

---

# 11. 서버 프로그램 구현 수행평가

---

Java2 bookstore

날짜 : 2023/01/10

이름 : 조주영



---

# INDEX

1

프로젝트 생성 및 구성

---

2

화면 구현

---

3

코드 첨부

---

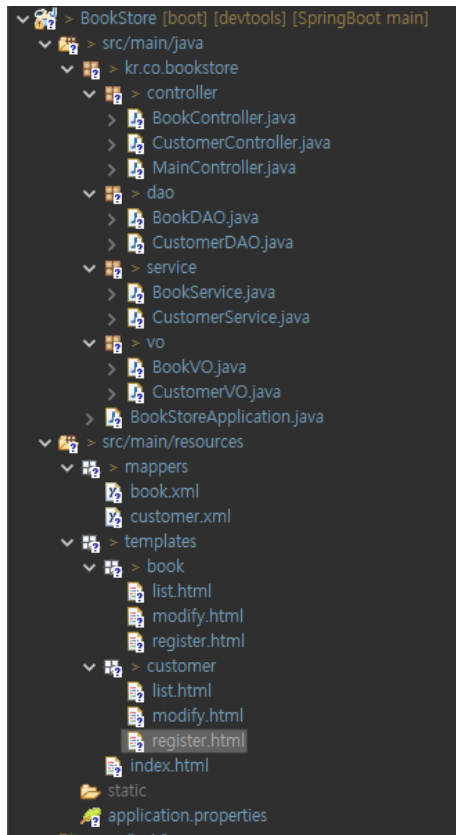
4

실행 (기능 구현)

---

# 1. 프로젝트 생성 및 구성

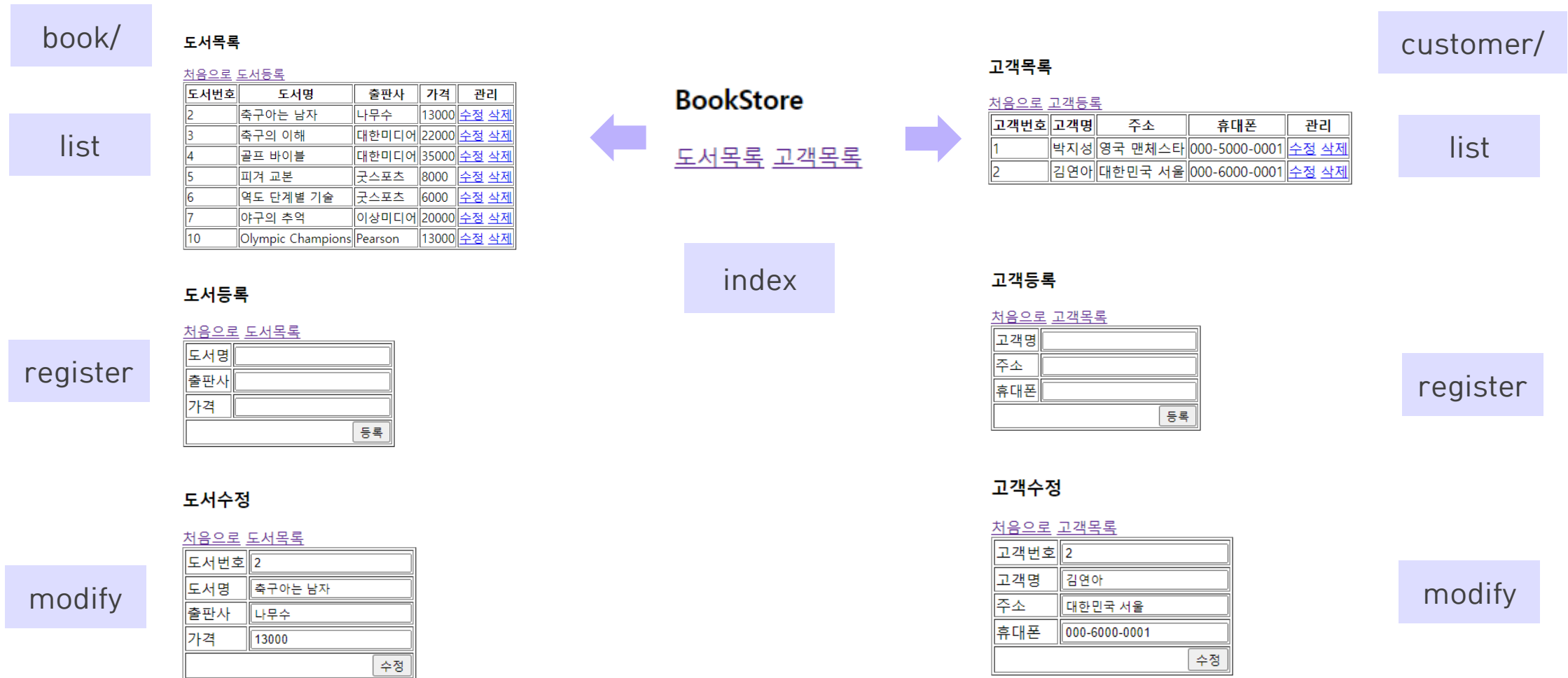
## 프로젝트 구성



## application.properties 설정

```
application.properties x
1 server.servlet.context-path=/BookStore
2 server.port=8080
3 spring.thymeleaf.cache=false
4
5 #Mybatis 설정
6 spring.datasource.url=jdbc:mysql://127.0.0.1:3306/java2_bookstore
7 spring.datasource.username=root
8 spring.datasource.password=1234
9 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
10
11 #MyBatis Mapper 경로설정 -> Ch06Application 클래스 상단에 @MapperScan("kr.co.ch06.persistence") 추가
12 mybatis.mapper-locations=classpath:mappers/**/*.xml
13
```

## 2. 화면 구현



### 3. 코드 첨부 - Controller

#### MainController

```
package kr.co.bookstore.controller;

import org.springframework.stereotype.Controller;

@Controller
public class MainController {

    @GetMapping(value = {"/", "/index"})
    public String index() {
        return "/index";
    }
}
```

#### BookController

```
1 package kr.co.bookstore.controller;
2
3 import java.util.List;
4
14
15 @Controller
16 @RequestMapping("/book")
17 public class BookController {
18
19     @Autowired
20     private BookService service;
21
22     @GetMapping("/list")
23     public String list(Model model) {
24         List<BookVO> books = service.selectBooks();
25         model.addAttribute("books", books);
26         return "/book/list";
27     }
28
29     @GetMapping("/register")
30     public String register() {
31         return "/book/register";
32     }
33
34     @PostMapping("/register")
35     public String register(BookVO vo) {
36         service.insertBook(vo);
37         return "redirect:/book/list";
38     }
39
40     @GetMapping("/modify")
41     public String modify(int bookId, Model model) {
42         BookVO book = service.selectBook(bookId);
43         model.addAttribute("book", book);
44         return "/book/modify";
45     }
46
47     @PostMapping("/modify")
48     public String modify(BookVO vo) {
49         service.updateBook(vo);
50         return "redirect:/book/list";
51     }
52
53     @GetMapping("/delete")
54     public String delete(int bookId) {
55         service.deleteBook(bookId);
56         return "redirect:/book/list";
57     }
58
59 }
60
```

#### CustomerController

```
1 package kr.co.bookstore.controller;
2
3 import java.util.List;
4
14
15 @Controller
16 @RequestMapping("/customer")
17 public class CustomerController {
18
19     @Autowired
20     private CustomerService service;
21
22     @GetMapping("/list")
23     public String list(Model model) {
24         List<CustomerVO> custs = service.selectCustomers();
25         model.addAttribute("custs", custs);
26         return "/customer/list";
27     }
28
29     @GetMapping("/register")
30     public String register() {
31         return "/customer/register";
32     }
33
34     @PostMapping("/register")
35     public String register(CustomerVO vo) {
36         service.insertCustomer(vo);
37         return "redirect:/customer/list";
38     }
39
40     @GetMapping("/modify")
41     public String modify(int custId, Model model) {
42         CustomerVO cust = service.selectCustomer(custId);
43         model.addAttribute("cust", cust);
44         return "/customer/modify";
45     }
46
47     @PostMapping("/modify")
48     public String modify(CustomerVO vo) {
49         service.updateCustomer(vo);
50         return "redirect:/customer/list";
51     }
52
53     @GetMapping("/delete")
54     public String delete(int custId) {
55         service.deleteCustomer(custId);
56         return "redirect:/customer/list";
57     }
58
59 }
60
```

### 3. 코드 첨부 - DAO

#### BookDAO

```
1 package kr.co.bookstore.dao;
2
3 import java.util.List;
4
10
11 @Repository
12 public class BookDAO {
13
14     @Autowired
15     private SqlSessionTemplate mybatis;
16
17     public void insertBook(BookVO vo) {
18         mybatis.insert("store.book.insertBook", vo);
19     }
20
21     public BookVO selectBook(int bookId) {
22         return mybatis.selectOne("store.book.selectBook", bookId);
23     }
24
25     public List<BookVO> selectBooks() {
26         return mybatis.selectList("store.book.selectBooks");
27     }
28
29     public void updateBook(BookVO vo) {
30         mybatis.update("store.book.updateBook", vo);
31     }
32
33     public void deleteBook(int bookId) {
34         mybatis.delete("store.book.deleteBook", bookId);
35     }
36
37 }
38
```

#### CustomerDAO

```
1 package kr.co.bookstore.dao;
2
3 import java.util.List;
4
10
11 @Repository
12 public class CustomerDAO {
13
14     @Autowired
15     private SqlSessionTemplate mybatis;
16
17     public void insertCustomer(CustomerVO vo) {
18         mybatis.insert("customer.insertCustomer", vo);
19     }
20
21     public CustomerVO selectCustomer(int custId) {
22         return mybatis.selectOne("customer.selectCustomer", custId);
23     }
24
25     public List<CustomerVO> selectCustomers() {
26         return mybatis.selectList("customer.selectCustomers");
27     }
28
29     public void updateCustomer(CustomerVO vo) {
30         mybatis.update("customer.updateCustomer", vo);
31     }
32
33     public void deleteCustomer(int custId) {
34         mybatis.delete("customer.deleteCustomer", custId);
35     }
36
37 }
38
```

### 3. 코드 첨부 - Service

#### BookService

```
1 package kr.co.bookstore.service;
2
3 import java.util.List;
10
11 @Service
12 public class BookService {
13
14     @Autowired
15     private BookDAO dao;
16
17     public void insertBook(BookVO vo) {
18         dao.insertBook(vo);
19     }
20
21     public BookVO selectBook(int bookId) {
22         return dao.selectBook(bookId);
23     }
24
25     public List<BookVO> selectBooks() {
26         return dao.selectBooks();
27     }
28
29     public void updateBook(BookVO vo) {
30         dao.updateBook(vo);
31     }
32
33     public void deleteBook(int bookId) {
34         dao.deleteBook(bookId);
35     }
36
37 }
38
```

#### CustomerService

```
1 package kr.co.bookstore.service;
2
3 import java.util.List;
10
11 @Service
12 public class CustomerService {
13
14     @Autowired
15     private CustomerDAO dao;
16
17     public void insertCustomer(CustomerVO vo) {
18         dao.insertCustomer(vo);
19     }
20
21     public CustomerVO selectCustomer(int custId) {
22         return dao.selectCustomer(custId);
23     }
24
25     public List<CustomerVO> selectCustomers() {
26         return dao.selectCustomers();
27     }
28
29     public void updateCustomer(CustomerVO vo) {
30         dao.updateCustomer(vo);
31     }
32
33     public void deleteCustomer(int custId) {
34         dao.deleteCustomer(custId);
35     }
36
37 }
```

### 3. 코드 첨부 - VO, Application, Index

#### BookVO

```
1 package kr.co.bookstore.vo;
2
3 import lombok.AllArgsConstructor;
4
5 @Getter
6 @Setter
7 @AllArgsConstructor
8 @NoArgsConstructor
9 @ToString
10 public class BookVO {
11     private int bookId;
12     private String bookName;
13     private String publisher;
14     private int price;
15 }
16
```

#### CustomerVO

```
1 package kr.co.bookstore.vo;
2
3 import lombok.AllArgsConstructor;
4
5 @Getter
6 @Setter
7 @AllArgsConstructor
8 @NoArgsConstructor
9 @ToString
10 public class CustomerVO {
11     private int custId;
12     private String name;
13     private String address;
14     private String phone;
15 }
16
```

#### Index.html

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3     <head>
4         <meta charset="UTF-8">
5         <title>index</title>
6     </head>
7     <body>
8         <h3>BookStore</h3>
9         <a th:href="@{/book/list}">도서목록</a>
10        <a th:href="@{/customer/list}">고객목록</a>
11    </body>
12 </html>
```

#### BookStoreApplication 위 어노테이션

```
@MapperScan("kr.co.bookstore.dao")
```



### 3. 코드 첨부 - XML

#### mappers/book.xml

```
-//mybatis.org//DTD Mapper 3.0//EN (doctype)
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE mapper
3   PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
4   "https://mybatis.org/dtd/mybatis-3-mapper.dtd">
5 <mapper namespace="store.book">
6   <insert id="insertBook">
7     INSERT INTO `book`
8       (`bookName`, `publisher`, `price`)
9     VALUES ({bookName}, #{publisher}, #{price});
10  </insert>
11 <select id="selectBook" resultType="kr.co.bookstore.vo.BookVO">
12   SELECT * FROM `book` WHERE `bookId`=#{bookId};
13 </select>
14 <select id="selectBooks" resultType="kr.co.bookstore.vo.BookVO">
15   SELECT * FROM `book`;
16 </select>
17 <update id="updateBook">
18   UPDATE `book` SET
19     `bookName` = #{bookName},
20     `publisher` = #{publisher},
21     `price` = #{price}
22   WHERE
23     `bookId` = #{bookId};
24 </update>
25 <delete id="deleteBook">
26   DELETE FROM `book` WHERE `bookId`=#{bookId};
27 </delete>
28 </mapper>
```

#### mappers/customer.xml

```
-//mybatis.org//DTD Mapper 3.0//EN (doctype)
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE mapper
3   PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
4   "https://mybatis.org/dtd/mybatis-3-mapper.dtd">
5 <mapper namespace="customer">
6   <insert id="insertCustomer">
7     INSERT INTO `customer`
8       (`name`, `address`, `phone`)
9     VALUES ({name}, #{address}, #{phone});
10  </insert>
11 <select id="selectCustomer" resultType="kr.co.bookstore.vo.CustomerVO">
12   SELECT * FROM `customer` WHERE `custId`=#{custId};
13 </select>
14 <select id="selectCustomers" resultType="kr.co.bookstore.vo.CustomerVO">
15   SELECT * FROM `customer`;
16 </select>
17 <update id="updateCustomer">
18   UPDATE `customer` SET
19     `name` = #{name},
20     `address` = #{address},
21     `phone` = #{phone}
22   WHERE
23     `custId` = #{custId};
24 </update>
25 <delete id="deleteCustomer">
26   DELETE FROM `customer` WHERE `custId`=#{custId};
27 </delete>
28 </mapper>
```

### 3. 코드 첨부 - list

#### book/list

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <meta charset="UTF-8">
5     <title>book::list</title>
6   </head>
7   <body>
8     <h3>도서목록</h3>
9     <a th:href="@{/index}">처음으로</a>
10    <a th:href="@{/book/register}">도서등록</a>
11    <table border="1">
12      <tr>
13        <th>도서번호</th>
14        <th>도서명</th>
15        <th>출판사</th>
16        <th>가격</th>
17        <th>관리</th>
18      </tr>
19      <tr th:each="book:${books}">
20        <td th:text="${book.bookId}">도서번호</td>
21        <td th:text="${book.bookName}">도서명</td>
22        <td th:text="${book.publisher}">출판사</td>
23        <td th:text="${book.price}">가격</td>
24        <td>
25          <a th:href="@{/book/modify?(bookId=${book.bookId})}">수정</a>
26          <a th:href="@{/book/delete?(bookId=${book.bookId})}">삭제</a>
27        </td>
28      </tr>
29    </table>
30  </body>
31 </html>
```

#### customer/list

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <meta charset="UTF-8">
5     <title>customer::list</title>
6   </head>
7   <body>
8     <h3>고객목록</h3>
9     <a th:href="@{/index}">처음으로</a>
10    <a th:href="@{/customer/register}">고객등록</a>
11    <table border="1">
12      <tr>
13        <th>고객번호</th>
14        <th>고객명</th>
15        <th>주소</th>
16        <th>휴대폰</th>
17        <th>관리</th>
18      </tr>
19      <tr th:each="cust:${custs}">
20        <td th:text="${cust.custId}">고객번호</td>
21        <td th:text="${cust.name}">고객명</td>
22        <td th:text="${cust.address}">주소</td>
23        <td th:text="${cust.phone}">휴대폰</td>
24        <td>
25          <a th:href="@{/customer/modify?(custId=${cust.custId})}">수정</a>
26          <a th:href="@{/customer/delete?(custId=${cust.custId})}">삭제</a>
27        </td>
28      </tr>
29    </table>
30  </body>
31 </html>
```

### 3. 코드 첨부 - modify

book/modify

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <meta charset="UTF-8">
5     <title>book::modify</title>
6   </head>
7   <body>
8     <h3>도서수정</h3>
9     <a th:href="@{/index}">처음으로</a>
10    <a th:href="@{/book/List}">도서목록</a>
11    <form action="/BookStore/book/modify" method="post">
12      <table border="1">
13        <tr>
14          <td>도서번호</td>
15          <td><input type="text" name="bookId" th:value="${book.bookId}" readonly/></td>
16        </tr>
17        <tr>
18          <td>도서명</td>
19          <td><input type="text" name="bookName" th:value="${book.bookName}"/></td>
20        </tr>
21        <tr>
22          <td>출판사</td>
23          <td><input type="text" name="publisher" th:value="${book.publisher}"/></td>
24        </tr>
25        <tr>
26          <td>가격</td>
27          <td><input type="text" name="price" th:value="${book.price}"/></td>
28        </tr>
29        <tr>
30          <td colspan="2" align="right"><input type="submit" value="수정"></td>
31        </tr>
32      </table>
33    </form>
34  </body>
35 </html>
```

customer/modify

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <meta charset="UTF-8">
5     <title>customer::modify</title>
6   </head>
7   <body>
8     <h3>고객수정</h3>
9     <a th:href="@{/index}">처음으로</a>
10    <a th:href="@{/customer/List}">고객목록</a>
11    <form action="/BookStore/customer/modify" method="post">
12      <table border="1">
13        <tr>
14          <td>고객번호</td>
15          <td><input type="text" name="custId" th:value="${cust.custId}" readonly/></td>
16        </tr>
17        <tr>
18          <td>고객명</td>
19          <td><input type="text" name="name" th:value="${cust.name}"/></td>
20        </tr>
21        <tr>
22          <td>주소</td>
23          <td><input type="text" name="address" th:value="${cust.address}"/></td>
24        </tr>
25        <tr>
26          <td>휴대폰</td>
27          <td><input type="text" name="phone" th:value="${cust.phone}"/></td>
28        </tr>
29        <tr>
30          <td colspan="2" align="right"><input type="submit" value="수정"></td>
31        </tr>
32      </table>
33    </form>
34  </body>
35 </html>
```

### 3. 코드 첨부 - register

#### book/register

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <meta charset="UTF-8">
5     <title>book::register</title>
6   </head>
7   <body>
8     <h3>도서등록</h3>
9     <a th:href="@{/index}">처음으로</a>
10    <a th:href="@{/book/list}">도서목록</a>
11    <form action="/BookStore/book/register" method="post">
12      <table border="1">
13        <tr>
14          <td>도서명</td>
15          <td><input type="text" name="bookName"/></td>
16        </tr>
17        <tr>
18          <td>출판사</td>
19          <td><input type="text" name="publisher"/></td>
20        </tr>
21        <tr>
22          <td>가격</td>
23          <td><input type="text" name="price"/></td>
24        </tr>
25        <tr>
26          <td colspan="2" align="right"><input type="submit" value="등록"></td>
27        </tr>
28      </table>
29    </form>
30  </body>
31 </html>
```

#### customer/register

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <meta charset="UTF-8">
5     <title>customer::register</title>
6   </head>
7   <body>
8     <h3>고객등록</h3>
9     <a th:href="@{/index}">처음으로</a>
10    <a th:href="@{/customer/list}">고객목록</a>
11    <form action="/BookStore/customer/register" method="post">
12      <table border="1">
13        <tr>
14          <td>고객명</td>
15          <td><input type="text" name="name"/></td>
16        </tr>
17        <tr>
18          <td>주소</td>
19          <td><input type="text" name="address"/></td>
20        </tr>
21        <tr>
22          <td>휴대폰</td>
23          <td><input type="text" name="phone"/></td>
24        </tr>
25        <tr>
26          <td colspan="2" align="right"><input type="submit" value="등록"></td>
27        </tr>
28      </table>
29    </form>
30  </body>
31 </html>
```

## 4. 실행

### Book 목록

#### 도서목록

[처음으로 도서등록](#)

도서번호	도서명	출판사	가격	관리
2	축구아는 남자	나무수	13000	<a href="#">수정</a> <a href="#">삭제</a>
3	축구의 이해	대한미디어	22000	<a href="#">수정</a> <a href="#">삭제</a>
4	골프 바이블	대한미디어	35000	<a href="#">수정</a> <a href="#">삭제</a>
5	피겨 교본	굿스포츠	8000	<a href="#">수정</a> <a href="#">삭제</a>
6	역도 단계별 기술	굿스포츠	6000	<a href="#">수정</a> <a href="#">삭제</a>
7	야구의 추억	이상미디어	20000	<a href="#">수정</a> <a href="#">삭제</a>
10	Olympic Champions	Pearson	13000	<a href="#">수정</a> <a href="#">삭제</a>

### Book 등록

#### 도서등록

[처음으로 도서목록](#)

도서명	<input type="text" value="문학의 이해"/>
출판사	<input type="text" value="동아대학교"/>
가격	<input type="text" value="11000"/>
<input type="button" value="등록"/>	

#### 도서목록

[처음으로 도서등록](#)

도서번호	도서명	출판사	가격	관리
2	축구아는 남자	나무수	13000	<a href="#">수정</a> <a href="#">삭제</a>
3	축구의 이해	대한미디어	22000	<a href="#">수정</a> <a href="#">삭제</a>
4	골프 바이블	대한미디어	35000	<a href="#">수정</a> <a href="#">삭제</a>
5	피겨 교본	굿스포츠	8000	<a href="#">수정</a> <a href="#">삭제</a>
6	역도 단계별 기술	굿스포츠	6000	<a href="#">수정</a> <a href="#">삭제</a>
7	야구의 추억	이상미디어	20000	<a href="#">수정</a> <a href="#">삭제</a>
10	Olympic Champions	Pearson	13000	<a href="#">수정</a> <a href="#">삭제</a>
17	문학의 이해	동아대학교	11000	<a href="#">수정</a> <a href="#">삭제</a>

### Book 수정

#### 도서수정

[처음으로 도서목록](#)

도서번호	<input type="text" value="17"/>
도서명	<input type="text" value="문학의 이해"/>
출판사	<input type="text" value="동아대학교"/>
가격	<input type="text" value="22000"/>
<input type="button" value="수정"/>	

#### 도서목록

[처음으로 도서등록](#)

도서번호	도서명	출판사	가격	관리
2	축구아는 남자	나무수	13000	<a href="#">수정</a> <a href="#">삭제</a>
3	축구의 이해	대한미디어	22000	<a href="#">수정</a> <a href="#">삭제</a>
4	골프 바이블	대한미디어	35000	<a href="#">수정</a> <a href="#">삭제</a>
5	피겨 교본	굿스포츠	8000	<a href="#">수정</a> <a href="#">삭제</a>
6	역도 단계별 기술	굿스포츠	6000	<a href="#">수정</a> <a href="#">삭제</a>
7	야구의 추억	이상미디어	20000	<a href="#">수정</a> <a href="#">삭제</a>
10	Olympic Champions	Pearson	13000	<a href="#">수정</a> <a href="#">삭제</a>
17	문학의 이해	동아대학교	22000	<a href="#">수정</a> <a href="#">삭제</a>

### Book 삭제

#### 도서목록

[처음으로 도서등록](#)

도서번호	도서명	출판사	가격	관리
2	축구아는 남자	나무수	13000	<a href="#">수정</a> <a href="#">삭제</a>
3	축구의 이해	대한미디어	22000	<a href="#">수정</a> <a href="#">삭제</a>
4	골프 바이블	대한미디어	35000	<a href="#">수정</a> <a href="#">삭제</a>
5	피겨 교본	굿스포츠	8000	<a href="#">수정</a> <a href="#">삭제</a>
6	역도 단계별 기술	굿스포츠	6000	<a href="#">수정</a> <a href="#">삭제</a>
7	야구의 추억	이상미디어	20000	<a href="#">수정</a> <a href="#">삭제</a>
17	문학의 이해	동아대학교	22000	<a href="#">수정</a> <a href="#">삭제</a>

## 4. 실행

### Customer 목록

#### 고객목록

[처음으로 고객등록](#)

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스터	000-5000-0001	<a href="#">수정</a> <a href="#">삭제</a>
2	김연아	대한민국 서울	000-6000-0001	<a href="#">수정</a> <a href="#">삭제</a>

### Customer 등록

#### 고객등록

[처음으로 고객목록](#)

고객명	<input type="text" value="이강인"/>
주소	<input type="text" value="서울 종로구"/>
휴대폰	<input type="text" value="010-1234-1234"/>
<input type="button" value="등록"/>	

#### 고객목록

[처음으로 고객등록](#)

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스터	000-5000-0001	<a href="#">수정</a> <a href="#">삭제</a>
2	김연아	대한민국 서울	000-6000-0001	<a href="#">수정</a> <a href="#">삭제</a>
12	이강인	서울 종로구	010-1234-1234	<a href="#">수정</a> <a href="#">삭제</a>

### Customer 수정

#### 고객수정

[처음으로 고객목록](#)

고객번호	<input type="text" value="12"/>
고객명	<input type="text" value="이강인"/>
주소	<input type="text" value="부산 연제구"/>
휴대폰	<input type="text" value="010-1234-1234"/>
<input type="button" value="수정"/>	

#### 고객목록

[처음으로 고객등록](#)

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스터	000-5000-0001	<a href="#">수정</a> <a href="#">삭제</a>
2	김연아	대한민국 서울	000-6000-0001	<a href="#">수정</a> <a href="#">삭제</a>
12	이강인	부산 연제구	010-1234-1234	<a href="#">수정</a> <a href="#">삭제</a>

### Customer 삭제

#### 고객목록

[처음으로 고객등록](#)

고객번호	고객명	주소	휴대폰	관리
2	김연아	대한민국 서울	000-6000-0001	<a href="#">수정</a> <a href="#">삭제</a>
12	이강인	부산 연제구	010-1234-1234	<a href="#">수정</a> <a href="#">삭제</a>

감사합니다.