

로그인 페이지 만들기

널포유

이현덕

목차

Table of Contents



기능 소개

기능 확인

코드 확인

목차

Table of Contents



기능 소개

기능 확인

코드 확인

목차

Table of Contents

기능 소개

웹 페이지 소개

코드 확인



기능 소개

로그인, 메모장 기능

1

로그인, 로그아웃, 계정가입,메모 기능

login, logout, signup, memo

2

겹치는 아이디 가입 불가

겹쳐지는 계정을 가입하지 못하도록
제한함

3

메모장 기능

언어설정(영어),휴지통

웹페이지 소개<login>

webpage

로그인 페이지

회원가입으로 넘어가는 페이지,
아이디와 비밀번호가 존재한다면 가입이 불가함,



로그인

아이디

비밀번호

로그인

[회원가입 하러가기](#)

로그인 페이지

웹 페이지 소개<signup>

webpage

회원가입 페이지

간단한 회원가입 페이지



회원가입

아이디

비밀번호

비밀번호 확인

가입하기

[로그인으로 돌아가기](#)

회원가입 하는 페이지

웹 페이지 소개<welcome>

webpage

환영합니다 페이지

회원가입 완료 후 로그인시 나오는 페이지



웹 페이지 소개

webpage

메모장 버튼 누르면 ?

두근두근

"메모장" 버튼을 누른 후 확인 버튼 클릭



메모장으로 가버리기

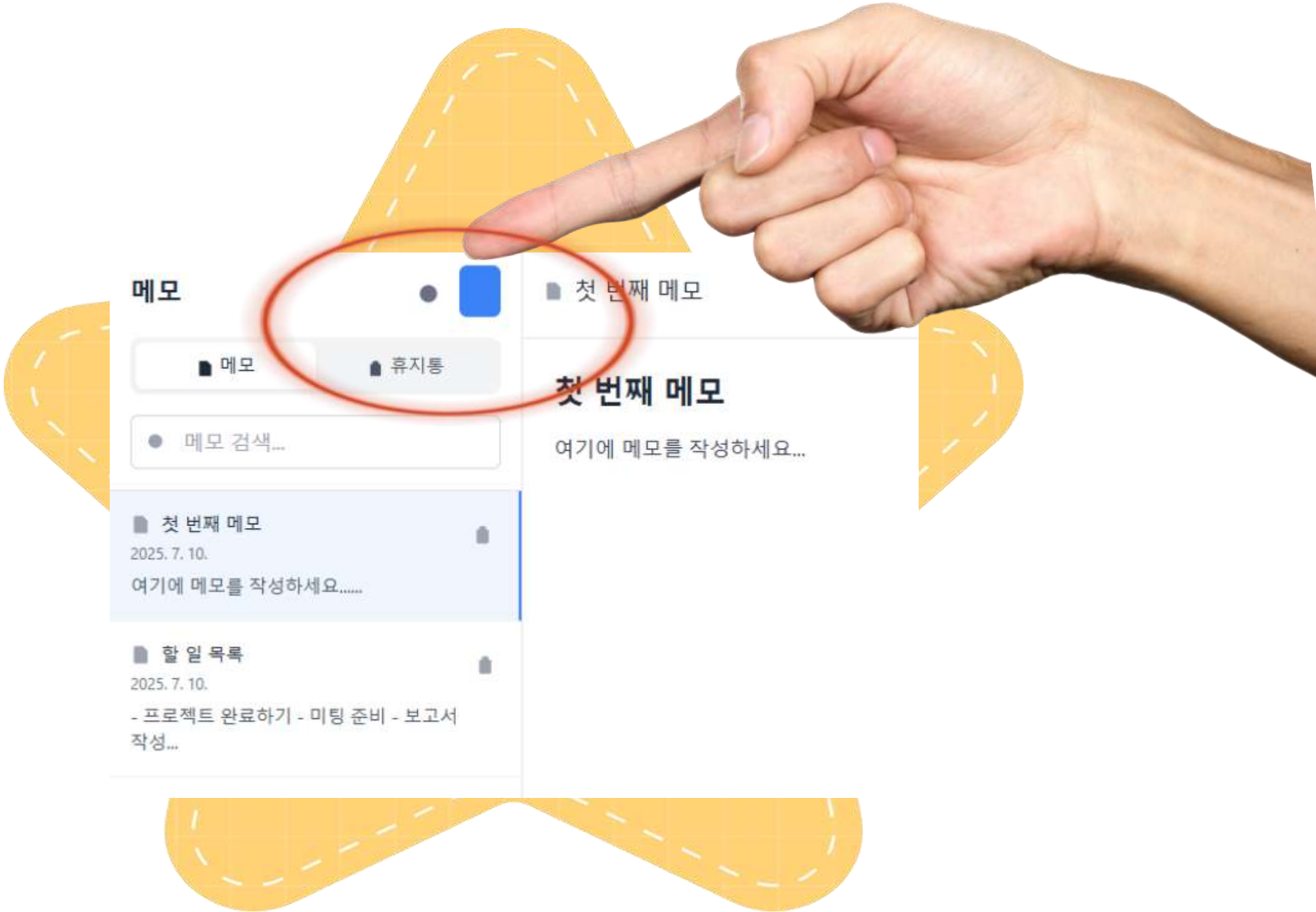
웹 페이지 소개

webpage

메모 페이지



동그란버튼을 누르면 영어로 바뀔
네모난 버튼을 누르면 메모 생성

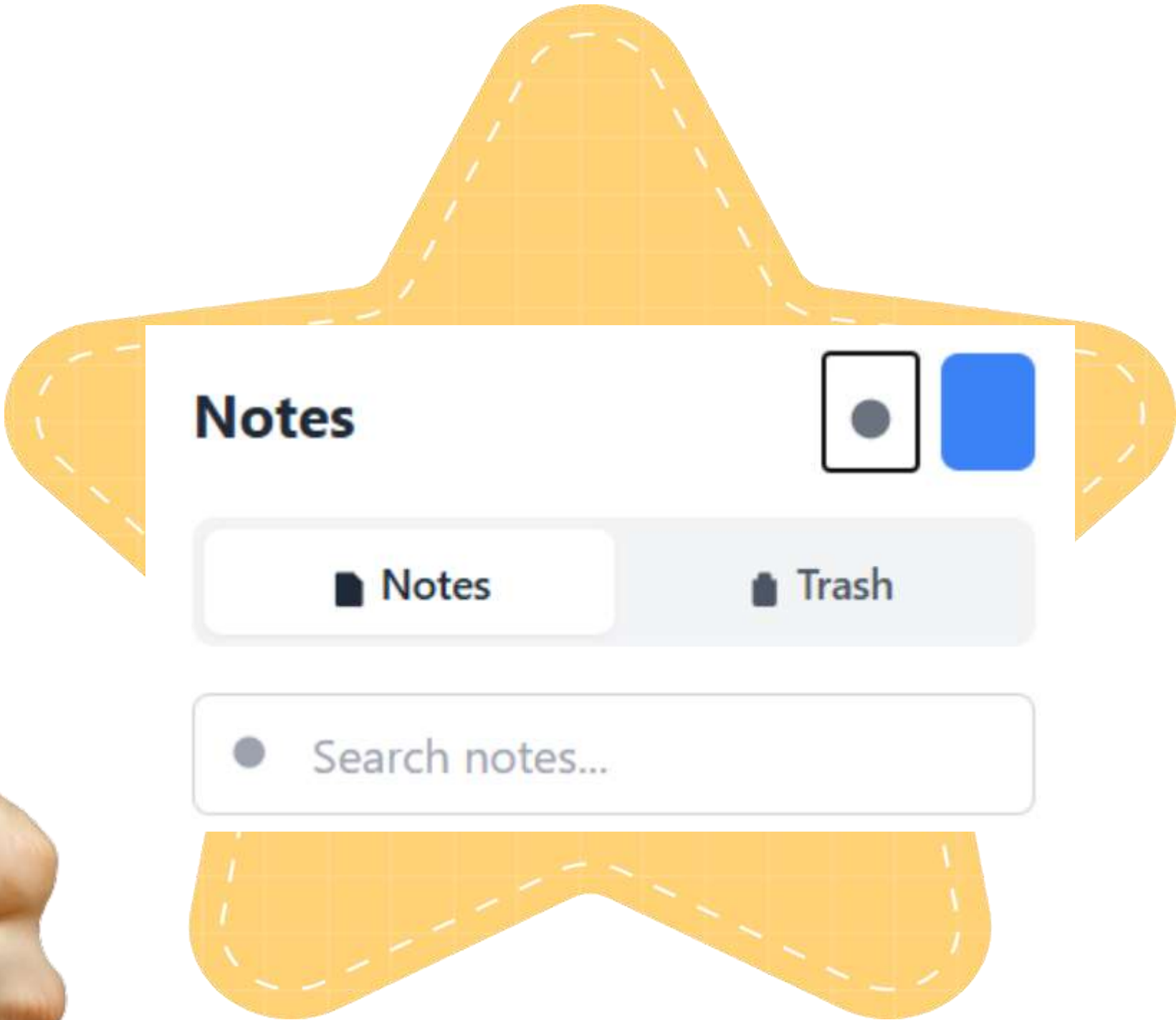


웹 페이지 소개

webpage

메모 페이지

영어로 바뀐 모습



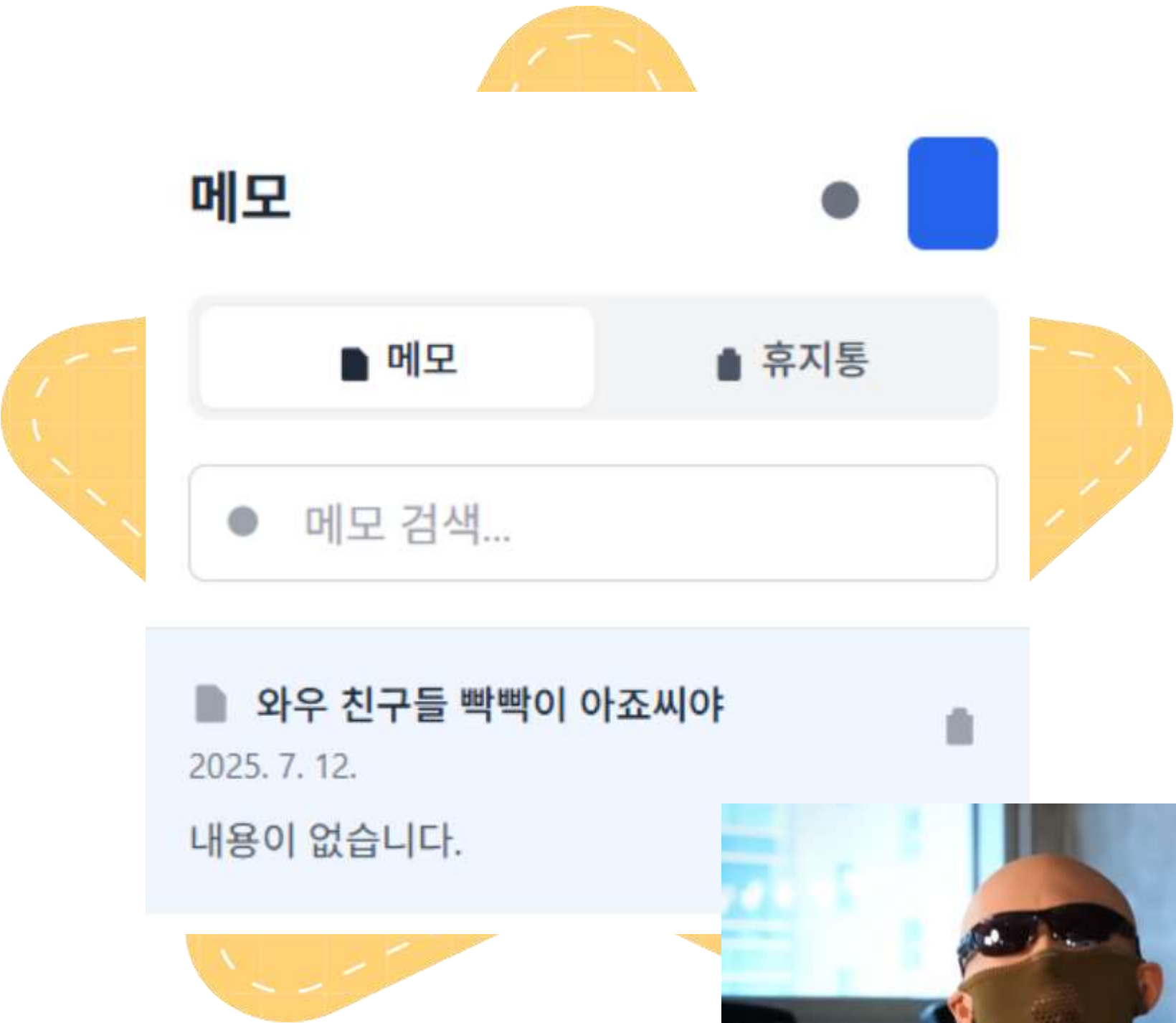
t사진에 대한 설명을 간략하게 입력해주세요.

웹 페이지 소개

webpage

메모 페이지

메모장 추가되기 전

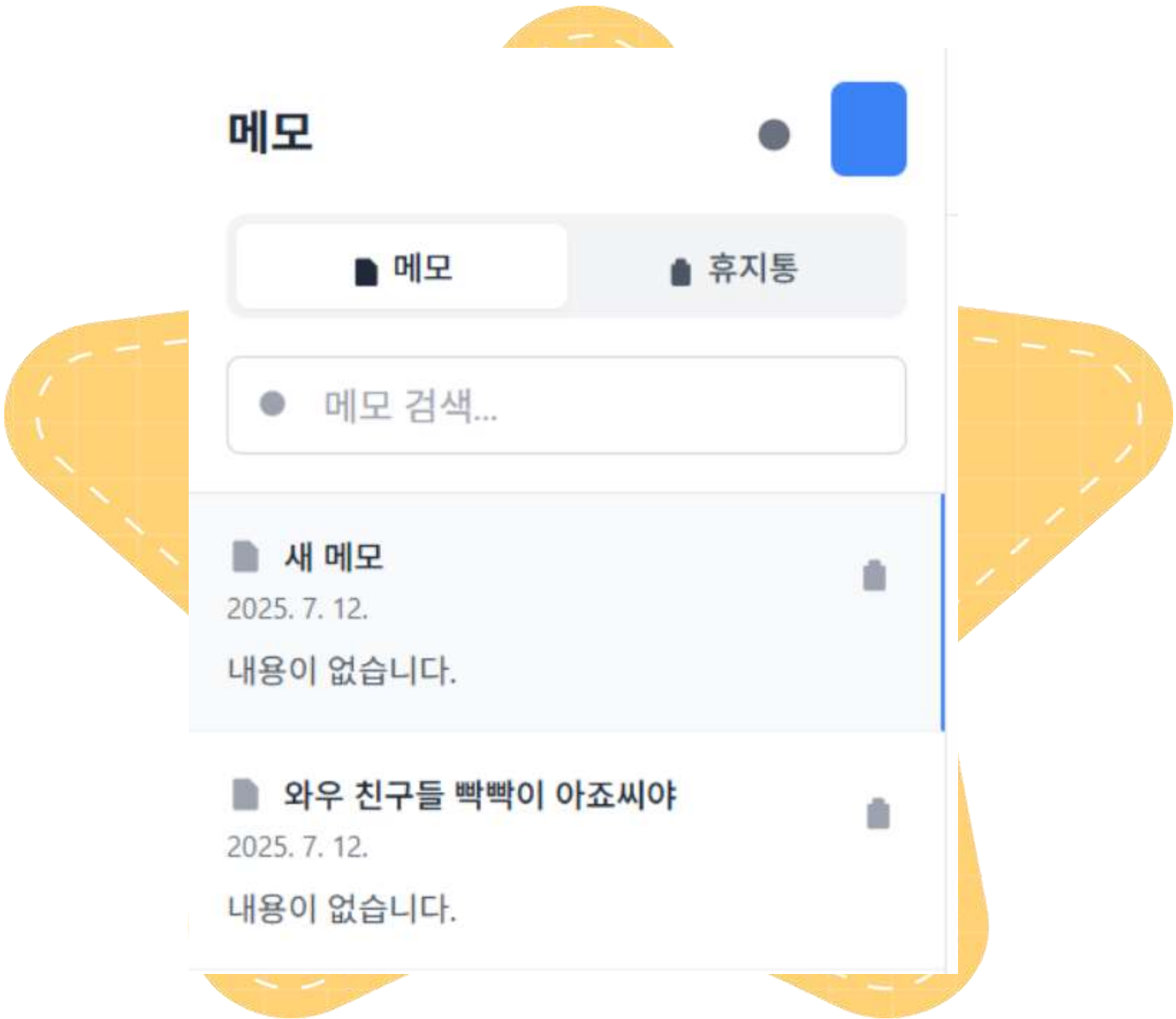


웹 페이지 소개

webpage

메모 페이지

메모장 페이지가 추가된 모습



t사진에 대한 설명을 간략하게 입력해주세요.

웹 페이지 소개

webpage

메모 페이지 , 로그인 페이지
구현 시연



코드 확인

code check

login code

이미지 삽입하는 코드

```
<div class="left-panel">
  
</div>

<div class="center-panel">
  <div class="login-box">
    <h2>로그인</h2>
    <form id="loginForm">
      <input type="text" id="loginUsername" placeholder="아이디" required>
      <input type="password" id="loginPassword" placeholder="비밀번호" required>
      <button type="submit">로그인</button>
    </form>
    <a href="signup.html">회원가입 하러가기</a>
  </div>
</div>

<div class="right-panel">
  
</div>
```

이미지 삽입 하는 코드 (세명컴고 로고)

코드 확인

code check

login code

```
<div class="left-panel">
  
</div>

<div class="center-panel">
  <div class="login-box">
    <h2>로그인</h2>
    <form id="loginForm">
      <input type="text" id="loginUsername" placeholder="아이디" required>
      <input type="password" id="loginPassword" placeholder="비밀번호" required>
      <button type="submit">로그인</button>
    </form>
    <a href="signup.html">회원가입 하러가기</a>
  </div>
</div>

<div class="right-panel">
  
</div>
```

3개로 나눈거 중 가운데에 로그인 박스 삽입

코드 확인

code check

login code

```
<div class="left-panel">
  
</div>

<div class="center-panel">
  <div class="login-box">
    <h2>로그인</h2>

    <form id="loginForm">
      <input type="text" id="loginUsername" placeholder="아이디" required>
      <input type="password" id="loginPassword" placeholder="비밀번호" required>
      <button type="submit">로그인</button>
    </form>

    <a href="signup.html">회원가입 하러가기</a>
  </div>
</div>

<div class="right-panel">
  
</div>
```

text 아이디와 비밀번호 입력받고, <로그인>버튼 추가

코드 확인

code check

login code

```
<div class="left-panel">
  
</div>

<div class="center-panel">
  <div class="login-box">
    <h2>로그인</h2>
    <form id="loginForm">
      <input type="text" id="loginUsername" placeholder="아이디" required>
      <input type="password" id="loginPassword" placeholder="비밀번호" required>
      <button type="submit">로그인</button>
    </form>
    <a href="signup.html">회원가입 하러가기</a>
  </div>
</div>

<div class="right-panel">
  
</div>
```

a href="@.html"> 다른 html 으로 이동하는 코드

코드 확인

code check

login code

```
<div class="left-panel">
  
</div>
```

```
<div class="center-panel">
  <div class="login-box">
    <h2>로그인</h2>
    <form id="loginForm">
      <input type="text" id="loginUsername" placeholder="아이디" required>
      <input type="password" id="loginPassword" placeholder="비밀번호" required>
      <button type="submit">로그인</button>
    </form>
    <a href="signup.html">회원가입 하러가기</a>
  </div>
</div>
```

```
<div class="right-panel">
  
```

gif 삽입, img src="" 뒤에 loop=infinite 으로 gif 무한 반복

코드 확인

code check

login code

```
<script>
  async function hashValue(value) {
    const encoder = new TextEncoder();
    const data = encoder.encode(value);
    const hashBuffer = await crypto.subtle.digest('SHA-256', data);
    const hashArray = Array.from(new Uint8Array(hashBuffer));
    return hashArray.map(b => b.toString(16).padStart(2, '0')).join('');
  }
}
```

입력받은 value(문자열)를 SHA-256으로 해시하는 비동기 함수
TextEncoder로 문자열을 UTF-8 바이트로 변환

crypto.subtle.digest()를 통해 SHA-256 해시값

SHA-256은 보안, 암호화, 인증 등에서 매우 널리 사용되는 해시 함수입니다.

코드 확인

code check

login code

```
document.getElementById("loginForm").addEventListener("submit", async function (e) {  
    e.preventDefault();
```

폼이 서버에 전송되는 기본 동작을 막고, js 로만 로그인 처리하게 함

코드 확인

code check

login code

```
const username = document.getElementById("loginUsername").value;
const password = document.getElementById("loginPassword").value;

const hashedUsername = await hashValue(username);
const hashedPassword = await hashValue(password);

const storedUser = localStorage.getItem("user_" + hashedUsername);
if (!storedUser) {
  alert("존재하지 않는 아이디입니다.");
  return;
}

const userData = JSON.parse(storedUser);
if (userData.password !== hashedPassword) {
  alert("비밀번호가 일치하지 않습니다.");
  return;
}

alert(username + "님, 로그인 성공!");
localStorage.setItem("savedUsername", username);
window.location.href = 'welcome.html';
});
```

(사용자로 부터 입력된 값 가져오기 (비번,아이디))

코드 확인

code check

login code

```
const username = document.getElementById("loginUsername").value;
const password = document.getElementById("loginPassword").value;

const hashedUsername = await hashValue(username);
const hashedPassword = await hashValue(password);

const storedUser = localStorage.getItem("user_" + hashedUsername);
if (!storedUser) {
  alert("존재하지 않는 아이디입니다.");
  return;
}

const userData = JSON.parse(storedUser);
if (userData.password !== hashedPassword) {
  alert("비밀번호가 일치하지 않습니다.");
  return;
}

alert(username + "님, 로그인 성공!");
localStorage.setItem("savedUsername", username);
window.location.href = 'welcome.html';
});
```

입력된 비번과 아이디를 SHA-256 해시값으로 변환

코드 확인

code check

login code

```
const username = document.getElementById("loginUsername").value;
const password = document.getElementById("loginPassword").value;

const hashedUsername = await hashValue(username);
const hashedPassword = await hashValue(password);

const storedUser = localStorage.getItem("user " + hashedUsername);
if (!storedUser) {
    alert("존재하지 않는 아이디입니다.");
    return;
}

const userData = JSON.parse(storedUser);
if (userData.password !== hashedPassword) {
    alert("비밀번호가 일치하지 않습니다.");
    return;
}

alert(username + "님, 로그인 성공!");
localStorage.setItem("savedUsername", username);
window.location.href = 'welcome.html';
});
```

해시된 ID를 user_<해시된 ID> 형태로 조합하여 localStorage에서 해당 데이터를 가져옵니다.

코드 확인

code check

login code

```
const username = document.getElementById("loginUsername").value;
const password = document.getElementById("loginPassword").value;

const hashedUsername = await hashValue(username);
const hashedPassword = await hashValue(password);

const storedUser = localStorage.getItem("user_" + hashedUsername);
if (!storedUser) {
  alert("존재하지 않는 아이디입니다.");
  return;
}

const userData = JSON.parse(storedUser);
if (userData.password !== hashedPassword) {
  alert("비밀번호가 일치하지 않습니다.");
  return;
}

alert(username + "님, 로그인 성공!");
localStorage.setItem("savedUsername", username);
window.location.href = 'welcome.html';
});
```

localStorage에 해당 키가 없으면 → "존재하지 않는 아이디입니다." 출력 후 종료

코드 확인

code check

login code

```
const username = document.getElementById("loginUsername").value;
const password = document.getElementById("loginPassword").value;

const hashedUsername = await hashValue(username);
const hashedPassword = await hashValue(password);

const storedUser = localStorage.getItem("user_" + hashedUsername);
if (!storedUser) {
  alert("존재하지 않는 아이디입니다.");
  return;
}

const userData = JSON.parse(storedUser);
if (userData.password !== hashedPassword) {
  alert("비밀번호가 일치하지 않습니다.");
  return;
}

alert(username + "님, 로그인 성공!");
localStorage.setItem("savedUsername", username);
window.location.href = 'welcome.html';
});
```

저장된 해시된 비밀번호와 입력한 비밀번호의 해시를 비교

다르면 "비밀번호가 일치하지 않습니다."

코드 확인

code check

login code

```
const username = document.getElementById("loginUsername").value;
const password = document.getElementById("loginPassword").value;

const hashedUsername = await hashValue(username);
const hashedPassword = await hashValue(password);

const storedUser = localStorage.getItem("user_" + hashedUsername);
if (!storedUser) {
  alert("존재하지 않는 아이디입니다.");
  return;
}

const userData = JSON.parse(storedUser);
if (userData.password !== hashedPassword) {
  alert("비밀번호가 일치하지 않습니다.");
  return;
}

alert(username + "님, 로그인 성공!");
localStorage.setItem("savedUsername", username);
window.location.href = 'welcome.html';
});
```

환영 메시지를 띄우고

savedUsername이라는 이름으로 로그인한 사용자의 아이디를 localStorage에 저장

그리고 welcome.html 페이지로 이동

코드 확인

code check

signup code

```
<div class="center-panel">
  <div class="signup-box">
    <h2>회원가입</h2>
    <form id="signupForm">
      <input type="text" id="signupUsername" placeholder="아이디" required>
      <input type="password" id="signupPassword" placeholder="비밀번호" required>
      <input type="password" id="signupPassword1" placeholder="비밀번호 확인" required>
      <button type="submit">가입하기</button>
    </form>
    <a href="login.html">로그인으로 돌아가기</a>
  </div>
</div>
```

login.html 으로 돌아가는 코드

코드 확인

code check

signup code

```
<script>  
const username = localStorage.getItem('savedUsername');  
  
if (!username) {  
  alert("로그인이 필요합니다.");  
  window.location.href = 'login.html';  
} else {  
  document.getElementById('welcomeMessage').innerText = `${username}님, 환영합니다!`;  
}
```

저장된 유저이름을 가져오기

코드 확인

code check

welcome code

```
<script>
  const username = localStorage.getItem('savedUsername');

  if (!username) {
    alert("로그인이 필요합니다.");
    window.location.href = 'login.html';
  } else {
    document.getElementById('welcomeMessage').innerText = `${username}님, 환영합니다!`;
  }
}
```

바로 welcome.html 에 접속을 차단
아니라면 접속 허용

코드 확인

code check

memo code

```
const translations = {
  ko: {
    memos: '메모',
    trash: '휴지통',
    searchPlaceholder: '메모 검색...',
    newMemo: '새 메모',
    untitled: '제목 없음',
    empty: '내용이 없습니다.',
    save: '저장',
    cancel: '취소',
    titlePlaceholder: '제목을 입력하세요...',
    contentPlaceholder: '내용을 입력하세요...',
    selectMemo: '메모를 선택하거나 새 메모를 만들어보세요',
    noMemos: '메모가 없습니다',
    noTrash: '휴지통이 비어있습니다',
    restore: '복원',
    deleteForever: '영구 삭제',
    moveToTrash: '휴지통으로 이동',
    confirmDelete: '정말로 영구 삭제하시겠습니까?',
    restored: '복원되었습니다',
    movedToTrash: '휴지통으로 이동했습니다',
    appTitle: '노션 스타일 메모 앱'
  },
}
```

한국어로 번역된 모습

코드 확인

code check

memo code

```
en: {
  memos: 'Notes',
  trash: 'Trash',
  searchPlaceholder: 'Search notes...',
  newMemo: 'New Note',
  untitled: 'Untitled',
  empty: 'No content.',
  save: 'Save',
  cancel: 'Cancel',
  titlePlaceholder: 'Enter title...',
  contentPlaceholder: 'Enter content...',
  selectMemo: 'Select a note or create a new one',
  noMemos: 'No notes',
  noTrash: 'Trash is empty',
  restore: 'Restore',
  deleteForever: 'Delete Forever',
  moveToTrash: 'Move to Trash',
  confirmDelete: 'Are you sure you want to delete forever?',
  restored: 'Restored',
  movedToTrash: 'Moved to trash',
  appTitle: 'Notion-Style Memo App'
}
```

영어로 번역된 모습

감사
합니다

103110이현덕