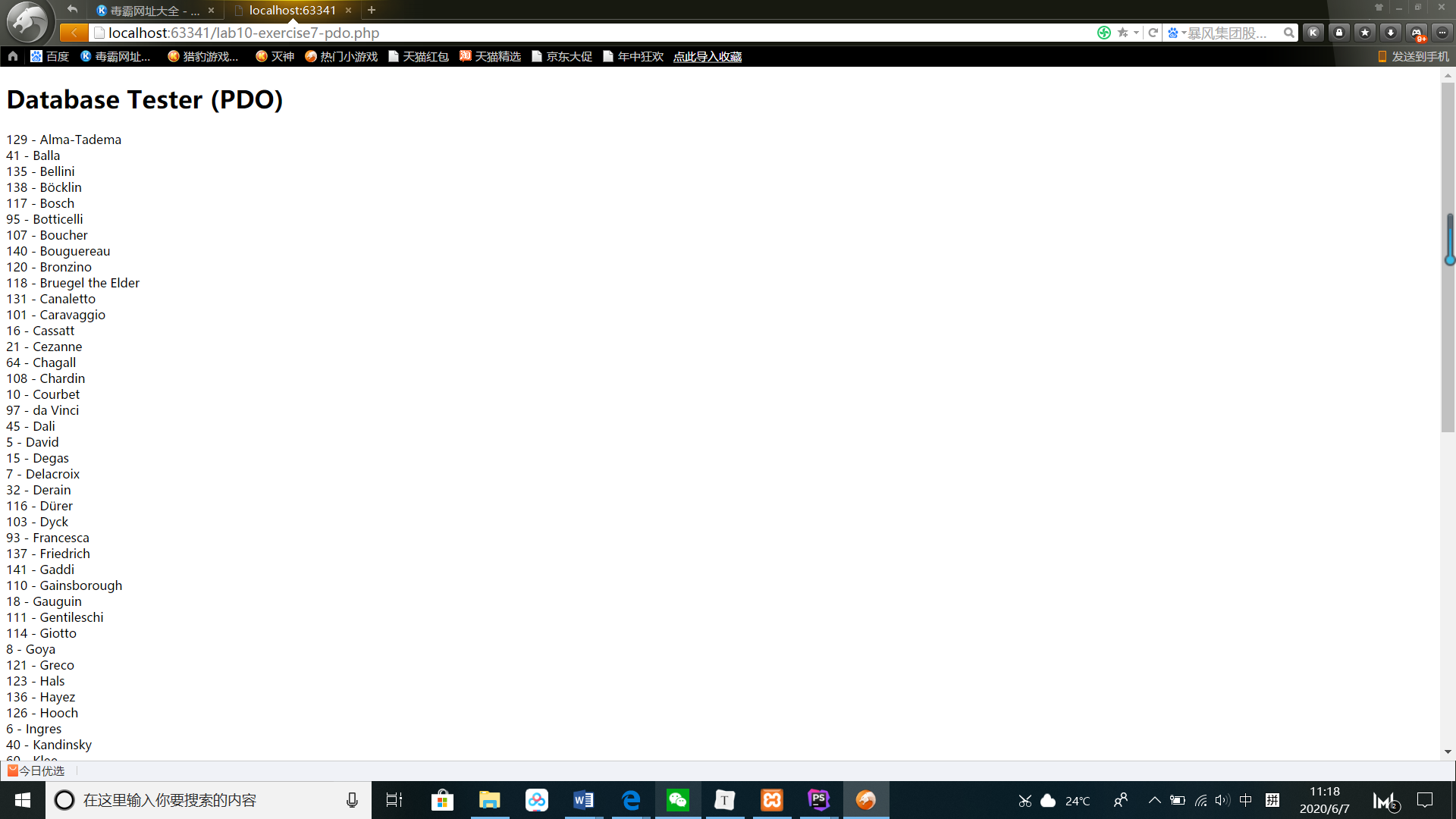
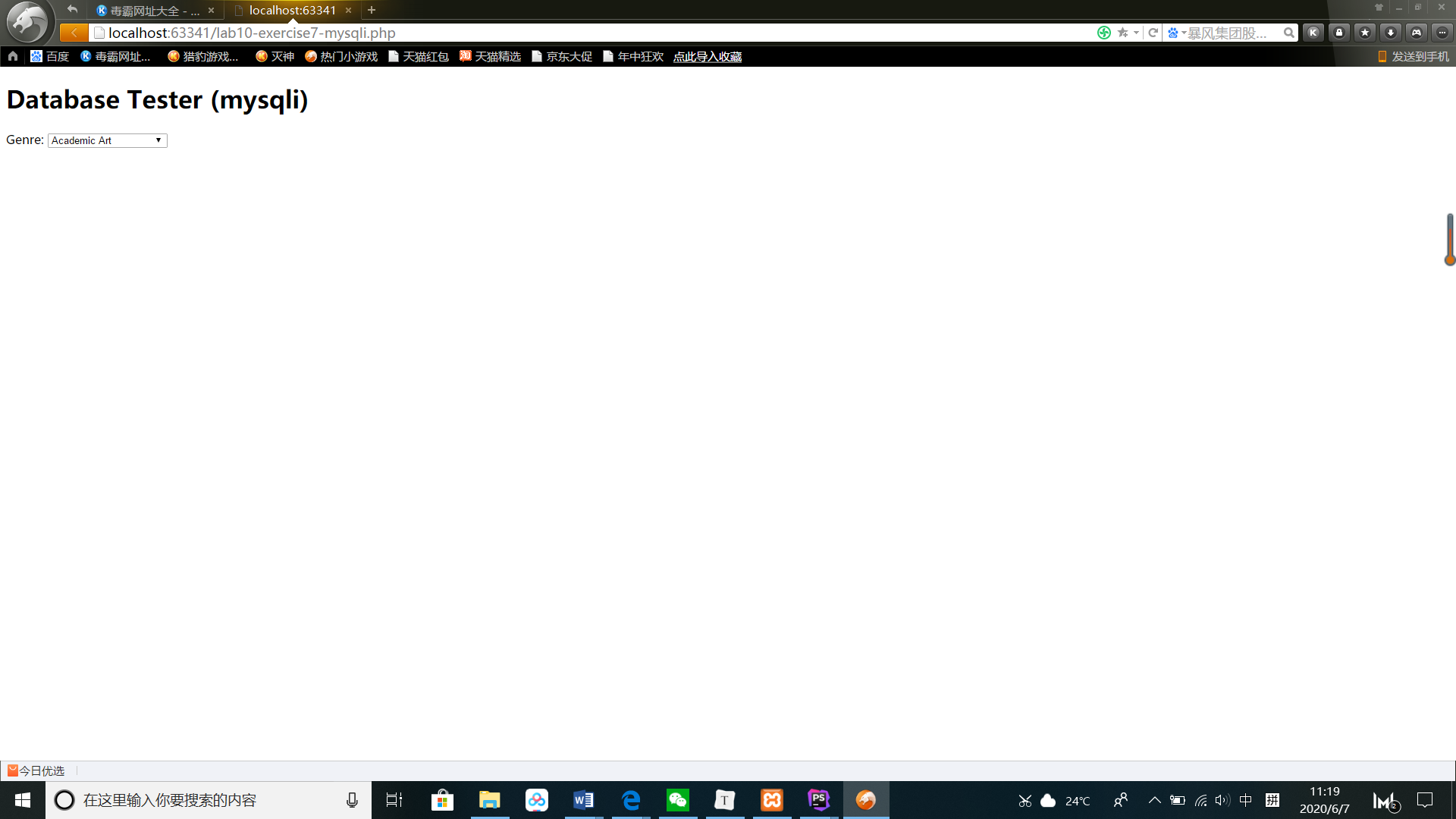
**Exercise 7**





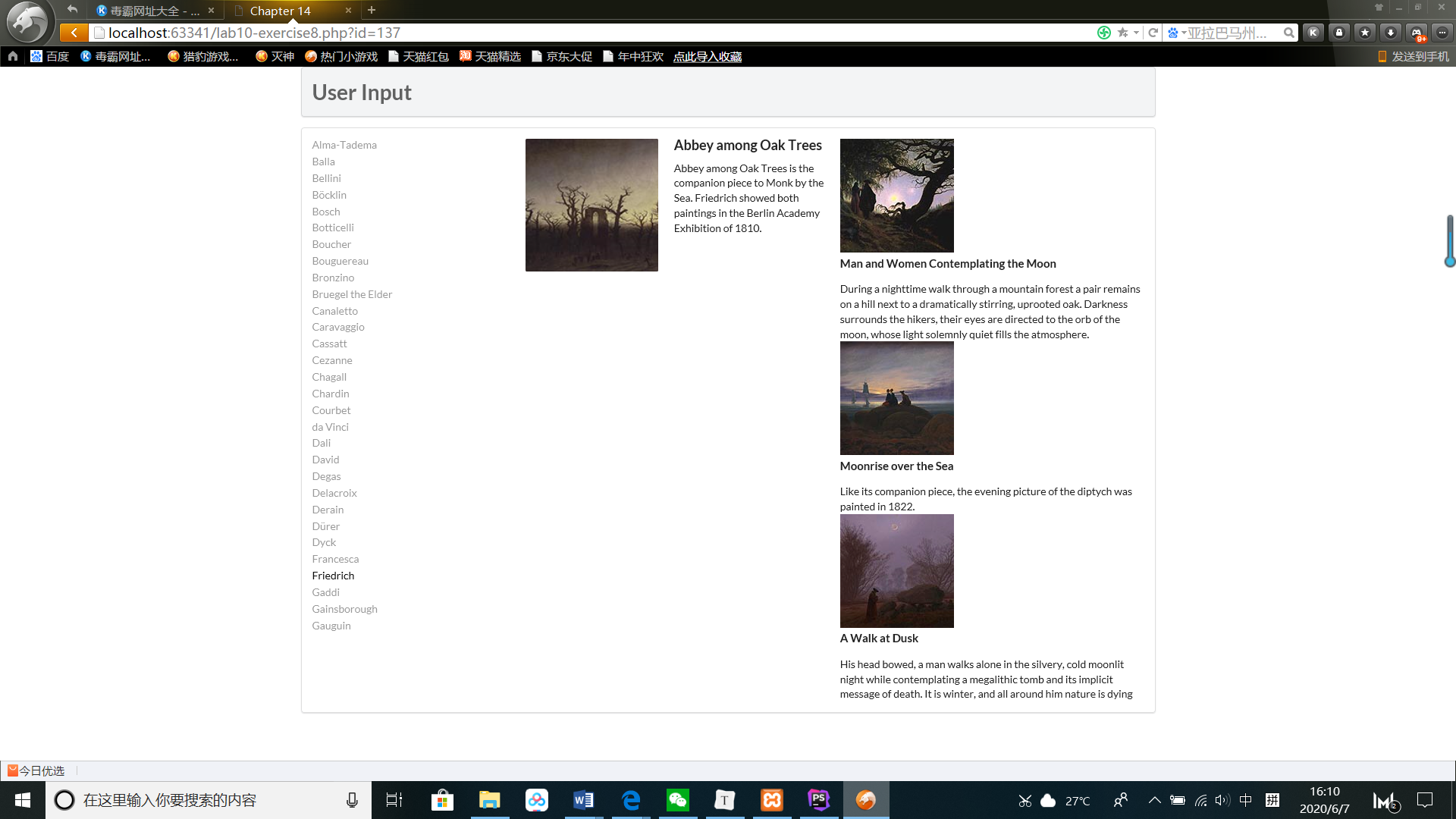
Config.php定义了用户，包括用户名和密码

lab10-exercise7-pdo.php功能是将艺术家按 last name 排序。首先创建了pdo类的实例，sql为排序方法，result 是 pdo 按 sql 排序后的结果数组。通过循环打印的方式打印。

lab10-exercise7-mysqli.php通过mysqli和数据库连接，通过mysqli\_connect函数创建connection连接，sql排序条件是将Genres中的全部按GenreName排序，result是筛选后结果，循环的方式加入到select的option，最后删除result，关闭连接。

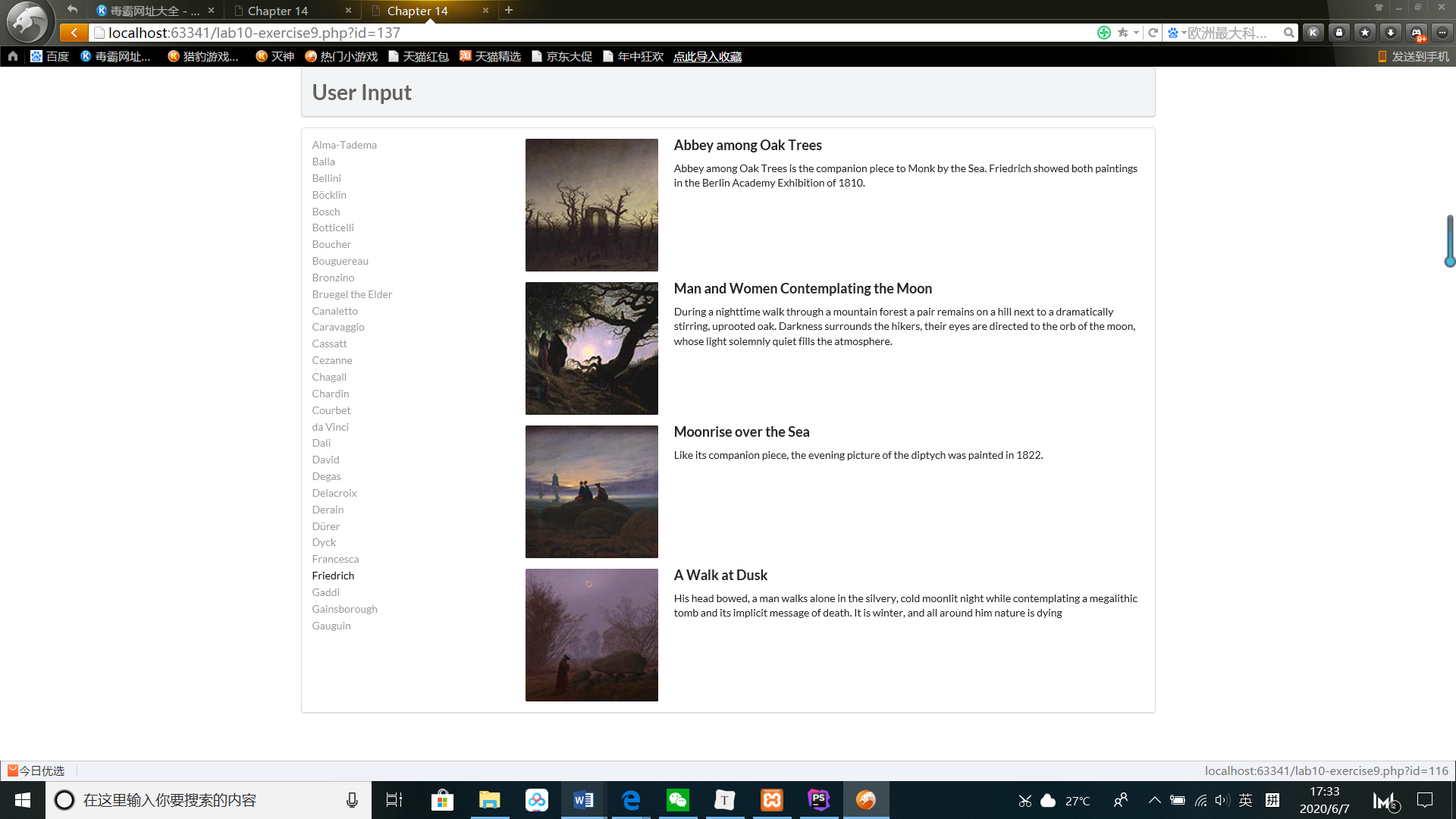
两者区别：前者创建实例操作实例来操作数据库，后者与数据库连接，通过连接操作

**Exercise 8**



用exercise7的第一种方式操作数据库，outputArtists()按last name的顺序筛选出前30个艺术家，用循环的方式将每一个都设置为<a>，并放在“ui link list”下。outputpainting筛选出对应艺术家id索引下的图片信息，达到点击艺术家获得图片信息的结果。输出图片的方法为调用outputSinglePainting($row)函数，通过echo函数在页面上输出一块内容，并通过$row的属性得到位置和名称，包含名字和描述的内容div。这一整块在点击左边链接后显示的图片位于<div class="ui items"></div>中，动态生成。

**Exercise 9**



1. 使用execute方法执行SQL语句：

使用execute()的返回值是boolean,用getResultSet(): 该方法返回Statement执行查询语句所返回的ResultSet对象，getUpdateCount():该方法返回statementDDL,DML语句所影响的记录行数。

1. 使用PreparedStatement执行SQL语句：

可以使用带占位符(?)参数的SQL语句来代替它们，即：

insert into tb\_test values(null,?,?)

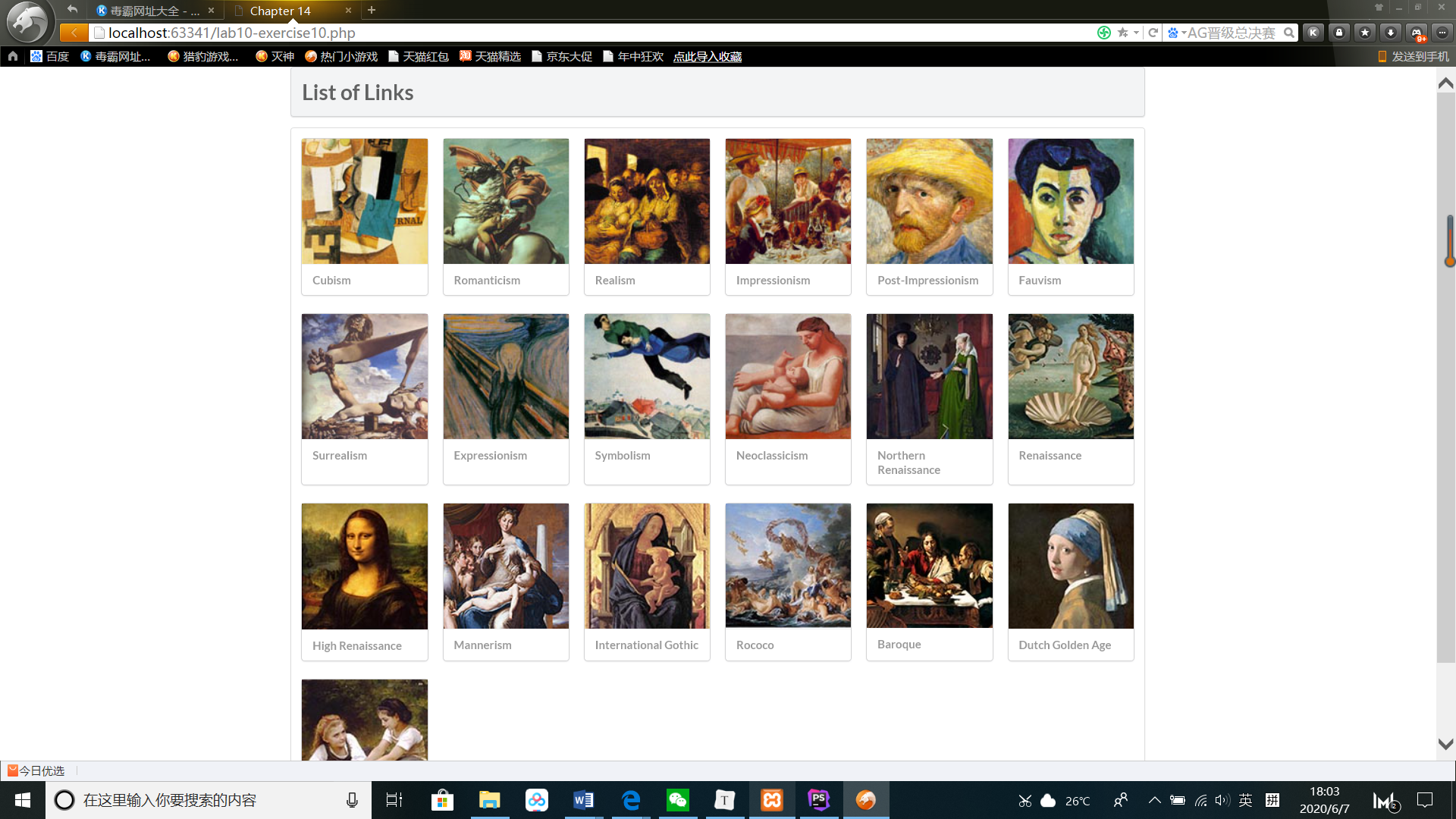
同时提供了一系列的setXxx(int index,Xxx value)方法传入参数值。

相比Statement，该优势在于预编译SQL语句，只解析查询一次，然后可以使用相同或不同的参数执行多次，避免不必要重复优化；无须“拼接”SQL语句；减少SQL注入。

1. 使用CallableStatement调用存储过程：

该对象是由Connection的perpareCall(String sql)方法来创建。可以通过CallableStatement的setXxx()方法为传入参数设置值；所谓传出参数就是Java可以通过该参数获取存储过程里的值，CallableStatement需要调用registerOutParameter(int index,int SQLTypes)方法来注册该参数

**Exercise 10**



outputGenres()函数用上述方法搜索出Genres中的 GenreId, GenreName, Description 并用GenreId进行排序，通过outputSingleGenre输出。