INDEX.PHP

<?php

include('db.php');

try {

    $pdo = new PDO("mysql:host=$host;dbname=$db", $user, $pass);

    // Setel atribut error mode untuk PDO

    $pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

} catch (PDOException $e) {

    die("Koneksi gagal: " . $e->getMessage());

}

// Tambah todo

if (isset($\_POST['task']) && isset($\_POST['due\_date']) && !isset($\_POST['update\_id'])) {

    $task = htmlspecialchars($\_POST['task']); // Sanitasi input

    $due\_date = $\_POST['due\_date'];

    $status = $\_POST['status']; // Ambil status dari form

    $sql = "INSERT INTO todos (task, due\_date, status) VALUES (:task, :due\_date, :status)";

    $stmt = $pdo->prepare($sql);

    $stmt->bindParam(':task', $task);

    $stmt->bindParam(':due\_date', $due\_date);

    $stmt->bindParam(':status', $status);

    $stmt->execute();

}

// Update status todo

if (isset($\_POST['update\_status\_id']) && isset($\_POST['status'])) {

    $id = $\_POST['update\_status\_id'];

    $status = $\_POST['status'];

    $sql = "UPDATE todos SET status = :status WHERE id = :id";

    $stmt = $pdo->prepare($sql);

    $stmt->bindParam(':id', $id);

    $stmt->bindParam(':status', $status);

    $stmt->execute();

}

// Update todo

if (isset($\_POST['update\_id']) && isset($\_POST['task']) && isset($\_POST['due\_date'])) {

    $id = $\_POST['update\_id'];

    $task = htmlspecialchars($\_POST['task']); // Sanitasi input

    $due\_date = $\_POST['due\_date'];

    $status = $\_POST['status']; // Ambil status dari form

    $sql = "UPDATE todos SET task = :task, due\_date = :due\_date, status = :status WHERE id = :id";

    $stmt = $pdo->prepare($sql);

    $stmt->bindParam(':id', $id);

    $stmt->bindParam(':task', $task);

    $stmt->bindParam(':due\_date', $due\_date);

    $stmt->bindParam(':status', $status);

    $stmt->execute();

}

// Hapus todo

if (isset($\_POST['delete\_id'])) {

    $id = $\_POST['delete\_id'];

    $sql = "DELETE FROM todos WHERE id = :id";

    $stmt = $pdo->prepare($sql);

    $stmt->bindParam(':id', $id);

    $stmt->execute();

}

// Ambil semua todo

$stmt = $pdo->prepare("SELECT \* FROM todos ORDER BY created\_at DESC");

$stmt->execute();

$todos = $stmt->fetchAll();

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Todo List</title>

    <link rel="stylesheet" href="css/hasill.css">

    <link rel="stylesheet" href="css/aksii.css">

    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css" rel="stylesheet">

</head>

<body>

    <h1 id="awal">List Kegiatan Dan Tugas</h1>

    <button id="addTodoBtn">+</button>

    <div id="addTodoForm" style="display:none;">

    <div id="TodoForm">

        <form action="index.php" method="post">

            <h2>Tambah Tugas</h2>

            <input type="text" name="task" placeholder="Tambah tugas" required>

            <input type="date" name="due\_date" required>

            <!-- Menambahkan form untuk status -->

            <select class="pilihanstatus" name="status" required>

                <option value="High">High</option>

                <option value="Medium">Medium</option>

                <option value="Low">Low</option>

            </select>

            <div class="buttons">

                <button type="button" id="cancelBtn">Batal</button>

                <button type="submit">Tambah</button>

            </div>

        </form>

    </div>

</div>

<!-- Tampilkan Tugas -->

<table class="show-todo-section">

<thead class="desatas">

    <tr>

    <td class="Tugas">Tugas</td>

            <td class="Dibuat">Dibuat</td>

            <td class="JW">Jangka Waktu</td>

            <td class="status">Prioritas</td>

            <td class="aksi">Aksi</td>

    </tr>

</thead>

<tbody>

    <?php foreach ($todos as $todo): ?>

        <tr>

            <td>

                <div class="todo-item">

                <div class="list">

    <div style="text-align: center;">

        <input type="checkbox" name="todo\_check[]" value="<?php echo $todo['id']; ?>" />

    </div>

    <div class="taskk"><?php echo htmlspecialchars($todo['task']); ?></div>

</div>

<div class="todo-info">

    <div class="created-at"><?php echo date('d - m - Y', strtotime($todo['created\_at'])); ?></div>

    <div class="shift-right"><?php echo date('d - m - Y', strtotime($todo['due\_date'])); ?></div>

    <div class="st"><strong><?php echo htmlspecialchars($todo['status']); ?></strong></div>

    <button class="menu-toggle" onclick="toggleSlide(<?php echo $todo['id']; ?>)">⋮</button>

</div>

                    <div id="todo-actions-<?php echo $todo['id']; ?>" class="kotakbelakang" style="display: none;">

                        <div class="todo-actions">

                        <button class="edit-btn" data-id="<?php echo $todo['id']; ?>" onclick="showUpdateForm(<?php echo $todo['id']; ?>)">

                                        <i class="fas fa-edit"></i>

                                    </button>

                        <!-- Hapus Tugas -->

                            <form action="index.php" method="post" style="display:inline;" onsubmit="return confirmDelete();">

                                <input type="hidden" name="delete\_id" value="<?php echo $todo['id']; ?>">

                                <button type="submit" class="delete" style="background: none; border: none; cursor: pointer;">

                                    <i class="fas fa-trash-alt"></i>

                                </button>

                            </form>

                        </div>

                    </div>

                </div>

            </td>

        </tr>

                <!-- Modal Edit -->

                <div id="edit-modal-<?php echo $todo['id']; ?>" class="edit-modal" style="display: none;">

                    <div class="modal-content">

                        <form action="index.php" method="post">

                            <input type="hidden" name="update\_id" value="<?php echo $todo['id']; ?>">

                            <h2>Edit Tugas</h2>

                            <input type="text" name="task" value="<?php echo htmlspecialchars($todo['task']); ?>" required>

                            <input type="date" name="due\_date" value="<?php echo $todo['due\_date']; ?>" required>

                            <select class="pilihanstatus" name="status" required>

                                <option value="High" <?php echo ($todo['status'] == 'High') ? 'selected' : ''; ?>>High</option>

                                <option value="Medium" <?php echo ($todo['status'] == 'Medium') ? 'selected' : ''; ?>>Medium</option>

                                <option value="Low" <?php echo ($todo['status'] == 'Low') ? 'selected' : ''; ?>>Low</option>

                            </select>

                            <div style="display: flex; justify-content: space-between; align-items: center;">

                                <button type="button" onclick="cancelUpdateForm(<?php echo $todo['id']; ?>)">Cancel</button>

                                <button type="submit">Update</button>

                            </div>

                        </form>

                    </div>

                </div>

                <!-- Overlay untuk modal -->

                <div id="overlay-<?php echo $todo['id']; ?>" class="overlay" style="display: none;"></div>

    <?php endforeach; ?>

</tbody>

</table>

<script src="js/script.js"></script>

</body>

</html>

DB.PHPD

<?php

// Tentukan informasi koneksi ke database

$host = 'localhost';

$db = 'todo\_app'; // Pastikan nama database ini benar

$user = 'root'; // Username untuk MySQL

$pass = ''; // Password untuk MySQL (kosongkan jika tidak ada password)

try {

// Membuat koneksi PDO

$conn = new PDO("mysql:host=$host;dbname=$db", $user, $pass);

// Set error mode PDO ke exception

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

} catch (PDOException $e) {

echo "Koneksi gagal: " . $e->getMessage();

}

?>

CONNECTIONS.SQL

USE todo\_app;

-- Create the 'todos' table

CREATE TABLE IF NOT EXISTS todos (

id INT AUTO\_INCREMENT PRIMARY KEY,

task VARCHAR(255) NOT NULL,

due\_date DATE NOT NULL,

status ENUM('High', 'Medium', 'Low') DEFAULT 'Medium',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

JS/SCRIPT.JS

   // Fungsi untuk toggle aksi tambahan (misalnya status)

   function toggleSlide(todoId) {

    const actions = document.getElementById('todo-actions-' + todoId);

    console.log(actions); // Debugging: cek apakah kotak ditemukan

    if (actions) {

        actions.style.display = (actions.style.display === 'block' ? 'none' : 'block');

    }

}

// Fungsi untuk menampilkan modal ketika tombol Edit ditekan

function showUpdateForm(id) {

    console.log("Modal ID: " + id); // Debugging: cek apakah id benar

    toggleModal(id, 'show'); // Tampilkan modal dan overlay

}

// Fungsi untuk menutup modal ketika tombol Cancel ditekan

function cancelUpdateForm(id) {

    toggleModal(id, 'hide'); // Sembunyikan modal dan overlay

}

// Fungsi untuk menangani interaksi tombol tambah todo

document.getElementById('addTodoBtn').addEventListener('click', function() {

    document.getElementById('addTodoForm').style.display = 'block';

});

// Fungsi untuk menangani interaksi tombol batal di form tambah todo

document.getElementById('cancelBtn').addEventListener('click', function() {

    document.getElementById('addTodoForm').style.display = 'none';

});

// Fungsi untuk menampilkan konfirmasi penghapusan

function confirmDelete() {

    // Menggunakan confirm() untuk menampilkan kotak konfirmasi

    var confirmation = confirm("Apakah Anda yakin ingin menghapus tugas ini?");

    // Jika pengguna memilih 'OK' (true), maka formulir akan disubmit

    if (confirmation) {

        return true;

    }

    // Jika pengguna memilih 'Cancel' (false), formulir tidak akan disubmit

    return false;

}

// Fungsi untuk menampilkan modal edit

// Fungsi untuk menampilkan modal edit

function showUpdateForm(id) {

    // Dapatkan elemen modal dan overlay berdasarkan ID

    const modal = document.getElementById('edit-modal-' + id);

    const overlay = document.getElementById('overlay-' + id);

    // Pastikan elemen modal dan overlay ditemukan

    if (modal && overlay) {

        modal.style.display = 'block'; // Tampilkan modal

        overlay.style.display = 'block'; // Tampilkan overlay

        console.log(`Modal dengan ID edit-modal-${id} berhasil ditampilkan.`);

    } else {

        console.error(`Modal atau overlay dengan ID ${id} tidak ditemukan.`);

    }

}

// Fungsi untuk menyembunyikan modal

function cancelUpdateForm(id) {

    const modal = document.getElementById('edit-modal-' + id);

    const overlay = document.getElementById('overlay-' + id);

    if (modal && overlay) {

        modal.style.display = 'none'; // Sembunyikan modal

        overlay.style.display = 'none'; // Sembunyikan overlay

    } else {

        console.error(`Modal atau overlay dengan ID ${id} tidak ditemukan.`);

    }

}

// Fungsi untuk menangani klik pada tombol Edit (pada tiap Todo)

document.querySelectorAll('.edit-btn').forEach(function(button) {

    button.addEventListener('click', function() {

        const todoId = button.getAttribute('data-id'); // Ambil ID todo dari data-id

        showUpdateForm(todoId); // Menampilkan modal dengan ID yang sesuai

    });

});

// Fungsi untuk menangani klik pada overlay untuk menutup modal

document.querySelectorAll('.overlay').forEach(function(overlay) {

    overlay.addEventListener('click', function() {

        const todoId = overlay.getAttribute('data-id');

        cancelUpdateForm(todoId); // Menyembunyikan modal dan overlay

    });

});

// Fungsi untuk menangani klik pada tombol Edit (pada tiap Todo)

document.querySelectorAll('.edit-btn').forEach(function(button) {

    button.addEventListener('click', function() {

        const todoId = button.getAttribute('data-id'); // Ambil ID todo dari data-id

        showUpdateForm(todoId); // Menampilkan modal dengan ID yang sesuai

    });

});

// Fungsi untuk menangani klik pada tombol Cancel pada setiap modal

document.querySelectorAll('.cancel-btn').forEach(function(button) {

    button.addEventListener('click', function() {

        const todoId = button.getAttribute('data-id'); // Ambil ID todo dari data-id

        cancelUpdateForm(todoId); // Menyembunyikan modal dengan ID yang sesuai

    });

});

// Fungsi untuk menangani klik pada overlay untuk menutup modal

document.querySelectorAll('.overlay').forEach(function(overlay) {

    overlay.addEventListener('click', function() {

        const todoId = overlay.getAttribute('data-id');

        cancelUpdateForm(todoId); // Menyembunyikan modal dan overlay

    });

});

// Fungsi untuk menangani perubahan status tugas

function toggleModal(id, action) {

    const modal = document.getElementById('edit-modal-' + id);

    const overlay = document.getElementById('overlay-' + id);

    if (action === 'show') {

        modal.style.display = 'block';

        overlay.style.display = 'block';

    } else {

        modal.style.display = 'none';

        overlay.style.display = 'none';

    }

}

document.addEventListener("DOMContentLoaded", function () {

    var dueDateInput = document.querySelector("input[name='due\_date']");

    var submitButton = document.querySelector("button[type='submit']");

    dueDateInput.addEventListener("change", function () {

        var dueDate = dueDateInput.value;

        var today = new Date().toISOString().split("T")[0];

        if (dueDate < today) {

            alert("Tanggal tugas tidak boleh sebelum hari ini!");

            submitButton.disabled = true; // Disable tombol submit

        } else {

            submitButton.disabled = false; // Enable kalau tanggal valid

        }

    });

});

document.addEventListener("DOMContentLoaded", function () {

    var dueDateInput = document.querySelector(".modal-content input[name='due\_date']");

    var updateButton = document.querySelector(".modal-content button[type='submit']");

    dueDateInput.addEventListener("change", function () {

        var dueDate = dueDateInput.value;

        var today = new Date().toISOString().split("T")[0];

        if (dueDate < today) {

            alert("Anda tidak bisa mengedit tanggal sebelum hari ini!");

            updateButton.disabled = true; // Disable tombol update

        } else {

            updateButton.disabled = false; // Enable kalau tanggal valid

        }

    });

});

document.addEventListener("DOMContentLoaded", function () {

    var dueDateInput = document.querySelector("input[name='due\_date']");

    var submitButton = document.querySelector("button[type='submit']");

    dueDateInput.addEventListener("change", function () {

        var dueDate = dueDateInput.value;

        var today = new Date().toISOString().split("T")[0];

        if (dueDate < today) {

            alert("Tanggal tugas tidak boleh sebelum hari ini!");

            submitButton.disabled = true; // Disable tombol submit

        } else {

            submitButton.disabled = false; // Enable kalau tanggal valid

        }

    });

});

document.addEventListener("DOMContentLoaded", function () {

    var dueDateInput = document.querySelector(".modal-content input[name='due\_date']");

    var updateButton = document.querySelector(".modal-content button[type='submit']");

    dueDateInput.addEventListener("change", function () {

        var dueDate = dueDateInput.value;

        var today = new Date().toISOString().split("T")[0];

        if (dueDate < today) {

            alert("Anda tidak bisa mengedit tanggal sebelum hari ini!");

            updateButton.disabled = true; // Disable tombol update

        } else {

            updateButton.disabled = false; // Enable kalau tanggal valid

        }

    });

});

CSS/TODOLIST.CSS

body {

    background: #AECBFA;

}

#awal {

    text-align: center; /\* Untuk memusatkan teks secara horizontal \*/

    padding: 20px 0; /\* Opsional, untuk memberi ruang di atas dan bawah \*/

    color: #000000;

    text-shadow: 2px 2px 4px rgb(53, 53, 53); /\* Menambahkan bayangan pada teks \*/

}

/\* start tanda + dan isi \*/

/\* start tanda + \*/

#addTodoBtn {

    background:rgb(200, 90, 160);

    color: #141414; border-radius: 50%; padding: 15px;

    font-size: 40px; cursor: pointer; box-shadow: 0 4px 6px rgb(0, 0, 0);

    width: 60px; height: 60px; position: fixed;

    bottom: 20px; right: 20px; display: flex;

    justify-content: center;

    align-items: center;

}

/\* end tanda + \*/

#addTodoForm {

    display: flex; /\* Menggunakan flexbox untuk memudahkan penyusunan \*/

    justify-content: center; /\* Menyusun modal di tengah secara horizontal \*/

    align-items: center; /\* Menyusun modal di tengah secara vertikal \*/

    position: fixed;

    top: 0;

    left: 0;

    width: 100%;

    height: 100%;

    background: rgba(0, 0, 0, 0.5);

    z-index: 1000;

}

#TodoForm {

    background: white;

    padding: 40px;

    border-radius: 10px;

    width: 80%;

    max-width: 500px;

    position: relative; /\* Pastikan modal berada di atas overlay \*/

    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3); /\* Tambahkan bayangan untuk memperjelas \*/

    top: 100px;  /\* Menyesuaikan posisi ke atas sedikit \*/

    left: 400px; /\* Menyesuaikan posisi ke kanan sedikit \*/

}

/\* Membatasi lebar div task agar teks tidak melebar \*/

.list div {

    word-wrap: break-word; /\* Memaksa teks panjang untuk turun ke baris berikutnya \*/

    max-width: 200px; /\* Membatasi lebar teks supaya tidak melebar \*/

    white-space: normal; /\* Memungkinkan teks untuk membungkus di baris baru \*/

    max-width: 28ch; /\* Membatasi lebar berdasarkan karakter \*/

}

/\* Membuat elemen created-at, shift-right, st, dan menu-toggle tetap pada posisi \*/

.list {

    display: flex;

    align-items: center;

    gap: 10px; /\* Jarak antara checkbox dan teks \*/

    flex-wrap: nowrap; /\* Jangan membungkus elemen ke baris baru \*/

    width: 25%; /\* Mengurangi lebar kotak agar tidak terlalu lebar \*/

    box-sizing: border-box; /\* Agar padding dan border tidak mempengaruhi ukuran lebar \*/

    margin-right: 10px; /\* Memberikan jarak kanan \*/

}

.todo-info {

    display: flex; /\* Membuat elemen-elemen dalam satu baris \*/

    justify-content: flex-start; /\* Menyelaraskan elemen-elemen di kiri \*/

    align-items: center; /\* Menyelaraskan elemen secara vertikal \*/

    padding: 10px; /\* Memberikan ruang di dalam kotak \*/

    border-radius: 10px; /\* Membuat sudut kotak lebih melengkung \*/

    width: 50%; /\* Menyesuaikan lebar kotak sesuai dengan kontainer induk \*/

    box-sizing: border-box; /\* Memastikan padding tidak mempengaruhi lebar total \*/

    margin-left: 60px; /\* Menggeser kotak lebih ke kiri \*/

}

.created-at, .shift-right, .st, .menu-toggle {

    font-size: 22px;

    white-space: nowrap; /\* Agar teks tetap dalam satu baris \*/

    margin: 0 19px; /\* Memberikan jarak lebih besar antar elemen \*/

}

.st {

    margin-left: 30px; /\* Menambah jarak ke kiri untuk menggeser elemen ke kanan \*/

}

.taskk {

    margin-left: 35px; /\* Menambah jarak ke kiri untuk menggeser elemen ke kanan \*/

}

/\* Menjaga jarak untuk tombol menu \*/

.menu-toggle {

    cursor: pointer;

    background-color: rgba(0, 0, 0, 0.1); /\* Menambahkan kotak transparan di belakang tombol \*/

    padding: 5px;

    border-radius: 5px; /\* Memberikan sedikit kelengkungan pada tombol \*/

}

h2 {

   text-align: center;

}

input, button {

   width: 95%;

   height: 35px;

   padding: 8px;

   margin-bottom: 10px;

   border-radius: 4px;

}

input[type="text"], input[type="date"] {

   border: 1px solid #ccc;

   top: 100px;  /\* Menyesuaikan posisi ke atas sedikit \*/

}

.buttons {

   display: flex; justify-content: space-between;

}

button {

   background-color: #4CAF50; color: white;

   border: none; cursor: pointer;

}

button[type="button"] {

   background-color: #f44336;

}

button:hover {

   opacity: 0.8;

}

/\* end tanda + dan isi \*/

/\* start hasil list\*/

.show-todo-section {

    width: 80%;

    margin: 0 auto;

    padding: 20px;

    border-radius: 5px;

}

.todo-item {

    max-width: 800px; padding: 10px;

    display: flex; gap: 40px; align-items: center;

    border-radius: 15px; margin: 8px auto;

    box-shadow: 0 6px 12px rgba(17, 17, 17, 0.788), 0 8px 24px rgba(0, 0, 0, 0.2);

    background: rgb(255, 153, 204); /\* Warna pink susu yang sedikit lebih tua \*/

    position: relative;

    transition: transform 0.3s ease;

    left: -10px; /\* Menyesuaikan posisi ke kanan sedikit \*/

}

.desatas{

    max-width: 800px; padding: 10px;

    display: flex; gap: 40px; align-items: center;

    border-radius: 15px; margin: 8px auto;

    box-shadow: 0 6px 12px rgba(17, 17, 17, 0.788), 0 8px 24px rgba(0, 0, 0, 0.2);

    background: rgb(200, 90, 160); /\* Pink susu yang lebih tua dan lebih ungu \*/

    position: relative;

    transition: transform 0.3s ease;

    left: -10px; /\* Menyesuaikan posisi ke kanan sedikit \*/

    text-align: left;

}

td {

    font-size: 20px;  /\* Ukuran font 20px \*/

    font-weight: bold;  /\* Membuat font menjadi tebal \*/

}

/\* Gaya khusus untuk masing-masing kolom \*/

td.Tugas {

    padding-right: 310px; /\* Atur sesuai kebutuhan \*/

    position: relative;  /\* Memungkinkan penggunaan left \*/

    left: 80px;  /\* Menyesuaikan posisi ke kanan sedikit \*/

    cursor: move;  /\* Menambahkan kursor bergerak untuk indikasi drag \*/

}

td.Dibuat {

    padding-right: 20px; /\* Atur sesuai kebutuhan \*/

}

td.JW {

    padding-left: 20px;  /\* Menambahkan jarak ke kiri \*/

    padding-right: 25px; /\* Atur sesuai kebutuhan \*/

    white-space: nowrap;  /\* Mencegah teks terpotong \*/

}

td.status {

    padding-right: 40px; /\* Atur sesuai kebutuhan \*/

}

td.aksi {

    padding-right: 30px; /\* Atur sesuai kebutuhan \*/

}

.shift-left {

    transform: translateX(-100%);

}

.menu-toggle {

    position: absolute;

    right: 30px; top: 50%;

    transform: translateY(-50%);

    background: none; border: none; font-size: 24px;

    cursor: pointer; z-index: 1;

}

.todo-item div {

    font-size: 18px;

    font-weight: bold;

    color: #1a1919 ;

}

/\* end hasil list\*/

CSS/AKSI.CSS

.kotakbelakang {

    display: none;

    position: absolute;

    top: -40px;  /\* Menyesuaikan posisi ke atas sedikit \*/

    left: 820px; /\* Menyesuaikan posisi ke kanan sedikit \*/

    width: 20%;  /\* Memperlebar kotak menjadi 80% dari lebar layar \*/

    max-height: 85px;

    padding: 5px;

    overflow: auto;

    box-shadow: 0 6px 10px rgba(0, 0, 0, 0.788), 0 8px 24px rgba(0, 0, 0, 0.2);

    background:rgb(255, 153, 204);

    border-radius: 10px;

}

.pilihanstatus {

    font-size: 13px; /\* Ukuran teks lebih besar \*/

    padding: 10px; /\* Memberikan jarak dalam \*/

    border-radius: 8px; /\* Membulatkan sudut \*/

    background-color: #f9f9f9; /\* Warna latar belakang dropdown \*/

    color: #333; /\* Warna teks \*/

    width: 97px; /\* Lebar dropdown \*/

    cursor: pointer; /\* Penunjuk mouse \*/

    font-weight: bold;

    margin-bottom: 15px; /\* Jarak di bawah kotak \*/

}

.todo-actions form {

    display: inline-block;

    margin: 0 5px;

}

.edit, .delete {

    border-radius: 5px; border: none; padding: 10px 20px; cursor: pointer; color: #fff;

    display: inline-block; text-align: center; margin: 5px;

    position: relative; /\* Menambahkan positioning relatif \*/

    left: 15px; /\* Menambahkan pergeseran ke kanan \*/

}

.edit-btn {

    background-color: #bdba05 !important;

    width: 70px; height: 45px; padding: 10px; border-radius: 5px;

}

.edit:hover {

    background-color: #746110 !important;

}

.delete {

    background-color: #d11d10 !important;

    width: 70px; height: 45px; padding: 10px; border-radius: 5px;

    margin-left: -2px; /\* Menambahkan jarak tambahan untuk tombol delete \*/

}

.delete:hover {

    background-color: #720c04 !important;

}

button i {

    font-size: 20px; /\* Ganti angka ini untuk mengatur ukuran ikon \*/

}

/\* start edit list \*/

/\* Modal \*/

.edit-modal {

    display: flex; /\* Menggunakan flexbox untuk memudahkan penyusunan \*/

    justify-content: center; /\* Menyusun modal di tengah secara horizontal \*/

    align-items: center; /\* Menyusun modal di tengah secara vertikal \*/

    position: fixed;

    top: 0;

    left: 0;

    width: 100%;

    height: 100%;

    background: rgba(0, 0, 0, 0.5);

    z-index: 1000;

}

.modal-content {

    background: white;

    padding: 40px;

    border-radius: 10px;

    width: 80%;

    max-width: 500px;

    position: relative; /\* Pastikan modal berada di atas overlay \*/

    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3); /\* Tambahkan bayangan untuk memperjelas \*/

    top: 90px;  /\* Menyesuaikan posisi ke atas sedikit \*/

    left: 400px; /\* Menyesuaikan posisi ke kanan sedikit \*/

}

.overlay {

    display: flex; /\* Menggunakan flexbox untuk overlay \*/

    justify-content: center; /\* Memastikan overlay berada di tengah \*/

    align-items: center; /\* Memastikan overlay berada di tengah \*/

    position: fixed;

    top: 0;

    left: 0;

    width: 100%;

    height: 100%;

    background: rgba(0, 0, 0, 0.5);

    z-index: 999;

}

/\* end edit list \*/

.menu-toggle {

    position: absolute;

    right: 35px; top: 10%;

    transform: translateY( -5% );

    color: #0c0c0c;

    background: none; border: none; font-size: 24px;

    cursor: pointer; z-index: 1;

    width: 45px; height: 50px; padding: 10px; border-radius: 5px;

    left: 727px; /\* Menyesuaikan posisi ke kanan sedikit \*/

}

URUTAN ISI FOLDER

Todo-listFaizah

/todo-list

│── index.php

│── db.php

│── connections.sql

│── /css

│ │── aksi.css

│ │── todolist.css

│── /js

│ │── script.js