



여행 계획을 위한 웹 페이지

# TRIP-관광지 웹 사이트

웹페이지 구현 프레젠테이션

양경민

E-mail:mi568977@naver.com



## 1. 프로젝트 개요

- 기획 의도 및 프로젝트 설명
- 개발 인원 및 개발 환경

## 2. 경쟁사 분석

- 경쟁사 분석
- 벤치마킹

## 3. 웹 사이트 및 DB 설계

- 사이트 설계
- DB 설계

## 4. 기능 구현

- 기능 구현 및 시연

## 5. 기대 효과

- 기대 효과



# 1. 프로젝트 개요

기획 의도 및 프로젝트 설명



## TRIP 기획 의도



- 코로나 19 대응 확산
- 백신 접종의 확산
- 코로나 19 감염자 감소



- 여행에 대한 수요 증가

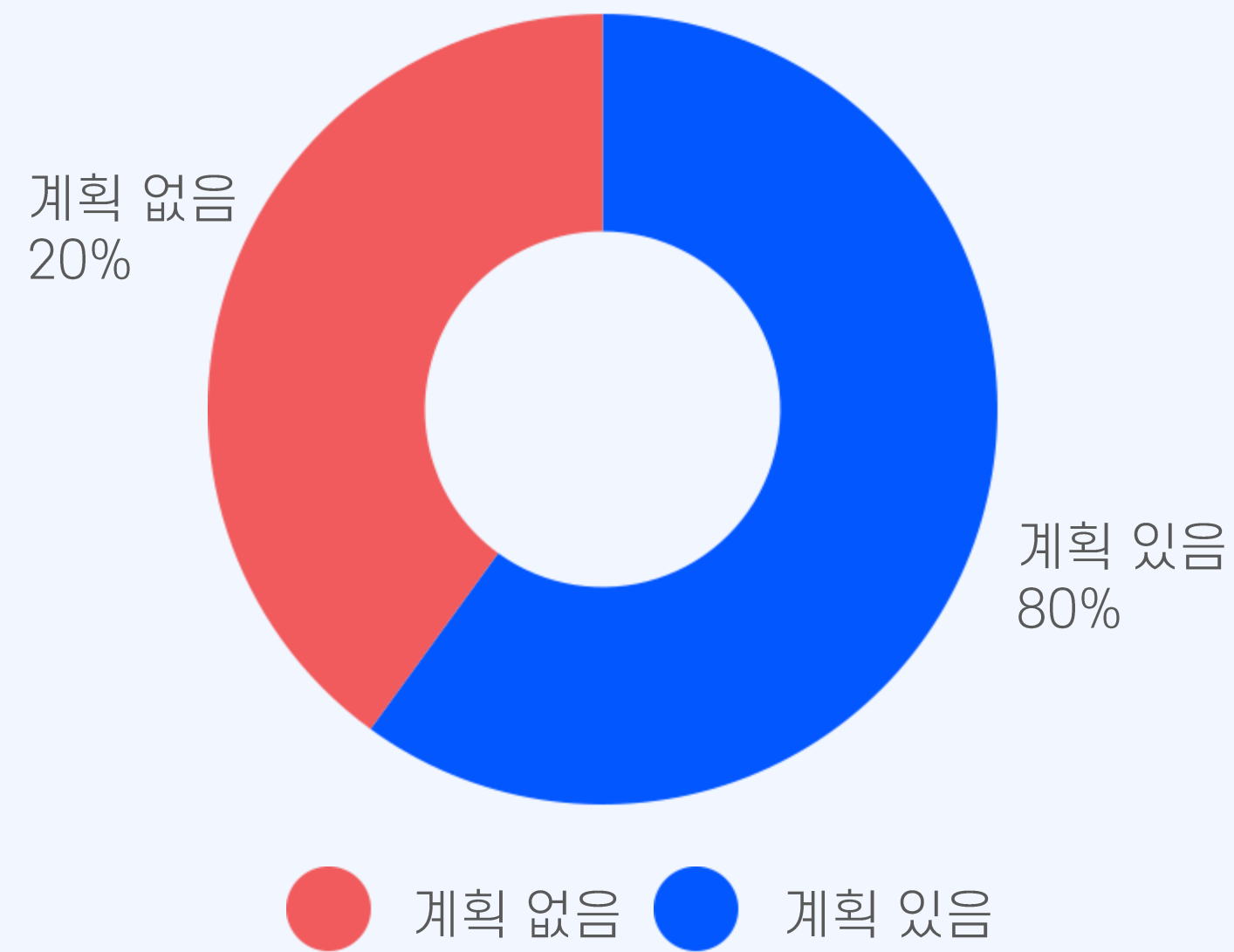


- 여행지 위치 및 정보 제공
- 여행 루트 설계 가능

여행 계획을 위한 웹 페이지

## 여행 계획

서울, 경기 및 5대 광역시에 거주하는 성인남녀 1,000명을 대상





# 1. 프로젝트 개요

개발 환경 및 언어

ORACLE  
GoldenGate



spring



Logstash



elastic

kibana

elasticsearch

여행 계획을 위한 웹 페이지

- 개발 인원: 6명
- 개발 기간:
  - 1차) 2023.04.25 ~ 2023.05.12
  - 2차) 2023.07.26 ~ 2023.08.14
- 맡은 역할 :
  - (1)회원 가입 및 로그인 기능
  - (2)회원 데이터 베이스 구성 및 관리
  - (3)회원 즐겨찾기 기능 구성
  - (4)회원 즐겨찾기 데이터 베이스 구성 및 관리
  - (5)검색 기능 구현
  - (6)검색어 자동완성 기능



## 2. 경쟁사 분석

경쟁사 분석

여행 계획을 위한 웹 페이지



- 하나 투어
- <https://m.hanatour.com/dcr/>

- 호텔스 닷컴
- <https://kr.hotels.com/>

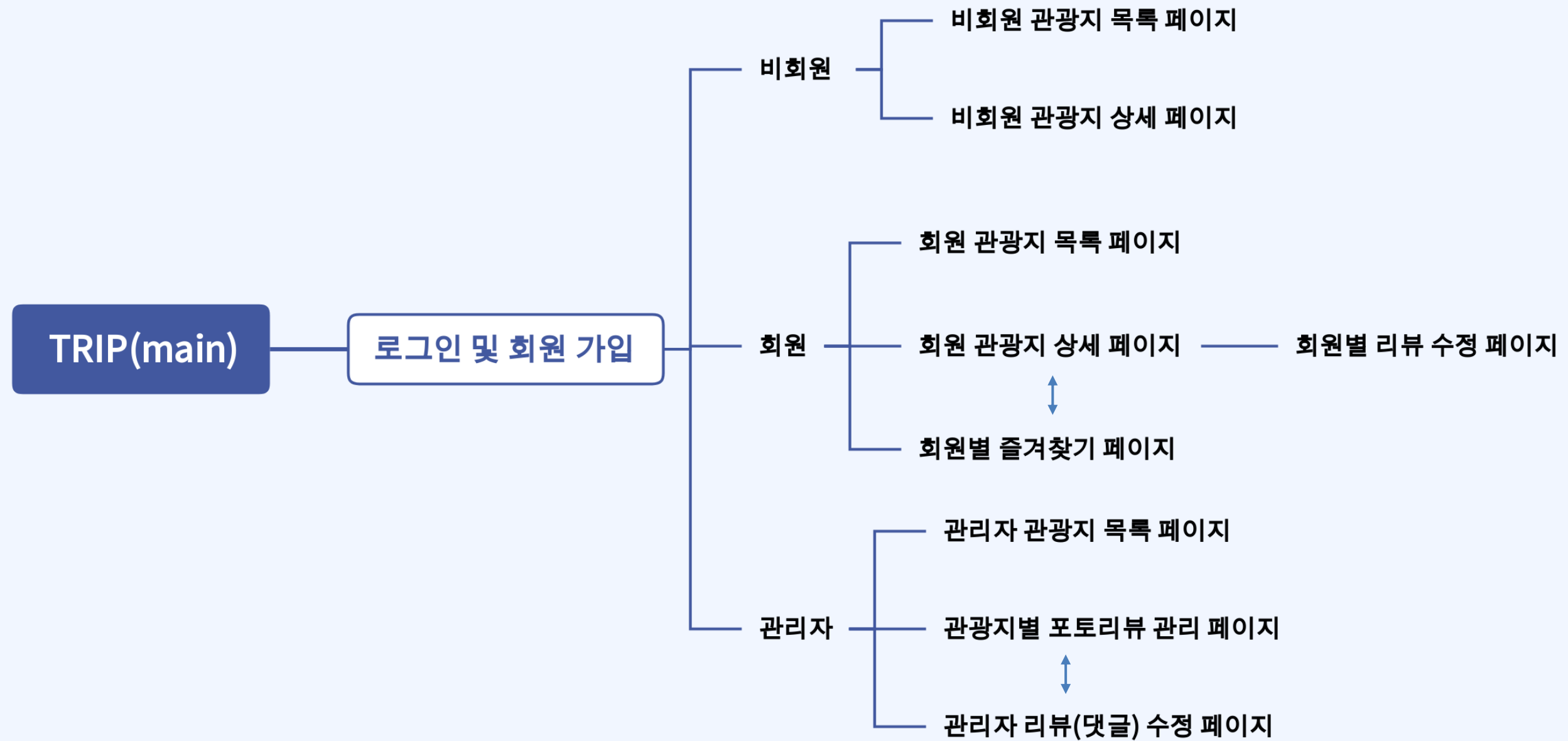
- 트리플
- <https://triple.guide/>



# 3. 웹 사이트 및 DB 설계

웹 사이트 설계

여행 계획을 위한 웹 페이지

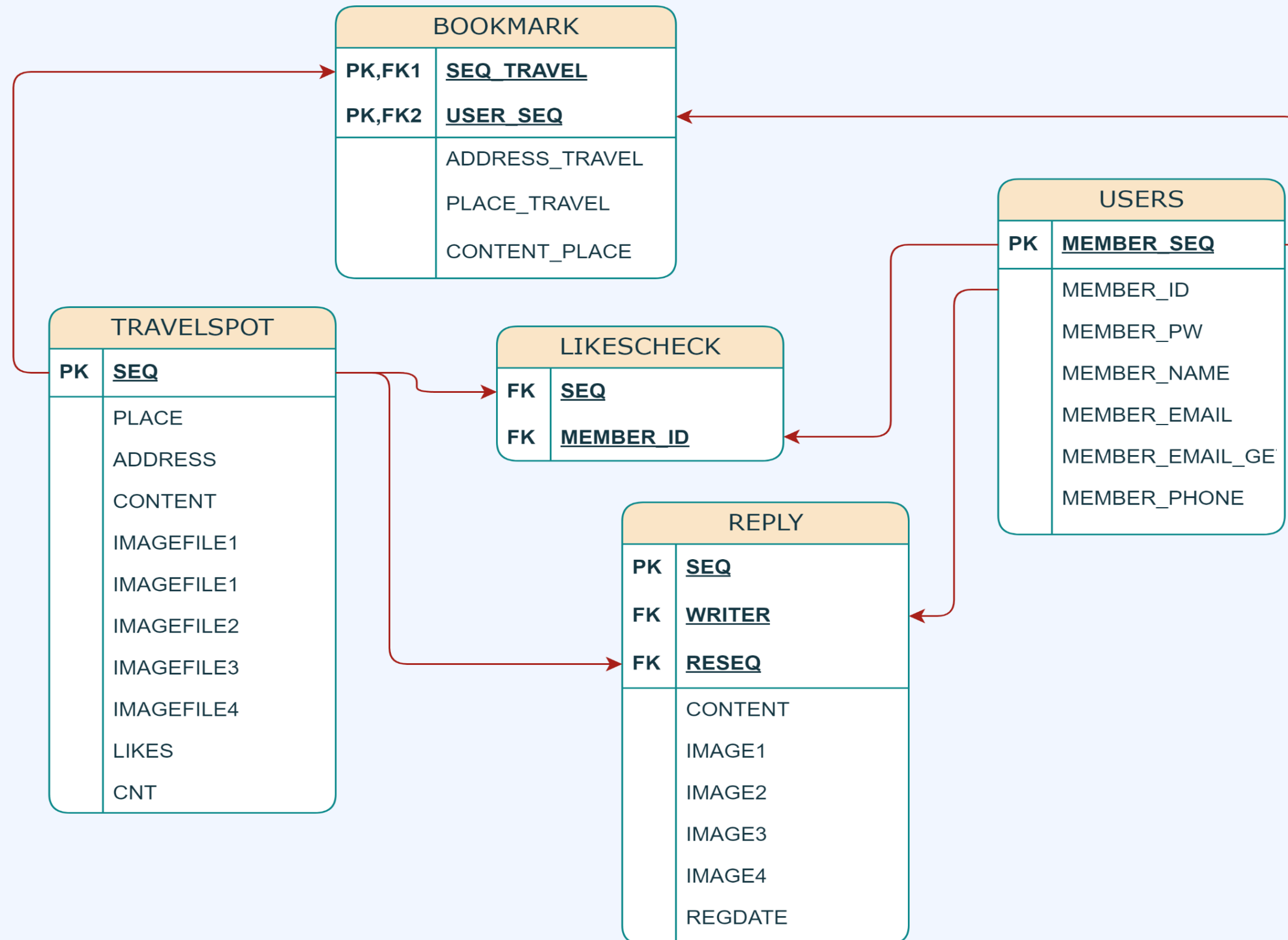




# 3. 웹 사이트 및 DB 설계

DB 설계

여행 계획을 위한 웹 페이지

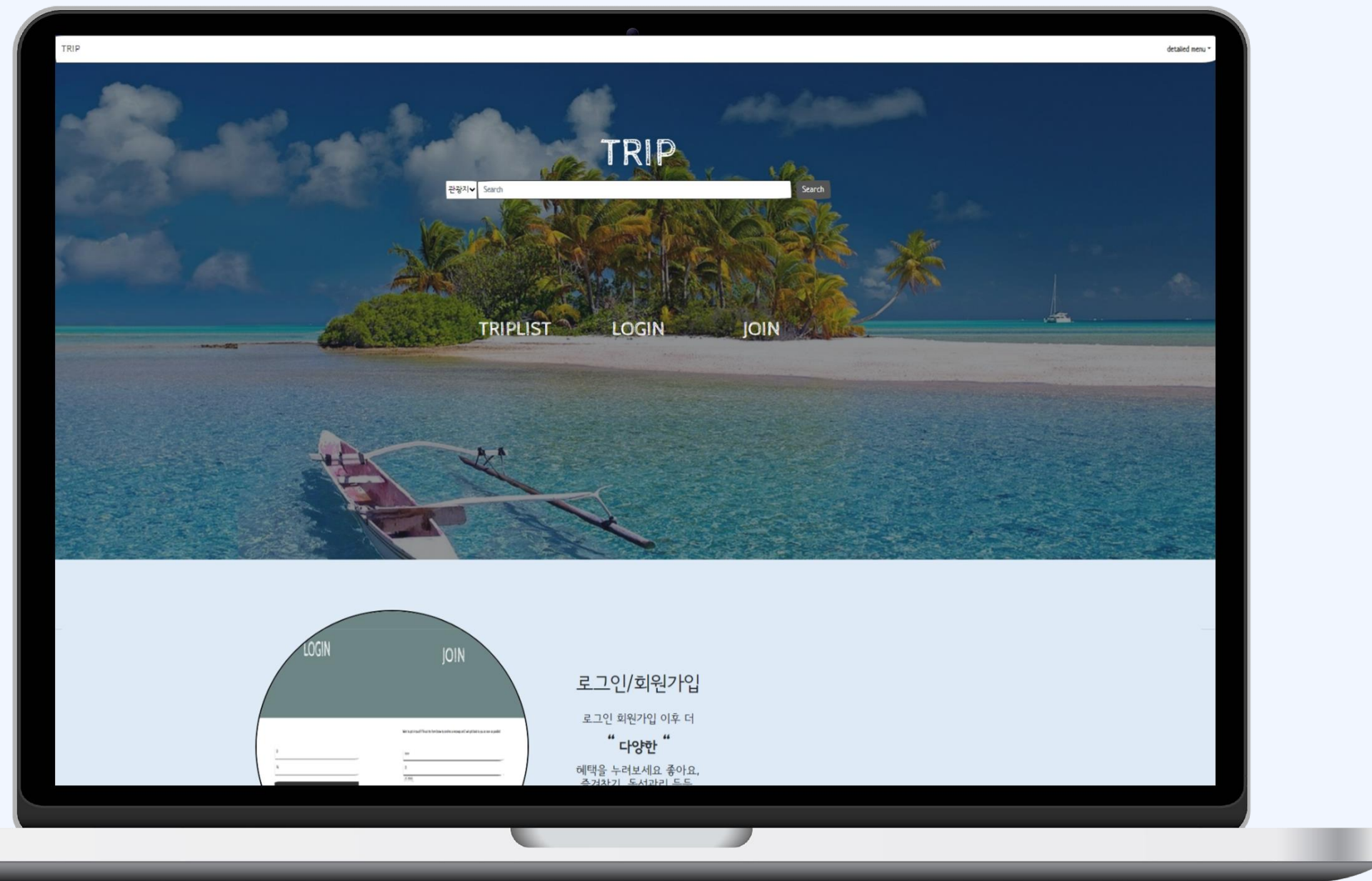




## 4. 기능 구현

메인 화면

여행 계획을 위한 웹 페이지



- ① 현재 보고 있는 관광지 주변 관광지 추천
- ② 즐겨찾기 기능
- ③ 검색어 자동완성

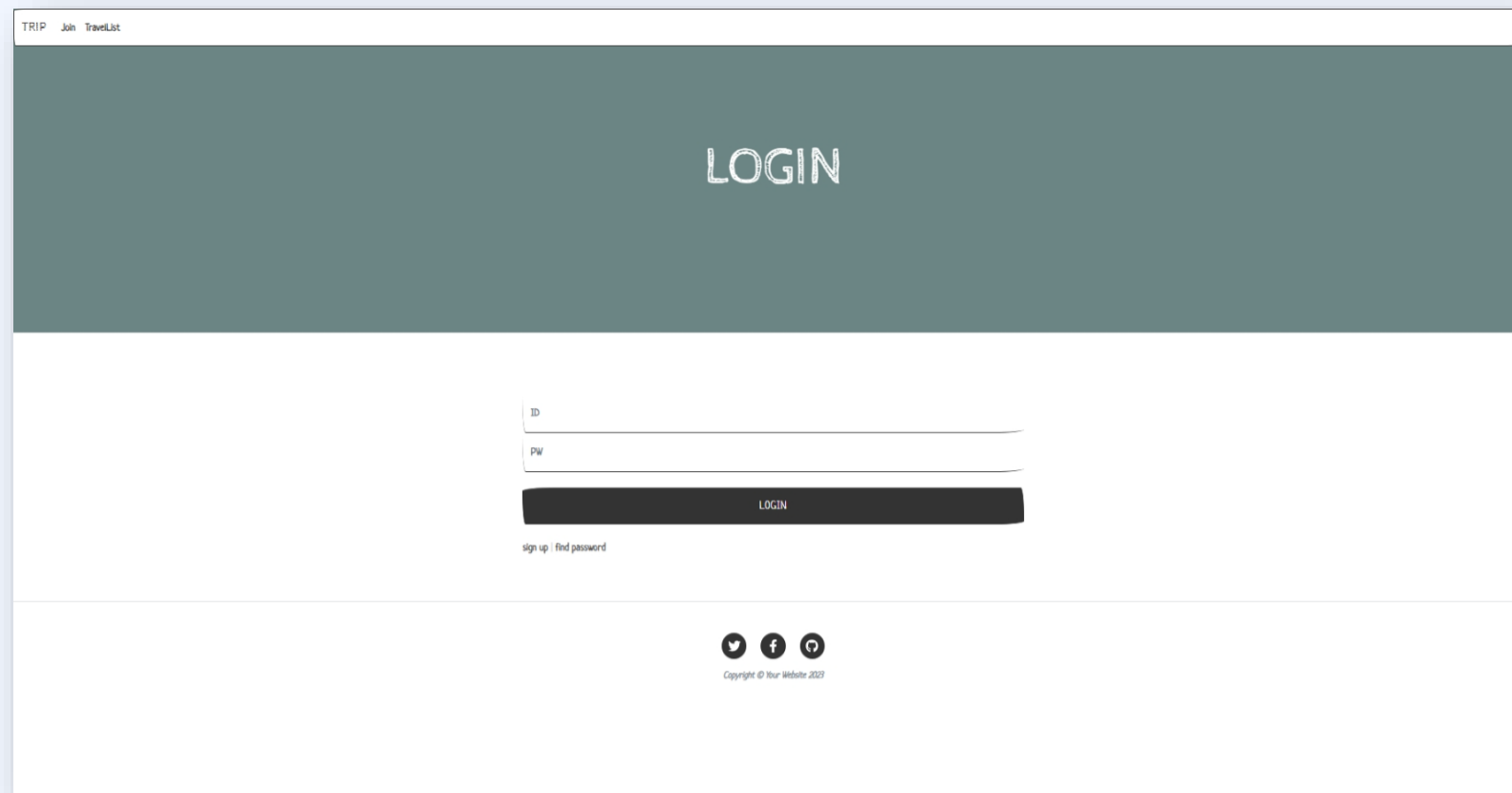




# 4. 기능 구현

기능 구현 및 시연

여행 계획을 위한 웹 페이지



\*회원 정보를 확인하시려면 비밀번호를 입력해주세요

아이디  
비밀번호

hong

확인

```
// 로그인 POST
@PostMapping(value = "/login.me") // 4.3 이후 버전
public String loginProc(UserVO vo, HttpSession session) {

    if (vo.getMEMBER_ID() == null || vo.getMEMBER_ID().equals("")) {
        throw new ArithmeticException("아이디는 반드시 입력해야 합니다.");
    }

    String retVal = null;
    UserVO user = userService.getUser(vo);
    System.out.println(user);
    if ((user != null) && (user.getMEMBER_PW().equals(vo.getMEMBER_PW()))) {
        System.out.println(user);
        session.setAttribute("id", vo.getMEMBER_ID());
        session.setAttribute("user", user);
        if (user.getMEMBER_ID().equals("admin")) {
            retVal = "redirect:getTravelSpotList.do";
        } else {
            retVal = "redirect:getTravelSpotListuser.do";
        }
    } else {
        session.setAttribute("id", vo.getMEMBER_ID());
        retVal = "redirect:login.me";
    }

    return retVal;
}

// 로그아웃
@RequestMapping("logout.me")
public String logoutProc(HttpSession session) {

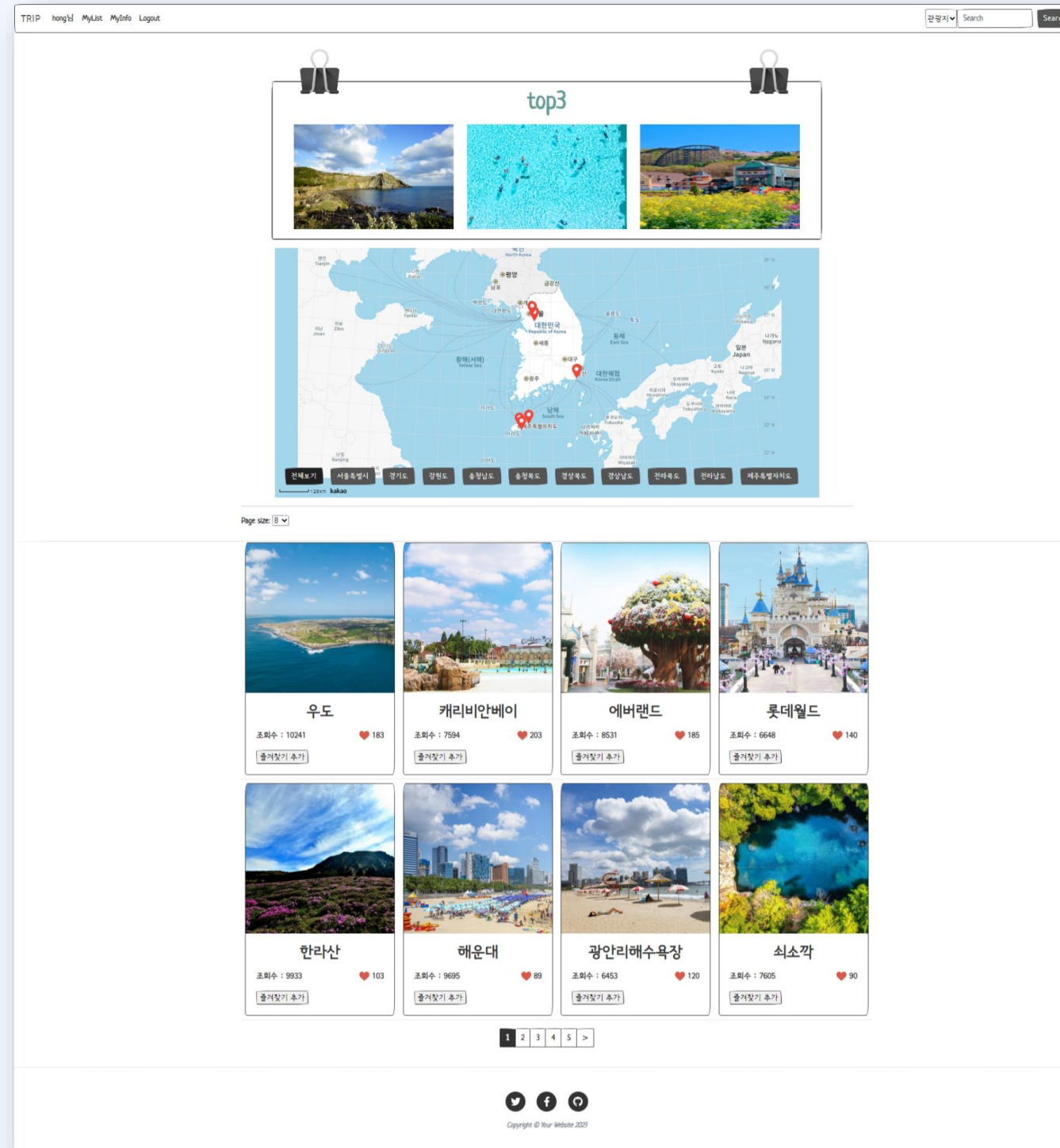
    session.invalidate();

    return "redirect:login.me";
}
```



# 4. 기능 구현

## 기능 구현 및 시연



여행 계획을 위한 웹 페이지

```
// 명소 목록 회원
@RequestMapping(value = "/getTravelSpotListuser.do")
public String getTravelSpotListuser(PagingVO vo, Model model, HttpServletRequest request,
    Elastickeyword elastickwordVO, BookMarkVO bookMarkVO,
    @RequestParam(value = "searchCondition", required = false) String searchCondition,
    @RequestParam(value = "searchKeyword", required = false) String searchKeyword,
    @RequestParam(value = "nowPage", required = false) String nowPage,
    @RequestParam(value = "cntPerPage", required = false) String cntPerPage) {
    if (vo.getSearchCondition() == null) {
        vo.setSearchCondition("PLACE");
    }
    if (vo.getSearchKeyword() == null) {
        vo.setSearchKeyword("");
    }

    int total = travelSpotService.countBoard(vo);
    if (nowPage == null && cntPerPage == null) {
        nowPage = "1";
        cntPerPage = "8";
    } else if (nowPage == null) {
        nowPage = "1";
    } else if (cntPerPage == null) {
        cntPerPage = "8";
    }
    System.out.println(nowPage + cntPerPage); // 18에서 28이 뜸.
    vo = new PagingVO(total, Integer.parseInt(nowPage), Integer.parseInt(cntPerPage), searchCondition,
        searchKeyword);
    System.out.println(vo);
    model.addAttribute("paging", vo);
    model.addAttribute("travelList", travelSpotService.getTravelSpotList(vo));
    model.addAttribute("top3", travelSpotService.getTopThree());

    model.addAttribute("checkMark", request.getParameter("checkMark"));
    model.addAttribute("checksseq", request.getParameter("seq_travel"));
    model.addAttribute("checkseq", request.getParameter("checkseq"));
    elastickwordVO.setKword("");

    if (vo.getSearchKeyword() != null && elastickwordVO.getKword() == "") {
        StringTokenizer Kwordtoken = new StringTokenizer(vo.getSearchKeyword(), " ");

        System.out.println(Kwordtoken.hasMoreTokens());
        while (Kwordtoken.hasMoreTokens()) {
            elastickwordVO.setKword(Kwordtoken.nextToken());
            System.out.println(elastickwordVO.getKword());
            elastickwordVO.setTimestamp(Instant.now());
            System.out.println(elastickwordVO.getTimestamp());

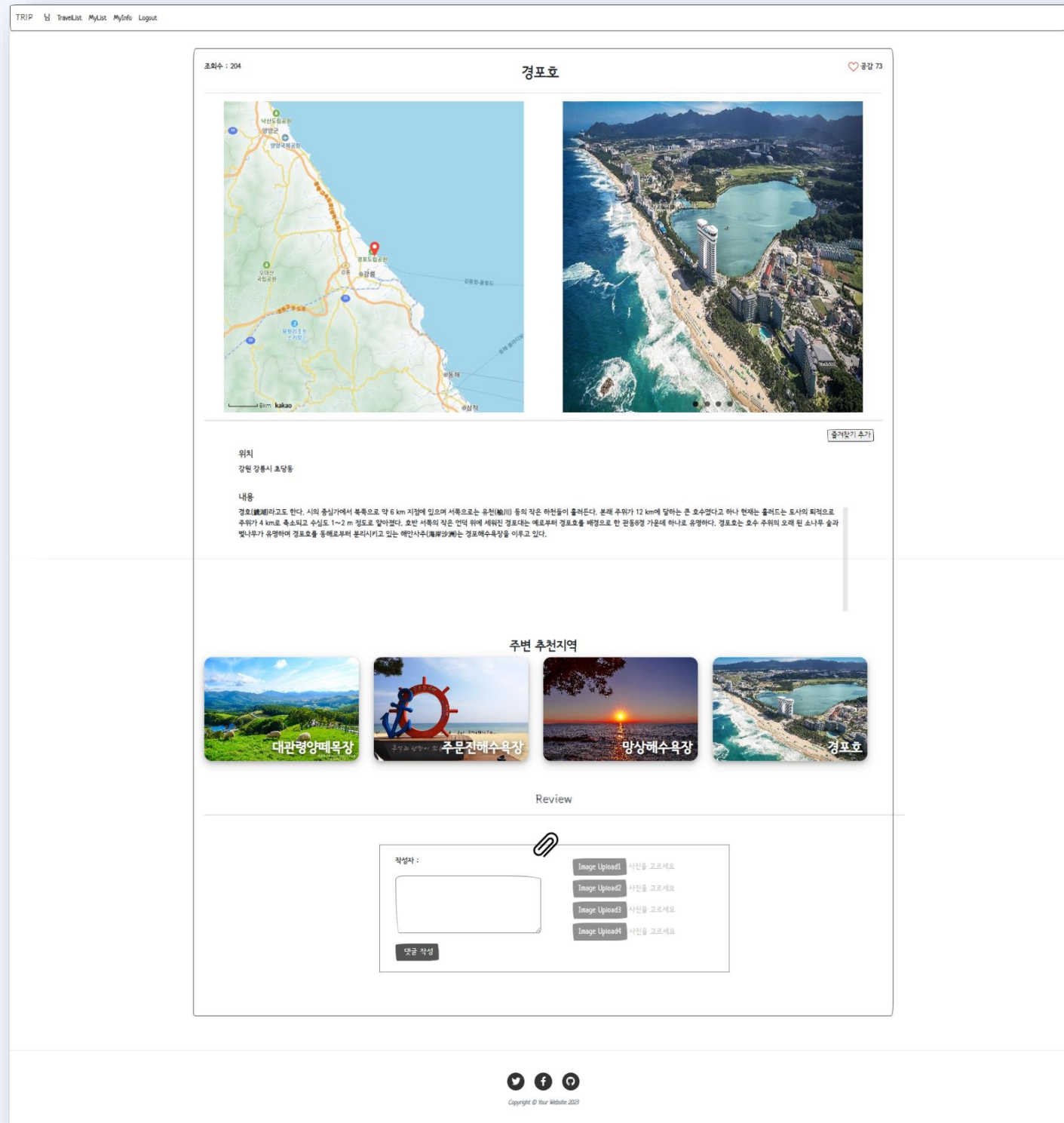
            save(elastickwordVO);
        }
    }
    return "getTravelSpotListuser";
}
```



# 4. 기능 구현

기능 구현 및 시연

여행 계획을 위한 웹 페이지



```
// 명소 상세 조회(회원)
@RequestMapping(value = "/getTravelSpotuser.do")
public String getTravelSpotuser(TravelSpotVO vo, ReplyVO replyVO, LikesVO likes, Model model,
    HttpServletRequest request) {
    HttpSession session = request.getSession();
    String sessionId = session.getId();

    Integer viewCnt = (Integer) session.getAttribute("viewCnt_" + vo.getSeq());

    if (!sessionId.equals(session.getAttribute("viewSessionId_" + vo.getSeq())) {
        travelSpotService.incrementViewCount(vo.getSeq());
        session.setAttribute("viewSessionId_" + vo.getSeq(), sessionId);
        viewCnt = (viewCnt == null) ? 1 : viewCnt + 1;
    } else {
        viewCnt = (viewCnt == null) ? 0 : viewCnt;
    }

    session.setAttribute("viewCnt_" + vo.getSeq(), viewCnt);

    // 댓글 조회 호출
    List<ReplyVO> list = replyService.getReplyList(replyVO);
    model.addAttribute("replyList", list);

    // 여기서 travelSpotService.getTravelSpot 메서드가 TravelSpotVO 객체를 반환하는 것으로 가정합니다.
    TravelSpotVO travel = travelSpotService.getTravelSpot(vo.getSeq());
    model.addAttribute("travel", travel);

    // 좋아요 기능
    if ((String) session.getAttribute("id") != null) {
        likes.setMEMBER_ID((String) session.getAttribute("id"));

        int result = likesService.checkLikeSelect(likes);
        session.setAttribute("result", result);
    }

    // 30킬로미터 반경 내 장소 가져오기
    List<PlaceInfo> nearbyPlaces = searchService.getPlacesNearby(vo.getSeq(), 30);
    model.addAttribute("nearbyPlaces", nearbyPlaces);

    return "getTravelSpotuser";
}
```





# 4. 기능 구현

## 기능 구현 및 시연

여행 계획을 위한 웹 페이지



Logstash-카카오 맵 api에서 위치정보로 위도, 경도 값을 가져옴

```
filter {
  ruby {
    code => '
      require "net/http"
      require "json"
      require "uri"

      # 장소 이름 추출
      place = event.get("place")

      # 주소 정보 추출
      address = event.get("address")
      kakao_api_key = "51da9463665748bbe0f6e31847b83198" # 실제 카카오 API 키로 대체

      # 주소 정보를 이용하여 카카오 API 호출
      encoded_address = URI.encode_www_form_component(address)
      uri = URI("https://dapi.kakao.com/v2/local/search/address.json?query=#{encoded_address}")
      http = Net::HTTP.new(uri.host, uri.port)
      http.use_ssl = true

      request = Net::HTTP::Get.new(uri)
      request["Authorization"] = "KakaoAK #{kakao_api_key}"

      response = http.request(request)
      data = JSON.parse(response.body)

      # 카카오 API 호출 결과 확인 및 위치 정보 추출
      if response.code == "200" && data["documents"].any?
        latitude = data["documents"][0]["y"].to_f
        longitude = data["documents"][0]["x"].to_f

        # 경도와 위도 정보를 location 필드에 저장
        event.set("location", [longitude, latitude])

        # place_suggest 필드에 장소 이름 저장
        event.set("place_suggest", place)
      else
        event.set("location", nil)
        event.set("place_suggest", nil)
      end
    '
  }
}
```

```
PUT /travelspot
{
  "settings": {
    "number_of_shards": 1,
    "number_of_replicas": 1
  },
  "mappings": {
    "properties": {
      "seq": { "type": "integer" },
      "place": { "type": "text" },
      "address": { "type": "text" },
      "location": { "type": "geo_point" },
      "place_suggest": {
        "type": "completion"
      }
    }
  }
}
```

```
public List<PlaceInfo> getPlacesNearby(int seq, double distanceInKilometers) {
  try {
    SearchRequest searchRequest = new SearchRequest("travelspot"); // Elasticsearch index name
    SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();

    TermQueryBuilder termQuery = QueryBuilders.termQuery("seq", seq);
    searchSourceBuilder.query(termQuery);
    searchRequest.source(searchSourceBuilder);

    SearchResponse searchResponse = elasticsearchClient.search(searchRequest, RequestOptions.DEFAULT);

    List<PlaceInfo> nearbyPlaces = new ArrayList<>();
    double latitude = 0.0;
    double longitude = 0.0;
    String placeName = null;
    int seqn = 0;

    for (SearchHit hit : searchResponse.getHits().getHits()) {
      Map<String, Object> sourceAsMap = hit.getSourceAsMap();
      List<Double> location = (List<Double>) sourceAsMap.get("location");
      placeName = (String) sourceAsMap.get("place");
      seqn = (int) sourceAsMap.get("seq");

      if (location != null && location.size() == 2 && placeName != null & seqn != -1) {
        latitude = location.get(1);
        longitude = location.get(0);
      }

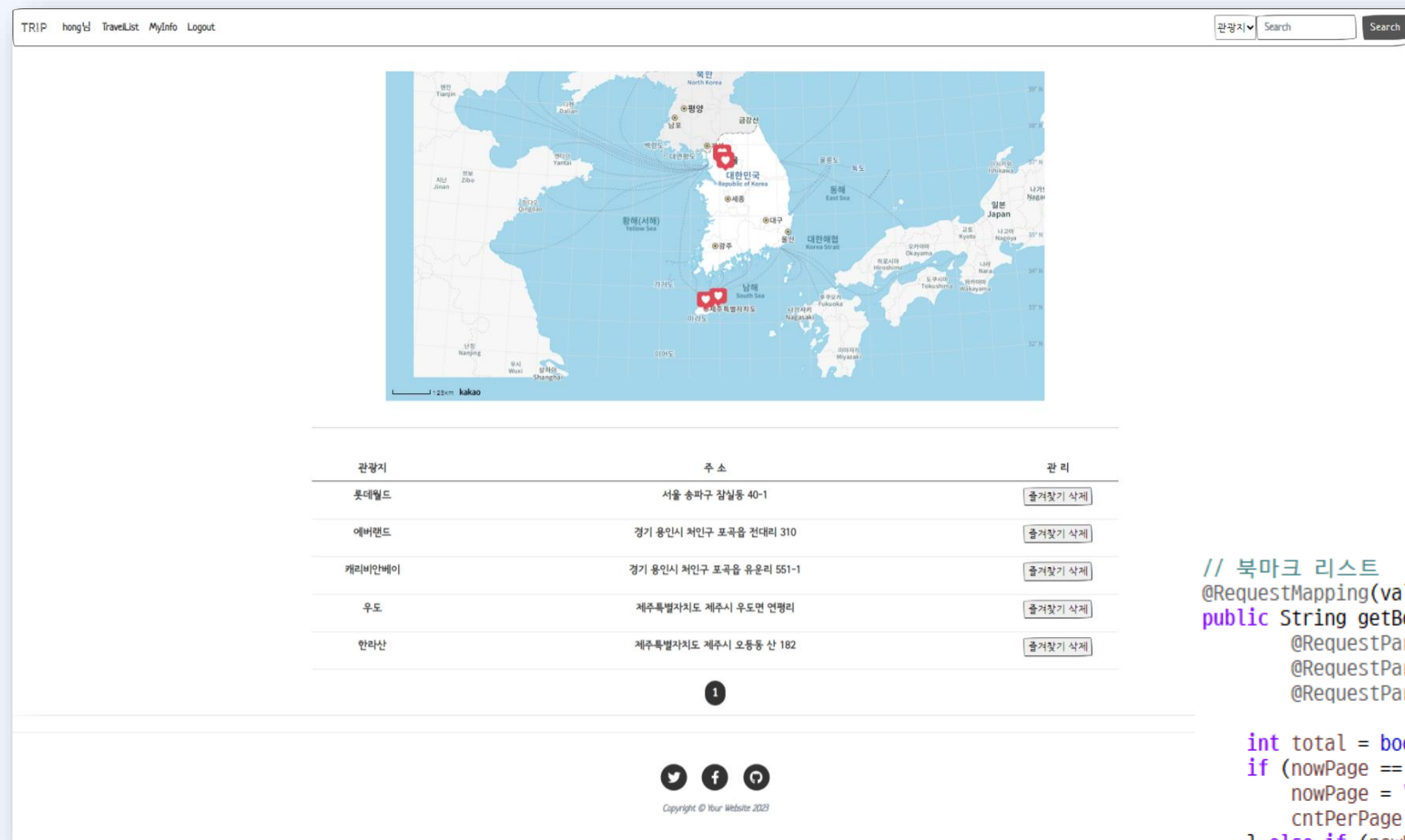
      if (latitude != 0.0 && longitude != 0.0) {
        List<PlaceInfo> nearbyPlacesList = getPlacesNearbyFromCoordinates(latitude, longitude, distanceInKilometers);
        return nearbyPlacesList;
      } else {
        return new ArrayList<>(); // Return an empty list if no valid location found
      }
    }
  } catch (IOException e) {
    // Handle IOException with more appropriate error handling
    e.printStackTrace();
    throw new RuntimeException("Error while retrieving nearby places.", e);
  }
}
```



# 4. 기능 구현

기능 구현 및 시연

여행 계획을 위한 웹 페이지



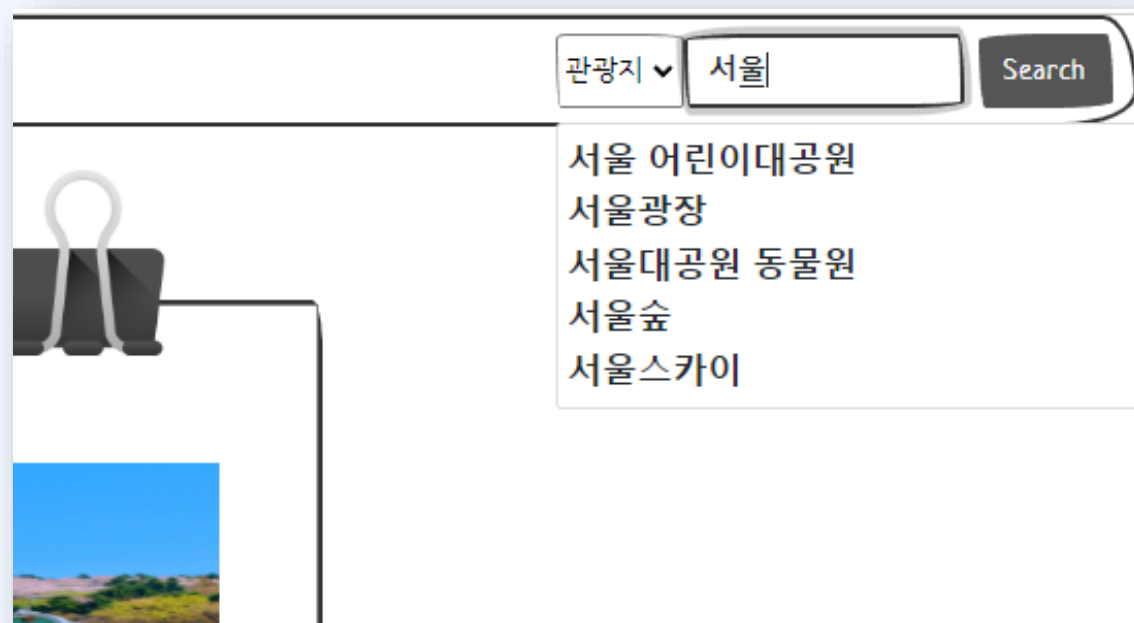
```
// 북마크 리스트
@RequestMapping(value = "/bookmarklist.do")
public String getBookMarkTravelList(BMPagingVO vo, Model model,
    @RequestParam(value = "user_seq", required = false) int user_seq,
    @RequestParam(value = "nowPage", required = false) String nowPage,
    @RequestParam(value = "cntPerPage", required = false) String cntPerPage) {

    int total = bookMarkService.countBoard(vo);
    if (nowPage == null && cntPerPage == null) {
        nowPage = "1";
        cntPerPage = "8";
    } else if (nowPage == null) {
        nowPage = "1";
    } else if (cntPerPage == null) {
        cntPerPage = "8";
    }
    vo = new BMPagingVO(total, Integer.parseInt(nowPage), Integer.parseInt(cntPerPage), user_seq);
    model.addAttribute("paging", vo);
    model.addAttribute("bookmarklist", bookMarkService.getBookMarkList(vo));
    return "getBookMarkTravelSpotList";
}
```



## 4. 기능 구현

기능 구현 및 시연



```
POST /_reindex
{
  "source": {
    "index": "travelspot"
  },
  "dest": {
    "index": "autocomplete"
  },
  "script": {
    "source": ""
    | ctx._source.place_suggest = [ctx._source.place];
    ""
  }
}
```

여행 계획을 위한 웹 페이지

```
public List<String> generateRelatedKeywords(List<String> tokens) throws IOException {
    List<String> relatedKeywords = new ArrayList<>();
    SearchRequest searchRequest = new SearchRequest("travelspot"); // Elasticsearch index name
    SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();

    for (String token : tokens) {
        CompletionSuggestionBuilder completionSuggestionBuilder = SuggestBuilders
            .completionSuggestion("place_suggest").prefix(token).size(5);

        searchSourceBuilder
            .suggest(new SuggestBuilder().addSuggestion("place_suggest", completionSuggestionBuilder));
    }

    searchRequest.source(searchSourceBuilder);

    try {
        SearchResponse searchResponse = elasticsearchClient.search(searchRequest, RequestOptions.DEFAULT);

        for (String token : tokens) {
            CompletionSuggestion suggest = searchResponse.getSuggest().getSuggestion("place_suggest");

            if (suggest != null) {
                List<String> suggestions = suggest.getEntries().stream()
                    .flatMap(entry -> entry.getOptions().stream())
                    .map(option -> option.getText().string())
                    .collect(Collectors.toList());

                relatedKeywords.addAll(suggestions);
            }
        }
    } catch (ElasticsearchStatusException e) {
        // Handle Elasticsearch exception
        e.printStackTrace();
    }

    return relatedKeywords;
}
```



## 5. 기대 효과

기대효과

여행 계획을 위한 웹 페이지



국내 관광지 홍보 효과

관광객의 편리성 증가

관광지 별 리뷰 확인



# 감사합니다.

## TRIP-관광지 웹 사이트

양경민

E-mail:mi568977@naver.com

Tel:010-2144-6263