* 1. **הנחיית פרויקטים – מיון טיפים לפי עקרונות פדגוגיים כלליים**

**א-1. למידה מתוך תהליך ולא רק תוצאה**

* החלום הוא לתת לתלמיד את ניהול לוח הזמנים, מניסיון שלי נכון להגדיר מראש בתחילת השנה חלון הגשות אחת לחודש, בכל חלון לדרוש הגשה של שלב מוגדר בפרויקט.
* זה לא מחשבים שעושים שגיאות, זה בני אדם שכותבים שגיאות. למד והדגם בדיקות והרצות קוד דחופות. אין לכתוב הרבה קוד במכה ולצפות שזה יעבוד. כותבים כל פעם שתיים/שלוש שורות ומריצים. כך יודעים היכן השגיאה.
* חשוב מאד ליצור אבני דרך ברורות. תלמידים חלשים, פעמים רבות, מרגישים אבודים, כי הם לא יודעים מאיפה להתחיל, איך לחפש מידע ואיך להתקדם באופן עצמאי. עבורם יש ליצור משימות קטנות עם הסברים ברורים ולוודא שהם מבינים כל שלב לפני שממשיכים הלאה. אפשר ליצור יחד איתם את סדר המשימות ולוודא שהם עומדים במשימה לפני מעבר למשימה הבאה. במקרה הצורך אפשר להצמיד לתלמיד חלש, תלמיד חזק יותר שילווה אותו.
* יצירת אווירה חיובית בכיתה: תן מקום לטעויות – חשוב שהתלמידים יבינו שטעויות הן חלק מהתהליך. אם תלמיד מפחד לטעות, הוא לא ינסה דברים חדשים. אתה יכול להדגים באג מכוון, ולשאול את הכיתה איך לתקן אותו.
* למדו את התלמידים איך לדבג תוכנה, כולל עבודה עם כלי debug , מה שיאפשר לתלמיד יותר הבנה בחומר הנלמד.
* ניהול ציפיות אישיות: תבין שהפרויקט הוא של התלמיד, לא שלך – לפעמים תרגיש צורך "להציל" תלמידים על ידי פתרון בעיות עבורם, אבל זה ימנע מהם לפתח מיומנויות קריטיות. תן להם להתמודד, גם אם זה אומר שהם יטעו וילמדו בדרך הקשה.

**א-2. למידה מתוך עניין, תחביב, וחיבור אישי**

* אתגרים לתלמידים: אני מוצאת שמה שמאתגר זה בעיקר לשלב פרויקטים עם מגמות אחרות ועם התעשייה .
* בחירת נושא הפרויקט - אם זה אפשרי לבחור נושא שמאפשר מימוש ברמות שונות: בסיסית, מתקדמת, אידאלית. כדאי לכוון את התלמיד לסיים את הגירסה הבסיסית של הפרויקט. אחר כך, אם נשאר זמן, לעבור לרמה מתקדמת יותר. למשל, משחק XO נגד בוט: בפרויקט הבסיסי בוט אקראי (טיפש), רמה מתקדמת: בוט חכם (תלמיד כותב אלגוריתם), רמה אידאלית – שימוש באתר AI לניהול המשחק.
* כתיבת קוד נכונה מצמצמת שגיאות. הדגם כתיבה רעה מול טובה. הסבר כיצד הכתיבה הזו מצמצמת שגיאות וכמובן בדוק שהתלמידים אכן מבצעים זאת. כאשר הם טועים, הצבע על המקום הבעייתי ובקש שהם יתקנו אותו.
* לגבי אתגרים ב-30%, שאלתי בתחילת השנה מה התחביבים של התלמידים ולדוגמא כשראיתי שחלק מהתלמידים לא מבצעים את הפרויקט עם מוטיבציה גבוהה, בניתי מערך שיעור חדש ששילב כדורגל (רוב הבנים בכיתה אוהבים את התחום), במקרה אחר התלמידים בנו עמוד למשחק שח-מט (תחביב של כמה בנות בכיתה). בניגוד לשנה שעברה שלקחתי נושא של חנות אינטרנטית והלבשתי את מירב תחומי הלימוד על נושא החנות, השנה אני מגוון מאוד.
* ניהול ציפיות אישיות: אתה לא חייב לדעת הכל – תלמידים יגיעו עם שאלות על טכנולוגיות חדשות, ספריות מתקדמות ובעיות שאתה לא נתקלת בהן. זה טבעי. במקום להילחץ, השתמש בזה כהזדמנות להדגים חקר ולמידה עצמאית. תגיד לתלמידים *"אני לא בטוח, בואו נבדוק יחד איך פותרים את זה."*
* שילוב טכנולוגיות מתקדמות בפיתוח אפליקציות כדי להפוך את הלמידה ליותר מעניינת ורלוונטית. לדוגמה, אפשר לשלב פיתוח אפליקציות שמשתמשות בטכנולוגיות מתקדמות, כמו שעונים חכמים, משחקי רשת, או שליפת נתונים מהאינטרנט וביצוע מניפולציות עליהן. זה לא מתאים לכל התלמידים אבל כולם מתלהבים במיוחד כשתלמיד אחד מפתח דברים מיוחדים ויכולים אפילו להיעזר בזה לפרויקטים אחרים.

**א-3. התלמיד במרכז – אמון, מסוגלות והתקדמות**

* באופן כללי מציעה ליצור קשר אישי עם התלמיד, הקשבה לצרכים ולקשיים שלו, מיפוי סוג הקושי: רגשי, ידע חסר, חוסר במוטיבציה וכדומה, ומתן עזרה לפי הקושי כדלהלן: ידע חסר - עזרה בפרטני, רגשי - שיתוף צוות הייעוץ של בית הספר, חוסר במוטיבציה - הצגת פרויקטים מוכנים, מה ייתן לו הידע הזה בעתיד וכדומה.
* ההרגשה שלי בפרויקטים היא שאני והתלמידים שותפים בבניית הפרויקט. אני לא באמת יודעת יותר מהם, הרצון שלי הוא לשדר להם שאנחנו לומדים יחד (עם זה שיש לי ידע מה אני רוצה מהם ואולי ידע מקצועי רב מהם)
* לתווך את המשימה לפי הרמה – בגובה העיניים - אם יש תלמיד שנתקע, מושך כל הזמן ולא מראה ביצועים במשימות שהוגדרו לו (על פי שלבי חוברת הפרויקט), כדאי לקבוע איתו פגישה אישית ולתת לו הנחיה אישית, אם נדרש, אפילו כותבים ביחד משימה ראשונה ובהמשך מנחים אותו בתתי משימות יותר קטנות על פי לוח זמנים.
* עבור תלמיד מתקשה אני מפרקת את המשימה לתתי משימות נוספות, פשוטות ממש, בכדי לאפשר לו לחוות הצלחה (גם אם קטנה) כך שתחושת המסוגלות שלו תגדל ויקבל ביטחון גם במשימות המורכבות יותר. כמו כן מצוותת אותם לתלמידים חזקים יותר בכדי שאלה ימשכו אותם כלפי מעלה, וברוב המקרים מצליחים להסביר את החומר בפשטות מנקודת מבטם, מה שמתגלה כאפקטיבי מאוד.
* פרויקט מותאם לכל תלמיד (לכל תלמיד יכולת וקצב עבודה משלו). במהלך הנחיה אישית או קבוצתית, חשוב להבין את רמת התלמיד ולהתאים את המשימות בהתאם.
* תתחיל ללמד עם הצלחות לתלמידים. בחר חומרים קלים, ביצוע פרויקטים פשוטים שיאפשרו לתלמיד להצליח ולקבל ציונים גבוהים בהתחלה.

**א-4. למידה מבוססת חקר, פרויקטים והתנסות**

* **איך מוצאים אתגרים מעניינים לתלמידים מתקדמים (נעזרתי ב-**AI**): חבר את האתגרים לעולם האמיתי ולתחומי העניין של התלמידים. לדוגמה, אם תלמיד מתעניין במשחקי מחשב, הצע לו ליצור משחק פשוט בעצמו. בנוסף, אפשר לאתגר אותם בפרויקטים הכוללים פתרון בעיות אמיתיות, כמו בניית אפליקציה שתסייע בארגון משימות כיתתיות או פיתוח תוכנה שמדמה תהליכים הנדסיים. השתמש במאגרי אתגרים מקוונים, כמו** Project Euler **או** Advent of Code**, כדי לחשוף אותם לבעיות מורכבות ולוגיות.**
* יצירת עניין – חיבור לנושאים מעולם ההייטק מאד מומלץ. תלמידים מתחברים יותר לנושאים כשהם רואים איך זה מתקשר לעולם האמיתי. כדאי להביא דוגמאות מפרויקטים אמיתיים, להזמין אורחים מהתעשייה (מתכנתים, מהנדסים – בדרך כלל יש הורה בכיתה שישמח להגיע ולספר על תחום עבודתו בהייטק), או לשלב נושאים עדכניים כמו בינה מלאכותית וסייבר.
* למדו את התלמידים איך ללמוד לבד – לחקור, ללמוד ולהתעניין בטכנולוגית חדשות
* מציאת אתגרים מעניינים לתלמידים מתקדמים. אני ממליץ לעודד שילוב בפרויקט של נושאים חדשים שלא נלמדו בכיתה. הדרך הכי טובה לאתגר תלמידים היא לתת להם ללמוד בעצמם עם תמיכה וליווי במקום להכתיב להם פתרונות מוכנים. ממליץ מאד להיערך לשיתופי פעולה עם התעשייה בפרויקטים – מתאים מאד לתלמידים מתקדמים שמתלהבים מטכנולוגיה ומעוניינים להשקיע בפרויקט.
* נוח ומעניין להתרכז בתלמידים הטובים. אלה שחוקרים, מנסים דברים יצירתיים ומחדשים. נכון לתת תשומת לב לתלמידים המתקשים יותר.

**א-5. גמישות וחשיבה פתוחה**

* איזון בין גמישות למסגרת ברורה: תהיה קשוח עם דדליינים, אבל גמיש עם הדרך להגיע אליהם – לוח זמנים ברור הוא קריטי, אבל אפשר לאפשר לתלמידים לבחור איך להגיע לכל שלב. לדוגמה, לאפשר להם לבחור אם להגיש תיעוד בכתב, בסרטון או במצגת.
* התאמת אתגרים מכיוון שבכל כיתה יש טווח רחב של יכולות, יש צורך להתאים את האתגרים על פי יכולת התלמיד. לתלמידים שמתקשים, צריך לתת מטלות קטנות יחסית וברות יישום לרמתו. אפשר גם להצמיד תלמיד חזק שיסייע לתלמיד חלש וכך הם גם מרוויחים מחיזוק הידע. לתלמידים מתקדמים, אפשר לתת אתגרים נוספים כמו השתתפות בתחרויות קידוד, או חקר של טכנולוגיות חדשות או בכלל נושאים שיכולים לתרום לכולם.
* לא כל התלמידים חייבים להיות מתכנתים בעתיד – יש תלמידים שיבחרו ללמוד הנדסת תוכנה כי זה מסקרן אותם, אבל בסוף ילכו למקצועות אחרים. גם אם הם לא ימשיכו בפיתוח תוכנה, הם ירוויחו מיומנויות חשיבה ביקורתית, פתרון בעיות וניהול פרויקטים.
* ללמוד לתקן שגיאות הוא נושא חשוב לא פחות מכתיבת קוד. כאשר אתה מדגים קוד על המחשב מול התלמידים צור בכוונה שגיאות שונות בקוד ותן לתלמידים למצוא אותם. הקפד על גם על שגיאות קומפילציה וגם שגיאות אלגוריתמיות. למד את התלמידים למצוא שגיאות בעזרת הדפסות ובעזרת נקודות עצירה. למד אותם כיצד קוראים את השגיאות שסביבת הפיתוח מציגה והמידע הרב שניתן למצוא בסביבת הפיתוח (כמו ה call stack של איזה פעולה קראה לאיזה פעולה טרם השגיאה) וכדומה.
* שמור על שקיפות מלאה - פרסם לוח זמנים לכל השנה כבר מההתחלה, תסביר איך הפרויקט יתקדם, ותעדכן את התלמידים בכל שינוי. ככל שהם ירגישו שליטה, הם ינהלו את הזמן שלהם טוב יותר.