The Note

Adrian Bernardino, Mark Laow, Stephania Rey, Matthew Roman

Team 3 – CIS 444

Dr. Jing Hou

Report version 2.0

Name of website/company: The Note

OUR LOGO

1. Project Description-**Mark**

The real-life problem we want to solve with our Web server is to have a platform where we can have a store to purchase music-related products. We understand that customers want a fast and accessible service, and our shopping cart feature provides just that. We offer competitive prices on everything from classic to modern products. In addition to providing a wide variety of products, we also strive to offer a seamless and user-friendly experience to our customers. Our website is designed to be intuitive, allowing users to browse through our extensive catalog and make purchases with ease. We have implemented secure payment gateways to ensure that all transactions are safe and secure. With society moving towards online shopping, We recognize the importance of providing a reliable and trustworthy platform for our customers. Our team is dedicated to continuously improving our website's functionality and features, to enhance the user experience and cater to the evolving needs of our customers.

2. Project Specification-**Mark**

The requirements for this project are listed below:

- Multiple pages linked
- Server-PHP for database/backend
- Front-end coded in HTML, CSS, and JavaScript

3. User’s Manual-**Matt**

Regarding the user’s manual for our webpage, we have crafted our site with a user-friendly and intuitive design, making it visually appealing and easy to navigate. Our website is built using PHP, CSS, and JS that provide a hassle-free experience for the user.

4. Web Pages and their Interrelationships-**Matt**

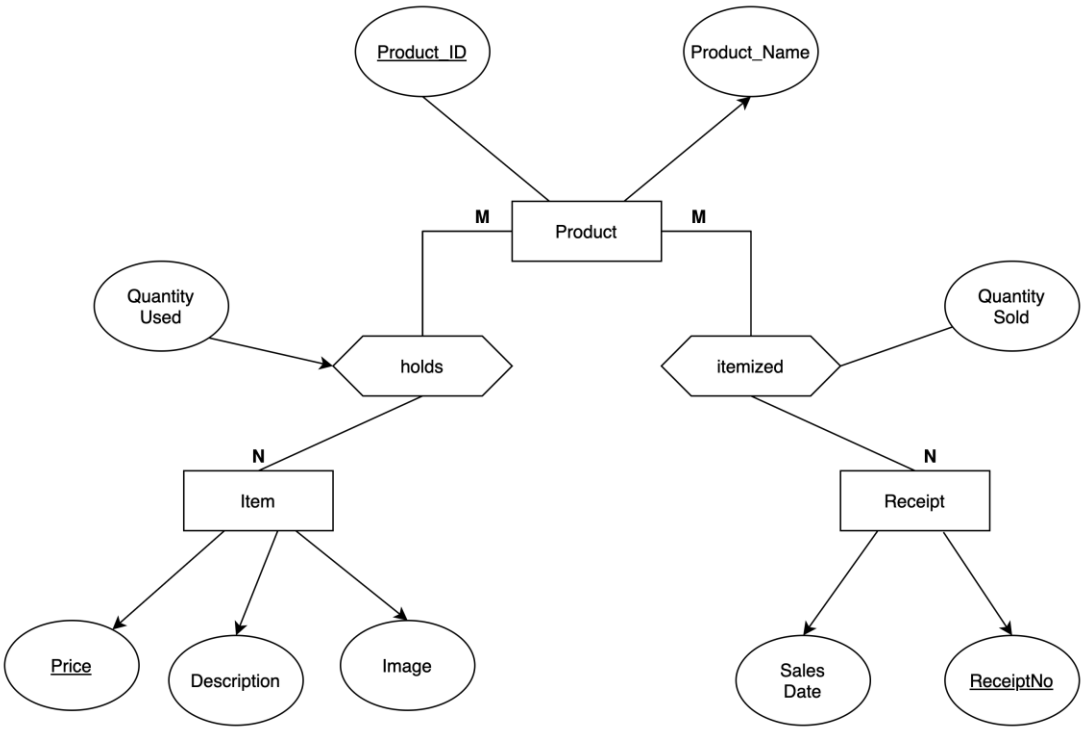
Our home page will showcase various services, including a navigation bar positioned across the screen that links all pages together seamlessly. This bar features our website's logo, which redirects the user back to the homepage. Users can sign in or sign up, access a shopping cart, manage their purchases, and use the search tab to find specific products. Our footer section includes an "About Us" summary and a FAQ section that addresses common user queries. The sign-up and account pages are interlinked through PHP, enhancing the user's experience. Additionally, we have employed scripts to facilitate functions such as password protection and value modification.

Furthermore, our navigation bar features a drop-down menu button, offering relevant information or options. For instance, our CD's/DVD's and vinyl's drop-down menu allows users to browse through our entire inventory or look for a specific artist or genre. Similarly, our clothing button's drop-down menu will suggest options like sweatshirts or shirts, while the instruments button offer users the choice to shop for all our instruments or select a specific one.

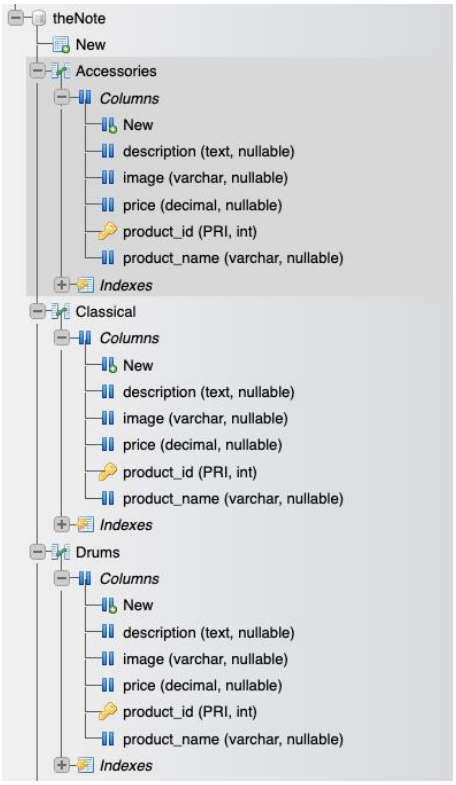
5. Database Design and Implementation-**Adrian**

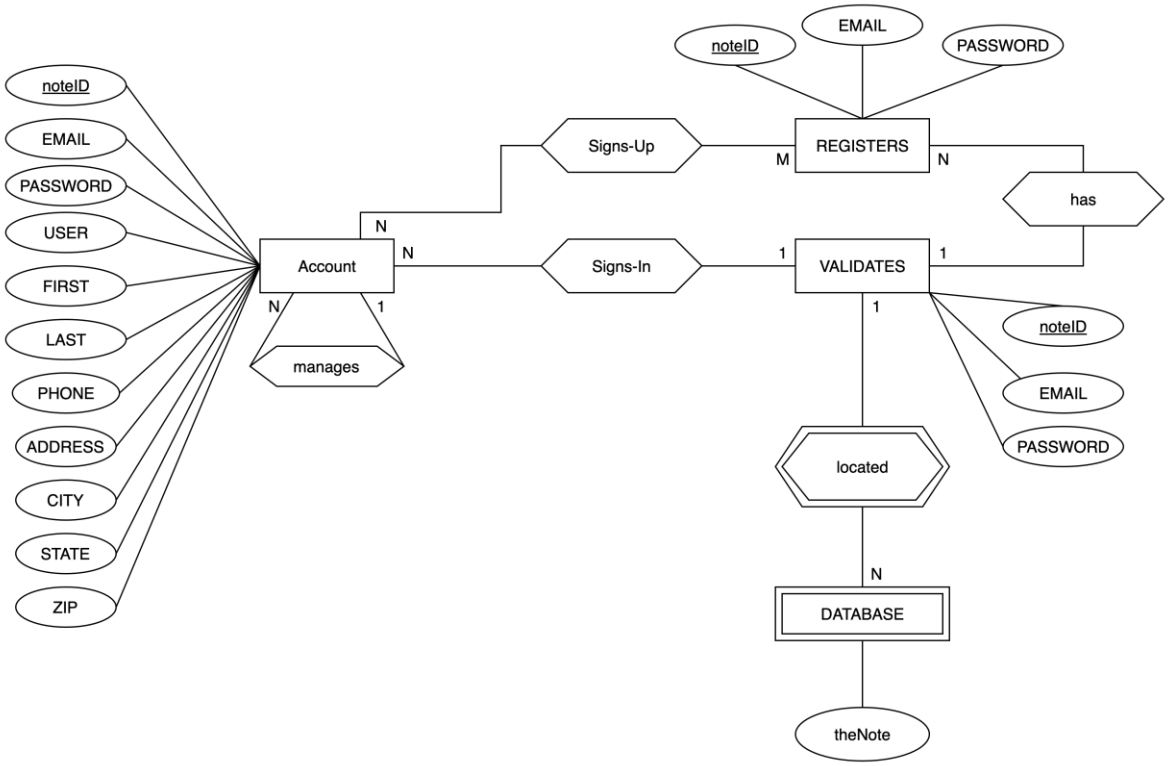
(Database should be well designed to support your website)

- p1) Database ER diagram: picture and the corresponding description

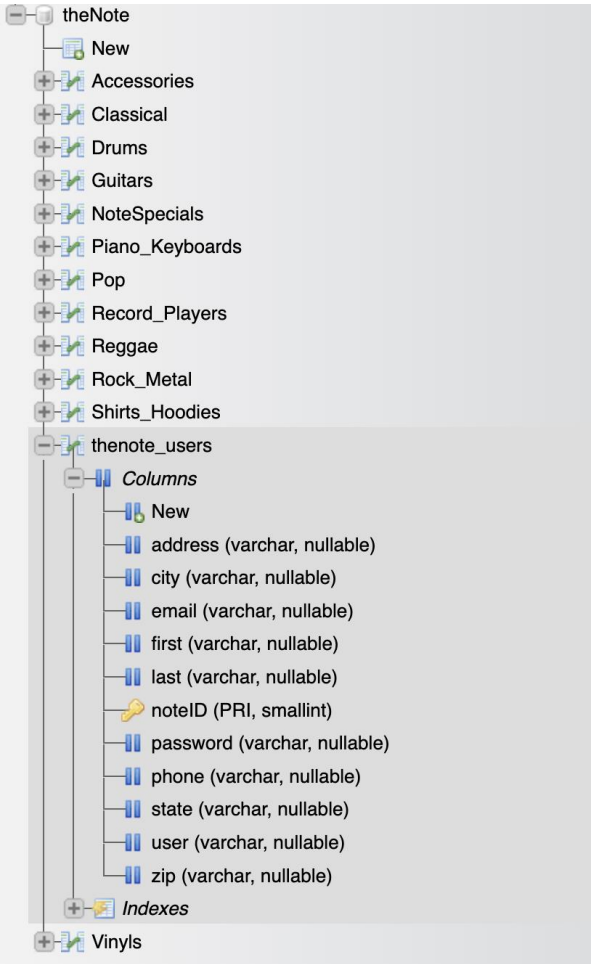


For this ER diagram, we implemented the same structure for all our products. Minimizing complex attribute notations helped us focus on a clear outlook when it came to creating our tables. As you can see in the above diagram, we wanted to focus on how the Product data is grabbing items or the receipt for when it's time to check out. The tables that are derived from this ER diagram were Accessories, Classical, Drums, Guitars, NoteSpecials, Piano_Keyboards, Pop, Record_Players, Reggae, Rock_Metal, Shirts_Hoodies, & Vinyls.





For our second ER diagram, we wanted to illustrate the process of what is occurring in our database and when the User is either registering a new account or if the user has already signed up, they are able to still retrieve that information from our database and sign back in based upon existing noteID, email and password. Some things we still wanted to implement soon were where the user is able to manage their account and add their information which is then saved onto the database as well. This part of the feature was going to be used to link to our user's checkout information, in order to receive orders successfully and to the right person.



- p2)How the tables are implemented, i.e., show your SQL CREATE queries.

```
CREATE DATABASE theNote;

USE theNote;

CREATE TABLE thenote_users (
```

```
        noteID SMALLINT NOT NULL AUTO_INCREMENT PRIMARY KEY,

        email VARCHAR(40),

        password VARCHAR(40),

        first VARCHAR(40),

        last VARCHAR(40),

        phone VARCHAR(40),

        address VARCHAR(40),

        city VARCHAR(40),

        state VARCHAR(2),

        zip VARCHAR(10)

);



CREATE TABLE Accessories (
        product_id INT PRIMARY KEY,
        product_name VARCHAR(100),
        price DECIMAL(10, 2),
        description TEXT,
        image VARCHAR(100)
);


 CREATE TABLE Classical (
      product_id INT PRIMARY KEY,
      product_name VARCHAR(100),
```

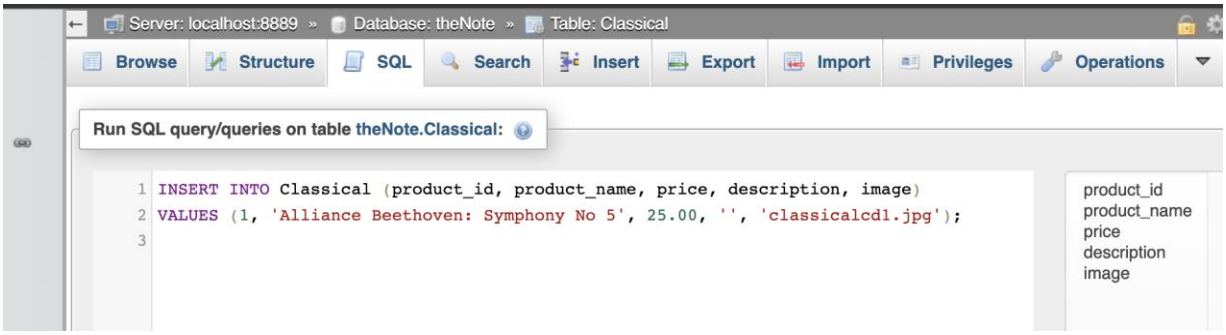
```
    price DECIMAL(10, 2),
    description TEXT,
    image VARCHAR(100)
);



CREATE TABLE Drums (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100),
    price DECIMAL(10, 2),
    description TEXT,
    image VARCHAR(100)
);

CREATE TABLE Guitars (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100),
    price DECIMAL(10, 2),
    description TEXT,
    image VARCHAR(100)
);

CREATE TABLE NoteSpecials (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100),
    price DECIMAL(10, 2),
    description TEXT,
    image VARCHAR(100)
);




CREATE TABLE Piano_Keyboards (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100),
    price DECIMAL(10, 2),
    description TEXT,
    image VARCHAR(100)
);

CREATE TABLE Pop (
    product_id INT PRIMARY KEY,
```

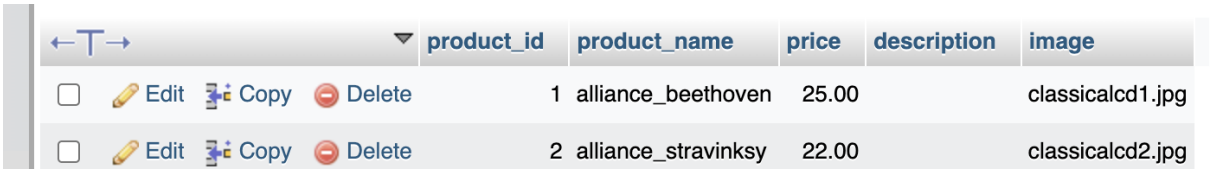
```
    product_name VARCHAR(100),
    price DECIMAL(10, 2),
    description TEXT,
    image VARCHAR(100)
);



CREATE TABLE Record_Players (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100),
    price DECIMAL(10, 2),
    description TEXT,
    image VARCHAR(100)

);


CREATE TABLE Reggae (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100),
    price DECIMAL(10, 2),
    description TEXT,
    image VARCHAR(100)

);


CREATE TABLE Rock_Metal (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100),
    price DECIMAL(10, 2),
    description TEXT,
    image VARCHAR(100)
);

CREATE TABLE Shirts_Hoodies (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100),
    price DECIMAL(10, 2),
    description TEXT,
    image VARCHAR(100)
```

```
);

CREATE TABLE Vinyls (

    product_id INT PRIMARY KEY,
    product_name VARCHAR(100),
    price DECIMAL(10, 2),
    description TEXT,
    image VARCHAR(100)

);
```

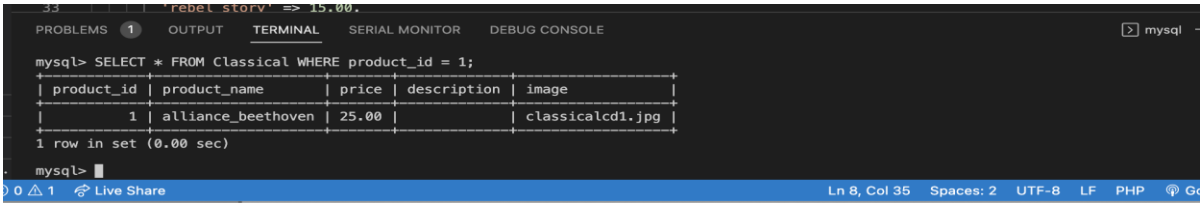
- p3) An illustration on how your test INSERT, SELECT, UPDATE and DELETE queries for your table(s) on MySQL

There was a total of 116 inventory items that needed to be added to each corresponding table and category. For generalization purposes, I'll show the Classical category. This was the way we inserted our values into our database. This helped run php methods smoothly by creating an inventory array so that when the item is selected it automatically updates into our shopping cart total based on the values in our database.





	product_id	product_name	price	description	image
Edit Copy Delete	1	alliance_beethoven	25.00		classicalcd1.jpg
Edit Copy Delete	2	alliance_stravinksy	22.00		classicalcd2.jpg



If we wanted to update our prices, we could run this statement:

Run SQL query/queries on table theNote.Classical:

```sql
UPDATE Classical
SET price = 14.99
WHERE product_name = 'alliance_beethoven';
```

PROBLEMS 1 OUTPUT TERMINAL SERIAL MONITOR DEBUG CONSOLE

```
mysql> SELECT * FROM Classical WHERE product_id = 1;
+------------+--------------------+-------+-------------+------------------+
| product_id | product_name       | price | description | image            |
+------------+--------------------+-------+-------------+------------------+
|          1 | alliance_beethoven | 14.99 |             | classicalcd1.jpg |
+------------+--------------------+-------+-------------+------------------+
1 row in set (0.00 sec)

mysql>
```

0 ⚠ 1 Live Share Ln 391, Col 7

```
3
```

```php
<?php
if (empty($_GET['product_name']) || empty($_GET['price'])) {
exit("<p>You must select an item to Add to cart! </p>");
}
$PName = $_GET['product_name'];
$Price = $_GET['price'];
```

```php
// Connecting to the database with PDO
require_once "config.php";
$TableName = "NoteSpecials";
$TableName = "Accessories";
$TableName = "Classical";
$TableName = "Drums";
$TableName = "Guitars";
$TableName = "Piano_Keyboards";
$TableName = "Pop";
$TableName = "Record_Players";
$TableName = "Reggae";
$TableName = "Rock_Metal";
$TableName = "Shirts_Hoodies";
$TableName ="Vinyls";

$sql = "SELECT * FROM $TableName
WHERE product_name = :productName
AND price = :price";
$stmt = $pdo->prepare($sql);
$stmt->bindParam(':productName', $PName);
$stmt->bindParam(':price', $Price);
$stmt->execute();

if (!$row = $stmt->fetch()) {
exit("You must enter a valid product name and price. Click your browser's Back button to return to the previous page.</p>");
} else {
$ProductID = $row['product_id'];
$PName = $row['product_name'];
$Price = $row['price'];
$Description = $row['description'];
```

```
$Image = $row['image'];
}
// Closes the connection and frees the resources used by the PDO object
$pdo = null;
?>
```

- p4) Show the content of the tables after populating them with some sample data

Here is some sample data that shows when the user chooses a new item from the database, our website will update the shopping cart total price.

Showing rows 0 - 2 (3 total, Query took 0.0002 seconds.)

```
SELECT * FROM `Classical`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

←T→	product_id	product_name	price	description	image
Edit Copy Delete	1	alliance_beethoven	25.00		classicalcd1.jpg
Edit Copy Delete	2	alliance_stravinksy	22.00		classicalcd2.jpg
Edit Copy Delete	3	alliance_maria_callas	45.00		classicalcd3.jpg

Classical CDs

Alliance Beethoven: Symphony No 5

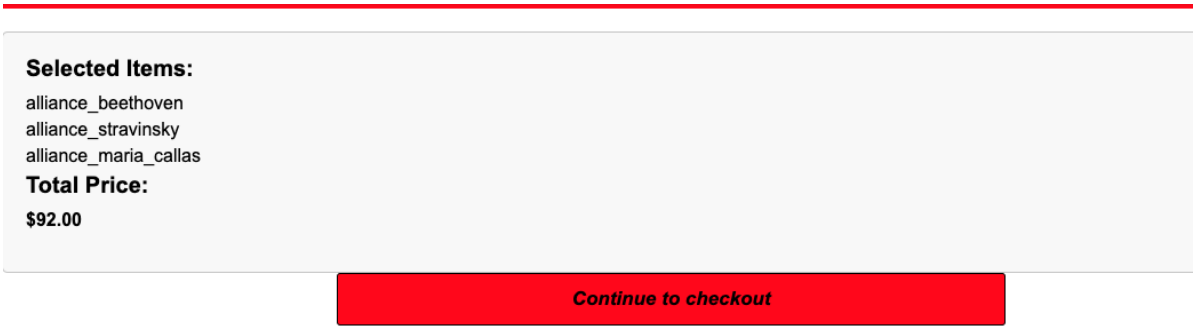
Price: $25.00

Alliance Igor Stravinsky - Stravinsky Conducts Stravinsky: Firebird

Price: $22.00

Alliance Maria Callas - La Traviata (1953 Studio Recording)

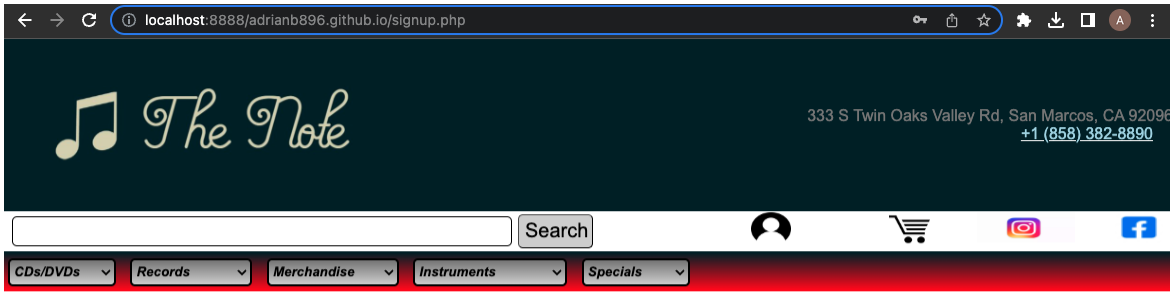
Price: $45.00



5) A description on how some or all your Web pages are supposed to interact with your database(s), i.e., what queries will be posted from which pages and how the results will be displayed in those pages -**Adrian**

For the computation portion of when the user is signing up, I went ahead and added a php method to get from our database or update our database when the user is registering for the first time it will add that information or when the user is logging in, it will validate if the email and password exists within our database already.

Ex:
1. User is signing up for the first time (signup.php)



Please fill in this form to create an account

Email

testing@email.com

Confirm Email

testing@email.com

Password

•••••

Repeat Password

•••••

☑ Remember me

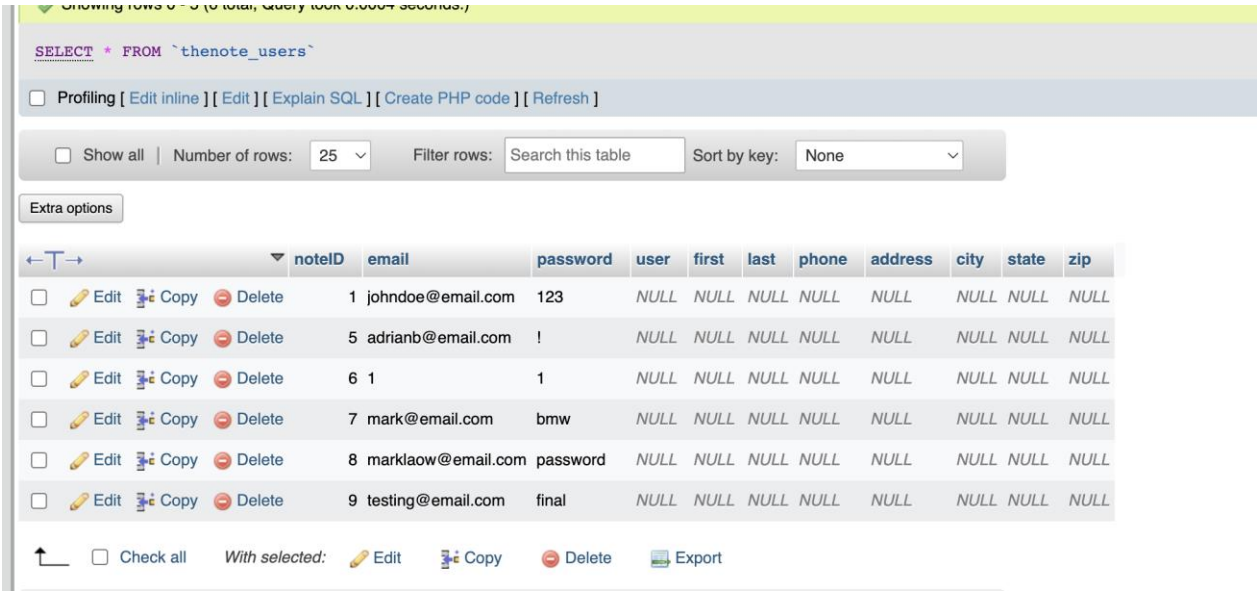
By creating an account you agree to our
Terms & Privacy

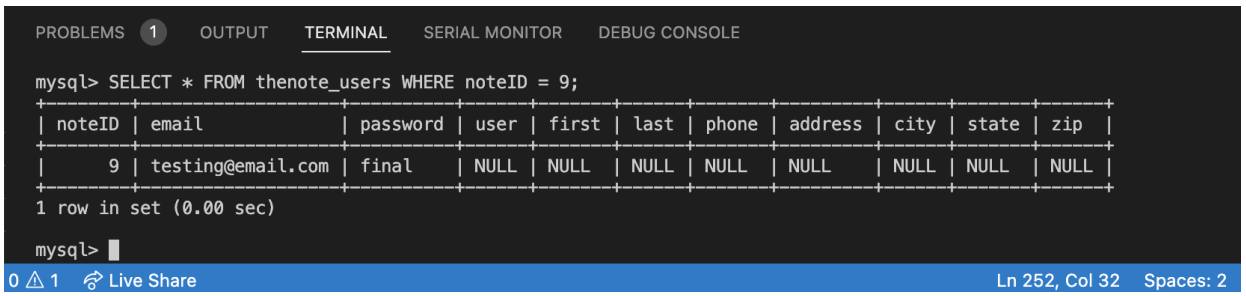
Cancel ***Sign Up***

2. Once the user clicks on sign up, it will direct them to their account info (account_test.php)

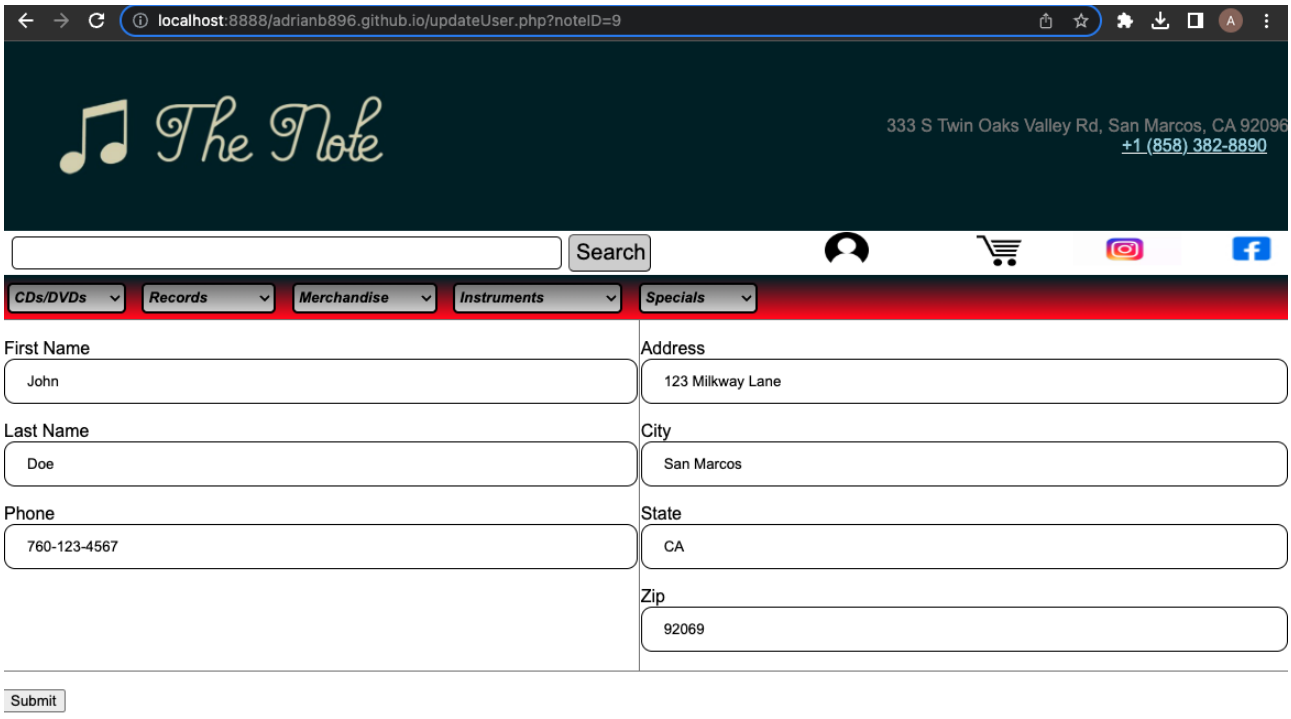
```
<?php
//check the user input first
if
(empty($_GET['email'])||empty($_GET['email_confirm'])||empty($_GET['password'])||empty($_
GET['password_confirm']))
exit ("<p> You must enter values in all fields ! Click your browser's Back button to return to the
previous page.</p>");
else if ($_GET["email"]!=$_GET["email_confirm"])
exit ("<p> You did not enter the same email address! Click your browser's Back button to return
to the previous page.</p>");
else if ($_GET["password"]!=$_GET["password_confirm"])
exit ("<p> You did not enter the same password! Click your browser's Back button to return to
the previous page.</p>");
//connecting to the database with PDO
require_once("config.php");
$TableName = "thenote_users";
$Email=$_GET['email'];
$Password = $_GET['password'];
// check if already registered with the email
$sql = "SELECT * FROM $TableName";
$result = $pdo->query($sql);
while ($row = $result->fetch()) {
//the email match the field from the table
if ($row['email']==$Email)
exit("<p>The email address you entered is already registered! Click your browser's Back button
to return to the previous page.</p>");
}
```

```php
//insert into new user data to the table
$sql = "INSERT INTO $TableName ( email, password) VALUES ('$Email','$Password')";
$pdo->exec($sql);
//retrieve the flyerID
$sql = "SELECT * FROM $TableName WHERE email = '$Email'";
$result= $pdo->query($sql);
if($row = $result->fetch())
$NoteID = $row['noteID'];
// closes connection and frees the resources used by the PDO object
$pdo = null;
?>
```

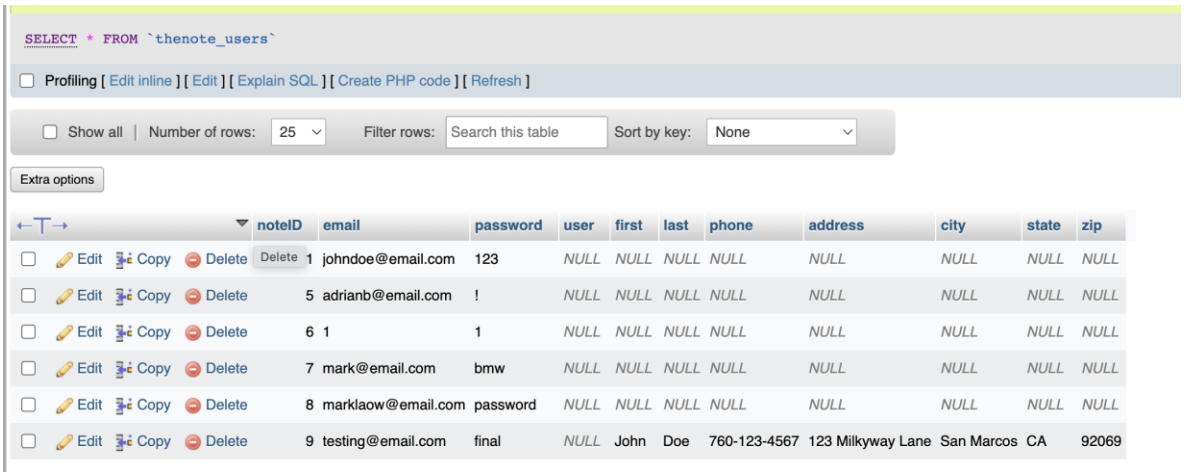




3. A future change I want to make is to have the user be able to update their information inside of their account page by clicking on the Change Info button. Currently the account change info is being implemented with purely JavaScript. However, if a user was to enter their information here in this form our database does update with that new information. (updateUser.php)



```php
<?php
require_once("config.php");
if(isset($_GET['noteID']))
$FlyerID = $_GET['noteID'];
//get existing user info from the database, to be shown in the form
$TableName = "thenote_users";
$sql = "SELECT * FROM $TableName WHERE noteID = '$NoteID' ";
$result = $pdo->query($sql);
if($row = $result->fetch()){
$Last=$row["last"];
$First=$row["first"];
$Phone=$row["phone"];
$Address=$row["address"];
$State=$row["state"];
$City=$row["city"];
$Zip=$row["zip"];
}
else{
$Last="";
$First="";
$Phone="";
$Address="";
$State="";
$City="";
$Zip="";
}
// closes connection and frees the resources used by the PDO object
$pdo = null;
?>
```



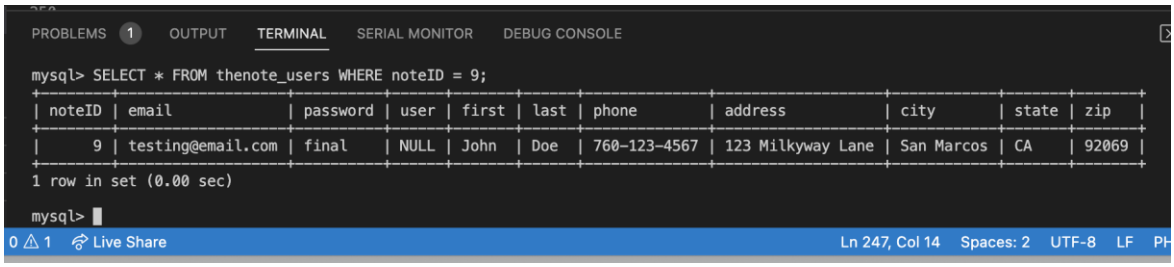
```
SELECT * FROM `thenote_users`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	noteID	email	password	user	first	last	phone	address	city	state	zip
Edit Copy Delete	1	johndoe@email.com	123	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Edit Copy Delete	5	adrianb@email.com	!	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Edit Copy Delete	6	1	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Edit Copy Delete	7	mark@email.com	bmw	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Edit Copy Delete	8	marklaow@email.com	password	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Edit Copy Delete	9	testing@email.com	final	NULL	John	Doe	760-123-4567	123 Milkyway Lane	San Marcos	CA	92069



```
PROBLEMS 1  OUTPUT  TERMINAL  SERIAL MONITOR  DEBUG CONSOLE

mysql> SELECT * FROM thenote_users WHERE noteID = 9;
+--------+------------------+----------+------+-------+------+--------------+-------------------+------------+-------+-------+
| noteID | email            | password | user | first | last | phone        | address           | city       | state | zip   |
+--------+------------------+----------+------+-------+------+--------------+-------------------+------------+-------+-------+
|      9 | testing@email.com | final    | NULL | John  | Doe  | 760-123-4567 | 123 Milkyway Lane | San Marcos | CA    | 92069 |
+--------+------------------+----------+------+-------+------+--------------+-------------------+------------+-------+-------+
1 row in set (0.00 sec)

mysql>

0 ⚠ 1  Live Share                                  Ln 247, Col 14  Spaces: 2  UTF-8  LF  PH
```

4. Lastly, now that the user has signed up and registered a new account, the new user is now able to sign in using the email and password they used to sign up with. The way this part works is that the database is connected to an existing value that the user added when signing up. When trying to log in, the php method is checking or validates to see if there is an existing value for this user. It looks for both the email and password within our 'thenote_users' table. (signin.php)

5. Then once the user clicks 'Sign In,' they will be prompted with their account info which is being processed through our validation method in PHP that checks for existing email and password within our database. (validateUser.php -> account_test.php)

```
<?php
if (empty($_GET['email'])||empty($_GET['password']))
exit ("<p> You must enter values in all fields! Click your browser's Back button to return to the

$Email=$_GET['email'];
$Password = $_GET['password'];
//connecting to the database with PDO
require_once "config.php";
$TableName = "thenote_users";
$sql = "SELECT * FROM $TableName



WHERE email = '$Email'
AND password = '$Password'";
$result = $pdo->query($sql);
//echo "<pre>$sql<pre>";
if(!$row = $result->fetch())
exit("You must enter a valid email address and password. Click your browser's Back button to return to the previous page.</p>");
else {
$NoteID = $row['noteID'];}
// closes connection and frees the resources used by the PDO object
$pdo = null;

?>
```

6) Web Design Issues such as resolved difficulties, possible future improvements and enhancements

Resolve difficulties:

The team faced several challenges during the project, including inconsistencies with the navbar color across all the web pages. This issue was addressed by revisiting the CSS code and making necessary modifications to ensure that the color scheme was consistent throughout the website.

Another challenge we encountered was related to the PHP files not functioning properly with our repository name. To resolve this issue, we had to rename our repository and update all the file paths and references accordingly. This required us to spend additional time testing the website thoroughly to ensure that all PHP files were working correctly.

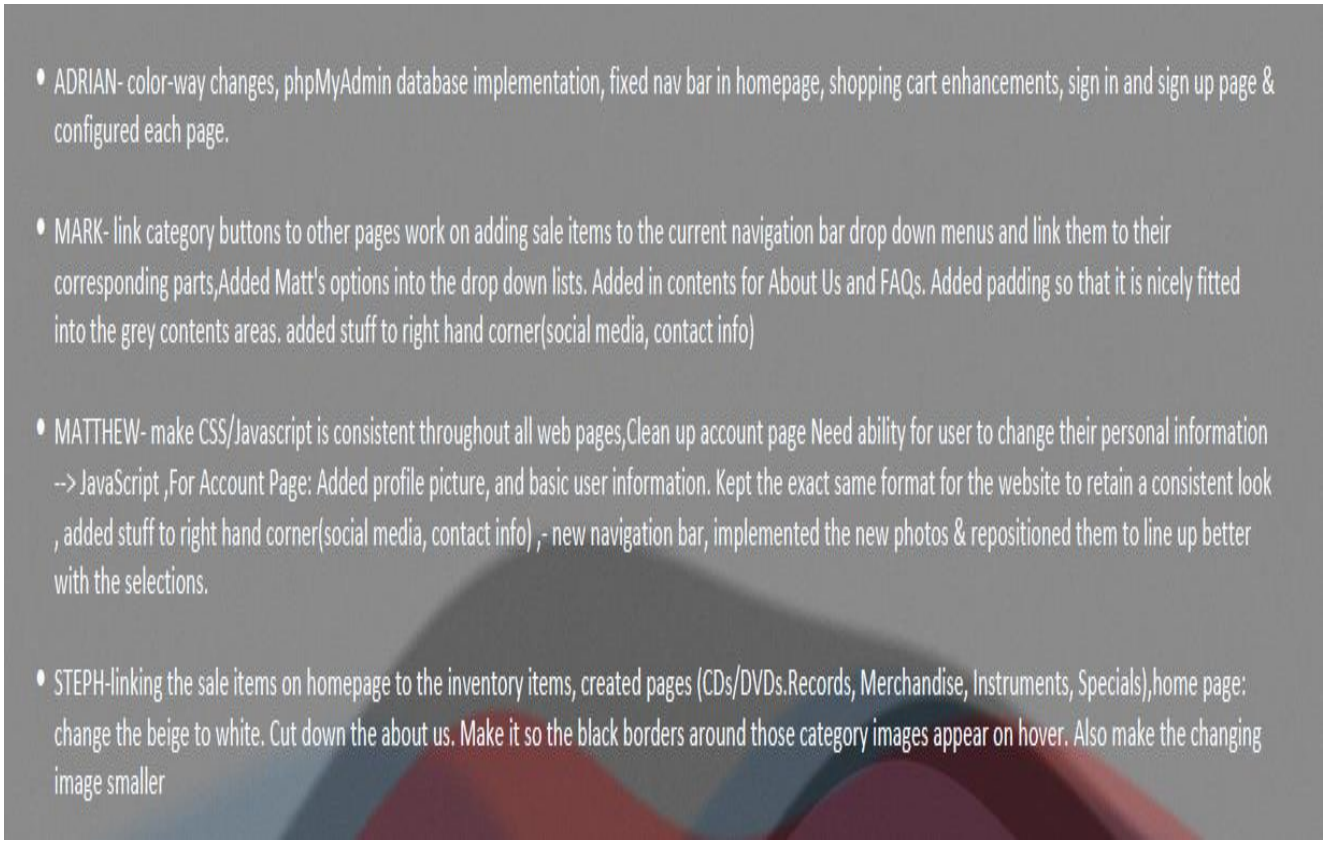
Overall, these challenges taught us valuable lessons and helped us improve our skills and problem-solving abilities.

Future improvements:

We have considered a future improvement to utilize cloud services to enhance the performance and reliability of our webpage. By leveraging cloud computing technology, we can ensure seamless access to our website, improve page load times, and enhance security features.

To make our website dynamic and interactive, we plan tto implementPHP throughout all our web pages. This will enable us to add advanced functionality to our website, such as user authentication, database connectivity, and dynamic content creation. These various tools and techniques will ensure users to have to accessibility, usability, and search engine optimization of our webpages.

7) The individual contribution of each member –Steph



Dates of Meetings:

Date: 2/27/2023

Time: 7pm-8pm

Who attended: Adrian Bernardino, Mark Laow, Matthew Roman, Stephania Rey

What you have discussed:

-topics pertaining to the project

-Discussing Team's availability and schedules

-Discussing topics of choice for project's backend side

-Discussing team's programming skills

What is the current challenge: Selecting topic for project and project name

Next plan:

-Meeting with Professor Hou to get clarity of project specifics

-Coordinate meeting times/dates for team

Date: 3/02/2023

Time: 7pm-8pm

Who attended: Adrian Bernardino, Mark Laow, Matthew Roman, Stephania Rey

What you have discussed:

-Reviewing questions with response asked to Professor Hou regarding the project

-Discussing project's topic of choice

-Creating project outline report

What is the current challenge: Selecting project name

Next plan:

-Work on report outline completion

Date: 3/05/2023

Time: 11am-12am

Who attended: Adrian Bernardino, Mark Laow, Matthew Roman, Stephania Rey

What you have discussed:

-Creating web page design uml diagram

-Working on finalization of project outline report

What is the current challenge: Selecting ways to incorporate PHP to our project

Next plan:

-Review project report before submitting

-Discuss how we want to design and stylize our page, functionalities

Date: 3/10/2023

Time: 11am to 12pm

Who attended: Adrian Bernardino, Mark Laow, Matthew Roman, Stephania Rey

What you have discussed:

-Discussed the team's availability to meet in the upcoming weeks

-Discussed the web page layouts and structure overlook

-Set up the project's presentation layout

-Approval and submission of the team's report

-Discussed the team's availability to meet in the upcoming weeks

-Discussed the web page layouts and structure overlook

What is the current challenge: Working on coding the webpage

-Set up the project's presentation layout

Next plan:

What is the current challenge: Working on coding the webpage

Next plan:

-assign tasks for each team member to work on their respective pages

-work on presentation slides edits

Date: 3/12/2023

Time:11am to 12:30pm

Who attended: Adrian Bernardino, Mark Laow, Matthew Roman, Stephania Rey

What you have discussed:

-Discussed the team's availability to meet in the upcoming weeks

-Discussed the web page layouts and structure overlook

-Set up the project's presentation layout

What is the current challenge: Working on coding the webpage

Next plan:

-Assign tasks for each team member to work on their respective pages

-Work on presentation slides edits

Date: 3/19/2023

Time: 11am to 12pm

Who attended: Adrian Bernardino, Mark Laow, Stephania Rey

What you have discussed:

- Team provided updates on webpages (what is worked on and what hasn't)
-Schedule upcoming meetings with team for next week

-Select day for rehearsal practice –work on presentation slides

-Work on presentation slides

Next plan:

-Work on powerpoint slides and preparation

Date: 3/24/2023

Time: 11am to 12pm

Who attended: Adrian Bernardino, Mark Laow, Stephania Rey

What you have discussed:

-Team provided updates on webpages (what is worked on and what hasn't)
- worked on powerpoint slides

Next plan:

-Worked on presentations script
-Dcide on rehearsal date

Date: 3/30/2023

Time: 2:30pm to 5pm

Who attended: Adrian Bernardino, Mark Laow, Stephania Rey

What you have discussed:

-Team provided updates on webpages (what is worked on and what hasn't)
-Discuss next steps in project

Next plan:

-Work on the web pages

Date: 4/13/2023

Time: 2:30pm to 5pm

Who attended: Adrian Bernardino, Matt Roman, Stephania Rey

What you have discussed:

-Team mates reported updates

-Finalized report

Next plan:

-Work on pages to submit on Saturday

Date: 4/20/2023

Time: 2:30pm to 5pm

Who attended: Adrian Bernardino, Matt Roman, Stephania Rey

What you have discussed:

-Team mates reported updates

Updates:

Shopping cart page/account page-Adrian CSS stuff background

Implemented account page(change your info) - Matt

Work on page structure and layout for cds/dvds, records, merchandise, instruments-Steph

Next plan:

-Continue working on corresponding parts for each team member

-Consider ways to implement database

Date: 4/27/2023

Time: 2:30pm to 5pm

Who attended: Adrian Bernardino, Matt Roman,Mark Laow, Stephania Rey

What you have discussed:

-Team mates reported updates

Next plans:

-Discussed team's next plans and assigned tasks

-Continue database research

-Coordinate meeting times

-Assigned report and slides parts to each teammember

Date: 05/4/2023

Time: 2:30pm to 5pm

Who attended: Adrian Bernardino, Matt Roman,Mark Laow, Stephania Rey

What you have discussed:

-Team mates reported updates

-decided database implementation

-coordinated meeting times