



---

# Term-Project

## 언어 학습 사이트

과목명	모바일 프로그래밍
분반	1 분반
교수	문일영
학번	2018136045
이름	박 형 진
제출일	2023년 12월 2일 토요일

# 목 차

## I. 서론

- 1. 프로젝트 목표 및 요구사항 ..... 1
- 2. 주제 및 주제 선정 동기 ..... 1

## II. 본론

- 1. 주요 기능소개 PC버전 ..... 2
  - 1) 헤더 네비게이션 기능 ..... 2
  - 2) 무작위 단어 제공 기능 ..... 3
  - 3) 네이버 히라가나 테이블 제공 기능 ..... 4
  - 4) 언어 번역 기능 ..... 5
  - 5) 언어 듣기 기능 ..... 6
  - 6) 메일 보내기(피드백) 기능 ..... 7
- 2. 주요 기능소개 모바일 버전(반응형) ..... 8
  - 1) 헤더 네비게이션 기능 ..... 8
  - 2) 무작위 단어 제공 기능 ..... 9
  - 3) 네이버 히라가나 테이블 제공 기능 ..... 10
  - 4) 언어 번역 기능 ..... 11
  - 5) 언어 듣기 기능 ..... 12
  - 6) 메일 보내기(피드백) 기능 ..... 13
- 3. 주요 기능 구현 방법 ..... 14
  - 1) 헤더 네비게이션 기능 ..... 14
  - 2) 무작위 단어 제공 기능 ..... 15
  - 3) 네이버 히라가나 테이블 제공 기능 ..... 16
  - 4) 언어 번역 기능 ..... 17
  - 5) 언어 듣기 기능 ..... 19
  - 6) 메일 보내기(피드백) 기능 ..... 20

## III. 결론

- 1. 추후 개발 계획 ..... 21
- 2. 소감 ..... 21

# I. 서론

## 1. 프로젝트 목표 및 요구사항

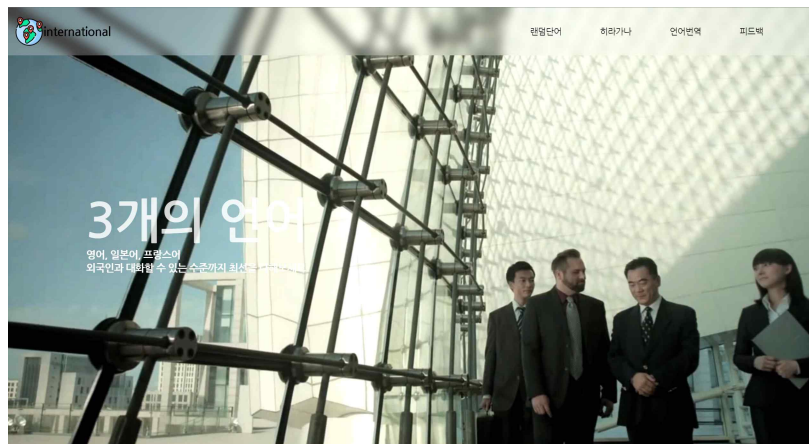
본 보고서는 2023학년도 한국기술교육대학교 컴퓨터 공학부 모바일 프로그래밍 교과목에서 요구된 텀프로젝트에 대하여, "반응형 웹사이트"를 상세히 알리는 것을 목적으로 한다. 본 프로젝트의 목표는 학기 중에 배운 HTML, CSS, Javascript 등을 통해 사용자에게 여러 서비스를 제공하는 반응형 웹사이트를 구축하는 것이다.

### 프로젝트 요구사항

- 해상도에 따라 웹과 모바일 화면으로 각각 변하는 반응형 웹앱으로 개발한다.
- 사용자에게 여러 서비스를 제공해주는 기능들을 제공한다.
- 서비스를 이용할 수 있도록 사용자 인터페이스 구현한다.

## 2. 주제 및 주제 선정 동기

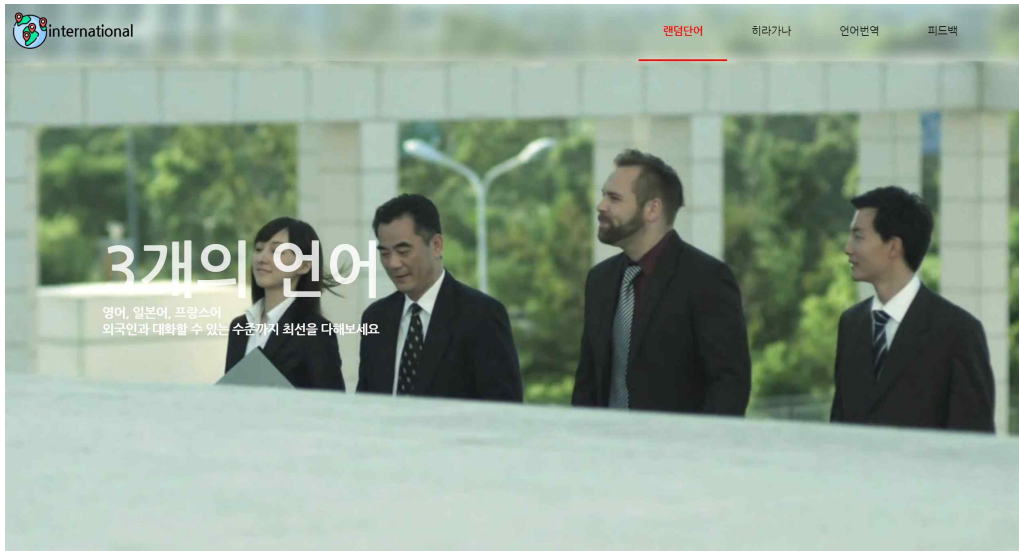
본 프로젝트의 웹사이트 주제는 언어를 영어, 일본어, 프랑스어 총 3개의 언어를 공부할 수 있는 웹사이트이다. 21세기 글로벌 시대에 세계와의 소통은 어떤 분야에서든 매우 중요한 요소로 자리 잡고 있다. 그러나 많은 사람들은 자신의 모국어 외에는 다른 언어를 구사하기 어려운 경우가 많다. 이에 따라, 번역 기술과 TTS(Text To Speech) 등을 활용하여 영어, 일본어, 프랑스어를 체험하고 학습할 수 있는 웹사이트를 개발하고자 한다. 이 웹사이트를 통해 이용자들은 보다 쉽게 언어를 습득할 수 있을 것으로 기대된다.



## II. 본론

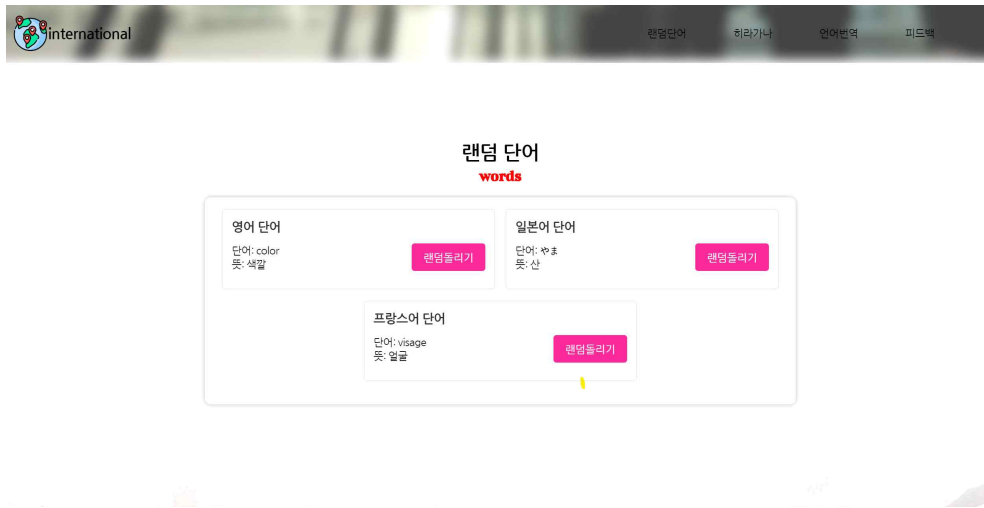
### 1. 주요 기능소개 PC버전

#### 1) 헤더 네비게이션 기능

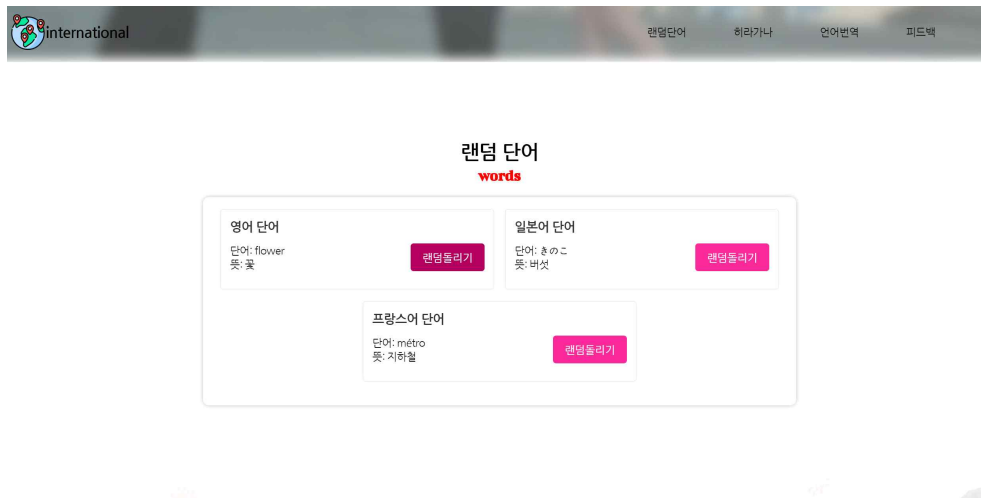


위의 사진과 같이 마우스를 각 네비게이션 리스트에 갖다 대면 빨간 글씨와 빨간 줄이 생기도록 구현하였다. 저 각각의 네비게이션 리스트를 누르면 밑에 각각의 섹션으로 이동되도록 구현하였다.

## 2) 무작위 단어 제공 기능



영어 단어와 일본어 단어, 프랑스어 단어 각각 랜덤돌리기 버튼을 클릭하면 새로운 단어가 나오도록 구현하였다.



실제로 버튼을 클릭하여 새로운 단어가 출력된 모습이다.

### 3) 네이버 히라가나 테이블 제공 기능



international

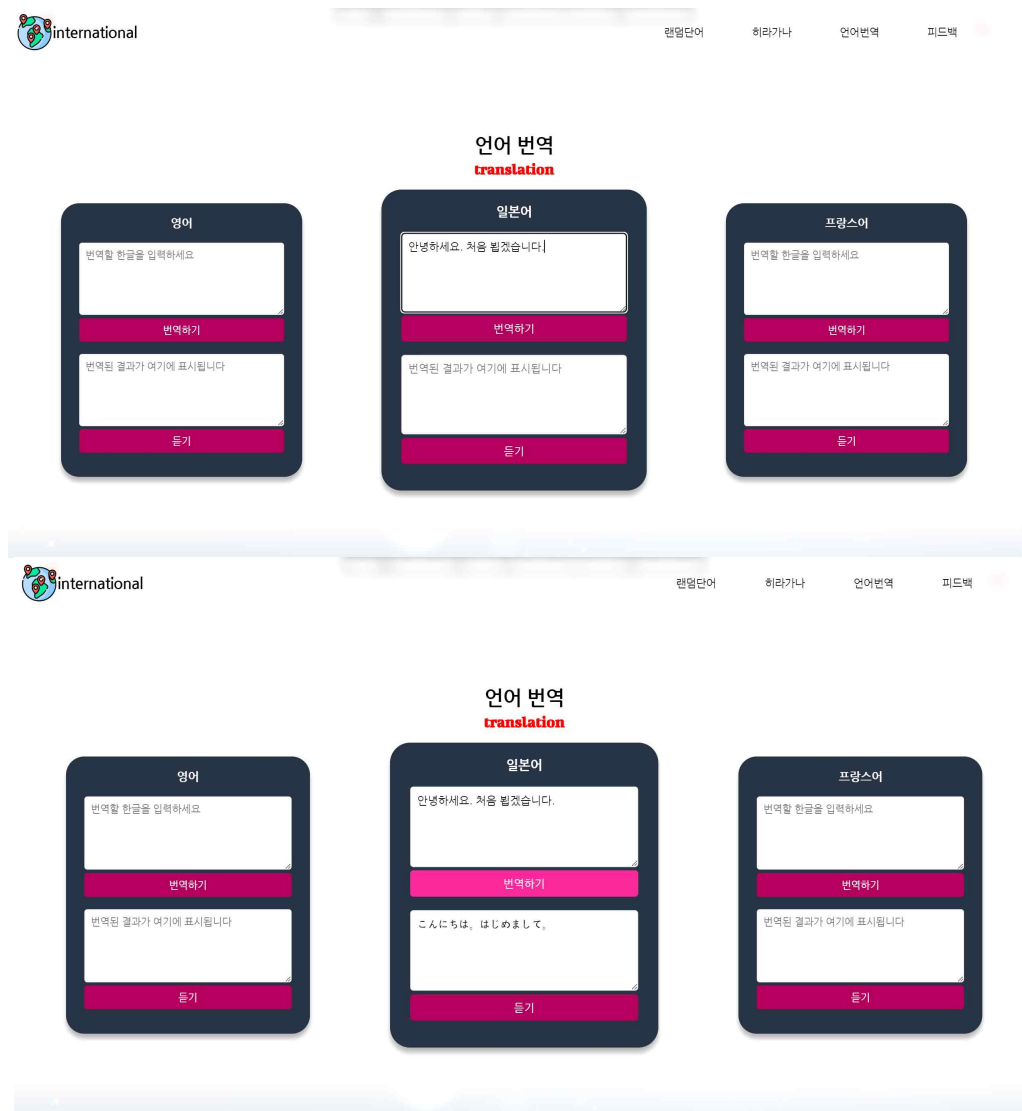
한글단어 히라가나 언어번역 피드백

히라가나  
study

	あ	い	う	え	お
あ	ka	ki	ku	ke	ko
か	ga	gi	gu	ge	go
が	sa	shi	su	se	so
さ	za	ji	zu	ze	zo
ざ	ta	chi	tsu	te	to
た	da	chi	du	de	do
だ					
	na	ni	nu	ne	no
な	ha	hi	fu	he	ho
は	ba	bi	bu	be	bo
ば	pa	pi	pu	pe	po
ぱ	ma	mi	mu	me	mo
ま	ya	ye	yu		yo
や	ra	ri	ru	re	ro
ら	wa				wo
わ					

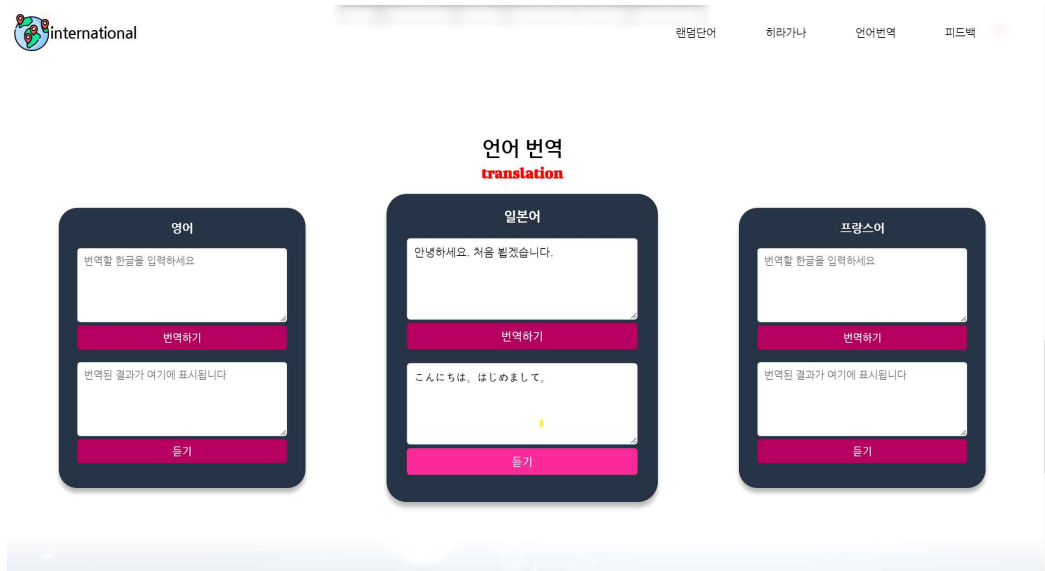
실제로 네이버에 히라가나 테이블을 치면 나오는 히라가나 테이블을 웹스 크래핑 기술을 통해 가져온 것이다. 이 섹션에서는 background-image로 벚꽃 일러스트와 이를 그레이디언트 효과로 투명도를 위로 갈수록 투명하도록 자연스럽게 맞춰주었다.

#### 4) 언어 번역 기능



일본어에서 “안녕하세요. 처음 뵙겠습니다.” 라고 입력하고 번역하기 버튼을 클릭하여 실제로 일본어로 번역된 것을 확인할 수 있다.

## 5) 언어 듣기 기능



위에서 번역한 일본어를 듣기 버튼을 통해 젊은 일본인 여성의 음성 발음을 들을 수 있는 것을 확인할 수 있다. 다른 언어들도 마찬가지로 정상적으로 동작한다.



## 6) 메일 보내기(피드백) 기능

international

랜딩단어

하러가나

언어번역

피드백

메일 보내기  
contact

Name  
이름 입력

Email  
이메일 입력

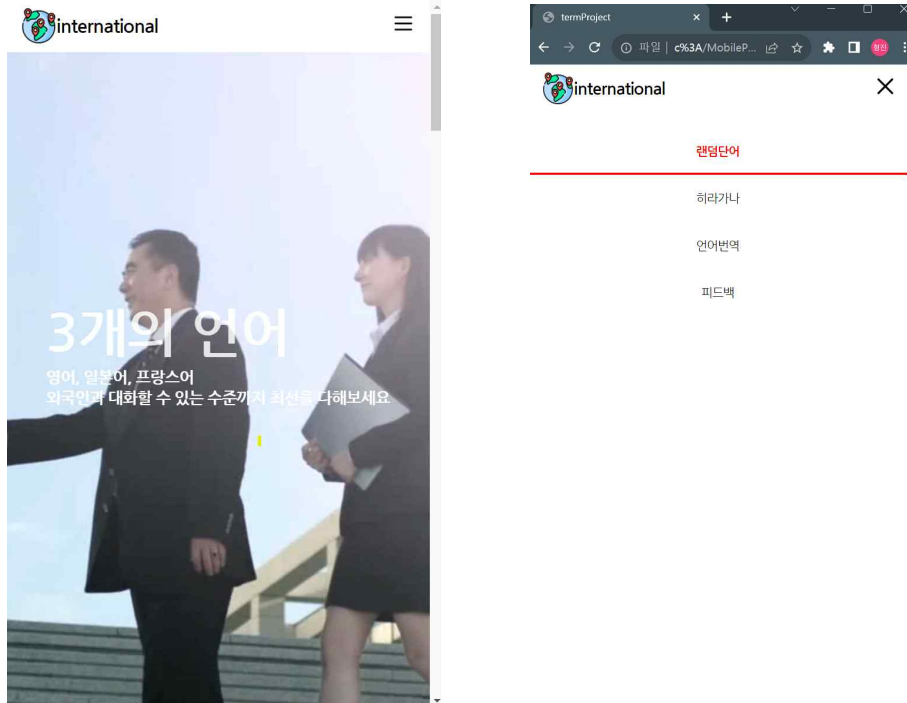
Message

제출하기

사용자는 본인의 이름과 이메일을 입력하고 하고 싶은 말을 메시지로 입력하여 제출하기 버튼을 누르면 자동으로 내 메일인 [bark000104@koreatech.ac.kr](mailto:bark000104@koreatech.ac.kr)로 전송된다.

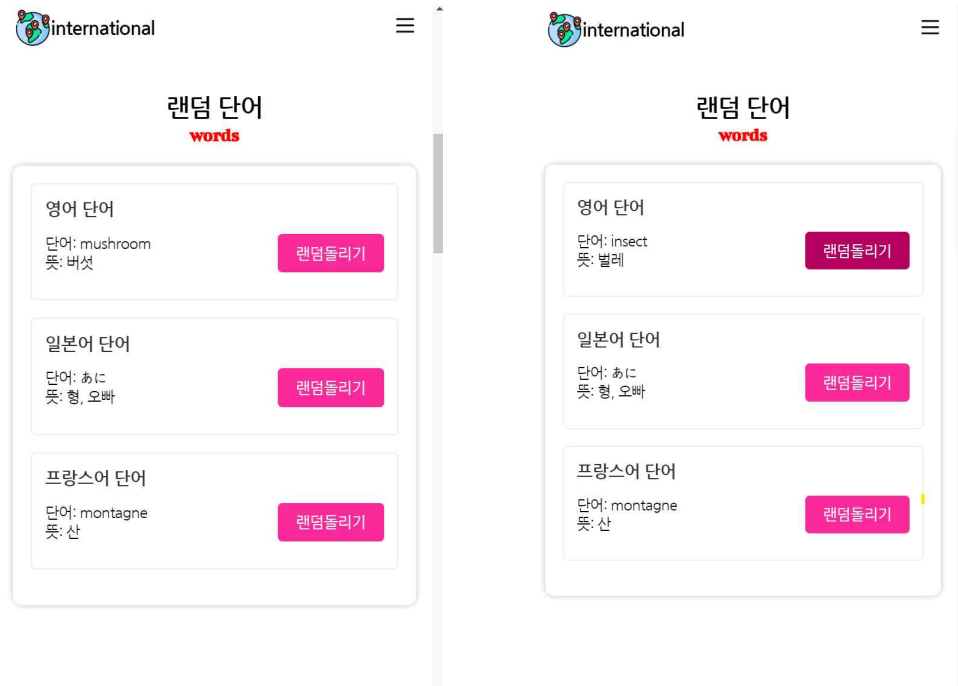
## 2. 주요 기능소개 모바일 버전(반응형)

### 1) 헤더 네비게이션 기능




모바일 버전에서는 햄버거 바로 메뉴바를 선택할 수 있도록 하였다. 햄버거 바를 클릭하면 오른쪽 사진과 같이 메뉴가 나타나고 해당 사진은 랜덤단어를 선택하려고 마우스를 갖다 댄 상태이다. 선택하면 랜덤단어 섹션으로 이동한다.


## 2) 무작위 단어 제공 기능



실제로 영어 단어의 랜덤돌리기 버튼을 클릭하여 새로운 단어가 출력된 것을 확인할 수 있다.

### 3) 네이버 히라가나 테이블 제공 기능


international



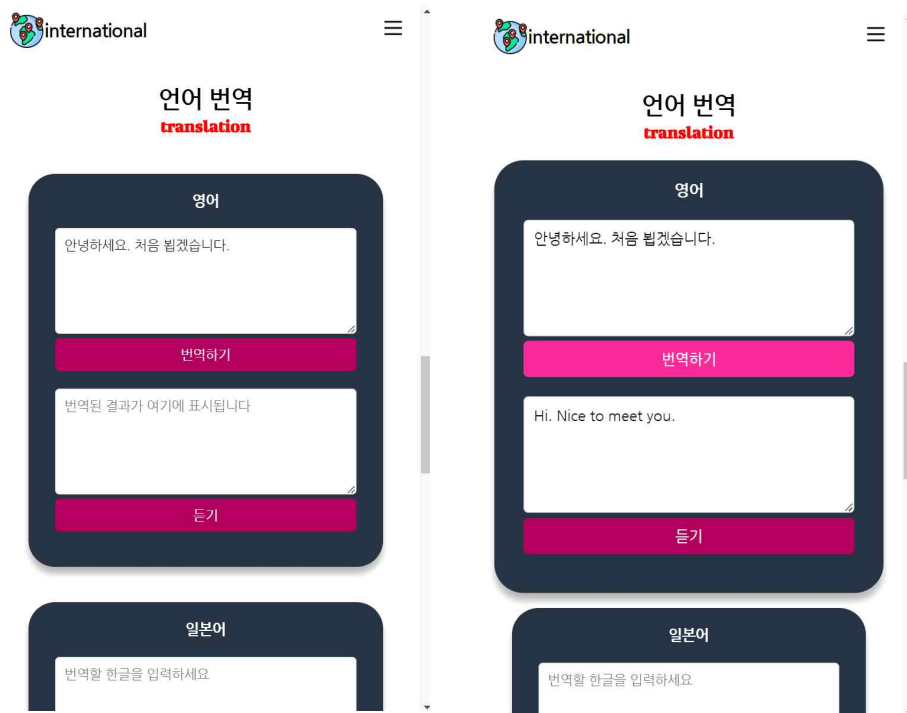
## 히라가나

study

	ㄱ		ㄴ		ㄷ		ㄹ		ㅁ	
ㄱ	あ [a]	か [ka]	い [i]	き [ki]	う [u]	く [ku]	え [e]	け [ke]	お [o]	こ [ko]
ㄴ	あ [a]	が [ga]	い [i]	ぎ [gi]	う [u]	ぐ [gu]	え [e]	げ [ge]	お [o]	ご [go]
ㄷ	さ [sa]	し [si/shi]	す [su]	せ [se]	そ [so]					
ㄹ	ざ [za]	じ [zi/ji]	ず [zu]	ぜ [ze]	ぞ [zo]					
ㅁ	た [ta]	ち [ti/chi]	つ [tu/tsu]	て [te]	と [to]					
ㄷ	だ [da]	ぢ [di/ji]	づ [du/zu]	で [de]	ど [do]					
축			っ							
ㄴ	な [na]	に [ni]	ぬ [nu]	ね [ne]	の [no]					
ㅎ	は [ha]	ひ [hi]	ふ [hu/fu]	へ [he]	ほ [ho]					
ㅁ	ば [ba]	び [bi]	ぶ [bu]	べ [be]	ぼ [bo]					
ㅁ	ぱ [pa]	ぴ [pi]	ぷ [pu]	ぺ [pe]	ぽ [po]					
ㅁ	ま [ma]	み [mi]	む [mu]	め [me]	も [mo]					
야	や [ya]	や [lya]		ゆ [yu]	ゆ [lyu]			よ [yo]	よ [lyo]	
ㄹ	ら [ra]	り [ri]	る [ru]	れ [re]	ろ [ro]					
와	わ [wa]	わ [lwa]						を [wo]		
	ん									

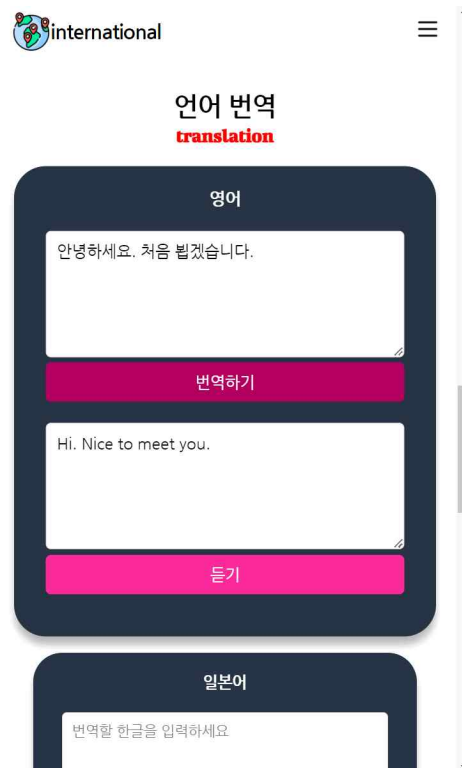
모바일 버전에서도 화면이 깨지지 않고 반응형으로 테이블이 잘 나오는 것을 확인할 수 있다.

#### 4) 언어 번역 기능



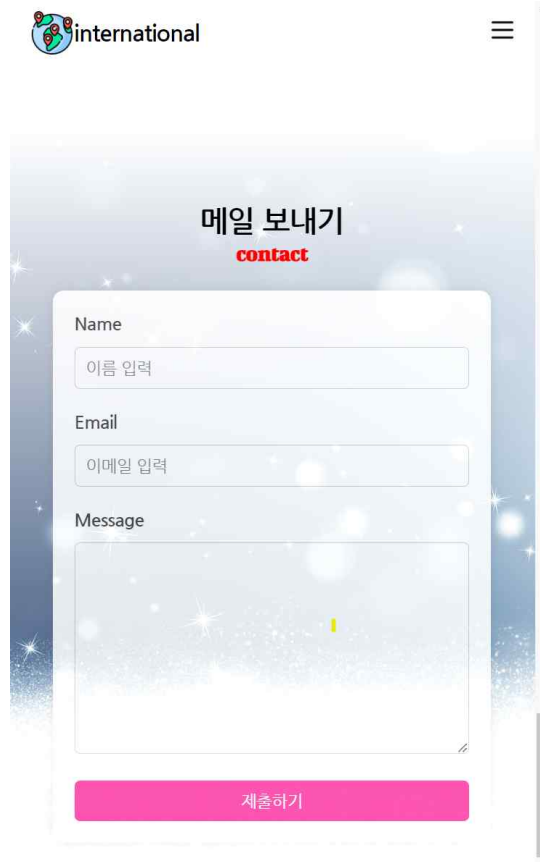
PC 버전에서는 3개의 카드가 가로로 연달아서 나왔지만, 모바일 버전에서는 세로로 연달아서 나오도록 반응형으로 구현하였다.

## 5) 언어 듣기 기능



듣기 역시 번역한 결과를 가지고 듣기 버튼을 클릭하면 잘 들리는 것을 확인할 수 있다.

## 6) 메일 보내기(피드백) 기능



The image shows a mobile application interface for sending an email. At the top, there is a logo with a globe icon and the text 'international', and a hamburger menu icon. The main title is '메일 보내기' (Send Email) in bold black text, with the word 'contact' in red below it. The form contains three input fields: 'Name' with a placeholder '이름 입력', 'Email' with a placeholder '이메일 입력', and 'Message' with a large text area and a yellow cursor. At the bottom is a pink button labeled '제출하기' (Submit).

반응형으로 PC 버전에서보다 크기가 더 작아지고 글씨 크기 역시 작아진 것을 확인할 수 있다.

### 3. 주요 기능 구현 방법

#### 1) 헤더 네비게이션 기능

```
const menuBtn = document.querySelector(".mobileMenuBtn");
const headerNav = document.querySelector(".headerNav");

menuBtn.addEventListener("click", () => {
  menuBtn.classList.toggle("active");
  headerNav.classList.toggle("active");
});

for (let i = 1; i <= 4; i++) {
  const headerLink = document.querySelector(`#headerLink${i}`);

  headerLink.addEventListener("click", () => {
    menuBtn.classList.toggle("active");
    headerNav.classList.toggle("active");
  });
}
```

모바일 버전에서 메뉴 대신 토글 버튼이 생겼을 때 토글 버튼을 클릭할 시 메뉴가 보이도록 하기 위한 자바스크립트 코드이다. 여기서 각각 메뉴 리스트를 클릭했을 시 해당 섹션으로 이동하고 다시 메뉴판을 감추기 위한 작업도 진행한다.



## 2) 무작위 단어 제공 기능

```
// 영어 단어와 뜻 배열
const english_word = [
  { word: 'love', meaning: '사랑' },
  { word: 'face', meaning: '얼굴' },
  { word: 'lie', meaning: '거짓말' },
  { word: 'subway', meaning: '지하철' },
  { word: 'cat', meaning: '고양이' },
  { word: 'age', meaning: '나이' },
  { word: 'mushroom', meaning: '버섯' },
  { word: 'sea', meaning: '바다' },
  { word: 'color', meaning: '색깔' },
  { word: 'brother', meaning: '형, 오빠' },
  { word: 'insect', meaning: '벌레' },
  { word: 'rain', meaning: '비' },
  { word: 'flower', meaning: '꽃' },
  { word: 'person', meaning: '사람' },
  { word: 'mountain', meaning: '산' },
];

/* 랜덤으로 영어 단어 출력 */
function showRandomEnglishWord() {
  // 랜덤으로 인덱스 선택
  const randomIndex = Math.floor(Math.random() * english_word.length);

  // 선택된 단어와 뜻 표시
  const wordElement = document.getElementById('english_word');
  const meaningElement = document.getElementById('english_meaning');

  wordElement.textContent = english_word[randomIndex].word;
  meaningElement.textContent = english_word[randomIndex].meaning;
}

showRandomEnglishWord();
```

영어 단어와 뜻이 배열로 저장되어 있고 랜덤돌리기 버튼을 클릭하면 showRandomEnglishWord() 함수가 실행되도록 설계했다. 함수 내부에서는 표시할 html의 id를 얻어와서 랜덤으로 선택된 단어가 표시되도록 구현했다.

### 3) 네이버 히라가나 테이블 제공 기능

```
app.get('/api/hiragana', async (req, res) => {
  try {
    const searchURL =
      'https://search.naver.com/search.naver?where=nexearch&sm=top_hy&fbm=0&ie=utf8&query=%ED%9E%88%EB%9D%BC%EA%80%80%EB%82%98+%ED%91%9C';
    const response = await axios.get(searchURL);
    const data = response.data;

    res.send(data);
  } catch (error) {
    console.error('Error:', error);
    res.status(500).send('Internal Server Error');
  }
});
```

Node.js를 활용해 실행시킬 server.js 파일안의 코드이다. 웹스크래핑할 URL 주소를 입력한 뒤 그 데이터를 로컬 자바스크립트 파일으로 넘겨준다.

```
/* 웹 스크래핑 테이블 추출 */
// const alphabetURL = 'http://localhost:3000/api/alphabet';
const hiraganaURL = 'http://localhost:3000/api/hiragana';
// const frenchURL = 'http://localhost:3000/api/french';

// 네이버 검색 결과 페이지에서 테이블을 추출하고 동적으로 생성하는 함수
async function createTableFromNaver(url, tableName) {
  try {
    // 네이버 검색 결과 페이지에서 HTML 가져오기
    const response = await fetch(url);
    const html = await response.text();

    // HTML에서 원하는 테이블 부분 추출
    const parser = new DOMParser();
    const doc = parser.parseFromString(html, 'text/html');
    const tableElement = doc.querySelector('table');

    // 동적으로 생성한 테이블을 표시할 div 요소 찾기
    const resultTableContainer = document.getElementById(tableName);

    // 동적으로 생성한 테이블을 표시
    resultTableContainer.appendChild(tableElement);
  } catch (error) {
    console.error('Error:', error);
  }
}

// 페이지 로드 시 테이블 생성 함수 호출
window.onload = function () {
  const functionsToCall = [
    //() => createTableFromNaver(alphabetURL, "resultTableAlphabet"),
    () => createTableFromNaver(hiraganaURL, 'resultTableHiragana'),
    //() => createTableFromNaver(frenchURL, "resultTableFrench"),
  ];

  functionsToCall.forEach((func) => func());
};
```

로컬 자바스크립트 소스코드에서는 실제로 데이터에서 원하는 부분을 추출 해서 이를 활용할 수 있다. 히라가나 테이블을 추출한 후 내 html의 id를 통해 화면에 표시할 수 있도록 하였다. 서버가 실행되고 있다면 테이블 생성될 때 자동으로 화면에 표시된다.

#### 4) 언어 번역 기능

```
app.post('/translateToEnglish', async (req, res) => {
  const { text } = req.body;
  const clientId = '91jWgWQDh8wydwsCK1yR'; // api key id
  const clientSecret = 'dW4i3ee7Cc'; // api key secret
  const papagoUrl = 'https://openapi.naver.com/v1/papago/n2mt';

  const headers = {
    'Content-Type': 'application/x-www-form-urlencoded',
    'X-Naver-Client-Id': '91jWgWQDh8wydwsCK1yR',
    'X-Naver-Client-Secret': 'pnb82zY9oZ',
  };

  const data = new URLSearchParams({
    source: 'ko',
    target: 'en',
    text: text,
  });

  try {
    const response = await fetch(papagoUrl, {
      method: 'POST',
      headers: headers,
      body: data,
    });

    if (!response.ok) {
      throw new Error('API 호출이 실패했습니다.');
```

Node.js 활용해 파파고 API를 호출하는 코드이다. POST 방식으로 로컬 자바스크립트 파일과 통신한다.

```

async function translateEnglishText() {
  const inputText = document.getElementById('inputEnglishText').value;

  if (inputText.trim() === '') {
    alert('번역할 텍스트를 입력하세요.');
```

```

    return;
  }

  try {
    const response = await fetch('http://localhost:3000/translateToEnglish', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({ text: inputText }),
    });

    const result = await response.json();

    if (result?.translatedText) {
      document.getElementById('outputEnglishText').value =
        result.translatedText;
    } else {
      alert('번역에 실패했습니다.');
```

```

    }
  } catch (error) {
    console.error('에러:', error);
    alert('번역에 실패했습니다.');
```

```

  }
}

```

번역하기 버튼을 클릭했을 시 동작하는 traslateEnglishText() 함수이다. 서버를 통해 파파고 API를 이용하여 입력된 TextArea의 텍스트를 번역하여 출력용 TextArea에 출력한다.

## 5) 언어 듣기 기능

```
/* Text To Speech */  
function speakEnglishText() {  
  const text = document.getElementById('outputEnglishText').value;  
  
  if (text.trim() === '') {  
    alert('텍스트가 없습니다.');    return;  
  }  
  
  const utterance = new SpeechSynthesisUtterance(text);  
  utterance.lang = 'en-US';  
  utterance.rate = 0.8;  
  
  // 음성 재생  
  window.speechSynthesis.speak(utterance);  
}
```

TTS 기술은 브라우저에 내장되어 있는 SpeechSynthesisUtterance API를 활용하여 따로 서버를 접속할 필요는 없다. 생성된 utterance 객체를 통해 text Area의 텍스트를 들을 수 있다.

## 6) 메일 보내기(피드백) 기능

```
<script type="text/javascript">
  (function () {
    emailjs.init("TD7hQdaBRHsmTdFMq");
  })();
</script>
```

```
/* 메일 보내기 */
function sendMail() {
  var params = {
    name: document.getElementById('name').value,
    email: document.getElementById('email').value,
    message: document.getElementById('message').value,
  };

  const serviceID = 'service_hyeong';
  const templateID = 'template_flzqocd';

  emailjs
    .send(serviceID, templateID, params)
    .then((res) => {
      document.getElementById('name').value = '';
      document.getElementById('email').value = '';
      document.getElementById('message').value = '';
      alert('메일 전송이 완료되었습니다.');
```

이메일 기능은 emailjs API를 이용하였다. 위의 코드는 API를 사용하는 코드이다.

메일을 보내면 위의 형식에 맞춰서 [bark000104@koreatech.ac.kr](mailto:bark000104@koreatech.ac.kr)로 전송되는 것을 확인할 수 있다.

## Ⅲ. 결론

### 1. 추후 개발 계획

Node.js를 활용하여 서버를 실행시켰고 이를 통해 번역 기능과 웹 스크래핑 기술을 활용할 수 있었다. 자바스크립트의 배열에 저장된 3개 언어의 단어들을 json으로 활용하여 완성도를 높이고 싶다. CSS를 통해 최대한 이쁘게 꾸미고자 노력했지만, 아직 많이 부족하다고 생각된다. 더 노력해서 자바스크립트를 더욱 활용해 더 생동감 있는 웹사이트를 구현해보고 싶다. 여러 기능을 최대한 넣어보고자 노력했고, 결과물로 잘 나온 것 같아 만족스럽다. 앞으로도 여기서 안주하지 않고 세세한 기능들, 예로 들자면 퀴즈에 대한 점수제공 기능 등과 같은 여러 기능을 더 넣어서 프로젝트의 완성물을 더 높이고 싶다.

### 2. 개인적 소감

강의시간에 배웠던 미디어 쿼리 등의 CSS 지식과 HTML, 자바스크립트 등을 직접 활용하며 반응형 웹앱을 만들어가면서 더 확실하게 내 지식으로 흡수할 수 있었다. 앱이나 GUI 프로그램 개발은 많이 해봤지만, 웹 프로그램 제작은 사실상 처음이나 다른 없어서 처음에는 막막하기도 하고 도중에 막히는 부분도 많았지만, 따로 Node.js도 공부하여 서버까지 연결해보기도 하는 등 끝내 완성까지 해보면서 웹 프론트엔드, 백엔드의 실력 모두 업그레이드한 좋은 계기가 되었다.