TEMA ESAME 2023 INFORMATICA

SOLUZIONE

**PUNTO 1: descrizione della realtà e modello concettuale**

Si chiede di realizzare una piattaforma web per una scuola che consenta ai docenti della scuola di mettere a disposizione degli studenti i link a videogiochi didattici. La piattaforma non deve realizzare i videogiochi, deve fornire i link ai videogiochi agli studenti e raccogliere i risultati delle partite in modo che possa essere memorizzato il punteggio ottenuto dagli studenti nelle partite. Il punteggio viene valutato in “monete raccolte” durante una partita. Ciascun gioco è afferente ad uno specifico argomento didattico, pertanto il numero di monete raccolte da uno studente durante una partita è proporzionale al livello di preparazione dello studente.

I docenti devono poter creare delle classi virtuali e devono assegnare a ciascuna classe i giochi, ogni classe virtuale è caratterizzata da un codice di accesso. I docenti avranno uno username e una password per poter accedere alla piattaforma in modo da poter aggiungere classi, videogiochi e associazioni fra queste due entità, oltre a poter visualizzare i dati degli studenti nelle partite. Si ipotizza che gli studenti si iscrivano alla piattaforma mediante username e password, una volta effettuato il login, possano accedere alle classi inserendo il codice classe (numerico o qr code) fornito dal docente che ha creato la classe.

Si ipotizza che uno studente, dopo aver effettuato il login ed essersi iscritto ad una classe, possa accedere alla classe e giocare ai videogiochi associati a tale classe. Si ipotizza che al termina di una partita vengano memorizzati i seguenti dati per la partita: la data, l’ora, il numero di monete raccolte dal determinato studente per quel determinato videogioco. Si ipotizza che i dati di una partita di uno studente vengano memorizzati solamente se è la prima partita per il dato studente nel dato gioco oppure se le monete raccolte superano le monete raccolte nella precedente partita svolta. Si decide quindi di memorizzare solo la “miglior partita” giocata dallo studente in ogni determinato gioco.

I videogame educativi e la piattaforma dovranno comunicare fra loro poiché al termine di una partita è necessario che la piattaforma conosca il numero di monete raccolte dallo studente che ha giocato. Questo potrebbe avvenire mediante delle WEB API.

Si individuano le seguenti entità:

DOCENTI: Rappresenta l’anagrafica dei docenti. La password verrà memorizzata crittografata

CLASSE\_VIRTUALE: Rappresenta la classe e la materia di uno specifico gruppo di studenti. Poicè si presume che la piattaforma posa essere utilizzata per più anni, si memorizza, per ogni classe, l’anno scolastico a cui si riferisce. Ad esempio: 5BINT Matematica 2024/2025

STUDENTE: Rappresenta l’anagrafica degli studenti. La password sarà memorizzata crittografata

PARTITA: Rappresenta un partita svolta da uno studente in un determinato videogioco. E’ descritta dalla data, l’ora, il numero di monete raccolte dal giocatore/studente. Una partita viene memorizzata solo se è la prima partita giocata da un determinato giocatore in un gioco o solo se le monete raccolte superano le monete raccolte nelle precedenti partite di quel gioco da quel determinato giocatore.

VIDEOGIOCO: Rappresenta un videogioco che può essere assegnato dal docente ad una o più classi, l’entità è descritta da una descrizione breve (160 caratteri e una estesa) viene memorizzato il numero massimo di monete virtuali che uno studente può raccogliere durante una partita. Viene memorizzato un link al videogioco.

ARGOMENTO: Rappresenta gli argomenti didattici a cui si riferiscono i videogiochi (triangoli, legge di hom, verismo…) sono descritti da un identificativo numerico e da un titolo

IMMAGINI: Rappresentano delle immagini, ciascuna delle quali è associata ad un videogioco. Le immagini verranno memorizzate in una cartella del web server che ospita la piattaforma web, nel database verrà memorizzato il link a tale immagine nel file system del web server. Oltre al link vengono memorizzati un identificativo e una descrizione di quanto rappresentato nell’immagine. Per ogni videogioco vengono memorizzate al massimo 3 immagini.

Diagramma e/R

**fare..**

Vincoli/note:

* Lo studente può svolgere una partita solamente se il videogioco è stato assegnato ad una classe di cui lo studente fa parte
* Sia le password dei docenti che quelle degli studenti sono crittografate

**PUNTO 2: Schema logico**

DB schema

DOCENTI: id\_docente (PK), username, password, cognome, nome

ARGOMENTI: id\_argomento (PK), nome

STUDENTI: id\_studente (PK), cognome, nome, username, password

VIDEOGIOCHI: id\_videogioco (PK), titolo, descrizione\_breve, descrizione\_estesa, max\_monete, link, id\_argomento(FK)

CLASSI\_VIRTUALI: codice\_iscrizione (PK), classe, sezione, indirizzo, anno\_scolastico, materia, id\_docente(FK)

IMMAGINI: id\_immagine (PK) , nome, descrizione, pathname, id\_videogioco(FK)

CLASSI\_STUDENTI: id\_studente (PK,FK) codice\_classe (PK,FK)

CLASSI\_VIDEOGIOCHI: codice\_classe(PK), id\_videogioco (PK,FK)

PARTITE: id\_studente (PK, FK), id\_videogioco (PK, FK), data\_ora, monete\_raccolte

Scrivere i dettagli di ogni relazione ..fare

DOCENTI

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NOME | CHIAVE | TIPO | DIMENSIONE | NOT NULL | VINCOLI/NOTE |
| Id\_docente | PK | INTEGER |  | X | AUTO\_INCREMENT |
| username |  |  |  |  |  |
| ….. |  |  |  |  |  |
| …. |  |  |  |  |  |

**PUNTO 3: SQL per relazioni**

CREATE TABLE STUDENTI

( id\_studente integer auto\_increment, not null, primary key,

cognome varchar (30) not null,

nome varchar (30 not null

username varchar (30) unique not null

password varchar (255) unique not null

)

CREATE TABLE VIDEOGIOCHI

(id\_videogioco integer primary key not null,

titolo varchar (30 ) not null,

descrizione breve varchar (160) not null,

descrizione estesa text,

max\_monete integer not null,

link varchar(255) not null unique,

id\_argomento integer not null,

CONSTRAINT fk\_videogiochi\_argomenti FOREIGN KEY (id\_argomento) REFERENCES Argomenti(id\_argomento)

CONSTRAINT check\_monete CHECK (max\_monete>0));

CREATE TABLE PARTITE

(id\_studente integer not null,

id\_videogioco integer not null,

data\_ora datetime not null,

monete\_raccolte integer not null,

CONSTRAINT pk\_partite PRIMARY KEY (id\_studente, id\_videogioco),

CONSTRAINT fk\_partite\_studenti FORIGN KEY (id\_studente) REFERENCES Studenti(id\_studente),

CONSTRAINT fk\_partite\_videogiochi FOREIGN KEY(id\_videogiochi) REFERENCES Videogiochi(id\_videogioco),

CONSTRAINT check\_monete\_raccolte CHECK (monete\_raccolte>0));

**PUNTO 4: INTERROGAZIONI IN LINGUAGGIO SQL**

1. l’elenco in ordine alfabetico dei giochi classificati per uno specifico argomento;

SELECT v.titolo, v.descrizione\_breve

FROM videogiochi as v, argomenti as a

WHERE v.id\_argomento,argomenti.id\_argomento

AND a.id\_argomento=’argomento\_scelto’

ORDER BY v.titolo ASC

//Si ipotizza che rimanga memorizzata solamente la migliore partita giocata da uno studente in un determinato videogioco. Ogni volta che lo studente effettua una partita, se la partita si conclude con un numero maggiore di monete raccolte, rispetto alle precedenti partite, viene memorizzata la nuova partita, altrimenti rimane in memoria la partita precedente. Il controllo può essere svolto con un trigger

1. la classifica degli studenti di una certa classe virtuale, in base alle monete raccolte per un certo gioco;

SELECT s.id\_studente,s.cognome, s.nome, p.monete\_raccolte

FROM videogiochi as v, partite, as p, studenti as s, classi\_studenti as cs, classi as c

WHERE v.id\_videogioco=p.id\_videogioco,

AND p.id\_studente=s.id\_studente

AND s.id\_studente=cs\_id\_studente

AND cs.id\_classe=c.id\_classe

AND c.id\_classe=’classe\_scelta’

AND v.id\_videogioco=’videogioco\_scelto’

ORDER BY p.monete\_raccolte

1. il numero di classi in cui è utilizzato ciascun videogioco del catalogo;

SELECT v.id\_videogioco, v.titolo, count (\*) as numero\_classi

FROM videogioco as v, classi\_videogiochi as cv

WHERE v.id\_videogioco=cv.id\_videogioco

GROUP BY v.id\_videogioco

**PUNTO 5: progetto di massima dell’applicazione web**

Il progetto di un’applicazione web richiede la realizzazione di pagine web ospitate su una piattaforma (ad esempio Aruba) che fornisce:

* servizio di hosting
* servizio web server (ad esempio Apache)
* DBMS (mySql o MariaDB)

L’applicazione sarà utilizzabile per mezzo di una connessione internet con qualsiasi dispositivo (smartphone, pc, tablet)

Le pagine web saranno realizzate con la seguente tecnologia

* Linguaggi lato client HTML5, CSS, Javascript per realizzare l’interfaccia utente (inserimento dei dati e relativo controllo e validazione dell’input). Per rendere la web app responsive si sceglie di avvalersi delle funzioni della libreria Bootstrap.
* Linguaggio lato server PHP per l’interazione con il database.
* Linguaggio SQL embeddato nelle pagine PHP per le operazioni CRUD sulla base di dati.

L’accesso al database prevede tre diverse tipologie di utenti, ciascuno dei quali potrà accedere con permessi diversi al database:

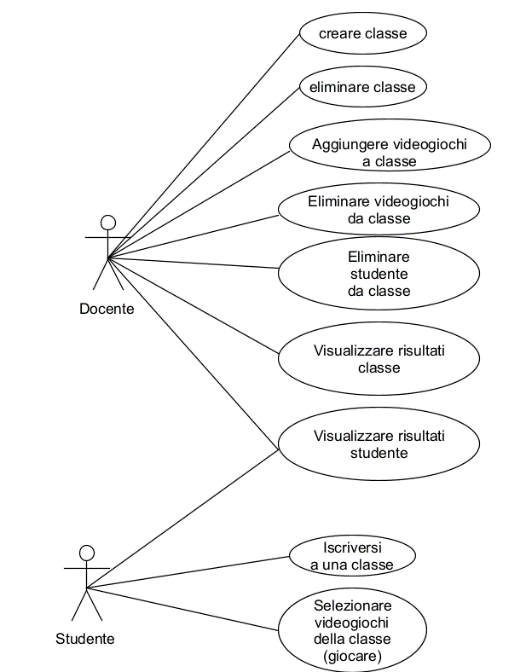
* profilo Amministratore del database: può creare nuove tipologie di utenti, può creare e modificare qualsiasi tabella, può leggere i dati di qualsiasi tabella
* profilo Docente: può creare classi, può aggiungere videogiochi alle proprie classi, può visualizzare i dati delle classi e i dati degli studenti iscritti alle proprie classi (dati di iscrizione, risultati delle partite), può eliminare uno studente da una classe.
* profilo Studente: può iscriversi ad una classe, può accedere ai videogiochi delle classi a cui è iscritto, può visualizzare i risultati delle proprie partite.

L’utilizzo dell’applicazione prevede due tipologie di utenti (attori): “Docente” e “Studente”. Tuti gli utenti, di entrambe le tipologie, accedono all’applicazione con procedura di login mediante username e password. Le password vengono memorizzate nel database in seguito a crittografia in modo da garantire la riservatezza dei dati.

Il diagramma dei casi d’uso di due attori è il seguente:

* Docente: accede mediante login con username e password, può creare classi, può aggiungere/eliminare videogiochi alle proprie classi, può visualizzare i dati delle classi e i dati degli studenti iscritti alle proprie classi (dati di iscrizione, risultati delle partite), può eliminare uno studente da una classe.
* Studente: accede mediante login con username e password, può iscriversi ad una classe (inserendo il codice di iscrizione fornito dal docente), può accedere ai videogiochi delle classi a cui è iscritto, può visualizzare i risultati delle proprie partite.

Diagramma dei casi d’uso:



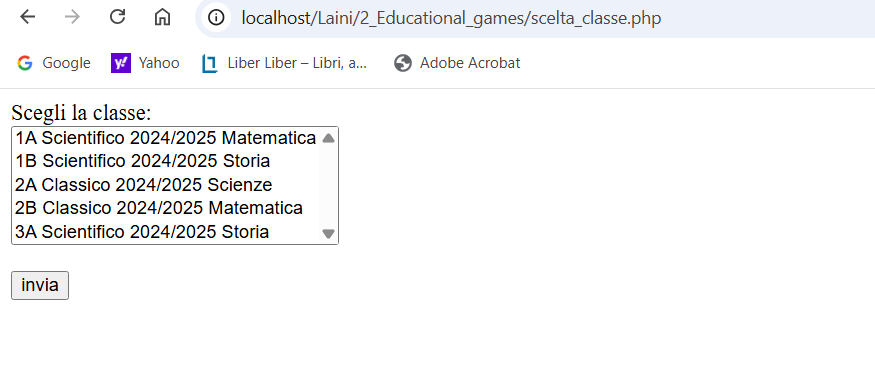
La struttura di massima dell’applicazione prevede la realizzazione delle seguenti pagine.

* Home\_page.html
* login\_Docente.php
  + Crea\_nuova\_classe.php
  + Seleziona\_mia\_classe.php
    - Visualizza risultati\_classe.php
    - aggiungi\_elimina\_videogioco.php
    - elimina\_studente\_dalla\_classe.php
    - elimina classe.php
  + Seleziona\_studente.php
    - Visualizza risultati\_studente.php
    - elimina\_studente\_dalle\_mie\_classi.php
* Login\_Studente.php
  + Iscrizione\_classe.php
  + Seleziona\_classe.php.php
    - Visualizza\_giochi.php
    - Visualizza\_miei\_risultati.php

**PUNTO 6: parte di applicazione web con interazione con database**

Vengono mostrate le pagine web che consentono di visualizzare la classifica di una determinata classe in un determinato videogioco:

Pagina 1: scelta della classe



Codice del file scelta\_classe.php:

<html>

<head>

    <title>Scelta classe</title>

</head>

<body>

    Scegli la classe:<br>

    <form method='POST' action="scelta\_videogioco.php">

        <select name="classe" size=5>

    <?php

        $servername="localhost";

        $username="root";

        $password="";

        $dbname="educationalgames";

        $connection=new mysqli($servername,$username,$password,$dbname);

        if($connection->connect\_errno!=0)

            die("Errore di connessione".$connection->connect\_errno."<br>");

        else

            {

                echo"connessione ok<br>";

                $query="select codice\_iscrizione,classe,indirizzo, anno\_scolastico,materia from classi\_virtuali;";

                $result=$connection->query($query);

                if($result->num\_rows==0)

                    echo"<br>nessun risultato";

                else

                    {

                        while($row=$result->fetch\_assoc())

                        {

                           echo "<option value=$row[codice\_iscrizione]>$row[classe] $row[indirizzo] $row[anno\_scolastico] $row[materia]</option>";

                        }

                        $result->free();

                        $connection->close();

                    }

            }

    ?>

        </select>

        <br><br>

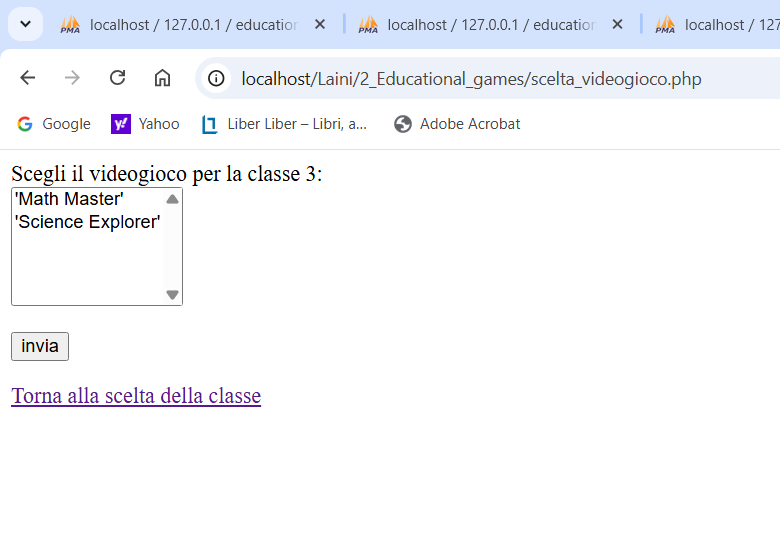
        <input type="submit" value="invia"/>

    </form>

</body>

</html>

Pagina 2: scelta del videogioco, la select box mostra solamente i videogiochi associati alla classe precedentemente selezionata:



Codice del file scelta\_videogioco.php

<html>

<head>

    <title>Scelta videogioco</title>

</head>

<body>

    <?php

        if (!isset($\_POST))

            die ("Nessuna classe scelta");

        else

        {

            $classe\_scelta=$\_POST['classe'];

            echo"Scegli il videogioco per la classe $classe\_scelta: <br>";

        }

    ?>

    <form method="POST" action="visualizza\_classifica.php">

        <input type="hidden" name="classe" value="<?php echo $classe\_scelta;?>"> <!--Invia la classe scelta con $\_POST-->

        <select name="videogioco" size=5>

    <?php

        $servername="localhost";

        $username="root";

        $password="";

        $dbname="educationalgames";

        $connection=new mysqli($servername,$username,$password,$dbname);

        if($connection->connect\_errno!=0)

            die("Errore di connessione".$connection->connect\_errno."<br>");

        else

            {

                echo"connessione ok<br>";

                $query="select v.id\_videogioco,v.titolo from classi\_videogiochi as cv,videogiochi as v

                where cv.id\_videogioco=v.id\_videogioco

                and cv.codice\_classe=$classe\_scelta;";

                $result=$connection->query($query);

                if($result->num\_rows==0)

                    echo"<br>nessun risultato";

                else

                    {

                        while($row=$result->fetch\_assoc())

                        {

                           echo "<option value=$row[id\_videogioco]>'$row[titolo]'</option>";

                        }

                        $result->free();

                        $connection->close();

                    }

            }

    ?>

        </select>

        <br><br>

        <input type="submit" value="invia"/>

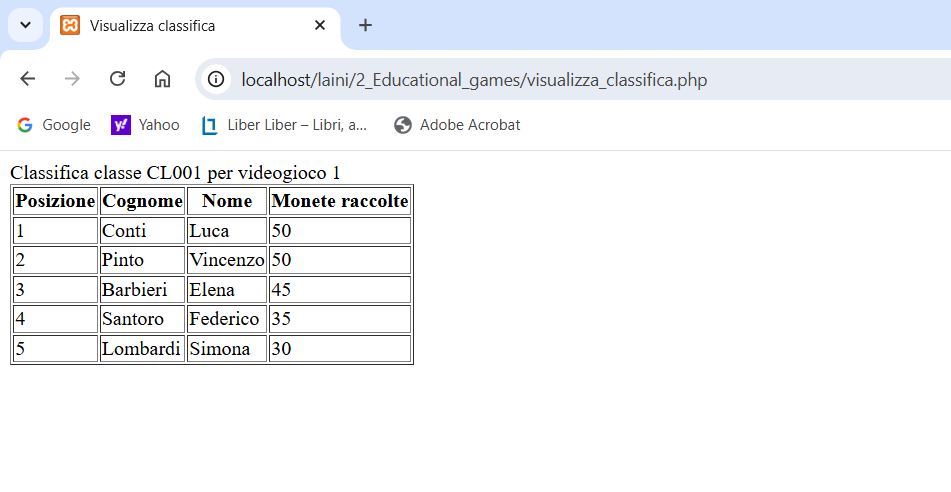
    </form>

    <a href='scelta\_classe.php'>Torna alla scelta della classe</href>

</body>

</html>

Pagina 3: Visualizzazione della classifica



Codice del file visualizza\_classifica.php

<html>

<head>

    <title>Visualizza classifica</title>

</head>

<body>

  <?php

  //var\_dump($\_POST);

  if (!(isset($\_POST['classe']) &&  isset($\_POST['videogioco'])))

  {

      die("valori non settati correttamente<br>");

  }

  else

  {

    $servername="localhost";

    $username="root";

    $password="";

    $dbname="videogiochieducativi";

    $connection=new mysqli($servername,$username,$password,$dbname);

    if($connection->connect\_errno)

    {

      echo"Error";

    }

    else

    {

      $codice\_classe=$\_POST['classe'];

      $id\_videogioco=$\_POST['videogioco'];

      $query="select s.cognome, s.nome, p.monete\_raccolte

              from classi\_virtuali as cv, classi\_studenti as cs, studenti as s, partite as p,videogiochi as v

              where cv.codice\_iscrizione=cs.codice\_classe

              and cs.id\_studente=s.id\_studente

              and s.id\_studente=p.id\_studente

              and p.id\_videogioco=v.id\_videogioco

              and v.id\_videogioco=$id\_videogioco

              and cv.codice\_iscrizione='$codice\_classe'

              group by s.id\_studente

              order by p.monete\_raccolte DESC;";

      $result=$connection->query($query);

      if($connection->connect\_errno!=0)

      {

         die("Query error ".$connection->connect\_errno);

      }

      else

      {

        echo "Classifica classe $codice\_classe per videogioco $id\_videogioco<br>";

        $row=$result->fetch\_assoc()

          ?>

          <table border="1">

            <tr>

                <th>Posizione</th>

                <th>Cognome</th>

                <th>Nome</th>

                <th>Monete raccolte</th>

            </tr>

          <?php

          $i=1;

          while($row=$result->fetch\_assoc())

          {

             $cognome=$row['cognome'];

             $nome=$row['nome'];

             $monete\_raccolte=$row['monete\_raccolte'];

             //var\_dump($row);

            ?>

            <tr>

                <td><?php echo"$i" ?></td>

                <td><?php echo"$cognome" ?></td>

                <td><?php echo"$nome" ?></td>

                <td><?php echo"$monete\_raccolte" ?></td>

            </tr>

            <?php

            $i++;

          }

      }

    }

  }

  ?>

  </table>

</body>

<html>