2018314737\_김명수 데이터베이스개론 최종보고서

제가 진행하는 프로젝트의 주제는 중고 도서 등록 및 판매 가능 웹페이지를 만드는 것입니다. 이러한 웹 페이지를 만들게 된 계기는 집에 읽지 않는 책이 넘쳐나지만 버리기엔 약간 아까운 책들이 넘쳐나 이를 처리할 방법을 생각해 보다가 중고도서로 판매를 하면 되겠다는 생각을 했습니다. 이 웹페이지를 통해 구매자의 입장에서는 구매하기 어려운 책이나 비싼 책 등을 저렴하게 구매할 수 있습니다. 또한 판매자의 입장에서는 더 이상 읽지 않고, 쓸모가 없는 책을 버리지 않고 돈으로 바꿀 수 있습니다. 세션 유지 시간을 설정하여 일정시간이 지나면 자동 로그아웃 되는 기능이 평소에 궁금했는데, 이번 프로젝트를 통해서 원리를 알게 되었습니다. 제가 만든 웹페이지의 기능은 크게 4가지입니다.

1. 회원 가입, 탈퇴 및 로그인/로그 아웃기능
2. 도서 검색, 추천
3. 도서 구매, 구매 도서 목록 조회
4. 도서 등록, 등록 도서 목록 조회
5. 회원의 포인트제로 Vip고객 관리

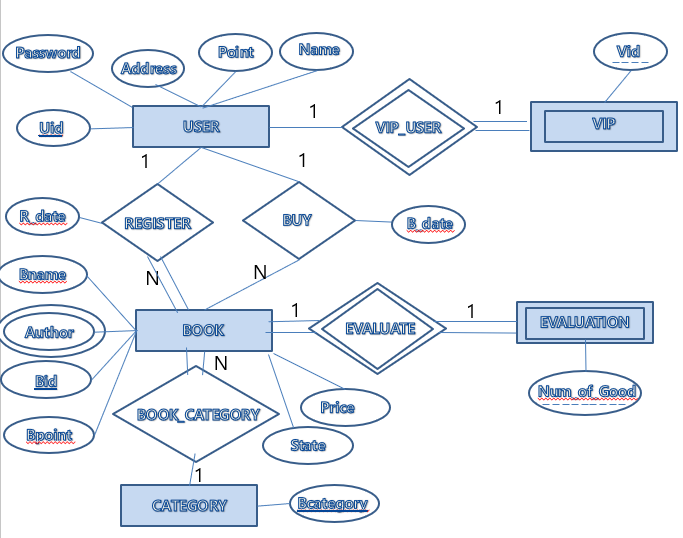
먼저 기본 홈페이지에서 책을 검색하기 위해서는 로그인이 되어야 합니다. ‘로그인’을 클릭하여 로그인과 회원가입을 할 수 있는 페이지로 이동할 수 있습니다. 회원가입시 아이디, 비밀번호, 비밀번호 확인, 주소, 이름 필드가 있습니다. 필드 중 아이디는 중복이 될 수 없습니다. 또한 어느 하나의 필드가 비어서도 안됩니다. 아이디와 비밀번호를 입력하여 로그인 할 수 있습니다. 로그인 시 세션이 생성되고 로그아웃시 세션이 파기됩니다. 또한 10분이 지나면 자동으로 세션이 파기됩니다. ‘회원 탈퇴’를 클릭하여 회원의 정보 및 구매, 등록 도서를 모두 삭제할 수 있습니다.

도서 검색은 검색어를 입력하고 ‘검색’버튼을 클릭하여 수행할 수 있습니다. 도서 제목, 저자, 장르 중 한 글자라도 일치하는 목록을 모두 출력해 줍니다. 검색된 책에는 제목, 저자, 가격, 쌓을 수 있는 포인트, 장르, 추천 수, 구매 가능 여부 필드가 있습니다. 검색한 책에서 ‘추천’ 버튼을 클릭하여 도서를 추천할 수 있습니다.

도서 등록은 로그인 후 ‘도서 등록 하기’를 클릭하여 할 수 있습니다. 책 제목, 저자, 장르, 가격을 입력하여 도서 등록을 할 수 있습니다. 장르는 사전에 데이터 베이스에 있는 장르 중 하나를 선택하도록 합니다. 어느 하나의 필드가 비어서도 안됩니다. 구매는 도서 검색을 완료한 후 ‘도서 구매’ 버튼을 클릭하여 구매할 수 있습니다. 구매한 도서는 ‘도서 구매 목록 조회’를 클릭하여 구매한 날짜를 확인할 수 있습니다. 등록한 도서는 ‘도서 등록 목록 조회’를 클릭하여 등록한 날짜와 도서의 상태(판매 완료/판매 중)을 알 수 있습니다.

도서를 구매 시 일정한 포인트를 적립할 수 있습니다. 포인트가 3000점 이상 쌓이면 Vip 회원이 되어 모든 책을 2000원 할인된 가격에 구매할 수 있습니다.

**ER-diagram**



USER

|  |  |
| --- | --- |
| 사용자의 개인정보(아이디, 비밀번호, 포인트, 이름, 주소)를 나타내는 Regular Entity | |
| Uid | 회원가입시 등록한 사용자의 고유 ID, USER entity의 primary key입니다. 자료형은 문자열-varchar(20)입니다. |
| Password | 회원가입시 등록하 사용자의 비밀번호입니다. 자료형은 문자열-varchar(20)입니다. |
| Address | 회원가입시 등록한 사용자의 주소입니다. 자료형은 문자열-varchar(40)입니다. |
| Point | 회원의 적립 포인트입니다. 자료형은 정수-int(11)입니다. |
| Name | 회원가입시 등록한 사용자의 이름입니다. 자료형은 문자열-varchar(5)입니다. |

**ENTITY**

BOOK

|  |  |
| --- | --- |
| 웹 페이지에 등록된 책의 정보(제목, 저자, 가격, 쌓을 수 있는 포인트, 상태)를 나타내는 Regular Entity입니다. | |
| Bid | 책 등록 시 부여되는 책 고유의 아이디입니다. 책을 등록할 때 마다 1씩 증가되어 모든 책이 다 다른 Bid를 가지고 있습니다. BOOK entity의 primary key입니다. 자료형은 정수-int(11)입니다. |
| Bname | 책 등록 시 등록한 책의 제목을 나타냅니다. 자료형은 문자열-varchar(30)입니다. |
| Author | 책 등록 시 등록한 책의 저자를 나타냅니다. 공저자가 있을 수 있으므로 Multi-valued attribute로 했습니다. 자료형은 문자열-varchar(20)입니다. |
| Price | 책 등록 시 등록한 책의 가격을 나타냅니다. 자료형은 정수 int(11)입니다. |
| Bpoint | 책을 구매하면 쌓을 수 있는 포인트의 양을 나타냅니다. 포인트는 책 가격의 5%입니다. 자료형은 정수-int(11)입니다. |
| State | 책의 상태(구매 가능, 구매 불가)를 나타냅니다. 자료형은 정수-int(11)이며 1은 구매가 가능한 상태, 0은 구매가 불가능한 상태를 나타냅니다. |

VIP

|  |  |
| --- | --- |
| 회원의 포인트가 10000점 이상이면 Vip회원이 됩니다. Vip 회원은 모든 책을 2000원 할인된 가격에 구매할 수 있습니다. USER가 정의되어야 VIP도 정의되므로 weak entity입니다. | |
| Vid | Vip회원의 id이며 회원이 VIP에 등록될 시USER의 id와 동일한 값을 가집니다. VIP entity의 partial key입니다. |

EVALUATION

|  |  |
| --- | --- |
| 각 책의 평가(추천 수)를 나타냅니다. BOOK이 정의되어야 EVALUATION도 정의되므로 weak entity입니다. | |
| Num\_of\_Good | 사용자들이 해당 책의 ‘추천’버튼을 누른 횟수를 나타냅니다. 자료형은 정수-int(11)입니다. EVALUATION entity의 partial key입니다. |

CATEGORY

|  |  |
| --- | --- |
| 책 등록 시 선택할 장르의 목록을 나타내는 Regular entity입니다. | |
| Bcategory | 미리 데이터베이스에 저장되어 있는 장르의 목록입니다. 책 등록 시 이 목록 중 한 장르를 선택하게 됩니다. 자료형은 문자열-varchar(20)입니다. |

**RELATION**

BUY

|  |  |
| --- | --- |
| 사용자가 책을 구매하는 행위를 나타내는 Relation입니다. BUY관계에서 USER와 BOOK모두 partial participate입니다. USER와 BOOK은 BUY 관계에서 1:N의 관계를 가집니다. | |
| B\_date | 사용자가 책을 구매한 시점의 날짜(“yyyy-mm-dd”)를 나타냅니다. 자료형은 날짜-date입니다. |

REGISTER

|  |  |
| --- | --- |
| 사용자가 책을 등록하는 행위를 나타내는 Relation입니다. REGISTER관계에서 USER는 partial participate이고 BOOK은 total participate입니다. USER와 BOOK은 REGISTER관계에서 1:N관계를 가집니다. | |
| R\_date | 사용자가 책을 등록한 시점의 날짜(“yyyy-mm-dd”)를 나타냅니다. 자료형은 날짜-date입니다. |

VIP\_USER

|  |
| --- |
| 회원이 VIP회원에 등록되는 것을 나타내는 Relation입니다. VIP\_USER관계에서 USER는 partial participate이고 VIP는 total participate입니다. USER와 VIP는 1:1 관계입니다. |

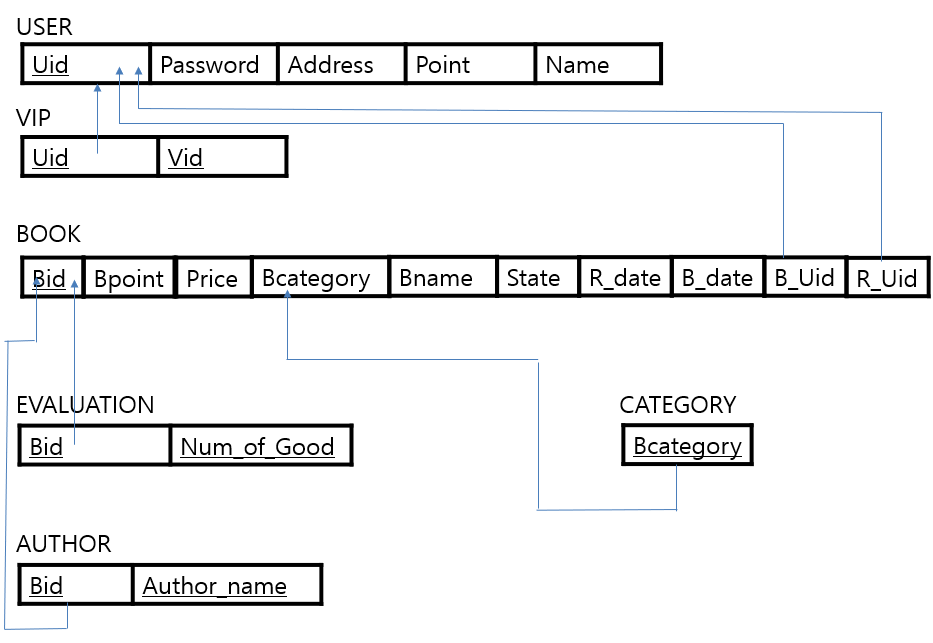
EVALUATE

|  |
| --- |
| 각각의 책이 평가되는 것을 나타내는 Relation입니다. EVALUATE에서 BOOK과 EVALUATION은 total participate입니다. BOOK과 EVALUATION는 1:1관계입니다. |

BOOK\_CATEGORY

|  |
| --- |
| 책이 어떤 장르에 속하는지를 나타내는 Relation입니다. BOOK\_CATEGORY에서 BOOK은 total participate이고 CATEGORY는 partial participate입니다. BOOK과 CATEGORY는 N:1관계입니다. |

**Relational Model**



TABLE

USER

|  |  |
| --- | --- |
| 회원들의 기본정보가 저장되는 table입니다. | |
| Uid | 회원가입시 등록한 사용자의 고유 ID, USER table의 primary key입니다. 자료형은 문자열-varchar(20)입니다. |
| Password | 회원가입시 등록하 사용자의 비밀번호입니다. 자료형은 문자열-varchar(20)입니다. NULL값이 될 수 없습니다. |
| Address | 회원가입시 등록한 사용자의 주소입니다. 자료형은 문자열-varchar(40)입니다. NULL값이 될 수 없습니다. |
| Point | 회원의 적립 포인트입니다. 자료형은 정수-int(11)입니다. 초기 기본값은 0 입니다. |
| Name | 회원가입시 등록한 사용자의 이름입니다. 자료형은 문자열-varchar(5)입니다. NULL값이 될 수 없습니다. |

VIP

|  |  |
| --- | --- |
| 회원들 중 Vip 회원의 id가 저장되는 table입니다. (Uid,Vid)의 조합이 VIP table의 primary key가 됩니다. | |
| Uid | VIP 가 Weak entity이므로 USER의 primary key 인 Uid를 참조하는 foreign key를 갖습니다. |
| Vid | Vip회원의 id이며 회원이 VIP에 등록될 시USER의 id와 동일한 값을 가집니다. |

CATEGORY

|  |  |
| --- | --- |
| 책 장르의 목록을 저장하고 있는 table입니다. | |
| Bcategory | CATEGORY table의 primary key입니다. |

BOOK

|  |  |
| --- | --- |
| 웹 페이지에 등록된 책의 정보가 저장되는 table입니다. | |
| Bid | 책의 고유한 id 번호입니다. 정수이며 BOOK table에 데이터가 추가될 때 마다 1씩 증가합니다. BOOK table의 primary key 입니다. |
| Bpoint | 해당 책을 구매했을 때 얻을 수 있는 포인트의 양을 나타냅니다. |
| Price | 해당 책의 가격을 나타냅니다. |
| Bcategory | 해당 책의 장르를 나타냅니다. BOOK table과 CATEGORY table이 N:1관계이기 때문에 CATEGORY의 primary key인 Bcategory를 BOOK의 column으로 추가하고,이를 CATEGORY의 Bcategory를 참조하는 foreign key로 설정했습니다. |
| Bname | 해당 책의 제목을 나타냅니다. |
| State | 해당 책의 상태(구매가능/구매불가)를 나타냅니다. |
| R\_date | 회원이 책을 등록한 시점의 날짜를 나타냅니다. BOOK table과 USER table이 N:1관계이기 때문에 만들어진 column입니다. REGISTER의 attribute인 R\_date를 BOOK의 column에 추가했습니다. |
| B\_date | 회원이 책을 구매한 시점의 날짜를 나타냅니다. BOOK table과 USER table이 N:1관계이기 때문에 만들어진 column입니다. BUY의 attribute인 B\_date를 BOOK의 column에 추가했습니다. |
| R\_Uid | 해당 책을 등록한 회원의 id를 나타냅니다. BOOK table과 USER table이 N:1관계이기 때문에 만들어진 column입니다. USER의 primary key인 Uid를 BOOK의 column으로 추가했습니다. 그리고 이를 USER의 Uid를 참조하는 foreign key로 설정했습니다. 책을 구매한 회원의 아이디와 구분하기 위해 이름을 R\_Uid로 설정했습니다. |
| B\_Uid | 해당 책을 구매한 회원의 id를 나타냅니다. BOOK table과 USER table이 N:1관계이기 때문에 만들어진 column입니다. USER의 primary key인 Uid를 BOOK의 column으로 추가했습니다. 그리고 이를 USER의 Uid를 참조하는 foreign key로 설정했습니다. 책을 등록한 회원의 아이디와 구분하기 위해 이름을 B\_Uid로 설정했습니다. |

EVALUATION

|  |  |
| --- | --- |
| 책의 평가(추천 수)가 저장되는 table입니다. (Bid, Num\_of\_Good)의 조합이 primary key입니다. | |
| Bid | EVALUATION 이 Weak entity이므로 BOOK의 primary key 인 Bid를 참조하는 foreign key를 갖습니다. 자료형은 정수-int(11)입니다 |
| Num\_of\_Good | 해당 책의 평가(추천 수)를 나타냅니다. |

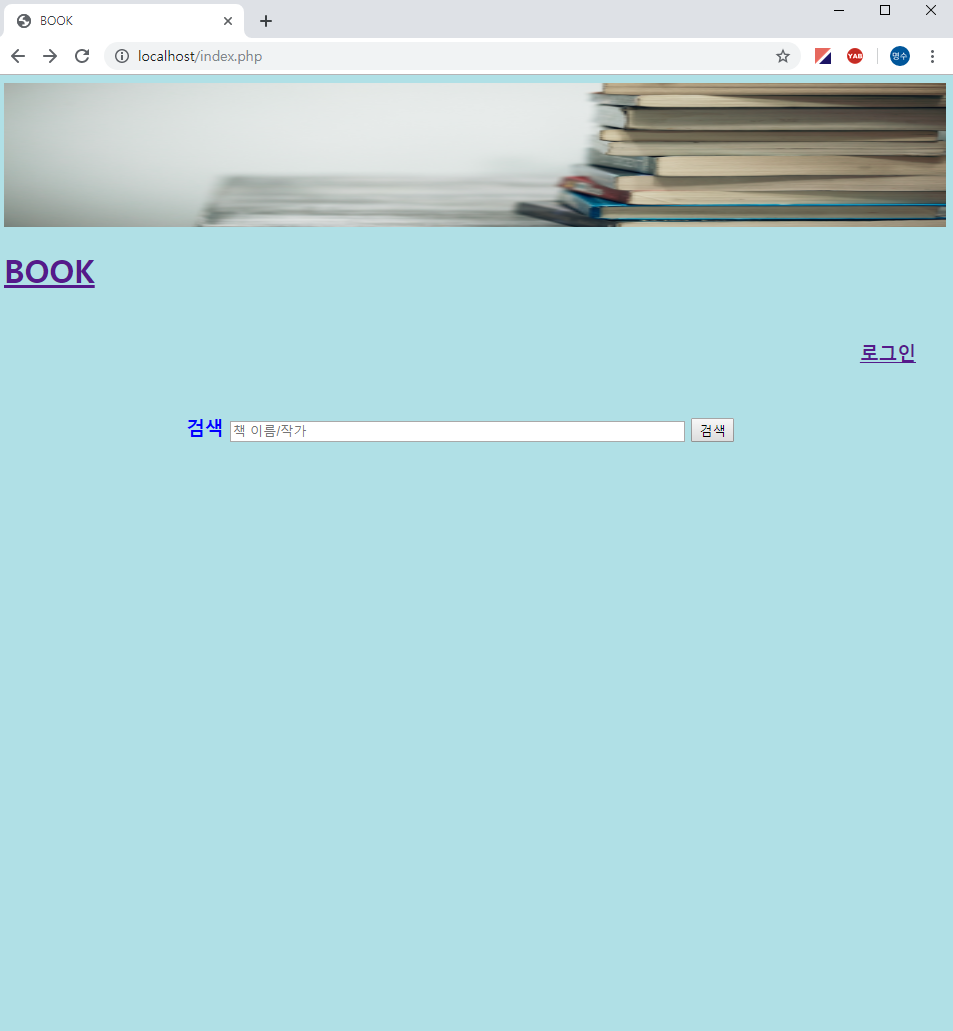
AUTHOR

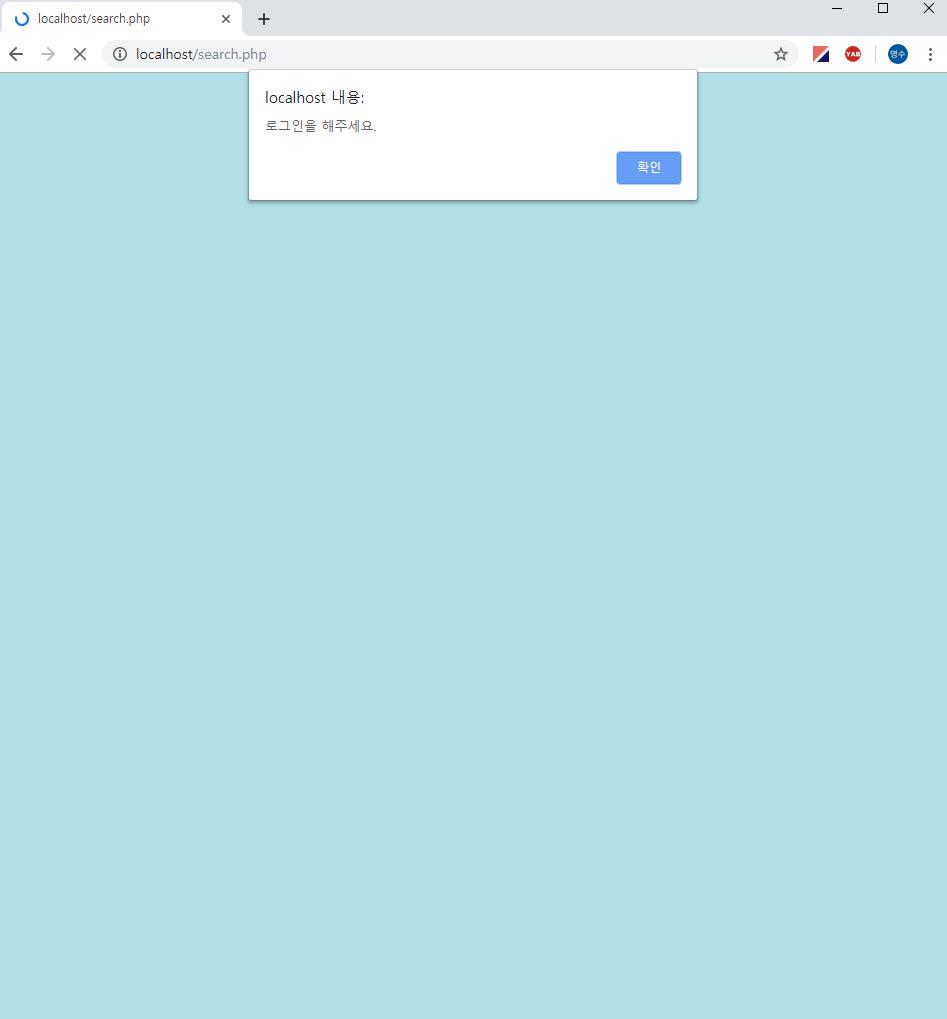
|  |  |
| --- | --- |
| 책의 저자가 저장되는 table입니다 BOOK entity의 attribute인 Author가 multi-valued attribute이기 때문에 생성된 table입니다. (Bid, Author\_name)의 조합이 primary key입니다. | |
| Bid | BOOK의 primary key인 Bid를 참조하는 foreign key입니다. 자료형은 정수(11)입니다 |
| Author\_name | 해당 책의 저자를 나타냅니다. 자료형은 문자열-varchar(20)입니다. |

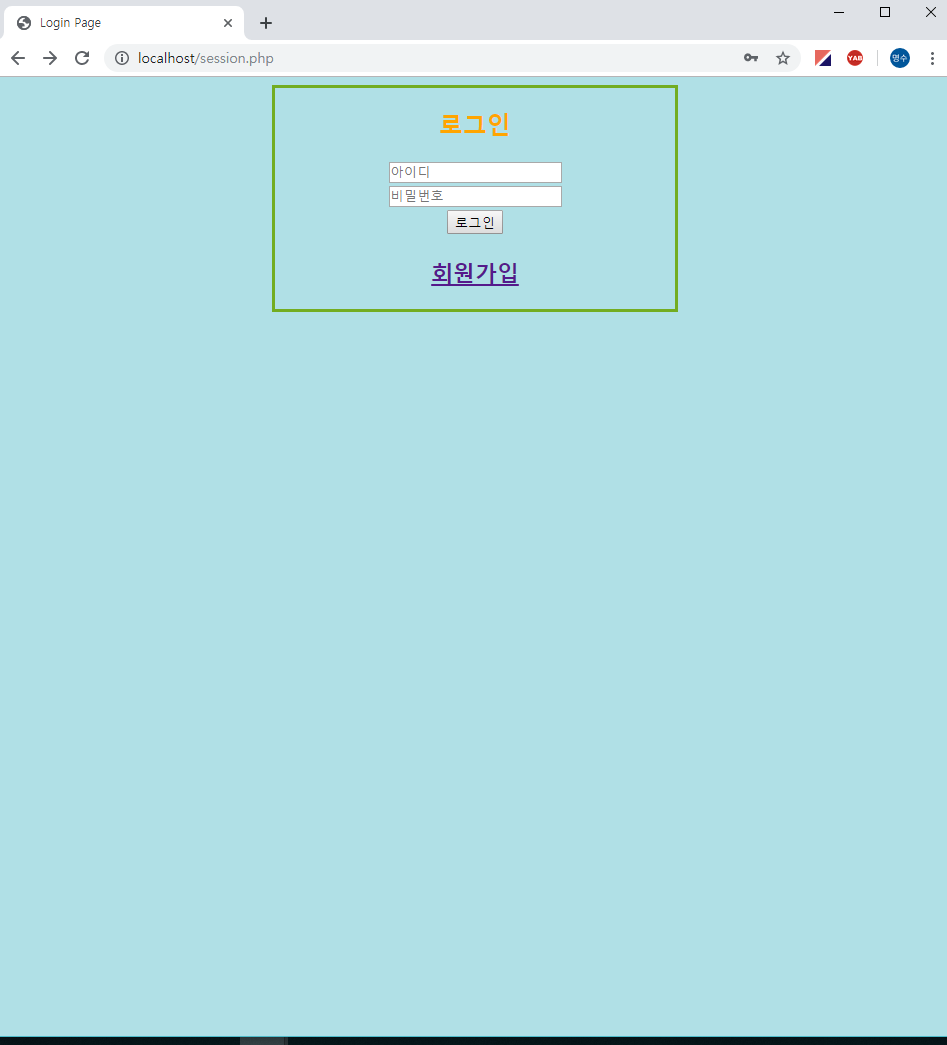
**Implementation and Results**

1. 회원 가입, 탈퇴 및 로그인/로그 아웃기능
2. 도서 검색, 추천
3. 도서 구매, 구매 도서 목록 조회
4. 도서 등록, 등록 도서 목록 조회
5. 회원의 포인트제로 Vip고객 관리

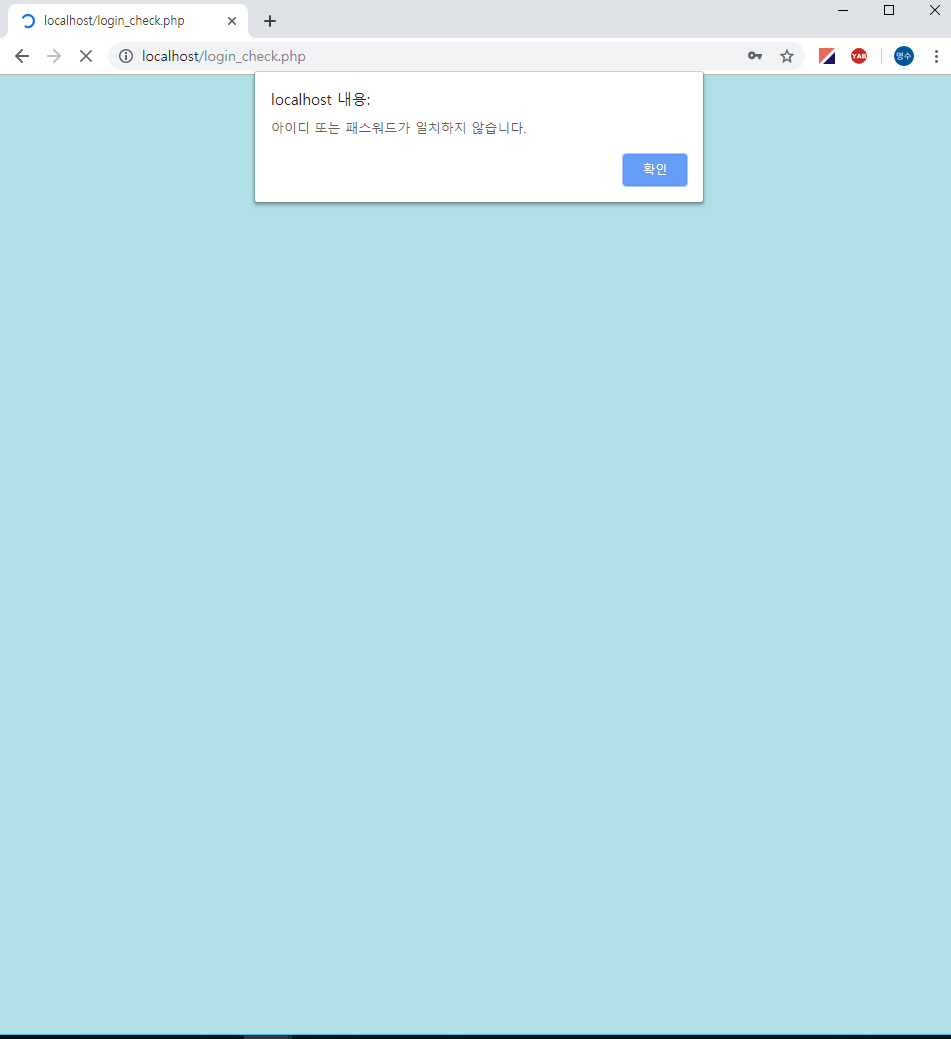
먼저 회원 가입, 탈퇴 및 로그인, 로그아웃 기능을 구현했습니다. 로그인을 하지 않은 상태의 메인 페이지는 index.php로 다음과 같습니다.

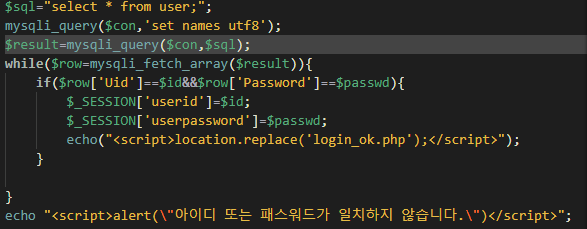


여기서 ‘검색’ 버튼을 누르게 되면 다음과 같이 로그인을 요구하는 알람과 함께 로그인을 할 수 있는 session.php로 이동합니다. Index.php의 ‘로그인’을 클릭해도 session.php로 이동하게 됩니다.



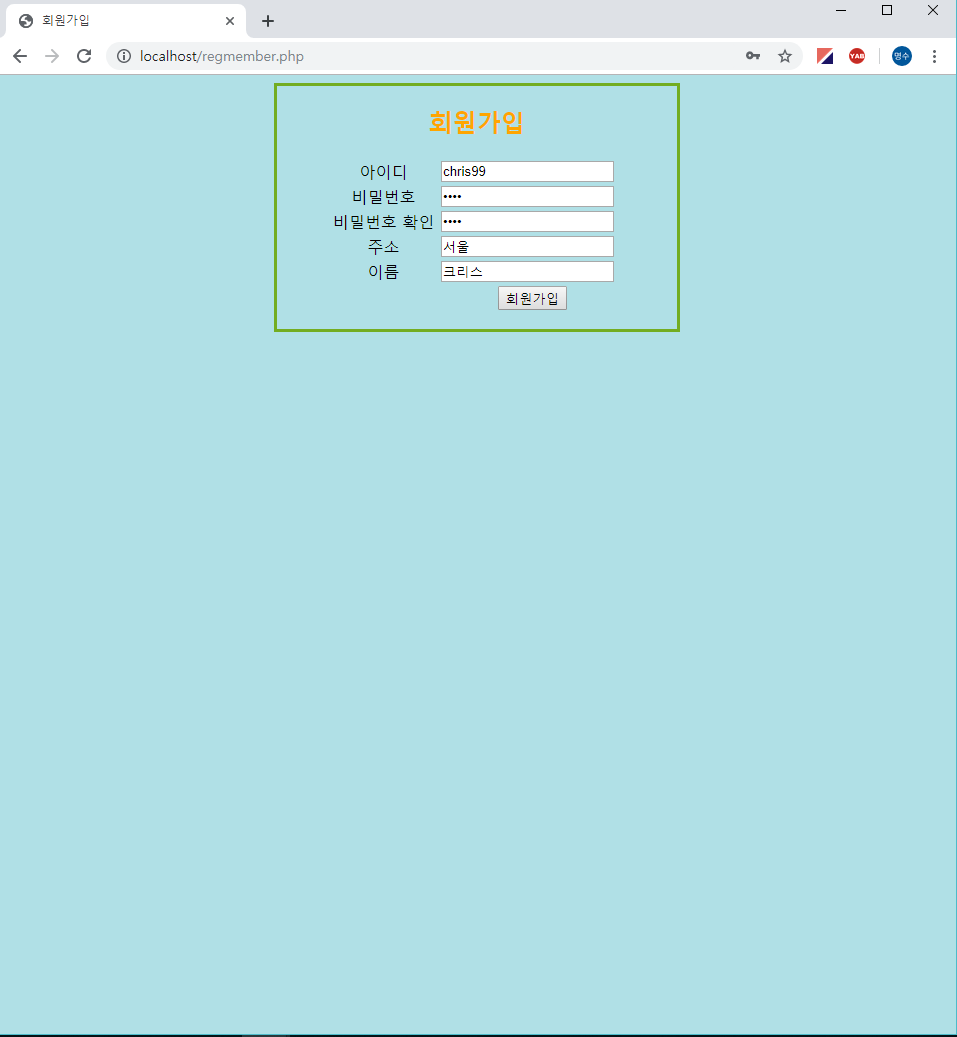
‘아이디’ 필드와 ‘비밀번호’ 필드에 정보를 입력한 뒤 ‘로그인’ 버튼을 누르면 post 형식으로 아이디와 비밀번호가login\_check.php로 전달이 됩니다. login\_check.php에서 전달받은 아이디와 비밀번호가 데이터베이스에 있는 지 확인을 한 후에 올바른 정보면 로그인에 성공하여 login\_ok.php로 이동합니다. 확인방법은 post로 받아온 아이디와 비밀번호를 USER테이블에서 일일이 대조하는 방식입니다. 만약 아이디가 등록되지 않았거나, 정보가 올바르지 않다면 다음과 같이 “아이디 또는 패스워드가 일치하지 않습니다”라는 알림이 생성되고 다시 session.php로 돌아갑니다.



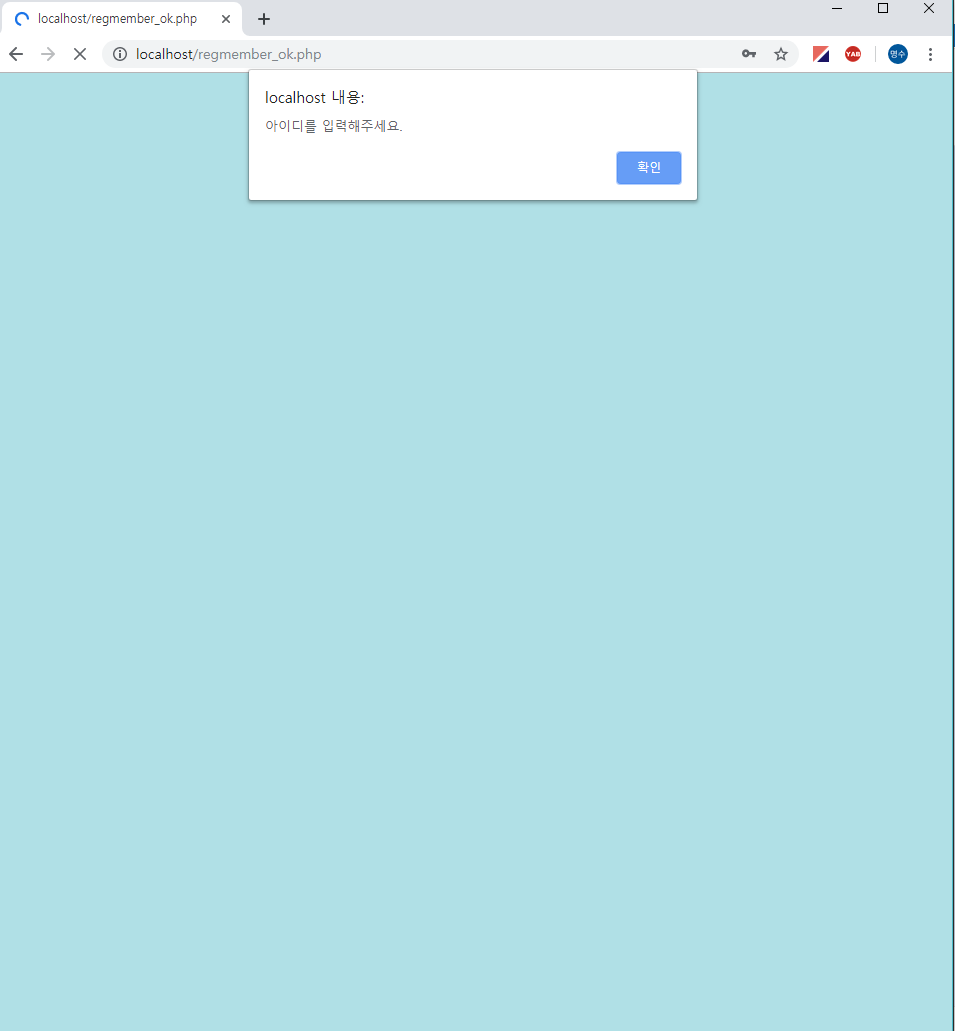


login\_check.php에서

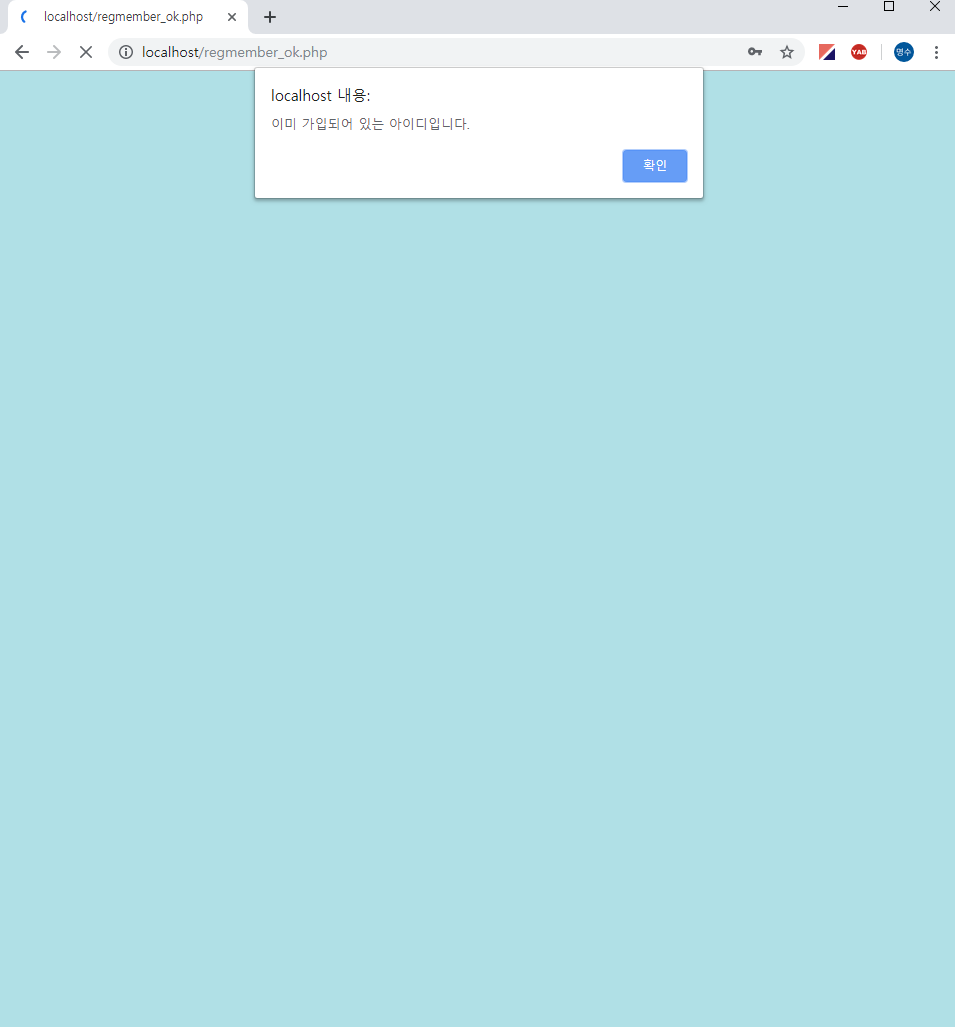
아이디, 패스워드 검사

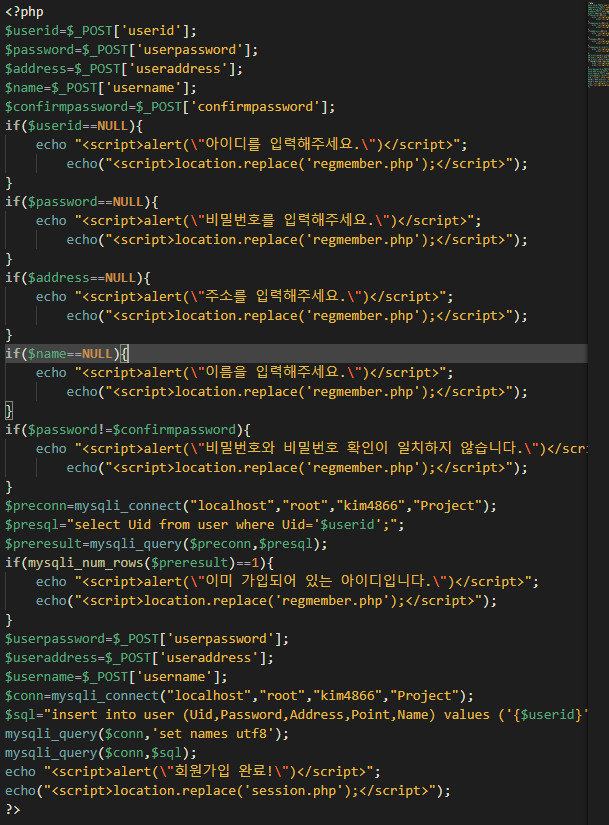
session.php에서 ‘회원가입’을 클릭하면 회원가입을 할 수 있는 regmember.php로 이동합니다.

각각의 필드를 모두 채우고, 아이디가 이미 데이터베이스에 존재하는 것이 아니라면 데이터가 regmember\_ok.php로 넘어가 회원가입이 완료되었다는 알림과 함께 회원가입이 완료됩니다. 이후 다시 session.php로 돌아옵니다. 하나의 필드라도 비어 있다면 그 필드를 채워 달라는 알림이 뜨면서 regmember.php로 돌아옵니다.



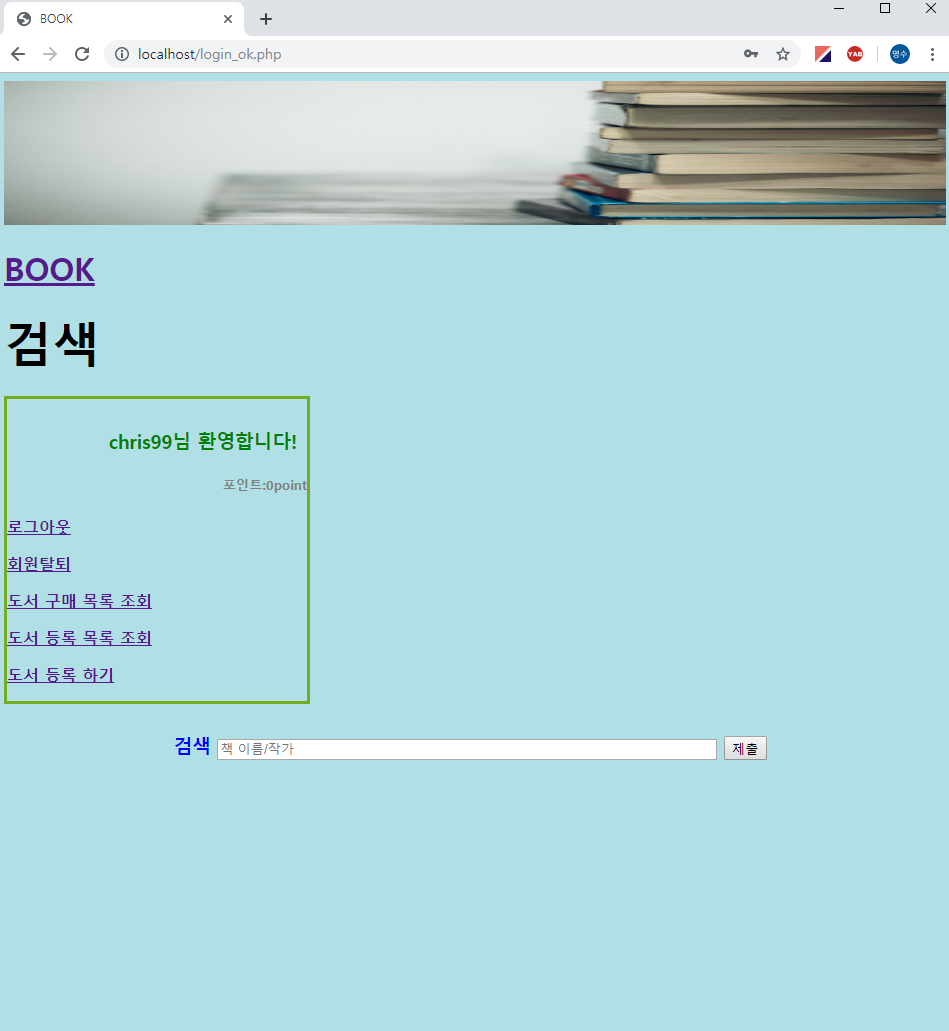
이미 존재하는 아이디를 입력한 경우 “이미 가입되어 있는 아이디입니다” 라는 알림과 함께 regmember.php로 돌아옵니다. 이미 존재하는 아이디인지 확인하는 방법으로 입력한 아이디를 USER테이블에서 where 문을 이용하여 검색했을 때 row가 1개라면 이미 가입된 아이디라는 점을 이용했습니다.





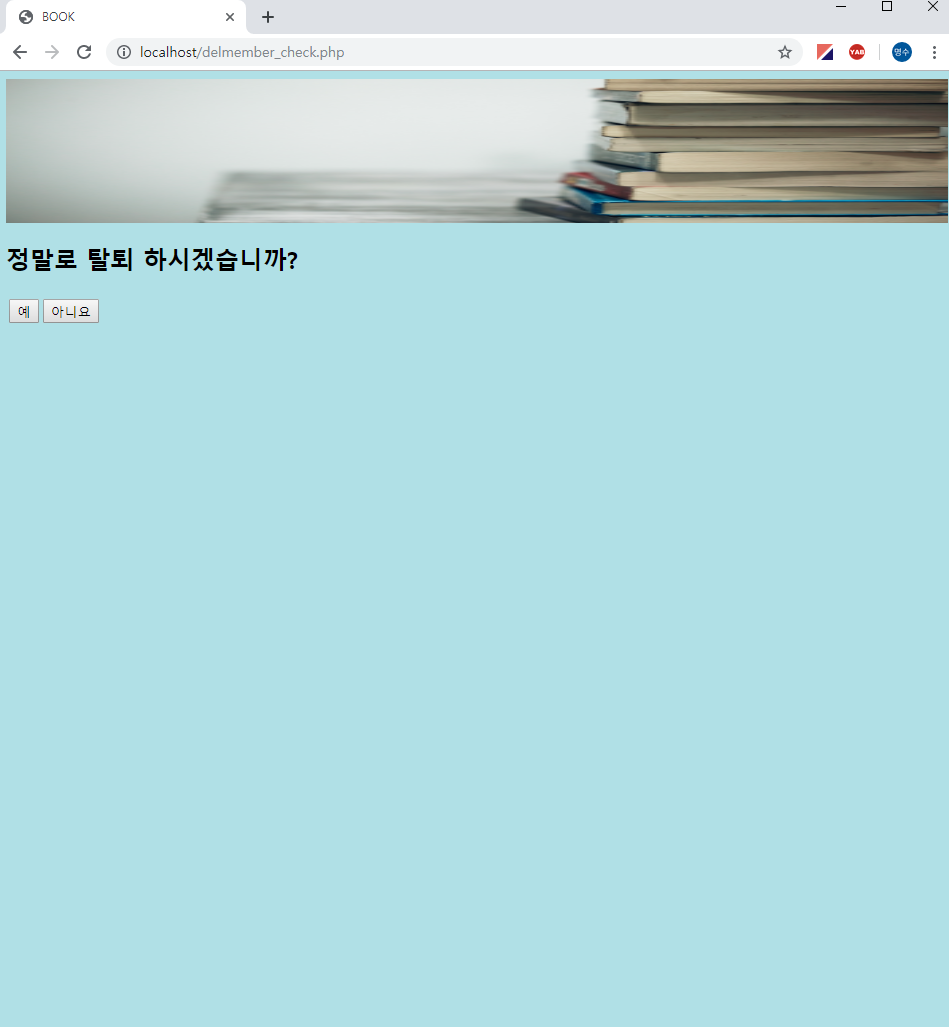
regmember\_ok에서 아이디 중복체크 및 값이 비어 있는 필드 체크

로그인을 성공했을 때의 기본페이지는 login\_ok.php입니다. 로그인 성공시 login\_check.php에서 세션 값을 생성합니다.

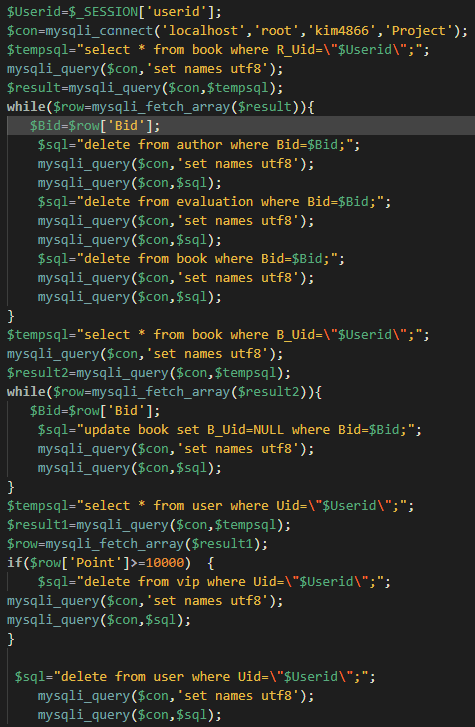
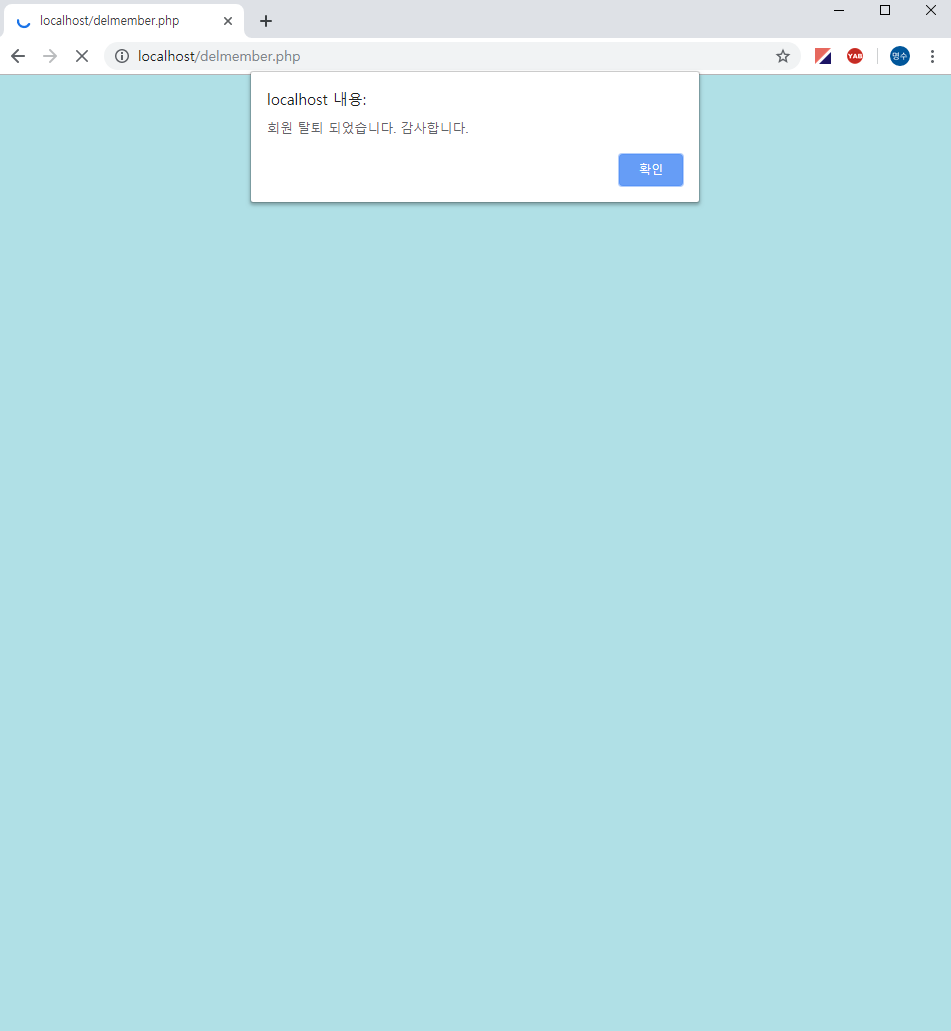


좌측에 ‘BOOK’은 로그인이 되었다면 login\_ok.php로 아니라면 index.php로 이동합니다.

초록색 박스안에는 회원의 현재 적립한 point를 출력해 줍니다. ‘로그아웃’을 클릭하면 logout.php로 이동하여 세션을 파기하고 index.php로 넘어갑니다. ‘화원탈퇴’를 클릭하면 회원탈퇴를 재차 물어보는 delmember\_check.php로 이동합니다.

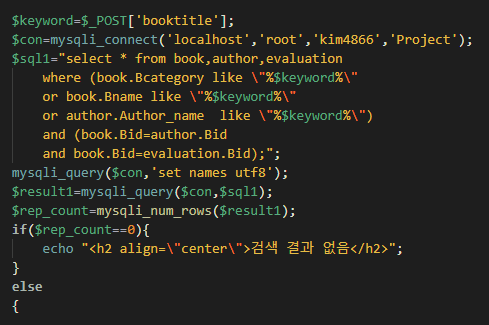
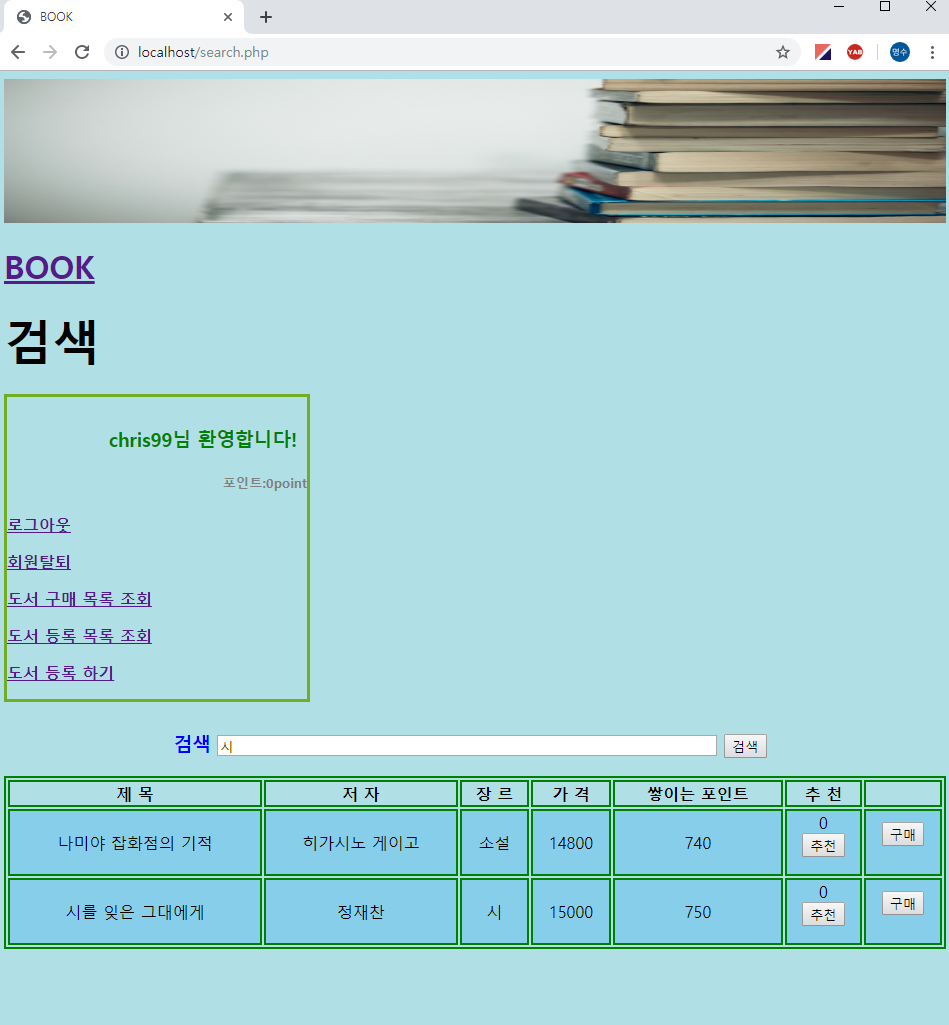


‘예’ 버튼을 누르면 로그인 되어있는 회원의 정보(세션 값)이 post 형식으로 delmember.php에 넘어갑니다. delmember.php에서 회원관련 정보를 삭제한 후 회원탈퇴가 완료되었다는 알림과 함께 회원탈퇴가 완료됩니다. 삭제되는 회원 관련 정보는 회원이 등록한 책, 회원이 등록한 책의 저자, 회원이 등록한 책의 평가(추천 수), 회원의 개인 정보 가 있습니다. Foreign key로 인해 삭제의 순서를 AUTHOR 테이블->EVALUATION 테이블->BOOK 테이블->USER테이블 순으로 삭제가 진행됩니다. 먼저 BOOK 테이블에서 R\_Uid와 로그인한 회원의 Uid가 같은 행의 Bid값을 추출합니다. 그리고 추출한 Bid들을 이용하여 AUTHOR, EVALUATION, BOOK 테이블애서 삭제를 진행합니다. 만약 회원이 책을 구매한 이력이 있을 시 BOOK테이블의 B\_Uid가 NULL로 바뀌게 됩니다.



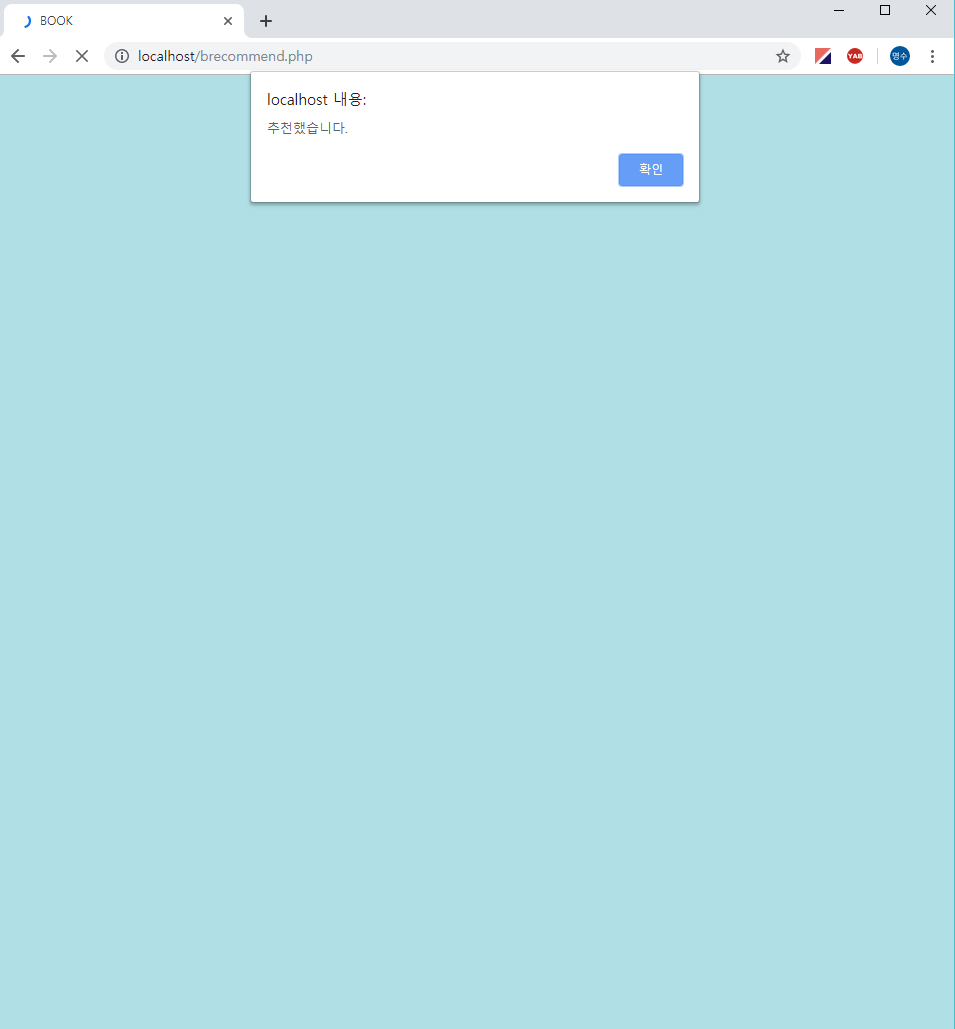
회원 정보 삭제

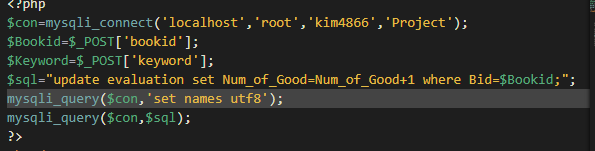
다음으로 도서 검색 및 추천 기능을 구현했습니다. 검색은 로그인이 된 상태에서 실행 할 수 있습니다. 검색 창에 값을 입력하면 post 형식으로 search.php에 값이 전달됩니다. search.php에서는 post로 받아온 글자 BOOK 테이블의 Bname, Bcategory 그리고 AUTHOR 테이블의 Author\_name 중에 한 글자라도 일치하는 책을 모두 출력해서 보여줍니다. 출력되는 정보로는 제목, 저자, 장르, 가격, 쌓이는 포인트, 추천, 구매가능 여부 가 있습니다. 출력을 위해 BOOK 테이블과 AUTHOR 테이블, EVALUATION 테이블을 JOIN 했습니다.



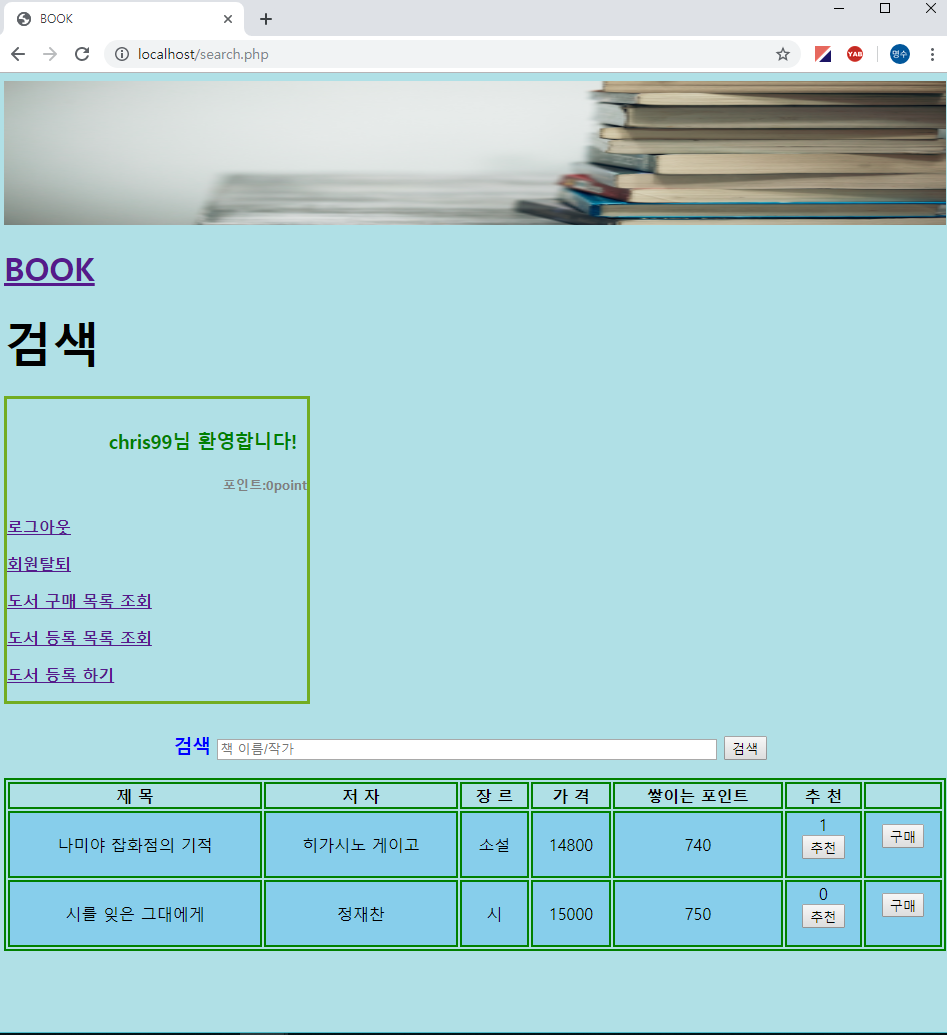
‘like’를 이용하여 키워드 포함하는 데이터를 추출

검색 후 책 목록에서 ‘추천’ 버튼을 누르면 해당 책을 추천할 수 있습니다. 추천한 후에는 해당책의 추천 수가 1증가하고 “추천했습니다”라는 알림과 함께 search.php로 돌아갑니다. ‘추천’ 버튼을 누르면 해당 책의 Bid가 brecommend.php로 전달이 되어 update 문을 통해 Num\_of\_Good의 수가 1증가하게 됩니다.

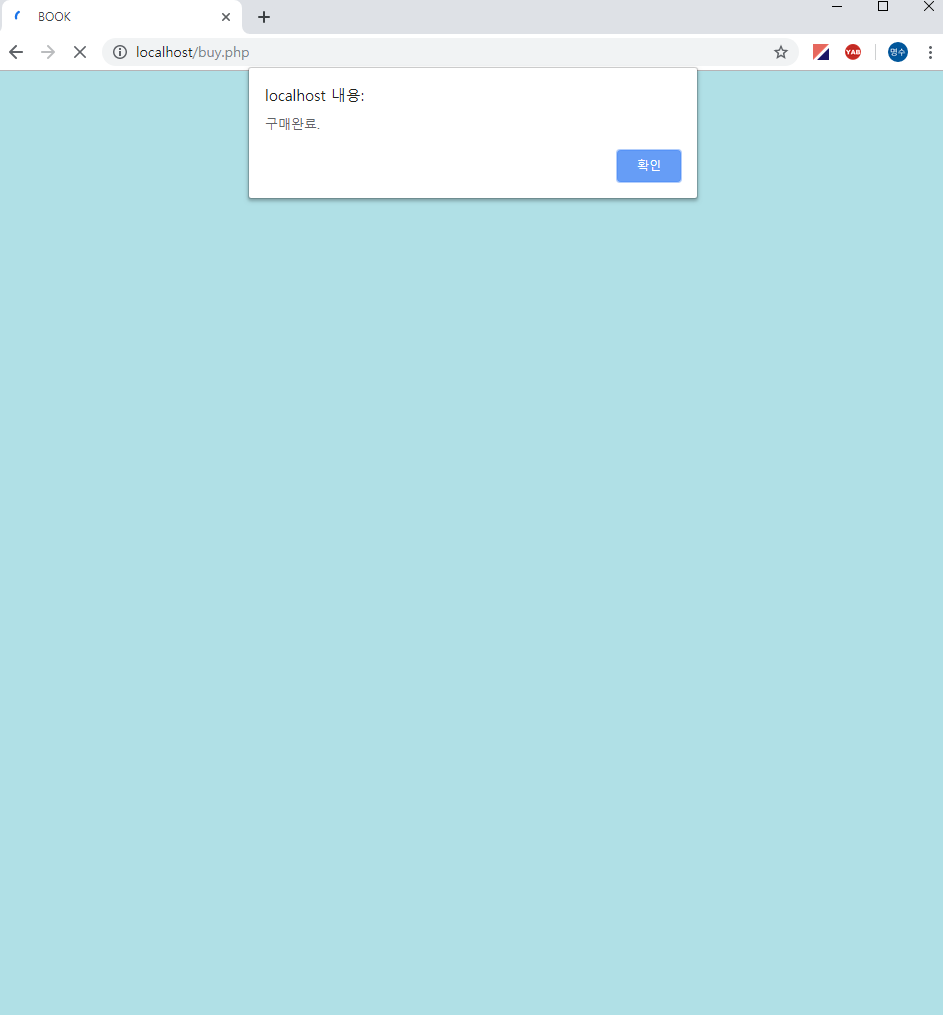
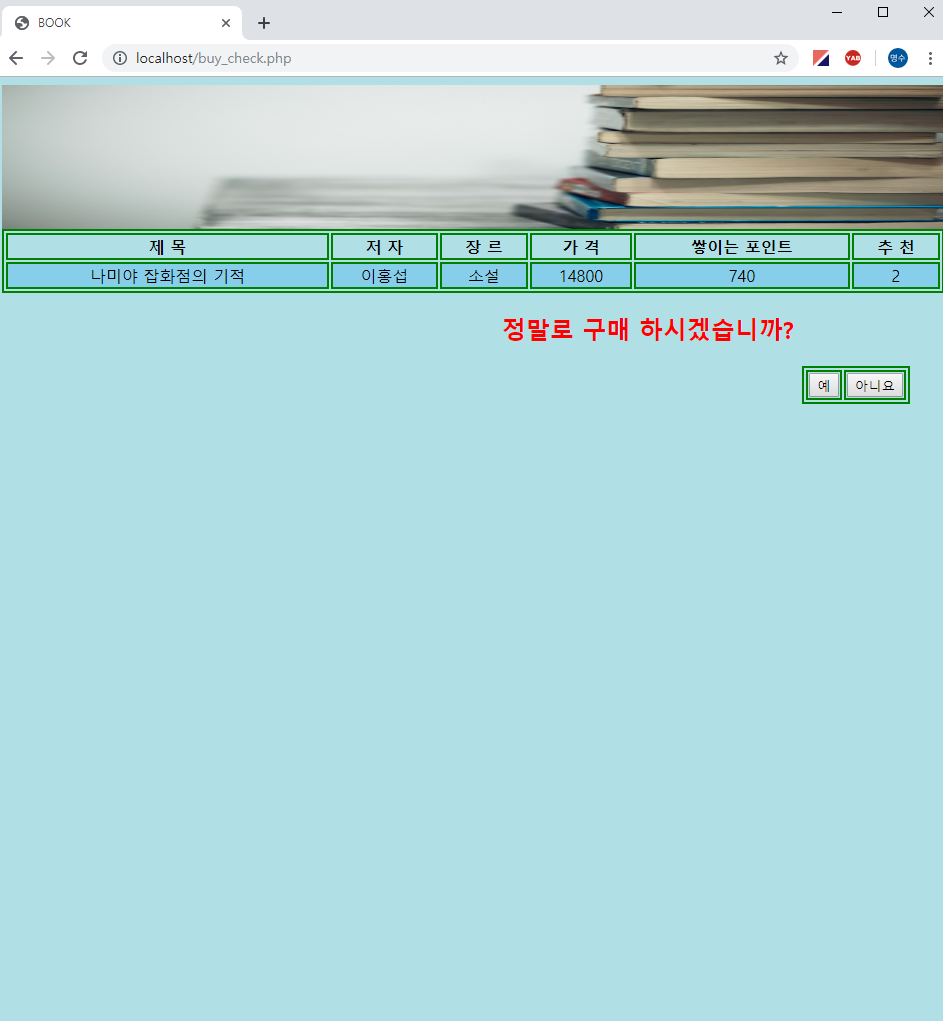


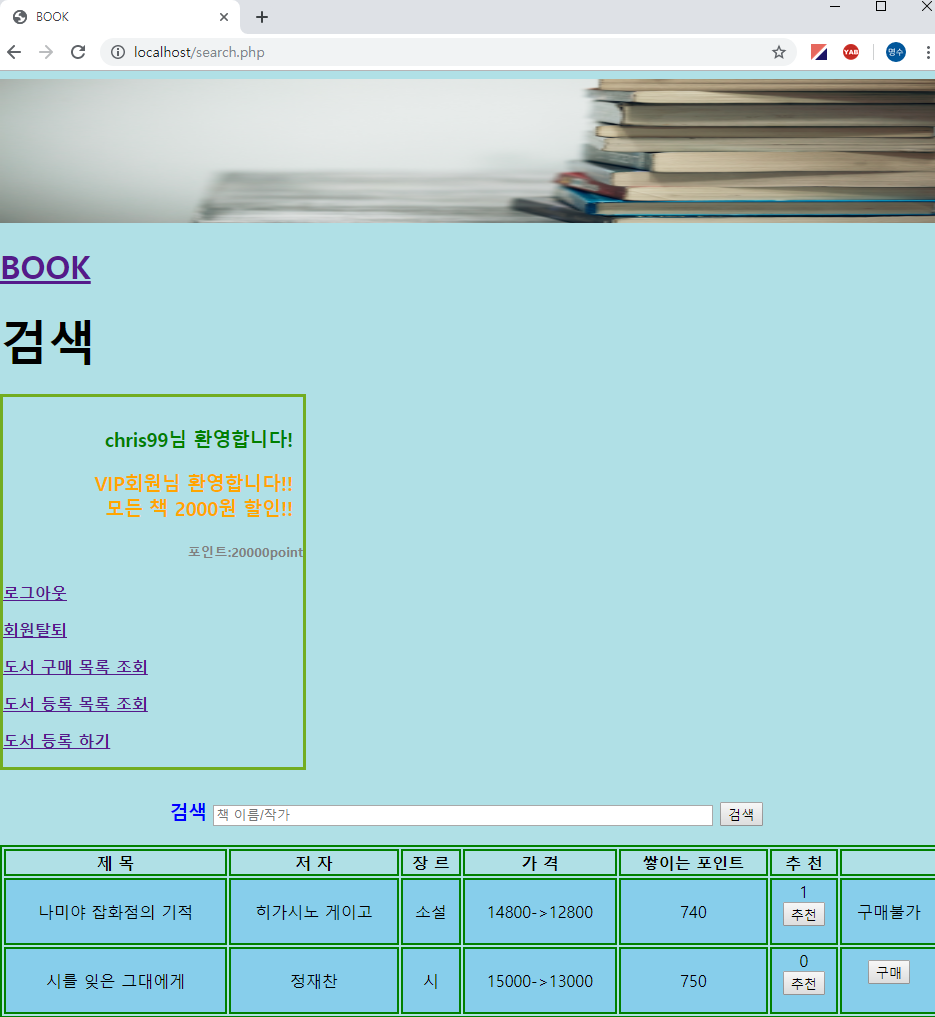
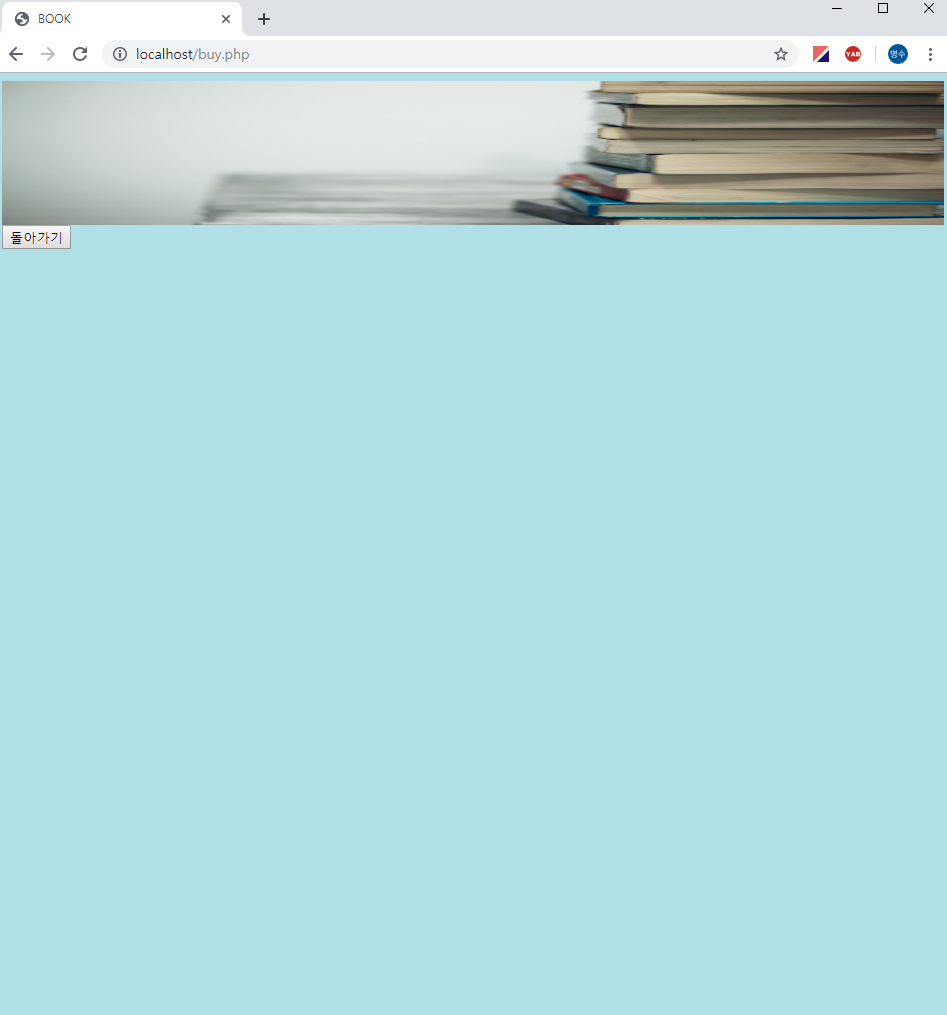


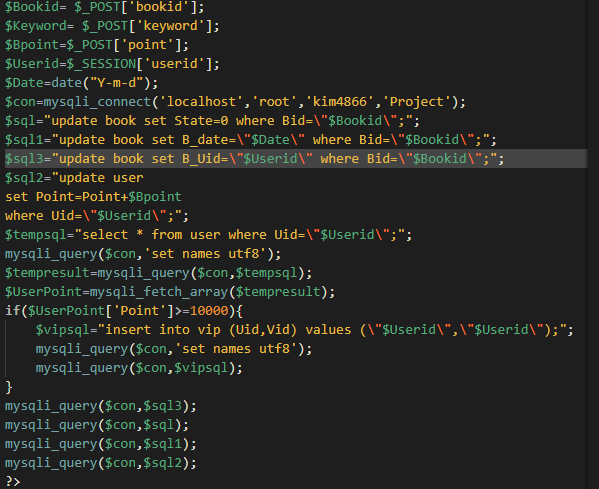
책 추천 코드



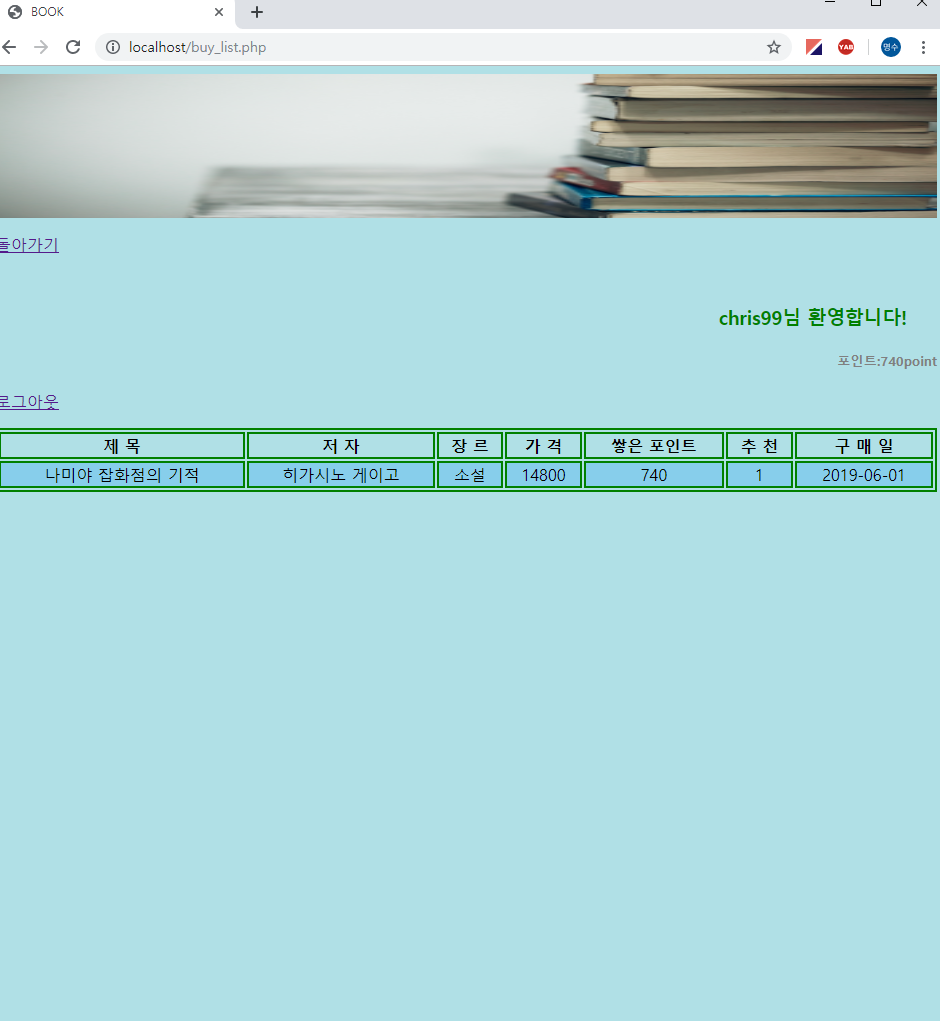
다음으로 도서 구매, 구매 도서 목록 조회 기능을 구현했습니다. 로그인을 한 뒤 검색을 합니다. Search.php에서 ‘구매’ 버튼을 눌러 해당 책을 구매할 수 있습니다. 버튼을 누르면 해당 책의 Bid, 검색 키워드, 쌓이는 포인트 정보가 buy\_check.php로 전달됩니다. Buy\_check에서는 해당 책을 정말로 구매할 것인지 재차 확인합니다. ‘아니요’를 누르면 검색 키워드를 다시 search.php로 전달하여 이전에 검색한 결과를 다시 보여줄 수 있도록 구현했습니다. ‘예’를 누르면 buy.php로 해당 책의 정보와 회원의 정보가 전달되어 구매를 할 수 있습니다. 구매를 하면 쌓이는 포인트만큼 회원의 포인트가 증가하고 해당 책의 State가 0이 됩니다. 이후로 해당 책은 구매 불가 상태가 됩니다. 또한, 구매버튼을 누른 시점의 날짜를 BOOK 테이블에서 해당 Bid 튜플의 B\_date 값을 저장하고 회원의 id를 B\_Uid에 저장합니다. “구매완료”라는 알림이 뜨고 확인을 누르면 ‘돌아가기 버튼이 있는 buy.php로 이동합니다. ‘돌아가기’ 버튼을 누르면 검색 키워드를 search.php로 전달하여 이전에 검색한 결과를 다시 보여줄 수 있도록 구현했습니다.

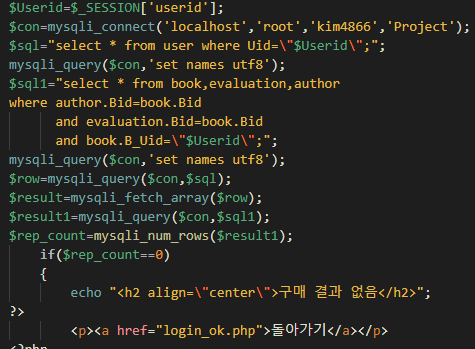


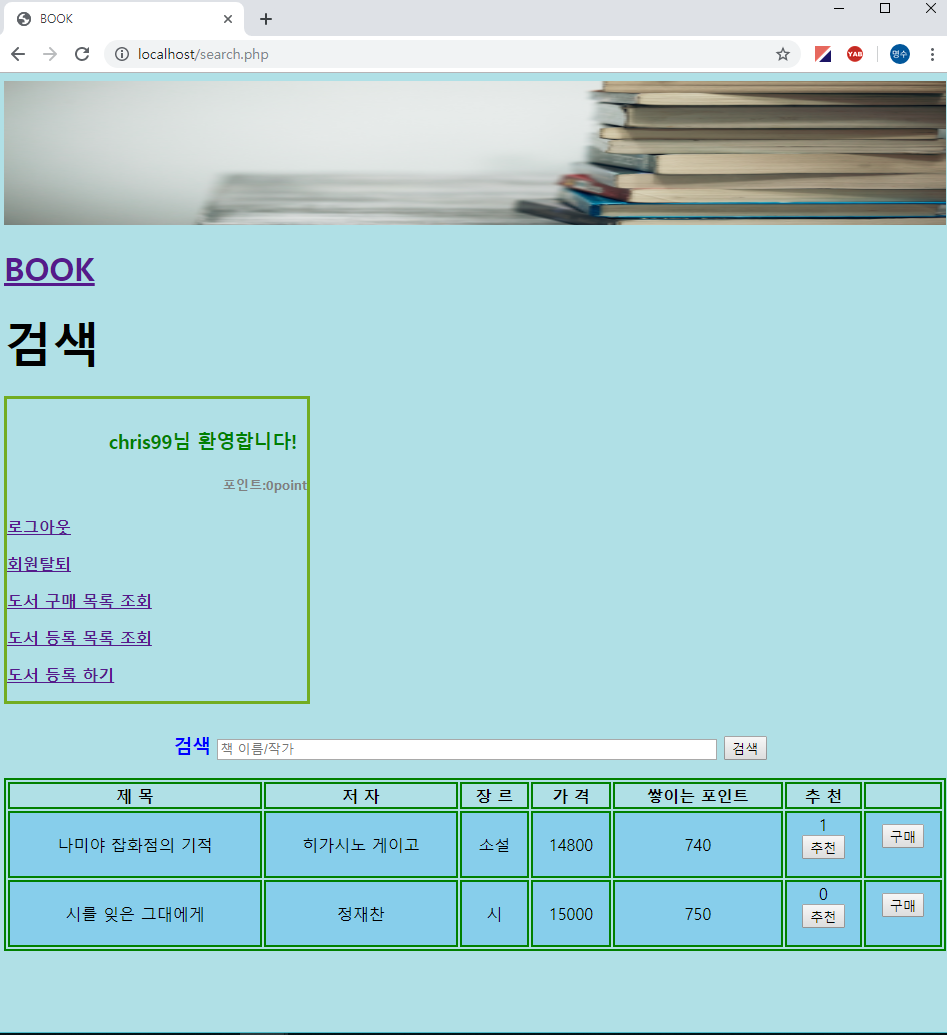




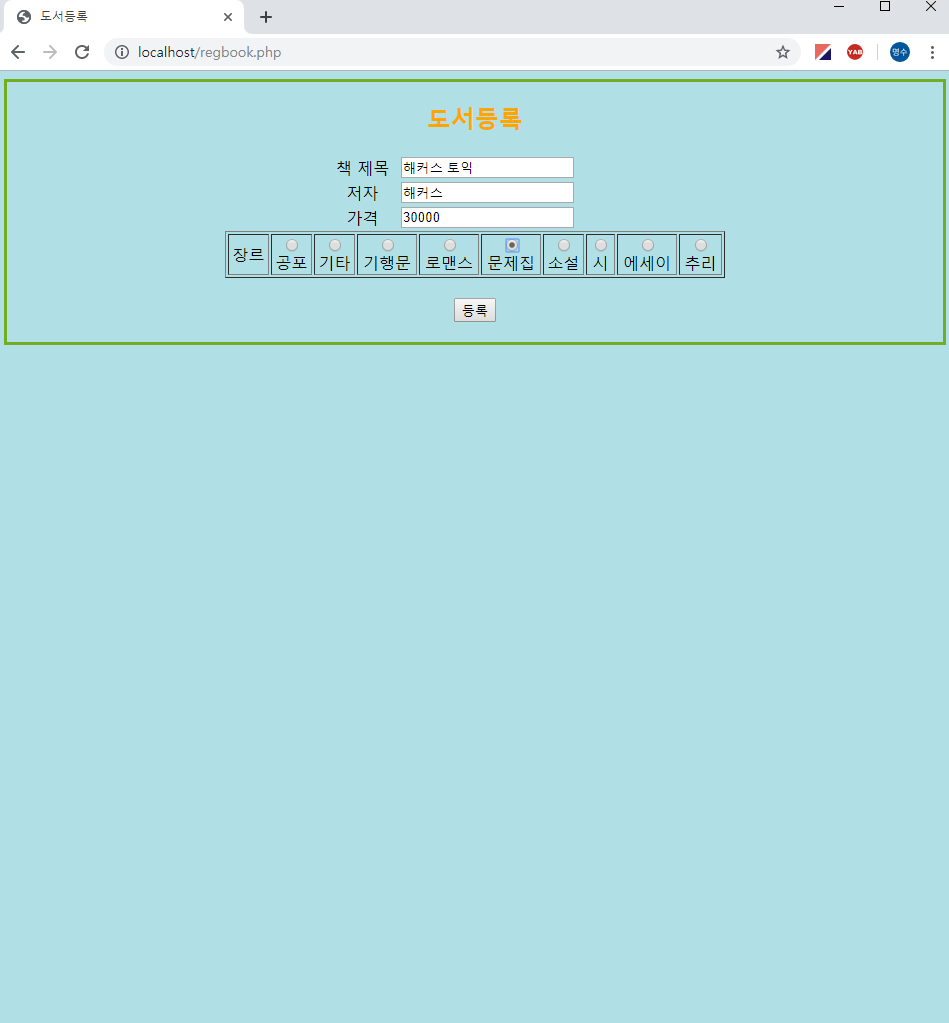
구매 코드- 구매한 책의 정보를 업데이트합니다.

구매 리스트는 로그인을 한 후 search.php 또는 login\_ok.php에서 ‘도서 구매 목록 조회’를 클릭하여 조회할 수 있습니다. ‘도서 구매 목록 조회’를 클릭하면 회원의 id가 buy\_list.php로 전달됩니다. buy\_list.php에서는 BOOK테이블에서 B\_Uid와 전달받은 id가 같은 튜플을 찾고 그러한 튜플들의 Bid를 추출하여 AUTHOR, EVALUATION테이블에서 Author\_name, Num\_of\_Good 정보를 얻어냅니다. 

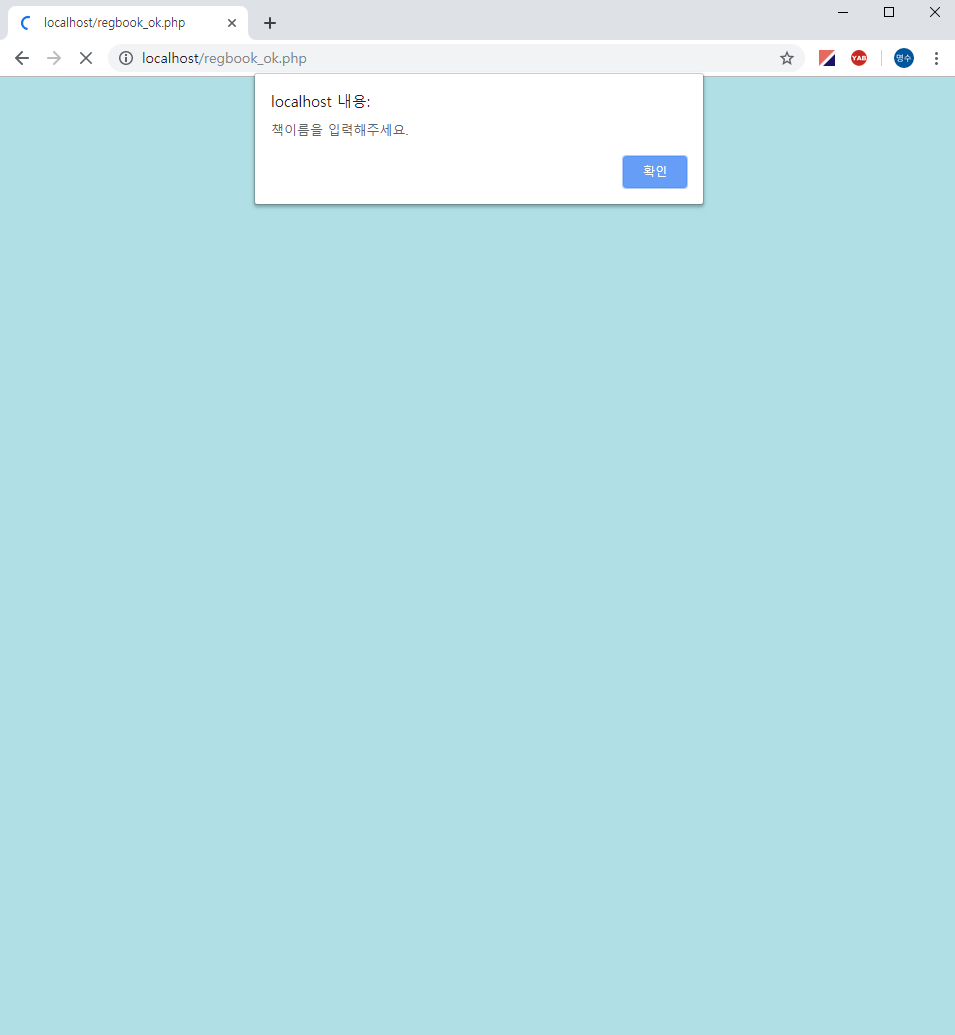


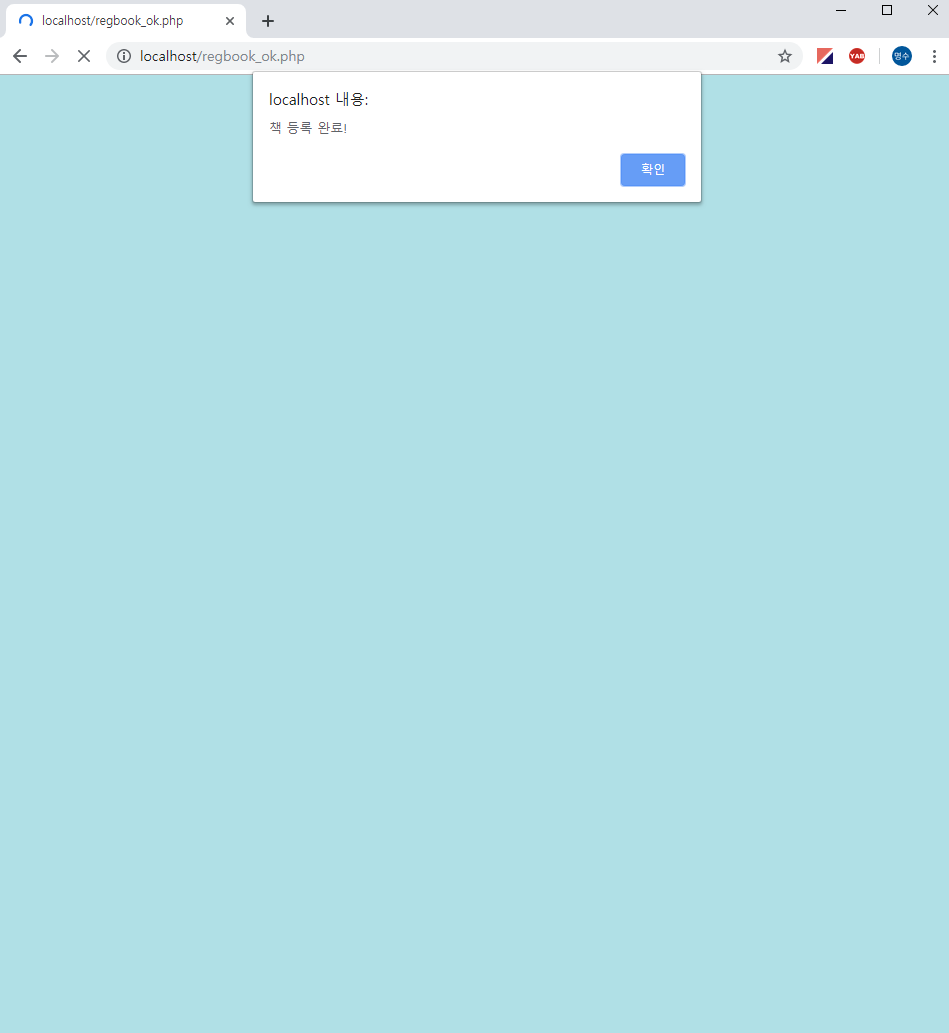
구매 목록 조회-$result1에 구매 도서의 데이터들이 있습니다.

다음은 도서 등록, 등록 도서 조회 기능입니다. 로그인을 한 후 search.php 또는 login\_ok.php에서 ‘도서 등록 하기’를 클릭하면 regbook.php 로 이동하면서 회원의 id가 전달됩니다.

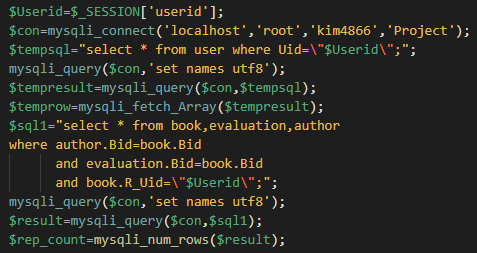
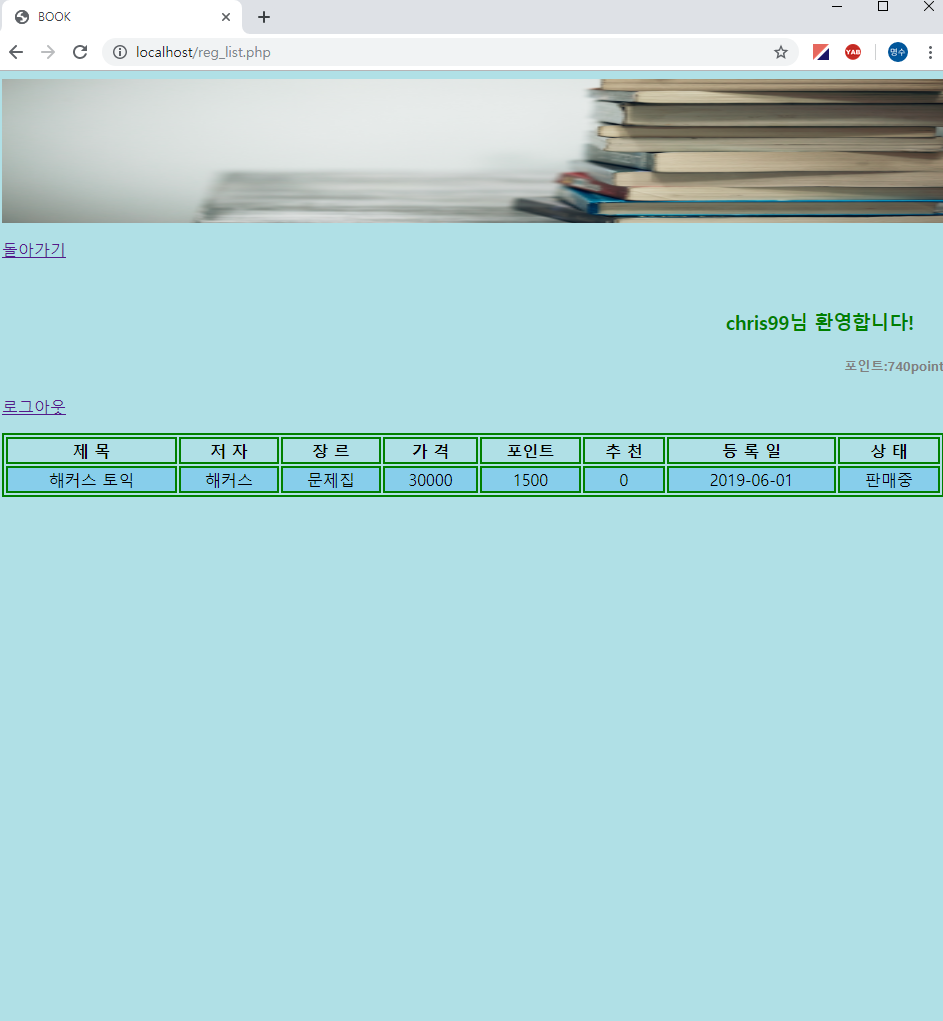


regbook.php에는 책 제목, 저자, 가격, 장르를 입력할 수 있는 필드가 있습니다. 어느 한 필드라도 값이 입력되지 않고 ‘등록’버튼을 누르면 해당 필드에 값을 입력해 달라는 알림과 함께 regbook.php로 돌아옵니다. 모든 필드에 값을 입력하고 ‘등록’버튼을 누르면 필드의 정보와 회원의 id, 등록한 시점의 날짜가 regbook\_pk.php로 전달됩니다. regbook\_ok.php에서는 전달받은 값들을 알맞은 테이블에 저장합니다. BOOK 테이블의 Bname에 책의 제목이, Price에 책의 가격이 Bpoint에는 책 가격의 5%만큼이, Bcategory에는 책의 장르가, State는 1, R\_date에는 date함수를 이용하여 얻은 등록한 시점의 날짜가 저장이 됩니다. 이후 “책 등록 완료”라는 알림과 함께 책 등록이 완료됩니다.



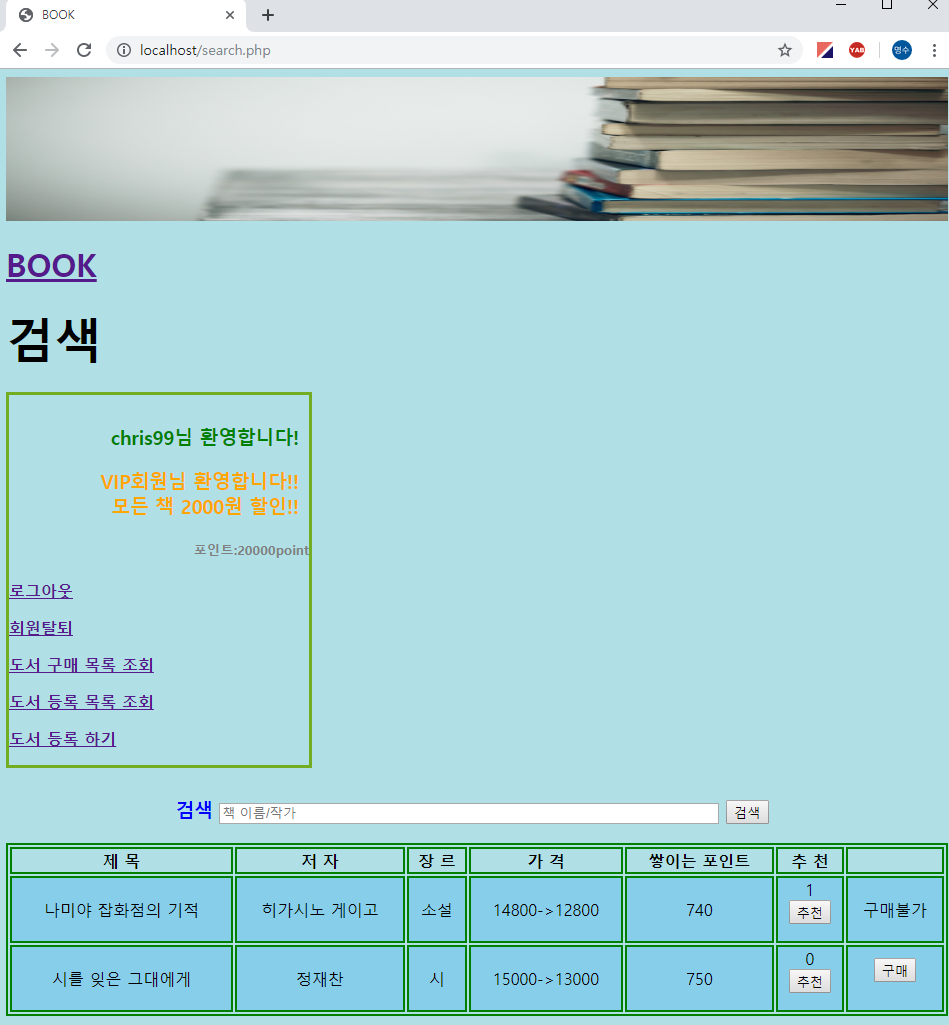


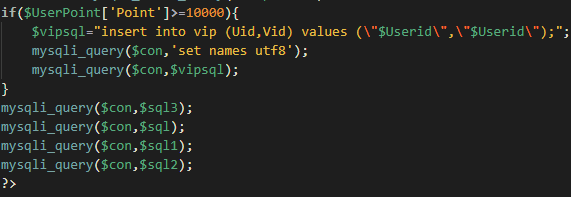
등록한 도서 목록은 로그인을 한 후 search.php 또는 login\_ok.php에서 ‘구매 도서 목록 조회’를 클릭하여 조회할 수 있습니다. ‘구매 도서 목록 조회’를 클릭하면 회원의 id가 reg\_list.php로 전달됩니다. reg\_list.php에서는 buy\_list.php에서와 비슷하게 BOOK테이블의 R\_Uid와 전달받은 회원의 id가 같은 행을 찾고 그 행들의 Bid에 대응하는 AUTHOR, EVALUATION테이블의 행을 읽어옵니다.



등록 도서 목록 코드- $result에 등록 도서들의 데이터가 있습니다.

마지막으로 회원의 포인트제로 인한 Vip 회원 기능입니다. 포인트가 10000점 이상이면 자동으로 Vip회원이 되어 VIP 테이블에 회원의 id가 저장됩니다. Vip 회원이 되면 모든 책을 2000원 할인된 가격에 구매할 수 있습니다. buy.php에서 책을 구매하여 포인트가 적립될 때 마다 회원의 누적 포인트가 10000점 이상인지 체크를 하고 10000점 이상이라면 VIP회원에 등록합니다.





buy.php에서 포인트를 적립할 때 마다 Vip회원 자격이 있는지 체크