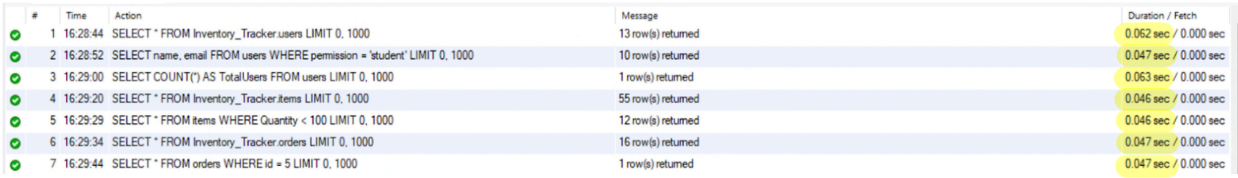
Database Validation

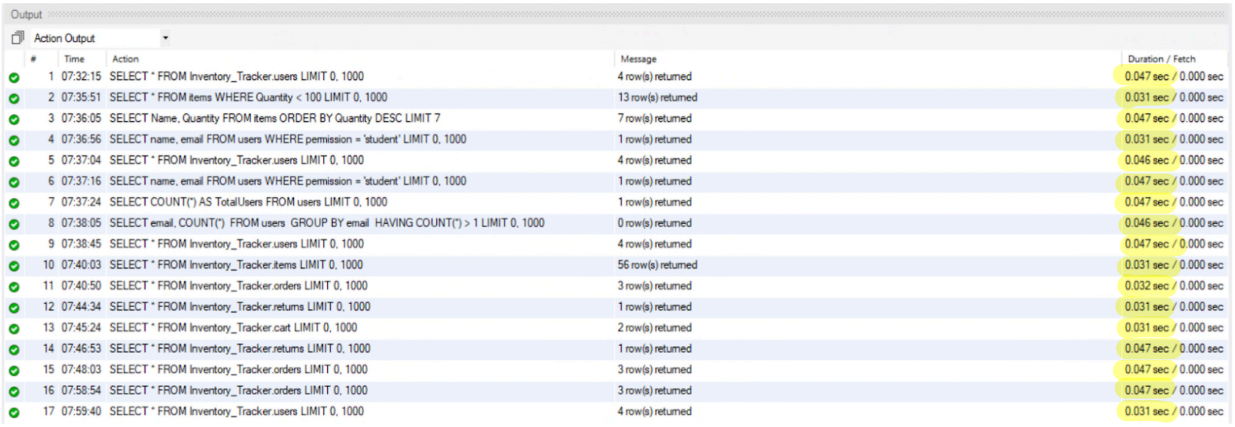
Requirement	**Target Specification**	**Method of Validation**
DB <-> Website Integration	100% data sync between UI and DB	Used Website UI to verify changes reflected in both (have a screen recording of it)
DB <-> Mobile App Integration	100% data sync between app and DB	Simulated transactions via mobile and confirmed DB updates (did it in person, it works w website, so only have a recording of the website)
Insert/Update/Delete Accuracy	No corrupted or missing records	Ran CRUD operations; checked results with SELECT queries
Query Response Time	≤ 500 ms average	Logged execution time using MySQL Workbench
Backup Availability	Recovery points within a 3-day window	Checked AWS RDS snapshots for date range

- DB <-> Website/Mobile App Integration:
    - Have a screen recording of the website and DB

- Query Response Time:
  - 

#	Time	Action	Message	Duration / Fetch
1	16:28:44	SELECT * FROM Inventory_Tracker.users LIMIT 0, 1000	13 row(s) returned	0.062 sec / 0.000 sec
2	16:28:52	SELECT name, email FROM users WHERE permission = 'student' LIMIT 0, 1000	10 row(s) returned	0.047 sec / 0.000 sec
3	16:29:00	SELECT COUNT(*) AS TotalUsers FROM users LIMIT 0, 1000	1 row(s) returned	0.063 sec / 0.000 sec
4	16:29:20	SELECT * FROM Inventory_Tracker.items LIMIT 0, 1000	55 row(s) returned	0.046 sec / 0.000 sec
5	16:29:29	SELECT * FROM items WHERE Quantity < 100 LIMIT 0, 1000	12 row(s) returned	0.046 sec / 0.000 sec
6	16:29:34	SELECT * FROM Inventory_Tracker.orders LIMIT 0, 1000	16 row(s) returned	0.047 sec / 0.000 sec
7	16:29:44	SELECT * FROM orders WHERE id = 5 LIMIT 0, 1000	1 row(s) returned	0.047 sec / 0.000 sec

    Average: 0.358/7 = 0.05 sec



Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	07:32:15	SELECT * FROM Inventory_Tracker.users LIMIT 0, 1000	4 row(s) returned	0.047 sec / 0.000 sec
2	07:35:51	SELECT * FROM items WHERE Quantity < 100 LIMIT 0, 1000	13 row(s) returned	0.031 sec / 0.000 sec
3	07:36:05	SELECT Name, Quantity FROM items ORDER BY Quantity DESC LIMIT 7	7 row(s) returned	0.047 sec / 0.000 sec
4	07:36:56	SELECT name, email FROM users WHERE permission = 'student' LIMIT 0, 1000	1 row(s) returned	0.031 sec / 0.000 sec
5	07:37:04	SELECT * FROM Inventory_Tracker.users LIMIT 0, 1000	4 row(s) returned	0.046 sec / 0.000 sec
6	07:37:16	SELECT name, email FROM users WHERE permission = 'student' LIMIT 0, 1000	1 row(s) returned	0.047 sec / 0.000 sec
7	07:37:24	SELECT COUNT(*) AS TotalUsers FROM users LIMIT 0, 1000	1 row(s) returned	0.047 sec / 0.000 sec
8	07:38:05	SELECT email, COUNT(*) FROM users GROUP BY email HAVING COUNT(*) > 1 LIMIT 0, 1000	0 row(s) returned	0.046 sec / 0.000 sec
9	07:38:45	SELECT * FROM Inventory_Tracker.users LIMIT 0, 1000	4 row(s) returned	0.047 sec / 0.000 sec
10	07:40:03	SELECT * FROM Inventory_Tracker.items LIMIT 0, 1000	56 row(s) returned	0.031 sec / 0.000 sec
11	07:40:50	SELECT * FROM Inventory_Tracker.orders LIMIT 0, 1000	3 row(s) returned	0.032 sec / 0.000 sec
12	07:44:34	SELECT * FROM Inventory_Tracker.returns LIMIT 0, 1000	1 row(s) returned	0.031 sec / 0.000 sec
13	07:45:24	SELECT * FROM Inventory_Tracker.cart LIMIT 0, 1000	2 row(s) returned	0.031 sec / 0.000 sec
14	07:46:53	SELECT * FROM Inventory_Tracker.returns LIMIT 0, 1000	1 row(s) returned	0.047 sec / 0.000 sec
15	07:48:03	SELECT * FROM Inventory_Tracker.orders LIMIT 0, 1000	3 row(s) returned	0.047 sec / 0.000 sec
16	07:58:54	SELECT * FROM Inventory_Tracker.orders LIMIT 0, 1000	3 row(s) returned	0.047 sec / 0.000 sec
17	07:59:40	SELECT * FROM Inventory_Tracker.users LIMIT 0, 1000	4 row(s) returned	0.031 sec / 0.000 sec

Average: 0.686 / 17 = 0.04 sec

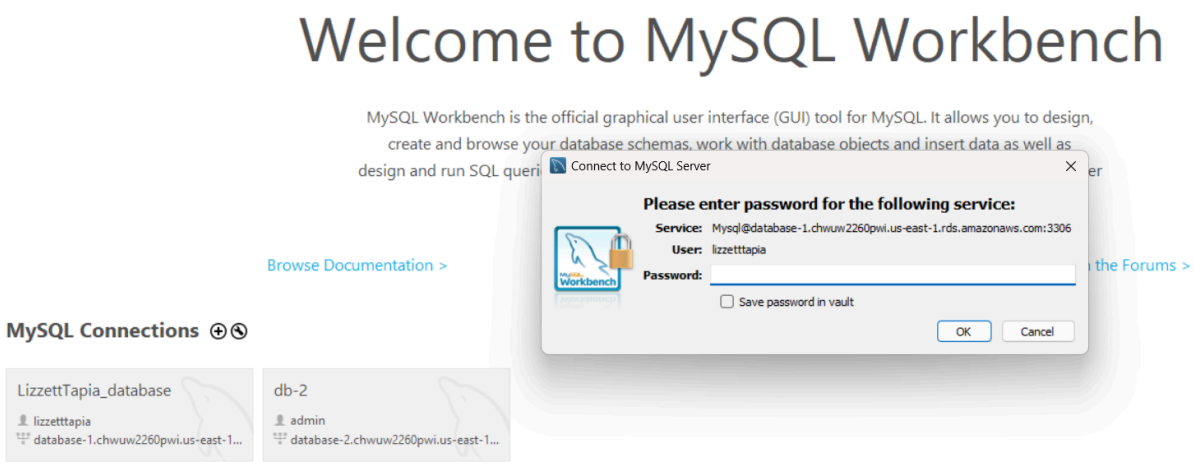
- Backup Availability: Insert Screenshot of AWS backup

Current Region backups (1) Info

Filter by current region backups

DB instance or cluster	Earliest restorable time	Latest restorable time
database-1	April 18, 2025, 01:41 (UTC-05:00)	April 20, 2025, 22:10 (UTC-05:00)

- Security:



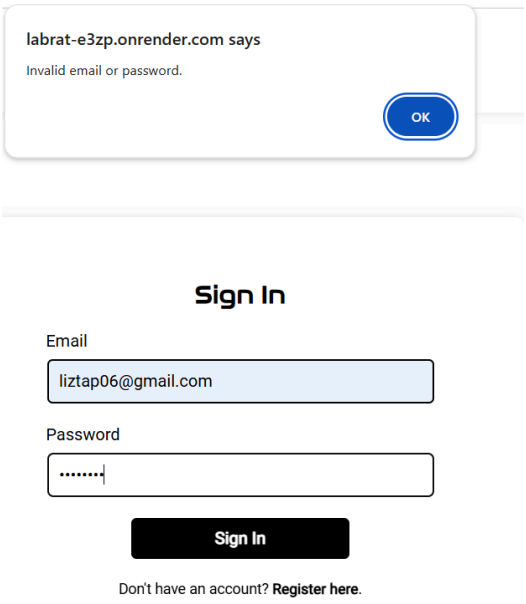
MySQL Python code asks for my specific AWS info

```python
mysql_password = getpass.getpass("Enter your MySQL password: ") #Get MySQL password securely from the user (for security purposes)

# Setup database connections
sqlite_engine = create_engine('sqlite:///inventoryTracker2.db') #create_engine(): helps create a connection to the SQLite database, and it points to
mysql_engine = create_engine(
    f'mysql+mysqlconnector://lizzetttapia:{mysql_password}@database-1.chwuw2260pwi.us-east-1.rds.amazonaws.com:3306/Inventory_Tracker' #connects to
)

#Create sessions, for both databases, for interacting w the SQLite and MySQL databases. Purpose of the session to create a connection to the databas
SQLiteSession = sessionmaker(bind=sqlite_engine) #telling SQLAlchemy that this session is sonnected to the SQLite database(can now produce sessions
sqlite_session = SQLiteSession() #the actual session object, which enables doing all the operations of adding, updating, or querying data in the SQL

MySQLSession = sessionmaker(bind=mysql_engine) #creating a session for the MySQL database, linking it to the mysql_engine
mysql_session = MySQLSession() #creating a new session for interaction w the MySQL database, which allows adding, querying, or updating data in the

#Ensure tables exist in both databases
Base.metadata.create_all(sqlite_engine) #creates all tables in the database if they dont already exists. specifies to create the tables in SQLite
Base.metadata.create_all(mysql_engine) #creates all tables in the database if they dont already exists. specifies to create the tables in MySQL

```

When code gets ran, it also asks for the password

```
PS C:\Users\lizta\OneDrive\Documents\ecen403> & C:/Users/lizta/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/lizta/OneDrive/Documents/ecen403/main.py
Enter your MySQL password:
Data transferred successfully from SQLite to MySQL.
```

- Deleting a user (my login no longer works)



- Insert/Update/Delete Accuracy: 100%

Find orders by a specific user

```sql
/*find orders by a specific user*/
SELECT * FROM orders WHERE id = 5; /*can change the id number
```

id	item_id	item_name	returnable	quantity	user_name	user_email	timestamp
5	13	Diodes (Schottky)	0	40	Jonathan Moore	jonbakermoore@gmail.com	2025-04-21 13:08:37
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Count total orders

```sql
/*count total orders*/
SELECT COUNT(*) AS TotalOrders FROM orders;
```

TotalOrders
11

Adding a new user

before

id	email	permission	password	name	color	last_login	last_logout
1	jonbakermoore@gmail.com	admin	$2b$10$TZaLtjd6CZ95Jq4EuuKPpOKwiO0TsQy...	Jonathan Moore	#500000	2025-04-21 13:27:39	2025-04-21 13:29:33
2	jonbmoore@tamu.edu	professor	$2b$10$HFyUZb5uCgdYfIomwOiI8utM7KADl3hJ...	Professor Jonny	#1e00ff	2025-04-21 13:29:43	2025-04-21 09:12:48
3	jonathancompeng@gmail.com	student	$2b$10$pYyYjXW36R9TK6dbrL1QQesqyCzWdM...	Student Jonny	#8400ff	2025-04-21 08:06:34	2025-04-21 08:06:42
4	test@test.com	admin	$2b$10$9.1YN2B17ln.i4mUKRTX5uWy61KIm/cm...	test	#6f6f6f	2025-04-21 08:29:22	2025-04-21 08:29:27
5	e@e.com	student	$2a$10$cZFSpk/qIYiDGf2/VQ2enO7mJ5mXxiOc...	Evan	#6f6f6f	2025-04-21 13:20:25	2025-04-21 13:20:25
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

→ after

id	email	permission	password	name	color	last_login	last_logout
2	jonbmoore@tamu.edu	professor	$2b$10$HFyUZb5uCgdYfIomwOiI8utM7KADl3hJ...	Professor Jonny	#1e00ff	2025-04-21 13:29:43	2025-04-21 09:12:48
3	jonathancompeng@gmail.com	student	$2b$10$pYyYjXW36R9TK6dbrL1QQesqyCzWdM...	Student Jonny	#8400ff	2025-04-21 08:06:34	2025-04-21 08:06:42
4	test@test.com	admin	$2b$10$9.1YN2B17ln.i4mUKRTX5uWy61KIm/cm...	test	#6f6f6f	2025-04-21 08:29:22	2025-04-21 08:29:27
5	e@e.com	admin	$2a$10$cZFSpk/qIYiDGf2/VQ2enO7mJ5mXxiOc...	Evan	#6f6f6f	2025-04-21 13:20:25	2025-04-21 13:20:25
6	liztap06@gmail.com	student	$2b$10$HPciuGYdl0iZEOu.ixud0uieD9/aIkpdvP0...	Liz	#6f6f6f	2025-04-21 13:52:27	2025-04-21 13:52:27
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Finding users with a specific role

```sql
/*finding users with a specific role*/
SELECT name, email
FROM users
WHERE permission = 'admin'; /*can change the role*/
```

name	email
Jonathan Moore	jonbakermoore@gmail.com
test	test@test.com
Evan	e@e.com
Liz	liztap06@gmail.com

Count the number of users

```sql
/*count the number of users*/
SELECT COUNT(*) AS TotalUsers FROM users;

/*find users w duplicate emails*/
```

TotalUsers
6

Selecting certain items where quantity is a certain number

```sql
/*selecting certain items where quantity is a certain number*/
SELECT * FROM items
WHERE Quantity < 100; /*can change the quantity*/

/*pulling a certain amount of items and counting their quantities and putting in descending order*/
```

Id	Name	Description	CategoryID	Quantity	Unit	Location	Supplier	re
1	Resistors (1kΩ)	A 1kΩ for basic electronic components	Basic Electronic Components	66	kits	Bin C	Digikey Electronics	0
7	Inductors (10µH)	A 10µH for basic electronic components	Basic Electronic Components	0	kits	Bin A	Farnell (Newark)	0
12	Diodes (LEDs)	A LEDs for basic electronic components	Basic Electronic Components	50	pieces	Bin B	Texas Instruments	0
21	Op-Amps (TL081)	A TL081 for integrated circuits	Integrated Circuits	29	units	Bin B	Farnell (Newark)	0
25	Logic Gates (OR)	A OR for integrated circuits	Integrated Circuits	99	pieces	Bin B	Mouser Electronics	0
27	Logic Gates (NAND)	A NAND for integrated circuits	Integrated Circuits	75	pieces	Bin B	Mouser Electronics	0
34	Batteries (AA)	A AA for power components	Power Components	33	units	Bin A	Farnell (Newark)	0
42	Multimeters (Digital Multimeter)	A Digital Multimeter for measurement and testin...	Measurement and Testing Equipment	35	kits	Checkout Area	Texas Instruments	1
48	Breadboards (Large)	A Large for prototyping and development tools	Prototyping and Development Tools	92	units	Checkout Area	Mouser Electronics	0

Updating quantity of a certain item

before

```sql
/*updating the quantity of a certain item*/
UPDATE items
SET Quantity = 100 /*can change the quantity*/
WHERE Id = 1; /*can change the Id num*/
```

Id	Name	Description	CategoryID	Quantity	Unit	Location	Supplier	returnable
1	Resistors (1kΩ)	A 1kΩ for basic electronic components	Basic Electronic Components	66	kits	Bin C	Digikey Electronics	0

→ after

```sql
/*updating the quantity of a certain item*/
UPDATE items
SET Quantity = 100 /*can change the quantity*/
WHERE Id = 1; /*can change the Id num*/
```

Id	Name	Description	CategoryID	Quantity	Unit	Location	Supplier	returnable
1	Resistors (1kΩ)	A 1kΩ for basic electronic components	Basic Electronic Components	100	kits	Bin C	Digikey Electronics	0