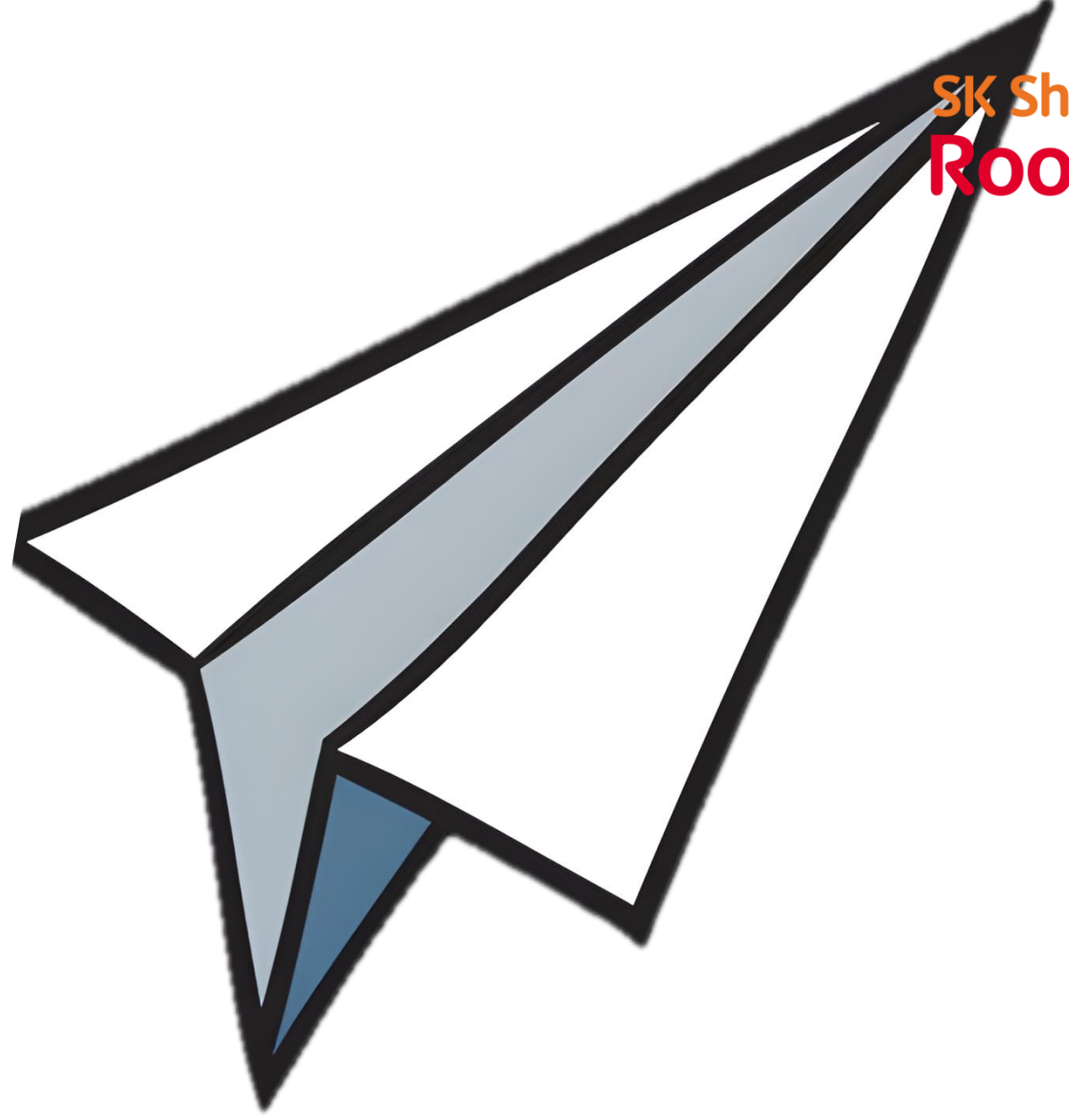


여행자의 기록

강민수



1. 프로젝트 개요

프로젝트 개요

[프로젝트 배경]

- CRUD가 가능한 주제 정의
- 회원, 파일 첨부, 게시글 관리 기능 포함
- > 여행지 추천 사이트로 적합

[프로젝트 목표]

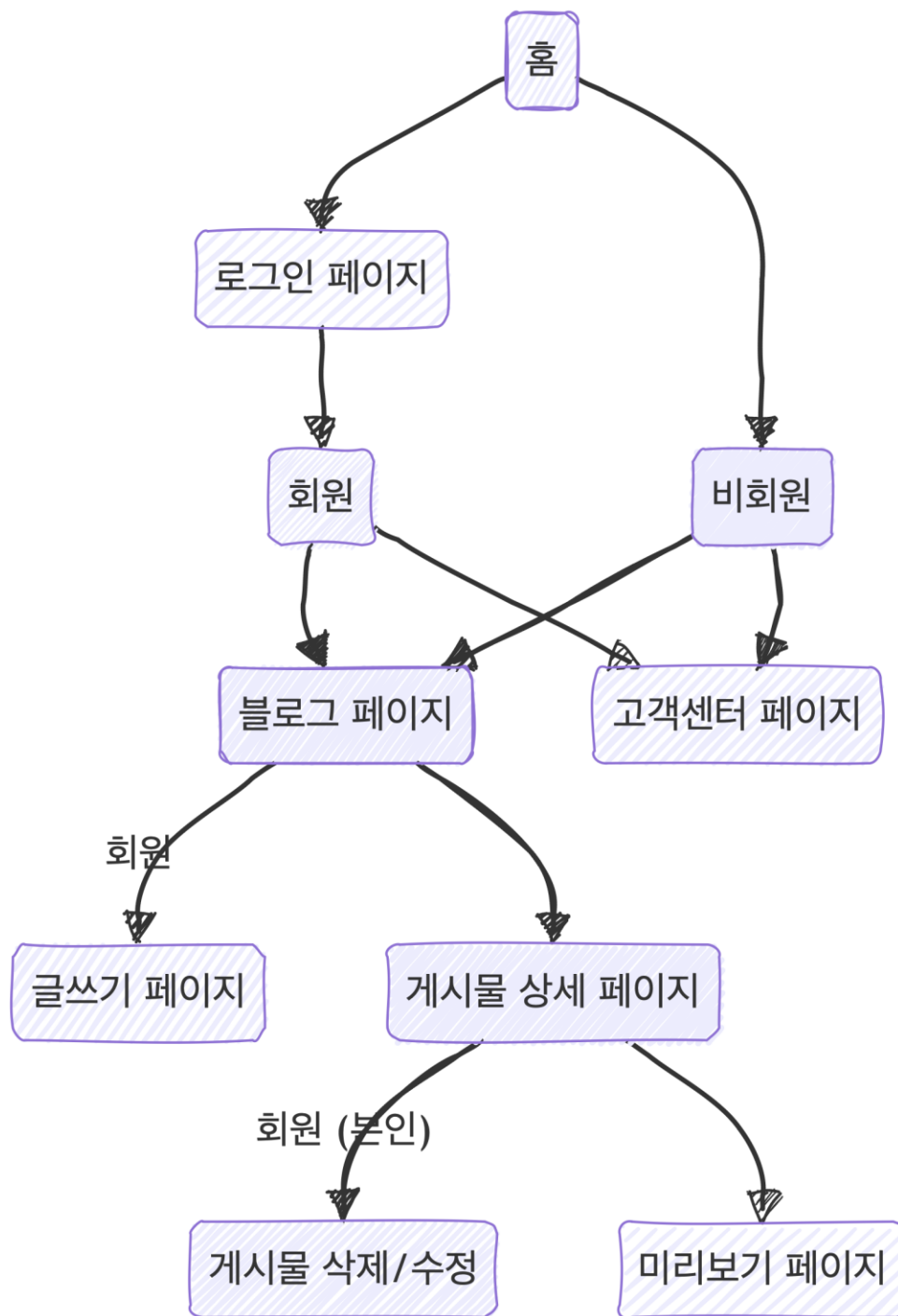
- 사용자 간 여행지 추천 웹 개발
- 여행자 정보 교류 및 커뮤니티 형성
- 여행 경험 공유 및 유용한 정보 제공

2. 프로젝트 계획

프로젝트 수행 절차

구분	기간	활동	비고
사전 기획	▶ 07/24 ~ 07/24	프로젝트 계획서 작성	
개발	▶ 07/25 ~ 07/28	웹 개발 구현	
기능 구현 및 연동	▶ 07/29 ~ 07/30	프론트-백 통합	테스트 및 오류수정
보고서 작성	▶ 07/31 ~ 08/01	PPT 및 시연영상	
총 개발기간	▶ 07/24 ~ 08/01		

다이아그램



3. 프로젝트 수행 결과

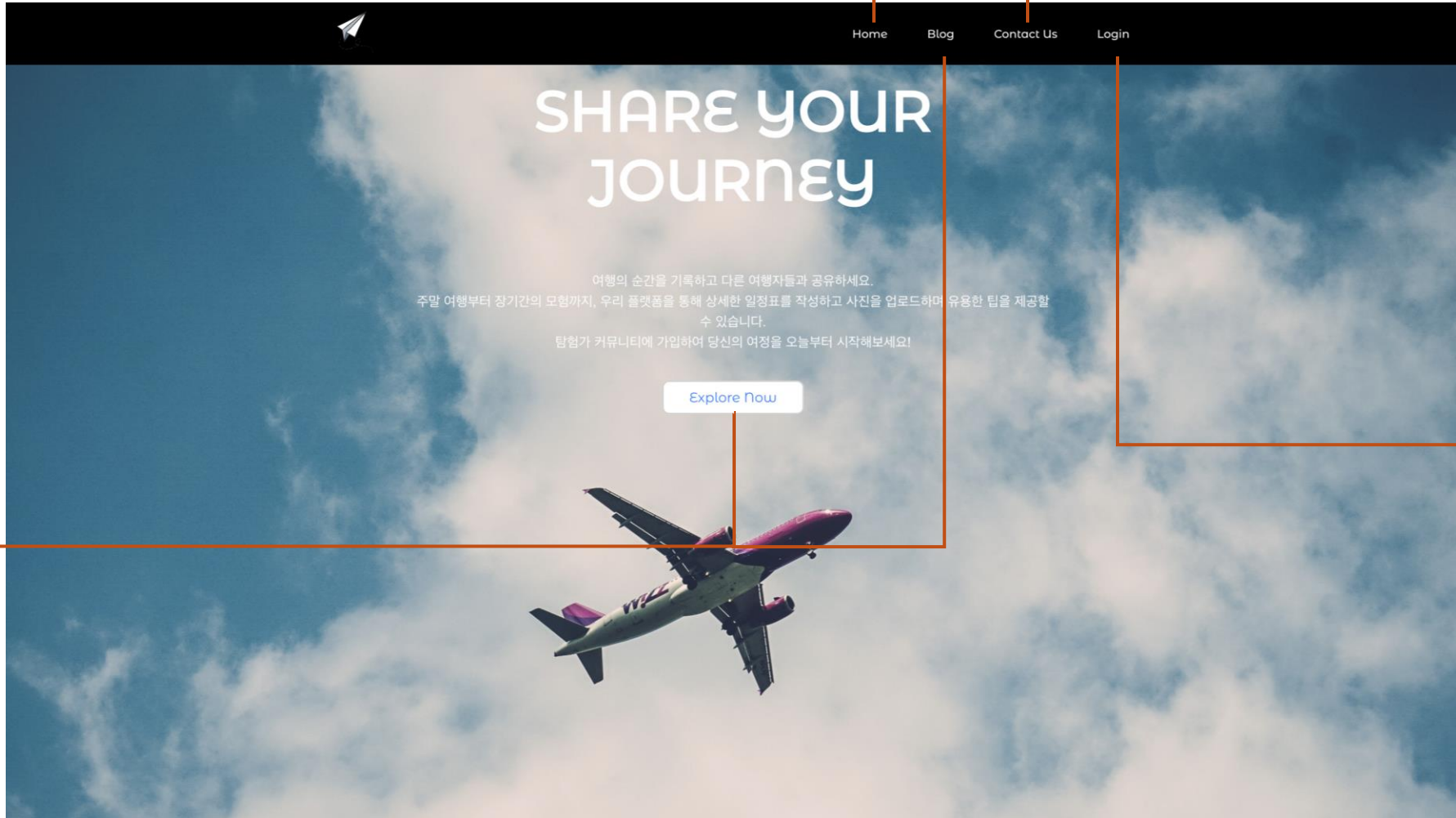
/home

홈 버튼

고객센터 버튼

로그인 버튼

블로그 버튼



/signup

Sign Up

required // HTML 자동 유효성 검사

ID 입력 필드

- 생성할 ID 입력

ID

Password

Confirm Password

비밀번호 입력 & 재입력 필드

- 비밀번호 입력
- 입력한 비밀번호와 동일한 내용 입력

이름 입력 필드

- 본인 이름 입력

Name

이메일 입력 필드

- 본인 이메일 입력

Email

회원가입 버튼

- 입력된 데이터 DB에 저장
- 가입 성공시 로그인 페이지로 이동

Sign Up

로그인 페이지 링크

- 로그인 페이지로 이동

[Log In](#)

/login

ID 입력 필드

- 생성한 ID 입력

Log In

Login failed. Please try again.

ID

Password

에러메시지

- ID | 패스워드 오류시 출력

비밀번호 입력 필드

- 생성한 비밀번호 입력

로그인 버튼

- 로그인 성공시 메인 페이지로 이동

Log In

[Sign Up](#)

회원가입 페이지 링크

- 회원가입 페이지로 이동

키워드 검색

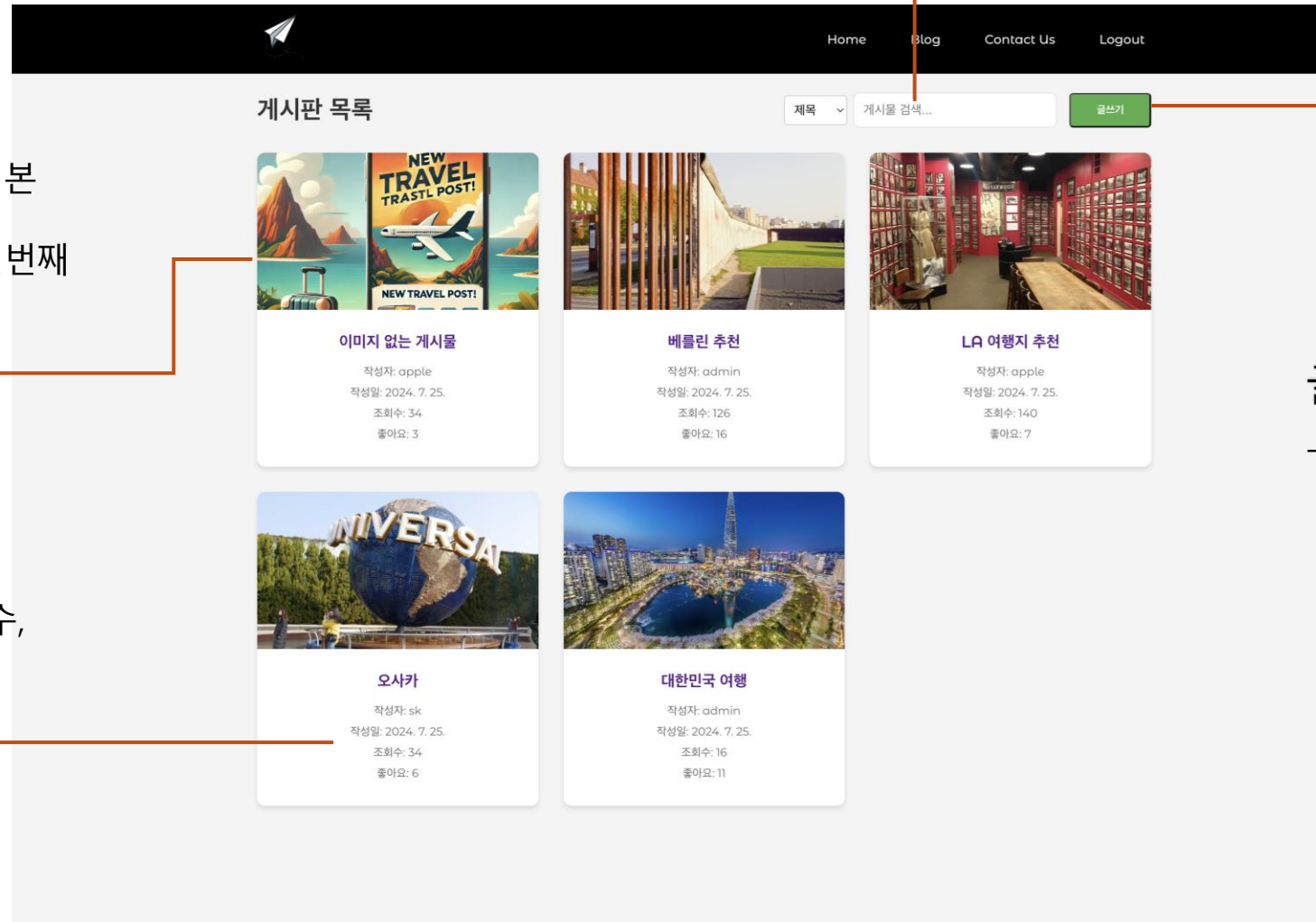
- 제목 및 작성자 검색 기능

게시물

- 첨부파일이 없으면 기본 사진 적용
- 첨부파일이 있으면 첫번째 사진 적용

게시물 디테일

- 작성자, 작성일, 조회수, 좋아요 수



글쓰기 버튼

- 로그인 안하면 글쓰기 불가

글쓰기 버튼 클릭 시 로그인 여부를 확인하는 기능

```
const handleLogin = (username) => {  
  setIsLoggedIn(true);  
  setUsername(username);  
  localStorage.setItem('isLoggedIn', 'true');  
  localStorage.setItem('username', username);  
};
```

```
const handleLogout = () => {  
  setIsLoggedIn(false);  
  setUsername('');  
  localStorage.removeItem('isLoggedIn');  
  localStorage.removeItem('username');  
};
```

BoardList.js

- 로그인 상태가 아니면 '/login'으로 이동
- 로그인 상태이면 '/write'로 이동

App.js

- React 라이브러리 기능을 사용하여 isLoggedIn 상태 설정

```
const handleWriteClick = () => {  
  if (!isLoggedIn) {  
    alert('로그인이 필요합니다.');    navigate('/login');  } else {  
    navigate('/write');  }  
};
```

i 게시물 이미지 기능

RestApiBoardController.java

- selectBoardFileInfo를 호출하여 파일 정보 조회
- 파일 정보가 있으면 URL을 생성하여 반환

// 추가 : 이미지 파일 URL을 반환하는 메서드 >> 프론트내에서 이미지를 띄우기 위한 코드

@GetMapping("/board/file/url/{boardIdx}/{idx}")

```
public ResponseEntity<String> getBoardFileUrl(@PathVariable("idx") int idx, @PathVariable("boardIdx") int boardIdx)
    throws Exception {
    BoardFileDto boardFileDto = boardService.selectBoardFileInfo(idx, boardIdx);
    if (ObjectUtils.isEmpty(boardFileDto)) {
        return ResponseEntity.status(HttpStatus.NOT_FOUND).body("File not found");
    }
    String fileUrl = "/api/board/file/" + boardIdx + "/" + idx;
    return ResponseEntity.status(HttpStatus.OK).body(fileUrl);
}
```

```
useEffect(() => {
  axios
    .get('http://localhost:8080/api/board')
    .then(res => {
      console.log('API Response:', res.data);
      res && res.data && setDatas(res.data);
    })
    .catch(err => console.log(err));
}, []);
```

BoardList.js

- 파일이 있으면 첫 번째 파일의 이미지 설정
- 파일이 없으면 대체 이미지 설정

```
<img
  src={board.fileInfoList && board.fileInfoList.length > 0
    ? `http://localhost:8080/api/board/file/${board.boardIdx}/${board.fileInfoList[0].idx}`
    : boardImage}
  className="board-card-img"
/>
```

/write

게시글 작성

제목

제목을 입력하세요

내용

내용을 입력하세요

제목 및 내용

- 제목 및 내용 쓰기

첨부파일 버튼

- 최소 0개 / 최대 10개 가능

파일 첨부 (최대 5개)

파일 선택 선택된 파일 없음

저장/취소 버튼

- 저장 및 취소

저장

취소

i 글쓰기 페이지

```
const handleTitleChange = e => setTitle(e.target.value);
const handleContentsChange = e => setContents(e.target.value);
```

‘handleTitleChange’와 ‘handleContentsChange’ 함수
상태 업데이트

```
const handleSubmit = e => {
  e.preventDefault();
  const formData = new FormData();
  formData.append('data', new Blob([JSON.stringify({ title, contents })], { type: 'application/json' }));
  files.forEach(file => formData.append('files', file));

  axios.post('http://localhost:8080/api/board/write', formData, {
    headers: { 'Content-Type': 'multipart/form-data' },
    withCredentials: true
  })
  .then(res => {
    if (res && res.status === 200) {
      navigate('/blog');
    }
  })
  .catch(err => console.error("Error submitting form:", err));
};
```

```
const handleFileChange = e => {
  const selectedFiles = Array.from(e.target.files);
  if (selectedFiles.length > 10) {
    alert('이미지는 최대 10개까지만 업로드 가능합니다.');
```

‘handleSubmit’ 함수는 폼 제출 시 호출

/detail/:boardIdx

좋아요 버튼

- 사용자 마다 한번씩 좋아요 버튼 가능

제목 & 설명

- 제목 & 설명 TextField

LA 여행지 추천

작성자: apple

작성일: 2024.07.25 14:39:42

조회수: 142



9

첨부파일 ▼

더 할리우드 뮤지엄 - 할리우드 영화에서 사용되었던 소품들과 사진들을 전시하는 박물관. 마릴린 먼로의 사진이 엄청 많습니다.

워크 오브 페임 - 약 2690여명의 세렘들의 이름이 들어가 있는 별 그림이 2.1km 펼쳐져 있는 거리. 좋아하는 연예인의 이름을 찾아보는 재미가 있습니다.

죽음의 박물관 - 살아있는 사람들을 행복하게 하기 위해서("to make people happy to be alive")라는 슬로건을 가지고 만든 죽음을 주제로 한 박물관입니다.

할리우드 벽화 - 할리우드 유명배우와 캐릭터들을 담은 벽화.

#미국 #la

첨부파일 버튼

- Dropdown 기능으로 파일이름을 본 후 다운로드 가능

첨부파일 필드

- 클릭 시 미리보기 가능



더할리우드뮤지엄.jpg



워크오브페임.png



죽음의박물관.png



할리우드벽화.png

목록 / 수정 / 삭제 버튼

- 본인 일시 수정/삭제 버튼 렌더링
- 본인이 아닐 시 수정/삭제 버튼 제거

목록

수정

삭제

목록

i 첨부파일 기능

```
<div className="attachment-dropdown">
  <button onClick={() => setShowAttachments(!showAttachments)}>
    첨부파일 {showAttachments ? '▲' : '▼'}
  </button>
  {showAttachments && (
    <ul className="attachment-list">
      {board.fileInfoList && board.fileInfoList.length > 0 ? (
        board.fileInfoList.map(fileInfo => (
          <li key={fileInfo.idx} onClick={() => handleDownload(fileInfo)}>
            {fileInfo.originalFileName}
          </li>
        ))
      ) : (
        <li>첨부파일 없음</li>
      )}
    </ul>
  )}
</div>
```

```
const handleDownload = (fileInfo) => {
  const { boardIdx, idx, originalFileName } = fileInfo;
  axios({
    url: `http://localhost:8080/api/board/file/${boardIdx}/${idx}`,
    method: 'GET',
    responseType: 'blob'
  })
  .then(res => {
    const url = window.URL.createObjectURL(new Blob([res.data]));
    const link = document.createElement('a');
    link.href = url;
    link.setAttribute('download', originalFileName);
    document.body.appendChild(link);
    link.click();
    link.parentNode.removeChild(link);
  })
  .catch(err => console.error("Error downloading file:", err));
};
```

BoardDetail.js

- 파일이 있으면 RestApiBoardController에서 설정된 URL을 통해 이미지를 표시 후 클릭시 다운로드 가능

이미지 미리보기 & 수정/삭제 버튼 기능

```
<div className="attached-images">
  {board.fileInfoList && board.fileInfoList.map(fileInfo => (
    <div key={fileInfo.idx} className="attached-image-item">
      <img
        src={`http://localhost:8080/api/board/file/${fileInfo.boardIdx}/${fileInfo.idx}`}
        alt={fileInfo.originalFileName}
        onClick={() => window.open(`/preview/${boardIdx}/${fileInfo.idx}`, '_blank')}
      />
      <span>{fileInfo.originalFileName}</span>
    </div>
  )]}
</div>
```

BoardDetail.js

- 이미지를 클릭하면 RestApiBoardController에서 설정된 URL을 통해 해당 이미지의 미리 보기 창으로 이동

```
<button className="btn list-btn" onClick={() => handleAction('list')}>목록으로</button>
{board.creatorId === username && (
  <>
    <button className="btn update-btn" onClick={() => handleAction('update')}>수정하기</button>
    <button className="btn delete-btn" onClick={() => handleAction('delete')}>삭제하기</button>
  </>
)}
}
```

BoardDetail.js

- 로그인된 사용자의 username (ID)과 게시글의 creatorId (작성자)가 동일 시 버튼 렌더링

i 좋아요 버튼

```
setLikes(prevLikes => {
  const updatedLikes = { ...prevLikes, [boardIdx]: hasLiked ? (prevLikes[boardIdx] || 1) - 1 : (prevLikes[boardIdx] || 0) + 1 };
  localStorage.setItem('likes', JSON.stringify(updatedLikes));
  return updatedLikes;
});
```

```
setUserLikes(prevUserLikes => {
  const userLikeList = prevUserLikes[username] ? [...prevUserLikes[username]] : [];
  if (hasLiked) {
    const index = userLikeList.indexOf(boardIdx);
    if (index > -1) userLikeList.splice(index, 1);
  } else {
    userLikeList.push(boardIdx);
  }
  const updatedUserLikes = { ...prevUserLikes, [username]: userLikeList };
  localStorage.setItem('userLikes', JSON.stringify(updatedUserLikes));
  return updatedUserLikes;
});
```

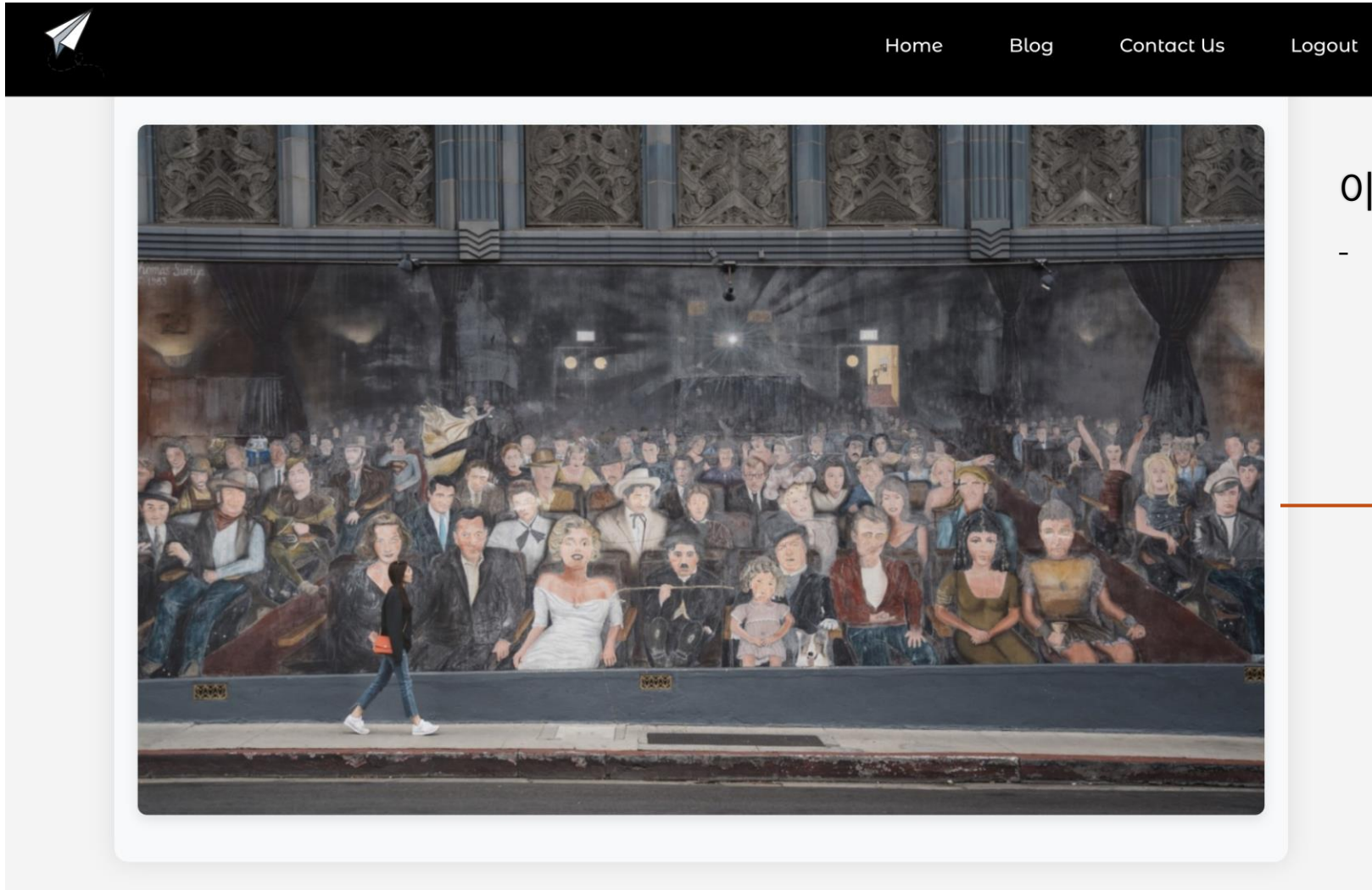
```
<div className="like-section">
  <button
    className={`like-button ${userLikes[username] && userLikes[username].includes(boardIdx) ? 'liked' : ''}`}
    onClick={handleLikeClick}
    disabled={!username}
  >
    <span className="like-icon">♥</span>
    <span className="like-count">{likes[boardIdx] || 0}</span>
  </button>
</div>
```

BoardDetail.js

- 좋아요 수 업데이트
- hasLiked가 true이면 현재 좋아요 수에서 1을 빼고, 그렇지 않으면 1을 더함

- 로그인을 하지 않으면 비활성화
- 버튼을 클릭하면 handleLikeClick 함수 실행

/preview/:boardIdx/:fileIdx



이미지 미리보기

- 단순 크게 보기 페이지

이미지 미리보기 기능

```
return (  
  <div className="preview-container">  
    <div className="preview-images">  
      {board.fileInfoList && board.fileInfoList.map(fileInfo => (  
        <div key={fileInfo.idx} className="image-wrapper">  
          <img  
            src={`http://localhost:8080/api/board/file/${fileInfo.boardIdx}/${fileInfo.idx}`}  
            className="preview-image"  
          />  
        </div>  
      ) )}  
    </div>  
  </div>  
);
```

BoardPreview.js

- RestApiBoardController에서 설정된 URL을 통해 이미지를 표시

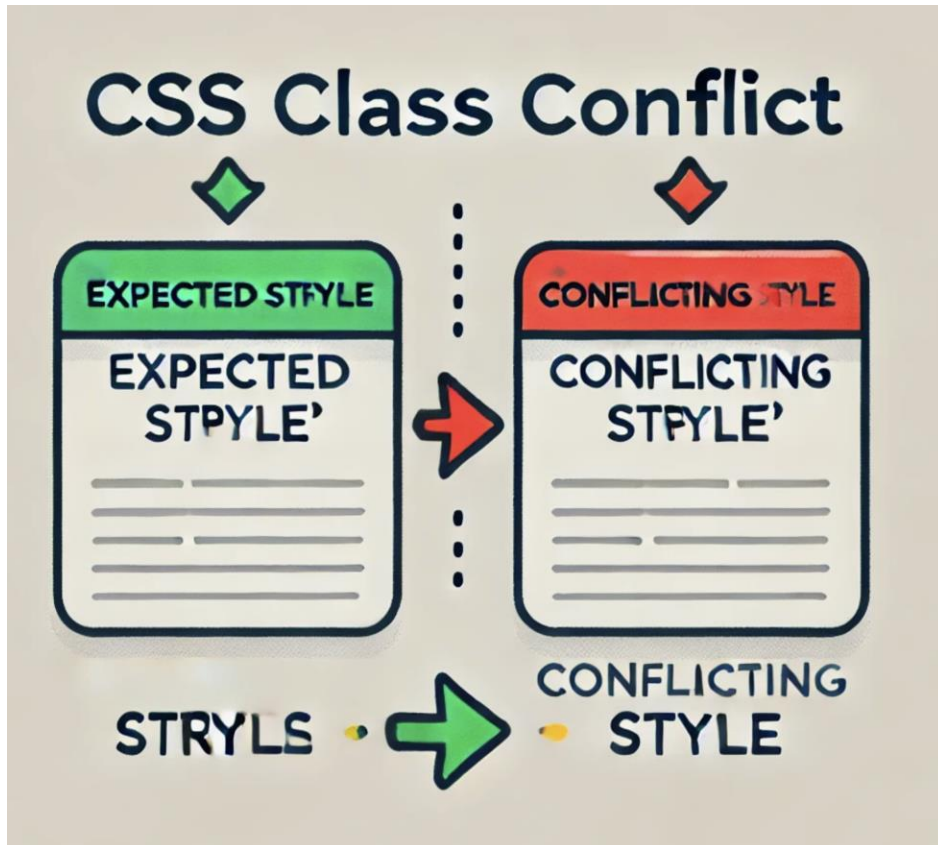
4. 자체 평가 의견

자체 평가 의견

II 개발 경험이 부족하고 역량의 한계로 인해 어려움을 겪었지만, 프로젝트를 진행하면서 강의를 통해 습득한 지식을 적극 활용했습니다.

II 이러한 과정을 통해 체계적으로 문제를 해결하고, 프로젝트를 성공적으로 완료할 수 있었습니다. 이번 경험은 실질적인 개발 역량을 크게 향상시키는 중요한 계기가 되었습니다.

Issue & Solution #1



NG ISSUE

- React에 CSS가 적용되지 않는 문제

OK SOLUTION

- 중복된 클래스 네임으로 충돌 문제 발생, 전체적인 클래스 네임 수정 완료

Issue & Solution #2

```
:43,910 INFO {jdbc.resultsettable}
```

board_idx	contents	created_datetime	creator_id	hit_cnt	title	updated_datetime	updater_id

```
:44,834 INFO {board.BoardApplication} Started BoardApplication in 2.84 seconds (process running for 3.121
:50,946 DEBUG {board.interceptor.LoggerInterceptor} ***** START *****
:50,946 DEBUG {board.interceptor.LoggerInterceptor} Request URI /loginProc
:50,954 DEBUG {board.interceptor.LoggerInterceptor} ***** START *****
:50,954 DEBUG {board.interceptor.LoggerInterceptor} Request URI /error
:50,983 DEBUG {board.interceptor.LoggerInterceptor} ***** END *****
```

NG ISSUE

- React에서 로그인 시도 시 로그인 API가 정상적으로 호출되지 않는 문제

OK SOLUTION

- 로그인 데이터를 URL-encoded 형식으로 수신하도록 수정

마무리 하며...

✓ 이번 프로젝트를 통해 많은 것을 배우고 성장할 수 있었습니다. 처음에는 개발 경험이 부족해 어려움이 있었지만, 꾸준한 학습과 문제 해결 과정을 통해 성공적으로 마무리할 수 있었습니다.

✓ 이번 경험을 바탕으로 앞으로도 계속 성장해 나가겠습니다. 스스로 도전하고 문제를 해결하는 과정에서 얻은 자신감은 앞으로의 도전에도 큰 힘이 될 것입니다.

시연영상

BGM = One Way Back by Rhett Nelson

