

BabPool

풀잎이들을 위한 밥 추천 웹 사이트

웹서버 프로그래밍 기말 프로젝트

2020303002 김동국

2022301032 박수연

목차

01

프로젝트
주제 설명

어떤 사이트인가?
프로젝트의 목적은?

02

프로젝트 구조
전체 기능 구성
설명

ERD + 페이지 · 기능
로드맵 첨부

03

시연 영상

수업에 포함된 기능
수업 외 추가한 기능

04

핵심 기능 및
코드 설명

어필하고 싶은 부분
노력한 부분

05

시행착오

진행하며 생긴 문제점
& 해결 방안

06

팀원의 참여비율

합의하여 나온 결과

01. BabPool?

“

학교를 다니면서 가장 많이 했던 고민

뭐 먹지?

누가 좀 물어보줬으면...
주변에 뭐 먹을 곳 어디 없나?

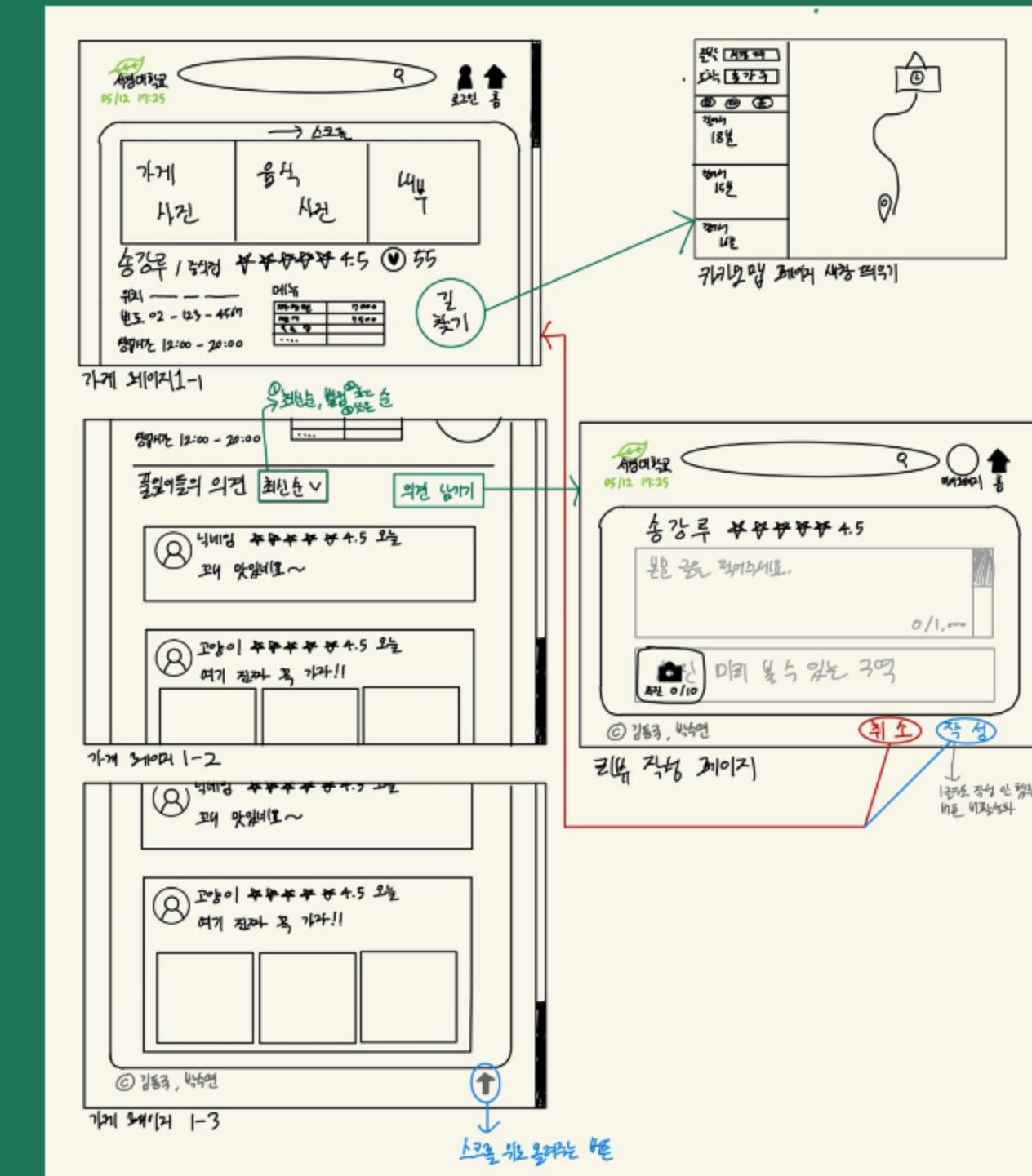
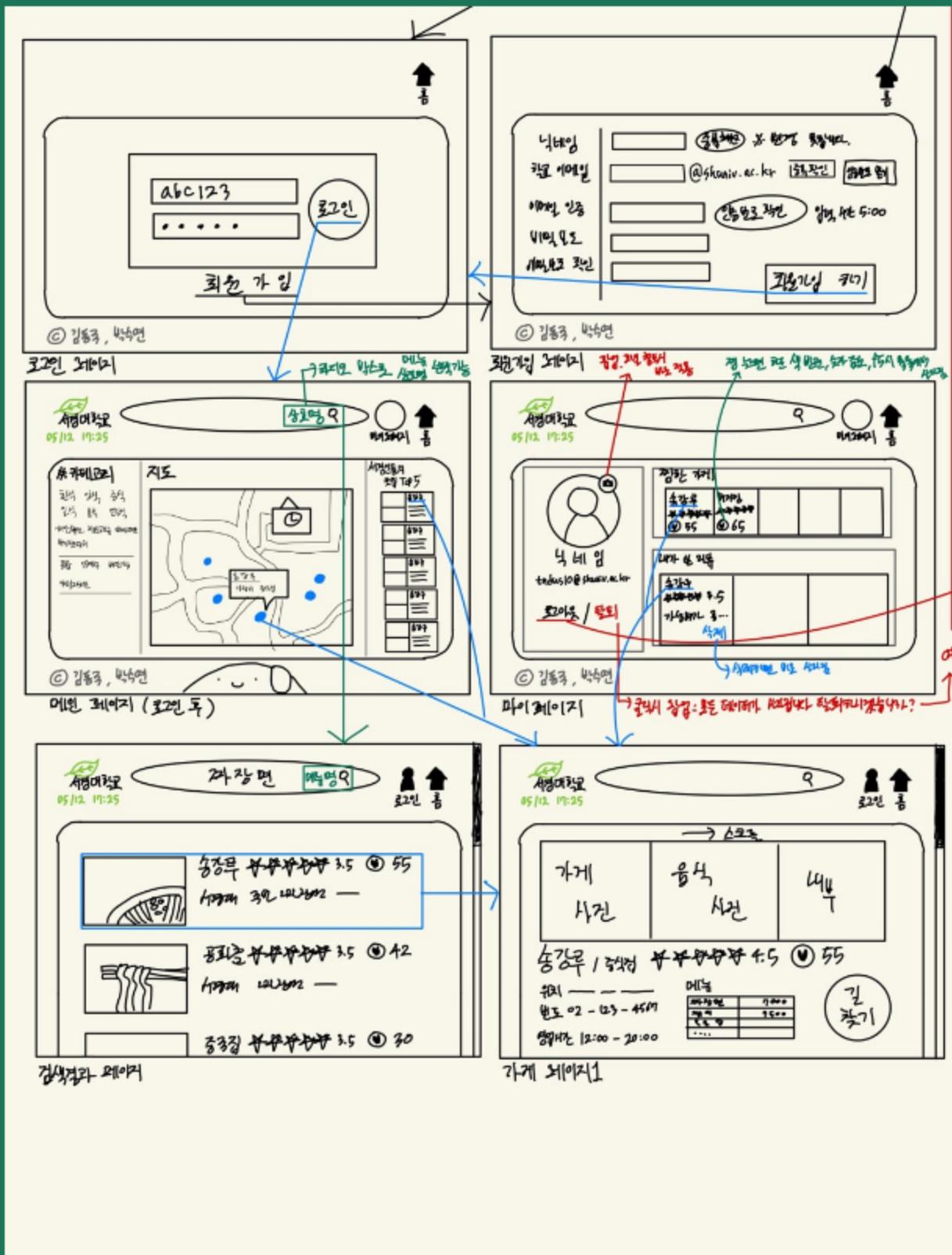
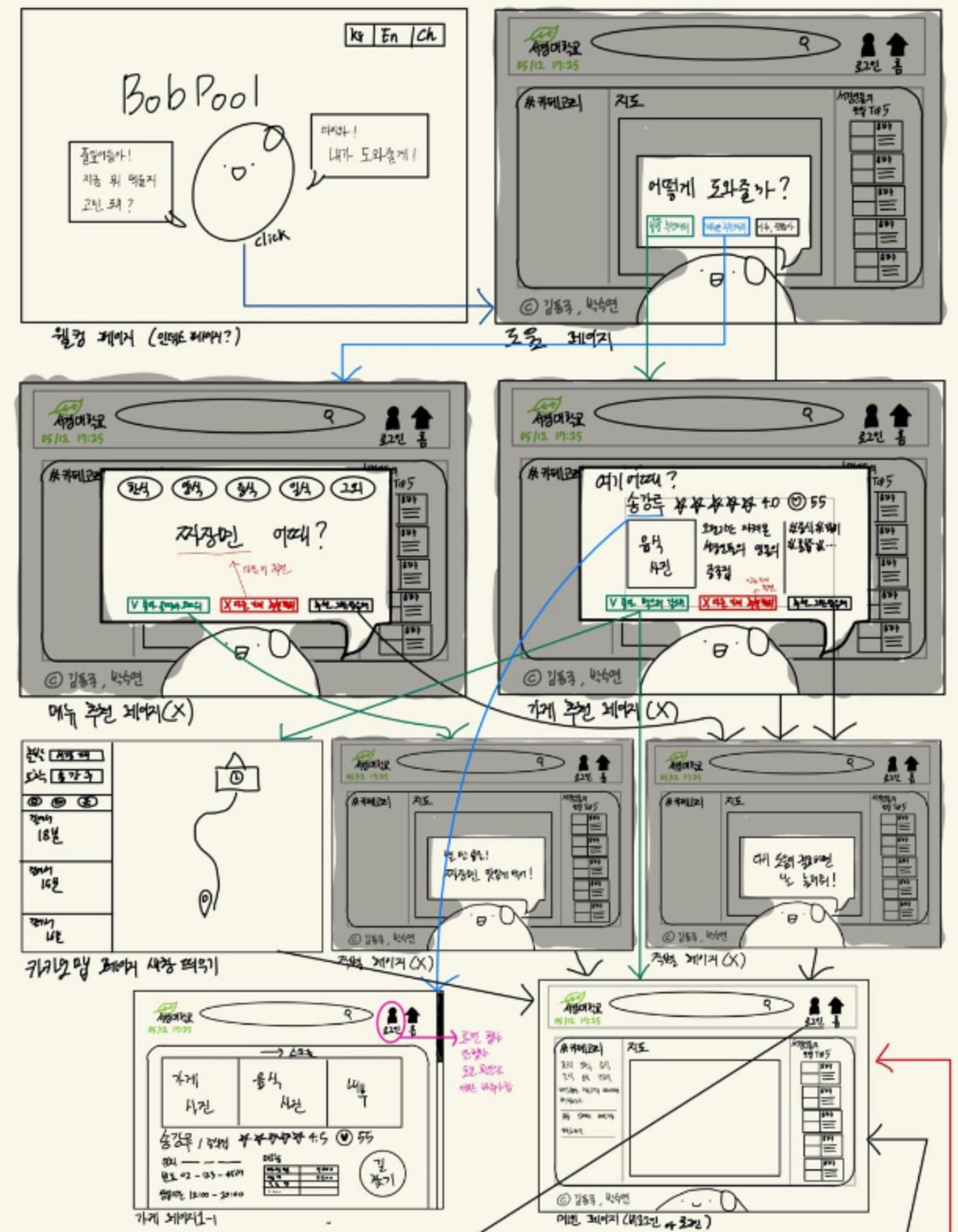
**뭐 먹을지 고민하는 것도
귀찮다...**



01. BabPool?

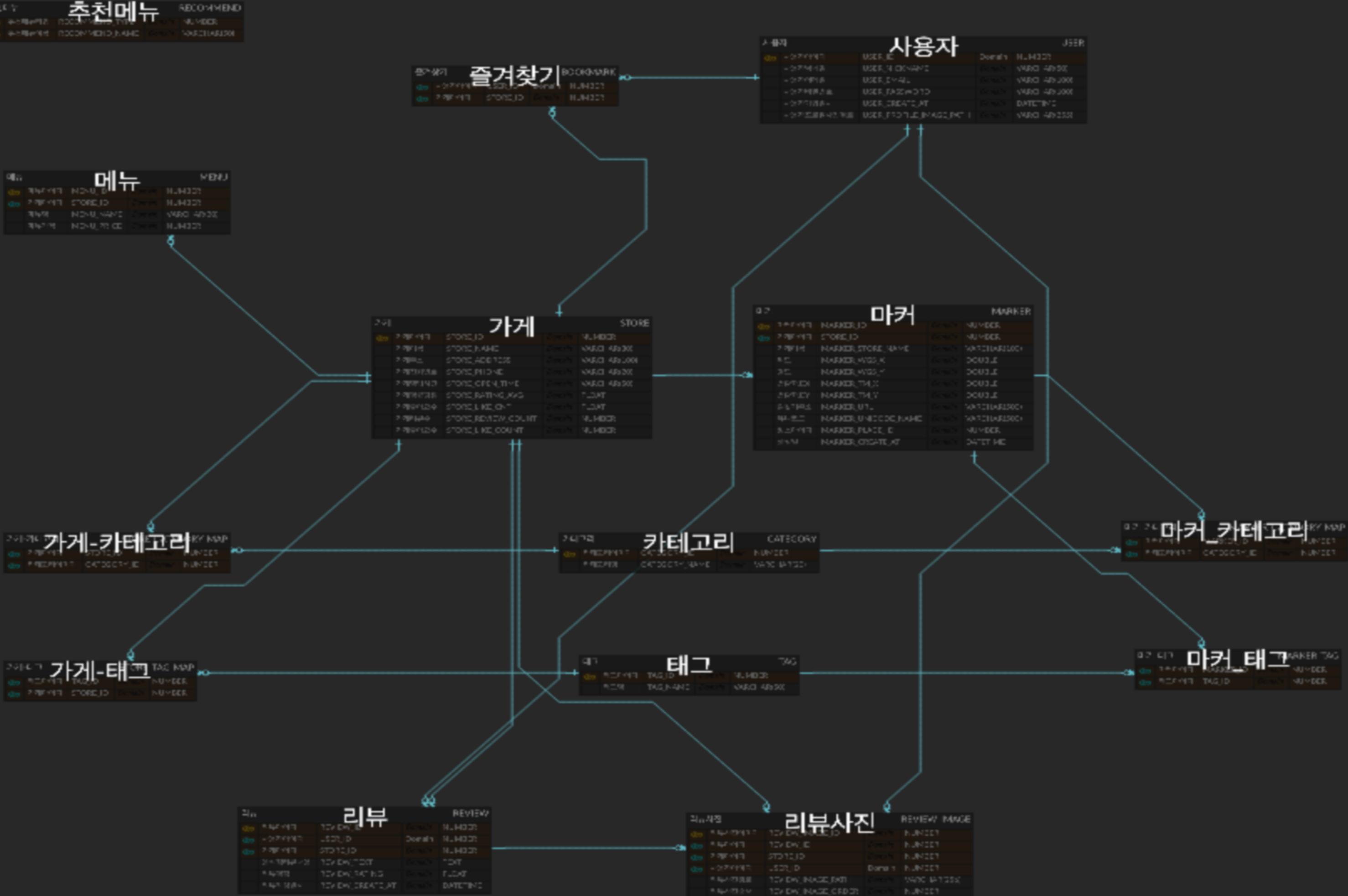


01. BabPool !

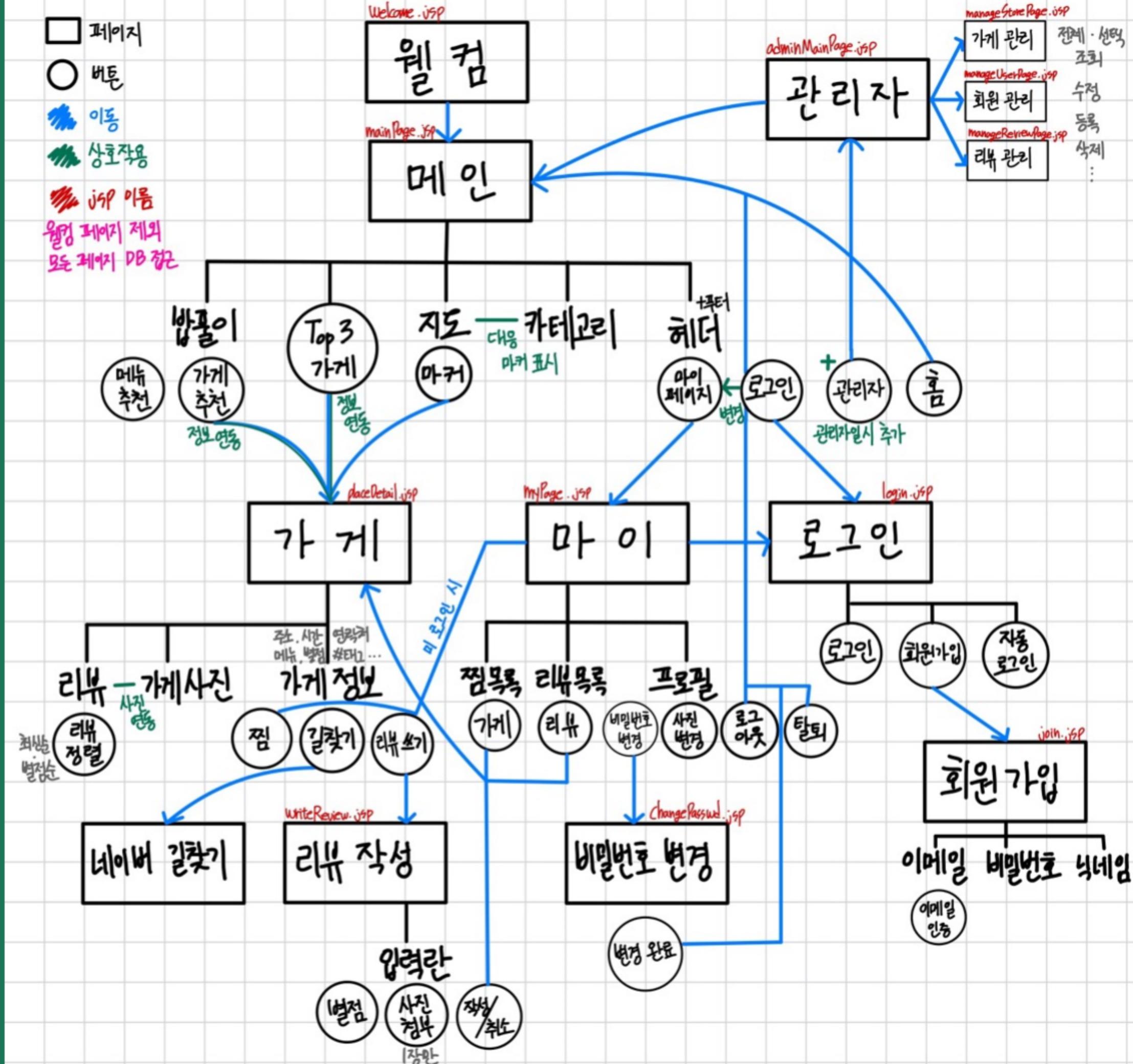


0

2.

E
R
D

02. 페이지 로드맵



03. 시연 영상

04. 핵심 기능

메인 페이지 - 지도 & 마커 - 카테고리 매팅

가게 페이지 - 검색 API를 통한 길찾기 URL

메인 페이지 - 밥풀이 UI

전체 페이지 - DB 연동, 다수의 기능 활용

04. 핵심 기능 코드 설명

```
<section class="map-section"> 네이버 지도 띄우기
<div id="map" style="width: 100%; height: 100%; border-radius: 12px;"></div>
```

<script type="text/javascript" src="[https://oapi.map.naver.com/openapi/v3/maps.js?ncpKeyId=%ApiKeyUtil.getId\("ncpId"\)%&language=<%= mapLang %>](https://oapi.map.naver.com/openapi/v3/maps.js?ncpKeyId=%ApiKeyUtil.getId()"*></script> → 쿼리로 받아 로케일- 언어를 통해서
지도 언어 변경

```
<script>
const markerList = [
<c:forEach var="m" items="${markerList}" varStatus="loop">
{
    lat: ${m.wgsY},
    lng: ${m.wgsX},
    storeId: ${m.storeId},
    storeName: "${fn:replace(m.storeName, "", "\\\\")}"
}<c:if test="${!loop.last}"></c:if>
</c:forEach>
];
```

```
let map;
let markers = [];
```

```
document.addEventListener("DOMContentLoaded", function () {
    map = new naver.maps.Map('map', {
        center: new naver.maps.LatLng(37.6154, 127.0115), → 중심 좌표 설정
        zoom: 17
    });
    drawMarkers(markerList);
});
```

04. 핵심 기능 코드 설명

```
function drawMarkers(data) {
    clearMarkers();
    data.forEach(marker => {
        const m = new naver.maps.Marker({
            position: new naver.maps.LatLng(marker.lat, marker.lng),
            map: map
        });

        const infoWindow = new naver.maps.InfoWindow({ content:'<div style="padding:5px;">' + marker.storeName
+ '</div>' });

        naver.maps.Event.addListener(m, "mouseover", () => infoWindow.open(map, m)); → 마커 위 마우스 시 가게 이름
        naver.maps.Event.addListener(m, "mouseout", () => infoWindow.close());
        naver.maps.Event.addListener(m, "click", function () {
            window.open('<%= request.getContextPath() %>/PlaceDetailServlet?storeId=' + marker.storeId, '_blank');
        }); → 마커 클릭 시 해당 가게의 상세페이지로 이동
    });
}

function clearMarkers() {
    markers.forEach(m => m.setMap(null));
    markers = [];
}
```

</script>

</section>

04. 핵심 기능 코드 설명

public class AdminMarkerInsertServlet extends HttpServlet { → 관리자 페이지에서 마커 추가 버튼에 대한 서블릿

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,  
IOException {
```

```
    Connection conn = null;
```

```
    try {
```

```
        request.setCharacterEncoding("UTF-8");
```

```
        int storeId = Integer.parseInt(request.getParameter("storeId")); → 입력받은 매핑할 가게 ID
```

```
// storeId 기반 가게 정보 조회
```

```
    conn = DBUtil.getConnection();
```

```
    StoreDAO storeDAO = new StoreDAO(conn);
```

```
    StoreDTO store = storeDAO.getStoreById(storeId);
```

```
    if (store == null) {
```

```
        response.sendRedirect("manageStorePage.jsp?error=notfound");
```

```
        return;
```

```
}
```

```
String placeName = store.getName();
```

```
double[] wgs = SearchAndConvertPlaceUtil.getPlaceCoordinates(placeName); → API 메서드를 활용한 좌표 삽입
```

```
if (wgs == null) {
```

```
    response.sendRedirect("manageStorePage.jsp?error=placefail");
```

```
    return;
```

```
}
```

```
double[] tm = SearchAndConvertPlaceUtil.convertWGS84To3857(wgs[0], wgs[1]); → 메서드를 사용한 좌표 변환 및 삽
```

입

```
String unicode = URLEncoder.encode(placeName, "UTF-8"); → URL에 사용될 unicode 리턴
```

04. 핵심 기능 코드 설명

```
// 수동 추출한 placeId를 추후 관리자가 입력해야 함
int placeId = 0; // 또는 DB에서 가져오는 구조로 변경 가능
String directionUrl = NaverMapUrlUtil.generateDirectionUrl(tm[0], tm[1], unicode, placeId); → 메서드를 통해 URL 리턴

// DTO 구성
MarkerDTO dto = new MarkerDTO();
dto.setStoreId(storeId);
dto.setStoreName(placeName);
dto.setWgsX(wgs[0]);
dto.setWgsY(wgs[1]);
dto.setTmX(tm[0]);
dto.setTmY(tm[1]);
dto.setUnicodeName(unicode);
dto.setPlaceId(placeId);
dto.setUrl(directionUrl);

// DB 저장
MarkerDAO dao = new MarkerDAO(conn);
boolean success = dao.insertMarker(dto);

...
}
```

04. 핵심 기능 코드 설명

```
private static final String CLIENT_ID = ApiKeyUtil.get("naver.clientId");
private static final String CLIENT_SECRET = ApiKeyUtil.get("naver.clientSecret");
```

// 장소명으로 WGS84 좌표 가져오기

```
public static double[] getPlaceCoordinates(String placeName) throws IOException {
    String encodedPlaceName = URLEncoder.encode(placeName, StandardCharsets.UTF_8);
    String apiURL = "https://openapi.naver.com/v1/search/local.json?query=" + encodedPlaceName + "&display=1";
```

```
HttpURLConnection connection = null;
```

```
int attempts = 0;
```

while (attempts < 3) { API의 호출 실패 시 연속 호출을 통하여 값을 받아오기 위해 3번까지 호출하기

```
try {
```

```
    URL url = new URL(apiURL);
    connection = (HttpURLConnection) url.openConnection();
    connection.setRequestMethod("GET");
    connection.setRequestProperty("X-Naver-Client-Id", CLIENT_ID);
    connection.setRequestProperty("X-Naver-Client-Secret", CLIENT_SECRET);
```

```
    int responseCode = connection.getResponseCode();
```

```
    if (responseCode == 200) {
```

```
        BufferedReader in = new BufferedReader(new InputStreamReader(connection.getInputStream(),
        StandardCharsets.UTF_8));
    
```

```
        StringBuilder response = new StringBuilder();
```

```
        String line;
```

```
        while ((line = in.readLine()) != null) response.append(line);
```

```
        in.close();
```

04. 핵심 기능 코드 설명

```
JSONObject json = new JSONObject(response.toString());
JSONArray items = json.getJSONArray("items");
if (items.length() > 0) {
    JSONObject item = items.getJSONObject(0);
    // 이 부분은 네이버 검색 API 는 소수가 아닌 정수로 리턴하므로 기존 wgs 좌표의 소수점까지 소수점 조정
    double x = Double.parseDouble(item.getString("mapx")) / 1e7;
    double y = Double.parseDouble(item.getString("mapy")) / 1e7;
    return new double[]{x, y};
} else {
    BufferedReader err = new BufferedReader(new InputStreamReader(connection.getErrorStream(),
StandardCharsets.UTF_8));
    while (err.readLine() != null); // 오류 메시지 무시
    err.close();
}
} catch (IOException e) {
    e.printStackTrace();
} finally {
    if (connection != null) connection.disconnect();
}
attempts++;
}
return null;
}
```

04. 핵심 기능 코드 설명

```
// WGS84 → TM (EPSG:3857)
public static double[] convertWGS84To3857(double wgsX, double wgsY) { → WGS → TM (네이버 길찾기 URL)
    double x = wgsX * 20037508.34 / 180;
    double y = Math.log(Math.tan((90 + wgsY) * Math.PI / 360)) / (Math.PI / 180);
    y = y * 20037508.34 / 180;
    return new double[]{x, y};
}
```

04. 핵심 기능 코드 설명

public class NaverMapUrlUtil { → 정해진 길찾기 URL 파라미터로 받은 값들을 통하여 서경대 북악관 → 도착지 길찾기 URL 리턴 }

```
private static final String BASE_URL = "https://map.naver.com/p/directions/";
private static final String START_TM_X = "14138890.8265643";
private static final String START_TM_Y = "4525316.5087329";
private static final String START_UNICODE =
"%EC%84%9C%EA%B2%BD%EB%8C%80%ED%95%99%EA%B5%90%EB%B6%81%EC%95%85%EA%B4%80";
private static final String START_PLACE_ID = "17563706";

public static String generateDirectionUrl(double destTmX, double destTmY, String destUnicode, int destPlaceId) {
    return BASE_URL
        + START_TM_X + "," + START_TM_Y + "," + START_UNICODE + "," + START_PLACE_ID + ",PLACE_POI/"
        + destTmX + "," + destTmY + "," + destUnicode + "," + destPlaceId + ",PLACE_POI/-/walk?c=17.00,0,0,0,dh";
}
```

04. 핵심 기능 코드 설명

```
<!-- 마스코트 영역 (수면 확장 적용) -->






<!-- 걸정 오버레이 -->


</div>

<!-- 1단계 : 처음 질문 모달 -->


<p><fmt:message key="mascot.help.question" /></p>
    <button onclick="recommendNearby()"><fmt:message key="mascot.help.recommend.store" /></button>
    <button onclick="recommendMenu()"><fmt:message key="mascot.help.recommend.menu" /></button>
    <button onclick="closeHelp()"><fmt:message key="mascot.help.cancel" /></button>



<!-- 2단계 : 추천 결과 모달 -->


<p>여기 어때?</p>

    <!-- 상단: 가게 이름 / 별점 / 하트 -->
    <div class="recommend-header">
        <span id="storeName"></span>
        <span id="ratingStars"></span>
        <span id="ratingAvg"></span>
        <span class="recommend-like">♡ <span id="likeCount"></span></span>
    </div>

    <!-- 3분할 영역: 사진 / 리뷰 / 태그 -->
    <div class="recommend-body">
        <!-- 원쪽: 음식 사진 -->
        <div class="recommend-image">
            <img id="storeImage" src="" alt="대표사진" onerror="handleNoImage()"/>
            <p id="noImageText" style="display: none; color: gray; font-size: 14px; margin-top: 8px;">
                사진 리뷰가<br>
                등록되지 않은<br>
                가게예요. ⓘ
            </p>
        </div>

        <!-- 가운데: 리뷰 -->
        <div class="recommend-reviews" id="reviewList"></div>

        <!-- 오른쪽: 태그 -->
        <div class="tag-box" id="tagList"></div>
    </div>

    <!-- 하단 버튼 3개 -->
    <div class="recommend-buttons">
        <button class="goStoreBtn" onclick="goStorePlace()">O 좋다! 가게 보러갈래</button>
        <button class="nextBtn" onclick="recommendNearby()">X 다른 가게 추천해줘</button>
        <button class="closeBtn" onclick="closeHelp()">추천 그만 닫을래</button>
    </div>



<!-- 2단계 : 메뉴 추천 모달 -->


<div class="menu-filter-buttons">
        <button class="typeBtn" data-type="1">한식</button>
        ...
    </div>


```

04. 핵심 기능 코드 설명

```
function openHelp() {
    document.getElementById('helpOverlay').style.display = 'block';
    document.getElementById('mascotModal').style.display = 'block';
    document.getElementById('mascotCharacter').style.display = 'block';
    document.getElementById('recommendPlace').style.display = 'none';
    document.getElementById('recommendMenu').style.display = 'none';
    document.getElementById('ByeMessage').style.display = 'none';
    document.getElementById('mainMascot').style.display = 'none';
}

// 가게 추천 데이터 불러오기 (완전 안정화버전)
function recommendNearby() {
    document.getElementById("recommendPlace").style.display = "block";
    document.getElementById("ByeMessage").style.display = "none";
    document.querySelector(".goStoreBtn").disabled = true;

    fetch("RandomStoreServlet")
        .then(response => response.json())
        .then(data => {
            console.log("받은 데이터:", data);

            // 가게이름 (* textContent 로 안정화)
            document.getElementById("storeName").textContent = data.storeName;

            // 별점 수치 (* 안정화)
            const avg = parseFloat(data.ratingAvg);
            document.getElementById("ratingAvg").textContent = "(" + avg.toFixed(1) + ")";

            // 별점 이모지
            let stars = "*****";
            if (avg <= 1.0) stars = "★ ★ ★ ★";
            else if (avg <= 1.5) stars = "★★ ★ ★ ★";
            else if (avg <= 2.5) stars = "★★★ ★ ★";
            else if (avg <= 3.5) stars = "★★★★ ★";
            else if (avg <= 4.5) stars = "★★★★ ★";
            document.getElementById("ratingStars").textContent = stars;

            // 필수
            document.getElementById("likeCount").textContent = data.likeCount;

            // * 리뷰 안정화 렌더링 (JS로 title 포함)
            const reviewList = document.getElementById("reviewList");
            reviewList.innerHTML = ""; // 초기화
            const reviewTitle = document.createElement("div");
            reviewTitle.className = "review-section-title";
            reviewTitle.textContent = "풀일이들의 리뷰";
            reviewList.appendChild(reviewTitle);

            const reviews = (data.reviews || []).filter(r => r.trim() !== "");
            if (reviews.length === 0) {
                const emptyDiv = document.createElement("div");
                emptyDiv.className = "review-box";
                emptyDiv.textContent = "아직 리뷰가 없습니다";
                reviewList.appendChild(emptyDiv);
            } else {
                reviews.forEach(review => {
                    const reviewBox = document.createElement("div");
                    reviewBox.className = "review-box";
                    const span = document.createElement("span");
                    span.textContent = review;
                    reviewBox.appendChild(span);
                });
            }
        });
}

function goStorePlace() {
    if (!currentStoreId || isNaN(currentStoreId)) {
        alert("가게 정보를 아직 불러오는 중입니다. 잠시만 기다려 주세요.");
        return;
    }

    const storeIdForRedirect = currentStoreId;

    document.getElementById("recommendPlace").style.display = "none";
    document.getElementById("ByeMessage").style.display = "block";
    document.getElementById("messageText").innerText = "내가 도움이 됐다니 기쁘다! 밥 맛있게 먹어!";

    setTimeout(function() {
        console.log("이동 직전 storeIdForRedirect:", storeIdForRedirect);
        window.location.href = "PlaceDetailServlet?storeId=" + currentStoreId;
    }, 1500);
}

function goStoreMenu() {
    document.getElementById("recommendMenu").style.display = "none";
    document.getElementById("ByeMessage").style.display = "block";
    document.getElementById("messageText").innerText = "내가 도움이 됐다니 기쁘다! 밥 맛있게 먹어!";

    setTimeout(function() {
        document.getElementById('helpOverlay').style.display = 'none';
        document.getElementById('ByeMessage').style.display = 'none';
        document.getElementById('mascotCharacter').style.display = 'none';
        document.getElementById('mainMascot').style.display = 'block';
    }, 1500);
}

const messagePressAgain = "<fmt:message key='mascot.message.pressagain' />";

function closeHelp() {
    document.getElementById("mascotModal").style.display = "none";
    document.getElementById("recommendPlace").style.display = "none";
    document.getElementById("recommendMenu").style.display = "none";
    document.getElementById("ByeMessage").style.display = "block";
    document.getElementById("messageText").innerText = messagePressAgain;

    setTimeout(function() {
        document.getElementById('helpOverlay').style.display = 'none';
        document.getElementById('ByeMessage').style.display = 'none';
        document.getElementById('mascotCharacter').style.display = 'none';
        document.getElementById('mainMascot').style.display = 'block';
    }, 1500);
}

function recommendMenu() {
    randomMenu();
    document.getElementById('mascotModal').style.display = 'none';
    document.getElementById('recommendMenu').style.display = 'block';

    document.querySelectorAll(".typeBtn").forEach(btn => {
        const type = parseInt(btn.dataset.type);
        if (selectedTypes.includes(type)) {
            btn.classList.add("active");
        } else {
            btn.classList.remove("active");
        }
    });
}
```

04. 핵심 기능 코드 설명

```
Project Explorer  
com.babpool  
  com.babpool.controller.user  
    ChangePasswordServlet.java  
    DeleteUserServlet.java  
    JoinServlet.java  
    LoginServlet.java  
    LogoutServlet.java  
    MainPageServlet.java  
    MyPageServlet.java  
    PlaceDetailServlet.java  
    RandomStoreServlet.java  
    ReviewDeleteServlet.java  
    SendEmailServlet.java  
    SubmitReviewServlet.java  
    ToggleBookmarkServlet.java  
    UploadProfileServlet.java  
  com.babpool.dao  
    BookmarkDAO.java  
    CategoryDAO.java  
    MarkerCategoryDAO.java  
    MarkerCategoryMapDAO.java  
    MarkerDAO.java  
    MarkerTagDAO.java  
    MarkerTagMapDAO.java  
    MenuDAO.java  
    RecommendDAO.java  
    ReviewDAO.java  
    ReviewImageDAO.java  
    StoreCategoryMapDAO.java  
    StoreDAO.java  
    StoreTagMapDAO.java  
    TagDAO.java  
    UserDAO.java  
  com.babpool.controller.user  
    ChangePasswordServlet.java  
    DeleteUserServlet.java  
    JoinServlet.java  
    LoginServlet.java  
    LogoutServlet.java  
    MainPageServlet.java  
    MyPageServlet.java  
    PlaceDetailServlet.java  
    RandomStoreServlet.java  
    ReviewDeleteServlet.java  
    SendEmailServlet.java  
    SubmitReviewServlet.java  
    ToggleBookmarkServlet.java  
    UploadProfileServlet.java
```

04. 핵심 기능 코드 설명

```
//  별점 통계 (리뷰 점수 그래프)
public Map<Integer, Integer> getScoreSummaryByStore(int storeId) {
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    Map<Integer, Integer> result = new HashMap<>();
    try {
        for (int i = 1; i <= 5; i++) result.put(i, 0);
        String sql = "SELECT FLOOR(rating) AS rating, COUNT(*) AS count FROM review WHERE store_id = ? GROUP BY FLOOR(rating)";
        pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, storeId);
        rs = pstmt.executeQuery();
        while (rs.next()) {
            result.putInt("rating", rs.getInt("rating"));
            result.put("count", rs.getInt("count"));
        }
    } catch (SQLException ex) {
        LogFileFilter.getWriter().println(ex);
    } finally {
        try { if (rs != null) rs.close(); } catch (SQLException ex) {
            LogFileFilter.getWriter().println(ex);
        }
    }
    return result;
}

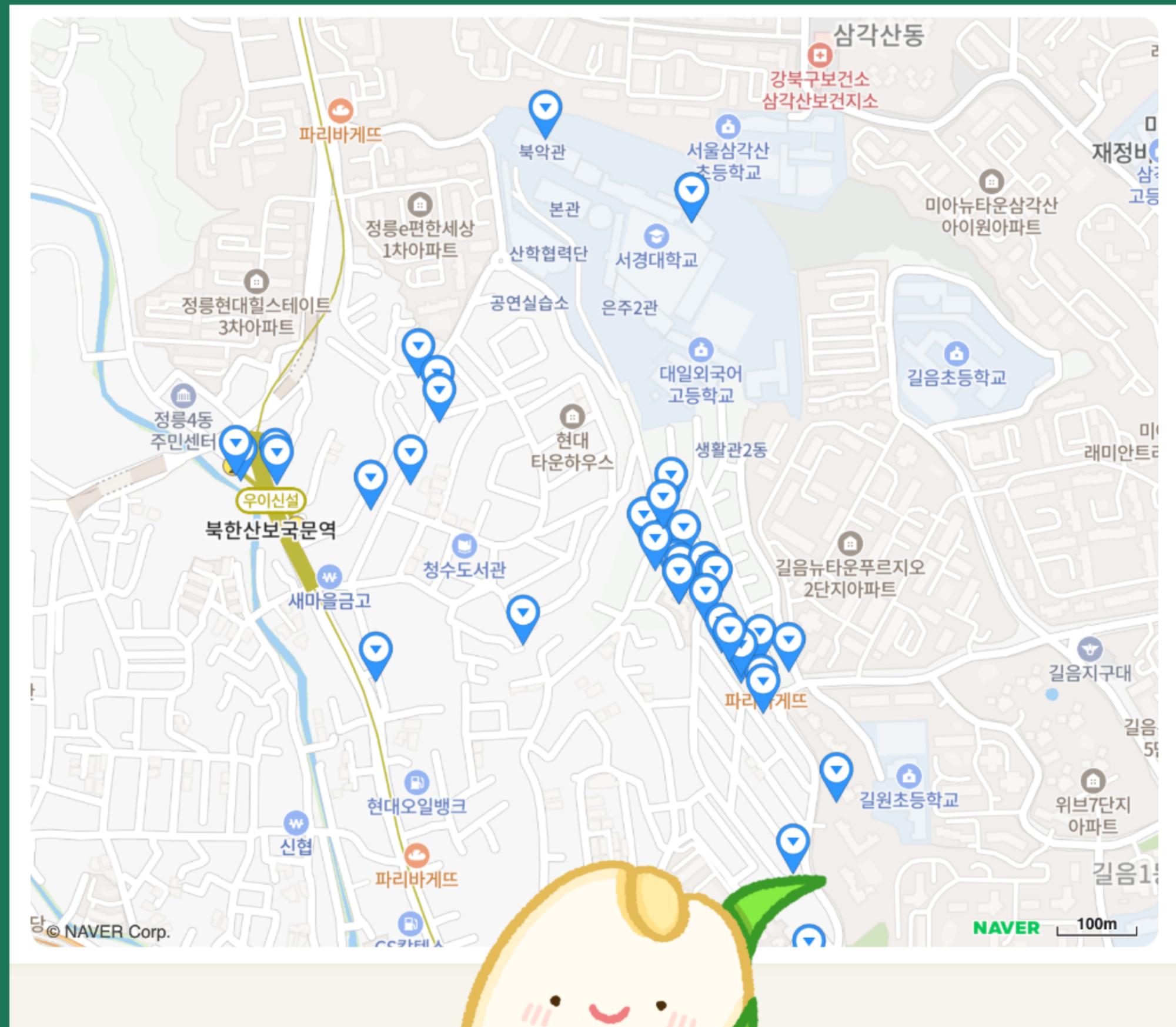
//  이메일 기준으로 회원 조회
public UserDTO getUserByEmail(String email) {
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    UserDTO user = null;
    try {
        String sql = "SELECT * FROM user WHERE email = ?";
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, email);
        rs = pstmt.executeQuery();
        if (rs.next()) {
            user = mapUserDTO(rs);
        }
    } catch (SQLException ex) {
        handleSQLException("[UserDAO] getUserByEmail() READ-Error", ex);
    } finally {
        close(rs, pstmt);
    }
    return user;
}

//  이메일 기준 삭제 (회원 탈퇴)
public boolean deleteUserByEmail(String email) {
    PreparedStatement pstmt = null;
    try {
        String sql = "DELETE FROM user WHERE email = ?";
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, email);
        return pstmt.executeUpdate() > 0;
    } catch (SQLException ex) {
        handleSQLException("[UserDAO] deleteUserByEmail() DELETE-Error", ex);
        return false;
    } finally {
        close(pstmt);
    }
}
```

05. 시행착오



05. 시행착오



05. 시행착오

마커 정보

마커 전체 조회

마커 전체 조회

마커 선택 조회

마커 ID 입력

마커 선택 조회

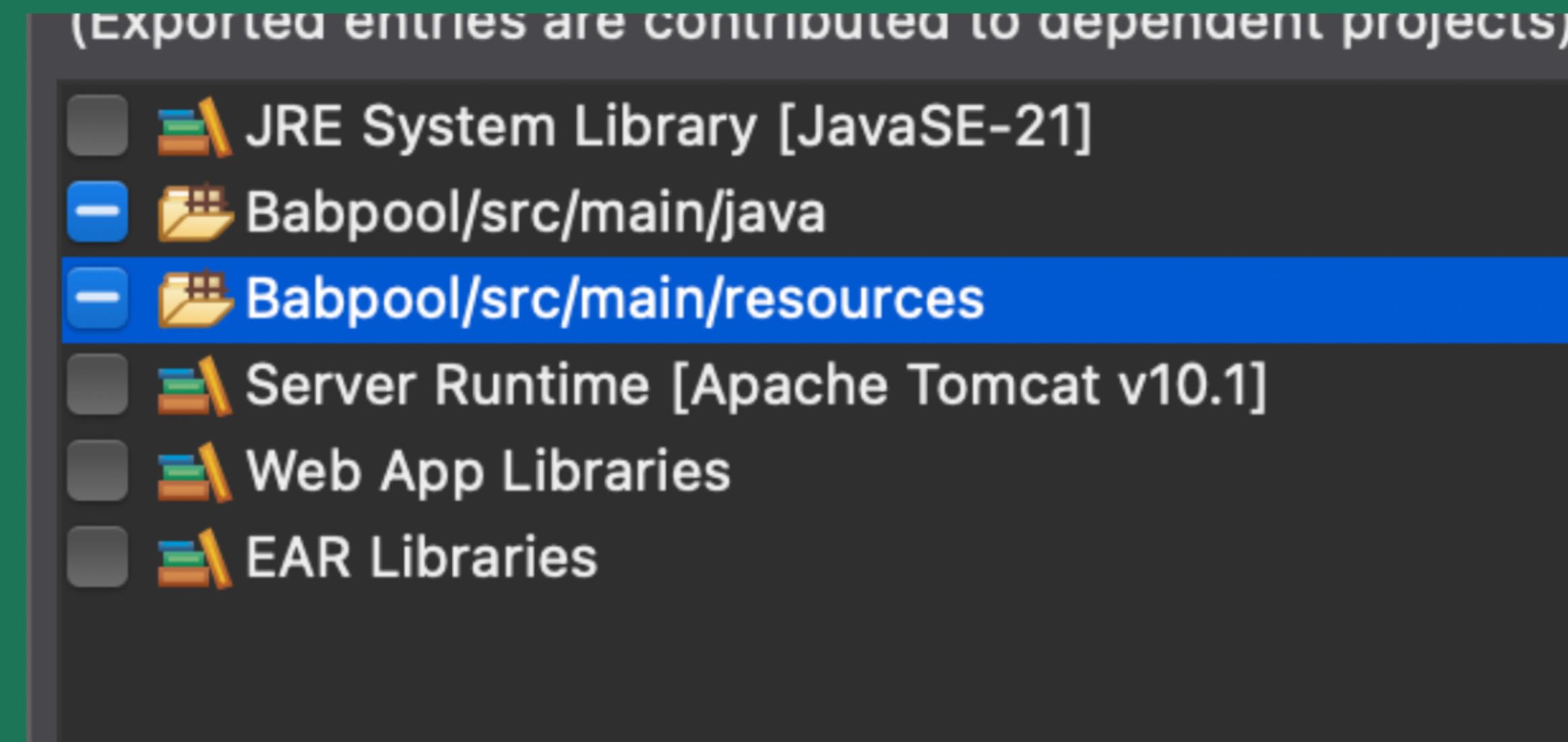
마커 등록

가게 ID

네이버 장소 ID

마커 등록

05. 실행작오



05. 시행착오

여기 어때?

구름파스타 ★★★★☆ (0) ❤ 5



풀잎이들의 리뷰

#

○ 좋다! 가게 보러갈래

X 다른 가게 추천해줘

추천 그만 받을래

여기 어때?

구름파스타 ★★★★☆ (4.1) ❤ 6



풀잎이들의 리뷰

음

ㅜㅜ

맛!

#solo

○ 좋다! 가게 보러갈래

X 다른 가게 추천해줘

추천 그만 받을래

06. 팀원의 참여비율

수연

주 담당 프론트
프로젝트 기획
전체 페이지 구상 및 디자인, 프론트 작업
 웰컴 페이지(다국어)
 메인 페이지
(세션, 헤더푸터, 카테고리, top3, 밍풀이)
 가게 페이지
 리뷰 작성 페이지
 에러페이지
 Mvc 분리
 가게 데이터추출
 dto/dao 확장
Sql 쿼리문 및 트리거 작업
 리팩토링 작업
Ppt, 영상 제작 및 발표

50%

동국

주 담당 백엔드
ERD 제작
메인 페이지
(지도 api, 지도-카테고리 매핑)
로그인 페이지(자동로그인-쿠키)
회원가입 페이지(이메일인증, 유효성)
 마이페이지
 비번변경 페이지
 관리자페이지(보안)
 Web.xml 연결
 로그 기록
 에러페이지
 Mvc분리
 가게 데이터추출
 dto/dao 초기 작업
Sql 쿼리문 작업

50%



Thank you !

웹서버 프로그래밍 기말 프로젝트

2020303002 김동국
2022301032 박수연