

**FINAL REPORT**  
**WEB APPLICATION DEVELOPMENT**



**Member of Group:**

Fatwa Putri Jingga	[ 012202400051 ]
Marsha Aulia Rizky	[ 012202400013 ]
Putri Armalia	[ 012202400017 ]

## 1. Admin\_dashboard.php

The screenshot shows the Admin Dashboard interface. At the top, there are four summary boxes: 'Total Queue' (4), 'Tables Ready' (10), 'Tables in Use' (0), and 'Reservations Today' (0). Below these are two tables: 'Queue List' and 'Table Status'. The 'Queue List' table shows 10 entries with columns for Queue Number, User, Party Size, Status, and Action. The 'Table Status' table shows 10 tables with columns for Table Number, Capacity, Status, and Action. A message at the bottom indicates a new table has been added successfully.

The `admin_dashboard.php` file is the **main dashboard page for admin users** in a restaurant queue and reservation management web application. It provides a comprehensive interface for managing queues, tables, and reservations. Below is a breakdown of its functionality:

### Core Functionalities:

#### 1. Access Control

- The file **starts a session** and includes a configuration file (`config.php`).
- It **restricts access** to users who are not logged in or are not admins by redirecting them to `login.php`.

#### 2. Summary Statistics

- Retrieves and displays:

**Total Queue:** Number of people currently waiting.

**Tables Ready:** Number of available tables.

**Tables in Use:** Number of tables currently occupied.

**Reservations Today:** Total number of reservations for the current day.

#### 3. Queue Management

Displays a **table of queue entries** showing queue number, user, party size, status (`waiting`, `called`, or `seated`), and possible actions:

```
1 <?php
2 session_start();
3 include 'config.php';
4
5 if (!isset($_SESSION['user_id']) || $_SESSION['role'] != 'admin') {
6     header("location: login.php");
7     exit;
8 }
9
10 $totalQueue = mysqli_num_rows(mysqli_query($conn, "SELECT * FROM queue WHERE status = 'waiting'"));
11 $totalTables = mysqli_num_rows(mysqli_query($conn, "SELECT * FROM tables"));
12 $tablesAvailable = mysqli_num_rows(mysqli_query($conn, "SELECT * FROM tables WHERE is_available = 1"));
13 $reservationsToday = mysqli_num_rows(mysqli_query($conn, "SELECT * FROM reservations WHERE DATE(reservation_time) = CURDATE()"));
14
15 include 'header_admin.php';
16
17 <?php if (isset($_GET['msg'])) && $_GET['msg'] == 'table_added': ?>
18     <div class="alert alert-success text-center" role="alert">
19         &#xA; New table has been added successfully!
20     </div>
21 </?php endif; ?>
22
23 <?php if (isset($_GET['msg'])): ?>
24     <div class="alert alert-success text-center">
25         <?php
26             if ($_GET['msg'] == 'table_added') echo &#x27; Table added successfully!&#x27;;
27             if ($_GET['msg'] == 'table_updated') echo &#x27; Table updated successfully!&#x27;;
28             if ($_GET['msg'] == 'table_deleted') echo &#x27; Table deleted successfully!&#x27;;
29         </?php
30     </div>
31 </?php endif; ?>
32
33 <?php endif; ?>
34
35
36 <div class="container py-5">
37     <h2 class="text-center mb-4 text-pink">Admin Dashboard &#xA; </h2>
38
39     <!-- Card Summary -->
40     <div class="row text-white mb-4">
41         <div class="col-md-3">
42             <div class="card shadow rounded p-3" style="background-color: #A7D9F3;">
43                 <h5>Total Queue</h5>
44                 <h2><?= $totalQueue ></h2>
45             </div>
46         </div>
47         <div class="col-md-3">
48             <div class="card shadow rounded p-3" style="background-color: #C4F1C5;">
49                 <h5>Tables Ready</h5>
50                 <h2><?= $tablesAvailable ></h2>
51             </div>
52         </div>
53         <div class="col-md-3">
54             <div class="card shadow rounded p-3" style="background-color: #FFEC6B;">
55                 <h5>Tables in Use</h5>
56                 <h2><?= $totalTables - $tablesAvailable ></h2>
57             </div>
58         </div>
59         <div class="col-md-3">
60             <div class="card shadow rounded p-3" style="background-color: #FFFAC2;">
61                 <h5>Reservations Today</h5>
62                 <h2><?= $reservationsToday ></h2>
63             </div>
64         </div>
65     </div>
66
67     <!-- Section Queue -->
68     <div class="bg-white p-4 rounded shadow-sm mb-4">
69         <h4 class="mb-3 text-pink"> Queue List </h4>
70         <table class="table table-bordered">
71             <thead class="table-pink text-white">
72                 <tr>
73                     <th>No/<th>
74                     <th>Queue Number</th>
75                     <th>User</th>
76                     <th>Party Size</th>
77                     <th>Status</th>
78                     <th>Action</th>
79                 </tr>
80             </thead>
81             <tbody>
82                 <?php
83                     $queue = mysqli_query($conn, "SELECT q.*, u.username FROM queue q
84                         JOIN users u ON q.user_id = u.id
85                         WHERE q.status != 'done'
86                         ORDER BY q.created_at ASC");
87
88                     $no = 1;
89                     while ($row = mysqli_fetch_assoc($queue)):
90
91                         <tr>
92                             <td><?= $no++ ></td>
93                             <td><?= str_pad($row['queue_number'], 3, '0', STR_PAD_LEFT) ></td>
94                             <td><?= $row['username'] ></td>
95                             <td><?= $row['party_size'] ></td>
96                             <td><?= $row['status'] ></td>
97                             <td><?php if ($row['status'] == 'waiting'): ?>
98                                 <a href="call_queue.php?queue_id=<?= $row['id'] >"> Call </a>
99                             <?php elseif ($row['status'] == 'called'): ?>
100                                 <a href="assign_table.php?queue_id=<?= $row['id'] >"> Assign Table </a>
101                             <?php else: ?>
102                                 <span class="text-muted">No action</span>
103                             </?php endif; ?>
104                         </td>
105                     </tr>
106                 <?php endwhile; ?>
107             </tbody>
108         </table>
109     </div>
```

- **Call:** Move the customer to the next stage.
- **Assign Table:** Allocate a table to the queue.
- No action for **seated**.

## 4. Table Management

- Provides a summary table of all restaurant tables including: Table number, Capacity, Availability status (Available/Occupied), Actions: Free the table, edit table info, delete table (with confirmation).
- Includes an “**Add Table**” button and hidden form to add new tables dynamically.

## 5. Reservation Management

Lists all **upcoming reservations** in a table format showing: Date & time, User info, Party size, Status (**pending**, **seated**, or **completed**), Actions : (**Seat Now**: Mark as seated, **Edit**: Modify reservation, **Delete**: Remove reservation with confirmation.)

## 6. Feedback Alerts (UI Enhancements)

- Uses **SweetAlert2** to show success or warning popups for actions like adding, updating, or deleting a table.
- Also displays contextual success/danger bootstrap alerts for various operations via `$_GET['msg']` parameters.

### 2. Admin\_menu.php

The screenshot shows a table titled "99 Admin Menu List". The table has columns: #, Image, Name, Price, and Actions. The data rows are:

#	Image	Name	Price	Actions
1		Strawberry Cake	Rp 25.000	<a href="#">Edit</a> <a href="#">Delete</a>
2		Rosy Latte	Rp 20.000	<a href="#">Edit</a> <a href="#">Delete</a>
3		Pink Creamy Pasta	Rp 22.000	<a href="#">Edit</a> <a href="#">Delete</a>
4		Strawberry Cupcake	Rp 35.000	<a href="#">Edit</a> <a href="#">Delete</a>
5		Pink Milkshake	Rp 40.000	<a href="#">Edit</a> <a href="#">Delete</a>
6		Sweet Macarons	Rp 30.000	<a href="#">Edit</a> <a href="#">Delete</a>
7		Dessert Cake	Rp 45.000	<a href="#">Edit</a> <a href="#">Delete</a>

`$menuQuery = mysqli_query($conn, "SELECT * FROM menu");`

This is the **Read** part of CRUD – it fetches all menu items from the database and displays them in a table format. Each menu row shows: Menu ID, Image, Name, Price, Actions (Edit/Delete)

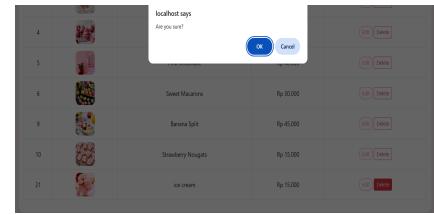
- if (isset(\$\_GET['delete'])) {  
    \$id = \$\_GET['delete'];

```

1 </php>
2 session_start();
3 if (!isset($_SESSION['user_id'])) || $_SESSION['role'] != 'admin') {
4     header("Location: login.php");
5     exit;
6 }
7
8 include 'config.php';
9 include 'header_admin.php';
10
11 // Handle Delete Menu
12 if (isset($_GET['delete'])) {
13     $id = $_GET['delete'];
14     mysqli_query($conn, "DELETE FROM menu WHERE id = $id");
15     header("Location: admin_menu.php?msg=deleted");
16     exit;
17 }
18
19 <div class="container py-5" style="background-color:#ffffcc">
20 <div class="d-flex justify-content-between align-items-center mb-4">
21 <div class="text-pink fw-bold" style="font-size:1.2em; color:#e67e22">99 Admin Menu List </div>
22 <a href="add_menu.php" class="btn btn-pink" style="font-size:0.9em; color:#fff">Add Menu</a>
23 </div>
24 </div>
25
26 <div class="table-responsive bg-white p-3 rounded-4 shadow">
27 <table class="table table-bordered align-middle text-center">
28 <thead class="table-pink text-pink">
29 <tr>
30 <th>#</th>
31 <th>Image</th>
32 <th>Name</th>
33 <th>Price</th>
34 <th>Actions</th>
35 </tr>
36 </thead>
37 <tbody>
38 <?php
39     $menuQuery = mysqli_query($conn, "SELECT * FROM menu");
40     while ($row = mysqli_fetch_assoc($menuQuery)) {
41         <tr>
42             <td><?php echo $row['id'] ?></td>
43             <td></td>
44             <td><?php echo htmlspecialchars($row['name']) ?></td>
45             <td><?php echo number_format($row['price'], 0, ',', '.') ?></td>
46             <td>
47                 <a href="edit_menu.php?id=<?php echo $row['id'] ?>" class="btn btn-sm btn-outline-pink" style="color:#e67e22; font-size:0.8em; border-radius:10px; padding:5px 10px; margin-right:10px">Edit</a>
48                 <a href="#" class="btn btn-sm btn-outline-danger" style="color:#e67e22; font-size:0.8em; border-radius:10px; padding:5px 10px; border:1px solid #e67e22; background-color:#fff; margin-right:10px">Delete</a>
49             </td>
50         </tr>
51     </tbody>
52 </table>
53 </div>
54 </div>
55 </div>
56 </div>
57

```

```
mysqli_query($conn, "DELETE FROM menu WHERE id = $id");
header("Location: admin_menu.php?msg=deleted");
exit;
}
```



This handles the **Delete** operation. When an admin clicks the "Delete" button, the menu item with the specified **id** is removed from the database.

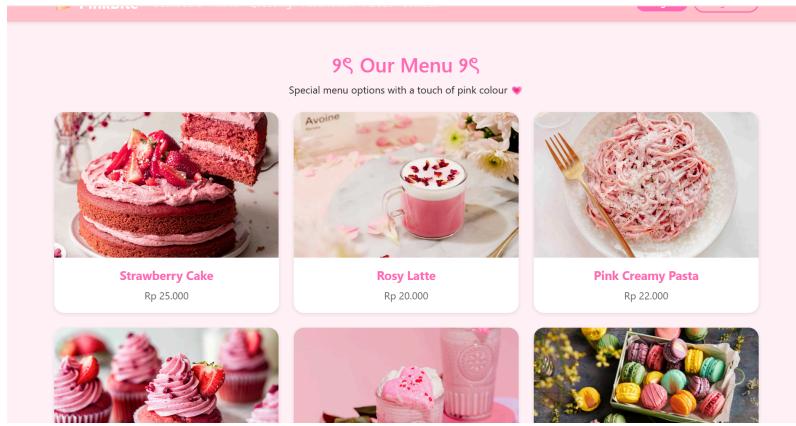
```
<a href="add_menu.php" class="btn btn-pink">+ Add Menu</a>
<a href="edit_menu.php?id=<?= $row['id'] ?>" class="btn btn-sm btn-outline-pink">Edit</a>
```

The "Add Menu" button links to `add_menu.php` (for the **Create** operation).

The "Edit" button links to `edit_menu.php`, passing the `id` for editing (**Update** operation).

<h2>9 🎀 Add New Menu Item 🎀 9</h2>	
Name	<input type="text"/>
Price	<input type="text"/>
Image Upload	<input type="file"/> Choose File No file chosen
<button>之心 Add Menu 心之</button> <button>Cancel</button>	
<hr/>	
<h2>9 🎀 Edit Menu 🎀 9</h2>	
Name	<input type="text"/> Strawberry Cake
Price	<input type="text"/> 25000
Current Image	
<input type="file"/> Upload New Image (optional)	
<input type="file"/> Choose File No file chosen	
<button>之心 Save Changes 心之</button> <button>Cancel</button>	

```
1 <?php
2 session_start();
3 if (!isset($_SESSION['user_id']) || $_SESSION['role'] != 'admin') {
4     header('location: login.php');
5     exit();
6 }
7
8 include 'config.php';
9 include 'header-admin.php';
10
11 // Handle add menu
12 if ($_POST['add_menu']) {
13     $name = $_POST['name'];
14     $price = $_POST['price'];
15
16     // Upload Image
17     $image_name = $_FILES['image']['name'];
18     $tmp_name = $_FILES['image']['tmp_name'];
19     $upload_dir = 'uploads/';
20     $target_file = $upload_dir . basename($image_name);
21
22     if (move_uploaded_file($tmp_name, $target_file)) {
23         mysql_query("com, INSERT INTO menu (name, price) VALUES ('$name', '$price')");
24         $target_file = $target_file;
25         header('location: admin_menu.php?msg=added');
26         exit();
27     } else {
28         echo "<script>alert('Image upload failed')</script>";
29     }
30 }
31
32 <div><container py-5 style="background-color: #fffff5;">
33 <div><row justify-content-center>
34 <div class="col-12">
35     <div class="card shadow-lg rounded-4 border-0">
36         <div class="card-header bg-white text-center">
37             <h3>Add New Menu Item <span style="color: red; font-weight: bold; text-pink;"># Add New Menu Item <span style="color: red; font-weight: bold; text-pink;"># Add New Menu Item </span></h3>
38         </div>
39         <div class="card-body p-4">
40             <form method="POST" enctype="multipart/form-data">
41                 <div class="mb-3">
42                     <label class="form-label text-pink">Name</label>
43                     <input type="text" name="name" class="form-control rounded-3" required>
44                 </div>
45                 <div class="mb-3">
46                     <label class="form-label text-pink">Price</label>
47                     <input type="number" name="price" class="form-control rounded-3" required>
48                 </div>
49                 <div class="mb-3">
50                     <label class="form-label text-pink">Image Upload</label>
51                     <input type="file" name="image" class="form-control rounded-3" accept="image/*" required>
52                 </div>
53                 <div class="text-center mt-4">
54                     <button type="submit" name="add_menu" class="btn btn-pink px-4">
55                         <span style="color: red; font-weight: bold; text-pink;"># Add New Menu Item <span style="color: red; font-weight: bold; text-pink;"># Add New Menu Item </span>
56                     </button>
57                 </div>
58             </div>
59         </div>
60     </div>
61 </div>
62 </div>
63
64 </div>
```



The form uses POST method with `enctype="multipart/form-data"` to support image uploading.

```
if (isset($_POST['add_menu'])) {
    ...
    mysqli_query($conn, "INSERT INTO menu (name, price, image) VALUES (...)");
}
```

When the form is submitted:

- The image is uploaded to the `uploads/` folder.
- A new menu item is inserted into the `menu` table using SQL `INSERT`.

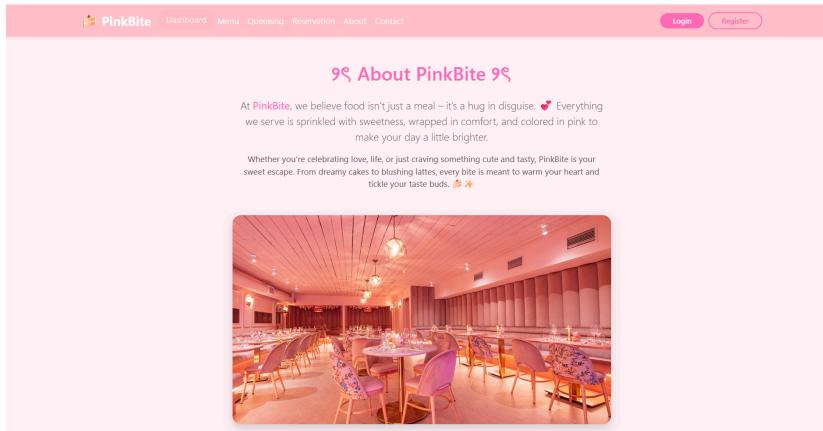
This is the **Create** part of **CRUD** (Create, Read, Update, Delete).

```
if (move_uploaded_file($tmp_name, $target_file)) {
    mysqli_query($conn, "INSERT INTO menu (name, price, image) VALUES ('$name', '$price',
    '$target_file')");
}
```

The script attempts to move the uploaded image from the temporary location to the target directory (`uploads/`). If the upload is successful, it inserts the new menu item into the `menu` table in the database, including the name, price, and image path. After successfully adding the menu item, it redirects to the `admin_menu.php` page with a success message. If the image upload fails, an alert is displayed to inform the user.

The form is structured using Bootstrap for a responsive design. It includes fields for the menu item name, price, and image upload.

### 3. About.php

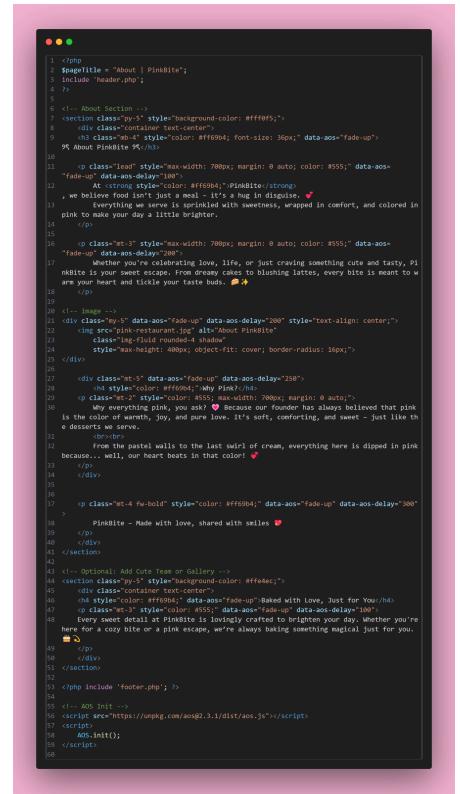


The `about.php` file displays an informative and welcoming **About page** for the PinkBite website.

**Header and footer** are included for consistent layout. A soft **pink design theme** is used throughout. Contains sections like:

- Introduction to PinkBite
- Why everything is pink
- Brand philosophy
- Cute image representing the restaurant
- Optional inspirational section about love and desserts

Uses **AOS (Animate On Scroll)** library for smooth animation effects. Includes an image and styled text with rounded cards and soft colors.



```
1 <?php
2 $pageTitle = "About | PinkBite";
3 $pageHeader = "header.php";
4 ?>
5
6 <!-- About Section -->
7 <section class="py-5" style="background-color: #fff#f51;">
8   <div class="container text-center">
9     <h2>About PinkBite</h2>
10    <p>At PinkBite, we believe food isn't just a meal – it's a hug in disguise. ❤️ Everything we serve is sprinkled with sweetness, wrapped in comfort, and colored in pink to make your day a little brighter. Whether you're celebrating love, life, or just craving something cute and tasty, PinkBite is your sweet escape. From dreamy cakes to blushing lattes, every bite is meant to warm your heart and tickle your taste buds. 🍫✿*
11    <img alt="Pink restaurant interior" data-aos="fade-up" data-aos-delay="100" style="max-width: 700px; margin: 0 auto; color: #fff;" data-aos="fade-up" data-aos-delay="100" data-aos-easing="ease-in-out">
12      At strong style="color: #ff69b4;">PinkBite, we believe food isn't a hug in disguise. ❤️ Everything we serve is sprinkled with sweetness, wrapped in comfort, and colored in pink to make your day a little brighter.
13
14    <p>At strong style="color: #ff69b4;">PinkBite, we believe food isn't a hug in disguise. ❤️ Everything we serve is sprinkled with sweetness, wrapped in comfort, and colored in pink to make your day a little brighter.
15
16    <p>At strong style="color: #ff69b4;">PinkBite, we believe food isn't a hug in disguise. ❤️ Everything we serve is sprinkled with sweetness, wrapped in comfort, and colored in pink to make your day a little brighter.
17
18    <img alt="Pink restaurant interior" data-aos="fade-up" data-aos-delay="200" style="text-align: center;" data-aos="fade-up" data-aos-delay="200" data-aos-easing="ease-in-out">
19      At strong style="color: #ff69b4;">PinkBite, we believe food isn't a hug in disguise. ❤️ Everything we serve is sprinkled with sweetness, wrapped in comfort, and colored in pink to make your day a little brighter.
20
21    <div class="mt-3" style="max-width: 700px; margin: 0 auto; color: #fff;" data-aos="fade-up" data-aos-delay="250" data-aos-easing="ease-in-out">
22      At strong style="color: #ff69b4;">PinkBite, we believe food isn't a hug in disguise. ❤️ Everything we serve is sprinkled with sweetness, wrapped in comfort, and colored in pink to make your day a little brighter.
23
24      <img alt="Pink dessert" data-aos="fade-up" data-aos-delay="300" style="width: 100%; object-fit: cover; border-radius: 10px;" data-aos="fade-up" data-aos-delay="300" data-aos-easing="ease-in-out">
25        At strong style="color: #ff69b4;">PinkBite, we believe food isn't a hug in disguise. ❤️ Everything we serve is sprinkled with sweetness, wrapped in comfort, and colored in pink to make your day a little brighter.
26
27      <div class="mt-5" data-aos="fade-up" data-aos-delay="250" style="text-align: center;" data-aos="fade-up" data-aos-delay="250" data-aos-easing="ease-in-out">
28        At strong style="color: #ff69b4;">PinkBite, we believe food isn't a hug in disguise. ❤️ Everything we serve is sprinkled with sweetness, wrapped in comfort, and colored in pink to make your day a little brighter.
29
30      <div class="mt-2" style="color: #fff; max-width: 700px; margin: 0 auto;" data-aos="fade-up" data-aos-delay="300" data-aos-easing="ease-in-out">
31        At strong style="color: #ff69b4;">PinkBite, you sick! ❤️ Because our founder has always believed that pink is the color of warmth, joy, and pure love. It's soft, comforting, and sweet – just like the desserts we serve.
32
33      <p>From the pastel walls to the last swirl of cream, everything here is dipped in pink because, well, our heart beats in that color! ❤️
34    </div>
35
36    <p>PinkBite – Made with love, shared with smiles ❤️
37
38  </div>
39
40  </div>
41
42  </section>
43
44  <!-- Optional: Add Cute Team or Gallery -->
45  <section class="py-5" style="background-color: #fff#f4c;">
46    <img alt="Pink dessert" data-aos="fade-up" style="width: 100%; object-fit: cover; border-radius: 10px;" data-aos="fade-up" data-aos-delay="100" data-aos-easing="ease-in-out">
47    <div class="mt-3" style="color: #fff;" data-aos="fade-up" data-aos-delay="100" data-aos-easing="ease-in-out">
48      At strong style="color: #ff69b4;">PinkBite, we believe food isn't a hug in disguise. ❤️ Everything we serve is sprinkled with sweetness, wrapped in comfort, and colored in pink to make your day a little brighter.
49
50      <img alt="Pink dessert" data-aos="fade-up" data-aos-delay="150" style="width: 100%; object-fit: cover; border-radius: 10px;" data-aos="fade-up" data-aos-delay="150" data-aos-easing="ease-in-out">
51        At strong style="color: #ff69b4;">PinkBite, we believe food isn't a hug in disguise. ❤️ Everything we serve is sprinkled with sweetness, wrapped in comfort, and colored in pink to make your day a little brighter.
52
53      <img alt="Pink dessert" data-aos="fade-up" data-aos-delay="200" style="width: 100%; object-fit: cover; border-radius: 10px;" data-aos="fade-up" data-aos-delay="200" data-aos-easing="ease-in-out">
54        At strong style="color: #ff69b4;">PinkBite, we believe food isn't a hug in disguise. ❤️ Everything we serve is sprinkled with sweetness, wrapped in comfort, and colored in pink to make your day a little brighter.
55
56      <img alt="Pink dessert" data-aos="fade-up" data-aos-delay="250" style="width: 100%; object-fit: cover; border-radius: 10px;" data-aos="fade-up" data-aos-delay="250" data-aos-easing="ease-in-out">
57        At strong style="color: #ff69b4;">PinkBite, we believe food isn't a hug in disguise. ❤️ Everything we serve is sprinkled with sweetness, wrapped in comfort, and colored in pink to make your day a little brighter.
58
59      <img alt="Pink dessert" data-aos="fade-up" data-aos-delay="300" style="width: 100%; object-fit: cover; border-radius: 10px;" data-aos="fade-up" data-aos-delay="300" data-aos-easing="ease-in-out">
60        At strong style="color: #ff69b4;">PinkBite, we believe food isn't a hug in disguise. ❤️ Everything we serve is sprinkled with sweetness, wrapped in comfort, and colored in pink to make your day a little brighter.
61
62      <img alt="Pink dessert" data-aos="fade-up" data-aos-delay="350" style="width: 100%; object-fit: cover; border-radius: 10px;" data-aos="fade-up" data-aos-delay="350" data-aos-easing="ease-in-out">
63        At strong style="color: #ff69b4;">PinkBite, we believe food isn't a hug in disguise. ❤️ Everything we serve is sprinkled with sweetness, wrapped in comfort, and colored in pink to make your day a little brighter.
64
65    </div>
66
67    <script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>
68    <script>
69      AOS.init();
70    </script>
71
72  </div>
```

## 4. Contact.php

The screenshot shows the 'Contact Us' page. At the top, there's a navigation bar with links for Dashboard, Menu, Queuing, Reservation, About, and Contact. A 'Logout' button is also present. The main content area has a pink header with the title 'Contact Us' and a small heart icon. Below this, a message says 'You can contact us anytime, we are ready to hear your sweet story ❤️'. The left sidebar contains 'Contact Info' with address, phone number, and email details. The right sidebar contains a 'Send Message' form with fields for Name, Gmail, and Message, followed by a 'Send' button.

### Purpose:

To offer users a way to reach the business via **location**, **phone**, **email**, or by sending a **message** using a built-in contact form.

### Key Features:

**Header and footer** are included for consistent site structure.

Left section displays:

- Business address
- Phone number
- Email contact

### Right section contains:

- A **contact form** with fields for name, email, and message
- Submit button (currently the form doesn't process data yet)

Uses **AOS animations** for smooth visual effects on scroll.

The screenshot shows the PHP code for the Contact.php page. It includes header and footer includes, AOS CSS and JS imports, and the main contact form. The form uses Bootstrap classes like 'form-control' and 'text-success' for validation. It also includes icons for location, phone, and email. The code is well-structured with comments and proper indentation.

```
1 <?php include 'header.php'; ?>
2
3 <!-- Contact Header -->
4 <section class="py-5 text-center" style="background-color: #ff64f0;" data-aos="fade-down">
5   <div class="row justify-content-center">
6     <p class="lead" data-aos="fade-up" data-aos-delay="100">Contact Us 🌟 </p>
7     You can contact us anytime, we are ready to hear your sweet story ❤️ </p>
8   </div>
9 </section>
10
11 <!-- Contact Info & Form -->
12 <section class="py-5">
13   <div class="row justify-content-between align-items-center" data-aos="fade-right">
14     <div class="col-md-6 mb-4" data-aos="fade-right">
15       <h4 style="color: #ff64f0;">Contact InfoSend MessageName</label>
27           <input type="text" class="form-control" id="name" name="name" placeholder=" " required>
28         </div>
29         <div class="mb-3">
30           <label for="email" class="form-label">Gmail</label>
31           <input type="email" class="form-control" id="email" name="email" placeholder=" " required>
32         </div>
33         <div class="mb-3">
34           <label for="message" class="form-label">Message</label>
35           <textarea class="form-control" id="message" name="message" rows="4" placeholder=" " required>
36         </div>
37         <button type="submit" class="btn text-white rounded-pill px-4" style="background-color: #ff64f0; border: none; color: white; font-weight: bold; font-size: 1em; padding: 0.5em 1em; text-decoration: none; transition: all 0.3s ease; width: 100%; ">Send 📩 </button>
38       </form>
39     </div>
40   </div>
41 </section>
42
43 <?php include 'footer.php'; ?>
44
45 <script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>
46 <script>AOS.init();</script>
47 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
```

## 5. Admin\_customer.php and edit\_customer

**Customer Management**  
Manage your beloved PinkBite customers ❤️

**Add New Customer**

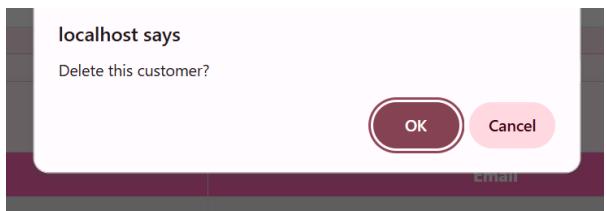
ID	Name	Email	Actions
74	zikra aulia	zikra@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
73	Zalfa Rahmatu Saqillah	zalfa@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
72	Destya Putri Maharani	destya@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
71	Ni Putu Vony	vony@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
70	Falishia Bulan	falistha@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
69	az Zahra putri kamilah	zaza@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>

**Edit Customer**

Customer Name  
jingga cantik

Email Address  
jican@gmail.co

[Back](#) [Save Changes](#)



This file is an admin page for managing customer data in the PinkBite restaurant website project. This file can only be accessed by users with the admin role and has CRUD (Create, Read, Delete) features for customer data.

```
if(isset($_POST['add'])) {
    $name = $_POST['name'];
    $email = $_POST['email'];

    mysqli_query($conn, "INSERT INTO customers (name, email) VALUES ('$name', '$email')");

    header("Location: admin_customers.php");
    exit;
}
```

```
1 <?php
2 session_start();
3 if (!isset($_SESSION['user_id']) || $_SESSION['role'] != 'admin') {
4     header('Location: login.php');
5     exit;
6 }
7
8 include 'config.php';
9 include_header_admin.php';
10
11 // Handle Insert
12 if (isset($_POST['add'])) {
13     $name = $_POST['name'];
14     $email = $_POST['email'];
15     mysqli_query($conn, "INSERT INTO customers (name, email) VALUES ('$name', '$email')");
16     header('Location: admin_customers.php');
17     exit;
18 }
19
20 // Handle Delete
21 if (isset($_GET['delete'])) {
22     $id = $_GET['delete'];
23     mysqli_query($conn, "DELETE FROM customers WHERE id=$id");
24     header('Location: admin_customers.php');
25     exit;
26 }
27
28 // Fetch customers
29 $customers = mysqli_query($conn, "SELECT * FROM customers ORDER BY id DESC");
30 >
31
```

```
1 <?php while ($row = mysqli_fetch_assoc($customers)): ?>
2     <tr>
3         <td class="text-center">?= $row['id'] ?</td>
4         <td>?= htmlspecialchars($row['name']) ?</td>
5         <td>?= htmlspecialchars($row['email']) ?</td>
6         <td class="text-center">
7             <a href="edit_customers.php?id=?= $row['id'] ?>" class="btn btn-sm btn-outline-pink">Edit</a>
8             <a href="#" onclick="return confirm('Delete this customer?') ?> class="btn btn-sm btn-danger">Delete</a>
9         </td>
10     </tr>
11 </php endwhile; ?>
12
```

```
1 <?php if (isset($_GET['msg'])): ?>
2     <script>
3         document.addEventListener('DOMContentLoaded', function() {
4             if ($_GET['msg'] == 'customer_added'):
5                 Swal.fire({
6                     icon: 'success',
7                     title: 'Customer Added!',
8                     text: 'New customer added successfully 🎉',
9                     confirmButtonColor: '#ff69b4'
10                 });
11             elseif ($_GET['msg'] == 'customer_updated'):
12                 Swal.fire({
13                     icon: 'success',
14                     title: 'Customer Updated!',
15                     text: 'Customer information updated successfully 🎉',
16                     confirmButtonColor: '#ff69b4'
17                 });
18             elseif ($_GET['msg'] == 'customer_deleted'):
19                 Swal.fire({
20                     icon: 'success',
21                     title: 'Customer Deleted!',
22                     text: 'Customer successfully deleted 🗑',
23                     confirmButtonColor: '#ff69b4'
24                 });
25         });
26     </script>
27 </php endif; ?>
28 <?php endif; ?>
29
```

This code runs when the admin submits the "Add New Customer" form.  
It saves the customer's **name** and **email** into the **customers** table.

```
if(isset($_GET['delete'])) {  
  
    $id = $_GET['delete'];  
  
    mysqli_query($conn, "DELETE FROM customers WHERE id=$id");  
  
    header("Location: admin_customers.php");  
  
    exit;  
  
}
```

This code runs when the admin clicks the "Delete" button.  
It deletes the customer with the selected **id** from the database.

```
$customers = mysqli_query($conn, "SELECT * FROM customers ORDER BY id DESC");
```

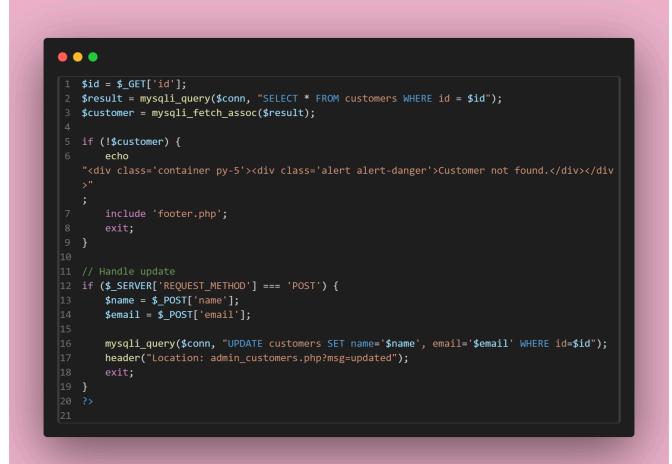
This code gets all customers from the database and sorts them by newest first.

**Read:** Fetch the customer record by its **id** from the **customers** table to pre-fill the edit form.

**Error handling:** If no record is found, show “Customer not found.” and stop.

**Update:** When the form submits, take the new name and email values and run an UPDATE on the **customers** table for that **id**.

**Redirect on success:** After updating, redirect back to the customer list with a “msg=updated” flag.



```
1 $id = $_GET['id'];  
2 $result = mysqli_query($conn, "SELECT * FROM customers WHERE id = $id");  
3 $customer = mysqli_fetch_assoc($result);  
4  
5 if (!$customer) {  
6     echo "7     include 'footer.php';  
8     exit;  
9 }  
10  
11 // Handle update  
12 if ($_SERVER['REQUEST_METHOD'] === 'POST') {  
13     $name = $_POST['name'];  
14     $email = $_POST['email'];  
15  
16     mysqli_query($conn, "UPDATE customers SET name='$name', email='$email' WHERE id=$id");  
17     header("Location: admin_customers.php?msg=updated");  
18     exit;  
19 }  
20 ?>  
21
```

## 6. Assign\_table.php and call\_queue.php

Queue List					
No	Queue Number	User	Party Size	Status	Action
1	#019	ileen daneaya	5 persons	Waiting	<button>Call</button>
2	#020	Salwa Reva Andini	2 persons	Waiting	<button>Call</button>
3	#021	Kaila Anisa Syafitri	7 persons	Waiting	<button>Call</button>
4	#022	Balqis Putri Azzahra	4 persons	Called	<button>Assign Table</button>

This file is used by the admin to assign a customer in the queue to an available table based on their party size. It is part of the PinkBite restaurant queue and table management system. To automatically find and assign a suitable table for a customer (based on `queue_id`) when called from the admin dashboard.

Once `queue_id` is confirmed, the script retrieves the **party size** for that queue ID from the `queue` table. This value is crucial to determine which table is suitable for the group.

The system searches for a table that is:

- Still **available** (`is_available = 1`)
- Has **enough capacity** to fit the party size
- Chosen from the **smallest suitable table** first (ORDER BY `capacity ASC`)

Only **one table** is selected (`LIMIT 1`).

### If a Table is Found

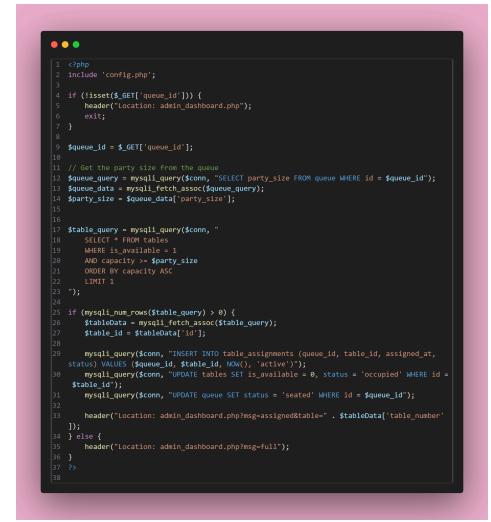
If a suitable table is found:

1. A new entry is added to the `table_assignments` table to record the assignment.
2. The table's status is updated to '`occupied`' and `is_available = 0`.
3. The queue status for that customer is updated to '`seated`'.
4. The admin is redirected to the dashboard with a success message (`msg=assigned&table=...`).

### If No Table is Available

If no table is available that matches the party size:

The admin is redirected back to the dashboard with a failure message (`msg=full`), indicating **all tables are currently full**.



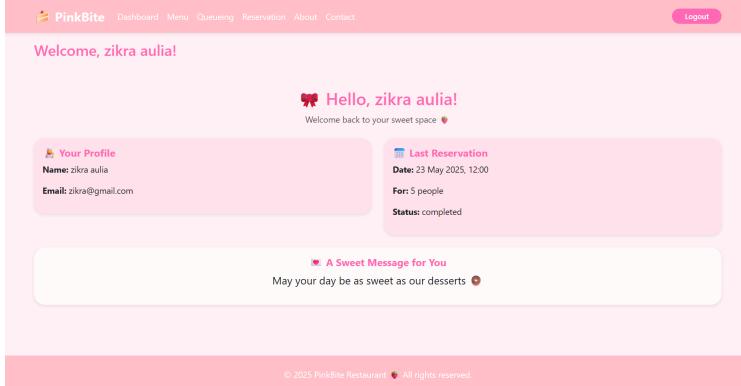
```
1 <?php
2 include 'config.php';
3
4 if (!isset($_GET['queue_id'])) {
5     header("location: admin_dashboard.php");
6     exit();
7 }
8
9 $queue_id = $_GET['queue_id'];
10
11 // Get the party size from the queue
12 $queue_query = mysqli_query($conn, "SELECT party_size FROM queue WHERE id = $queue_id");
13 $queue_data = mysqli_fetch_assoc($queue_query);
14 $party_size = $queue_data['party_size'];
15
16
17 $table_query = mysqli_query($conn, "
18     SELECT * FROM tables
19     WHERE is_available = 1
20     AND capacity >=$party_size
21     ORDER BY capacity ASC
22     LIMIT 1
23 ");
24
25 if (mysqli_num_rows($table_query) > 0) {
26     $tableData = mysqli_fetch_assoc($table_query);
27     $table_id = $tableData['id'];
28
29     mysqli_query($conn, "INSERT INTO table_assignments (queue_id, table_id, assigned_at, status) VALUES ($queue_id, $table_id, NOW(), 'active')");
30     mysqli_query($conn, "UPDATE tables SET is_available=0, status='occupied' WHERE id = $table_id");
31     mysqli_query($conn, "UPDATE queue SET status = 'seated' WHERE id = $queue_id");
32
33     header("location: admin_dashboard.php?msg=assigned&table=$tableData['table_number']");
34 } else {
35     header("location: admin_dashboard.php?msg=full");
36 }
37 ?>
```

The status of that queue entry is updated to '`called`' in the `queue` table of the database.



```
1 $id = $_GET['id'];
2 mysqli_query($conn, "UPDATE queue SET status = 'called' WHERE id = $id");
```

## 7. Customer\_dashboard.php



This file is the **dashboard for customers** after they log in. It displays the logged-in user's customer data and ensures it exists in the database.

```
1 <?php
2 session_start();
3 if (!isset($_SESSION['user_id']) || $_SESSION['role'] != 'customer') {
4     header('Location: login.php');
5     exit();
6 }
7
8 include 'config.php';
9 include 'header.php';
10
11 $user_id = $_SESSION['user_id'];
12
13 $query = mysqli_query($conn, "SELECT * FROM customers WHERE user_id = $user_id");
14 $customer = mysqli_fetch_assoc($query);
15
16 if (!$customer) {
17     $getUser = mysqli_query($conn, "SELECT username, email FROM users WHERE id = $user_id");
18     $userData = mysqli_fetch_assoc($getUser);
19     $username = $userData['username'] ?? 'Customer';
20     $email = $userData['email'] ?? '';
21
22     mysqli_query($conn, "INSERT INTO customers (name, email, user_id) VALUES ('$username',
23     '$email', '$user_id')");
24
25     $query = mysqli_query($conn, "SELECT * FROM customers WHERE user_id = $user_id");
26     $customer = mysqli_fetch_assoc($query);
27
28     $cust_name = htmlspecialchars($customer['name'] ?? 'Customer');
29     $cust_email = htmlspecialchars($customer['email'] ?? 'Not set');
30 }
31
```

## Customer Data Retrieval

It queries the **customers** table to find customer info based on **user\_id**.

If **no customer record is found**, the system:

- Queries the **users** table to get **username** and **email**.
- Automatically **inserts a new customer profile** into the **customers** table.
- After that, it re-fetches the new customer data to use in the dashboard.
- The name and email are saved in **\$cust\_name** and **\$cust\_email**, sanitized using **htmlspecialchars()** for safe output.

## 8. Delete \_table.php , delete \_reservation.php , delete \_queue.php

- delete\_table.php

```
1 if (isset($_GET['id'])) {  
2     $id = $_GET['id'];  
3     mysqli_query($conn, "DELETE FROM tables WHERE id = $id");  
4 }  
5
```

- delete\_reservation.php

```
1 mysqli_query($conn, "DELETE FROM reservations WHERE id = $id");  
2  
3
```

- delete\_queue.php

```
1 $id = $_GET['id'];  
2 mysqli_query($conn, "DELETE FROM queue WHERE id = $id");  
3
```

Deletes a specific **queue record** based on the **id** from the URL. Used to remove a customer from the waiting list.

Deletes a **reservation entry** using its **id**. Used when cancelling a reservation.

Deletes a **table record** from the database. Only runs if an **id** is provided in the URL.

## 9. Edit\_table.php, add\_table.php, delete\_table.php,

```

1 <?php include 'header.php'; ?>
2 <?php if (isset($_GET['msg'])) : ?>
3   <script>
4     document.addEventListener('DOMContentLoaded', function() {
5       if ($_GET['msg'] == 'table_added') {
6         Swal.fire({
7           icon: 'success',
8           title: 'New table added!',
9           text: 'Table added successfully 🎉',
10          confirmButtonColor: '#ff9900'
11        });
12      } else if ($_GET['msg'] == 'table_updated') {
13        Swal.fire({
14          icon: 'success',
15          title: 'Table updated',
16          text: 'Table information updated successfully 🎉',
17          confirmButtonColor: '#ff9900'
18        });
19      } else if ($_GET['msg'] == 'table_deleted') {
20        Swal.fire({
21          icon: 'success',
22          title: 'Table Deleted!',
23          text: 'Table successfully deleted 🚫',
24          confirmButtonColor: '#ff9900'
25        });
26      }
27    })
28  </script>;
29 </body>;
30 </html>;
31 <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>;
32 <script>
33   function confirmDelete(tableId, tableName) {
34     Swal.fire({
35       title: 'Oh no! 🚫',
36       text: 'Do you really wanna remove table #' + tableName,
37       icon: 'warning',
38       showCancelButton: true,
39       confirmButtonColor: '#ff9900',
40       cancelButtonColor: '#333',
41       confirmButtonText: 'Yes, remove it 🚫',
42       cancelButtonText: 'Nope, keep it!'
43     }).then(result => {
44       if (result.isConfirmed) {
45         window.location.href = 'delete_table.php?id=' + tableId;
46       }
47     });
48   }
49 </script>;
50 <?php if (isset($_GET['user'])) : ?>
51   <div class="alert alert-success alert-dismissible fade show" role="alert">
52     User has been successfully assigned to <strong>$_GET['user']</strong>!
53   </div>
54 <?php endif; ?>
55 <?php if (isset($_GET['msg']) && $_GET['msg'] == 'assigned' && isset($_GET['user'])) : ?>
56   <div class="alert alert-success alert-dismissible fade show" role="alert">
57     User has been successfully assigned to <strong>$_GET['user']</strong>!
58   </div>
59 <?php endif; ?>
60 <?php if (isset($_GET['msg']) && $_GET['msg'] == 'full') : ?>
61   <div class="alert alert-danger alert-dismissible fade show" role="alert">
62     No available table for reservation 🚫
63   </div>
64 <?php endif; ?>
65 <?php elseif (isset($_GET['msg']) && $_GET['msg'] == 'full') : ?>
66   <div class="alert alert-danger alert-dismissible fade show" role="alert">
67     No available table for reservation 🚫
68   </div>
69 <?php endif; ?>
70 <?php if (isset($_GET['msg'])) : ?>
71   <div class="alert alert-success text-center">
72     <?php if ($_GET['msg'] == 'reservation_deleted') echo
73       "Reservation has been successfully deleted. 🚫";
74     if ($_GET['msg'] == 'reservation_updated') echo
75       "Reservation has been updated successfully 🎉";
76   </div>
77 <?php endif; ?>
78 <?php endif; ?>
79 <?php endif; ?>
80 <?php endif; ?>

```

SweetAlert is used to show success alerts after **Create, Update, or Delete**:

- Messages like `table_added`, `table_updated`, `table_deleted` are used.

Function `confirmDelete(tableId, tableName)`:

- Shows a SweetAlert confirmation popup before deleting a table.
- If confirmed, it redirects to: `delete_table.php?id=${tableId}` → which deletes the table by its `id`.

Also includes alerts for reservation actions like `reservation_deleted`.

### `add_table.php (CREATE)`

- This page is used to **add a new table** to the `tables` database.
- It contains a form that sends data (`table_number`, `capacity`, `status`) to `process_add_table.php` using POST.
- **CRUD Action: Create** – inserting new data into the database.

### `edit_table.php (READ + UPDATE)`

First, it checks the URL for `id` using `$_GET['id']` and runs:

`SELECT * FROM tables WHERE id = $id` → to **read a specific table's data** from the database.

Then it shows that data in a form for editing.

After editing, the form sends updated data to `process_edit_table.php`.

#### **CRUD Actions:**

- **Read** – getting table data from the database
- **Update** – sending changes to update the table data

## 10. Free\_table.php

Table Status			
Table	Capacity	Status	Action
T01	2 people	Available	Ready <a href="#">Delete</a> <a href="#">Edit</a>
T02	2 people	Available	Ready <a href="#">Delete</a> <a href="#">Edit</a>
T03	4 people	Occupied	<a href="#">Free Table</a> <a href="#">Delete</a> <a href="#">Edit</a>
T04	4 people	Available	Ready <a href="#">Delete</a> <a href="#">Edit</a>

This PHP file is used when a table or reservation is finished and needs to be **marked as available** again.

If a **table ID** is sent (via URL), it:

- Updates the table to be available again.
- Marks the table assignment as **completed**.
- Sets the related **queue** and **reservation** (if any) as **done**.

If a **reservation ID** is sent instead, it:

- Marks the reservation's table assignment as **completed**.
- Also sets the **queue** and reservation status to **done**.

```
1 <?php
2 include 'config.php';
3
4 if (!isset($_GET['table_id']) && !isset($_GET['reservation_id'])) {
5     header("Location: admin_dashboard.php");
6     exit;
7 }
8
9 if (isset($_GET['table_id']) && !empty($_GET['table_id'])) {
10    $table_id = $_GET['table_id'];
11
12    mysqli_query($conn, "UPDATE tables SET is_available = 1 WHERE id = $table_id");
13
14    mysqli_query($conn, "UPDATE table_assignments SET status = 'completed' WHERE table_id = $table_id AND status = 'active'");
15
16    mysqli_query($conn, "
17        UPDATE queue
18        SET status = 'done'
19        WHERE id IN (
20            SELECT queue_id FROM table_assignments
21            WHERE table_id = $table_id AND status = 'completed'
22        )
23    ");
24
25    mysqli_query($conn, "
26        UPDATE reservations
27        SET status = 'completed'
28        WHERE id IN (
29            SELECT reservation_id FROM table_assignments
30            WHERE table_id = $table_id
31            AND status = 'completed' AND reservation_id IS NOT NULL
32        );
33    } elseif (isset($_GET['reservation_id']) && !empty($_GET['reservation_id'])) {
34        $reservation_id = $_GET['reservation_id'];
35
36        mysqli_query($conn, "UPDATE table_assignments SET status = 'completed' WHERE
37        reservation_id = $reservation_id AND status = 'active'");
38
39        mysqli_query($conn, "
40            UPDATE queue
41            SET status = 'done'
42            WHERE id IN (
43                SELECT queue_id FROM table_assignments
44                WHERE reservation_id = $reservation_id AND status = 'completed',
45            )
46        ");
47
48        mysqli_query($conn, "
49            UPDATE reservations
50            SET status = 'completed'
51            WHERE id = $reservation_id
52        ");
53
54    header("Location: admin_dashboard.php?msg=freed");
55}
```

## 11. Queueing.php

The screenshot shows a pink-themed web application. At the top, there's a navigation bar with links like 'Dashboard', 'Menu', 'Queueing', 'Reservation', 'About', 'Contact', and 'Logout'. Below the navigation is a section titled 'Your Current Queue' with a sub-section 'Your Queue History'. A message box says 'You haven't taken a queue number yet' with a 'Take Queue' button. Below this, a table shows a single row of data: Queue Number #034, Party Size 5 persons, Status DONE, and Date 22 May 2025, 14:36. At the bottom is a 'Back to Dashboard' link.

This file lets a **customer** view their **current queue**, take a **new queue number**, and see their **queue history**.

It interacts with the **queue** table in the database.

**C = Create:** When a customer takes a new queue, it inserts a new row into the **queue** table.

**R = Read:** It fetches the current queue, queue history, and the current called number.

No **Update** or **Delete** in this file.

The screenshot shows a pink-themed web application. At the top, there's a section titled 'Take Queue Number' with a sub-section 'Number of People'. A dropdown menu is open, showing 'Select number of people'. Below this is a large pink button with a white 'Take Queue' icon and text.

The screenshot shows a pink-themed web application. At the top, there's a section titled 'Your Current Queue' with a sub-section 'Queue Number'. A large bold number '#035' is displayed. Below it is a table with five rows: 'Restaurant: PinkBite Resto', 'Date: Friday, 23 May 2025', 'People ahead: 34', 'Estimated wait: 102 minutes', and 'Location: President University Campus Area'. At the bottom is a message: 'Please stay nearby. You'll be called soon' with a smiling face emoji.

## 12. Reservation.php

This PHP script handles the creation and display of restaurant reservations for logged-in customers. The flow can be divided into two main parts:

### 1. Reservation Submission (Insert Operation)

When a POST request is detected (meaning the user submitted the reservation form), the script captures the following inputs from the form:

- name – the customer's name
- date and time – which are combined into a datetime format
- people – the number of guests

Each input is sanitized using `mysqli_real_escape_string()` to mitigate SQL injection risks. Then, an INSERT query is constructed to store this reservation data in the `reservations` table with the following fields.

- `user_id`: taken from the current session
- `name`: customer input
- `reservation_time`: a combination of date and time
- `party_size`: total number of people
- `status`: set to 'pending' by default
- `created_at`: automatically recorded using `NOW()`

If the query runs successfully, a flag `$success` is set to `true` to indicate the booking was stored.

### 2. Fetching the Latest Reservation (Select Operation)

After a reservation is submitted (or whenever the page is accessed), the script fetches the user's most recent reservation using:

```
SELECT * FROM reservations WHERE user_id = $user_id ORDER BY created_at DESC LIMIT 1
```

This ensures that only the latest booking is displayed to the customer. The data is stored in the `$reservation` variable and later shown as a confirmation summary.

```
1 // Handle form submission
2 if($_SERVER['REQUEST_METHOD'] == "POST") {
3     $name = mysqli_real_escape_string($conn, $_POST['name']);
4     $date = mysqli_real_escape_string($conn, $_POST['date']);
5     $time = mysqli_real_escape_string($conn, $_POST['time']);
6     $people = (int) $_POST['people'];
7
8     $insert = "INSERT INTO reservations (user_id, name, reservation_time, party_size,
9         status, created_at)
10         VALUES ('{$user_id}', '$name', '$date $time', '$people', 'pending', NOW());
11
12     if(mysqli_query($conn, $insert)) {
13         $success = true;
14     }
15 }
16
17 // Fetch latest reservation
18 $query = "SELECT * FROM reservations WHERE user_id = {$user_id} ORDER BY created_at DESC
19
20 $resResult = mysqli_query($conn, $query);
21 if($resResult && mysqli_num_rows($resResult) > 0) {
22     $reservation = mysqli_fetch_assoc($resResult);
23 }
24
25
26
```

```
1 <?php if ($success): ?>
2     <div class="success-message">
3         Your reservation has been made successfully! See you at PinkBite ❤️
4     </div>
5     </?php endif; ?>
6
7
8     <?php if ($reservation): ?>
9         <div class="reservation-card mt-5">
10             <h3>Your Latest Reservation</h3>
11             <ul>
12                 <li><strong>Name:</strong> <code>{$reservation['name']}</code>
13                 </li>
14                 <li><strong>Date:</strong> <code>date('Y, d M Y', strtotime($reservation[reservation_time]))</code> WIB</li>;
15                 <li><strong>Time:</strong> <code>date('H:i', strtotime($reservation[reservation_time]))</code> WIB</li>;
16                 <li><strong>Party Size:</strong> <code>{$reservation['party_size']}</code></li>;
17                 <li><strong>Status:</strong> <code>ucfirst($reservation['status'])</code></li>;
18             </ul>
19             <div class="text-muted small">Show this to staff on your reservation day ❤️</div>
20         </div>
21     </?php endif; ?>
22
23
```

### a. reservation\_add.php

```
1 <?php while ($table = mysqli_fetch_assoc($tables)): ?>
2   <option value=<?= $table['id'] ?>>Table #<?= $table['table_number'] ?> (
3     <?= $table['capacity'] ?> seats); />
4   </option>
5 </table>
```

Before displaying the form, the script queries the database to retrieve all tables where `status = 'available'`. This ensures the admin can only assign reservations to free tables.

```
1 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
2   $customer_name = $_POST['customer_name'];
3   $reservation_time = $_POST['reservation_time'];
4   $guest_count = $_POST['guest_count'];
5   $table_id = $_POST['table_id'];
6
7   $stmt = mysqli_prepare($conn, "INSERT INTO
8     reservations (customer_name, reservation_time, guest_count, table_id, status) VALUES
9     (?, ?, ?, ?, pending)");
10  mysqli_stmt_bind_param($stmt, "ssii", $customer_name, $reservation_time, $guest_count,
11    $table_id);
12  mysqli_stmt_execute($stmt);
13
14  echo "<script>alert('Reservation added!'); window.location='reservations_view.php';</script>";
15  exit;
16 }
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
```

```
$tables = mysqli_query($conn, "SELECT * FROM tables WHERE status = 'available'");
```

### 1. Handling the Form Submission

When the form is submitted using the POST method, the script:

- Collects form input: customer name, reservation time, number of guests, and selected table.
- Uses a **prepared statement** to safely insert the reservation into the `reservations` table, setting the status to 'pending'.

### 2. Redirection and Confirmation

After successful insertion, a JavaScript `alert` is triggered to confirm the action, and the user is redirected to `reservations_view.php`.

### b. reservation\_edit.php

```
1 $id = $_GET['id'] ?? null;
2
3 if (!$id) {
4   echo "Invalid reservation ID.";
5   exit;
6 }
7
8 $res = mysqli_query($conn, "SELECT * FROM reservations WHERE id = $id");
9 $reservation = mysqli_fetch_assoc($res);
10
11 if (!$reservation) {
12   echo "Reservation not found.";
13   exit;
14 }
15
16 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
17   $new_status = $_POST['status'];
18
19   $stmt = mysqli_prepare($conn, "UPDATE reservations SET status = ? WHERE id = ?");
20   mysqli_stmt_bind_param($stmt, "si", $new_status, $id);
21   mysqli_stmt_execute($stmt);
22
23   if ($new_status === 'seated') {
24     mysqli_query($conn, "UPDATE tables SET status = 'occupied' WHERE id = ($reservation
25     ['table_id'])");
26   }
27
28   echo "<script>alert('Reservation updated!'); window.location='reservations_view.php';</script>";
29 }
30
31
```

Date	Time	User	People	Status	Action
23 May 2025	12:00	zikra aulia	5 people	Completed	-
24 May 2025	10:24	jicaput	3 people	Pending	<button>Seat Now</button> <button>Edit</button> <button>Delete</button>

**Edit Reservation**

Name: jicaput

Phone: 087783021100

Reservation Time: 05/24/2025 10:24 AM

Party Size: 3

Status: Pending

This file allows an **admin** to update the status of a specific reservation. It includes validation, data retrieval, and status updating functionalities.

## **1. Reservation ID Validation**

It retrieves the `id` from the URL (`$_GET['id']`). If the ID is missing or invalid, the script stops and shows an error. This prevents undefined or unauthorized access to reservation data.

## **2. Fetch Reservation Data**

The reservation details are fetched from the database using the ID. If no matching reservation is found, an error message is shown.

```
$res = mysqli_query($conn, "SELECT * FROM reservations WHERE id = $id");
```

## **3. Update Logic on Form Submission**

When the admin submits the form:

- The new status is read from the form.
- A prepared statement is used to securely update the reservation's status.
- If the new status is '`seated`', the corresponding table's status is also updated to '`occupied`' in the `tables` table.

This ensures data consistency between reservations and tables.

## **4. Confirmation & Redirection**

After updating, an alert is shown to confirm the update, and the admin is redirected to the reservation view page.

## 13. login.php

# Login to PinkBite

Email address

Password

 • o

**Login**

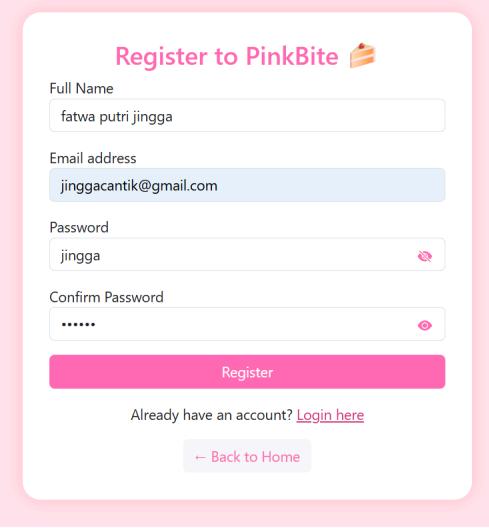
Don't have an account? [Register here](#)

[← Back to Dashboard](#)

```
include 'config.php'; → Connects to the database.  
session_start(); → Starts the session to track login  
status.  
if ($_SERVER["REQUEST_METHOD"] == "POST") →  
Checks if the login form is submitted.  
$email & $password → Takes user input (email &  
password).  
mysqli_query(...) → Finds user in users table by  
email.  
password_verify(...) → Verifies input password with  
hashed one in DB.
```

```
1 $error = "";
2
3 if ($_SERVER['REQUEST_METHOD'] == "POST") {
4     $email = mysqli_real_escape_string($conn, $_POST['email']) ?? '';
5     $password = $_POST['password'] ?? '';
6
7     $query = mysqli_query($conn, "SELECT * FROM users WHERE email='$email'");
8
9     if ($query && mysqli_num_rows($query) == 1) {
10         $user = mysqli_fetch_assoc($query);
11
12         if ($password_verify($password, $user['password'])) {
13             if (isset($user['role'])) {
14                 if ($user['role'] == 'admin') {
15                     if ($SESSION['admin_logged_in'] == true) {
16                         $SESSION['user_id'] = $user['id'];
17                         $SESSION['role'] = "admin";
18                         header('Location: admin_dashboard.php');
19                         exit();
20                     } else if ($user['role'] == 'customer') {
21                         if ($SESSION['customer_logged_in'] == true) {
22                             $SESSION['customer_id'] = $user['id'];
23                             $SESSION['customer_name'] = $user['username'];
24                             $SESSION['user_id'] = $user['id'];
25                             $SESSION['role'] = "customer";
26                             header('Location: customer_dashboard.php');
27                             exit();
28                         } else {
29                             $error = "Role tidak dikenali.";
30                         }
31                     } else {
32                         $error = "Role tidak ditemukan.";
33                     }
34                 }
35             }
36         }
37
38         $error = "Invalid email or password.";
39     }
40 }
41
42 <?php if (isset($_SESSION["success_message"])): >>
43     <div class="alert alert-success"><?= $SESSION["success_message"]>
44     <?= unset($_SESSION["success_message"])> </div>
45 </?php endif; ?>
46
47 <?php if ($error): ?>
48     <div class="alert alert-danger"><?= $error ?> </div>
49 <?php endif; ?>
50
51 <script>
52     function togglePassword(icon) {
53         const input = icon.previousElementSibling;
54         if (input.type === "password") {
55             input.type = "text";
56             icon.classList.remove("bi-eye-fill");
57             icon.classList.add("bi-eye-slash-fill");
58         } else {
59             input.type = "password";
60             icon.classList.remove("bi-eye-slash-fill");
61             icon.classList.add("bi-eye-fill");
62         }
63     }
64 </script>
65
66
```

## 14. register.php



Full Name  
fatwa putri jingga

Email address  
jinggacantik@gmail.com

Password  
jingga

Confirm Password  
\*\*\*\*\*

Register

Already have an account? [Login here](#)

← Back to Home

```
1 $success = "";
2 $error = "";
3
4 if ($_SERVER["REQUEST_METHOD"] == "POST") {
5     $fullname = mysqli_real_escape_string($conn, $_POST['fullname'] ?? '');
6     $email = mysqli_real_escape_string($conn, $_POST['email'] ?? '');
7     $password = $_POST['password'] ?? '';
8     $confirm = $_POST['confirm'] ?? '';
9
10    if ($password != $confirm) {
11        $error = "Passwords do not match!";
12    } elseif (strlen($password) < 4) {
13        $error = "Password must be at least 4 characters!";
14    } else {
15        $checkEmail = mysqli_query($conn, "SELECT * FROM users WHERE email='$email'");
16        if (mysqli_num_rows($checkEmail) > 0) {
17            $error = "Email already registered!";
18        } else {
19            $hashedPassword = password_hash($password, PASSWORD_DEFAULT);
20            $insertUser = mysqli_query($conn, "INSERT INTO users (username, email, password
21 , role) VALUES ('$fullname', '$email', '$hashedPassword', 'customer')");
22
23            if ($insertUser) {
24                $user_id = mysqli_insert_id($conn);
25
26                $insertCustomer = mysqli_query($conn, "INSERT INTO customers (name
27 , email, user_id) VALUES ('$fullname', '$email', '$user_id')");
28
29                if ($insertCustomer) {
30                    $SESSION['success_message'] = "Registration successful! Please login."
31                    ;
32                    header("Location: login.php");
33                    exit();
34                } else {
35                    $error = "Failed to create customer profile. Please try again.";
36                }
37            }
38        }
39    }
40}
41
42
```

include 'config.php'; → Connects to the database.

session\_start(); → Starts a session to store messages or login data.

When the form is submitted (POST request):

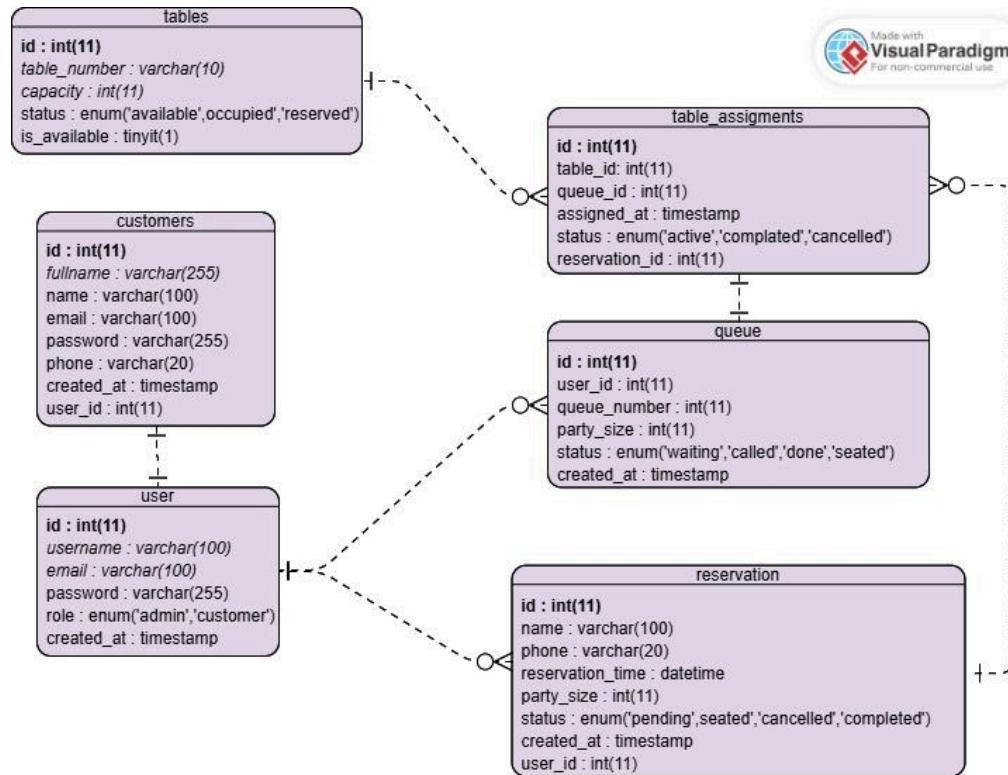
- Gets **fullname**, **email**, **password**, and **confirm** from the form.
- Checks if password and confirmation match.
- Checks if the password is at least 4 characters long.
- Checks if the email is already registered in the **users** table.
- If not registered:
  - Hashes the password using **password\_hash**.
  - Inserts the new user into the **users** table with role **customer**.
  - Gets the new **user\_id** just created.
  - Inserts customer details into the **customers** table.
  - If all successful → redirects to **login.php** with a success message.
- If any error happens → shows it in a red alert box.

## logout.php



```
1 <?php
2 session_start();
3 session_destroy();
4 header("Location: index.php");
5 exit;
6 ?>
7
```

# ERD



## 1. Users

- **PK: id**
- Stores **login info** for all users (admin & customer).
- **1-to-1 → customers:**

- A user *can be* a customer (if role = 'customer')
- `customers.user_id` is a **FK** to `users.id`

## 2. Customers

- **PK: id**
- **FK: user\_id → users.id**
- Stores **extra profile info** for customers only (like name & phone).

 Linked to:

- `users` (1-to-1)
- `queue & reservations` indirectly via `user_id` (from `users`)

### 3. Tables

- **PK: `id`**
- Stores all **table data** (number, size, status).
-  Linked to `table_assignments`
  - **1-to-many → `table_assignments`:**
  - 1 table can be assigned multiple times (different reservations/queues)
  - `table_assignments.table_id` is a **FK** to `tables.id`

### 4. Queue

- **PK: `id`**
- **FK: `user_id` → `users.id`**
- Stores **walk-in queue info** for customers.
-  Linked to:
  - `users` (many queues for one user)
  - `table_assignments` (queue assigned to a table)

### 5. Reservations

- **PK: `id`**
- **FK: `user_id` → `users.id`**
- Stores **reservation info** (name, time, party size).
-  Linked to:
  - `users` (many reservations per user)
  - `table_assignments` (reservation assigned to table)

## 6. Table\_assignments

- **PK:** id
- **FKs:**
  - table\_id → tables.id
  - queue\_id → queue.id
  - reservation\_id → reservations.id
- Used to **assign tables** either for queue or reservation.
- This table **connects multiple entities!**

## 7. Menu

- **PK:** id
- Just stores menu items. Not related to other tables in this diagram.