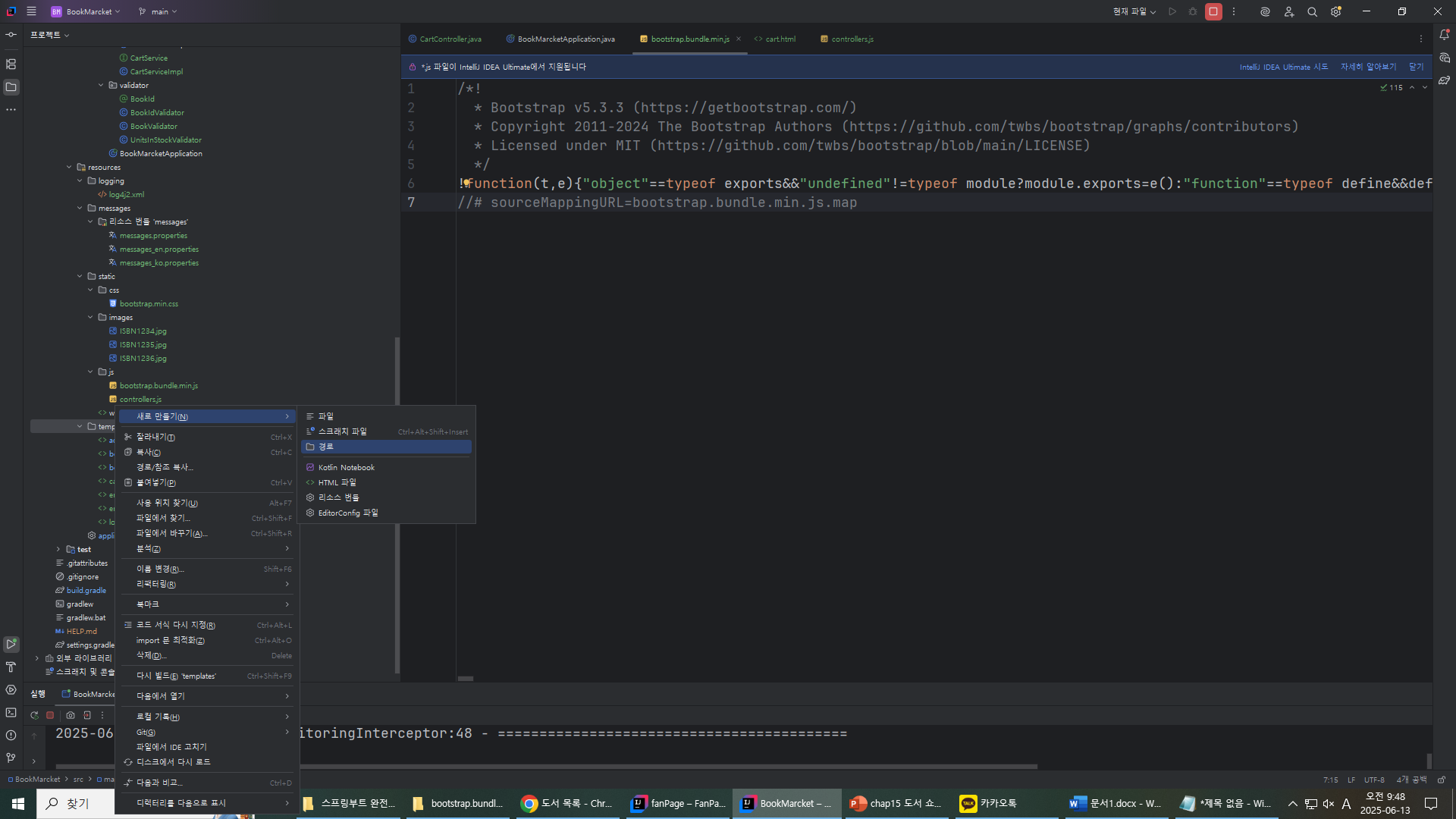


resources 폴더 안 Js폴더에

bootstrap.bundle.min.js 추가



Templates에 module폴더 만들기

Header.html, footer.html 만들기

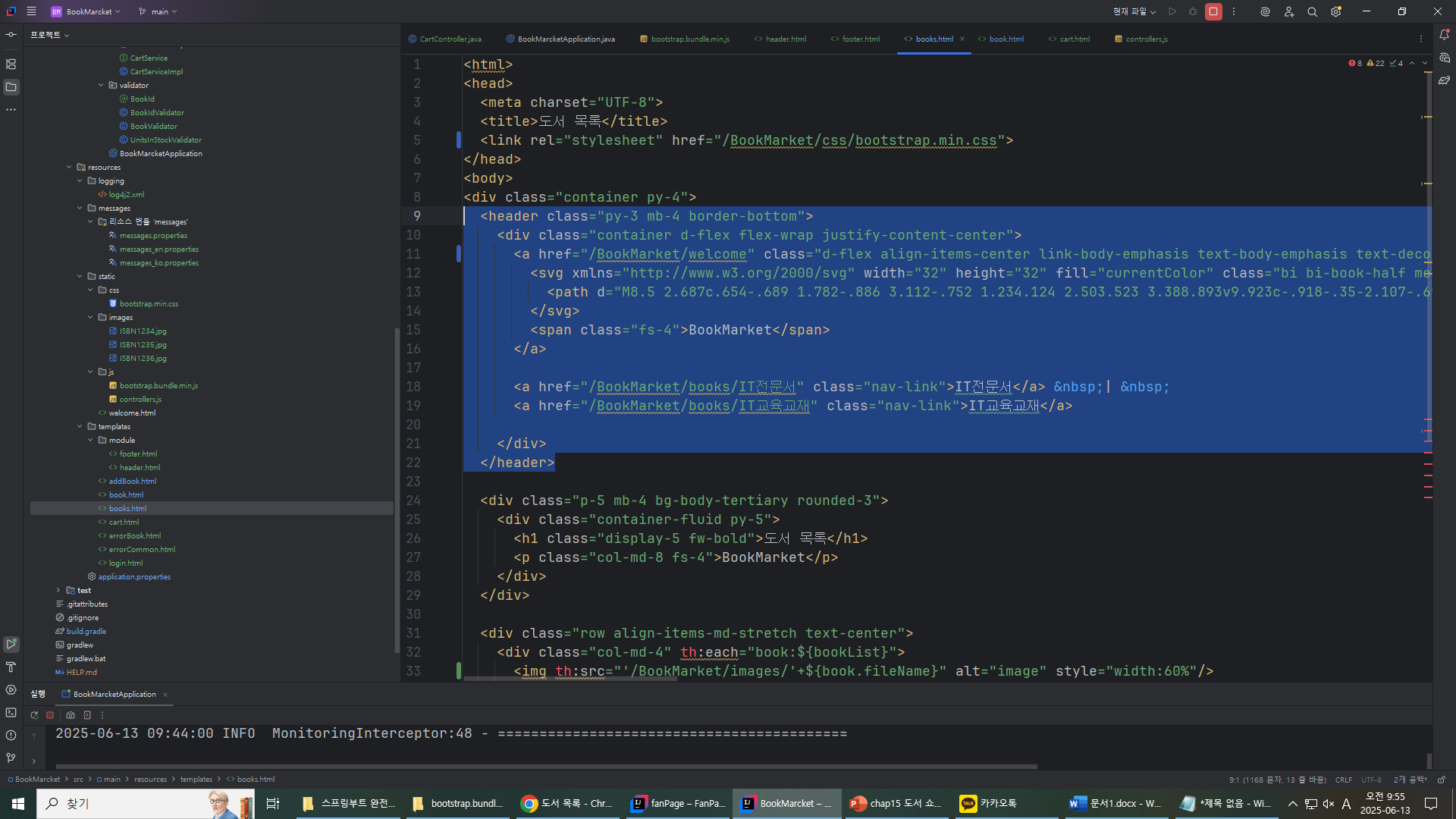
Header에 있는 내용 다 지우고 아래 내용 붙여넣기

|  |
| --- |
| <header class="py-3 mb-4 border-bottom">  <div class="container d-flex flex-wrap justify-content-center">  <a href="/BookMarket/home" class="d-flex align-items-center link-body-emphasis text-body-emphasis text-decoration-none mb-3 mb-lg-0 me-lg-auto">  <svg xmlns="http://www.w3.org/2000/svg" width="32" height="32" fill="currentColor" class="bi bi-book-half me-2" viewBox="0 0 16 16">  <path d="M8.5 2.687c.654-.689 1.782-.886 3.112-.752 1.234.124 2.503.523 3.388.893v9.923c-.918-.35-2.107-.692-3.287-.81-1.094-.111-2.278-.039-3.213.492zM8 1.783C7.015.936 5.587.81 4.287.94c-1.514.153-3.042.672-3.994 1.105A.5.5 0 0 0 0 2.5v11a.5.5 0 0 0 .707.455c.882-.4 2.303-.881 3.68-1.02 1.409-.142 2.59.087 3.223.877a.5.5 0 0 0 .78 0c.633-.79 1.814-1.019 3.222-.877 1.378.139 2.8.62 3.681 1.02A.5.5 0 0 0 16 13.5v-11a.5.5 0 0 0-.293-.455c-.952-.433-2.48-.952-3.994-1.105C10.413.809 8.985.936 8 1.783"/>  </svg>  <span class="fs-4">BookMarket</span>  </a>  <a class="nav-link dropdown-toggle" href="#" data-bs-toggle="dropdown" aria-expanded="false">도서목록</a>  <ul class="dropdown-menu">  <li><a class="dropdown-item" href="/BookMarket/books">전체도서</a></li>  <li><a class="dropdown-item" href="/BookMarket/books/IT전문서">IT전문서</a></li>  <li><a class="dropdown-item" href="/BookMarket/books/IT교육교재">IT교육교재</a></li>  </ul>  &nbsp;| &nbsp;</a>    <a sec:authorize="hasRole('ROLE\_ADMIN')" href="/BookMarket/books/add" class="nav-link">도서등록 &nbsp;| &nbsp; </a>  <a sec:authorize="isAuthenticated()" href="/BookMarket/cart" class="nav-link">장바구니 &nbsp;| &nbsp; </a>  <a sec:authorize="hasRole('ROLE\_ADMIN')" href="/BookMarket/order/list" class="nav-link">주문목록 &nbsp;| &nbsp; </a>  <a href="/BookMarket/board/list" class="nav-link">게시판 &nbsp;| &nbsp; </a>  <a sec:authorize="isAnonymous()" href="/BookMarket/members/add" class="nav-link">회원가입 &nbsp;| &nbsp;</a>  <a sec:authorize="isAnonymous()" href="/BookMarket/login" class="nav-link">로그인 &nbsp;| &nbsp;</a>  <a sec:authorize="isAuthenticated()" th:href="'/BookMarket/members/update/'+${session.userLoginInfo.memberId}" class="nav-link">회원수정 &nbsp;| &nbsp;</a>  <a sec:authorize="isAuthenticated()" href="/BookMarket/logout" class="nav-link">로그아웃 &nbsp;| &nbsp; </a>  <b sec:authorize="isAuthenticated()"><span sec:authentication='name'></span>님</b>    </div>  </header>  <script src="/BookMarket/js/bootstrap.bundle.min.js" rel="stylesheet" ></script> |

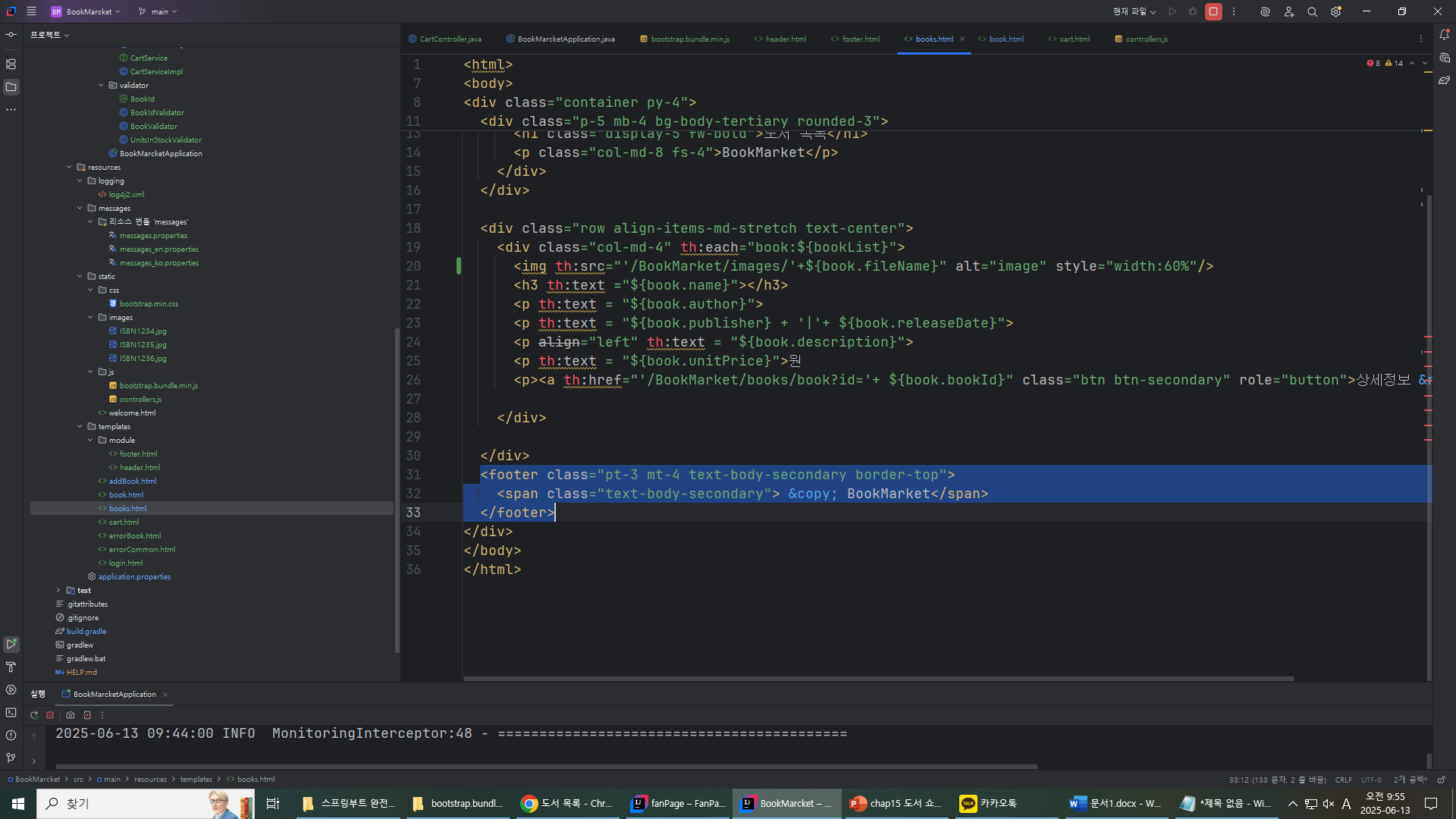
footer.html 내용

|  |
| --- |
| <footer class="pt-3 mt-4 text-body-secondary border-top">  <span class="text-body-secondary"> &copy; BookMarket</span>  </footer> |

Books.html 안에 <header> 제거

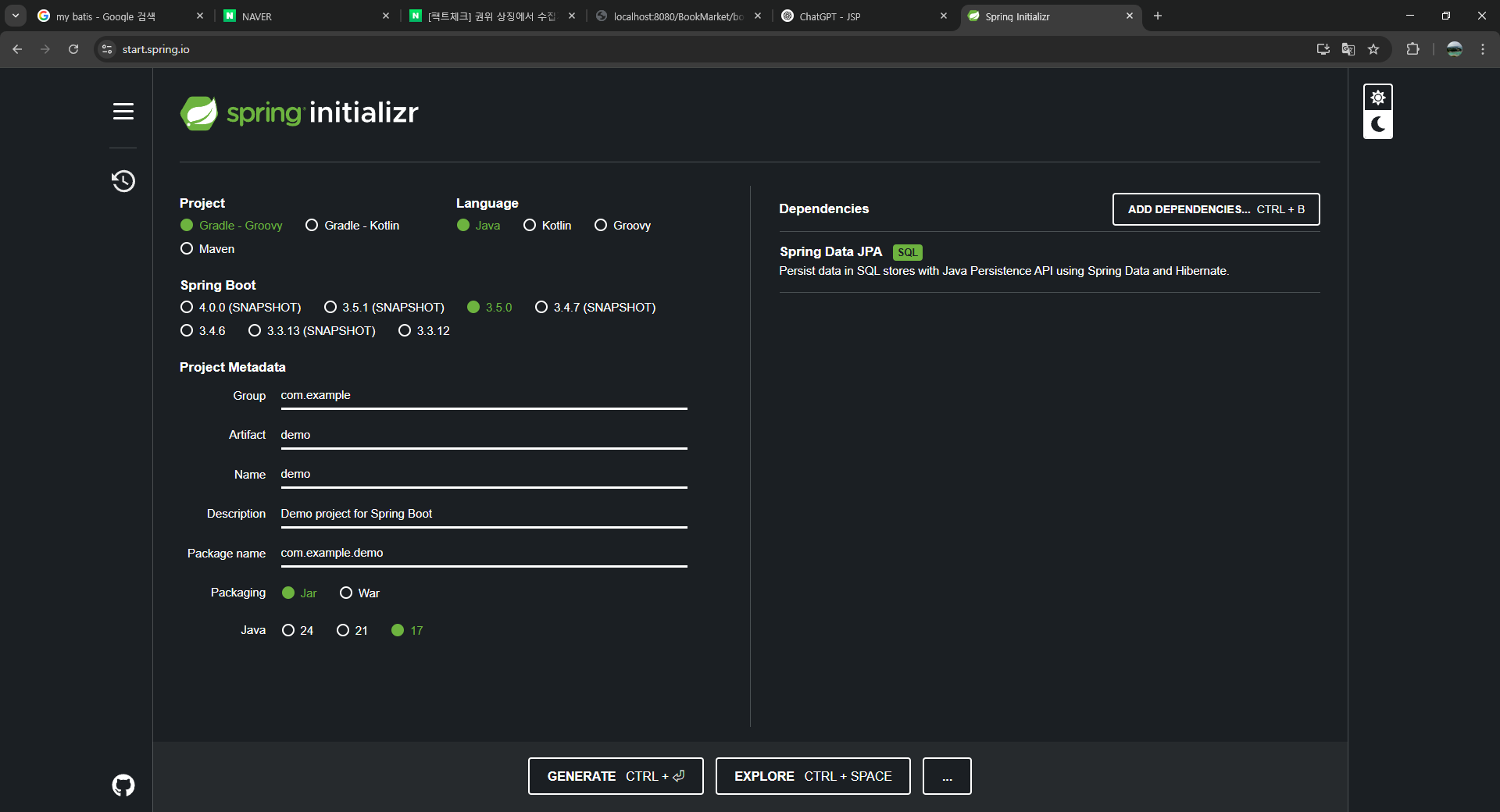


Books.html 안에 <footer> 제거



헤더에 <th:block th:replace="~{/module/menu}"></th:block>로 수정

푸터에 <th:block th:replace="~{/module/footer}"></th:block>로 수정



build.gradle에 JPA Dependencies 추가 (@Entity를 사용하기 위함)

domain폴더에 Role.enum파일(열거형)과 Member.java파일 생성

Member.java 내용 중

@Column(unique = true) // 중복허용 금지

@Enumerated(EnumType.STRING)

domain폴더에 MemberFormDto.java 생성하고 아래 내용 넣기

|  |
| --- |
| @NotBlank(message = "아이디는 필수 입력 값입니다.")  private String memberId;      @NotEmpty(message = "비밀번호는 필수 입력 값입니다.")  @Length(min=8, max=16, message = "비밀번호는 6자 이상, 16자 이하로 입력해주세요")  private String password;      @NotBlank(message = "이름은 필수 입력 값입니다.")  private String name;  @NotBlank(message = "연락처는 필수 입력 값입니다.")  private String phone;    @NotEmpty(message = "이메일은 필수 입력 값입니다.")  @Email(message = "이메일 형식으로 입력해주세요.")  private String email;    @NotEmpty(message = "주소는 필수 입력 값입니다.")  private String address;  } |

Member.java 안에 createMember메서드 작성

|  |
| --- |
| public static Member createMember(MemberFormDto memberFormDto, PasswordEncoder passwordEncoder){  Member member = new Member();  member.setMemberId(memberFormDto.getMemberId());  member.setName(memberFormDto.getName());  member.setPhone(memberFormDto.getPhone());  member.setEmail(memberFormDto.getEmail());  member.setAddress(memberFormDto.getAddress());  String password = passwordEncoder.encode(memberFormDto.getPassword());  member.setPassword(password);  member.setRole(Role.USER);  return member;  } |

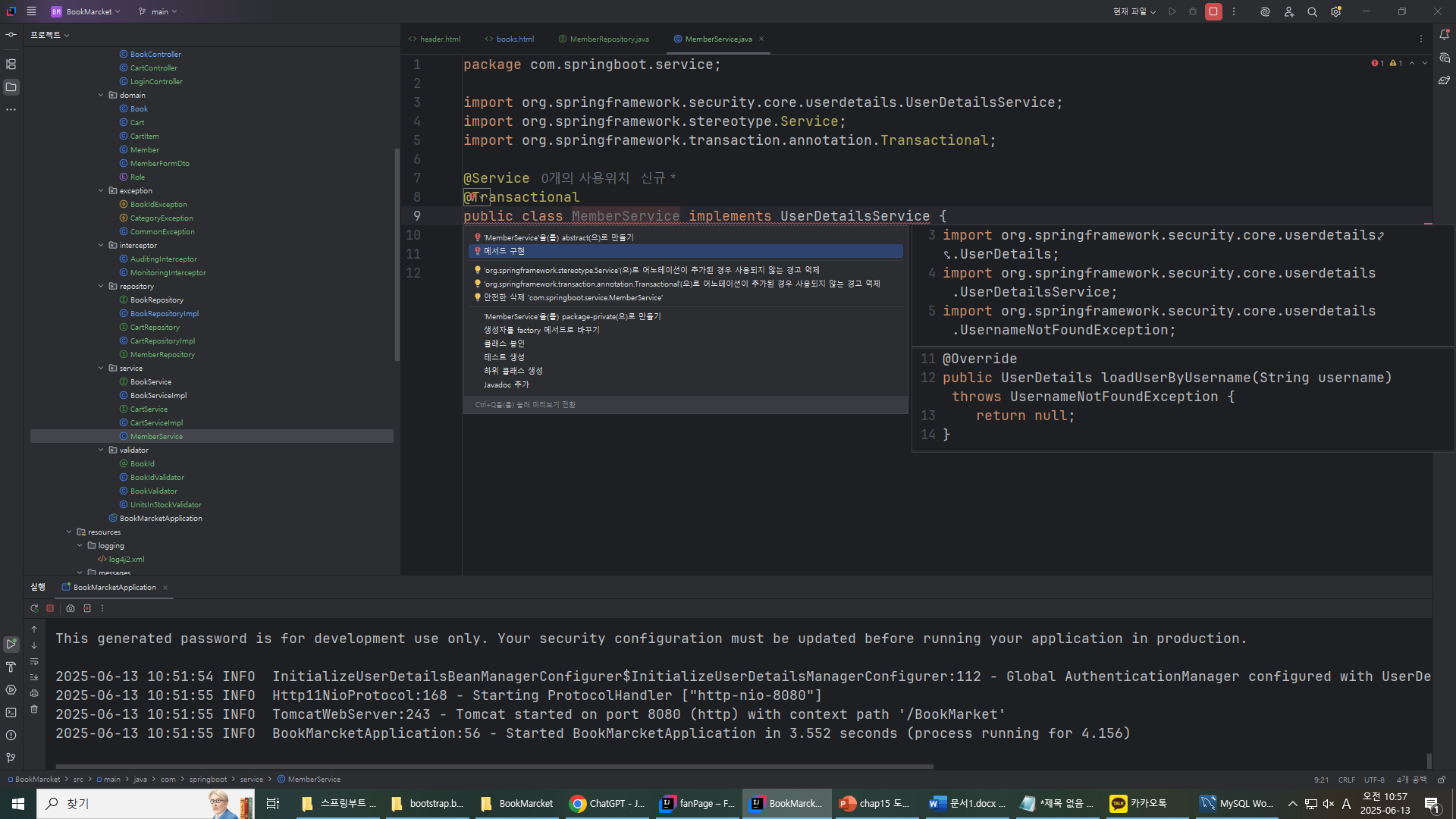
build.gradle에 MYSQL Dependencies 추가 (@Entity를 사용하기 위함)

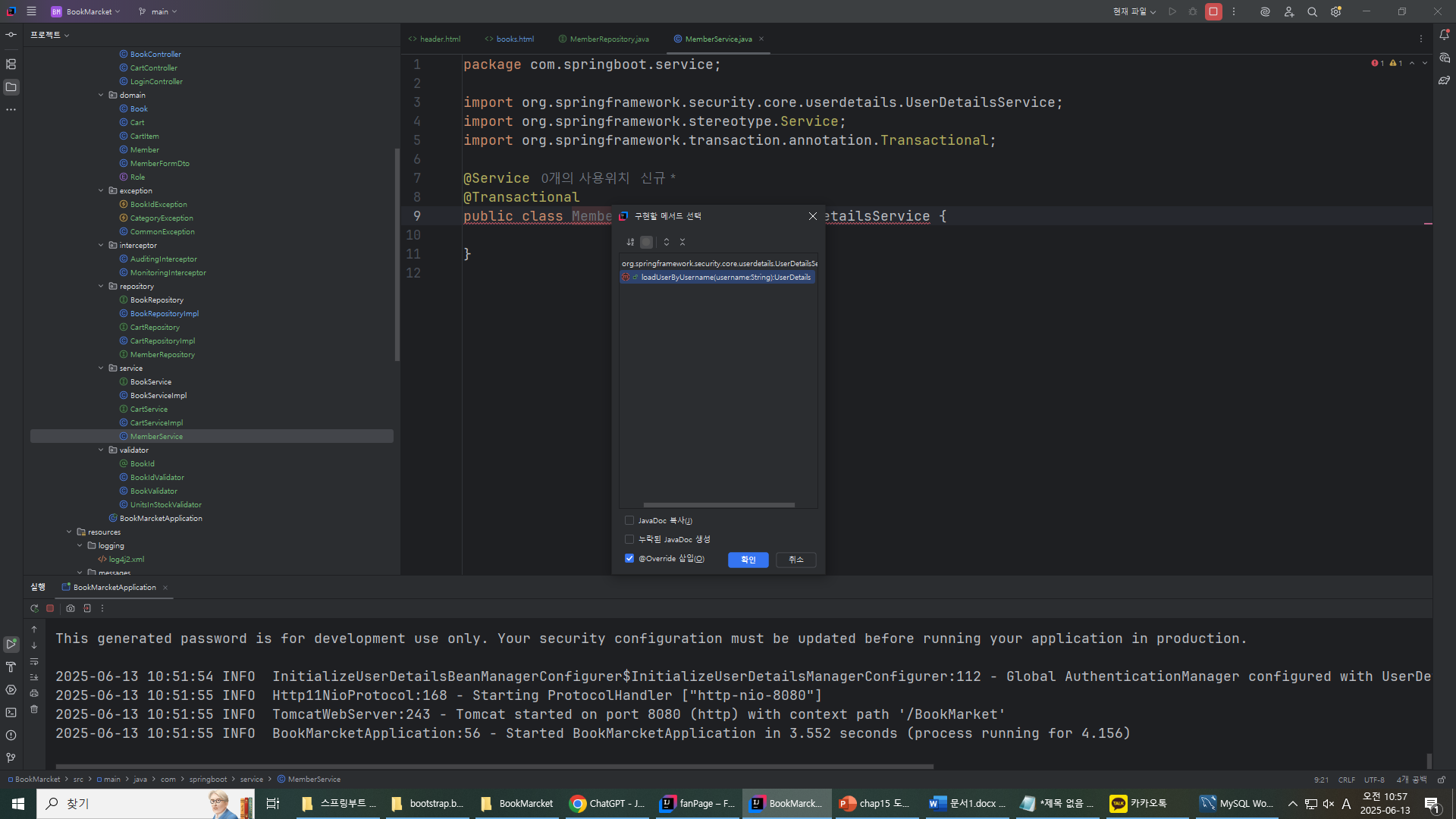
application.properties에 아래 내용 추가

|  |
| --- |
| spring.datasource.url=jdbc:mysql://localhost:3306/bookmarket\_DB  spring.datasource.username=root  spring.datasource.password=1234  spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  spring.jpa.database=mysql  spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect  spring.jpa.hibernate.ddl-auto=create  spring.jpa.generate-ddl=false  spring.jpa.show-sql=true  spring.jpa.properties.hibernate.format\_sql=true |

MemberRepository.java만들기

MemberService.java만들기





@Override  
public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {  
 return null;  
}

구현

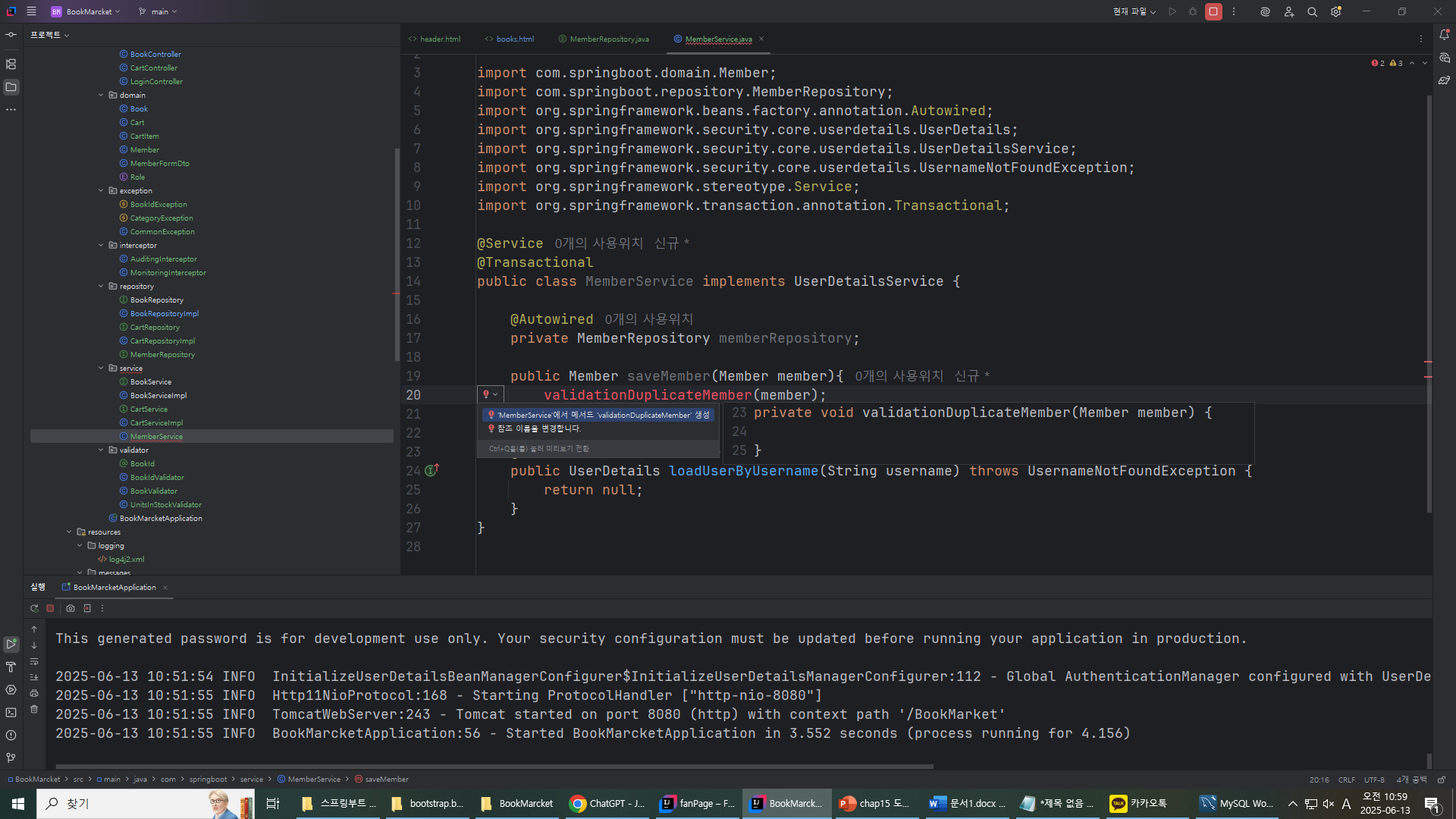
@Autowired  
private MemberRepository memberRepository;

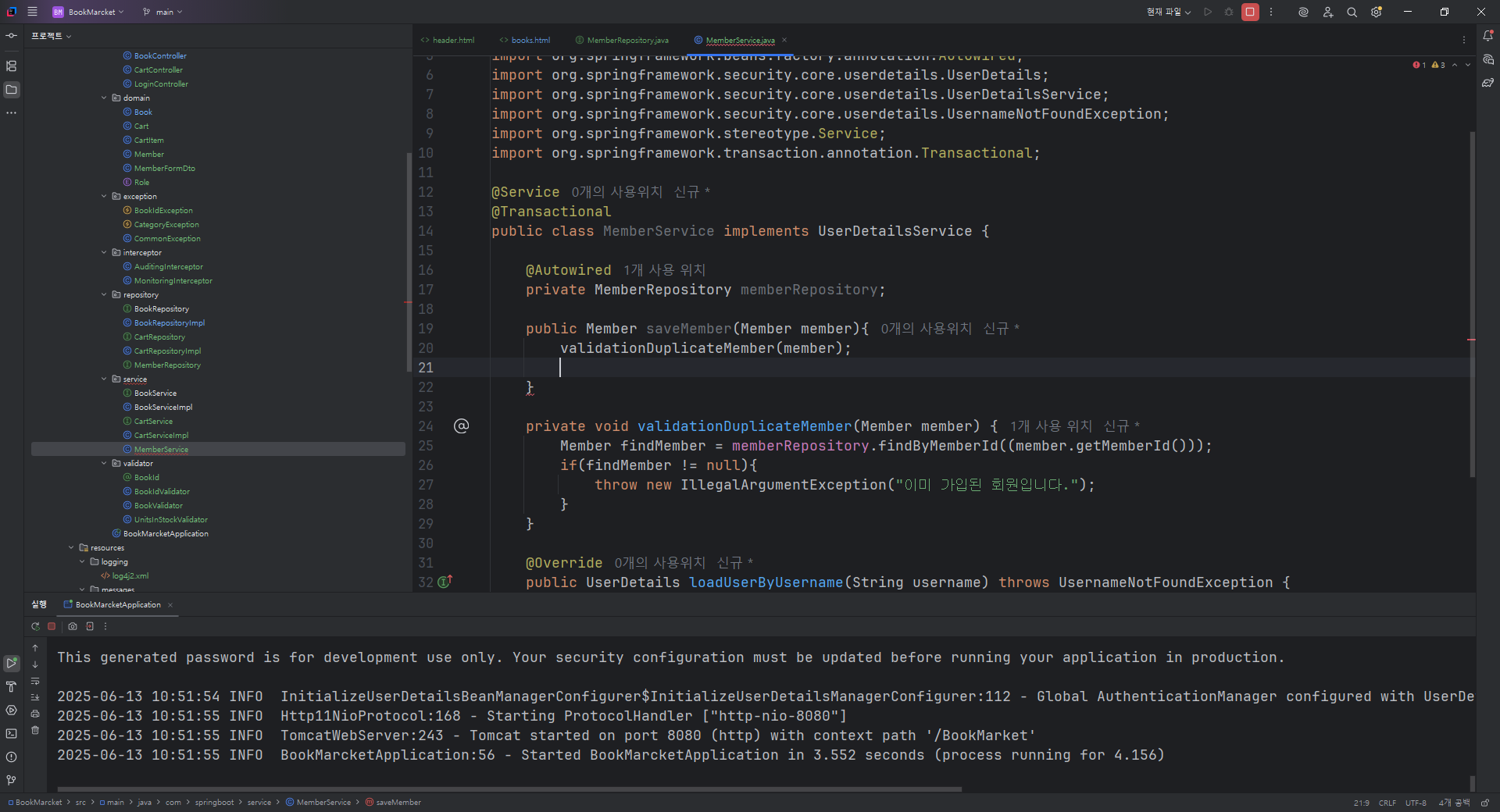
구현

public Member saveMember(Member member){  
 validationDuplicateMember(member);  
}

구현

아래 메서드 생성





return memberRepository.save(member);

memberService

memberController

memberRepository 만들기

member폴더 만들고

addMember.html만들기

name이 있어야 script를 읽을 수 있음

<form action="/BookMarket/members/add" method="post" th:object="${memberFormDto}" name="member">