

MBTI 프로젝트 만들기

<https://zzzmini-mbti.netlify.app/>

1. github에서 관리할 Project 만들기

- 전체 프로젝트의 형상관리를 위해 git hub을 사용하며, git hub과 vscode 연결을 위한 소프트웨어로 gitKraken 을 사용합니다.

① MBTI Project 생성하기

#1-프로젝트 기본구조 만들기

- 기본구조 만들기
- index.html 만들기

#2. 폰트 설정 및 브라우저 css 설정

- 브라우저 마다 다른 CSS 표현방식을 통일
- pretendard github에서 저작권 문제없는 폰트 사용해 보기

#3. 메인페이지 구조만들기

- index.html 조정하기
- 공통 사용 css 생성하기

#4. 메인페이지 CSS만들기

- home.css 만들기

#5. 질문페이지 만들기

- question.html 생성하고 구현하기
- 질문페이지 css 작업하기

#6. js로 질문보여주고 답하기

- 질문에 답하는 JavaScript 생성
- question.js 생성하고 연결하기
- 화면에 질문보여주기
- ProgressBar 만들기

#7. 결과페이지 만들기

- 결과페이지를 위한 html 과 css 생성
- results.html 생성
- results.css 생성

#8. 질문결과 코딩

- 질문에 맞는 결과를 보여주도록 코딩하기
- result.js 생성하고 코딩하기

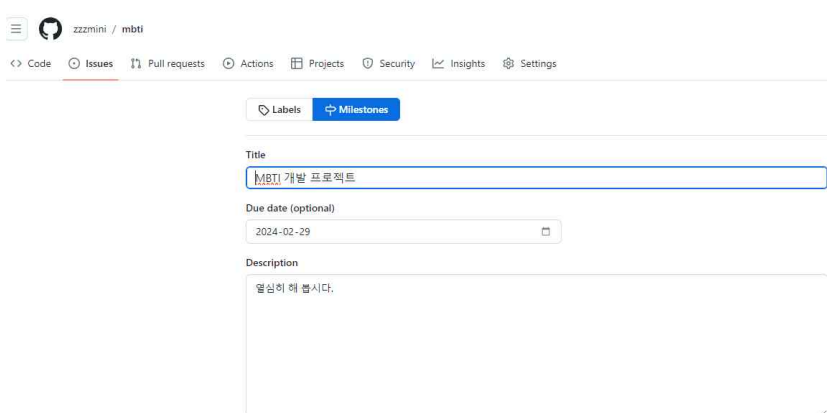
#9. 웹사이트에 업로드하기

- 실제 웹사이트에 업로드하기

② github에 프로젝트 카드와 Todo List 만들기

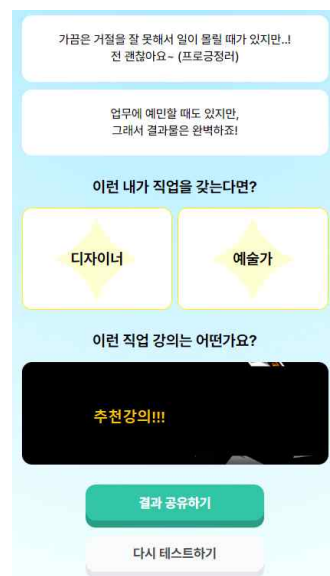
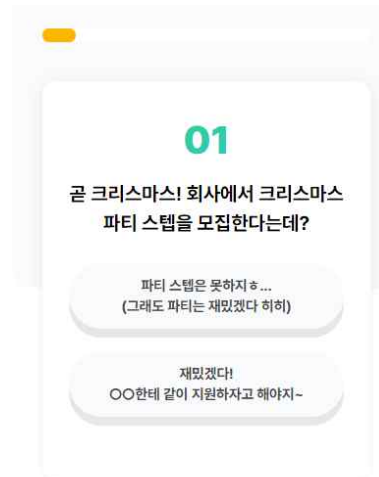
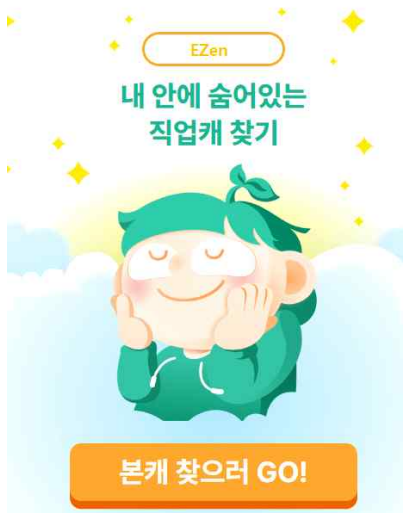


③ 마일스톤 만들기



2. 프로젝트 구조 및 실행화면

① 실행화면



② data.js 내부 구조

※ export : 자바스크립트 파일 내에서 정의된 변수명을 외부(다른 js 파일에서) 사용 시)로 내보낼 때 사용하는 키워드

```
export const questions = [
  {
    number: '01',
    question: '곧 크리스마스! 회사에서 크리스마스 파티 스템을 모집한다는데?',
    choices: [
      { text: '파티 스템은 못하지ㅎ...<br>(그래도 파티는 재밌겠다 히히)', value: 'i' },
      { text: '재밌겠다!<br>○○한테 같이 지원하자고 해야지~', value: 'e' }
    ]
  }
]
```

```
]
},
{
  number: '02',
  question: '해외 출장이 잡혔다!<br>전날 잠들기 전 나는?',
  choices: [
    { text: '짐 다 챙겼고 일찍 자야지~', value: 's' },
    { text: '다 챙겼겠지?<br>여분 옷을 좀 더 챙길까..(뒤척뒤척)', value: 'n' }
  ]
},
```

=====

```
export const results = [
  {
    title: '출근 시간은<br>내가 정해!',
    character: '/images/result_character1.png',
    results: [
      '혼자서 멘탈 케어가 가능해요!<br>소속이 없어도, 별이가 불안정해도 저는 지금이 좋아요!',
      '일에 대한 욕심이 많아요.<br>행복한 야근이라고 할 수 있어요!',
      '자유로운 환경에서 더 잘해요!<br>하지만 누구보다 스스로 시간 관리가 철저해요:)',
      '리더십 있다는 소리를 자주 들어요.<br>추진력이 있는 편이죠~ (뿌듯)'
    ],
    jobs: ['프리랜서', '창업가'],
    lectureImg: '/images/result_lecture1.png',
    lectureUrl: 'https://bit.ly/3Wr0kt6'
  },
],
```

=====

```
export const mbtis = {
  entj: 0,
  entp: 0,
```

3. 브라우저 및 폰트 설정하기

① 브라우저마다 다른 css 표현방식 통일하기

- 구글검색 : reset .css cdn → <https://www.jsdelivr.com/package/npm/reset-css> →



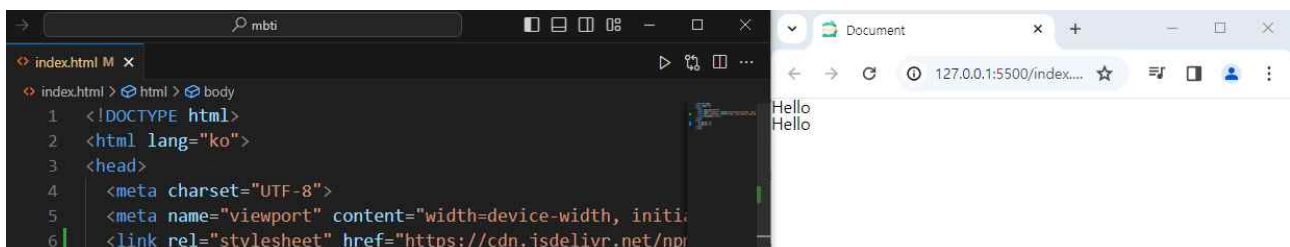
선택 후 HTML Copy하기 → index.html에 적용하기

- 변환 전 : 적용 전



- 적용 후 :

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/reset-css@5.0.2/reset.min.css">



② 웹폰트 사용하기

- 사용이유 : 인터넷 속도를 고려해 경량화 된 폰트로 저작권의 문제가 없음.
- 구글검색 : pretendard github → <https://github.com/orioncactus/pretendard> → 아래쪽 『가변 다이나믹 서브셋』 찾기 → HTML 하단 코드 복사 후 사용

가변 다이나믹 서브셋

가변 다이나믹 서브셋으로 가변 Weight 속성과 함께 기존 다이나믹 서브셋보다 현저히 적은 용량으로 Pretendard를 사용할 수 있습니다. 모던 브라우저에서 더욱 쾌적하게 Pretendard를 사용하려면 아래 코드를 사용하세요. 사용하는 font-family 이름은 "Pretendard Variable" 입니다.

HTML

```
<link rel="stylesheet" as="style" crossorigin href="https://cdn.jsdelivr.net/gh/orioncactus/pretendard@
```



```
<script defer src="/js/test.js"></script>
```

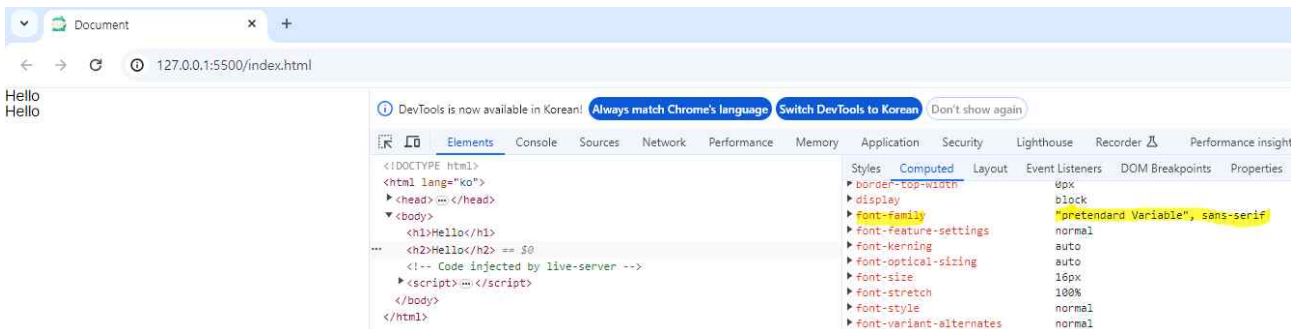
- 적용할 css 생성하고 연결하기 → css 폴더생성 → common.css 파일 생성

```
body{  
  font-family: "pretendard Variable", sans-serif;  
}
```

→ index.html에 css 연결하기

```
<link rel="stylesheet" href="/css/common.css">
```

→ 개발도구에서 확인하기



4. 메인페이지 구도 정하기



① index.html 코딩하기

```
<body>
  <h2 class="page-subtitle">
    E-Zen Academy
  </h2>
  <h1 class="page-title">
    내 안에 숨어있는 <br>
    직업캐 찾기!
  </h1>
  
  <a href="/question.html" class="btn btn-orange">
    본캐 찾으러 Go!
  </a>
</body>
```

② common.css

```
body{
  font-family: "pretendard Variable", sans-serif;
}

.page-title{
  font-size: 50px;
  font-weight: 800;
  line-height: 1.3;
  color: #13B491;
  text-align: center;
  margin-bottom: 20px;
}

.page-subtitle{
  width: 240px;
  height: 50px;
  border: 5px solid #FFA32D;
  display: flex;
  justify-content: center;
  align-items: center;
  font-size: 24px;
  font-weight: 700;
  color: #FFA32D;
  /* page 가운데 정렬로 너비가 정해지면 자동으로 가운데 정렬된다. */
  margin-left: auto;
  margin-right: auto;
  margin-bottom: 20px;
}

/* 모든 버튼의 기본이 되는 css */
.btn{
  max-width: 500px;
  height: 100px;
  display: flex;
  justify-content: center;
  align-items: center;
  font-size: 50px;
  font-weight: 700;
```



```
/* a 태그 사용시 생기는 밑줄 제거 */
text-decoration: none;
cursor: pointer;
border-radius: 20px;
margin-bottom: 35px;
margin-left: auto;
margin-right: auto;
/* 전체 버튼 눌리는 효과 지정 */
transition: 0.1s;
}

/* 기본 버튼에 추가할 css */
.btn-orange{
  background-color: #FFA32D;
  box-shadow: 0 14px 0 #E66000;
  color: #FFFFFF;
}

/* 버튼 눌림 효과 만들기 */
.btn-orange:active{
  box-shadow: 0 4px 0 #E66000;
}

.btn-green{
  background-color: #32bfa1;
  box-shadow: 0 14px 0 #299a82;
  color: #ffffff;
}

.btn-green:active{
  box-shadow: 0 14px 0 #299a82;
  transform: translate(10px);
}

.btn-small{
  max-width: 350px;
  height: 70px;
  font-size: 26px;
  margin-bottom: 30px;
}
```

```
.btn-gray{
  background-color: #f2f4f6;
  box-shadow: 0 14px 0 #dfe1e2;
  color: #444444;
}

.btn-gray:active{
  box-shadow: 0 4px 0 #dfe1e2;
  transform: translate(10px);
}
```

5. 메인페이지 CSS 만들기

① index.html 코딩하기

```
<link rel="stylesheet" href="/css/home.css">
```

② home.css 코딩하기

```
body{
  background-color: #f6fcff;
  background-image: url("../images/main_bg.jpg");
  background-position: top center;
  background-repeat: no-repeat;
}

/* 중앙이미지를 정가운데 위치시키기 */
/* 화면크기(가로)가 줄어들때 화면너비를 100% 이용해서 */
/* 자동으로 이미지 크기를 줄이기 */
.character{
  width: 100%;
  max-width: 364px;
  margin-left: auto;
  margin-right: auto;
  /* 이미지는 글자요소라 중앙정렬이 안됨. */
  /* 상자요소로 전환해서 처리 */
  /* display: block; */
  margin-bottom: 40px;
}

/* 최근에 와서는 image를 layout을 잡는 상자요소로 사용하는 추세 */
```

```
/* home.css의 display : block; 을 모든 내부에서 사용할 수 있도록*/  
/* common.css로 이동하여 처리한다. */
```

③ common.css 에 추가하기

```
img{  
    display: block;  
}
```

④ 메인화면 수정하기 : common.css

: 화면 크기가 작아질 때 띄어쓰기가 안되는 부분 해결

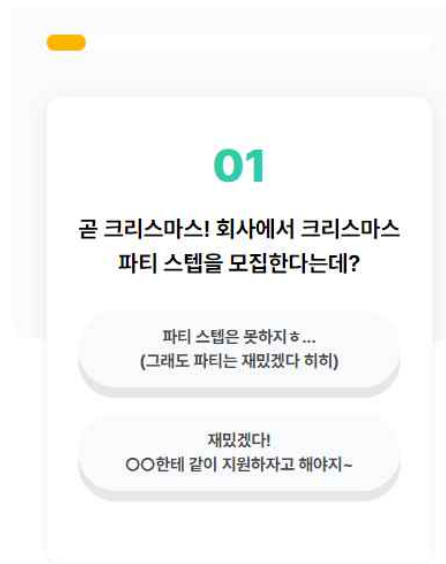
▷ 제목 및 상자(버튼) 안 글씨 정리하기

▷ 맨위/맨아래 버튼 간격조정하기

```
body{  
    font-family: "pretendard Variable", sans-serif;  
    word-break: keep-all;  
    padding-top: 80px;  
    padding-bottom: 80px;  
}
```

6. 질문페이지 구현하기

① question.html 코딩하기



```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/reset-css@5.0.2/reset.min.css">
  <link rel="stylesheet" as="style" crossorigin
href="https://cdn.jsdelivr.net/gh/orioncactus/pretendard@v1.3.9/dist/web/variable/pretendardvariable
-dynamic-subset.min.css" />
  <link rel="stylesheet" href="/css/common.css">
  <title>Document</title>
</head>
<body>
  <div class="progress">
    <div class="value"></div>
  </div>
  <div class="question-box">
    <div class="number"></div>
    <div class="question"></div>
    <div class="btn btn-gray choice choice1"></div>
    <div class="btn btn-gray choice choice2"></div>
  </div>
</body>
```

```
</html>
```

② data.js 첫 질문 확인하기

```
export const questions = [
  {
    number: '01',
    question: '곧 크리스마스! 회사에서 크리스마스 파티 스텝을 모집한다는데?',
    choices: [
      { text: '파티 스텝은 못하지ㅎ...<br>(그래도 파티는 재밌겠다 히히)', value: 'i' },
      { text: '재밌겠다!<br>○○한테 같이 지원하자고 해야지~', value: 'e' }
    ]
  },
],
```

③ data.js 첫 질문을 question.html로 옮기기

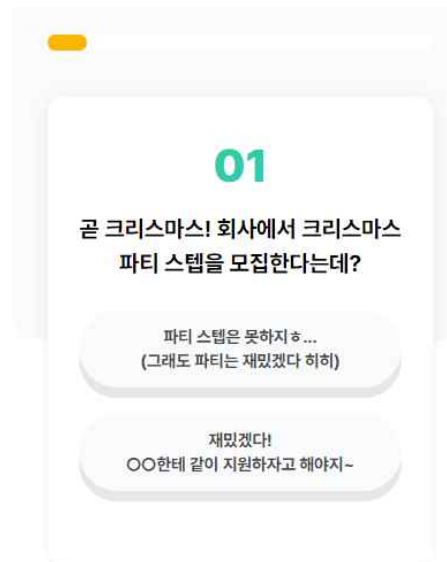
```
<div class="question-box">
  <div class="number">01</div>
  <div class="question">곧 크리스마스! 회사에서 크리스마스 파티 스텝을 모집한다는데?</div>
  <div class="btn btn-gray choice choice1">파티 스텝은 못하지ㅎ...<br>(그래도 파티는 재밌겠다 히히)</div>
  <div class="btn btn-gray choice choice2">재밌겠다!<br>○○한테 같이 지원하자고 해야지~</div>
</div>
```

```
<body>
  <div class="progress">
    <div class="value"></div>
  </div>
  <div class="question-box">
    <div class="number">01</div>
    <div class="question">곧 크리스마스! 회사에서 크리스마스 파티 스텝을 모집한다는데?</div>
    <div class="btn btn-gray choice choice1">파티 스텝은 못하지ㅎ...<br>(그래도 파티는 재밌겠다 히히)</div>
    <div class="btn btn-gray choice choice2">재밌겠다!<br>○○한테 같이 지원하자고 해야지~</div>
  </div>
</body>
```

파티 스텝은 못하지ㅎ...
(그래도 파티는 재밌겠다
히히)
재밌겠다!
○○한테 같이
지원하자고 해야지~

7. 질문페이지 css 작업하기

① question.css 작업



```
body{
  background-image: url("../images/questions_bg.jpg");
  background-position: top center;
  background-repeat: no-repeat;
}
```

```
.progress{
  max-width: 500px;
  height: 20px;
  background-color: #FFFFFF;
  margin-left: auto;
  margin-right: auto;
  border-radius: 10px;
  margin-bottom: 60px;
}
```

```
.progress .value{
  /* 부모인 .progress width의 10% */
  /* progress bar의 크기(길이) 총 10단계 */
  width: 10%;
  height: 20px;
  background-color: #ffb103;
  border-radius: 10px;
}
```

```
/* progress의 애니메이션 효과 */
transition: 0.2s;
}

.question-box{
  max-width: 420px;
  background-color: #FFFFFF;
  padding-top: 60px;
  padding-bottom: 60px;
  padding-left: 40px;
  padding-right: 40px;
  margin-left: auto;
  margin-right: auto;
  box-shadow: 0 4px 25px #dde2e8;
}

.question-box .number{
  font-size: 60px;
  /* 제일 큰 사이즈 */
  font-weight: 900;
  text-align: center;
  color: #2ec7a6;
  margin-bottom: 30px;
}

.question-box .question{
  font-size: 30px;
  font-weight: 700;
  line-height: 1.5;
  text-align: center;
  margin-bottom: 40px;
}

.question-box .choice{
  border-radius: 120px;
  font-size: 22px;
}
```

8. 질문 처리하기

① question.html

- question.js 에서는 동일 폴더위치의 data.js를 import 해서 사용할 예정입니다.
- 위의 경우

```
<!-- 외부자바스크립트를 import 하는 파일을 참조할 때는 type="module"로 써야한다. -->
<script type="module" defer src="/js/question.js"></script>
```

② question.html에 자료 가져오기

```
// data.js에서 가져오기
// data.js 파일에서 배열형태로 가져온다.
import { questions } from './data.js';

const progressValueEl = document.querySelector('.progress .value')
const numberEl = document.querySelector('.number')
const questionEl = document.querySelector('.question')
const choice1El = document.querySelector('.choice1')
const choice2El = document.querySelector('.choice2')

let currentNumber = 0
let mbti = ""

function renderQuestion(){
  const question = questions[currentNumber]
  numberEl.innerHTML = question.number
  questionEl.innerHTML = question.question
  choice1El.innerHTML = question.choices[0].text
  choice2El.innerHTML = question.choices[1].text
  progressValueEl.style.width = (currentNumber + 1)*10 + '%'
}

// 처음 시작화면에서는 첫 질문을 가져와서 뿌려준다.
renderQuestion()
```

② 질문에 답변한 경우 처리

```
// 질문에 대한 첫번째, 두번째 답을 클릭했을 경우
// choices: [
//   { text: '파티 스텝은 못하지ㅎ...<br>(그래도 파티는 재밌겠다 히히)', value: 'i' },
//   { text: '재밌겠다!<br>○○한테 같이 지원하자고 해야지~', value: 'e' }
```



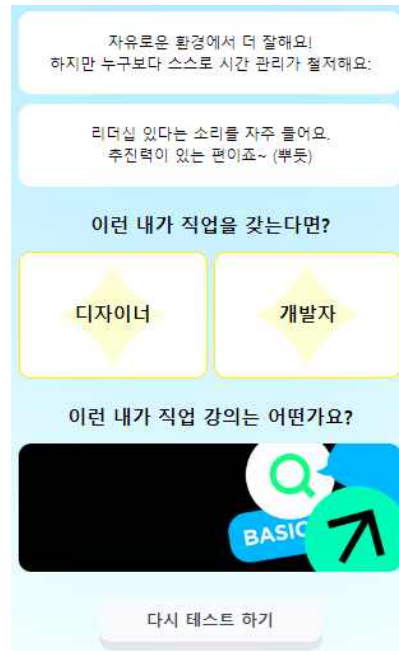
```
// ]
// choices[0] : 첫번째 text 선택 → 첫번째 value 값 가져오기
// choices[1] : 두번째 text 선택 → 두번째 value 값 가져오기
choice1El.addEventListener('click', function(){nextQuestion(0)})
choice2El.addEventListener('click', function(){nextQuestion(1)})

// 1. 선택한 답변을 mbti 변수에 저장
// 2. 다음 질문번호를 생성해서 질문을 뿌려주는 renderQuestion() 함수 호출
function nextQuestion(choiceNumber){
  console.log('nextQuestion Function')
  const question = questions[currentNumber]
  // 만약 맨 마지막 질문이면 결과 보여주기 페이지로 이동
  if(currentNumber === questions.length - 1){
    showResultPage()
    return
  }
  mbti = mbti + question.choices[choiceNumber].value
  currentNumber = currentNumber + 1;
  renderQuestion()
}

function showResultPage(){
  // 쿼리 스트링을 만들어 결과페이지 호출
  location.href = './results.html?mbti=' + mbti
}
```

9. 결과 페이지 만들고 css 적용하기

① 결과페이지 구성



② results.html 생성하기

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="/css/results.css">
  <link rel="stylesheet" href="/css/common.css">
  <title>Document</title>
</head>
<body>
  <h2 class="page-subtitle">
    내 본캐는?!
  </h2>
  <!-- 나중에 data.js에서 가져옴 -->
  <h1 class="page-title">
    출근 시간은 <br> 내가 정해!
  </h1>
  
  <div class="result">
    <div class="box">혼자서 멘탈 케어가 가능해요!<br>소속이 없어도, 별이가 불안정해도 저는 지
```

```

금이 좋아요!</div>
  <div class="box">일에 대한 욕심이 많아요.<br>행복한 야근이라고 할 수 있어요!</div>
  <div class="box">자유로운 환경에서 더 잘해요!<br>하지만 누구보다 스스로 시간 관리가 철저
해요:</div>
  <div class="box">리더십 있다는 소리를 자주 들어요.<br>추진력이 있는 편이죠~ (뿌듯)</div>
</div>
<div class="result">
<h3>이런 내가 직업을 갖는다면?</h3>
  <div class="jobs">
    <div class="job">디자이너</div>
    <div class="job">개발자</div>
  </div>
</div>
<div class="result">
  <h3>이런 내가 직업 강의는 어떤가요?</h3>
  <a href="" target="_blank" class="lecture">
    
  </a>
</div>
<a href="/" class="btn btn-gray btn-small">
  다시 테스트 하기
</a>
</html>

```

③ results.css

```

body{
  background-color: #eaf8ff;
  background-image: url("../images/result_bg.jpg");
  background-position: top center;
  background-repeat: no-repeat;
}

.character{
  width: 100%;
  max-width: 600px;
  margin-left: auto;
  margin-right: auto;
  margin-bottom: 20px;
}

```

```
.result{
  max-width: 600px;
  margin-left: auto;
  margin-right: auto;
}

.result .box{
  min-height: 70px;
  background-color: #ffffff;
  margin-bottom: 15px;
  border-radius: 20px;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 24px;
  font-weight: 500;
  line-height: 1.4;
  text-align: center;
  padding-top: 30px;
  padding-bottom: 30px;
  padding-left: 10px;
  padding-right: 10px;
}

.result h3{
  font-size: 30px;
  font-weight: 700;
  line-height: 1.4;
  text-align: center;
  margin-top: 40px;
  margin-bottom: 20px;
}

.result .jobs{
  height: 200px;
  /* 요소의 자식 요소는 수평정렬됨. */
  display: flex;
  gap: 10px;
}
```

```

.result .job{
  flex: 1;
  background-color: #ffffff;
  background-image: url("../images/result_star.png");
  background-position: center center;
  background-repeat: no-repeat;
  display: flex;
  justify-content: center;
  align-items: center;
  font-size: 30px;
  font-weight: 700;
  line-height: 1.4;
  text-align: center;
  border: 1px solid #ffe603;
  border-radius: 20px;
}

.result .lecture img{
  margin-bottom: 40px;
  width: 100%;
}

```

10. 결과 페이지 js 작업하기

① results.html 수정

```

<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="/css/results.css">
  <link rel="stylesheet" href="/css/common.css">
  <script type="module" defer src="/js/results.js"></script>
  <title>Document</title>
</head>
<body>
  <h2 class="page-subtitle">
    내 본캐는?!

```

```

</h2>
<!-- 나중에 data.js에서 가져옴 -->
<h1 class="page-title">

</h1>

<div class="result">
  <div class="box"></div>
  <div class="box"></div>
  <div class="box"></div>
  <div class="box"></div>
</div>
<div class="result">
<h3>이런 내가 직업을 갖는다면?</h3>
  <div class="jobs">
    <div class="job"></div>
    <div class="job"></div>
  </div>
</div>
<div class="result">
  <h3>이런 내가 직업 강의는 어떤가요?</h3>
  <a href="" target="_blank" class="lecture">
    <img src="" alt="강의">
  </a>
</div>
<a href="/" class="btn btn-gray btn-small">
  다시 테스트 하기
</a>
</html>

```

② results.js

※ 결과를 담고 있는 data.js 내용 확인하기

```

export const results = [
  {
    title: '출근 시간은<br>내가 정해!',
    character: '/images/result_character1.png',
    results: [
      '혼자서 멘탈 케어가 가능해요!<br>소속이 없어도, 별이가 불안정해도 저는 지금이 좋아요!',
      '일에 대한 욕심이 많아요.<br>행복한 야근이라고 할 수 있어요!',
      '자유로운 환경에서 더 잘해요!<br>하지만 누구보다 스스로 시간 관리가 철저해요.);',
    ]
  }
]

```

```

    '리더십 있다는 소리를 자주 들어요.<br>추진력이 있는 편이죠~ (뿌듯)'
  ],
  jobs: ['프리랜서', '창업가'],
  lectureImg: '/images/result_lecture1.png',
  lectureUrl: 'http://localhost:5500/lecture.html'
},

export const mbtis = {
  entj: 0,
  entp: 0,
  estp: 0,
  esfp: 1,
  esfj: 1,
  enfj: 1,
  infp: 2,
  isfp: 2,
  isfj: 2,
  infj: 2,
  estj: 3,
  istp: 3,
  intj: 4,
  intp: 4,
  istj: 4,
  enfp: 5
}

```

③ results.js 만들기

```

import { results, mbtis } from "./data.js";

// http://127.0.0.1:5000/results.html?mbti=isfj
// 위 내용 중 mbti=isfj 를 쿼리스트링이라고 함.
// 쿼리스트링을 가져오는 방법

const mbti = new URLSearchParams(location.search).get('mbti')
console.log(mbti)

// const result = results.mbtis['istf']
const result = results[mbtis[mbti]]
console.log(result)

```

```
const titleEl = document.querySelector('.page-title')
const characterEl = document.querySelector('.character')
const boxEls = document.querySelectorAll('.box')
const jobEls = document.querySelectorAll('.job')
const lectureEl = document.querySelector('.lecture')
const lectureImgEl = document.querySelector('.lecture img')

titleEl.innerHTML = result.title
characterEl.src = result.character

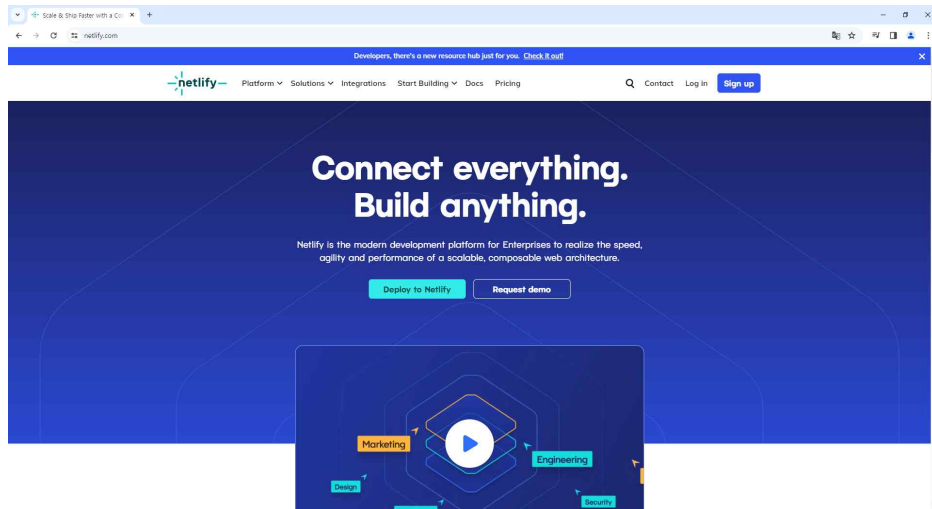
boxEls.forEach(function(boxEl, index){
  boxEl.innerHTML = result.results[index]
})

jobEls.forEach(function(jobEl, index){
  jobEl.innerHTML = result.jobs[index]
})

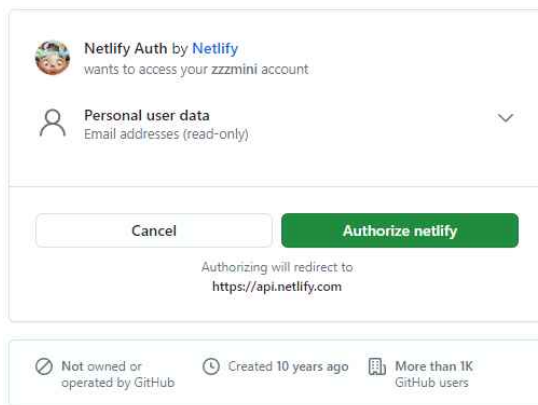
lectureEl.href = result.lectureUrl
lectureImgEl.src = result.lectureImg
```


11. 배포하기

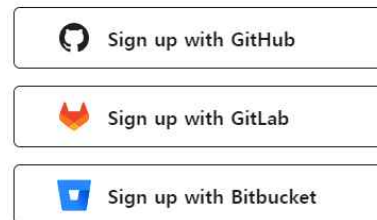
① 구글에서 netlify(네트리파이) 검색



② 우측 상단 Sign Up



Sign up to deploy your project



- Sign up with email : netlify에서 보낸 이메일로 확인

③ 깃헙으로 로그인 후 기본정보 입력 → Deploy with GitHub

Deploy your first project.

Not ready yet? [Skip this step for now.](#)

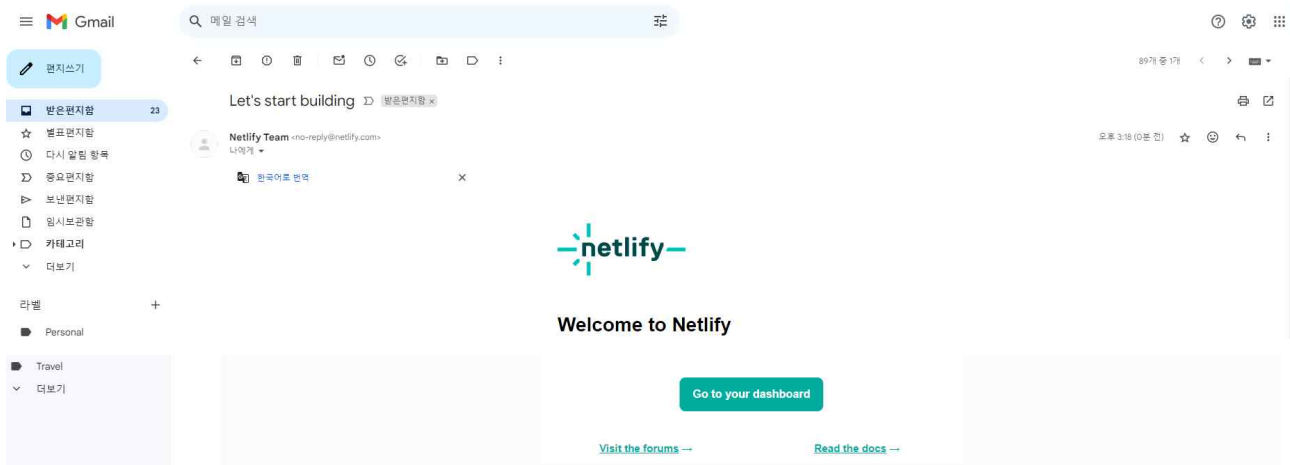


④ 이메일로 permission 정보 전송

Netlify by Netlify would like permission to:

The image shows a GitHub authorization screen. At the top, it says 'Netlify by Netlify would like permission to:'. Below this, there are three permissions listed: 'Verify your GitHub identity (zzzmini)', 'Know which resources you can access', and 'Act on your behalf' with a 'Learn more' link. Under 'Resources on your account', there is a section for 'Email addresses (read)' with a link to 'View your email addresses'. At the bottom, there are 'Cancel' and 'Authorize Netlify' buttons. Below the buttons, it says 'Authorizing will redirect to https://api.netlify.com'. At the very bottom, there are three status indicators: 'Not owned or operated by GitHub', 'Created 6 years ago', and 'More than 1K GitHub users'.

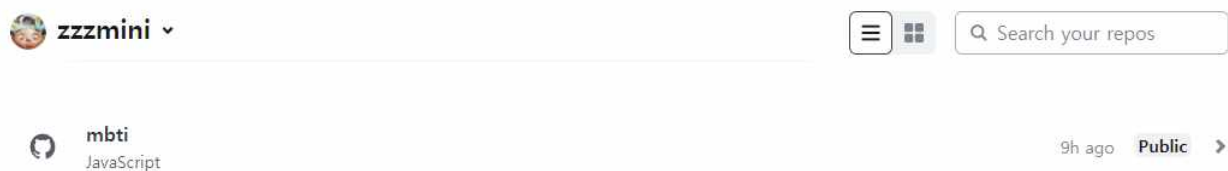
⑤ 이메일 정보 확인



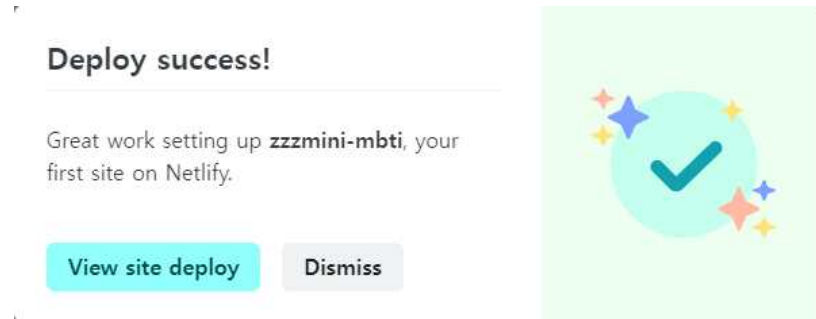
⑥ 승인 후 import from Git → Deploy with GitHub → Authorize Netlify → One select repositories → mbti repository 선택 → Install

⑦ GitHub 연결 후 배포 준비

Let's deploy your project.



⑧ 잠시 기다리면...



⑨ 배포완료

