MULTIMEDIA UNIVERSITY ®

CCS6344 Database and Cloud Security Assignment 1

Group Name: 2

Lecturer Name:

Dr. Navaneethan A/L C.Arjuman

Prepared By:

Student ID	Name
1201200722	Nur Ayu Amira binti Idris
1221303085	Mannoj Sakthivel
1201201537	Muhammad Dhiyaul Naufal Bin Zainuddin

YouTube link: https://youtu.be/4yXjNV-xZIM
GitHub link: https://github.com/NurAyuAmira/Museum_Ticket_Booking_Website

Table of Contents

Preparation of the Proposal 3

Implementation of the application using SQL Database 6

Security Measures Implementation 26

Threat Modelling (STRIDE and DREAD) 33

PDPA 2010 35

Preparation of the Proposal

PROJECT PROPOSAL: Museum Ticket Reservation Website

1. **Objectives of the Project**

   The main objectives of the Museum Management Systems are:

   a. To implement robust security measures to protect sensitive user data and transaction information against unauthorized access and cyber threats.
   b. To create a comprehensive platform that streamlines the management of museum exhibits and reservations.
   c. To provide a user-friendly interface that caters to the needs of different user groups, including museum visitors and administrators.

2. **Proposed Design and Implementation of the Application**

   The system involves the development of a Museum Ticket Booking website. The website is responsible for the booking of museum tickets by the consumer. The application will prioritize the security of the application and the user experience. This is to ensure both the user and administrator sides have a safe experience using the application. The proposed application includes the following features:

   a. Exhibit Management - Administrators can manage exhibits with access restricted to authenticated administrators to ensure only authorized changes are made and tracked.
   b. User Management - Users can register, login and manage their profiles, with password securely hashed, and input validation to prevent injection attacks, ensuring secure access and preventing automated attacks.
   c. Forgot Password - Users and admins can reset their password by receiving a one-time password (OTP) via email, ensuring secure verification of user identity during the password recovery process.
   d. Reservation System - Users can make reservations to visit the museum, with all reservation data encrypted during transmission and access restricted to authenticated users.
   e. Payment System - Handles payments for reservations securely, following PCI-DSS standards, encrypting payment information during transmission, tokenizing data to minimize storage of sensitive information, and implementing input validation to

prevent fraudulent data entry, ensuring the integrity and security of payment transactions.

3. **<u>Proposed Hardware and Software to Develop the Application</u>**
    a. Programming Language: PHP for backend, HTML/CSS/JavaScript for frontend.
    b. Database: MySQL for data storage. phpMyAdmin for database management.
    c. Operating System: Windows 11.
    d. Server: XAMPP Local Server

4. **<u>System Design</u>**

    The system is designed using a multi-layered approach:

    a. Application Layer: Backend server and business logic, which handles the processing of user requests, application logic, and communication between the presentation layer and the data layer.
    b. Presentation Layer: User Interface, which includes the web pages and forms that users interact with.
    c. Data Layer: MySQL database for data storage, which stores all the data required by the application, including user information, exhibit details, reservations, payments, and activity logs.

5. **<u>Database Design</u>**

    The database includes the following tables:

    a. users: To store the user's personal details.
    b. admins: To store the admin's personal details.
    c. exhibits: To store the exhibit's details.
    d. reservations: To manage user reservations.
    e. payments: To manage payment details.
    f. Activity logs: To record user activities.
    g. Traffic logs: To monitor traffic from user.

6. **<u>Database Security</u>**

    To secure the database, the following measures are implemented:

    a. Authentication: Users must verify their identity before accessing the database by validating their usernames and passwords against stored credentials.

b. Authorization: After authentication, users are granted access only to the data they are authorized to view or modify, based on their roles and privileges.

c. Data Encryption: Sensitive data stored in the database is encrypted to protect it from unauthorized access.

d. Parameterized Queries and Prepared Statements: These techniques are used to prevent SQL Injection attacks by ensuring that user inputs are treated as data and not executable code.

e. Password Hashing: Passwords are hashed before being stored in the database to ensure that even if the data is compromised, the actual passwords are not exposed.

f. Data Masking: Sensitive information, such as credit card numbers, is masked within the database.

g. Input Validation: All user inputs are validated to ensure they meet the required criteria and do not contain malicious data.

h. Regular Backups and Recovery: Regular backups are scheduled to prevent data loss due to accidental deletion, corruption, or security breaches. These backups are securely stored to ensure data can be recovered when needed.

i. Auditing and Logging: Activities such as logins, and access attempts are monitored and logged.

j. Patch Management and Updates: The database management system (DBMS) is regularly updated with the latest security patches and fixes to address known vulnerabilities and improve security.

Implementation of the application using SQL Database

1. System Design

Museum Ticket Booking Website is an online ticketing system which allows users to purchase and book tickets. The website mainly has 2 roles which are admin and user. In terms of user, the user logs onto the website and sees the availability of exhibits. The user then can proceed to mark how many tickets they wish to buy based on the selected show. Then, the user will have to fill in their cardholder's name, number, expiration date, and CVV number of their debit/credit card. When all the mentioned are done correctly, the user will have successfully booked their respective ticket(s). In terms of admin, the admin can log onto the system to see all exhibits available, reservations made, payments made and user activity log. The admin can also add an exhibit by inserting the name, description, date, tickets available and price per ticket of the exhibit. The admin can also delete an exhibit as well as edit a show currently on display.

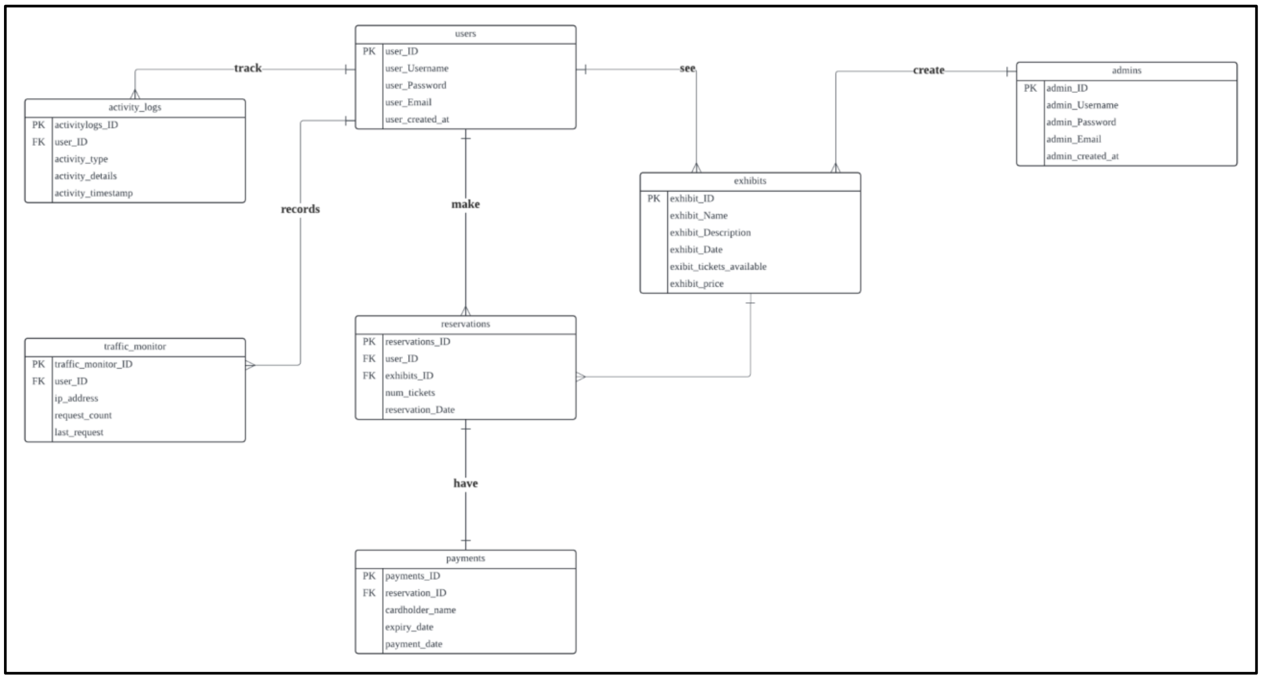


Figure 1.1: Museum Entity Relational Diagram.

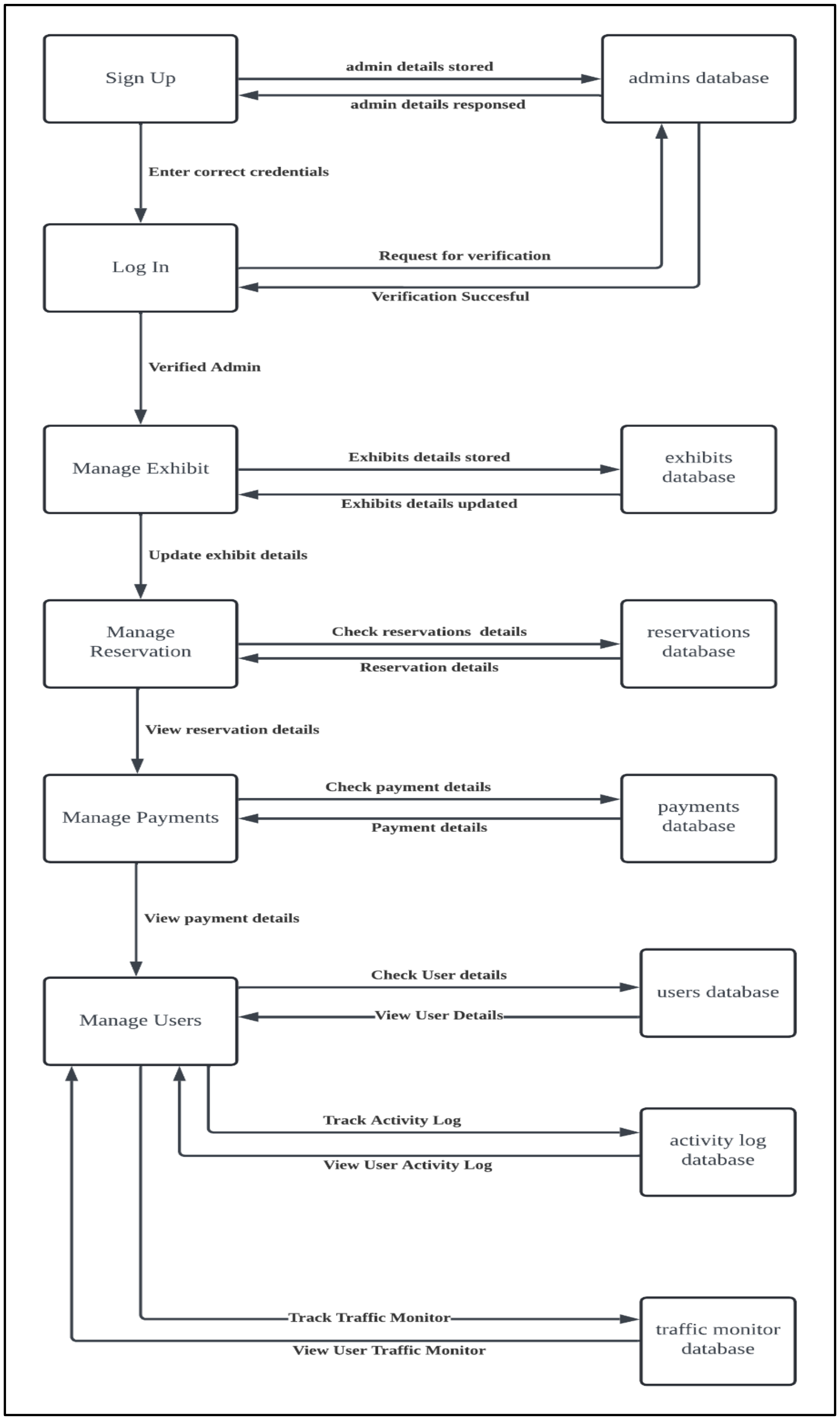


Figure 1.2: Admin Workflow on Website

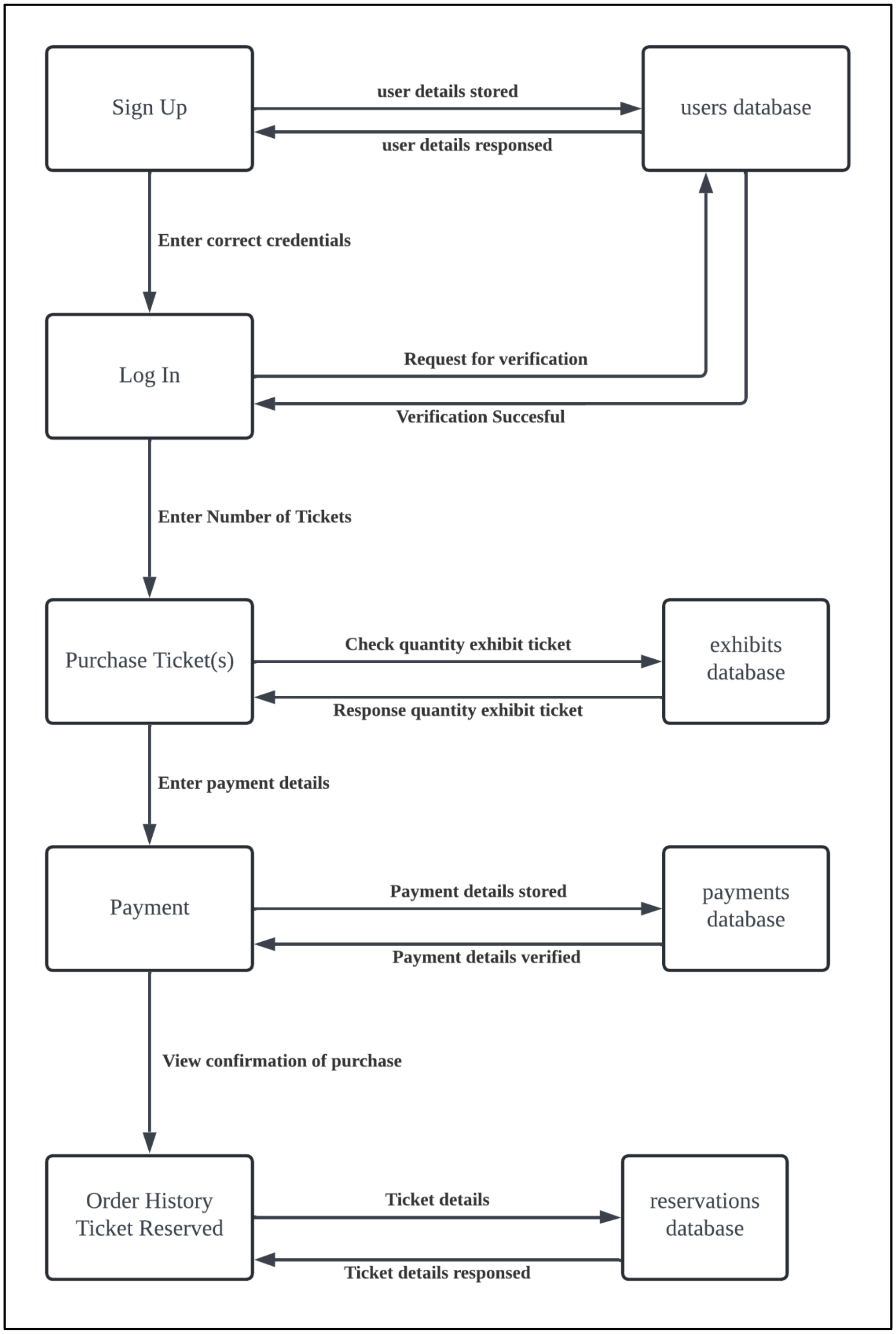


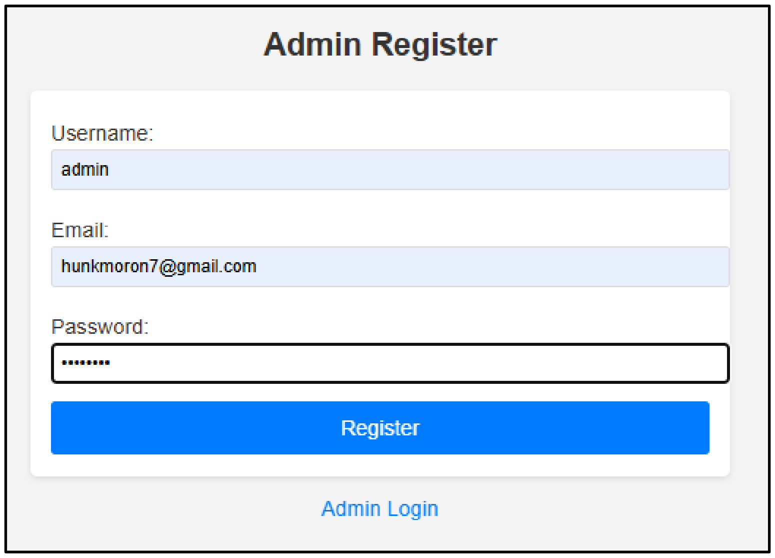
Figure 1.3: User Workflow on Website.

2. <u>Whole Workflow Application Screenshot</u>

a. Administrator

- **Register**

  Security Measure: The script employs multiple layers of security measures to ensure the safe handling of user registration data. It uses the htmlspecialchars() function to sanitize user inputs for username and email, mitigating Cross-Site Scripting attacks. To safeguard passwords, the password_hash() function employs the PASSWORD_BCRYPT algorithm, making them difficult to crack even if the database is compromised. Prepared statements via the prepare() and bind_param() methods effectively prevent SQL Injection attacks by treating user inputs as data, not executable code. HTML pattern attributes further reinforce security by enforcing format requirements for username and password inputs. Additionally, thorough error handling mechanisms ensure prompt feedback in case of issues, aiding in debugging and maintaining a smooth registration process. Upon successful registration, users are redirected to the login page, minimizing form resubmission vulnerabilities.



Username: admin

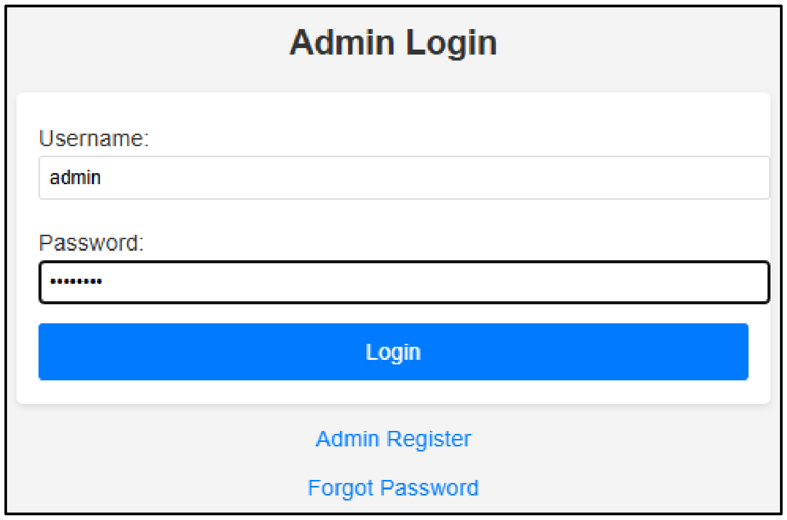
Email: hunkmoron7@gmail.com

Password: ••••••••

Figure 2.1: Admin can register their account by inserting the Username, Email and Password.

- **Login**

  Security Measure: The admin_login.php file is responsible for handling the login process for administrators. It initializes a session for tracking login status and sanitizes the username input to prevent XSS vulnerabilities. The script verifies the entered password against a hashed version stored in the database using password_verify(). Prepared statements are employed to mitigate SQL Injection risks. Upon successful login, the admin's ID is securely stored in a session variable, protecting against session hijacking. Error handling mechanisms provide feedback for invalid credentials, guiding users. Upon successful authentication, the script redirects the user to the admin dashboard, enhancing the user experience and maintaining security by preventing access to the login page after login.



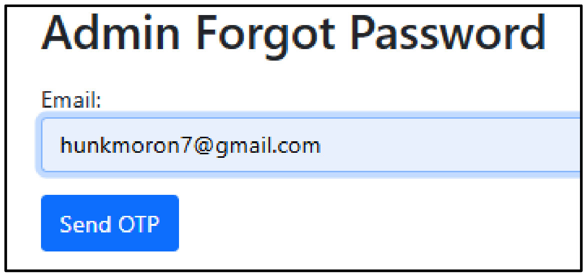
Username: admin

Password: ••••••••

Figure 2.2: After successful registration, admin can login to their account using their Username and Password.

- **Admin Forgot Password**

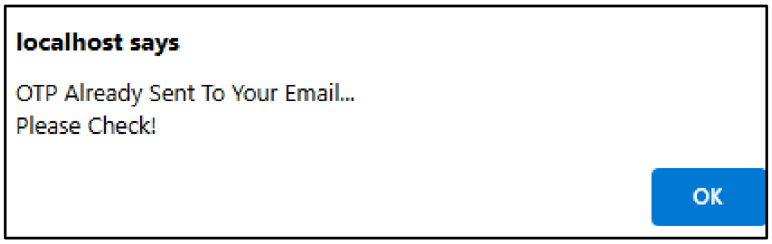
  Security Measure: The admin_forgotpwd.php file facilitates password resets for administrators through a One-Time Password (OTP) mechanism. It initializes a session to maintain process flow and sanitizes email input to thwart XSS attacks. Upon receiving a valid email, a random 6-digit OTP is generated, stored in the session, and dispatched securely to the user's email using PHPMailer with SMTP and STARTTLS encryption. Robust error handling mechanisms ensure feedback for email delivery issues. The user-submitted OTP is then validated against the session-stored OTP. Upon successful validation, the user proceeds to password reset, where the new password is hashed with bcrypt before storage. HTML pattern attributes ensure the new password's strength. Session variables track the process, and error feedback assists users throughout.


Admin Forgot Password

Email:

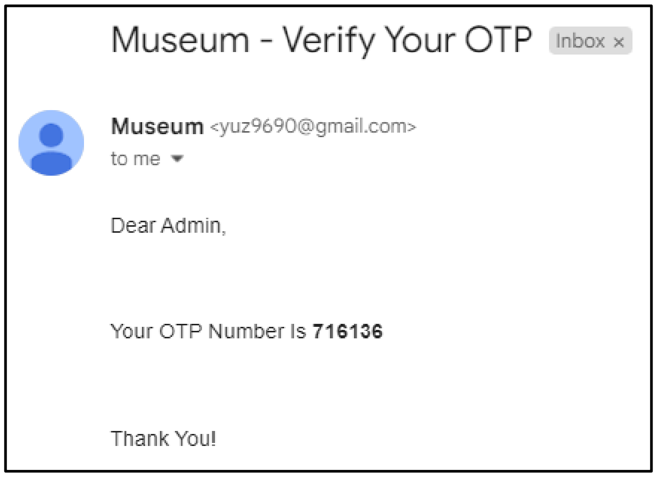
hunkmoron7@gmail.com

Figure 2.3: If the admin forgot their password, they could click the forgot password page and insert their Email to retrieve the OTP via Email.


localhost says

OTP Already Sent To Your Email...
Please Check!

Figure 2.4: After inserting the Email, it will pop up the notification that OTP has sent to their Email.


Museum - Verify Your OTP

Museum <yuz9690@gmail.com>

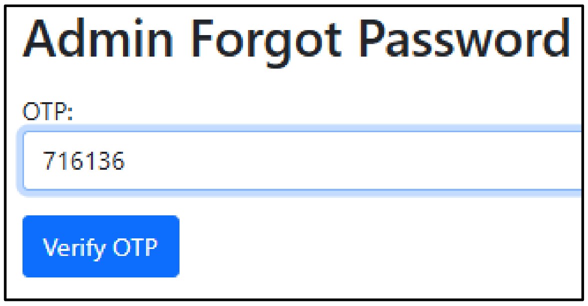
to me

Dear Admin,

Your OTP Number Is **716136**

Thank You!

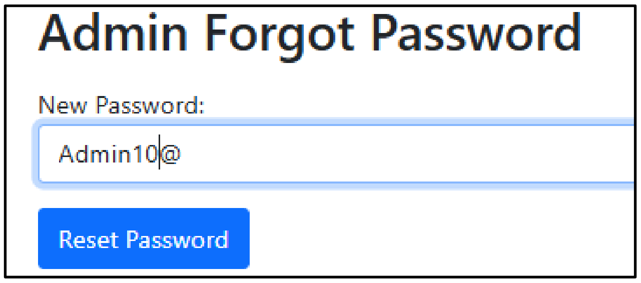
Figure 2.5: Admin can check their Email to check the OTP Number.


Admin Forgot Password

OTP:

716136

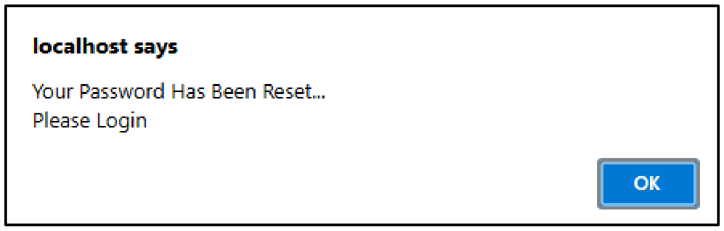
Figure 2.6: Admin can insert the OTP from Email and verify it.


Admin Forgot Password

New Password:

Admin10@

Figure 2.7: After successfully verifying OTP, it will ask the admin to insert a New Password.



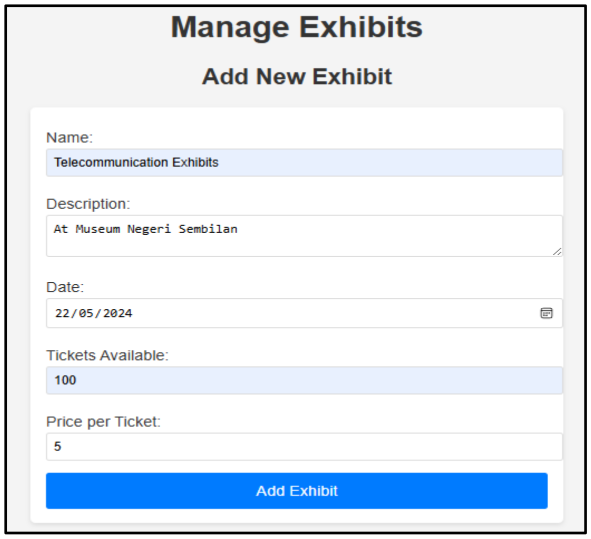
localhost says

Your Password Has Been Reset...
Please Login

Figure 2.8: After the admin clicks the reset password button, it will pop up the notification that mentioned the password has been reset.

- **Manage Exhibits**

  Security Measure: The exhibits.php and edit_exhibit.php files work in tandem to handle exhibit management tasks. Both files initiate a session to verify the admin's login status, preventing unauthorized access. Input sanitization through htmlspecialchars() is employed to mitigate XSS vulnerabilities. Prepared statements are utilized for database operations (INSERT, UPDATE, DELETE) in exhibits.php to thwart SQL Injection attacks. Error handling mechanisms ensure smooth operation and prompt feedback. edit_exhibit.php processes form submissions for editing exhibit details, enforcing redirection for invalid access, and utilizing hidden fields for data persistence. Both files use HTML form attributes for client-side validation, ensuring data integrity before server processing.



Manage Exhibits

Add New Exhibit

Name: Telecommunication Exhibits

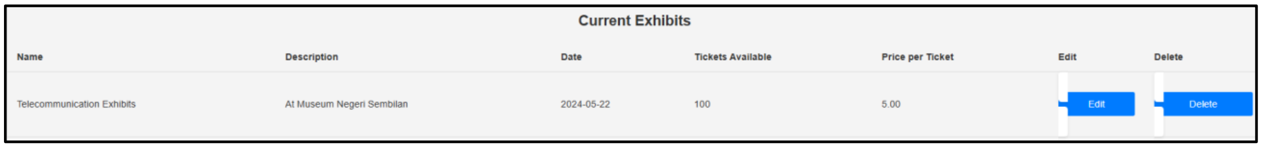
Description: At Museum Negeri Sembilan

Date: 22/05/2024

Tickets Available: 100

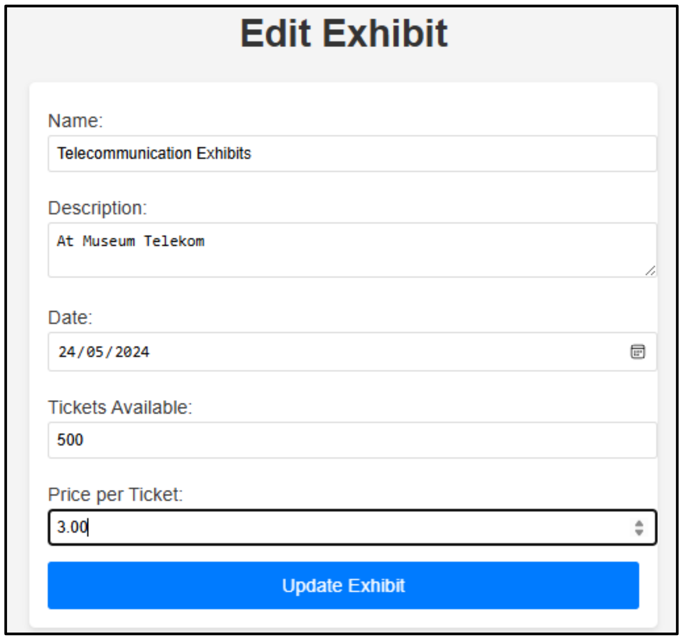
Price per Ticket: 5

Figure 2.9: Admin can add the exhibit by inserting name, description, date, ticket available and ticket price.


Current Exhibits

Name	Description	Date	Tickets Available	Price per Ticket	Edit	Delete
Telecommunication Exhibits	At Museum Negeri Sembilan	2024-05-22	100	5.00	Edit	Delete

Figure 2.10: After clicking the Add Exhibit button, it will add the exhibits into the current exhibits. Admin also can edit or delete the exhibits.


Edit Exhibit

Name: Telecommunication Exhibits

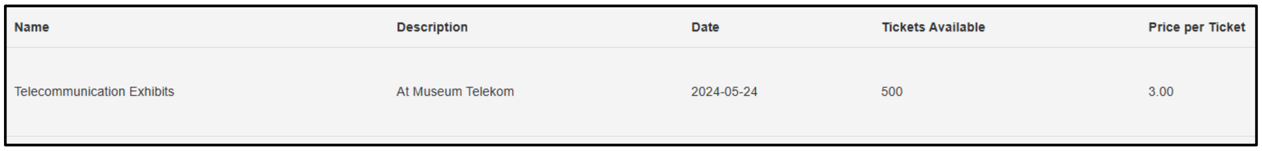
Description: At Museum Telekom

Date: 24/05/2024

Tickets Available: 500

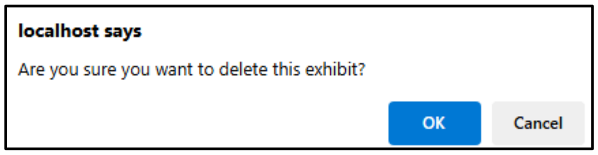
Price per Ticket: 3.00

Figure 2.11: Admin can edit and change any information about the exhibit.



Name	Description	Date	Tickets Available	Price per Ticket
Telecommunication Exhibits	At Museum Telekom	2024-05-24	500	3.00

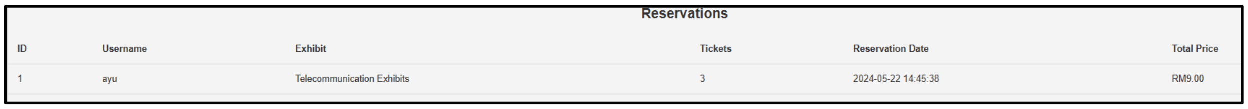
Figure 2.12: After clicking the Update Exhibit button, it will update to the dashboard.


localhost says

Are you sure you want to delete this exhibit?

Figure: 2.13: Admin can also delete the exhibit.

- **Manage Reservations**



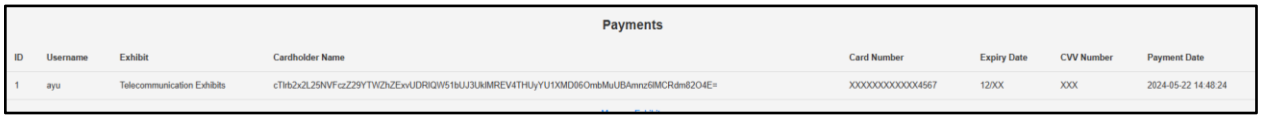
Reservations

ID	Username	Exhibit	Tickets	Reservation Date	Total Price
1	ayu	Telecommunication Exhibits	3	2024-05-22 14:45:38	RM9.00

Figure 2.14: Admin can also view the reservations.

- **Manage Payments**

  Security Measure: The managed payments demonstrate several security measures employed to protect sensitive cardholder data. Firstly, the cardholder's name is encrypted rendering it unreadable without the decryption key. Secondly, the card number is masked, displaying only the last four digits for identification while safeguarding the remaining digits. Additionally, the expiry date and CVV number are partially masked, adding another layer of protection against unauthorized transactions. While not explicitly shown, the presence of usernames and payment dates suggests the implementation of session management for controlling access to this information. Finally, the inclusion of payment dates indicates the maintenance of audit trails for monitoring and investigating transactions. These combined measures contribute to a more secure environment for handling cardholder data, reducing the risk of unauthorized access and potential fraud.



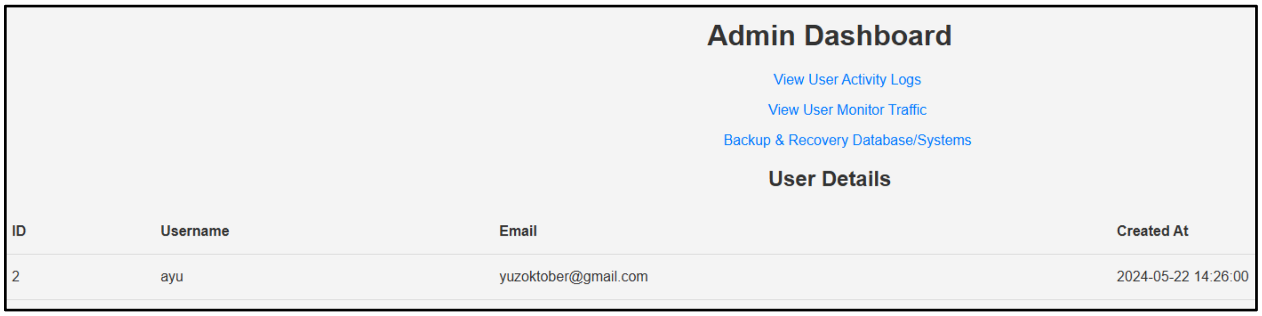
Payments

ID	Username	Exhibit	Cardholder Name	Card Number	Expiry Date	CVV Number	Payment Date
1	ayu	Telecommunication Exhibits	cTlrb2x2L25NVFczZ29YTWZhZExvUDRIQW51bUJ3UkIMREV4THUyYU1XMD06OmbMuUBAmnz6lMCRdm82O4E=	XXXXXXXXXXXX4567	12/XX	XXX	2024-05-22 14:48:24

Figure 2.15: Admin can view the payment details that have been paid from users.

- **Manage Users**

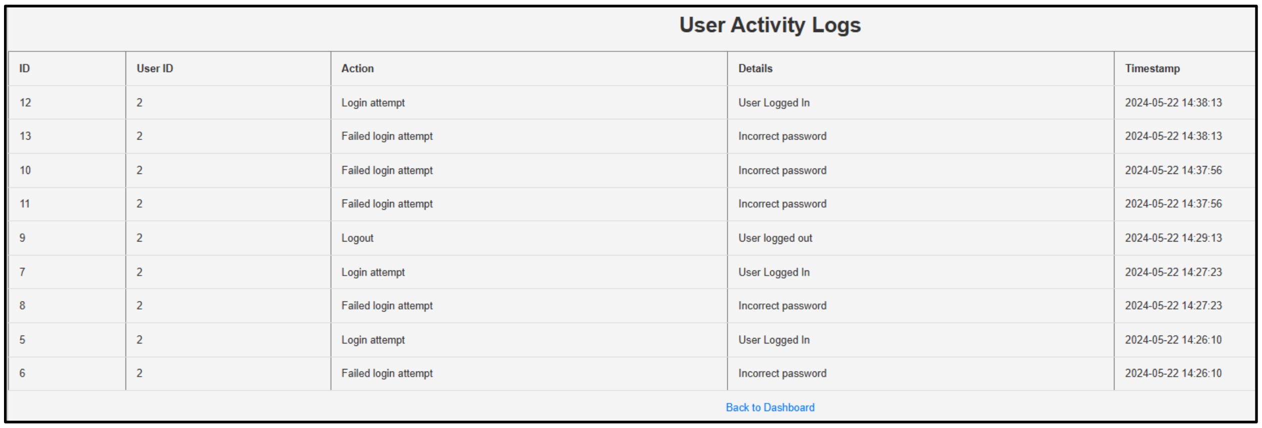
  Security Measure: The provided PHP files serve the purpose of admin monitoring and system backup. admin_activity_logs.php and admin_monitor_traffic.php are responsible for displaying user activity logs and monitoring traffic respectively, both protected by session management and enhanced with error reporting for debugging. backup_system.php facilitates comprehensive backups, encompassing a SQL dump of the entire database with data sanitization and a ZIP archive of the project directory, ensuring data integrity and easy restoration.



User Details

ID	Username	Email	Created At
2	ayu	yuzoktober@gmail.com	2024-05-22 14:26:00

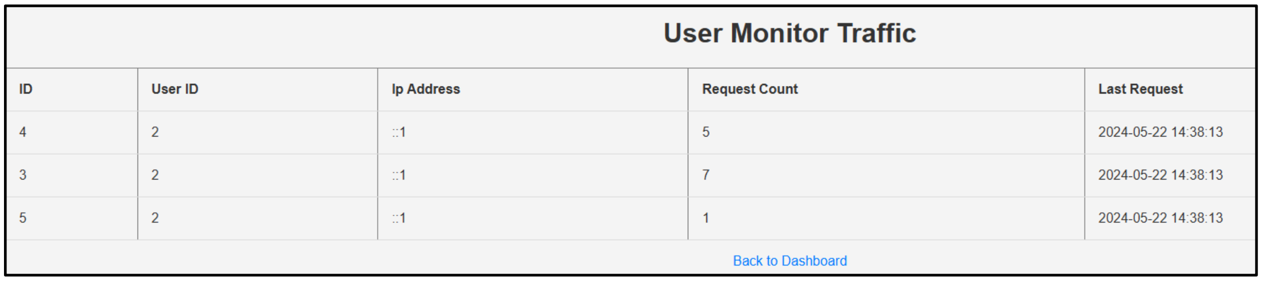
Figure 2.16: Admin can view the users' details.



User Activity Logs

ID	User ID	Action	Details	Timestamp
12	2	Login attempt	User Logged In	2024-05-22 14:38:13
13	2	Failed login attempt	Incorrect password	2024-05-22 14:38:13
10	2	Failed login attempt	Incorrect password	2024-05-22 14:37:56
11	2	Failed login attempt	Incorrect password	2024-05-22 14:37:56
9	2	Logout	User logged out	2024-05-22 14:29:13
7	2	Login attempt	User Logged In	2024-05-22 14:27:23
8	2	Failed login attempt	Incorrect password	2024-05-22 14:27:23
5	2	Login attempt	User Logged In	2024-05-22 14:26:10
6	2	Failed login attempt	Incorrect password	2024-05-22 14:26:10

Figure 2.17: Admin can view the user activity logs to track which user login and logout.

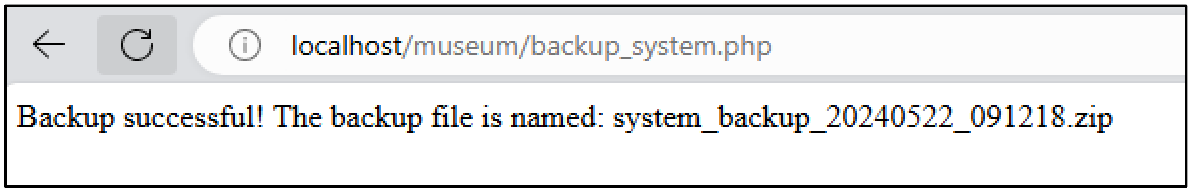


User Monitor Traffic

ID	User ID	Ip Address	Request Count	Last Request
4	2	::1	5	2024-05-22 14:38:13
3	2	::1	7	2024-05-22 14:38:13
5	2	::1	1	2024-05-22 14:38:13

Back to Dashboard

Figure 2.18: Admin can monitor the user traffic.



localhost/museum/backup_system.php

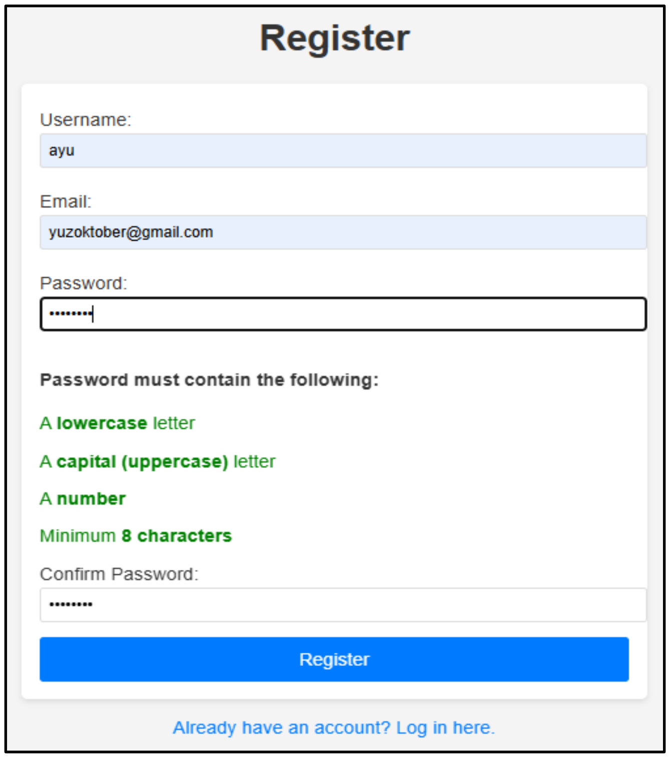
Backup successful! The backup file is named: system_backup_20240522_091218.zip

Figure 2.19: Admin can also back up the database and whole system.

b. User

- **Register**

  Security Measure: The provided code securely handles user registration through a combination of session management, input sanitization, server-side validation, and password hashing. It initializes a session, establishes a database connection, and enables error reporting for development purposes. User inputs for usernames and email are sanitized using htmlspecialchars() to prevent XSS attacks. Server-side validation ensures all fields are filled and checks for valid email and username formats, as well as password strength and confirmation. The password is securely hashed using bcrypt before being stored. Prepared statements with bind_param() are used to prevent SQL injection. Error handling provides user feedback, HTML form validation, along with real-time JavaScript password validation, enhances the user experience.



Username: ayu

Email: yuzoktober@gmail.com

Password: ••••••••

Password must contain the following:

A **lowercase** letter

A **capital (uppercase)** letter

A **number**

Minimum **8 characters**

Confirm Password: ••••••••

Figure 2.20: Users can register their account by inserting their Username, Email and Password.

- **Login**

  Security Measure: The login.php file securely manages user logins through a multi-faceted approach. It initiates a session, establishes a database connection, and incorporates activity logging for enhanced security. Client IP addresses are retrieved for monitoring and rate limiting, which prevents brute-force attacks by restricting the number of logins attempts from a specific IP. User input validation ensures required fields are filled, and a prepared statement prevents SQL injection vulnerabilities. Password verification securely compares the entered password with the hashed version stored in the database. Both successful and failed login attempts are meticulously logged for audit purposes. Upon successful login, the user's ID is stored in a session variable, maintaining authentication across pages. The script also monitors traffic, contributing to the rate-limiting mechanism, and provides informative feedback to users in case of errors.

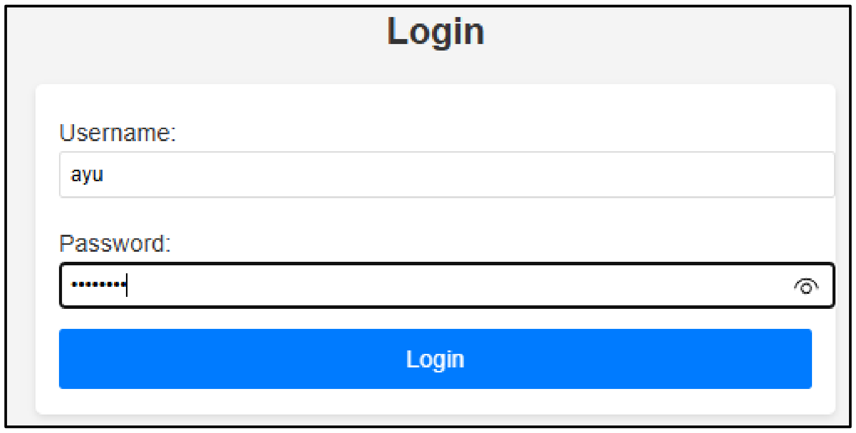


Figure 2.21: Users can login to their account by inserting their Username and Password.

- **User Forgot Password**

  Security Measure: The forgotpwd.php file enables users to reset their passwords securely through a One-Time Password (OTP) process. It initializes a session for state management and sanitizes email input to prevent XSS vulnerabilities. Prepared statements are used to interact with the database, ensuring protection against SQL injection attacks. A randomly generated OTP is sent securely via PHPMailer with encryption, and this OTP is verified against user input for identity confirmation. Upon successful verification, the user can reset their password, which is then securely hashed with bcrypt before being stored. Throughout the process, the script provides clear feedback and redirects users as needed to streamline the password reset experience while maintaining security.

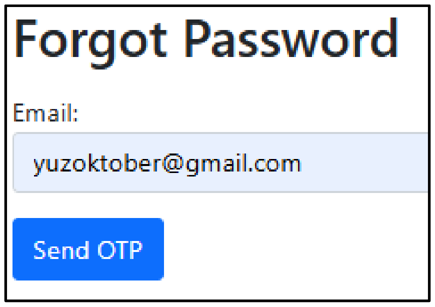
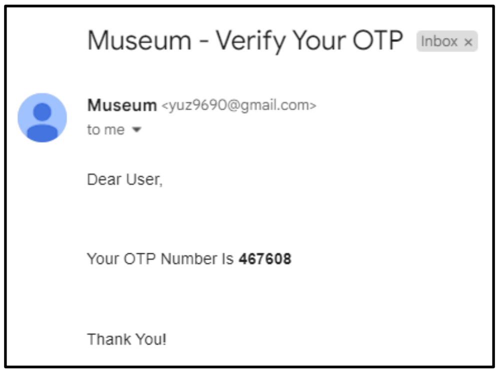


Figure 2.22: If the user forgot their password, they could click the forgot password page and insert their Email to retrieve the OTP via Email.


Museum - Verify Your OTP Inbox ×

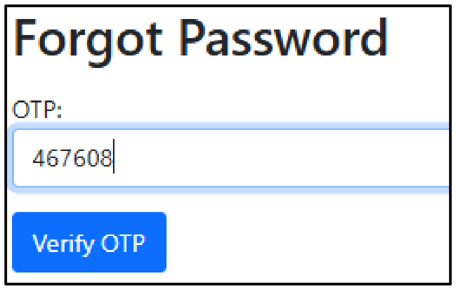
Museum <yuz9690@gmail.com>
to me

Dear User,

Your OTP Number Is **467608**

Thank You!

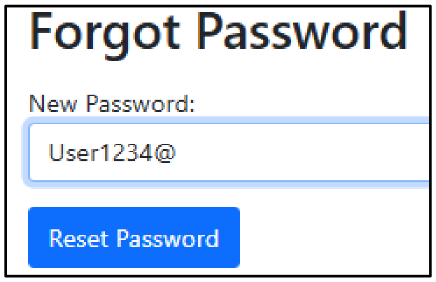
Figure 2.23: Users can check their Email to check the OTP Number.


Forgot Password

OTP:

467608

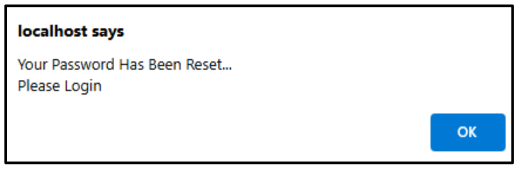
Figure 2.24: Users can insert the OTP from Email and verify it.


Forgot Password

New Password:

User1234@

Figure 2.25: After successfully verifying OTP, it will ask the user to insert a New Password.

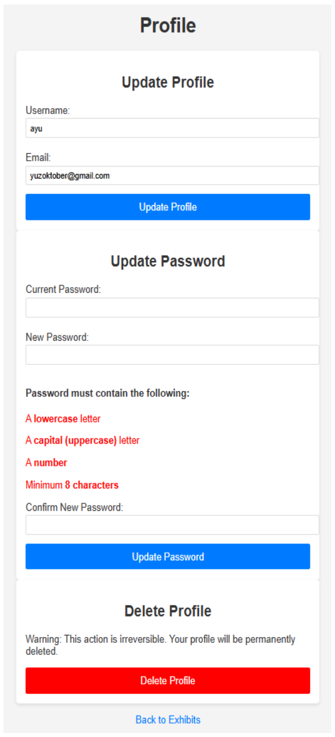

localhost says

Your Password Has Been Reset...
Please Login

Figure 2.26: After the user clicks the reset password button, it will pop up the notification that mentioned the password has been reset.

- **Manage Profile**

  Security Measure: The profile.php file is designed to manage user profiles securely. It starts by initiating a session to track user actions within the profile section. Any user inputs are meticulously sanitized using the htmlspecialchars() function, mitigating the risk of Cross-Site Scripting (XSS) attacks. When updating profile information, the script employs prepared statements with the prepare() and bind_param() methods. This practice safeguards against SQL Injection attacks by ensuring that user-provided values are treated as data and not as part of the SQL query. In cases where the user modifies their password, the script hashes the new password using password_hash() with the robust PASSWORD_BCRYPT algorithm. Finally, to maintain consistency between the database and the user's session, relevant session variables are updated to reflect the changes made to the user's profile.



Profile

Update Profile

Username: ayu

Email: yuzoktober@gmail.com

Update Password

Current Password:

New Password:

Password must contain the following:

A lowercase letter

A capital (uppercase) letter

A number

Minimum 8 characters

Confirm New Password:

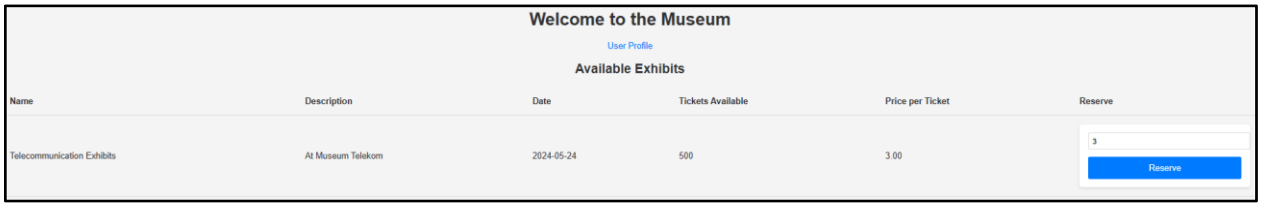
Delete Profile

Warning: This action is irreversible. Your profile will be permanently deleted.

Figure 2.27: Users can update their profile which is username and email, password and delete their profile.

- **Reserve Tickets**

  Security Measure: The reserve.php file securely handles exhibit reservations by utilizing session management, input sanitization, and prepared statements. It starts by initializing a session to track user interactions and ensures data integrity by sanitizing all inputs using htmlspecialchars(), effectively mitigating XSS attacks. To further enhance security, the script employs prepared statements with prepare() and bind_param() for all database operations, particularly when inserting reservation details. This approach prevents SQL Injection attacks by ensuring that user inputs are treated as data and not as part of the SQL query. Additionally, robust error handling mechanisms are in place to identify and address any issues during the reservation process, providing feedback to the user and improving the overall user experience.



Welcome to the Museum

User Profile

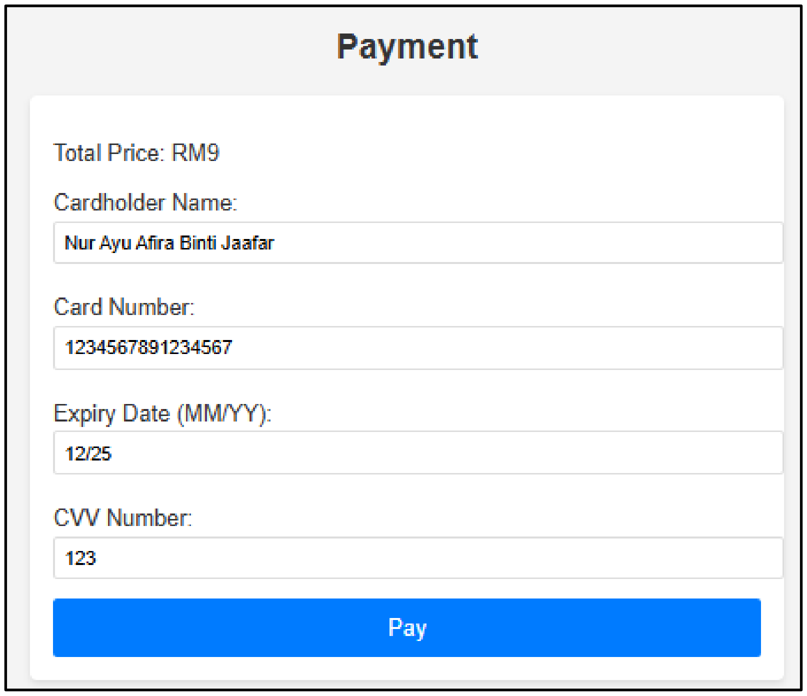
Available Exhibits

Name	Description	Date	Tickets Available	Price per Ticket	Reserve
Telecommunication Exhibits	At Museum Telekom	2024-05-24	500	3.00	3

Figure 2.28: Users can reserve their tickets by inserting the quantity ticket.

- **Purchase Tickets**

  Security Measure: The payment.php file securely processes payments by leveraging session management, input sanitization, and prepared statements. It starts by initializing a session to track user interactions throughout the payment process. To mitigate XSS vulnerabilities, it sanitizes URL parameters and form inputs using htmlspecialchars(). Furthermore, it employs client-side validation through HTML attributes to enforce format requirements for cardholder details like name, number, expiry date, and CVV. Server-side validation complements this by ensuring all required fields are filled before processing. To prevent SQL injection attacks, the script relies on prepared statements with prepare() and bind_param() for all database operations related to storing payment information. Additionally, it prioritizes the protection of sensitive data by masking or encrypting cardholder names and numbers, reducing the risk of unauthorized access.



Payment

Total Price: RM9

Cardholder Name:

Nur Ayu Afira Binti Jaafar

Card Number:

1234567891234567

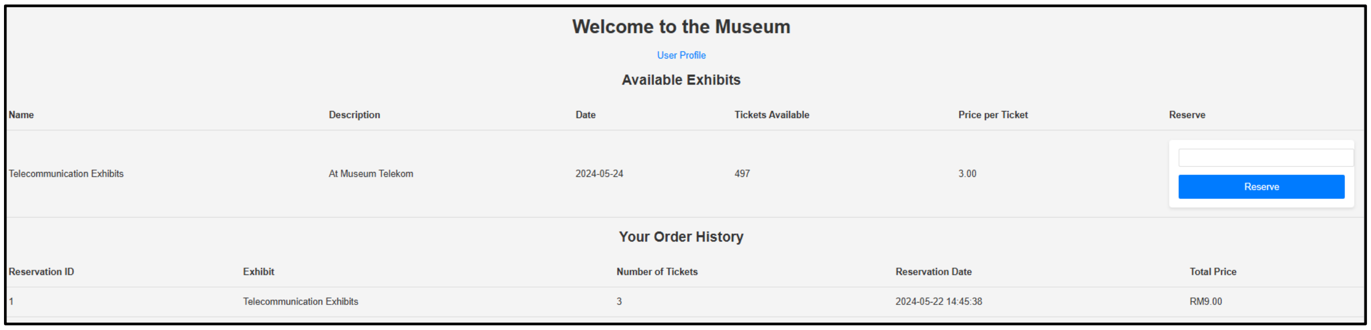
Expiry Date (MM/YY):

12/25

CVV Number:

123

Figure 2.29: After clicking the reserve button, it will direct to the payment page. Next, the users need to insert their payment details.



Welcome to the Museum

User Profile

Available Exhibits

Name	Description	Date	Tickets Available	Price per Ticket	Reserve
Telecommunication Exhibits	At Museum Telekom	2024-05-24	497	3.00	Reserve

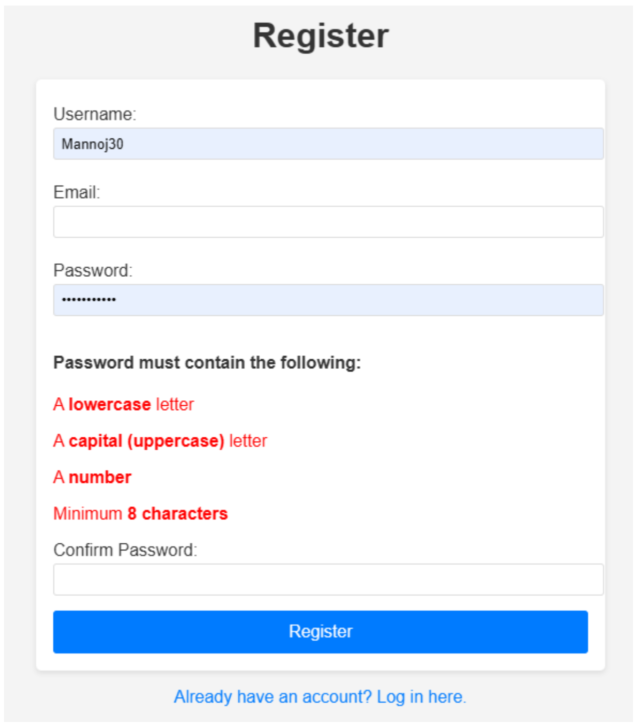
Your Order History

Reservation ID	Exhibit	Number of Tickets	Reservation Date	Total Price
1	Telecommunication Exhibits	3	2024-05-22 14:45:38	RM9.00

Figure 2.30: After clicking the pay button, it will direct to the index page and the users can view their payment in the order history.

Security Measures Implementation

Authentication: In terms of authentication, the user must sign up to purchase tickets. Then the user will need to enter the sign-up details into the login page to log in to their respective accounts. The username and password on the sign-up page must be watched with the login page for a successful signup and log in for a new user. The data of the user will be stored in the user database.



Register

Username: Mannoj30

Email:

Password: ••••••••••

Password must contain the following:

A **lowercase** letter

A **capital (uppercase)** letter

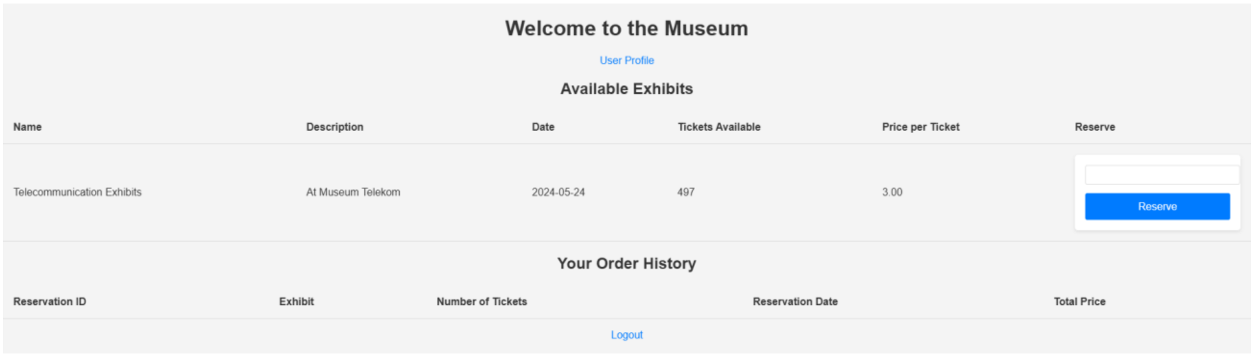
A **number**

Minimum **8 characters**

Confirm Password:

Figure 3.1: Register Page

Authorization: Based on the website, authorization can be classed in a way such that users are only limited to certain actions as well as what they can see. The user can only look at the current shows available, then proceed to purchase a certain number of tickets. Finally purchase them using his or her credit/debit card details.



Welcome to the Museum

User Profile

Available Exhibits

Name	Description	Date	Tickets Available	Price per Ticket	Reserve
Telecommunication Exhibits	At Museum Telekom	2024-05-24	497	3.00	Reserve

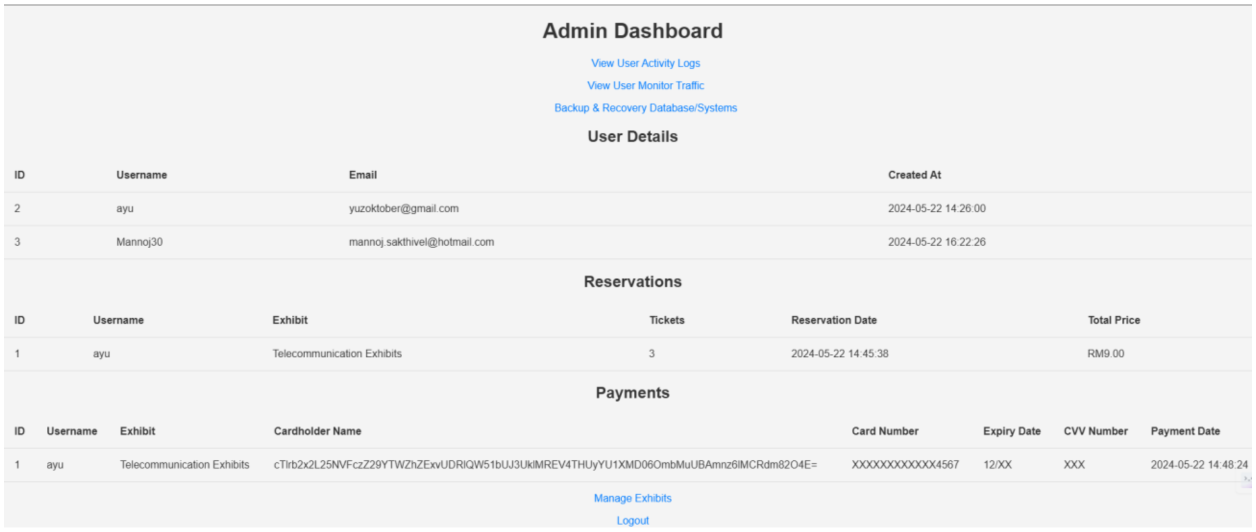
Your Order History

Reservation ID	Exhibit	Number of Tickets	Reservation Date	Total Price

Logout

Figure 3.2: Customer View

The admin on the other hand, can view user details, reservations of tickets and payments. As shown at the top the admin also can view user activity logs, user monitor traffic and backup the database. The admin can also manage the shows by adding, modifying, and deleting a particular show.



Admin Dashboard

View User Activity Logs
View User Monitor Traffic
Backup & Recovery Database/Systems

User Details

ID	Username	Email	Created At
2	ayu	yuzoktober@gmail.com	2024-05-22 14:26:00
3	Mannoj30	mannoj.sakthivel@hotmail.com	2024-05-22 16:22:26

Reservations

ID	Username	Exhibit	Tickets	Reservation Date	Total Price
1	ayu	Telecommunication Exhibits	3	2024-05-22 14:45:38	RM9.00

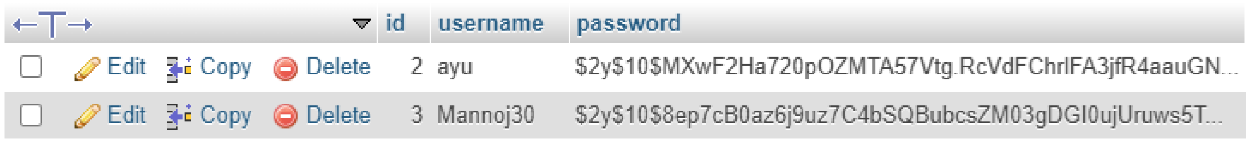
Payments

ID	Username	Exhibit	Cardholder Name	Card Number	Expiry Date	CVV Number	Payment Date
1	ayu	Telecommunication Exhibits	cTlrb2x2L25NVFczZ29YTWZhZExvUDRIQW51bUJ3UklMREV4THUyYU1XMD06OmbMuUBAmnz6lMCRdm82O4E=	XXXXXXXXXXXX4567	12/XX	XXX	2024-05-22 14:48:24

Manage Exhibits
Logout

Figure 3.3: Admin View

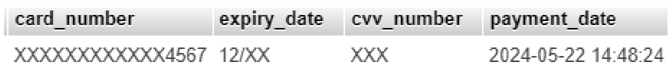
Hashing: Hashing is done through the likes of data stored in the database. Every time a new user creates an account, the user's password will be hashed in the database. Thus, this process implements data integrity. Since the password is hashed in the database, an unauthorized user cannot obtain the password. This is since the password is hashed randomly and can be detected easily if the database has been altered since the hash value is very particular and will change completely even if it's slightly altered.



←T→				id	username	password
☐	Edit	Copy	Delete	2	ayu	$2y$10$MXwF2Ha720pOZMTA57Vtg.RcVdFChrlFA3jfR4aauGN...
☐	Edit	Copy	Delete	3	Mannoj30	$2y$10$8ep7cB0az6j9uz7C4bSQBubcsZM03gDGI0ujUruws5T...

Figure 3.4: Hashing of Passwords

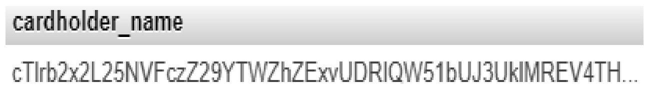
Masking: The deployment of masking techniques as a security measure in a web application is depicted in Figure 3.5 below. Credit card numbers and other sensitive data are protected with this method. XXXX XXXX XXXX 4567 is the card number that is partially hidden. The remaining four digits are the only ones that are visible. 'X's are used in place of the first twelve numbers to keep the entire card number concealed. By using this procedure, the likelihood of unwanted access and possible card information misuse is greatly decreased. Furthermore, there is a partial masking of the expiration date. Month is visible, but the year is hidden by substituting 'XX', so the entire expiration date is hidden. By keeping potential attackers from getting the complete facts of the card's validity term, this provides an extra degree of security. Finally, 'XXX', the CVV number, is completely hidden. It protects the confidentiality of this vital security feature by not displaying any portion of the CVV.



card_number	expiry_date	cvv_number	payment_date
XXXXXXXXXXXX4567	12/XX	XXX	2024-05-22 14:48:24

Figure 3.5: Masking of card_number, expiry_date, cvv_number

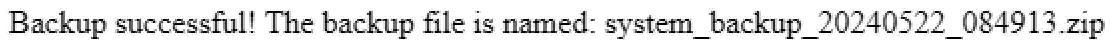
Encryption: Figure 3.6 below shows an encrypted cardholder_name that exemplifies the use of encryption to protect sensitive personal information. In this context, encryption transforms the plaintext cardholder's name into a ciphertext using a cryptographic algorithm (aes-256-cbc) and an encryption key. This process ensures that the data remains confidential, readable only by entities possessing the decryption key. Such encryption is crucial in protecting personal and financial data against unauthorized access and breaches, particularly in handling sensitive information like banking and online transactions (such as used in our website). By encrypting this information, our website complies with data protection regulations and reduces the risk of data theft, ensuring that even if the data is intercepted, it cannot be misused without the corresponding decryption mechanism.



cardholder_name
cTlrb2x2L25NVFczZ29YTWZhZExvUDRIQW51bUJ3UklMREV4TH...

Figure 3.6 cardholder_name being encrypted.

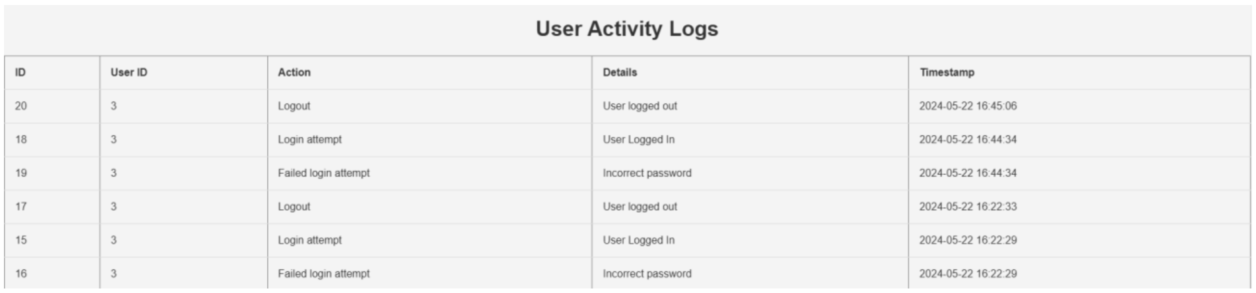
Backup and Disaster Recovery: The backup process in the provided PHP script involves creating a SQL dump of the database and compressing the project directory into a zip file. It starts by retrieving all table names from the database and generating SQL CREATE TABLE statements and INSERT statements for each table's data. These statements are written to a file named with a timestamp for uniqueness. Additionally, the entire project directory is recursively zipped into another file, also timestamped. The script ensures that both the database and application files can be restored if needed. To secure the backup process, it is crucial to use proper file permissions, consider data encryption, and implement regular backup schedules to maintain data integrity and availability.



```
Backup successful! The backup file is named: system_backup_20240522_084913.zip
```

Figure 3.7: Indication of a Successful Backup

Audit Logs: "User Activity Logs" in Figure 3.8 below shows a detailed log of user activities, including login attempts, unsuccessful login attempts, logins, and logouts. Every record has a unique timestamp, the user ID, the particular action taken, and details about the action. Administrators may actively monitor user behavior, identify any unusual activity, and quickly address possible security breaches thanks to this thorough tracking. To safeguard user accounts, additional research or prompt action may be necessary if multiple unsuccessful attempts at login are indicative of a potential brute-force attack.

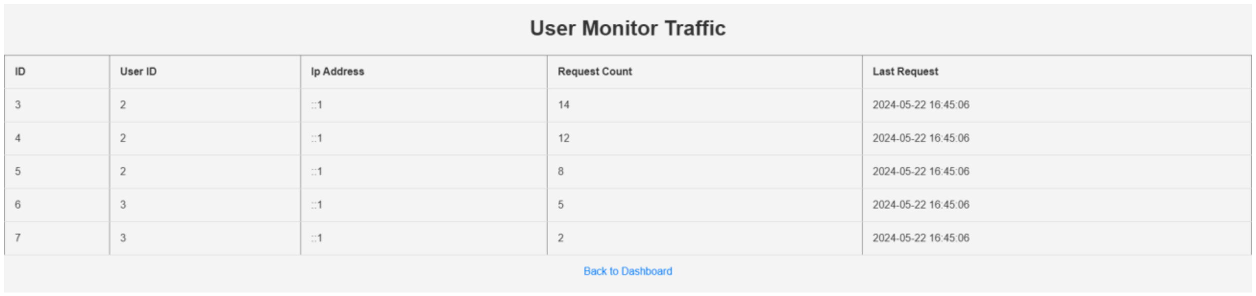


User Activity Logs

ID	User ID	Action	Details	Timestamp
20	3	Logout	User logged out	2024-05-22 16:45:06
18	3	Login attempt	User Logged In	2024-05-22 16:44:34
19	3	Failed login attempt	Incorrect password	2024-05-22 16:44:34
17	3	Logout	User logged out	2024-05-22 16:22:33
15	3	Login attempt	User Logged In	2024-05-22 16:22:29
16	3	Failed login attempt	Incorrect password	2024-05-22 16:22:29

Figure 3.8: User Activity Logs Page

Monitor Traffic: Figure 3.9 below shows a 'User Monitor Traffic" that logs user requests. Each entry in the table includes an ID, user ID, IP address, request count, and the timestamp of the last request. This log provides a granular view of user interactions with the application, including the frequency and timing of their requests. By tracking the IP addresses and the number of requests, administrators can identify unusual patterns, such as a sudden spike in activity from a single IP address, which may signify a distributed denial-of-service (DDoS) attack or other malicious behavior.

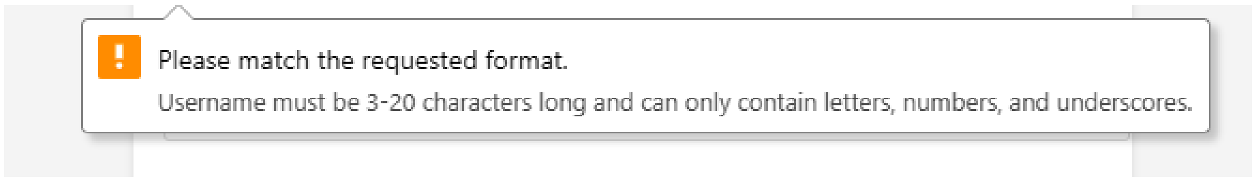


User Monitor Traffic

ID	User ID	Ip Address	Request Count	Last Request
3	2	::1	14	2024-05-22 16:45:06
4	2	::1	12	2024-05-22 16:45:06
5	2	::1	8	2024-05-22 16:45:06
6	3	::1	5	2024-05-22 16:45:06
7	3	::1	2	2024-05-22 16:45:06

Figure 3.9: User Monitor Traffic Page.

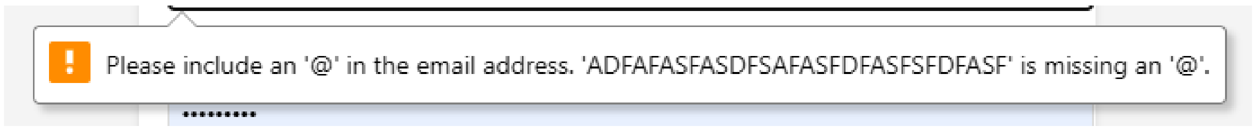
Input Validation: Figure 3.10 below showcases a format requirement for creating a username, serving as a form of input validation in user registration systems. Specifically, it mandates that a username must be between 3 to 20 characters in length and can only include letters, numbers, and underscores. This constraint not only standardizes the format of user identifiers across the system but also enhances security and database integrity by preventing the use of special characters that could potentially be exploited in security attacks. Additionally, the character length limit helps maintain consistency and usability in the system's interface design.



Please match the requested format.
Username must be 3-20 characters long and can only contain letters, numbers, and underscores.

Figure 3.10: Username Validation

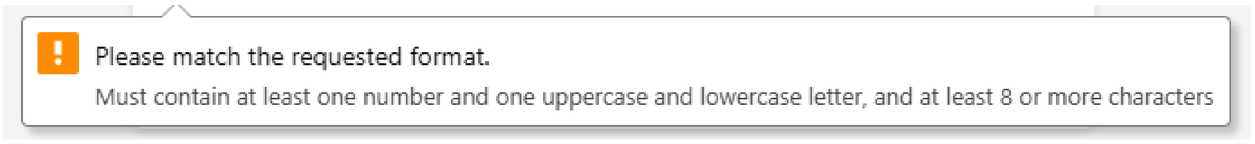
Figure 3.11 below showcases an error message related to input validation for an email address field, indicating that the input provided lacks the '@' symbol, a critical component of a valid email address format. This form of validation ensures that the user enters a correctly formatted email, separating the local part from the domain, which is essential for routing and delivering emails accurately. By enforcing this specific criterion, systems maintain data integrity and functionality, preventing user errors and ensuring that communications are correctly addressed and sent to intended recipients. This method is a standard practice in digital forms and applications where email addresses are required.



Please include an '@' in the email address. 'ADFAFASFASDFSAFASFDFASFSFDFASF' is missing an '@'.

Figure 3.11: Email Validation

Figure 3.12 illustrates input validation rules aimed at enhancing security, specifically for password creation. It mandates that all inputs must contain at least one number and both uppercase and lowercase letters to increase complexity and resist brute-force attacks. Additionally, the requirement for the input to be a minimum of eight characters long ensures that the password is sufficiently complex, making it difficult for potential unauthorized access through common cracking methods. These rules are standard practices in cybersecurity, designed to safeguard user data by preventing the use of weak and easily guessable passwords.



Please match the requested format.
Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters

Figure 3.12: Password Validation

Threat Modelling (STRIDE and DREAD)

Spoofing Identity	To prevent spoofing: - The use of strong authentication mechanisms and password policies.
Tampering with data	To prevent tampering with data: - The application's usage of role-based access control to grant authorized people access and modification rights.
Repudiation	To prevent repudiation: - The use of auditing and logging to document all database activity, including transactions and user actions. By doing this, a transparent audit trail that can be used to confirm actions made inside the database will be established.
Information Disclosure	To prevent information disclosure: - Adding role-based permissions and access control to the database to restrict access to private data. To prevent unauthorized access to sensitive data, data encryption is used.
Denial of Service	To prevent denial of service: - Our web application implements a comprehensive traffic monitoring system, enforces time limits, and applies request limits to detect and prevent excessive traffic, ensuring continuous and reliable service availability.
Elevation of Privileges	To prevent elevation of privileges: - To restrict the permissions of unapproved workers, we use role-based permissions and access control. - To consider any modifications to the roles within the organization and database, we will examine and update the access restrictions and permissions.

Character	Acronym for	Rating
D	Damage potential	1 for minimal damage, 10 indicates a catastrophically vulnerability
R	Reproducibility	1 is almost impossible to reproduce, 10 is very easy to reproduce, more likely to be automated attacks like Bots.
E	Exploitability	1 Indicates a vulnerability is difficult to exploit. 10 is easily exploitable.
A	Affected Users	Take the percentage of users affected and divide by 10 and round to the nearest whole number. E.g. if 80% of users are affected, the rating is 8. For 25% the rating is 3.
D	Discoverability	1 for obscure vulnerabilities. 10 is easily discovered, such as SQL injection.

Stride Category	Threat Detection	D	R	E	A	D	Threat Rating
S	Unauthorized access, data breaches.	3	1	5	7	5	4.2
T	Incorrect data, operational issues.	2	3	4	6	3	3.6
R	Trust issues, legal disputes.	3	4	3	5	3	3.6
I	Exposure of personal/payment data.	2	3	5	6	5	4.2
D	Service downtime, availability issues.	7	6	7	8	7	7.0
E	Total system compromise, security issues.	4	3	5	7	6	5.0

PDPA 2010

Categorize of Personnel According to PDPA 2010:

1. Data Users (Data Controllers)

- **Definition**: Organizations or individuals who collect, store, and process personal data.
- **Responsibilities**:
    - Ensure compliance with data protection principles.
    - Implement appropriate security measures.
    - Inform data subjects about data processing activities.
    - Respond to data subjects' requests regarding their personal data.

2. Data Processors

- **Definition**: Entities or individuals that process data on behalf of the data user.
- **Responsibilities**:
    - Process data only on the instructions of the data user.
    - Implement appropriate technical and organizational measures to protect personal data.
    - Assist the data user in fulfilling data subject requests.

3. Data Subjects

- **Definition**: Individuals whose personal data is being collected, stored, or processed.
- **Rights**:
    - Right to be informed about the processing of their data.
    - Right to access their personal data.
    - Right to correct inaccurate data.
    - Right to withdraw consent for data processing.
    - Right to prevent processing likely to cause damage or distress.

4. Data Protection Officer (DPO)

- **Definition**: A designated individual within an organization responsible for overseeing data protection strategy and compliance.
- **Responsibilities**:
    - Ensure the organization’s compliance with the PDPA.
    - Conduct regular data protection audits.
    - Serve as a point of contact for data protection authorities and data subjects.
    - Provide training and awareness programs on data protection.

5. Third-Party Service Providers

- **Definition**: External entities that provide services to the data user, which may involve processing personal data.
- **Responsibilities**:
    - Ensure compliance with the data protection requirements as stipulated in their contract with the data user.
    - Implement adequate security measures to protect personal data.
    - Report any data breaches to the data user promptly.

6. Data Protection Authorities

- **Definition**: Government bodies responsible for enforcing data protection laws and regulations.
- **Responsibilities**:
    - Monitor and enforce compliance with the PDPA.
    - Investigate complaints and conduct audits.
    - Issue guidance and regulations regarding data protection.
    - Impose penalties and sanctions for non-compliance.

7. Internal Audit and Compliance Teams

- **Definition**: Teams within an organization responsible for ensuring internal policies and procedures comply with the PDPA.
- **Responsibilities**:
  - Conduct regular audits to ensure data protection practices are followed.
  - Review and update data protection policies and procedures.
  - Provide recommendations for improving data protection measures.

By categorizing personnel in this way, organizations can ensure that all roles and responsibilities related to data protection are clearly defined and managed according to the requirements of the PDPA 2010.

Mapping Data Lifecycle to PDPA 2010 Requirements:

1. Collection

- **PDPA Requirements**:
  - **Notice and Consent**: Inform data subjects about the purpose of data collection and obtain their explicit consent.
  - **Purpose Limitation**: Collect data only for specified, legitimate purposes.
  - **Data Minimization**: Collect only the data necessary for the specified purposes.

2. Storage

- **PDPA Requirements**:
  - **Security Measures**: Implement appropriate technical and organizational measures to protect personal data against loss, misuse, unauthorized access, disclosure, alteration, and destruction.
  - **Data Integrity**: Ensure that the data stored is accurate, complete, and kept up to date.

        - **Retention Policies**: Establish clear data retention policies, ensuring data is not kept longer than necessary.

3. Usage

- **PDPA Requirements**:

    - **Purpose Adherence**: Use data strictly for the purposes for which it was collected, as stated in the notice to the data subjects.
    - **Access Control**: Restrict access to personal data to authorized personnel only.
    - **Data Quality**: Ensure the data remains accurate, relevant, and up to date for its intended use.

4. Sharing and Disclosure

- **PDPA Requirements**:

    - **Third-Party Agreements**: Ensure third parties processing personal data on behalf of the data user have adequate data protection measures in place.
    - **Data Transfer Conditions**: Follow specific conditions for transferring data to third parties, especially across borders, ensuring an adequate level of protection.
    - **Consent for Sharing**: Obtain consent from data subjects before sharing their data with third parties, unless exempted by the law.

5. Retention and Disposal

- **PDPA Requirements**:

    - **Data Retention**: Retain data only as long as necessary to fulfill the purposes for which it was collected.
    - **Secure Disposal**: Ensure secure disposal of data when it is no longer needed, preventing unauthorized access or disclosure during the disposal process.

6. Access and Correction

- **PDPA Requirements**:
    - **Access Rights**: Allow data subjects to access their personal data upon request.
    - **Correction Rights**: Enable data subjects to correct inaccurate or incomplete data.
    - **Response to Requests**: Respond to data access and correction requests within a reasonable time frame.

7. Audit and Review

- **PDPA Requirements**:
    - **Compliance Audits**: Regularly conduct audits to ensure compliance with the PDPA requirements.
    - **Policy Review**: Periodically review and update data protection policies and procedures.
    - **Incident Management**: Have procedures in place to manage and report data breaches and other security incidents.

Summary

Data Lifecycle Stage	PDPA 2010 Requirements
Collection	Notice and Consent, Purpose Limitation, Data Minimization
Storage	Security Measures, Data Integrity, Retention Policies
Usage	Purpose Adherence, Access Control, Data Quality
Sharing and Disclosure	Third-Party Agreements, Data Transfer Conditions, Consent for Sharing
Retention and Disposal	Data Retention, Secure Disposal
Access and Correction	Access Rights, Correction Rights, Response to Requests
Audit and Review	Compliance Audits, Policy Review, Incident Management