* 1. **הנחיית פרויקטים – מיון טיפים לפי אתגרים לתלמידים מתקדמים**

**ז-1. התאמה אישית של אתגרים**

* אין תחליף לפגישות אחד-על-אחד צפופות עם תלמידים. זה בעיקר מה שאני עושה בתקופה הזו. אני מבקש מהתלמידים דו"חות דו-שבועיים על התקדמות ותוכניות לשבועיים הבאים, אבל אני לא ממש מספיק לקרוא אותם בעיון. עם תלמיד שנתקע אני פשוט מבלה יותר זמן באחד-על-אחד ומפליג איתו בסיעור מוחות. לפעמים אנחנו נפרדים עם משימות בדיקה וחוזרים בשיעור שאחרי עם יותר תובנות. כך גם עם תלמידים מתקדמים יותר, בפגישות האלה אנחנו "תופרים" אתגרים מתאימים.
* 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.
* הנחייה אישית/קבוצתית: עליך לזכור שכל תלמיד הוא אינדיבידואל, בעל יכולת קוגניטיבית ומערכתית משלו, וגם אם הוא במגמת הנדסת תוכנה אין הדבר מעיד על הומוגניות מוחלטת בינו לבין חבריו למגמה. לפיכך המלצתי היא לבנות מסלול התקדמות אישית לכל תלמיד, כמובן תוך התחשבות בלוחות הזמנים המחייבים (מועדי בחינות של משרד החינוך, למשל). כלי שמאוד עוזר לי באופן אישי הוא גאנט מסודר, ובו טבלה בה מפורטות כל המשימות בחלוקה לפי נושאים, כאשר לכל תלמיד מוקדשת שורה, וכל תלמיד מסמן בתיבת הבחירה (צ'קבוקס) את המשימות שכבר ביצע. אני מוצאת שזהו כלי יעיל כיוון שהוא מאפשר לתלמידים לעקוב בעצמם אחר ההתקדמות שלהם, וגם להשוות את עצמם לקצב הכללי בכיתה בכדי להבין האם עליהם להדביק איזשהו פער. תלמידים מתקדמים מאותגרים יותר, למשל בהכוונה למציאת פתרונות בעצמם ברחבי הרשת, עידוד לכתיבת קוד גנרי יותר ושימוש בכלים שלא לימדתי בכיתה, לבחירתם.
* 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.
* חשוב מאוד להתאים את הפרויקט לתלמיד: לתלמיד חלש – פרויקט למידותיו, לתלמיד חזק – פרויקט מאתגר. למרות שהם יגיעו עם בקשת אחרות, לזכור תמיד keep it simple כדי להגיע לקו הגמר עם פרויקט.
* נוח ומעניין להתרכז בתלמידים הטובים. אלה שחוקרים, מנסים דברים יצירתיים ומחדשים. נכון לתת תשומת לב לתלמידים המתקשים יותר.
* 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.
* תלמידים מתקדמים שרצים קדימה מהר, אני נותן להם אתגרים אישיים להוספת דברים בפרויקט.

**ז-2. למידה עצמאית וחקר**

* **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.**
* יש הרבה נושאים שהם די קלים ללמידה עצמית ולמימוש, וניתן לתת לתלמיד ללמוד את הנושא ולממש לבד (כמובן שניתן להיעזר ברשת וב- AI כדי להבין איך זה מתבצע.) לאחר מכן לומר לתלמיד לעזור לתלמיד נוסף לממש את אותו נושא. זה מקצר מאד זמנים כי בפעם השניה והשלישית זה הרבה יותר מהר, והם לומדים לעזור ולהיעזר. כך ניתן להרחיב את הנושאים שמממשים בפרויקטים, מעבר למה שהמורה מלמד/ת פרונטלית בכתה.
* כחלק מציון ההגשה, על כל תלמיד לחקור ולהתמקצע בנושא כלשהו שלא לימדתי ולהיות "מומחה" לנושא – עליו להכין מצגת או דף הסבר ולהיות מסוגל לתמוך בחבריו שירצו לשלב נושא זה בפרויקט.
* 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.
* למדו את התלמידים איך ללמוד לבד – כמו שכל מתכנת בהייטק עושה!
* 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.
* 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."
* 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.

**ז-3. הרחבה טכנולוגית ומקצועית**

* אתגרים לתלמידים: אני מוצאת שמה שמאתגר זה בעיקר לשלב פרויקטים עם מגמות אחרות ועם התעשייה .
* בכל נושא שימוש ב- GenAI , לפני הטמעת הקוד, אני מבקש מהם לעבור על שורות קוד חשובות, בהן יש שימוש במחלקות או מבנים לא מוכרים, לקרוא עליהם ולראות שהם מבינים את הפונקציונליות שלהם, זאת כדי שגם אם ההטמעה תצליח, חשוב שלא ידקלמו אח"כ את הקוד בהסבר מול הבוחן.
* הנחיות תלמידים בזמן ביצוע פרויקט גמר - בזמן העבודה על פרויקט הגמר תמיד ישנם נושאים שתלמידים לא יודעים ואתם נדרשים ללמד. כמובן אם אתם מכירים את הנושא והוא קשה ללימוד עצמי. כדאי לזהות עוד תלמידים שצריכים ללמוד את הנושא הזה, כך שאפשר ללמד את כל הקבוצה. למשל, גרפיקה. נושא לא קטן ולא פשוט ללימוד עצמי. במידע ולימדתם תלמיד/מספר תלמידים נושא כלשהו, כאשר לתלמידים אחרים יש שאלות בנושא זה – שלחו אותם לתלמידים "המומחים".
* 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.
* 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.
* למדו את התלמידים איך לדבג תוכנה, כולל עבודה עם כלי debug , מה שיאפשר לתלמיד יותר הבנה בחומר הנלמד.
* תלמיד חייב להגיע לעבודה על פרויקט גמר בהנדסת תוכנה לאחר שצבר ניסיון בלמידה מבוססת פרויקטים בכיתה י' וגם בכיתה יא' (אחרת אין סיכוי לתהליך למידה ראוי בפרויקט בכיתה יב'!)

**ז-4. תלמידים מובילים אחרים**

* יש הרבה נושאים שהם די קלים ללמידה עצמית ולמימוש, וניתן לתת לתלמיד ללמוד את הנושא ולממש לבד (כמובן שניתן להיעזר ברשת וב- AI כדי להבין איך זה מתבצע.) לאחר מכן לומר לתלמיד לעזור לתלמיד נוסף לממש את אותו נושא. זה מקצר מאד זמנים כי בפעם השניה והשלישית זה הרבה יותר מהר, והם לומדים לעזור ולהיעזר. כך ניתן להרחיב את הנושאים שמממשים בפרויקטים, מעבר למה שהמורה מלמד/ת פרונטלית בכתה.
* כאשר יש תלמידים מתקדמים, אפשר לשחרר אותם או לתת להם בונוס בתנאי שימצו ויעזרו לתלמידים שקשה להם
* לשים לב שכשעוזרים לתלמידים שיש להם באג/תקלה/וכדומה, אם התלמידים מצליחים, לבקש מהם לעצור אחרי כמה זמן, לתת להם הנחיות להמשך בדיקה, ולבקש מהם לתמוך גם בתלמידים אחרים.
* 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.