

VERIFICA DI INFORMATICA - PHP + MySQL

Esercizio 1 – Database e CRUD (20 punti)

Utilizza phpMyAdmin per creare un nuovo database chiamato TUOCOGNOME_azienda. All'interno di questo database crea due tabelle:

- Tabella **ruoli** composta dai campi: id, nome_ruolo
- Tabella **dipendenti** composta da campi: id, nome, cognome, ruolo_id, data_assunzione

Dopo aver creato le tabelle, effettua l'esportazione del database scegliendo l'opzione "Personalizzato", assicurandoti di selezionare CREATE DATABASE IF NOT EXISTS. Rinomina il file SQL TUOCOGNOME_db_azienda.sql e inseriscilo nella cartella esercizio1.

La tabella ruoli dovrà essere popolata con 4 ruoli predefiniti: Sviluppatore, Project Manager, Designer, Analista. La tabella una volta creata non dovrà più essere modificata.

Creare nel file "esercizio1.php" una pagina che contenga 4 form (come in figura).

Ognuno di questi utilizzerà il metodo post per svolgere una delle seguenti azioni:

- Inserimento dati (insert.php)
- Cancellazione dati usando come discriminante l'id (delete.php)
- Aggiornamento dati (update.php)
- Lettura di tutti i dati (select-all.php)
- Lettura dati filtrando per ruolo e in ordine per cognome (select-role.php)

Insert

Name	<input type="text"/>
Surname	<input type="text"/>
Ruolo	<input type="text" value="Seleziona ruolo"/>
Data assunzione	<input type="text" value="gg/mm/aaaa"/>
<input type="button" value="Insert"/>	

Delete

id	<input type="text"/>
<input type="button" value="Delete"/>	

Update

id	<input type="text"/>
Name	<input type="text"/>
Surname	<input type="text"/>
Ruolo	<input type="text" value="— nessuna modifica —"/>
Data assunzione	<input type="text" value="gg/mm/aaaa"/>
<input type="button" value="Edit"/>	

Select *

<input type="button" value="Select"/>

Select by Ruolo

Ruolo	<input type="text" value="Seleziona ruolo"/>
<input type="button" value="Select by Ruolo"/>	

Tutti i form, dopo aver completato le operazioni sul database, devono reindirizzare alla pagina esercizio1.php

I form di tipo **select**, invece, devono visualizzare i risultati della query in una pagina dedicata

Bonus +5 Punti

Sia nei form che nelle visualizzazione del select usare una **JOIN** per il ruolo

Barbieri Valentina Designer 2021-03-25
Bianchi Laura Project Manager 2019-06-10
Conti Davide Sviluppatore 2022-05-27
Costa Marco Designer 2020-12-03
Esposito Chiara Sviluppatore 2023-02-14
Ferrari Alessandro Analista 2018-11-30
Fontana Silvia Analista 2019-04-18
Gallo Elena Sviluppatore 2021-07-11
Greco Luca Project Manager 2016-09-12

Esercizio 2 – Tabella Dinamica con Sessioni (7 punti)

Sviluppare nella cartella Esercizio2 il file esercizio2.php con una pagina che contiene un form come in immagine, con campi per R (righe), C (colonne) e un input di testo per il contenuto.

Dati tabella

Righe	<input type="text" value="6"/>
Colonne	<input type="text" value="3"/>
Contenuto	<input type="text" value="Rossi"/>
<input type="button" value="Invia"/> <input type="button" value="Home"/> <input type="button" value="Destroy"/>	

Utilizzando la tecnica di postback, salvare i valori in variabili di sessione e visualizzare una tabella HTML composta da R righe e C colonne sotto al form a partire da tali valori. Ogni cella deve contenere il contenuto inserito seguito dall'indice della cella (es: Rossi-1, Rossi-2, ...).

Il bottone Home deve tornare a index.php; il bottone Destroy deve distruggere la sessione e cancellare la tabella.

Dando ai bottoni submit lo stesso attributo name (es. "action") e valori diversi puoi andare a differenziare le azioni

Dati tabella

Righe	<input type="text" value="6"/>
Colonne	<input type="text" value="3"/>
Contenuto	<input type="text" value="Rossi"/>
<input type="button" value="Invia"/> <input type="button" value="Home"/> <input type="button" value="Destroy"/>	

Rossi-1	Rossi-2	Rossi-3
Rossi-4	Rossi-5	Rossi-6
Rossi-7	Rossi-8	Rossi-9
Rossi-10	Rossi-11	Rossi-12
Rossi-13	Rossi-14	Rossi-15
Rossi-16	Rossi-17	Rossi-18

Esercizio 3 – Sistema di Login (15 punti)

Leggere attentamente e importare il file verifica.sql tramite phpMyAdmin nella cartella Esercizio3.

Sviluppare nella cartella Esercizio3 il file esercizio3.php contenente un form di login come in figura.

Login

Username

Password

Accedi

Dopo aver effettuato il controllo del login, reindirizzare alla pagina profilo.php che visualizza i dati dell'utente e include un bottone di logout come nella seconda figura.

Profilo

Nome: alberto

Cognome: costa

Username: acosta

Email: acosta@email.it

Logout

Ad ogni accesso effettuato con successo, richiamare una funzione contenuta nel file mail.php che inserisce nella tabella mail un record con i campi: destinatario, oggetto, testo e timestamp_invio.

Esercizio 4 – Configurazione Routes (8 punti)

Decommentare il file .htaccess per attivare la riscrittura degli URL e modificare il file index.php rimuovendo le estensioni ".php" dai tag <a>.

Aggiungere nel file routes.php le routes corrispondenti a esercizioX/esercizioX per ciascun esercizio.

Nota importante: Nei form che utilizzano postback, assicurarsi di specificare esplicitamente il file PHP di destinazione nell'attributo action (es: action="esercizio2.php") per evitare problemi di routing con il sistema di riscrittura URL.

PUNTEGGIO TOTALE: 45 punti

Esercizio 1 – Database e CRUD (20 punti)

- **Creazione database e tabelle:** 15 punti
- **Pagina con 4 form:** 5 punti
- **Insert.php:** 2 punti
- **Delete.php:** 2 punti
- **Update.php:** 2 punti
- **Select-all.php:** 2 punti
- **Select-role.php:** 2 punti
- **Query con JOIN:** 5 punti

Esercizio 2 – Tabella Dinamica con Sessioni (7 punti)

- **HTML:** 3 punti
- **Postback e variabili di sessione:** 4 punti

Esercizio 3 – Sistema di Login (15 punti)

- **Form di login:** 3 punti
- **Controllo credenziali:** 5 punti
- **Pagina profilo.php:** 3 punti
- **Logout:** 2 punti
- **Funzione mail.php:** 2 punti

Esercizio 4 – Configurazione Routes (8 punti)

Consegna finale

Rinominare la cartella sostituendo TUO COGNOME con il tuo cognome, alla fine comprimere la cartella e inserirla nella cartella che vi dirà il prof