

Full Stack Web Developer פרויקט JQUERY -AJAX



בפרויקט זה עליכם לפתח דף יחיד, המגיש רשימות השמעה (playlists). כל המידע ישמר בשרת RESTful אותו תתקינו, יש ליישם הוספה, עריכה והסרה של כל הנתונים. השירים יהיו URL לקובץ MP3 חיצוני שינוגנו עם גן HTML5.

בפרויקט הזה יש ליישם את הנושאים הבאים:

- HTML + CSS
 - New HTML5 tags
 - CSS3 media queries and advanced selectors
 - Dynamic page layouts
 - Bootstrap & Font Awesome
- JavaScript
 - Objects
 - Function Closures
 - jQuery
 - **Single Page Application** foundations
 - Events
 - Ajax (RESTful API)
 - Documentation

הגשת הפרויקט: חודש וחצי ממפגש ההנחיה

פירוט הפרויקט

העמוד מציג רשימות השמעה בצורה של דיסקים. לכל רשימת השמעה יש שם ותמונת רקע אותם ניתן יהיה לשנות.

התמונה היא URL חיצוני לקובץ תמונה כלשהו.

לכל רשימת השמעה יש רשימת שירים אותן ניתן יהיה לשנות. הרשימה היא בעצם מערך אובייקטים שכל אחד מכיל URL חיצוני לקובץ MP3 ושם השיר.

בלחיצה על רשימת ההשמעה ינוגנו השירים שלה באמצעות גן פשוט HTML5 ותמונת הרקע תסתובב כמו תקליט. בסיום שיר ינוגן השיר הבא.

ניתן גם לחפש רשימות השמעה לפי שם, דרך תיבת חיפוש. כאשר בתיבת הטקסט יש יותר משתי אותיות, יש להראות רשימות השמעה בעל שם מתאים בלבד.

כל הנתונים נמצאים על שרת Restful אליו יש לגשת למשך ולערוך נתונים באמצעות API. תיעוד מלא שלו מופיע בהמשך.

האייקונים המתוארים אינם תמונות אלא תווים, המוגשים ע"י ספריית Bootstrap. לחילופין, ניתן להשתמש ב-Font Awesome.

יש להשתמש בספריית jQuery ככל שניתן.

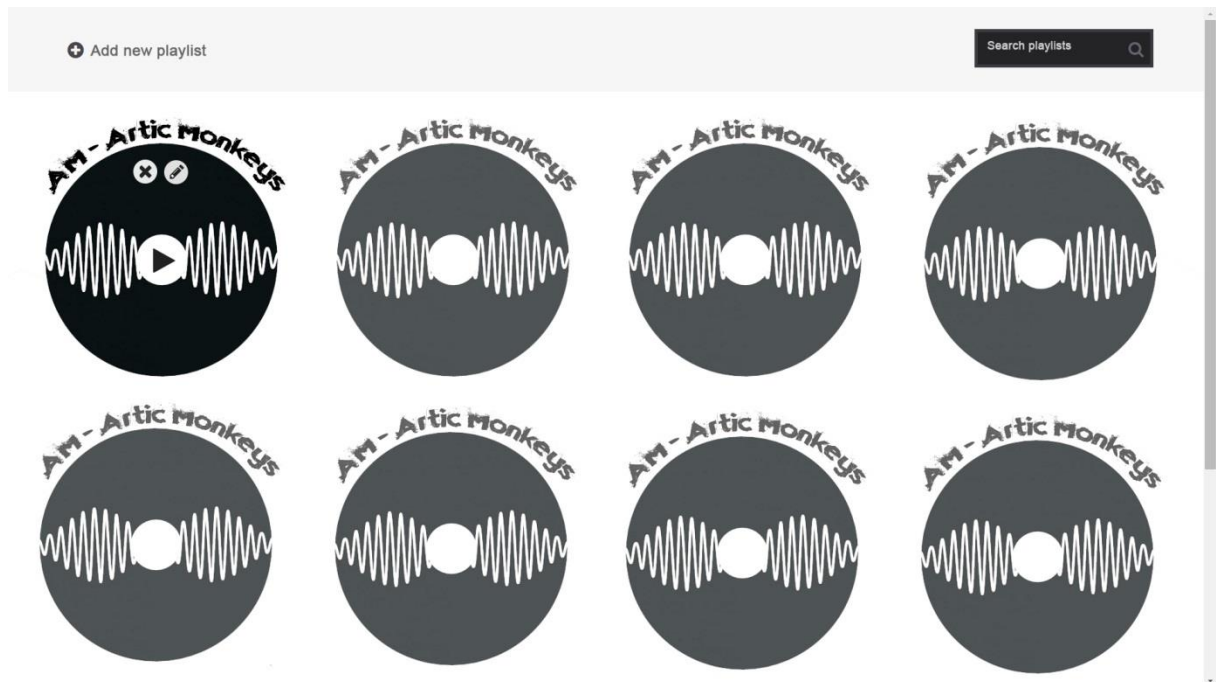
יש לבצע ולידציה לכל תיבות הקלטים מהמשתמש באמצעות Regex.

מסכים

אפיון מסכים גרפי (mockups) נמצא בתקיית Docs

מסך ראשי

מסך הכניסה הוא בעצם תצוגה של כל רשימות ההשמעה הקיימות במערכת.



בחלק העליון קיים תפריט, המכיל כפתור הוספה של רשימת השמעה חדשה ותיבת חיפוש, המאפשרת לסנן רשימות לפי שם. החלק העליון מקובע (fixed), כך שגלילת העמוד לא תשפיע עליו. רשימות ההשמעה מוצגות עם השם מעליהם (יש למצוא פתרון לכתיבת השם בקשת יפה, ע"י שימוש ב-CSS3 /או Jquery Plugin) והתמונה המקושרת תוצג כעיגול (border-radius CSS3).

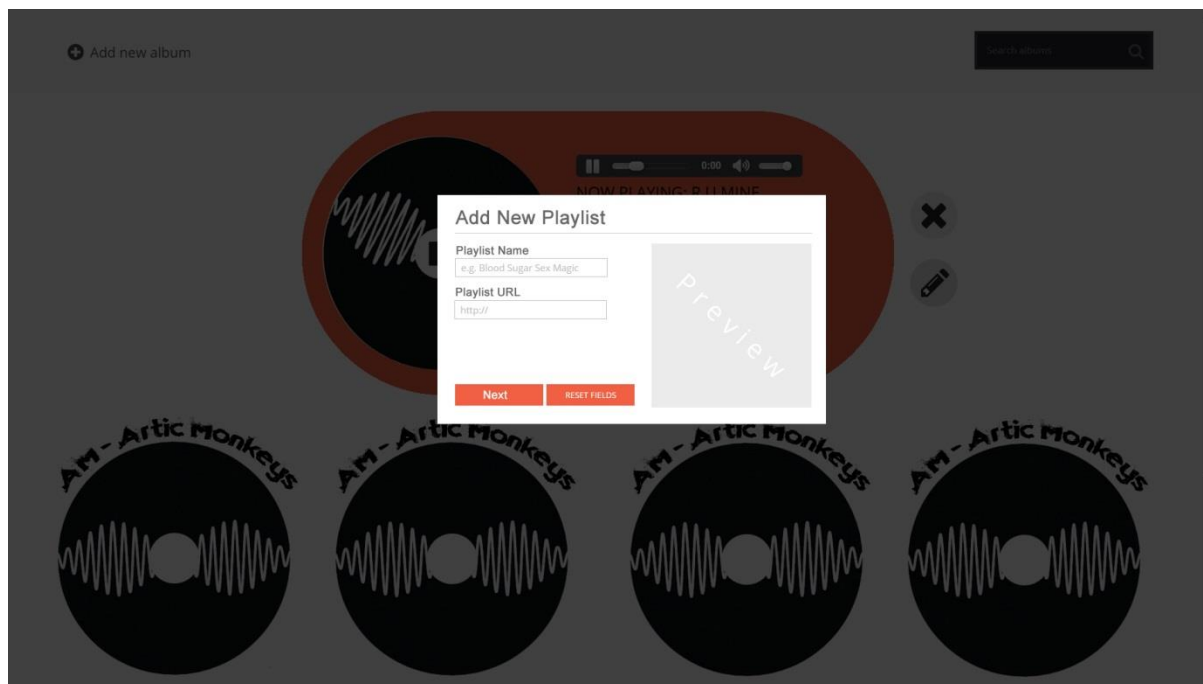
כל האלבומים שקופים בשיעור של 0.85 ובעת מעבר העכבר עליהם יקרו הדברים הבאים: השקיפות תוסר והתמונה תהיה מלאה, במרכז יופיע כפתור "play" בחלק העליון יופיע כפתורי מחיקה ועריכה. השינוי הנ"ל לא יבוצע באמצעות JavaScript אלא CSS. כמו כן, יש ליצור את הכפתורים באמצעות שימוש בספריה חיצונית ולא image.

בלחיצה על מחיקה תעלה השאלה "Are you Sure??" בתוך חלון מודל (modal window), כפי ששאר החלונות יתוארו בהמשך, ובאישור תתבצע המחיקה.

המסכים הבאים מתארים את פעולות ניגון, עריכה והוספה.

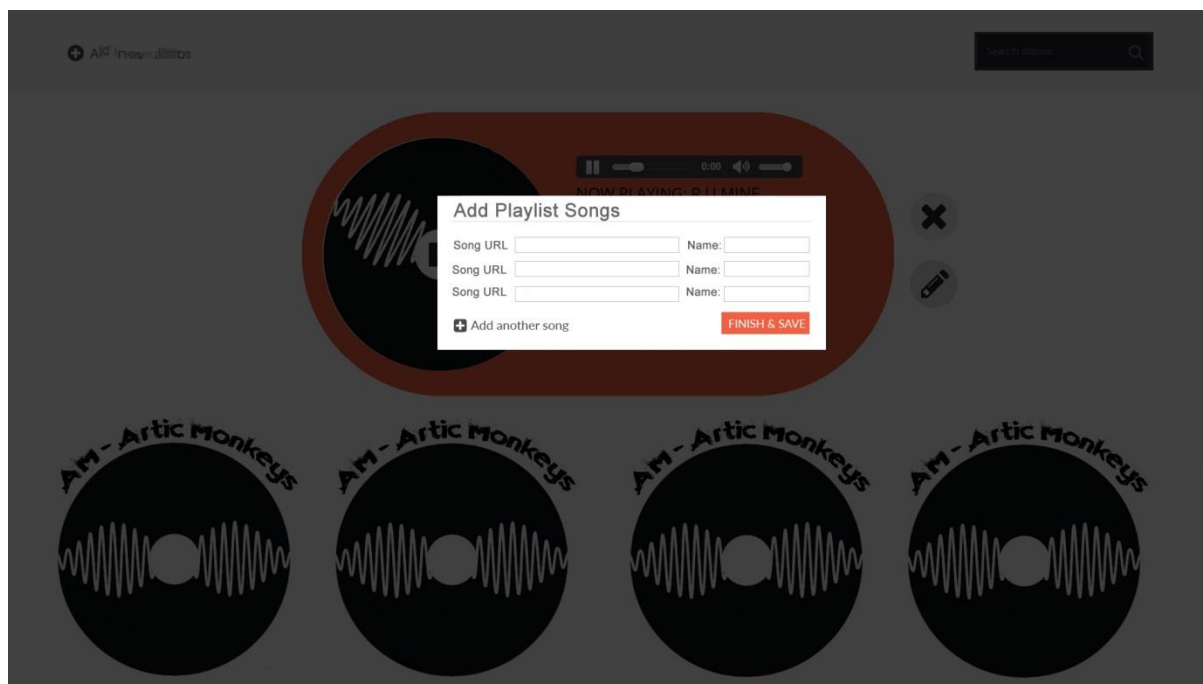
מסך הוספת ועריכה של רשימת השמעה

במסך זה תתאפשר הוספת אלבום חדש לאוסף האלבומים ועריכתו. הכותרות מראות "add", אך בעריכה יש לטעון את כל הפרטים הקיימים ולהודיע בהתאם (כותרת edit, כפתור update וכו').



המסך כולו מוחשך (שימוש בposition וב-opacity modal window או של ספריה חיצונית) ובאמצע תופיע תיבת דיאלוג המכילה מספר שדות: שם האלבום, ותמונת cover (לינק חיצונית). לפרטים אלה יש לבצע ולידציה באמצעות Regex. מיד לאחר הזנת כתובת התמונה (כאשר התבנית נכונה), יש להציג את התמונה במקום המיועד לה (היכן שכתוב Preview).

בלחיצה על "Next", אם כל הנתונים תקינים, תתחלף התיבה, ותוצג תיבה בה ניתן להוסיף את רשימת השירים לאלבום.



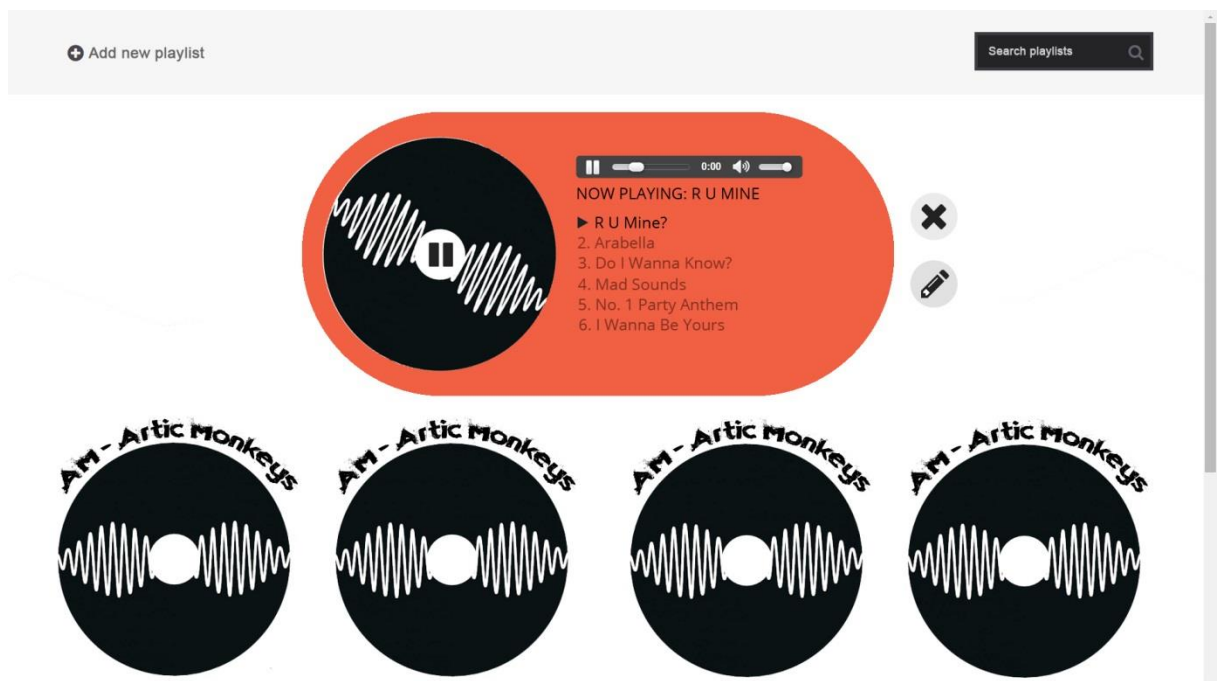
כל שורה מייצגת פרטי שיר: כתובת חיצונית של קובץ MP3 ו שם השיר. כתובת השיר חייבת להסתיים ב-
 ".mp3". על כן, יש לבצע בדיקה באמצעות ביטוי רגולרי, שהכתובת היא אכן URL של MP3. אם הכתובת לא
 נכונה, אזי ברגע שהמשתמש יצא מהשדה, הוא יצבע באדום (או יכיל אייקון אזהרה או כל דבר אחר שתבחרו)
 ולא ניתן יהיה לשמור את האלבום. לחיצה על האייקון של הפלוס, תאפשר להוסיף שדות נוספים, כדי להוסיף
 עוד שירים לפי צורך המשתמש.

*** רשות: ניתן לבצע בדיקת Ajax על כתובת ה-URL של הקובץ ולוודא שה-Header המכיל את ה-Content-Type, מציג קובץ סאונד (audio/mpeg).

בלחיצה על "Save & Finish", האלבום ישמר בשרת ולאחר אישור מהשרת וקבלת ID יתווסף לתוך ה-DOM.
 שים לב: במצב עריכה יש לשלוח עדכונים לשרת, לאחר כל שלב, באמצעות הקריאה הנכונה, לעומת מצב
 יצירת רשימה חדשה שם יש לשלוח את כל הנתונים בקריאה אחת.

מסך נגן

בלחיצה על כפתור "Play" במסך האלבומים, יפתח נגן בחלקו העליון, שיתחיל לנגן את השיר הראשון. הנגן
 גם מקובע למקומו כך שגלילה של העמוד לא תשפיע עליו. מלבד זאת, גם הכותרת של העמוד תשתנה (הכוונה
 היא לתגית <title> המופיעה ב-<head>), ותציג את שם רשימת ההשמעה ואת השיר שמתנגן כעת.



הנגן מציג את התמונה שנבחרה לרשימת ההשמעה בצג שמאל. כששיר מנגן, התמונה תסתובב כמו תקליט
 (שימוש ב-CSS3 באמצעות Keyframes ו-Transform). במרכז יתחלף האייקון, באייקון של "Pause". מצד
 ימין לדיוקן תופיע רשימת השירים של הדיסק והשיר הראשון יחל להתנגן.

הנגן הוא נגן פשוט ביותר, שימוש בתגית <audio> באמצעות HTML5 מאפשרת לנגן שיר Mp3, כל שיש
 לעשות הוא להחליף את המאפיין src (Attribute) של התגית, יש ללמוד כיצד להגיב לאירוע סיום שיר כדי
 להחליף לשיר הבא.

כל שמות השירים יהיו שקופים (opacity כברירת מחדל), למעט השיר המתנגן כרגע. במעבר על אחד השירים
 יוצג לצידו אייקון של "Play". בלחיצה עליו השיר הנבחר יתנגן ויעבור להיות בולט ולא שקוף. במעבר על השיר

המתנגן כפתור ה-"Play" יוחלף בכפתור "Pause". שם השיר המתנגן יופיע מתחת לנגן. לצד הנגן יופיעו שני אייקונים, אייקון עריכה ואייקון מחיקה. לחיצה עליהם יראו את מסכי עריכה ומחיקה כפי שתוארו, והנגן יסקף את השינוי, אם בוצע.

התקנה

כדי להריץ את הפרוייקט, יש להתקין סביבת שתתמוך ב PHP ו MySQL כגון XAMPP או WAMP. זהו מבנה תיקיית הפרוייקט:

```
playlist/
  api/
    .htaccess
    playlist.php
  Docs/
    instructions.pdf
    playlistDB.sql
    mockup screens
    index.html
```

את תיקיית הפרוייקט playlist יש למקם ישירות תחת תיקיית ה root, לדוגמא htdocs. יש לייצר סכמת DB חדשה ע"י שימוש בקובץ playlistDB.sql שנמצא בתקיית ה Docs. העמוד הראשי של הפרוייקט יהיה index.html הנמצא ישירות תחת תיקיית playlist.

API Docs

Endpoint: **http://localhost/playlist/api/**

Description	Method + Path	Request / Response
Get all playlists	GET /playlist	Request: empty Response: { "success": true, "data": [{ "id": integer, "name": string, "image": string }, . . .] }

Get existing playlist	GET /playlist/{id}:integer	Request: empty Response: { "success": true, "data": { "id": integer "name": string, "image": string } } <i>*"data":false if no such id</i>
Get playlist songs	GET /playlist/{id}:integer/songs	Request: empty Response: { "success": true, "data": { "songs": [{ "name": string, "url": string }, ...] } } <i>*"songs":[] if no such id or no songs</i>
Create new playlists	POST /playlist	<i>* All request fields required, minimum one song in songs array, both name & url per song required</i> Request: { "name": string, "image": string, "songs": [{ "name": string, "url": string }, ...] } Response: { "success": true, "data": { "id": integer } } <i>*returning newly created id</i>
Update existing playlist info	POST /playlist/{id}:integer	<i>* Request requires at least one field: name and/or image</i> Request: { "name": string, "image": string } Response: { "success": true }

Update playlist songs	POST /playlist/{id}:integer/songs	<p><i>* Request requires at least one song with both name & url</i></p> <p>Request: { "songs": [{ "name": string, "url": string }, . . .] } Response: { "success": true }</p>
Delete existing playlist	DELETE /playlist/{id}:integer	<p>Request: empty Response: { "success": true }</p>

Response header codes:

- 200: Success
- 400: Bad request
- 404: Bad URL / Page not found
- 422: DB Error

לינקים

- אייקונים לכפתורים מ Font Awesome
[/http://fontawesome.io/icons](http://fontawesome.io/icons)
- תגית AUDIO של HTML5
http://www.w3schools.com/tags/ref_a

ב-ה-צ-ל-ח-ה