1. 用SQL语句数据库建表（books，mybook, students）

二、编写过滤器Filter：

1.导包：pom.xml

2.编写过滤器（过滤请求乱码）filter(实现Filter接口，重写对应方法即可)

Filter.main.java.com.tian,filter.CharacterEncodingFilter

3.在web.xml中配置过滤器

1. 项目搭建&工具类的实现&依赖导入
2. 编写数据库配置文件：src.main.resources.database.properties

(propoties文件：用于表达配置信息)

# 配置数据库驱动

driver=com.mysql.cj.jdbc.Driver

# 数据库地址

url=jdbc:mysql://localhost:3306/cqutlms?serverTimezone=UTC&useUnicode=true&characterEncoding=utf8&useSSL=true

# 配置数据库用户名

user=root

# 配置密码

password=15220lbs

2.建立操作数据库的公共类（src.main.java.com.tian.utils.BaseDao）

Init方法：初始化连接参数，从配置文件database.properties中获得

Connection方法：加载数据库驱动，连接数据库

执行相关的SQL语句：executeQuery方法：查询操作

executeUpdate方法：增删改操作

closeResource方法：关闭资源

3.导入相关依赖（pom.xml）

1. 建立数据库的实体类（pojo）

src.main.java.com.tian.pojo.book

Student

由于mybooks表和books表的字段基本一致，所以只建立了2个实体了类

1. 实现登录功能
2. 创建登录页面login.jsp(web.login.jsp)
3. 创建登录页面的css样式（login.css）
4. 把登录页面（login.jsp）设置为首页（在web.xml中配置）

三层结构：View层：用于接受用户提交请求的代码在此编写

Service层：系统的业务逻辑主要在这里编写（如StudentService判断学生学号和密码是否匹配）

Dao层：之间操作数据库的代码在这里编写（如StudentDao通过StuId获取学生信息）

1. 建立StudentDao：1.创建StudentDao接口（StudentDao.java）

2.创建StudentDaoImpl（StudentDao接口的实现类）

用来实现“通过StuId获得登录图书管理系统的学生信息”

(通过BaseDao.executeQuery查询数据库实现)

5． 建立StudentService：1.创建StudentService接口（service.student.StudentService）

2.创建StudentServiceImpl（StudentService接口的实现类）

用来实现判断密码是否匹配（和对应学号）

（通过studentDao的方法studentDao.getLoginStudent实现）

.

6.写一个登录请求：LoginServlet：

1.编写LoginServlet（若账号密码匹配则跳转到下一个页面）

2.在web.xml中配置Servlet

1. 查询书籍所有功能：

（一开始页面跳转到/book（管理系统主页）的时候没有值，现在要在跳转过去的时候，就从后台数据库取值）

1. 创建BookDao：1.创建BookDao接口

2.创建BookDao接口的实现类：BookDaoImpl

（作用是查阅所有书籍（通过BaseDao.executeQuery查询数据库实现））

2． 创建BookService：1.创建BookService接口

2.创建BookService接口的实现类：BookServiceImpl

（作用是获得所有书籍信息通过BookDaoImpl的getAllbook方法）

3.创建AllBookServlet：查阅所有书籍的请求，获得值并实现页面跳转

4．配置AllBookServlet：在web.xml配置文件中配置相关内容

5.修改LoginServlet：（使重定向到/books）.

web.xml web项目的配置文件，在web运行时，首先会加载xml文件

Java Servlet 是运行在 Web 服务器或应用服务器上的程序，它是作为来自 Web 浏览器或其他 HTTP 客户端的请求和 HTTP 服务器上的数据库或应用程序之间的中间层。

使用 Servlet，您可以收集来自网页表单的用户输入，呈现来自数据库或者其他源的记录，还可以动态创建网页

database.properties：编写数据库配置文件

driver:一个接口，能从其中获取数据库连接，可以通过Driver类的实现类的对象获取数据库连接

导入相关依赖：不用再一个个new对象了，这样会使new满天飞

pojo：建立数据库的实体类

out文件夹：项目输出文件夹。包含所有已编译的类（存放该项目下所有Module模块编译的结果），当从IDE运行程序时，所有类都将编译在out目录中

target文件夹：存放单个Module模块的编译结果

实现登录功能：

index.jsp：网站的首页（在web.xml里面配置了它是首页）

login.jsp：网站的登录页面

session对象：session 对象（会话）是由服务器自动创建的与用户请求相关的对象。服务器为每个用户都生成一个session对象，用于保存该用户的信息数据，跟踪用户的操作状态。session对象内部使用Map类来保存数据，因此保存数据的格式为 “Key/value”。

EL表达式：主要作用是在Java Web应用程序嵌入到网页（如[JSP](https://zh.wikipedia.org/wiki/JSP)）中，用以访问页面的上下文以及不同作用域中的[对象](https://zh.wikipedia.org/wiki/%E5%AF%B9%E8%B1%A1_(%E8%AE%A1%E7%AE%97%E6%9C%BA%E7%A7%91%E5%AD%A6))，取得对象属性的值，或执行简单的运算或判断操作。

表达式语法：${变量名}

Servlet：一个web程序，主要用于保存传输网页的数据

两个Servlet之间传值：

* 发送方: request.setAttribute(key, value)
* 接收方: request.getAttribute(key)