

פרויקט סופי – ניהול מערכת שעות לסטודנט

השוואה בין אפשרויות מימוש

הנחיות כלליות

יש לממש אפליקציה לניהול מערכת שעות לסטודנט ב – 3 דרכים:

1. מימוש עצמי. ניתן להיעזר בבינה מלאכותית כדי לקבל הסברים, אבל לא כדי לקבל קוד.
2. מימוש ע"י כלי בינה מלאכותית יוצרת כלשהו.
3. מימוש ע"י Vibe Coding.

יש להתחיל מהמימוש העצמי, לאחר מכן למימוש עם בינה ולבסוף מימוש עם vibe coding. הפרויקט יתבצע בזוגות, ויוגש עד ה 14.8.25 ב 9:00 בבוקר. יש [להשתבץ למבחן בע"פ](#) בזום של 20 דקות ב 14.8.25.

מימוש עצמי

בפרויקט זה עליכם לממש אפליקציה לניהול מערכת שעות. הסטודנט יכול לבנות כמה מערכות כרצונו, כאשר הן ימוספרו באופן סדרתי, החל מ-1. לשם כך עליכם להכין את הקבצים הבאים:

שם קובץ	תיאור
schedules.csv	המערכות שבנה הסטודנט.
courses.csv	כל המקצועות.
xxxxx_lectures.csv	(xxxxx מתאר את מספר הקורס). כל ההרצאות בקורס xxxxx.
xxxxx_tutorials.csv	(xxxxx מתאר את מספר הקורס). כל התרגולים בקורס xxxxx.
xxxxx_labs.csv	(xxxxx מתאר את מספר הקורס). כל המעבדות בקורס xxxxx.

מחלקות עיקריות:

Course – מתארת קורס. נתונים: מספר קורס, שם, נק"ז, תאריך מועד א, תאריך מועד ב, מרצה אחראי.

Lesson – מתארת שיעור לשיבוץ במערכת. מחלקה מופשטת. נתונים: מספר קורס, יום, שעת התחלה, משך, כיתה (מספר חדר ובניין), שם מורה, מספר קבוצה (לכל השיעורים באותו קורס יהיה מספר קבוצה שונה).

Lecture – יורשת מהמחלקה Lesson ומתארת הרצאה.

Tutorial – יורשת מהמחלקה Lesson ומתארת תרגול.

Lab – יורשת מהמחלקה Lesson ומתארת מעבדה

Schedule – מתארת מערכת שעות. המידע יישמר ב data structure מה STL לבחירתכם.

התכנון של המחלקות, מבני הנתונים ויחסי הגומלין ביניהם נעשה על ידיכם.

שימו לב:

- ניתן להוסיף מחלקות.
- ניתן להוסיף שדות ומתודות במחלקות לעיל.
- אוסף הקורסים והשיעורים יישמרו ב Data structure מה STL לבחירתכם.
- בפרויקט יהיה ממשק משתמש מבוסס DOS, ללא ממשק גרפי. ממשק כזה נקרא CLI (Command Line Interface).

להלן תיאור הפעולות המופיעות למשתמש על גבי המסך. שימו לב: כאשר מופיעה מילה בסוגריים זוויתיים, למשל <unique_id>, הכוונה שהמשתמש יחליף את המילה ואת הסוגריים הזוויתיים. למשל כדי להדפיס פרטי קורס שמספרו הוא 1000 המשתמש ירשום:

Print 1000

- PrintCourse <course_id> - להדפיס למסך את הפרטים של הקורס, כולל כל השיעורים הקיימים בו.
- PrintCourses - להדפיס את 10 הקורסים הראשונים ברשימת הקורסים. אם יש פחות מ 10 קורסים יודפסו רק הקורסים הקיימים.
- More – פקודה זו חוקית רק מייד לאחר פקודת PrintCourses וניתן לקרוא לה מספר פעמים ברצף. היא תגרום להדפסת 10 הקורסים הבאים ברשימת הקורסים. בחירה נוספת של האופציה הזו תדפיס את 10 הבאים, וכך הלאה. אם נגמרו הקורסים פקודה זו תדפיס הודעה מתאימה למסך. אם בשלב כלשהו יש פחות מ 10 קורסים יודפסו רק הקורסים הקיימים.
- PrintSchedule <schedule_id> - להדפיס את המערכת על פי מספרה. המערכת תודפס בצורה של טבלה, השורה הראשונה תהיה הימים, העמודה הראשונה תהיה השעות (ברזולוציה של שעה). לכל קורס יודפסו הפרטים הבאים: מספר קורס, האם הוא הרצאה, תרגול או מעבדה, וכיתה. אין צורך להדפיס את הזמן והמשך כיוון שהם יתוארו בשוליים של הטבלה. אם שיעור מסוים מתמשך יותר משעה, יש להדפיסו במספר שורות על פי המשך שלו (שורה לכל שעה). אם יש חפיפה בין קורסים, יש להדפיס את כולם.
- PrintSchedules – הדפסת כל המערכות שהכין הסטודנט, על פי הפורמט שתואר קודם לכל מערכת.
- AddSchedule – הוספת מערכת ריקה. מספרה ייקבע באופן אוטומטי להיות המספר הגדול ביותר הקיים ועוד 1.
- RmSchedule <schedule_id> - מחיקת מערכת. לאחר מחיקת המערכת, המספרים המזהים יצטמצמו, כך שלא יהיו קפיצות במספור.
- Add <schedule_id> <course_id> <group_id> - להוסיף את הקורס למערכת (מספר הקבוצה יכול לתאר הרצאה, תרגול או מעבדה).
- Rm <schedule_id> <course_id> <group_id> - למחוק קורס מהמערכת על פי מספר קורס ומספר קבוצה.

כאשר נסיים את התוכנית ונבצע כניסה מחודשת, המערכות שנבנו לא יימחקו. כלומר הפקודה Print תדפיס את כל המערכות שהסטודנט בנה עד כה. לשם כך עליכם לשמור את המערכות בקובץ schedules.csv.

שאר הקבצים יש להכין מראש ידנית.

כמו כן, אתם נדרשים להוסיף לפחות 3 פעולות מקוריות (לא טריוויאליות) שלכם שלא ביקשנו. פעולות אלו צריכות לחייב מעבר על מבני הנתונים שקיים אצלכם בתוכנית (למשל פעולה

שדורשת get או set בלבד, היא פעולה טריוויאלית. פעולה שמדפיסה את כל השיעורים בהם יש חפיפה היא פעולה מקורית לא טריוויאלית).

דרישות כלליות

- בכל המחלקות יש מתודות get/set.
- שימוש ב const, initialization list, call by reference בכל מקום שצריך.
- בכל מקום שצריך, עבודה עם operator overload.
- האופרטור << לכל מחלקה.
- שימוש ב STL עבור מבני הנתונים שלכם (שימו לב שכל מבני הנתונים צריכים להיות כך שיעבדו יעיל).
- שימוש באיטרטורים.
- שימוש ב try-throw-catch במקומות הנחוצים.
- תיעוד (באנגלית בלבד).
- חלוקה לקבצים.

מה אסור שיהיה:

- דליפות זיכרון.
 - שגיאות ריצה (שימו לב לטפל בקלט לא חוקי).
 - לולאות אינסופיות.
 - שכפולי קוד.
 - במימוש העצמי, לא להשתמש בדברים שלא נלמדו בקורס, כגון auto.
- שימו לב: עבודה עם קבצים היא איטית. הגישה לקבצים תתבצע פעם אחת בתחילת התוכנית ופעם אחת בסיומה. וזהו.

מימוש ע"י כלי בינה מלאכותית יוצרת

יש להשתמש באחד מהכלים המוכרים (ChatGPT, Claude, Gemini, Cursor, CoPilot, Grok וכו') כדי לממש את הפרויקט כפי שתואר במימוש העצמי. באחריותכם לוודא שהפרויקט עובד ללא תקלות. ניתן לבקש מהכלי לתקן את הבאגים או לתקן בעצמכם.

דרישות כלליות

- בכל המחלקות יש מתודות get/set.
- שימוש ב const, initialization list, call by reference בכל מקום שצריך.
- בכל מקום שצריך, עבודה עם operator overload.
- האופרטור << לכל מחלקה.
- שימוש ב STL לשמירת הנתונים (שימו לב שכל מבני הנתונים צריכים להיות כך שיעבדו יעיל).
- שימוש באיטרטורים.
- שימוש ב try-throw-catch במקומות הנחוצים.
- תיעוד (באנגלית בלבד).
- חלוקה לקבצים.

מה אסור שיהיה:

- דליפות זיכרון.
- שגיאות ריצה.
- לולאות אינסופיות.
- שכפולי קוד.

Vibe Coding

בקשו מאחד מכלי הבינה הסבר על המושג Vibe Coding.

אנו נשתמש ב Base44 למימוש הפרויקט.

בקישור: <https://docs.base44.com/VideoTutorials> תמצאו מספר סרטוני הדרכה קצרים.

שימו לב: יש להשתמש אך ורק ברשיון החינמי. הרשיון החינמי מקנה לכם 7 שאילתות ביום, ועד 25 בחודש. נצלו אותן בחוכמה.

בניגוד למימושים הקודמים, אנחנו לא אומרים ל Base44 כיצד לממש את האפליקציה. כלומר, אל תגדירו לו כיצד לשמור את המידע, או מה המחלקות, רק אימרו לו מה הפרויקט אמור לעשות. הגדירו לו מה היכולות שיש למשתמש (הדפסת מערכת, הוספת מערכת, הוספת קורס למערכת וכו). כמו כן הוא מייצר אפליקציה וובית, כך שאין משמעות ל CLI. כדי שאוכל לראות את האפליקציה יש לעשות לה publish ולהוסיף להגשה קישור אליה.

הגשת הפרויקט

הקוד יהיה בפונט מינימלי של 12. המצגת תהיה בפונט מינימלי של 24. נא לא לצרף תמונת מסך של הקוד או של דוגמאות הרצה.

יש להגיש את הפרויקט [בתיקיה משותפת](#).

עליכם ליצור בתוך התיקיה המשותפת תיקיה חדשה, ששמה יהיה שמות הסטודנטים באנגלית, ללא רווחים וללא תווים מיוחדים. בתוך תיקיה זו תגישו את:

- קובץ זיפ ששמו מורכב מהת.ז. של הסטודנטים והמילה manual. בקובץ זה יהיה כל הפרויקט במימוש עצמי כך שאפשר להריץ אותו בזמן המבחן בע"פ.
- קובץ זיפ ששמו מורכב מהת.ז. של הסטודנטים והמילה genai ולאחר מכן שם הכלי איתו פיתחתם. בקובץ זה יהיה כל הפרויקט שנוצר עם כלי GenAI כך שאפשר להריץ אותו בזמן המבחן בע"פ.
- מצגת של 10 דקות, ובה:
 - קישור לאפליקצית Vibe Coding
 - השוואה בין שלוש שיטות המימוש:
 - איכות הקוד (סגנון, יעילות)
 - מאמץ זמן השקעה
 - הכישורים להם נדרשתם בפיתוח
 - מבני הנתונים בשימוש
 - הבנה של התוצר ושליטה בו
 - תקלות/בעיות
 - תרומה ללמידה

- סיפוק/תסכול
- מהלך העבודה (בקצרה)
- שלושה דברים שלמדתם מהמימוש העצמי.
- שלושה דברים שלמדתם מהמימוש עם בינה.
- שלושה דברים שלמדתם מהמימוש עם Base44.
- רשימה של כל המקורות בהם נעזרתם בכתיבת התוכנית.
- רשימה של הבעיות הידועות בתוכנית (אם קיימות).