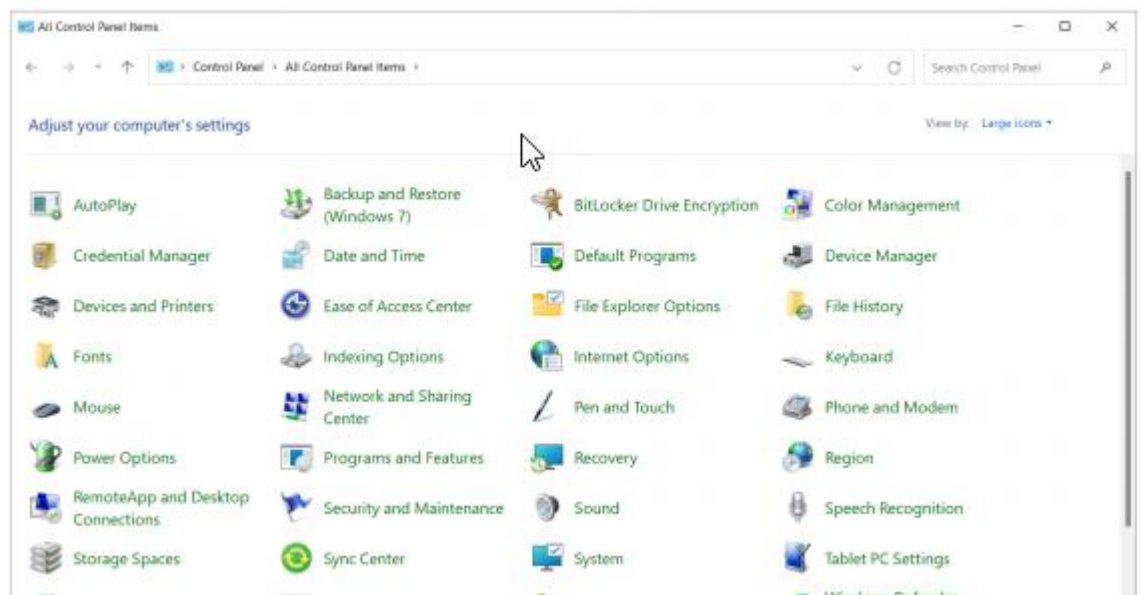
שימוש בכלי הניהול של ווינדוס 11 :

ישנם מספר כלי ניהול שמאפשרים לנו לבצע שינויי הגדרות ואיתור תקלות , בהמשך החוברת נבצע סקירה מעמיקה של הכלים הללו אך חשוב שנכיר אותם כבר בשלב זה .

אפליקציית Settings – תפריט ההגדרות הינו למעשה אפליקציית ווינדוס – גישה מהירה באמצעות Win+I , פה נוכל להגדיר כמעט כל הגדרה של ווינדוס ובשונה מווינדוס 10 שם היה משקל רב גם ללוח הבקרה אנו רואים לאט לאט שישנה מגמה לרכז את כל ההגדרות תחת האפליקציה המודרנית

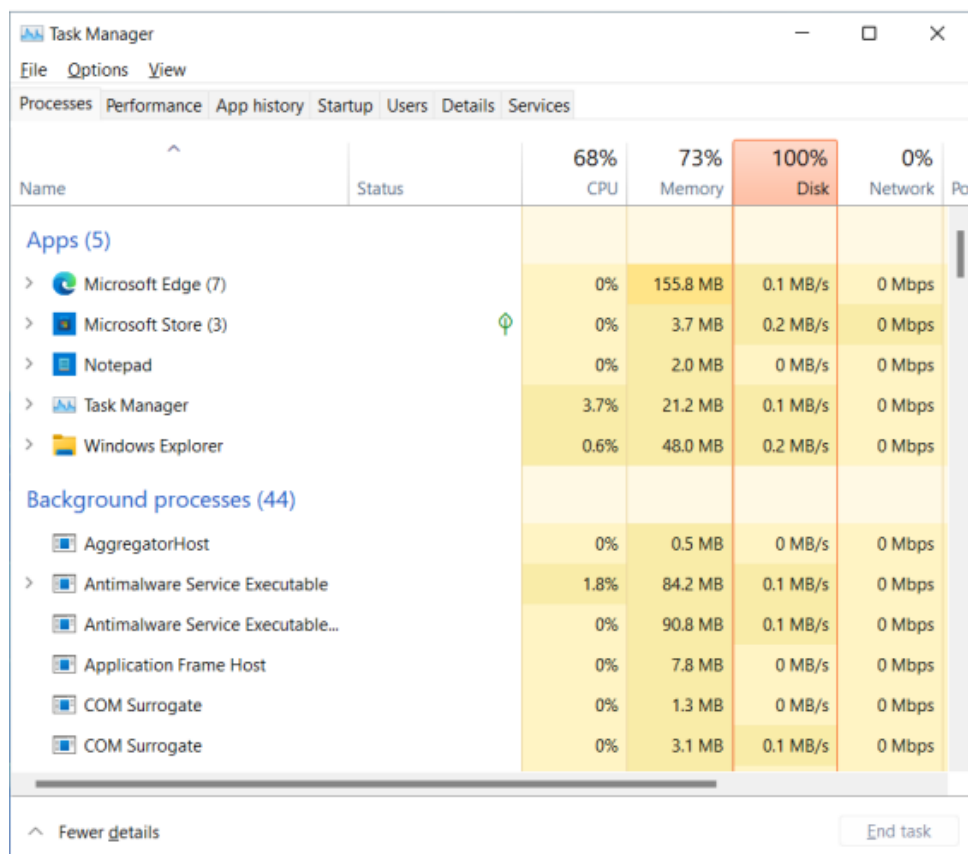


אחד הדברים היותר יעילים הוא לבצע שימוש ביכולת החיפוש על מנת למצוא את ההגדרה הרלוונטית עבורנו. עדיין נוכל להבחין שחלק מההגדרות נמצאות בלוח הבקרה ונוכל עדיין להשתמש בו

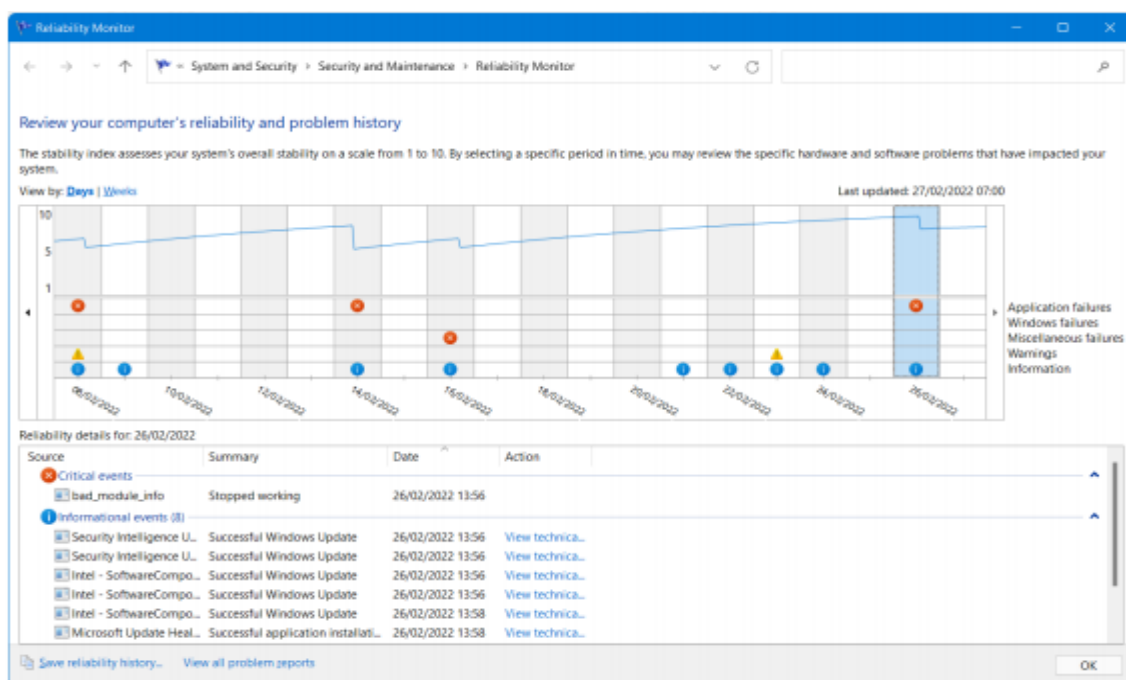


הכניסה יכולה להתבצע בחיפוש של המילה Control בתפריט התחל

כלי נוסף שנשתמש בו הינו Task Manager הגישה אליו תבוצע באמצעות CTRL+Shift+Esc והוא יאפשר לנו לקבל מידע על האפליקציות הרצות שירותי המע' ועוד... נלמד עליו יותר לעומק בפרק בדיקות הביצועים



מד האמינות – כלי זה ירכז מידע חשוב הנוגע לאירועים המשפיעים על ביצועי המערכת שלנו , הגישה אליו דרך תפריט התחל וחיפוש Reliability



המסך ייתן לנו תמצית מידע שממנה נוכל להתחיל לקבל תמונה על הדברים המשפיעים על ביצועי המערכת

MMC - קונסולות מייקרוסופט , כלים רבים בהם נשתמש הינם למעשה קונסולות ניהול המובנות במע' אך נוכל גם להשתמש בקונסולות שלא זמינות ישירות ולנהל שירותי מע' בצורה יעילה.

קונסולות זמינות :

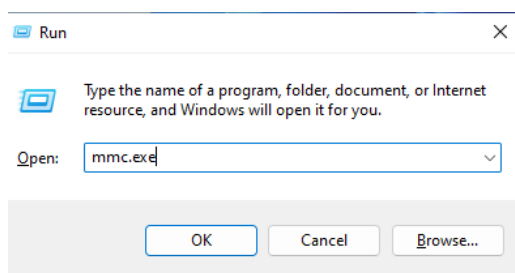
Diskmgmt.msc - משמש לניהול דיסקים

Compmgmt.msc – ניהול המחשב

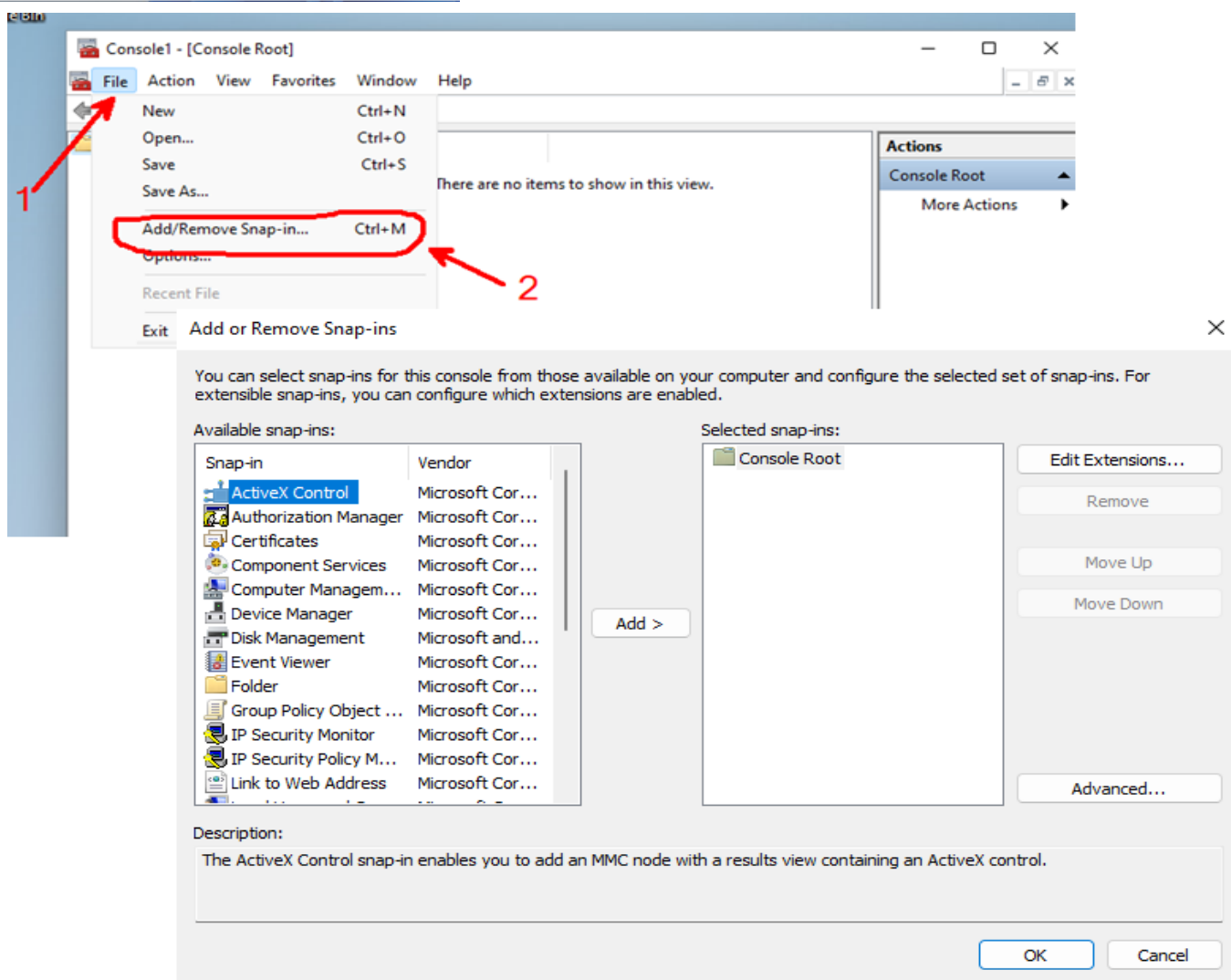
Lusrmgr.msc – ניהול משתמשים מקומיים

Devmgmt.msc – ניהול התקנים

כדי לראות את כל הקונסולות הזמינות עבורנו נוכל להיכנס בתפריט Run אל mmc.exe ולבחור להוסיף קונסולה



בתפריט שייפתח נבחר File ולאחר מכן Add Remove Snap-in



תוכלו להתרשם מכמות הקונסולות הזמינות עבורנו , בהמשך הקורס נבצע סקירה של חלק גדול מיכולות אילו

הכרות בסיסית עם ממשק הפקודה POWERSHELL

בעולם שפות הסקריפטים של מייקרוסופט עבדנו עם CMD ו-BATCH ויש כאלו שהתנסו עם VBSCRIPT שתי הטכנולוגיות הללו נחשבות מיושנות כעת והיות ומבחינת מייקרוסופט כולנו עוברים ל POWERSHELL. למנהלי הרשת היותר וותיקים זכורה בוודאי מע' ההפעלה DOS שהביאה לנו את COMMAND.COM, הממשק היחידי שהיה לנו היה מבוסס טקסט וכשהיינו רוצים לבצע פקודות ברצף היינו כותבים סקריפט באמצעות קבצי BATCH

בהמשך שחררה מיקרוסופט את VBscript על מנת להחליף את קבצי BATCH, השפה הייתה שימושית אך לא מתועדת מספיק ולא נתמכת בצורה אחידה במוצרי מייקרוסופט. את החלל הצליחה POWERSHELL למלא, מווינדוס 7 היא מובנת במע'. היתרון המשמעותי של PS הוא בכך שהשפה מבוססת על .NET. שהיא השפה שבה כתובה מע' ההפעלה.

כמו כן חשוב להבין שהשפה של PS היא Object Oriented – כלומר מונחת עצמים

הסבר קצר : תכנות מונחה עצמים

השיטה של תכנות מונחה עצמים (אובייקטים) הינה למעשה, חיקוי של חשיבת האדם, כלומר, היא אינטואיטיבית וקרובה יותר לדרך החשיבה של רוב האנשים המטפלים בבעיה העומדת בפניהם.

אנו באופן טבעי ואוטומטי מסווגים את כל העצמים על פי קטגוריות שונות. לדוגמא, העולם הפיזי מחולק על ידנו לחי, צומח ודומם. קיימת גם חלוקה פנימית יותר ובה את החי אנו מחלקים ליונקים, עופות, דגים וכך הלאה לחלוקות פנימיות יותר. כל עצם או מושג המוכר לנו בעולם הזה משתייך למספר רב של קטגוריות בסדר היררכי כלשהו. קל ונח יותר לשלוט בכמות האדירה של האינפורמציה בה אנו מוצפים וכך יעיל ונח יותר לסדר אינפורמציה זו במוחנו, בכדי שנוכל לשלפה ולזהותה בקלות בכל עת שנחפוץ.

נראה דוגמא – בבואנו לספר לחבר כי רכשנו רכב חדש אין צורך לפרט לו כי רכשנו מנוע, קרבורטור(מאייד), מצמד, דוושת דלק, רדיאטור, הגה וכו'.... אלא בעצם הגדרת המושג רכב יידע החבר כי כל אלו נכללים, ומכאן שאין צורך בפירוט מצדנו וזה יתרון עצום! אחרת עבור כל רכב היינו צריכים לציין את כל מרכיביו, בעוד שבזכות העובדה כי אנו מסווגים ומקטלגים את הדברים כל שנצטרך לציין הן התכונות המייחדות את הרכב שלנו ולא את המשותפות לכלל הרכבים.

תכונה נוספת המאפיינת את תהליך החשיבה של האדם הינה – שיוך של פעולה לעצם כלשהו. למשל, נשווה בין שתי הפעולות הבאות: הדלקת נר והדלקת המחשב – לכאורה, מבחינה לוגית משמעותן של שתי הפעולות זהה – שתיהן פעולות "הדלקה", אך הנגזרת של כל פעולה היא מימוש פיזי אחר – כלומר, סידרת הצעדים אשר נבצע שונה לחלוטין בין פעולה אחת לאחרת.

ומכאן כשנתבקש לבצע פעולת "הדלקה" אנו נדע איזו פעילות עלינו לבצע על פי שיוך של הפעולה לעצם אותו אנו מתבקשים להדליק. כלומר, קיים קשר חשוב בין הפעולה לבין העצם עליו או איתו היא מתבצעת.

שני התהליכים החשיבתיים שהגדרנו אצל בני האדם מיושמים בצורה דומה בגישה מונחית העצמים ע"י יצירת טיפוס חדש המכונה Class (מחלקה) המאפשר ריכוז של מאפיינים (משתנים) ומטודות (פונקציות) תחת הגדרה אחת.

לאחר הגדרת המחלקה ניתן לייצר מופעים שלה, כל מופע של מחלקה קרוי Object (אובייקט). כאשר המחלקה הינה תבנית ליצירת אובייקטים מסוגה, כלומר, המחלקה מהווה את ההגדרה של המשתנים והשיטות שיכילו אח"כ כל מופעיה (האובייקטים מסוגה).

דוגמא מעשית ליכולות של PS בנושא עבודה עם אובייקטים :

בתרגיל הבא נפנה למחלקה האחראית על יצירת הדיבור במחשב ונסקור את הפרמטרים ואת הפעולות שהיא יודעת לבצע

ובאמצעות פקודות PS נגרום למחשב לדבר. הדוגמא הזו תמחיש את היכולת של PS לקרוא לכל מחלקה שקיימת במע' ההפעלה ולבצע פקודות ישירות עליה. כלומר PS משוחררת מכל מגבלה ומסוגלת לבצע כל פעולה אפשרית על מע' ההפעלה. אם בעולם הישן היינו מוגבלים בפקודות CMD לקוד שנכתב עבור הפקודה כעת יש לנו חופש מוחלט לבצע כל מה שנרצה במע' ההפעלה אם רק נכיר את המחלקות והאובייקטים הנכונים.

נפתח ממשק PS ונקליד את הפקודה הראשונה :

Add-Type -AssemblyName System.speech

```
PS C:\Users\Administrator> Add-Type -AssemblyName System.speech
PS C:\Users\Administrator>
```

למעשה אנו טוענים רכיב DLL הנמצא במע' ההפעלה ומכיל את המחלקה

<https://docs.microsoft.com/en-us/dotnet/api/system.speech.synthesis.speechsynthesizer?view=netframework-4.8>

SpeechSynthesizer Class

Namespace: System.Speech.Synthesis

Assembly: System.Speech.dll

Provides access to the functionality of an installed speech synthesis engine.

בשלב הבא נרצה ליצור משתנה שיכיל את האובייקט אותו טענו לזיכרון :

\$speak = New-Object System.Speech.Synthesis.SpeechSynthesizer

```
$speak = New-Object System.Speech.Synthesis.SpeechSynthesizer
```

על מנת לבחור את האובייקט שיצרנו נרשום את הפקודה הבאה :

\$speak | Get-Member

רשמנו את המשתנה ובאמצעות PIPE הסימן : | כתבנו את הפקודה Get-MEMBER או GM בקיצור על מנת לראות את כל המאפיינים והפעולות של האובייקט

```
PS C:\Windows\System32\WindowsPowerShell\v1.0\en-US> $speak | Get-Member

TypeName: System.Speech.Synthesis.SpeechSynthesizer

Name MemberType Definition
-----
BookmarkReached Event System.EventHandler<1[System.Speech.Synthesis.BookmarkReachedEventArgs]>
PhonemeReached Event System.EventHandler<1[System.Speech.Synthesis.PhonemeReachedEventArgs]>
SpeakCompleted Event System.EventHandler<1[System.Speech.Synthesis.SpeakCompletedEventArgs]>
SpeakProgress Event System.EventHandler<1[System.Speech.Synthesis.SpeakProgressEventArgs]>
SpeakStarted Event System.EventHandler<1[System.Speech.Synthesis.SpeakStartedEventArgs]>
StateChanged Event System.EventHandler<1[System.Speech.Synthesis.StateChangedEventArgs]>
VisemeReached Event System.EventHandler<1[System.Speech.Synthesis.VisemeReachedEventArgs]>
VoiceChange Event System.EventHandler<1[System.Speech.Synthesis.VoiceChangeEventArgs]>
AddLexicon Method void AddLexicon(uri uri, string mediaType)
Dispose Method void Dispose(), void IDisposable.Dispose()
Equals Method bool Equals(System.Object obj)
GetCurrentlySpokenPrompt Method System.Speech.Synthesis.Prompt GetCurrentlySpokenPrompt()
GetHashCode Method int GetHashCode()
GetInstalledVoices Method System.Collections.ObjectModel.ReadOnlyCollection[System.Speech.Synthesis.Voice] GetInstalledVoices()
GetType Method type GetType()
Pause Method void Pause()
RemoveLexicon Method void RemoveLexicon(uri uri)
Resume Method void Resume()
SelectVoice Method void SelectVoice(string name)
SelectVoiceByHints Method void SelectVoiceByHints(System.Speech.Synthesis.VoiceGender gender), void SelectVoiceByHints(System.Speech.Synthesis.VoiceGender gender, System.Speech.Synthesis.VoiceGender gender)
SetOutputToAudioStream Method void SetOutputToAudioStream(System.IO.Stream audioDestination, System.Speech.Synthesis.Prompt speakAsyncCancel)
SetOutputToDefaultAudioDevice Method void SetOutputToDefaultAudioDevice()
SetOutputToWaveFile Method void SetOutputToWaveFile(string path), void SetOutputToWaveFile(string path, System.Speech.Synthesis.Prompt speakAsyncCancel)
SetOutputToWaveStream Method void SetOutputToWaveStream(System.IO.Stream audioDestination), void SetOutputToWaveStream(System.IO.Stream audioDestination, System.Speech.Synthesis.Prompt speakAsyncCancel)
Speak Method void Speak(string textToSpeak), void Speak(System.Speech.Synthesis.Prompt textToSpeak), void Speak(System.Speech.Synthesis.Prompt textToSpeak, System.Speech.Synthesis.Prompt speakAsyncCancel)
SpeakAsync Method void SpeakAsync(string textToSpeak), void SpeakAsync(System.Speech.Synthesis.Prompt textToSpeak), void SpeakAsync(System.Speech.Synthesis.Prompt textToSpeak, System.Speech.Synthesis.Prompt speakAsyncCancel)
SpeakAsyncCancel Method void SpeakAsyncCancel(System.Speech.Synthesis.Prompt speakAsyncCancel)
SpeakSsml Method void SpeakSsml(string textToSpeak), void SpeakSsml(System.Speech.Synthesis.Prompt textToSpeak), void SpeakSsml(System.Speech.Synthesis.Prompt textToSpeak, System.Speech.Synthesis.Prompt speakAsyncCancel)
SpeakSsmlAsync Method void SpeakSsmlAsync(string textToSpeak), void SpeakSsmlAsync(System.Speech.Synthesis.Prompt textToSpeak), void SpeakSsmlAsync(System.Speech.Synthesis.Prompt textToSpeak, System.Speech.Synthesis.Prompt speakAsyncCancel)
ToString Method string ToString()
Rate Property int Rate {get;set;}
State Property System.Speech.Synthesis.SynthesizerState State {get;}
Voice Property System.Speech.Synthesis.VoiceInfo Voice {get;}
Volume Property int Volume {get;set;}
```

אנו יכולים לראות שהאובייקט מחולק למאפיינים , Method (פעולות) ו-Evnet (פעולות שמבצעות רישום או תגובה על מנת להפעיל טריגר)

נוכל להבחין שיש פה שיטה (פעולה) שנקראת Speak שלוקחת מחרוזת ומוציאה ממנו קול :

SetOutputWaveStream	Method	void SetOutputWaveStream(System.IO.Stream addrobect m...
Speak	Method	void Speak(string textToSpeak), void Speak(System.Speec...
SpeakAsync	Method	System.Speech.Synthesis.Prompt SpeakAsync(string textTo...

`$speak.Speak("This is PowerShell says Hello")`

נכתוב את הפקודה ונוכל לשמוע דרישת שלום מהמחשב

```
PS C:\Users\Administrator> $speak.Speak("This is PowerShell says Hello")
PS C:\Users\Administrator>
```

כמה משחקים קטנים :

`$speak.voice`

יראה לנו את הקול הנוכחי באמצעותו מופעל הדיבור

`$speak.GetInstalledVoices().VoiceInfo`

יראה לנו את הקולות המותקנים במחשב

`$speak.SelectVoice('Microsoft Zira Desktop')`

יאפשר לנו לבחור קול אחר מהרשימה

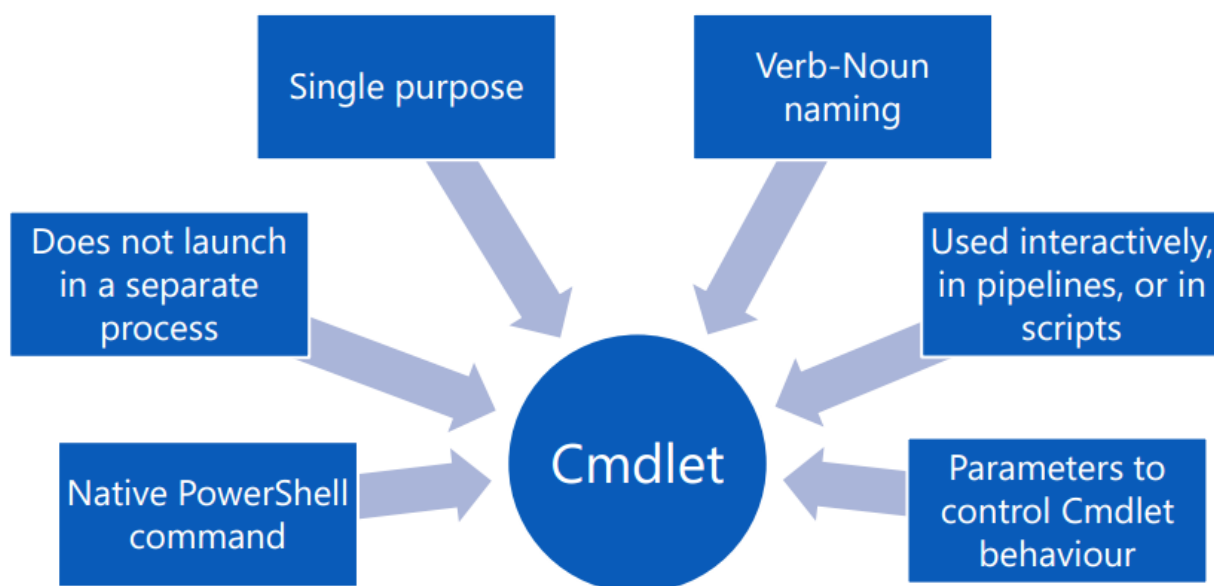
מה למדנו מהדוגמא הזו ?

למדנו שPS יודעת לבצע קריאה לכל אובייקט שנמצא במחשב לא משנה אם זה אאוטלוק או AD

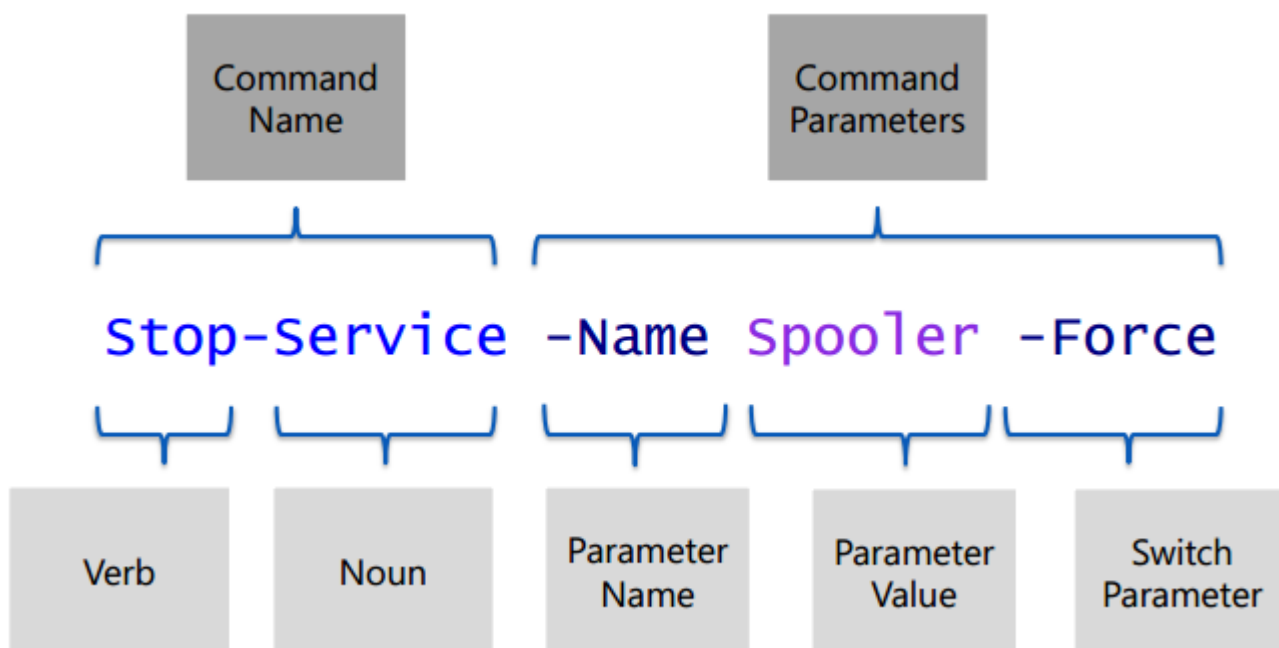
ראינו שכל אובייקט מורכב ממחלקות וכל מחלקה מכילה פרמטרים שונים ופעולות , נוכל לגשת למידע הזה באמצעות הפקודה Get-Member או GM בקיצור.

PS מכילה מראש המון פקודות אשר חוסכות מאיתנו את הצורך לגשת לאובייקט ולבצע עליו פעולה בצורה ישירה , לדוגמא הפקודה SET-ADuser מבצעת שינויים הקשורים למשתמש על האובייקט המנהל את בסיס הנתונים של AD .

PS מכיל תאימות לאחר לפקודות ואפליקציות CMD , אך המבנה של פקודת PS שונה מפקודות אילו. מבנה הפקודה מחולק לשניים VERB-NOUN – שם פעולה ושם עצם , או בקיצור תעשה משהו למישהו. לפקודות PS קוראים בשפה מקצועית Cmdlet. תכונות של פקודת PS נראה בתרשים הבא :



נוכל לסקור את הפעלים השונים באמצעות הקלדה של פקודת Get-Verb , בתמונה הבאה מבנה של פקודה טיפוסית:



```
PS C:\> Test-Connection -ComputerName 2012R2-MS -Count 1 -Quiet
True
```

אפשרות מעניינת היא להוסיף לפקודה את ה-SWITCH Whatif ואז נוכל לקבל מידע על מה הפקודה תבצע במידה ונחליט להריץ אותה, ה-SWITCH Confirm יאפשר לנו להחליט אם לבצע את הפקודה או לא.

```
PS C:\> Stop-Process -Name * -WhatIf
```

```
What if: Performing the operation "Stop-Process" on target "AcroRd32 (8160)".
What if: Performing the operation "Stop-Process" on target "AcroRd32 (12756)".
What if: Performing the operation "Stop-Process" on target "armsvc (2468)".
What if: Performing the operation "Stop-Process" on target "atieclxx (3220)".
```

במידה ונרצה להריץ שתי פקודות אחת אחר השניה נשתמש בסימן נקודה פסיק ;

```
PS C:\> Get-Service BITS ; Get-Process System
```

אם נרצה לחפש פקודות נוכל להשתמש בפקודה : Get-Command ולאחריה את שם הפקודה או חלק מהאותיות שמרכיבות את שמה. בדוגמא למטה ביקשתי לראות את כל הפקודות המכילות את שם הפועל SET והמילה DNS מופיעה בשם העצם

```
PS C:\Users\Administrator> Get-Command set-*dns*
```

CommandType	Name	Version	Source
Function	Set-DnsClient	1.0.0.0	DnsClient
Function	Set-DnsClientGlobalSetting	1.0.0.0	DnsClient
Function	Set-DnsClientNrptGlobal	1.0.0.0	DnsClient
Function	Set-DnsClientNrptRule	1.0.0.0	DnsClient
Function	Set-DnsClientServerAddress	1.0.0.0	DnsClient
Function	Set-NetDnsTransitionConfiguration	1.0.0.0	NetworkTransition
Function	Set-VpnConnectionTriggerDnsConfiguration	2.0.0.0	VpnClient
Cmdlet	Set-AzDnsRecordSet	1.1.2	Az.Dns
Cmdlet	Set-AzDnsZone	1.1.2	Az.Dns
Cmdlet	Set-AzFrontDoorCdnSecret	1.8.3	Az.Cdn
Cmdlet	Set-AzFrontDoorCdnSecurityPolicy	1.8.3	Az.Cdn
Cmdlet	Set-AzPrivateDnsRecordSet	1.0.3	Az.PrivateDns
Cmdlet	Set-AzPrivateDnsVirtualNetworkLink	1.0.3	Az.PrivateDns
Cmdlet	Set-AzPrivateDnsZone	1.0.3	Az.PrivateDns
Cmdlet	Set-AzPrivateDnsZoneGroup	4.16.1	Az.Network
Cmdlet	Set-AzSqlServerDnsAlias	3.8.0	Az.Sql

```
PS C:\Users\Administrator>
```

הפקודה SHOW-COMMAND ולאחריה פקודה כלשהי תפתח לנו ממשק גרפי שייאפשר לנו לראות את הפרמטרים של הפקודה ולהזין אותם. הדבר שימושי כאשר אנו רוצים להרחיב את ההכרות שלנו עם היכולות של הפקודה.

```
PS C:\Users\Administrator> Show-Command Set-DnsClient
```

The screenshot shows a PowerShell window with the command 'Show-Command Set-DnsClient' executed. A dialog box titled 'Set-DnsClient' is open, displaying the parameters for the 'Set-DnsClient' command. The dialog has tabs for 'ByAlias', 'ByName', and 'InputObject (cdxml)'. The 'ByAlias' tab is selected. The parameters listed are: InterfaceAlias: *, AsJob (checkbox), CimSession: (text box), Confirm (checkbox), ConnectionSpecificSuffix: (text box), PassThru (checkbox), RegisterThisConnectionsAddress: (text box), ResetConnectionSpecificSuffix (checkbox), ThrottleLimit: (text box), and UseSuffixWhenRegistering: (text box). At the bottom, there is a 'Common Parameters' section with a dropdown arrow. The 'Run', 'Copy', and 'Cancel' buttons are at the bottom right.

ל-PS יש מספר מנגנונים המאפשרים לנו לקבל עזרה, הפקודה Get-Help או שימוש בפרמטר -? יאפשרו לקבל עזרה עבור הפקודה.

```
PS C:\> Get-Help Get-ChildItem
#or
PS C:\> Get-ChildItem -?

NAME
    Get-ChildItem
SYNOPSIS
    Gets the files and folders in a file system drive.
SYNTAX
    Get-ChildItem [[-Path] <String[]>] [[-Filter] <String>]...
...
DESCRIPTION
    The Get-ChildItem cmdlet gets the items in one or more...
```

נוכל להשתמש בעוד אופרטורים על מנת להרחיב את המידע שאותו נקבל (נסו את זה אצלכם במחשב)

```
PS C:\> Get-Help Get-ChildItem
PS C:\> Get-Help Get-ChildItem -Full
PS C:\> Get-Help Get-ChildItem -Examples
PS C:\> Get-Help Get-ChildItem -Detailed
```

כמו כן תמיד נוכל להוסיף לכל פקודה את האפשרות Get-Member | או בקיצור GM

בתוצאה נקבל את המחלקה שאליה שייכת הפקודה עם כל המאפיינים, אירועים ושיטות (פעולות) אותן ניתן לבצע על האובייקט. נסו לכתוב פקודה ולהוסיף לה GM

```
PS C:\Users\Administrator> Get-Process | gm

TypeName: System.Diagnostics.Process

Name      MemberType Definition
-----
Handles   AliasProperty Handles = Handlecount
Name       AliasProperty Name = ProcessName
NPM        AliasProperty NPM = NonpagedSystemMemorySize64
PM         AliasProperty PM = PagedMemorySize64
SI         AliasProperty SI = SessionId
VM         AliasProperty VM = VirtualMemorySize64
WS         AliasProperty WS = WorkingSet64
Disposed  Event        System.EventHandler Disposed(System.Object, System.EventArgs)
ErrorDataReceived Event        System.Diagnostics.DataReceivedEventHandler ErrorDataReceived(Sys
Exited     Event        System.EventHandler Exited(System.Object, System.EventArgs)
OutputDataReceived Event        System.Diagnostics.DataReceivedEventHandler OutputDataReceived(Sy
BeginErrorReadLine Method        void BeginErrorReadLine()
BeginOutputReadLine Method        void BeginOutputReadLine()
CancelErrorRead Method        void CancelErrorRead()
CancelOutputRead Method        void CancelOutputRead()
Close      Method        void Close()
CloseMainWindow Method        bool CloseMainWindow()
CreateObjRef Method        System.Runtime.Remoting.ObjRef CreateObjRef(type requestedType)
Dispose    Method        void Dispose(), void IDisposable.Dispose()
Equals     Method        bool Equals(System.Object obj)
GetHashCode Method        int GetHashCode()
GetLifetimeService Method        System.Object GetLifetimeService()
```

היות ו-PS לא יכול להציג את כל המידע שהאובייקט מכיל על המסך, פקודת GM היא שימושית כיוון שהיא תראה לנו איזה עוד מידע נוסף לא נכלל בפלט ואולי דווקא אותו נרצה להציג.

היות וזהו לא קורס העוסק ב-PS נסתפק בהכרות ראשונית עם השפה ובהמשך נסקור פקודות שימושיות שיעזרו לנו לטפל ולאתר תקלות ביעילות.