

Progetto di Basi di Dati

Forum di libri

Giacomo Zanatta - 859156

AA 2016-17

1 Introduzione

Ho scelto di portare il progetto "Forum di recensione di libri". Il progetto consiste in un'applicazione web composta da pagine dinamiche PHP e un database. L'applicazione permette di:

- Cercare i libri inseriti nel database, e visualizzare i dati e le eventuali recensioni
- Elencare i libri per ordine di giudizio positivi
- Scrivere commenti e/o recensioni
- Votare i commenti e le recensioni
- Registrare un nuovo utente
- Effettuare il login e il logout

L'applicazione è reperibile all'indirizzo <http://wwwstud.dsi.unive.it/gzanatta/progettoBasi2016-17/>

2 Descrizione del progetto e casi d'uso

2.1 Descrizione

La pagina iniziale permette, mediante appositi link, di effettuare il login e la registrazione. Inoltre, fornisce un form di ricerca di libri, per poterne vedere recensioni e informazioni.

Se l'utilizzatore non si è ancora registrato al sito (o non ha ancora effettuato il login), il sito è in sola consultazione: l'utilizzatore non può quindi inserire dati (libri, recensioni, commenti e autori) ma solamente 'sfogliare' la base di dati, cercando ad esempio un titolo e visualizzarne le recensioni e i commenti a quest'ultime.

Per gli utenti registrati e loggati invece è offerta anche la possibilità di inserire una recensione, di commentare una recensione o un commento, di votare (+1 o -1) un commento e di inserire dei libri e i relativi autori.

2.2 Casi d'uso

2.2.1 Login

Per effettuare il login, un utente clicca sul link LOGIN presente nell'header. Viene presentato un form composto da 2 campi testuali (username e password) e un bottone. L'utente dovrà quindi inserire il proprio username e la propria password, e se queste due informazioni sono inserite in modo corretto allora l'utente verrà autenticato.

2.2.2 Signup

Un utente può registrarsi al sito, cliccando sul link SIGNUP. A questo punto è necessario completare il form proposto, inserendo l'username del nuovo utente, la sua password, e le informazioni personali quali nome (facoltativo), cognome (facoltativo), email e data di nascita (facoltativa). L'email viene inserita, sebbene non riporti utilità per questo progetto, per poter evitare problemi riguardo ad eventuali sviluppi futuri dell'applicazione (ad esempio, se in futuro verrà implementato un servizio di recupero password, o un servizio di notifica alla ricezione di commenti riguardanti una recensione, o un altro ipotetico servizio, allora avere l'email dell'utente all'interno della base di dati risulterà molto utile, in quanto si andrebbe ad evitare di richiedere la mail in seguito).

Dopo aver inserito le informazioni richieste l'utente dovrà cliccare sul bottone SIGNUP, e se le informazioni sono corrette (ossia l'username e la mail scelta non appartengono ad un altro utente) allora l'utente verrà inserito nel database. Dopo essersi registrato, sarà possibile effettuare il login inserendo le nuove credenziali.

Dopo una registrazione, non si viene autenticati in modo automatico per motivi di sicurezza.

2.2.3 Logout

La funzione di logout è permessa se e solo se un utente è autenticato al sito. L'utente in questione, quando vorrà scollegarsi dal sito, cliccherà nel link LOGOUT presente sotto il titolo, e verrà disconnesso senza ulteriori richieste. Dopo essersi disconnesso, è possibile autenticarsi con un altro profilo o chiudere la pagina, senza preoccuparsi della sessione che è stata distrutta al momento della disconnessione.

2.2.4 Cerca

Un utente (autenticato o non) può cercare i libri nel database. Inizialmente verranno presentati all'utente tutti i libri presenti. Dopo aver inserito le informazioni richieste per la ricerca ed aver premuto il bottone presentato (CERCA) nella pagina principale, verranno stampati i risultati della ricerca.

È possibile a questo punto cliccare sul libro cercato (se presente nel database)

per visualizzare ulteriori informazioni ed eventuali recensioni del libro in questione.

2.2.5 Inserire un libro

Se l'utente è autenticato viene permesso di inserire un libro.

Cliccando sul link INSERISCI, presente sotto alla funzione cerca, si viene reindirizzati ad una pagina contenente un form da completare con le informazioni del libro: vengono richiesti titolo, autore/i, genere, ed eventualmente una breve trama.

2.2.6 Inserire un autore

Un utente registrato ed autenticato, oltre ad inserire un libro, può inserire un autore. Può essere inserito mediante l'apposito link nella pagina iniziale o al momento dell'inserimento di un libro. Vengono richiesti nome, cognome, nazionalità e data di nascita.

2.2.7 Recensire un libro

Un utente registrato ed autenticato può scrivere una recensione di un libro. Per scrivere una recensione, deve prima cercare il libro che vuole recensire e successivamente, nella pagina del libro, compilare il form (titolo, testo e voto). Se l'utente è soddisfatto della sua recensione, può premere il bottone INSERISCI che effettuerà il caricamento della recensione nella base di dati.

2.2.8 Commentare una recensione o un commento

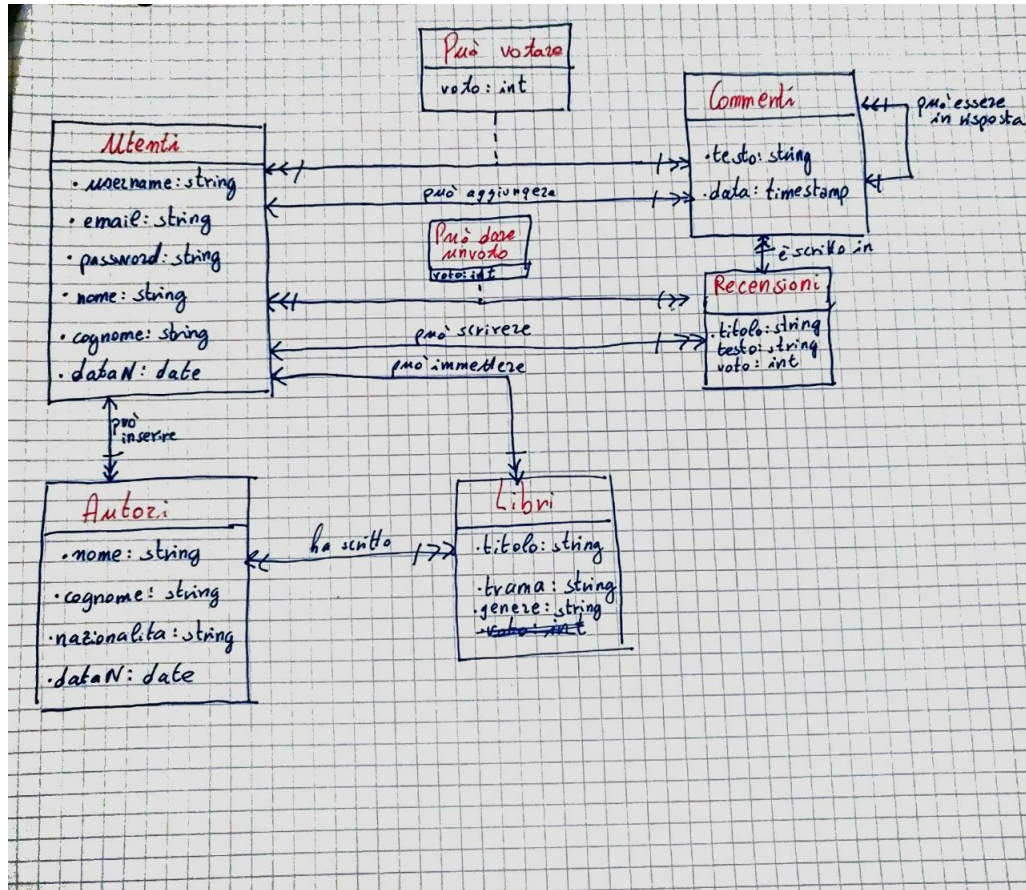
Per commentare una recensione è necessario recarsi, dopo previa autenticazione, alla recensione in questione. Alla fine della recensione, si preme il bottone COMMENTA, e si scrive il proprio commento. I commenti non sono anonimi ma vengono pubblicati con l'username dell'utente.

È possibile anche commentare un commento: il procedimento è simile a quello appena visto, con la sola differenza che è necessario cliccare su COMMENTA posto al di sotto del commento che si vuole commentare.

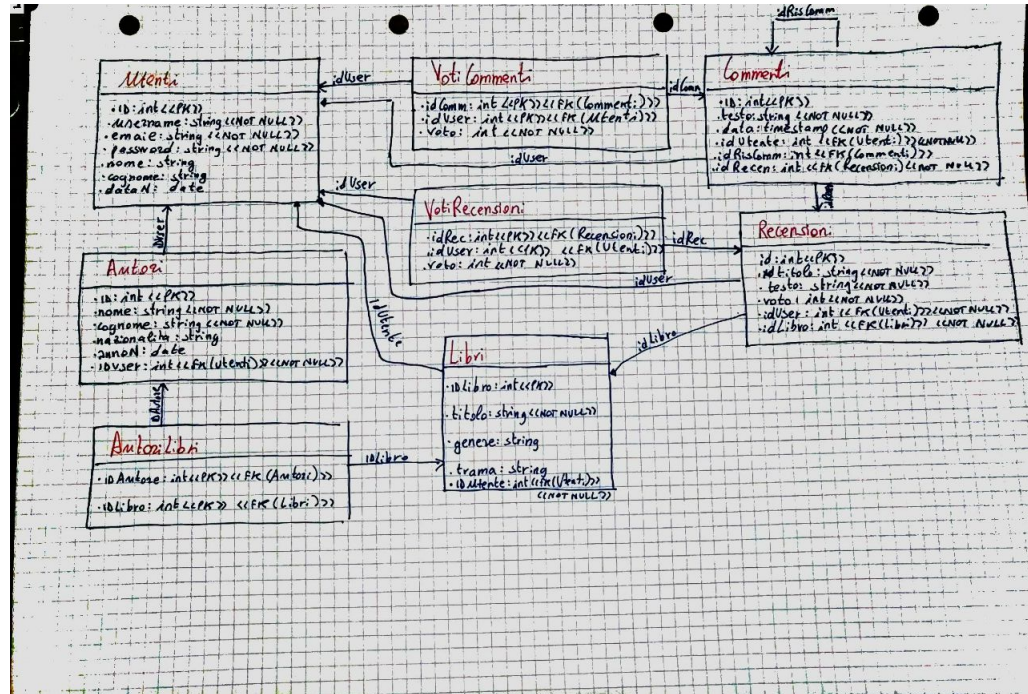
2.2.9 Votare un commento o una recensione

Di fianco al bottone COMMENTA sono presenti tre bottoni: +1, 0 e -1. Questi bottoni permettono di votare un commento o una recensione. L'utente se desidera votare una recensione o un commento deve quindi essere autenticato e pigiare sull'apposito bottone che esprime la sua preferenza: +1 se la recensione (o il commento) è valido, -1 se invece non lo è. Per annullare il voto si clicca sul link 0.

3 Schema concettuale



4 Schema relazionale



5 Codice di creazione del database

5.1 Definizione dello schema

```
CREATE SCHEMA progetto_db
```

5.2 Creazione tabelle

```
CREATE TABLE progetto_db.utenti(  
    id SERIAL,  
    username text not null UNIQUE,  
    password text not null,  
    nome text,  
    cognome text,  
    email text not null UNIQUE,  
    data_nascita date,  
    primary key(id));
```

```
CREATE TABLE progetto_db.libri(
    id SERIAL,
    titolo text,
```

```

        trama text,
        genere text,
        valutazione float(2) DEFAULT NULL,
        id_user int,
        primary key(id));

ALTER TABLE progetto_db.libri
    ADD foreign key (id_user) REFERENCES progetto_db.utenti
    ON DELETE SET NULL;

CREATE TABLE progetto_db.autori(
    id SERIAL,
    nome text not null,
    cognome text not null,
    data_nascita date,
    nazionalita text,
    aggiunto_da int not null,
    primary key(id));

CREATE TABLE progetto_db.libri_autori(
    id_autore int not null,
    id_libro int not null,
    primary key(id_autore, id_libro));

ALTER TABLE progetto_db.libri_autori
    ADD foreign key (id_autore) REFERENCES progetto_db.autori
    ON DELETE SET NULL;

ALTER TABLE progetto_db.libri_autori
    ADD foreign key (id_libro) REFERENCES progetto_db.libri
    ON DELETE SET NULL;

CREATE TABLE progetto_db.recensioni(
    id_recensione SERIAL,
    titolo text NOT NULL,
    testo text NOT NULL,
    voto int NOT NULL,
    punteggio int NOT NULL DEFAULT 0,
    id_autore int NOT NULL,
    id_libro int NOT NULL,
    primary key(id_recensione));

ALTER TABLE progetto_db.recensioni
    ADD foreign key (id_autore) REFERENCES progetto_db.utenti
    ON DELETE SET NULL;

```

```

ALTER TABLE progetto_db.recensioni
    ADD foreign key (id_libro) REFERENCES progetto_db.libri
    ON DELETE CASCADE;

CREATE TABLE progetto_db.voti_recensioni(
    id_utente int NOT NULL,
    id_recensione int NOT NULL,
    voto int NOT NULL,
    primary key(id_utente, id_recensione));

ALTER TABLE progetto_db.voti_recensioni
    ADD foreign key (id_utente) REFERENCES progetto_db.utenti
    ON DELETE CASCADE;

ALTER TABLE progetto_db.voti_recensioni
    ADD foreign key (id_recensione) REFERENCES progetto_db.recensione
    ON DELETE CASCADE;

CREATE TABLE progetto_db.commenti(
    id_commento SERIAL,
    testo text NOT NULL,
    punteggio int default 0,
    id_user int NOT NULL,
    id_recensione int not null,
    id_ref_comm int,
    data_commento timestamp,
    primary key(id_commento));

ALTER TABLE progetto_db.commenti
    ADD foreign key (id_user) REFERENCES progetto_db.utenti
    ON DELETE SET NULL;

ALTER TABLE progetto_db.commenti
    ADD foreign key (id_ref_comm) REFERENCES progetto_db.commenti
    ON DELETE CASCADE;

ALTER TABLE progetto_db.commenti
    ADD foreign key (id_recensione) REFERENCES progetto_db.recensioni
    ON DELETE CASCADE;

CREATE TABLE progetto_db.voti_commenti(
    id_utente int NOT NULL,
    id_commento int NOT NULL,
    voto int NOT NULL,
    primary key(id_utente, id_commento));

```

```

ALTER TABLE progetto_db.voti_commenti
    ADD foreign key (id_utente) REFERENCES progetto_db.utenti
    ON DELETE CASCADE;

ALTER TABLE progetto_db.voti_commenti
    ADD foreign key (id_commento) REFERENCES progetto_db.commenti
    ON DELETE CASCADE;

```

5.3 Creazione funzioni

```

//esegui_login: ritorna la password (criptata) di un utente, per poter
    fare la verifica con la password inserita
create or replace function esegui_login(u text) returns text as $$
declare
    pwd text;
begin
    select password into pwd from progetto_db.utenti where u = username;
    return pwd;
end;
$$ language plpgsql;
//get id: ritorna l'id di un utente
create or replace function get_id(u text) returns int as $$
declare
    id_user text;
begin
    select id into id_user from progetto_db.utenti where u = username;
    return id_user;
end;
$$ language plpgsql;
//inserisci utente: inserisce un utente nel db
create or replace function inserisci_utente(u text, p text, e text, nom
    text, co text) returns void as $$
    insert into progetto_db.utenti (username, password, email, nome,
        cognome) values (u, p, e, nom, co);
$$ language sql;
//inserisci utente con data: inserisce un utente nel db, specificando
    anche la data di nascita
create or replace function inserisci_utente_con_data(u text, p text, e
    text, d date, nom text, co text) returns void as $$
    insert into progetto_db.utenti (username, password, email,
        data_nascita, nome, cogome) values (u, p, e, d, nom, co);
$$ language sql;
//username esistente: controlla se un username esiste
create or replace function username_esistente(u text) returns boolean as
    $$
declare
    usr text;
begin

```



```

        select username into usr from progetto_db.utenti where u = username;
    if u = usr
        then return true;
        else return false;
    end if;
end;
$$ language plpgsql;
//mail esistente: controlla se una mail esiste
create or replace function mail_esistente(m text) returns boolean as $$
    declare
        mail text;
    begin
        select email into mail from progetto_db.utenti where m = email;
    if m = mail
        then return true;
        else return false;
    end if;
end;
$$ language plpgsql;
//get autori: ritorna tutti gli autori presenti
create or replace function get_autori() returns autori as $$
    select * from progetto_db.autori;
$$ language plpgsql;
//inserisci libro_X_X: funzioni che permettono di inserire un libro
    all'interno del database, specificando ulteriori dati aggiuntivi
create or replace function inserisci_libro_genere_trama(t text, tr text,
    ge text, us int) returns int as $$
    declare
        i int;
    begin
        insert into progetto_db.libri (titolo,trama, genere, id_user) values
            (t, tr, ge, us) returning id into i;
        return i;
    end;
$$ language plpgsql;

create or replace function inserisci_libro_trama(t text, tr text, us
    int) returns int as $$
    declare
        i int;
    begin
        insert into progetto_db.libri (titolo,trama, id_user) values (t, tr,
            us) returning id into i;
        return i;
    end;
$$ language plpgsql;

create or replace function inserisci_libro_genere(t text, ge text, us
    int) returns int as $$
    declare

```

```

        i int;
begin
    insert into progetto_db.libri (titolo, genere, id_user) values (t, ge,
        us) returning id into i;
    return i;
end;
$$ language plpgsql;

create or replace function inserisci_libro(t text, us int) returns int
as $$
declare
    i int;
begin
    insert into progetto_db.libri (titolo, id_user) values (t, us)
        returning id into i;
    return i;
end;
$$ language plpgsql;
//inserisci_autore_X_X: funzioni che permettono di inserire autori nel
db, scegliendo anche dati aggiuntivi non obbligatori.
create or replace function inserisci_autore_nazionalita_data(n text, c
    text, na text, da date, us int) returns void as $$
    insert into progetto_db.autori (nome,cognome, nazionalita,
        data_nascita, aggiunto_da) values (n, c, na, da, us);
$$ language sql;

create or replace function inserisci_autore_data(n text, c text, da
    date, us int) returns void as $$
    insert into progetto_db.autori (nome,cognome, data_nascita,
        aggiunto_da) values (n, c, da, us);
$$ language sql;

create or replace function inserisci_autore_nazionalita(n text, c text,
    na text, us int) returns void as $$
    insert into progetto_db.autori (nome, cognome, nazionalita,
        aggiunto_da) values (n, c, na, us);
$$ language sql;

create or replace function inserisci_autore(n text, c text, us int)
    returns void as $$
    insert into progetto_db.autori (nome, cognome, aggiunto_da) values (n,
        c, us);
$$ language sql;

```

5.4 Creazione trigger

```

//TRIGGER PER AGGIORNAMENTO MEDIA LIBRO
create or replace function aggiorna_media_libro() returns trigger as $$

```

```

declare
    idl int = new.id_libro;
begin
    update progetto_db.libri set valutazione = (SELECT avg(voto) FROM
        progetto_db.recensioni where id_libro=idl) WHERE id=idl;
return new;
end
$$ language plpgsql;

drop trigger if exists trigger_aggiorna_media_libro on
    progetto_db.recensioni;
create trigger trigger_aggiorna_media_libro after insert on
    progetto_db.recensioni
    for each row execute procedure aggiorna_media_libro();

//TRIGGER PER AGGIORNAMENTO VOTO RECENSIONE
create or replace function aggiorna_voti_recensione() returns trigger as
    $$
declare
    idr int = new.id_recensione;
begin
    update progetto_db.recensioni set punteggio =
        (select coalesce((select sum(voto)
            from progetto_db.voti_recensioni
            where id_recensione=idr),0))
        WHERE id_recensione=idr;
    return new;
end
$$ language plpgsql;

drop trigger if exists trigger_aggiorna_voti_recensione on
    progetto_db.voti_recensioni;
create trigger trigger_aggiorna_voti_recensione after insert on
    progetto_db.voti_recensioni
    for each row execute procedure aggiorna_voti_recensione();

create or replace function aggiorna_voti_recensione_after_delete()
    returns trigger as $$
declare
    idr int = old.id_recensione;
begin
    update progetto_db.recensioni set punteggio =
        (select coalesce((select sum(voto)
            from progetto_db.voti_recensioni
            where id_recensione=idr),0))
        where id_recensione=idr;
    return old;
end
$$ language plpgsql;

```

```

drop trigger if exists trigger_aggiorna_voti_recensione_after_delete on
    progetto_db.voti_recensioni;
create trigger trigger_aggiorna_voti_recensione_after_delete after
    delete on progetto_db.voti_recensioni
    for each row execute procedure aggiorna_voti_recensione_after_delete();

//TRIGGER PER AGGIORNAMENTO VOTO COMMENTO

create or replace function aggiorna_voti_commento() returns trigger as $$
declare
    idc int = new.id_commento;
begin
    update progetto_db.commenti set punteggio =
        (select coalesce((select sum(voto)
            from progetto_db.voti_commenti
            where id_commento=idc),0))
        where id_commento=idc;
    return new;
end
$$ language plpgsql;

drop trigger if exists trigger_aggiorna_voti_commento on
    progetto_db.voti_commenti;
create trigger trigger_aggiorna_voti_commento after insert on
    progetto_db.voti_commenti
    for each row execute procedure aggiorna_voti_commento();

create or replace function aggiorna_voti_commento_after_delete() returns
    trigger as $$
declare
    idc int = old.id_commento;
begin
    update progetto_db.commenti set punteggio =
        (select coalesce((select sum(voto)
            from progetto_db.voti_commenti
            where id_commento=idc),0))
        where id_commento=idc;
    return old;
end
$$ language plpgsql;

drop trigger if exists trigger_aggiorna_voti_commento_after_delete on
    progetto_db.voti_commenti;
create trigger trigger_aggiorna_voti_commento_after_delete after delete
    on progetto_db.voti_commenti
    for each row execute procedure aggiorna_voti_commento_after_delete();

```

6 Codice PHP

6.1 File funzioni_db.php

```
<?php
//CONNESSIONE: permette di connettersi al server*/
function connessione() {
    $connection = new
        PDO('pgsql:host=dblab.dsi.unive.it;port=5432;dbname=a2016u130','a2016u130','Y8q1600V');
    $connection -> setAttribute (PDO::ATTR_ERRMODE,
        PDO::ERRMODE_EXCEPTION);
    return $connection;
}
//LOGIN: esegue il login
function login($username){
    try{
        $dbconn = connessione();
        $statement = $dbconn->prepare('select esegui_login(?)');
        $statement->execute(array($username));
        return $statement->fetch(PDO::FETCH_NUM)[0];
    } catch (PDOException $e) { echo $e->getMessage(); }
}
//GET ID USER: ritorna l'id di un utente, dato l'username
function get_id_user($username){
    try{
        $dbconn = connessione();
        $statement = $dbconn->prepare("select get_id(?)");
        $statement->execute(array($username));
        return $statement->fetch(PDO::FETCH_NUM)[0];
    } catch (PDOException $e) { echo $e->getMessage(); }
}
//USER ESISTE: verifica se l'username esiste
function user_esiste($username){
    try{
        $dbconn = connessione();
        $statement = $dbconn->prepare('select
            username_esistente(?)');
        if($statement->execute(array($username))[0]==1)
            return true;
        else
            return false;
    } catch (PDOException $e) { echo $e->getMessage(); }
}
//MAIL ESISTE: verifica se l'indirizzo mail esiste
function mail_esiste($email){
    try{
        $dbconn = connessione();
        $statement = $dbconn->prepare('select mail_esistente(?)');
        if($statement->execute(array($email))[0]==1)
```

```

        return true;
    else
        return false;
    } catch (PDOException $e) { echo $e->getMessage(); }
}

//ESEGUI REGISTRAZIONE: registra l'utente del database
function esegui_registrazione($username, $hashed_pwd, $email, $nome,
    $cognome){
    try{
        $dbconn = connessione();
        echo $username." ".$email." ".$nome;
        $statement = $dbconn->prepare('select inserisci_utente(?, ?,
            ?, ?, ?)');
        $statement->execute(array($username,$hashed_pwd, $email,
            $nome, $cognome));
    } catch (PDOException $e) { echo $e->getMessage(); }
}

function esegui_registrazione_con_data($username, $hashed_pwd,
    $email, $data, $nome, $cognome){
    try{
        $dbconn = connessione();
        $statement = $dbconn->prepare('select
            inserisci_utente_con_data(?, ?, ?, ?,?,?)');
        $statement->execute(array($username,$hashed_pwd, $email,
            $data, $nome, $cognome));
    } catch (PDOException $e) { echo $e->getMessage(); }
}

//GET AUTORI: ritorna un array associativo contenente le
    informazioni di tutti gli autori
function get_autori(){
    try{
        $dbconn = connessione();
        $statement = $dbconn->prepare('select cognome, nome, id from
            progetto_db.autori ORDER BY cognome, nome');
        $statement->execute();
        return $statement->fetchAll(PDO::FETCH_ASSOC);
    } catch (PDOException $e) { echo $e->getMessage(); }
}

//GET LIBRI ORDINATI: ritorna tutti i libri all'interno del DB
    ordinato in un determinato modo
function get_libri_ordinati($ordinamento){
    try{
        $dbconn = connessione();
        $statement = $dbconn->prepare("select id, titolo, valutazione
            FROM progetto_db.libri ORDER BY $ordinamento");
        $statement->execute();
        return $statement->fetchAll(PDO::FETCH_ASSOC);
    } catch (PDOException $e) { echo $e->getMessage(); }
}

```

```

//GET LIBRI CERCA: ricerca un libro per titolo
function get_libri_cerca($titolo){
    try{
        $dbconn = connessione();
        $statement = $dbconn->prepare("select id, titolo, valutazione
            FROM progetto_db.libri WHERE titolo ilike ?");
        $statement->execute(array("%$titolo%"));
        return $statement->fetchAll(PDO::FETCH_ASSOC);
    } catch (PDOException $e) { echo $e->getMessage(); }
}

//INS LIBRO: inserisce un libro nel DB
function ins_libro($titolo, $trama, $genere, $id_user, $autori){
    try{
        $dbconn = connessione();
        if(empty($trama) && empty($genere)){
            $statement = $dbconn->prepare('select
                inserisci_libro(?,?)');
            $statement->execute(array($titolo, $id_user));
        }
        else if(!empty($trama) && !empty($genere)){
            $statement = $dbconn->prepare('select
                inserisci_libro_genere_trama(?,?,?,?)');
            $statement->execute(array($titolo, $trama, $genere,
                $id_user));
        }
        else if(!empty($trama) && empty($genere)){
            $statement = $dbconn->prepare('select
                inserisci_libro_trama(?,?,?)');
            $statement->execute(array($titolo, $trama, $id_user));
        }
        else if(empty($trama) && !empty($genere)){
            $statement = $dbconn->prepare('select
                inserisci_libro_genere(?,?,?)');
            $statement->execute(array($titolo, $genere, $id_user));
        }
        $id_libro = $statement->fetch(PDO::FETCH_NUM)[0];
        foreach($autori as $autore){
            $statement = $dbconn->prepare('insert into
                progetto_db.libri_autori(id_autore, id_libro)
                values(?, ?)');
            $statement->execute(array($autore, $id_libro));
        }
    } catch (PDOException $e) { echo $e->getMessage();}
}

//INS AUTORE: inserisce un autore nel DB
function ins_autore($nome, $cognome, $nazionalita, $data_n,
    $user_id){
    try{
        $dbconn = connessione();
        if(empty($nazionalita) && empty($data_n)){

```

```

        $statement = $dbconn->prepare('select
            inserisci_autore(?, ?, ?)');
        $statement->execute(array($nome, $cognome, $user_id));
    }
    else if(!empty($nazionalita) && !empty($data_n)){
        $statement = $dbconn->prepare('select
            inserisci_autore_nazionalita_data(?,?,?, ?, ?)');
        $statement->execute(array($nome, $cognome, $nazionalita,
            $data_n, $user_id));
    }
    else if(!empty($nazionalita) && empty($data_n)){
        $statement = $dbconn->prepare('select
            inserisci_autore_nazionalita(?,?,?, ?)');
        $statement->execute(array($nome, $cognome, $nazionalita,
            $user_id));
    }
    else if(empty($nazionalita) && !empty($data_n)){
        $statement = $dbconn->prepare('select
            inserisci_autore_data(?,?,?, ?)');
        $statement->execute(array($nome, $cognome, $data_n,
            $user_id));
    }
} catch (PDOException $e) { echo $e->getMessage();}
}
//GET INFO LIBRO: ritorna le informazioni di un determinato libro
function get_info_libro($id){
    try{
        $dbconn = connessione();
        $statement = $dbconn->prepare('select titolo, genere, trama,
            username from progetto_db.libri join progetto_db.utenti
            on progetto_db.utenti.id=id_user where
            progetto_db.libri.id=?');
        $statement->execute(array($id));
        return $statement->fetchAll(PDO::FETCH_ASSOC);
    } catch (PDOException $e) { echo $e->getMessage(); }
}
//GET AUTORI LIBRO: ritorna le informazioni degli autori di un libro
function get_autori_libro($id){
    try{
        $dbconn = connessione();
        $statement = $dbconn->prepare('select nome, cognome from
            progetto_db.autori join progetto_db.libri_autori on
            id=id_autore where id_libro=?');
        $statement->execute(array($id));
        return $statement->fetchAll(PDO::FETCH_ASSOC);
    } catch (PDOException $e) { echo $e->getMessage(); }
}
//GET RECENSIONE: ritorna una recensione, specificata da un ID
function get_recensione($id){
    try{

```



```

$dbconn = connessione();
$stmt = $dbconn->prepare('select
    progetto_db.recensioni.titolo, username, testo, voto,
    progetto_db.libri.titolo as titolo_libro from
    progetto_db.recensioni join progetto_db.utenti on
    id=id_autore join progetto_db.libri on
    progetto_db.libri.id=id_libro where id_recensione=?');
$stmt->execute(array($id));
return $stmt->fetchAll(PDO::FETCH_ASSOC)[0];
} catch (PDOException $e) { echo $e->getMessage(); }
}

//GET RECENSIONI LIBRO: ritorna tutte le recensioni di un
determinato libro
function get_recensioni_libro($id){
    try{
        $dbconn = connessione();
        $stmt = $dbconn->prepare('select id_recensione, titolo,
            username from progetto_db.recensioni join
            progetto_db.utenti on id=id_autore where id_libro=?');
        $stmt->execute(array($id));
        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    } catch (PDOException $e) { echo $e->getMessage(); }
}

//GIA RECENSITO: controlla se l'utente ha gia' recensito un libro
function gia_recensito($id_libro, $username){
    try{
        $dbconn = connessione();
        $stmt = $dbconn->prepare('select id_recensione from
            progetto_db.recensioni where id_libro=? and
            id_autore=?');
        $stmt->execute(array($id_libro, $username));
        return !empty($stmt->fetchAll(PDO::FETCH_ASSOC));
    } catch (PDOException $e) { echo $e->getMessage(); }
}

//INSERISCI RECENSIONE: inserisce una recensione all'interno del DB
function inserisci_recensione($titolo, $testo, $valutazione,
    $id_username, $id_libro){
    try{
        $dbconn = connessione();
        $stmt = $dbconn->prepare('insert into
            progetto_db.recensioni(titolo, testo, voto, id_autore,
            id_libro) values (?, ?, ?, ?, ?)');
        $stmt->execute(array($titolo, $testo, $valutazione,
            $id_username, $id_libro));
        /*aggiorno punteggio medio*/
        $stmt = $dbconn->prepare('update progetto_db.libri set
            valutazione = (SELECT avg(voto) FROM
            progetto_db.recensioni where id_libro=?) WHERE id=?');
        $stmt->execute(array($id_libro, $id_libro));
    } catch (PDOException $e) { echo $e->getMessage(); }
}

```

```

}
//GET COMMENTI: ritorna i commenti di una recensione
function get_commenti($id_rec){
    try{
        $dbconn = connessione();
        $statement = $dbconn->prepare('select id_commento, testo,
            punteggio, data_commento, username from
            progetto_db.commenti join progetto_db.utenti on
            id_user=id where id_recensione=? and id_ref_comm is NULL
            order by data_commento');
        $statement->execute(array($id_rec));
        return $statement->fetchAll(PDO::FETCH_ASSOC);
    } catch (PDOException $e) { echo $e->getMessage(); }
}
//GET COMMENTI COMMENTO: ritorna i commenti di un determinato
commento
function get_commenti_commento($id_comm){
    try{
        $dbconn = connessione();
        $statement = $dbconn->prepare('select id_commento, testo,
            punteggio, data_commento, username from
            progetto_db.commenti join progetto_db.utenti on
            id_user=id where id_ref_comm=? order by data_commento');
        $statement->execute(array($id_comm));
        return $statement->fetchAll(PDO::FETCH_ASSOC);
    } catch (PDOException $e) { echo $e->getMessage(); }
}
//INSERISCI COMMENTO: inserisce un commento nel DB
function inserisci_commento($testo, $id_user, $id_rec,
    $commento_padre, $data){
    try{
        $dbconn = connessione();
        $statement = $dbconn->prepare('insert into
            progetto_db.commenti(testo, id_user, id_recensione,
            id_ref_comm, data_commento) values (?, ?, ?, ?, ?)');
        $statement->execute(array($testo, $id_user, $id_rec,
            $commento_padre, $data));
    } catch (PDOException $e) { echo $e->getMessage(); }
}
//INSERISCI VOTO COMMENTO: inserisce un voto al commento.
//N.B: un utente puo' votare un commento una volta sola, se vuole
ri votare, e' necessario cancellare il voto vecchio
function inserisci_voto_commento($id_user, $id_commento, $voto){
    try{
        $dbconn = connessione();
        $statement = $dbconn->prepare('delete from
            progetto_db.voti_commenti where id_utente=? and
            id_commento=?');
        $statement->execute(array($id_user, $id_commento));
        if($voto!=0){

```

```

        $statement = $dbconn->prepare('insert into
            progetto_db.voti_commenti (id_utente, id_commento,
            voto) values(?,?,?)');
        $statement->execute(array($id_user, $id_commento, $voto));
    }
} catch (PDOException $e) { echo $e->getMessage(); }
}

function inserisci_voto_recensione($id_user, $id_recensione, $voto){
    try{
        $dbconn = connessione();
        $statement = $dbconn->prepare('delete from
            progetto_db.voti_recensioni where id_utente=? and
            id_recensione=?');
        $statement->execute(array($id_user, $id_recensione));
        if($voto!=0){
            $statement = $dbconn->prepare('insert into
                progetto_db.voti_recensioni (id_utente,
                id_recensione, voto) values(?,?,?)');
            $statement->execute(array($id_user, $id_recensione,
                $voto));
        }
    } catch (PDOException $e) { echo $e->getMessage(); }
}

//GET INFO UTENTE
function get_info_utente($id){
    try{
        $dbconn = connessione();
        $statement = $dbconn->prepare('select username, nome,
            cognome, data_nascita, email from progetto_db.utenti
            where id=?');
        $statement->execute(array($id));
        return $statement->fetchAll(PDO::FETCH_ASSOC)[0];

    } catch (PDOException $e) { echo $e->getMessage(); }
}

//GET RECENSIONI UTENTI
function get_recensioni_utente($id){
    try{
        $dbconn = connessione();
        $statement = $dbconn->prepare('select id_recensione as id,
            progetto_db.recensioni.titolo as titolo,
            progetto_db.libri.titolo as titolo_libro from
            progetto_db.recensioni join progetto_db.utenti on
            progetto_db.utenti.id = id_autore
            join progetto_db.libri on progetto_db.recensioni.id_libro =
            progetto_db.libri.id where id_autore=?');
        $statement->execute(array($id));
        return $statement->fetchAll(PDO::FETCH_ASSOC);
    } catch (PDOException $e) { echo $e->getMessage(); }
}

```

```
}  
?>
```

6.2 File libreria.php

```
<?php  
error_reporting(E_ALL & ~E_NOTICE);  
include "funzioni_db.php";  
//CONTROLLA DATA: ritorna true se la data e' stata inserita  
correttamente, false altrimenti  
function controlla_data($date){  
    $dt = explode('-', $date);  
    if (!empty($dt[0]) && !empty($dt[1]) && !empty($dt[2]) &&  
        strlen($date)==10) {  
        return checkdate((int) $dt[1], (int) $dt[2], (int) $dt[0]);  
    } else {  
        return false;  
    }  
}  
  
//STAMPA LIBRI ORDINATI: stampa una tabella contenente i libri nel  
database, ordinati  
function stampa_libri_ordinati($ordinamento){  
    $arr_libri = get_libri_ordinati($ordinamento);  
    echo "<table border=1><tr><th>Libro</th><th>Valutazione  
media</th></tr>";  
    foreach($arr_libri as $libro){  
        echo '<tr><th><a  
href="libro.php?libro='.$libro["id"].' ">'. $libro["titolo"].' </a></th><th>';  
        if(empty($libro["valutazione"]))  
            echo "N/A";  
        else  
            echo $libro["valutazione"];  
        echo "</th></tr>";  
    }  
    echo "</table>";  
}  
  
//STAMPA FORM CERCA: stampa il form per cercare i libri  
function stampa_form_cerca(){  
    echo' <form name="cerca" method="get"action="cerca.php">  
        Titolo: <input type="text" name="titolo">  
        <input type="submit" value="cerca">  
    </form>';  
}  
  
//STAMPA RICERCA LIBRI PER: stampa una tabella contenente le  
informazioni dei libri cercati  
function stampa_ricerca_libri_per($titolo){  
    $arr_libri = get_libri_cerca($titolo);
```

```

echo "<table border=1><tr><th>Libro</th><th>Valutazione
    media</th></tr>";
foreach($arr_libri as $libro){
    echo '<tr><th><a
        href="libro.php?libro='.$libro["id"].'>'. $libro["titolo"].'</a></th><th>';
    if(empty($libro["valutazione"]))
        echo "N/A";
    else
        echo $libro["valutazione"];
    echo "</th></tr>";
}
echo "</table>";
}

//STAMPA CHECKBOX AUTORI: stampa il form di scelta degli autori
function stampa_checkbox_autori(){
    $arr_autori=get_autori();
    //print_r($arr_autori);
    foreach($arr_autori as $aut)
        echo '<input type="checkbox" name="autori[]"
            value="'.$aut["id"].'"/>'. $aut["cognome"].'
            '. $aut["nome"].'<br>';
}

//STAMPA FORM NUOVA RECENSIONE: stampa il form per inserire una
    recensione
function stampa_form_nuova_recensione($id){
    echo ' <form name="nuova_recensione" method="post"
        action="nuova_recensione_validation.php">
        Titolo: <input type="text" name="titolo"><br>
        Testo: <textarea name="testo"></textarea><br>
        Punteggio: <select name="punteggio">
            <option value="1">1</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4">4</option>
            <option value="5">5</option>
        </select>
        <input type="hidden" name="id_libro" value="'.$id.'">
        <input type="submit" value="Inserisci">
    </form>';
}

//STAMPA INFO LIBRO: stampa informazioni su un libro (titolo,
    autori, trama, inserito da)
function stampa_info_libro($id){
    $arr_libro = get_info_libro($id);
    if(empty($arr_libro))
        echo "<font color=red>Errore: nessun libro trovato!
            :(</font>";
    else{
        echo "<p><h3>Titolo: ".$arr_libro[0]["titolo"]."</h3><br>";
        echo "Genere: ".$arr_libro[0]["genere"]."<br>";
    }
}

```

```

        echo "Autori: ";
        $arr_autori = get_autori_libro($id);
        foreach($arr_autori as $autore){
            echo $autore["cognome"]." ".$autore["nome"].",";
        }
        echo "<br>Trama: ".$arr_libro[0]["trama"];
        echo "<br>Inserito da: ".$arr_libro[0]["username"];
        if(empty($_SESSION["id_user"])){
            $_SESSION['last_page']="http://" . $_SERVER['HTTP_HOST']
                . $_SERVER['REQUEST_URI'];
            echo '<br><a href="login.php">Accedi</a> per recensire il
                libro!<br>';
        }
        else if(!gia_recensito($id, $_SESSION["id_user"])){
            if($_GET['errore']=='probleminput')
                echo "<br><font color=red>Errore: tutti i campi sono
                    obbligatori!</font>";
            echo "<br>Scrivi una recensione per questo libro!<br>";
            stampa_form_nuova_recensione($id);
        }
        else{
            echo "<br>Hai gia' recensito questo libro!<br>";
        }
        echo "<br><h4>RECENSIONI</h4>";
        $arr_recensioni = get_recensioni_libro($id);
        echo "<table border=1><tr><th>Titolo</th><th>Recensore</th>";
        foreach($arr_recensioni as $recensione)
            echo '<tr><td><a
                href="recensione.php?id='.$recensione["id_recensione"].">'. $recensione["titolo"].'
            echo "</table>";
    }
}

//STAMPA INFO RECENSIONE: stampa informazioni sulla recensione scelta
function stampa_info_recensione($id){
    $arr_recensione = get_recensione($id);
    echo "<h2>Recensione di: ".$arr_recensione["username"]."</h2>";
    echo "<h3>Titolo: ".$arr_recensione["titolo"]."</h3>";
    echo "<br>Libro: ".$arr_recensione["titolo_libro"]."<br>";
    echo "<br>Testo: ".$arr_recensione["testo"]."<br>";
    echo "<br>Voto: ".$arr_recensione["voto"]."<br>";
}

//STAMPA FORM COMMENTA RECENSIONE: stampa il form per permettere ad
    un utente di commentare una recensione
function stampa_form_commenta_recensione($id_rec){
    echo "<h4>Commenta</h4>";
    if(empty($_SESSION["id_user"])){
        $_SESSION['last_page']="http://" . $_SERVER['HTTP_HOST'] .
            $_SERVER['REQUEST_URI'];
        echo '<br><a href="login.php">Accedi</a> per commentare!<br>';
    }
}

```

```

    }
    else{
        echo '<form name="commenta_recensione" method="post"
            action="commento_validation.php">
            Testo: <textarea name="testo" rows="1"
                cols="50"></textarea>
            <input type="hidden" name="id_recensione"
                value="'. $id_rec. '">
            <input type="submit" value="Commenta">
            </form>';
    }
}
//STAMPA FORM COMMENTA COMMENTO: permette di commentare un commento,
    e di votarlo.
function stampa_form_commenta_commento($id_rec, $id_commento, $span,
    $voto){
    if(empty($_SESSION["id_user"])){
        $_SESSION['last_page']="http://" . $_SERVER['HTTP_HOST'] .
            $_SERVER['REQUEST_URI'];
        echo stampa_tab($span). '<a href="login.php">Accedi</a> per
            commentare!<br>';
    }
    else{
        echo stampa_tab($span);
        echo ' | <a
            href="vota_validation.php?commento="'. $id_commento. '&voto=1">+</a>
            / ' ;
        echo '<a
            href="vota_validation.php?commento="'. $id_commento. '&voto=0">0</a>
            / ' ;
        echo '<a
            href="vota_validation.php?commento="'. $id_commento. '&voto=-1">-</a>
            | punteggio: '. $voto. ' | ' ;
        echo '<form name="commenta_commento" method="post"
            action="commento_validation.php">
            '. stampa_tab($span). '<textarea name="testo" rows="1"
                cols="50"></textarea>
            <input type="hidden" name="id_recensione"
                value="'. $id_rec. '">
            <input type="hidden" name="id_commento_padre"
                value="'. $id_commento. '">
            <input type="submit" value="Commenta">
            </form>';
    }
}
}
//STAMPA TAB: stampa N tab in html
function stampa_tab($tab){
    $str = "";
    for($i=0; $i<$tab; $i++){
        $str = $str. "&nbsp;";
    }
}

```

```

        return $str;
    }
    //STAMPA COMMENTI COMMENTO: stampa i commenti di un commento, e i
    //commenti di ogni commento del commento padre
    function stampa_commenti_commento($id_rec,$id_comm, $span_int){
        $array_commenti = get_commenti_commento($id_comm);
        foreach($array_commenti as $commento){
            echo stampa_tab($span_int).$commento['data_commento'].' ->
                '.$commento['username'].": ".$commento['testo'].'<br>';
            echo stampa_tab($span_int);
            stampa_form_commenta_commento($id_rec,
                $commento["id_commento"], $span_int,
                $commento["punteggio"]);
            $parent_comm=$commento["id_commento"];
            stampa_commenti_commento($id_rec, $parent_comm, $span_int+1);
        }
    }
    //STAMPA COMMENTI RECENSIONE: stampa i commenti di una
    //recensione.Inoltre stampera' anche i commenti di ogni commento
    function stampa_commenti_recensione($id_rec){
        $array_commenti = get_commenti($id_rec);
        $span_int=2;
        foreach($array_commenti as $commento){
            echo stampa_tab(1).$commento['data_commento'].' ->
                '.$commento['username'].": ".$commento['testo'].'<br>';
            echo stampa_tab(1);
            stampa_form_commenta_commento($id_rec,
                $commento["id_commento"], 1, $commento["punteggio"]);
            $parent_comm=$commento["id_commento"];
            stampa_commenti_commento($id_rec, $parent_comm, $span_int);
        }
    }
}

function stampa_info_utente($id){
    $utente = get_info_utente($id);
    echo "<h2>".$utente['username']. "</h2>";
    echo "<br><em><b>Nome: </b></em> ".$utente['nome'];
    echo '<br><em><b>Cognome:</b></em>'. $utente["cognome"];
    echo '<br><em><b>Data di nascita:</b></em>
        '.$utente["data_nascita"];
    echo '<br><em><b>Email:</b></em> '. $utente["email"];
    echo "<hr>";
    echo "<h3>Le tue recensioni:</h3>";
    stampa_recensioni_utente($id);
}

function stampa_recensioni_utente($id){
    $arr_recensioni=get_recensioni_utente($id);
    $i=1;
    foreach($arr_recensioni as $recensione){

```



```

        echo "$i. <a
            href='recensione.php?id=".$recensione['id']."'>".$recensione['titolo_libro']."
            (". $recensione['titolo'].")</a><br>";
        $i++;
    }
}
?>
```

6.3 File header.php

```
<div class="header">
    <center>
        <h1>FORUM DI RECENSIONE DI LIBRI</h1>
        <a href="index.php">HOME</a>
        <?php if (!isset($_SESSION['username'])) {?>
            <a href="login.php">LOGIN</a>
            <a href="signup.php">SIGNUP</a><br>
        <?php } else {?>
            Benvenuto, <?php echo $_SESSION['username']; ?>!
            <a href="logout.php">LOGOUT</a><br>
        <?php } ?>
    </center>
</div>
```

6.4 File index.php

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<?php
    include "includes/libreria.php";
    session_start();
    $_SESSION['last_page']="http://" . $_SERVER['HTTP_HOST'] .
        $_SERVER['REQUEST_URI']; //mi permette di trovare l'ultima
        pagina (utile per il login, o per l'inserimento di un
        autore/libro)
?>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
            charset=utf-8"/>
        <link
            href="https://fonts.googleapis.com/css?family=Old+Standard+TT"
            rel="stylesheet">
        <link href="css.css" rel="stylesheet" type="text/css">
        <title>
            Progetto basi di dati
        </title>
```

```

</head>
<body>
    <?php require "header.php" ?>
    <p>
        <a href="inserisci_autore.php">Inserisci nuovo autore</a><br>
        <a href="inserisci_libro.php">Inserisci nuovo libro</a><br>
    </p>
    <hr>
    <?php
        stampa_form_cerca();
        echo "<h2>LISTA LIBRI</h2>";
        echo 'Ordina per: <a
            href="index.php?order=titolo">titolo</a>, <a
            href="index.php?order=titolodesc">titolo desc</a>, <a
            href="index.php?order=valutazione">valutazione</a>, <a
            href="index.php?order=valutazionedesc">valutazione
            desc</a>';
        if($_GET["order"]=="valutazione")
            stampa_libri_ordinati("valutazione");
        else if($_GET["order"]=="valutazionedesc")
            stampa_libri_ordinati("valutazione desc");
        else if($_GET["order"]=="titolodesc")
            stampa_libri_ordinati("titolo desc");
        else
            stampa_libri_ordinati("titolo");
    ?>
</body>
</html>

```

6.5 File cerca.php

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
    <?php session_start(); ?>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
            charset=utf-8"/>
        <link
            href="https://fonts.googleapis.com/css?family=Old+Standard+TT"
            rel="stylesheet">
        <link href="css.css" rel="stylesheet" type="text/css">
        <title>
            Progetto basi di dati
        </title>
    </head>
    <body>
        <?php
            require "header.php";

```

```

        include "includes/libreria.php";
    ?>
<h1>CERCA</h1>
<?php
    stampa_form_cerca();
    if(empty($_GET['titolo']))
        echo "<font color=red>Errore: devi inserire il titolo del
            libro che vuoi cercare!</font>";
    else
        echo "Risultati di ricerca per: ".$_GET['titolo'];
    stampa_ricerca_libri_per($_GET['titolo']);

?>
</body>
</html>

```

6.6 File login.php

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<?php
    session_start() ;
    //stampa_form_login: visualizza il form di login (username e
    password)
    function stampa_form_login(){ ?>
        <form name="login" method="post" action="login_validation.php">
            Username: <input type="text" name="username">
            Password: <input type="password" name="password">
            <input type="submit" value="login">
        </form>
    <?php } ?>

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
            charset=utf-8"/>
        <link
            href="https://fonts.googleapis.com/css?family=Old+Standard+TT"
            rel="stylesheet">
        <link href="css.css" rel="stylesheet" type="text/css">
        <title>Progetto basi di dati</title>
    </head>
    <body>
        <?php require "header.php" ?>
        <h2>LOGIN</h2>
        <?php if(!isset($_SESSION['username'])) {
            if($_GET['errore']=="problemaintput")
                echo "<p><font color=red>Attenzione: inserisci username
                    e/o password!</font></p>";
        }
    }
}

```

```

        else if($_GET['errore']=="noncorretti")
            echo "<p><font color=red>Attenzione: hai sbagliato ad
                inserire l'username e/o la password!</font></p>";
        else if($_GET['errore']=="nonpermesso")
            echo "<p><font color=red>Attenzione: funzione non
                permessa da utente ospite. Accedi!</font></p>";
        stampa_form_login();
    } else {?>
        <center>Attenzione: un utente e' gia' connesso. Devi prima
            effettuare il logout!</center>
    <?php } ?>
</body>
</html>

```

6.7 File login_validation.php

```

<?php
    session_start() ;
    require "includes/libreria.php";
    if((empty($_POST['username'])) || (empty($_POST['password'])))
        header('Location:login.php?errore=problemainput');
    else{
        /*eseguo il login:
        /*ora si cerca se esiste l'utente nel database*/
        $hashed_pwd = login($_POST['username']);
        if(password_verify($_POST['password'], $hashed_pwd)){
            header("Location: ".$_SESSION['last_page']);
            $_SESSION['username'] = $_POST['username']; // si
                inserisce il nome utente
            $_SESSION['id_user']=get_id_user($_POST['username']);
        } else {
            header('Location:login.php?errore=noncorretti');
        }
    }
?>

```

6.8 File signup.php

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<?php
    session_start() ;
    //stampa_form_signup: visualizza il form di signup
    function stampa_form_signup(){ ?>
        <form name="signup" method="post" action="signup_validation.php">
            Username: <input type="text" name="username">
            Password: <input type="password" name="password">

```

```

        Email: <input type="text" name="email">
        Data di Nascita: <input type="text" name="data_nascita">
        <br>Nome: <input type="text" name="nome">
        Cognome: <input type="text" name="cognome">
        <input type="submit" value="Registrati">
    </form>
<?php } ?>

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
            charset=utf-8"/>
        <link
            href="https://fonts.googleapis.com/css?family=Old+Standard+TT"
            rel="stylesheet">
        <link href="css.css" rel="stylesheet" type="text/css">
        <title>Progetto basi di dati</title>
    </head>
    <body>
        <?php require "header.php" ?>
        <h2>SIGNUP</h2>
        <?php if(!isset($_SESSION['username'])) {
            if($_GET['registrato']=="true")
                echo "<p><font color=darkgreen><center>Sei stato
                    registrato correttamente! Ora puoi eseguire il login
                    con i tuoi dati.</center></font></p> ";
            else{
                if($_GET['errore']=="problemainput")
                    echo "<p><font color=red>Attenzione: obbligatorio
                        inserire username, password, email!</font></p>";
                else if($_GET['errore']=="mailesistente")
                    echo "<p><font color=red>Attenzione: mail gia'
                        presente nel database!</font></p>";
                else if($_GET['errore']=="useresistente")
                    echo "<p><font color=red>Attenzione: username gia'
                        presente nel database!</font></p>";
                else if($_GET['errore']=="data")
                    echo "<p><font color=red>Attenzione: data errata!
                        Inserire la data nel seguente modo: GG-MM-AAAA
                        (es: 02-12-1975)!</font></p>";
                stampa_form_signup();
            }
        } else {?>
            <center>Attenzione: un utente e' gia' connesso. Devi prima
                effettuare il logout!</center>
        <?php } ?>
    </body>
</html>

```

6.9 File signup_validation.php

```
<?php
    session_start() ;
    require "includes/libreria.php";
    function preparapwd($password){
        return password_hash($password, PASSWORD_BCRYPT);
    }

    if(empty($_POST['username']) || empty($_POST['password']) ||
        empty($_POST['email']))
        header('Location:signup.php?errore=problemainput');
    else{
        /*controllo data*/
        if(!empty($_POST['data_nascita']) &&
            !controlla_data($_POST['data_nascita']))
            header('Location:signup.php?errore=data');
        else{
            /*controllo se esiste gia un utente con l'username scelto*/
            if(user_esiste($_POST['username']))
                header('Location:signup.php?errore=usersistente');
            else if(mail_esiste($_POST['email']))
                header('Location:signup.php?errore=mailesistente');
            else{
                if(empty($_POST["data_nascita"]))
                    esegui_registrazione($_POST['username'],preparapwd($_POST['password']),
                        $_POST['email'], $_POST['nome'],
                        $_POST['cognome']);
                else
                    esegui_registrazione_con_data($_POST['username'],preparapwd($_POST['password']),
                        $_POST['email'], $_POST['data_nascita'],
                        $_POST['nome'], $_POST['cognome']);
                header("Location:signup.php?registrato=true");
            }
        }
    }
}
?>
```

6.10 File logout.php

```
<?php
    session_start() ;
    session_destroy();
    header('Location: index.php');
?>
```

6.11 File inserisci_autore.php

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
  <?php session_start();
    if(empty($_SESSION['username'])) {
      $_SESSION['last_page'] = "http://" . $_SERVER['HTTP_HOST'] .
        $_SERVER['REQUEST_URI'];
      header("Location: login.php?errore=nonpermeso");
    }
  ?>

  <head>
    <meta http-equiv="Content-Type" content="text/html;
      charset=utf-8"/>
    <link
      href="https://fonts.googleapis.com/css?family=Old+Standard+TT"
      rel="stylesheet">
    <link href="css.css" rel="stylesheet" type="text/css">
    <title>
      Progetto basi di dati
    </title>
  </head>
  <body>
    <?php require "header.php" ?>
    <h1>Inserisci autore</h1>
    <?php
      if($_GET["errore"]=="problemaintput")
        echo "<font color=red>Errore: nome e cognome
          obbligatori!</font>";
      else if($_GET["errore"]=="problemadata")
        echo "<font color=red>Errore: inserisci la data
          correttamente!</font>";
      else if($_GET["inserito"]=="true")
        echo "<font color=darkgreen>Autore inserito
          correttamente!</font>";
    ?>
    <form name="inserisci_autore" method="post"
      action="inserisci_autore_validation.php">
      Nome: <input type="text" name="nome"><br>
      Cognome: <input type="text" name="cognome"><br>
      Data di Nascita (AAAA-MM-GG) : <input type="text"
        name="data_nascita"><br>
      Nazionalita': <input type="text" name="nazionalita"><br>
      <input type="submit" value="Inserisci">
    </form>
  </body>
</html>
```

6.12 File inserisci_autore_validation.php

```
<?php
    session_start() ;
    require "includes/libreria.php";

    if(empty($_POST['nome']) && empty($_POST['cognome']))
        header('Location:inserisci_autore.php?errore=problemainput');
    else if(!empty($_POST['data_nascita']) &&
        !controlla_data($_POST['data_nascita']))
        header('Location:inserisci_autore.php?errore=problemadata');
    else{
        ins_autore($_POST['nome'], $_POST['cognome'],
            $_POST['nazionalita'], $_POST['data_nascita'],
            $_SESSION['id_user']);
        header('Location:inserisci_autore.php?inserito=true');
    }
?>
```

6.13 File inserisci_libro.php

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
    <?php session_start();
    include "includes/libreria.php";
    if(empty($_SESSION['username'])) {
        $_SESSION['last_page']="http://" . $_SERVER['HTTP_HOST'] .
            $_SERVER['REQUEST_URI'];
        header("Location: login.php?errore=nonpermesso");
    }
?>

    <head>
        <meta http-equiv="Content-Type" content="text/html;
            charset=utf-8"/>

        <link
            href="https://fonts.googleapis.com/css?family=Old+Standard+TT"
            rel="stylesheet">
        <link href="css.css" rel="stylesheet" type="text/css">
        <title>
            Progetto basi di dati
        </title>
    </head>
    <body>
        <?php
            require "header.php";
```



```

        $_SESSION['last_page']="http://" . $_SERVER['HTTP_HOST'] .
        $_SERVER['REQUEST_URI'];
    ?>
<h1>Inserisci libro</h1>
<?php
    if($_GET['errore']=="problemainput")
        echo "<font color=red>Errore: titolo
            obbligatorio!</font>";
    if($_GET['inserito']=="true")
        echo"<font color=darkgreen>Libro inserito
            correttamente!</font>";
    if($_GET['errore']=="problemaaautore")
        echo "<font color=red>Attenzione: obbligatorio
            selezionare almeno 1 autore!</font>";
    ?>
    <form name="inserisci_libro" method="post"
        action="inserisci_libro_validation.php">
        Titolo: <input type="text" name="titolo"><br>
        Genere: <input type="text" name="genere"><br>
        Trama: <textarea name="trama"></textarea><br>
        Autori: <br>
        <a href="inserisci_autore.php" >Inserisci un autore</a>
        <br>
        <?php stampa_checkbox_autori(); ?>
        <input type="submit" value="Inserisci">
    </form>
</body>
</html>

```

6.14 File inserisci_libro_validation.php

```

<?php
    session_start() ;
    require "includes/libreria.php";

    if(empty($_POST['titolo']))
        header('Location:inserisci_libro.php?errore=problemainput');
    else if(!isset($_POST['autori']))
        header('Location:inserisci_libro.php?errore=problemaaautore');
    else{
        ins_libro($_POST['titolo'], $_POST['trama'], $_POST['genere'],
            $_SESSION['id_user'], $_POST['autori']);
        header('Location:inserisci_libro.php?inserito=true');
    }
    ?>

```

6.15 File libro.php

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<?php
    include "includes/libreria.php";
    session_start();
    $_SESSION['last_page']="http://" . $_SERVER['HTTP_HOST'] .
        $_SERVER['REQUEST_URI'];
?>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
            charset=utf-8"/>
        <link
            href="https://fonts.googleapis.com/css?family=Old+Standard+TT"
            rel="stylesheet">
        <link href="css.css" rel="stylesheet" type="text/css">
        <title>
            Progetto basi di dati
        </title>
    </head>
    <body>
        <?php require "header.php" ?>
        <?php
            if(empty($_GET["libro"]))
                echo "<font color=red>Errore: devi prima selezionare un
                    libro!</font>";
            else{
                echo "<h2>LIBRO</h2>";
                stampa_info_libro($_GET["libro"]);
            }
        ?>
    </body>
</html>
```

6.16 File recensione.php

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<?php
    include "includes/libreria.php";
    session_start();
    $_SESSION['last_page']="http://" . $_SERVER['HTTP_HOST'] .
        $_SERVER['REQUEST_URI'];
?>
<html>
    <head>
```

```

<meta http-equiv="Content-Type" content="text/html;
    charset=utf-8"/>
<link
    href="https://fonts.googleapis.com/css?family=Old+Standard+TT"
    rel="stylesheet">
<link href="css.css" rel="stylesheet" type="text/css">
<title>
    Progetto basi di dati
</title>
</head>
<body>
    <?php require "header.php" ?>
    <?php
        if(empty($_GET["id"]))
            echo "<font color=red>Errore: devi prima selezionare una
                recensione!</font>";
        else{
            if($_GET['errore']=='notesto')
                echo "<font color=red>Errore: un commento non puo'
                    essere vuoto!</font>";
            stampa_info_recensione($_GET["id"]);
            stampa_form_commenta_recensione($_GET["id"]);
            echo "<h4>Commenti degli utenti</h4>";
            stampa_commenti_recensione($_GET["id"]);
        }
    ?>
</body>
</html>

```

6.17 File nuova_recensione_validation.php

```

<?php
    session_start() ;
    require "includes/libreria.php";
    if(empty($_POST['titolo']) || empty($_POST['testo']))
        header('Location:libro.php?libro='.$_POST['id_libro'].'&errore=problemainput');
    else{
        /*inserisco la recensione nel database.*/
        inserisci_recensione($_POST['titolo'], $_POST['testo'],
            $_POST['punteggio'], $_SESSION['id_user'],
            $_POST['id_libro']);
        header('Location:libro.php?libro='.$_POST['id_libro']);
    }
?>

```

6.18 File commento_validation.php

```
<?php
    session_start();
    include "includes/libreria.php";
    if(empty($_POST['id_recensione']))
        header("Location: index.php");
    else if(empty($_POST['testo']))
        header("Location:
                recensione.php?id=".$_POST['id_recensione']."&errore=notesto");
    else{
        header("Location: recensione.php?id=".$_POST['id_recensione']);
        /*calcolo la data corrente ed inserisco il commento*/
        $data = date('Y-m-d h:i:s', time());
        inserisci_commento($_POST["testo"], $_SESSION["id_user"],
                           $_POST["id_recensione"], $_POST["id_commento_padre"], $data);
    }
?>
```

6.19 File commento_validation.php

```
<?php
    session_start();
    include "includes/libreria.php";
    if(empty($_POST['id_recensione']))
        header("Location: index.php");
    else if(empty($_POST['testo']))
        header("Location:
                recensione.php?id=".$_POST['id_recensione']."&errore=notesto");
    else{
        header("Location: recensione.php?id=".$_POST['id_recensione']);
        /*calcolo la data corrente ed inserisco il commento*/
        $data = date('Y-m-d h:i:s', time());
        inserisci_commento($_POST["testo"], $_SESSION["id_user"],
                           $_POST["id_recensione"], $_POST["id_commento_padre"], $data);
    }
?>
```

6.20 File utente.php

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<?php
    include "includes/libreria.php";
    session_start();
```

```

$_SESSION['last_page']="http://" . $_SERVER['HTTP_HOST'] .
    $_SERVER['REQUEST_URI'];
if(empty($_GET["id"]))
    header("Location: index.php");
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
    charset=utf-8"/>
<link
    href="https://fonts.googleapis.com/css?family=Old+Standard+TT"
    rel="stylesheet">
<link href="css.css" rel="stylesheet" type="text/css">
<title>
    Progetto basi di dati
</title>
</head>
<body bgcolor="FFFFFF3">
    <?php require "header.php" ?>
    <?php
        stampa_info_utente($_GET['id']);
    ?>
</body>
</html>

```

6.21 File css.css

```

body {
    font-family: 'Old Standard TT', serif;
    margin: 20px 8% 100px 8%;
}
#testo {
    margin-right: 50%;
}

```

7 Diagramma della struttura dell'applicazione WEB

