

1) As Is

- Main Page Hooker 화면 구현
- Footer 정리
- JavaScript 기능 구현
 - 공지사항 게시판 구현
 - 로그인 성공시의 기능 구현
 - 로그아웃 버튼 등장
 - 로그아웃시 초기화면으로 리셋
 - 현 위치 확인의 Text 변경 : Guest -> 로그인 아이디로 변경
 - 예약정보
 - 공지사항 게시판에서 글 작성 버튼 생성 및 글 작성
 - 회원가입 정보 : ID, PW 이외의 정보를 받을 수 있도록 구현
- CSS 통일 (후순위)
 - 색상, 글꼴, 이미지 구성등
 - 메인 컬러, 서브 컬러, 폰트 사이즈
 - Padding, Margin 규격화
 - 가급적 중앙 정렬을 기준으로 Frame 구성

2) JavaScript 기능 구현

2-1) 로그인 기능 변경전

```
function handleSubmitId() {  
    let currentId = loginId.value;  
    let currentPassword = LoginPassword.value;  
    localStorage.setItem(currentId, currentPassword);  
    alert("회원가입이 완료되었습니다");  
}
```

기존의 소스코드중 일부

- 이전 로직은 단순히 Text 를 받아와서 로컬스토리지에 저장하기까지
- 이는 가장 기초적인 기능의 구현임을 알게 되었다.
- Key : Value 의 한쌍의 오브젝트만으로는 다른 기능을 구현하기에는 한계가 있었다.
 - 단순히 Key : Value 를 받아오는 변경전의 모습과

afadrf	3r535r
df4g	335fg
abcd	1234
1	3434

변경전

2-2) 로그인 기능 변경후

```
const memberId = document.querySelector(".placeholder-text");
const memberPassword = document.querySelector(".userpw");
//localStorage.clear(); //로컬스토리지 초기화
let user = JSON.parse(localStorage.getItem("user") || "{}");
// 위의 코드 설명 : user JSON 데이터 받아오기 /
// 없으면 빈 배열 -> JSON.parse로 STRING 형태 => JSON 형태로 변환
let member = user.member || []; // user(JSON) > member(arr)

function handleSubmitId(){
    let currentId = memberId.value;
    let currentPassword = memberPassword.value;
    alert("회원가입이 완료되었습니다");
    let param = {};
    param.currentId = memberId.value;
    param.currentPassword = memberPassword.value;

    member.push(param) // member 배열에 새로운 데이터 push
    user.member = member; //user (JSON) 에 추가된 member 넣기

    localStorage.setItem("user", JSON.stringify(user)); //로컬 스토리지에
    추가된 user (JSON) 넣기
}
```

회원가입의 소스코드

- 이제는 Value 자체를 object 로 받아 ID 와 PW 를 받게 됨

Key	Value
user	{"member":[{"currentId":"aaa","currentPassword":"111"},{"curre..."}
▼ {member: [{currentId: "aaa", currentPassword: "111"}, {currentId: "bbb", currentPassword: "222"}]}	
▼ member: [{currentId: "aaa", currentPassword: "111"}, {currentId: "bbb", currentPassword: "222"}]	
▶ 0: {currentId: "aaa", currentPassword: "111"}	
▶ 1: {currentId: "bbb", currentPassword: "222"}	

변경 후

```
const check_id = document.querySelector(".loginId");
const check_password = document.querySelector(".loginPassword");
// 로컬 > user (JSON) > member(arr)
var user = JSON.parse(localStorage.getItem("user")); //로컬에 저장된
유저 JSON
console.log(user)
let member = user.member; //로컬에 저장된 멤버 배열
```

```
function CheckValue() {
    const ID = check_id.value;
    const PASSWORD = check_password.value;

    let currentPassword;
    for(let i=0; i<member.length; i++){
        if(member[i].currentId === ID){
            currentPassword = member[i].currentPassword;
            console.log(currentPassword)
        }
    }

    if(currentPassword === PASSWORD && currentPassword !== null &&
currentPassword !== ""){
        localStorage.setItem("ID", ID);
        alert("로그인 성공");
    }else{
        alert("ID 와 PASSWORD 를 확인해주세요");
    }
}
```

로그인 소스코드

2-3) 로그인으로 인한 기능 구현

- MyLocation 에서 Guest -> aaa(로그인한 아이디) 로 변경

현재
aaa
위치는 서울대입구역 입니다.



- 로그인시 로그인, 회원가입 버튼이 사라지고 로그아웃 버튼이 생성

The image shows a web page layout. At the top is a navigation bar with buttons: '홈으로' (Home), '공지사항' (Notice), '출석체크' (Check-in), '내위치확인' (Check location), '쿠폰' (Coupon), '로그인' (Login), and '회원가입' (Sign up). Below the navigation bar is a login form with two input fields: '아이디' (ID) and '비밀번호' (Password). Below these fields is a large orange button labeled '로그인' (Login). At the bottom of the form is a link labeled '회원가입' (Sign up).

3) 완료된 내용

- Main Page Hooker 화면 구현



- 소비자의 클릭을 유도할 페이지
- reference를 참고하여 구성, 회원가입을 권장한다.
- reference : <https://theboss.imweb.me/>

- Phase3 : 리뷰 게시판 추가

The image shows a form for adding a review. It has a title '리뷰 작성란' (Review writing area). Below the title are three input fields: '제목' (Title), '내용' (Content), and '비밀번호' (Password). There are also buttons for '제목 입력' (Enter title), '내용 입력' (Enter content), and '비밀번호 입력' (Enter password). At the bottom of the form are two buttons: '작성완료' (Finish writing) and '취소' (Cancel).

- 글 작성시 리뷰 게시판으로 이동

4) To do

- 메인의 로고 재구성
- 본인이 한 파트에 대해서 기술명세서 작성해오기
- CSS 통일 (후순위)
 - 색상, 글꼴, 이미지 구성등
 - 메인 컬러, 서브 컬러, 폰트 사이즈
 - **Padding, Margin** 규격화
 - 가급적 중앙 정렬을 기준으로 **Frame** 구성