

BookFinder

2020-2학기 웹DB프로그래밍
중간고사 과제

교 수: 왕보현 교수님
학 과: 컴퓨터공학과
학 번: 201533792
이 름: 유석환

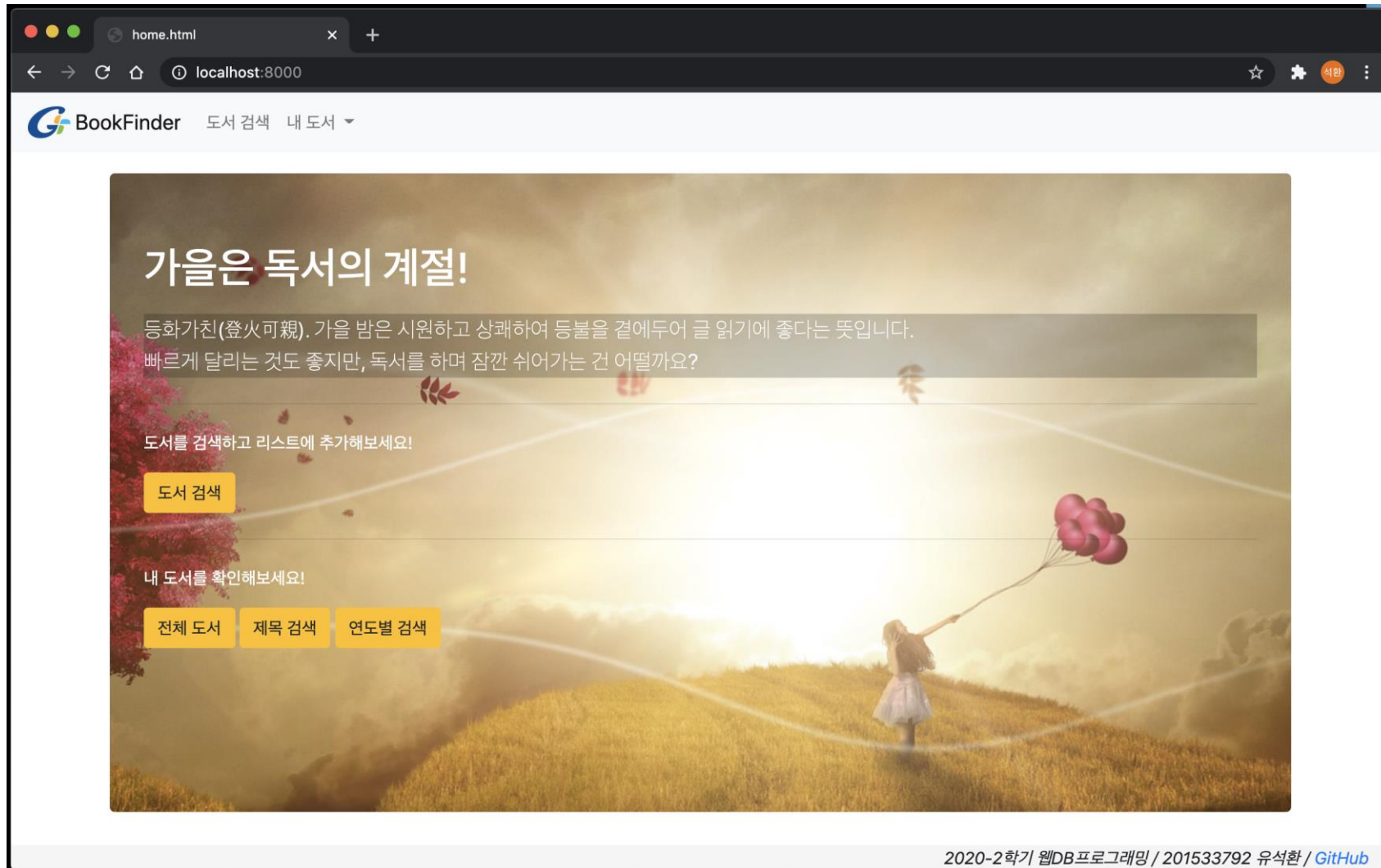
목차

1. 주제 및 App 구성	3p
2. 실행화면	4p
3. Model 설계	15p
4. URL 구성	16p
5. View 구현	19p
6. Form 구현	25p
7. HTML 구현	26p

1. 주제 및 App 구성

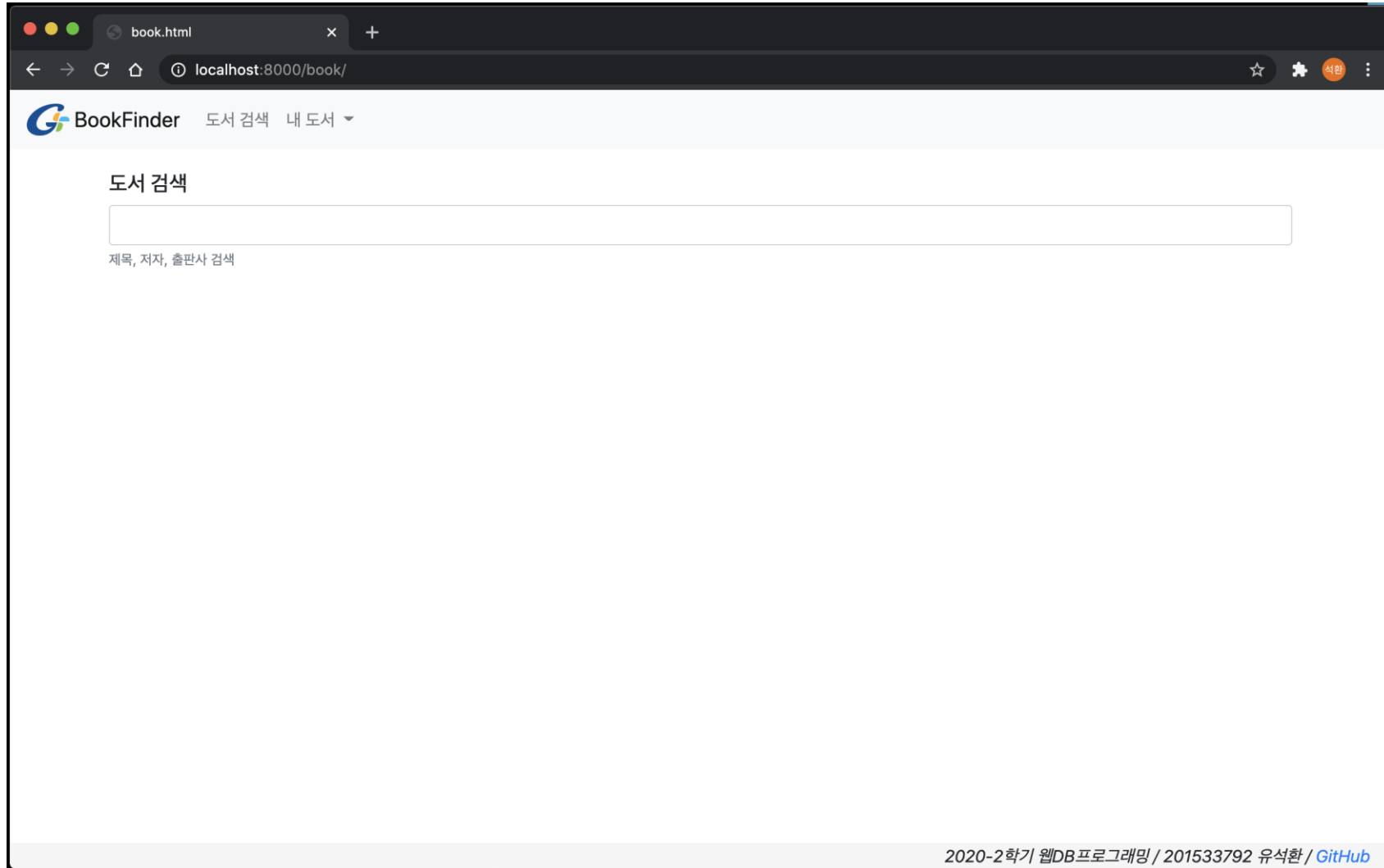
- 주제: **도서 검색 사이트**
- book과 mypage 2가지 App으로 구성
 - book: 네이버 책 API를 활용하여 실제 도서를 검색하고,
원하는 도서는 '내 도서'에 추가
 - mypage: '내 도서' 목록 조회 및 검색

2. 실행화면(1/11)



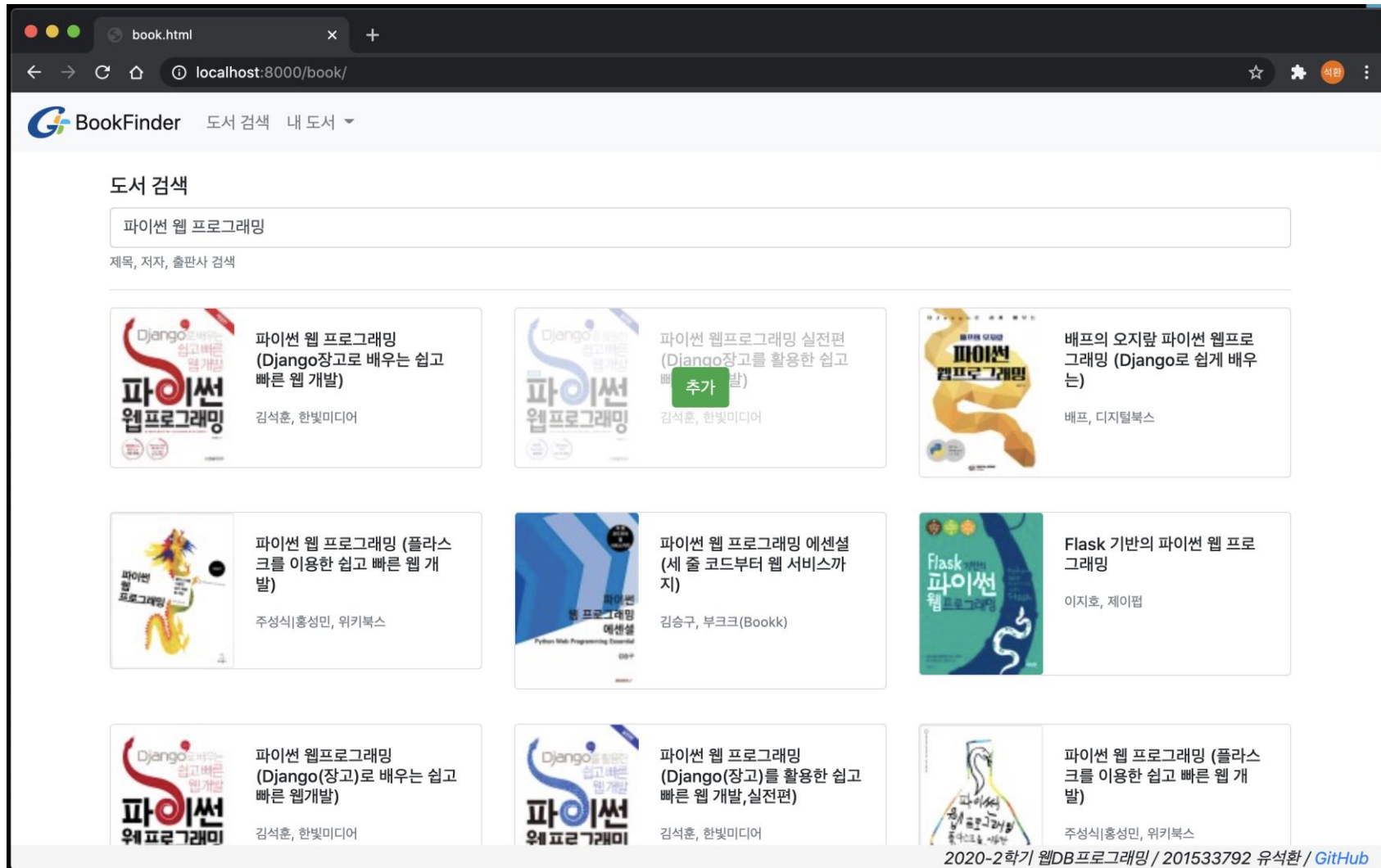
홈 화면

2. 실행화면(2/11)



도서 검색

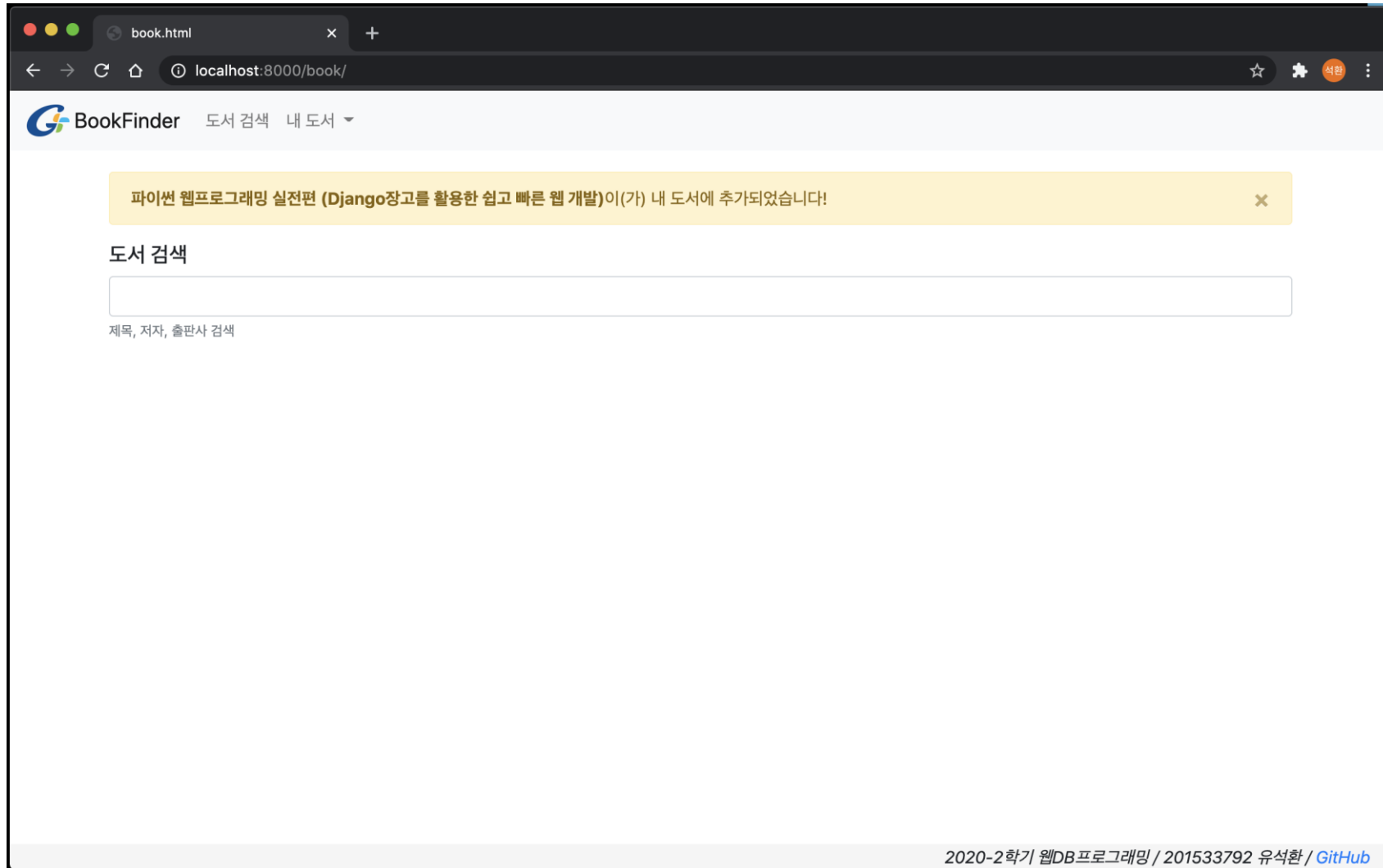
2. 실행화면 (3/11)



* hover시 '추가' 버튼 표시

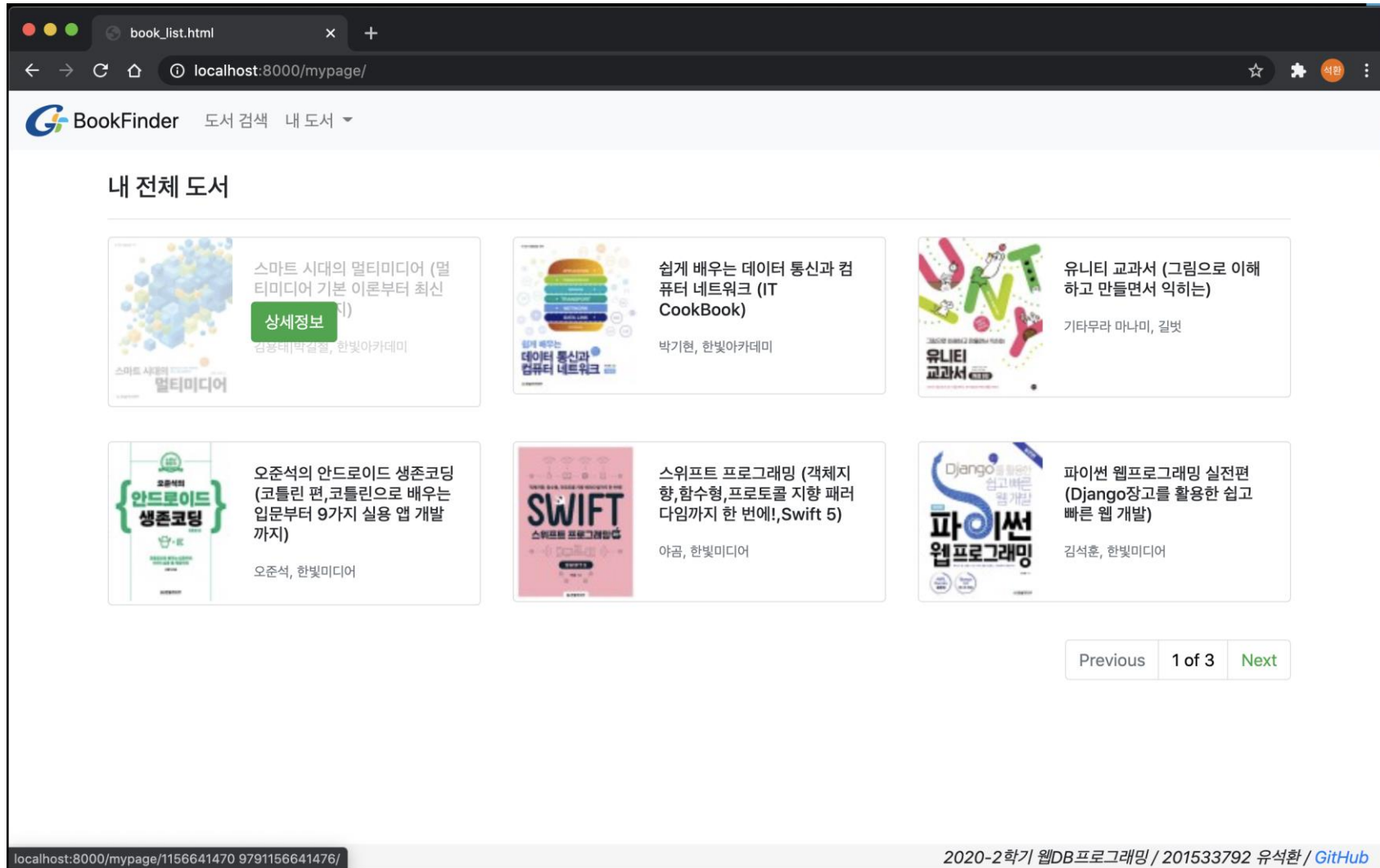
도서 검색

2. 실행화면(4/11)



도서 검색

2. 실행화면(5/11)



* hover시 '상세정보' 버튼 표시

내 전체 도서


2. 실행화면 (6/11)

book_detail.html

localhost:8000/mypage/1162242264%209791162242261/

BookFinder 도서 검색 내 도서

← 상세정보



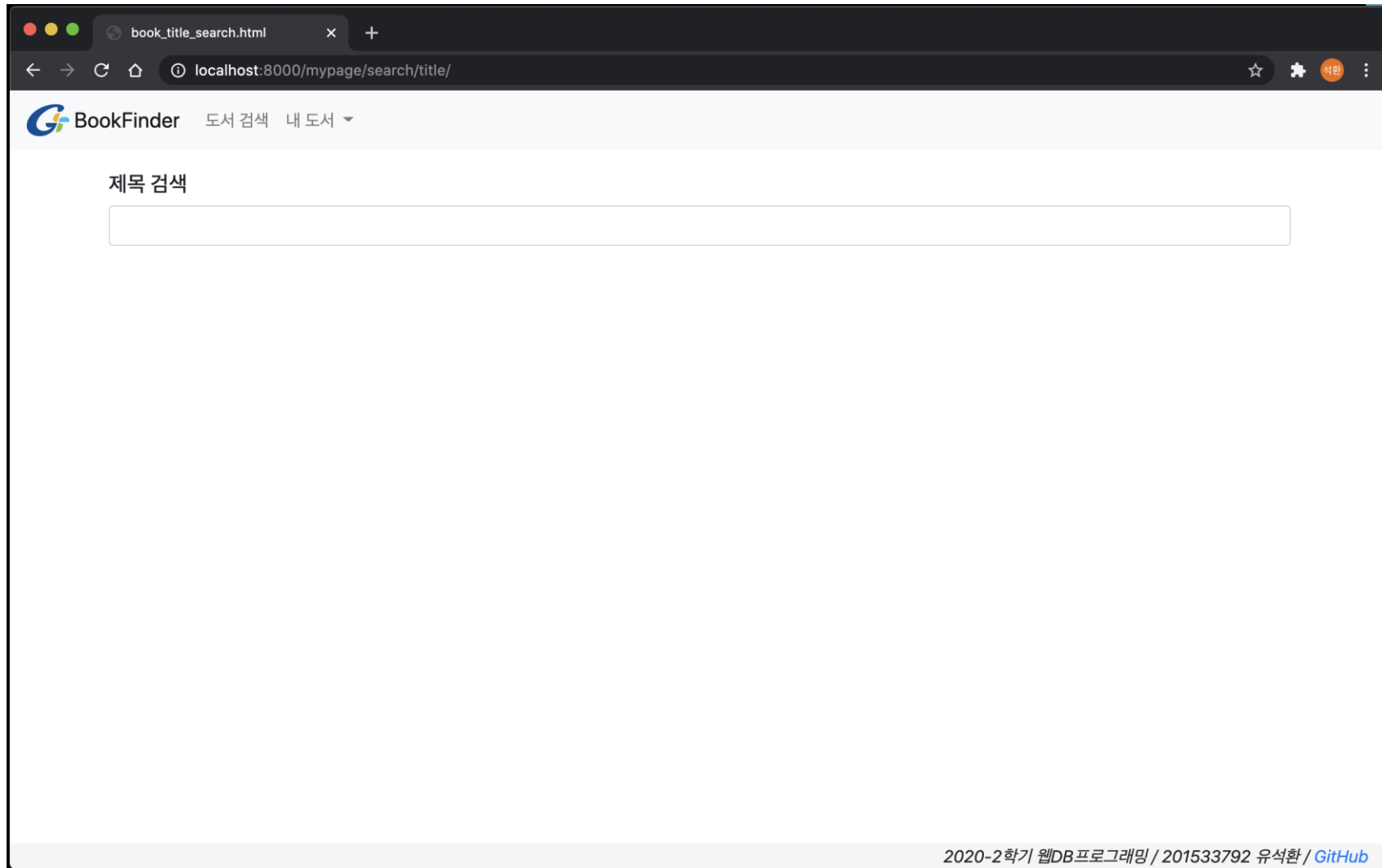
파이썬 웹프로그래밍 실전편 (Django장고를 활용한 쉽고 빠른 웹 개발)
김석훈

출판사	출판일
한빛미디어	2019년 11월 1일
ISBN	정가
1162242264 9791162242261	32000원
설명	
이 책이 제시하는 핵심 내용장고 프레임워크를 사용해서 상용화 수준의 앱을 만들고, 앱의 기능을 확장하고, 클라우드 서버에서 배포하는 것까지 다룬다.	

2020-2학기 웹DB프로그래밍 / 201533792 유석환 / [GitHub](#)

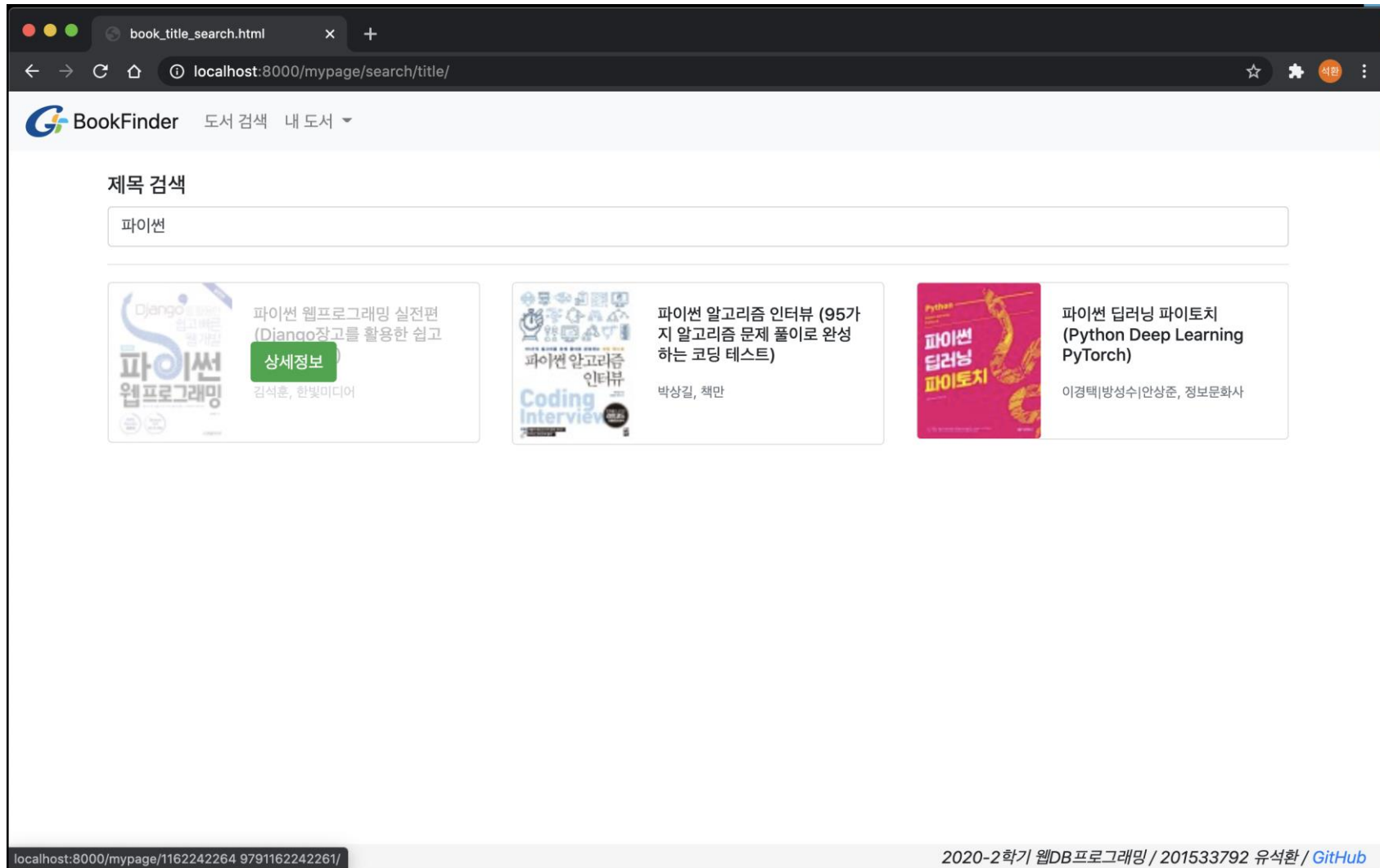
상세정보

2. 실행화면(7/11)



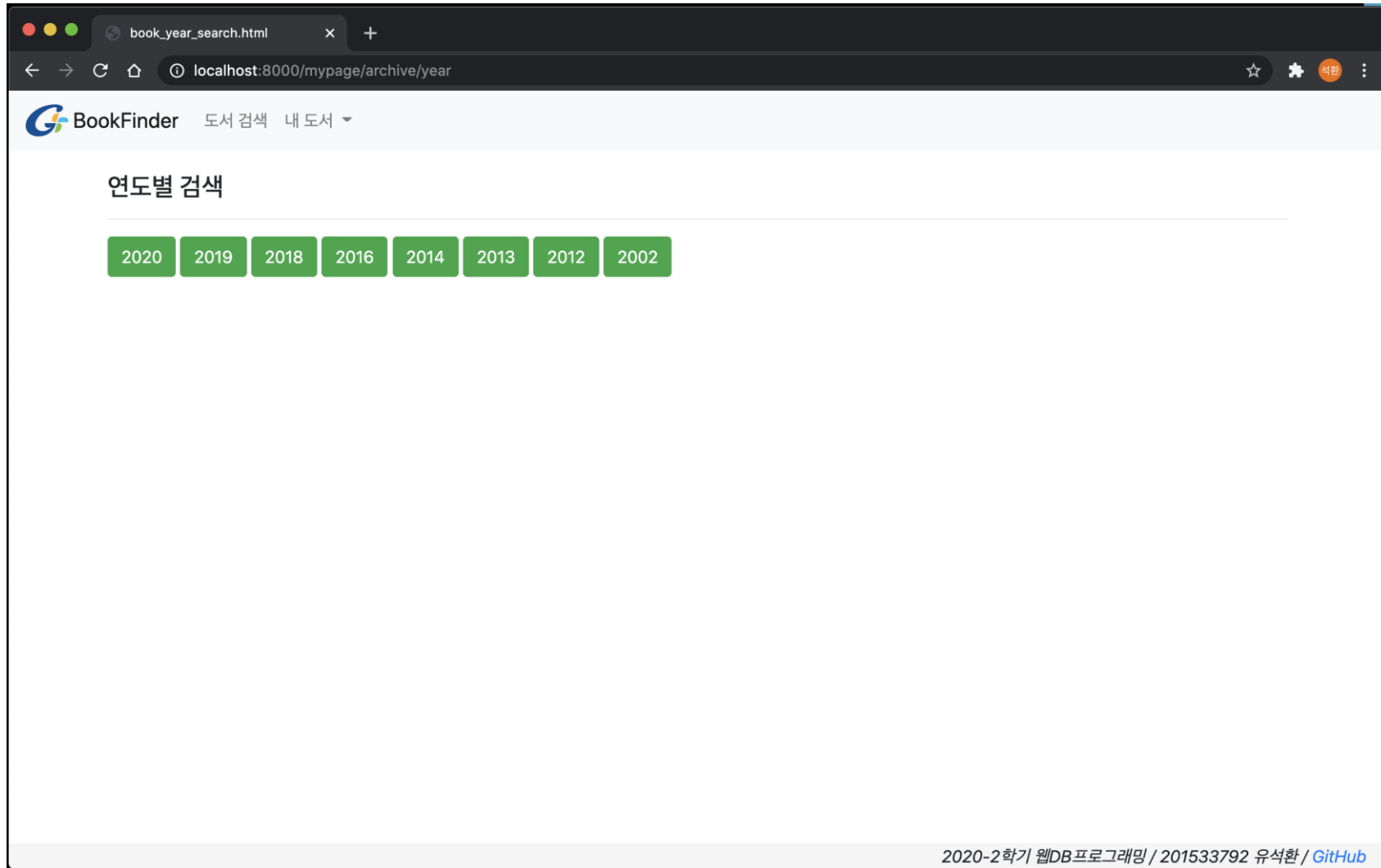
제목 검색

2. 실행화면 (8/11)



제목 검색

2. 실행화면(9/11)



연도별 검색


2. 실행화면(10/11)

book_archive_year.html x +


localhost:8000/mypage/archive/2020/

BookFinder 도서 검색 내 도서 ▾


← 2020년에 출간된 도서



파이썬 딥러닝 파이토치
(Python Deep Learning PyTorch)
이경택|방성수|안상준, 정보문화사



파이썬 알고리즘 인터뷰 (95가지 알고리즘 문제 풀이로 완성하는 코딩 테스트)
박상길, 책만



오준석의 플러터 생존 코딩(소문난 명강의) (Flutter와 Dart 입문부터 안드로이드와 iOS용 3가지 앱 개발까지)
오준석, 한빛미디어

Previous 1 of 1 Next

2020-2학기 웹DB프로그래밍 / 201533792 유석환 / [GitHub](#)

연도별 검색

2. 실행화면(11/11)

The screenshot shows the Django admin interface for a book management system. The browser address bar indicates the URL is `localhost:8000/admin/book/book/`. The page title is "Django 관리". The left sidebar contains a "BOOK" section with a "Books" link and a "+ 추가" button. Below this, there are sections for "인증 및 권한" (Authentication and Permissions) with links for "그룹" (Groups) and "사용자(들)" (Users), each with a "+ 추가" button. The main content area is titled "변경할 Book 선택" (Select book to change). It features a search bar with a dropdown menu and a "실행" (Execute) button. Below the search bar, there is a table listing books with columns for ISBN, TITLE, AUTHOR, PUBLISHER, and PUBDATE. The table contains 10 rows of book data. A "+ BOOK 추가" button is located in the top right corner of the main content area.

ISBN	TITLE	AUTHOR	PUBLISHER	PUBDATE
1156641470 9791156641476	스마트 시대의 멀티미디어 (멀티미디어 기본 이론부터 최신 기술 동향까지)	김용태 박길철	한빛아카데미	2014년 12월 15일
115664304X 9791156643043	쉽게 배우는 데이터 통신과 컴퓨터 네트워크 (IT CookBook)	박기현	한빛아카데미	2016년 12월 26일
1160509417 9791160509410	유니티 교과서 (그림으로 이해하고 만들면서 익히는)	기타무라 마나미	갈벗	2019년 10월 30일
1162241195 9791162241196	오준석의 안드로이드 생존코딩 (코틀린 편, 코틀린으로 배우는 입문부터 9가지 실용 앱 개발까지)	오준석	한빛미디어	2018년 10월 1일
1162242221 9791162242223	스위프트 프로그래밍 (객체지향, 함수형, 프로토콜 지향 패러다임까지 한 번에!, Swift 5)	야곰	한빛미디어	2019년 10월 1일
1162242264 9791162242261	파이썬 웹프로그래밍 실전편 (Django장고를 활용한 쉽고 빠른 웹 개발)	김석훈	한빛미디어	2019년 11월 1일
1162242833 9791162242834	오준석의 플러터 생존 코딩(소문난 명강의) (Flutter와 Dart 입문부터 안드로이드와 iOS용 3가지 앱 개발까지)	오준석	한빛미디어	2020년 3월 1일
1163030848 9791163030843	Do it! 코틀린 프로그래밍	황영덕	이지스퍼블리싱	2019년 5월 28일
1163031275 9791163031277	Do it! 스위프트로 아이폰 앱 만들기 입문	이범근 송호정	이지스퍼블리싱	2019년 12월 23일
1185123466 9791185123462	리눅스마스터 1급 정복하기 (1차·2차 시험대비)	정성재 배유미	북스출력퍼블리싱	2019년 7월 17일

* ID/PW는 교안대로
bhwang99
bh123456 입니다.

admin

3. Model 설계

book 앱과 mypage 앱에서 공통으로 사용하는 Book 모델

```
class Book(models.Model):
    isbn = models.CharField('ISBN', max_length=50, primary_key=True) # ISBN
    title = models.CharField('TITLE', max_length=200) # 도서 제목
    author = models.CharField('AUTHOR', max_length=200) # 저자
    publisher = models.CharField('PUBLISHER', max_length=100) # 출판사
    pubdate = models.DateField('PUBDATE') # 출간일
    price = models.IntegerField('PRICE', default=0) # 정가
    description = models.CharField('DESCRIPTION', max_length=1000) # 줄거리
    image = models.URLField('IMAGE', max_length=200) # 썸네일 URL
```

book/models.py

4. URL 구성(1/3)

```
urlpatterns = [  
    # admin  
    path('admin/', admin.site.urls),  
  
    # root  
    path('', HomeView.as_view(), name='home'),  
  
    # app  
    path('book/', include('book.urls')),  
    path('mypage/', include('mypage.urls')),  
]
```

→ admin

→ home 화면

→ book 앱 연결

→ mypage 앱 연결

midterm/urls.py

4. URL 구성(2/3)

```
app_name = 'book'
urlpatterns = [
    # root
    path('', BookSearchFormView.as_view(), name='index'),

    # 내 도서 추가
    path('insert/', BookInsertAndSearchFormView.as_view(), name='insert')
]
```

book/urls.py

→ 1. 도서 검색 Form 출력
2. 검색 결과 출력

→ 1. '내 도서'에 추가
2. 도서 검색 Form 출력

4. URL 구성(3/3)

```
app_name = 'mypage'
urlpatterns = [
    # root - 전체 도서 목록
    path('', BookLV.as_view(), name='index'),

    # DetailView - 도서 상세정보
    path('<str:pk>/', BookDV.as_view(), name='detail'),

    # FormView - 제목 검색
    path('search/title/', TitleSearchFormView.as_view(), name='title_search'),

    # ArchiveIndexView - 연도별 검색
    path('archive/year', BookAV.as_view(), name='year_search'),

    # YearArchiveView - 해당 연도의 도서 목록
    path('archive/<int:year>/', BookYAV.as_view(), name='year_archive')
]
```

→ '내 도서' 전체 목록 조회

→ 도서 상세정보 조회

→ 제목으로 검색

→ 연도별 검색
(해당되는 연도 출력)

→ 선택한 연도에 출간된 도서 조회

mypage/urls.py

5. View 구현 (1/6)

```
class HomeView(TemplateView):  
    template_name = 'home.html'
```

midterm/views.py

TemplateView를 사용하여 홈 화면을 출력합니다.

5. View 구현 (2/6)

도서 검색 Form에서 keyword를 받고
그 keyword로 네이버 책 API를 호출하여
도서 목록을 출력하는 FormView입니다.

```
11 class BookSearchFormView(FormView):
12     form_class = BookSearchForm
13     template_name = 'book/book.html'
14
15     def form_valid(self, form):
16         keyword = form.cleaned_data['keyword']
17         url = 'https://openapi.naver.com/v1/search/book.json?query='\
18             + parse.quote(keyword) + '&display=30&start=1'
19
20         # 네이버 책 API 호출
21         api_request = request.Request(url)
22         api_request.add_header('X-Naver-Client-Id', 'qk7TJRgtBZFSBy3pIoIN')
23         api_request.add_header('X-Naver-Client-Secret', 'J2IQ57jCr0')
24         response = request.urlopen(api_request)
```

```
25
26
27     if response.getcode() == 200:
28         body = response.read()
29         result = json.loads(body.decode('utf-8'))
30         items = result.get('items')
31         pprint(items)
32
33         # 문자열에서 HTML 태그 제거
34         for item in items:
35             item['title'] = removeHtmlTag(item['title'])
36             item['author'] = removeHtmlTag(item['author'])
37             item['publisher'] = removeHtmlTag(item['publisher'])
38             item['description'] = removeHtmlTag(item['description'])
39
40         # 이미지가 없을 경우 대체 이미지 표시
41         if item['image'] == '':
42             item['image'] = '/static/img/no-image.png'
43
44     context = {
45         'form': form,
46         'keyword': keyword,
47         'object_list': items,
48     }
49
50     return render(self.request, self.template_name, context)
51
52     elif response.getcode() == 400:
53         pprint("잘못된 쿼리입니다.")
54
55     elif response.getcode() == 500:
56         pprint("네이버 API 서버 오류입니다.")
57
58     else:
59         pprint("알 수 없는 에러: %s" % response.getcode())
```

book/views.py

5. View 구현 (3/6)

```
class BookInsertAndSearchFormView(TemplateView):
    template_name = 'book/book.html'

    # 도서 추가
    def post(self, req):
        try:
            book = Book(
                isbn=req.POST['isbn'],
                title=req.POST['title'],
                author=req.POST['author'],
                publisher=req.POST['publisher'],
                pubdate=datetime.datetime.strptime(req.POST['pubdate'], "%Y%m%d").date(),
                price=req.POST['price'],
                description=req.POST['description'],
                image=req.POST['image'],
            )
            book.save()

            # success message
            messages.info(req, req.POST['title'])
        finally:
            return redirect('/book')
```

book/views.py

‘추가’ 버튼을 클릭했을 때
Book 모델에 데이터를 추가(Insert)합니다.

이후 /book으로 redirect하여 검색창을 띄우되,
Django의 message 객체를 이용하여
어떤 도서를 추가했는지 Alert를 띄웁니다.
(7p 실행화면 참고)

5. View 구현 (4/6)

```
# ListView
class BookLV(ListView):
    model = Book
    template_name = 'mypage/book_list.html'
    paginate_by = 6

# DetailView
class BookDV(DetailView):
    model = Book
    template_name = 'mypage/book_detail.html'
```

mypage/views.py

내 전체 도서를 띄우는 ListView와
'상세정보'를 클릭하면 출력되는 DetailView입니다.

5. View 구현 (5/6)

```
# FormView
class TitleSearchFormView(FormView):
    form_class = TitleSearchForm
    template_name = 'mypage/book_title_search.html'

    def form_valid(self, form):
        title = form.cleaned_data['title']
        book_list = Book.objects.filter(Q(title__icontains=title)).distinct()

        context = {
            'form': form,
            'search_term': title,
            'object_list': book_list
        }

        return render(self.request, self.template_name, context)
```

mypage/views.py

도서를 제목으로 검색하는
FormView입니다.

5. View 구현 (6/6)

```
# ArchiveIndexView
class BookAV(ArchiveIndexView):
    model = Book
    template_name = 'mypage/book_year_search.html'
    date_field = 'pubdate'
    allow_empty = True

# YearArchiveView
class BookYAV(YearArchiveView):
    model = Book
    template_name = 'mypage/book_archive_year.html'
    date_field = 'pubdate'
    make_object_list = True
    paginate_by = 6
```

mypage/views.py

연도별 검색에 사용되는 ArchiveIndexView와 YearArchiveView 입니다.

BookAV에서 데이터가 있는 연도를 모두 표시하고, 버튼을 클릭하면 해당되는 연도에 출간된 도서를 BookYAV를 통해 표시합니다.

6. Form 구현

```
class BookSearchForm(forms.Form):  
    keyword = forms.CharField(  
        widget = forms.TextInput(attrs={'class': 'form-control'}),  
        label = '',  
    )
```

book/forms.py

네이버 책 API를 호출하기 위한 Form입니다.

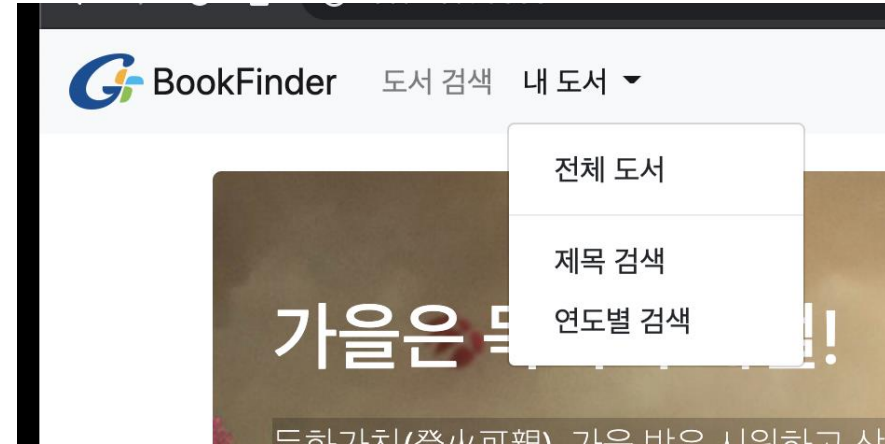
```
class TitleSearchForm(forms.Form):  
    title = forms.CharField(  
        widget = forms.TextInput(attrs={'class': 'form-control'}),  
        label = '',  
    )
```

mypage/forms.py

Book 모델의 title 항목을 기준으로
검색하는 Form입니다.

7. HTML 구현 (1/12)

```
{# 상단 Navbar #}  
<nav class="navbar navbar-expand-lg navbar-light bg-light">  
  {# 로고 #}  
  <a class="navbar-brand" href="{% url 'home' %}">  
      
    BookFinder  
  </a>  
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav"  
    aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">  
    <span class="navbar-toggler-icon"></span>  
  </button>  
  {# 각 App 메뉴 #}  
  <div class="collapse navbar-collapse" id="navbarNav">  
    <ul class="navbar-nav">  
      {# book 앱 #}  
      <li class="nav-item">  
        <a class="nav-link" href="{% url 'book:index' %}">도서 검색</a>  
      </li>  
      {# mypage 앱 #}  
      <li class="nav-item dropdown">  
        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button"  
          data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">  
          내 도서  
        </a>  
        <div class="dropdown-menu" aria-labelledby="navbarDropdown">  
          <a class="dropdown-item" href="{% url 'mypage:index' %}">전체 도서</a>  
          <div class="dropdown-divider"></div>  
          <a class="dropdown-item" href="{% url 'mypage:title_search' %}">제목 검색</a>  
          <a class="dropdown-item" href="{% url 'mypage:year_search' %}">연도별 검색</a>  
        </div>  
      </li>  
    </ul>  
  </div>  
</nav>
```



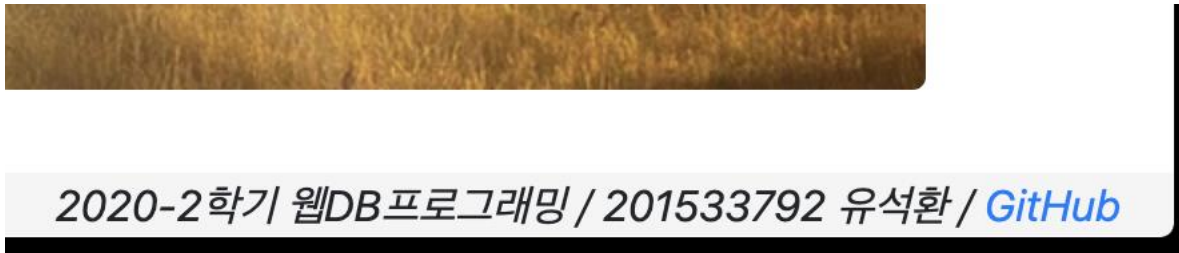
base.html의 Navbar 구현입니다.

templates/base.html

7. HTML 구현 (2/12)

```
{# footer #}  
<footer class="fixed-bottom bg-info">  
  <div class="font-italic text-right footer">  
    2020-2학기 웹DB프로그래밍 / 201533792 유석환 / <a href="https://github.com/youseokhwan/BookFinder" target="_blank">GitHub</a>  
  </div>  
</footer>
```

templates/base.html



base.html의 footer 구현입니다.

생략된 base.html의 나머지 부분들은 Bootstrap CDN, CSS 등입니다.

7. HTML 구현(3/12)

```
{% block content %}
    {# 메인 이미지 #}
    <div class="jumbotron">
        <h1 class="display-5">가을은 독서의 계절!</h1>
        <p class="lead">
            등화가친(登火可親). 가을 밤은 시원하고 상쾌하여 등불을 곁에두어 글 읽기에 좋다는 뜻입니다.<br>
            빠르게 달리는 것도 좋지만, 독서를 하며 잠깐 쉬어가는 건 어떨까요?
        </p>
        <hr class="my-4">
        <p>도서를 검색하고 리스트에 추가해보세요!</p>
        <a class="btn btn-warning" href="{% url 'book:index' %}" role="button">도서 검색</a>
        <hr class="my-4">
        <p>내 도서를 확인해보세요!</p>
        <a class="btn btn-warning" href="{% url 'mypage:index' %}" role="button">전체 도서</a>
        <a class="btn btn-warning" href="{% url 'mypage:title_search' %}" role="button">제목 검색</a>
        <a class="btn btn-warning" href="{% url 'mypage:year_search' %}" role="button">연도별 검색</a>
    </div>
{% endblock %}
```

templates/home.html

Jumbotron을 이용한 Home 화면 구현입니다.



7. HTML 구현(4/12)

```
{# 도서 검색 form #}  
<form action="" method="post">  
  <h5 for="book-search">도서 검색</h5>  
  {% csrf_token %}  
  {{ form.as_table }}  
  <button type="submit" class="btn btn-warning" hidden>검색</button>  
  <small id="book-search-desc" class="form-text text-muted">  
    제목, 저자, 출판사 검색  
  </small>  
</form>
```

book/templates/book/book.html

도서를 검색하기 위한 Form 부분입니다.
(Model에서 데이터를 검색하는 Form이 아닌 API 호출을 위한 Form입니다.)

도서 검색

제목, 저자, 출판사 검색

7. HTML 구현 (5/12)

```
{# 검색 결과 #}
{% if object_list %}
<hr>

{# 도서 목록 #}
<div class="row row-cols-2 row-cols-md-3">
  {% for item in object_list %}
    <div class="col mb-3">
      <div class="card mb-3" style="...">
        <div class="row no-gutters">
          {# 썸네일 이미지 #}
          <div class="col-md-4">
            
          </div>

          {# 제목, 저자, 출판사 정보 #}
          <div class="col-md-8">
            <div class="card-body">
              <h6 class="card-title">{{ item.title }}</h6>
              <p class="card-text text-truncate"><small class="text-muted">{{ item.author }}, {{ item.publisher }}</small>
            </div>
          </div>
        </div>
      </div>
    </div>
  {% endfor %}
</div>
```

book/templates/book/book.html

API 호출 후 response의 결과를
Card 형식으로 출력합니다.

도서 검색

안드로이드

제목, 저자, 출판사 검색



Do it! 안드로이드 앱 프로그래밍
정재곤, 이지스퍼블리싱



Android Studio를 활용한 안드로이드 프로그래밍
우재남(박길식, 한빛아카데미)



이것이 안드로이드다
이것이 안드로이드다 with 코틀린 (안드로이드 입문의 3가지 장벽, 언어+실전+환경 완벽 대응!)
고돈호, 한빛미디어



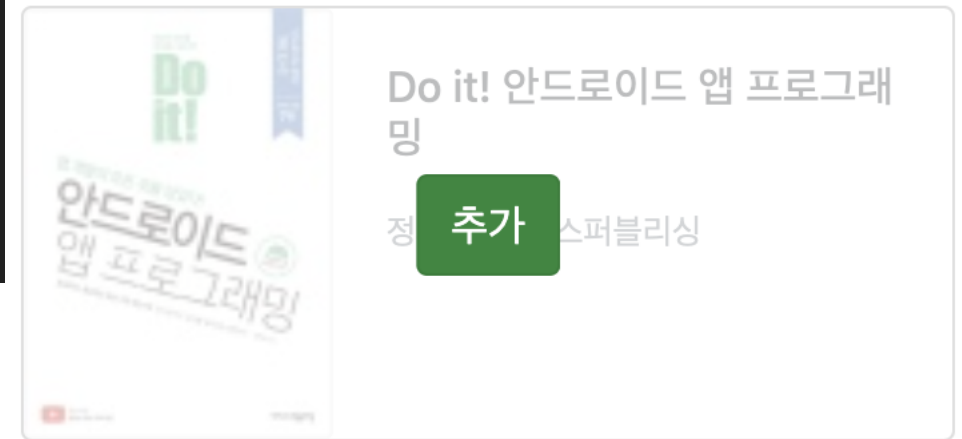
안드로이드 프로그래밍(그림으로 쉽게 설명하는)(개정판 5판) (개정5판)
천인국, 생능출판

7. HTML 구현 (6/12)

```
{# hover시 '추가' 버튼 표시 #}  
<div class="middle">  
  <form action="{% url 'book:insert' %}" method="post">  
    {% csrf_token %}  
    <input type="text" name="isbn" value="{{ item.isbn }}" hidden>  
    <input type="text" name="title" value="{{ item.title }}" hidden>  
    <input type="text" name="author" value="{{ item.author }}" hidden>  
    <input type="text" name="publisher" value="{{ item.publisher }}" hidden>  
    <input type="text" name="pubdate" value="{{ item.pubdate }}" hidden>  
    <input type="text" name="price" value="{{ item.price }}" hidden>  
    <input type="text" name="description" value="{{ item.description }}" hidden>  
    <input type="text" name="image" value="{{ item.image }}" hidden>  
    <button type="submit" class="btn btn-success">추가</button>  
  </form>  
</div>  
</div>  
</div>  
{% endfor %}  
</div>  
{% endif %}  
{% endblock %}
```

book/templates/book/book.html

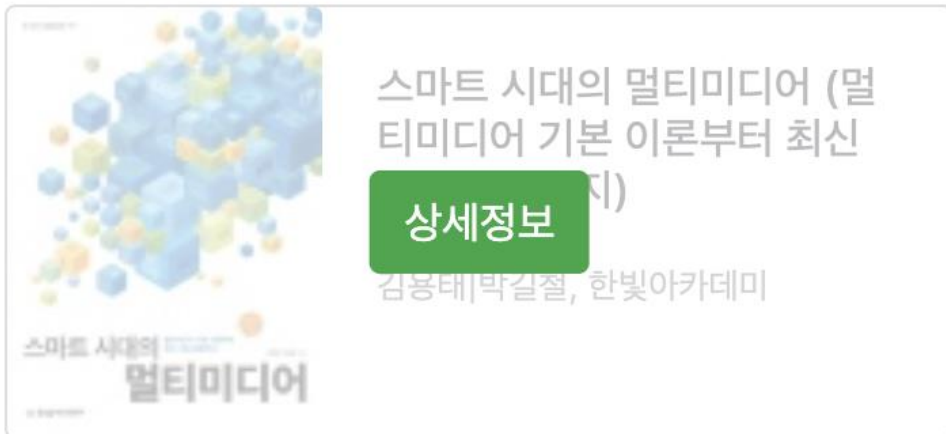
마우스를 hover하면 아래와 같이
'추가'버튼이 표시되고, 버튼을 클릭하면
'book/Insert/' URL이 호출되어
Book 모델에 해당 데이터를 Insert합니다.



7. HTML 구현(7/12)

```
{# hover시 '추가' 버튼 표시 #}  
<div class="middle">  
  <a class="btn btn-outline-success" href="{{ item.get_absolute_url }}">상세정보</a>  
</div>
```

mypage/templates/mypage/book_list.html



'내 도서'의 목록을 출력하는 화면입니다.
Card 구현은 거의 비슷하여 생략하였고,
앞과 다르게 '추가' 버튼이 아닌 '상세정보' 버튼이 표시됩니다.
클릭하면 상세정보 페이지로 이동합니다.

7. HTML 구현 (8/12)

```
{# 페이지네이션 #}  
<nav aria-label="Page navigation">  
  <ul class="pagination justify-content-end">  
    {# 이전 페이지 #}  
    {% if page_obj.has_previous %}  
      <li class="page-item">  
        <a class="page-link text-success" href="?page={{ page_obj.previous_page_number }}">Previous</a>  
      </li>  
    {% else %}  
      <li class="page-item disabled" data-toggle="tooltip" data-placement="bottom" title="첫 페이지입니다">  
        <a class="page-link" href="#">Previous</a>  
      </li>  
    {% endif %}  
  
    {# 현재 페이지 of 전체 페이지 #}  
    <li class="page-item disabled">  
      <a class="page-link" href="#" style="...">{{ page_obj.number }} of {{ page_obj.paginator.num_pages }}</a>  
    </li>  
  
    {# 다음 페이지 #}  
    {% if page_obj.has_next %}  
      <li class="page-item">  
        <a class="page-link text-success" href="?page={{ page_obj.next_page_number }}">Next</a>  
      </li>  
    {% else %}  
      <li class="page-item disabled" data-toggle="tooltip" data-placement="bottom" title="마지막 페이지입니다">  
        <a class="page-link" href="#">Next</a>  
      </li>  
    {% endif %}  
  </ul>  
</nav>
```

Pagination 구현입니다.

내 전체 도서



Previous 1 of 3 Next

첫 페이지입니다

mypage/templates/mypage/book_list.html

7. HTML 구현 (9/12)

```
29 {% block content %}
30     {# 제목 #}
31     <h4>
32         {# 뒤로가기 버튼 #}
33         <a class="text-success" href="{% url 'mypage:index' %}"
34             data-toggle="tooltip" data-placement="bottom" title="전체 도서" style="..."></a>
35         상세정보
36     </h4>
37     <hr>
38     {# 도서 정보 #}
39     <div class="jumbotron">
40         <div class="row no-gutters">
41             {# 썸네일 #}
42             <div class="col img-col">
43                 
44             </div>
45             {# 제목, 저자 #}
46             <div class="col">
47                 <div class="card-body">
48                     <h4 class="card-title">{{ object.title }}</h4>
49                     <p class="card-text">{{ object.author }}</p>
50                 </div>
51             </div>
52         </div>
53         <hr>
54         <div class="row no-gutters">
55             {# 출판사 #}
56             <div class="col my-col">
57                 <label for="publisher">출판사</label>
58                 <input type="text" class="form-control" id="publisher" value="{{ object.publisher }}" readonly>
59             </div>
60             {# 출판일 #}
61             <div class="col my-col">
```

상세정보 페이지 구현입니다.

← 상세정보



파이썬 웹프로그래밍 실전편 (Django장고를 활용한 쉽고 빠른 웹 개발)

김석훈

출판사

한빛미디어

출판일

2019년 11월 1일

ISBN

1162242264 9791162242261

정가

32000원

설명

이 책이 제시하는 핵심 내용장고 프레임워크를 사용해서 상용화 수준의 앱을 만들고, 앱의 기능을 확장하고, 클라우드 서버에서 배포하는 것까지 다룬다.

mypage/templates/mypage/book_detail.html

7. HTML 구현(10/12)

```
{% block content %}
    {# 도서 검색 form #}
    <form action="" method="post">
        <h5 for="book-search">제목 검색</h5>
        {% csrf_token %}
        {{ form.as_table }}
        <button type="submit" class="btn btn-warning" hidden>검색</button>
    </form>

    {# 도서 목록 #}
    {% if object_list %}
        <hr>

        <div class="row row-cols-2 row-cols-md-3">
            {% for item in object_list %}
```

mypage/templates/mypage/book_title_search.html

Book 모델에서 제목(title)을 기준으로
검색하는 화면입니다.

제목 검색

파이썬



파이썬 웹프로그래밍 실전편
(Django장고를 활용한 쉽고
빠른 웹 개발)

김석훈, 한빛미디어



파이썬 알고리즘
인터뷰

박상길, 책만

7. HTML 구현(11/12)

```
{% block content %}
    {# 제목 #}
    <h4>연도별 검색</h4>
    <hr>
    {# 데이터가 있을 경우 #}
    {% if date_list %}
        {% for date in date_list %}
            <a class="btn btn-success"
                href="{% url 'mypage:year_archive' date|date:"Y" %}">{{ date|date:"Y" }}</a>
        {% endfor %}
    {# 데이터가 없을 경우 #}
    {% else %}
        <p>검색된 내 도서가 없습니다.</p>
        <p><a href="{% url 'book:index' %}">도서 검색</a>에서 도서를 추가해주세요!</p>
    {% endif %}
{% endblock %}
```

연도별로 도서를 검색하는 화면입니다.
해당 연도에 1개 이상의 데이터가 있으면
아래처럼 버튼으로 표시됩니다.

연도별 검색

mypage/templates/mypage/book_year_search.html

2020

2019

2018

2016

2014

2013

2012

2002

7. HTML 구현 (12/12)

```
{% block content %}
    {# 제목 #}
    <h4>
        <a class="text-success" href="{% url 'mypage:year_search' %}"
           data-toggle="tooltip" data-placement="bottom" title="연도별 검색" style="...">
            {{ year|date:"Y" }}년에 출간된 도서
        </a>
    </h4>
    <hr>
    {# 도서 목록 #}
    {% if object_list %}
        <div class="row row-cols-2 row-cols-md-3">
            {% for item in object_list %}
                <div class="col mb-3">
                    <div class="card mb-3" style="...">
```

mypage/templates/mypage/book_archive_year.html

YearArchiveView를 통해 해당 연도에 출간된 도서만 출력하는 화면입니다.

Card 출력 부분은 유사하여 생략하였습니다.

← 2019년에 출간된 도서



Do it! 스위프트로 아이폰 앱 만들기 입문

이범근|송호정, 이지스퍼블리싱



파이썬 (Django) 빠른

김석훈

감사합니다.