```
#include "stdafx.h"
#include <conio.h>

/* Copyright 2008, 2010, Oracle and/or its affiliates. All rights reserved.

/* Standard C++ includes */
#include <stdlib.h>
#include <iostream>
#include <stdio.h>

/*
  Include directly the different
  headers from cppconn/ and mysql_driver.h + mysql_util.h
  (and mysql_connection.h). This will reduce your build time!
*/
#include "mysql_connection.h"

#include <cppconn/driver.h>
#include <cppconn/exception.h>
#include <cppconn/resultset.h>
#include <cppconn/statement.h>
#include <cppconn/prepared_statement.h>



using namespace std;

sql::Driver *driver;
  sql::Connection *con;
  sql::Statement *stmt;
  sql::ResultSet *res;
  sql::PreparedStatement *pstmt;

void options();

void search_by_movies ();

void sign_up () ;

void sign_in () ;

void insert_movies () ;

void review_movies () ;

void search_movies() ;

void rating_movies() ;

bool check_rating (int a) ;

void search_by_actor() ;

void search_by_actress() ;

void search_by_director() ;

void search_by_year() ;
void the_main();

//void add_to_favourites ();

string username_str;

void search_movies()
{
       char q;
       system ("cls");
       cout<<"    \nThanks for using iitMdB to search for your favourite
movies!!!!!!"<<endl;
       cout<<endl;


       cout<<"1.Hit 'M' to search by the movie name\n2.Hit 'A' to search by the actor
name\n3.Hit 'S' to search by the actress name\n4.Hit 'D' to search by the director's
name\n5.Hit 'Y' to search by the yeaar of relase"<<endl;
       cin>>q;
       system ("cls") ;
       switch(q)
       {
       case 'M': search_by_movies();
              break;
       case 'A':search_by_actor();
              break;
       case 'S':{search_by_actress();
                      break;}
       case 'D':{search_by_director();
                      break;}
       case 'Y':{search_by_year();
                      break;}
        default:{cout<<"\nInvalid Entry";
                                   search_movies();
                      }
       }

}

void sign_up ()
{       int p ;
   string i, j, l, k ;

   do
   {
       cout << "\nEnter the username (DataBase will check its avaibility) : " ;
       cin >> j;
       pstmt = con->prepareStatement("SELECT * FROM users where username=(?)");
  pstmt->setString(1, j);
  res = pstmt->executeQuery();
   p= res->rowsCount();
  if (p==1)
  {
         cout <<"Oops this id is already taken! ... Please enter another id"<<endl;
  }
 }while (p==1);
```

```
do
{
   cout << " \n Enter login password : " ;
   cin >> i ;
   cout <<  "\n Re-Enter password : "  ;
   cin >> k  ;

 if (i!=k)
 {
        cout << "Re-Enter same password ";
 }
}
while(i!=k);

   pstmt = con->prepareStatement("INSERT INTO users(username,password) VALUES
((?),(?))");
  pstmt->setString(1, j);
  pstmt->setString(2, k);
  pstmt->executeUpdate();
  system("cls");
  sign_in();

 }
void options()
 {
        cout << "\nEnter 1 to INSERT MOVIES" <<endl;
  cout << "Enter 2 to SEARCH MOVIES " << endl;
  cout << "Enter 3 to REVIEW MOVIES " << endl;
  //cout << "Enter 5 to ADD YOUR FAVOURITE MOVIES TO YOU FAVOURITES LIST " << endl;

   int w;

       cin >> w;

       system("cls");

  switch(w)
  {
  case 1 : insert_movies();
         break;
  case 2 : search_movies();
         break;
  case 3 : review_movies();
         break;
 // case 5 : add_to_favourates();
  default : cout << "\nInvalid Entry!!" ;
              options();

  }
}


void sign_in ()
{   int p ;
       string j,k,y,q;

       cout << "\nWelcome to iitMdB !!!";
   do
   {
       cout << "\nEnter the your username : " ;
       cin >> j;
       try{
       pstmt = con->prepareStatement("SELECT * FROM users where username=(?)");
  pstmt->setString(1, j);
  res = pstmt->executeQuery();
   p= res->rowsCount();
   res->next();
   q=res->getString("password");
       }
       catch (sql::SQLException &e) {
         cout << "# ERR: SQLException in " << __FILE__;
         cout << "(" << __FUNCTION__ << ") on line "  <<__LINE__  << endl;
         cout << "# ERR: " << e.what();
         cout << " (MySQL error code: " << e.getErrorCode();
         cout << ", SQLState: " << e.getSQLState() << " )" << endl;
       }

  if (p==0)
  {
         cout <<"\nThis username does not exist !! Please enter correct user id"<<endl;
  }
 }while (p==0);

    do
   {
       cout << "Enter your login password : " ;
       cin >> y;

  if (y!=q)
  {
         cout <<"\nInvalid Password... Enter correct Password "<<endl;
  }
 }while (y!=q);

   username_str=j;

   options();
}



bool check_rating(int b)
{
    if (b<1 || b>10)
    {
        cout << "Oops!! wrong rating \nPlease rate between 1-10 " << endl;
        return 1;
    }
    return 0 ;
}


void review_movies()
{
       int x,m,i;
       string j, k, l ;


   cout << "\nEnter the movie name which you want to review :" ;
    cin.ignore();
       getline(cin, k);//cin>>k
       pstmt = con->prepareStatement("SELECT * FROM movies where name like (?)");
  pstmt->setString(1, "%"+k+"%");
  res = pstmt->executeQuery();

  cout << "\nNo. Of movies by related search in our DataBase : " << res->rowsCount() <<
endl;

  while (res->next())
  {

       cout << "\n The Movie you are looking for reviewing may be : "<< res-
>getString("name")<<endl;
       //looks good if we write the name of the movie before giving its desciption

       cout << "S No. in the list " << res->getString(1) << endl;
       cout<<"___________________________________________________________________________
"<<endl;

  }



  cout << "Enter S No. from the given movie to review : " ;
       cin>>x;
   pstmt = con->prepareStatement("SELECT * from review where username =(?) and moviesno
=(?)");
  pstmt->setInt(2, x);
  pstmt->setString(1, username_str);
  res = pstmt->executeQuery();
  int count = res->rowsCount();
   //select statement from review table where username=username_str and movie_sno=x;


  if(count == 0){
         cout << "Enter the review : " << endl;
         cin.ignore();
         getline(cin, l);

         cout << "Enter the rating : " << endl;
         cin >> m;
         pstmt = con->prepareStatement("INSERT INTO
review(moviesno,username,review,rating) VALUES ((?),(?),(?),(?))");
         pstmt->setInt(1, x);
         pstmt->setString(2, username_str);
         pstmt->setString(3, l);
         pstmt->setInt(4, m);
         pstmt->executeUpdate();


  }
  else{
         cout << "You have already reviewed and rated "<< endl;
         cout<< "Your rating ";
         res->next();
         cout<< res->getInt("rating")<< endl;//rating is a integer
         cout<< "Your review ";
         cout<< res->getString("review");

         cout <<"If you want to still update enter 1 "<<endl;
         cin >> i;
         if(i==1)
         {
                //do you want to update
         pstmt = con->prepareStatement("UPDATE review set review=(?), rating=(?) where
moviesno=(?) and username=(?)");
         pstmt->setInt(1, x);
         pstmt->setString(2, username_str);
         pstmt->setString(3, l);
         pstmt->executeUpdate();
         }
         else
         {
                options();
         }

  }
system ("cls");
options();
}

void insert_movies()
{
       string i,j,k,l;
       int p;
       cout << "\n\nEnter the Movie Name or enter '0' to escape: " ;
       cin >> i;

       if(i=="0")
       {
              system ("cls");
              options();

       }

       system("cls");

       cout << "\n\nEnter Actor's Name enter '0' to escape: : " ;
       cin>> j;

       system("cls");

       if(j=="0")
       {
              system ("cls");
              options();

       }

       cout << "\n\nEnter Actress's Name : " ;
       cin>> k;

       system("cls");

       cout << "\n\nEnter Director's Name : " ;
       cin>> l;

       system("cls");

       cout << "\n\nEnter the Movie's Release year : " ;
       cin >> p;

       system("cls");

        pstmt = con->prepareStatement("INSERT INTO 
movies(name,actor,actress,director,year) VALUES ((?),(?),(?),(?),(?))");
  pstmt->setString(1, i);
  pstmt->setString(2, j);
  pstmt->setString(3, k);
  pstmt->setString(4, l);
  pstmt->setInt(5, p);
  pstmt->executeUpdate();

  cout << "\n\nYour information is recorded in the database" <<endl;

}

 void search_by_movies ()
 {
        string j, k;
        int o, avg_rating;
   cout << "\nEnter the movie name which you want to search :" ;
   //cin >> j ;
   cin.ignore();
       getline(cin, j);
       pstmt = con->prepareStatement("SELECT * FROM movies where name like (?)");
  pstmt->setString(1, "%"+j+"%");
  res = pstmt->executeQuery();

  cout << "\nNo. Of movies by related search in our DataBase : " << res->rowsCount() << 
endl;

  while (res->next()) {

       cout << "\n The Movie you are looking for may be : "<< res-
>getString("name")<<endl;
       //looks good if we write the name of the movie before giving its desciption

    cout << "\n Year of release is : ";
    /* Accessing column data by column name */
       cout << res->getString("year")<< endl;

       cout<<"\n Directed by : "<< res->getString("director")<<endl;
       /* Access column data by alias or column name */

       cout << "\n Starring : " << res->getString("actor") << " , " << res-
>getString("actress") << endl;
```

```
  sql::ResultSet *res2;
  sql::PreparedStatement *pstmt2;

       j=res->getString(1);
       o=res->getInt(1);

       pstmt2 = con->prepareStatement("SELECT AVG(rating) as avg_rating FROM review where
moviesno=(?)");
  pstmt2->setInt(1, o);
  res2 = pstmt2->executeQuery();
  res2->next();

  avg_rating = res2->getInt("avg_rating");

  cout << "Rating:" << avg_rating << endl;

  sql::ResultSet *res3;
  sql::PreparedStatement *pstmt3;

  pstmt3 = con->prepareStatement("SELECT * FROM review where moviesno=(?)");
  pstmt3->setInt(1, o);
  res3 = pstmt3->executeQuery();
  int i=0;
   while (res3->next()) {
          i++;

         string review = res3->getString("review");

  cout << "Review " << i <<":" << review << endl;

   }
       cout << "\nS No. in the list " << res->getString(1) << endl;

       cout<<"___________________________________________________________________________
"<<endl;

  }
  pstmt = con->prepareStatement("SELECT * FROM movies where name=(?)");
  pstmt->setString(1, j);
  res = pstmt->executeQuery();

   char t;
  cout << "\nEnter any KEY to go back to search page or '0' to go back to MAIN PAGE";
  cin>>t;
  if(t=='0')
  {the_main();}
  else
  {  search_movies();}
 }


  void search_by_actor ()
 {
        string j, k;
        int o, avg_rating;
   cout << "\nEnter the actor name which you want to search :" ;
   //cin >> j ;
   cin.ignore();
```

```
       getline(cin, j);
       pstmt = con->prepareStatement("SELECT * FROM movies where actor like (?)");
  pstmt->setString(1, "%"+j+"%");
  res = pstmt->executeQuery();

  cout << "\nNo. Of movies by related search in our DataBase : " << res->rowsCount() <<
endl;

  while (res->next()) {

       cout << "\n The Movie you are looking for may be : "<< res-
>getString("name")<<endl;
       //looks good if we write the name of the movie before giving its desciption

    cout << "\n Year of release is : ";
    /* Accessing column data by column name */
       cout << res->getString("year")<< endl;

       cout<<"\n Directed by : "<< res->getString("director")<<endl;
       /* Access column data by alias or column name */

       cout << "\n Starring : " << res->getString("actor") << " , " << res-
>getString("actress") << endl;

  sql::ResultSet *res2;
  sql::PreparedStatement *pstmt2;

       j=res->getString(1);
       o=res->getInt(1);

       pstmt2 = con->prepareStatement("SELECT AVG(rating) as avg_rating FROM review where
moviesno=(?)");
  pstmt2->setInt(1, o);
  res2 = pstmt2->executeQuery();
  res2->next();

  avg_rating = res2->getInt("avg_rating");

  cout << "Rating:" << avg_rating << endl;

  sql::ResultSet *res3;
  sql::PreparedStatement *pstmt3;

  pstmt3 = con->prepareStatement("SELECT * FROM review where moviesno=(?)");
  pstmt3->setInt(1, o);
  res3 = pstmt3->executeQuery();
  int i=0;
   while (res3->next()) {
          i++;

         string review = res3->getString("review");

  cout << "Review " << i <<":" << review << endl;

   }
       cout << "\nS No. in the list " << res->getString(1) << endl;
       cout<<"___________________________________________________________________________
"<<endl;

  }
  pstmt = con->prepareStatement("SELECT * FROM movies where name=(?)");
  pstmt->setString(1, j);
  res = pstmt->executeQuery();
  char t;
  cout << "\nEnter any KEY to go back to search page or '0' to go back to MAIN PAGE";
  cin>>t;
  if(t=='0')
  {the_main();}
  else
  {  search_movies();}
  }

   void search_by_actress ()
{
        string j, k;
        int o, avg_rating;
   cout << "\nEnter the actress name which you want to search :" ;
   //cin >> j ;
   cin.ignore();
       getline(cin, j);
       pstmt = con->prepareStatement("SELECT * FROM movies where actress like (?)");
  pstmt->setString(1, "%"+j+"%");
  res = pstmt->executeQuery();

  cout << "\nNo. Of movies by related search in our DataBase : " << res->rowsCount() <<
endl;

  while (res->next()) {

       cout << "\n The Movie you are looking for may be : "<< res-
>getString("name")<<endl;
       //looks good if we write the name of the movie before giving its desciption

    cout << "\n Year of release is : ";
    /* Accessing column data by column name */
       cout << res->getString("year")<< endl;

       cout<<"\n Directed by : "<< res->getString("director")<<endl;
       /* Access column data by alias or column name */

       cout << "\n Starring : " << res->getString("actor") << " , " << res-
>getString("actress") << endl;

  sql::ResultSet *res2;
  sql::PreparedStatement *pstmt2;

       j=res->getString(1);
       o=res->getInt(1);

       pstmt2 = con->prepareStatement("SELECT AVG(rating) as avg_rating FROM review where
moviesno=(?)");
  pstmt2->setInt(1, o);
  res2 = pstmt2->executeQuery();
  res2->next();
  avg_rating = res2->getInt("avg_rating");

  cout << "Rating:" << avg_rating << endl;

  sql::ResultSet *res3;
  sql::PreparedStatement *pstmt3;

  pstmt3 = con->prepareStatement("SELECT * FROM review where moviesno=(?)");
  pstmt3->setInt(1, o);
  res3 = pstmt3->executeQuery();
  int i=0;
   while (res3->next()) {
          i++;

         string review = res3->getString("review");

  cout << "Review " << i <<":" << review << endl;

   }
       cout << "\nS No. in the list " << res->getString(1) << endl;

       cout<<"___________________________________________________________________________
"<<endl;

  }
  pstmt = con->prepareStatement("SELECT * FROM movies where name=(?)");
  pstmt->setString(1, j);
  res = pstmt->executeQuery();
  char t;
  cout << "\nEnter any KEY to go back to search page or '0' to go back to MAIN PAGE";
  cin>>t;
  if(t=='0')
  {the_main();}
  else
  {  search_movies();}
  }


void search_by_director()
 {
        string j, k;
        int o, avg_rating;
   cout << "\nEnter the director name which you want to search :" ;
   //cin >> j ;
   cin.ignore();
       getline(cin, j);
       pstmt = con->prepareStatement("SELECT * FROM movies where director like (?)");
  pstmt->setString(1, "%"+j+"%");
  res = pstmt->executeQuery();

  cout << "\nNo. Of movies by related search in our DataBase : " << res->rowsCount() <<
endl;

  while (res->next()) {

       cout << "\n The Movie you are looking for may be : "<< res-
>getString("name")<<endl;
```

```
       //looks good if we write the name of the movie before giving its desciption

    cout << "\n Year of release is : ";
    /* Accessing column data by column name */
       cout << res->getString("year")<< endl;

       cout<<"\n Directed by : "<< res->getString("director")<<endl;
       /* Access column data by alias or column name */

       cout << "\n Starring : " << res->getString("actor") << " , " << res-
>getString("actress") << endl;

  sql::ResultSet *res2;
  sql::PreparedStatement *pstmt2;

       j=res->getString(1);
       o=res->getInt(1);

       pstmt2 = con->prepareStatement("SELECT AVG(rating) as avg_rating FROM review where
moviesno=(?)");
  pstmt2->setInt(1, o);
  res2 = pstmt2->executeQuery();
  res2->next();

  avg_rating = res2->getInt("avg_rating");

  cout << "Rating:" << avg_rating << endl;

  sql::ResultSet *res3;
  sql::PreparedStatement *pstmt3;

  pstmt3 = con->prepareStatement("SELECT * FROM review where moviesno=(?)");
  pstmt3->setInt(1, o);
  res3 = pstmt3->executeQuery();
  int i=0;
   while (res3->next()) {
          i++;

         string review = res3->getString("review");

  cout << "Review " << i <<":" << review << endl;

   }
       cout << "\nS No. in the list " << res->getString(1) << endl;

       cout<<"___________________________________________________________________________
"<<endl;

  }
  pstmt = con->prepareStatement("SELECT * FROM movies where name=(?)");
  pstmt->setString(1, j);
  res = pstmt->executeQuery();
  char t;
  cout << "\nEnter any KEY to go back to search page or '0' to go back to MAIN PAGE";
  cin>>t;
  if(t=='0')
  {the_main();}
  else
  {  search_movies();}
  }


void search_by_year()
{
       int j;
        string  k;
        int o, avg_rating;
   cout << "\nEnter the director name which you want to search :" ;
   cin >> j ;
       pstmt = con->prepareStatement("SELECT * FROM movies where year like (?)");
  pstmt->setInt(1, j);
  res = pstmt->executeQuery();

  cout << "\nNo. Of movies by related search in our DataBase : " << res->rowsCount() <<
endl;

  while (res->next()) {

       cout << "\n The Movie you are looking for may be : "<< res-
>getString("name")<<endl;
       //looks good if we write the name of the movie before giving its desciption

    cout << "\n Year of release is : ";
    /* Accessing column data by column name */
       cout << res->getString("year")<< endl;

       cout<<"\n Directed by : "<< res->getString("director")<<endl;
       /* Access column data by alias or column name */

       cout << "\n Starring : " << res->getString("actor") << " , " << res-
>getString("actress") << endl;

  sql::ResultSet *res2;
  sql::PreparedStatement *pstmt2;

       j=res->getInt(1);
       o=res->getInt(1);

       pstmt2 = con->prepareStatement("SELECT AVG(rating) as avg_rating FROM review where
moviesno=(?)");
  pstmt2->setInt(1, o);
  res2 = pstmt2->executeQuery();
  res2->next();

  avg_rating = res2->getInt("avg_rating");

  cout << "Rating:" << avg_rating << endl;

  sql::ResultSet *res3;
  sql::PreparedStatement *pstmt3;

  pstmt3 = con->prepareStatement("SELECT * FROM review where moviesno=(?)");
  pstmt3->setInt(1, o);
  res3 = pstmt3->executeQuery();
  int i=0;
   while (res3->next()) {
          i++;

         string review = res3->getString("review");

  cout << "Review " << i <<":" << review << endl;

   }
       cout << "\nS No. in the list " << res->getString(1) << endl;

       cout<<"___________________________________________________________________________
"<<endl;

  }
  pstmt = con->prepareStatement("SELECT * FROM movies where name=(?)");
  pstmt->setInt(1, j);
  res = pstmt->executeQuery();
  char t;
  cout << "\nEnter any KEY to go back to search page or '0' to go back to MAIN PAGE";
  cin>>t;
  if(t=='0')
  {the_main();}
  else
  {  search_movies();}
}

void the_main()
{
        cout << "Are you an existing user ?? Wanna SIGN IN? "<< endl;
  cout << "New User ?? Want to SIGN UP ??" << endl;
  cout << "Or would u like to just explore ou DataBase (by just searching movies)  "
<<endl ;

  cout << "\nEnter 1 to SIGN IN" <<endl;
  cout << "Enter 2 to SIGN UP" <<endl;
  cout << "Enter 3 to just use DataBase As a Guest User" << endl;
  int j;
  cin >> j;

  system("cls");

  switch(j)
  {
  case 1 : sign_in();
         break;
  case 2 : sign_up();
         break;
  case 3 : search_movies();
         break;
  }
}


int main(void)
{
cout << endl;
//cout << "Running 'SELECT 'Hello World!' AS _message'..." << endl;

try {
  /* Create a connection */
  driver = get_driver_instance();
  con = driver->connect("tcp://127.0.0.1:3306", "root", "root");
  /* Connect to the MySQL test database */
  con->setSchema("iitmdb");

 the_main();


  //cout << "Enter 1 to search by movies " << endl;
  //cout << "Enter 2 to search by actor " << endl;
  //cout << "Enter 3 to search by actress " << endl;
  //cout << "Enter 4 to search by year " << endl;

  delete res;
  //delete stmt;
  delete pstmt;
  delete con;
  getch();

} catch (sql::SQLException &e) {
  cout << "# ERR: SQLException in " << __FILE__;
  cout << "(" << __FUNCTION__ << ") on line "  <<__LINE__  << endl;
  cout << "# ERR: " << e.what();
  cout << " (MySQL error code: " << e.getErrorCode();
  cout << ", SQLState: " << e.getSQLState() << " )" << endl;
}

cout << endl;

return EXIT_SUCCESS;
}
```