

이미지 기반 블로그 플랫폼 서비스 'Daily'



| | |
|-----|------------|
| 과목명 | 웹프로그래밍 |
| 분반 | 1 분반 |
| 제출일 | 2021.06.08 |
| 학부 | 컴퓨터공학부 |
| 학번 | 2019136034 |
| 이름 | 김은서 |

'Daily' 바로 가기 ⇨ <http://dailyme.kro.kr/>

목차

| | |
|---------------------------|----|
| 1부 웹사이트 ‘Daily’ 소개 | 3 |
| 1. 웹사이트 기획의도 | 4 |
| 2. ‘Daily’ 주요 기능 소개 | 5 |
| 2부 ‘Daily’ 구현 과정 | 17 |
| 1. DB 및 전체 구조 설계 | 18 |
| 2. 기능별 구현 방법 설명 | 20 |

1 부

웹사이트 ‘Daily’ 소개

01 | 웹사이트 기획의도

02 | ‘Daily’ 주요 기능 소개

1. 웹사이트 기획의도

제가 제작한 웹사이트는 이미지 기반 블로그 플랫폼 서비스인 'Daily'입니다.



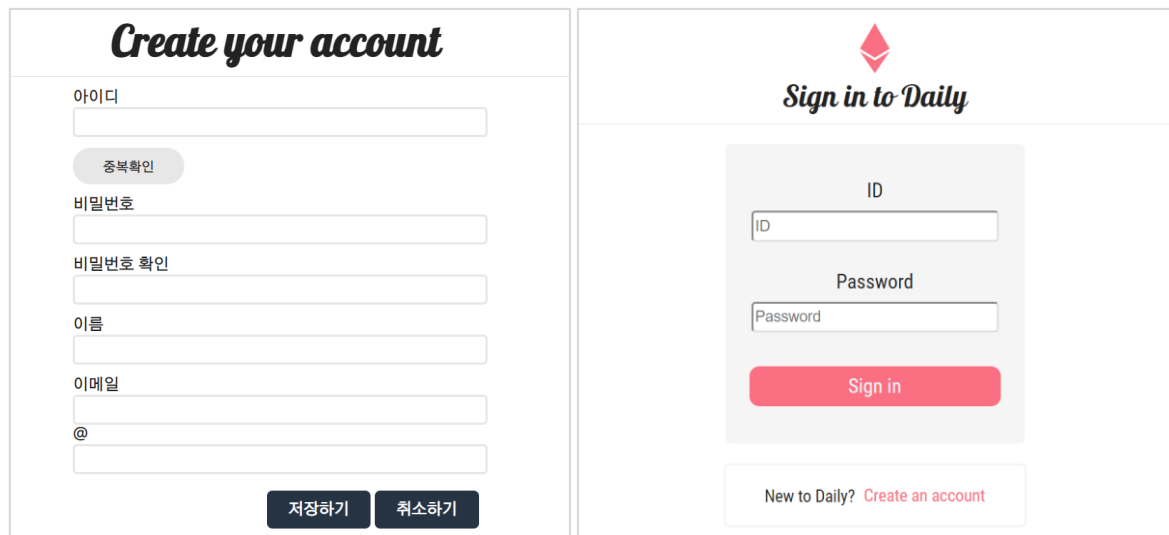
- 이미지 기반 블로그 플랫폼 'Daily'의 로고 사진

대표적인 이미지 기반 SNS 플랫폼인 'instagram'과 기존의 다양한 블로그 플랫폼으로부터 영감을 받아, 이미지 중심으로 소통하는 블로그 플랫폼을 만들어보면 좋겠다고 생각하여 제작하게 된 것이 'Daily'입니다. 이미지 기반 플랫폼인만큼, 블로그의 주 용도는 일상 공유, 사진 기반 정보(맛집, 제품 리뷰 등) 공유 등에 적합하다고 생각했습니다.

2. 'Daily' 주요 기능 소개

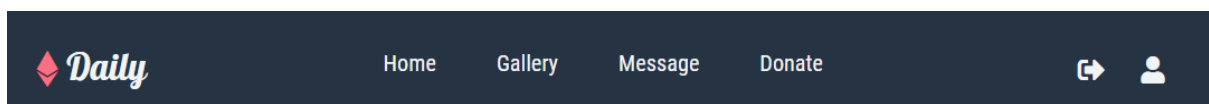
이미지 기반 블로그 플랫폼 서비스 'Daily'를 기획하면서 작성한 웹사이트의 기능적인 요구사항들과 이를 실제 구현한 화면은 다음과 같습니다.

1. 회원가입, 로그인, 로그아웃, 마이페이지(회원정보수정) 기능



- (좌) 회원 가입 (우) 로그인 기능

'Daily'는 개인이 스스로 자신의 공간을 관리할 수 있는 블로그 플랫폼이므로, 회원가입과 로그인 기능이 필수적으로 구현되어야 합니다. 회원가입 시, 다른 회원과 ID가 겹치지 않도록 아이디 중복 확인을 할 수 있는 기능을 추가적으로 구현하였습니다.



- 오른쪽에서 2번째 아이콘이 '로그아웃' 아이콘입니다.

로그아웃은 메뉴바의 '로그아웃' 아이콘을 눌러 로그아웃 할 수 있도록 하였습니다.

회원 정보 수정

아이디
eunseo

비밀번호

....

비밀번호 확인

....

이름

김은서

이메일

eunseo

@

kakao.com

수정하기

취소하기

- 마이페이지(회원 정보 수정) 화면

또한 이렇게 마이페이지 기능을 구현하여 회원 아이디는 유지하고, 회원의 비밀번호, 이름, 이메일을 수정할 수 있도록 구현했습니다.

2. 이미지 기반 글쓰기 기능

제목을 입력하세요

파일 선택

선택된 파일 없음

당신의 이야기를 적어보세요...

목록보기

출간하기

- 글쓰기 화면. 파일을 선택하여 이미지(혹은 다른 종류) 파일을 첨부할 수 있습니다.

동탄 스시풍

김은서 · 2021-05-30 (17:35)



맛있네요 별점 5개 드립니다.

동탄스시풍.jpg [저장]

목록보기

글쓰기

수정

삭제

- 이미지 파일을 첨부하게 되면 다음과 같이 저장 후 실제 이미지를 글과 함께 볼 수 있습니다.

‘Daily’는 이미지 기반 블로그 플랫폼이므로, ‘글쓰기’ 기능에서 필수적으로 파일을 첨부할 수 있어야 합니다. 해당 파일은 어떤 파일이든지(img, pdf, ppt, mp4 등) 업로드 가능하지만, img 파일을 업로드 할 경우에는 게시물을 저장했을 때 실제 이미지를 글과 함께 보여줄 수 있어야 합니다.

3. 글 수정 및 삭제 기능

의왕 레일바이크

📁 의왕레일바이크.jpg

의왕 레일바이크를 탔다.
힘들었다

🔗 목록보기

수정하기

- 본인이 작성한 글의 텍스트 부분을 수정할 수 있습니다.

마들렌

김은혜 · 2021-06-03 (17:14)



제가 만든 마들렌 자랑해요~

📁 20210602_195529.jpg [\[저장\]](#)

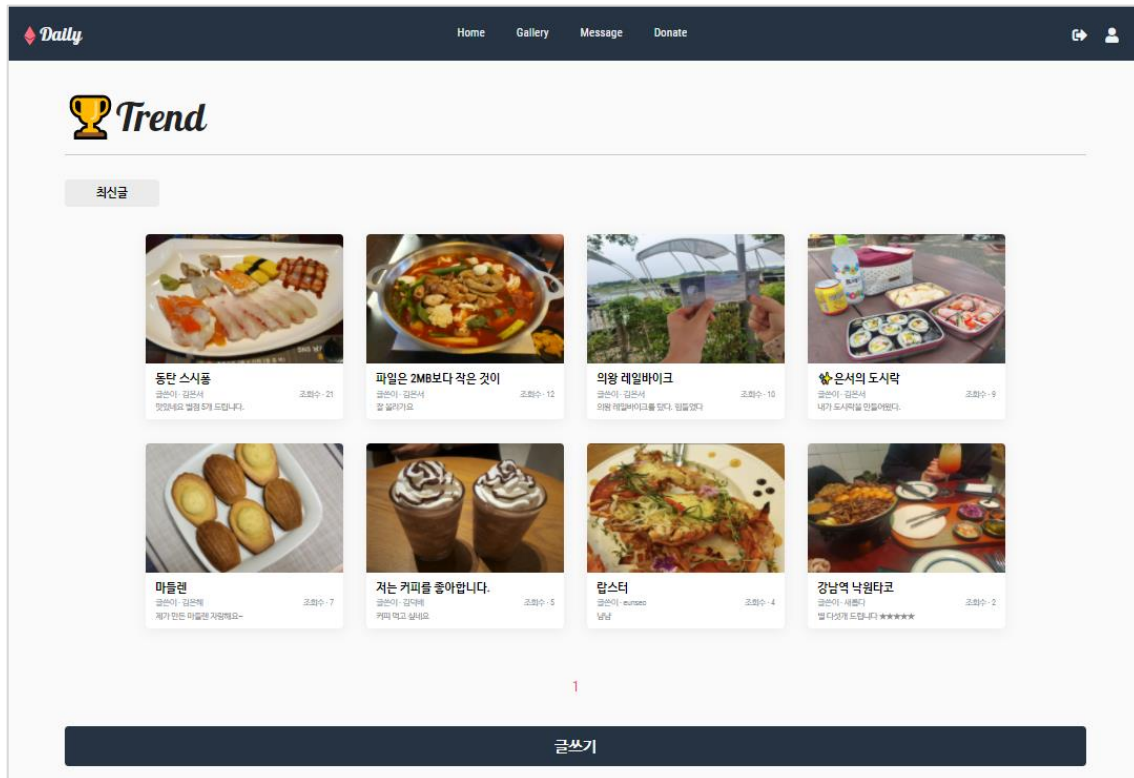
🔗 목록보기

글쓰기

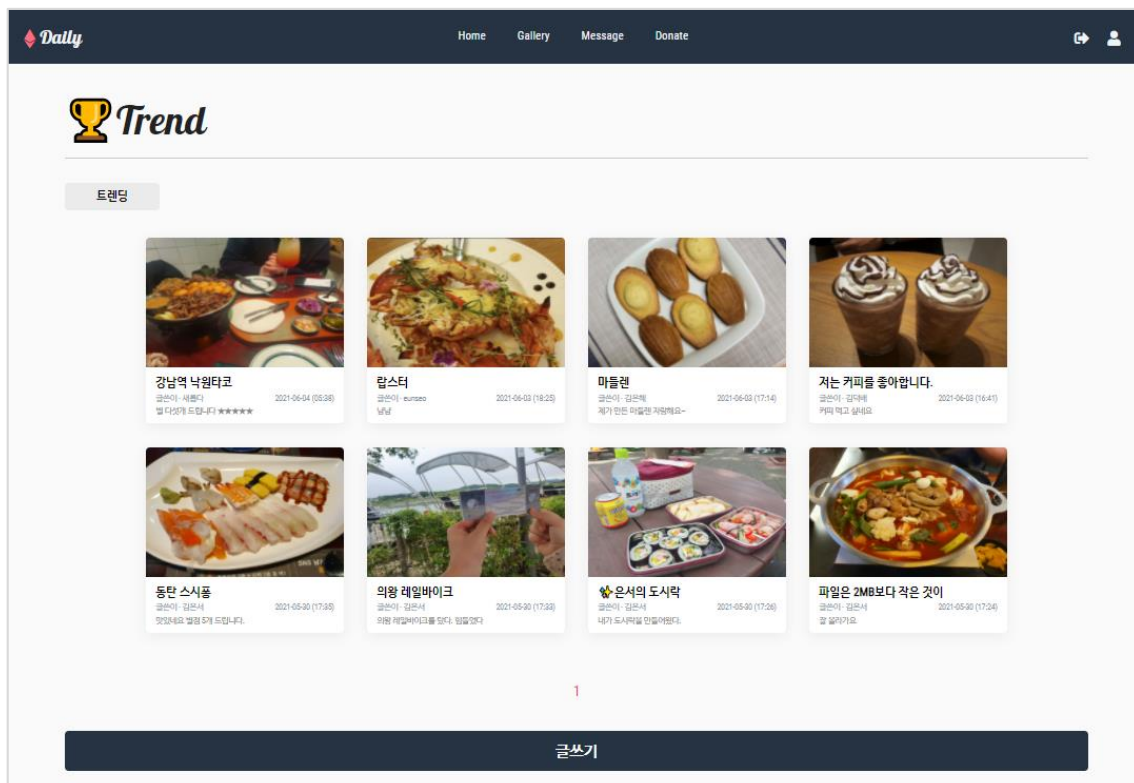
- 다른 사용자가 작성한 글을 누르면 '수정', '삭제' 버튼이 보이지 않습니다.

사용자는 본인이 작성한 글에 대하여 게시물 수정 및 삭제를 할 수 있어야 합니다. 따라서 본인이 작성한 게시물에 한해서만 '수정', '삭제' 버튼이 보이도록 해야 합니다. 수정은 제목, 내용과 같은 텍스트 수정만 가능하도록 하였고, 삭제 버튼을 누르면 바로 게시물이 삭제되도록 했습니다.

4. 전체 사용자 글 보기 기능



- '조회수가 많은 순서'로 정렬된 메인 화면





- '최신순'으로 정렬된 메인 화면

‘Daily’는 개방형 서비스입니다. 따라서 자신이 작성한 글을 다른 어떤 사용자들도 볼 수 있어야 합니다. 이를 가능하도록 하기 위해 ‘Daily’의 ‘Home’ 화면은 전체 사용자의 글을 모두 볼 수 있는 공간으로 설정했습니다.

추가적으로, ‘Home’ 화면은 기본적으로 ‘조회수가 많은 순서’로 전체 사용자의 게시물이 내림차순 정렬됩니다. 만약 ‘최신순’ 버튼을 누르게 되면 ‘최신 게시물 순서’로 전체 사용자의 게시물이 정렬된 화면을 볼 수 있습니다. 마찬가지로 ‘트렌딩’ 버튼을 다시 누르게 되면 ‘조회수가 많은 순서’로 정렬된 화면을 다시 볼 수 있습니다.

5. 조회수 기능 (새로고침을 할 때마다 조회수가 무한정 증가하는 문제 해결)


| | |
|--|---|
|  |  |
| <p>동탄 스시풍 글쓴이 · 김은서 조회수 · 23 맛있네요 별점 5개 드립니다.</p> | <p>파일은 2MB보다 작은 것이 글쓴이 · 김은서 조회수 · 13 잘 올라가요</p> |

- 게시물 별로 조회수를 확인할 수 있습니다.

조회수 는 게시물을 클릭할 때 1씩 증가하되, 조회할 때마다(새로고침) 조회수가 1씩 무한정 증가하는 문제를 방지하기 위해 사용자마다 마지막 방문 이후 최소 24시간이 지나야 다시 조회수가 측정되도록 구현했습니다.

6. 내가 쓴 글 확인하기 기능


Gallery



랍스터

남남

작성자 · eunseo
등록일 · 2021-06-03 (18:25) 조회수 · 4



동탄 스시풍

맛있네요 별점 5개 드립니다.

작성자 · 김은서
등록일 · 2021-05-30 (17:35) 조회수 · 21

- 내가 쓴 글만 확인할 수 있는 'Gallery' 게시판

'블로그'의 특성에 맞게, 본인이 작성한 글만 모아서 확인할 수 있어야 합니다. 따라서 'gallery'라는 탭을 따로 마련하여 본인이 작성한 글만 모아서 확인할 수 있는 게시판을 만들었습니다.

Gallery



이미지 사진이 없는 경우

이렇게 기본 사진이 들어갑니다.

작성자 · eunseo

등록일 · 2021-06-08 (06:36)

조회수 · 0

- 이미지 사진이 없는 경우의 미리보기 화면

추가적으로 만약 게시물에 이미지 파일이 존재하지 않는 경우라 하더라도, 게시판 목록 화면에서는 디자인의 일관성을 위해 기본 사진을 대신 출력하여 보여주었습니다.

7. 사용자 간 쪽지 보내기 기능(받은 쪽지함, 보낸 쪽지함)



Send a Message

받은 쪽지함

보낸 쪽지함

보낸 사람

eunseo

받는 사람

제목

내용

보내기

- 쪽지 보내기 화면

받은 쪽지함

보낸 쪽지함

쪽지 보내기

 받은 쪽지함

| 번호 | 제목 | 보낸 사람 | 날짜 |
|----|---------------------|------------------|--------------------|
| 2 | 안녕하세요 친하게 지내고 싶습니다. | hehe(hehe) | 2021-06-03 (16:30) |
| 1 | 엠평에 중국당면 추가 | 살생냥이 (eunjin) | 2021-05-30 (20:47) |

1

- (받은)쪽지함 화면

블로그에 가입한 사용자들간에 서로 소통할 수 있는 방법이 있으면 좋겠다고 생각하여 '쪽지 보내기' 기능을 추가하였습니다. 받은 쪽지함, 보낸 쪽지함을 통해 서로 주고 받은 쪽지를 확인할 수 있습니다.

8. 쪽지 답장하기, 삭제하기 기능

 받은 쪽지함

제목 : 안녕하세요 친하게 지내고 싶습니다.

hehe / 2021-06-03 (16:30)

반갑습니다.

보낸 쪽지함

받은 쪽지함

답장하기

삭제하기

- 우측 하단에 '답장하기'를 통해 답신을 보낼 수 있습니다.

 답장 보내기

보낸 사람 | eunseo

받는 사람 | hehe(hehe)

제목 | RE: 안녕하세요 친하게 지내고 싶습니다.

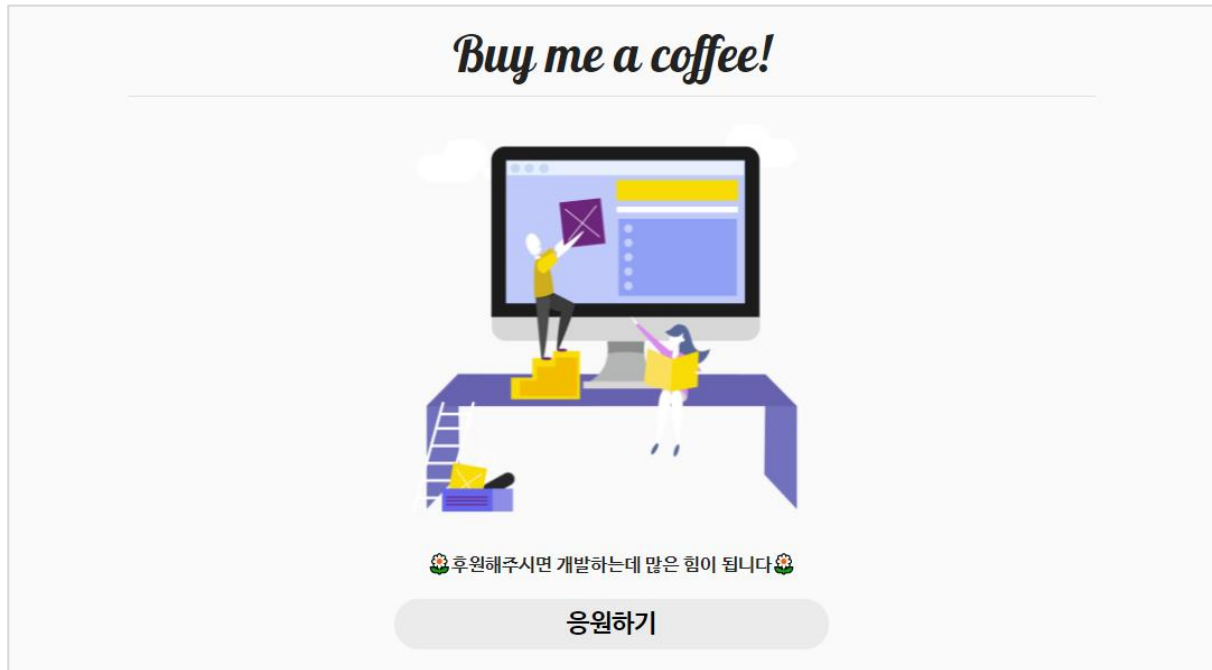
내용 |
----- Original Message -----
반갑습니다.

보내기

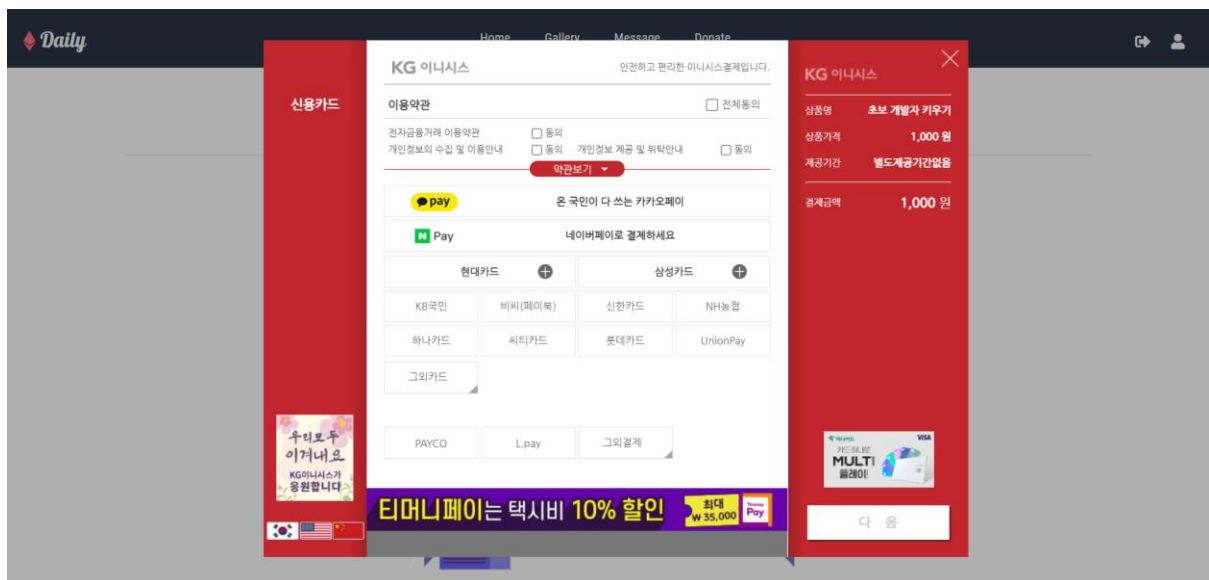
- 답신 메시지 기본 양식

받은 쪽지에 답신을 보내는 기능도 추가했습니다. '답장하기'를 누르면 이전에 주고 받은 쪽지 내용이 'Original Message'로 표시되며, 쪽지 제목도 기존의 제목 앞에 'RE:'가 표시되어 해당 쪽지가 답신임을 표시할 수 있도록 하였습니다. 또한 '삭제하기' 버튼을 통해 쪽지를 삭제할 수 있습니다.

9. 웹사이트 개발자 후원하기 기능



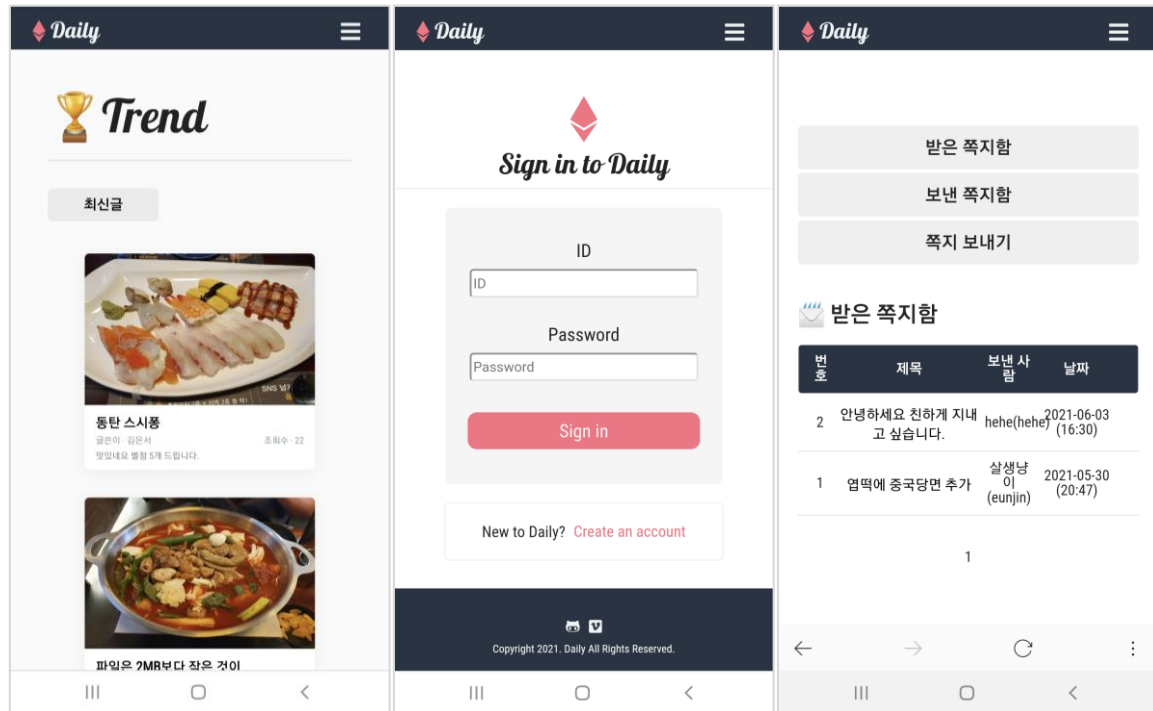
- 웹사이트 개발자에게 후원하는 'donate' 게시판 화면



- '응원하기'를 누르면 실제 결제창이 뜨고 결제를 진행할 수 있습니다.

마지막으로 웹사이트 개발자를 후원할 수 있는 'Donate' 게시판을 추가적으로 만들어보았습니다. 웹 서비스를 제공한 개발자에게 커피 한 잔 정도의 값을 후원할 수 있도록 추가한 기능입니다. 실제로 '응원하기' 버튼을 누르면 결제창이 뜨고 1000원을 결제할 수 있도록 구현했습니다.

10. 반응형 웹사이트 기능



- 모바일에서 웹사이트를 들어갔을 때의 화면입니다.

모바일에서 잘 동작할 수 있도록 화면이 767px 이하가 되는 경우를 고려하여 반응형 웹을 구현하였습니다. 따라서 모바일에서도 위와 같이 화면 깨짐 없이 예쁘게 동작함을 확인할 수 있습니다

2 부

‘Daily’ 구현 과정

01 | DB 및 전체 구조 설계

02 | 기능별 구현 방법 설명



1. DB 및 전체 구조 설계

DB 테이블은 members, message, board로 나누어 설계했습니다.

(1) members - 회원 정보 저장 DB 테이블

| 필드명 | 데이터형 | null | 설명 |
|------------|----------|----------|------|
| num | int(11) | not null | 일련번호 |
| id | char(15) | not null | 아이디 |
| pass | char(15) | not null | 비밀번호 |
| name | char(10) | not null | 이름 |
| email | char(80) | | 이메일 |
| regist_day | char(20) | | 가입일 |

(2) message - 쪽지 DB 테이블

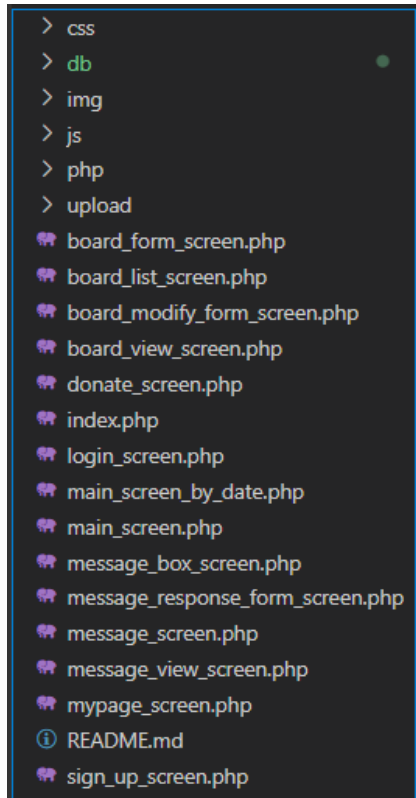
| 필드명 | 데이터형 | null | 설명 |
|------------|-----------|----------|-------|
| num | int(11) | not null | 일련번호 |
| send_id | char(20) | not null | 보낸 사람 |
| rv_id | char(20) | not null | 받는 사람 |
| subject | char(200) | not null | 제목 |
| content | text | not null | 내용 |
| regist_day | char(20) | | 보낸 날짜 |

(3) board - 'Gallery' 게시판 DB 테이블

| 필드명 | 데이터형 | null | 설명 |
|------------|-----------|----------|---------|
| num | int(11) | not null | 일련번호 |
| id | char(15) | not null | 글쓴이 아이디 |
| name | char(10) | not null | 글쓴이 이름 |
| subject | char(200) | not null | 글 제목 |
| content | text | not null | 글 내용 |
| regist_day | char(20) | not null | 작성 일시 |
| hit | int(11) | not null | 조회수 |

| | | | |
|-------------|----------|----------|---------|
| file_name | char(40) | | 파일명 |
| file_type | char(40) | | 파일 형식 |
| file_copied | char(40) | | 저장된 파일명 |
| like | int(11) | not null | 좋아요 개수 |

다음으로 웹사이트 코드의 전체 구조는 다음과 같이 설계하였습니다.



코드는 CSS, Javascript, database, image, upload(업로드 된 파일을 저장하는 폴더), php 폴더와 나머지 php 파일로 구성하였습니다.

이때 전체 php 파일 중 일부를 'php 폴더 내'로 분리한 이유는 다음과 같습니다.

- 먼저 php 폴더 안에 있는 php 코드들은 화면에 실제로 보이지는 않지만 어떠한 기능이나 동작을 처리하는 파일입니다. (예를 들어 [로그인/로그아웃 처리하기], [마이페이지 정보 수정하기], [메시지 전송하기] 등의 기능을 말합니다. 이는 실제로 화면에 보이지 않지만 기능적으로 처리되는 부분입니다.)
- 나머지 php 파일은 자세히 보면 파일명 끝에 대부분 'screen'이 붙습니다. 즉, 'php 폴더 밖'의 php 파일들은 실제로 화면에 보이는 부분들입니다. (예를 들어 [로그인 창], [메시지 보내기 화면], [메인 게시판 화면] 등을 말합니다.)

2. 기능별 구현 방법 설명

지금부터는 1부의 '02. 'Daily' 주요 기능 소개'에서 소개한 각각의 기능들을 어떻게 구현하였는지 코드와 함께 설명 드리겠습니다.

1. 회원가입(아이디 중복 확인 기능), 로그인, 로그아웃

```
$con = mysqli_connect("localhost", "root", "s6139350!", "diary");
$sql = "select * from members where id='$id'";
$result = mysqli_query($con, $sql);
$num_record = mysqli_num_rows($result);
if ($num_record)
{
    echo "<li>".$id." 아이디는 중복됩니다.</li>";
    echo "<li>다른 아이디를 사용해 주세요.</li>";
}
else
{
    echo "<li>".$id." 아이디는 사용 가능합니다.</li>";
}
mysqli_close($con);
```

- members DB에서 동일한 아이디가 있는지 검사합니다.

사용자가 아이디를 입력하면 members 테이블에 동일한 아이디가 있는지 검사하고, 만약에 동일한 아이디가 존재하면 \$num_record에 해당 row값이 저장되므로, 아이디 중복 여부를 판단할 수 있습니다.

```
else
{
    session_start();
    $_SESSION["userid"] = $row["id"];
    $_SESSION["username"] = $row["name"];
    echo("
    <script>
        location.href = '../main_screen.php';
    </script>
    ");
}
```

- 로그인 성공 시 세션 변수에 id, name 값 저장

```
<!-- 로그인 상태 판단 & nav -->
<?php
    session_start();
    if (isset($_SESSION["userid"])) $userid = $_SESSION["userid"];
    else $userid = "";
    if (isset($_SESSION["username"])) $username = $_SESSION["username"];
    else $username = "";
?>
```

- 로그인 상태 판단

로그인 기능은, 로그인 페이지에서 사용자가 입력한 아이디와 비밀번호를 POST로 입력 받아

DB에 저장된 id, pass와 비교하여 로그인할 수 있도록 하였습니다. 이때, 로그인 상태를 판단하고 유지하도록 하기 위해서는 SESSION을 사용했습니다. 로그인을 완료하게 되면 사용자의 id값으로 세션 변수인 \$_SESSION["userid"]를 생성합니다. 따라서 \$_SESSION["userid"]가 존재하면 \$userid = \$_SESSION["userid"];를 통해 현재 로그인 된 사용자 정보를 식별하고 로그인 상태를 유지할 수 있습니다.

```
login.php
<?php
    session_start();
    unset($_SESSION["userid"]);
    unset($_SESSION["username"]);

    echo("
    <script>
        location.href = '../main_screen.php';
    </script>
    ");
?>
```

- 로그아웃 처리 부분

마찬가지로 로그아웃 처리 기능도 unset(\$_SESSION["userid"])를 통해 세션 변수를 삭제하고, 따라서 \$userid값이 null이 되면서 로그아웃 되었음을 확인할 수 있습니다.

2. 이미지 기반 글쓰기 기능

```
// 첨부 파일이 이미지인 경우, 화면에 해당 이미지를 띄운다
if ($file_type == "image/jpeg" || $file_type == "image/png" || $file_type == "image/jpg" || $file_type == "image/gif") {
    $image_file_image = "<img src='upload/{$_file_copied}' class='image_file'>";
}
else {
    $image_file_image = NULL;
}
```

- 첨부된 파일이 이미지인지 확인하는 코드입니다.

‘Daily’는 글쓰기 기능에서 이미지 파일을 첨부하게 되면 저장 이후 글과 함께 실제 이미지를 확인할 수 있습니다. 이를 구현하기 위해 업로드 된 파일의 타입이 이미지 파일인지 확인하고, 이미지 파일이라면 \$image_file_image 변수에 해당 파일의 저장 경로를 값으로 넣어주었습니다. 반면 이미지 파일이 아니라면 Null 값을 넣어주었습니다.

```
<li>
    <div class="row3">
        <div class="image_file"><?php echo $image_file_image; ?></div>
        <div class="content"><?=$content?></div>
    </div>
</li>
```

- \$image_file_image이 NULL이 아니라면 이미지 사진을 화면에 보여줍니다.

따라서 저장 이후, 게시물을 띄울 때 \$image_file_image를 화면에 출력해주었는데, 만약 해당 값이 Null이라면 화면에 아무것도 보이지 않게 될 것이고, 값에 이미지 파일 저장 경로가 저장되어 있다면 글과 함께 실제 이미지 사진을 볼 수 있게 됩니다.

3. 전체 사용자 글 보기 기능

전체 사용자의 글을 보는 기준은 '조회수가 많은 순서' 또는 '최신 글 순서' 입니다. 이를 구현하기 위해 각각 서로 다른 정렬 방법으로 DB에서 게시물을 가져왔습니다.

```
$con = mysqli_connect("localhost", "root", "s6139350!", "diary");
$sql = "select * from board where id='$userid' order by num desc";
$result = mysqli_query($con, $sql);
$total_record = mysqli_num_rows($result); // 전체 글 수
```

- 최신 글 순서대로 board DB에서 게시물을 가져오는 과정

먼저 '최신 글 순서'를 위해 \$sql = "select * from board where id='\$userid' order by num desc"; 을 통해 num(일련번호, 작성된 글 순서)를 내림차순으로 정렬한 뒤 해당 board DB의 정보를 가져왔습니다.

```
$con = mysqli_connect("localhost", "root", "s6139350!", "diary");
$sql = "select * from board order by hit desc"; // 조회수 많은 순서
$result = mysqli_query($con, $sql);
$total_record = mysqli_num_rows($result); // 전체 글 수
```

- 조회수가 많은 순서대로 board DB에서 게시물을 가져오는 과정

마찬가지로 '조회수가 많은 순서'를 위해서는 \$sql = "select * from board order by hit desc"; 을 통해 hit(조회수)을 내림차순으로 정렬한 뒤 board DB의 정보를 가져왔습니다.

4. 조회수 기능 (Cookie로 조회수 무한 증가 문제 방지)

```
// 방문자 조회수 쿠키로 판별 (마지막 접속 이후 24시간이 지나야 조회수 1 증가)
$hit_cookie_name = $num."_hit";
if(!isset($_COOKIE[$hit_cookie_name])) { // 쿠키가 없으면
    $new_hit = $hit + 1; // 조회수 1 증가
    $sql = "update board set hit=$new_hit where num=$num";
    mysqli_query($con, $sql);
}
setcookie("$hit_cookie_name", true, time() + (60*60*24)); // 매번 클릭 때마다 쿠키 시간 다시 초기화
```

- Cookie로 방문한 사용자를 기억해 조회수를 증가시키는 부분

조회수 기능을 구현할 때, 새로고침을 할 때마다 조회수가 무한정 증가하는 문제를 해결하기 위해 Cookie를 이용했습니다. 쿠키를 통해 해당 게시판 번호(num)과 관련된 쿠키를 생성해 저장해 놓음으로써 게시물 별 방문한 사용자를 판별하였습니다.

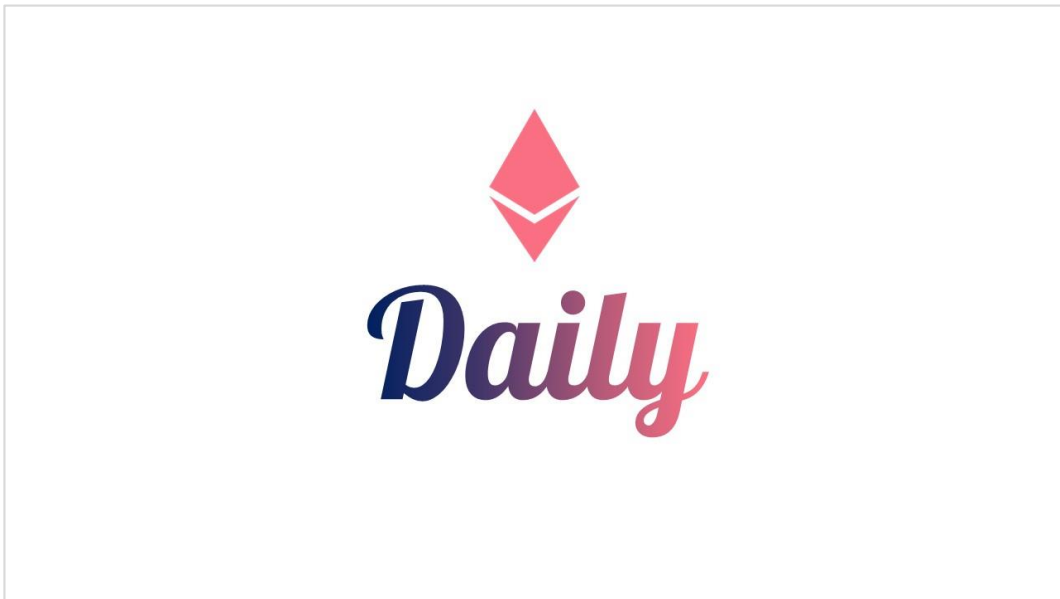
쿠키는 최대 24시간동안 유지되도록 설정함으로써 각 사용자마다 마지막 접속 이후 24시간이 지나야 다시 조회수가 1 증가되도록 설정하였습니다.

5. 내가 쓴 글 확인하기 기능(게시판 목록에서 이미지 미리보기 기능)

내가 쓴 글을 확인할 수 있는 'Gallery' 게시판을 구현할 때에는, 직접 게시물을 누르지 않아도 이미지 사진을 게시판 목록에서부터 보여주면 좋을 것 같다고 생각했습니다.

```
// 이미지 처리
if ($file_type == "image/jpeg" || $file_type == "image/png") {
    $image_file_image = "<img src='upload/{$_file_copied}' class='image_file'>";
}
else {
    $image_file_image = "<img src='img/Daily.jpg' class='image_file'>";
}
```

- 게시판 목록에서의 이미지 미리보기 처리하는 과정



- (게시물에 이미지 파일이 없을 때 대신 출력하는) 기본 사진

따라서 게시판 목록에서 이미지 미리보기 기능을 구현하기 위해 위의 코드와 같이 '게시물에 이미지 파일이 존재하는 경우'에는 해당 이미지 파일을 가져와 미리보기 화면에 출력해주었고,

‘게시물에 이미지 파일이 존재하지 않는 경우’에는 게시판 디자인의 일관성을 위해 미리 제작해 둔 기본 사진(img/Daily.jpg)을 출력하여 보여주었습니다.

6. 쪽지 답장하기 기능

```
<?php
$num = $_GET["num"];

$con = mysqli_connect("localhost", "root", "s6139350!", "diary");
$sql = "select * from message where num=$num";
$result = mysqli_query($con, $sql);

$row = mysqli_fetch_array($result);
$send_id = $row["send_id"];
$rv_id = $row["rv_id"];
$subject = $row["subject"];
$content = $row["content"];

$subject = "RE: ".$subject;

$content = str_replace("\n", "\n", $content);
$content = "\n\n\n----- Original Message ----- \n".$content;

$result2 = mysqli_query($con, "select name from members where id='$send_id'");
$record = mysqli_fetch_array($result2);
$send_name = $record["name"];
?>
```

- 이전 쪽지 내용을 가져와서 답신으로 처리하는 부분

쪽지 답신 기능의 구현을 위해서 기존의 쪽지 제목, 내용을 DB로부터 가져와 각각 \$subject, \$content에 저장한 뒤, html코드 내에서 각각 제목, 내용의 기본값으로 작성해서 화면에 출력해주었습니다.

```
<li>
  <span class="col1">제목</span>
  <span class="col2"><input name="subject" type="text" value="<?=$subject?>"></span>
</li>
<li id="text_area">
  <span class="col1"><span class="col1_textarea">내용</span></span>
  <span class="col2">
    <textarea name="content"><?=$content?></textarea>
  </span>
</li>
```

- 쪽지 답신 작성 시 초기값을 설정하는 부분

제목의 기본값으로 \$subject를 띄우기 위해서는 input 태그의 value 속성을 사용하여 구현하였고, 내용의 기본값으로 \$content를 띄우기 위해서는 textarea의 태그 사이에 \$content를 넣어줌으로써 초기값을 설정할 수 있었습니다.

7. 후원하기 기능

후원하기 기능에서는 결제 시스템 API인 아임포트(<https://www.iamport.kr/>)를 활용하여 결제 모듈을 생성했습니다. 아임포트 라이브러리를 코드 내에 삽입하고 결제창 호출 코드를 추가함으로써 구현할 수 있었습니다.

```
<script>
function donate() {
    var IMP = window.IMP; // 선택가능
    IMP.init('imp97593941'); // 'iamport' 대신 부여받은 "가맹점 식별코드"를 사용

    IMP.request_pay({
        pg : 'inicis', // version 1.1.0부터 지원.
        pay_method : 'card',
        merchant_uid : 'merchant_' + new Date().getTime(),
        name : '초보 개발자 키우기',
        amount : 1000,
        buyer_email : '',
        buyer_name : '',
        buyer_tel : '',
        buyer_addr : '',
        buyer_postcode : '',
    }, function(rsp) {
        if ( rsp.success ) {
            var msg = '결제가 완료되었습니다.';
            msg += '결제 금액 : ' + rsp.paid_amount;
        } else {
            var msg = '결제에 실패하였습니다.';
            msg += '에러내용 : ' + rsp.error_msg;
        }
        alert(msg);
    });
}
</script>
```

- 결제창 호출 코드

8. 반응형 웹사이트 구현

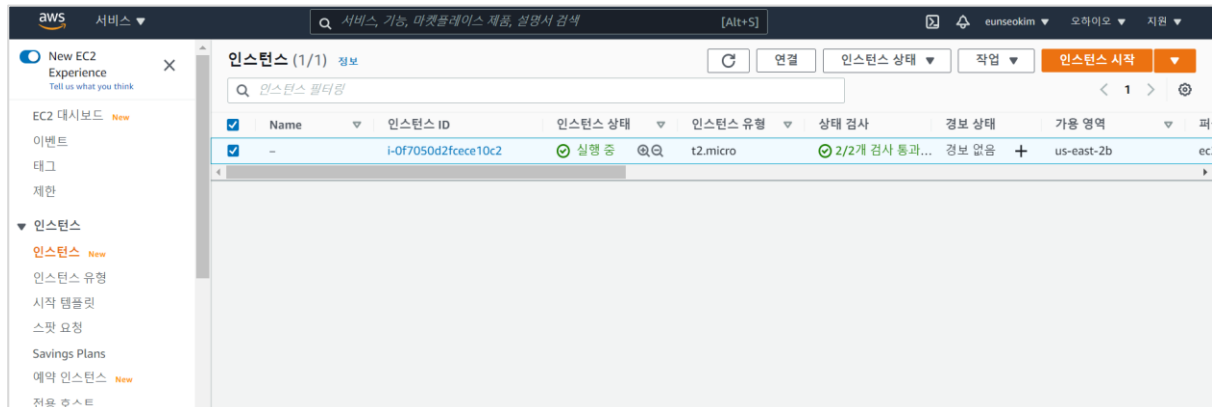
```
@media screen and (max-width: 767px) {
    .navbar {
        width: 100%;
        display: flex;
        flex-direction: column;
        align-items: flex-start;
        padding: 8px 24px;
    }

    .navbar__menu {
        display: none;
        flex-direction: column;
        align-items: center;
        width: 100%;
    }
}
```

- 메뉴바 반응형 웹 구현을 위한 CSS 부분

반응형 웹사이트의 구현을 위해서는 CSS media query를 사용하여 media screen의 max-width가 767px 이하가 되는 경우에 화면의 레이아웃을 다르게 배치함으로써 구현하였습니다.

9. AWS를 통한 웹 호스팅



- Daily의 AWS EC2 인스턴스

Daily는 실제로 AWS의 EC2를 사용하여 웹 호스팅까지 완료한 웹사이트입니다. 따라서 현재 (<http://dailyme.kro.kr/>)라는 도메인으로 운영되고 있습니다. (해당 링크를 누르면 실제로 Daily에 접속 가능합니다.)

AWS를 통한 웹 호스팅 방법을 간단하게 설명 드리자면 다음과 같습니다.

- (1) AWS의 EC2를 통해 인스턴스를 생성합니다.
- (2) 보안 그룹을 생성하여 ssh, http, https 포트를 열어줍니다.
- (3) 발급받은 private key로 SSH Client를 시작합니다. 그리고 SSH를 통해 EC2 Linux 인스턴스에 연결합니다.
- (4) AWS EC2에 APM(Apache, PHP, MariaDB)을 설치합니다.
- (5) 로컬에 저장된 phpmyadmin의 DB를 복사하고, AWS MariaDB의 외부 접속을 통해 복사한 DB를 옮겨 붙여 넣습니다.
- (6) 웹사이트의 IP주소를 바탕으로 도메인을 등록합니다.

도메인은 내도메인(<https://xn--220b31d95hq8o.xn--3e0b707e/>) 웹사이트를 통해 무료로 등록하여 사용했습니다.

모든 소스코드는 Github에서 확인하실 수 있습니다.

소스코드 보기: https://github.com/eunseo-kim/Website_Project

Daily 바로가기: <http://dailyme.kro.kr/>

