



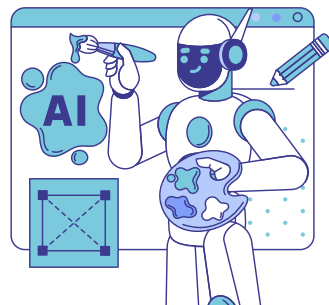
מיכאל שטאל

אחרי 35 שנות עבודה באינטל ישראל, מיכאל פרש מתפקידו כמהנדס בדיקות תוכנה בכיר ועוסק היום בייצור, ובפעילות התנדבותית ב-ITCB וב-ISTQB. מעבר לעבודת בדיקות ממש (שזה הכי כיף!), מיכאל הדריך לאורך השנים אלפי עובדים בתכנון בדיקות, שיטות בדיקה ומתודולוגיות עבודה, וממשיך ללמד בדיקות תוכנה בפקולטה למדעי המחשב באוניברסיטה העברית. ניתן לראות חלק מהמצגות והמאמרים שלו באתר <https://www.testprincipia.com>



תן ל-AI לעשות את העבודה

לאחרונה קראתי מספר בלוגים שמתארים איך מתראיינים מצליחים לעבור ראיון עבודה שמתנהל בווידיאו על ידי העברת השאלות של המראיין ל-AI וציטוט של התשובה. יש אפילו סרטוני YouTube שמייצעים איך לעשות את זה טוב. האמת היא שגם לפני AI, התקווה שסדרת ראיונות תבטיח לנו שנבחר את המועמדים המתאימים הייתה בדיוק כך: תקווה. הרושם הראשוני ומגבלות הזמן משאירים תמיד מידה מסוימת של ספק האם ההחלטה לגייס מישהו היא החלטה טובה. האפשרות שמתפתחת עכשיו, שמאפשרת למועמדים לעבור ראיון על ידי שימוש ב-AI, מוסיפה עוד חוסר ודאות לתהליך, ומעלה את הסיכון שנבחר מועמדים לא מתאימים. האם החשש הזה במקום? אני אחזור לזה, אבל תנו לי להתפזר קצת.



כתבתי אפליקציה!

לאחרונה כתבתי יישום רשת פשוט וקטן בפייתון. כשהיישום היה מוכן, הוספתי כמה בדיקות אוטומטיות שגם הן כתובות בפייתון, תוך שימוש בסלניום ו-Chromedriver.

זה לקח לי שלושה ימים. מסיבות שיתבררו מיד, אין לי מושג אם שלושה ימים זה סביר, מדהים או ממש עלוב. כדי להסביר, אתן קצת רקע על עצמי, אתאר מה למדתי תוך כדי החוויה של כתיבת היישום, ואתן לכם להחליט עד כמה ההישג הזה משמעותי.

הרקע שלי

מבחינת התואר, אני מהנדס אלקטרוניקה. ב-35 השנה האחרונות עבדתי באינטל, בתחילה במפעל הייצור בירושלים (FAB8), וב-25 השנה האחרונות במרכז הפיתוח בבניין שמעבר לכביש (המפעל, דרך אגב, נסגר ב-2007, וכעת כל האתר בירושלים עוסק בפיתוח). לפני המעבר לפיתוח למדתי קורסים באוניברסיטה הפתוחה, כך שאמנם אין לי תואר רשמי במדעי המחשב, אבל השלמתי את רוב הקורסים שהיו חסרים לי בתחום.

במרכז הפיתוח התחלתי לעבוד כבודק במעבדה, ועם השנים התקדמתי עד לדרגה של מהנדס בכיר, כארכיטקט בדיקות תוכנה. הרצתי בדיקות - או הובלתי את הצד הטכני של הבדיקות - בפרויקטים שונים ומגוונים, שהצד המשותף לרובם היה שהם היו במערכות משובצות מחשב. התוצאה היא שבכל הקריירה שלי ועד פרישתי לפני כמה חודשים, לא יצא לי לכתוב יישום רשת או בדיקות אוטומטיות ל-GUI.

אחד היתרונות של פרישה מהעבודה זה שאני עושה מה שבא לי. כך שכשהבתי שלי העלתה רעיון ליישום רשת פשוט, החלטתי לנסות לכתוב אותו בעצמי. בהתחשב בכך שהשכר שביקשתי היה נמוך משמעותית מההצעות שהיא קיבלה ב-Fiverr, קיבלתי את העבודה (לתקופת ניסיון 😊).

תיקון: Microsoft Copilot כתב אפליקציה!

התחלתי מסודר: כתבתי את הדרישות, כך שאדע מה עלי לפתח. הבת שלי, שמחזיקה בתואר במדעי המחשב, סקרה את הדרישות ואישרה שהן מדויקות.

כיוון שהידע שלי בפייתון ו-HTML לא מספיק על מנת לכתוב יישום כזה בעצמי, עשיתי שימוש מסיבי ב-Microsoft Copilot. כתבתי תיאור של מה אני צריך, והעתקתי-הדבקתי את הקוד לקבצי .py ו-HTML. מה שאומר שרוב הכתיבה שעשיתי הייתה כתיבת הנחיות (prompts), ומעט מאוד כתיבת קוד.

אחרי שהיישום עבד, נעזרתי שוב ב-Copilot לכתוב בדיקות אוטומטיות על מנת לוודא שהיישום יעבוד היטב עם כ-30 משתמשים במקביל. Copilot היה מי שהחליט להשתמש בסלניום ו-Chromedriver.

שרפתי איזה חצי יום בניסיונות להתקין את היישום אצל ספק האינטרנט שמאחסן את האתר שלי, עד שהבנתי שבמנוי המינימלי

שיש לי זה בכלל לא אפשרי. עברתי ל-AWS ושם זה לקח כחצי שעה. רוב הזמן הושקע במאבק קצר עם הרשאות התקשורת.

לסיכום:

- יישום רשת: יומיים
- בדיקות אוטומטיות: ½ יום
- התקנה (כולל התברבורות): ½ יום

אני מדלג על העבודה של העיצוב בעזרת CSS, שלקחה הרבה יותר זמן ממה שהייתי מנחש - אבל זה כבר נושא אחר. לצורך הדיון, אני עוצר ברגע שהיה לי משהו עובד, למרות שלא היה מוכן מבחינה ויזואלית.

עבורי, שלושה ימים היו זמן ממש קצר. התחלתי עם ידע אפסי על פיתוח יישום רשת, והנה יש לי משהו עובד. ההתפעלות נובעת גם מתוך ההכרה שאם הייתי מנסה לעשות את זה לפני רק כמה שנים, זה היה לוקח לי הרבה יותר משלושה ימים.

הייתי צריך למצוא את ההדרכות הבאות ברשת (tutorials):

- איך לבנות יישום רשת
- איך לייצר בסיס נתונים SQL בעזרת פייתון
- איך לכתוב בדיקות אוטומטיות עם סלניום
- איך להריץ מספר רב של הילכי משנה (threads) על מנת לדמות הרבה משתמשים בו זמנית
- איך להתקין אפליקציה על AWS

הייתי גם צריך ללמוד לא מעט על HTML, איך לייצר דפים עם כפתורים שמעבירים את המשתמש מדף אחד לשני; איך להעביר משתנים בין דפים; איך לקרוא לדפים מתוך קוד של פייתון; וכמובן גם משהו בסיסי על CSS. רק ללמוד את כל זה היה לוקח לי כחודש, ואז הייתי צריך לנסות ליישם את מה שלמדתי ולהיכשל פעמים רבות עד שהיה לי קוד עובד.

מסקנה: כשיש AI, לא צריך ללמוד הרבה. AI יכול לעשות את העבודה עבורנו בלי שנבין ממש מה אנחנו עושים.

האמנם?

"נהניתי מכך ש-Copilot עושה לי את העבודה הקשה"

מבט אל מתחת למכסה המנוע

תוך כדי שנהייתי מכך ש-Copilot עושה לי את העבודה הקשה, נחתה עלי ההבנה שללא הידע **בהמון דברים** שרכשתי בשלושים וחמש שנה בהי-טק, כל מה ש-Copilot ייצר עבורי היה חסר מובן ולא היה לי מושג מה לעשות עם זה.



כן – אפשר להתווכח ולטעון שהייתה לי עקומת למידה עלובה ואיטית; היה עלי ללמוד עוד ובמהירות רבה יותר. אני לא מתווכח. להגנתי אומר שלמדתי את מה שהייתי זקוק לו לצורך עבודתי. אם הגדרת התפקיד הייתה כתיבת יישומי רשת בפייטון, מן הסתם הייתי לומד את כל זה כבר בשנה הראשונה, וצובר ניסיון ובטחון בתוך שנה-שנתיים נוספות.

אז מה למדנו מזה?

יש לי שתי מסקנות מהפרייקט הזה:

א. AI הוא מכפיל כוח מדהים למתכנתים (אני כולל בזה גם כותבי אוטומציית בדיקות). זה מאפשר לעשות יותר, גם בטכנולוגיות ושפות תכנות שאין לנו שליטה מלאה בהן.

ב. על מנת ליהנות מכל הטוב הזה, צריך בסיס איתן בתכנות והכרת מגוון רחב של נושאים קשורים.

אני מעודד מההבנה הזאת. מצד אחד, אני יכול להתמודד עם אתגרים שהיו בלתי אפשריים עבורי ללא העזרה של AI. ומצד שני, יש משמעות לניסיון שרכשתי בלמידה מתמשכת לאורך השנים. אי אפשר להחליף אותי בסתם מישהו עם דופק.

ומה הקשר לראיונות עבודה?

המסקנה מההתנסות הקצרה שלי היא שלא כל אחד יכול לעבור בהצלחה ראיון תוך שימוש ב-AI. הצלחה כזאת מצביעה על מועמדים בעלי רקע טכנולוגי משמעותי ויכולת שימוש מעשית בידע זה.

"מועמדים שמצליחים להשתמש בכלי AI תוך כדי ראיון, הם כאלה שנרצה לגייס"

אם נשאל את השאלות הנכונות בראיון – ולא משהו שניתן לענות בקלות על ידי העברת השאלה כמו שהיא ל-AI (למשל: "כתבו פונקציה בפייטון שעושה X") – נוכל לזהות מועמדים עם רקע טכנולוגי עשיר. מועמדים שמצליחים להשתמש בכלי AI ולתת תשובות בזמן אמיתי, תוך כדי ראיון, הם כאלה שנרצה לגייס. אם הם יודעים לנסח הנחיות אפקטיביות ל-AI, ויכולים לדבר באופן אינטליגנטי על התוצאות שקיבלו, הם לא מזויפים. סביר שיהיו עובדים יעילים, גם בניצול הידע שכבר יש להם וגם בהפקת המיטב מ-AI כשיתקלו בטכנולוגיה או תחום שהם (עדיין) לא מכירים. למעשה, אני ממליץ לעודד מועמדים להשתמש ב-AI בראיון. לחלוק איתנו את המסך, ולתאר איך הם ניגשים לפתרון השאלה שלנו תוך שימוש בעוצמה של כלים חדשים אלה.



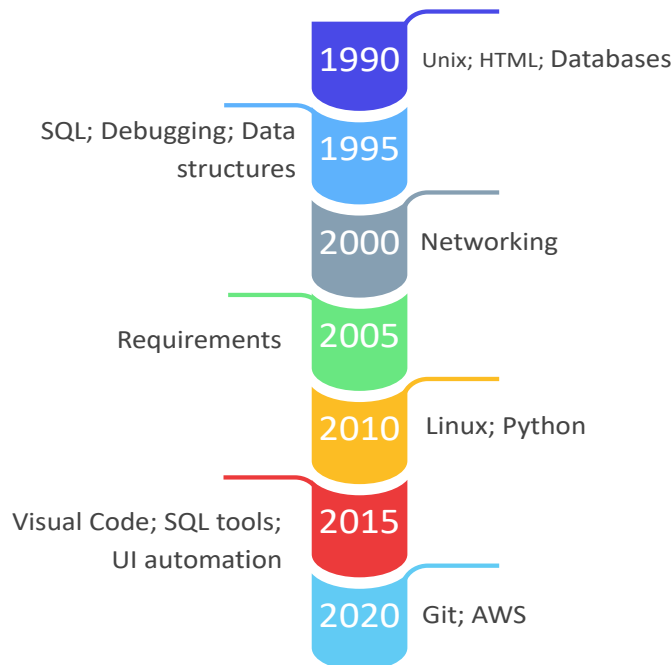
הנה רשימה של ידע שאיפשר לי להנות מהעבודה של Copilot. אין כאן שום סדר, ואני בטוח שיש עוד הרבה ידע שאני לוקח אותו ככל כך מובן מאליו שאני לא קולט שצריך להזכיר אותו.

- (1) HTML
- (2) בסיסי נתונים; ספציפית: SQLite ו-SQL
- (3) Linux
- (4) Git
- (5) Python
- (6) דיבוג קוד
- (7) Visual Code וההרחבות שלו (extensions)
- (8) תקשורת מחשבים (networking)
- (9) מבני נתונים

חלק מהנושאים ברשימה פשוטים למדי; אחרים לוקחים לא מעט זמן על מנת להגיע ליכולת מעשית, אפילו ברמה בסיסית, ורכישת מיומנות ובטחון בשימוש בהם לוקחת עוד יותר זמן. לא פחות חשובה מיכולת השימוש, היא הידיעה שהם בכלל קיימים, לאיזה צרכים הם טובים ומתי נכון להשתמש בהם.

תיאורטית, אם הייתי שואל את Copilot על כל אחד מהנושאים האלה, אני בטוח שהייתי מקבל עזרה והצעות איך להשתמש בהם. אבל זו בדיקת הנקודה: הייתי צריך להיות מודע לדברים כמו HTML, SQL, סביבות וירטואליות של פייטון, וכו' וכו' על מנת לבקש עזרה מ-Copilot. אם לא היה לי מושג עליהם, איך הייתי יודע לשאול עליהם? בכל מקרה, ודאי שלא הייתי מתקדם מהר.

קו הזמן למטה מראה (פחות או יותר) מתי התחלתי לרכוש את הידע שאפשר לי לנהל שיחה יעילה עם Copilot.



Graphics origin by <https://slidesgo.com> & <https://www.freepik.com>

כל נושא ברשימה הזאת נרכש בשעות רבות של השקעה בלימוד, ניסוי, טעיה ושימוש מעשי. חלק מהתחומים (כמו למשל הגדרה וניהול של דרישות) היו הגיוני העיקרי שלי במשך כמה שנים. התוצאה היא שיש לי ידע מעמיק וזמין, תיאורטי ומעשי, ביכולות שרכשתי.

"ללא הידע שרכשתי בשלושים וחמש שנה בהיטק, כל מה ש-Copilot ייצר עבורי היה חסר מובן"