**1. User (משתמש)**

* **Id** (Primary Key)
* **Username** (Unique)
* **Email**
* **PasswordHash**
* **CreatedAt** (תאריך יצירה)
* **UpdatedAt** (תאריך עדכון אחרון)
* **Role** (למשל: "Admin", "Developer", "Viewer" – תוכל להשתמש בזה לניהול הרשאות גישה)

**2. File (קובץ קוד)**

* **Id** (Primary Key)
* **FileName** (שם הקובץ)
* **FilePath** (מיקום הקובץ במערכת או בענן)
* **FileType** (למשל: "C#", "Java", "Python" – לסיווג הקובץ)
* **UploadDate** (תאריך העלאה)
* **Size** (גודל הקובץ)
* **UserId** (Foreign Key, מצביע על המשתמש שהעלה את הקובץ)
* **AnalysisResult** (תוצאה של ניתוח הקוד – אפשר לשמור תוצאות ניתוח קוד כמו אזהרות או תיקונים מומלצים)

**3. CodeAnalysis (ניתוח קוד)**

* **Id** (Primary Key)
* **FileId** (Foreign Key, מצביע על הקובץ המתואם)
* **AnalysisDate** (תאריך ביצוע הניתוח)
* **Result** (תוצאה מפורטת של הניתוח – אפשר לשמור שם את האזעקות, תיקונים מומלצים, וכו')
* **Status** (סטטוס הניתוח, לדוגמה: "Completed", "InProgress", "Failed")

**4. UserFileAccess (הרשאות גישה לקובץ)**

* **Id** (Primary Key)
* **UserId** (Foreign Key, מצביע על המשתמש)
* **FileId** (Foreign Key, מצביע על הקובץ)
* **AccessLevel** (למשל: "Read", "Write", "Delete" – מה שהמשתמש יכול לעשות עם הקובץ)

**5. AuditLog (יומן אודיט)**

* **Id** (Primary Key)
* **UserId** (Foreign Key)
* **Action** (הפעולה שבוצעה, למשל: "Upload", "Delete", "Edit")
* **ActionDate** (תאריך הפעולה)
* **Details** (פרטים נוספים על הפעולה – לדוגמה, איזה קובץ שונה או הועלה)

### 1. ****User (משתמש)****

המחלקה הזו מייצגת את המשתמשים במערכת, עם פרטים בסיסיים עליהם כמו שם משתמש, מייל, סיסמה ועוד.

#### קשרים:

* **User -> File**: משתמש יכול להעלות מספר קבצים. כלומר, כל קובץ יכול להיות שייך למשתמש אחד בלבד, אבל משתמש יכול להעלות כמה קבצים. קשר זה ייוצג על ידי UserId בטבלת **File** כ-**Foreign Key**.
* **User -> UserFileAccess**: כל משתמש יכול לקבל גישה לקבצים עם רמות הרשאה שונות. קשר זה מאפשר לנו לנהל מי יכול לקרוא, לערוך או למחוק קובץ. קשר זה ייוצג על ידי UserId בטבלת **UserFileAccess** כ-**Foreign Key**.
* **User -> AuditLog**: כל פעולה שהמשתמש מבצע תתועד ביומן אודיט. קשר זה ייוצג על ידי UserId בטבלת **AuditLog** כ-**Foreign Key**.

### 2. ****File (קובץ קוד)****

המחלקה הזו מייצגת קבצים שהמשתמשים מעלים למערכת.

#### קשרים:

* **File -> User**: כל קובץ שייך למשתמש אחד בלבד, כך שהקשר יהיה **Foreign Key** ב-**File** (למשל: UserId).
* **File -> CodeAnalysis**: כל קובץ יכול להיבדק וניתח מספר פעמים. כלומר, קובץ יכול להיות קשור למספר ניתוחים. קשר זה ייוצג ב-**Foreign Key** בתוך טבלת **CodeAnalysis** (למשל: FileId).
* **File -> UserFileAccess**: קובץ יכול להיות נגיש למספר משתמשים, כאשר לכל משתמש יש רמת גישה שונה. קשר זה ייוצג ב-**Foreign Key** בטבלת **UserFileAccess** (למשל: FileId).

### 3. ****CodeAnalysis (ניתוח קוד)****

המחלקה הזו מייצגת ניתוחים שמבוצעים על הקבצים. כל ניתוח מכיל תוצאה וסטטוס.

#### קשרים:

* **CodeAnalysis -> File**: כל ניתוח מתבצע על קובץ ספציפי. הקשר הזה יתבצע על ידי FileId כ-**Foreign Key** בתוך **CodeAnalysis**.
* **CodeAnalysis -> User**: אפשר להוסיף קשר בין ניתוח לבין המשתמש שביצע אותו אם תרצי, אבל זה תלוי אם את רוצה לעקוב אחרי מי ביצע את הניתוח (למשל, אם יש ניתוחים אוטומטיים שמבוצעים בשרת).

### 4. ****UserFileAccess (הרשאות גישה לקובץ)****

המחלקה הזו מייצגת את הגישות שהמשתמשים יכולים לקבל על הקבצים (לקרוא, לכתוב, למחוק וכו').

#### קשרים:

* **UserFileAccess -> User**: כל הרשאת גישה מקושרת למשתמש ספציפי. קשר זה ייוצג ב-**Foreign Key** בטבלת **UserFileAccess** (למשל: UserId).
* **UserFileAccess -> File**: כל הרשאת גישה מקושרת לקובץ ספציפי. קשר זה ייוצג ב-**Foreign Key** בטבלת **UserFileAccess** (למשל: FileId).

### 5. ****AuditLog (יומן אודיט)****

המחלקה הזו מתעדת את הפעולות שהמשתמשים מבצעים במערכת (העלאת קובץ, עדכון קובץ, מחיקת קובץ וכו').

#### קשרים:

* **AuditLog -> User**: כל רשומה ביומן האודיט מקושרת למשתמש שביצע את הפעולה. קשר זה יתבצע על ידי UserId כ-**Foreign Key** בטבלת **AuditLog**.
* **AuditLog -> File**: אפשר להוסיף קשר בין פעולה על קובץ לבין הקובץ עצמו, אם תרצי לתעד איזה קובץ שונה או הועלה. קשר זה ייוצג על ידי **Foreign Key** בטבלת **AuditLog** (למשל: FileId).

### איך זה יעבוד מבחינת EF?

1. **הגדרת קשרים:** כל הקשרים בין המחלקות יוגדרו באמצעות תכונה navigation property. לדוגמה:
   * ב-**User** תהיה תכונה של קובץ: public ICollection<File> Files { get; set; }
   * ב-**File** תהיה תכונה של משתמש: public User User { get; set; }
   * ב-**CodeAnalysis** תהיה תכונה של קובץ: public File File { get; set; }
2. **מיגרציות:** לאחר שתגדרי את המודלים, תוכל להשתמש ב-**Entity Framework Migrations** כדי ליצור את מסד הנתונים בפועל עם הטבלאות שהגדרת.
3. **עבודת CRUD:** תוכל ליצור, לקרוא, לעדכן ולמחוק רשומות באמצעות ה-**DbContext** של EF. לדוגמה, כדי להוסיף קובץ, פשוט תיצור אובייקט **File** ותוסיף אותו ל-**DbContext**, ואז תבצע שמירה (SaveChanges).