

# 텀프로젝트 최종 발표 반응형 웹페이지 구현 (포트폴리오)



지도교수: 문일영 교수님

모바일프로그래밍(캡스톤디자인)

02분반 | 2019136151 이상영



1. 주제 선정 배경 및 목적
2. 사례 분석
3. 구현 목표
4. 프로젝트 개발 일정
5. 화면 설계
6. 구현 기능
7. 향후 개선사항 및 회고
8. 참고 문헌
9. 웹사이트 시연

# 주제 선정 배경 및 목적

---

## 1

# 1

## 주제 선정 배경 및 목적

---

가장 중점적으로 생각한 것은 “**활용성**”

단순히 프로젝트에서 끝나는 것이 아니라 계속 사용가능한 사이트

첫 번째 주제: 단축키 사전 사이트

### 두 번째 주제: 개인 포트폴리오 사이트

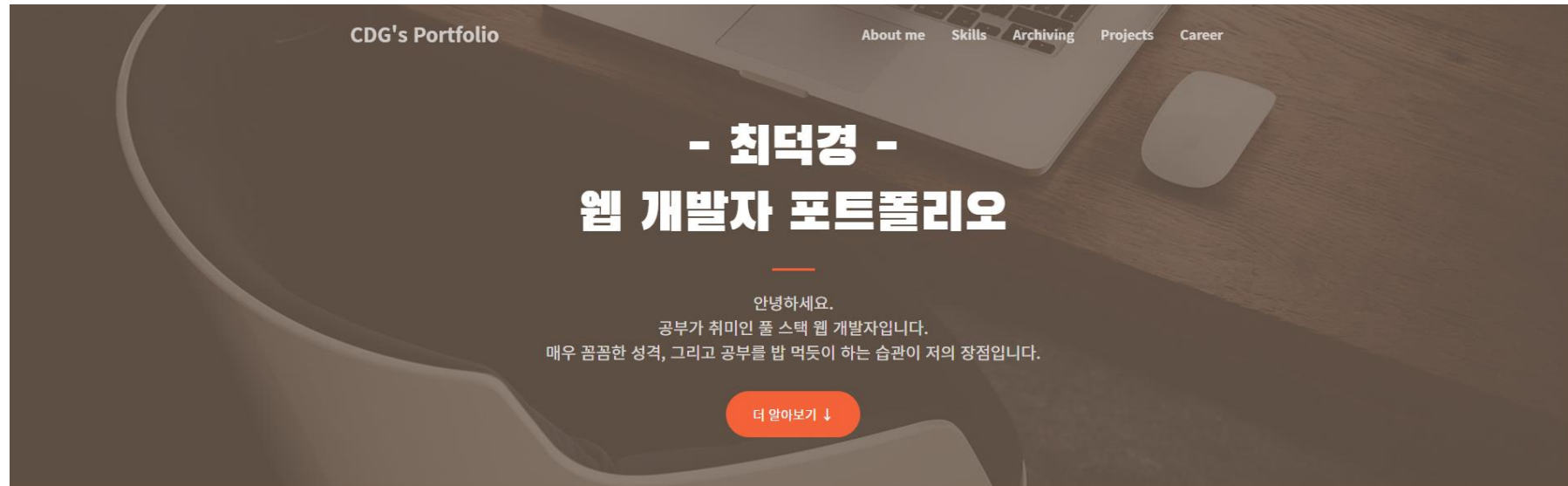
- 웹 개발자라면 필수 요소
- 나를 알릴 수 있는 기회
- 지금까지의 경험 정리

# 사례분석

---

## 2

# 사례분석(1/3)\_최덕경 웹 개발자 포트폴리오



## ABOUT ME



이름  
최덕경



연락처  
010-4582-5037



생년월일  
97.01.17



이메일  
hjcdg197@gmail.com



주소지  
서울특별시 송파구



학력  
서울대학교  
(컴퓨터공학부)

### [주요 특징]

1. 상단 메뉴가 고정
2. 하나의 페이지로 구성
3. 반응형 웹페이지

2

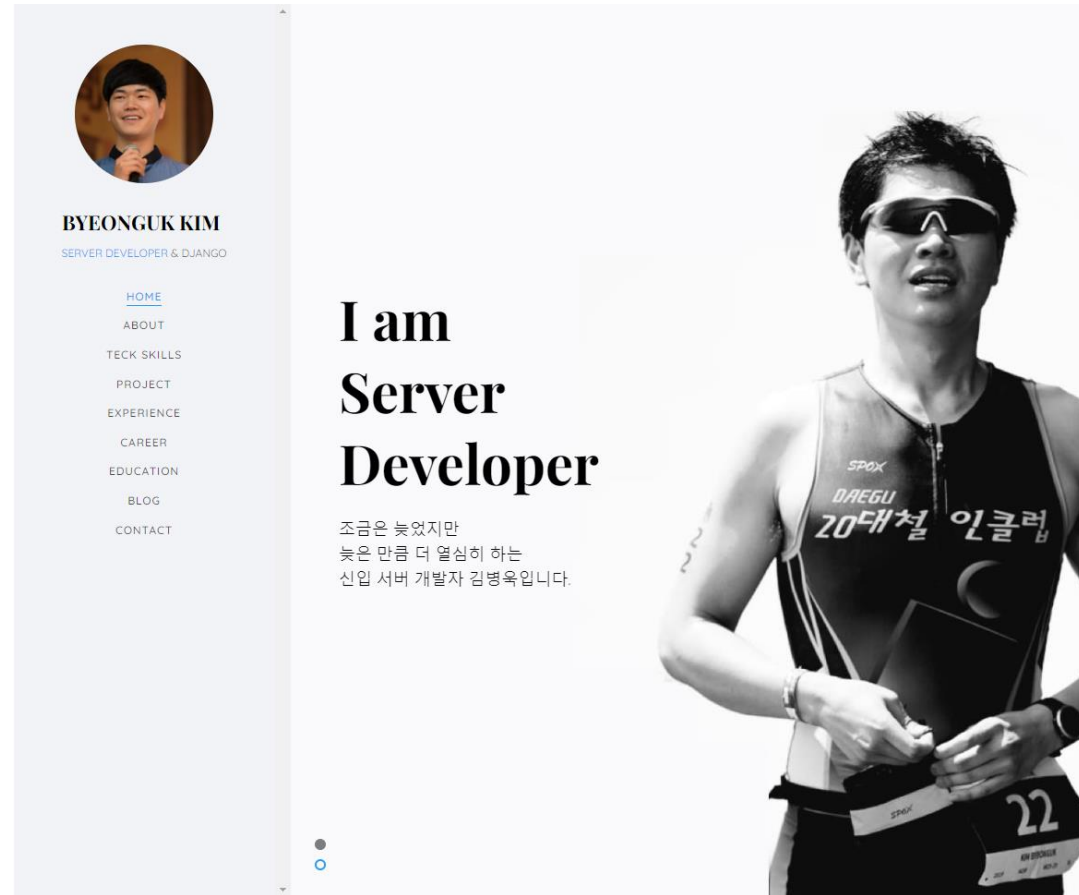
# 사례분석(2/3)\_배형준 웹 개발자 포트폴리오



## [주요 특징]

1. 좌측 메뉴가 고정
2. 하나의 페이지로 구성
3. 반응형 웹페이지
4. 직접 구현한 사이트가 아닌 부트스트랩 서식 이용

# 사례분석(3/3)\_김병욱 서버 개발자 포트폴리오



## [주요 특징]

1. 좌측 메뉴가 고정
2. 하나의 페이지로 구성
3. 반응형 웹페이지



# 구현 목표

3

## 3

## 구현 목표

---

### 첫 번째, 하나의 페이지로 구성

- 스크롤 형태로 페이지가 전환되도록 구현

### 두 번째, 메뉴는 상단에 고정

- 좌측아닌 상단에 메뉴를 고정(가독성 확보)
- 메뉴 클릭시 해당 페이지로 이동

### 세 번째, 직접 구현

- 수업시간에 배운 내용을 모두 써먹을 수 있도록 하드코딩 지향

# 프로젝트 개발 일정

---

4

## 4

# 프로젝트 개발 일정

프로젝트 시작일: 2022년 11월 14일 월요일

총 진행기간: 5주

일정 \ 주차	1	2	3	4	5
주제 선정					
사례 분석					
화면 설계					
구현					
보고서 및 발표 자료 작성					
발표 영상 촬영					

# 화면 설계

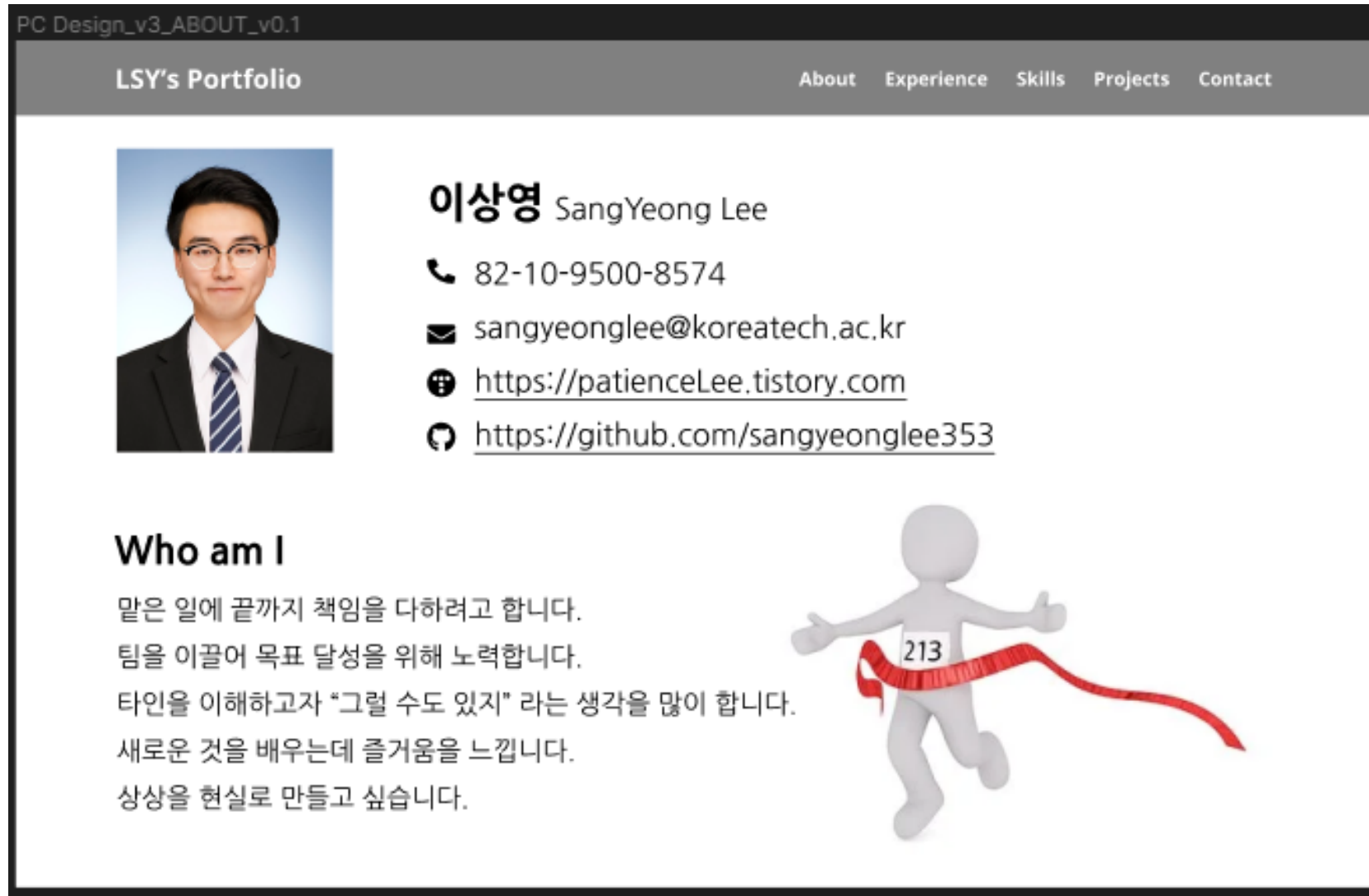
# 5



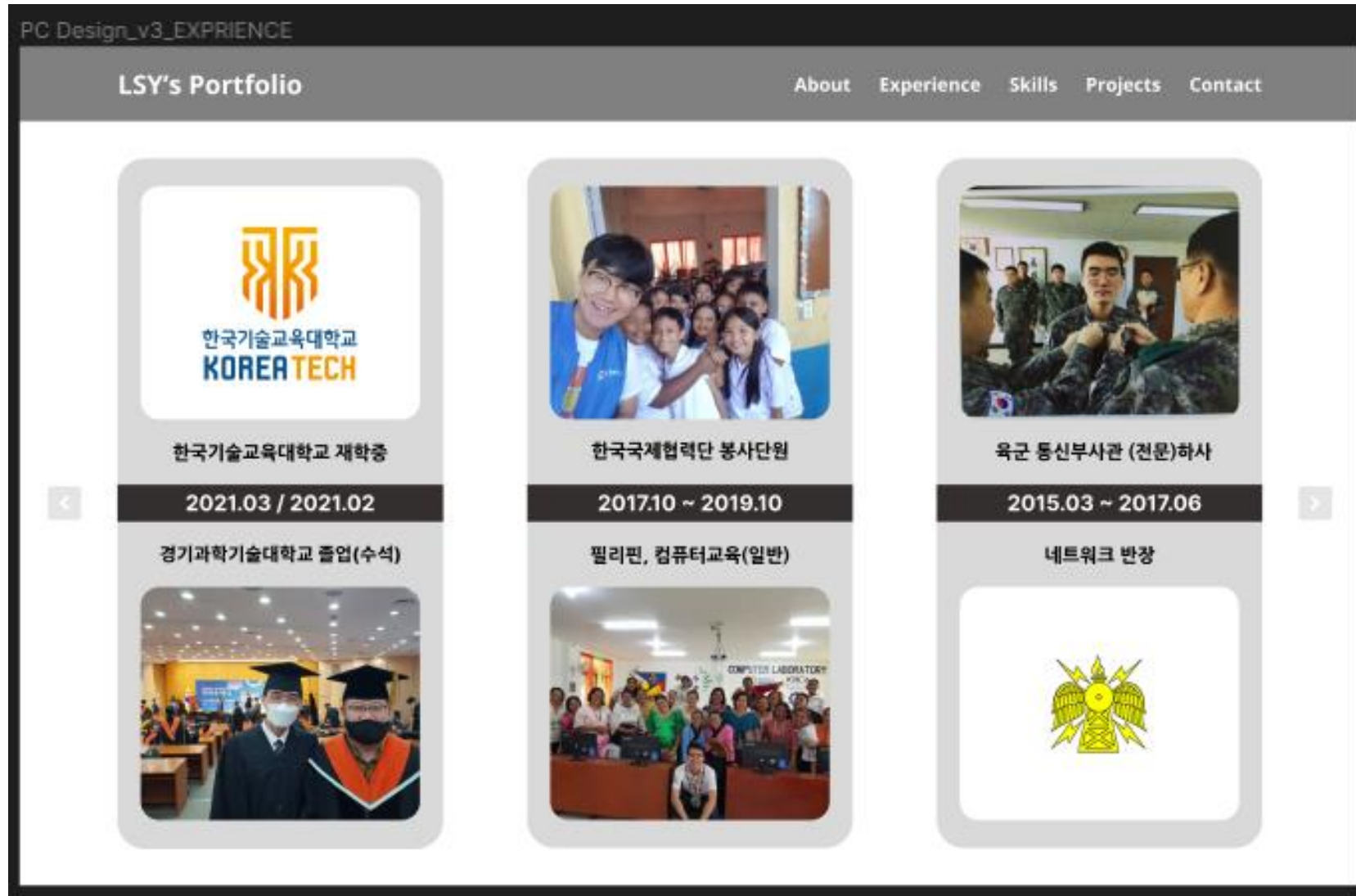
## 5 화면 설계(1/6)\_메인화면



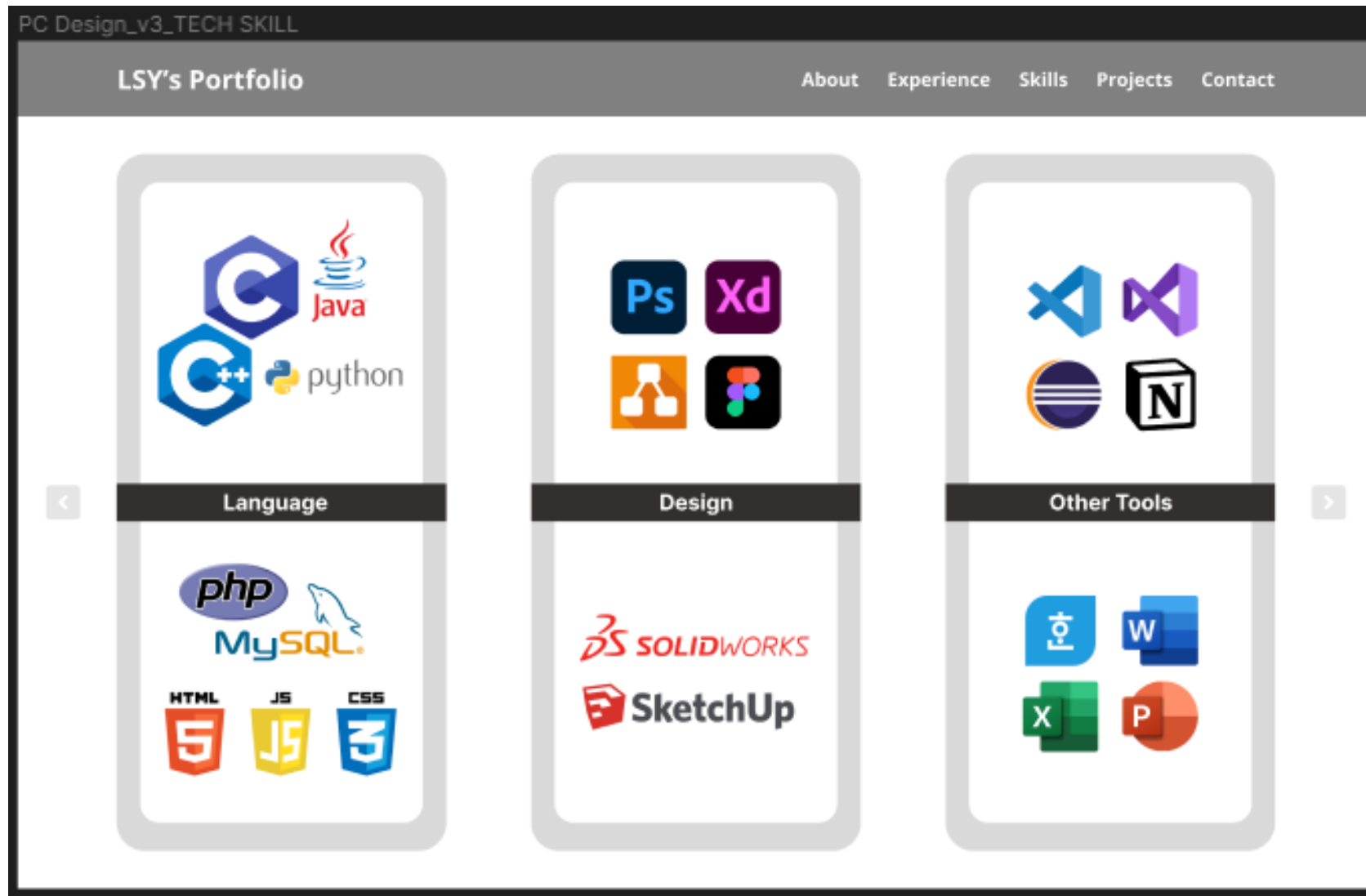
# 5 화면 설계(2/6)\_ABOUT



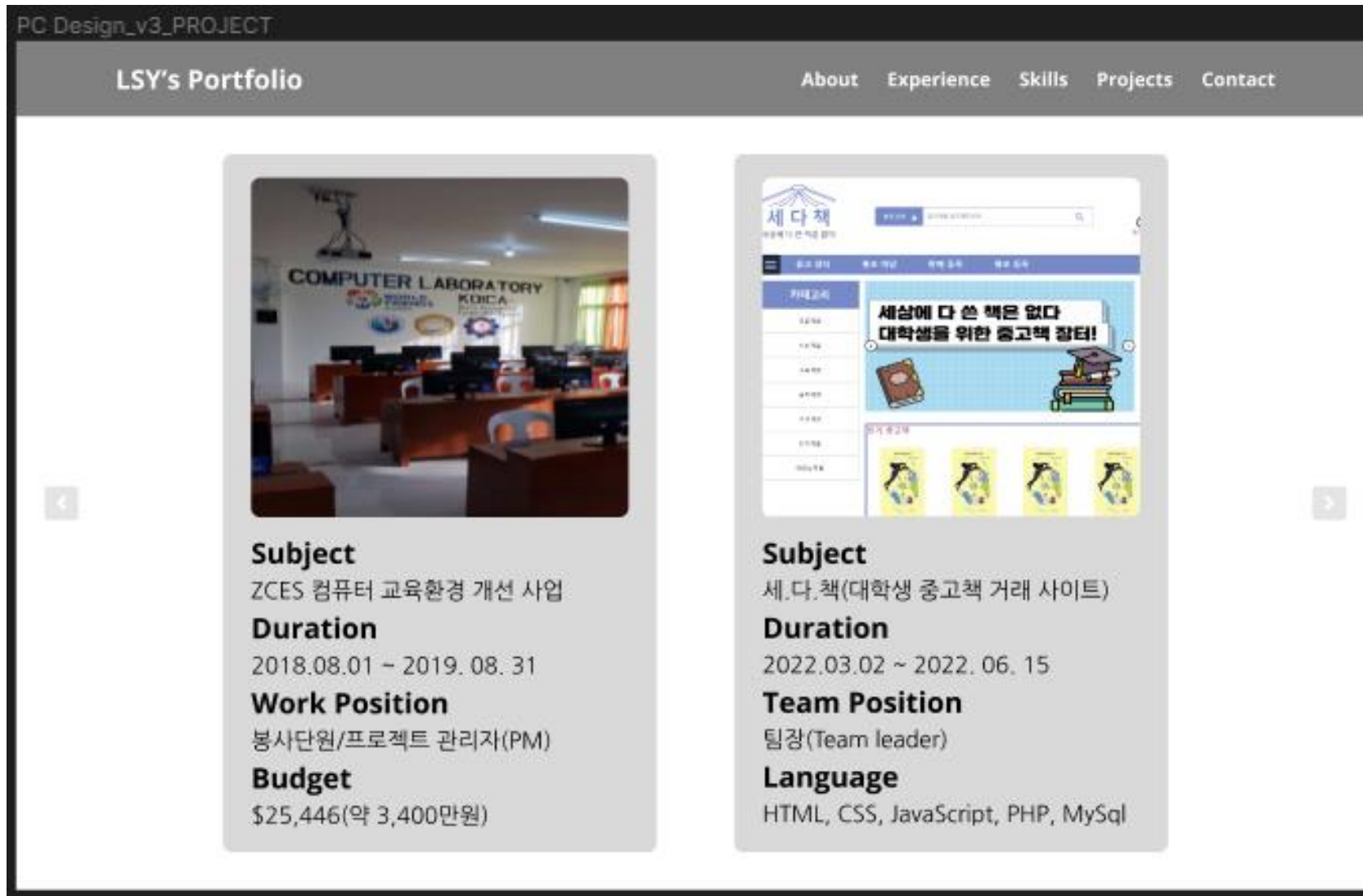
# 화면 설계(3/6)\_Experience



# 화면 설계(4/6)\_Skills



# 5 화면 설계(5/6)\_Projects





# 5 화면 설계(6/6)\_Contact

PC Design\_v3\_CONTACT

LSY's Portfolio

About Experience Skills Projects Contact

## Contact me


**Name**  
Sangyeong lee

**Email**  
sangyeong@gmail.com

**Subject**  
Please Type message title

**Message**  
Please type your message

SEND




# 구현 기능

---

# 6

## 6

## 구현 기능(1/4)\_FullPage 스크롤

```

1  /** 전체 페이지 슬라이드 */
2  window.addEventListener("wheel", function(e){
3      // e.preventDefault();
4      },{passive : false});
5
6  let $html = $("html");
7  let page = 1;
8  // let lastPage = $(".content").length;
9  let lastPage = $(".Home").length;
10
11  $html.animate({scrollTop:0},10);

```

```

13  $(window).on("wheel", function(e){
14
15      if($html.is(":animated")) return;
16
17      if(e.originalEvent.deltaY > 0){
18          if(page== lastPage) return;
19
20          page++;
21      }else if(e.originalEvent.deltaY < 0){
22          if(page == 1) return;
23
24          page--;
25      }
26      var posTop = (page-1) * $(window).height();
27
28      $html.animate({scrollTop : posTop});
29
30  });

```

```

<main>
  <article class="Home" id="Home">
    <div class="Home__contents">
      <div class="Home__contents__BG"></div>
      <div class="Home__contents__BG-Black"></div>
      <div class="Home__contents__text">
        <div class="Home__contents__text-title">
          <p>I am<br/>Front End<br/>Developer</p>
        </div>

```

```

32  /** Header 색상 변경 */
33  /* 방법: 높이에 따른 class 추가 제거 */
34  $(function(){
35      let rollHeader = 80;
36      $(window).scroll(function() {
37          let scroll = getCurrentScroll();
38          if ( scroll >= rollHeader ) {
39              $('header').addClass('roll');
40          }
41          else {
42              $('header').removeClass('roll');
43          }
44      });
45      function getCurrentScroll() {
46          return window.pageYOffset || document.documentElement.scrollTop;
47      }
48  });

```

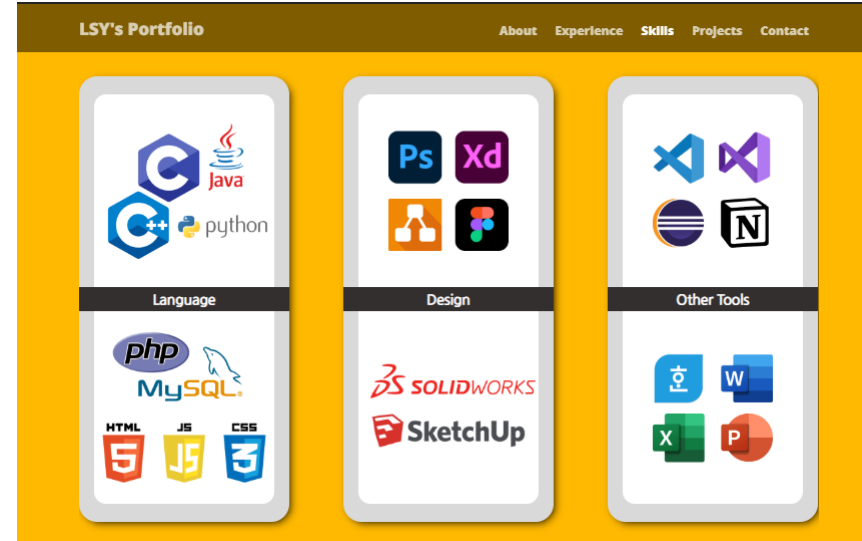
스크롤 화면 전환 &amp; 상단 헤더 고정

# 6 구현 기능(2/4)\_리스트 스크롤

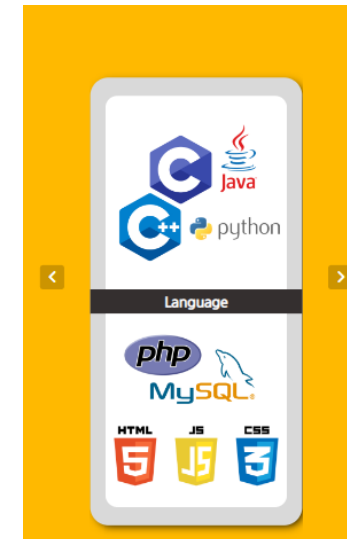
```

1  var swiper = new Swiper(".mySwiper", {
2      slidesPerView: 1,
3      spaceBetween: 10,
4      // loop: true,
5      pagination: {
6          el: ".swiper-pagination",
7          clickable: true
8      },
9      navigation: {
10         nextEl: ".next-btn",
11         prevEl: ".prev-btn",
12     },
13     breakpoints: {
14         // 화면의 넓이가 576px 이상일때
15         576: {
16             slidesPerView: 3,
17             spaceBetween: 90
18         },
19     },
20 });

```



데스크톱 화면



모바일 화면

모바일 화면 가로형 스크롤 구현(swiper.js)

## 6

## 구현 기능(3/4)\_Modal 창

```

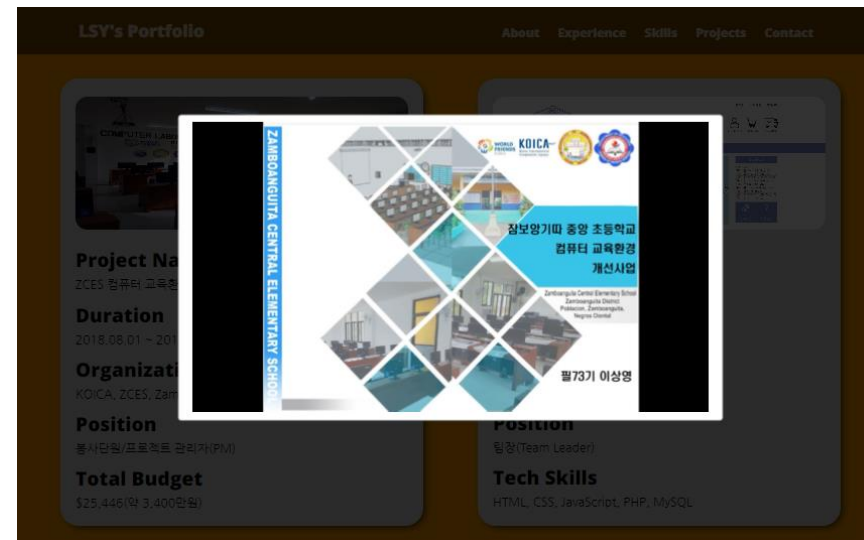
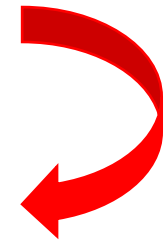
1  const modal = document.querySelector(".Project_card-modal-video");
2
3  // 방법 1
4  const selectButton = document.querySelector(".Project_card-modal-btn");
5  const modalBackground = document.querySelector(".modal__background");
6  const video = $("#player").get(0);
7  selectButton.addEventListener('click', () => {
8      modal.style.display = "block";
9      modalBackground.style.display = "block";
10     video.play();
11     // console.log("test");
12 });
13
14 // 배경 클릭시 사라지기
15 modalBackground.addEventListener('click', () => {
16     modal.style.display = 'none';
17     modalBackground.style.display = 'none';
18     /* 동영상 재생 중지(유튜브 영상인 경우) */
19     // $("#player")[0].contentWindow.postMessage({"event": "command", "func": "
20     /* 동영상 재생 중지(video인 경우) */
21     video.pause();
22     video.currentTime = 0;
23 });

```

Modal(모달) 비디오 창 구현



VIDEO 클릭 전



VIDEO 클릭 후



## 6 구현 기능(4/4)\_이메일 전송

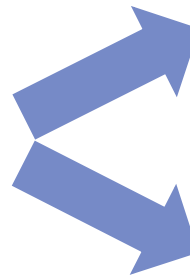


① Form Data 전달(JSON)



Google Apps Script

② Script 실행



Gmail

③ 내 메일로 이메일 전송



Google  
Sheets

③ 이메일 전송 로그 저장

## 6

## 구현 기능(4/4)\_이메일 전송



## ① From Data 전달(JSON)

```

1  (function() {
2      // get all data in form and return object
3      function getFormData(form) {
4          var elements = form.elements;
5          var honeypot;
6
7          var fields = Object.keys(elements).filter(function(k) {
8              if (elements[k].name === "honeypot") {
9                  honeypot = elements[k].value;
10                 return false;
11             }
12             return true;
13         }).map(function(k) {
14             if(elements[k].name !== undefined) {
15                 return elements[k].name;
16             } else if(elements[k].length > 0){
17                 return elements[k].item(0).name;
18             }
19         }).filter(function(item, pos, self) {
20             return self.indexOf(item) == pos && item;
21         });
22
23         var formData = {};
24         fields.forEach(function(name){
25             var element = elements[name];
26
27             // singular form elements just have one value
28             formData[name] = element.value;
29
30             // when our element has multiple items, get their values
31             if (element.length) {
32                 var data = [];
33                 for (var i = 0; i < element.length; i++) {
34                     var item = element.item(i);
35                     if (item.checked || item.selected) {
36                         data.push(item.value);
37                     }
38                 }
39                 formData[name] = data.join(', ');
40             }
41         });
42
43         // add form-specific values into the data
44         formData.formDataNameOrder = JSON.stringify(fields);
45         formData.formGoogleSheetName = form.dataset.sheet || "responses"; // d
46         formData.formGoogleSendEmail
47             = form.dataset.email || ""; // no email by default
48
49         return {data: formData, honeypot: honeypot};
50     }
51 }

```

## 6

## 구현 기능(4/4)\_이메일 전송



Google Apps Script

## ② Script 실행

Apps Script html contact form sendEmail

```

1  /*****
2  * This tutorial is based on the work of Martin Hawksey twitter.com/mhawksey *
3  * But has been simplified and cleaned up to make it more beginner friendly *
4  * All credit still goes to Martin and any issues/complaints/questions to me. *
5  *****/
6
7  // If you want to store your email server-side (hidden), uncomment the next line
8  var TO_ADDRESS = "sangyeong.lee353@gmail.com";
9
10 // spit out all the keys/values from the form in HTML for email
11 // uses an array of keys if provided or the object to detect
12 function formatMailBody(obj, order) {
13   var result = "";
14   if (order) {
15     order = Object.keys(obj);
16   }
17   // loop over all keys in the ordered form data
18   for (var idx in order) {
19     var key = order[idx];
20     result += "<div style='text-transform: capitalize; margin-bottom: 10px;'>";
21     // for every key, concatenate an <div /> /> <div /> pa
22     // and append it to the 'result' string created at the
23   }
24   return result; // once the looping is done, 'result' will
25 }
26
27 // sanitize content from the user - trust no one
28 // ref: https://developers.google.com/apps-script/reference/
29 function sanitizeInput(rawInput) {
30   var placeholder = HtmlService.createHtmlOutput("<div />");
31   placeholder.appendUntrusted(rawInput);
32   return placeholder.getContent();
33 }
34
35
36
37 function doPost(e) {
38   try {
39     Logger.log(e); // the Google Script version of console
40     record_data(e);
41   } catch (error) {
42     // shorter name for form data
43   }
44 }
45
46
47
48
49
50
51
52
53
54 // if you have your email uncommented above, it uses that 'TO_ADDRESS'
55 // otherwise, it defaults to the email provided by the form's data attribute
56 var sendEmailTo = (typeof TO_ADDRESS !== "undefined") ? TO_ADDRESS : mailData.formGoogleS
57
58 // send email if to address is set
59 if (sendEmailTo) {
60   MailApp.sendEmail({
61     to: String(sendEmailTo),
62     subject: "[Portfolio Contact] check the contents!!",
63     // replyTo: String(mailData.email), // This is optional and reliant on your form actu
64     htmlBody: formatMailBody(mailData, dataOrder)
65   });
66 }
67
68 return ContentService // return json success results
69   .createTextOutput(
70     JSON.stringify({"result": "success",
71                   "data": JSON.stringify(e.parameters) })))
72   .setMimeType(ContentService.MimeType.JSON);
73 } catch (error) { // if error return this
74   Logger.log(error);
75   return ContentService
76     .createTextOutput(JSON.stringify({"result": "error", "error": error}))
77     .setMimeType(ContentService.MimeType.JSON);
78 }
79 }
80

```

6

# 구현 기능(4/4)\_이메일 전송



## Gmail

### ③ 내 메일로 이메일 전송



## Google Sheets

### ③ 이메일 전송 로그 저장

Gmail interface showing an email from sangyeong.lee353@gmail.com to 나에게. The email content is highlighted in a red box:

[Portfolio Contact] check the contents!!

받은편지함 x

sangyeong.lee353@gmail.com 오후 5:10 (2분 전) ☆ ↶ ⋮

이름  
채용담당자

Email  
[syRecruit@worldjob.co.kr](mailto:syRecruit@worldjob.co.kr)

Message  
채용 제안 드리고자 연락드립니다.

HTML Form Responses의 사본 ☆ 📎 ☁

파일 수정 보기 삽입 서식 데이터 도구 확장 프로그램 도움말 이상영님이 2분 전에 마지막으로 수정했습니다.

Google Sheets interface showing a table with email logs. The table has columns: Timestamp, name, message, email, color, subject. The table is sorted by Timestamp. The last row is highlighted in red:

	A	B	C	D	E	F	G	H	I
1	Timestamp	name	message	email	color	subject			
2	19/10/2015 16:42:01	Ashley	I love this tutorial!!	ashley@test.com	apple green				
3	19/10/2015 17:10:52	Bob	And it's open-source too?	bob@test.com	blue				
4	09/03/2018 07:41:51	Carol	Wow, you guys rock!	carol@test.com	cerulean				
5	09/03/2018 07:53:39	Doug	Keep it coming.	doug@test.com	dandelion				
6	09/03/2018 09:30:46	Erin	Pull request on its way!	erin@test.com	emerald green				
7	10/12/2022 12:22:40	tester	메일 보내는 중입니다.	sangyeonglee@koreatech.ac.kr		테스트 메일			
8	10/12/2022 12:22:41	tester	메일 보내는 중입니다.	sangyeonglee@koreatech.ac.kr		테스트 메일			
9	10/12/2022 12:34:14	채용 담당자	당신을 보고 싶습니다. 회신주소	sangyeonglee@koreatech.ac.kr		채용 건의			
10	10/12/2022 12:34:15	채용 담당자	당신을 보고 싶습니다. 회신주소	sangyeonglee@koreatech.ac.kr		채용 건의			
11	11/12/2022 08:10:04	채용담당자	채용 제안 드리고자 연락드립니다	syRecruit@worldjob.co.kr					

# 향후 개선사항 및 회고

---

7



# 7 향후 개선사항 및 회고

## 첫 번째, 데스크톱과 모바일 간의 자연스러운 연결을 위한 CSS 수정

- 화면 전환을 더 자연스럽게 하기 위하여 중간 사이즈의 반응형 미디어 쿼리를 추가할 필요가 있음.

## 두 번째, CSS 표기법 수정

- BEM 표기법을 준수하고자 했으나, 미비한 부분이 많음.
- 중복되는 속성에 대해서는 Class를 병합할 필요가 있음.

## 세 번째, 웹사이트 배포

- 다른 사람들이 볼 수 있도록 github pages나 AWS 등을 이용한 배포가 필요함.

## 7 향후 개선사항 및 회고

**2022년 1학기 웹 프로그래밍(창업) 수업 때보다 display 사용이 익숙해짐.**

- 1학기 때엔 화면 배치가 어려웠으나, 이번 프로젝트를 진행하면서 해당 부분을 이해할 수 있게 됨.

**화면 설계 도구인 피그마(Figma) 사용이 익숙해짐.**

- 대중적으로 사용되는 화면 설계 도구를 이용해 나만의 페이지를 설계할 수 있게 됨.

**방학기간 동안 본 사이트 개선 및 첫 번째 주제(단축키 사전) 사이트도 구현해 볼 용기가 생김.**

- 웹 개발자를 목표로 하고 있지만, 항상 화면 구현에서 막혀서 길을 헤매고 있었는데 이번 기회를 통해 벽을 하나 허문 느낌을 받음.

## 참고 문헌

# 8

# 8 참고 문헌

1. underline 위치 조정. <https://hianna.tistory.com/680>
2. transition 속성. <https://skydoor2019.tistory.com/6>
3. 페이지 스크롤. <https://velog.io/@sklove96/한-페이지씩-올라가는-스크롤>
4. 일반 스크롤 + 원페이지 스크롤 혼합형. <https://do-dam.tistory.com/101>
5. 반응형 웹페이지. <https://studiomeal.com/archives/1004>
6. 상단 고정바. <https://www.daleseo.com/css-position-fixed-navigation/>
7. 하이퍼 링크 색상 제거. <https://storyhong.tistory.com/82>
8. 슬라이드 이동 자연스럽게 하기. <https://salgum1114.github.io/css/2019-04-28-scroll-behavior-smooth/#bottom>
9. article 태그와 section 태그의 차이. <https://velog.io/@apricotsoul/HTML-section요소-main-요소>
10. 웹페이지 초기 크기. <https://emmakwon.kr/ideal-screen-size-for-desktop/>
11. 메일 하이퍼링크. <https://m.blog.naver.com/tokyomari/221912264344>
12. font-family 사용법. <https://kcmschool.com/124>
13. 스크롤바 숨기기. <https://wooaoe.tistory.com/49>
14. fontawesome 적용하기. <https://stackoverflow.com/questions/17394760/content-property-is-not-working-correctly>
15. nth-child와 nth-type-of의 차이. <https://soharu.tistory.com/3>
16. modal 창 만들기. <https://7942yongdae.tistory.com/104>
17. background-size 조절. <https://devjhs.tistory.com/611>
18. CSS 속성 기술 순서. <https://velog.io/@gygy/CSS-작성-순서>
19. video 영상 정지. <https://stackoverflow.com/questions/52087822/how-to-stop-not-pause-html5-video-with-jquery>
20. 가로 슬라이드 기능. <https://swiperjs.com/swiper-api#pagination>
21. 이메일 보내기 기능. <https://yuuisang.github.io/blog-http-mail>
22. [도서] 웹 디자인 & 웹 퍼블리싱을 위한 피그마 완벽 활용서, 김동주 지음.

# 웹사이트 시연

---

9