**Analysis and Design Document for *musicOnline.com***

**Produced by Renat Oosthuizen**

**09/02/2021**

**Contents**

1. Introduction and Overview
2. Requirements Document
   1. Functional Requirements
   2. Non-Functional Requirements
3. Design Document
   1. Site Structure
   2. Wireframes
   3. Page Content Summaries
   4. Data Source Design

**Introduction and Overview**

We have been hired to create a prototype website for a business called musicOnline.com. It will ultimately allow users to sign up and sell or buy vinyl music disks, though at this stage no sales capability is needed. New users can sign up by entering an email, password, phone number and their date of birth which will be used to verify that the user is over 18 years of age. Additionally, they will be prompted to pick between a regular user account and a retailer user account. Picking a retailer account will prompt the user to make a payment.

Users can search for vinyls based on artists and titles. Clicking on a vinyl entry will generate a page with the complete information of the vinyl including an image, title, name of artist, publisher, price, release date, run time and description. This page will feature a ‘Buy’ button to simulate a purchase. Users can view, create edit and delete vinyls that they are selling.

Another type of user exists, called an admin user. Admin users have the capability to view, edit or delete any user or vinyl information. The website will have available free space for future advertisement, a clean, user friendly interface and will dynamically adjust content to fit screen space.

**Requirements Document**

**Functional Requirements**

1. The system will maintain two type of users:
   1. Normal user.
   2. Admin user.
2. The system will initially have two Administrator logins.
3. To gain access to the system Users first of all will need to register by providing the following information:
   1. Email
   2. Phone Number
   3. Date of Birth, which will be used to verify that the user is over 18
   4. Password
   5. If they wish to make a retailer or a regular account
4. Validated users will be allocated a unique User ID Code.
5. Users will be able to login back into the system at any point by typing in their email address and password into the login page.
6. Administrators will be able to login back into the system at any point by typing in their email address and password into the login page.
7. Logged-in Users will be able to create, delete and edit the vinyls that they have for sale.
8. Logged-in Users will be able to search for vinyls by entering information related to the artist, vinyl title.
9. Logged-in Users will be able to click on displayed vinyls to open a page with more detailed information about the vinyl.
10. Logged-in Users will be able to logout.
11. Administrators will be able to view and edit all user and vinyl information.
12. Administrators will be able to logout.
13. The following security measures will be used:
    1. Hashing and Salting for secure passwords, server-side
    2. ‘POST’ HTML method will be used to securely transfer user data between pages
    3. Single and double quotes will be removed server-side before data is passed to the SQL database in order to protect against SQL injection attacks
14. The website will be hosted on a Fife College server ([www.fifecomptech.net](http://www.fifecomptech.net)) that contains MySQL functionality for mariaDB. It runs PHP 7.4.3 on Apache 2.4.41 on top of an Ubuntu operating system. The hardware runs a 64-bit x86 instruction set. All code will be written using the Brackets software and FileZilla will be used to upload files to the server.

**Non-Functional Requirements**

Time-frame/delivery dates:

Website prototype must be online within 6 months.

Hardware/Software Compatibility:

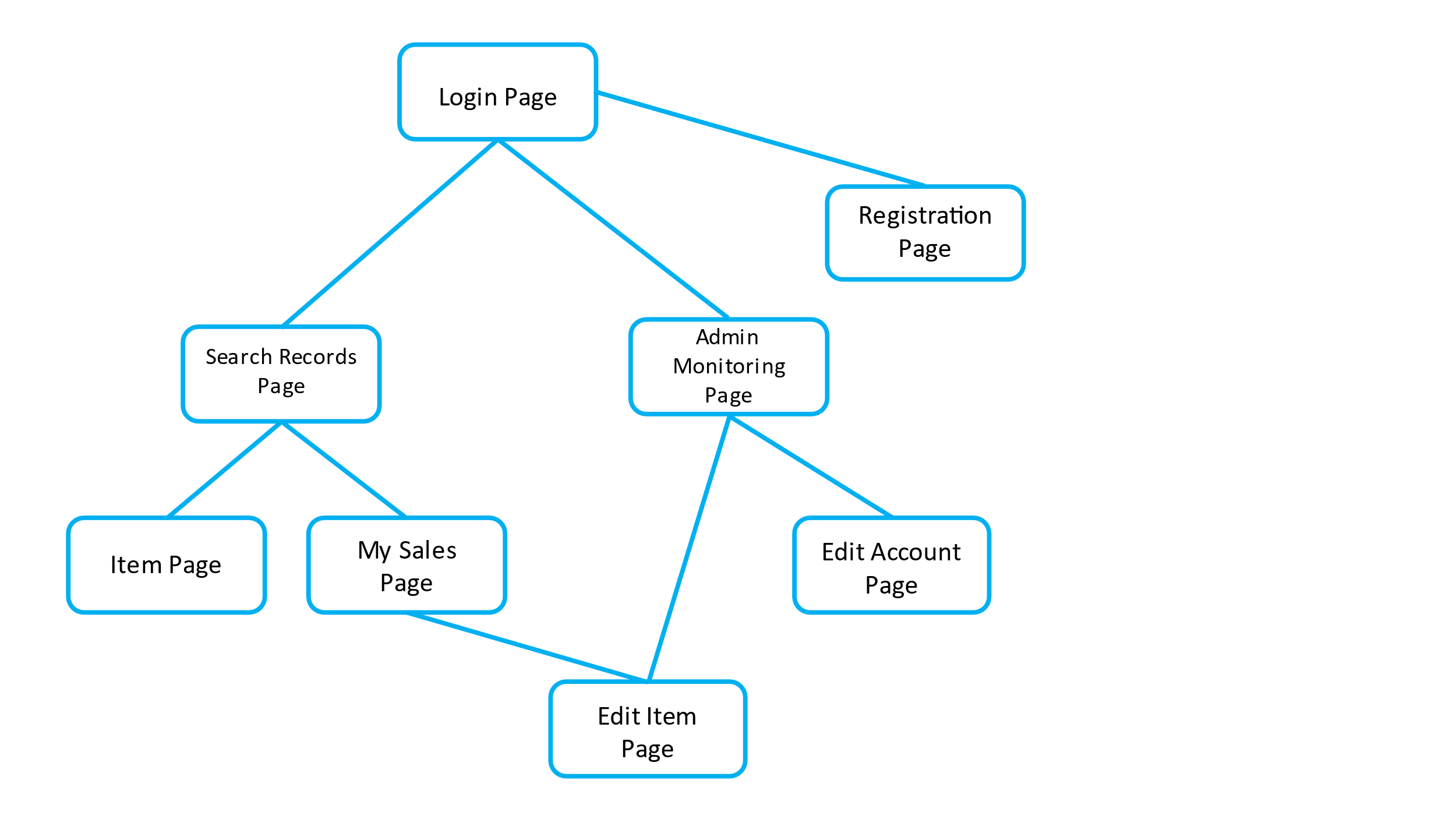
The website must be functional on any device running Google Chrome and Edge browsers.

Target End Users

The website’s target demographic is made up of individuals that are over 18 years of age with very limited competence in computer use. Website must therefore have a highly intuitive user interface. There will be no measures in place to make the website disability-friendly.

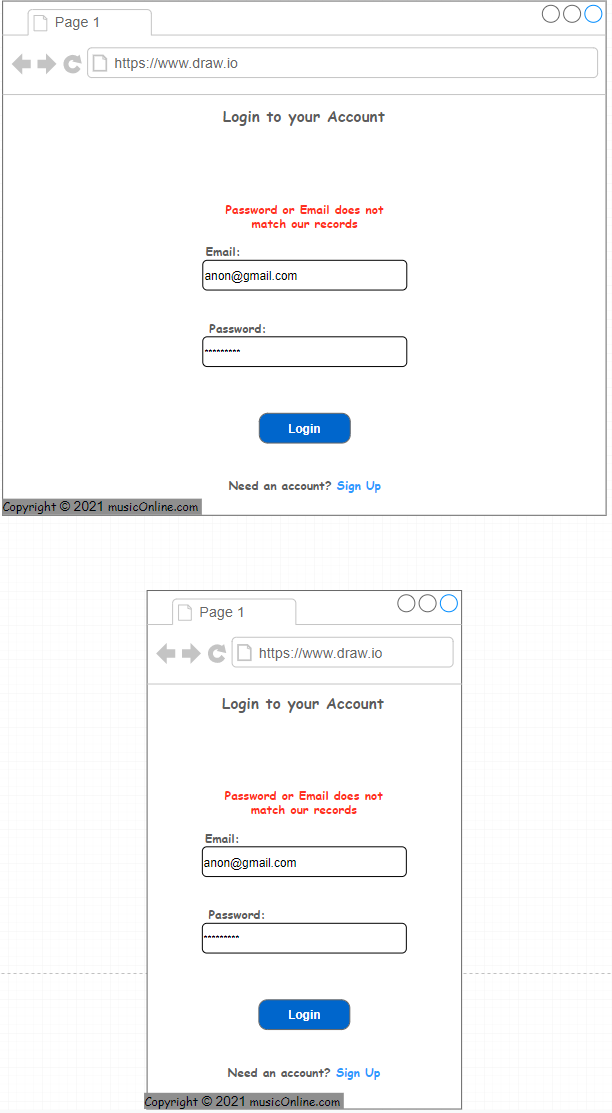
**Design Document**

**Site Structure**

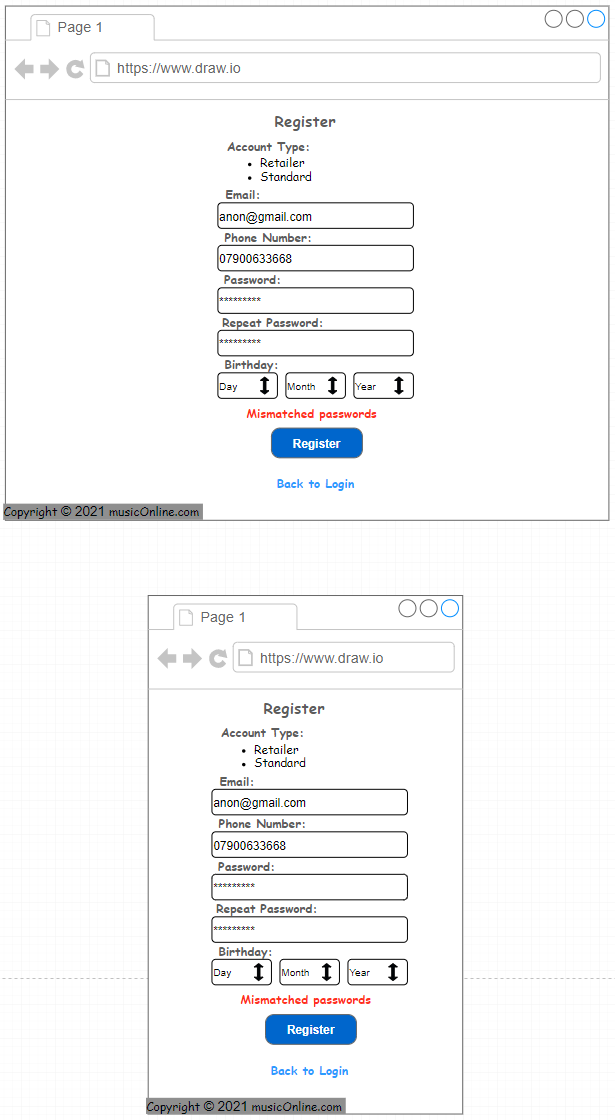


**Wireframes**

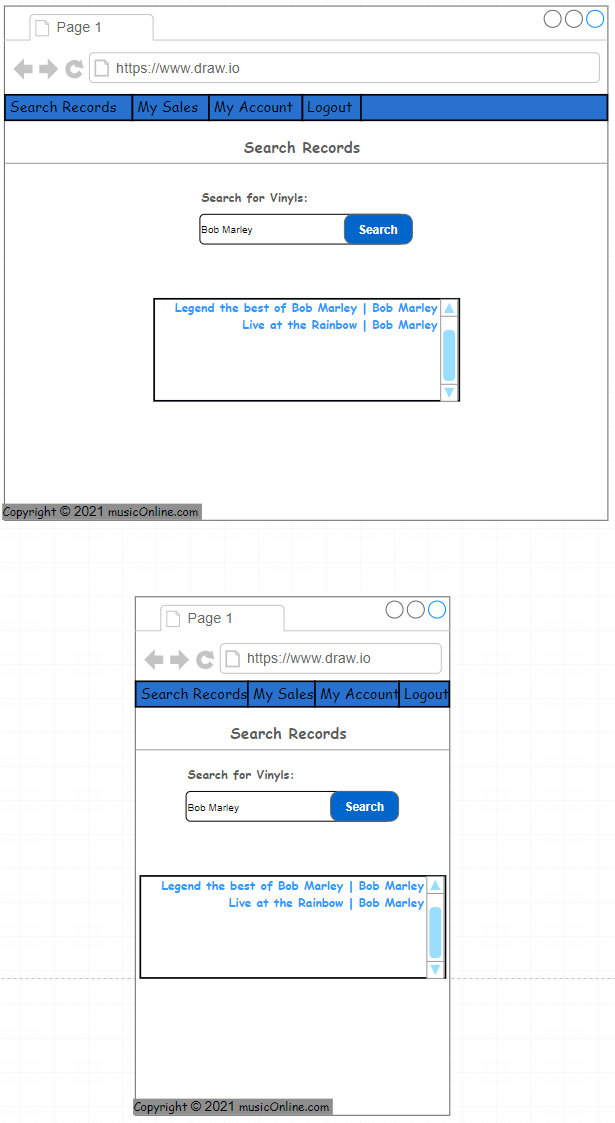
Login Page:



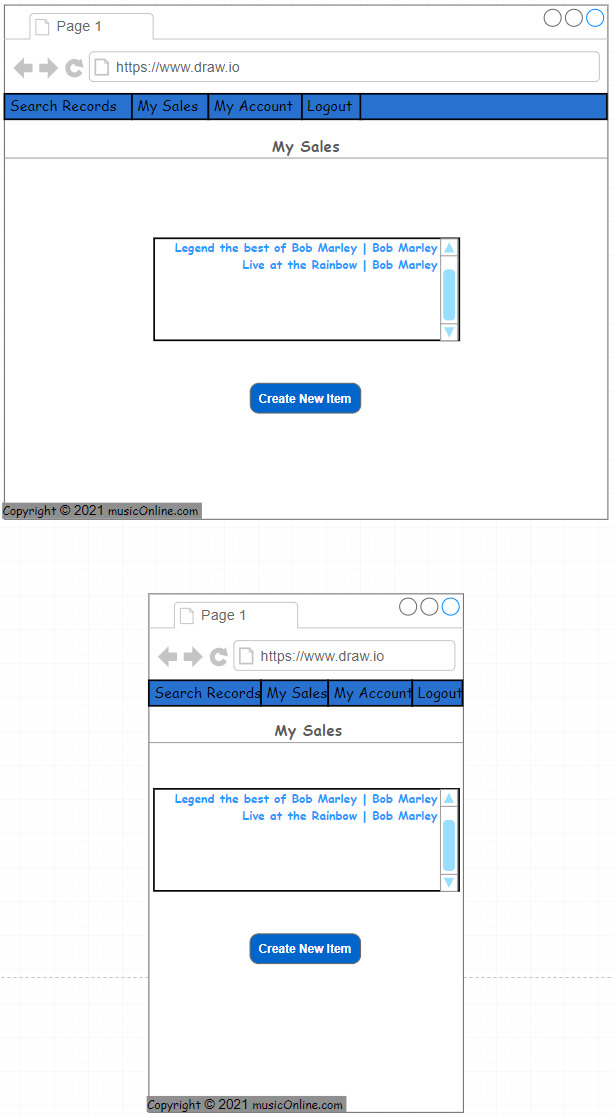
Register Page:



Search Records Page:



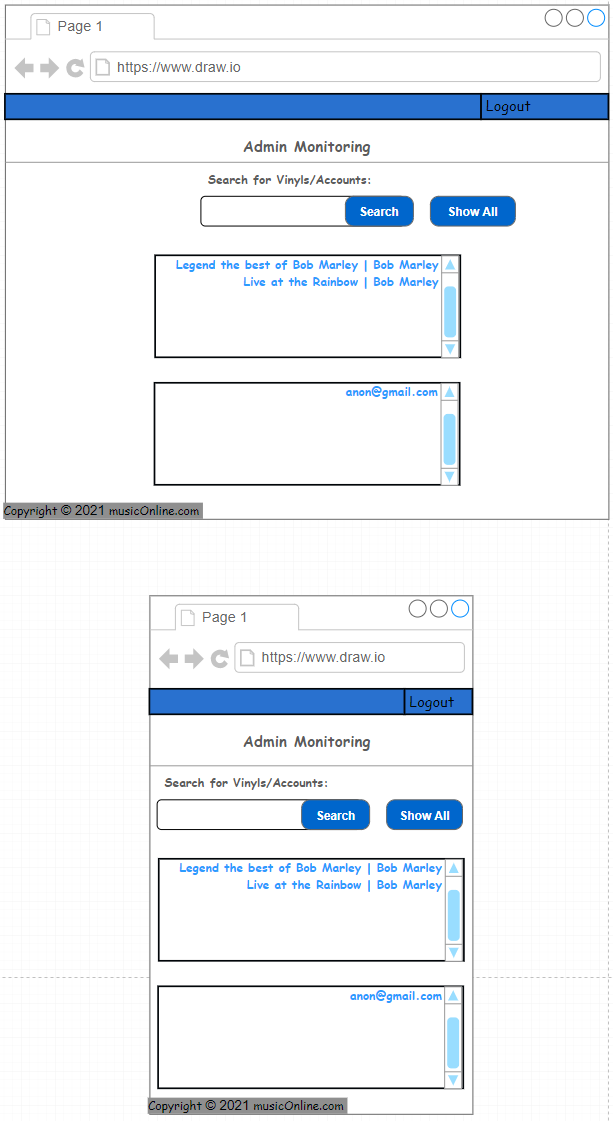
My Sales Page:



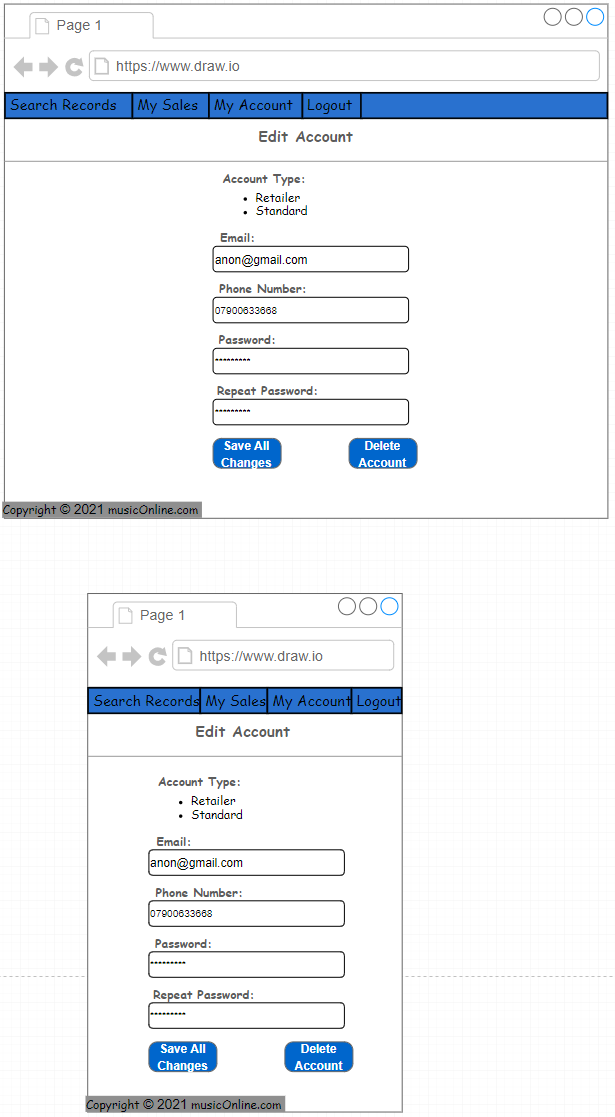
Item Page:



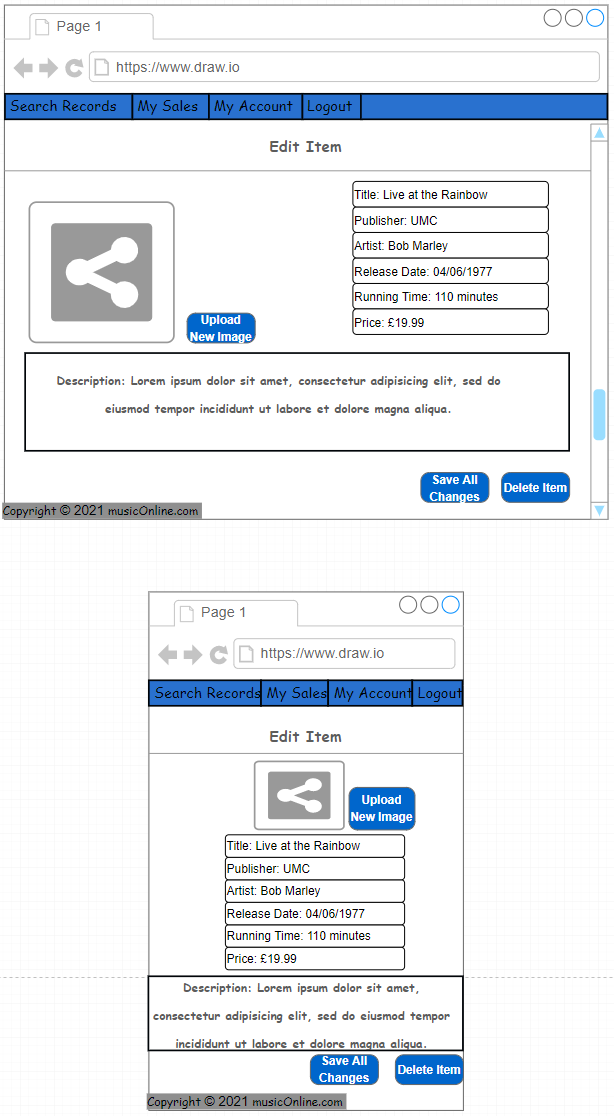
Admin Monitoring Page:



Edit Account Page:



Edit Item Page:



**Page Content Summaries**

Login Page:

* Field for typing in email.
* Field for typing in password.
* Login button.
* Hidden field for displaying login failure message.
* Registration Page navigation link.

Server-side PHP will be used to query the database for a matching email. If it matches then the entered password will be hashed and checked against the password in the database. If that matches then the userID and admin status will be stored in a PHP session and an appropriate page is loaded.

Registration Page:

* Field for typing in email.
* Field for typing in password (disguised characters).
* Field for typing in password repeat (disguised characters).
* Field for typing in a phone number.
* Hidden field for displaying error messages.
* Login Page navigation link.
* Radio buttons for picking between Regular and Retail Accounts.
* Popup prompting user to make a payment if they picked a Retail account.

Client-side JavaScript will be used to verify that the passwords match and that the user is over 18. Data will then be passed to server-side PHP which will check that the email is of a valid format send a query to the MySQL database to store the data after it has been cleared of quotations if the email does not already exist in the database.

Search Records Page:

* Field for typing in a search query.
* Button to launch search for a query.
* Search query results displayed in a table below the search field (includes title and artist). Clicking on an entry leads to a dynamic Item Page.
* Search Records navigation button.
* My Sales Page navigation button.
* My Account navigation button.
* Logout navigation button that clears the logged user and leads to the Login Page.

Entered data will be passed to server-side PHP which will remove quotations before sending a query to MySQL for authors and titles matching the search string. Returned data will then be used to construct the output.

My Sales Page:

* Displays all items on sale by logged user. Items can be clicked to go to a dynamic Edit Item Page for that item.
* ‘Create New Item’ button that will create a new item with no data in database and then load an Edit Item Page for that item allowing the user to fill in the details.
* Search Records navigation button.
* My Sales Page navigation button.
* My Account navigation button.
* Logout navigation button that clears the logged user and leads to the Login Page.

Server-side PHP will query the MySQL database for items linked to the UserID stored in the current session. Returned data will then be used to construct the output.

Item Page:

* Displays all item information in a large, easy to read format.
* A ‘Buy’ button to simulate a purchase.
* Search Records navigation button.
* My Sales Page navigation button.
* My Account navigation button.
* Logout navigation button that clears the logged user and leads to the Login Page.

Server-side PHP will query the MySQL database for data on the item with the selected ProductID. Returned data will then be used to construct the output.

Admin Monitoring Page:

* A field to search for a User Record or a Vinyl Record by email address, vinyl title or vinyl artist.
* A ‘Show All’ button display all User Records followed by all Vinyl Records.
* Records displayed in a table below the search field. Clicking on an entry leads to a dynamic Item/Account Page.
* Logout navigation button that clears the logged user and leads to the Login Page.

Entered data will be passed to server-side PHP which will remove quotations before sending a query to MySQL for email addresses, authors and titles matching the search string. If “Show All” is pressed then it will query the database for all non-admin users and items. Returned data will then be used to construct the output.

Edit Account Page:

* Displays all account information in a form with editable fields.
* ‘Save All Changes’ button that will replace the account details in the database with new information from the form.
* ‘Delete Account’ button allowing the account to be deleted from the database. This button will also take user to Admin Monitoring page if user is an admin and the Login page if user is a non admin.
* Admin Monitoring Page navigation button if user is an admin.
* Search Records navigation button if user is not an admin.
* My Sales Page navigation button if user is not an admin.
* My Account navigation button if user is not an admin.
* Logout navigation button that clears the logged user and leads to the Login Page.

Server-side PHP will query the MySQL database for data on the user with the selected UserID. Returned data will then be used to construct the output. When a field is edited client-side JavaScript will prompt user for new data and then send it to server-side PHP. PHP will remove all quotations before querying the MySQL database to replace the old data with the new one. For emails, PHP will first verify that it is valid and then query the database to check if the new email already exists before inserting it. Incorrect input will display an error. The “Delete Account” button will first display a JavaScript warning. Then, PHP will query the database to delete the account.

Edit Item Page

* Displays all account information in a form with editable fields.
* ‘Save All Changes’ button that will replace the item details in the database with new information from the form.
* ‘Delete Item’ button allowing the item to be deleted from the database. This button will also take user to Admin Monitoring page if user is an admin and the Search Records page if user is a non admin.
* Admin Monitoring Page navigation button if user is an admin.
* Search Records navigation button if user is not an admin.
* My Sales Page navigation button if user is not an admin.
* My Account navigation button if user is not an admin.
* Logout navigation button that clears the logged user and leads to the Login Page.

Server-side PHP will query the MySQL database for data on the item with the selected ProductID. Returned data will then be used to construct the output. When a field is edited client-side JavaScript will prompt user for new data and then send it to server-side PHP. PHP will remove all quotations before querying the MySQL database to replace the old data with the new one. For price, release date and running time, PHP will check that they are of the correct format first. Incorrect input will display an error. The “Delete Item” button will first display a JavaScript warning. Then, PHP will query the database to delete the item.

**Data Source Design -Data Dictionary (MySQL)**

* **AdminTable:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **PK/FK** | **Constraints** |
| UserID | INT | 10 | PK | Not Null, Auto Increment |
| Email | VARCHAR | 100 | - | Not Null, Unique |
| Phone | VARCHAR | 20 | - | Not Null |
| PassHash | VARCHAR | 255 | - | Not Null |

* **UserTable:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **PK/FK** | **Constraints** |
| UserID | INT | 10 | PK | Not Null, Auto Increment |
| Retailer | ENUM | - | - | Not Null |
| Email | VARCHAR | 100 | - | Not Null, Unique |
| Phone | VARCHAR | 20 | - | Not Null |
| PassHash | VARCHAR | 255 | - | Not Null |

* **MusicTable:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **PK/FK** | **Constraints** |
| ProductID | INT | 13 | PK | Not Null, Auto Increment |
| SellerID | INT | 10 | FK | Not Null |
| ImagePath | VARCHAR | 255 | - | Not Null, Unique |
| Title | VARCHAR | 100 | - | Not Null |
| Price | DECIMAL | 6,2 | - | Not Null |
| Artist | VARCHAR | 100 | - | Not Null |
| Publisher | VARCHAR | 100 | - | Not Null |
| ReleaseDate | DATE | - | - | Not Null |
| RunningTime | INT | 3 | - | Not Null |
| Description | VARCHAR | 255 | - | Not Null |

Note: MusicTable’s SellerID links to UserTable’s UserID.