**תרגיל # 3 Courses Service**

בתרגיל זה יהיה עליכם לממש את שירות ה Courses של הכולל מידע על קורסים ועל ציוני תלמידים המשתתפים בהם.

עליכם לתכנן בניית  שרות REST  הנקרא CoursesService המאחסן ב’ענן’ רשימות קורסים, רשימת הסטודנטים הלומדים בקורס בליווי הציון שנקבע לסטודנט בקורס זה.

השירות מאפשר למנהל המערכת לראות את רשימת הקורסים וגם לראות את רשימת הסטודנטים עם הציונים עבור קורס מסוים.

המידע המנוהל:

* לכל קורס נשמרים הפרטים הבאים: מזהה הקורס (יכול להכיל אותיות A-Z ומספרים), שם הקורס, שם המרצה, רשימת הסטודנטים המשתתפים בקורס בליווי הציון שכל סטודנט.ית קיבלו, רשימת קורסי קדם, תאריך תחילת הקורס, תאריך סיום הקורס.
* לכל סטודנט.ית נשמרים הפרטים הבאים: תז סטודנט (מזהה הסטודנט), שם פרטי ושם משפחה, תמונת פרופיל, וכמובן הציון שלו בקורס

המידע ייכתב לקובץ json. קובץ לדוגמא המכיל את המבנה מצורף לתרגיל.

כל שירות אחסון מאפשר לבצע על האובייקט המאוחסן את קבוצת הפעולות CRUD המורכבת מ-Create, Read, Update, Delete .

שירות זה יאפשר למשתמש את פעולות ה-CRUD המפורטות בהמשך

**ממשו בצד שרת ה-NodeJS, באמצעות express, את הפונקציות הבאות. לכל פונקציה קיים ניתוב מתאים עם http method מתאים המפנה לפונקציה:**

1. CreateCourse (CourseDetails)

פונקציה זו מקבלת את פרטי הקורס: מזהה הקורס (יכול להכיל אותיות A-Z ומספרים), שם הקורס, שם המרצה , רשימת קורסי קודם, תאריך תחילת הקורס, תאריך סיום הקורס.

כל השדות הם שדות חובה. הפונקציה תחזיר סטטוס מתאים במקרה הצלחה או כשלון.

1. updateCourse (string Course\_id, CourseDetails)

פונקציה זו תעדכן את השדות שעבורם נשלח מידע חדש עבור Course\_id. הפונקציה תחזיר הודעה אם Course\_id אינו קיים.

כל השדות יכולים להתעדכן חוץ ממזהה הקורס הייחודי ורשימת הסטודנטים.

1. AddStudentToCourse (string Course\_id, StuedntDetails )

פרטי הסטודנט הינם תז סטודנט (מזהה הסטודנט), שם פרטי ושם משפחה, תמונת פרופיל, וכמובן הציון שלו בקורס. פונקציה זו בודקת אם התלמיד כבר קיים ברשימה, ואם לא קיים מוסיפה אותו.

הפונקציה תחזיר הודעה אם Course\_id אינו קיים או במקרים אחרים שבהם הקלט אינו מתאים.

1. getCourse (Course\_id)

פונקציה זו מחזירה את פרטי הקורס Course\_id כולל כל המידע השמור לגביו

1. getCourses ()

פונקציה זו מציגה את רשימת הקורסים.

1. deleteStudentFromCourse (string Course\_id, string Student\_id)

פונקציה זו תמחק את הסטודנט מרשימת הסטודנטים של הקורס הנתון, הפונקציה תחזיר הודעה אם Course\_id אינו קיים או במקרים אחרים שבהם הקלט אינו מתאים

1. deleteCourse( string Course\_id)

פונקציה זו תמחק את כל המידע עבור Course\_id. הפונקציה תחזיר הודעה אם Course\_id אינו קיים.

* בתרגיל זה השרת שומר את המידע בקובץ JSON בפורמט של הקובץ המצורף לדוגמא.

**ממשו בצד לקוח באמצעות jquery +ajax**

דף /list המציג את רשימת הקורסים ברשימה או בטבלה.

עבור כל קורס ברשימה מוצגים הפרטים שלו:

מזהה הקורס (יכול להכיל אותיות A-Z ומספרים), שם הקורס, שם המרצה, רשימת הסטודנטים המשתתפים בקורס בליווי הציון שכל סטודנט.ית קיבלו, רשימת קורסי קדם, תאריך תחילת הקורס, תאריך סיום הקורס.

ליד כל קורס יש כפתורים עבור הפעולות הבאות:

1. מחיקת הקורס
2. עריכת פרטי הקורס האפשריים לעריכה
3. הוספת סטודנט לרשימת הסטודנטים
4. לצפייה ברשימת הסטודנטים. שבה, ליד כל סטודנט יש כפתור למחיקה. (הרשימה תיפתח בחלונית או בהרחבה של הרשימה וסגירה שלה)

מעל ומתחת לרשימה, יש כפתורים הפותחים טופס של "הוסף קורס".

טופס יכול להיפתח כחלונית מעל הרשימה או להוביל לניתוב נפרד שיציג את הטופס. בסיום פעולת הקורסה/עדכון תקינה, המשתמש ינותב חזרה לרשימת הקורסים ויראה אותה עדכנית.

הטופס יאפשר הזנת הפרטים הנדרשים כפי שתוארו לעיל בסעיף 1 בפונקציית יצירת קורס.

הערות למימוש והגשה:

1. השירות ירוץ בפורט 3001
2. יש להקפיד על בדיקות שלמות ותקינות קלט בצד לקוח ובצד שרת.
3. יש להקפיד על כתיבת קוד נקי, מסודר, מחולק לתיקיות.
4. יש לתעד כל פונקציה: שם, תיאור קצר של הפונקציה , תיאור פרמטרים וערך החזרה. בנוסף להוסיף תיעוד קצר בגוף הקוד עבור קטעי קוד הדורשים הסבר.
5. כל הניתובים בצד שרת צריכים להיות לפי עקרונות REST שנלמדו
6. כל קריאות הAJAX של הקליינט צריכות להיות עם JQUERY.
7. יש להגיש ZIP עם כל הקבצים. אין צורך לצרף את תיקיית node\_modules. (שימו לב שכל ספריה שאתם משתמשים בה ואינה מובנית בnodeJS צריכה להיות כתבה בקובץ package.json באופן שיאפשר התקנה מתאימה)
8. מומלץ להוסיף את פקודת ההפעלה של השרת, בקובץ package.json כך שניתן יהיה להריץ את הקוד באמצעות npm start