MMU

MULTIMEDIA UNIVERSITY

# CCS6344 T2410 Assignment 1 Submission

## Group Name: Group16

| | |
|---|---|
| ASYRANI SYAZWAN BIN YUHANIS | 1211103222 |
| OOI PUI KEAT | 1201101582 |
| DIVYASHREE A/P SELVANYGAM | 1221303777 |

YouTube link: https://youtu.be/roMNB26aMFo

## 1. OBJECTIVE

The objective of the project is to develop an application that centralizes and simplifies the management of wedding and catering services, reducing manual effort and increasing operational efficiency. From the application, we provided a user-friendly interface for admin to add staff, manage wedding staff and catering staff. Moreover, wedding manager and catering manager can only access the data about wedding and catering managers for viewing purposes. The aim of this project is also to show the security measures that we have implemented such as masking, user privilege, authentication, role-based control and minor encryption. Furthermore, are to show the implementation of STRIDE and DREAD threat modelling on application and the ability to pass through this test. Ensuring that the system complies with relevant data protection regulations, such as PDPA 2010, safeguarding personal data and ensuring legal compliance. Finally, design the system to be scalable and easy to maintain, allowing for future enhancements and the ability to handle growing user demands.

## 2. PROPOSED HARDWARE AND SOFTWARE TO DEVELOP THE APPLICATION

### 2.1 PROGRAMMING LANGUAGE AND DATABASE PROGRAMME

**Programming Language: PHP**

PHP (Hypertext Preprocessor) is a widely used open-source server-side scripting language for web development. PHP is compatible with various databases, including MySQL. PHP code can be embedded into HTML. It is efficient for creating dynamic and interactive web pages.

**Database Program: MySQL**

The SQL part of "MySQL" stands for "Structured Query Language". SQL is the most common standardized language used to access databases. It is an ideal choice for handling the data requirements of a wedding and catering management system.

### 2.2 TYPE OF SERVER OS AND WEBSERVER

**Server OS: Ubuntu Server**

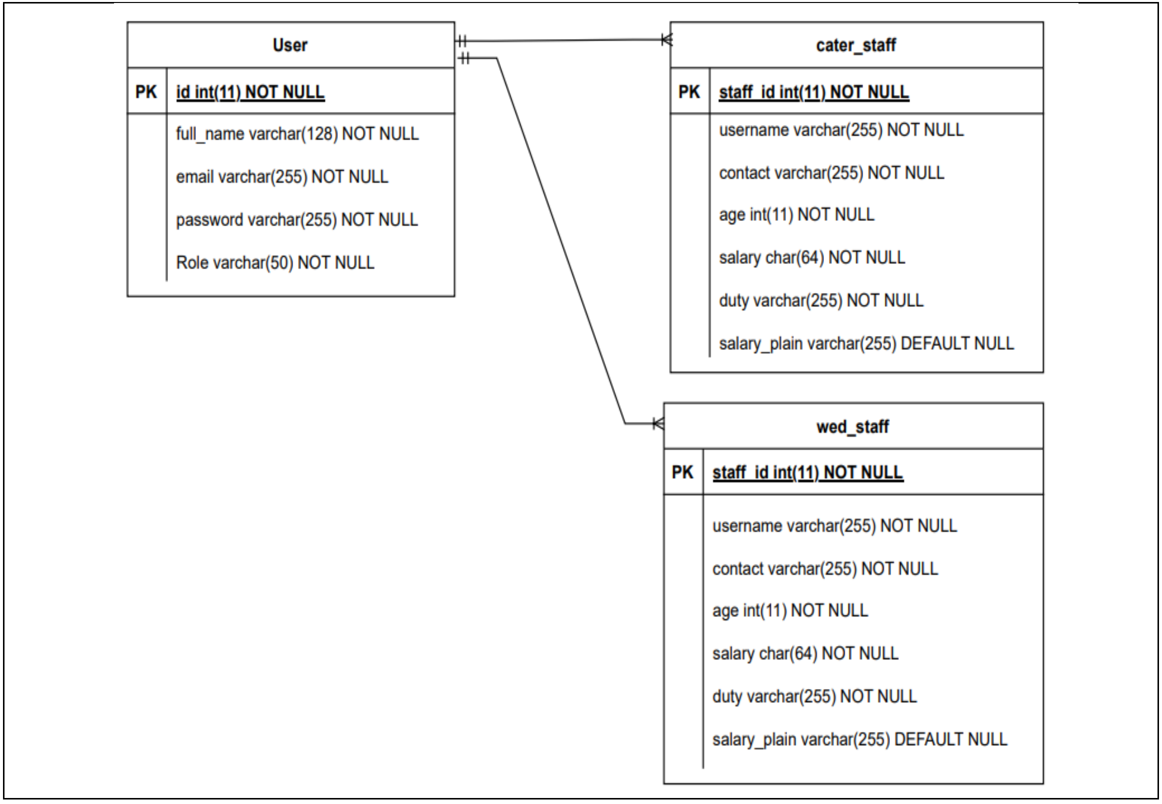
Ubuntu Server is a Linux-based operating system which provides a robust environment for hosting web applications and databases. Ubuntu Server is widely used in both development and production environments.

### Web Server: Apache HTTP Server (part of the XAMPP package)

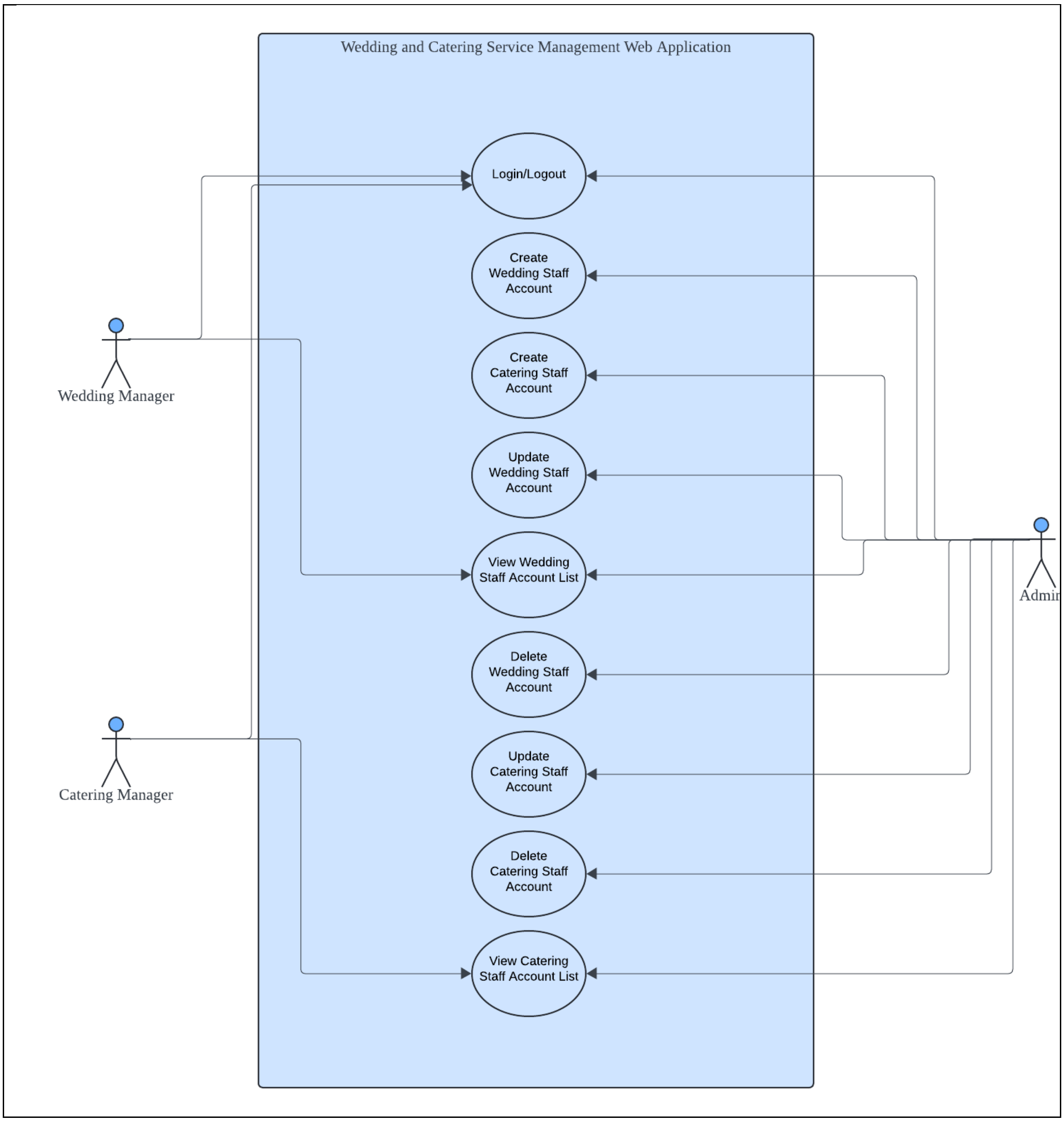
Apache is an open-source web server software that is widely used across various platforms, including Ubuntu Server. Apache is included in the XAMPP package which simplifies the setup process for development purposes.

## 3. SYSTEM DESIGN AND DATABASE DESIGN

### Database Design:


User
PK
id int(11) NOT NULL
full_name varchar(128) NOT NULL
email varchar(255) NOT NULL
password varchar(255) NOT NULL
Role varchar(50) NOT NULL
cater_staff
PK
staff_id int(11) NOT NULL
username varchar(255) NOT NULL
contact varchar(255) NOT NULL
age int(11) NOT NULL
salary char(64) NOT NULL
duty varchar(255) NOT NULL
salary_plain varchar(255) DEFAULT NULL
wed_staff
PK
staff_id int(11) NOT NULL
username varchar(255) NOT NULL
contact varchar(255) NOT NULL
age int(11) NOT NULL
salary char(64) NOT NULL
duty varchar(255) NOT NULL
salary_plain varchar(255) DEFAULT NULL

## System Design:


Wedding and Catering Service Management Web Application
Login/Logout
Create Wedding Staff Account
Create Catering Staff Account
Update Wedding Staff Account
View Wedding Staff Account List
Delete Wedding Staff Account
Update Catering Staff Account
Delete Catering Staff Account
View Catering Staff Account List
Wedding Manager
Catering Manager
Admir

## 4. PLAN TO SECURE THE DATABASE USING TRADITIONAL DATABASE SYSTEM

Securing a traditional SQL database system involves implementing multiple layers of defence to protect against various threats and vulnerabilities. Here is a comprehensive plan for securing database system for the wedding and catering staff services.

First, robust authentication and authorization mechanisms are essential. Strong password policies should be enforced, requiring complex passwords with a minimum length and regular expiration. Passwords should be securely stored using hashing algorithms such as bcrypt or SHA-256.

Dynamic data masking can further protect sensitive information by obfuscating it for non-privileged users. This involves defining masking rules for fields containing sensitive data, such as social security numbers or credit card information, to prevent unauthorized access to this data.

Regular security assessments, including vulnerability scanning and penetration testing, should be conducted to evaluate the effectiveness of security controls. Automated tools can identify and remediate security weaknesses, and findings from penetration tests should be addressed promptly to enhance security measures.

By implementing these comprehensive security measures, the risk of data breaches can be significantly reduced, ensuring the integrity, confidentiality, and availability of the database system.

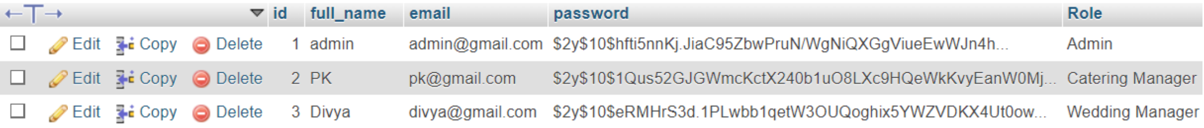
## 5. PROPOSED DESIGN AND IMPLEMENTATION OF THE APPLICATION USING SQL DATABASE

### 1. Design Description

The application is a web-based staff management system built with PHP and MySQL, designed for managing wedding and catering staff details. It includes functionality for adding, viewing, updating, and deleting staff records with role-based access control, ensuring that only admins can perform all actions while managers have restricted views. The frontend uses HTML, CSS, and JavaScript for the user interface, and the backend operates on an Apache server within a XAMPP environment, handling authentication and database interactions.

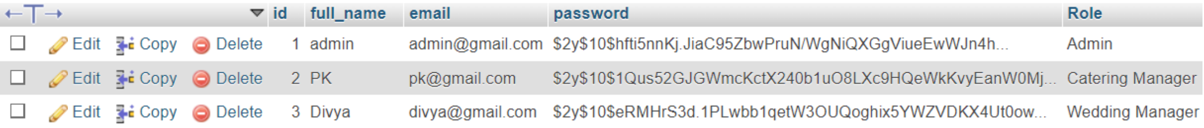
## 2. Step-by-Step Application Creation

a. Firstly, we created users and roles in our database for logins on the web based on their respective roles.

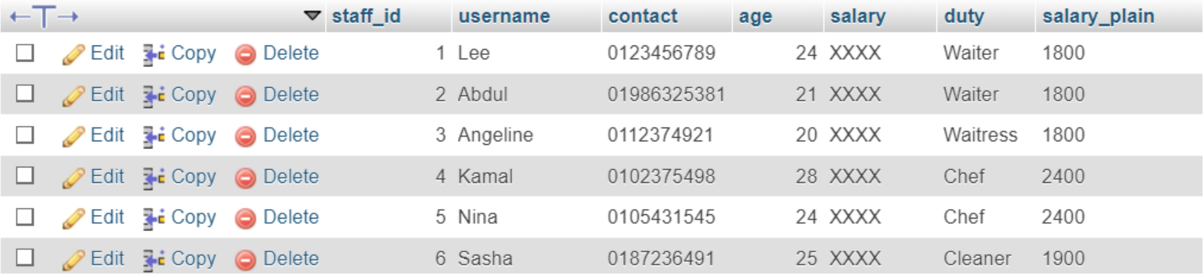


| ←T→ |  | id | full_name | email | password | Role |
|---|---|---|---|---|---|---|
| □ | Edit Copy Delete | 1 | admin | admin@gmail.com | $2y$10$hfti5nnKj.JiaC95ZbwPruN/WgNiQXGgViueEwWJn4h... | Admin |
| □ | Edit Copy Delete | 2 | PK | pk@gmail.com | $2y$10$1Qus52GJGWmcKctX240b1uO8LXc9HQeWkKvyEanW0Mj... | Catering Manager |
| □ | Edit Copy Delete | 3 | Divya | divya@gmail.com | $2y$10$eRMHrS3d.1PLwbb1qetW3OUQoghix5YWZVDKX4Ut0ow... | Wedding Manager |

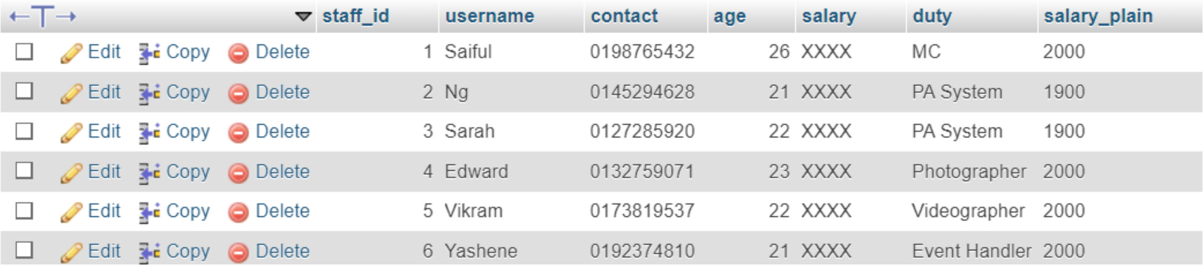
b. Next, we created 3 tables within the database called users, wed_staff and cater_staff that will be used for the application.



| ←T→ |  | id | full_name | email | password | Role |
|---|---|---|---|---|---|---|
| □ | Edit Copy Delete | 1 | admin | admin@gmail.com | $2y$10$hfti5nnKj.JiaC95ZbwPruN/WgNiQXGgViueEwWJn4h... | Admin |
| □ | Edit Copy Delete | 2 | PK | pk@gmail.com | $2y$10$1Qus52GJGWmcKctX240b1uO8LXc9HQeWkKvyEanW0Mj... | Catering Manager |
| □ | Edit Copy Delete | 3 | Divya | divya@gmail.com | $2y$10$eRMHrS3d.1PLwbb1qetW3OUQoghix5YWZVDKX4Ut0ow... | Wedding Manager |

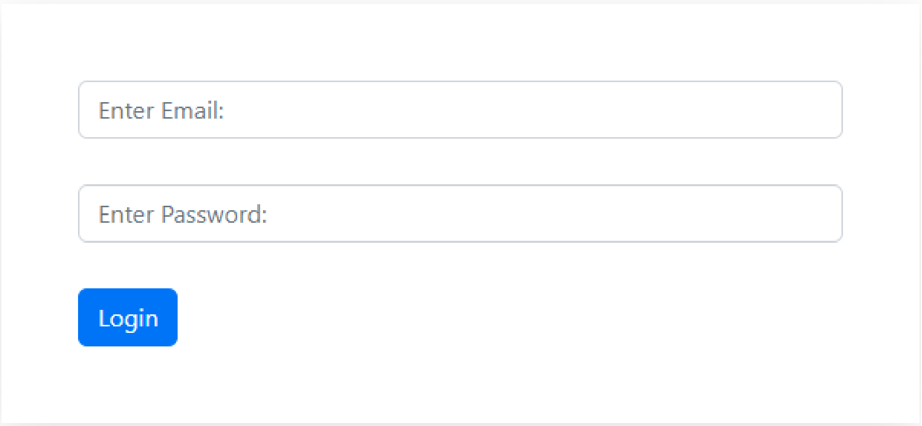


| ←T→ |  | staff_id | username | contact | age | salary | duty | salary_plain |
|---|---|---|---|---|---|---|---|---|
| □ | Edit Copy Delete | 1 | Lee | 0123456789 | 24 | XXXX | Waiter | 1800 |
| □ | Edit Copy Delete | 2 | Abdul | 01986325381 | 21 | XXXX | Waiter | 1800 |
| □ | Edit Copy Delete | 3 | Angeline | 0112374921 | 20 | XXXX | Waitress | 1800 |
| □ | Edit Copy Delete | 4 | Kamal | 0102375498 | 28 | XXXX | Chef | 2400 |
| □ | Edit Copy Delete | 5 | Nina | 0105431545 | 24 | XXXX | Chef | 2400 |
| □ | Edit Copy Delete | 6 | Sasha | 0187236491 | 25 | XXXX | Cleaner | 1900 |

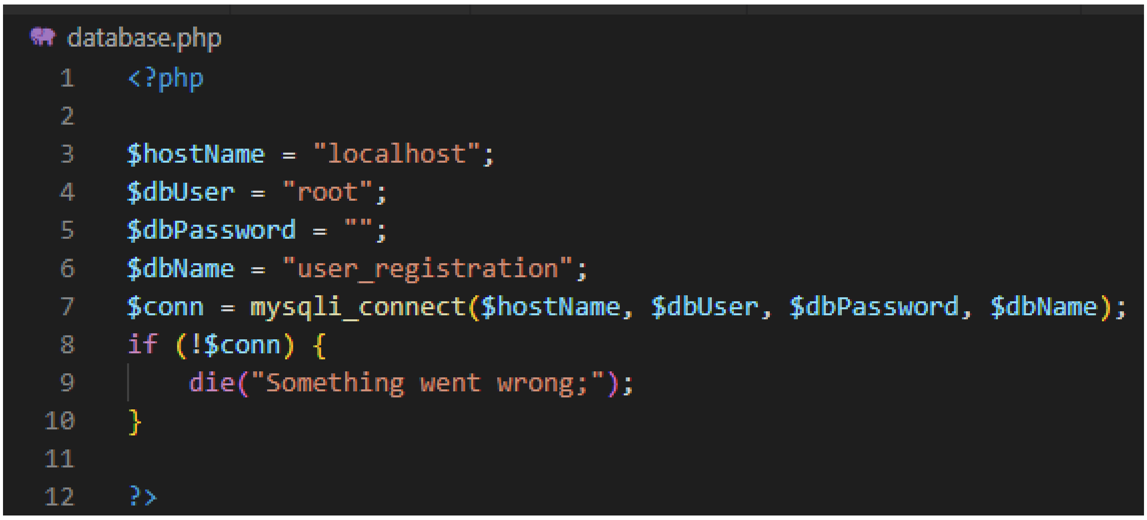


| ←T→ |  | staff_id | username | contact | age | salary | duty | salary_plain |
|---|---|---|---|---|---|---|---|---|
| □ | Edit Copy Delete | 1 | Saiful | 0198765432 | 26 | XXXX | MC | 2000 |
| □ | Edit Copy Delete | 2 | Ng | 0145294628 | 21 | XXXX | PA System | 1900 |
| □ | Edit Copy Delete | 3 | Sarah | 0127285920 | 22 | XXXX | PA System | 1900 |
| □ | Edit Copy Delete | 4 | Edward | 0132759071 | 23 | XXXX | Photographer | 2000 |
| □ | Edit Copy Delete | 5 | Vikram | 0173819537 | 22 | XXXX | Videographer | 2000 |
| □ | Edit Copy Delete | 6 | Yashene | 0192374810 | 21 | XXXX | Event Handler | 2000 |

c. After that, we created a login page using PHP with the implementation of HTML and CSS for a better UI for the users. By filling in the details, the system will check through the credentials from the users table in the database.

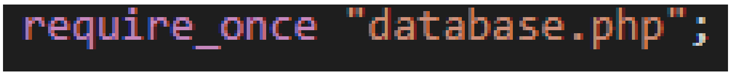


d. We have implemented a connection from the web application to the database by using PHP through database.php. This enables the function of insert, delete, update and view that are linked between the web application and the database.



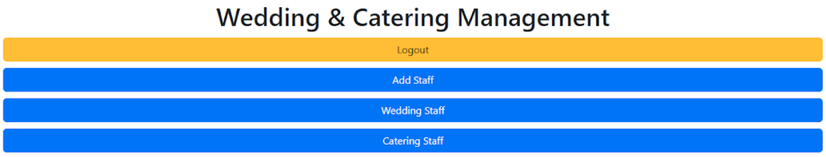
database.php

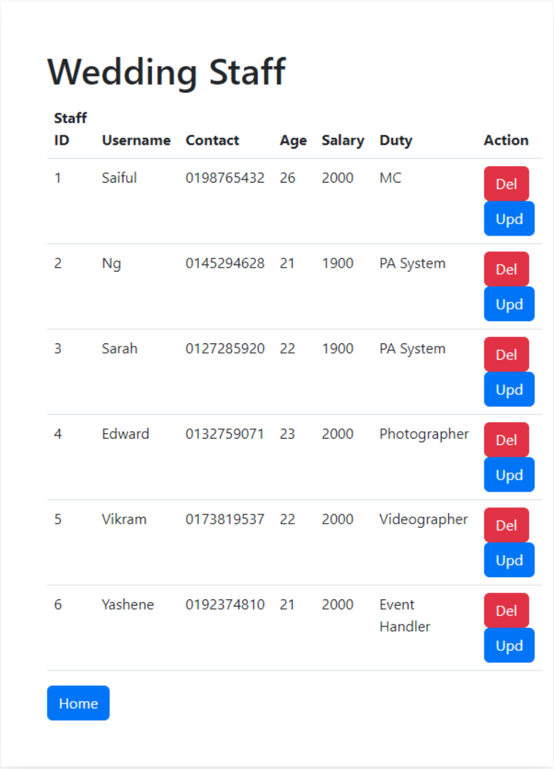
```php
<?php

$hostName = "localhost";
$dbUser = "root";
$dbPassword = "";
$dbName = "user_registration";
$conn = mysqli_connect($hostName, $dbUser, $dbPassword, $dbName);
if (!$conn) {
    die("Something went wrong;");
}

?>
```



```php
require_once "database.php";
```

e. With the ability to log in to the web application, we have created web pages that are implemented to have role-based authorization which can make it easier for the users to access the web based on their respective role. For example, the Wedding Manager is only able to view the Wedding Staff page and not the user which is Catering Manager. Only the Admin can view all pages with some additional features.

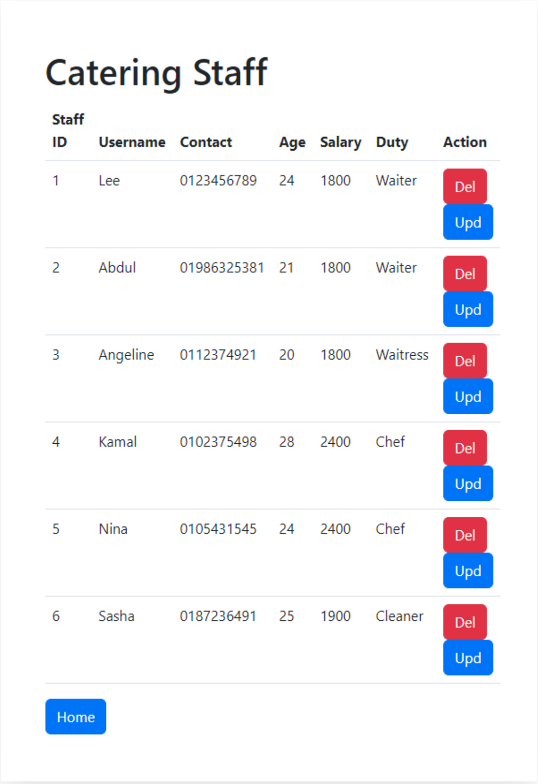




## Wedding Staff

| Staff ID | Username | Contact | Age | Salary | Duty | Action |
|---|---|---|---|---|---|---|
| 1 | Saiful | 0198765432 | 26 | 2000 | MC | Del<br>Upd |
| 2 | Ng | 0145294628 | 21 | 1900 | PA System | Del<br>Upd |
| 3 | Sarah | 0127285920 | 22 | 1900 | PA System | Del<br>Upd |
| 4 | Edward | 0132759071 | 23 | 2000 | Photographer | Del<br>Upd |
| 5 | Vikram | 0173819537 | 22 | 2000 | Videographer | Del<br>Upd |
| 6 | Yashene | 0192374810 | 21 | 2000 | Event Handler | Del<br>Upd |

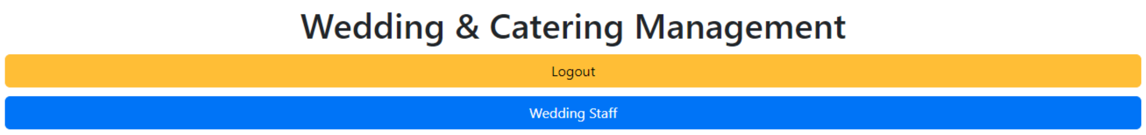
Home

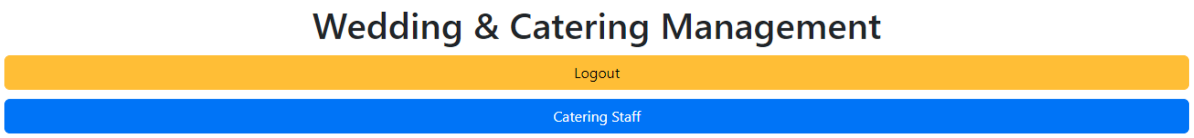


## Catering Staff

| Staff ID | Username | Contact | Age | Salary | Duty | Action |
|---|---|---|---|---|---|---|
| 1 | Lee | 0123456789 | 24 | 1800 | Waiter | Del<br>Upd |
| 2 | Abdul | 01986325381 | 21 | 1800 | Waiter | Del<br>Upd |
| 3 | Angeline | 0112374921 | 20 | 1800 | Waitress | Del<br>Upd |
| 4 | Kamal | 0102375498 | 28 | 2400 | Chef | Del<br>Upd |
| 5 | Nina | 0105431545 | 24 | 2400 | Chef | Del<br>Upd |
| 6 | Sasha | 0187236491 | 25 | 1900 | Cleaner | Del<br>Upd |

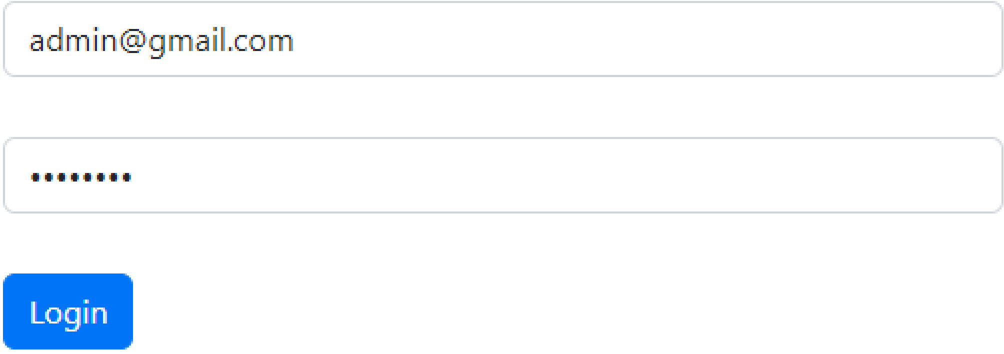
Home


Wedding & Catering Management


Wedding & Catering Management

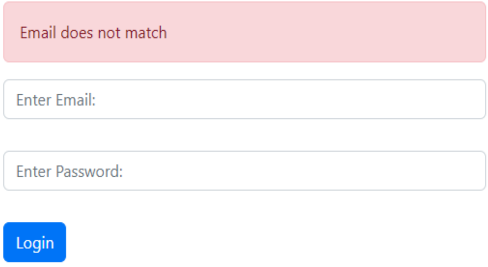
## 3. Details of Security Measures Implemented

a. All the users are implemented through SQL where the users created are within the database. If the user inserted wrong credentials, the system would notify.

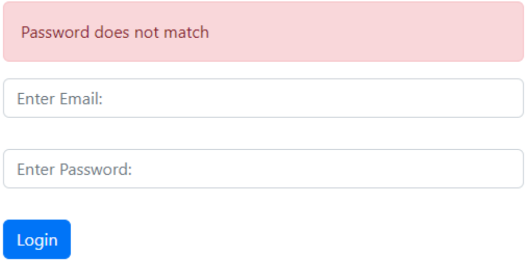


admin@gmail.com

••••••••

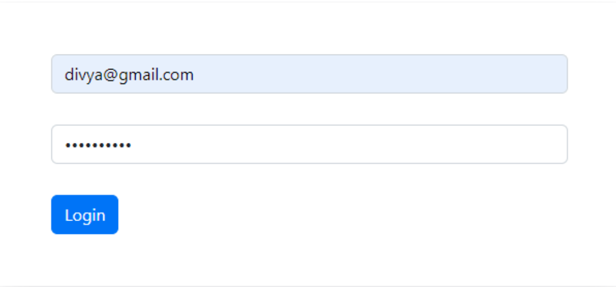


Email does not match

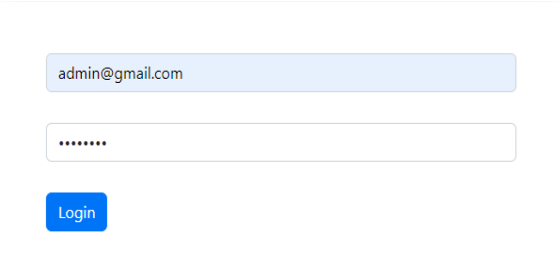


Password does not match

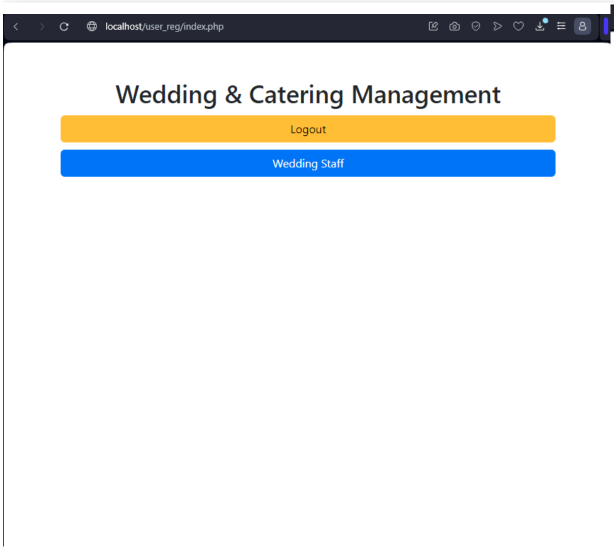
b. The user has their own authorization based on the roles that they are assigned to. For example, the Wedding Manager can only view the Wedding Staff list but not insert new staff. However, the admin can. If the Wedding Manager enters the add_staff.php, it will directly lead the user back to the index.php (Home page)


divya@gmail.com

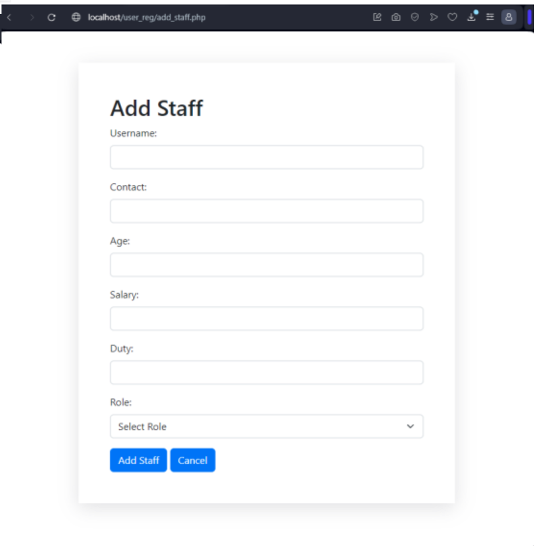
••••••••••


admin@gmail.com

••••••••


localhost/user_reg/index.php

Wedding & Catering Management


localhost/user_reg/add_staff.php

Add Staff

Username:

Contact:

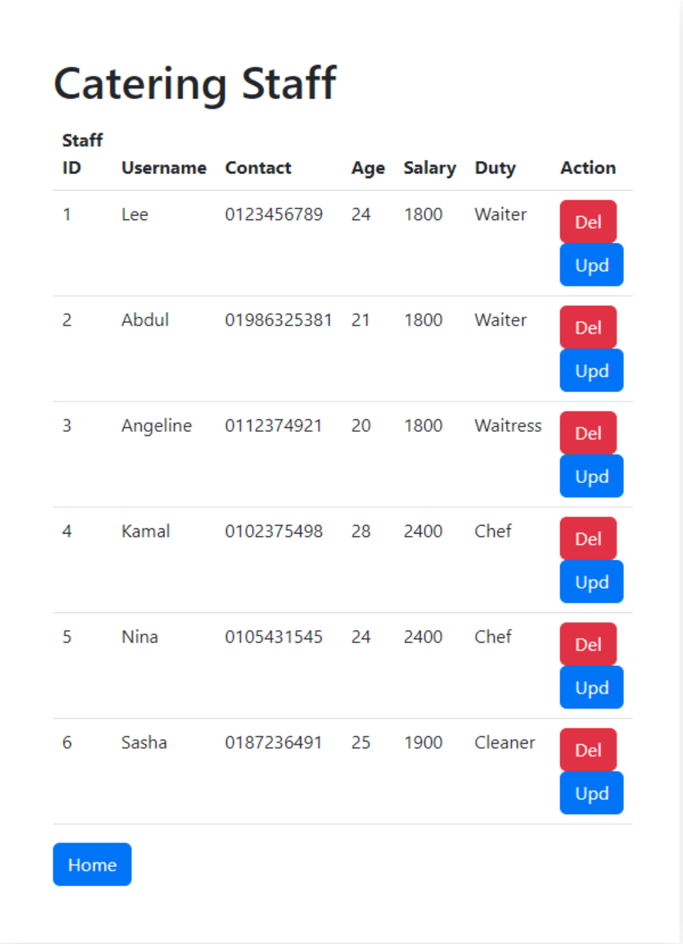
Age:

Salary:

Duty:

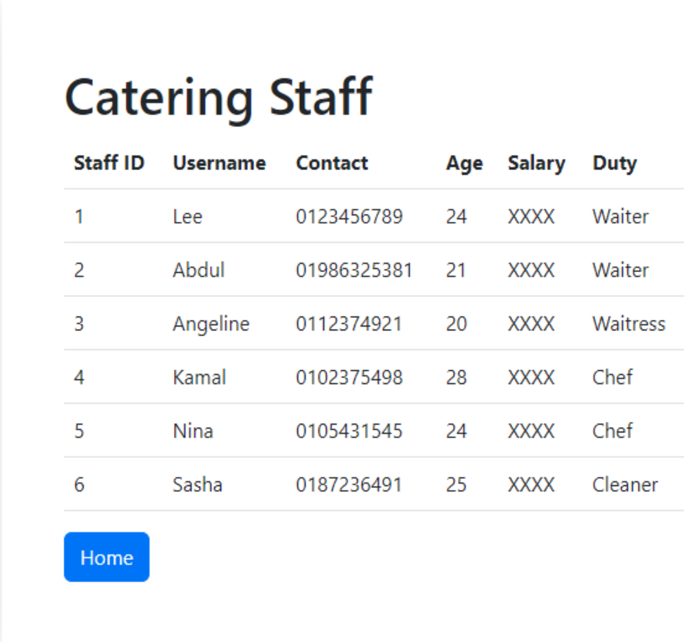
Role: Select Role

c. Only the Admin is given the authority to delete or update data from the database. Unauthorised users like the Catering Manager and the Wedding Manager are not able to access this feature.



## Catering Staff

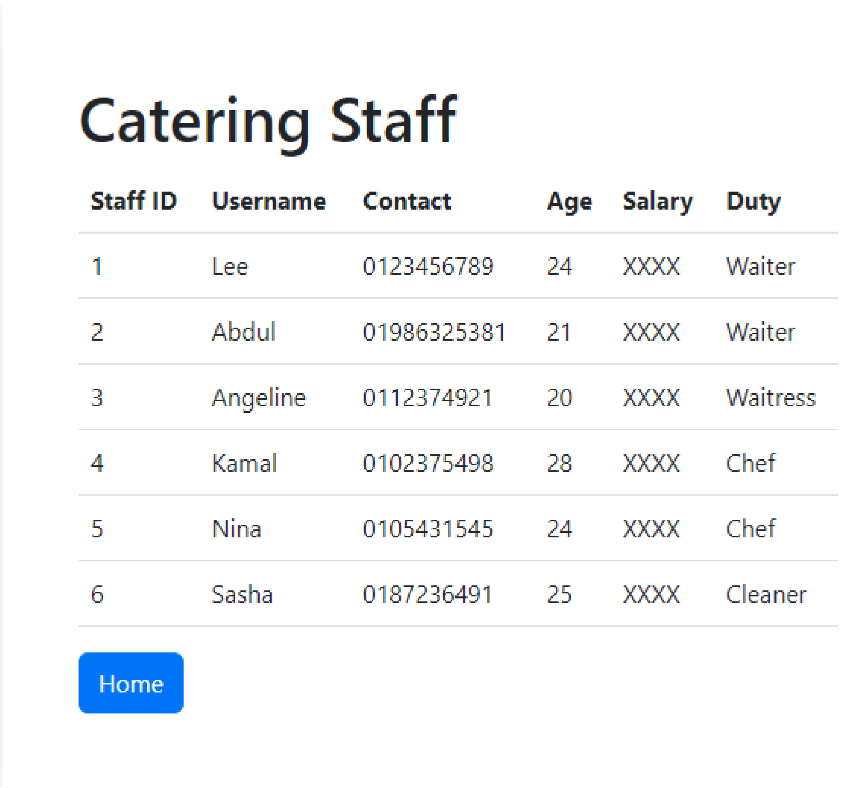
| Staff ID | Username | Contact | Age | Salary | Duty | Action |
|---|---|---|---|---|---|---|
| 1 | Lee | 0123456789 | 24 | 1800 | Waiter | Del<br>Upd |
| 2 | Abdul | 01986325381 | 21 | 1800 | Waiter | Del<br>Upd |
| 3 | Angeline | 0112374921 | 20 | 1800 | Waitress | Del<br>Upd |
| 4 | Kamal | 0102375498 | 28 | 2400 | Chef | Del<br>Upd |
| 5 | Nina | 0105431545 | 24 | 2400 | Chef | Del<br>Upd |
| 6 | Sasha | 0187236491 | 25 | 1900 | Cleaner | Del<br>Upd |



## Catering Staff

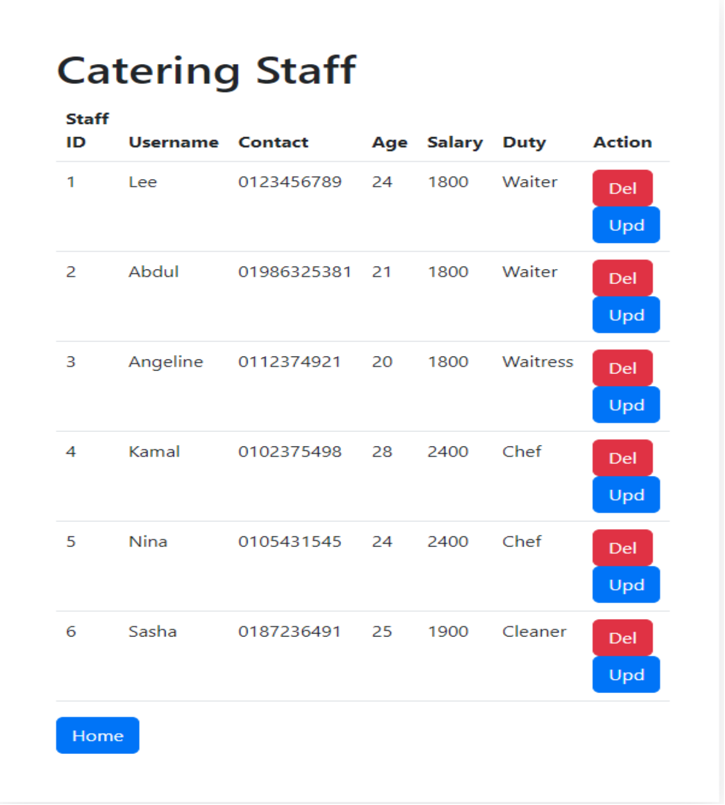
| Staff ID | Username | Contact | Age | Salary | Duty |
|---|---|---|---|---|---|
| 1 | Lee | 0123456789 | 24 | XXXX | Waiter |
| 2 | Abdul | 01986325381 | 21 | XXXX | Waiter |
| 3 | Angeline | 0112374921 | 20 | XXXX | Waitress |
| 4 | Kamal | 0102375498 | 28 | XXXX | Chef |
| 5 | Nina | 0105431545 | 24 | XXXX | Chef |
| 6 | Sasha | 0187236491 | 25 | XXXX | Cleaner |

d. The staff’s salary is considered as sensitive data. Therefore, we implement a dynamic masking to secure the data to be viewed by unauthorised users. Only the Admin can view the unmasked data.



## Catering Staff

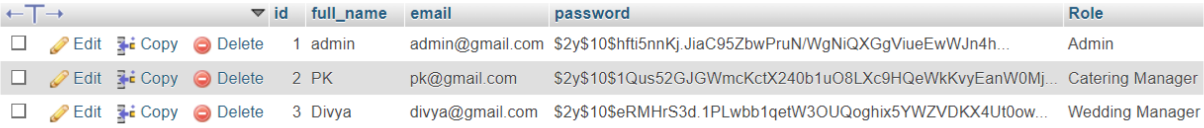
| Staff ID | Username | Contact | Age | Salary | Duty |
|---|---|---|---|---|---|
| 1 | Lee | 0123456789 | 24 | XXXX | Waiter |
| 2 | Abdul | 01986325381 | 21 | XXXX | Waiter |
| 3 | Angeline | 0112374921 | 20 | XXXX | Waitress |
| 4 | Kamal | 0102375498 | 28 | XXXX | Chef |
| 5 | Nina | 0105431545 | 24 | XXXX | Chef |
| 6 | Sasha | 0187236491 | 25 | XXXX | Cleaner |



## Catering Staff

| Staff ID | Username | Contact | Age | Salary | Duty | Action |
|---|---|---|---|---|---|---|
| 1 | Lee | 0123456789 | 24 | 1800 | Waiter | Del<br>Upd |
| 2 | Abdul | 01986325381 | 21 | 1800 | Waiter | Del<br>Upd |
| 3 | Angeline | 0112374921 | 20 | 1800 | Waitress | Del<br>Upd |
| 4 | Kamal | 0102375498 | 28 | 2400 | Chef | Del<br>Upd |
| 5 | Nina | 0105431545 | 24 | 2400 | Chef | Del<br>Upd |
| 6 | Sasha | 0187236491 | 25 | 1900 | Cleaner | Del<br>Upd |

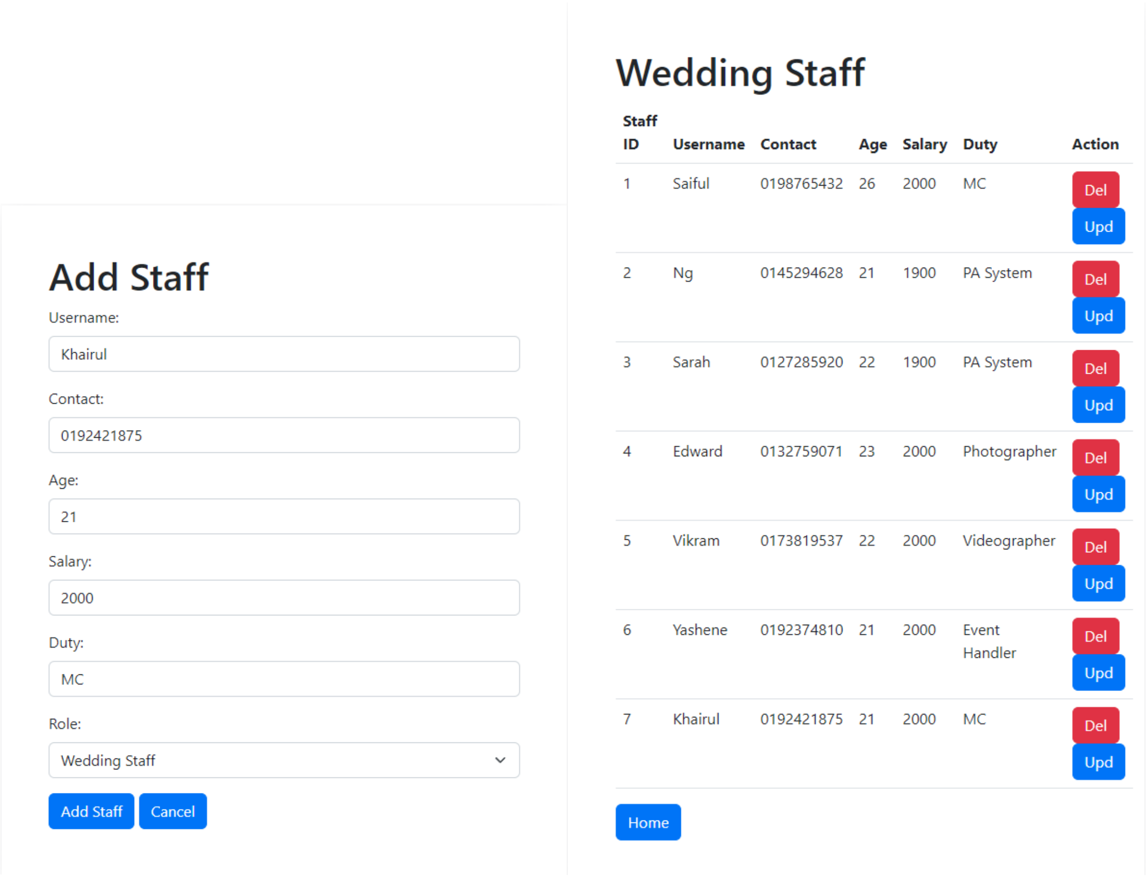
e. Important data such as the users’ passwords are encrypted by using hash to avoid any disclosure of information.



| | id | full_name | email | password | Role |
|---|---|---|---|---|---|
| □ Edit Copy Delete | 1 | admin | admin@gmail.com | $2y$10$hfti5nnKj.JiaC95ZbwPruN/WgNiQXGgViueEwWJn4h... | Admin |
| □ Edit Copy Delete | 2 | PK | pk@gmail.com | $2y$10$1Qus52GJGWmcKctX240b1uO8LXc9HQeWkKvyEanW0Mj... | Catering Manager |
| □ Edit Copy Delete | 3 | Divya | divya@gmail.com | $2y$10$eRMHrS3d.1PLwbb1qetW3OUQoghix5YWZVDKX4Ut0ow... | Wedding Manager |

## 4. Testing Activity

a. Insert new entry (Khairul - wed_staff)

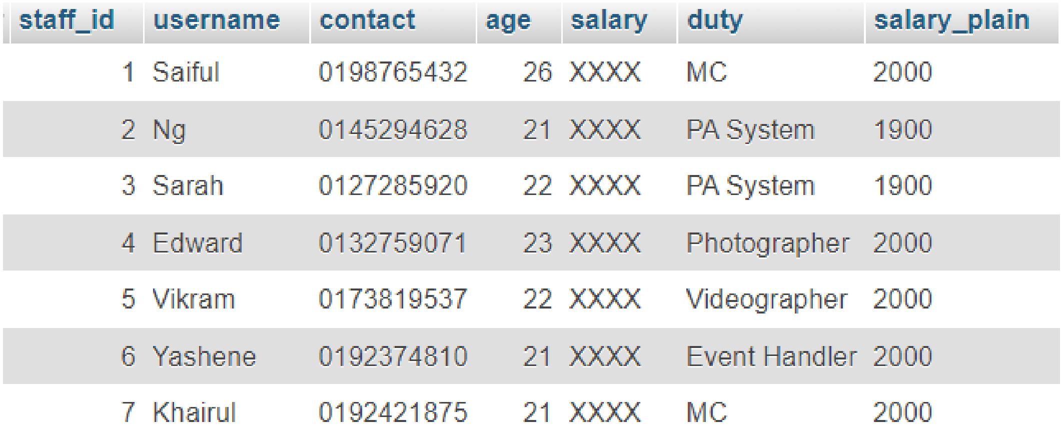


### Add Staff

| | |
|---|---|
| Username: | Khairul |
| Contact: | 0192421875 |
| Age: | 21 |
| Salary: | 2000 |
| Duty: | MC |
| Role: | Wedding Staff |

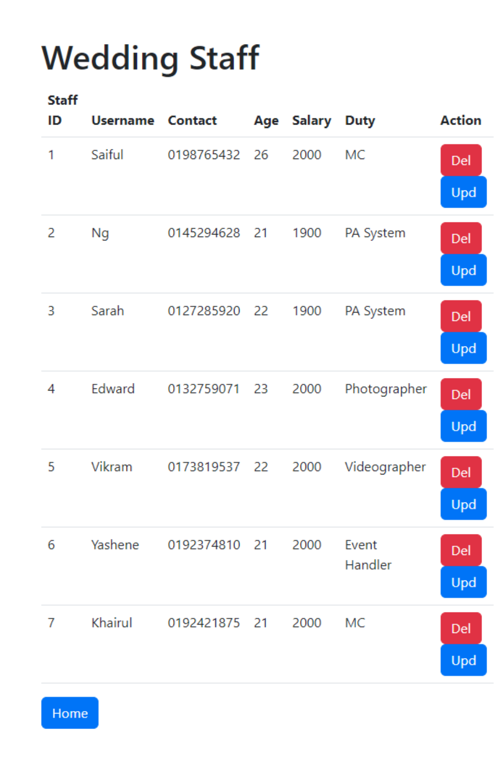
### Wedding Staff

| Staff ID | Username | Contact | Age | Salary | Duty | Action |
|---|---|---|---|---|---|---|
| 1 | Saiful | 0198765432 | 26 | 2000 | MC | Del<br>Upd |
| 2 | Ng | 0145294628 | 21 | 1900 | PA System | Del<br>Upd |
| 3 | Sarah | 0127285920 | 22 | 1900 | PA System | Del<br>Upd |
| 4 | Edward | 0132759071 | 23 | 2000 | Photographer | Del<br>Upd |
| 5 | Vikram | 0173819537 | 22 | 2000 | Videographer | Del<br>Upd |
| 6 | Yashene | 0192374810 | 21 | 2000 | Event Handler | Del<br>Upd |
| 7 | Khairul | 0192421875 | 21 | 2000 | MC | Del<br>Upd |



| staff_id | username | contact | age | salary | duty | salary_plain |
|---|---|---|---|---|---|---|
| 1 | Saiful | 0198765432 | 26 | XXXX | MC | 2000 |
| 2 | Ng | 0145294628 | 21 | XXXX | PA System | 1900 |
| 3 | Sarah | 0127285920 | 22 | XXXX | PA System | 1900 |
| 4 | Edward | 0132759071 | 23 | XXXX | Photographer | 2000 |
| 5 | Vikram | 0173819537 | 22 | XXXX | Videographer | 2000 |
| 6 | Yashene | 0192374810 | 21 | XXXX | Event Handler | 2000 |
| 7 | Khairul | 0192421875 | 21 | XXXX | MC | 2000 |

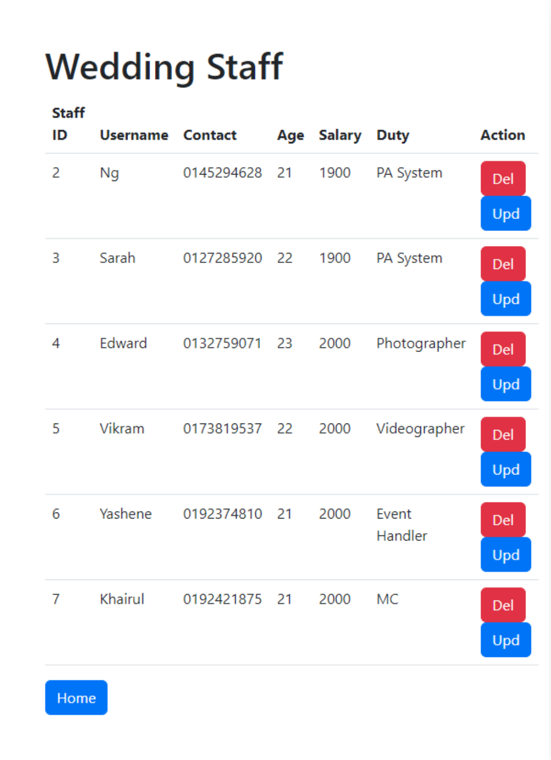
b. Delete one of the old entries (Saiful - wed_staff)



## Wedding Staff

| Staff ID | Username | Contact | Age | Salary | Duty | Action |
|---|---|---|---|---|---|---|
| 1 | Saiful | 0198765432 | 26 | 2000 | MC | Del<br>Upd |
| 2 | Ng | 0145294628 | 21 | 1900 | PA System | Del<br>Upd |
| 3 | Sarah | 0127285920 | 22 | 1900 | PA System | Del<br>Upd |
| 4 | Edward | 0132759071 | 23 | 2000 | Photographer | Del<br>Upd |
| 5 | Vikram | 0173819537 | 22 | 2000 | Videographer | Del<br>Upd |
| 6 | Yashene | 0192374810 | 21 | 2000 | Event Handler | Del<br>Upd |
| 7 | Khairul | 0192421875 | 21 | 2000 | MC | Del<br>Upd |

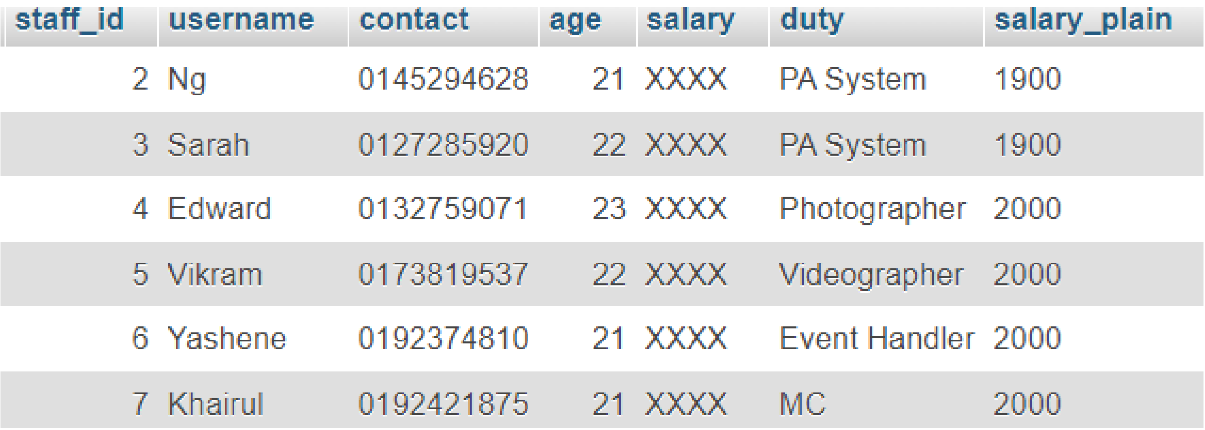
Home



## Wedding Staff

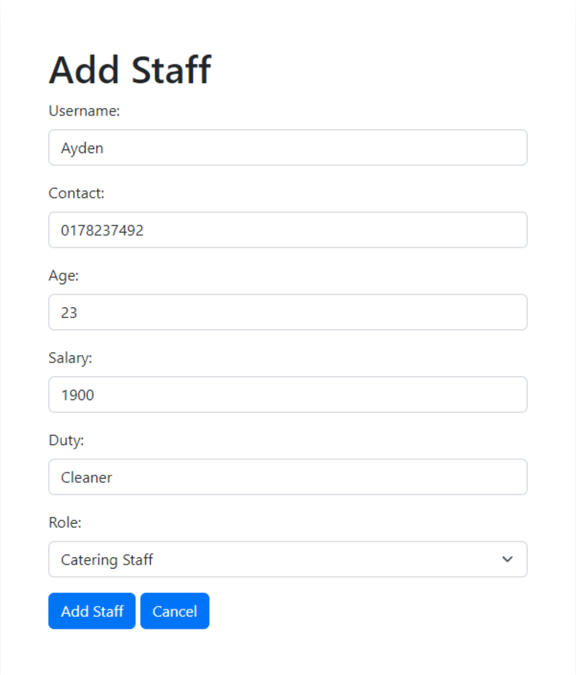
| Staff ID | Username | Contact | Age | Salary | Duty | Action |
|---|---|---|---|---|---|---|
| 2 | Ng | 0145294628 | 21 | 1900 | PA System | Del<br>Upd |
| 3 | Sarah | 0127285920 | 22 | 1900 | PA System | Del<br>Upd |
| 4 | Edward | 0132759071 | 23 | 2000 | Photographer | Del<br>Upd |
| 5 | Vikram | 0173819537 | 22 | 2000 | Videographer | Del<br>Upd |
| 6 | Yashene | 0192374810 | 21 | 2000 | Event Handler | Del<br>Upd |
| 7 | Khairul | 0192421875 | 21 | 2000 | MC | Del<br>Upd |

Home



| staff_id | username | contact | age | salary | duty | salary_plain |
|---|---|---|---|---|---|---|
| 2 | Ng | 0145294628 | 21 | XXXX | PA System | 1900 |
| 3 | Sarah | 0127285920 | 22 | XXXX | PA System | 1900 |
| 4 | Edward | 0132759071 | 23 | XXXX | Photographer | 2000 |
| 5 | Vikram | 0173819537 | 22 | XXXX | Videographer | 2000 |
| 6 | Yashene | 0192374810 | 21 | XXXX | Event Handler | 2000 |
| 7 | Khairul | 0192421875 | 21 | XXXX | MC | 2000 |

c. Insert another new entry (Ayden - cater_staff)



## Add Staff

Username: Ayden

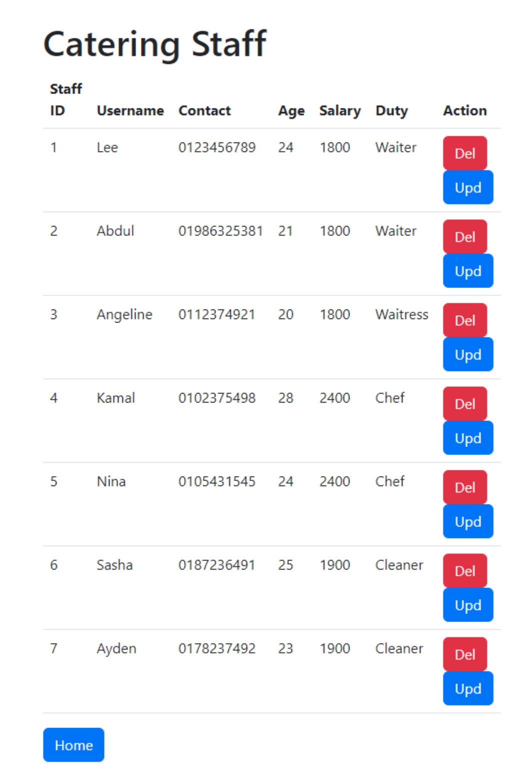
Contact: 0178237492

Age: 23

Salary: 1900

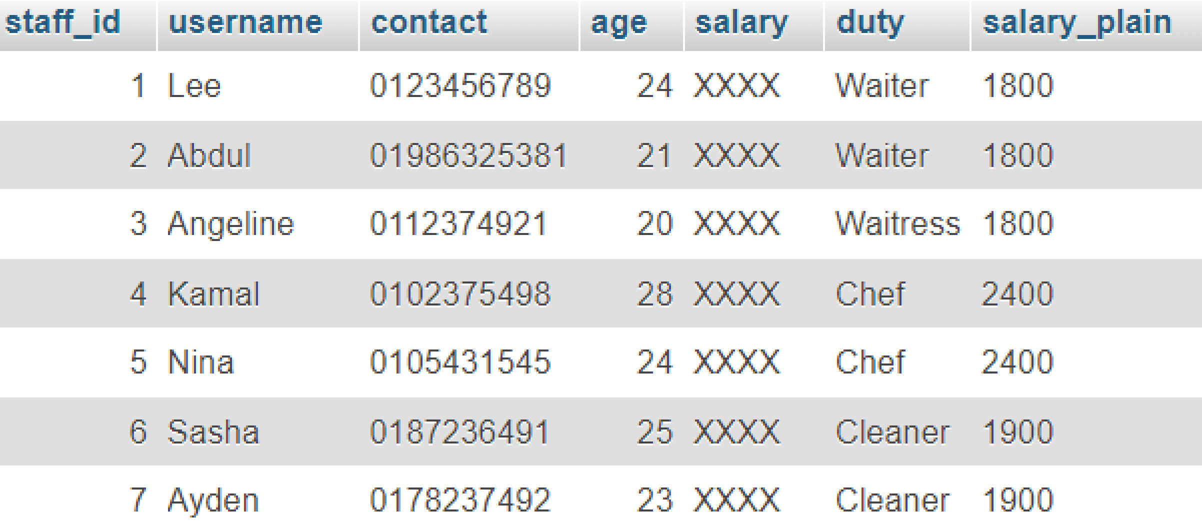
Duty: Cleaner

Role: Catering Staff



## Catering Staff

| Staff ID | Username | Contact | Age | Salary | Duty | Action |
|---|---|---|---|---|---|---|
| 1 | Lee | 0123456789 | 24 | 1800 | Waiter | Del<br>Upd |
| 2 | Abdul | 01986325381 | 21 | 1800 | Waiter | Del<br>Upd |
| 3 | Angeline | 0112374921 | 20 | 1800 | Waitress | Del<br>Upd |
| 4 | Kamal | 0102375498 | 28 | 2400 | Chef | Del<br>Upd |
| 5 | Nina | 0105431545 | 24 | 2400 | Chef | Del<br>Upd |
| 6 | Sasha | 0187236491 | 25 | 1900 | Cleaner | Del<br>Upd |
| 7 | Ayden | 0178237492 | 23 | 1900 | Cleaner | Del<br>Upd |



| staff_id | username | contact | age | salary | duty | salary_plain |
|---|---|---|---|---|---|---|
| 1 | Lee | 0123456789 | 24 | XXXX | Waiter | 1800 |
| 2 | Abdul | 01986325381 | 21 | XXXX | Waiter | 1800 |
| 3 | Angeline | 0112374921 | 20 | XXXX | Waitress | 1800 |
| 4 | Kamal | 0102375498 | 28 | XXXX | Chef | 2400 |
| 5 | Nina | 0105431545 | 24 | XXXX | Chef | 2400 |
| 6 | Sasha | 0187236491 | 25 | XXXX | Cleaner | 1900 |
| 7 | Ayden | 0178237492 | 23 | XXXX | Cleaner | 1900 |

## 6. THREAT MODELING

### STRIDE:

| RISK | STRIDE | JUSTIFICATION |
|---|---|---|
| **SQL Injection** | **S, T, I** | **Spoofing:** Attackers use malicious SQL queries to spoof identity.<br>**Tampering:** Attackers can manipulate the database with crafted queries.<br>**Information Disclosure:** Sensitive data can be extracted by attackers. |
| **DoS Attack** | **D** | **Denial of Service:** Attackers can overwhelm the server with excessive requests, leading to service downtime**.** |
| **Privilege Escalation** | **E** | **Elevation of Privilege:** Attackers can exploit vulnerabilities to gain higher-level access than intended. |
| **Unauthorized Data Access** | **I, R** | **Information Disclosure:** Sensitive data can be accessed without proper authorization.<br>**Repudiation**: Actions by unauthorized users can be hard to trace back and deny their activities. |
| **Data Tampering** | **T** | **Tampering:** Attackers can alter data in the database, compromising data integrity. |
| **Session Hijacking** | **S, E** | **Spoofing:** Attackers can hijack legitimate user sessions.<br>**Elevation of Privilege:** Attackers can gain unauthorized access by hijacking sessions. |

### DREAD:

| RISK | D | R | E | A | D | Threat Rating | Justification |
|---|---|---|---|---|---|---|---|
| **SQL Injection** | **8** | **7** | **9** | **8** | **8** | **8** | Highly damaging as it can compromise the entire database, easy to exploit, and affects all users. |
| **DoS Attack** | **5** | **3** | **4** | **6** | **6** | **5** | Can significantly disrupt service availability, though mitigation measures like rate limiting can reduce impact. |
| **Privilege Escalation** | **7** | **6** | **8** | **7** | **7** | **7** | Serve impact due to unauthorized access and control over the system, mitigated by stringent access controls and regular audits. |
| **Unauthorized Data Access** | **6** | **5** | **6** | **6** | **6** | **6** | Affects confidentiality and integrity of sensitive data, moderate ease of discovery and mitigation with proper access control. |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Data Tampering** | **7** | **6** | **6** | **7** | **7** | **7** | Compromises data integrity, affecting business operations and decision-making, mitigated by checks and logging. |
| **Session Hijacking** | **6** | **5** | **7** | **6** | **6** | **6** | Potential for unauthorized actions using hijacking sessions, mitigated by secure session management and encryption. |

## 7. PDPA 2010

### Categorization of Personnel

According to PDPA 2010, personnel can be categorized as follows:

**Data Users:** Individuals who process personal data on behalf of data users.

**Data Processors:** Individuals or entities who process personal data on behalf of data users.

**Data Controllers:** Individuals who determine the purpose and means of processing personal data

### Mapping Data Lifecycle to PDPA 2010 Requirements

**Data Collection:** Before collecting an individual's personal data, obtain their consent except in specific situations outlined in the law. This can be achieved by using checkboxes or digital signatures on forms to obtain explicit consent from individuals. Example, wedding managers that ensures customers provide consent during the booking process.

**Data Processing:** Personal data must be processed fairly, and lawfully, and only for the purposes for which it was collected. Ensuring the personal data is used only for the purpose stated during collection. For example, admins that monitor and enforce data processing policies

**Data Storage:** Personal data must be stored securely and protected against disclosure, destruction and unauthorized access. Compliance action would be encrypt sensitive data both at rest and in transit. This can be done by database administrators who oversee encryption and access control mechanisms.

**Data Transfer:** Personal data must be transferred securely, especially when transferred outside of Malaysia. Ensure compliance with PDPA 2010 requirements for cross-border data transfers. For instance, network administrators monitor data transfers.

**Data Retention and Disposal:** Personal data must not be kept longer than necessary for the fulfilment of the purposes for which it was collected. The compliance action would be ensuring data is securely deleted or destroyed once it is no longer needed. For example, admins enforcing data retention and disposal policies.

### Penalties for Non-Compliance

PDPA 2010 outlines various penalties for non-compliance, including:

**Administrative Penalties:** Imprisonment of up to three years, fines up to RM500,000 or both. For certain offenses.

**Compensation to Data Subject:** Individuals may be entitled to compensation for those affected by a breach of PDPA 2010 for damages suffered.

**Liability of Directors and Managers:** Directors, managers, and officers can be held personally liable for corporate offenses if committed with their consent or due to their neglect.

**Civil Remedies:** Individuals affected by a breach of PDPA 2010 may seek civil remedies, through civil proceedings.

## 8. SECURITY MEASURES IMPLEMENTATION

### Data Masking

Data masking involves obscuring data within a database to protect it from unauthorized access. Example, masking the salary of wedding staffs by displaying only X letter (eg., XXXX).

### User Privileges

User privileges control what actions a user can perform on the database. For instance, only the admin has privileges to insert, update and delete any data of the staffs in the database.

### Authentication

Authentication verifies the identity of users attempting to access the system. Example, a user has to enter their respective email address and password to access the database. Only authorized user can access the web application and database.

### Role-Based Access Control

It restricts system access to authorized users based on their roles within the organization. For instance, admin can manage user accounts and system settings.

### Minor Encryption

Encryption involves encoding data to protect it from unauthorized access. For example, encrypting sensitive data stored in the database such as the usage of hash for user’s password.