PES UNIVERSITY, Bengaluru
Department of Computer Science and Engineering
B. Tech (CSE) – 5th Semester – Aug-Dec 2024

UE22CS351A – Database Management System

PROJECT report on

TIME CAPSULE DATABASE SYSTEM

PES1UG22CS343	**Meeraja K**
PES1UG22CS314	**M S Chandana**

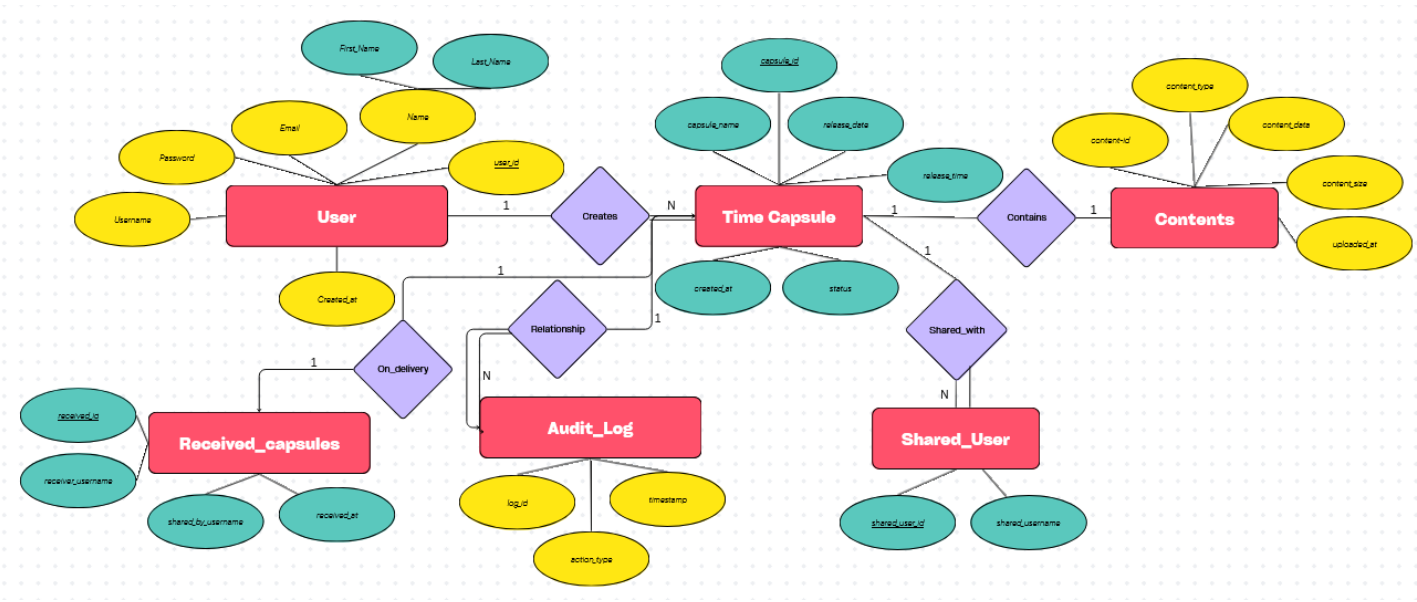
Description about the statement (Short abstract):

TimeLock is a secure and user-friendly database system designed to allow individuals to create and store digital time capsules, containing only image files (PNG, JPG, GIF). At any given time, only one user can create a time capsule, ensuring exclusive control over the creation process. The system allows users to seal capsules with a specified release date and time, preserving the content until its designated moment.

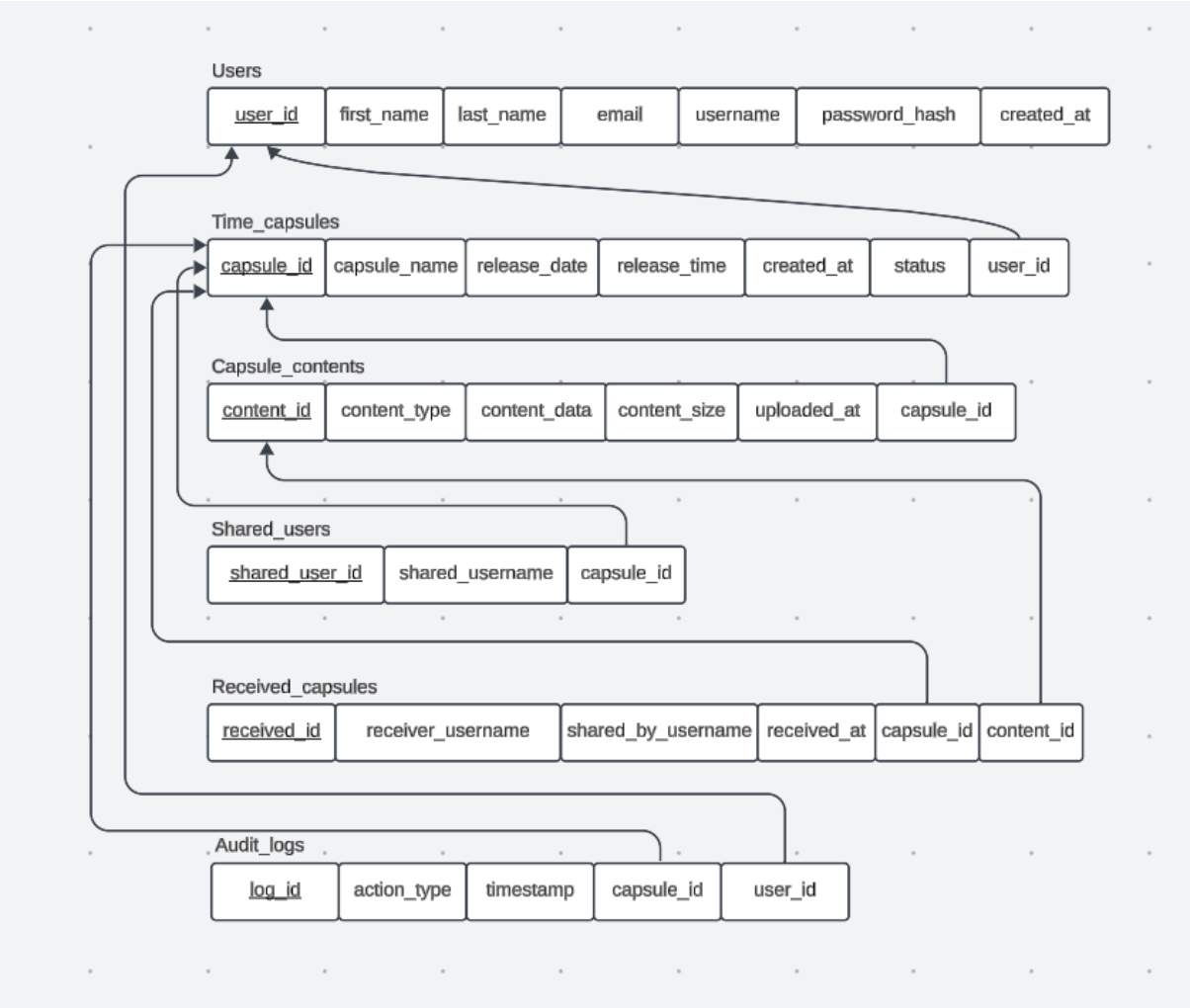
The platform supports role-based access control, where administrators can oversee system activity without compromising user privacy. Audit logging ensures accountability, tracking every user action. Additionally, TimeLock features content validation mechanisms, ensuring that only supported file types are saved and stored securely.

By blending nostalgia with modern technology, TimeLock offers a reliable platform for users to securely preserve memories, enabling future access to their time capsules.

ER - Diagram



Relational Schema



CRUD Operations

CREATE COMMAND

Tables

```
-- Users table creation
CREATE TABLE `users` (
  `user_id` int NOT NULL AUTO_INCREMENT,
  `first_name` varchar(50) NOT NULL,
  `last_name` varchar(50) NOT NULL,
  `email` varchar(100) NOT NULL,
  `username` varchar(50) NOT NULL,
  `password_hash` varchar(255) NOT NULL,
  `created_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`user_id`),
  UNIQUE KEY `email` (`email`),
  UNIQUE KEY `username` (`username`)
)
```

timecapsuledbs.users

Info | Columns | Indexes | Triggers | Foreign keys | Partitions | Grants | DDL

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
created_at	timestamp	CURRENT_TIMESTA...	YES			select,insert,update,references
email	varchar(100)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
first_name	varchar(50)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
last_name	varchar(50)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
password_hash	varchar(255)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
user_id	int		NO			select,insert,update,references
username	varchar(50)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references

Count: 7 Refresh

```
-- Time capsules table
CREATE TABLE `time_capsules` (
  `capsule_id` int NOT NULL AUTO_INCREMENT,
  `user_id` int NOT NULL,
  `capsule_name` varchar(100) NOT NULL,
  `release_date` date NOT NULL,
  `release_time` time NOT NULL,
  `created_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `status` enum('scheduled','delivered','failed') NOT NULL DEFAULT 'scheduled',
  PRIMARY KEY (`capsule_id`),
  UNIQUE KEY `user_id` (`user_id`,`capsule_name`),
  CONSTRAINT `time_capsules_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`user_id`) ON
DELETE CASCADE
)
```

timecapsuledbs.time_capsules

Info | Columns | Indexes | Triggers | Foreign keys | Partitions | Grants | DDL

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
capsule_id	int		NO			select,insert,update,references
capsule_name	varchar(100)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
created_at	timestamp	CURRENT_TIMESTA...	YES			select,insert,update,references
release_date	date		NO			select,insert,update,references
release_time	time		NO			select,insert,update,references
status	enum('scheduled','d...	scheduled	NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
user_id	int		NO			select,insert,update,references

Count: 7 Refresh

```
-- Capsule_contents table
CREATE TABLE `capsule_contents` (
  `content_id` int NOT NULL AUTO_INCREMENT,
  `capsule_id` int NOT NULL,
  `content_type` enum('text','image') NOT NULL,
  `content_data` longblob NOT NULL,
  `content_size` int NOT NULL,
  `uploaded_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`content_id`),
  KEY `capsule_id` (`capsule_id`),
  CONSTRAINT `capsule_contents_ibfk_1` FOREIGN KEY (`capsule_id`) REFERENCES `time_capsules`
(`capsule_id`) ON DELETE CASCADE,
  CONSTRAINT `capsule_contents_chk_1` CHECK ((`content_size` <= 52428800))
)
```

timecapsuledbs.capsule_contents

Info | Columns | Indexes | Triggers | Foreign keys | Partitions | Grants | DDL

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
capsule_id	int		NO			select,insert,update,references
content_data	longblob		NO			select,insert,update,references
content_id	int		NO			select,insert,update,references
content_size	int		NO			select,insert,update,references
content_type	enum('text','image')		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
uploaded_at	timestamp	CURRENT_TIMESTA...	YES			select,insert,update,references

Count: 6 Refresh

```
-- Shared_users table
CREATE TABLE `shared_users` (
  `shared_user_id` int NOT NULL AUTO_INCREMENT,
  `capsule_id` int NOT NULL,
  `shared_username` varchar(50) NOT NULL,
  PRIMARY KEY (`shared_user_id`),
  KEY `capsule_id` (`capsule_id`),
  CONSTRAINT `shared_users_ibfk_1` FOREIGN KEY (`capsule_id`) REFERENCES `time_capsules`
(`capsule_id`) ON DELETE CASCADE
)
```

timecapsuledbs.shared_users

Info | Columns | Indexes | Triggers | Foreign keys | Partitions | Grants | DDL

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
capsule_id	int		NO			select,insert,update,references
shared_user_id	int		NO			select,insert,update,references
shared_username	varchar(50)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references

Count: 3 Refresh

```
-- Received_capsules table
CREATE TABLE `received_capsules` (
  `received_id` int NOT NULL AUTO_INCREMENT,
  `capsule_id` int NOT NULL,
  `receiver_username` varchar(50) NOT NULL,
  `shared_by_username` varchar(50) NOT NULL,
  `content_id` int NOT NULL,
  `received_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`received_id`),
  UNIQUE KEY `capsule_receiver_unique` (`capsule_id`,`receiver_username`),
  KEY `capsule_id` (`capsule_id`),
  KEY `content_id` (`content_id`),
  CONSTRAINT `received_capsules_ibfk_1` FOREIGN KEY (`capsule_id`) REFERENCES `time_capsules`
(`capsule_id`) ON DELETE CASCADE,
  CONSTRAINT `received_capsules_ibfk_2` FOREIGN KEY (`content_id`) REFERENCES `capsule_contents`
(`content_id`) ON DELETE CASCADE
)
```

timecapsuledbs.received_capsul...

Info | Columns | Indexes | Triggers | Foreign keys | Partitions | Grants | DDL

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
capsule_id	int		NO			select,insert,update,references
content_id	int		NO			select,insert,update,references
received_at	timestamp	CURRENT_TIMESTA...	YES			select,insert,update,references
received_id	int		NO			select,insert,update,references
receiver_username	varchar(50)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
shared_by_username	varchar(50)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references

Count: 6 Refresh

```
-- Audit_logs table
CREATE TABLE `audit_logs` (
  `log_id` int NOT NULL AUTO_INCREMENT,
  `user_id` int NOT NULL,
  `action_type` varchar(50) NOT NULL,
  `capsule_id` int DEFAULT NULL,
  `timestamp` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`log_id`),
  KEY `user_id` (`user_id`),
  KEY `capsule_id` (`capsule_id`),
  CONSTRAINT `audit_logs_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`user_id`),
  CONSTRAINT `audit_logs_ibfk_2` FOREIGN KEY (`capsule_id`) REFERENCES `time_capsules`
(`capsule_id`)
)
```

timecapsuledbs.audit_logs

Info | Columns | Indexes | Triggers | Foreign keys | Partitions | Grants | DDL

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
action_type	varchar(50)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
capsule_id	int		YES			select,insert,update,references
log_id	int		NO			select,insert,update,references
timestamp	timestamp	CURRENT_TIMESTA...	YES			select,insert,update,references
user_id	int		NO			select,insert,update,references

Count: 5 Refresh

Explanation: These SQL scripts create a relational database schema for a time capsule application, defining tables for users, time capsules, their contents, shared and received capsules, and audit logs. The schema ensures data integrity with primary keys, foreign keys, unique constraints, and size checks while maintaining **3rd Normal Form (3NF)**. Each table's attributes depend only on the primary key, ensuring no redundancy or partial/transitive dependencies, which makes the design efficient and normalized.

CREATE USERS WITH VARIED PRIVILEDGES:

```
-- Create an Admin User (Read-Only Access)
CREATE USER 'admin'@'localhost' IDENTIFIED WITH mysql_native_password BY 'admin123';
-- Create a Regular User (Full Access)
CREATE USER 'user'@'localhost' IDENTIFIED WITH mysql_native_password BY 'user_password';

GRANT SELECT ON timecapsuledbs.* TO 'admin'@'localhost';
GRANT ALL PRIVILEGES ON timecapsuledbs.* TO 'user'@'localhost';
```

Purpose: These codes ensures that the admin user cannot perform INSERT, UPDATE, or DELETE operations while the regular user has full CRUD capabilities.

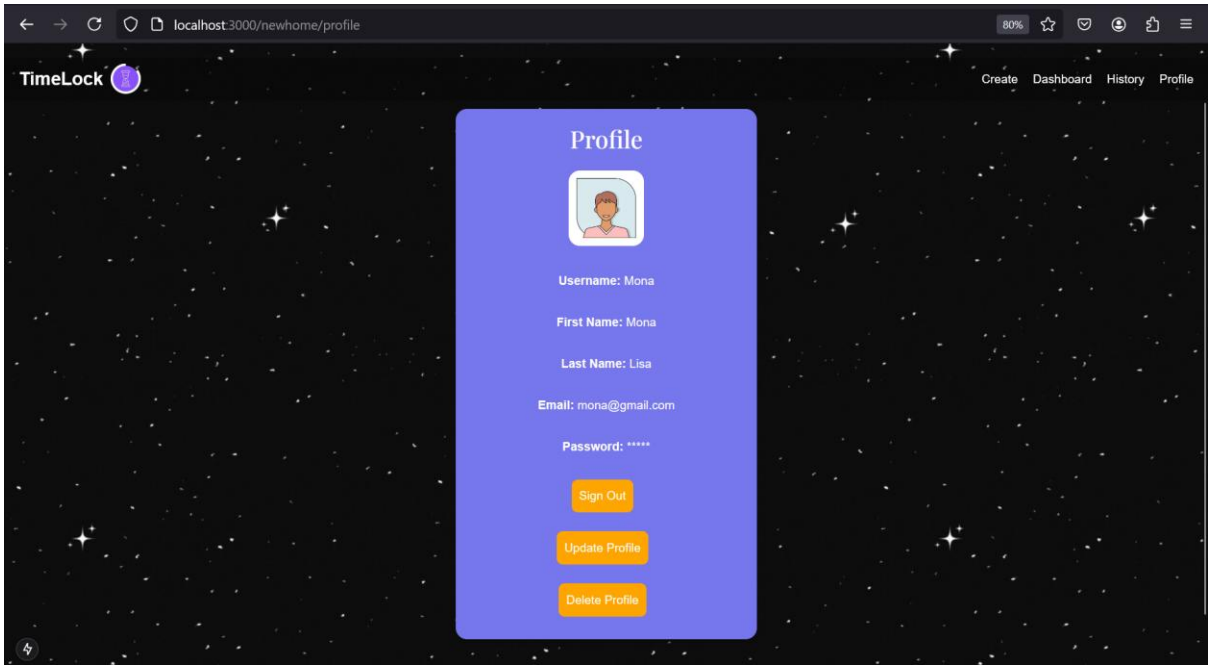
READ OPERATION

Users table:

```
mysql> select * from users;
+---------+------------+-----------+--------------------+----------+---------------+---------------------+
| user_id | first_name | last_name | email              | username | password_hash | created_at          |
+---------+------------+-----------+--------------------+----------+---------------+---------------------+
|       1 | Admin      | Main      | admin@gmail.com    | admin    | admin123      | 2024-11-17 12:50:42 |
|       2 | Sasha      | Neil      | sasha@gmail.com    | Sash     | sasha         | 2024-11-07 21:41:06 |
|       3 | Myla       | Sen       | myla12@gmail.com   | myla     | myla          | 2024-11-08 14:22:05 |
|       6 | Keem       | Krys      | keemy@gmail.com    | Keem     | keemy123      | 2024-11-10 08:02:24 |
|       8 | Akira      | Raj       | akiraraj@gmail.com | Akira    | akira         | 2024-11-10 20:12:37 |
|       9 | Disha      | Pandit    | disha@gmail.com    | Dishhaa  | disha123      | 2024-11-13 18:28:21 |
|      10 | Taylor     | Swift     | taylor@gmail.com   | Taylor   | taylor        | 2024-11-16 21:51:30 |
|      12 | Mona       | Lisa      | mona@gmail.com     | Mona     | mona          | 2024-11-19 21:21:50 |
+---------+------------+-----------+--------------------+----------+---------------+---------------------+
8 rows in set (0.00 sec)
```

Fetching user's profile from database and displaying it in the website:

*"SELECT * FROM users WHERE username = ?"*



From this we can see that Mona's profile is read from the backend.

Reading capsule contents of delivered capsules:

```
SELECT capsule_id
    FROM time_capsules
    WHERE capsule_name = ? AND status = 'delivered'

SELECT content_type, TO_BASE64(content_data) AS content_data
    FROM capsule_contents
    WHERE capsule_id = ?
```

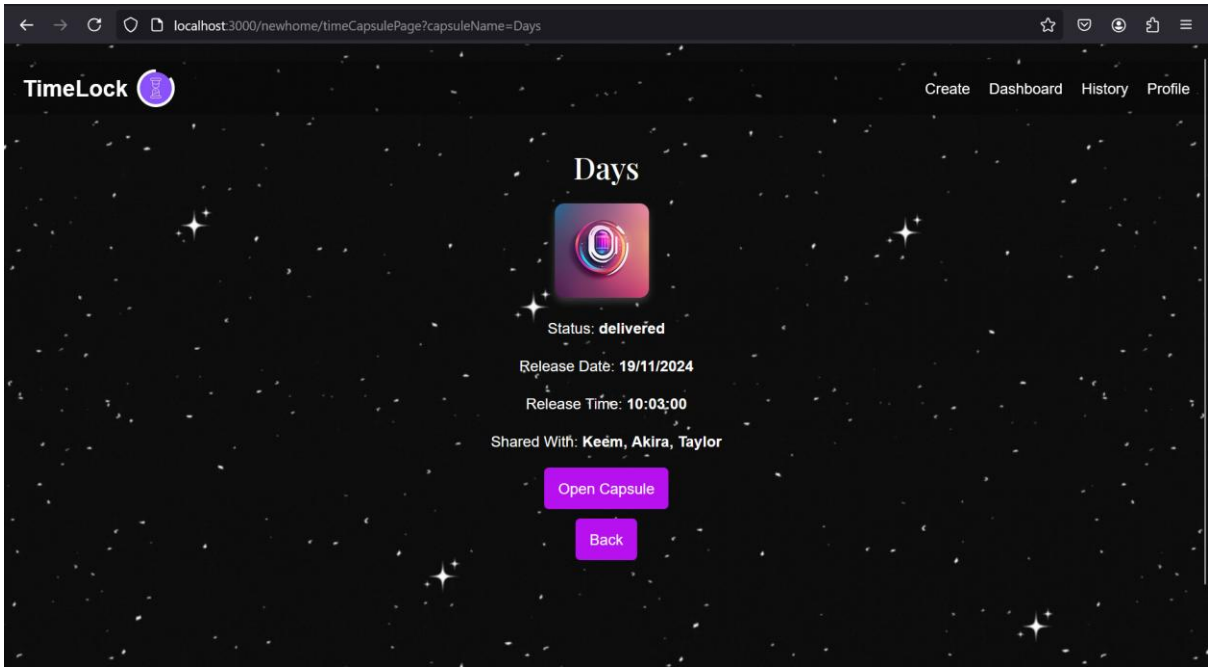
capsule_contents

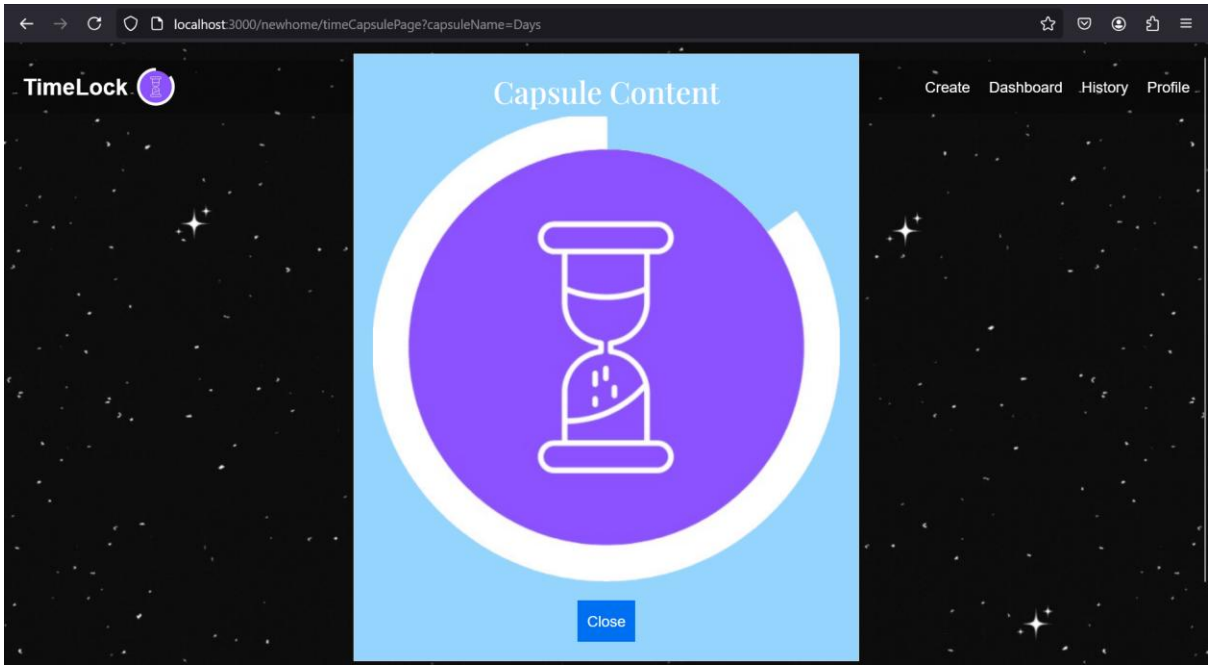
Limit to 1000 rows

```
1 •  SELECT * FROM timecapsuledbs.capsule_contents;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

content_id	capsule_id	content_type	content_data	content_size	uploaded_at
29	45	image	BLOB	3993	2024-11-16 22:39:03
30	46	image	BLOB	107988	2024-11-16 22:49:39
31	47	text	BLOB	21	2024-11-17 01:11:20
32	48	image	BLOB	320513	2024-11-17 01:13:06
33	49	image	BLOB	1854076	2024-11-17 01:14:43
34	50	text	BLOB	11	2024-11-17 01:24:40
35	51	text	BLOB	44	2024-11-17 01:27:01
36	55	image	BLOB	3993	2024-11-19 09:41:03
37	57	image	BLOB	107988	2024-11-19 10:01:17





Capsule contents are read from backend and displayed in the website.

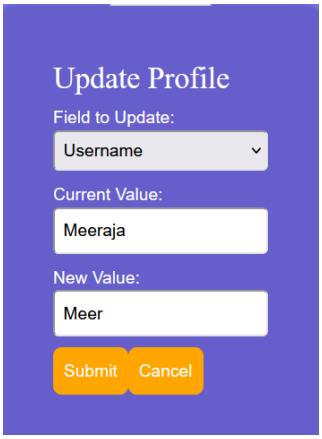
UPDATE OPERATION:

UPDATE users SET ${field} = ? WHERE user_id = ?

Before:

```
mysql> select * from users;
+---------+------------+-----------+---------------------+----------+---------------+---------------------+
| user_id | first_name | last_name | email               | username | password_hash | created_at          |
+---------+------------+-----------+---------------------+----------+---------------+---------------------+
|       1 | Admin      | Main      | admin@gmail.com     | admin    | admin123      | 2024-11-17 12:50:42 |
|       2 | Sasha      | Neil      | sasha@gmail.com     | Sash     | sasha         | 2024-11-07 21:41:06 |
|       3 | Myla       | Sen       | myla12@gmail.com    | myla     | myla          | 2024-11-08 14:22:05 |
|       6 | Keem       | Krys      | keemy@gmail.com     | Keem     | keemy123      | 2024-11-10 08:02:24 |
|       8 | Akira      | Raj       | akiraraj@gmail.com  | Akira    | akira         | 2024-11-10 20:12:37 |
|       9 | Disha      | Pandit    | disha@gmail.com     | Dishhaa  | disha123      | 2024-11-13 18:28:21 |
|      10 | Taylor     | Swift     | taylor@gmail.com    | Taylor   | taylor        | 2024-11-16 21:51:30 |
|      11 | Meeraja    | K         | meeraja.kn@gmail.com | Meeraja  | #InsideOut    | 2024-11-18 22:39:08 |
+---------+------------+-----------+---------------------+----------+---------------+---------------------+
8 rows in set (0.00 sec)
```

Updating profile details in the frontend changes them in the backend too.



After

```
mysql> select * from users;
+---------+------------+-----------+---------------------+----------+---------------+---------------------+
| user_id | first_name | last_name | email               | username | password_hash | created_at          |
+---------+------------+-----------+---------------------+----------+---------------+---------------------+
|       1 | Admin      | Main      | admin@gmail.com     | admin    | admin123      | 2024-11-17 12:50:42 |
|       2 | Sasha      | Neil      | sasha@gmail.com     | Sash     | sasha         | 2024-11-07 21:41:06 |
|       3 | Myla       | Sen       | myla12@gmail.com    | myla     | myla          | 2024-11-08 14:22:05 |
|       6 | Keem       | Krys      | keemy@gmail.com     | Keem     | keemy123      | 2024-11-10 08:02:24 |
|       8 | Akira      | Raj       | akiraraj@gmail.com  | Akira    | akira         | 2024-11-10 20:12:37 |
|       9 | Disha      | Pandit    | disha@gmail.com     | Dishhaa  | disha123      | 2024-11-13 18:28:21 |
|      10 | Taylor     | Swift     | taylor@gmail.com    | Taylor   | taylor        | 2024-11-16 21:51:30 |
|      11 | Meeraja    | K         | meeraja.kn@gmail.com | Meer     | #InsideOut    | 2024-11-18 22:39:08 |
+---------+------------+-----------+---------------------+----------+---------------+---------------------+
8 rows in set (0.00 sec)
```

Here, the username of user_id 11 was changed from Meeraja to Meer

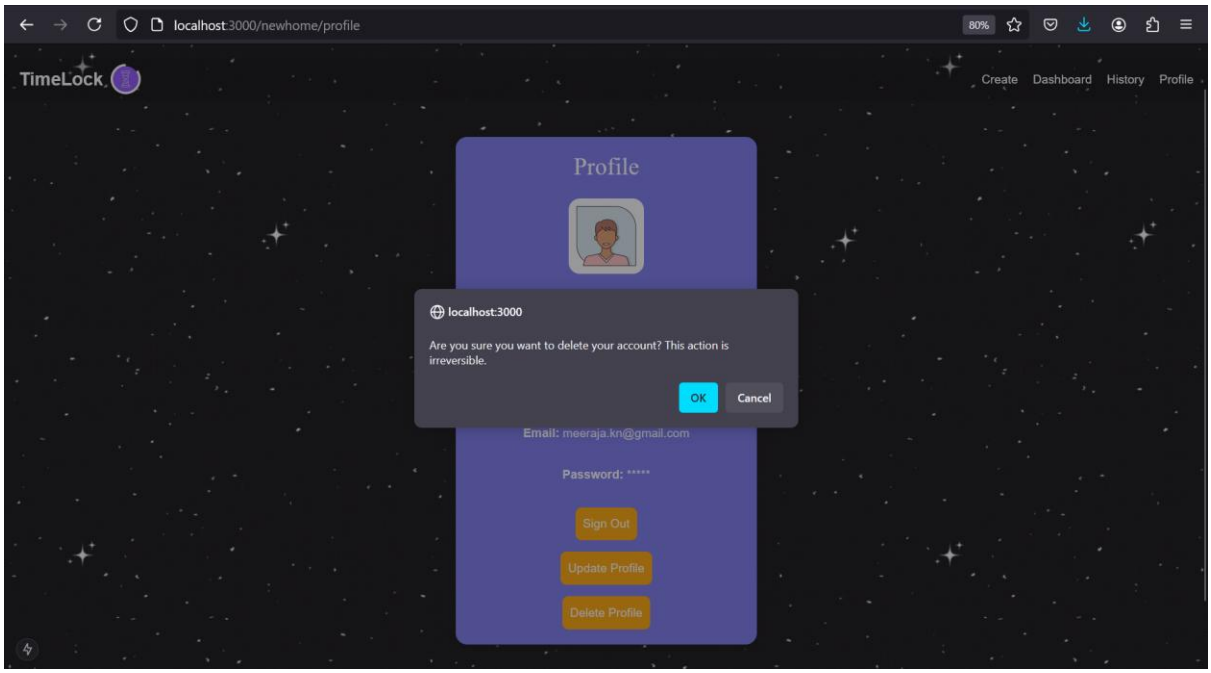
DELETE OPERATION:

Deleting user profile from the frontend deletes all users' related info from all tables.

DELETE FROM users WHERE username = ?", [username]

Before:

```
mysql> select * from users;
+---------+------------+-----------+---------------------+----------+---------------+---------------------+
| user_id | first_name | last_name | email               | username | password_hash | created_at          |
+---------+------------+-----------+---------------------+----------+---------------+---------------------+
|       1 | Admin      | Main      | admin@gmail.com     | admin    | admin123      | 2024-11-17 12:50:42 |
|       2 | Sasha      | Neil      | sasha@gmail.com     | Sash     | sasha         | 2024-11-07 21:41:06 |
|       3 | Myla       | Sen       | myla12@gmail.com    | myla     | myla          | 2024-11-08 14:22:05 |
|       6 | Keem       | Krys      | keemy@gmail.com     | Keem     | keemy123      | 2024-11-10 08:02:24 |
|       8 | Akira      | Raj       | akiraraj@gmail.com  | Akira    | akira         | 2024-11-10 20:12:37 |
|       9 | Disha      | Pandit    | disha@gmail.com     | Dishhaa  | disha123      | 2024-11-13 18:28:21 |
|      10 | Taylor     | Swift     | taylor@gmail.com    | Taylor   | taylor        | 2024-11-16 21:51:30 |
|      11 | Meeraja    | K         | meeraja.kn@gmail.com | Meer     | #InsideOut    | 2024-11-18 22:39:08 |
+---------+------------+-----------+---------------------+----------+---------------+---------------------+
8 rows in set (0.00 sec)
```



After:

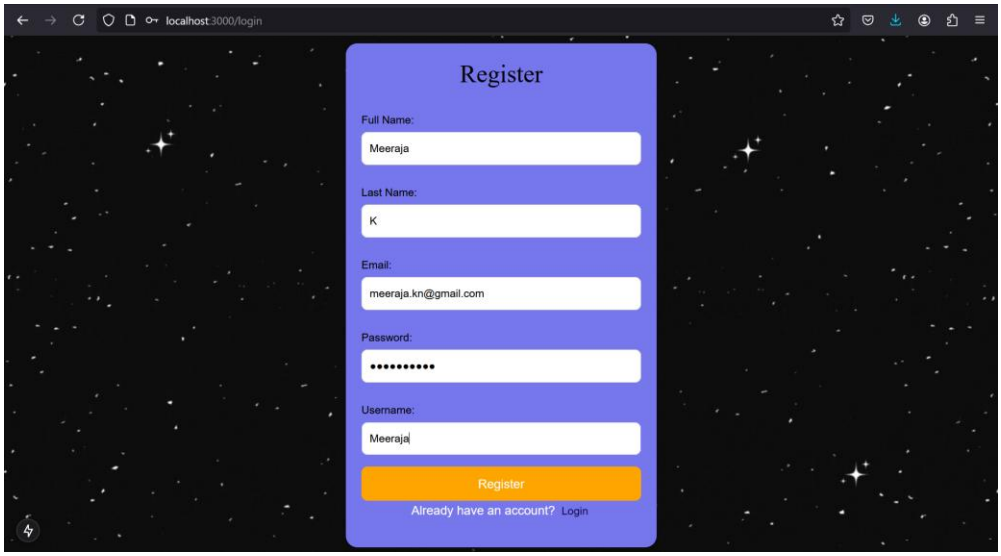
```
mysql> select * from users;
+---------+------------+-----------+--------------------+----------+---------------+---------------------+
| user_id | first_name | last_name | email              | username | password_hash | created_at          |
+---------+------------+-----------+--------------------+----------+---------------+---------------------+
|       1 | Admin      | Main      | admin@gmail.com    | admin    | admin123      | 2024-11-17 12:50:42 |
|       2 | Sasha      | Neil      | sasha@gmail.com    | Sash     | sasha         | 2024-11-07 21:41:06 |
|       3 | Myla       | Sen       | myla12@gmail.com   | myla     | myla          | 2024-11-08 14:22:05 |
|       6 | Keem       | Krys      | keemy@gmail.com    | Keem     | keemy123      | 2024-11-10 08:02:24 |
|       8 | Akira      | Raj       | akiraraj@gmail.com | Akira    | akira         | 2024-11-10 20:12:37 |
|       9 | Disha      | Pandit    | disha@gmail.com    | Dishhaa  | disha123      | 2024-11-13 18:28:21 |
|      10 | Taylor     | Swift     | taylor@gmail.com   | Taylor   | taylor        | 2024-11-16 21:51:30 |
+---------+------------+-----------+--------------------+----------+---------------+---------------------+
7 rows in set (0.00 sec)
```

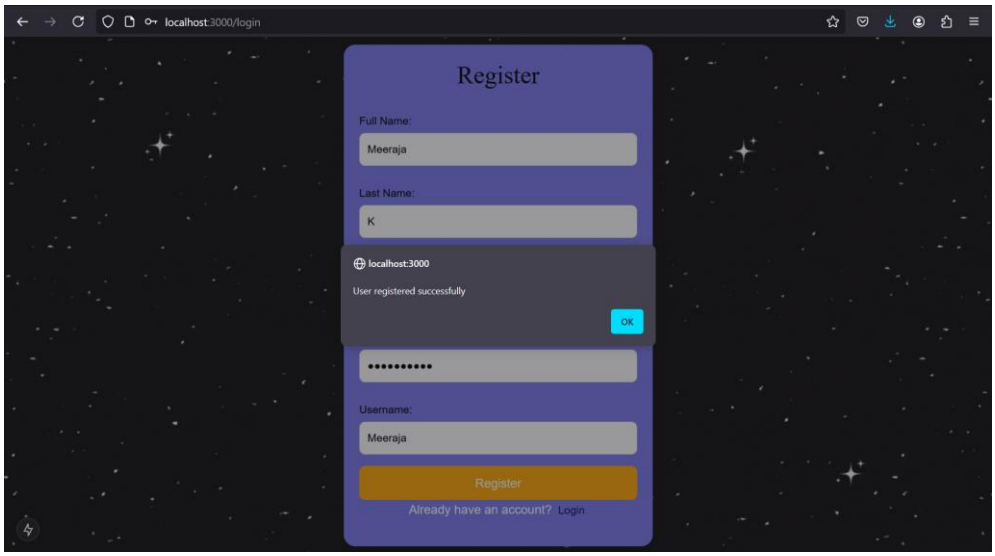
Here, the User with user_id = 11, Username: Meer was deleted.

INSERT OPERATION:

Inserting a new user entry to Users table:

INSERT INTO users (first_name, last_name, email, username, password_hash) VALUES (?, ?, ?, ?, ?)





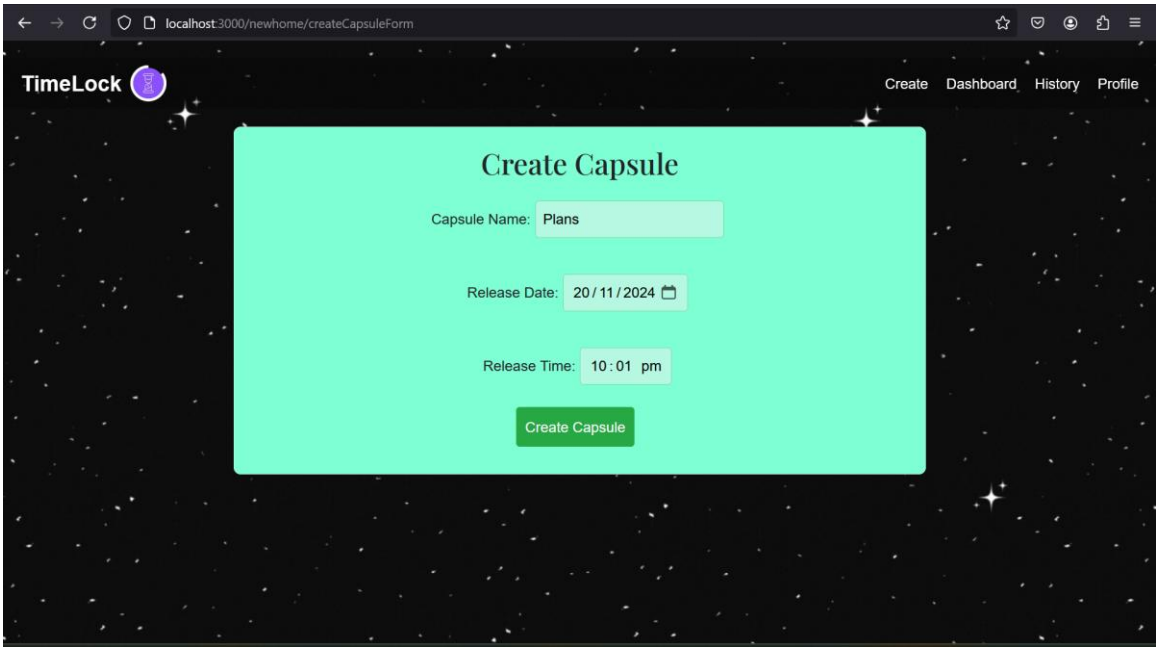
```
mysql> select * from users;
+---------+------------+-----------+---------------------+----------+---------------+---------------------+
| user_id | first_name | last_name | email               | username | password_hash | created_at          |
+---------+------------+-----------+---------------------+----------+---------------+---------------------+
|       1 | Admin      | Main      | admin@gmail.com     | admin    | admin123      | 2024-11-17 12:50:42 |
|       2 | Sasha      | Neil      | sasha@gmail.com     | Sash     | sasha         | 2024-11-07 21:41:06 |
|       3 | Myla       | Sen       | myla12@gmail.com    | myla     | myla          | 2024-11-08 14:22:05 |
|       6 | Keem       | Krys      | keemy@gmail.com     | Keem     | keemy123      | 2024-11-10 08:02:24 |
|       8 | Akira      | Raj       | akiraraj@gmail.com  | Akira    | akira         | 2024-11-10 20:12:37 |
|       9 | Disha      | Pandit    | disha@gmail.com     | Dishhaa  | disha123      | 2024-11-13 18:28:21 |
|      10 | Taylor     | Swift     | taylor@gmail.com    | Taylor   | taylor        | 2024-11-16 21:51:30 |
|      11 | Meeraja    | K         | meeraja.kn@gmail.com | Meeraja  | #InsideOut    | 2024-11-18 22:39:08 |
+---------+------------+-----------+---------------------+----------+---------------+---------------------+
8 rows in set (0.00 sec)
```

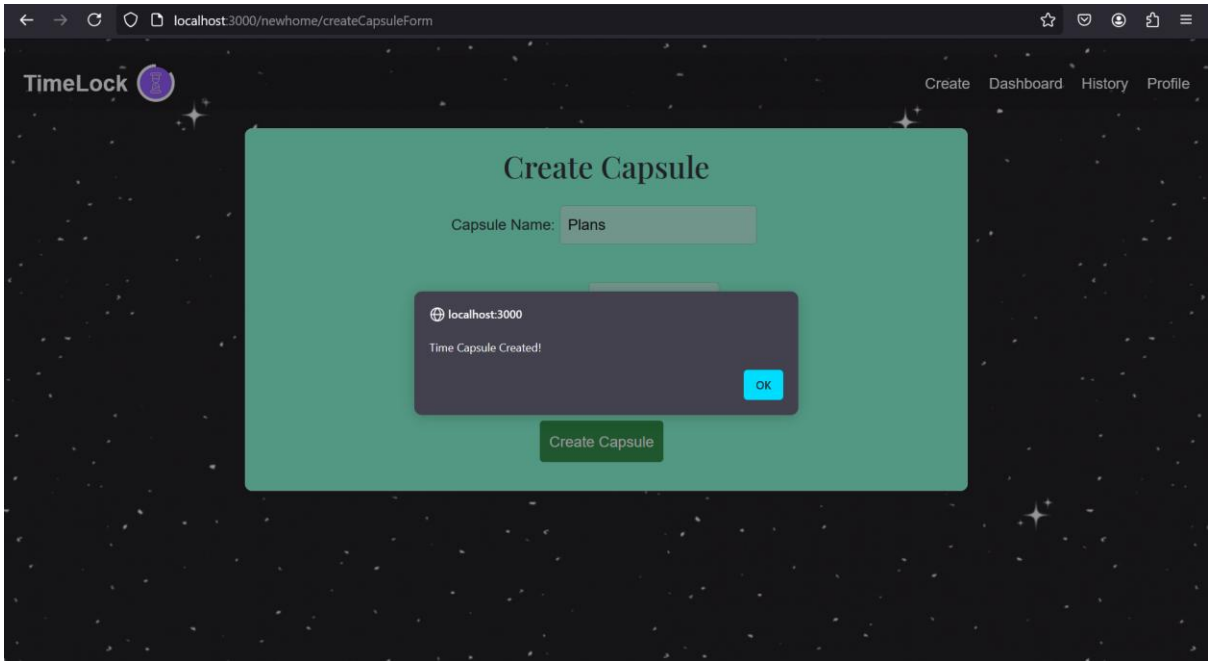
Inserting an entry to time capsules table:

INSERT INTO time_capsules (user_id, capsule_name, release_date, release_time, status)
VALUES (?, ?, ?, ?, ?)

Before:

```
mysql> select * from time_capsules;
+------------+---------+--------------+--------------+--------------+---------------------+-----------+
| capsule_id | user_id | capsule_name | release_date | release_time | created_at          | status    |
+------------+---------+--------------+--------------+--------------+---------------------+-----------+
|         45 |      10 | Cool         | 2024-11-16   | 22:40:00     | 2024-11-16 22:38:56 | delivered |
|         46 |       6 | Enable       | 2024-11-16   | 22:52:00     | 2024-11-16 22:49:32 | delivered |
|         47 |       6 | November     | 2024-11-17   | 01:13:00     | 2024-11-17 01:11:03 | delivered |
|         48 |       6 | Rain         | 2024-11-17   | 01:20:00     | 2024-11-17 01:12:43 | delivered |
|         49 |       6 | work         | 2024-11-17   | 01:20:00     | 2024-11-17 01:14:32 | delivered |
|         50 |       6 | New year     | 2024-11-17   | 12:10:00     | 2024-11-17 01:17:54 | delivered |
|         51 |       6 | Project      | 2024-11-17   | 01:30:00     | 2024-11-17 01:26:42 | delivered |
|         52 |      10 | Weekend      | 2024-11-19   | 09:20:00     | 2024-11-19 09:18:42 | delivered |
|         53 |      10 | WeekendPlan  | 2024-11-19   | 09:30:00     | 2024-11-19 09:24:17 | delivered |
|         54 |      10 | Monday       | 2024-11-19   | 09:40:00     | 2024-11-19 09:35:42 | delivered |
|         55 |      10 | friends      | 2024-11-19   | 09:45:00     | 2024-11-19 09:40:53 | delivered |
|         56 |      10 | Tuesday      | 2024-11-19   | 10:00:00     | 2024-11-19 09:59:56 | delivered |
|         57 |      10 | Days         | 2024-11-19   | 10:03:00     | 2024-11-19 10:01:04 | delivered |
|         58 |      10 | Tests        | 2024-11-19   | 14:10:00     | 2024-11-19 14:07:15 | delivered |
|         59 |      12 | Memories     | 2024-11-19   | 21:30:00     | 2024-11-19 21:24:30 | delivered |
|         60 |      12 | HolidayPlan  | 2024-11-19   | 21:55:00     | 2024-11-19 21:51:01 | delivered |
|         61 |      12 | Videos       | 2024-11-19   | 22:18:00     | 2024-11-19 22:14:48 | delivered |
|         62 |      12 | Hello        | 2024-11-20   | 22:45:00     | 2024-11-19 23:14:47 | scheduled |
+------------+---------+--------------+--------------+--------------+---------------------+-----------+
18 rows in set (0.00 sec)
```





After:

```
mysql> select * from time_capsules;
+------------+---------+--------------+--------------+--------------+---------------------+-----------+
| capsule_id | user_id | capsule_name | release_date | release_time | created_at          | status    |
+------------+---------+--------------+--------------+--------------+---------------------+-----------+
|         45 |      10 | Cool         | 2024-11-16   | 22:40:00     | 2024-11-16 22:38:56 | delivered |
|         46 |       6 | Enable       | 2024-11-16   | 22:52:00     | 2024-11-16 22:49:32 | delivered |
|         47 |       6 | November     | 2024-11-17   | 01:13:00     | 2024-11-17 01:11:03 | delivered |
|         48 |       6 | Rain         | 2024-11-17   | 01:20:00     | 2024-11-17 01:12:43 | delivered |
|         49 |       6 | work         | 2024-11-17   | 01:20:00     | 2024-11-17 01:14:32 | delivered |
|         50 |       6 | New year     | 2024-11-17   | 12:10:00     | 2024-11-17 01:17:54 | delivered |
|         51 |       6 | Project      | 2024-11-17   | 01:30:00     | 2024-11-17 01:26:42 | delivered |
|         52 |      10 | Weekend      | 2024-11-19   | 09:20:00     | 2024-11-19 09:18:42 | delivered |
|         53 |      10 | WeekendPlan  | 2024-11-19   | 09:30:00     | 2024-11-19 09:24:17 | delivered |
|         54 |      10 | Monday       | 2024-11-19   | 09:40:00     | 2024-11-19 09:35:42 | delivered |
|         55 |      10 | friends      | 2024-11-19   | 09:45:00     | 2024-11-19 09:40:53 | delivered |
|         56 |      10 | Tuesday      | 2024-11-19   | 10:00:00     | 2024-11-19 09:59:56 | delivered |
|         57 |      10 | Days         | 2024-11-19   | 10:03:00     | 2024-11-19 10:01:04 | delivered |
|         58 |      10 | Tests        | 2024-11-19   | 14:10:00     | 2024-11-19 14:07:15 | delivered |
|         59 |      12 | Memories     | 2024-11-19   | 21:30:00     | 2024-11-19 21:24:30 | delivered |
|         60 |      12 | HolidayPlan  | 2024-11-19   | 21:55:00     | 2024-11-19 21:51:01 | delivered |
|         61 |      12 | Videos       | 2024-11-19   | 22:18:00     | 2024-11-19 22:14:48 | delivered |
|         62 |      12 | Hello        | 2024-11-20   | 22:45:00     | 2024-11-19 23:14:47 | scheduled |
|         63 |      12 | Plans        | 2024-11-20   | 22:01:00     | 2024-11-19 23:50:59 | scheduled |
+------------+---------+--------------+--------------+--------------+---------------------+-----------+
19 rows in set (0.00 sec)
```

TRIGGERS

Trigger 1

```
DELIMITER $$

-- Trigger to delete time_capsules when user is deleted
CREATE TRIGGER delete_time_capsules
AFTER DELETE ON users
FOR EACH ROW
BEGIN
   DELETE FROM time_capsules WHERE user_id = OLD.user_id;
END $$
```

Before:

```
mysql> select * from time_capsules;
+------------+---------+--------------+--------------+--------------+---------------------+-----------+
| capsule_id | user_id | capsule_name | release_date | release_time | created_at          | status    |
+------------+---------+--------------+--------------+--------------+---------------------+-----------+
|         45 |      10 | Cool         | 2024-11-16   | 22:40:00     | 2024-11-16 22:38:56 | delivered |
|         46 |       6 | Enable       | 2024-11-16   | 22:52:00     | 2024-11-16 22:49:32 | delivered |
|         47 |       6 | November     | 2024-11-17   | 01:13:00     | 2024-11-17 01:11:03 | delivered |
|         48 |       6 | Rain         | 2024-11-17   | 01:20:00     | 2024-11-17 01:12:43 | delivered |
|         49 |       6 | work         | 2024-11-17   | 01:20:00     | 2024-11-17 01:14:32 | delivered |
|         50 |       6 | New year     | 2024-11-17   | 12:10:00     | 2024-11-17 01:17:54 | delivered |
|         51 |       6 | Project      | 2024-11-17   | 01:30:00     | 2024-11-17 01:26:42 | delivered |
|         52 |      10 | Weekend      | 2024-11-19   | 09:20:00     | 2024-11-19 09:18:42 | delivered |
|         53 |      10 | WeekendPlan  | 2024-11-19   | 09:30:00     | 2024-11-19 09:24:17 | delivered |
|         54 |      10 | Monday       | 2024-11-19   | 09:40:00     | 2024-11-19 09:35:42 | delivered |
|         55 |      10 | friends      | 2024-11-19   | 09:45:00     | 2024-11-19 09:40:53 | delivered |
|         56 |      10 | Tuesday      | 2024-11-19   | 10:00:00     | 2024-11-19 09:59:56 | delivered |
|         57 |      10 | Days         | 2024-11-19   | 10:03:00     | 2024-11-19 10:01:04 | delivered |
|         58 |      10 | Tests        | 2024-11-19   | 14:10:00     | 2024-11-19 14:07:15 | delivered |
|         59 |      12 | Memories     | 2024-11-19   | 21:30:00     | 2024-11-19 21:24:30 | delivered |
|         60 |      12 | HolidayPlan  | 2024-11-19   | 21:55:00     | 2024-11-19 21:51:01 | delivered |
|         61 |      12 | Videos       | 2024-11-19   | 22:18:00     | 2024-11-19 22:14:48 | delivered |
|         62 |      12 | Hello        | 2024-11-20   | 22:45:00     | 2024-11-19 23:14:47 | scheduled |
|         63 |      12 | Plans        | 2024-11-20   | 22:01:00     | 2024-11-19 23:50:59 | scheduled |
|         64 |       9 | Schedule     | 2024-11-20   | 09:55:00     | 2024-11-20 09:50:35 | scheduled |
+------------+---------+--------------+--------------+--------------+---------------------+-----------+
20 rows in set (0.00 sec)
```

After:

```
mysql> select * from time_capsules;
+------------+---------+--------------+--------------+--------------+---------------------+-----------+
| capsule_id | user_id | capsule_name | release_date | release_time | created_at          | status    |
+------------+---------+--------------+--------------+--------------+---------------------+-----------+
|         45 |      10 | Cool         | 2024-11-16   | 22:40:00     | 2024-11-16 22:38:56 | delivered |
|         46 |       6 | Enable       | 2024-11-16   | 22:52:00     | 2024-11-16 22:49:32 | delivered |
|         47 |       6 | November     | 2024-11-17   | 01:13:00     | 2024-11-17 01:11:03 | delivered |
|         48 |       6 | Rain         | 2024-11-17   | 01:20:00     | 2024-11-17 01:12:43 | delivered |
|         49 |       6 | work         | 2024-11-17   | 01:20:00     | 2024-11-17 01:14:32 | delivered |
|         50 |       6 | New year     | 2024-11-17   | 12:10:00     | 2024-11-17 01:17:54 | delivered |
|         51 |       6 | Project      | 2024-11-17   | 01:30:00     | 2024-11-17 01:26:42 | delivered |
|         52 |      10 | Weekend      | 2024-11-19   | 09:20:00     | 2024-11-19 09:18:42 | delivered |
|         53 |      10 | WeekendPlan  | 2024-11-19   | 09:30:00     | 2024-11-19 09:24:17 | delivered |
|         54 |      10 | Monday       | 2024-11-19   | 09:40:00     | 2024-11-19 09:35:42 | delivered |
|         55 |      10 | friends      | 2024-11-19   | 09:45:00     | 2024-11-19 09:40:53 | delivered |
|         56 |      10 | Tuesday      | 2024-11-19   | 10:00:00     | 2024-11-19 09:59:56 | delivered |
|         57 |      10 | Days         | 2024-11-19   | 10:03:00     | 2024-11-19 10:01:04 | delivered |
|         58 |      10 | Tests        | 2024-11-19   | 14:10:00     | 2024-11-19 14:07:15 | delivered |
|         59 |      12 | Memories     | 2024-11-19   | 21:30:00     | 2024-11-19 21:24:30 | delivered |
|         60 |      12 | HolidayPlan  | 2024-11-19   | 21:55:00     | 2024-11-19 21:51:01 | delivered |
|         61 |      12 | Videos       | 2024-11-19   | 22:18:00     | 2024-11-19 22:14:48 | delivered |
|         62 |      12 | Hello        | 2024-11-20   | 22:45:00     | 2024-11-19 23:14:47 | scheduled |
|         63 |      12 | Plans        | 2024-11-20   | 22:01:00     | 2024-11-19 23:50:59 | scheduled |
+------------+---------+--------------+--------------+--------------+---------------------+-----------+
19 rows in set (0.00 sec)
```

Here, the time_capsule entry 64 is deleted.

Purpose: This trigger automatically deletes all time capsules associated with a user when that user is deleted from the users table. It ensures that no orphaned time capsules remain in the database.

Trigger 2

```
-- Trigger to delete capsule_contents when time_capsules is deleted
CREATE TRIGGER delete_capsule_contents
AFTER DELETE ON time_capsules
FOR EACH ROW
BEGIN
   DELETE FROM capsule_contents WHERE capsule_id = OLD.capsule_id;
END $$
```

Before:

Result Grid | Filter Rows: 64 | Edit: | Export/Import: | Wrap Cell Content:

content_id	capsule_id	content_type	content_data	content_size	uploaded_at
43	64	image	BLOB	4712	2024-11-20 09:50:44
NULL	NULL	NULL	NULL	NULL	NULL

After:

Result Grid | Filter Rows: 64 | Edit: | Export/Import: | Wrap Cell Content:

content_id	capsule_id	content_type	content_data	content_size	uploaded_at
NULL	NULL	NULL	NULL	NULL	NULL

Purpose: This trigger deletes all contents associated with a time capsule when the capsule itself is deleted from the time_capsules table. It maintains data integrity by removing related content to prevent orphaned records.

Trigger 3

```
-- Trigger to delete shared_users when time_capsules is deleted
CREATE TRIGGER delete_shared_users
AFTER DELETE ON time_capsules
FOR EACH ROW
BEGIN
   DELETE FROM shared_users WHERE capsule_id = OLD.capsule_id;
END $$

DELIMITER ;
```

Purpose: This trigger removes all entries in the shared_users table that are linked to a time capsule when that capsule is deleted. It ensures that shared user relationships are cleaned up alongside the deletion of the capsule.

Before:

```
mysql> select * from shared_users;
+----------------+------------+-----------------+
| shared_user_id | capsule_id | shared_username |
+----------------+------------+-----------------+
|             54 |         45 | Taylor          |
|             55 |         45 | Keem            |
|             56 |         46 | Keem            |
|             58 |         47 | Keem            |
|             59 |         49 | Akira           |
|             60 |         50 | Sash            |
|             61 |         50 | Akira           |
|             62 |         51 | Keem            |
|             63 |         51 | AKira           |
|             64 |         55 | Taylor          |
|             65 |         55 | Keem            |
|             66 |         57 | Keem            |
|             67 |         57 | Akira           |
|             68 |         57 | Taylor          |
|             69 |         58 | Taylor          |
|             70 |         58 | Keem            |
|             71 |         59 | Mona            |
|             72 |         59 | Akira           |
|             74 |         59 | Taylor          |
|             75 |         59 | Keem            |
|             76 |         60 | Mona            |
|             77 |         60 | Akira           |
|             78 |         60 | Keem            |
|             80 |         61 | Mona            |
|             81 |         61 | Taylor          |
|             82 |         64 | Dishhaa         |
|             83 |         64 | Keem            |
+----------------+------------+-----------------+
27 rows in set (0.00 sec)
```

After:

```
mysql> select * from shared_users;
+----------------+------------+-----------------+
| shared_user_id | capsule_id | shared_username |
+----------------+------------+-----------------+
|             54 |         45 | Taylor          |
|             55 |         45 | Keem            |
|             56 |         46 | Keem            |
|             58 |         47 | Keem            |
|             59 |         49 | Akira           |
|             60 |         50 | Sash            |
|             61 |         50 | Akira           |
|             62 |         51 | Keem            |
|             63 |         51 | AKira           |
|             64 |         55 | Taylor          |
|             65 |         55 | Keem            |
|             66 |         57 | Keem            |
|             67 |         57 | Akira           |
|             68 |         57 | Taylor          |
|             69 |         58 | Taylor          |
|             70 |         58 | Keem            |
|             71 |         59 | Mona            |
|             72 |         59 | Akira           |
|             74 |         59 | Taylor          |
|             75 |         59 | Keem            |
|             76 |         60 | Mona            |
|             77 |         60 | Akira           |
|             78 |         60 | Keem            |
|             80 |         61 | Mona            |
|             81 |         61 | Taylor          |
+----------------+------------+-----------------+
25 rows in set (0.00 sec)
```

Trigger 4

```
DELIMITER $$

-- Trigger to delete audit_logs when user is deleted
CREATE TRIGGER delete_audit_logs
AFTER DELETE ON users
FOR EACH ROW
BEGIN
   DELETE FROM audit_logs WHERE user_id = OLD.user_id;
END $$

DELIMITER ;
```

Before:

Result Grid | Filter Rows: 64 | Edit: | Export/Import: | Wrap Cell Content:

log_id	user_id	action_type	capsule_id	timestamp
64	6	scheduled	49	2024-11-17 01:14:54
97	9	created	64	2024-11-20 09:50:35
98	9	scheduled	64	2024-11-20 09:50:53
99	9	delivered	64	2024-11-20 09:55:50
NULL	NULL	NULL	NULL	NULL

After:

Result Grid | Filter Rows: 64 | Edit: | Export/Import: | Wrap Cell Content:

log_id	user_id	action_type	capsule_id	timestamp
64	6	scheduled	49	2024-11-17 01:14:54
NULL	NULL	NULL	NULL	NULL

Here, the audit_logs of capsule_id = 64 is deleted.

Purpose: This trigger deletes all audit log entries associated with a user when that user is deleted from the users table. It helps maintain a clean audit trail by removing logs that are no longer relevant to existing users.

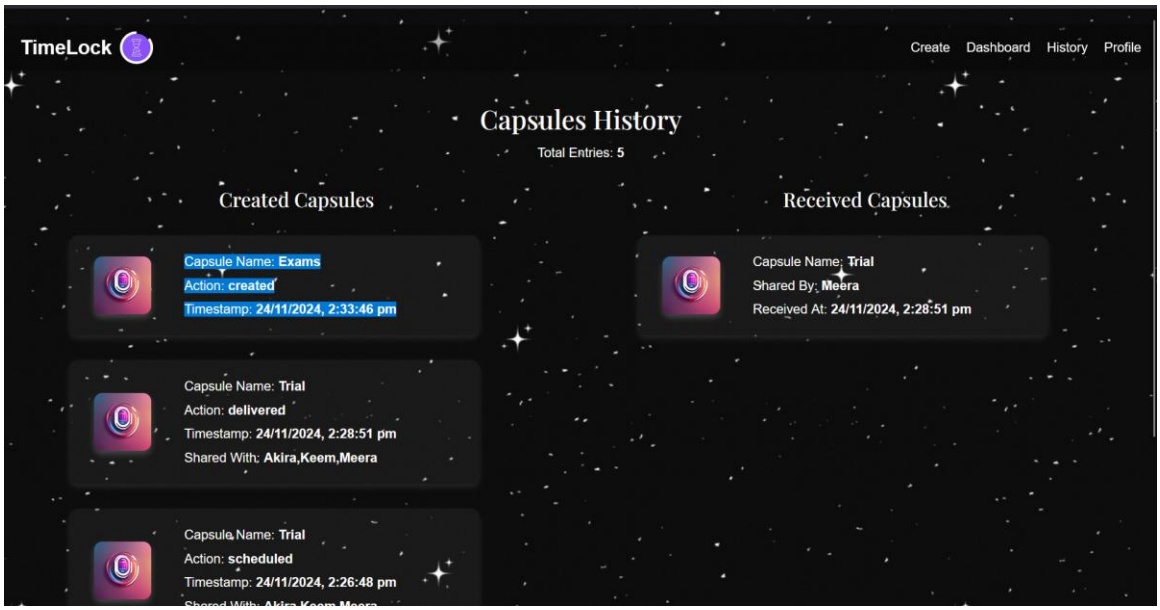
Trigger 5

```
DELIMITER $$

-- Trigger to create an entry in audit_logs when a new time capsule is created
CREATE TRIGGER after_time_capsule_insert
AFTER INSERT ON time_capsules
FOR EACH ROW
BEGIN
   -- Log the creation of a new time capsule
```

```
    INSERT INTO audit_logs (user_id, action_type, capsule_id)
    VALUES (NEW.user_id, 'created', NEW.capsule_id);
END $$

DELIMITER ;
```

```
mysql> select * from audit_logs;
+--------+---------+-------------+------------+---------------------+
| log_id | user_id | action_type | capsule_id | timestamp           |
+--------+---------+-------------+------------+---------------------+
|     56 |       6 | created     |         46 | 2024-11-16 22:49:32 |
|     57 |       6 | scheduled   |         46 | 2024-11-16 22:49:46 |
|     58 |       6 | delivered   |         46 | 2024-11-16 22:52:50 |
|     59 |       6 | created     |         47 | 2024-11-17 01:11:03 |
|     60 |       6 | scheduled   |         47 | 2024-11-17 01:11:30 |
|     61 |       6 | created     |         48 | 2024-11-17 01:12:43 |
|     62 |       6 | delivered   |         47 | 2024-11-17 01:13:50 |
|     63 |       6 | created     |         49 | 2024-11-17 01:14:32 |
|     64 |       6 | scheduled   |         49 | 2024-11-17 01:14:54 |
|     65 |       6 | created     |         50 | 2024-11-17 01:17:54 |
|     66 |       6 | delivered   |         48 | 2024-11-17 01:20:50 |
|     67 |       6 | delivered   |         49 | 2024-11-17 01:20:50 |
|     68 |       6 | scheduled   |         50 | 2024-11-17 01:24:54 |
|     69 |       6 | created     |         51 | 2024-11-17 01:26:42 |
|     70 |       6 | scheduled   |         51 | 2024-11-17 01:27:14 |
|     71 |       6 | delivered   |         51 | 2024-11-17 01:30:50 |
|     72 |       6 | delivered   |         50 | 2024-11-17 12:10:50 |
|    100 |      13 | scheduled   |         65 | 2024-11-20 21:58:34 |
|    101 |      13 | scheduled   |         65 | 2024-11-20 21:59:54 |
|    102 |      14 | created     |         66 | 2024-11-24 14:26:11 |
|    103 |      14 | scheduled   |         66 | 2024-11-24 14:26:48 |
|    104 |      14 | delivered   |         66 | 2024-11-24 14:28:51 |
|    105 |      14 | created     |         67 | 2024-11-24 14:33:46 |
+--------+---------+-------------+------------+---------------------+
23 rows in set (0.00 sec)
```

Frontend:



Purpose: This trigger automatically logs the creation of a new time capsule in the audit_logs table after a new row is inserted into the time_capsules table. It records the user ID, action type ('created'), and capsule ID for tracking purposes.

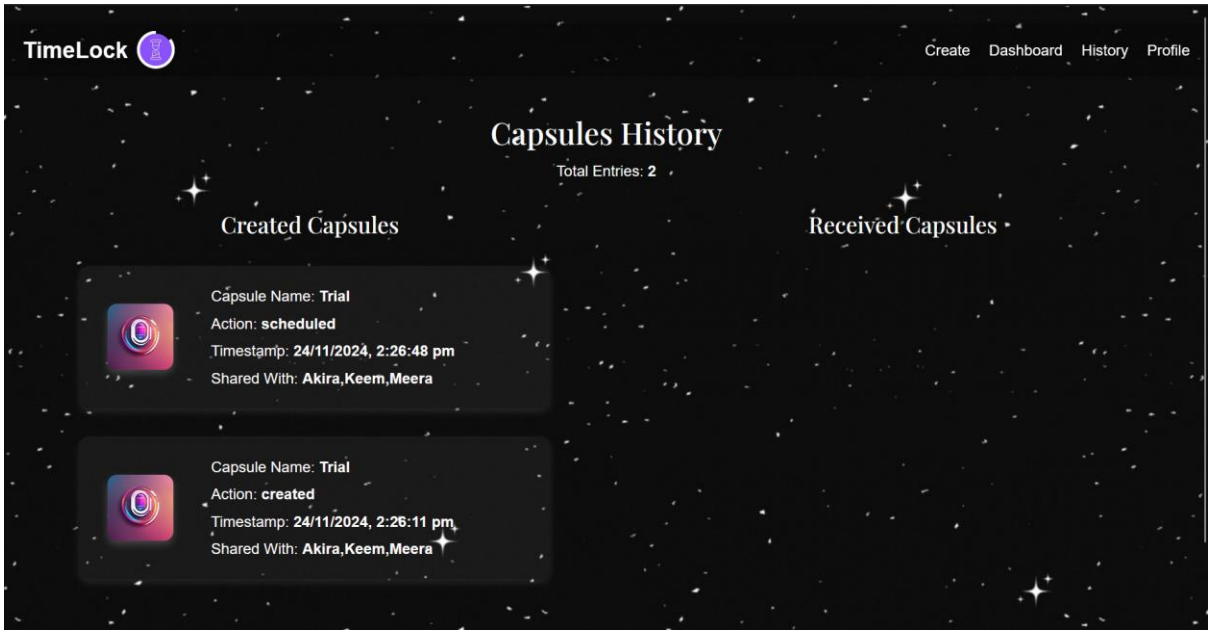
Trigger 6

```
DELIMITER $$

CREATE TRIGGER after_received_capsules_insert
AFTER INSERT ON received_capsules
FOR EACH ROW
BEGIN
   -- Check if an entry for the action 'delivered' already exists for the capsule_id
   IF NOT EXISTS (
      SELECT 1
      FROM audit_logs
      WHERE capsule_id = NEW.capsule_id AND action_type = 'delivered'
   ) THEN
      -- Insert the log only if no existing 'delivered' log exists for this capsule_id
      INSERT INTO audit_logs (user_id, action_type, capsule_id, timestamp)
      SELECT tc.user_id, 'delivered', NEW.capsule_id, NOW()
      FROM time_capsules tc
      WHERE tc.capsule_id = NEW.capsule_id;
   END IF;
END$$

DELIMITER ;
```

Before:

```
mysql> select * from audit_logs;
+--------+---------+-------------+------------+---------------------+
| log_id | user_id | action_type | capsule_id | timestamp           |
+--------+---------+-------------+------------+---------------------+
|     56 |       6 | created     |         46 | 2024-11-16 22:49:32 |
|     57 |       6 | scheduled   |         46 | 2024-11-16 22:49:46 |
|     58 |       6 | delivered   |         46 | 2024-11-16 22:52:50 |
|     59 |       6 | created     |         47 | 2024-11-17 01:11:03 |
|     60 |       6 | scheduled   |         47 | 2024-11-17 01:11:30 |
|     61 |       6 | created     |         48 | 2024-11-17 01:12:43 |
|     62 |       6 | delivered   |         47 | 2024-11-17 01:13:50 |
|     63 |       6 | created     |         49 | 2024-11-17 01:14:32 |
|     64 |       6 | scheduled   |         49 | 2024-11-17 01:14:54 |
|     65 |       6 | created     |         50 | 2024-11-17 01:17:54 |
|     66 |       6 | delivered   |         48 | 2024-11-17 01:20:50 |
|     67 |       6 | delivered   |         49 | 2024-11-17 01:20:50 |
|     68 |       6 | scheduled   |         50 | 2024-11-17 01:24:54 |
|     69 |       6 | created     |         51 | 2024-11-17 01:26:42 |
|     70 |       6 | scheduled   |         51 | 2024-11-17 01:27:14 |
|     71 |       6 | delivered   |         51 | 2024-11-17 01:30:50 |
|     72 |       6 | delivered   |         50 | 2024-11-17 12:10:50 |
|    100 |      13 | scheduled   |         65 | 2024-11-20 21:58:34 |
|    101 |      13 | scheduled   |         65 | 2024-11-20 21:59:54 |
|    102 |      14 | created     |         66 | 2024-11-24 14:26:11 |
|    103 |      14 | scheduled   |         66 | 2024-11-24 14:26:48 |
+--------+---------+-------------+------------+---------------------+
```

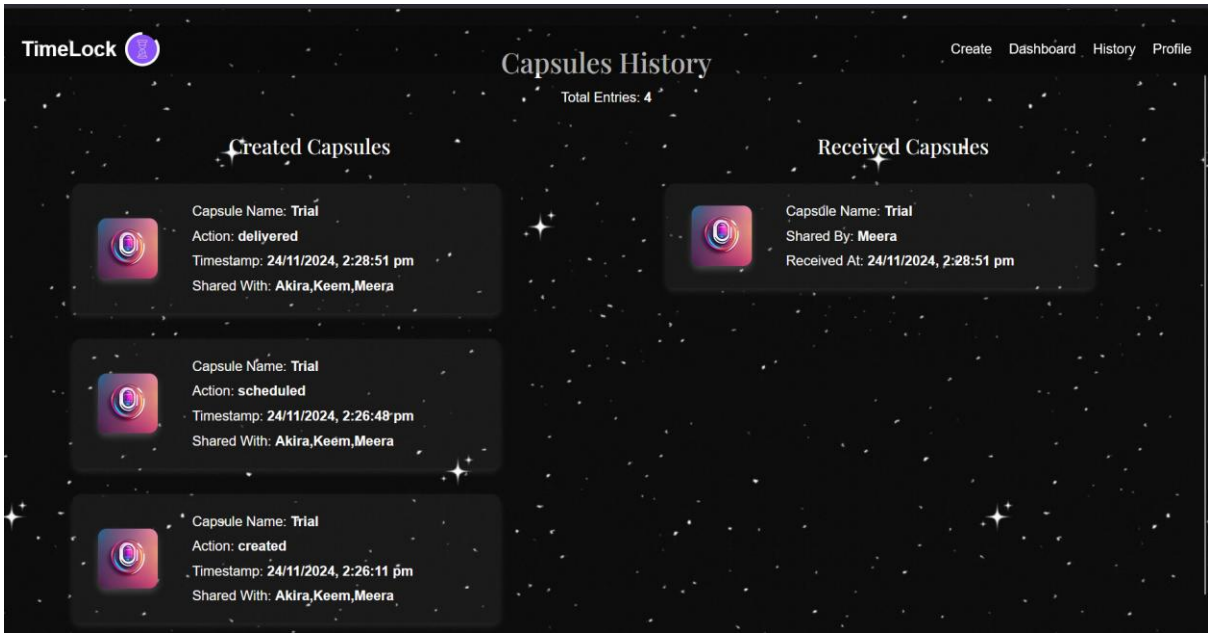
Frontend:



After:

```
mysql> select * from audit_logs;
+--------+---------+-------------+------------+---------------------+
| log_id | user_id | action_type | capsule_id | timestamp           |
+--------+---------+-------------+------------+---------------------+
|     56 |       6 | created     |         46 | 2024-11-16 22:49:32 |
|     57 |       6 | scheduled   |         46 | 2024-11-16 22:49:46 |
|     58 |       6 | delivered   |         46 | 2024-11-16 22:52:50 |
|     59 |       6 | created     |         47 | 2024-11-17 01:11:03 |
|     60 |       6 | scheduled   |         47 | 2024-11-17 01:11:30 |
|     61 |       6 | created     |         48 | 2024-11-17 01:12:43 |
|     62 |       6 | delivered   |         47 | 2024-11-17 01:13:50 |
|     63 |       6 | created     |         49 | 2024-11-17 01:14:32 |
|     64 |       6 | scheduled   |         49 | 2024-11-17 01:14:54 |
|     65 |       6 | created     |         50 | 2024-11-17 01:17:54 |
|     66 |       6 | delivered   |         48 | 2024-11-17 01:20:50 |
|     67 |       6 | delivered   |         49 | 2024-11-17 01:20:50 |
|     68 |       6 | scheduled   |         50 | 2024-11-17 01:24:54 |
|     69 |       6 | created     |         51 | 2024-11-17 01:26:42 |
|     70 |       6 | scheduled   |         51 | 2024-11-17 01:27:14 |
|     71 |       6 | delivered   |         51 | 2024-11-17 01:30:50 |
|     72 |       6 | delivered   |         50 | 2024-11-17 12:10:50 |
|    100 |      13 | scheduled   |         65 | 2024-11-20 21:58:34 |
|    101 |      13 | scheduled   |         65 | 2024-11-20 21:59:54 |
|    102 |      14 | created     |         66 | 2024-11-24 14:26:11 |
|    103 |      14 | scheduled   |         66 | 2024-11-24 14:26:48 |
|    104 |      14 | delivered   |         66 | 2024-11-24 14:28:51 |
+--------+---------+-------------+------------+---------------------+
22 rows in set (0.00 sec)
```

Frontend:



Purpose: This trigger logs an action into the audit_logs table whenever a new row is added to the received_capsules table. It records the user ID, action type ('delivered'), capsule ID, and the current timestamp by referencing the related time_capsules table.

AGGREGATE QUERY

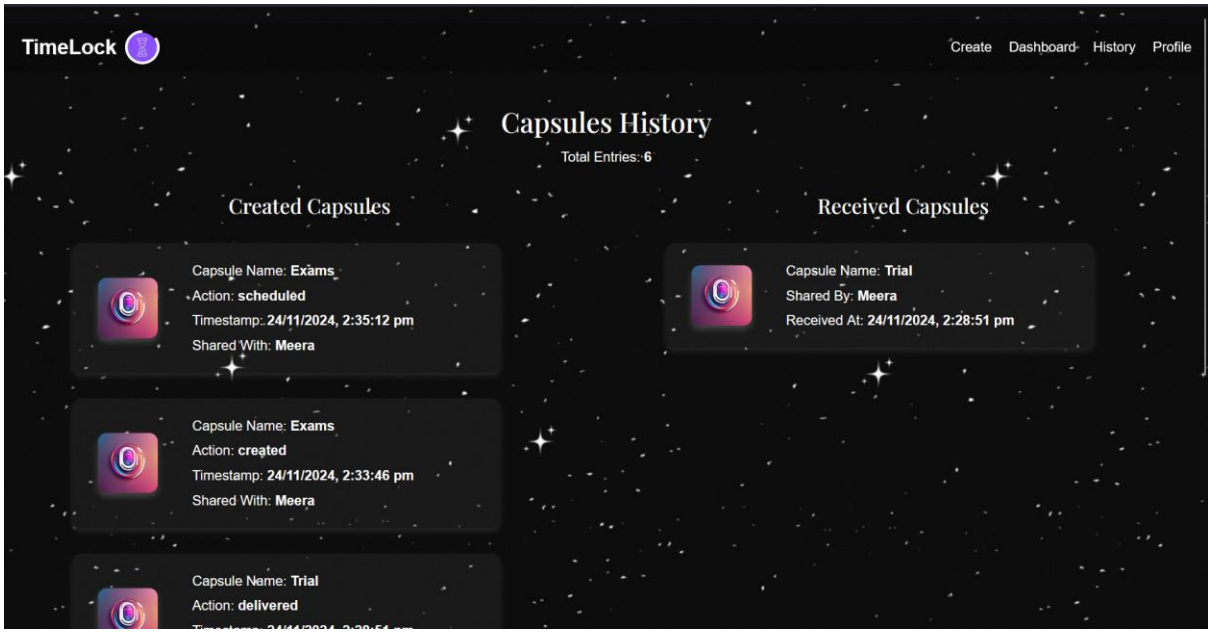
Trigger 7 + Aggregate 1

```
DELIMITER $$

-- Trigger to log action when a new entry is added to shared_users
CREATE TRIGGER after_shared_users_insert
AFTER INSERT ON shared_users
FOR EACH ROW
BEGIN
   DECLARE related_user_id INT;
   DECLARE capsule_contents_count INT;
   DECLARE already_scheduled INT;

   -- Retrieve the user_id from the related time_capsules entry
   SELECT user_id INTO related_user_id
   FROM time_capsules
   WHERE capsule_id = NEW.capsule_id;

   -- If no matching time capsule exists, exit
   IF related_user_id IS NULL THEN
      SIGNAL SQLSTATE '45000'
      SET MESSAGE_TEXT = 'No matching time capsule for shared_users entry';
   END IF;

   -- Check if a corresponding entry exists in the capsule_contents table
```

```
    SELECT COUNT(*) INTO capsule_contents_count
    FROM capsule_contents
    WHERE capsule_id = NEW.capsule_id;

    -- Check if "scheduled" is already logged
    SELECT COUNT(*) INTO already_scheduled
    FROM audit_logs
    WHERE capsule_id = NEW.capsule_id AND action_type = 'scheduled';

    -- Log "scheduled" only if:
    -- 1. There's an entry in capsule_contents, AND
    -- 2. "scheduled" has not already been logged
    IF capsule_contents_count > 0 AND already_scheduled = 0 THEN
        INSERT INTO audit_logs (user_id, action_type, capsule_id)
        VALUES (related_user_id, 'scheduled', NEW.capsule_id);
    END IF;
END $$

DELIMITER ;
```

```
mysql> select * from audit_logs;
+--------+---------+-------------+------------+---------------------+
| log_id | user_id | action_type | capsule_id | timestamp           |
+--------+---------+-------------+------------+---------------------+
|     56 |       6 | created     |         46 | 2024-11-16 22:49:32 |
|     57 |       6 | scheduled   |         46 | 2024-11-16 22:49:46 |
|     58 |       6 | delivered   |         46 | 2024-11-16 22:52:50 |
|     59 |       6 | created     |         47 | 2024-11-17 01:11:03 |
|     60 |       6 | scheduled   |         47 | 2024-11-17 01:11:30 |
|     61 |       6 | created     |         48 | 2024-11-17 01:12:43 |
|     62 |       6 | delivered   |         47 | 2024-11-17 01:13:50 |
|     63 |       6 | created     |         49 | 2024-11-17 01:14:32 |
|     64 |       6 | scheduled   |         49 | 2024-11-17 01:14:54 |
|     65 |       6 | created     |         50 | 2024-11-17 01:17:54 |
|     66 |       6 | delivered   |         48 | 2024-11-17 01:20:50 |
|     67 |       6 | delivered   |         49 | 2024-11-17 01:20:50 |
|     68 |       6 | scheduled   |         50 | 2024-11-17 01:24:54 |
|     69 |       6 | created     |         51 | 2024-11-17 01:26:42 |
|     70 |       6 | scheduled   |         51 | 2024-11-17 01:27:14 |
|     71 |       6 | delivered   |         51 | 2024-11-17 01:30:50 |
|     72 |       6 | delivered   |         50 | 2024-11-17 12:10:50 |
|    100 |      13 | scheduled   |         65 | 2024-11-20 21:58:34 |
|    101 |      13 | scheduled   |         65 | 2024-11-20 21:59:54 |
|    102 |      14 | created     |         66 | 2024-11-24 14:26:11 |
|    103 |      14 | scheduled   |         66 | 2024-11-24 14:26:48 |
|    104 |      14 | delivered   |         66 | 2024-11-24 14:28:51 |
|    105 |      14 | created     |         67 | 2024-11-24 14:33:46 |
|    106 |      14 | scheduled   |         67 | 2024-11-24 14:35:12 |
+--------+---------+-------------+------------+---------------------+
24 rows in set (0.00 sec)
```

Frontend:



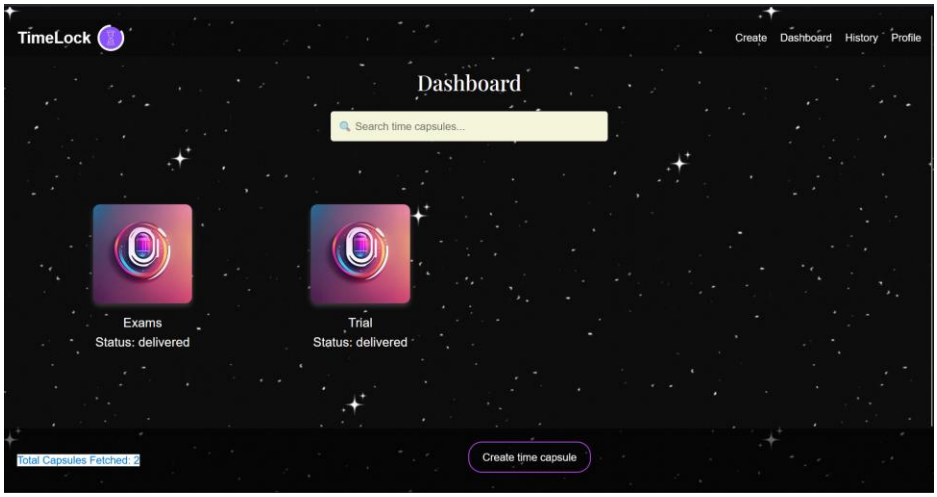
Purpose:

The trigger after_shared_users_insert is executed **after inserting a new row** into the shared_users table. It performs validations, checks data consistency across related tables, and conditionally logs actions in the audit_logs table.

The SELECT COUNT(*) aggregate query is used twice: once to count matching entries in the capsule_contents table and another to check if the scheduled action has already been logged in audit_logs.

Aggregate query 2 + Join Query 1:

```
SELECT COUNT(DISTINCT capsule_id) AS totalCount
    FROM (
      SELECT tc.capsule_id
      FROM users u
      INNER JOIN time_capsules tc ON u.user_id = tc.user_id
      WHERE u.username = ?
      UNION ALL
      SELECT tc.capsule_id
      FROM received_capsules rc
      INNER JOIN time_capsules tc ON rc.capsule_id = tc.capsule_id
      WHERE rc.receiver_username = ?
    ) AS combined_capsules
```



Purpose:

Aggregate query: This query calculates the total number of **unique capsule_ids** associated with a specific user.

Join query: This query joins the users and time_capsules tables to find time capsules created by a specific user, and the received_capsules and time_capsules tables to find those received by the user. It combines both results and counts the distinct capsule_ids to get the total number of unique capsules.

NESTED QUERY

```
INSERT INTO shared_users (capsule_id, shared_username)
    SELECT ?, ?
    WHERE NOT EXISTS (
      SELECT 1 FROM shared_users
      WHERE capsule_id = ? AND shared_username = ?
    )
    AND EXISTS (
      SELECT 1 FROM users
      WHERE username = ?
    )
```

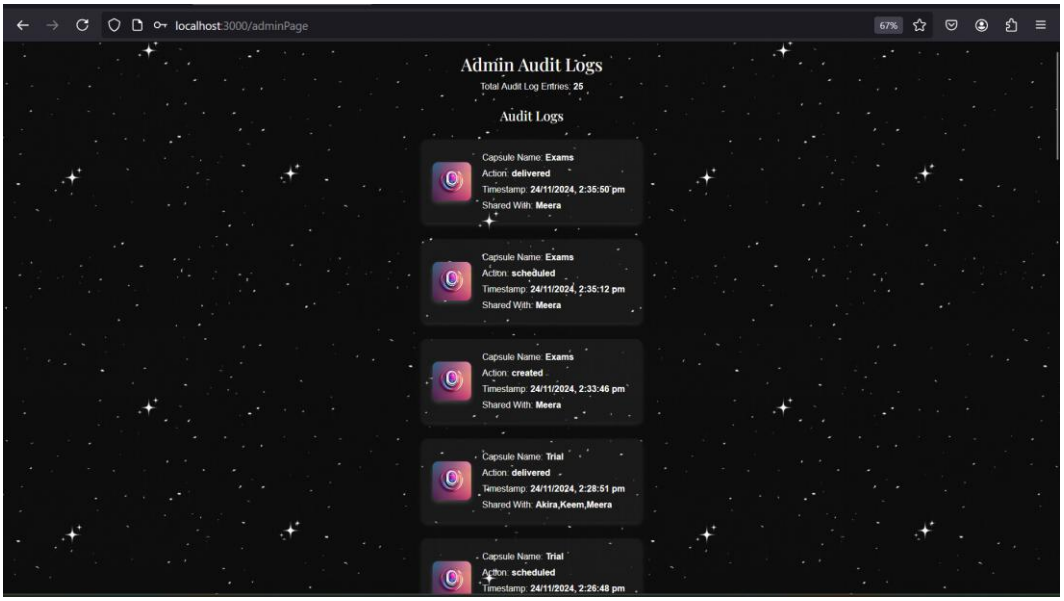
```
mysql> select * from time_capsules;
+------------+---------+--------------+--------------+--------------+---------------------+-----------+
| capsule_id | user_id | capsule_name | release_date | release_time | created_at          | status    |
+------------+---------+--------------+--------------+--------------+---------------------+-----------+
|         46 |       6 | Enable       | 2024-11-16   | 22:52:00     | 2024-11-16 22:49:32 | delivered |
|         47 |       6 | November     | 2024-11-17   | 01:13:00     | 2024-11-17 01:11:03 | delivered |
|         48 |       6 | Rain         | 2024-11-17   | 01:20:00     | 2024-11-17 01:12:43 | delivered |
|         49 |       6 | work         | 2024-11-17   | 01:20:00     | 2024-11-17 01:14:32 | delivered |
|         50 |       6 | New year     | 2024-11-17   | 12:10:00     | 2024-11-17 01:17:54 | delivered |
|         51 |       6 | Project      | 2024-11-17   | 01:30:00     | 2024-11-17 01:26:42 | delivered |
|         65 |      13 | Memories     | 2025-01-01   | 10:10:00     | 2024-11-20 21:58:34 | scheduled |
|         66 |      14 | Trial        | 2024-11-24   | 14:28:00     | 2024-11-24 14:26:11 | delivered |
|         67 |      14 | Exams        | 2024-11-24   | 14:35:00     | 2024-11-24 14:33:46 | delivered |
+------------+---------+--------------+--------------+--------------+---------------------+-----------+
9 rows in set (0.00 sec)

mysql> select * from shared_users;
+----------------+------------+-----------------+
| shared_user_id | capsule_id | shared_username |
+----------------+------------+-----------------+
|             56 |         46 | Keem            |
|             58 |         47 | Keem            |
|             59 |         49 | Akira           |
|             60 |         50 | Sash            |
|             61 |         50 | Akira           |
|             62 |         51 | Keem            |
|             63 |         51 | AKira           |
|             84 |         65 | Daisy1          |
|             85 |         66 | Meera           |
|             86 |         66 | Keem            |
|             87 |         66 | Akira           |
|             88 |         67 | Meera           |
+----------------+------------+-----------------+
12 rows in set (0.00 sec)
```

Purpose: This code inserts a new record into the shared_users table only if the given capsule_id and shared_username combination doesn't already exist and the shared_username exists in the users table. It ensures data consistency and prevents duplicate or invalid entries using nested subqueries.

JOIN QUERY

Join Query 2:

```
SELECT al.log_id, al.user_id, al.action_type, al.capsule_id, al.timestamp,
        tc.capsule_name,
        GROUP_CONCAT(DISTINCT su.shared_username ORDER BY su.shared_username ASC) AS
shared_usernames
    FROM audit_logs al
    LEFT JOIN time_capsules tc ON al.capsule_id = tc.capsule_id
    LEFT JOIN shared_users su ON al.capsule_id = su.capsule_id
    GROUP BY al.log_id, al.user_id, al.action_type, al.capsule_id, al.timestamp, tc.capsule_name
    ORDER BY al.timestamp DESC
```



Purpose: This query retrieves audit log details along with the capsule name and a list of usernames the capsule was shared with, grouped by each log entry. It uses GROUP_CONCAT to combine shared usernames into a single string and orders the results by the most recent timestamp.

Join Query 3:

```
SELECT al.log_id, al.user_id, al.action_type, al.capsule_id, al.timestamp,
        tc.capsule_name,
        GROUP_CONCAT(DISTINCT su.shared_username ORDER BY su.shared_username ASC) AS
shared_usernames
    FROM audit_logs al
    LEFT JOIN time_capsules tc ON al.capsule_id = tc.capsule_id
    LEFT JOIN shared_users su ON al.capsule_id = su.capsule_id
```

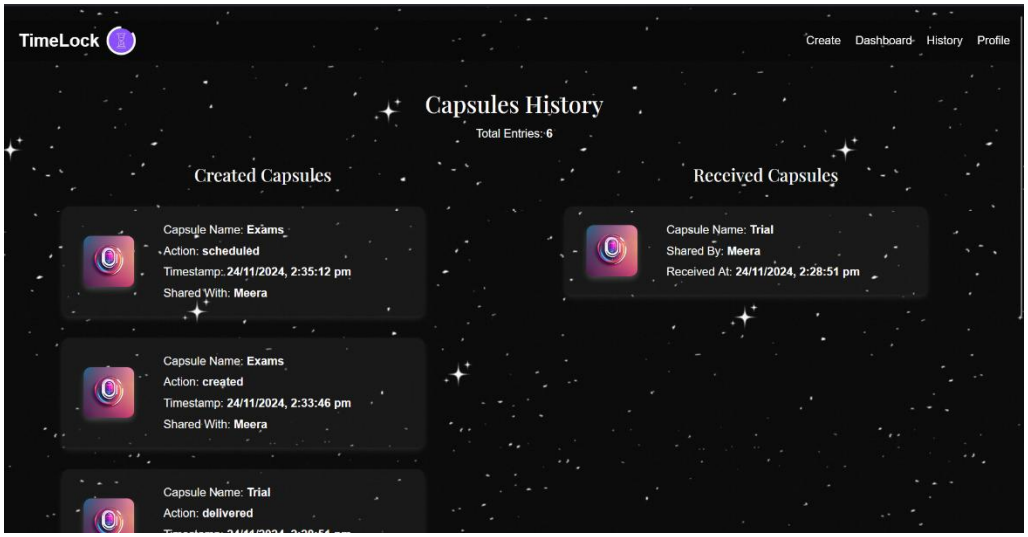
```
WHERE al.user_id = ?
GROUP BY al.log_id, al.user_id, al.action_type, al.capsule_id, al.timestamp, tc.capsule_name
ORDER BY al.timestamp DESC
```

Purpose: This query fetches audit log entries for a specific user (al.user_id = ?), including details like capsule name and a concatenated list of distinct shared usernames. It groups results by log details, orders shared usernames alphabetically, and sorts the overall output by the most recent log entry timestamp.

Join Query 4:

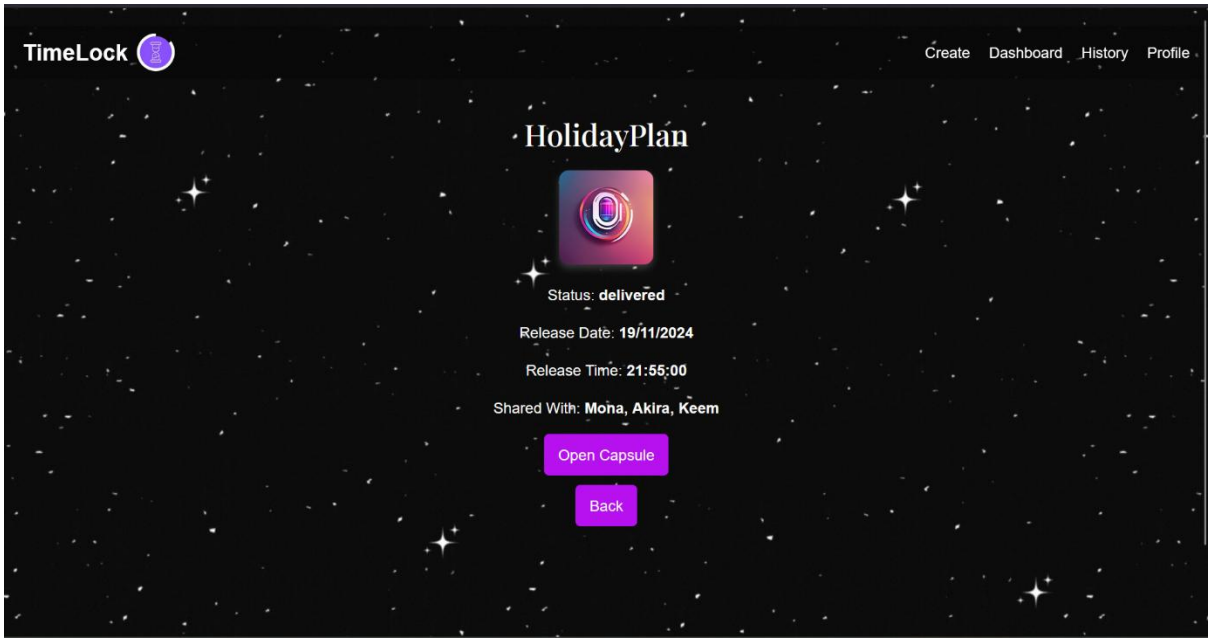
```
SELECT rc.received_id, rc.capsule_id, rc.shared_by_username, rc.received_at, tc.capsule_name
    FROM received_capsules rc
    LEFT JOIN time_capsules tc ON rc.capsule_id = tc.capsule_id
    WHERE rc.receiver_username = ?
    ORDER BY rc.received_at DESC
```

Purpose: This query retrieves details of received capsules for a specific receiver (rc.receiver_username = ?), including the capsule ID, shared by username, and received time, joining with the time_capsules table to fetch the capsule name. The results are sorted by the most recent received_at timestamp.



Join Query 5:

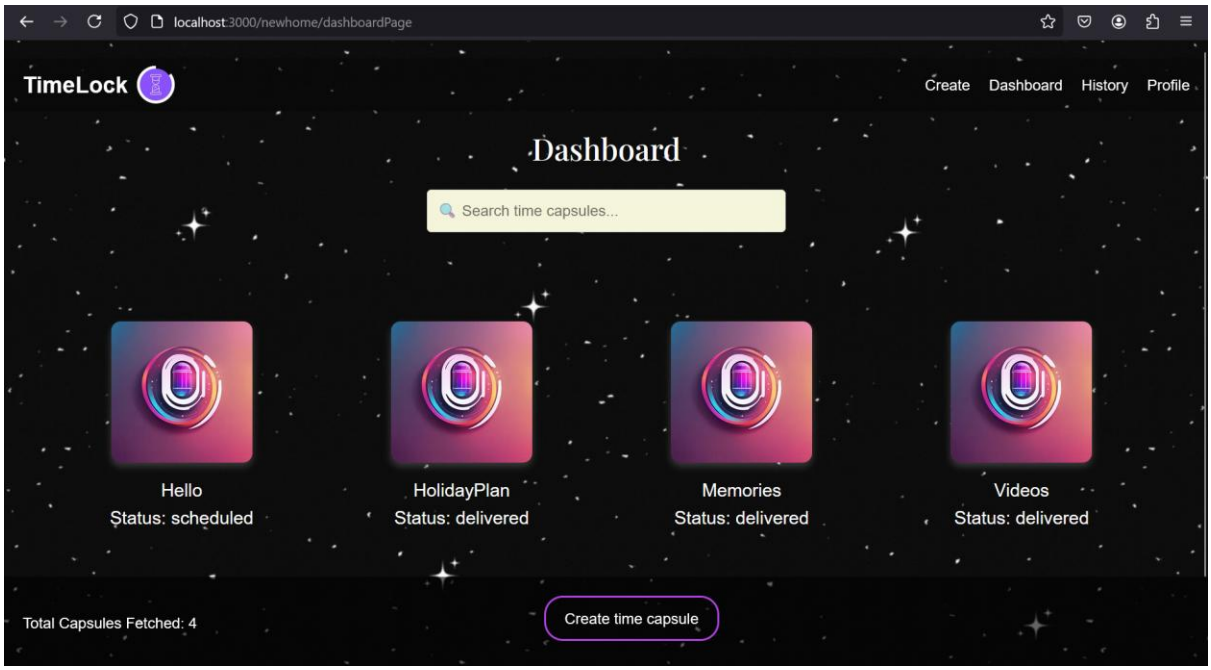
```
SELECT t.capsule_id, t.capsule_name, t.release_date, t.release_time, t.status, u.first_name,
u.last_name
    FROM time_capsules t
    JOIN users u ON t.user_id = u.user_id
    WHERE t.capsule_name = ?
```



Purpose: This query retrieves the details of a time capsule, including its ID, name, release date, and status, along with the creator's first and last name, by joining the time_capsules table with the users table. It filters the results based on a specific capsule_name.

Join Query 6:

```
SELECT tc.capsule_id, tc.capsule_name, tc.status
    FROM users u
    INNER JOIN time_capsules tc ON u.user_id = tc.user_id
    WHERE u.username = ?
```



Purpose: This query retrieves the capsule_id, capsule_name, and status of time capsules created by a specific user (u.username = ?) by joining the users table with the time_capsules table. It filters the results based on the provided username.

Procedures:

```
DELIMITER $$

CREATE PROCEDURE CheckAndReleaseCapsules()
BEGIN
   DECLARE done INT DEFAULT FALSE;
   DECLARE capsuleID INT;
   DECLARE userID INT;

   -- Declare a cursor for time capsules ready to be released
   DECLARE release_cursor CURSOR FOR
      SELECT tc.capsule_id, tc.user_id
      FROM time_capsules AS tc
      WHERE CONCAT(tc.release_date, ' ', tc.release_time) <= NOW()
      AND tc.status = 'scheduled';

   -- Declare a handler to exit the loop
   DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

   -- Open the cursor
   OPEN release_cursor;

   -- Loop through all capsules that are ready to be released
   release_loop: LOOP
      FETCH release_cursor INTO capsuleID, userID;

      IF done THEN
         LEAVE release_loop;
      END IF;

      -- Process the capsule: Insert for creator (ensure no duplicates)
      INSERT IGNORE INTO received_capsules (capsule_id, receiver_username, shared_by_username,
content_id, received_at)
      SELECT capsuleID, u.username, u.username, cc.content_id, NOW()
      FROM users u
      JOIN capsule_contents cc ON cc.capsule_id = capsuleID
      WHERE u.user_id = userID
      LIMIT 1;

      -- Process the capsule: Insert for shared users (ensure no duplicates)
      INSERT IGNORE INTO received_capsules (capsule_id, receiver_username, shared_by_username,
content_id, received_at)
      SELECT capsuleID, su.shared_username, u.username, cc.content_id, NOW()
      FROM shared_users su
      JOIN users u ON u.user_id = userID
      JOIN capsule_contents cc ON cc.capsule_id = capsuleID
      WHERE su.capsule_id = capsuleID;

      -- Mark the capsule as delivered
      UPDATE time_capsules
```

```
        SET status = 'delivered'
        WHERE capsule_id = capsuleID;
    END LOOP;

    -- Close the cursor
    CLOSE release_cursor;
END$$

DELIMITER ;
```

```
mysql> select * from received_capsules;
+-------------+------------+-------------------+-------------------+------------+---------------------+
| received_id | capsule_id | receiver_username | shared_by_username | content_id | received_at         |
+-------------+------------+-------------------+-------------------+------------+---------------------+
|         115 |         46 | Keem              | Keem              |         30 | 2024-11-16 22:52:50 |
|         117 |         47 | Keem              | Keem              |         31 | 2024-11-17 01:13:50 |
|         119 |         48 | Keem              | Keem              |         32 | 2024-11-17 01:20:50 |
|         120 |         49 | Keem              | Keem              |         33 | 2024-11-17 01:20:50 |
|         121 |         49 | Akira             | Keem              |         33 | 2024-11-17 01:20:50 |
|         122 |         51 | Keem              | Keem              |         35 | 2024-11-17 01:30:50 |
|         123 |         51 | AKira             | Keem              |         35 | 2024-11-17 01:30:50 |
|         124 |         50 | Keem              | Keem              |         34 | 2024-11-17 12:10:50 |
|         125 |         50 | Sash              | Keem              |         34 | 2024-11-17 12:10:50 |
|         126 |         50 | Akira             | Keem              |         34 | 2024-11-17 12:10:50 |
|         147 |         66 | Meera             | Meera             |         46 | 2024-11-24 14:28:51 |
|         148 |         66 | Keem              | Meera             |         46 | 2024-11-24 14:28:51 |
|         149 |         66 | Akira             | Meera             |         46 | 2024-11-24 14:28:51 |
|         151 |         67 | Meera             | Meera             |         47 | 2024-11-24 14:35:50 |
+-------------+------------+-------------------+-------------------+------------+---------------------+
14 rows in set (0.00 sec)
```

Purpose: The CheckAndReleaseCapsules procedure processes time capsules that are scheduled for release by checking if the release date and time have passed. It inserts records into the received_capsules table for both the capsule creator and shared users, ensuring no duplicates. After processing, it updates the capsule's status to "delivered" and continues until all relevant capsules are processed.

GITHUB LINK:

https://github.com/Meeraja-K/Time-Capsule-Database-System