

Documentazione

1. modifica_orario.php

Descrizione

Script PHP che consente ai docenti di visualizzare e modificare il proprio orario scolastico. Fornisce funzionalità per eliminare lezioni esistenti e aggiungere nuove voci all'orario. L'accesso è ristretto ai soli utenti con ruolo "docente".

Dipendenze

- `session_start()` : Gestione delle sessioni utente per l'autenticazione
- `login.php` : Reindirizzamento in caso di accesso non autorizzato
- Connessione al database (host, user, pass, dbname): Per interagire con i dati dell'orario
- `teacher_dashboard.php` : Link per tornare alla dashboard del docente
- `/uploads/back_white.png` : Immagine per il pulsante "Indietro"

Variabili

- `$_SESSION["user_id"]` : ID dell'utente loggato
- `$_SESSION["ruolo"]` : Ruolo dell'utente (deve essere "docente")
- `$host` , `$user` , `$pass` , `$dbname` : Dati di connessione al database
- `$conn` : Oggetto `mysqli` per la connessione al database
- `$docente_id` : ID del docente estratto dalla sessione
- `$_POST['delete_id']` : ID della lezione da eliminare (inviato via POST)
- `$_POST['classe_id']` , `$_POST['materia_id']` , `$_POST['giorno']` , `$_POST['ora_inizio']` : Dati per aggiungere una nuova lezione (inviati via POST)
- `$ora_fine` : Ora di fine lezione calcolata automaticamente (ora inizio + 1 ora)
- `$stmt` : Oggetto `mysqli_stmt` per le query preparate
- `$orario` : Array contenente i dati dell'orario del docente
- `$lezione` : Singolo elemento dell'array `$orario` durante l'iterazione

Funzioni principali

- `session_start()` : Avvia una nuova sessione o riprende quella esistente
- `header()` : Invia un header HTTP (usato per il reindirizzamento)
- `exit()` : Termina l'esecuzione dello script
- Query preparate: `$conn->prepare()` , `$stmt->bind_param()` , `$stmt->execute()`
- `date()` , `strtotime()` : Gestione delle date e orari
- `htmlspecialchars()` : Sanitizzazione dell'output

Logica del codice

1. **Autenticazione:** Verifica che l'utente sia loggato e abbia il ruolo di "docente"
2. **Connessione al database:** Stabilisce una connessione al database
3. **Gestione eliminazione lezione:** Elimina la lezione corrispondente all'ID ricevuto via POST
4. **Gestione aggiunta lezione:** Inserisce una nuova lezione con i dati ricevuti via POST
5. **Recupero orario:** Recupera tutte le lezioni del docente dal database
6. **Visualizzazione orario:** Genera una tabella HTML per visualizzare l'orario

Note sulla sicurezza

- Autenticazione e autorizzazione per limitare l'accesso ai soli docenti
- Utilizzo di query preparate per prevenire SQL injection
- Sanitizzazione dell'output con `htmlspecialchars()` per prevenire XSS

Codice:

```
<?php
session_start();
if (!isset($_SESSION["user_id"]) || $_SESSION["ruolo"] !== 'docente') {
    header("Location: login.php");
    exit();
}

$host = "localhost";
$user = "admin";
$pass = "Password";
$dbname = "registro_elettronico";
$conn = new mysqli($host, $user, $pass, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$docente_id = $_SESSION["user_id"];

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (isset($_POST['delete_id'])) {
        $delete_id = $_POST['delete_id'];
        $stmt = $conn->prepare("DELETE FROM orario WHERE id = ? AND docente_id = ?");
        $stmt->bind_param("ii", $delete_id, $docente_id);
        $stmt->execute();
        $stmt->close();
    } elseif (isset($_POST['classe_id'], $_POST['materia_id'], $_POST['giorno'],
$_POST['ora_inizio'])) {
        $classe_id = $_POST['classe_id'];
        $materia_id = $_POST['materia_id'];
        $giorno = $_POST['giorno'];
        $ora_inizio = $_POST['ora_inizio'];
```

```

        $ora_fine = date("H:i", strtotime($ora_inizio . " +1 hour"));

        $stmt = $conn->prepare("INSERT INTO orario (classe_id, docente_id, materia_id,
giorno, ora_inizio, ora_fine) VALUES (?, ?, ?, ?, ?, ?)");
        $stmt->bind_param("iiiiss", $classe_id, $docente_id, $materia_id, $giorno,
$ora_inizio, $ora_fine);
        $stmt->execute();
        $stmt->close();
    }
}

$query = "SELECT orario.id, classi.nome AS classe, materie.nome AS materia,
orario.giorno, orario.ora_inizio, orario.ora_fine FROM orario JOIN classi ON
orario.classe_id = classi.id JOIN materie ON orario.materia_id = materie.id WHERE
orario.docente_id = ? ORDER BY FIELD(orario.giorno, 'Luned\u00ec', 'Marted\u00ec',
'Mercoled\u00ec', 'Gioved\u00ec', 'Venerd\u00ec', 'Sabato'), orario.ora_inizio";
$stmt = $conn->prepare($query);
$stmt->bind_param("i", $docente_id);
$stmt->execute();
$result = $stmt->get_result();
$orario = $result->fetch_all(MYSQLI_ASSOC);
$stmt->close();
$conn->close();
?>

```

```

<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Modifica Orario</title>
    <style>
        body { background-color: #212121; color: white; font-family: monospace; text-align: center; }
        table { width: 80%; margin: 20px auto; border-collapse: collapse; }
        th, td { padding: 10px; border: 1px solid white; }
        th { background: rgba(255, 255, 255, 0.2); }
        .btn { padding: 8px; background: cyan; border: none; cursor: pointer; }
        .btn:hover { background: #00a2a2; }
        .back-button {
            position: absolute;
            top: 20px;
            left: 20px;
        }

        .back-button img {
            width: 32px;
            height: 32px;
            cursor: pointer;
        }
    </style>

```

```

</head>
<body>
    <a href="teacher_dashboard.php" class="back-button">
        
    </a>
    <h1>Modifica il tuo orario</h1>
    <table>
        <tr>
            <th>Classe</th>
            <th>Materia</th>
            <th>Giorno</th>
            <th>Ora Inizio</th>
            <th>Ora Fine</th>
            <th>Azioni</th>
        </tr>
        <?php foreach ($orario as $lezione): ?>
            <tr>
                <td><?= htmlspecialchars($lezione['classe']) ?></td>
                <td><?= htmlspecialchars($lezione['materia']) ?></td>
                <td><?= htmlspecialchars($lezione['giorno']) ?></td>
                <td><?= htmlspecialchars($lezione['ora_inizio']) ?></td>
                <td><?= htmlspecialchars($lezione['ora_fine']) ?></td>
                <td>
                    <form method="POST" style="display:inline;">
                        <input type="hidden" name="delete_id" value="<?=
$lezione['id'] ?>">
                        <button type="submit" class="btn">Elimina</button>
                    </form>
                </td>
            </tr>
        <?php endforeach; ?>
    </table>
    <a href="orario.php" class="btn">Aggiungi lezione</a>
</body>
</html>

```

2. student_dashboard.php

Descrizione

Dashboard principale per gli studenti. Fornisce un punto di accesso centralizzato per le funzionalità rilevanti per gli studenti, come la visualizzazione dell'orario, l'accesso alle classi e alle materie.

Dipendenze

- `session_start()` : Gestione delle sessioni utente per l'autenticazione
- `login.php` : Reindirizzamento in caso di accesso non autorizzato

- `classi.php`, `orario_studente.php`, `materie.php` : Link alle altre pagine dell'applicazione
- `logout.php` : Gestione del logout dell'utente

Variabili

- `$_SESSION["user_id"]` : ID dell'utente loggato
- `$_SESSION["nome"]` : Nome dello studente estratto dalla sessione

Funzioni principali

- `session_start()` : Avvia una nuova sessione o riprende quella esistente
- `header()` : Invia un header HTTP (usato per il reindirizzamento)
- `exit()` : Termina l'esecuzione dello script
- `htmlspecialchars()` : Sanitizzazione dell'output

Logica del codice

1. **Autenticazione:** Verifica che l'utente sia loggato
2. **Visualizzazione della dashboard:** Genera una pagina HTML con messaggi di benvenuto e link alle altre pagine

Note sulla sicurezza

- Autenticazione per garantire che solo gli studenti loggati possano accedere alla dashboard
- Sanitizzazione dell'output con `htmlspecialchars()` per prevenire XSS

Codice:

```
<?php
session_start();
if (!isset($_SESSION["user_id"])) {
    header("Location: login.php");
    exit();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dashboard</title>
    <style>
        *, ::after, ::before {
            box-sizing: border-box;
        }

        body {
```

```
background-color: #212121;
color: #fff;
font-family: monospace, serif;
letter-spacing: 0.05em;
margin: 0;
padding: 0;
display: flex;
flex-direction: column;
justify-content: center;
align-items: center;
height: 100vh;
text-align: center;
}

.navbar {
display: flex;
justify-content: space-between;
align-items: center;
background: rgba(255, 255, 255, 0.1);
padding: 15px 30px;
width: 100%;
position: absolute;
top: 0;
}

.navbar .links {
display: flex;
gap: 20px;
}

.navbar a {
color: white;
text-decoration: none;
font-size: 18px;
transition: 0.3s;
}

.navbar a:hover {
color: cyan;
}

.user-container {
display: flex;
align-items: center;
gap: 15px;
}

.username {
font-size: 18px;
color: cyan;
}
```

```

.logout-button {
    background-color: red;
    color: white;
    font-family: monospace;
    font-size: 16px;
    padding: 5px 10px;
    border: 2px solid white;
    cursor: pointer;
    text-transform: uppercase;
    transition: 0.3s;
}

.logout-button:hover {
    background-color: darkred;
    color: yellow;
}

.container {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}

h1 {
    font-size: 30px;
}
</style>
</head>
<body>
    <div class="navbar">
        <div class="links">
            <a href="http://localhost:8000/classi.php">Classi</a>
            <a href="http://localhost:8000/orario_studente.php">Orario</a>
            <a href="http://localhost:8000/materie.php">Materie</a>
        </div>
        <div class="user-container">
            <span class="username">
                <?php echo "Benvenuto/a, " . htmlspecialchars($_SESSION["nome"]) .
"!"; ?>
            </span>
            <a href="logout.php">
                <button class="logout-button">Logout</button>
            </a>
        </div>
    </div>
    <div class="container">
        <h1>Benvenuto, sei uno Studente!</h1>
    </div>

```

```
</body>
</html>
```

3. teacher_dashboard.php

Descrizione

Pannello di controllo per i docenti. Fornisce un'interfaccia centrale dove i docenti possono accedere alle funzionalità relative alla gestione delle classi, dell'orario e delle materie.

Dipendenze

- `session_start()` : Gestione delle sessioni utente per l'autenticazione
- `login.php` : Reindirizzamento in caso di accesso non autorizzato
- `config.php` : Include le impostazioni di connessione al database
- `classi.php` , `orario.php` , `materie.php` : Link alle pagine per la gestione di classi, orario e materie
- `logout.php` : Gestione del logout dell'utente
- Immagine del profilo del docente (o un'immagine predefinita)

Variabili

- `$_SESSION["user_id"]` : ID dell'utente docente autenticato
- `$docente_id` : ID del docente estratto dalla sessione
- `$query` : Query SQL per selezionare l'immagine del profilo
- `$stmt` : Oggetto `mysqli_stmt` per le query preparate
- `$immagine_profilo` : Nome del file dell'immagine del profilo
- `$immagine_profilo_url` : URL completo dell'immagine del profilo

Funzioni principali

- `session_start()` : Avvia una nuova sessione o riprende quella esistente
- `header()` : Invia un header HTTP (usato per il reindirizzamento)
- `require_once()` : Include ed evalua il file specificato
- Query preparate: `$conn->prepare()` , `$stmt->bind_param()` , `$stmt->execute()`
- `htmlspecialchars()` : Sanitizzazione dell'output

Logica del codice

1. **Autenticazione:** Verifica che l'utente sia autenticato
2. **Recupero dati del docente:** Ottiene l'immagine del profilo del docente
3. **Visualizzazione della dashboard:** Genera una pagina HTML con i link alle funzionalità

Note sulla sicurezza

- Autenticazione per garantire che solo i docenti loggati possano accedere
- Utilizzo di query preparate per prevenire SQL injection
- Sanitizzazione dell'output con `htmlspecialchars()` per prevenire XSS

Codice:

```
<?php
session_start();
if (!isset($_SESSION["user_id"])) {
    header("Location: login.php");
    exit();
}

require_once "config.php";

// Recupero i dettagli dell'utente (docente)
$docente_id = $_SESSION["user_id"];
$query = "SELECT immagine FROM utenti WHERE id = ?";
$stmt = $conn->prepare($query);
$stmt->bind_param("i", $docente_id);
$stmt->execute();
$stmt->bind_result($immagine_profilo);
$stmt->fetch();
$stmt->close();

$immagine_profilo_url = !empty($immagine_profilo) ? $immagine_profilo :
'default_profile_image.png'; // Usa un'immagine predefinita se non c'è
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dashboard</title>
    <style>
        *, ::after, ::before {
            box-sizing: border-box;
        }

        body {
            background-color: #212121;
            color: #fff;
            font-family: monospace, serif;
            letter-spacing: 0.05em;
            margin: 0;
            padding: 0;
            display: flex;
            flex-direction: column;
            justify-content: center;
```

```
    align-items: center;
    height: 100vh;
    text-align: center;
}

.navbar {
    display: flex;
    justify-content: space-between;
    align-items: center;
    background: rgba(255, 255, 255, 0.1);
    padding: 15px 30px;
    width: 100%;
    position: absolute;
    top: 0;
}

.navbar .links {
    display: flex;
    gap: 20px;
}

.navbar a {
    color: white;
    text-decoration: none;
    font-size: 18px;
    transition: 0.3s;
}

.navbar a:hover {
    color: cyan;
}

.user-container {
    display: flex;
    align-items: center;
    gap: 15px;
}

.username {
    font-size: 18px;
    color: cyan;
}

.logout-button {
    background-color: red;
    color: white;
    font-family: monospace;
    font-size: 16px;
    padding: 5px 10px;
    border: 2px solid white;
    cursor: pointer;
}
```

```
    text-transform: uppercase;
    transition: 0.3s;
}
```

```
.logout-button:hover {
    background-color: darkred;
    color: yellow;
}
```

```
.container {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    position: absolute;
}
```

```
h1 {
    font-size: 30px;
}
```

```
.profile-image {
    width: 100px;
    height: 100px;
    border-radius: 50%;
    object-fit: cover;
    margin-bottom: 20px;
    left: 50%;
    align-items: center;
    margin-top: -150px;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="navbar">
```

```
<div class="links">
```

```
<a href="http://localhost:8000/classi.php">Classi</a>
```

```
<a href="http://localhost:8000/orario.php">Orario</a>
```

```
<a href="http://localhost:8000/materie.php">Materie</a>
```

```
</div>
```

```
<div class="user-container">
```

```
<span class="username">
```

```
<?php echo "Sei un docente"?>
```

```
</span>
```

```
<a href="logout.php">
```

```
<button class="logout-button">Logout</button>
```

```
</a>
```

```
</div>
```

```
</div>
```

```


<div class="container">
    <h1><?php echo "Benvenuto/a, " . htmlspecialchars($_SESSION["nome"]) . "!"; ?>
</h1>
</div>
</body>
</html>
```

4. materie.php

Descrizione

Script PHP che permette di visualizzare l'elenco delle materie e dei docenti che le insegnano. Quando viene selezionata una materia, mostra i nomi, cognomi ed email dei docenti associati.

Dipendenze

- `session_start()` : Gestione delle sessioni utente per l'autenticazione
- `login.php` : Reindirizzamento in caso di accesso non autorizzato
- `config.php` : Include le impostazioni di connessione al database

Variabili

- `$_SESSION["user_id"]` : ID dell'utente autenticato
- `$materie_result` : Risultato della query per le materie
- `$docenti_materia` : Array con i dati dei docenti di una materia
- `$materia_selezionata` : ID della materia selezionata dall'utente
- `$stmt` : Oggetto `mysqli_stmt` per le query prepare
- `$result` : Risultato della query preparata
- `$row` : Singola riga del risultato

Funzioni principali

- `session_start()` : Avvia una nuova sessione o riprende quella esistente
- `$conn->query()` : Esegue una query SQL
- Query prepare: `$conn->prepare()`, `$stmt->bind_param()`, `$stmt->execute()`
- `htmlspecialchars()` : Sanitizzazione dell'output

Logica del codice

1. **Autenticazione:** Verifica che l'utente sia autenticato
2. **Recupero elenco materie:** Ottiene tutte le materie dal database

3. **Gestione selezione materia:** Recupera i docenti per la materia selezionata
4. **Visualizzazione dati:** Genera una pagina HTML con form e tabella risultati

Note sulla sicurezza

- Autenticazione per garantire che solo gli utenti loggati possano accedere
- Utilizzo di query preparate per prevenire SQL injection
- Sanitizzazione dell'output con `htmlspecialchars()` per prevenire XSS

Codice

```
<?php
session_start();
if (!isset($_SESSION["user_id"])) {
    header("Location: login.php");
    exit();
}

require_once "config.php";

// Recupera tutte le materie
$materie_result = $conn->query("SELECT id, nome FROM materie");

$docenti_materia = [];
$materia_selezionata = '';

if ($_SERVER["REQUEST_METHOD"] == "POST" && !empty($_POST["materia_id"])) {
    $materia_selezionata = $_POST["materia_id"];

    $stmt = $conn->prepare("SELECT DISTINCT u.nome, u.cognome, u.email
                           FROM orario o
                           JOIN utenti u ON o.docente_id = u.id
                           WHERE o.materia_id = ?");
    $stmt->bind_param("i", $materia_selezionata);
    $stmt->execute();
    $result = $stmt->get_result();

    while ($row = $result->fetch_assoc()) {
        $docenti_materia[] = $row;
    }
}

?>
<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <title>Docenti per Materia</title>
    <style>
        body {
            background-color: #212121;
```



```

        <option value="">Scegli una materia</option>
        <?php while ($row = $materie_result->fetch_assoc()): ?>
            <option value="<?php echo $row['id']; ?>" <?php if
($materia_selezionata == $row['id']) echo 'selected'; ?>
                <?php echo htmlspecialchars($row['nome']); ?>
            </option>
        <?php endwhile; ?>
    </select>
    <button type="submit">Visualizza</button>
</form>

<?php if (!empty($docenti_materia)): ?>
    <table>
        <thead>
            <tr>
                <th>Nome</th>
                <th>Cognome</th>
                <th>Email</th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($docenti_materia as $docente): ?>
                <tr>
                    <td class="highlight"><?php echo
htmlspecialchars($docente['nome']); ?></td>
                    <td class="highlight"><?php echo
htmlspecialchars($docente['cognome']); ?></td>
                    <td><?php echo htmlspecialchars($docente['email']); ?>
                </td>
            </tr>
            <?php endforeach; ?>
        </tbody>
    </table>
    <?php elseif ($_SERVER["REQUEST_METHOD"] == "POST"): ?>
        <p>Nessun docente attualmente insegna questa materia.</p>
    <?php endif; ?>
</div>
</body>
</html>

```

5. admin_dashboard.php

Descrizione

Dashboard dell'amministratore. Fornisce funzionalità per la gestione degli utenti (docenti e studenti), delle classi e delle materie. Solo gli utenti con ruolo "admin" possono accedere.

Dipendenze

- `session_start()` : Gestione delle sessioni utente per l'autenticazione
- `login.php` : Reindirizzamento in caso di accesso non autorizzato
- `config.php` : Include le impostazioni di connessione al database

Variabili

- `$_SESSION["user_id"]` : ID dell'amministratore autenticato
- `$_SESSION["ruolo"]` : Ruolo dell'utente (deve essere "admin")
- `$stmt` : Oggetto `mysqli_stmt` per le query preparate
- `$docenti`, `$studenti`, `$classi`, `$materie` : Risultati delle query

Funzioni principali

- `session_start()` : Avvia una nuova sessione o riprende quella esistente
- `require_once()` : Include ed evalua il file specificato
- `$conn->query()`, `$conn->prepare()` : Eseguono query SQL
- `htmlspecialchars()` : Sanitizzazione dell'output

Logica del codice

1. **Autenticazione:** Verifica che l'utente sia admin
2. **Gestione eliminazioni:** Elimina utenti, studenti, classi o materie
3. **Gestione aggiunte:** Aggiunge nuovi utenti, studenti, classi o materie
4. **Recupero dati:** Ottiene tutti i docenti, studenti, classi e materie
5. **Visualizzazione dashboard:** Genera una pagina HTML con form e tabelle

Note sulla sicurezza

- Autenticazione e autorizzazione per limitare l'accesso agli amministratori
- Utilizzo di query preparate per prevenire SQL injection
- Sanitizzazione dell'output con `htmlspecialchars()` per prevenire XSS

Codice

```
<?php
session_start();
if (!isset($_SESSION["user_id"]) || $_SESSION["ruolo"] !== 'admin') {
    header("Location: login.php");
    exit();
}

require_once "config.php";

// Gestione eliminazioni\
if (isset($_POST['delete_user'])) {
    $stmt = $conn->prepare("DELETE FROM utenti WHERE id = ? AND ruolo = 'docente'");
```



```

$stmt->bind_param("i", $_POST['delete_user']);
$stmt->execute();
}
if (isset($_POST['delete_studente'])) {
    $stmt = $conn->prepare("DELETE FROM utenti WHERE id = ? AND ruolo = 'studente'");
    $stmt->bind_param("i", $_POST['delete_studente']);
    $stmt->execute();
}
if (isset($_POST['delete_classe'])) {
    $stmt = $conn->prepare("DELETE FROM classi WHERE id = ?");
    $stmt->bind_param("i", $_POST['delete_classe']);
    $stmt->execute();
}
if (isset($_POST['delete_materia'])) {
    $stmt = $conn->prepare("DELETE FROM materie WHERE id = ?");
    $stmt->bind_param("i", $_POST['delete_materia']);
    $stmt->execute();
}

// Gestione aggiunte
if (isset($_POST['add_user'])) {
    $stmt = $conn->prepare("INSERT INTO utenti (nome, cognome, email, password_hash, ruolo) VALUES (?, ?, ?, ?, 'docente')");
    $stmt->bind_param("ssss", $_POST['nome'], $_POST['cognome'], $_POST['email'], $_POST['password']);
    $stmt->execute();
}
if (isset($_POST['add_studente'])) {
    $stmt = $conn->prepare("INSERT INTO utenti (nome, cognome, email, password_hash, ruolo) VALUES (?, ?, ?, ?, 'studente')");
    $stmt->bind_param("ssss", $_POST['nome_studente'], $_POST['cognome_studente'], $_POST['email_studente'], $_POST['password_studente']);
    $stmt->execute();
}
if (isset($_POST['add_classe'])) {
    $stmt = $conn->prepare("INSERT INTO classi (nome) VALUES (?)");
    $stmt->bind_param("s", $_POST['classe']);
    $stmt->execute();
}
if (isset($_POST['add_materia'])) {
    $stmt = $conn->prepare("INSERT INTO materie (nome) VALUES (?)");
    $stmt->bind_param("s", $_POST['materia']);
    $stmt->execute();
}

$docenti = $conn->query("SELECT * FROM utenti WHERE ruolo = 'docente'");
$studenti = $conn->query("SELECT * FROM utenti WHERE ruolo = 'studente'");
$classi = $conn->query("SELECT * FROM classi");
$materie = $conn->query("SELECT * FROM materie");
?>
<!DOCTYPE html>

```

```
<html lang="it">
<head>
  <meta charset="UTF-8">
  <title>Admin Dashboard</title>
  <style>
    body {
      background: #1e1e1e;
      color: #fff;
      font-family: monospace;
      padding: 20px;
    }
    h2 {
      margin-top: 40px;
      color: cyan;
    }
    form {
      margin-bottom: 20px;
    }
    input, button {
      padding: 5px 10px;
      font-family: monospace;
      margin: 5px;
    }
    table {
      width: 100%;
      border-collapse: collapse;
      background-color: #333;
      margin-top: 10px;
    }
    th, td {
      border: 1px solid #555;
      padding: 10px;
    }
    th {
      background-color: #444;
    }
    .delete-btn {
      background-color: red;
      color: white;
      border: none;
      padding: 5px 10px;
      cursor: pointer;
    }

    .logout-button {
      background-color: red;
      color: white;
      font-family: monospace;
      font-size: 16px;
      padding: 5px 10px;
      border: 2px solid white;
```

```

    cursor: pointer;
    text-transform: uppercase;
    transition: 0.3s;
    position: absolute; /* Posizionamento assoluto */
    top: 20px; /* Distanza dall'alto */
    right: 20px; /* Distanza da destra */
}

.logout-button:hover {
    background-color: darkred;
    color: yellow;
}

</style>
</head>
<body>

<a href="logout.php">
    <button class="logout-button">Logout</button>
</a>

<h1>Benvenuto nella Dashboard Admin</h1>

<h2>Aggiungi Docente</h2>
<form method="POST">
    <input type="text" name="nome" placeholder="Nome" required>
    <input type="text" name="cognome" placeholder="Cognome" required>
    <input type="email" name="email" placeholder="Email" required>
    <input type="text" name="password" placeholder="Password" required>
    <button type="submit" name="add_user">Aggiungi</button>
</form>

<table>
    <tr><th>Nome</th><th>Cognome</th><th>Email</th><th>Elimina</th></tr>
    <?php while ($row = $docenti->fetch_assoc()): ?>
    <tr>
        <td><?= htmlspecialchars($row['nome']) ?></td>
        <td><?= htmlspecialchars($row['cognome']) ?></td>
        <td><?= htmlspecialchars($row['email']) ?></td>
        <td>
            <form method="POST" style="display:inline;">
                <input type="hidden" name="delete_user" value="<?= $row['id'] ?>">
                <button class="delete-btn" type="submit">Elimina</button>
            </form>
        </td>
    </tr>
    <?php endwhile; ?>
</table>

<h2>Aggiungi Studente</h2>
<form method="POST">
    <input type="text" name="nome_studente" placeholder="Nome" required>

```

```

<input type="text" name="cognome_studente" placeholder="Cognome" required>
<input type="email" name="email_studente" placeholder="Email" required>
<input type="text" name="password_studente" placeholder="Password" required>
<button type="submit" name="add_studente">Aggiungi</button>
</form>

<table>
  <tr><th>Nome</th><th>Cognome</th><th>Email</th><th>Elimina</th></tr>
  <?php while ($row = $studenti->fetch_assoc()): ?>
    <tr>
      <td><?= htmlspecialchars($row['nome']) ?></td>
      <td><?= htmlspecialchars($row['cognome']) ?></td>
      <td><?= htmlspecialchars($row['email']) ?></td>
      <td>
        <form method="POST" style="display:inline;">
          <input type="hidden" name="delete_studente" value="<?= $row['id']
?>">
          <button class="delete-btn" type="submit">Elimina</button>
        </form>
      </td>
    </tr>
  <?php endwhile; ?>
</table>

<h2>Aggiungi Classe</h2>
<form method="POST">
  <input type="text" name="classe" placeholder="Nome classe" required>
  <button type="submit" name="add_classe">Aggiungi</button>
</form>
<table>
  <tr><th>Nome</th><th>Elimina</th></tr>
  <?php while ($row = $classi->fetch_assoc()): ?>
    <tr>
      <td><?= htmlspecialchars($row['nome']) ?></td>
      <td>
        <form method="POST" style="display:inline;">
          <input type="hidden" name="delete_classe" value="<?= $row['id'] ?
>">
          <button class="delete-btn" type="submit">Elimina</button>
        </form>
      </td>
    </tr>
  <?php endwhile; ?>
</table>

<h2>Aggiungi Materia</h2>
<form method="POST">
  <input type="text" name="materia" placeholder="Nome materia" required>
  <button type="submit" name="add_materia">Aggiungi</button>
</form>
<table>

```

```

<tr><th>Nome</th><th>Elimina</th></tr>
<?php while ($row = $materie->fetch_assoc()): ?>
<tr>
    <td><?= htmlspecialchars($row['nome']) ?></td>
    <td>
        <form method="POST" style="display:inline;">
            <input type="hidden" name="delete_materia" value="<?= $row['id'] ?
>">
            <button class="delete-btn" type="submit">Elimina</button>
        </form>
    </td>
</tr>
<?php endwhile; ?>
</table>
</body>
</html>

```

6. orario_studente.php

Descrizione

Script PHP che consente agli studenti di visualizzare il loro orario scolastico. Gli studenti possono selezionare un docente e una classe per visualizzare l'orario specifico.

Dipendenze

- `session_start()` : Gestione delle sessioni utente per l'autenticazione
- `login.php` : Reindirizzamento in caso di accesso non autorizzato
- `config.php` : Include le impostazioni di connessione al database

Variabili

- `$_SESSION["user_id"]` : ID dello studente autenticato
- `$docenti_result` : Elenco di tutti i docenti
- `$classi_result` : Elenco di tutte le classi
- `$orario` : Array con i dati dell'orario
- `$email_selezionata` : Email del docente selezionato
- `$classe_selezionata` : ID della classe selezionata
- `$stmt` : Oggetto per le query preparate
- `$fasce_orarie` : Array con le fasce orarie
- `$giorni` : Array con i giorni della settimana

Funzioni principali

- `session_start()` : Avvia una nuova sessione o riprende quella esistente

- `$conn->query()` : Esegue una query SQL
- Query prepare: `$conn->prepare()` , `$stmt->bind_param()` , `$stmt->execute()`
- `date()` , `strtotime()` : Gestione delle date e orari
- `htmlspecialchars()` : Sanitizzazione dell'output

Logica del codice

1. **Autenticazione:** Verifica che l'utente sia autenticato
2. **Recupero dati:** Ottiene elenchi di docenti e classi
3. **Gestione selezione:** Recupera l'orario per docente e classe selezionati
4. **Formattazione dati:** Organizza i dati dell'orario in un array
5. **Visualizzazione:** Genera una pagina HTML con form e tabella orario

Note sulla sicurezza

- Autenticazione per garantire che solo gli studenti loggati possano accedere
- Utilizzo di query prepare per prevenire SQL injection
- Sanitizzazione dell'output con `htmlspecialchars()` per prevenire XSS

Codice

```
<?php
session_start();
if (!isset($_SESSION["user_id"])) {
    header("Location: login.php");
    exit();
}

require_once "config.php";

// Fetch all docenti
$docenti_result = $conn->query("SELECT id, email FROM utenti WHERE ruolo = 'docente'");

// Fetch all classi
$classi_result = $conn->query("SELECT id, nome FROM classi");

$orario = [];
$email_selezionata = '';
$classe_selezionata = '';

if ($_SERVER["REQUEST_METHOD"] == "POST" && !empty($_POST["docente_email"]) && !empty($_POST["classe_id"])) {
    $email_selezionata = $_POST["docente_email"];
    $classe_selezionata = $_POST["classe_id"];

    $stmt = $conn->prepare("SELECT o.giorno, o.ora_inizio, o.ora_fine, m.nome AS materia
```

```

        FROM orario o
        JOIN utenti u ON o.docente_id = u.id
        JOIN materie m ON o.materia_id = m.id
        WHERE u.email = ? AND o.classe_id = ?");
$stmt->bind_param("si", $email_selezionata, $classe_selezionata);
$stmt->execute();
$result = $stmt->get_result();

while ($row = $result->fetch_assoc()) {
    $ora = date("H:i", strtotime($row['ora_inizio']));
    $orario[$ora][$row['giorno']] = $row['materia'];
}
}

$fasce_orarie = [
    "07:50" => "7.50 - 8.50",
    "08:50" => "8.50 - 9.50",
    "09:50" => "9.50 - 10.50",
    "10:50" => "10.50 - 11.50",
    "11:50" => "11.50 - 12.50",
    "12:50" => "12.50 - 13.50"
];

$giorni = ['Lunedì', 'Martedì', 'Mercoledì', 'Giovedì', 'Venerdì'];
?>
<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Orario Docente</title>
    <style>
        body {
            background-color: #212121;
            color: #fff;
            font-family: monospace, serif;
            padding: 20px;
        }
        h2, label, select, button {
            font-size: 18px;
        }
        table {
            width: 100%;
            border-collapse: collapse;
            margin-top: 20px;
            background-color: #333;
        }
        th, td {
            border: 1px solid #555;
            padding: 10px;
            text-align: center;

```

```

        font-size: 16px;
    }
    th {
        background-color: #444;
    }
    .materia-cell {
        font-weight: bold;
        color: #00e5ff;
        font-size: 17px;
    }

    .back-button {
        position: absolute;
        top: 20px;
        left: 20px;
    }

    .back-button img {
        width: 32px;
        height: 32px;
        cursor: pointer;
    }

    .centro-contenuto {
        text-align: center;
        padding: 20px;
    }
</style>
</head>
<body>
    <a href="student_dashboard.php" class="back-button">
        
    </a>
    <div class="centro-contenuto">
    <h2>Visualizza Orario del Docente</h2>
    <form method="POST">
        <label for="docente_email">Seleziona docente:</label>
        <select name="docente_email" id="docente_email" required>
            <option value="">Scegli un docente</option>
            <?php while ($row = $docenti_result->fetch_assoc()): ?>
                <option value="<?php echo htmlspecialchars($row['email']); ?>" <?php
if ($email_selezionata == $row['email']) echo 'selected'; ?>>
                    <?php echo htmlspecialchars($row['email']); ?>
                </option>
            <?php endwhile; ?>
        </select>

        <label for="classe_id">Seleziona classe:</label>
        <select name="classe_id" id="classe_id" required>
            <option value="">Scegli una classe</option>
            <?php while ($row = $classi_result->fetch_assoc()): ?>

```



```

        <option value="<?php echo $row['id']; ?>" <?php if
($classe_selezionata == $row['id']) echo 'selected'; ?> >
        <?php echo htmlspecialchars($row['nome']); ?>
    </option>
    <?php endwhile; ?>
</select>

    <button type="submit">Visualizza</button>
</form>

<?php if (!empty($orario)): ?>
    <table>
        <thead>
            <tr>
                <th>Ora</th>
                <?php foreach ($giorni as $giorno): ?>
                    <th><?php echo $giorno; ?></th>
                <?php endforeach; ?>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($fasce_orarie as $ora_inizio => $fascia): ?>
                <tr>
                    <td><?php echo $fascia; ?></td>
                    <?php foreach ($giorni as $giorno): ?>
                        <td>
                            <?php echo isset($orario[$ora_inizio][$giorno]) ?
'<span class="materia-cell">' . htmlspecialchars($orario[$ora_inizio][$giorno]) .
'</span>' : "-"; ?>
                        </td>
                    <?php endforeach; ?>
                </tr>
            <?php endforeach; ?>
        </tbody>
    </table>
    <?php elseif ($_SERVER["REQUEST_METHOD"] == "POST"): ?>
        <p>Nessun orario disponibile per il docente e classe selezionati.</p>
    <?php endif; ?>
</div>
</body>
</html>

```

7. orario.php

Descrizione

Script PHP che consente ai docenti di visualizzare il proprio orario e di aggiungere nuove lezioni. L'accesso è ristretto ai soli utenti con ruolo "docente".

Dipendenze

- `session_start()` : Gestione delle sessioni utente per l'autenticazione
- `login.php` : Reindirizzamento in caso di accesso non autorizzato
- `config.php` (implicito): Connessione al database
- `modifica_orario.php` : Link per modificare l'orario

Variabili

- `$_SESSION["user_id"]` : ID del docente autenticato
- `$_SESSION["ruolo"]` : Ruolo dell'utente (deve essere "docente")
- `$host`, `$user`, `$pass`, `$dbname` : Dati di connessione al database
- `$conn` : Oggetto per la connessione al database
- `$docente_id` : ID del docente dalla sessione
- `$orari` : Array con i dati dell'orario del docente
- `$classi`, `$materie` : Array con dati di classi e materie
- Form POST: `$_POST["classe"]`, `$_POST["materia"]`, `$_POST["giorno"]`, `$_POST["ora_inizio"]`
- `$ora_fine` : Calcolata automaticamente (ora inizio + 1)

Funzioni principali

- `session_start()` : Avvia o riprende una sessione PHP
- `new mysqli()` : Crea una nuova connessione al database
- Query prepare e standard: `$conn->prepare()`, `$conn->query()`
- `date()`, `strtotime()` : Gestione di date e orari
- `htmlspecialchars()` : Sanitizzazione dell'output

Logica del codice

1. **Autenticazione:** Verifica che l'utente sia docente
2. **Connessione al database:** Stabilisce una connessione
3. **Recupero orario:** Ottiene l'orario del docente
4. **Recupero classi e materie:** Ottiene elenchi dal database
5. **Gestione aggiunta lezione:** Inserisce nuove lezioni se richiesto
6. **Visualizzazione dati:** Genera una pagina HTML con orario e form

Note sulla sicurezza

- Autenticazione e autorizzazione per limitare l'accesso ai docenti
- Utilizzo di query prepare per prevenire SQL injection
- Sanitizzazione dell'output con `htmlspecialchars()` per prevenire XSS

Codice

```

<?php
session_start();
if (!isset($_SESSION["user_id"]) || $_SESSION["ruolo"] !== 'docente') {
    header("Location: login.php");
    exit();
}

$host = "localhost";
$user = "admin";
$pass = "Password";
$dbname = "registro_elettronico";
$conn = new mysqli($host, $user, $pass, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$docente_id = $_SESSION["user_id"];
$orari = [];
$query = "SELECT orario.id, classi.nome AS classe, materie.nome AS materia,
orario.giorno, orario.ora_inizio, orario.ora_fine
        FROM orario
        JOIN classi ON orario.classe_id = classi.id
        JOIN materie ON orario.materia_id = materie.id
        WHERE orario.docente_id = ?";
$stmt = $conn->prepare($query);
$stmt->bind_param("i", $docente_id);
$stmt->execute();
$result = $stmt->get_result();
while ($row = $result->fetch_assoc()) {
    $orari[] = $row;
}
$stmt->close();

// Recupero classi e materie disponibili
$classi = $conn->query("SELECT * FROM classi")->fetch_all(MYSQLI_ASSOC);
$materie = $conn->query("SELECT * FROM materie")->fetch_all(MYSQLI_ASSOC);

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $classe_id = $_POST["classe"];
    $materia_id = $_POST["materia"];
    $giorno = $_POST["giorno"];
    $ora_inizio = $_POST["ora_inizio"];
    $ora_fine = date("H:i", strtotime($ora_inizio) + 3600);

    $stmt = $conn->prepare("INSERT INTO orario (docente_id, classe_id, materia_id,
giorno, ora_inizio, ora_fine) VALUES (?, ?, ?, ?, ?, ?)");
    $stmt->bind_param("iiisss", $docente_id, $classe_id, $materia_id, $giorno,
$ora_inizio, $ora_fine);
    $stmt->execute();
    $stmt->close();
    header("Location: orario.php");
}

```

```
    exit();
}
$conn->close();
?>

<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <title>Orario Docente</title>
    <style>
        body {
            background-color: #212121;
            color: #fff;
            font-family: monospace, serif;
            text-align: center;
        }

        table {
            width: 80%;
            margin: 20px auto;
            border-collapse: collapse;
        }

        th, td {
            padding: 10px;
            border: 1px solid cyan;
        }

        .btn {
            padding: 10px;
            background: cyan;
            color: #000;
            text-decoration: none;
        }

        h1 {
            font-size: 32px;
        }

        h2 {
            font-size: 28px;
        }

        .back-button {
            position: absolute;
            top: 20px;
            left: 20px;
        }

        .back-button img {
```

```

        width: 32px;
        height: 32px;
        cursor: pointer;
    }
</style>
</head>
<body>
    <a href="teacher_dashboard.php" class="back-button">
        
    </a>
    <h1>Il tuo Orario</h1>
    <table>
        <tr><th>Classe</th><th>Materia</th><th>Giorno</th><th>Ora Inizio</th><th>Ora
Fine</th></tr>
        <?php foreach ($orari as $lezione): ?>
            <tr>
                <td><?= htmlspecialchars($lezione["classe"]) ?></td>
                <td><?= htmlspecialchars($lezione["materia"]) ?></td>
                <td><?= htmlspecialchars($lezione["giorno"]) ?></td>
                <td><?= htmlspecialchars($lezione["ora_inizio"]) ?></td>
                <td><?= htmlspecialchars($lezione["ora_fine"]) ?></td>
            </tr>
        <?php endforeach; ?>
    </table>
    <h2>Aggiungi una Lezione</h2>
    <form method="post">
        <select name="classe">
            <?php foreach ($classi as $classe) { echo "<option
value='{$classe['id']}'>{$classe['nome']}</option>"; } ?>
        </select>
        <select name="materia">
            <?php foreach ($materie as $materia) { echo "<option
value='{$materia['id']}'>{$materia['nome']}</option>"; } ?>
        </select>
        <select name="giorno">
            <option value="Lunedì">Lunedì</option>
            <option value="Martedì">Martedì</option>
            <option value="Mercoledì">Mercoledì</option>
            <option value="Giovedì">Giovedì</option>
            <option value="Venerdì">Venerdì</option>
            <option value="Sabato">Sabato</option>
        </select>
        <input type="time" name="ora_inizio" min="07:50" max="13:50" required>
        <button type="submit" class="btn">Aggiungi</button>
    </form>
    <br>
    <a href="modifica_orario.php" class="btn">Modifica Orario</a>
</body>
</html>

```

8. classi.php

Descrizione

Script PHP che visualizza l'elenco delle classi presenti nel sistema. L'accesso è consentito solo agli utenti autenticati.

Dipendenze

- `session_start()` : Gestione delle sessioni utente per l'autenticazione
- `login.php` : Reindirizzamento in caso di accesso non autorizzato
- `config.php` (implicito): Connessione al database
- `teacher_dashboard.php` : Link per tornare alla dashboard
- `/uploads/back_white.png` : Immagine per il pulsante "Indietro"

Variabili

- `$_SESSION["user_id"]` : ID dell'utente autenticato
- `$host`, `$user`, `$pass`, `$dbname` : Dati di connessione al database
- `$conn` : Oggetto per la connessione al database
- `$sql` : Query SQL per selezionare le classi
- `$result` : Risultato della query
- `$row` : Singola riga del risultato

Funzioni principali

- `session_start()` : Avvia o riprende una sessione PHP
- `new mysqli()` : Crea una nuova connessione al database
- `$conn->query()` : Esegue una query SQL
- `htmlspecialchars()` : Sanitizzazione dell'output

Logica del codice

1. **Autenticazione:** Verifica che l'utente sia autenticato
2. **Connessione al database:** Stabilisce una connessione
3. **Recupero classi:** Ottiene tutte le classi dal database
4. **Visualizzazione dati:** Genera una pagina HTML con l'elenco classi

Note sulla sicurezza

- Autenticazione per garantire che solo gli utenti loggati possano accedere
- Sanitizzazione dell'output con `htmlspecialchars()` per prevenire XSS

Codice

```

<?php
session_start();

if (!isset($_SESSION["user_id"])) {
    header("Location: login.php");
    exit();
}

$host = "localhost";
$user = "admin";
$pass = "Password";
$dbname = "registro_elettronico";

$conn = new mysqli($host, $user, $pass, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT id, nome FROM classi";
$result = $conn->query($sql);
?>

<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Elenco Classi</title>
    <style>
        body {
            background-color: #212121;
            color: #fff;
            font-family: monospace, serif;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
            flex-direction: column;
        }

        .back-button {
            position: absolute;
            top: 20px;
            left: 20px;
        }

        .back-button img {
            width: 32px;
            height: 32px;
        }
    </style>

```

```

        cursor: pointer;
    }

    .table-container {
        text-align: center;
        background: rgba(255, 255, 255, 0.1);
        padding: 20px;
        border-radius: 10px;
        box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.2);
    }

    table {
        width: 100%;
        border-collapse: collapse;
        margin-top: 10px;
    }

    th, td {
        padding: 10px;
        text-align: left;
        border-bottom: 1px solid #fff;
    }

    th {
        background-color: rgba(255, 255, 255, 0.2);
    }
</style>
</head>
<body>
    <a href="teacher_dashboard.php" class="back-button">
        
    </a>

    <div class="table-container">
        <h2>Elenco delle Classi</h2>
        <table>
            <tr>
                <th>ID</th>
                <th>Classe</th>
            </tr>
            <?php while ($row = $result->fetch_assoc()): ?>
                <tr>
                    <td><?php echo htmlspecialchars($row["id"]); ?></td>
                    <td><?php echo htmlspecialchars($row["nome"]); ?></td>
                </tr>
            <?php endwhile; ?>
        </table>
    </div>
</body>
</html>

```



```
<?php $conn->close(); ?>
```

9. index.php

Descrizione

Script PHP che gestisce il login degli utenti. Gli utenti inseriscono email e password, lo script verifica le credenziali nel database e reindirizza alla dashboard appropriata (studente o docente).

Dipendenze

- `session_start()` : Gestione delle sessioni utente
- `config.php` (implicito): Connessione al database
- `student_dashboard.php` : Reindirizzamento per gli studenti
- `teacher_dashboard.php` : Reindirizzamento per i docenti

Variabili

- `$host`, `$user`, `$pass`, `$dbname` : Dati di connessione al database
- `$conn` : Oggetto per la connessione al database
- `$_POST["email"]`, `$_POST["password"]` : Credenziali inserite dall'utente
- `$stmt` : Oggetto per le query preparate

Funzioni principali

- `session_start()` : Avvia o riprende una sessione PHP
- `new mysqli()` : Crea una nuova connessione al database
- Query preparate: `$conn->prepare()`, `$stmt->bind_param()`, `$stmt->execute()`
- `password_verify()` : Verifica la password inserita con l'hash nel database

Logica del codice

1. **Connessione al database:** Stabilisce una connessione
2. **Verifica credenziali:** Controlla email e password
3. **Gestione sessione:** Se le credenziali sono valide, crea la sessione
4. **Reindirizzamento:** In base al ruolo, reindirizza alla dashboard appropriata

Note sulla sicurezza

- Utilizzo di query preparate per prevenire SQL injection
- Verifica delle password con `password_verify()` per confrontare in modo sicuro
- Gestione delle sessioni per mantenere l'autenticazione degli utenti

Codice

```
<?php
session_start();
$host = "localhost";
$user = "admin";
$pass = "Password";
$dbname = "registro_elettronico";

// Connessione a MariaDB
$conn = new mysqli($host, $user, $pass, $dbname);

// Controllo connessione
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Gestione login
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = $_POST["email"];
    $password = $_POST["password"];

    // Recupera i dati dell'utente
    $stmt = $conn->prepare("SELECT id, nome, password_hash, ruolo FROM utenti WHERE
email = ?");
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $stmt->store_result();

    if ($stmt->num_rows > 0) {
        $stmt->bind_result($id, $nome, $stored_password, $ruolo);
        $stmt->fetch();

        // Confronto diretto della password (senza hash)
        if ($password === $stored_password) {
            // Memorizza i dati dell'utente nella sessione
            $_SESSION["user_id"] = $id;
            $_SESSION["nome"] = $nome;
            $_SESSION["email"] = $email;
            $_SESSION["ruolo"] = $ruolo;

            // Reindirizza in base al ruolo
            if ($ruolo === 'studente') {
                header("Location: student_dashboard.php");
            } else if ($ruolo === 'docente') {
                header("Location: teacher_dashboard.php");
            } else if ($ruolo === 'admin') {
                header("Location: admin_dashboard.php");
            }
            exit();
        } else {
```

```

        $error = "Password errata.";
    }
} else {
    $error = "Utente non trovato.";
}
$stmt->close();
}
$conn->close();
?>

<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <style>
        body {
            background-color: #212121;
            color: #fff;
            font-family: monospace, serif;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
        }
        .form-container {
            text-align: center;
            background: rgba(255, 255, 255, 0.1);
            padding: 60px;
            border-radius: 10px;
            box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.2);
        }
        .form-container h1 {
            font-size: 23px;
            margin-bottom: 20px;
        }
        .control {
            margin-bottom: 24px;
        }
        .control input {
            width: 100%;
            padding: 14px 16px;
            border: 0;
            background: transparent;
            color: #fff;
            font-size: 16px;
            border-bottom: 2px solid #fff;
            text-align: center;
        }
    </style>

```

```

        .control input:focus {
            outline: none;
            border-bottom: 2px solid cyan;
        }
        .btn {
            width: 100%;
            padding: 14px 16px;
            background: cyan;
            border: none;
            color: #000;
            font-weight: bold;
            font-size: 16px;
            cursor: pointer;
            border-radius: 5px;
            transition: 0.3s;
        }
        .btn:hover {
            background: #00a2a2;
        }
        .error {
            color: red;
            margin-top: 10px;
        }
    }
</style>
</head>
<body>
    <div class="form-container">
        
        <h1>Login</h1>
        <form method="POST" action="">
            <div class="control">
                <input type="email" name="email" placeholder="Email" required>
            </div>
            <div class="control">
                <input type="password" name="password" placeholder="Password"
required>
            </div>
            <button class="btn" type="submit">Accedi</button>
        </form>
        <?php if (isset($error)) echo "<p class='error'>$error</p>"; ?>
    </div>
</body>
</html>

```