

A decorative background featuring a network diagram with nodes and connecting lines. The nodes are represented by circles of varying sizes and colors (blue, grey, and white), and the lines are thin and grey. The network is more dense on the left and right sides of the slide, with a large cluster on the left and a smaller one on the right.

This is Museum App-web presentation



Hello!

Today Project

국내에 위치한 국립 박물관의
전시회 및 이벤트 종합 정보APP개발

- Spring, Mysql, Svelte, Html 로 구현

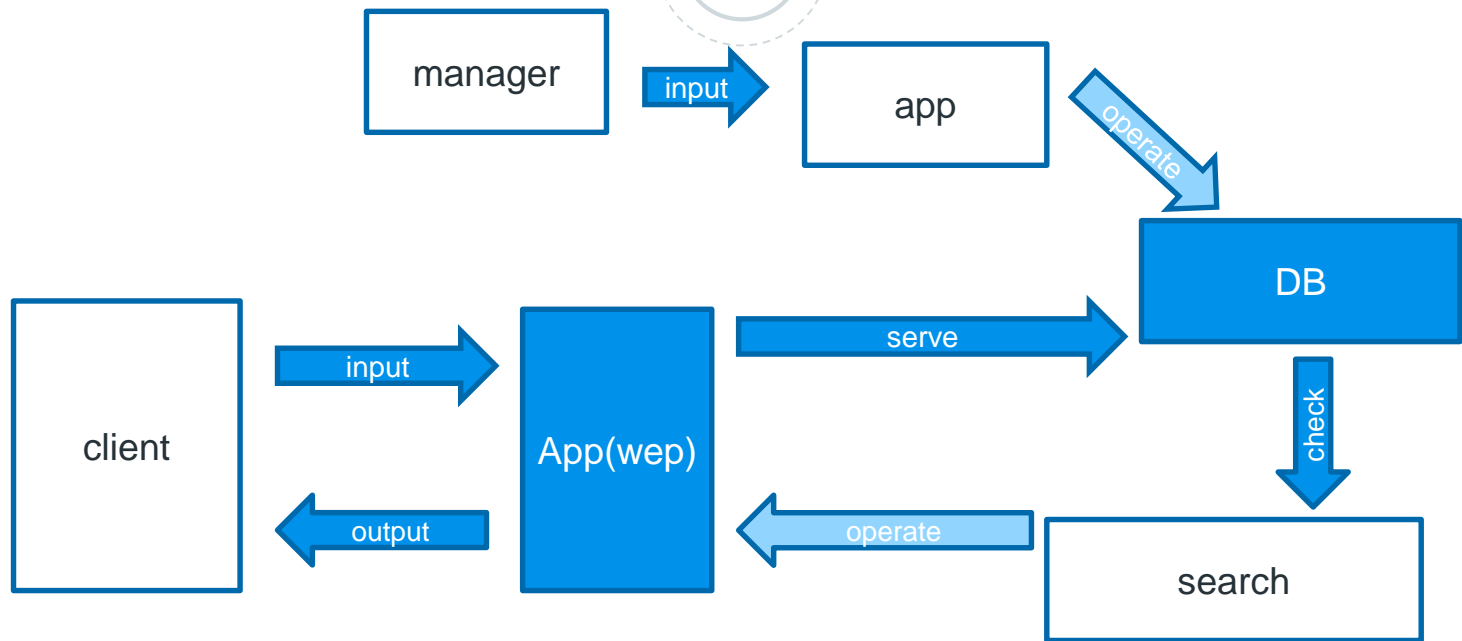
A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines, with some nodes highlighted in blue.

1.

Basic construct

DB를 통한 자료 저장 및 변경

APP 구동 구조



A decorative network diagram at the top of the slide, featuring a central node connected to several other nodes, which are further connected to a larger network of nodes and lines. The nodes are represented by circles of varying sizes, some with internal patterns, and the lines are thin and grey.

“

App을 통하여 DB에 저장된
내용을 기반으로 App에 정보가
확인 가능하게 된다. **DB에서**
모든 정보를 받아와 App을 구성
컴퓨터 서버로부터 **CRUD**를 진행

▼ JSPBoot-museum [boot] [devtools]

- ▼ src/main/java
 - ▼ com.example.demo
 - > JspBootMuseumApplication.java
 - ▼ com.example.demo.dto
 - > Event.java
 - > EventView.java
 - > Exhibition.java
 - > ExhibitionView.java
 - > Goods.java
 - > GoodsView.java
 - > **mainpage.java**
 - > MainView.java
 - > MuseumList.java
 - > PostBoard.java
 - > PostBoardView.java
 - > Relic.java
 - ▼ com.example.demo.repository
 - > eventRepository.java
 - > exhibitionRepository.java
 - > goodsRepository.java
 - > mainRepository.java
 - > MuseumListRepository.java
 - > postboardRepository.java
 - > relicRepository.java
 - > src/main/resources
 - > src/test/java
 - > JRE System Library [JavaSE-1.8]
 - > Maven Dependencies
 - > src

STS를 통해 TABLE 및 DB를 설정

```
@Entity
public class mainpage {

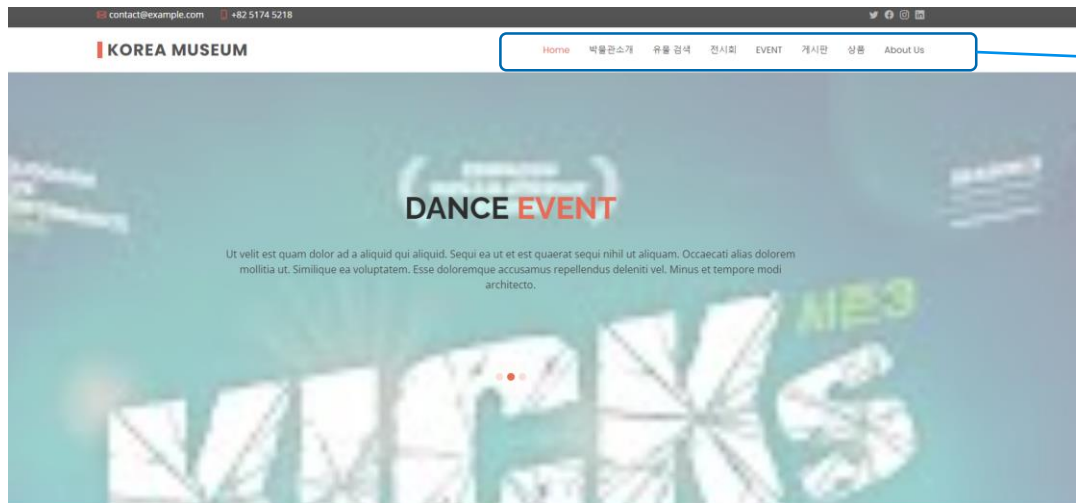
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer id;
```

Repository를 통해 control, service 설정

```
@RepositoryRestResource(collectionResourceRel = "mainpage", path="mainpage", excerptProjection = MainView.class)
@CrossOrigin
public interface mainRepository extends JpaRepository<mainpage, Integer>{

}
```

Bootstrap 활용



각 페이지는 html로 연결

반응형앱을 구성



Svelte 컴포넌트

App 안에 svelte를 활용한 구성

각각의 Table에서
작용하여 구성

- 관람시간
 - 일/화/목/금/토 10:00 ~ 18:00
 - 수/토 10:00 ~ 21:00
 - * 입장 마감은 폐관 30분 전까지
- 관람료
 - 무료 특별전시는 유료

전국 전시회 모음

전국에 진행중인 전시회



공지사항

• 공지사항 알림	2022-08-08
• 시범공연 도입	2022-08-12
• 하반기 채용 공고	2022-08-19
• 특설 공연은 석제 및 방역 조치합니다.	2022-08-12
• 광주박물관 전시회 예정 정보입니다.	2022-08-16
• 광주박물관 수장고 안내	2022-08-19
• 광주박물관 내부 공사 안내	2022-08-23
• 박물관 입체이도 안내	2022-09-08

EVENT

지금 신청 가능하거나 진행중인 EVENT 확인 소식!




```
import { onMount } from "svelte";
```

```
let exhib=[]  
let boards=[]  
let mainposter=[]  
let eventposter=[]
```

```
const upsidebanner="http://192.168.0.51:8080/mainpage/"  
const slidingbnn="http://192.168.0.51:8080/exhibition/"  
const boardbnn="http://192.168.0.51:8080/postboard/"  
const eventdb="http://192.168.0.51:8080/event/"
```

```
onMount(async function(){  
  const res=await fetch(upsidebanner)  
  const data = await res.json()  
  mainposter=data._embedded.mainpage  
  
  const red= await fetch(slidingbnn)  
  const open= await red.json()  
  exhib=open._embedded.exhibition  
  
  const rew= await fetch(boardbnn)  
  const por= await rew.json()  
  boards=por._embedded.postboard  
  
  const rsq= await fetch(eventdb)  
  const vev= await rsq.json()  
  eventposter=vev._embedded.event  
})
```

DB와의 연결

연결 된 DB 불러오기

이미지 슬라이드

```
let currId = 0;  
let positionLeft = 0;
```

```
const moveSlider = () => {  
  positionLeft = currId * 33;  
};
```

```
const next = () => {  
  currId = currId === eventposter.length - 1 ? 0 : currId + 1;  
  moveSlider();  
};
```

```
const prev = () => {  
  currId = currId === 0 ? eventposter.length-1 : currId - 1;  
  moveSlider();  
};
```

```
const getIndex = (index) => {  
  currId = index;  
  moveSlider();  
};
```

박물관 소개



각 알맞은 컴포넌트 적용

KOREA MUSEUM

Home

박물관소개

유물 검색

전시회

EVENT

게시판

상품

About Us

우리의 박물관을 찾아서

국립 중앙박물관

국립 경주박물관

국립 광주박물관

국립 전주박물관

국립 대구박물관

국립 부여박물관

국립 공주박물관

국립 진주박물관

국립 청주박물관

국립 김해박물관

국립 제주박물관

국립 춘천박물관

국립 나주박물관

국립 익산박물관

박물관 내역



- ◎ 사진
- ◎ 지도
- ◎ 정보 기입

- ◎ 위 내용은 DB에 저장하여 불러옴(DB에서 수정필요)



장소	국립경주박물관
주소	경북 경주시 일정로 186 국립경주박물관
전화번호	054-740-7500
개관	1945년 10월 7일
이용시간	10:00-18:00, 토/일요일 : 1시간 연장, 야간연장개관 -21:00, 매달 마지막 수요일, 3~12월 중 매주 토요일
휴무일	1월 1일, 설날, 추석
요금	무료
편의시설	청각장애인 수어상당, 시각장애인 점자 안내판, 엘리베이터, 장애인 화장실, 장애인 주차장, 임산부 및 유아휴게공간
홈페이지	https://gyeongju.museum.go.kr

▼ museumList

> infoimg

- BuyeoMuseum.svelte
- CentralMuseum.svelte
- CheongjuMuseum.svelte
- ChuncheonMuseum.svelte
- DaeguMuseum.svelte
- GimhaeMuseum.svelte
- GongjuMuseum.svelte
- GwangjuMuseum.svelte
- GyeongjuMuseum.svelte
- IksanMuseum.svelte
- JejuMuseum.svelte
- JeonjuMuseum.svelte
- JinjuMuseum.svelte
- MuseumInfo.svelte**
- NajuMuseum.svelte

```
<script>
  import { onMount } from "svelte";

  const museumList="http://192.168.0.51:8080/museumlist/"

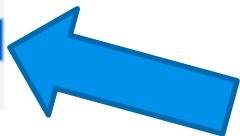
  let museuminfomation=[]

  onMount(async function(){
    const res=await fetch(museumList)
    const data = await res.json()
    museuminfomation=data._embedded.museumlist
    console.log(museuminfomation)
  })
</script>
```

DB 적용

```
<div id="main">
  <div id="contents">
    <div id="muse">
      {#each museuminfomation as museum}
      {#if museum.photo=="central3"}
        <div id="muse-img">
          
        </div>
        <div id="map">
          <iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3164.3241777948615!2d126.97828151564724!3d37.523"
            width="400" height="300" style="border:0;" loading="lazy" referrerpolicy="no-referrer-when-downgrade"></iframe>
        </div>
      </if>
    </div>
  </div>
</div>
```

조건부



각 컴포넌트로 조건에 맞아 컴포넌트 on

유물 검색 - 조건부 Read, 정보 확인

- ◎ 조건부 검색
- ◎ 유물 명 또는 조건부(지역, 소속)으로 조건에 부합하는 유물 Read

검색 상세 검색 전체 보기

☐ 국보 ☐ 보물 ☐ 기타

검색 상세 검색 전체 보기



검색 내용

전체 보기의 경우
DB의 모든 정보 검색
검색, 상세 검색을
이용하면 각 조건에
맞는 유물만 DB에서
Read.

DB의 정보가 많을 경우 페이지 나누기

```
<script>
import { onMount } from "svelte";
import RelicOption from "../RelicOption.svelte";
import RelicSearch from "../RelicSearch.svelte";
import { paginate, PaginationNav } from "svelte-easy-paginate";
```

```
const relic = "http://192.168.0.51:8080/relic/?page=0&size=200"
```

DB의 한 페이지 저장 용량 조절

```
let items=[]
```

```
onMount(async function(){
  const res=await fetch(relic)
  const data = await res.json()
  items=data._embedded.relic
  // console.log(relicinfo)
})
```

```
let currentPage = 1
```

```
let pageSize = 4
```

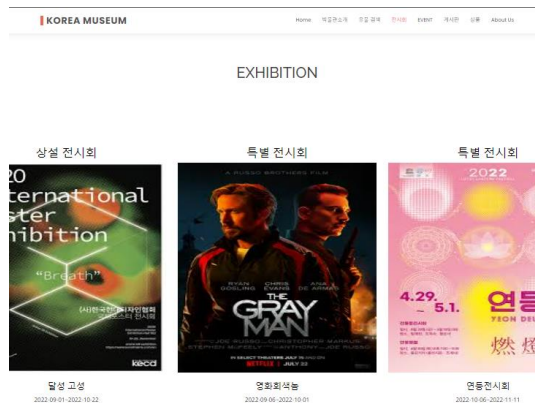
```
$.relicinfo = paginate({ items, pageSize, currentPage })
```

대량의 정보 페이지 나누기 기능

```
{/each}
<PaginationNav
  totalItems="{items.length}"
  pageSize="{pageSize}"
  currentPage="{currentPage}"
  limit="{1}"
  showStepOptions="{true}"
  on:setPage="{(e) => currentPage = e.detail.page}"
/>
</ul>
```

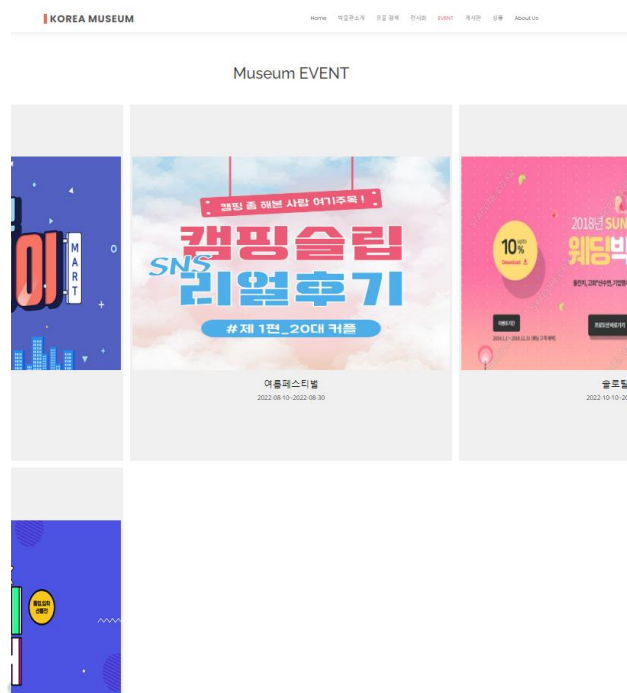
Body에 페이지 이동 및 클릭 이벤트

DB에서 이미지 주소 및 내용을 받아 표시



전시회 및 EVENT도
각 DB를 통해
이미지, 기간등 정보
Read

특이사항을 텍스트 또는
반투명상태로 표시



```
let items=[]
let inknum

const slidingbnn="http://192.168.0.51:8080/exhibition/"

onMount(async function(){
  const red= await fetch(slidingbnn)
  const open= await red.json()
  items=open._embedded.exhibition
})
function addno(ibs){
  inknum=ibs
  console.log(inknum)
}

$:callid.update(t=>inknum)

let page=true
let detail=false

function toggle(){
  page=!page
  detail=!detail
}
```

JS store.js

museum > src > lib > JS store.js > ...

```
1 import { writable } from "svelte/store";
2
3 export const callid = writable(0);
4 export const boardid = writable(0);
5 export const boardcount = writable(0);
```

Svelte의 store로 특정 번호(PK)저장

DB의 PK를 이용한 컴포넌트

전시회, EVENT 이미지
선택 시 정보의 보다
자세한 내용을 볼 수 있는
컴포넌트 가 활성화

자세히보기 컴포넌트 닫기버튼

EXHIBITION

다른 이벤트



영화회색늬

관람료	10000원
장소	익산박물관
시작일	2022-09-06
마감일	2022-10-01

내용 영화다

Event, Exhibition PK 읽어 Read하기

```
import { onMount } from "svelte";
import { callid } from "../store";
```

```
let callno
```

```
callid.subscribe(t=>{ callno=t})
```

```
// DB를 열어(특정 DB) ID 일치하는것 정보 가져오기!
```

```
let exhib=[]
```

```
const slidingbnn="http://192.168.0.51:8080/exhibition/"
```

```
onMount(async function(){
  const red= await fetch(slidingbnn)
  const open= await red.json()
  exhib=open._embedded.exhibition
})
```

```
</script>
```

```
<div id="total">
```

```
{#each exhib as exhibition}
```

```
{#if callno==exhibition.id}
```

```
<div id="poster">
```

```

```

```
<div id="content">
```

Exhibition 컴포넌트서 store한 내용 가져오기

가져온 번호로 PK찾아 내용 가져오기

게시판

Create 기능 부여

게시판

글쓰기

조회수 카운트
(DB자동 연결)

번호	장소	제목	작성자	작성일	조회
공지	본사	박물관 업데이트 안내	관리자	2022-09-08	24
공지	전주박물관	전주박물관 내부 공사 안내	관리-전주	2022-08-23	42
공지	공주박물관	공주박물관 수장고 안내	관리-공주	2022-08-19	21
공지	경주박물관	경주박물관 전시회 예정 정보입니다.	관리-경주	2022-08-16	7
공지	본사	육설 폭언은 삭제 및 법적 조치합니다.	관리자	2022-08-12	89
공지	본사	하반기 채용 공고	관리자	2022-08-19	36
공지	본사	서버점검 알림	관리자	2022-08-12	6
공지	본사	공지사항 알림	주인장	2022-08-08	24
23	전주박물관	글쓰기 테스트	사용자	2022-08-19	2
22	경주박물관	테스트	사용자	2022-08-19	4
21	대구박물관	asdfasdf	사용자	2022-08-19	3
20	경주박물관	테스트	사용자	2022-08-19	8
19	박물관	도배	사용자	2022-08-18	6
18	박물관	글 써지나	사용자	2022-08-18	6
17	박물관	박물관 인원제한 문의	사용자	2022-08-18	3
16	김해박물관	추가출토 유물 전시회	김대기 대리	2022-08-17	5
15	준천박물관	준천박물관 이용문의	사용자	2022-08-16	7
14	본사	테스트	관리자	2022-08-18	23

< 1 2 3 >

공지 제외 최신 글을
가장 위로 배치
글 개수 확인

특정 개수 넘어가면
페이지 자동생성

게시내용 확인 및 글쓰기

기입 내용 저장

게시판

사용자

제목 문의장소

비밀번호

내용을 입력해주세요

조건부 POST 기능

본인이 쓴 글을 삭제하기 위해서는 비밀번호가 필요합니다. 비밀번호를 기억해주시기 바랍니다.

욕설, 인신공격, 광고 글은 삭제되니 작성에 유의 바랍니다.

문의하여 주신 사항은 공지로 알려드립니다.

[뒤로가기](#)

DB 내용 호환

게시판

욕설 폭언은 삭제 및 법적 조치합니다.

담당자: 관리자 조회수: 89 작성일: 2022-08-1

게시판을 통해 욕설 및 폭언을 하는 이용자의 게시물은 삭제하고 이용을 제한하도록 저장해 유류드립니다. 아울러 욕설과 폭언의 수위가 심각할 경우 법적 조치를 취할 수 있습니다.

비밀번호를 입력하세요

[뒤로가기](#)

조건부 삭제 기능

```
function addno(ibs){
  inknum=ibs
  console.log(inknum)
}
```

```
$:boardid.update(t=>inknum)
$:boardcount.update(t=>inkcount)
```

```
let page=true
let content=false
let write=false
```

```
function aop(recnt){
  inkcount=recnt
  content=!content
  page=!page
  console.log(inkcount)
}
```

```
async function remove(pbd){
  await fetch(post+pbd.num,{
    method:"DELETE"
  })
  board=board.filter(t=>t!==pbd)
}
```

게시판 내용 확인 시
자동으로 조회수 증가

삭제
기능

JS store.js

museum > src > lib > JS store.js > ...

```
1 import { writable } from "svelte/store";
2
3 export const callid = writable(0);
4 export const boardid = writable(0);
5 export const boardcount = writable(0);
```

```
let callno
let modifcount
```

```
boardid.subscribe(t=>{callno=t})
boardcount.subscribe(t=>{modifcount=t})
```

```
const post="http://192.168.0.51:8080/postboard/?page=0&size=200"
let board=[]
```

```
onMount(async function(){
  const res=await fetch(post)
  const data = await res.json()
  board=data._embedded.postboard
  // console.log(board)
  // console.log(callno)
```

```
modifcount=modifcount+1
```

```
await fetch("http://192.168.0.51:8080/postboard/"+callno,{
  method:"PATCH", headers:{ //headers를 설정해야 오류가 안남
    "Content-Type":"application/json"
  },
  body:JSON.stringify({
    count:modifcount
  })
})
```

게시글 작성

```
let title, content, workplace, password
```

```
let official=false  
let whiter="사용자"
```

→ 공지 작성 불가, 이용자 이름 “사용자”로 고정

```
async function add(){
```

```
  const board = {
```

```
    | done: official , date : new Date , content: content, title: title, whiter: whiter, workplace: workplace, password: password
```

```
  }
```

```
  await fetch(post,{
```

```
    method:"POST", headers:{ //headers를 설정해야 오류가 안남
```

```
      "Content-Type":"application/json"
```

```
    },
```

```
    body:JSON.stringify({
```

```
      done:board.done, date : board.date, content : board.content, title: board.title, workplace: board.workplace, password: board.password, whiter: board.whiter
```

```
    })
```

```
  })
```

```
  boards=[board,...boards]
```

```
  title =""
```

```
  content =""
```

```
  workplace =""
```

```
  password =""
```

```
  whiter="사용자"
```

```
  official=false
```

```
  console.log(board)
```

```
}
```

```
create
```

→ 작성 후 내역 리셋

CREATE

Mall 파트

Web사이즈에 따라 변경
각 탭에 맞는 조건부
검색(컴포넌트 이용)

스크롤로 정보 나열



MANAGER APP

DB 직통 관리자 APP

관리자모드

Museum

Relic

Exhibition

Event

Board

Goods

각 Table마다 각자의 컴포넌트를 이용해 CRUD 가능

Manage App

Museum
Relic
Exhibition
Event
Board
Goods
Home

추가
장식물장비물
제목 [게시글 제목]
작성자 [이름+직책] 공지
내용
게시글 내용 적어주세요
저장

이벤트, 전시회 등 모든 Table 에 자료 추가 가능

삭제

고유번호	게시글 제목	작성자	기관	공지여부	비밀번호	작성일	
283	글쓰기 테스트	사용자	전주박물관	X	11111	2022-08-19	삭제
282	테스트	사용자	경주박물관	X	111111	2022-08-19	삭제
281	asdfasdf	사용자	대구박물관	X	2222	2022-08-19	삭제
278	테스트	사용자	경주박물관	X	11222	2022-08-19	삭제
272	도배	사용자	박물관	X	1234	2022-08-18	삭제
271	글 써지나	사용자	박물관	X	1234	2022-08-18	삭제
270	박물관 현황제한 문의	사용자	박물관	X	1234	2022-08-18	삭제
267	추가물트 유물 전시회	김대기 대리	김해박물관	X	0	2022-08-17	삭제
255	춘천박물관 이용문의	사용자	춘천박물관	X	1234	2022-08-16	삭제
244	테스트	관리자	본사	X	1234	2022-08-18	삭제
16	박물관 업데이트 안내	관리자	본사	공지	0	2022-09-08	삭제
15	제주박물관 이용시간 문의	제주스	이용자	X	0	2022-08-29	삭제

모든 글이 등록번호 역순으로(최신글이 가장위로) 정렬하여 관리자에서는 조건없이 삭제 가능

```
let title, content, whiter, workplace
let official=false
let password= 1111
```



기본값 설정

```
async function add(){
  const board = {
    done: official, date : new Date(), content: content, title: title, whiter: whiter, workplace: workplace, password: password
  }
  await fetch(savepoint,{
    method:"POST", headers:{ //headers를 설정해야 오류가 안남
      "Content-Type":"application/json"
    },
    body:JSON.stringify({
      done:board.done, date : board.date, content : board.content, title: board.title, whiter: board.whiter, workplace: board.workplace
    })
  })
  boards=[board,...boards]

  title =""
  content =""
  whiter =""
  workplace =""
  official=false
  password=1111
  console.log(board)
}
```

변경 값 저장

자동 설정(날짜)

```
async function remove(board){
  await fetch("http://192.168.0.51:8080/postboard/"+board.num,{
    method:"DELETE"
  })
  boards=boards.filter(t=>t!==board)
  alert("삭제되었습니다.");
}
```

Router

Home

Link

Path

```
<script>
  import {Router, Route} from "../node_modules/svelte-routing";
  import Board from "../lib/Board.svelte";
  import Event from "../lib/Event.svelte";
  import Exhibition from "../lib/Exhibition.svelte";
  import Goods from "../lib/Goods.svelte";
  import Home from "../lib/Home.svelte";
  import Museum from "../lib/Museum.svelte";
  import Relic from "../lib/Relic.svelte";

  export let url=""
```

```
</script>
```

```
<Router url={url}>
```

```
  <!-- <nav>
```

```
    <Link to="/">Home</Link>
```

```
    <Link to="infomation">Infomation</Link>
```

```
    <Link to="search">Search</Link>
```

```
    <Link to="option">Option</Link>
```

```
    <Link to="/">Exit</Link>
```

```
  </nav> -->
```

```
  <div>
```

```
    <h1>관리자모드</h1>
```

```
  </div>
```

```
  <div>
```

```
    <Route path="home" component={Home}><Home/></Route>
```

```
    <Route path="museum" component={Museum}></Route>
```

```
    <Route path="relic" component={Relic}></Route>
```

```
    <Route path="exhibition" component={Exhibition}></Route>
```

```
    <Route path="event" component={Event}></Route>
```

```
    <Route path="board" component={Board}></Route>
```

```
    <Route path="goods" component={Goods}></Route>
```

```
    <Route path="/"><Home/></Route>
```

```
  </div>
```

```
</Router>
```

Feedback



Update

기존 DB에 등록된 정보 변경이
어렵다(postman 필요)
이에 관하여 수정 보완이 필요



Over Table

초기 기획과는 달리 필요 없는
TABLE을 만들어 의미 없는 공간이
생성되었다.



Pagination

TABLE을 전체 검색은 페이지가 잘
나뉘 지지만 조건부 검색을 할
경우 페이지 나누는 기준은
TABLE로 나온다



Bootstrap

Footer을 사용함에 있어 호환이
되지 못하여 일부 페이지 오류로
인해 수정하지 못했다.



Limit DB

DB를 생성시 각 행마다 조건을
붙이지 못해 notNull 등의 특이
사항을 제대로 기입되지 못했다.



Thanks!

감사합니다!

Any questions?