**מסמך אפיון מערכת לניהול שירים ופלייליסטים.**

1. תיאור המערכת

1.1 מטרת הפרויקט

המערכת מאפשרת למשתמשים לנהל שירים ופלייליסטים בצורה נוחה, שיתופית ומתקדמת, עם אפשרויות האזנה, שיתוף, ותמלול שירים באמצעות AI. כל משתמש יכול ליצור פלייליסטים אישיים, לשתף אותם עם אחרים, ולהגיש שירים חדשים למאגר הציבורי. מנהל המערכת מנהל את האישור של השירים הציבוריים, עוקב אחרי פעילות המשתמשים, ומאפשר שליטה מלאה בהרשאות ובתוכן.

תכונות עיקריות:

* ניהול פלייליסטים אישיים ומשותפים
* מאגר שירים ציבורי
* שליחת בקשות להוספת שירים למאגר הציבורי
* ממשק מנהל לאישור/דחיית בקשות
* שליחת מיילים עם עדכון סטטוס בקשה
* תמלול שירים באמצעות AI

1.2 קהל היעד

האפליקציה מיועדת למשתמשים פרטיים שרוצים לנהל שירים ופלייליסטים באופן אישי או משותף, ולמנהל מערכת שמאשר את העלאות השירים למאגר הציבורי. הקהל כולל:

* משתמשים פרטיים שמעוניינים לנהל ולהשמיע שירים ופלייליסטים אישיים או משותפים
* יוצרים המעוניינים לשתף שירים עם הקהל הרחב ולהציע תכנים חדשים
* ארגונים או קבוצות המעוניינים בשיתוף תוכן מוזיקלי פנימי

1.3 פונקציונליות של המערכת

* ניהול משתמשים עם הרשאות(JWT + Roles)
* התחברות באמצעות חשבון Google
* ניהול פלייליסטים פרטיים ומשותפים
* שיתוף פלייליסטים באמצעות קישור ייחודי ומאובטח
* האזנה לשירים דרך נגן מובנה
* הוספת שירים לפלייליסטים
* בקשות משתמש להוספת שירים למאגר הציבורי
* אישור/דחיית בקשות על ידי מנהל ושליחת מייל למשתמש לגבי סטטוס הבקשה
* תמלול שיר דרך מודל AI (ע"פ שפת השיר)

1.4 בעיות שהמערכת פותרת

* ניהול ריכוזי של שירים במקום אחסון מבולגן
* שיתוף פשוט ומאובטח של פלייליסטים ללא צורך בהרשמה של המקבל
* סינון תוכן ציבורי בעזרת אישור מנהל
* אפשרות לשיתוף פעולה על רשימות שירים
* קושי בגישה לתמלולים של שירים

1.5 טכנולוגיות בשימוש

Backend: ASP.NET Core 9 לפיתוח API מאובטח ומודולרי

JWT לניהול גישה והרשאות

Frontend:Angular -ממשק ניהול

React -משתמשים רגילים

Database: MySQL

אחסון קבצים: Amazon S3

שליחת מיילים: MailKit

AI לתמלול שירים: שרת pythonהמשתמש ב OpenAI Whisper ו־ GPT להפקת מילות שירים

OAuth2 להתחברות עם google

1.6 מתודולוגיית פיתוח (Agile)

הפיתוח יתבצע בשיטת Agile, עם 4 ספרינטים של שבועיים.

ספרינט 1 – תשתיות ונתונים

* הקמת בסיס נתונים
* בניית תשתית API
* ניהול משתמשים והרשאות
* שליחת מייל בסיסית

ספרינט 2 – ניהול שירים ופלייליסטים

* CRUD לפלייליסטים
* הוספה/הסרה של שירים
* נגן שירים

ספרינט 3 – בקשות וניהול

* שליחת בקשה להוספת שיר
* ממשק מנהל לאישור ודחייה
* שליחת מייל על סטטוס בקשה

ספרינט 4 – תמלול שירים ושיפורים

* אינטגרציה עם מודל תמלול שירים AI
* שיפורי UI/UXבדיקות והרצת גרסה

2. עיצוב ממשק משתמש ויזואלי

* מסך התחברות: טופס התחברות פשוט עם קישור להרשמה
* מסך שירים ציבוריים: טבלת שירים, חיפוש,סינון, האזנה, כפתור הוספה לפלייליסט
* מסך הפלייליסטים שלי: טאב "בבעלותי" ו־"משותפים איתי", כפתורי עריכה ושיתוף
* דיאלוג העלאת שיר: בחירת קובץ, שם, תיאור ושינוי פרטי המטא דטה.
* אפליקצית ניהול (Admin): אישור שירים, סטטיסטיקות, ניהול משתמשים

3. איפיון תמציתי

3.1 רשימת פונקציות המערכת

הוספת פלייליסט חדש

Route: POST /api/playlists

Body: { name: string }

Response: { playlistId: string }

לוגיקה: שם ייחודי למשתמש, נוצר עם תאריך

הוספת שיר לפלייליסט

Route: POST /api/playlists/{playlistId}/songs

Body: { songId: string }

Response: 204

לוגיקה: בדיקת בעלות על הפלייליסט

שיתוף פלייליסט  
Route: GET /api/playlists/{playlistId}/share-link  
Response: { link: string }  
לוגיקה:

המערכת מייצרת קישור מוצפן המכיל מזהה פלייליסט חתום דיגיטלית

ניתן לשתף גם משתמש שאינו רשום למערכת

שליחת בקשה להוספת שיר ציבורי

Route: POST /api/song-requests

Body: { title: string, artist: string, link: string }

Response: { requestId: string }

לוגיקה: כל בקשה נשמרת בטבלת בקשות וממתינה לאישור מנהל

אישור בקשה ע"י מנהל

Route: POST /api/song-requests/{id}/approve

Response: 200

לוגיקה: יצירת שיר חדש בטבלת Songs, שליחת מייל למבקש

דחיית בקשה ע"י מנהל

Route: POST /api/song-requests/{id}/reject

Response: 200

לוגיקה: שליחת מייל עם סיבת דחייה

תמלול שיר

Route: POST /api/songs/{id}/transcribe

Response: { lyrics: string }

לוגיקה: שימוש ב־Whisper + GPT לקבלת תמלול לפי קובץ השיר

3.2 אימות והרשאות

מערכת משתמשים מבוססת JWT

תפקידים:

* User: גישה לפלייליסטים ובקשות
* Admin: אישור בקשות, ניהול שירים, שליחה במייל

קישורים שיתופיים מבוססי חתימה דיגיטלית כדי למנוע זיופים ולוודא אמינות

3.3 אפליקציית ניהול

ניהול משתמשים: הצגה, מחיקה, שינוי תפקיד

ניהול מאגר שירים ציבורי: הצגת שירים, עריכה, מחיקת שירים

ניהול בקשות משתמשים להעלאת שירים שירים: אישור ודחייה, צפייה בנתונים

צפייה בסטטיסטיקות מערכת: כמות משתמשים, שירים מאושרים, שירים ממתינים

4. מבנה הטבלאות

Users

| **שדה** | **טיפוס** | **תיאור** |
| --- | --- | --- |
| Id | GUID | מזהה ייחודי למשתמש |
| Email | string | אימייל ייחודי |
| PasswordHash | string | סיסמה מוצפנת |
| Role | enum | User / Admin |
| CreatedAt | datetime | תאריך הרשמה |

Songs

| **שדה** | **טיפוס** | **תיאור** |
| --- | --- | --- |
| Id | GUID | מזהה שיר ייחודי |
| Title | string | שם השיר |
| Artist | string | שם האמן |
| FileUrl | string | קישור לקובץ ב־S3 |
| Lyrics | text | תמלול (אם קיים) |
| IsPublic | bool | האם שיר מאושר לציבור |
| CreatedAt | datetime | תאריך הוספה |

Playlists

| **שדה** | **טיפוס** | **תיאור** |
| --- | --- | --- |
| Id | GUID | מזהה פלייליסט |
| Name | string | שם הפלייליסט |
| OwnerId | GUID | מקשר לטבלת Users |
| CreatedAt | datetime | תאריך יצירה |

PlaylistSongs

| **שדה** | **טיפוס** | **תיאור** |
| --- | --- | --- |
| Id | GUID | מזהה רשומה |
| PlaylistId | GUID | קישור לפלייליסט |
| SongId | GUID | קישור לשיר |

SongRequests

| **שדה** | **טיפוס** | **תיאור** |
| --- | --- | --- |
| Id | GUID | מזהה בקשה |
| Title | string | שם השיר המבוקש |
| Artist | string | שם האמן |
| Link | string | קישור לקובץ |
| UserId | GUID | מי הגיש את הבקשה |
| Status | enum | Pending / Approved / Rejected |
| Reason | string | סיבת דחייה (אם רלוונטי) |
| CreatedAt | datetime | מתי הבקשה נשלחה |

5. הגדרת סבבי פיתוח (ספרינטים)

ספרינט 1 – תשתיות ובסיס:

* הקמת פרויקט - NET הגדרת בסיס נתונים וטבלאות
* הטמעת + JWT ניהול הרשאות
* התחברות ל S3-והעלאה ראשונית

ספרינט 2 – ממשק משתמש ראשוני:

* דפי התחברות/הרשמה
* עמוד שירים ציבוריים + נגן - תצוגת טבלת שירים

ספרינט 3 – פלייליסטים ושיתוף:

* יצירה וניהול פלייליסטים
* שיתוף פלייליסט
* הוספת שיר לפלייליסט

ספרינט 4 – מערכת ניהול:

* שליחת בקשות להעלאת שירים ציבוריים
* מסך אישור מנהל
* ניהול משתמשים וסטטיסטיקות

ספרינט 5 – סיום והטמעה:

* טיפול בבאגים
* שיפורי UIו־ UX
* פריסה לשרת