

README

프로젝트

Doit은 "하루하루 목표를 이루면서 그 목표를 습관으로 바꿔나가는 커뮤니티"으로 서로 인증해 주기, 금잔디로 자신이 한 일들을 한눈에 확인, 물품 거래 등 다양한 기능으로 목표 달성에 좀 더 다가가기 쉽도록 도와줍니다.

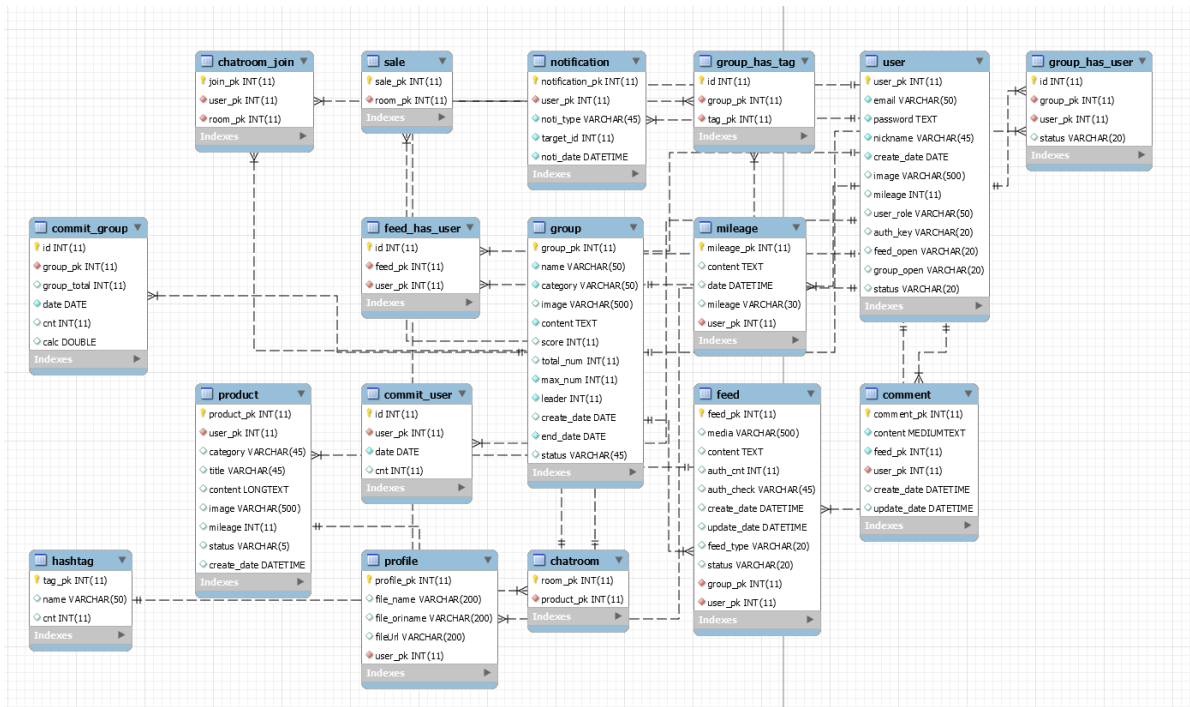
개발 환경



목표

- 반응형 웹
- axios 모듈화
- 채팅 구현

1. 데이터 모델링



- 유저정보, 커뮤니티, 그룹, 채팅 등 다양한 기능이 있었기 때문에 매우 복잡한 ERD를 가지게 되었다.
- 기능 수정이 반복적으로 이루어지면서 데이터 모델링도 수차례 변경 되었다.

2. 디자인

저희 Doit은 반응형 디자인을 통해 웹, 모바일 환경 모두에서 이용할 수 있게 디자인 하였습니다.



만들고 싶은 습관을 검색해보세요!

카테고리



Doit에서 가장 많이 나온 키워드를 확인해보세요!



Doit 그룹영향

아직 월간 그룹 순위가 생성되지 않았어요!
참여한 그룹에 인증글을 올리
인증마크를 획득하고 그룹 행동을 늘려보세요!



공부

X

총 7 개의 검색 결과가 있습니다.

수학의정석
수학 I
홍성대 커스
수학공부
#수학 #수학의 정석 #공부
상세보기

BIG DATA
빅데이터 공부 하실분
#공부 #빅데이터
상세보기

English
공부 모임 (영어)
#영어 #영어는 '못해도' #공부
상세보기

1일 1캠블리
#캠블리 #화상영어 #영어회화 #영어 #공부
상세보기

c108
#자바 #공부
상세보기

DO YOU SPEAK ENGLISH?
1일 1화상영어
#화상영어 #영어회화 #공부 #영어 #캠블리 #만병천유론
상세보기



투시투시

멤버 : 6/6

~ 2021-08-01

8만원 #투시투시 #7월

그룹 시작 날짜: 2021-08-08

그룹 수장

만들지도 그룹정보

템 스크어 : 0 그룹원 수 : 6

모두 투시투시의 다이어트 해볼까요...!!
7월까지 운동하고 여름에 몸 자랑 하요...

그룹 규칙

- 자킨은 월 1화만 먹기
- 인증글 올리는 '투시투시' 붙이기 <- 핵심
- 주말에는 몸무게 인증하기



이승준



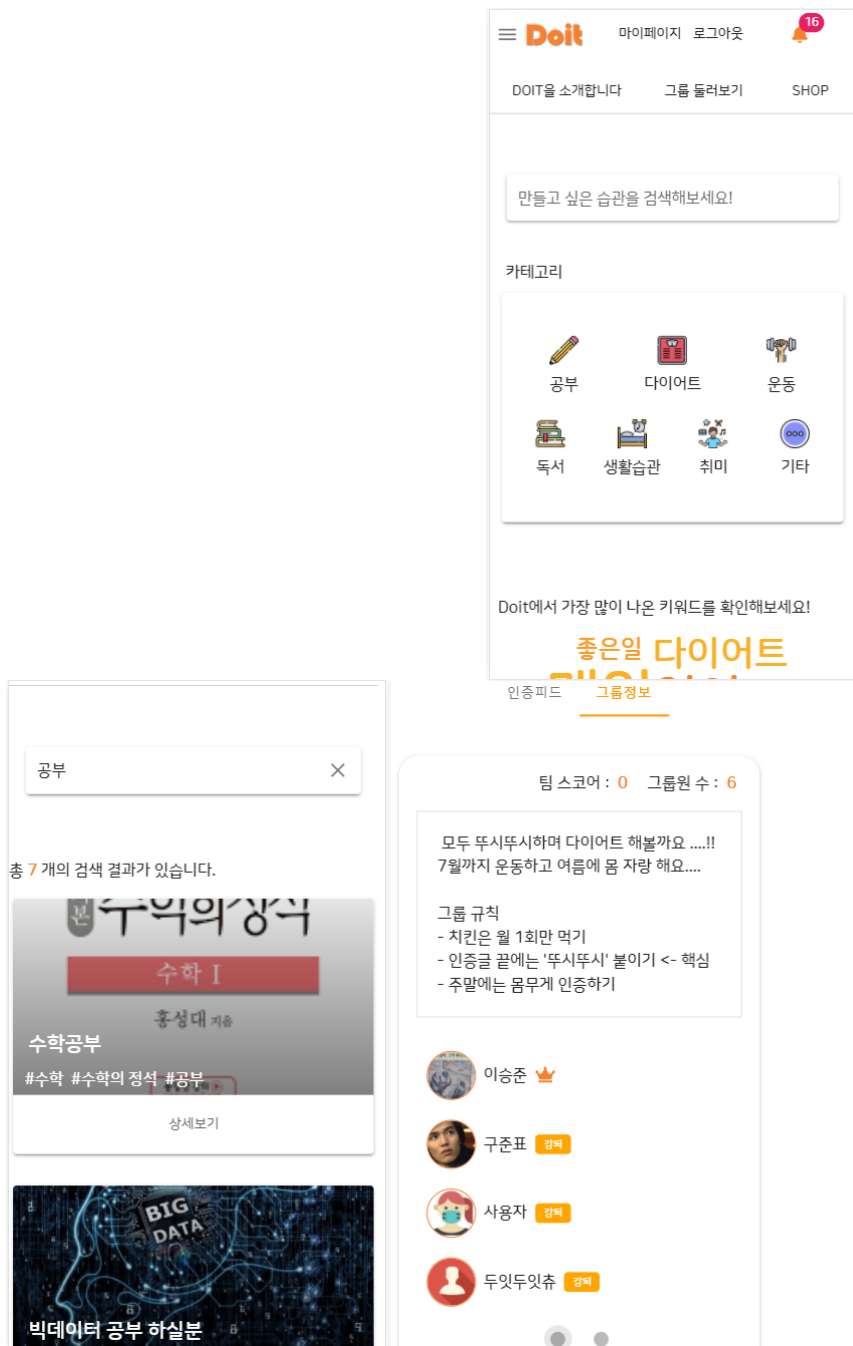
사용자



구준표



두잇두잇



3.기능

0. 핵심 코드

1. Axios 모듈화

서버와 통신할 때 자주 이용할 Axios의 중복(Header, 에러처리 등)을 처리하기 위해 따로 정의하여 각종 API요청시 간편하게 코드를 작성할 수 있게 하였습니다.

```
import axios from 'axios';
import { api, response } from './api/account/index'

const http = axios.create({
  baseURL: 'http://localhost:8080',
```

```

    headers: {
      'Content-type': 'application/json',
    },
  })

http.interceptors.response.use(function (response) {
  return response;
}, function (error) {
  if (error.response.status === response.UNAUTHORIZED) {
    {
      api.onUnauthorized();
    }
  }
  else if (error.response.status === response.FOBBIDEN)
    api.onFobidden();
  else {
    console.log(error)
  }
});

http.interceptors.request.use(function (config) {
  config.withCredentials = true;
  return config;
}, function (error) {
  return Promise.reject(error);
});

export default http;

```

2. Vuex

로그인 유저의 상시로 사용하는 데이터 정보를 Session에 저장하여 쉽게 접근할 수 있도록하였습니다.

```

import { api } from '../api/account/index'

const state = {
  accessToken: null,
  name: null,
  email: null,
  userpk: null,
  usingmg: null,
  role: ""
};

const getters = {
  isAuthenticated(state) {
    return !!state.accessToken;
  },
  getAccessToken(state) {

    .....
  }
};

const mutations = {
  SET_ACCESSTOKEN(state, { accessToken, email, name, userpk, usingmg }) {
    state.accessToken = accessToken;
    state.email = email;
    state.name = name;
  }
}

```

```

    state.userimg = userimg
    state.userpk = userpk;
  },
  SET_NAME(state, { name }) {
    state.name = name;
    sessionStorage.name = name;
  },
  LOGIN(state, { data }) {
    if (!data.status) return;
    state.accessToken = true;
    state.name = data.object.nickname;
    state.email = data.object.email;
    state.userpk = data.object.id;
    state.userimg = data.object.image
    state.role = data.object.role

    sessionStorage.accessToken = state.accessToken;
    sessionStorage.name = data.object.nickname;
    sessionStorage.email = data.object.email;
    sessionStorage.userpk = data.object.id; import { api } from
    '../..api/account/index'

    ....

```

3. 이미지

해당 프로젝트에서는 많은 부분에서 이미지가 이용되었습니다. 많은 이미지가 저장되기 때문에 데이터 베이스에 저장하지 않고 S3에 저장하여 서버에 부담을 줄여주었습니다.

```

<!-- 케이스별 이미지 -->
<div class="text-center d-flex align-center flex-column" >
  <v-img v-if="imageUrl" :src="imageUrl" class="profile-img"></v-img>
  <v-img v-else-if="image !== 'undefined' " :src="`http://ssafydoit.s3.ap-
northeast-2.amazonaws.com/` + image" class="profile-img"> </v-img>
  <v-img v-else src="@/assets/img/profile_temp.png" class="profile-img"> </v-
img>
  <!-- 사진변경 버튼 시작 -->
  <div class="pt-3 text-center">
    <input type="file" ref="imageInput" hidden @change="onImages"
accept="image/*">
    <v-btn class="mt-4" outlined type="button" @click="onClickImageUpload">업로
드</v-btn>
    ....

```

```

onImages(e) {
  this.file = e.target.files[0];
  this.imageUrl = URL.createObjectURL(this.file)
},
onClickImageUpload() {
  this.$refs.imageInput.click();
}

...
const formData = new FormData()
formData.append('file', this.file)
// 이미지 업로드 axios

```

```

http.post('user/updateImg', formData)
.then(()=>{
  if (this.c_Nick) {
    http.put(`/user/updateInfo`, {nickname:this.name})
    .then((res) => {
      if (res.data.status) {
        alert("회원정보가 변경되었습니다");
        const name = this.name;
        this.$store.commit("SET_NAME", { name });
        this.$router.push("/user/profile");
      }
    })
  }
})

```

4. 소켓통신

```

const mutations = {
  SET_CONNECTED(state, flag) {
    state.isConnected = flag;
  },
  CONNECT(state, response) {
    state.stompClient = response;
  },
  DISCONNECT(state) {
    state.stompClient = null;
  },
};

const actions = {
  async CONNECT({ commit }) {
    commit('SET_CONNECTED', false);
    const response = await connect();
    commit('CONNECT', response);
    return response;
  },
  async DISCONNECT({ commit }) {
    const response = await disconnect();
    commit('DISCONNECT');
    return response;
  },
};

```

1. 메인 페이지

만들고 싶은 습관을 검색해보세요!

카테고리



공부



다이어트



운동



독서



생활습관

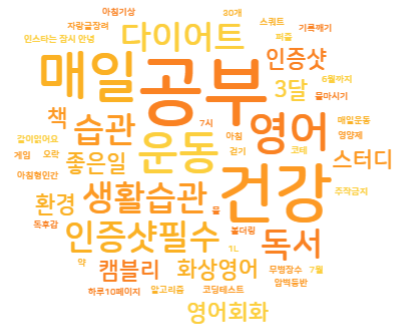


취미



기타

Doit에서 가장 많이 나온 키워드를 확인해보세요!



Doit 그룹랭킹 🏆

아직 일간 그룹 순위가 생성되지 않았어요!
 참여한 그룹에 인증글을 올려
 인증마크를 획득하고 그룹 랭킹을 높여보세요!



1. 워드클라우드 기능

- 그룹에서 지정한 해쉬태그를 기반으로 워드 클라우드를 구성하고 해당 단어를 선택하면 관련 그룹이 검색되어진다.

2. 일간 그룹 랭킹

- 하루마다 팀별 스코어 점수가 책정되고 그 점수로 랭킹이 갱신되어집니다.

3. 카테고리 그룹 검색

- Doit에서 정한 카테고리 7가지 별 그룹을 나누고 카테고리별 그룹을 검색할 수 있습니다.

4. 해쉬태그 검색 기능

- 그룹을 만들때 지정한 해쉬태그로 검색을 합니다.

2. 검색 기능

- 검색창에서 검색한 단어를 통해 그룹들을 보여줍니다.
- created, updated, watch를 통해서 데이터가 변할때 자연스럽게 데이터를 변경시킬 수 있게 제작 하였습니다.
- 로그인 이후에만 그룹에 들어가서 가입 및 활동을 할 수 있게 제작하였습니다.

3. 그룹 관리

- 멤버별 노출 버튼을 다르게 하였습니다.
- 그룹장(리더)만이 그룹을 수정하고 그룹원을 강퇴시킬 수 있습니다.
- 그룹 활동에는 인증/정보공유 글로 나뉘어져 있으며 멤버들은 다른 멤버의 인증 글에 인증 버튼을 누르면서 당사자에게 인증을 해줄 수 있습니다.

4. 유저 관리

1. 프로필 페이지에서 닉네임과 프로필 이미지를 바꿀 수 있습니다.
2. 회원가입, 로그인, 비밀번호 찾기, 프로필 기능이 제공됩니다.
3. 프로필 페이지에서는 자기의 최근 30일간의 활동이력을 재미있게 볼 수 있는 **금잔디**를 제공하며 유저가 쓴 글, 가입한 그룹을 확인할 수 있습니다.



마이페이지 로그아웃



DOIT을 소개합니다

그룹 둘러보기

SHOP

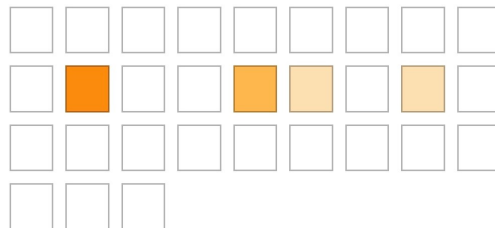


이승준

회원정보 수정 >

금잔디 🏆

잔디를 진하게 채워나가며 성취감을 가져보세요.



30일 동안의 이승준님의 기록입니다.

4. 물품 거래

- 물품을 등록하고 다른 사용자가 해당 물품상세페이지에서 채팅을 시도하면 판매자에게 알림을 통해 해당 이벤트를 알려줍니다.
- 알림을 통해 채팅리스트 페이지로 이동하고 다른 사람들과 해당 물품에 대한 대화를 할 수 있습니다.
- 판매자는 물품판매 예약, 완료, 취소를 할 수 있습니다.\



구리어엄 판매중

화장품

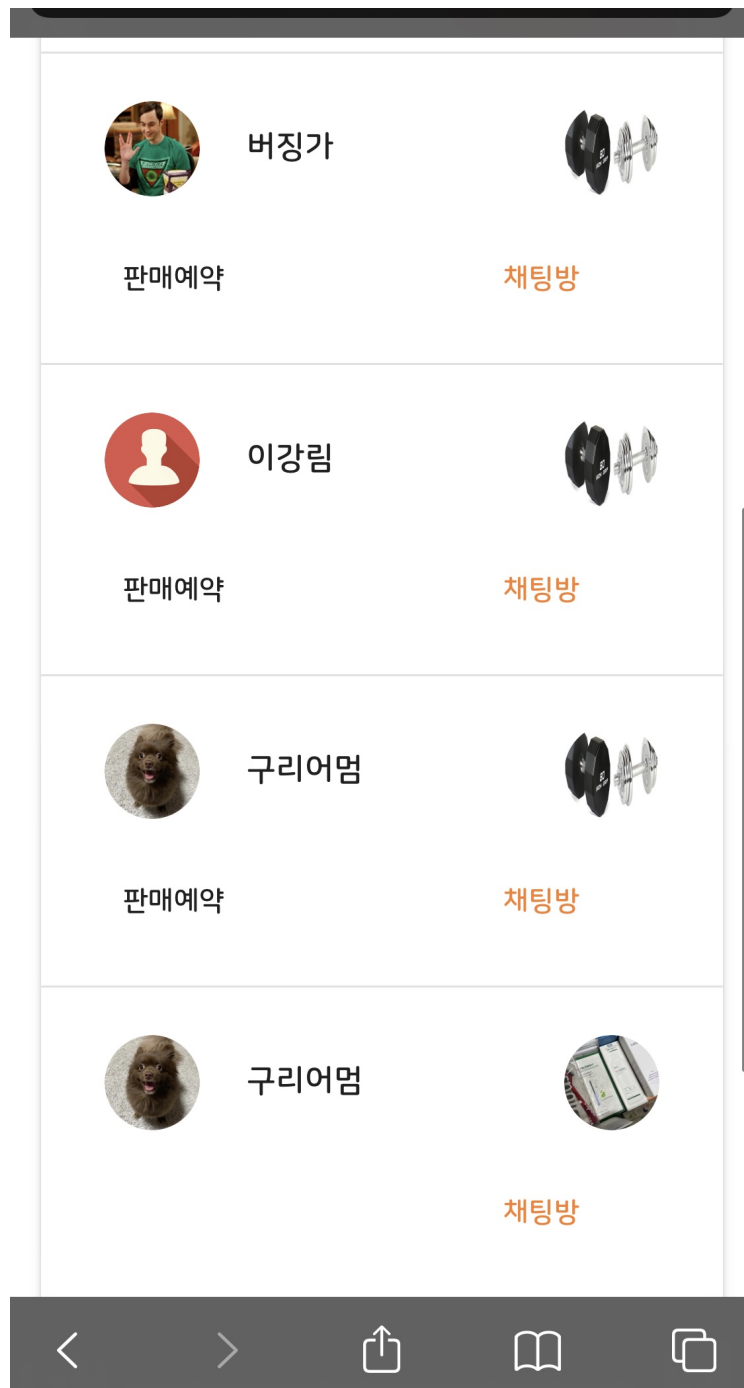
카테고리: 기타

5000 마일리지

안 사용하는 화장품들 많아서 팔아요! 원하는 마일리지 가격있으면 말해주세요:)

채팅으로 거래하기

[물품상세페이지]



[채팅리스트]

3. 특징점

- ""목표 이루기""라는 주제로 함께 만들어가는 목표 관리 커뮤니티
- 해시태그를 이용한 간편 검색 가능 (그룹 내 정보란에 있는 태그 클릭 시에도 검색 가능)
- 인증사진만 올리는 '척' 하는 꿈을 줄일 수 있는 '서로 인증하기' 기능
- 오늘 작성한 피드가 '인증 완료'되면 마이페이지에서 개수에 따라 다른 잔디 확인 가능
- 그룹마다 그날 인증 수를 계산하여 일간 그룹 순위 제공하여 흥미 유발
- 매주 월요일마다 그 전주동안 그룹의 활동 수를 합하여 1~5위까지 그룹 점수를 제공하며, 그룹에 가입되어있는 그룹원들에게도 마일리지 점수 제공
- Doit 내 활동으로 얻은 마일리지로 다른 사용자와 1:1 채팅으로 물품을 거래할 수 있는 기능
- 판매자와 구매자의 판매로직에 '판매예약/구매확정' 등 중간과정 삽입하여 거래 기능 강화"

