הבנתי. אתחיל בתחקיר שימפה כיצד לבנות מערכת תרגול ופענוח אק"ג מבוססת על מאגר PTB-XL, שתהיה משופרת ביחס לבחנים אונליין קיימים בכך שתספק מידע קליני מפורט לכל תרשים – כולל הסברים טקסטואליים, סיווגים רפואיים, הדמיות ECG, ומטא-דאטה מתוך המאגר. אחפש השראה ממערכות קיימות, אתגרים נפוצים, והצעות ליישום אופטימלי כולל רכיבים מומלצים בקוד. אחזור עם המלצות מפורטות.

**המלצות לעיצוב ובנייה של מערכת תרגול ולמידת פענוח אק"ג מבוססת PTB-XL**

מערכת תרגול ולמידה לפענוח אק"ג (ECG) המבוססת על מאגר PTB-XL צריכה לספק חוויית לימוד אינטראקטיבית ועשירה, המשפרת את הקיים בבחנים רפואיים מקוונים. להלן המלצות מפורטות לעיצוב המערכת, כולל מבנה רכיבים, אופן יצירת השאלות וההסברים, טכנולוגיות מתאימות ורעיונות לשיפור חוויית הלמידה.

**מבנה המערכת המוצע**

המערכת תיבנה בצורה מודולרית, כך שכל רכיב אחראי על חלק מוגדר בתהליך:

* **טעינת הנתונים (Data Loader):** מודול זה יטען את נתוני הגלם של אק"ג ממאגר PTB-XL. הוא ישתמש בקבצים המקוריים (ptbxl\_database.csv, scp\_statements.csv וכן קבצי האותות .hea/.dat). ניתן להסתמך על ספריית Python כמו wfdb לקריאת קבצי WFDB של Physionet. מודול זה יפיק מבני נתונים (לדוגמה, DataFrame של Pandas) המכילים את המידע הדרוש לכל רשומת אק"ג: הנתיב לקובצי האות (ליצירת התרשים), נתוני מטופל (גיל, מין, תאריך הקלטה וכו'), קודי האבחנה (scp\_codes) ועוד.
* **יצירת תרשים אק"ג (ECG Image Generator):** רכיב שאחראי להמרת אות ה-ECG הגולמי לתמונה גרפית (למשל בפורמט PNG). ניתן להשתמש בספריית matplotlib כדי לשרטט את 12-החיבורים של האק"ג על גרף ולשמור את הפלט כקובץ תמונה. עבור כל רשומה, המודול יקבל את נתוני האות (באמצעות wfdb או המירוץ ל-numpy) וייצר תרשים הניתן להצגה למשתמש בשאלה. רצוי לשמור את התמונות בתיקייה ייעודית (לדוגמה data/ecg\_images/) עם שם אחיד (כגון ecg\_<ecg\_id>.png) כדי שיהיה קל לקשר בין תמונה לשאלה.
* **מחולל השאלות (Quiz Generator):** מודול הליבה ליצירת השאלות. הוא יקבל עבור כל רשומת אק"ג את הנתונים הנחוצים (תרשים + מידע מטא) וייצר שאלה בסגנון רב-ברירה. לכל שאלה צריכה להיות אבחנה נכונה אחת לגבי התרשים הנתון, ושלוש תשובות שגויות (מסיחים). המחולל ישתמש במידע מתוך ptbxl\_database.csv כדי לקבוע את התשובה הנכונה – למשל, לבחור את האבחנה המרכזית או המשמעותית ביותר מת dintre קודי ה-SCP של אותה רשומה. יתר על כן, מודול זה יהיה אחראי גם לבחור מסיחים הגיוניים (פירוט בתת-סעיף להלן). פלט המודול יכול להיות מבנה נתונים של שאלה, למשל אובייקט או מילון עם השדות: טקסט השאלה, רשימת 4 האפשרויות, אינדקס התשובה הנכונה, ונתוני עזר (כמו מזהה התרשים לקישור לתמונה).
* **מערך ההסברים (Explanation Generator):** רכיב שאחראי על יצירת טקסט תיאורי ברור לכל תרשים אק"ג, במיוחד כמשוב לאחר מענה על שאלה. רכיב זה ישתמש בקודי ה-SCP ובמטא-דאטה של הרשומה כדי להרכיב הסבר קליני מקיף: הוא יתרגם כל קוד לאבחנה טקסטואלית (בעזרת scp\_statements.csv המכיל תיאורים של 71 הצהרות אק"ג שונות לפי תקן SCP-ECG ([PTB-XL, a large publicly available electrocardiography dataset v1.0.3](https://physionet.org/content/ptb-xl/#:~:text=The%20PTB,The%20dataset%20is))), יספק הקשר לסיווג הרפואי של כל קוד (האם הוא קוד אבחנתי, קוד קצב או קוד צורה – **diagnostic/form/rhythm** ([PTB-XL, a large publicly available electrocardiography dataset v1.0.3](https://physionet.org/content/ptb-xl/#:~:text=two%20cardiologists%2C%20who%20assigned%20potentially,The%20dataset%20is)) ([PTB-XL, a large publicly available electrocardiography dataset v1.0.3](https://physionet.org/content/ptb-xl/#:~:text=,diagnostic_subclass)), וכן לאיזו קטגוריה היררכית הוא שייך כמו diagnostic\_class), ויציג נתוני מטופל ופרמטרים רלוונטיים (גיל ומין המטופל, ציר הלב אם קיים, קצב לב משוער, מדדי זמן כמו QRS ו-QT אם ניתנים לחילוץ, ועוד). ההסבר ישמש הן לחיזוק הלמידה אחרי שהתלמיד ענה והן כ"תשובת פתרון" המצורפת לשאלה.
* **ממשק משתמש והצגת השאלות:** למרות שהדגש הוא על רכיבי backend, יש לתכנן גם כיצד יוצגו השאלות למשתמש. זה יכול להיות ממשק וובי (למשל יישום מבוסס Flask/Streamlit) או אפליקציה שולחנית. הממשק יציג בכל פעם תרשים אק"ג (כתמונה סטטית או כגרף אינטראקטיבי) עם טקסט השאלה ומתחתיו 4 אפשרויות. לאחר בחירת התשובה, המערכת תחשוף את ההסבר המלא לאותה שאלה. מומלץ לכלול כפתורים למעבר לשאלה הבאה, וסטטיסטיקות בסיסיות (ניקוד, התקדמות וכו'). **הערה:** הממשק עצמו יכול להיבנות בשלב שני, אך חשוב שהארכיטקטורה תאפשר שילובו בקלות (למשל, ע"י הפרדת מחולל השאלות מלוגיקת התצוגה).

*הערת ארכיטקטורה:* ניתן לארוז את הרכיבים הנ"ל בספריות/מחלקות נפרדות. למשל, קובץ ראשי ecg\_quiz\_generator.py שיקרא לפונקציות מתוך מודולי עזר (data\_loader.py, image\_generator.py, quiz\_generator.py, explanation\_generator.py). בצורה זו, כל חלק של המערכת ניתן לבדיקה ואחזקה בנפרד. כמו כן, רצוי להוסיף **מודול לוגים (Logger)** שיתעד תקלות או בעיות (כגון כשל בטעינת קובץ נתונים או יצירת תמונה) לקובץ לוג ייעודי (אפיון ECG.docx), כדי לסייע בדיבאגינג ובאמינות המערכת.

**תהליך יצירת שאלה ובחירת מסיחים**

יצירת שאלה אמריקאית מכל רשומת אק"ג תתבצע במספר שלבים, במטרה להבטיח שאלה מאתגרת והגיונית:

1. **בחירת האבחנה הנכונה:** לכל תרשים אק"ג, יש לעיתים מספר קודי אבחנה (לדוגמה, גם הפרעת קצב וגם ממצא של אוטם או הפרעה בהולכה). המערכת צריכה לקבוע מה תהיה ה"אבחנה המרכזית" שתשמש כתשובה הנכונה. ניתן לקבוע כללים כגון: לבחור בקוד האבחנתי החמור ביותר או הבולט ביותר קלינית. למשל, אם בתרשים קיימים קודים של אוטם בשריר הלב (MI) לצד קוד של קצב סינוסי איטי, סביר שהאוטם יהיה האבחנה העיקרית לשאלה. לחלופין, ניתן להגריל אחד מהקודים של הרשומה כשאלה – ולאורך מבחן להציג את כלל ההיבטים. כך או כך, השאלה תנוסח בצורה כללית כ"הממצא העיקרי באק"ג" או "מהי האבחנה?" לגבי אותו תרשים.
2. **ניסוח השאלה (Stem):** הקפד שנוסח השאלה יהיה ברור ותמציתי. למשל: *"מהו האבחון המתאים ביותר לאק"ג שלמטה?"* או *"איזו הפרעה משתקפת בתרשים האק"ג המצורף?"*. יש לוודא שהשאלה ממוקדת בממצא מסוים ולא מבלבלת את הקורא. רצוי לכלול בשאלה את ההקשר הנחוץ, למשל "גבר בן 65, התלונן על כאבים בחזה..." אם משלבים נתוני מטופל בהצגת השאלה. עם זאת, אם המערכת משתמשת תמיד בנתוני המטופל, אפשר לכלול אותם כחלק מהתרשים/נתונים מוצגים ולאו דווקא בגוף השאלה.
3. **בחירת שלוש מסיחים (תשובות שגויות):** בחירת מסיחים היא שלב קריטי בעיצוב שאלה טובה. על המסיחים להיות **סבירים אך לא נכונים**, כדי לאתגר את הלומד מחד ולא להטעות באופן בלתי הוגן מאידך. דרך מומלצת היא להשתמש בסיווגי הקוד כדי לבחור מסיחים מאותה קטגוריה כמו התשובה הנכונה. למשל, אם התשובה הנכונה היא קצב פרפור פרוזדורים (AFIB), המסיחים יכולים להיות קצבים חריגים אחרים (כגון רפרוף פרוזדורים AFLT, טכיקרדיה על-חדרית SVT, או קצב סינוסי מהיר) במקום אפשרויות שלא קשורות לקצב כלל. באופן דומה, אם התשובה היא אוטם בשריר הלב תחתון, מסיחים אפשריים יהיו אבחנות אק"ג שונות (כמו אוטם קדמי, חסם הולכה, אק"ג תקין) – מספיק קרובות כדי להיות מבלבלות ללומד שאינו בטוח. חשוב להימנע ממסיחים שהם מופרכים לחלוטין או חופפים מדי לתשובה הנכונה ([Designing Quality Multiple Choice Questions | Poorvu Center for Teaching and Learning](https://poorvucenter.yale.edu/MultipleChoiceQuestions#:~:text=,answer%20from%20question%20to%20question)). מומלץ לשלב בכל מסיח רכיב שמקורו בקוד אמיתי מהמאגר (להגדיל את האותנטיות), אך לוודא שלא מדובר בקוד שמופיע באמת ברשומת האק"ג הזו (כדי שלא יציגו יותר מאבחנה אמיתית אחת). ניתן להשתמש ב־scp\_statements.csv כדי לשאוב רשימת קודים מאותה קטגוריית-על. למשל, אם האבחנה הנכונה היא מקטגוריית *rhythm* (הפרעות קצב), לבחור עוד 3 קודים מקטגוריית rhythm כמסיחים.
4. **הצגה וערבוב:** לאחר שיש ארבע אפשרויות (תשובה נכונה + 3 מסיחים), יש לערבב אותן בסדר אקראי בכל שאלה, כדי למנוע הטיה של מיקום התשובה הנכונה. במבני הנתונים, אפשר לסמן איזה מבין הארבע הוא הנכון (למשל אינדקס או ערך). בשלב התצוגה למשתמש, ודא שכל האפשרויות מוצגות בפורמט אחיד (למשל כל האפשרויות הן שמות אבחנות תמציתיים בני מילה-שתיים).
5. **דוגמא למסיחים מבוססי בלבול נפוץ:** נניח תרשים אק"ג שמציג פרפור פרוזדורים מהיר. תשובה נכונה: "פרפור פרוזדורים (AFIB)". מסיחים אפשריים: "רפרוף פרוזדורים (AFLutter)", "טכיקרדיה על-חדרית (SVT)", "קצב סינוסי בלתי סדיר" – כולם מצבים שיכולים לבלבל סטודנט שאינו בקיא (למשל בלבול בין פרפור לרפרוף), אך רק אחד נכון לתרשים. דוגמה נוספת: תרשים עם סימני **LBBB** (חסם צרור שמאלי מלא) – תשובה: "חסם צרור שמאלי מלא"; מסיחים: "חסם צרור ימני (RBBB)", "היפרטרופיה חדרית שמאלית (LVH)", "אק"ג תקין". כאן שני המסיחים הראשונים חולקים חלק מהתכונות (QRS רחב ב-LBBB וברBBB, מתח חשמלי גבוה ב-LVH יכול להזכיר QRS רחב), והמסיח השלישי של "תקין" נועד לבחון שהלומד אכן מזהה שהאק"ג פתולוגי.
6. **התאמת רמת הקושי:** ניתן לקבוע כללים דינמיים לבחירת המסיחים כדי לשלוט ברמת הקושי. לדוגמה, עבור מתחילים – לבחור מסיחים "קלילים" יותר (שונים יותר מהתשובה הנכונה). עבור מתקדמים – מסיחים דומים מאוד לתשובה הנכונה. בנוסף, המערכת יכולה לנטר בחירות נפוצות שגויות של משתמשים ולשפר את המסיחים בהתאם (כדי ללכוד טעויות נפוצות). בטווח הארוך, אפשר אף להשתמש במודלי AI ליצירת מסיחים מתוחכמים המבוססים על טקסט חופשי (למשל לתת ל-GPT-4 לתאר אבחנה שגויה באופן אמין) – **הרחבה אפשרית** לשיפור הגיוון של השאלות (אפיון ECG.docx).

**יצירת טקסט תיאורי ברור עבור כל תרשים ECG**

אחד המרכיבים המרכזיים בלמידה אפקטיבית הוא המשוב המוסבר. לאחר שהמשתמש ענה על שאלה (נכון או לא נכון), המערכת צריכה להציג הסבר מילולי ברור ושלם המבוסס על אותו תרשים. להלן עקרונות ליצירת טקסט תיאורי לכל רשומת אק"ג:

* **שליפת מידע קליני מהמאגר:** בעזרת ה־Data Loader, נקבל עבור כל תרשים את כל המידע הזמין: קודי האבחון (לדוגמה {'AFIB': 100.0, 'LVH': 50.0}), סיווגי הקוד (איזה מהם קצב, איזה מבני-צורה וכו' – מידע שקיים ב-scp\_statements.csv ([PTB-XL, a large publicly available electrocardiography dataset v1.0.3](https://physionet.org/content/ptb-xl/#:~:text=,diagnostic_subclass))), נתוני המטופל (גיל, מין) ונתוני ההקלטה (כגון ציר הלב = heart\_axis, שלבי אוטם = infarction\_stadium אם רלוונטי, ועוד). לדוגמה, **ציר הלב** (Axis) מחושב ע"י מכשיר האק"ג וזמין ברשומה כערך כמו LAD/ RAD (סטיית ציר שמאלית/ימנית) (ptbxl\_database.csv), ויכול להיכלל בהסבר. גם **שלב האוטם** (Stadium I/II/III) זמין לחלק מהרשומות – למשל Stadium I עשוי לרמז על אוטם חד, בעוד Stadium III על ממצא ישן.
* **מבנה ההסבר הטקסטואלי:** מומלץ לבנות תבנית כללית להסבר, כדי להבטיח אחידות ושההסברים יהיו מקיפים. למשל, ניתן לחלק לפסקאות קצרות:
  1. **נתוני רקע:** תיאור גיל ומין המטופל ונסיבות ההקלטה (אם יש). לדוגמה: "*מטופל בן 68, בזמן מנוחה, הקלטת אק"ג בת 10 שניות.*"
  2. **קצב ולב וחיבורי הבסיס:** ציון מהו קצב הלב הכללי ומקצב היסוד. אם יש קוד קצב (rhythm) ייעודי – להשתמש בו. למשל "מקצב: פרפור פרוזדורים מהיר עם תגובה חדרית ~140 לדקה (בלתי סדיר)" או "קצב סינוס תקין, ~75 פעימות לדקה". את קצב הלב אפשר לאמוד: ברשומות רבות של PTB-XL, אם מצוין קוד כמו SBRAD (סינוס ברדיקרדיה) או STACH (סינוס טכיקרדיה) – זה רמז לקצב איטי/מהיר. ניתן גם לחשב גס את הדופק מהאות (למשל, איתור מרווח R-R מתוך הנתונים הגולמיים). במידה ויש **קוצב לב (PACE)** או פעימות מוקדמות (PAC/PVC) הנרשמות בקודים, יש להזכיר זאת. למשל: "*נראים סימנים לפעימות חדריות מוקדמות (PVCs) הפזורות בתרשים*".
  3. **ממצאים עיקריים באק"ג:** כאן נשתמש בקודי ה-SCP האבחנתיים והצורתיים. עבור כל קוד משמעותי, נשלב אותו בהסבר בשפה טבעית. למשל, אם אחד הקודים הוא CLBBB (חסם צרור שמאלי מלא), נסביר את המשמעות: "*נראה חסם מלא של הצרור השמאלי – קומפלקסי QRS רחבים (≥120ms) עם עיוות אופייני (scp\_statements.csv).*" אם קיים קוד LVH (היפרטרופיה חדרית שמאלית), נציין: "*מתח חשמלי מוגבר המחשיד להיפרטרופיה של החדר השמאלי.*" יש לשים דגש על תרגום הקוד להסבר קליני ברור. לשם כך, ניתן להעזר בעמודת התיאור (description) בקובץ scp\_statements.csv המכילה תיאור טקסטואלי של כל קוד (באנגלית) – ולתרגמה לעברית. למשל, קוד IMI מתואר כ"inferior myocardial infarction" (scp\_statements.csv), שניתן לתרגם "אוטם שריר הלב תחתון". אפשר אף לשלב בסוגריים את הקוד עצמו באנגלית לאחר התיאור בעברית כדי לחבר את הלומד לקידוד (לדוגמה: "*אוטם תחתון ישן (IMI)*").
  4. **סיווג וחומרה:** אם רלוונטי, ציין תחת איזה סיווג נופל כל ממצא. למשל: "*האבחנה שייכת לקטגוריית* ***הפרעות הולכה*** *(Conduction Disturbances)*", או "*ממצא זה הוא מסוג* ***אבחנה צוּרָתִית*** *(Form) המתארת שינויי ST/T ללא משמעות חד-משמעית.*" מידע זה קיים בעמודות diagnostic\_class ו-statement category שבקובץ הקודים (scp\_statements.csv). למשל, קוד LAFB (חסם צרור קדמי שמאלי) משויך לקטגוריית "הפרעות הולכה תוך-חדריות" (Intraventricular Conduction Disturbances).
  5. **סיכום ומשמעות קלינית:** סכם את המשמעות הכוללת של הממצאים. למשל: "*הממצאים הללו מצביעים על פרפור פרוזדורים בלתי מבוקר בשילוב עם סימני אוטם ישן בקיר תחתון. מבחינה קלינית, שילוב כזה עשוי להעיד על מחלת לב איסכמית בעבר והפרעת קצב מתמשכת הדורשת טיפול בנוגדי קרישה.*"
* **כתיבה בהירה ומובנית:** ההסבר צריך להיות כתוב בשפה קריאה (עברית רפואית מובנת). הימנע משימוש במונחים טכניים מדי ללא הסבר. עדיף משפטים קצרים יחסית, או נקודות, כדי שהלומד יוכל לעקוב. לדוגמה, במקום "*AV block first degree observed*", לכתוב "*ניכר הארכת מקטע PR המצביעה על חסם עליתי-חדרי מדרגה ראשונה*".
* **הוספת נתונים מספריים וקישורים:** במידת האפשר, שלב בהסבר פרטים כמותיים כדי להמחיש את הנקודות. אם חושב קצב הלב – ציין "≈80 לדקה". אם נמדד אורך QRS או QT – אפשר לציין "משך QRS -45°)"). בנוסף, **"קישורים"** יכולים להיות בצורה של הפניה למונחים: למשל, לא צריך לינק אמיתי אבל אפשר לסמן מונחים חשובים בכוכבית או סוגריים עם הסבר. לדוגמה: "*QT מוארך*\* (אשר עלול לגרום להפרעות קצב חמורות)..." – ובמשפט הבא להסביר מהו QT ארוך. אם הממשק מאפשר, אפשר שבקליק על מונח יוצג ToolTip עם הסבר או שייפתח קישור להגדרה מורחבת (כגון בערך ויקיפדיה רלוונטי או אתר רפואי).
* **התאמת השפה לרמת הלומד:** ייתכן ויהיה כדאי להציג גרסאות הסבר ברמות שונות. למשל, הסבר בסיסי לסטודנט שנה מוקדמת והסבר מפורט למתקדמים. זהו שיפור אופציונלי - ניתן להתחיל בהסבר אחד לכולם.

**דוגמה לטקסט תיאורי עבור תרשים נתון**

נניח שהמערכת ייצרה שאלה על תרשים עם קוד אבחנה עיקרי של **חסם צרור שמאלי מלא (CLBBB)**, וכן ברשומה ישנו קוד של פעימות חדריות מוקדמות **(PVC)**. לאחר שהמשתמש ענה, יוצג למשל ההסבר הבא:

**תרשים אק"ג מס׳ 256** – מטופל בן 80. בקצב הבסיס נצפה קצב סינוס (גלי P נוכחים) עם חסם מלא של הצרור השמאלי (Complete LBBB). קומפלקסי QRS רחבים במיוחד (כ-160 מילישניות) עם פיצול אופייני – ממצא התואם לחסם שמאלי (scp\_statements.csv). בנוסף, ניכרות פעימות חדריות מוקדמות (PVCs) בודדות – קומפלקסי QRS המופיעים טרם זמנם, רחבים ושונים בצורתם, ללא גל P קודם. ציר הלב סטה שמאלה (LAD). בסיכום, האק"ג אינו תקין: הוא מדגים הפרעת הולכה תוך-חדרית משמעותית (LBBB, מסוג הפרעות הולכה על-סיביות) יחד עם הפרעת קצב חדרית בנונית (פעימות מוקדמות). הממצא מצריך בירור גורמי רקע (לרוב LBBB מעיד על מחלת לב מבנית כגון אוטם קודם) ומעקב אחר שכיחות ה-PVC.

בדוגמה זו אפשר לראות שילוב של כל המרכיבים: פתיחה עם נתוני מטופל, תיאור הקצב, תיאור הממצאים העיקריים עם הסבריהם, והקשר קליני. מערכת הלמידה תפיק באופן אוטומטי טקסט דומה לכל תרשים, באמצעות שילוב התבניות עם הנתונים הספציפיים של אותו תרשים.

**טכנולוגיות Python וספריות מתאימות**

להלן כלים וטכנולוגיות מבוססי Python שיסייעו במימוש המערכת על רכיביה:

* **קריאת ועיבוד נתונים (Data Loading):** ספריית pandas מתאימה לטעינת קובצי ה-CSV (ptbxl\_database.csv ו-scp\_statements.csv) ולניהול הנתונים בטבלאות. עבור קריאת אותות האק"ג הגולמיים (פורמט WFDB), מומלץ להשתמש בספריית [wfdb](https://pypi.org/project/wfdb/) אשר מיועדת למטרה זו. ספרייה זו תאפשר לקרוא את קובצי ה-.dat/.hea ולקבל מערכים של אותות הזמן לכל 12 הערוצים, וכן מטא-דאטה מהכותרת (כמו דגימת תדר, Gain וכדומה). אם המערכת צריכה לחשב קצב לב או לזהות קומפלקסים, ניתן להעזר גם בספריות עיבוד אות כמו scipy.signal (לסינון וגרירת נתונים) או בחבילות ייעודיות ל-ECG דוגמת heartpy או neurokit2 לצורך איתור פיקים של R.
* **יצירת תמונות וגרפים:** matplotlib היא הבחירה הקלאסית לשרטוט האק"ג על גרף ולשמירתו כתמונה. ניתן ליצור פונקציה שתפרוס את 12 הלידים במבנה 4x3 (או 3x4) על Canvas ותשרטט כל אחד בצבע שונה או כולם בשחור על רקע לבן, בדומה לדפי אק"ג קליניים. לשיפור הנראות, ניתן להשתמש גם ב-matplotlib יחד עם seaborn עבור סטייל גרפי, או plotly אם רוצים גרף אינטראקטיבי (למשל שהמשתמש יוכל לרחף עם העכבר ולראות ערכי זמן-מתח). עם זאת, לרוב תמונה סטטית מספיקה ללמידה. ודא לציין על התרשים את קצב הדגימה/משך (10 שניות) ולסמן צירים של מתח (mV) וזמן (ms) כדי לטפח הרגלי קריאה נכונים.
* **לוגיקה של יצירת השאלות:** כתיבת הקוד הלוגי יכולה להעשות ישירות בפייתון סטנדרטי. שימוש במחלקות יעזור, אך אין צורך בספריות צד-שלישי מיוחדות לשלב זה. עם זאת, ייתכן שתרצה להשתמש ב-numpy לחישובים (למשל להגריל בחירה או לערבב את רשימת התשובות בצורה וקטורית). כמו כן, ספריית random של פייתון יכולה לשרת לערבוב ובחירת מסיחים אקראית מתוך רשימות.
* **ניהול תהליכים ואוטומציה:** אם הכוונה לייצר **מאגר שאלות גדול** ולשמור אותו כקובץ (לשימוש בפלטפורמה חיצונית), ניתן להשתמש בספריית tqdm להצגת Progress Bar בזמן יצירת אלפי שאלות (אפיון ECG.docx). בנוסף, כדאי לממש אפשרות לייצא את התוצאות כקובץ CSV או JSON (כפי שהוגדר במפרט) (אפיון ECG.docx). זה יאפשר גמישות – למשל טעינת השאלות המאורגנות לאפליקציה אינטרנטית, או עדכון חוזר של מסיחים.
* **ממשק משתמש אינטראקטיבי:** עבור אבטיפוס מהיר, ניתן להשתמש בספריית Streamlit ליצירת ממשק וובי פשוט בפייתון ללא צורך בכתיבת HTML/CSS/JS. Streamlit מאפשר להציג תמונה ושאלות רב-ברירה, ולטפל באינטראקציה של המשתמש באופן פשוט. אפשרות אחרת היא פריימוורק וובי כמו Flask או Django כדי לחשוף את השאלות דרך דפדפן, אך זה ידרוש מעט יותר עבודת פיתוח Front-End. אם מחשיבים אפליקציה מקומית, אפשר גם להשתמש ב-library כמו tkinter ל-GUI בסיסי, אם כי חוויית המשתמש בוובי תהיה לרוב טובה ונגישה יותר (למשל, מאפשרת שימוש במכשירים ניידים). **הערה:** המפרט מציע שיפור עתידי על ידי הוספת ממשק Streamlit לנוחות (אפיון ECG.docx) – זה אכן כיוון מומלץ.
* **ספריות נוספות:** כדאי לכלול טיפול בלוגים וחריגות – למשל להשתמש בספריית logging המובנית של פייתון כדי לתעד שגיאות (כמו כשל בטעינת קובץ נתונים) לקובץ טקסט (אפיון ECG.docx). לבדיקות יחידה ואינטגרציה של המודולים, להשתמש ב-PyTest או unittest (כדי לוודא שיצירת השאלות עובדת, שההסברים נוצרים נכון וכו').
* **ביצועים:** מאגר PTB-XL מכיל 21,879 רשומות – עלולה להיות כבידות ביצועים אם ננסה לטעון את כולו לזיכרון או לייצר אלפי תמונות בבת אחת. מומלץ לתכנן טעינת נתונים עצלה (lazy loading) או עבודה במנות. לחלופין, אם מטרת המערכת היא *להתאמן* ולאו דווקא לייצר את כל השאלות מראש, אפשר בכל פעם לבחור רשומה אקראית, לטעון אותה, ליצור שאלה ולהציג – וכך לא צריך זיכרון ענק. זמן הטעינה והשרטוט לכל שאלה אמור להיות קטן משנייה-שתיים (ניתן אף למדוד ולוודא שהמערכת עומדת בזה כמדד איכות (אפיון ECG.docx)).

**רעיונות לשיפור חוויית הלמידה ביחס למבחנים רגילים**

היתרון במערכת מבוססת-נתונים ותגובתית הוא האפשרות לספק חוויית לימוד עשירה וגמישה בהרבה ממבחן אמריקאי סטטי. הנה כמה הצעות לשיפור חוויית המשתמש והלמידה בהשוואה לבחנים רגילים:

* **משוב מיידי ומפורט:** בניגוד למבחן רגיל שבו הסטודנט מגלה מה התשובה הנכונה רק בסוף (וללא הסבר), כאן לכל שאלה מתלווה הסבר מפורט מיד לאחר המענה. המשוב המיידי, הכולל נימוק קליני ואפילו סימון המרכיבים על גבי האק"ג, יעזור ללומד להבין את טעויותיו ולהפנים את החומר מהר יותר. ניתן לשקול הדגשה ויזואלית על התרשים בעת מתן ההסבר – למשל, לסמן בצבע את מקטעי ST הרלוונטיים באוטם, או להדליק חץ לפעימה מוקדמת – כדי לקשר בין ההסבר הטקסטואלי לתרשים עצמו.
* **התאמה אישית וניתוח ביצועים:** המערכת יכולה לעקוב אחר ביצועי המשתמש לאורך הזמן. למשל, לזהות שהוא שוגה לעיתים קרובות בשאלות על הפרעות קצב, ולהציע לו יותר תרגול באותו נושא. זה יוצר מסלול למידה אדפטיבי ואישי – מה שקשה מאוד בבחנים מסורתיים. בנוסף, אפשר להציג סטטיסטיקות אישיות: אחוזי הצלחה לפי קטגוריות (קצב, הולכה, אוטמים), זמנים ממוצעים למתן תשובה, התקדמות מנקודה לנקודה. פידבק כזה מעודד שיפור עצמי ומיקוד בנקודות חלשות.
* **מצבי תרגול מגוונים:** בעוד מבחן רגיל הוא חד-פעמי ולרוב בפורמט אחיד, המערכת יכולה להציע מצבי תרגול שונים. למשל: **מצב למידה מודרכת** – שבו לאחר כל שאלה ההסבר מופיע אוטומטית, ואף יש רמזים טרם בחירה (אם המשתמש רוצה). לעומת זאת **מצב בחן עצמי** – ללא רמזים והסברים רק בסוף הסשן, כדי לדמות מבחן אמיתי. אפשרות נוספת היא **בחנים לפי נושאים**: לאפשר למשתמש לבחור להתמקד רק בקטגוריה אחת (למשל רק קצב לב/הפרעות הולכה) ולתרגל אותה בנפרד. גמישות זו אינה קיימת במבחנים מקובעים מראש.
* **מאגר שאלות רחב ודינמי:** בניגוד לבוחן רגיל שבו מספר השאלות מוגבל, כאן ניתן לשאוב מעשרות אלפי הרשומות במאגר PTB-XL וליצור וריאציות רבות. המשמעות היא שהלומד כמעט ולא יקבל את אותה שאלה פעמיים, וייחשף למגוון אדיר של תרשימים אמיתיים ומקרים קליניים. זה כשלעצמו משפר את הלמידה (כי הלומד לא "זוכר את התשובה" משאלה קודמת אלא באמת צריך לנתח כל תרשים חדש). בנוסף, ניתן לעדכן את המאגר או לשלב **תרשימי אק"ג נוספים** ממקורות פתוחים כדי להרחיב עוד את השאלות – דבר שקשה לעשות במבחנים רגילים ללא הפקת תוכן חדשה.
* **מישחוק (Gamification):** אפשר להוסיף אלמנטים של משחק כדי להעלות מוטיבציה. לדוגמה, ניקוד מצטבר, לוח תוצאות של משתמשים מובילים, "תגים" שמוענקים על הישגים (כגון "אשף הפרעות קצב" למי שענה נכון על 50 שאלות רצופות בקטגוריה זו). אלמנטים כאלו מדרבנים את הלומדים להתמיד ולהשתפר, והם בלתי אפשריים בבחנים סטטיים חד-פעמיים.
* **עדכניות ודיוק מקצועי:** המערכת יכולה להיעזר בעדכוני מאגר הקודים וההנחיות הרפואיות. למשל, אם מתפרסמות הנחיות חדשות לגבי אבחון אק"ג (כגון קריטריונים מעודכנים ל-LVH או הארכת QT), ניתן לשלב אותן בהסברים או באופן ניקוד השאלות. זאת לעומת שאלות סטטיות שעשויות להיות מבוססות על ידע שהוא מעט מיושן. מעבר לכך, שימוש במאגר PTB-XL שפותח למחקר מאפשר ודאות גבוהה לגבי נכונות האבחנות (הרי כל תרשים שם סווג ע"י קרדיולוג/ים) ([PTB-XL, a large publicly available electrocardiography dataset v1.0.3](https://physionet.org/content/ptb-xl/#:~:text=The%20PTB,The%20dataset%20is)), לעומת מבחנים שאולי נבנו ביד אנושית בודדת ועשויים לכלול טעויות.
* **שילוב מקורות מידע נוספים:** שיפור חוויית הלמידה יכול להתבטא גם במתן אפשרות "חקירה עצמית". למשל, לאחר הצגת ההסבר, לספק כפתור "למידע נוסף" שיוביל את המשתמש למאמר קצר או קטע מספר לימוד על אותו נושא (קישור לחומר תאורטי על פרפור פרוזדורים, למשל). כך המערכת משמשת לא רק כבודקת ידע אלא כנקודת מוצא ללמידה מעמיקה. ניתן לשלב קישורים למקורות פתוחים או לתכני PhysioNet הרלוונטיים.
* **התנסות ריאליסטית יותר:** מכיוון שיש לנו נתוני מטופל וקליניקה, ניתן לדמות **סצנות קליניות** קטנות בתוך השאלות. למשל, להציג שאלה בסגנון מקרה: "*מטופל בן 50 הגיע לחדר מיון עם תלונה על דפיקות לב מהירות. מצורף תרשים אק"ג שלו. מה האבחנה הסבירה?*" ובתשובות לשים אבחנות מבדלות. זה מאמן את הסטודנט בצורה שיותר קרובה לאופן החשיבה במציאות. מאגר PTB-XL יכול לתמוך בכך כי לכל תרשים יש הקשר (למשל, פרפור פרוזדורים שכיח יותר במבוגרים עם היסטוריה מסוימת וכו'). באופן זה הלמידה עולה מדרגה לעומת שאלה תיאורטית קרה.

לסיכום, שילוב מאגר PTB-XL במערכת תרגול אק"ג מאפשר ליצור פלטפורמה דינמית, מקיפה ומעמיקה ללימוד פענוח אק"ג. המבנה המוצע עם רכיבים מודולריים, שאלות רב-ברירה מבוססות תרשימים אמיתיים, הסברים קליניים מפורטים ושימוש בכלים פייתוניים חזקים – כל אלו יאפשרו אלטרנטיבה משופרת לבחנים המקוונים הרגילים, ויתרמו להכשרת דור חדש של מפענחי אק"ג מיומנים. ([PTB-XL, a large publicly available electrocardiography dataset v1.0.3](https://physionet.org/content/ptb-xl/#:~:text=The%20PTB,The%20dataset%20is))

כמובן! הנה הוראות מפורטות וברורות לגבי כל התהליך, כולל כל המידע שנאסף עד כה. כל שלב כתוב בצורה כזו שניתן להעתיק את הקוד ולהשתמש בו כחלק מהפיתוח, תוך מיקוד במזעור שגיאות ובלבול. ההוראות כוללות גם את כל החלקים הנדרשים להתחלה והמשך הפיתוח בצורה מסודרת.

**1. הגדרת תיקיית מאגר הנתונים**

בשלב הראשון, יש צורך לאפשר למשתמש לבחור תיקיית מאגר שבה מאוחסנים הנתונים (קובצי ה-CSV והנתונים של PTB-XL). נשתמש ב-tkinter ליצירת חלונית בחירת תיקייה.

**קוד לבחירת תיקיית המאגר:**

import tkinter as tk

from tkinter import filedialog

def select\_directory():

# יצירת חלונית בחירת תיקייה

root = tk.Tk()

root.withdraw() # מסתיר את חלון ברירת המחדל של Tkinter

directory = filedialog.askdirectory(title="Select PTB-XL Dataset Directory") # בחירת תיקייה

if directory:

print(f"Selected directory: {directory}")

else:

print("No directory selected.")

return directory

# קריאה לפונקציה

dataset\_directory = select\_directory()

**2. טעינת הנתונים**

לאחר בחירת תיקיית המאגר, נטען את קובצי ה-CSV (ptbxl\_database.csv ו-scp\_statements.csv) עם הנתונים של PTB-XL באמצעות pandas.

**קוד לטעינת הנתונים:**

import pandas as pd

import os

def load\_data(dataset\_directory):

# בודק אם הקבצים קיימים בתיקייה שנבחרה

ptbxl\_file = os.path.join(dataset\_directory, 'ptbxl\_database.csv')

scp\_file = os.path.join(dataset\_directory, 'scp\_statements.csv')

if os.path.exists(ptbxl\_file) and os.path.exists(scp\_file):

# טוען את קובצי ה-CSV

ptbxl\_data = pd.read\_csv(ptbxl\_file)

scp\_data = pd.read\_csv(scp\_file)

print(f"Loaded PTB-XL dataset with {len(ptbxl\_data)} records.")

print(f"Loaded SCP statements dataset with {len(scp\_data)} records.")

return ptbxl\_data, scp\_data

else:

print("Required files are missing in the selected directory.")

return None, None

# קריאה לפונקציה

ptbxl\_data, scp\_data = load\_data(dataset\_directory)

**3. יצירת שאלה**

בשלב הבא, המערכת תבחר רישום אק"ג אקראי מהנתונים, תיצור את ההבחנה הנכונה מתוך קודי ה-SCP, תבחר מסיחים ותיצור את השאלה.

**קוד ליצירת השאלה:**

import random

def generate\_question(ptbxl\_data, scp\_data):

# בוחרים רישום אק"ג אקראי

random\_record = ptbxl\_data.sample(1).iloc[0] # בוחר רישום אקראי

ecg\_id = random\_record['ecg\_id']

scp\_codes = random\_record['scp\_codes']

# מוצאים את ההבחנה העיקרית מתוך קודי ה-SCP

main\_diagnosis\_code = max(scp\_codes, key=lambda x: x[1]) # קוד עם ההסתברות הגבוהה ביותר

main\_diagnosis = scp\_data[scp\_data['code'] == main\_diagnosis\_code]['description'].iloc[0]

# יצירת מסיחים (מסיחים מקרוב אך שגויים)

possible\_diagnoses = list(scp\_data['description'].sample(5)) # בחר 5 הצהרות אקראיות

possible\_diagnoses.remove(main\_diagnosis) # מסיר את ההבחנה הנכונה

distractors = random.sample(possible\_diagnoses, 3) # בחר 3 מסיחים

# מערבבים את האפשרויות (כולל התשובה הנכונה)

options = [main\_diagnosis] + distractors

random.shuffle(options) # ערבוב אפשרויות

# יצירת שאלה

question = f"What is the most suitable diagnosis for ECG {ecg\_id}?"

return question, options, main\_diagnosis

# קריאה לפונקציה

question, options, correct\_answer = generate\_question(ptbxl\_data, scp\_data)

print(f"Question: {question}")

print(f"Options: {options}")

print(f"Correct Answer: {correct\_answer}")

**4. הסבר לאחר מענה**

לאחר שהמשתמש ענה על השאלה, המערכת תספק הסבר מפורט, תציג אם התשובה נכונה או שגויה ותסביר למה התשובה נכונה או לא נכונה.

**קוד להסבר:**

def provide\_explanation(selected\_answer, correct\_answer, ecg\_id, scp\_data):

if selected\_answer == correct\_answer:

explanation = f"Correct! ECG {ecg\_id} shows {correct\_answer}. This is the most suitable diagnosis."

else:

explanation = f"Incorrect. ECG {ecg\_id} actually shows {correct\_answer}. Here's why:\n"

explanation += f"The correct diagnosis was {correct\_answer} because it matches the key features in the ECG pattern, such as ST elevation, QRS duration, etc."

# תיאור נוסף של ההבחנה (הסבר רפואי)

explanation += f"\nDetailed explanation: {scp\_data[scp\_data['description'] == correct\_answer]['description'].iloc[0]}"

return explanation

# קריאה לפונקציה לאחר תשובה:

selected\_answer = "Inferior Myocardial Infarction" # דוגמת תשובה שנבחרה

explanation = provide\_explanation(selected\_answer, correct\_answer, ecg\_id, scp\_data)

print(explanation)

**5. הערות כלליות על המערכת**

1. **חלונית בחירת תיקיית המאגר**: באמצעות tkinter, המערכת מאפשרת למשתמש לבחור את תיקיית המאגר שמכילה את הקבצים הנדרשים (כמו ptbxl\_database.csv ו-scp\_statements.csv).
2. **טעינת הנתונים**: המערכת בודקת אם הקבצים קיימים בתיקייה שנבחרה. אם כן, היא טוענת את הנתונים בעזרת pandas.
3. **הפקת שאלות**: השאלות מבוססות על רישומים אקראיים מתוך הנתונים. השאלות כוללות את ההבחנה הנכונה שנבחרה על פי ההסתברות של קודי ה-SCP, והמסיחים נבחרים מתוך אפשרויות דומות אך שגויות.
4. **הסברים**: לאחר שהמשתמש ענה על השאלה, המערכת מספקת הסבר מפורט אם התשובה נכונה או לא, תוך ציון הסיבה והתיאור הרפואי של ההבחנה.

**6. אפשרויות שדרוג (בשלב מאוחר יותר)**

* **תמיכה בממשק וובי**: אפשרות להוסיף ממשק משתמש בעזרת ספריית Streamlit או Flask כדי להציג את השאלות וההסברים בצורה נוחה.
* **שיפור חוויית הלמידה**: ניתן להוסיף מעקב אחרי ביצועים של המשתמש לאורך זמן ולספק משוב על שגיאות נפוצות.
* **סינכרון עם מקור מידע אמין נוסף**: אפשר לשלב API או כלי חיפוש שיאפשרו להציג הסברים רפואיים נוספים מאתרים אמינים (אם תבחר בכך בהמשך).

**סיכום**

ההוראות מפורטות של התהליך כוללות את כל השלבים מההתחלה ועד הסוף. כל פונקציה מוגדרת בצורה ברורה, והקוד מותאם להתפתחות הדרגתית. בהמשך, אפשר לשדרג ולהוסיף פונקציות נוספות בהתאם לצורך.

אם תצטרך להציג לי משהו או להוסיף שדרוגים, תמיד תוכל לפנות אליי כדי לעדכן את המערכת בהתאם לשינויים שתבחר לעשות.

מעולה, אני אבדוק מהי עמודת המפתח המשותפת בין הקבצים ptbxl\_database.csv ו-scp\_statements.csv מתוך מאגר PTB-XL ואסביר את הקשר ביניהם. אעדכן אותך בעוד רגע עם הממצאים.

**עמודת המפתח המשותפת בין ptbxl\_database.csv ו-scp\_statements.csv במאגר PTB-XL**

**PTB-XL** הוא מאגר נתונים גדול של תרשימי אק"ג קליניים (12-הד־לב, 10 שניות כל אחד) הכולל 21,799 רשומות מ-18,869 מטופלים ([PTB-XL, a large publicly available electrocardiography dataset v1.0.3](https://physionet.org/content/ptb-xl/#:~:text=The%20PTB,The%20dataset%20is)). כל תרשים אק"ג במאגר סומן (annotated) על ידי קרדיולוג אחד או שניים עם מספר **הצהרות אבחון** לפי תקן **SCP-ECG** – אוסף של 71 תגיות/הצהרות אפשריות המתארות ממצאים אבחנתיים, צורת גלים או קצב לב בתרשים ([PTB-XL, a large publicly available electrocardiography dataset v1.0.3](https://physionet.org/content/ptb-xl/#:~:text=The%20PTB,The%20dataset%20is)). הנתונים מאורגנים בשתי טבלאות עיקריות: ptbxl\_database.csv ו-scp\_statements.csv. להלן פירוט כל טבלה ועמודת המפתח המשותפת ביניהן.

**ptbxl\_database.csv – רשומות ECG ותיוגים אבחנתיים**

טבלת **ptbxl\_database.csv** היא טבלת המטא-נתונים הראשית של המאגר, בה כל שורה מייצגת **רשומת ECG יחידה**. עבור כל תרשים אק"ג, הטבלה מכילה מזהה ייחודי של הרישום (ecg\_id) ומזהה מטופל (patient\_id), וכן נתוני דמוגרפיה ופרטי הקלטה (כגון גיל, מין, גובה, משקל, תאריך הקלטה ועוד). בנוסף, כוללת הטבלה שדות המתארים את פענוח התרשים: למשל **דוח טקסטואלי** חופשי (report) שנוצר ע"י המכשיר או הקרדיולוג, וחשוב לענייננו – עמודת **scp\_codes** ([PTB-XL, a large publicly available electrocardiography dataset v1.0.3](https://physionet.org/content/ptb-xl/#:~:text=3,validated_by_human)).

עמודת scp\_codes מכילה **מילון (dictionary)** של קודי ההצהרות לפי תקן SCP-ECG עבור אותו רישום. כל ערך במילון הוא זוג של **קוד: הסתברות/מידת ודאות** שמציין אילו הצהרות אבחנתיות זוהו באק"ג ובאיזו סבירות. למשל, רשומת אק"ג תקינה עשויה להכיל את הקוד 'NORM': 100 (נורמלי בוודאות גבוהה), בעוד שרשומה עם חשד לאוטם תחתון תכיל אולי קוד כמו 'IMI': 35.0 לצד קודים נוספים. אם הצהרה מסוימת אינה רלוונטית, מידת הוודאות שלה תהיה 0 ([PTB-XL, a large publicly available electrocardiography dataset v1.0.3](https://physionet.org/content/ptb-xl/#:~:text=3,validated_by_human)). בצורה זו, לכל תרשים משויכים אחד או יותר **קודי SCP-ECG** המתארים את האבחנות והממצאים שלו.

**scp\_statements.csv – הצהרות SCP-ECG וקטגוריותיהן**

טבלת **scp\_statements.csv** היא טבלה המסכמת את **כל 71 קודי ההצהרות** (SCP-ECG statements) הקיימים במאגר. כל שורה בטבלה זו מייצגת **קוד הצהרה ייחודי** (למשל NORM, IMI, LAFB וכו') ומספקת מידע אודותיו: תיאור מילולי של ההצהרה, וסיווגים שונים. בין העמודות בטבלה ניתן למצוא:

* **description** – תיאור טקסטואלי של ההצהרה (לדוגמה: "normal ECG" עבור NORM, "inferior myocardial infarction" עבור IMI).
* **קטגוריות אבחון** – עמודות תאוריות המציינות האם הקוד שייך לקטגוריה אבחנתית, צורתית (form), או של קצב (rhythm) באמצעות סימון (diagnostic, form, rhythm – ערך 1/0 או True/False). קוד יכול להשתייך ליותר מקטגוריה אחת, בהתאם למה שהוא מתאר.
* **diagnostic\_class** ו-**diagnostic\_subclass** – סיווג היררכי עבור קודים אבחנתיים, המאגד קודים דומים לקבוצות-על. למשל, קודים של אוטמי לב (MI) יחלקו מחלקה משותפת, וקוד NORM (אק"ג תקין) משויך למחלקת NORM הן ברמת class והן ב-subclass, שכן הוא מייצג תרשים תקין ([PTB-XL, a large publicly available electrocardiography dataset v1.0.3](https://physionet.org/content/ptb-xl/#:~:text=All%20information%20related%20to%20the,diagnostic_subclass)).
* בנוסף, הטבלה מכילה מיפויים לתקני קידוד אחרים (כגון קודי AHA, CDISC, DICOM) עבור כל הצהרה, למטרות התאמה לתוויות במערכות אחרות ([PTB-XL, a large publicly available electrocardiography dataset v1.0.3](https://physionet.org/content/ptb-xl/#:~:text=All%20information%20related%20to%20the,diagnostic_subclass)).

בקצרה, scp\_statements.csv משמשת כ**מילון התגים/הצהרות** של המאגר: היא מתארת מה משמעות כל קוד SCP-ECG ובאיזו קטגוריה רפואית הוא נמצא.

**עמודת המפתח המקשרת וקשר בין הטבלאות**

**עמודת המפתח המשותפת** לשתי הטבלאות היא **קוד ההצהרה עצמו** (SCP-ECG Statement Code). בטבלת ptbxl\_database.csv קודים אלה מופיעים כמפתחות בתוך שדה ה-scp\_codes של כל רשומה (Keys במילון התגים של אותה רשומת ECG). בטבלת scp\_statements.csv אותם קודים מופיעים בעמודה הראשונה (שם השורה) של הטבלה, כמזהה ההצהרה המתואר באותה שורה. למעשה, **כל קוד המופיע ב-scp\_codes אמור להימצא כערך בעמודה הראשונה של scp\_statements.csv**, ולהפך – scp\_statements.csv מונה את כל הקודים האפשריים שיכולים להופיע ב-scp\_codes ([PTB-XL, a large publicly available electrocardiography dataset v1.0.3](https://physionet.org/content/ptb-xl/#:~:text=All%20information%20related%20to%20the,diagnostic_subclass)).

**הקשר בין הטבלאות:** טבלת הרשומות משתמשת בקודי SCP-ECG כדי לתייג כל תרשים בממצאים אבחנתיים, וטבלת ההצהרות מספקת את *הפירוש והקטלוג* של אותם קודים. כלומר, ptbxl\_database.csv אומרת אילו מצבים אובחנו בכל ECG (באמצעות הקודים), ו-scp\_statements.csv מפרטת מה פירוש כל קוד כזה והאם הוא אבחנתי, מתאר צורת גל, או קשור לקצב הלב, וכו' ([PTB-XL, a large publicly available electrocardiography dataset v1.0.3](https://physionet.org/content/ptb-xl/#:~:text=The%20PTB,The%20dataset%20is)) ([PTB-XL, a large publicly available electrocardiography dataset v1.0.3](https://physionet.org/content/ptb-xl/#:~:text=All%20information%20related%20to%20the,diagnostic_subclass)). בדרך זו מבטיחים עקביות בהתאם לתקן SCP-ECG – כל התרשימים במאגר מתוארים באמצעות סט אחיד של תגיות מוגדרות מראש.

**מיזוג על בסיס העמודה המשותפת:** כן, ניתן למזג (להצליב) בין הטבלאות באמצעות קוד ההצהרה. בפועל, כדי לבצע חיבור נתונים, יש להשתמש בקוד כערך מקשר: לכל קוד שמופיע בשדה scp\_codes של רשומה, ניתן לאתר את השורה המתאימה לו ב-scp\_statements.csv ולקבל את תיאורו וקטגוריותיו. לדוגמה, אם ברשומה מסוימת ב-ptbxl\_database מופיע הקוד "IMI" (אוטם תחתון), אפשר לפנות לטבלת scp\_statements ולמצוא את שורת הקוד "IMI" כדי לדעת שהיא מתארת "inferior myocardial infarction" ומשויכת לקטגוריית אבחון של אוטם לב.

חשוב לציין שקובץ scp\_statements.csv **אינו מכיל מזהה רשומה** או מידע ספציפי על מטופלים – הוא משמש כמילון סטטי של הקודים. לכן, **עמודת המפתח המדויקת למיזוג** היא *שם הקוד עצמו*: מצד אחד, מפתח המילון (Key) במערך scp\_codes בכל רשומה, ומצד שני, הערך בעמודת הקוד (הטור הראשון) בטבלת ההצהרות. באמצעות מפתח זה, ניתן לקשר בין התגיות שבכל רשומת אק"ג לבין המשמעויות והמאפיינים שלהן.

בסיכום, ptbxl\_database.csv מספקת את **רשומות ה-ECG והתגיות הרפואיות** שהוקצו להן, בעוד scp\_statements.csv מספקת **הגדרות וסיווגים לכל תגית (קוד SCP-ECG)**. עמודת המפתח המשותפת – **קוד ההצהרה** – מקשרת בין התרשים הבודד לתיאור הטקסטואלי והקטגוריה של כל ממצא, ומאפשרת מיזוג מידע בין שני הקבצים על בסיס קוד זה בצורה ישירה. זאת בהתאם לתיעוד המגדיר שקודי SCP-ECG הם הבסיס לאנוטציית הממצאים בכל התרשימים, ומיוצגים כמילון בנתוני הרשומות וכרשימת קודים מפורשת בקובץ הנפרד ([PTB-XL, a large publicly available electrocardiography dataset v1.0.3](https://physionet.org/content/ptb-xl/#:~:text=3,validated_by_human)) ([PTB-XL, a large publicly available electrocardiography dataset v1.0.3](https://physionet.org/content/ptb-xl/#:~:text=All%20information%20related%20to%20the,diagnostic_subclass)).