팀 이름 : 개발자국

* 개요

반려동물 상품 구매 사이트 및 게시판 제작

* 팀원 및 역할

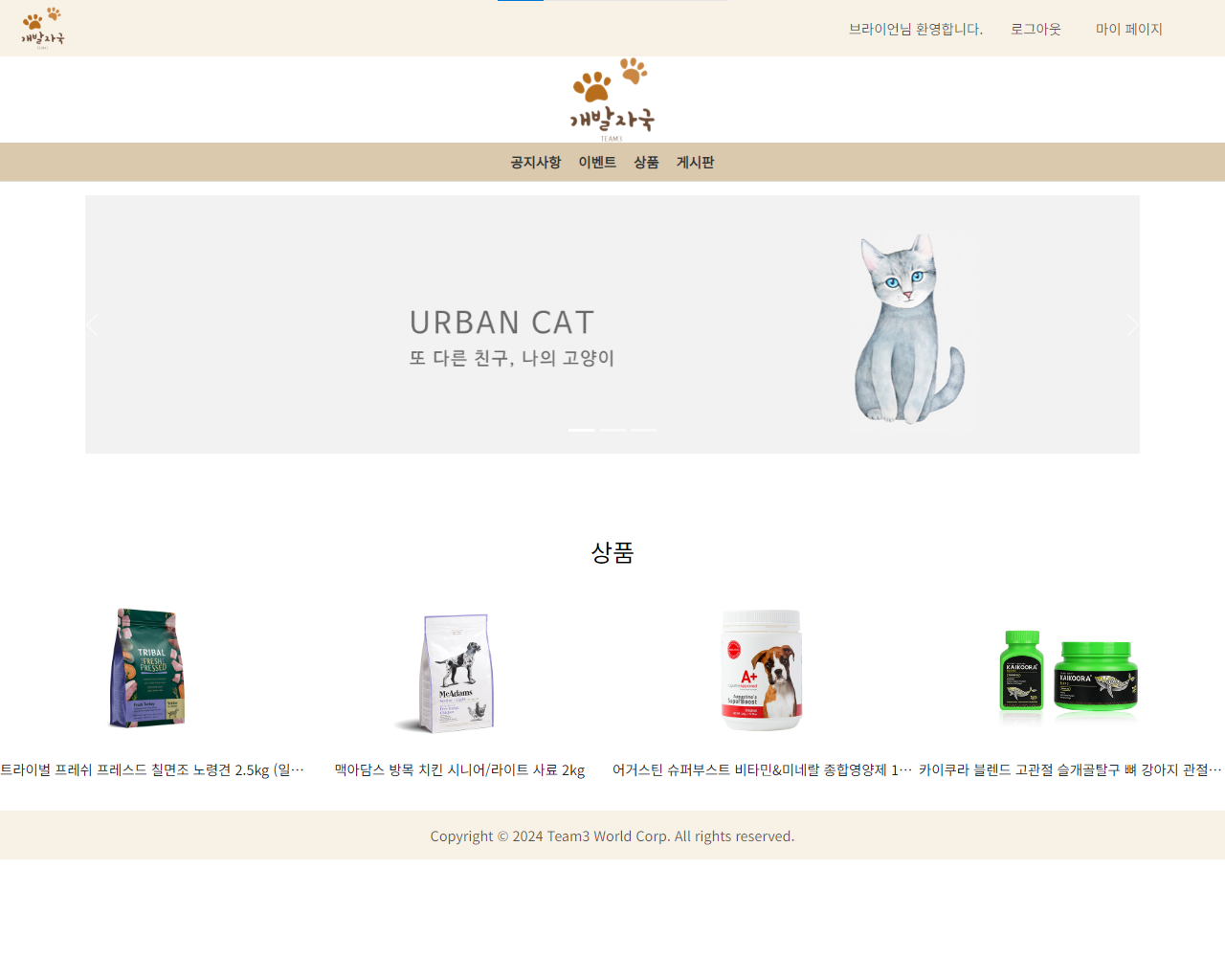
이병학(팀장) : 데이터베이스, DAO, 컨트롤러 제작

김동호 : 데이터베이스, jsp 파일 제작

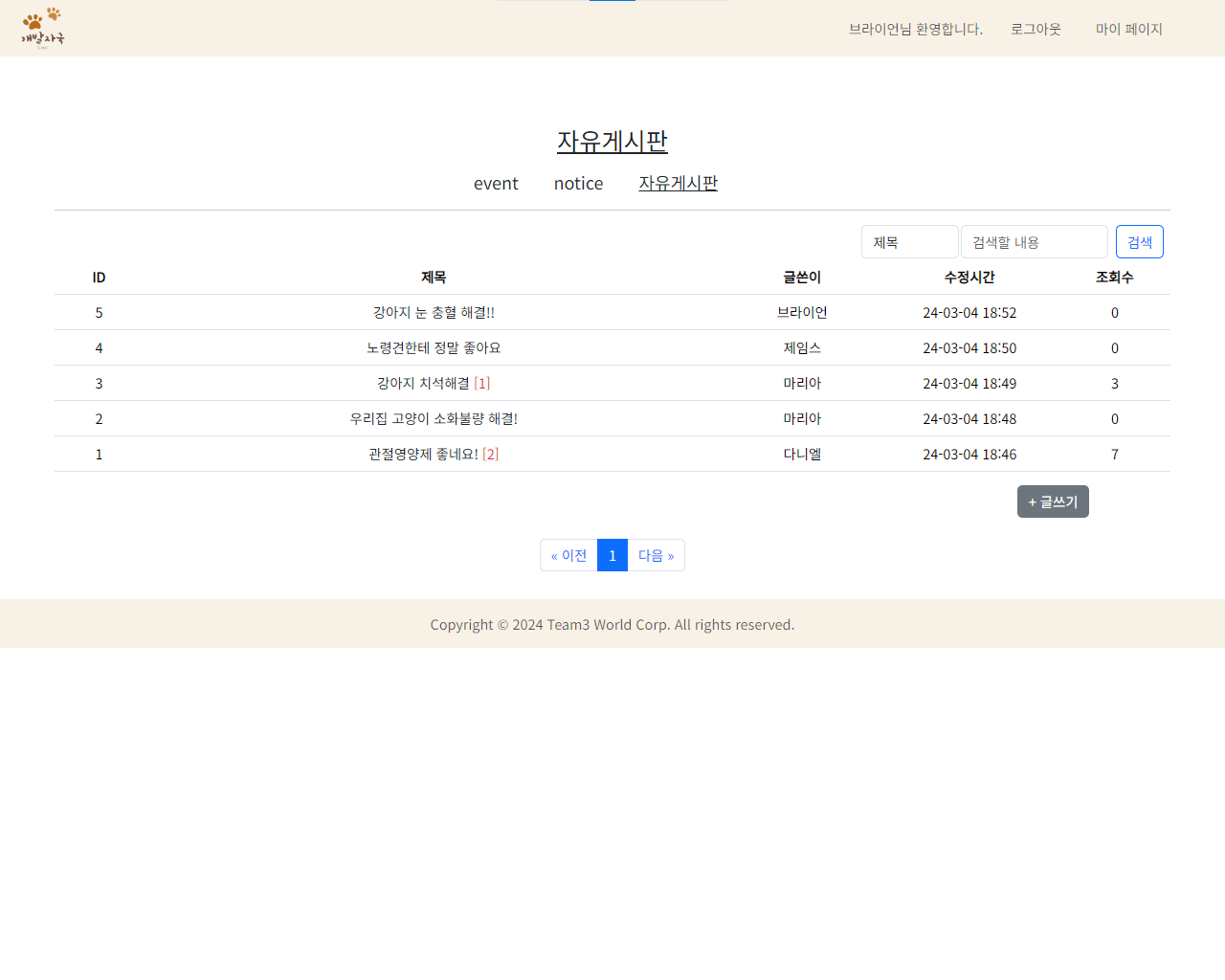
곽주영 : jsp 파일 및 웹 디자인 제작

윤영준 : 웹 디자인 제작 및 자료 조사

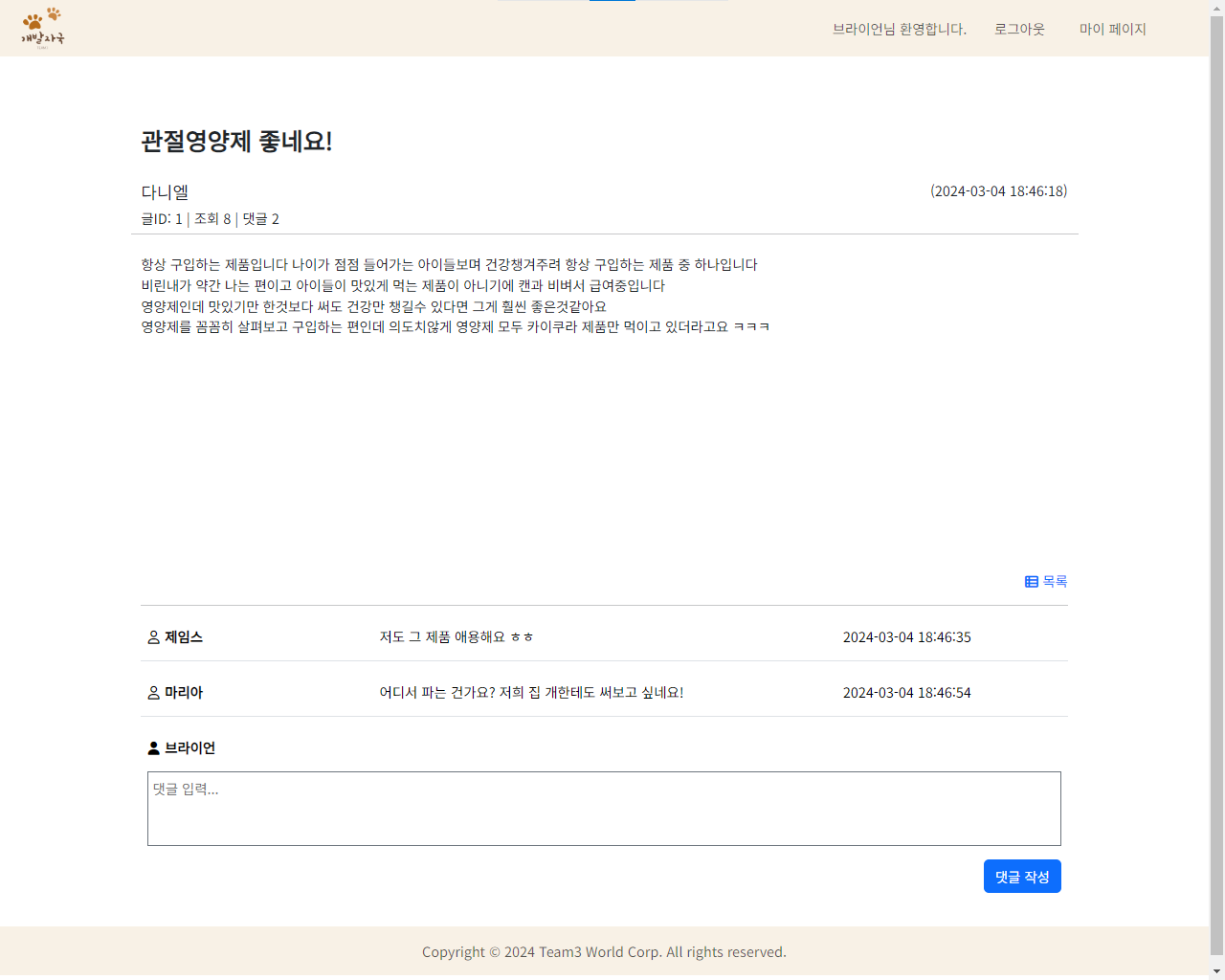
* 사이트 구성

메인페이지

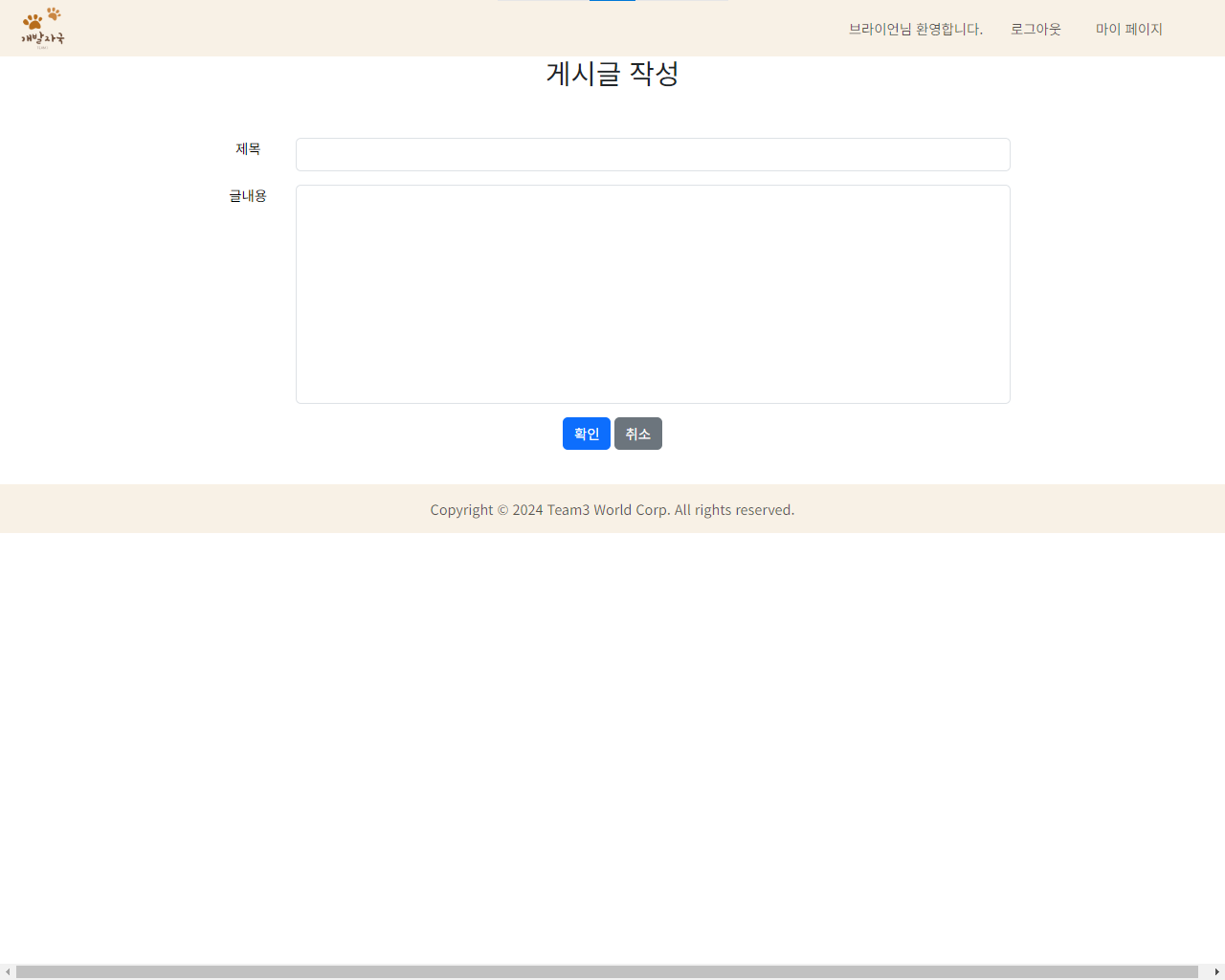
게시글 목록 페이지



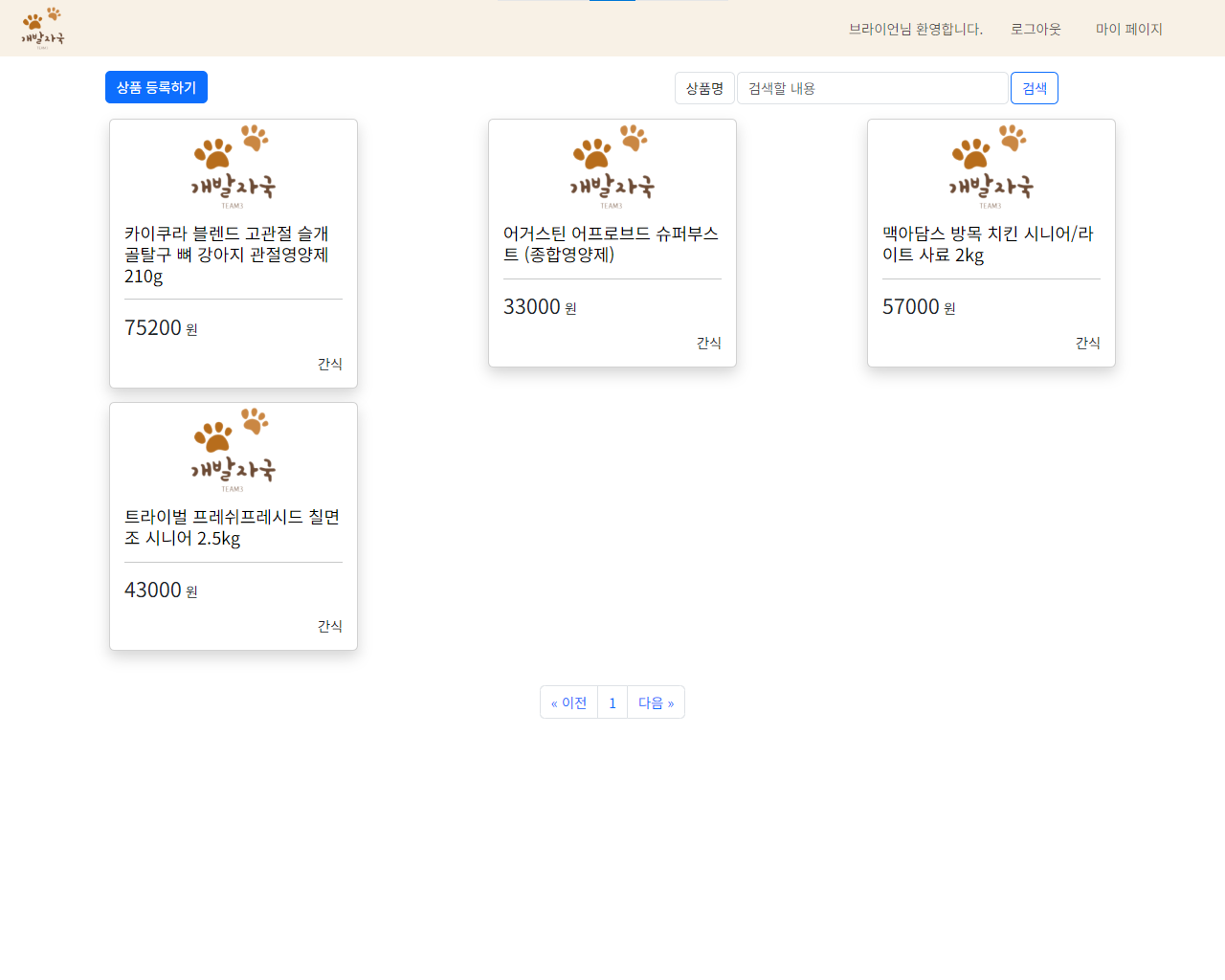
게시글 페이지 + 댓글



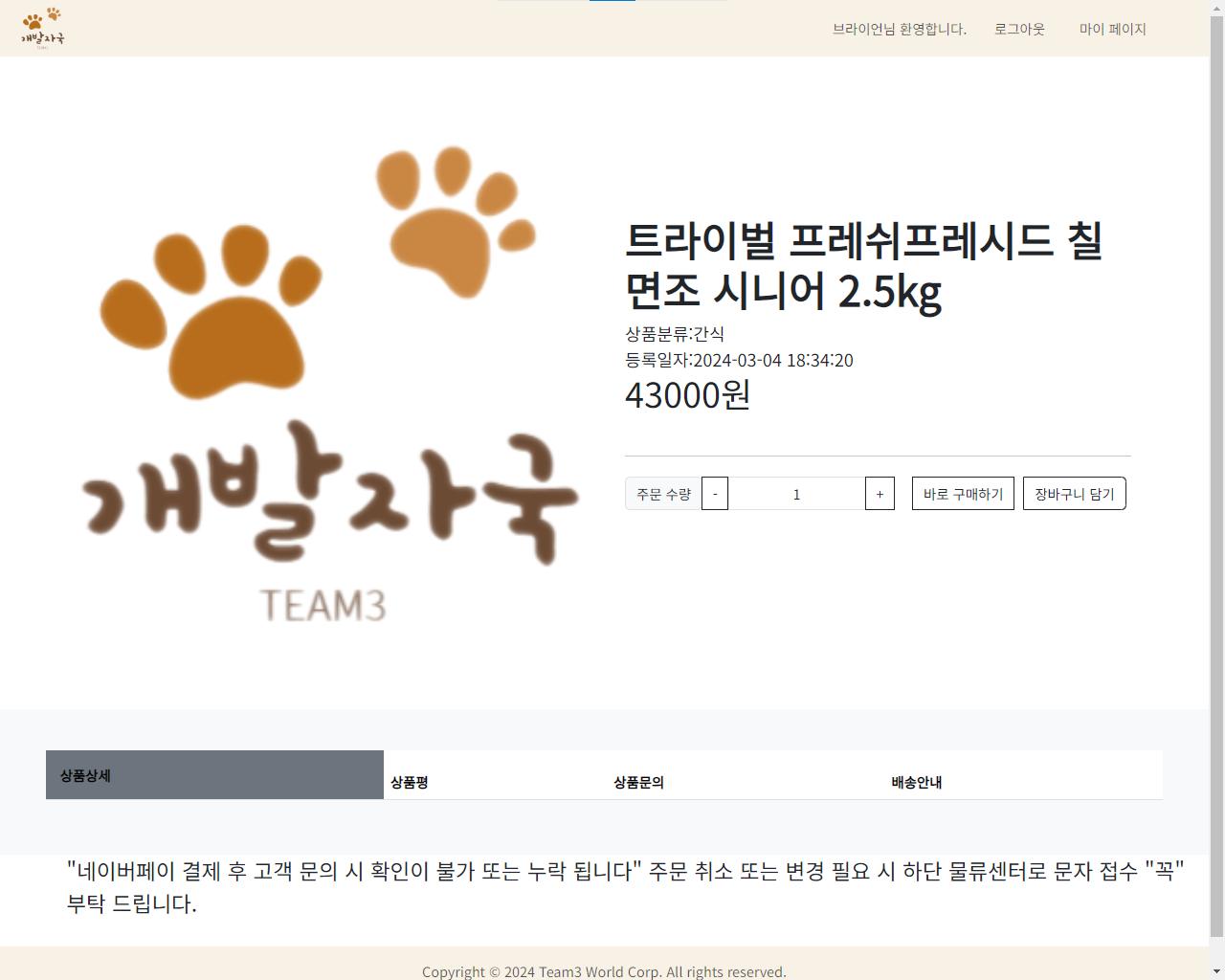
글 작성 페이지



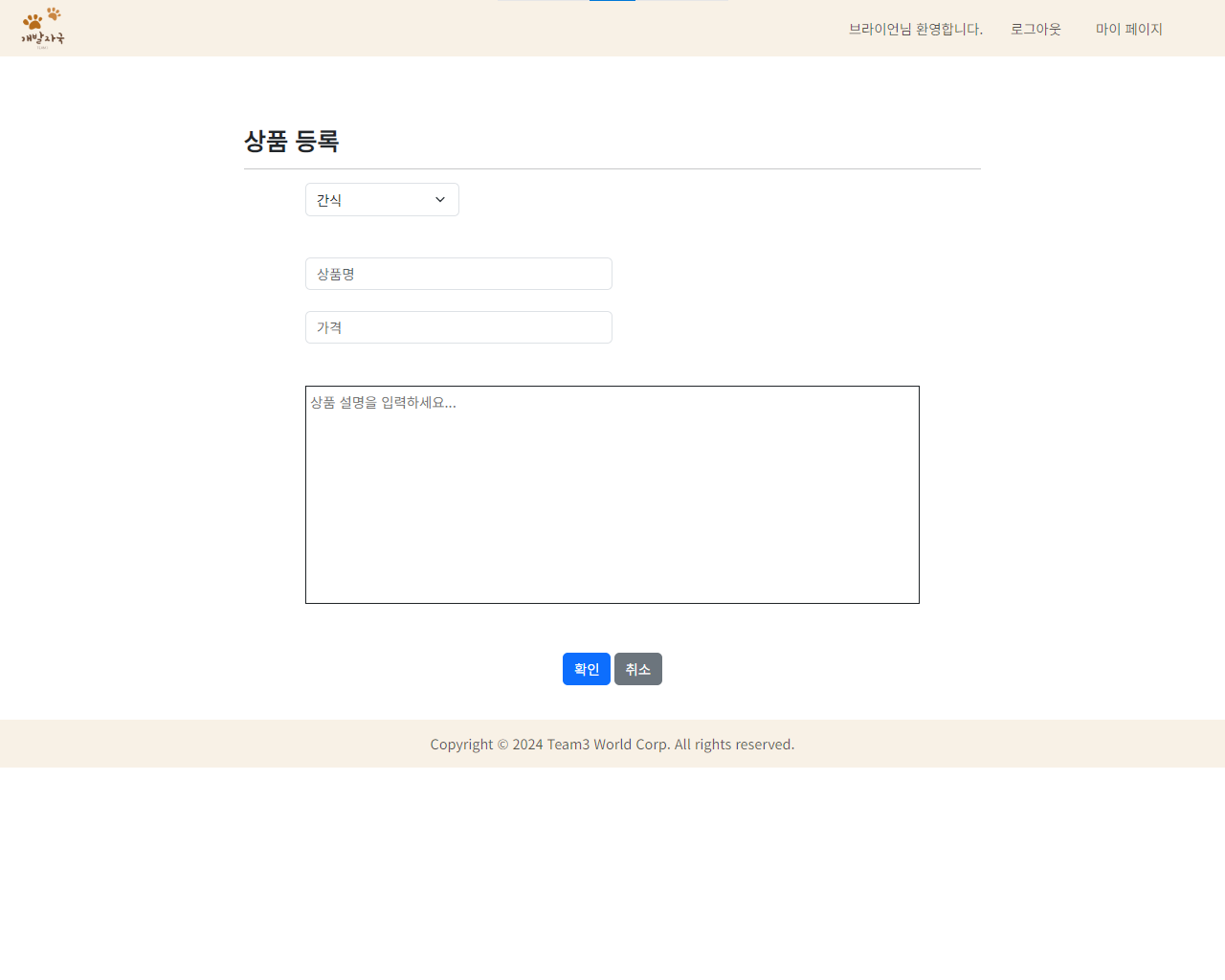
상품 목록 페이지



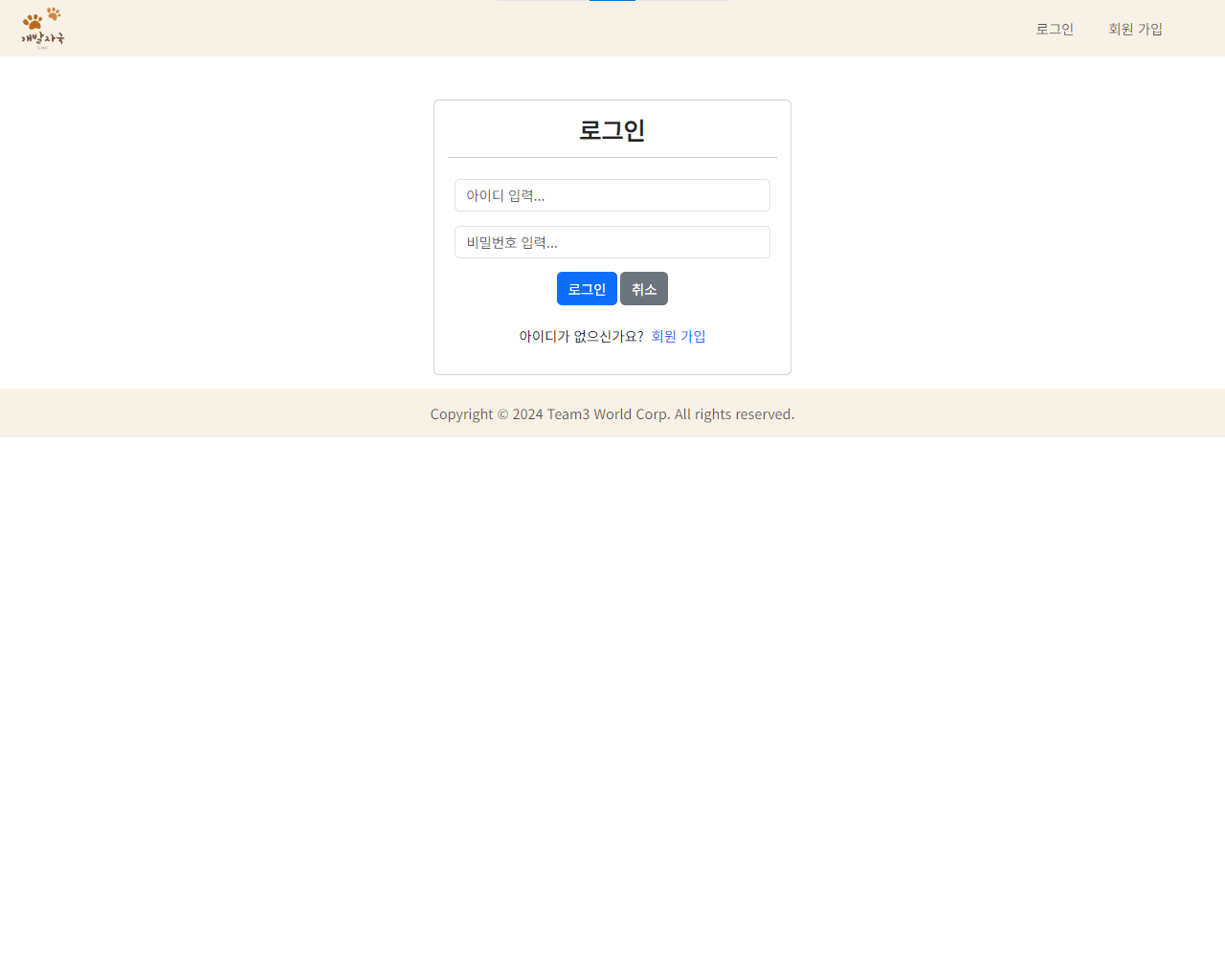
상품 페이지



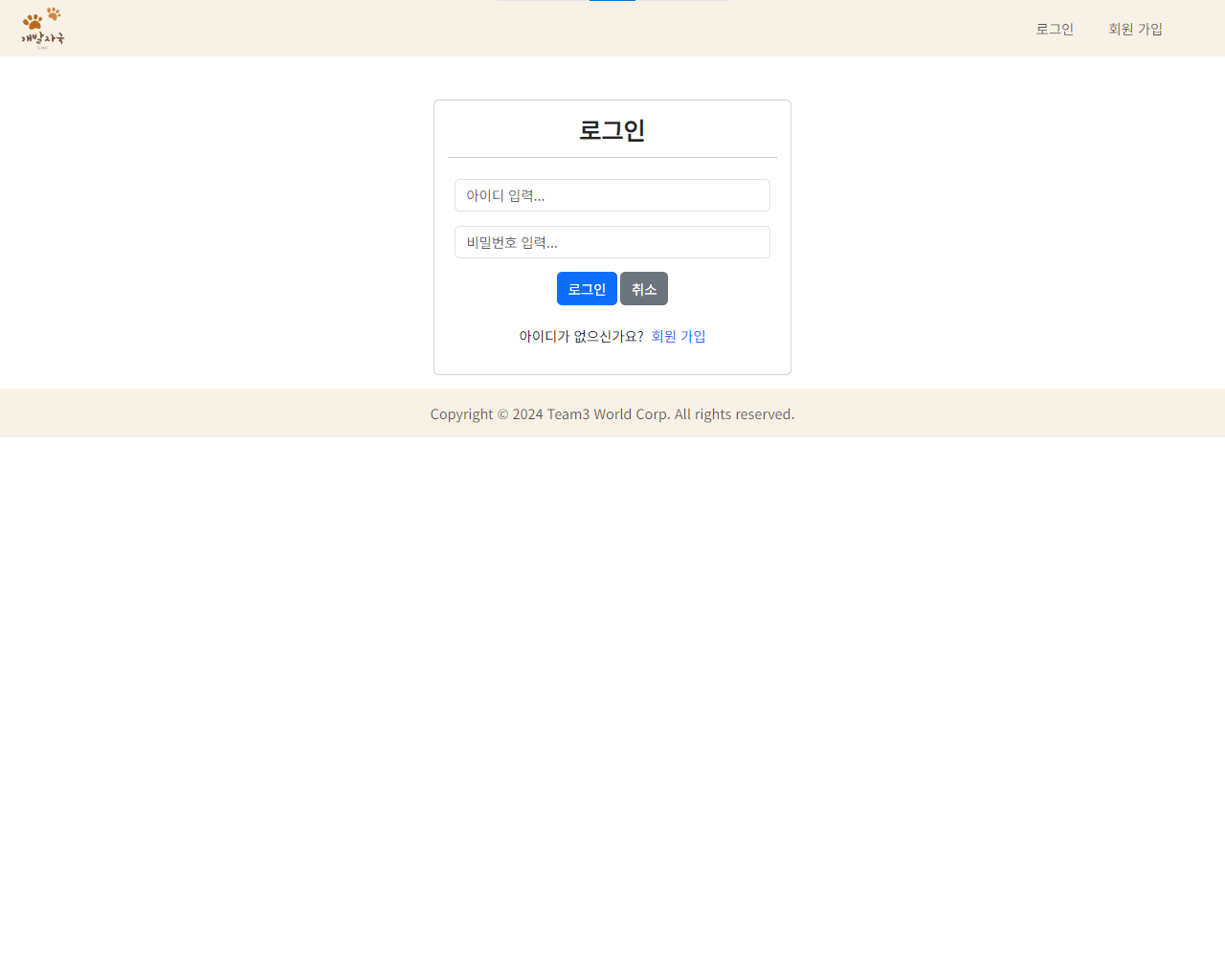
상품 등록 페이지



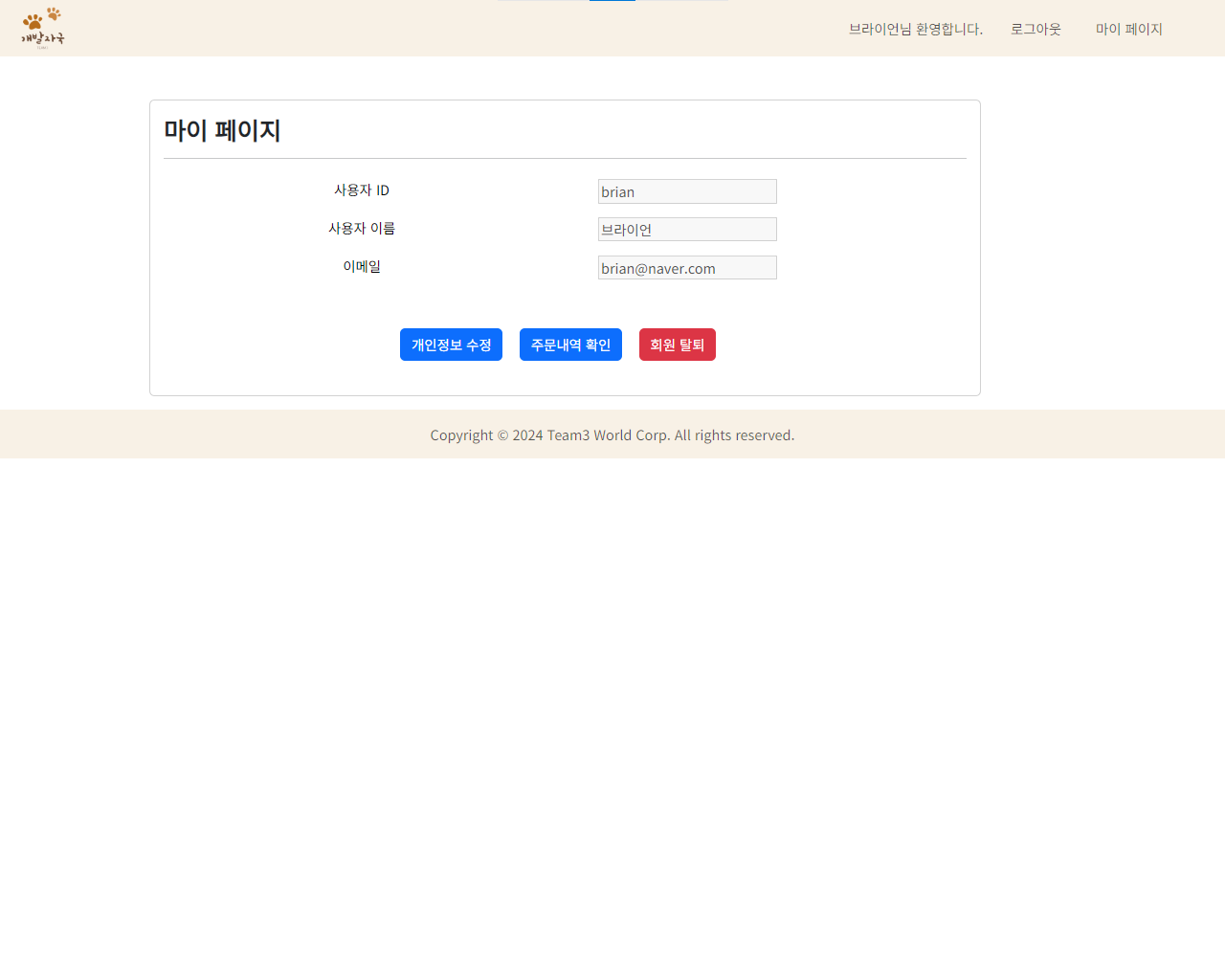
로그인 페이지



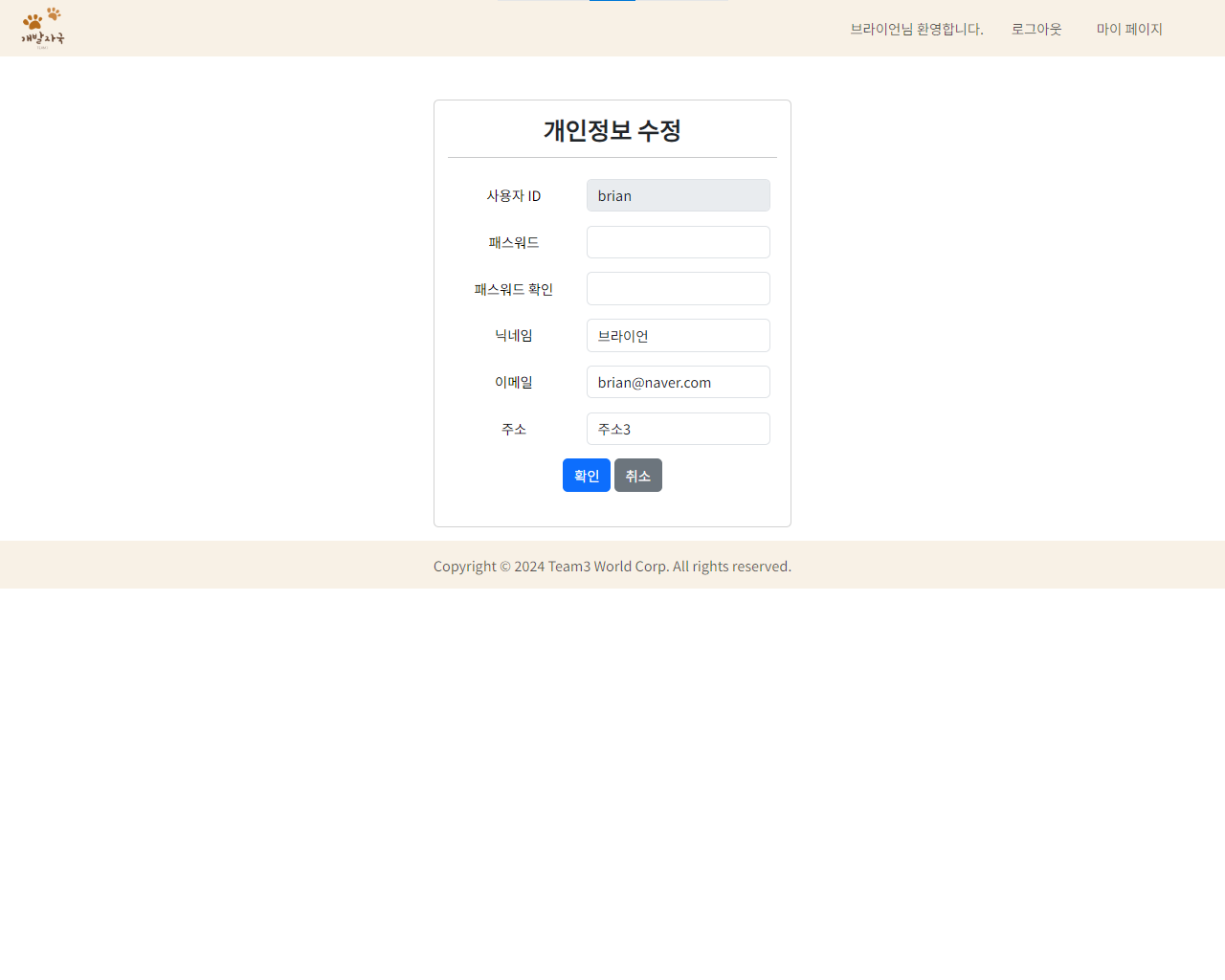
회원가입 페이지



마이페이지



회원정보 수정 페이지



* 라우팅 테이블

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **패스** | **메서드** | **입력 파라메터** | **행선지** | **넘겨줄 파라메터** |
| /main | GET |  | Main.jsp |  |
| 메인 페이지 | | | |
| /user/login | GET |  | login.jsp |  |
| 로그인, \_head.jspf를 통해 호출 | | | |
| POST | uid, pwd | /login or main |  |
| 로그인 페이지 입력 후 보내기 | | | |
| /user/logout | GET |  | /user/login |  |
| 로그아웃, \_head.jspf를 통해 호출 | | | |
| /user/register | GET |  | register.jsp |  |
| 사용자 등록 | | | |
| POST | uid, pwd, pwd2, email, address, name | /register or main | User 객체 |
| 사용자 등록 후 DB에 정보 보내기 | | | |
| /user/delete | GET | uid | /user/login |  |
| 사용자 삭제 | | | |
| /user/update | GET | uid | update/jsp | User 객체 |
| 사용자 정보 수정 | | | |
| POST | uid, pwd, pwd2, email, address, name | /update or main |  |
| 사용자 정보 수정 후 DB에 보내기 | | | |
| /user/order | GET | uid | user/order.jsp | uid, orderList |
| 주문 내역 보기 | | | |
| POST | behavior, oid | /user/order |  |
| 주문 내역 확인 혹은 취소 | | | |
| /user/orderDetail | GET | oid | user/orderDetail.jsp | order, salesList |
| 주문 내역 내 구매내역 리스트 확인 | | | |
| POST | oid, sid | user/orderDetail |  |
| 주문 내역 내 구매내역 취소 | | | |
| /board/list | GET | page, field, query | board/list.jsp | BoardList, pageList, page, field, query |
| 게시글 리스트 | | | |
| /board/detail | GET | bid | board/detail.jsp | Board 객체 |
| 게시글 화면 | | | |
| POST | content, bid, behavior | board/detail.jsp | bid, Reply 객체 |
| 게시글 안에서 댓글 작성 및 삭제 | | | |
| /board/insert | GET | sessUid | board/insert.jsp |  |
| 게시글 작성 페이지 | | | |
| POST | title, content | /board/insert or /board/detail | Stitle, Scontent |
| 게시글 작성 후 DB에 보내기 | | | |
| /board/delete | GET | bid | /board/list |  |
| 게시글 삭제 | | | |
| /board/update | GET |  | /board/update.jsp | Board 객체 |
| 게시글 수정 | | | |
| POST | bid, uid, title, content | /board/update or /board/detail | bid |
| 게시글 수정 후 DB에 보내기 | | | |
| /item/detail | GET | mid | item/detail.jsp | Item 객체 |
| 상품 페이지 | | | |
| POST | uid, mid, quantity, | item/detail | mid |
| 상품 구매 | | | |
| /item/list | GET | page, field, query | item/list.jsp | itemList, pageList, page, field, query |
| 상품 목록 | | | |
| /item/insert | GET |  | item/insert.jsp |  |
| 상품 추가 페이지 | | | |
| POST | uid, name, price | /item/insert or /item/list |  |
| 상품 추가 후 DB에 보내기 | | | |

* 사이트 작동 구조

|  |  |
| --- | --- |
| 객체 (변수는 모두 private) | |
| User | uid (String) 사용자 id - primary key  pwd (String) 비밀번호  uname (String) 사용자 이름  email (String) 이메일  regDate (LocalDate) 등록일  isDeleted (int) 삭제여부  address (String) 주소  ——————————————————————————————————————  사용자 정보를 담는 객체 |
| Board | bid (int) 게시글 id - primary key  uid (String) 작성자 id  title (String) 제목  content (String) 글 내용  writeTime (LocalDateTime) 작성 시간  viewCount (int) 조회수  replyCount (int) 댓글 수  uname (String) 작성자 이름 - DB 미포함 변수  ——————————————————————————————————————  게시글 정보를 담는 객체 |
| Item | mid (int) 상품 id - primary key  name (String) 상품 이름  price (int) 가격  itemTime (LocalDateTime) 등록일자  category (String) 분류  image (String) 사진파일  description (String) 상품 설명  ——————————————————————————————————————  상품 정보를 담는 객체 |
| Reply | rid (int) 댓글 번호 - primary key  bid (int) 게시글 번호  uid (String) 작성자 uid  content (String) 댓글 내용  replyTime (LocalDateTime) 작성 시간  uname (String) 작성자 이름 - DB 미포함 변수  ——————————————————————————————————————  댓글 정보를 담는 객체 |
| Item\_order | oid (int) 주문번호 - primary key  uid (String) 주문자 id  totalPrice (int) 주문 내 내역의 총 가격  orderTime (LocalDateTime) 주문시간  status (int) 주문 상태  ——————————————————————————————————————  주문 정보를 담는 객체  Order가 이미 sql에서의 축약어라 객체명은 Order가 아닌 Item\_order로 대체 |
| Sales | sid (int) 구매번호- primary key  mid (int) 상품번호  oid (int) 주문번호  quantity (int) 구매수량  price (int) 가격 - DB 미포함 변수  name (String) 상품이름 - DB 미포함 변수  ——————————————————————————————————————  구매 정보를 담는 객체 |
| DAO | |
| UserDao | Users getUsersByUid(String uid)  List<Users> getUsersList(**int** num, **int** offset)  **void** insertUsers(Users Users)  **void** updateUsers(Users Users)  **void** deleteUsers(String uid)  **int** getUsersCount() |
| BoardDao | Board getBoard(**int** bid)  List<Board> getBoardList(String field, String query, **int** num, **int** offset)  **void** insertBoard(Board board)  **void** setReplyCount(Board board) - 댓글 수가 바뀌면 이를 바로 반영하기 위한 메소드  **void** updateBoard(Board board) - 위와 달리 글 제목과 내용을 바꾸는 메소드  **void** deleteBoard(**int** bid)  **void** increaseCount(String field, **int** bid)  **int** getBoardCount(String field, String query) |
| ItemDao | Item getItem(**int** mid)  List<Item> getItemList(String field, String query, **int** limit, **int** offset)  **void** insertItem(Item item)  **void** updateItem(Item item)  **void** deleteItem(**int** mid)  **int** getItemCount(String field, String query) |
| ReplyDao | Reply getReply(**int** rid)  List<Reply> getReplyList(**int** bid)  **void** insertReply(Reply reply)  **void** updateReply(Reply reply)  **void** deleteReply(**int** rid)  **int** getReplyCount(**int** bid) |
| OrderDao | Item\_order getOrder(String uid, **int** status) - 유저명과 상태로 주문정보 찾기. 새 주문을 만들기 전에 특정 상태의 기존 주문이 있는지 확인하기 위해 사용  Item\_order getOrderOid(**int** oid)  List<Item\_order> getOrderList(String uid)  **void** insertOrder(Item\_order order)  **void** updateOrder(Item\_order order)  **void** deleteOrder(**int** oid) |
| SalesDao | Sales getSales(**int** sid)  Sales getSalesOid(**int** oid, **int** mid) - 주문번호와 상품번호로 구매정보 찾기. 새 구매정보를 만들기 전에 기존에 동일한 상품을 동일한 주문번호에서 구매하는 정보가 있는지 확인하기 위해 사용  List<Sales> getSalesList(**int** oid)  **void** insertSales(Sales sales)  **void** updateSales(Sales sales)  **void** deleteSales(**int** sid) |
| 서비스/인터페이스 | |
| UserService  impl | 호출되는 DAO - UserDao  ——————————————————————————————————————  **void** registerUser(Users users)  - 유저 등록, 이 과정에서 입력된 비밀번호는 암호화  Users getUserByUid(String uid)  - uid를 통해 유저 찾기  **int** getUserCount()  - 전체 유저 수 찾기  **void** updateUser(Users users)  - 유저 정보 업데이트  **void** deleteUser(String uid)  - 유저 삭제  **int** login(String uid, String pwd)  - hash 값으로 등록된 비밀번호를 BCrypt를 이용하여 입력된 비밀번호와 비교하여 로그인 결과 리턴  (0 = 로그인 성공, 1 = 비밀번호 오류, 2 = id 오류) |
| BoardService  impl | 호출되는 DAO - BoardDao, ReplyDao  ——————————————————————————————————————  List<Board> getBoardList(**int** page, String field, String query)  - 게시판 리스트 + 각 게시글마다 댓글 수 찾아서 댓글 수 올바르게 초기화  Board getBoard(**int** bid)  - 번호를 통해 게시글 DB에서 찾기  **int** getBoardCount(String field, String query)  - 특정 조건을 만족하는 게시글의 개수  **void** insertBoard(Board board)  - 게시글 DB에 추가  **void** updateBoard(Board board)  - 게시글 수정  **void** deleteBoard(**int** bid)  - 게시글 삭제  **void** increaseViewCount(**int** bid)  - 게시글 조회수 증가 (고정 +1)  **void** increaseReplyCount(**int** bid)  - 게시글 댓글수 증가 (고정 +1) |
| ItemService  impl | 호출되는 DAO - ItemDao  ——————————————————————————————————————  Item getItem(**int** mid)  - 상품 객체 받기  List<Item> getItemList(String field, String query, **int** page)  - 상품 리스트 리턴  **int** getItemCount(String category, String query)  - 페이지를 만들기 위해 특정 조건의 상품 수 계산  **void** insertItem(Item item)  - 상품 등록  **void** updateItem(Item item)  - 상품 정보 수정  **void** deleteItem(**int** mid)  - 상품 삭제 |
| ReplyService  impl | 호출되는 DAO - BoardDao, ReplyDao  ——————————————————————————————————————  Reply getReply(**int** rid)  - 댓글 객체 받기  List<Reply> getReplyList(**int** bid)  - 댓글 리스트 리턴 + 리스트 내의 댓글 갯수만큼 게시글의 댓글 수 올바르게 수정  **void** insertReply(Reply reply)  - 댓글 등록 + 게시글의 댓글 수(ReplyCount) 증가  **void** updateReply(Reply reply)  - 댓글 정보 수정  **void** deleteReply(**int** rid)  - 댓글 삭제 + 게시글의 댓글 수(ReplyCount) 감소 |
| OrderService  impl | 호출되는 DAO - OrderDao  ——————————————————————————————————————  Item\_order getOrder(String uid, **int** status)  - 사용자 id와 주문 상태로 주문 찾기  0 = 대기(초기 상태)  1 = 요청  2 = 운송  3 = 완료  -1 = 취소  Item\_order getOrderOid(**int** oid)  - 주문번호로 주문 찾기  List<Item\_order> getOrderList(String uid)  - 특정 id를 가진 사용자의 주문정보 리스트 리턴  **void** insertOrder(Item\_order order)  - 주문 등록  **void** updateOrder(Item\_order order)  - 주문정보 수정  **void** deleteOrder(**int** oid)  - 주문 삭제  **void** cancelOrder(Item\_order order)  - 주문 취소 |
| SalesService  impl | 호출되는 DAO - SalesDao, OrderDao, ItemDao  ——————————————————————————————————————  Sales getSales(**int** sid)  - 구매정보 찾기  Sales getSalesOid(**int** oid, **int** mid)  - 주문번호와 상품번호로 구매정보 찾기  List<Sales> getSalesList(**int** oid)  - 특정 주문 번호의 구매정보 리스트 리턴  **void** insertSales(Sales sales)  - 구매정보 등록 + 등록된 주문정보의 총 가격(totalPrice) 증가  **void** updateSales(Sales sales)  - 구매정보 수정 + 등록된 주문정보의 총 가격 변경  **void** deleteSales(Sales sales)  - 구매정보 삭제 + 등록된 주문정보의 총 가격 감소 |

* 개선할 점

1. 상품 이미지 추가 방식을 DB와 연동해서 하는 방식 필요
2. 회원 정보를 세분화하여 ‘기업’ 속성을 갖는 회원만 상품 추가 가능하도록 기능 수정
3. 이벤트와 공지사항 페이지 추가

* 참고한 사이트

1. <https://pecanori.co.kr/main/index.php>
2. <https://puppydog.co.kr>
3. <https://www.dogpang.com>
4. <https://petbtob.co.kr/>

* 제작 과정 및 회의록

|  |
| --- |
| 사용한 Notion 링크  https://www.notion.so/Human-3-899250bf094c46b282719409e98460c2  구글드라이브  https://drive.google.com/drive/folders/1R80l7\_kK9HNuWZMxo5IymI1x5y5Kj5Hb  GitHub 링크 |
| 2024-02-27  **프로젝트 시작 및 기초적인 설계 구상**  ——————————————————————————————————————  페이지 주제 정하기 – 상품 + 게시판으로 결졍  페이지 구성요소 정하기 – User, Board, Item, Reply  페이지 플로우 차트 만들기  페이지 디자인 구상 정하기 |
| 2024-02-28  **제작에 필요한 기초 요소 작성**  ——————————————————————————————————————  DAO 등의 파일 작성규칙 정하기  DB 만들기 – User, Board, Item, Reply  DB 초기값 입력  인터페이스, 서비스, 객체 생성 |
| 2024-02-29  **페이지 제작 1**  ——————————————————————————————————————  서블릿 제작 – main, board, user  jsp, jspf 제작 – user/login, user/register, user/mypage, board/list, \_top, \_bottom |
| 2024-03-01  **페이지 제작 2 및 데이터 수정**  ——————————————————————————————————————  서블릿 제작 - item  jsp, jspf 제작 – user/update, board/insert, board/detail, item/list, item/insert, item/detail  메인 페이지 디자인 제작 |
| 2024-03-04  **페이지 제작 마무리**  ——————————————————————————————————————  모든 페이지 디자인 제작  주문내역 추가 (order, orderDao 등)  보고서 및 ppt 작성 |