



שיפור תהליכי פיתוח תוכנה באמצעות AI

תיאור הקורס:

בעולם של 2026, כתיבת קוד ידנית היא כבר לא צוואר הבקבוק - היכולת לנהל את ה-AI שכותב עבורכם היא המפתח. נשנה את התפיסה מ"מתכנתים" ל"אדריכלי מערכות" ונלמד איך להשתמש ב"עוזרי פיתוח" חכמים כדי לבנות אפליקציות מורכבות בתוך ימים במקום חודשים.

החשיבות נובעת מהצורך של השוק באנשים שיודעים להוביל פרויקט מקצה לקצה, תוך שמירה על איכות קוד גבוהה ואבטחה, כשהמכונה עושה את העבודה ה"שחורה". למידת התחום תעניק לכם יתרון אדיר: היכולת לייצר פתרונות טכנולוגיים שלמים לבד או בצוות קטן, מה שהופך אתכם לכוח עבודה מבוקש וחדשני בצורה בלתי רגילה.

קהל יעד:

מפתחי JavaScript, בוגרי קורסי פיתוח, Full-Stack, Frontend, מתחילים-ביניים, וכל מי שרוצה להשתלב בצוותים מודרניים.

סילבוס נושאים:

פרק 1: מבוא לפיתוח מואץ וסקירת כלי AI

- מודל "טייס ונווט": הבנת חלוקת העבודה החדשה בין המתכנת האנושי לבינה המלאכותית.
- אקו-סיסטם של כלי פיתוח: סקירת כלי ה-AI המובילים (GitHub Copilot, Cursor, Ghostwriter) והתאמתם לסביבות עבודה שונות.
- הנחיה הנדסית (Prompt Engineering) למפתחים: טכניקות מתקדמות לכתיבת הנחיות המייצרות קוד מדויק ובר-תחזוקה.

פרק 2: לוגיקה, ארכיטקטורה ותכנון מערכת

- תכנון לפני ביצוע: שימוש ב-AI לבניית תרשימי זרימה, מבני נתונים (ERD) ואפיון דרישות פונקציונליות.
- בחירת מחסנית טכנולוגית (Tech Stack): התייעצות עם מודלי שפה לבחירת הכלים הנכונים לכל פרויקט.
- פירוק משימות (Decomposition): הפיכת רעיון מופשט לרשימת משימות פיתוח מוגדרות עבור ה-AI.

פרק 3: פיתוח צד-לקוח (Frontend) במהירות שיא

- מיצירת עיצוב לקוד: הפיכת סקיצות וממשקים ויזואליים לרכיבי React/Vue דינמיים תוך דקות.
- Styling ועיצוב רספונסיבי: שימוש בבינה מלאכותית להחלת סגנונות (Tailwind/CSS) והתאמה לכלל סוגי המסכים.
- חוויית משתמש (UX): אופטימיזציה של ממשקים ושיפור הנגישות בעזרת המלצות AI בזמן אמת.

פרק 4: בניית ה"מנוע" (Backend) ולוגיקה עסקית

- הקמת שרתים ו-APIs: בניית נקודות קצה (Endpoints) וניהול תקשורת נתונים במינימום כתיבה ידנית.
- ניהול בסיסי נתונים: יצירת שאילתות מורכבות ואופטימיזציה של מבנה המידע בעזרת עוזרי פיתוח.
- מימוש לוגיקה עסקית: הזרקת חוקים עסקיים מורכבים לתוך הקוד בצורה עקבית ומאורגנת.

פרק 5: בדיקות (Testing) ובקרת איכות

- אוטומציה של בדיקות: כתיבת Unit Tests ו-Integration Tests באופן אוטומטי על בסיס הקוד הקיים.
- איתור באגים חכם: שימוש ב-AI לניתוח שגיאות לוגיות והצעת תיקונים (Refactoring) מיידיים.
- שיפור ביצועים: זיהוי צווארי בקבוק בקוד ויעול צריכת המשאבים.

פרק 6: ניהול גרסאות ושיתוף פעולה (Git Smart)

- ניהול Git מונחה AI : כתיבת הודעות Commit חכמות וסיכומי שינויים (PR Descriptions) באופן אוטומטי.
- סקירת קוד (Code Review) : שימוש ב-AI כ"עין נוספת" לביקורת עמיתים לפני מיזוג הקוד.
- תיעוד פרויקט : יצירת דוקומנטציה מלאה (README, API Docs) תוך כדי תנועה.