

# 블로그 만들기

2조

정현규, 임송재, 김민정, 오탈경, 이경환



# Firebase



Firebase는 웹과 모바일 개발에 필요한 기능을 제공하는 **BaaS(Backend as a Service)**입니다. 쉽게 말해 백 엔드 개발을 통해 서버를 따로 설계, 구현하지 않고 프론트 엔드 개발에 집중할 수 있도록 도와주는 서비스입니다



# Firebase



Authentication



간편하고 안전한 방법으로 사용자를 관리해줍니다.

Firebase 인증은 이메일/비밀번호, 타사 **제공업체** 기존 계정 시스템 직접 사용 등의 다양한 인증 방법을 제공합니다.



Firestore Database



파이어베이스 및 구글 클라우드 플랫폼의 모바일, 웹, 서버 개발에 사용되는 유연하고 확장 가능합니다.



Realtime Database



NoSql클라우드 데이터베이스에 JSON 형태로 데이터를 저장하고 클라이언트에 실시간으로 동기화 합니다.



# Firebase

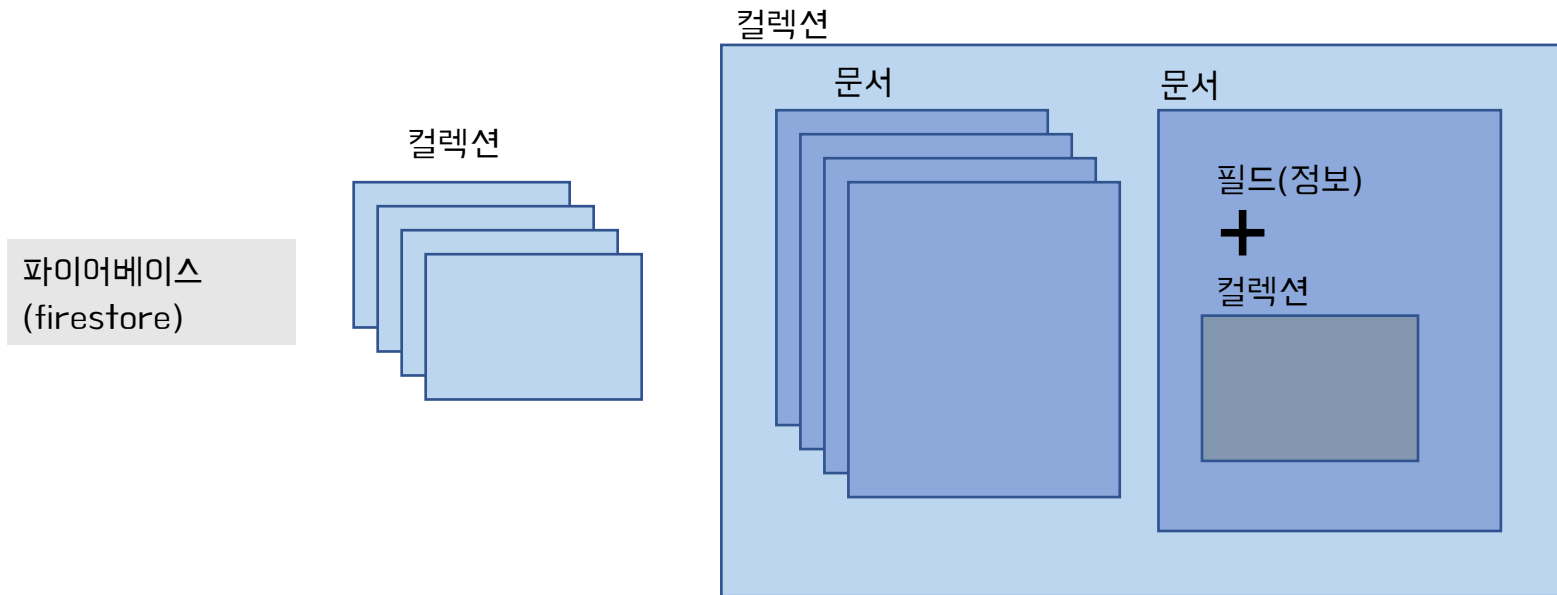


사용자가 제작한 이미지, 오디오, 동영상의 콘텐츠를 저장하고 필요에 따라 저장된 콘텐츠를 제공합니다.



HTML, CSS, 자바스크립트 등과 같은 정적 콘텐츠를 빠르고 안전하게 호스팅해줍니다. SSL을 제공하기 때문에 안전하게 콘텐츠를 전송할 수 있으며 전세계 CDN에 캐싱되어 있기 때문에 사용자가 어디에 있든지 제한없이 콘텐츠를 빠르게 전송이 가능합니다.

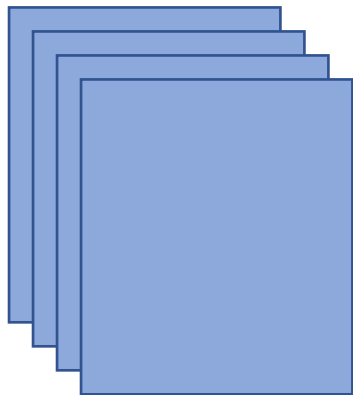
# 파이어베이스 구조



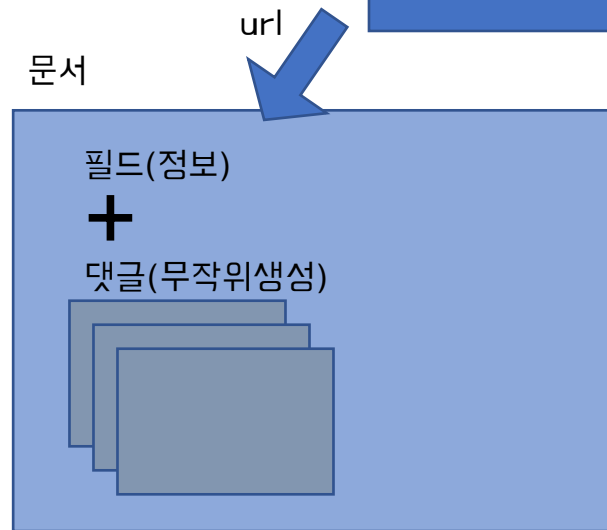
## 2조의 데이터베이스



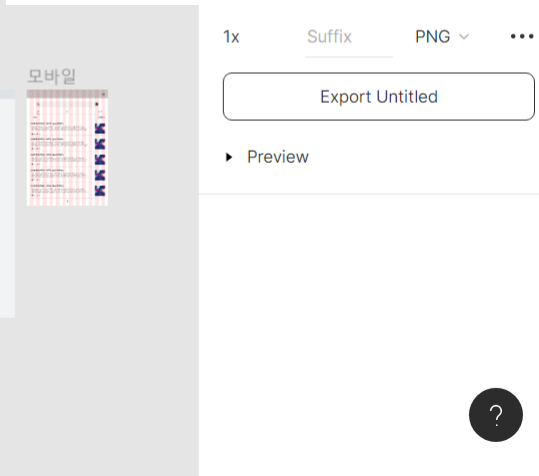
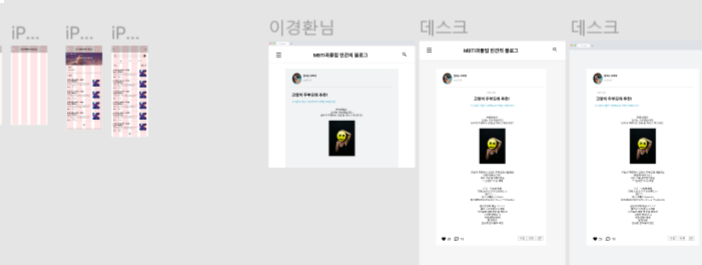
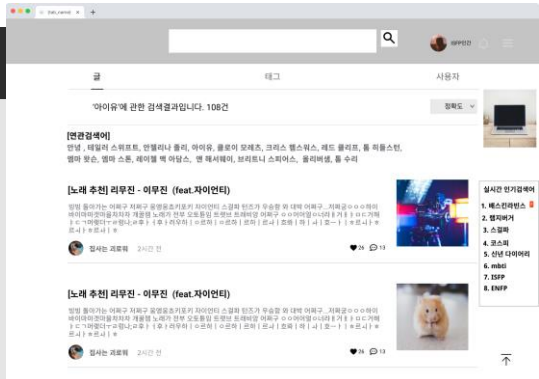
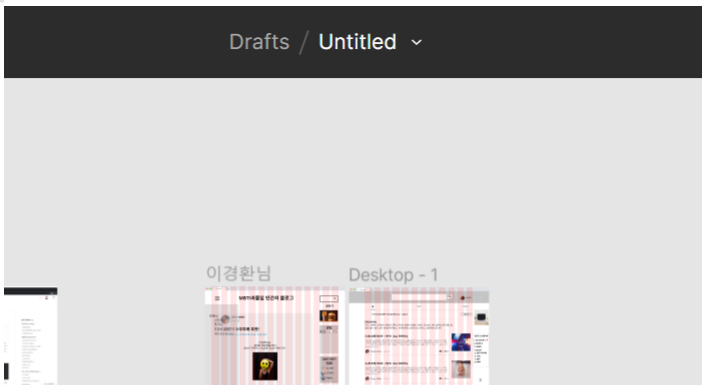
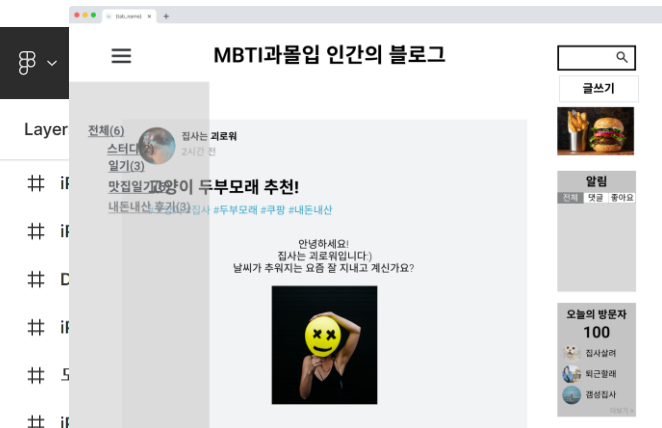
문서(시간순)

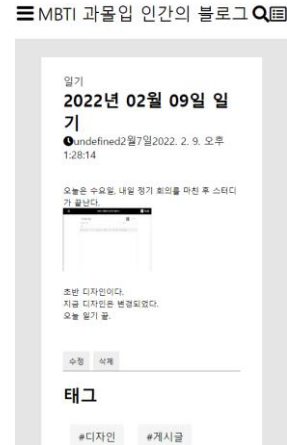


문서



storage(이미지)





# 게시글 페이지 화면구성



# 쿼리스트링-문서와 문서를 연결

≡ 🔍 MBTI과몰입 인간의 블로그

전체글 인기순 최신순

study2 hansei  
1일 28분 46초 전

≡ MBTI 과몰입 인간의 블로그 🔍

2022년 02월 09일 일요일  
#스터디 #일기 #게시글 #디자인

오늘은 수요일, 내일 정기 회의를 마친다. 초반 디자인이다. 지금 디자인은 일기 끝.

스터디  
마무리 하는 날.  
🕒 20222월5일오후 2:54:38

오늘 열심히 마무리 합니다.  
끝.

수정 삭제

태그  
#날 #마지막 #오늘

댓글

정현규/컴퓨터공학과  
Wed Feb 09 2022 22:22:59 GMT+0900 (한국)



≡ 🔍 MBTI과몰입 인간의 블로그

전체글 인기순 최신순

study2 hansei  
1일 28분 46초 전

≡ MBTI 과몰입 인간의 블로그 🔍

2022년 02월 09일 일기  
#스터디 #일기 #게시글 #디자인

오늘은 수요일, 내일 정기 회의를 마친 후 스터디. 초반 디자인이다. 지금 디자인은 변경되었 일기 끝.

스터디  
마무리 하는 날.  
🕒 20222월5일오후 2:54:38

오늘 열심히 마무리 합니다.  
끝.

수정 삭제

태그  
#날 #마지막 #오늘

댓글

정현규/컴퓨터공학과



```
'./post.html?id=${i.data().docId}'
```

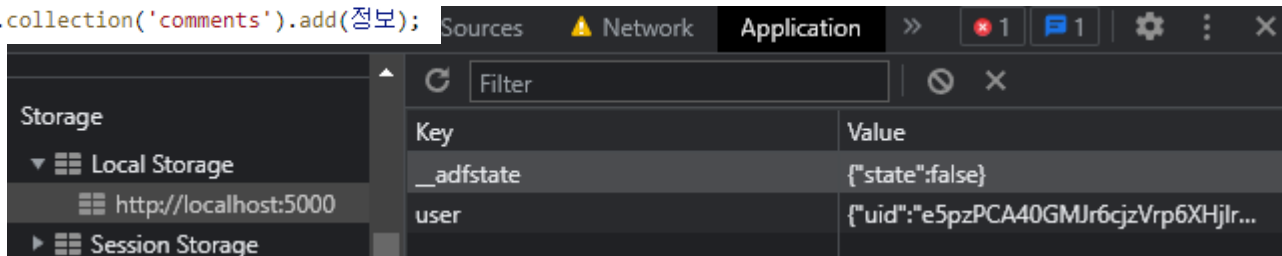
```
var params = new URLSearchParams(window.location.search)

//본문//
db.collection('post').doc(params.get('id')).get().then((result) => {
```

# 로그인 정보를 확인, local storage에 저장

```
//////////로그인 정보 확인//////////
firebase.auth().onAuthStateChanged((user) => {
  if (user) {
    localStorage.setItem('user', JSON.stringify(user))
  }
})
var myUid = JSON.parse(localStorage.getItem('user')).uid;
var mydisplayName = JSON.parse(localStorage.getItem('user')).displayName;
//////////댓글작성//////////
$('#comment-send').on("click", function () {
  var 정보 = {
    content: $('#form-control').val(),
    date: new Date(),
    uid: JSON.parse(localStorage.getItem('user')).uid,
    displayName: JSON.parse(localStorage.getItem('user')).displayName,
    photoURL: JSON.parse(localStorage.getItem('user')).photoURL,
  }
  db.collection('post').doc(params.get('id')).collection('comments').add(정보);
});
```

로그인 정보를 확인해서 localStorage에 저장시키고 localStorage에 있는 uid, displayName과 photoURL을 저장하는 코드



# 게시글 작성 페이지 화면

MBTI 과몰입 인간의 불로그

Q

게시글 작성

등록취소

카테고리 선택

공개

제목

B

U

-

14

Segoe UI

A

T

#해시태그

MBTI 과몰입 인간의...

Q

게시글 작성

등록취소

카테고리 선택

공개

제목

B

U

-

14

Segoe UI

A

T

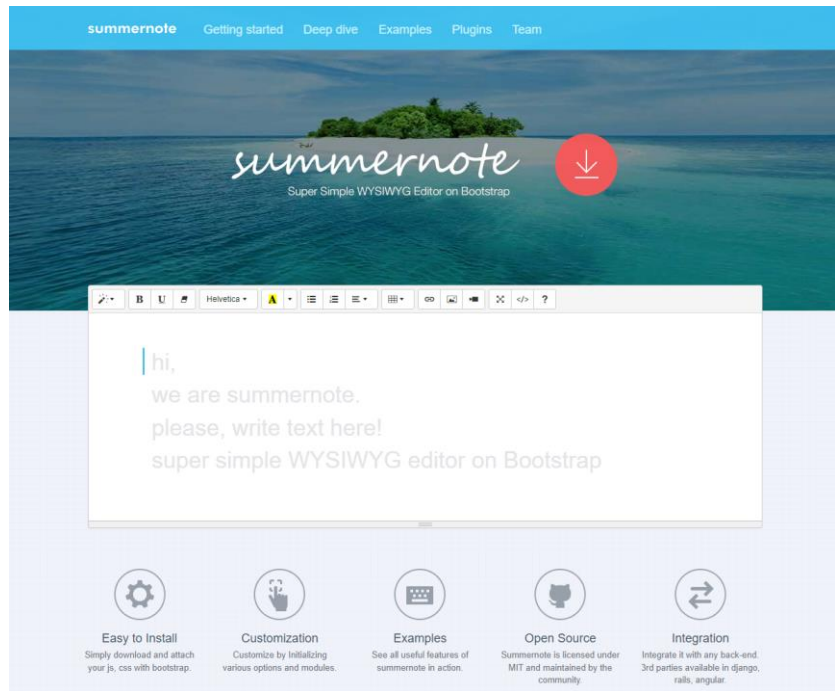
#해시태그

# Summernote 에디터

무료 오픈소스 WYSIWYG 텍스트 에디터 사용.

사용하기 쉽고, 많은 예제들을 쉽게 찾아볼 수 있었기 때문에 선택.

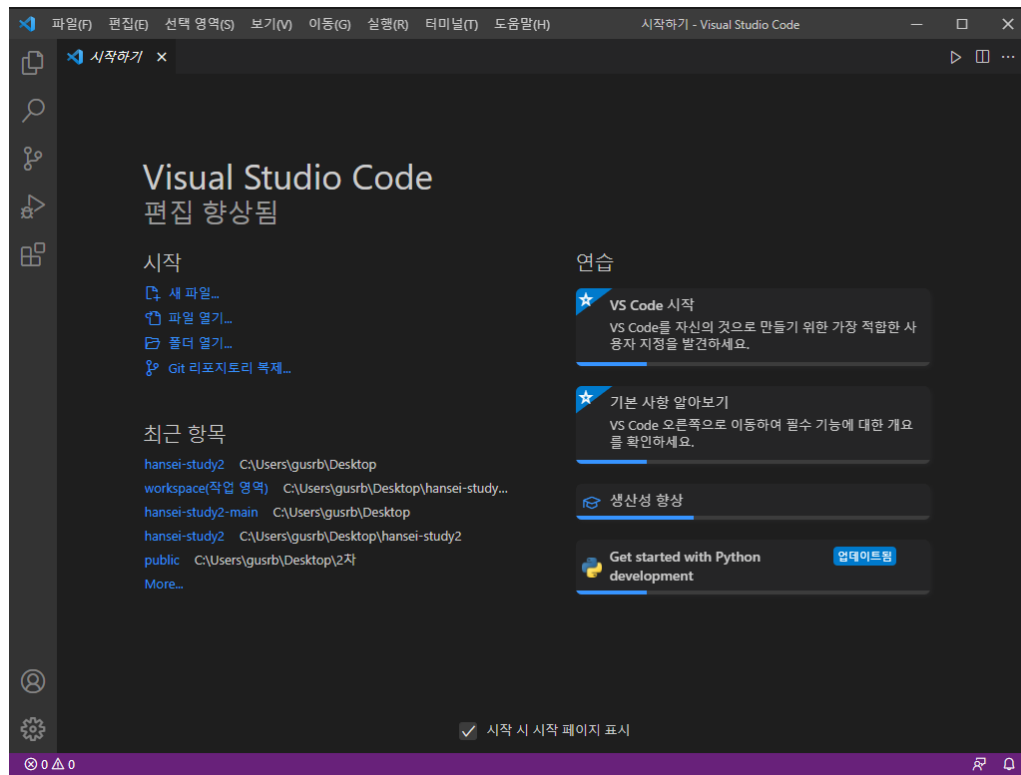
특히, 한국인 개발자가 참여하여 개발.



# Visual Studio Code

마이크로소프트에서 개발한 텍스트 에디터

크로스 플랫폼(윈도우, macOS, 리눅스)을 지원하며 다양한 확장 기능을 지원하며 마이크로소프트가 GitHub을 인수함에 따라 Atom의 자리를 바꿀것을 보인다.



# 이미지 첨부

Summernote 에 이미지를 첨부 시, 이미지가 존재하는 local 경로로 img 태그가 삽입되는데 summernote 에서 제공하는 함수를 사용하여 첨부한 이미지를 firebase storage에 업로드 한 뒤, 해당 경로를 가져와서 img 태그를 설정 후 삽입하는 방식으로 해결.

```
// Summernote 이미지 첨부 시 실행
const uploadImage = (file, editor) => {
  console.log('Upload start')

  const image = file;
  const storageUrl = 'images/' + image.name
  const storageUpRef = storage.ref()
  const task = storageUpRef.child(storageUrl).put(file)

  // console.log('storageRef: ', storageUpRef)
  // console.log('image is: ', image.name)

  // Summernote 이미지 업로드 후 해당 링크로 경로 설정
  task.on('state_changed', (snapshot) => {
    console.log('Uploading...')
  }, (error) => {
    console.log('!!Uploading Error!! => ', error)
  }, () => {
    console.log('Upload Success!!')
    const storageRef = storage.ref()
    storageRef.child(storageUrl).getDownloadURL().then((url) => {
      // console.log('Image URL => ', url)
      const imageUrl = $('<img>').attr({
        src: url
      })
      $('#summernote').summernote('insertNode', imageUrl[0])
    }).catch((error) => {
      console.log('Get Url ERROR => ', error)
    })
  })
}
```

# 게시글 수정

URL 파라미터를 이용하여 게시글에서 id값을 가져온 후, 해당 문서를 조회하여 데이터를 불러오는 방식으로 코드 구성.

```
314     ... // 게시글 수정 부분
315     ... const updateDocId = new URLSearchParams(window.location.search).get('id');
316     ... if (updateDocId) {
317     ...   db.collection('post').doc(updateDocId).get().then(async (res) => {
318     ...     const content = await res.data();
319     ...     console.log(content)
320     ...     const {
321     ...       category,
322     ...       open,
323     ...       title,
324     ...       description,
325     ...       hashtag
326     ...     } = content;
327     ...     // console.log(description)
328     ...     document.querySelector(`#${category}`).setAttribute('selected', true);
329     ...     document.querySelector(`#${open}`).setAttribute('selected', true)
330     ...     document.querySelector('#title').value = title
331     ...     $('#summernote').summernote('pasteHTML', description)
332     ...     // console.log(description)
333     ...     document.querySelector('#hashtag').value = hashtag
334     ...   })
```

# 정렬기능

```
<script>
  async function change(id){ //await 쓸 거라 async 꼭 쓰기!! 아니면 오류남
    const postsRef = db.collection('post'); // 참조객체 만들기

    id = document.getElementById("filter");
    if(id.value=="date"){
      const firstThreeRes = await postsRef.orderBy('좋아요','desc').limit(1).get(); // 참조객체에 따라 좋아요를 내림
      firstThreeRes.forEach(doc => {
        console.log(doc.id,'->',doc.data()); // 콘솔에 해당 데이터 표시
      });
    }
    else{
      return;
    }
  }
</script>
```

비동기식으로 정렬이 되게 구현하기

문서는 기본적으로 시간순으로 정렬되어있음, 하지만 이를 좋아요가 많은 순으로 보고 싶을 땐  
쿼리 기능을 사용할 수 있다

참조객체를 만든 후 orderBy를 통해 정렬한 후, 화면에 표시되는 부분에 뿌려준다.



# 검색기능

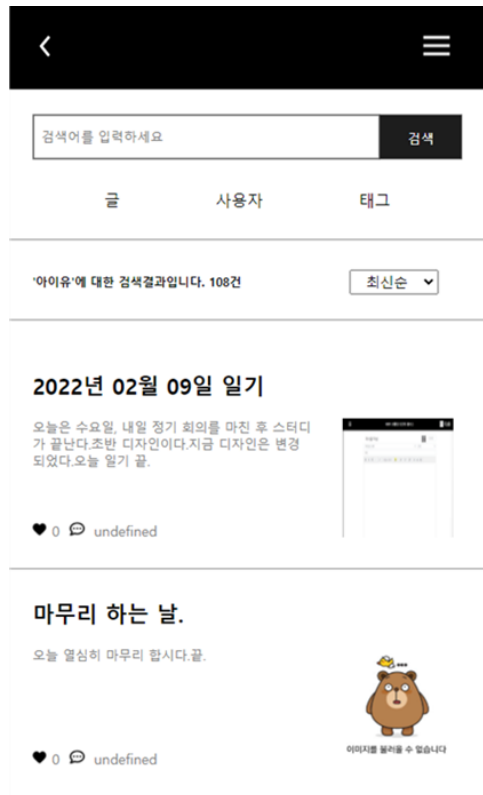
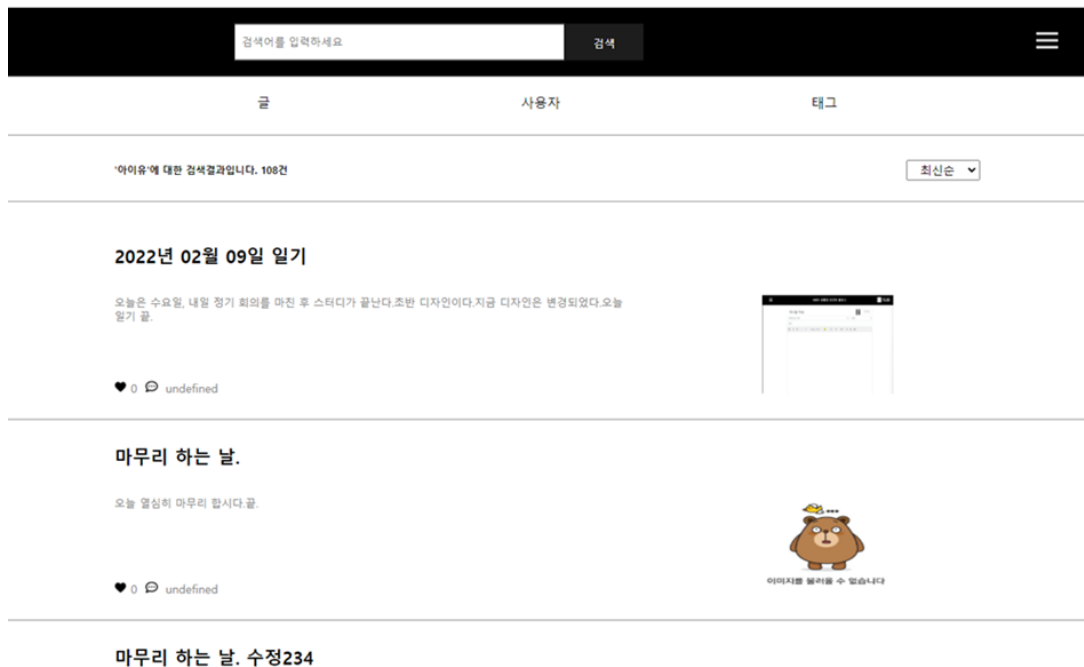
```
var kw = document.querySelector(".search").value; //search클래스의 value를 가져옴
console.log(kw)
document.querySelector('.media').innerHTML = ``; // media클래스를 비운 후
db.collection("post").where(`title`,`==`, kw) // 검색창에 쓴 제목과 db안의 제목들 중 같은 목록을 찾음
.get()//가져와라
.then(function(querySnapshot) { //QuerySnapshot 을 얻으면 컬렉션 정보가 다 들어있음
```

문제점이 있다면... 정말 정확하게 입력해야 검색결과가 나온다는 것과... 결과물이 여러개 나온다는 거... 추후에 고쳐봐야겠다...

상단의 input 태그인 .search의 텍스트값을 가져온 후  
앞서 했던대로 콘텐츠 표시된 곳을 비워준다.

그 후 .search안의 키워드와 db내의 게시물 중 제목필드를 비교하여  
그 위치를 찾은 후 그것들을 다시 부리는 방식을 사용하였다

하지만 게시물이 여러개씩 나오고 정확하게 입력하지 않으면 안나온다  
-> 그래서 파이어베이스는 검색기능의 경우 서드파티 사용을 권한다.



반응형 홈페이지를 미디어쿼리를 통해 구현

# 블로그 홈 화면 구성



## MBTI과몰입 인간의 블로그



study2 hansei

1일 3시간 10분 35초 전

2022년 02월 09일 일기

#스터디, #일기, #게시글, #디자인

오늘은 수요일, 내일 정기 회의를 마친 후 스터디가 끝난다. 초반 디자인이다. 지금 디자인은 변경되었다. 오늘 일기 끝.





# 블로그 홈 화면 구성 코드

```
<body>
  <header>
    <div class="MenuButton">
      <input type="checkbox" id="menuicon" />
      <label for="menuicon">
        <span></span>
        <span></span>
        <span></span>
      </label>
      <div class="sidebar">...
    </div>
  </div>
  <div class="mobile-searching">
    <a href="search.html"><i class="fas fa-search"></i> </a>
  </div>
  <div class="blogname">MBTI과몰입 인간의 블로그</div>
  <div class="mobile-category">
    <input type="checkbox" id="category" />
    <label for="category">...
  </label>
  <div>...
</div>
<div class="icons">
  <a href="search.html"><i class="fab fa-sistrix"></i></a>
</div>
</header>
```

# 블로그 홈 화면 CSS

```
header {
  display: flex;
  justify-content: space-between;
  padding: 25px 60px;
  align-items: center;
  background: #ffff;
  /* background-color: #c4c4c4; */
  position: relative;
}

header .mobile-searching {
  font-size: 30px;
  display: none;
}

header .mobile-searching a {
  text-decoration-color: none;
  color: black;
}

header .mobile-category {
  display: none;
}

header .mobile-category img {
  height: 30px;
  width: 30px;
}

header .mobile-category input {
  display: none;
}

header .mobile-category label:active {
  transform: translate(5%, 5%);
}

header .mobile-category label:hover {
  color: #27a3ca;
  cursor: pointer;
}

header .mobile-category > input:checked + label + div {
  display: block;
```

```
header .mobile-category label + div > div {
  border-bottom: 1px solid #c4c4c4;
  display: flex;
  align-items: center;
  padding: 4px 15px;
  font-weight: 900;
  font-size: 20px;
}

header .mobile-category label + div > div .fas {
  font-size: 25px;
  padding: 10px 10px;
  margin-right: 10px;
}

header .mobile-category label + div ul {
  list-style: none;
  padding: 0px 10px;
}

header .mobile-category label + div ul li {
  display: flex;
  padding: 10px 0px;
  justify-content: space-between;
  border-bottom: 0.5px solid #c4c4c4;
}

header .mobile-category label + div ul li:hover {
  font-weight: 900;
}

header .mobile-category label + div ul li a {
  text-decoration: none;
  color: black;
}

header .mobile-category label + div ul li:nth-child(1),
header .mobile-category label + div ul li:nth-child(7) {
  border-bottom: 0.5px solid gray;
}

header .blogname {
  font-size: 30px;
```

```
header .mobile-category label + div ul li:nth-child(7) {
  border-bottom: 0.5px solid gray;
}

header .blogname {
  font-size: 30px;
  font-weight: 900;
  font-family: inherit;
  overflow: hidden;
  white-space: nowrap;
  text-overflow: ellipsis;
  position: absolute;
  left: 50%;
  top: 50%;
  transform: translate(-50%, -50%);
}

header .blogname a {
  text-decoration: none;
  color: #000;
}

header .icons {
  font-size: 30px;
  display: flex;
  align-items: center;
  position: relative;
}

header .icons a {
  text-decoration-color: none;
  color: black;
}
```



study2 hansei

1일 3시간 10분 35초 전

## 2022년 02월 09일 일기

#스터디, #일기, #게시글, #디자인

오늘은 수요일, 내일 정기 회의를 마친 후 스터디가 끝난다. 초반 디자인이다. 지금 디자인은 변경되었다. 오늘 일기 끝.



# 반응형 모바일 홈 화면 구성

## 데스크탑 홈 화면과 다른 점

- 모바일 홈 화면 header 추가 (전체글, 인기순, 최신순, 레아웃 정렬 방법)
- 인기순 최신순으로 게시물들 정렬 구현
- 홈 header 폰트(메뉴바 돋보기 블로그이름) 크기 조절
- 홈 header 돋보기 위치 왼쪽으로 이동

## 반응형 모바일 홈 화면 HTML

```
<!-- 모바일 전용 정렬 방법 -->
<div class="range">
  <div>
    <span class="normal">전체글</span> <span class="popular">인기순</span>
    <span class="recent">최신순</span>
  </div>
  <div class="range-icon">
    <i class="fas fa-list"></i><i class="fas fa-grip-lines"></i>
  </div>
</div>
<script type="text/javascript" src="range.js"></script>

<div class="wrapper">
  <!-- <div class="main-box">...
</div>

<div class="mobile-box">tnemnorivnelee, a day ago • 게시글 프로필 수정
  <!-- <div class="mobile-wrapper">...
</div>
```



## 반응형 모바일 홈 화면 CSS

```
.range {  
  display: none;  
  justify-content: space-between;  
  align-items: center;  
  width: 100%;  
  padding: 4px 37px;  
  border-bottom: 0.5px solid #f2f3f4;  
  font-family: 'inherit';  
  font-size: 14px;  
}  
  
.range span:hover {  
  font-weight: 900;  
  cursor: pointer;  
}  
  
.range .range-icon i {  
  border-radius: 100%;  
  padding: 10px 10px;  
}  
  
.range .range-icon i:hover {  
  cursor: pointer;  
  background-color: #979797;  
}
```

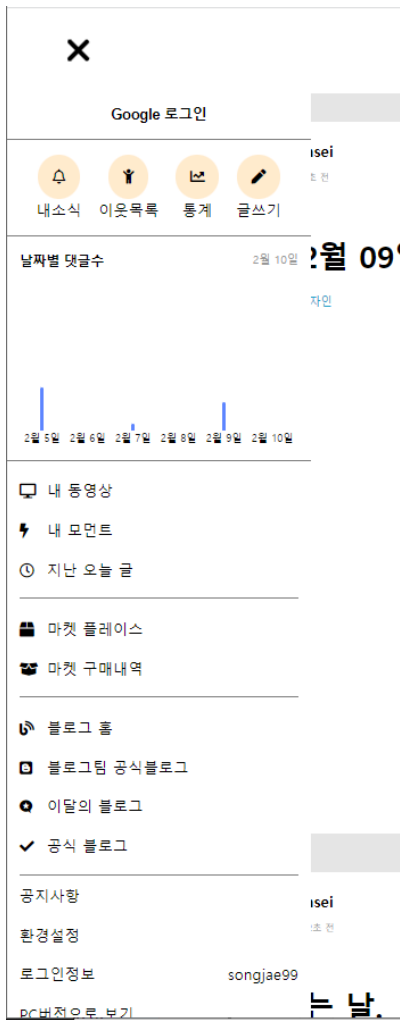
# 반응형 모바일 홈 화면 CSS

```
@media screen and (max-width: 991px) {
  header {
    position: relative;
    padding: 25px, 60px;
    justify-content: space-between;
  }
  header .icons {
    position: static;
    display: none;
  }
  header .mobile-searching {
    font-size: 30px;
    display: block;
    position: absolute;
    left: 110px;
    top: 20px;
  }
  header .mobile-category {
    display: block;
  }
  header .mobile-category label + div {
    width: 100%;
    height: 362px;
    position: absolute;
    top: 85px;
    left: 0;
    display: flex;
    flex-direction: column;
    background-color: white;
    font-family: monospace;
    display: none;
  }
  .wrapper {
    display: block;
  }
  .wrapper .main-box {
    margin-left: auto;
    margin-right: auto;
  }
}
```

```
.wrapper .main-box {
  margin-left: auto;
  margin-right: auto;
  width: 80%;
}
.wrapper .main-box .contents-box {
  margin-bottom: 30px;
}
.wrapper .side-box {
  display: none;
}
}

@media screen and (max-width: 700px) {
  .range {
    display: flex;
  }
  .sidebar {
    top: 72px;
  }
  #menuicon + label {
    width: 13px;
    height: 13px;
  }
  #menuicon + label > span {
    height: 2px;
  }
  header .blogname {
    font-size: 13px;
  }
  header .mobile-category img {
    height: 13px;
    width: 13px;
  }
  header .mobile-searching {
    font-size: 13px;
    top: 27px;
    left: 80px;
  }
  header .mobile-category label + div {
    top: 72px;
  }
}
```

```
@media screen and (max-width: 700px) {
  .range {
    display: flex;
  }
  .sidebar {
    top: 72px;
  }
  #menuicon + label {
    width: 13px;
    height: 13px;
  }
  #menuicon + label > span {
    height: 2px;
  }
  header .blogname {
    font-size: 13px;
  }
  header .mobile-category img {
    height: 13px;
    width: 13px;
  }
  header .mobile-searching {
    font-size: 13px;
    top: 27px;
    left: 80px;
  }
  header .mobile-category label + div {
    top: 72px;
  }
}
```



->



header 부분 메뉴바 눌렀을 시

구현한 기능들

- 로그인
- 글쓰기 링크 연결
- 날짜별 댓글수 통계
- 메뉴 버튼 눌렀을 시 모양 변경

# 블로그 홈 화면 메뉴바 구성 코드

```
<div class="sidebar">
  <div class="sidebar-box1">
    <button id="logInBtn">Google 로그인</button>
    <div id="Myinfo">...
  </div>
  <button id="logOutBtn" style="display: none; margin-left: auto;" ...
</button>
</div>

<ul class="sidebar-box2">
  <li>
    <a href="#"><i class="far fa-bell"></i></a>내소식
  </li>
  <li>...
</li>
  <li>...
</li>
  <li>...
</li>
</ul>

<div class="sidebar-box3">
  <div>
    <span>날짜별 댓글수</span>
    <span class="date"></span>
    <script>
      let i = new Date();
      let date = i.getMonth() + 1 + "월 " + i.getDate() + "일";
      document.querySelector(".date").innerHTML = date;
      console.log(date)
    </script>
  </div>
  <ul class="graph">
```

```

<ul class="graph">
  <li>
    <div class="rect1">...
    </div>
    <p id="one"></p>
    <script>
      i = new Date();
      date = i.getMonth() + 1 + "월 " + (i.getDate() - 5) + "일";
      document.querySelector("#one").innerHTML = date;
    </script>
  </li>
  <li>
    <div class="rect2">...
    </div>
    <p id="two"></p>
    <script>...
    </script>
  </li>
  <li>...
</li>
<li>...
</li>
<li>...
</li>
<li>...
</li>
<li>...
</li>
</ul>
</div>

```

```

<div class="sidebar-box4">
  <ul>
    <a href="#">
      <li><i class="fas fa-desktop"></i>내 동영상</li>
    </a>
    <a href="#">...
    </a>
    <a href="#">...
    </a>

```

```

<div class="sidebar-box4">
  <ul>
    <a href="#">
      <li><i class="fas fa-desktop"></i>내 동영상</li>
    </a>
    <a href="#">...
    </a>
    <a href="#">...
    </a>
  </ul>
  <ul>...
</ul>
</div>
<div class="sidebar-box5">
  <ul>
    <a href="#">
      <li>공지사항</li>
    </a>
    <a href="#">...
    </a>
    <ul>...
    </ul>
    <a href="#">...
    </a>
    <a href="#">...
    </a>
  </ul>
</div>
</div>

```

# 블로그 홈 화면 메뉴바 통계부분 CSS

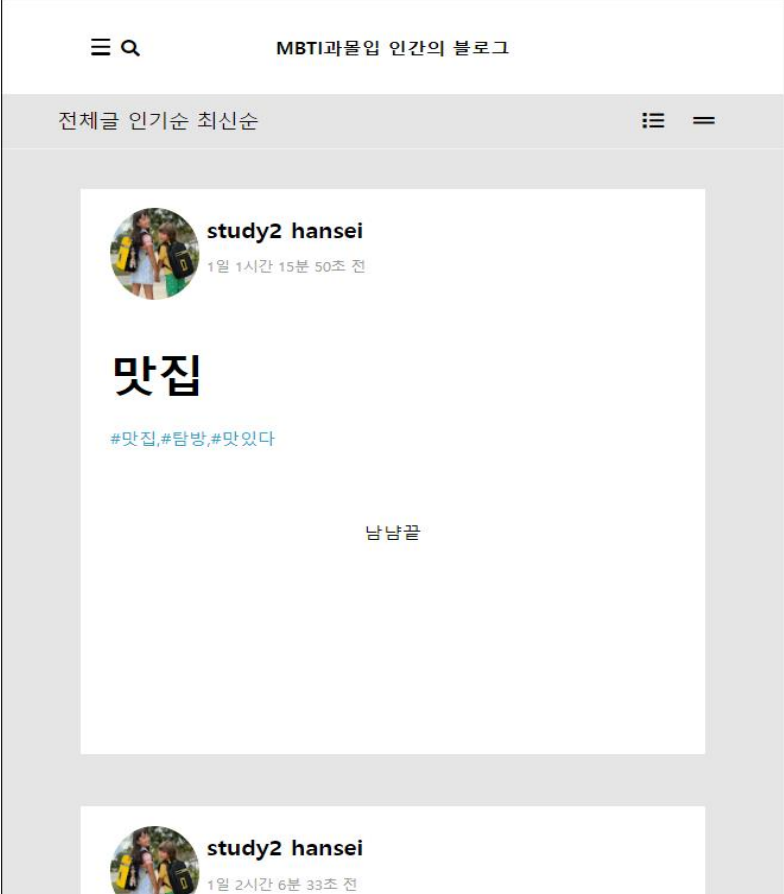
```
.sidebar .sidebar-box3 {  
  padding: 16px 13px;  
  border-bottom: 2px solid gray;  
  display: flex;  
  flex-direction: column;  
}  
.sidebar .sidebar-box3 > div {  
  display: flex;  
  justify-content: space-between;  
  margin-bottom: 20px;  
}  
.sidebar .sidebar-box3 > div span:nth-child(1) {  
  font-weight: 900;  
  font-size: 14px;  
}  
.sidebar .sidebar-box3 > div span:nth-child(2) {  
  color: #979797;  
  font-size: 12px;  
}  
.sidebar .sidebar-box3 .graph {  
  display: flex;
```

```
.sidebar .sidebar-box3 .graph li {  
  padding: 0 5px;  
  text-align: center;  
  font-size: 11px;  
}  
.sidebar .sidebar-box3 .graph li div {  
  position: relative;  
  width: 4px;  
  height: 150px;  
  border-radius: 4px 4px 0 0;  
  margin: 0 auto;  
}  
.sidebar .sidebar-box3 .graph li div span {  
  position: absolute;  
  left: 0;  
  right: 0;  
  bottom: 0;  
  background-color: #5e86ff;  
  border-radius: 4px 4px 0 0;  
}
```

모바일 홈 화면 리스트 정렬 방식 눌렀을 때 화면



모바일 홈 화면 햄버거 방식 눌렀을 때 화면



# 모바일 레이아웃 정렬 방식 눌렀을 때 코드

JS range.js > ...

tnemnorivnelee, 18 hours ago | 1 author (tnemnorivnelee)

```
1  const list = document.querySelector('.fa-list');
2  const line = document.querySelector('.fa-grip-lines');
3
4  list.addEventListener('click', () => {
5      $('.mobile-box').css('display', 'block');
6      $('.wrapper').css('display', 'none');
7  });
8
9  line.addEventListener('click', () => {
10     $('.mobile-box').css('display', 'none');
11     $('.wrapper').css('display', 'flex');
12 });
13
```



# 모바일 게시물 정렬 방식 놀렀을 때 코드

```
// 인기순 정렬
popular.addEventListener("click", () => {
  postsRef
    .orderBy("like", "desc")
    .get()
    .then((결과) => {
      document.querySelector(".wrapper").innerHTML = ``; // 공백화
      document.querySelector(".mobile-box").innerHTML = ``; // 공백화
      결과.forEach((i) => {
        const post = `
          <div class="main-box">
            <div class="info">
              
              <div class="nickname">
                <p class="name">${userName}</p>
                <p class="timer">${timeFunc(i.data().docId)}</p>
              </div>
            </div>
            <div class="title">
              <h1><a href="/post.html?id=${i.data().docId}">${i.data().title}</a></h1>
              <ul class="hashtag">
                <li><a href="#">${i.data().hashtag}</a></li>
              </ul>
            </div>
            <div class="content">
              <div class="greeting">
                <p>${i.data().text}</p>
              </div>
              
            </div>
          </div>`;

```

```
const MobileClick = `
  <div class="mobile-wrapper">
    <div class="mobile-main-box">
      <div><a href="/post.html?id=${i.data().docId}">${i.data().title}</a></div>
      <div>${i.data().hashtag}</div>
      <div>${timeFunc(i.data().docId)}</div>
      <div>
        <i class="fas fa-heart"></i> ${i.data().like}
        <i class="far fa-comment-dots"> ${i.data().commentNumber} </i>
      </div>
    </div>
    <div class="mobile-side-box"></div>
  </div>`;

$(".wrapper").append(post);
$(".mobile-box").append(MobileClick);
});

// 최신순 정렬
recent.addEventListener("click", () => {
  postsRef
    .orderBy("docId", "desc")
    .get()
    .then((결과) => { ...
  });
});
```

**감사합니다! 😊**