Samsung Software Academy
For Youth

Web / Mobile
Project

박사홍, 양동권, 박근형

목차

소개

팀원 소개 및 역할

구성

전체 프로젝트 구성

프론트엔드 구성

백엔드 구성

기능

기능 소개

코드 리뷰

QnA

질문 및 답변

첫번째.

소개

1. 팀원 소개 및 역할

팀장: 박사홍

프로젝트 관리

백엔드

팀원: 양동권

챗봇

프론트엔드

팀원: 박근형

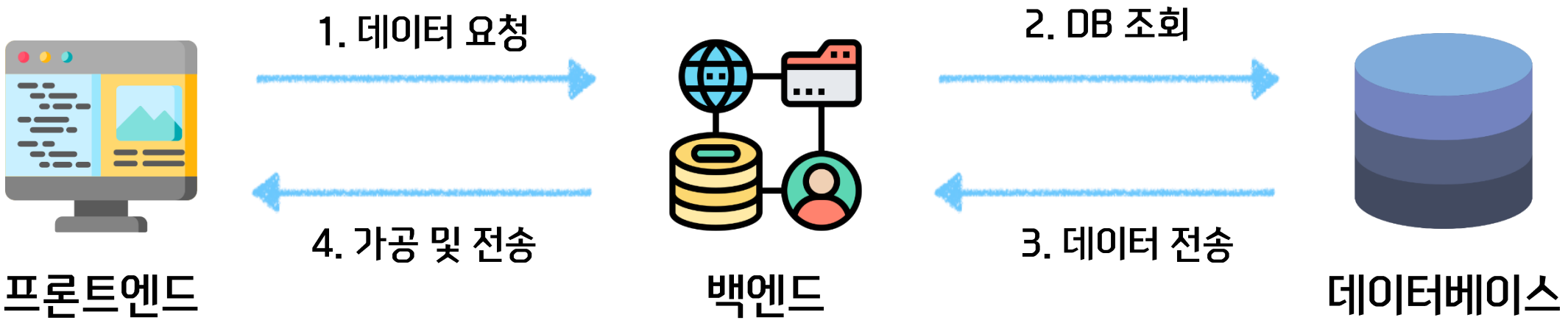
PWA

미들웨어

두번째.

구성

1. 전체 프로젝트 구성
2. 프론트엔드 구성
3. 백엔드 구성



Vue.js

Vuetify

Python

Flask

MySQL

WebMobile

Backend

API 서버

Public

서비스워커

Src

렌더링
데이터 처리

Src

Components

각 기능 수행

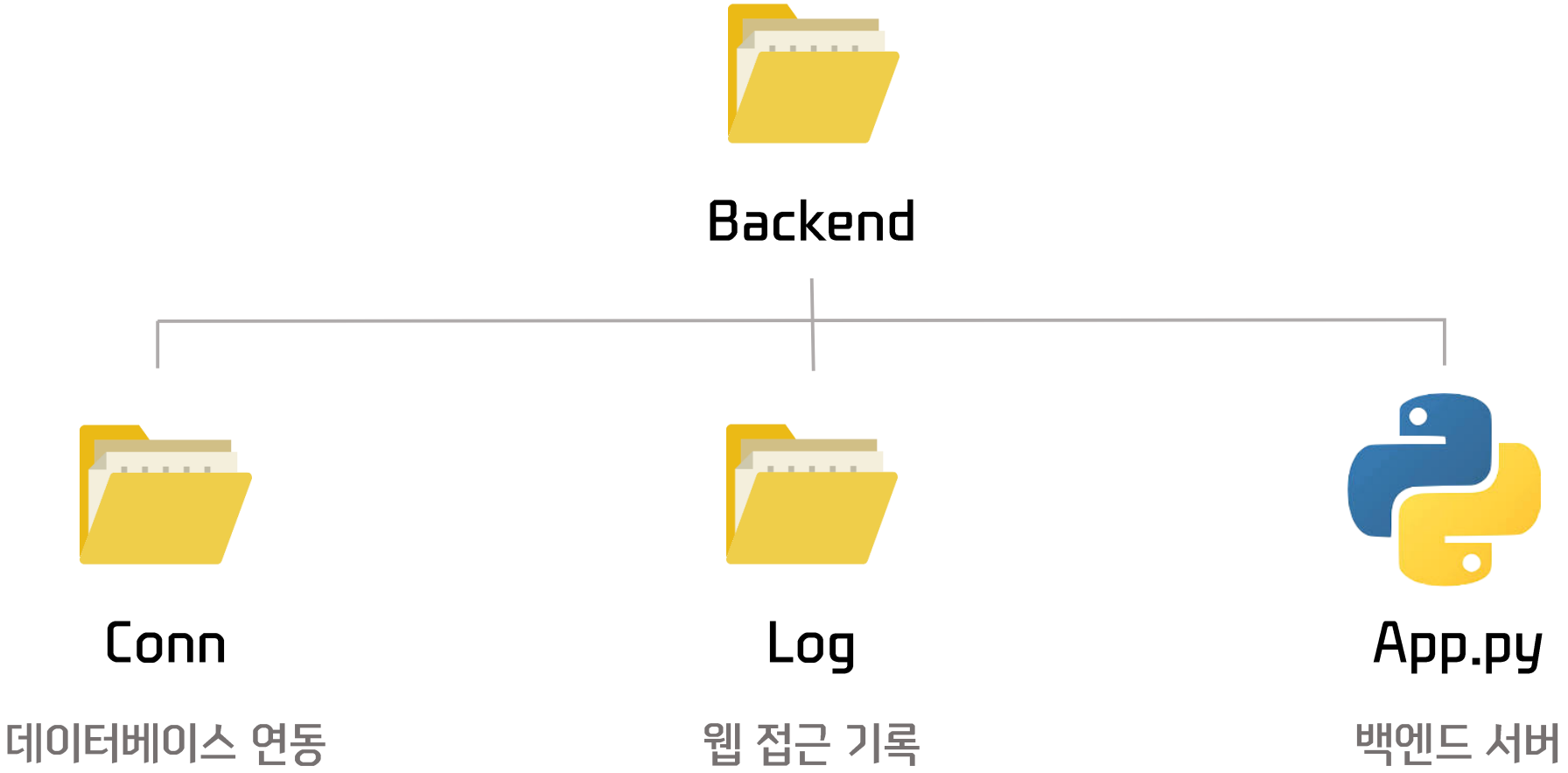
Services

비동기 통신

외부 API 통신

Views

렌더링


Backend
Conn
데이터베이스 연동
Log
웹 접근 기록
App.py
백엔드 서버

세번째.

기능

1. 기능 소개
2. 코드 리뷰

```
<template>

</template>


<script>

</script>


<style scoped>

</style>
```

template

사용자에게 보여지는 코드

script

조건부 렌더링, 데이터 가공 / Vue 라이프 사이클

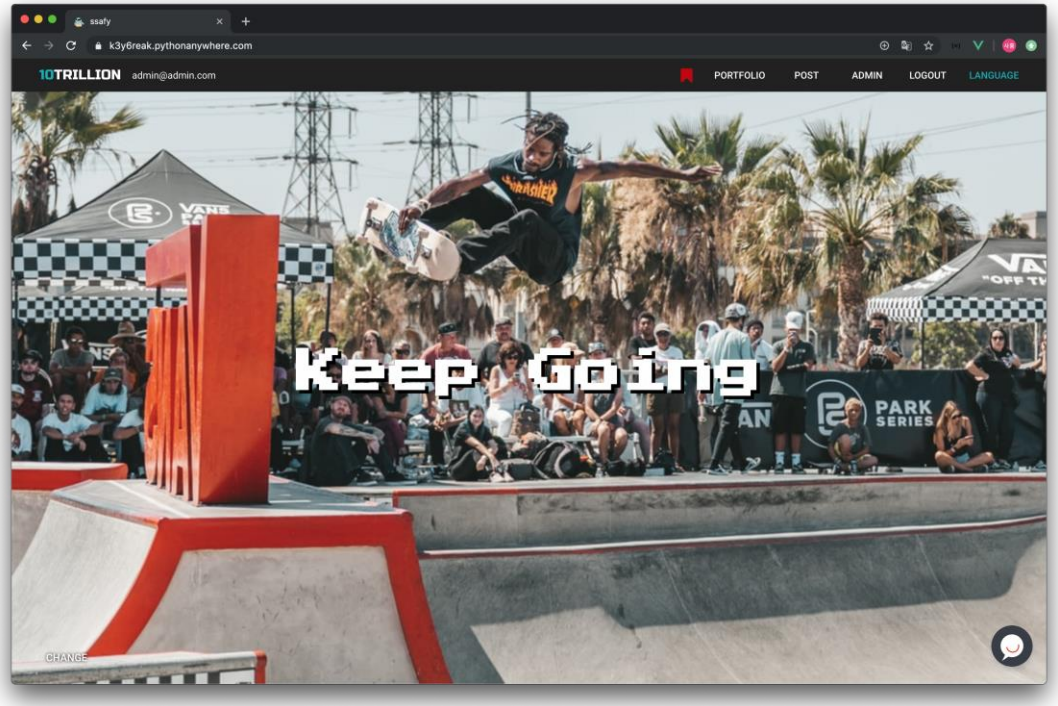
style

화면 디자인

* scoped: 해당 파일의 스타일만 존재하는 경우

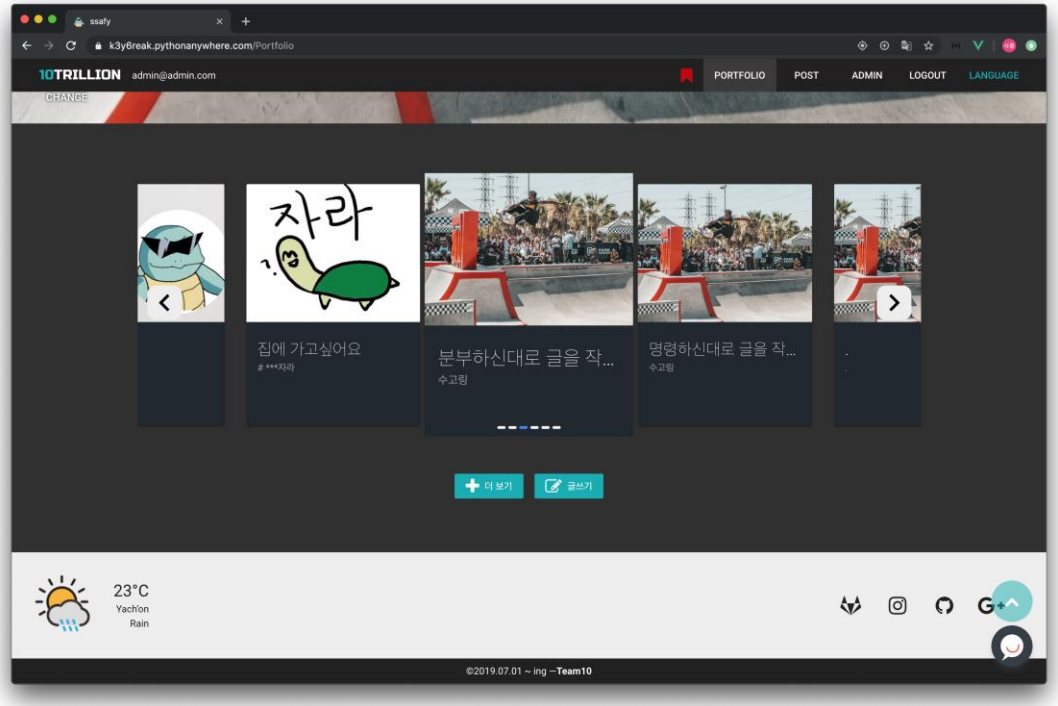
기능 소개 - 레이아웃

메인 페이지



풀 페이지

기타 페이지



일반 페이지

```
1.  <template>
2.    <div>
3.      <v-app dark>
4.        <Header />
5.
6.        <v-content>
7.          <router-view />
8.        </v-content>
9.
10.       <!-- Bottom to Top Button -->
11.       <back-to-top bottom="90px" right="20px">
12.         <button type="button" class="btn btn-info btn-to-top">
13.           <i class="fa fa-chevron-up" />
14.         </button>
15.       </back-to-top>
16.
17.       <Footer v-if="this.$route.path != '/'" />
18.     </v-app>
19.   </div>
20. </template>
```

Header

상단 메뉴

v-content

헤더, 푸터를 제외한 영역

Footer

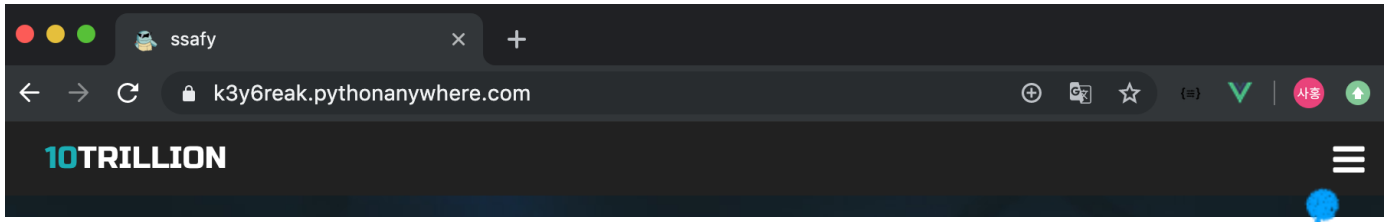
날씨, 팀 정보

메인 페이지에서는 나타나지 않음

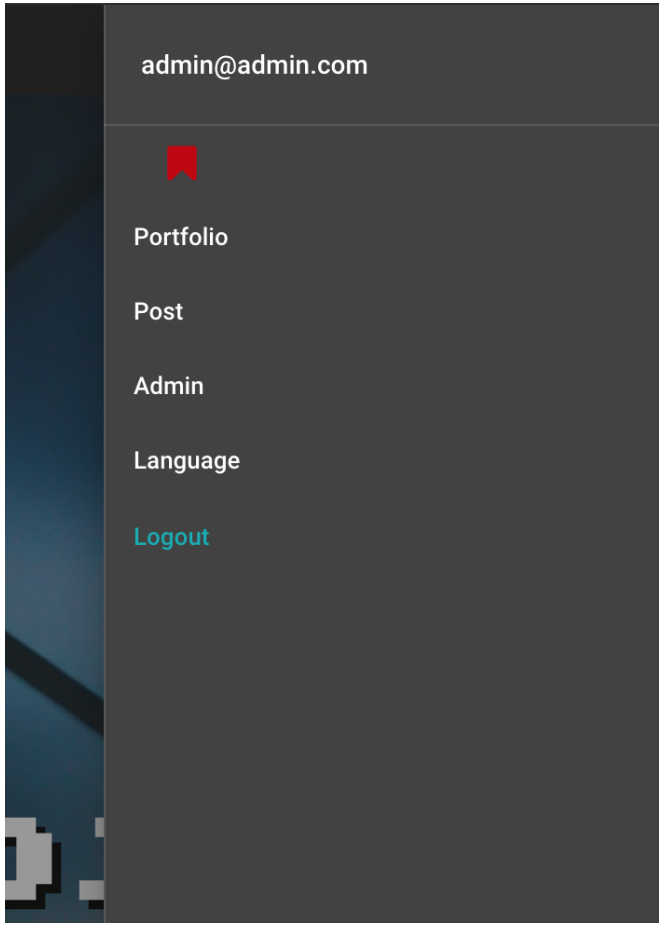
헤더 영역


ssafy
k3y6reak.pythonanywhere.com
10TRILLION
admin@admin.com
PORTFOLIO
POST
ADMIN
LOGOUT
LANGUAGE
팀 로고
계정정보
즐겨찾기
메뉴
관리자 메뉴
번역

헤더 영역(모바일)



모바일 3단 바



코드 리뷰 - 헤더


```
1.  <!-- Appers after login -->
2.  <div class="hidden-sm-and-down" v-if="this.$store.state.accessToken">
3.    {{ this.$store.state.umail }}
4.  </div>
5.
6.  <v-toolbar-items class="hidden-sm-and-down">
7.    <!-- Bookmark Button -->
8.    <v-btn flat href="javascript:bookmarksite('home','/')" icon>
9.      <v-icon color="#c10000">fa-bookmark</v-icon>
10.   </v-btn>
11.
12.   <!-- Menu -->
13.   <v-btn
14.     v-for="menu in menus"
15.     :key="menu.title"
16.     :to="menu.route"
17.     flat
18.     router
19.   >
20.     {{ menu.title }}
21.   </v-btn>
```

계정 정보

Vuex 상태정보 관리

즐겨찾기

메뉴

데이터 영역 메뉴 리스트

코드 리뷰 - 헤더


```
1.  <!-- Admin Menu -->
2.  <v-btn v-if="this.$store.state.uauth == 2" to="/Admin" flat router>
3.    Admin
4.  </v-btn>
5.
6.  <!-- Login Logout Button-->
7.  <v-btn v-if="this.$store.state.accessToken" v-on:click="logout()" flat>
8.    logout
9.  </v-btn>
10. <v-btn v-else @click="dialog_login = true" flat>login</v-btn>
11.
12. <!-- Translate Button -->
13. <v-btn id="highlight-fontColor" @click="dialog = true" flat>
14.   Language
15. </v-btn>
```

관리자 메뉴

로그인 / 로그아웃

Vuex 상태정보 관리

```
export default new Vuex.Store({
  state: {
    SERVER_URL: "http://localhost:5000/",
    IMGUR_URL: "https://api.imgur.com/3/",
    DISQUS_URL: "https://webmobile-team10.disqus.com",
    accessToken: localStorage.accessToken,
    refreshToken: localStorage.refreshToken,
    user: "",
    umail: localStorage.umail,
    uauth: localStorage.uauth,
    url: "https://source.unsplash.com/random",
    random_url: "https://source.unsplash.com/random"
  },
  mutations: {
    setUrl(state, _url) {
      state.url = _url;
    },
    login(state, userInfo) {

    },
    logout(state) {

    }
  },
  actions: {
    login({ commit }, userInfo) {
      commit("login", userInfo);

    },
    logout({ commit }) {
      commit("logout");

    }
  }
});
```

state

외부 데이터 접근

mutation

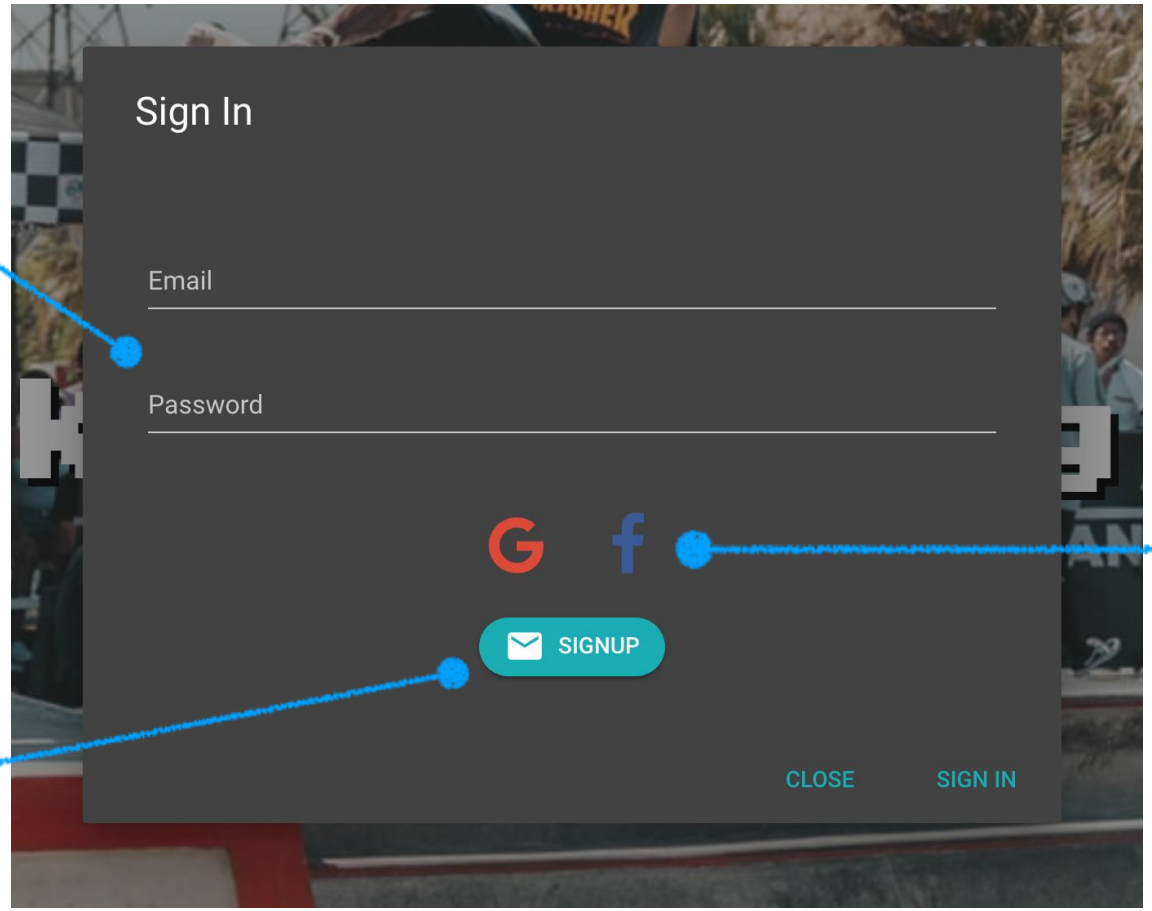
actions 데이터를 이용하여
state 정보 변경

actions

특정 기능 호출 시 데이터 저장

기능 구현에 필요한 코드는 공간상 제외함.

로그인



로그인

회원가입

구글, 페이스북 로그인

코드 리뷰 - 로그인


```
async login() {
  var form = new FormData();
  form.append("umail", this.email);
  form.append("upasswd", this.password);

  await Server(this.$store.state.SERVER_URL)
    .post("/api/login", form)
    .then(res => {
      if (res.data.success) {
        this.$store.dispatch("login", {
          accessToken: res.data.session.accessToken,
          refreshToken: res.data.session.refreshToken,
          umail: res.data.user.umail,
          uauth: res.data.user.uauth
        });
      } else {
        alert(res.data.msg);
      }
    });
  this.postToken();
}
```

아이디, 비밀번호

FormData로 사용자의 정보 전송

토큰, 권한

인증된 사용자의 권한, 토큰 저장

3 코드 리뷰 - 로그인

```
1.  # Get one user using login
2.  @app.route("/api/login", methods=["POST"])
3.  def login():
4.      umail = request.form.get("umail")
5.      upasswd = request.form.get("upasswd") + SALT
6.      upasswd = hashlib.sha256(upasswd.encode()).hexdigest()
7.
8.      cursor = conn.db().cursor()
9.      sql = "select * from users where umail = %s and upasswd = %s"
10.     cursor.execute(sql, (umail, upasswd))
11.     result = cursor.fetchone()
12.
13.     if isinstance(result, type(None)):
14.         return jsonify({"msg": "해당 정보가 없습니다.", "success": False})
15.
16.     # Set session
17.     token_identity = {"umail": umail}
18.     accessToken = create_access_token(identity=token_identity)
19.     refreshToken = create_refresh_token(identity=token_identity)
20.
21.     session = ({"accessToken": accessToken,
22.                 "refreshToken": refreshToken})
23.
24.     add_log("Login", umail)
25.
26.     return jsonify({"msg": "로그인 성공", "success": True, "user": result, "session": session})
```

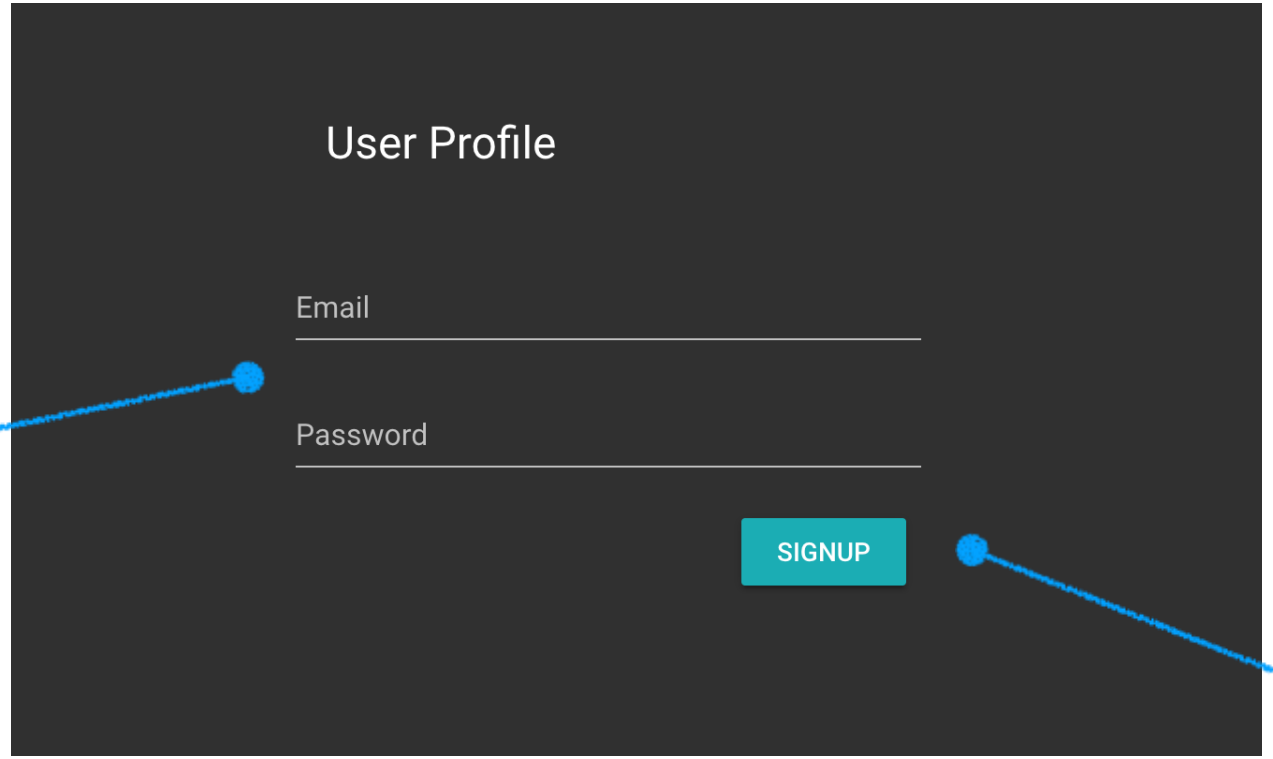
아이디, 비밀번호

비밀번호 + SLAT 값 을 SHA256 해싱

계정 여부 확인

SQL을 이용하여 계정 확인

회원가입



아이디, 비밀번호

가입 버튼

```python
# Insert user
@app.route("/api/add/user", methods=["POST"])
def add_user():
    umail = request.form.get("umail")
    upasswd = request.form.get("upasswd") + SALT
    upasswd = hashlib.sha256(upasswd.encode()).hexdigest()

    db = conn.db()
    cursor = db.cursor()
    sql = "insert into users (umail, upasswd, uauth) values(%s, %s, 0)"

    try:
        cursor.execute(sql, (umail, upasswd))
    except pymysql.err.IntegrityError as e:
        return jsonify({"msg": "중복된 계정입니다."})

    db.commit()

    add_log("Add user", umail)

    return jsonify({"msg" : "가입완료!"})
```

비밀번호

비밀번호 + SLAT 값 을 SHA256 해싱

SQL 에러

중복된 계정일 경우 메시지 전달

푸터 영역

날씨

팀 정보

SNS 버튼

코드 리뷰 - 푸터


```
1.  <!-- weather -->
2.  <div id="weather">
3.    <v-card light flat class="grey lighten-3">
4.      <v-layout>
5.        <v-flex xs6class="align-center">
6.          <img id="weather_icon_id" :src="weather_icon" />
7.        </v-flex>
8.        <v-flex xs6>
9.          <v-card-title primary-title text-xs-right>
10.           <div class="text-xs-right">
11.             <div class="headline">{{ temp }}°C</div>
12.             <div>{{ city }}</div>
13.             <div>{{ now_weather }}</div>
14.           </div>
15.         </v-card-title>
16.       </v-flex>
17.     </v-layout>
18.   </v-card>
19. </div>
```

날씨

외부 API를 이용해 데이터 출력

기온, 도시 명, 현재 날씨

코드 리뷰 - 푸터


```
1.  methods: {
2.  async getLocation() {
3.    if (navigator.geolocation) {
4.      //위치 정보를 얻기
5.      navigator.geolocation.getCurrentPosition(this.showPosition);
6.    } else {
7.      alert("이 브라우저에서는 Geolocation이 지원되지 않습니다.");
8.    }
9.  },
10. showPosition(position) {
11.   this.latitude = position.coords.latitude;
12.   this.longitude = position.coords.longitude;
13.   var apiKey = "843e94f1b4901a3e5dbad93a0befb2e5";
14.   const apiURL = "https://api.openweathermap.org/data/2.5/weather";
15.   this.$http({
16.     method: "GET",
17.     url: apiURL,
18.     params: {
19.       appid: apiKey,
20.       lon: this.longitude,
21.       lat: this.latitude
22.     }
23.   }).then(res => {
24.     this.now_weather = res.data.weather[0].main;
25.     this.city = res.data.name;
26.     this.temp = res.data.main.temp - 273.15;
27.     this.weather_icon = require("../assets/img/weather/" +
28.       res.data.weather[0].icon +
29.       ".png");
30.   });
31. }
```

사용자의 위치

브라우저의 위치정보 기능

Open Weather Map API

API를 이용하여 현재 위치의 날씨 정보 가공

해당 날씨에 대한 이미지 출력

기능 소개 – 사진 첨부

사진 첨부 버튼

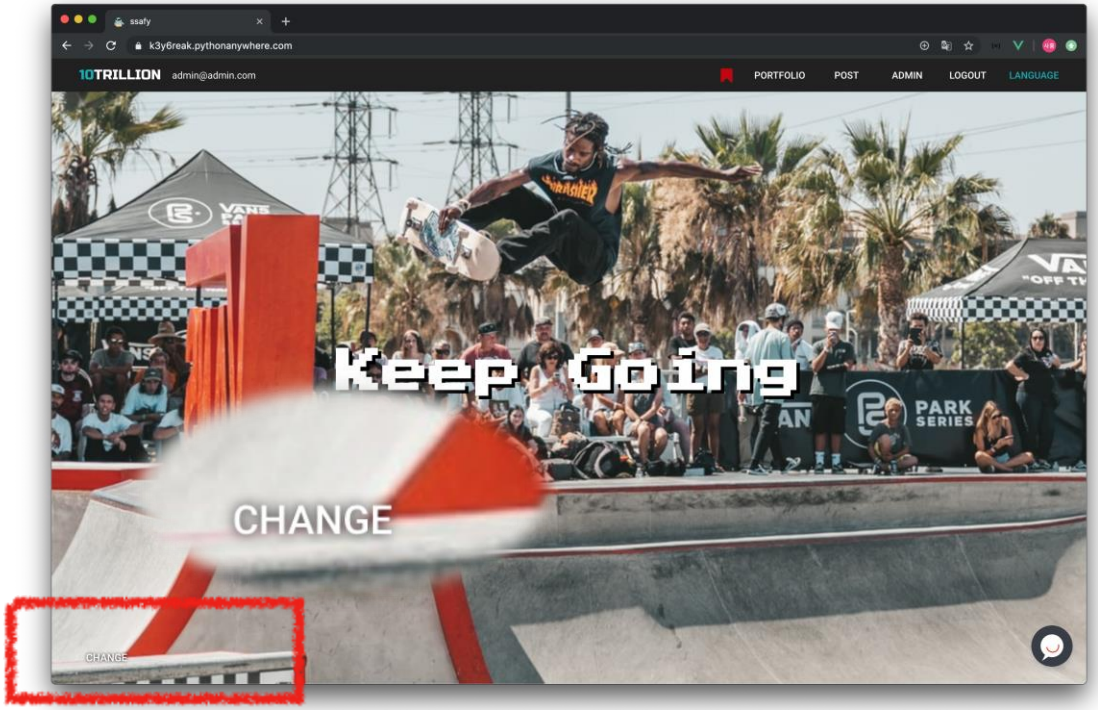
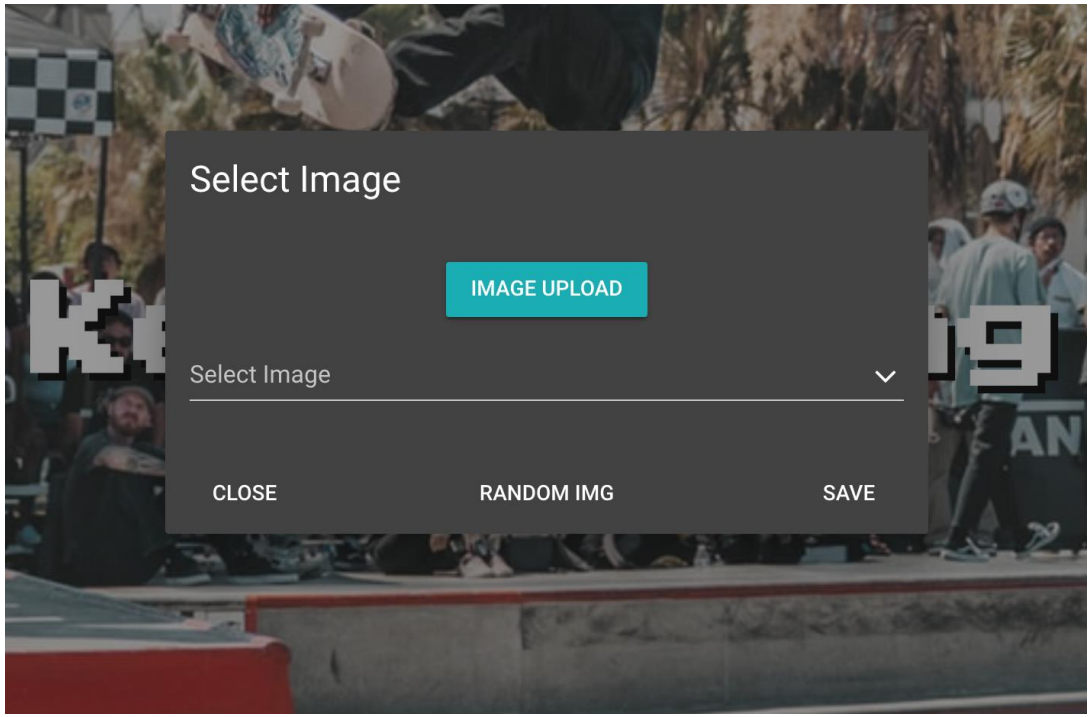


사진 첨부 창



기능 소개 – 사진 첨부

사진 선택

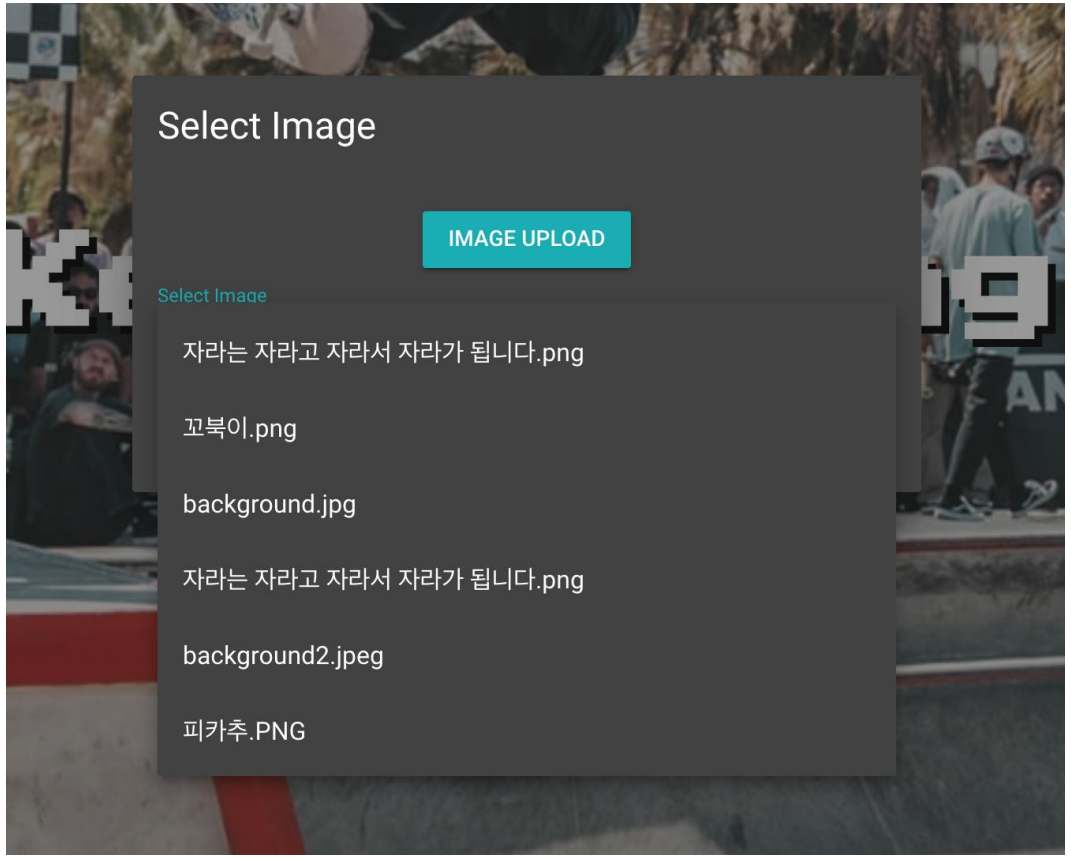
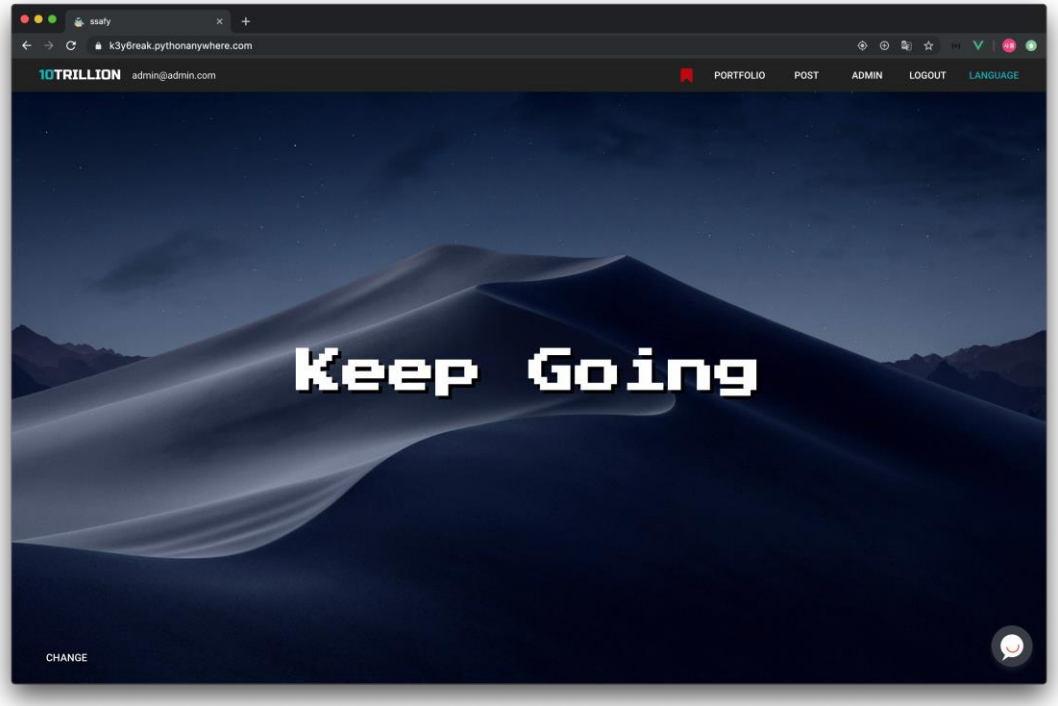


사진 변경



코드 리뷰 – 사진 첨부


```
1.  methods: {
2.   RandomImg() {
3.     this.$store.commit("setUrl", this.random_base_url);
4.     this.imgSrc = this.$store.state.url;
5.     this.dialog = false;
6.   },
7.   ChangeImg() {
8.     if (this.selectedImg == "") {
9.       alert("이미지를 선택해주세요");
10.      return;
11.    }
12.    this.$store.commit("setUrl", this.selectedImg);
13.    this.imgSrc = this.selectedImg;
14.    this.dialog = false;
15.  },
16.  getAlbumImg()
17.    var images = ImgurApi(this.$store.state.IMGUR_URL).get(`album/pQivCF9`);
18.    images.then(data => {
19.      var response = data.data;
20.      this.albumImgs = response.data.images;
21.    });
22.  },
23.  Upload() {
24.    this.$EventBus.$emit("uploadImgBanner")
25.    this.uploadBannerDialog = false;
26.  }
27. }
```

랜덤 이미지

Unsplash 사이트의 랜덤 이미지

이미지 변경

선택한 이미지로 배너 변경

이미지 선택

사용자가 업로드한 Imgur 이미지

Imgur API를 구성

이미지 업로드

이벤트 버스를 이용하여 업로드

```
1.  mounted() {
2.   var self = this;
3.
4.   // Image Banner upload
5.   this.$EventBus.$on("uploadImgBanner", function() {
6.     self.sendBanner();
7.   });
8.  },
9.
10. methods: {
11.   async sendBanner() {
12.     const bannerID = "N9DRFuvC9ppf6r4";
13.     var form = this.makeFormData(bannerID);
14.     await ImgurApi(this.$store.state.IMGUR_URL).post(`image`, form);
15.   },
16. }
```

이벤트 버스

$emit으로 보낸 데이터를 $on으로 받음

이미지 업로드

해당 이미지를 Imgur 서버에 전송

코드 리뷰 – 사진 첨부


```
1.  import axios from "axios";
2.
3.  const clientID = "Client-ID 2547e24f9344b8c";
4.
5.  export default baseURL => {
6.    return axios.create({
7.      baseURL: baseURL,
8.      timeout: 0,
9.      headers: {
10.       Authorization: clientID
11.     }
12.   });
13. };
```

깃랩 그래프


Project
k3y6reak/ai-sub2
k3y6reak
Recently Commit
박사홍: Merge branch 'feature/3-1_sh' into 'develop' S1P3142002-60 ...
양동권: [+] Edit data source path
박근형: Merge branch 'feature/1-4' into 'develop' S1P3142002-49 [+] ...

```
<v-sparkline
  id="commit_graph"
  :value="commitsCnt"
  :gradient="gradient"
  :smooth="10 || false"
  :padding="8"
  :line-width="3"
  :stroke-linecap="`round`"
  :gradient-direction="gradientDirection"
  :auto-draw="autoDraw"
  xs0
  md12
></v-sparkline>

<!-- Recently Commit -->
<p>Recently Commit</p>
<div v-for="idx in users.length" :key="idx">
  <p id="commit_msg_large">
    {{ users[idx - 1] }}: {{ userCommits[idx - 1] }}
  </p>
  <p id="commit_msg_small">
    {{ users[idx - 1] }}<br />
    {{ userCommits[idx - 1] }}
  </p>
</div>
```

v-sparkline

최근 일주일간 커밋 개수를 나타낸 그래프

개인 최근 커밋

개인별 최근 커밋 메시지

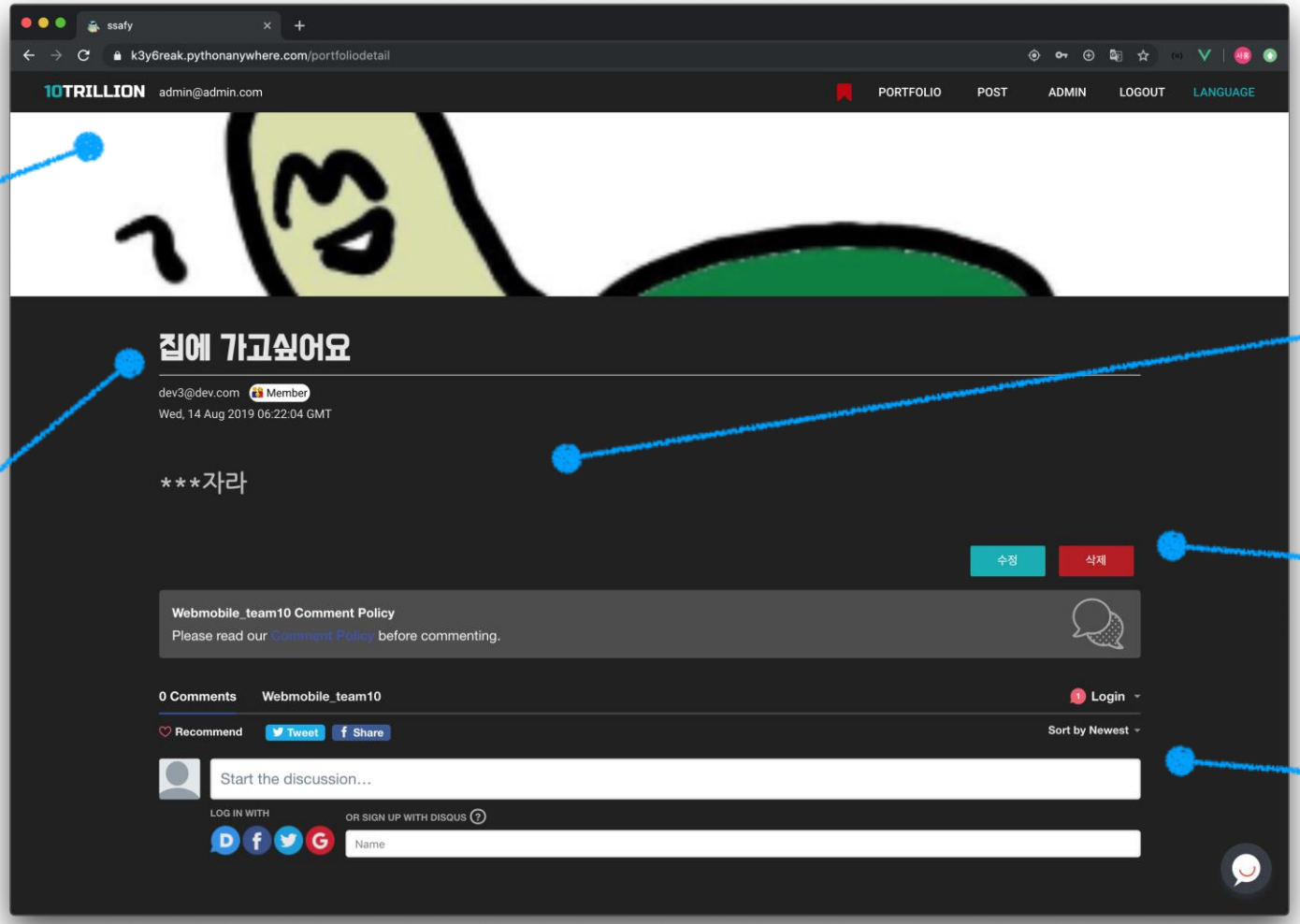
```
1.  methods: {
2.   async drawStatGraph() {
3.     this.autoDraw = false;
4.
5.     for (var day = -7; day <= 2; day++) {
6.       var now = new Date();
7.       var since = new Date();
8.       var until = new Date();
9.
10.      since.setDate(now.getDate() + day);
11.      until.setDate(now.getDate() + day);
12.      since.setHours(-15, 0, 0, 0);
13.      until.setHours(+8, 59, 59, 999);
14.
15.      var commits_7day = await GitlabService.getCommits(
16.        this.repos.id,
17.        since,
18.        until
19.      );
20.
21.      this.commitsCnt.push(commits_7day.data.length);
22.    }
23.    this.autoDraw = true;
24.  },
25.  async usersCommits() {
26.    this.userCommits = await GitlabService.getUsersCommits(this.repos.id);
27.  }
28. }
```

7일 계산

현재부터 7일 전 날짜 및 시간 계산

00시 00분 00초 ~ 23시 59분 59초

포스트 / 포트폴리오



이미지

제목

내용
(마크다운)

수정, 삭제

댓글

코드 리뷰 – 상세 페이지


```
<!-- Portfolio body readonly -->
<div class="postcontext my-5">
  <vue-markdown>{{ portfolio.body }}</vue-markdown>
</div>

<!-- Edit and Delte button -->
<template v-if="authCheck">
  <div class="editBtn">
    <v-btn color="#00adb5" @click="updatePortfolio" depressed>
      수정
    </v-btn>
    <v-btn color="error" @click="deletePortfolio" depressed>
      삭제
    </v-btn>
  </div>
</template>

<!-- Portfolio comments -->
<div class="comments">
  <VueDisqus
    shortname="webmobile-team10"
    :url="this.$store.state.DISQUS_URL + '/portfolio' + portfolio.num"
    v-on:new-comment="newComment"
    :identifier="'portfolio' + portfolio.num"
  ></VueDisqus>
</div>
```

vue-markdown

본문 내용을 마크다운 형식으로 표시

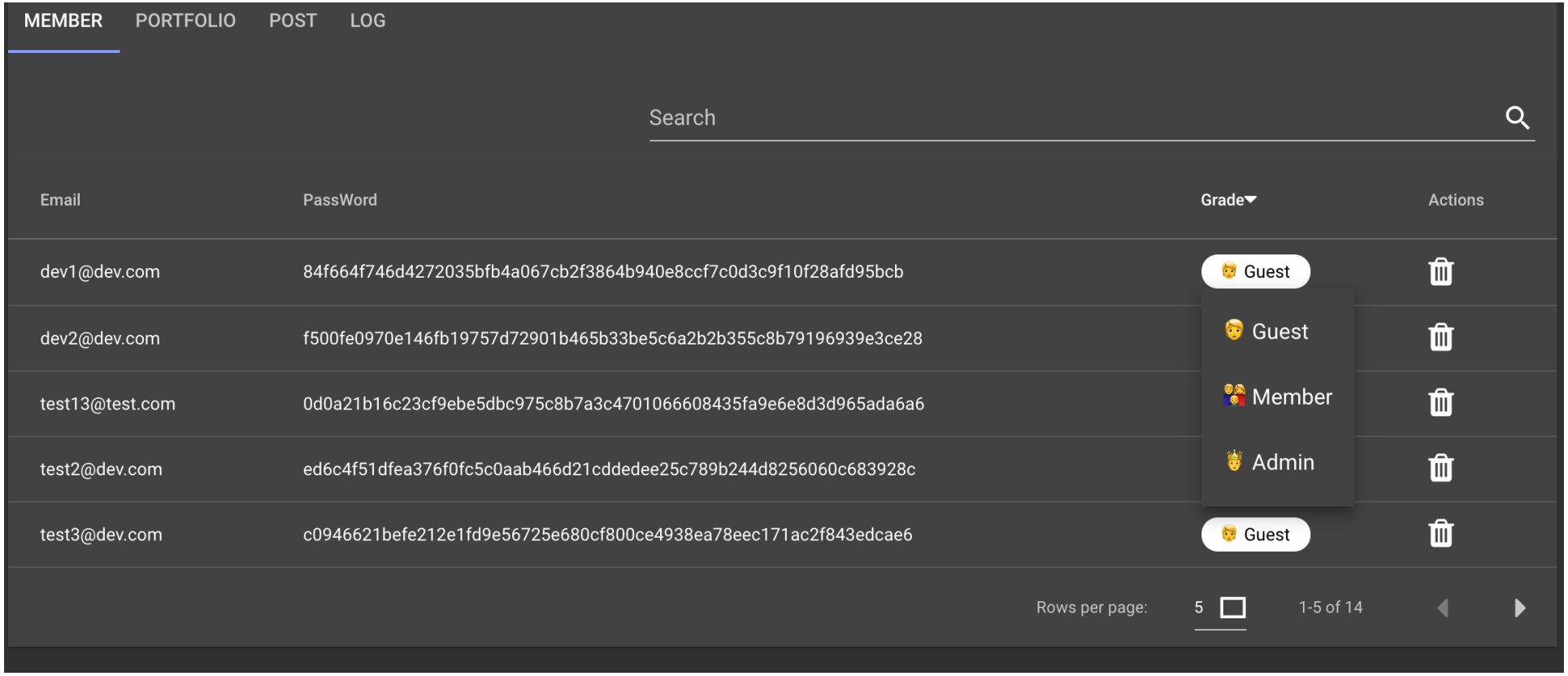
수정, 삭제

작성자, 관리자 외 불가능

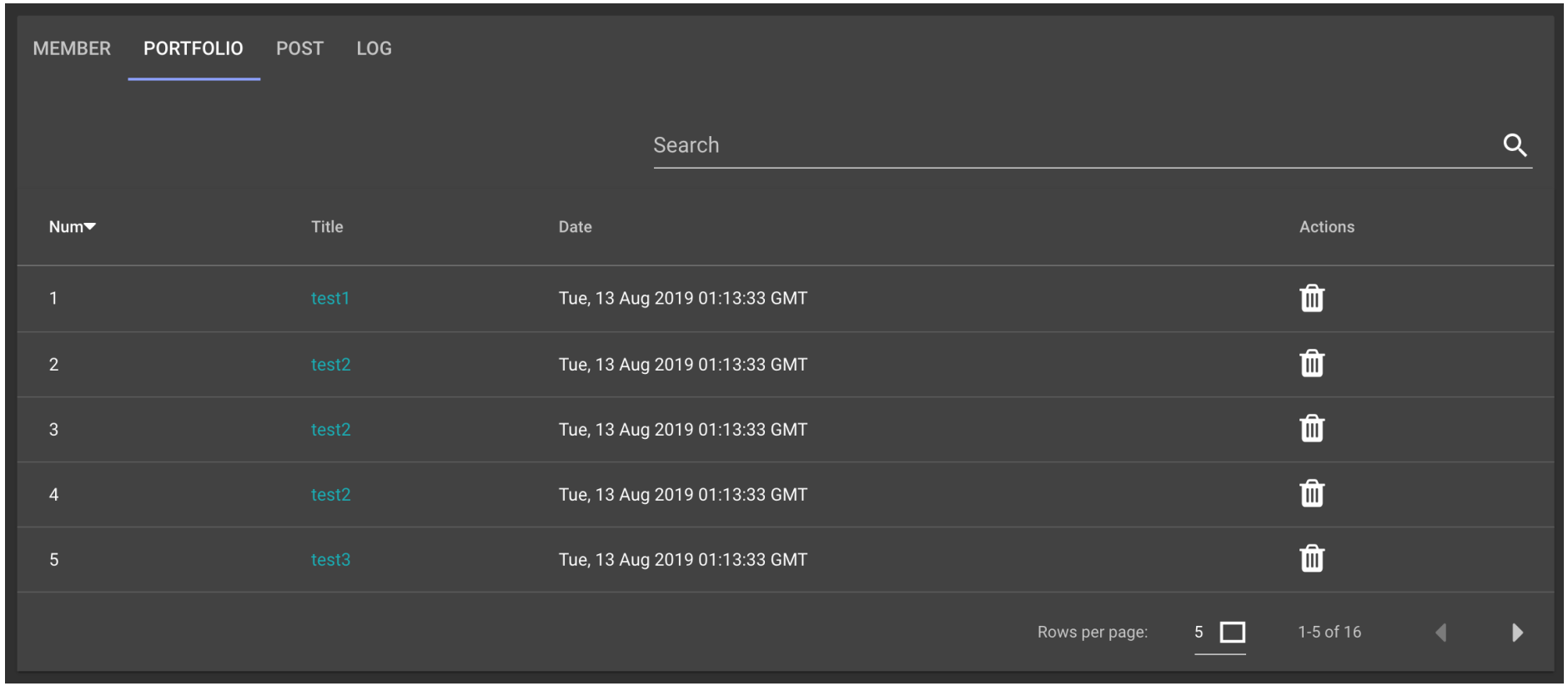
VueDisqus

게시글 별 댓글 관리

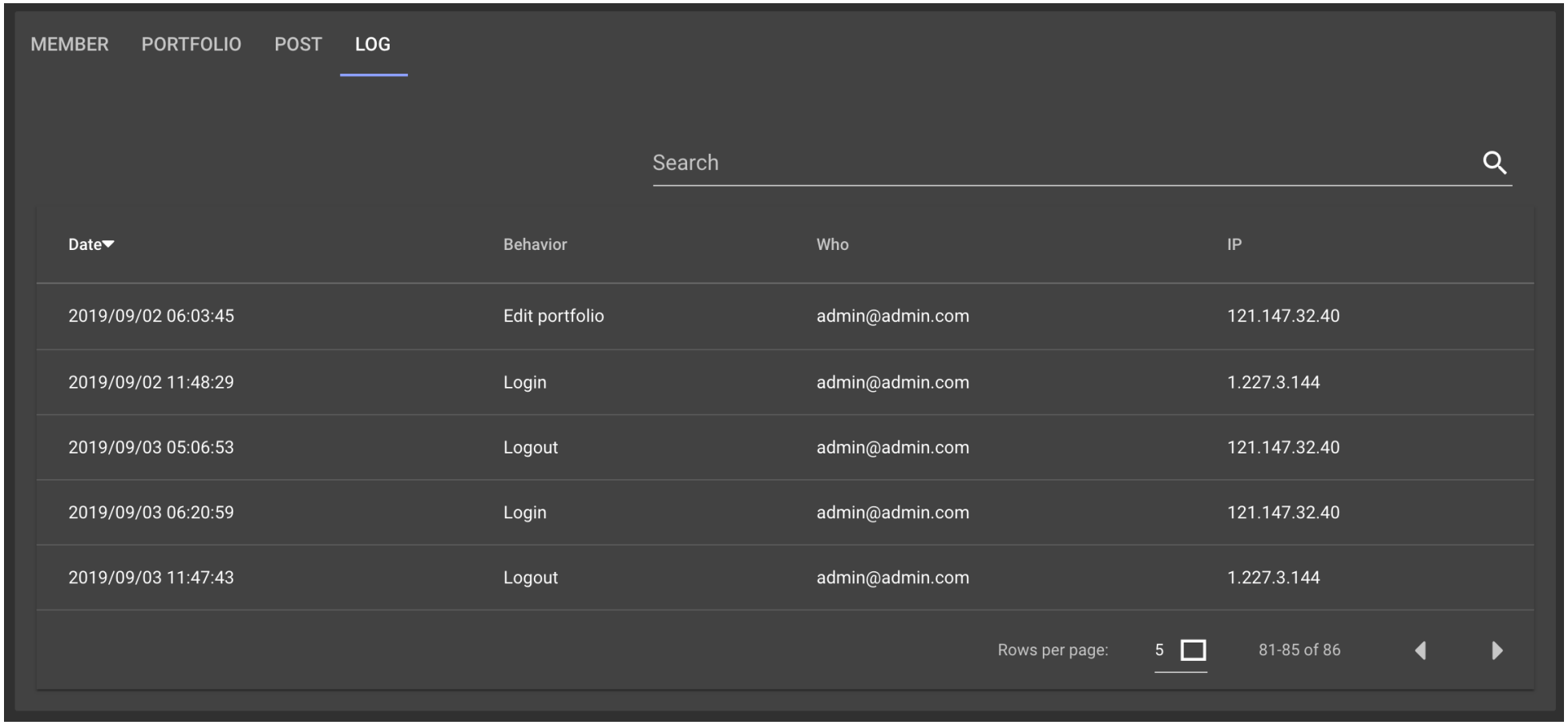
멤버 관리


MEMBER
PORTFOLIO
POST
LOG
Search
Email
PassWord
Grade
Actions
dev1@dev.com
84f664f746d4272035bfb4a067cb2f3864b940e8ccf7c0d3c9f10f28afd95bcb
Guest
dev2@dev.com
f500fe0970e146fb19757d72901b465b33be5c6a2b2b355c8b79196939e3ce28
Guest
Member
Admin
test13@test.com
0d0a21b16c23cf9ebe5dbc975c8b7a3c4701066608435fa9e6e8d3d965ada6a6
test2@dev.com
ed6c4f51dfea376f0fc5c0aab466d21cddedee25c789b244d8256060c683928c
test3@dev.com
c0946621befe212e1fd9e56725e680cf800ce4938ea78eec171ac2f843edcae6
Guest
Rows per page:
5
1-5 of 14

포트폴리오 / 포스트 관리


MEMBER
PORTFOLIO
POST
LOG
Search
Num
Title
Date
Actions
1
test1
Tue, 13 Aug 2019 01:13:33 GMT
2
test2
Tue, 13 Aug 2019 01:13:33 GMT
3
test2
Tue, 13 Aug 2019 01:13:33 GMT
4
test2
Tue, 13 Aug 2019 01:13:33 GMT
5
test3
Tue, 13 Aug 2019 01:13:33 GMT
Rows per page:
5
1-5 of 16

접근 기록 관리


MEMBER
PORTFOLIO
POST
LOG
Search
Date
Behavior
Who
IP
2019/09/02 06:03:45
Edit portfolio
admin@admin.com
121.147.32.40
2019/09/02 11:48:29
Login
admin@admin.com
1.227.3.144
2019/09/03 05:06:53
Logout
admin@admin.com
121.147.32.40
2019/09/03 06:20:59
Login
admin@admin.com
121.147.32.40
2019/09/03 11:47:43
Logout
admin@admin.com
1.227.3.144
Rows per page:
5
81-85 of 86

코드 리뷰 – 관리자 페이지


```
1.  # Get client user ip
2.  def get_ip_addr():
3.      return request.environ.get("HTTP_X_REAL_IP", request.remote_addr)
4.
5.
6.  # Get current time
7.  def get_current_time():
8.      now = time.localtime()
9.      return "%04d/%02d/%02d %02d:%02d:%02d" % (now.tm_year, now.tm_mon, now.tm_mday, now.tm_hour, now.tm_min, now.tm_sec)
10.
11.
12. # Add log
13. def add_log(action, who):
14.     lock.acquire()
15.
16.     f = open(LOG_PATH, "a")
17.     f.write(get_current_time() + "$" + action + "$" + who + "$" + get_ip_addr() + "\n")
18.     f.close()
19.
20.     lock.release()
21.     return ""
```

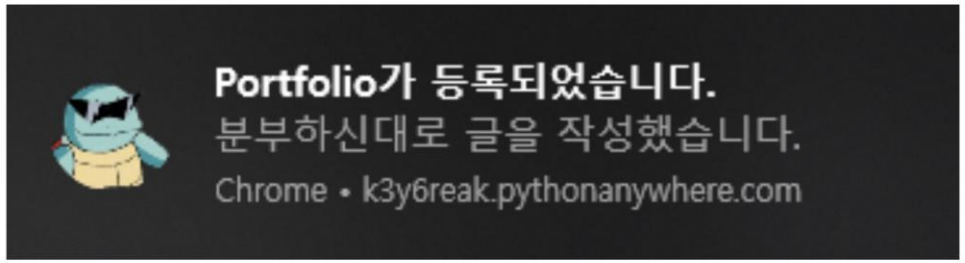
사용자 IP

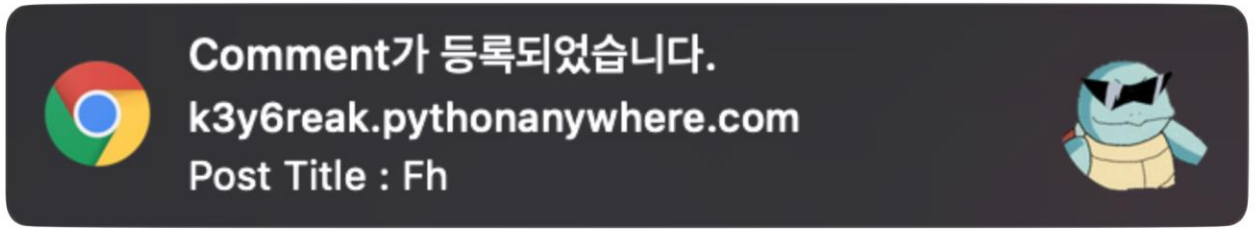
특정 행위 시간

로그 기록

파일에 기록, 데드락, 레이스 컨디션 방지

알람

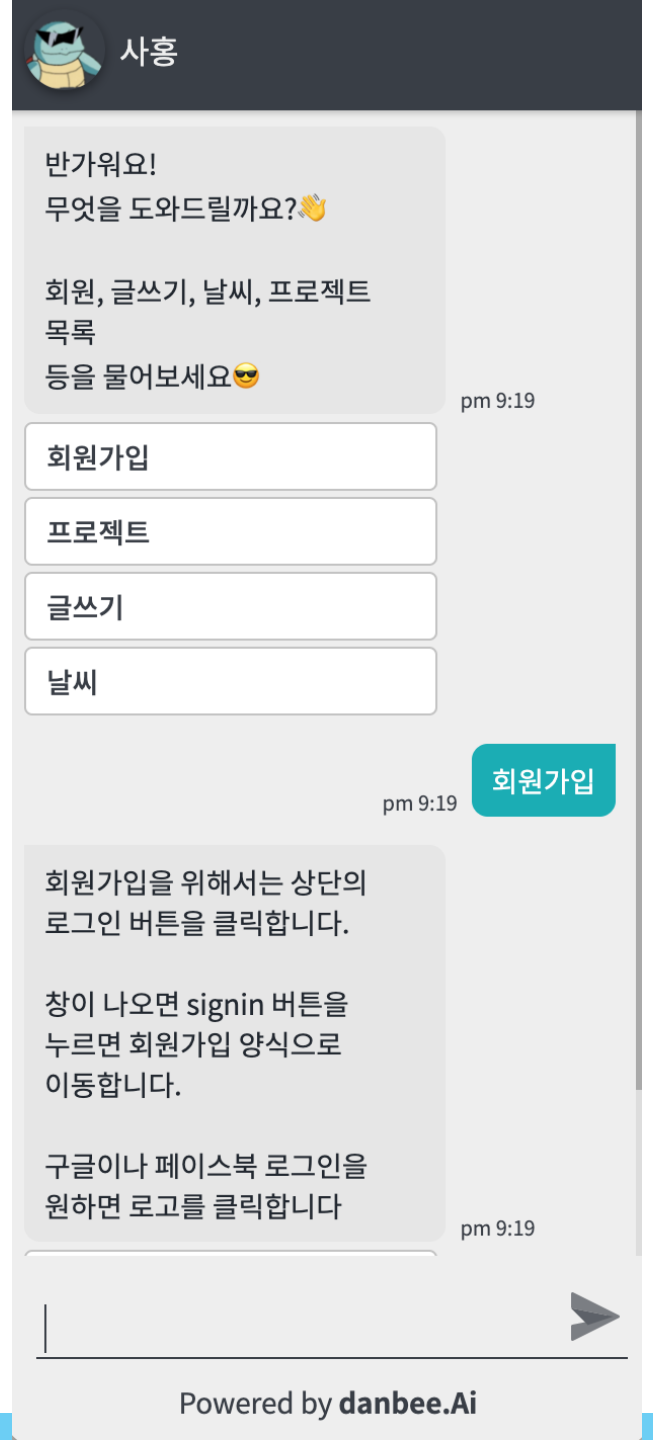


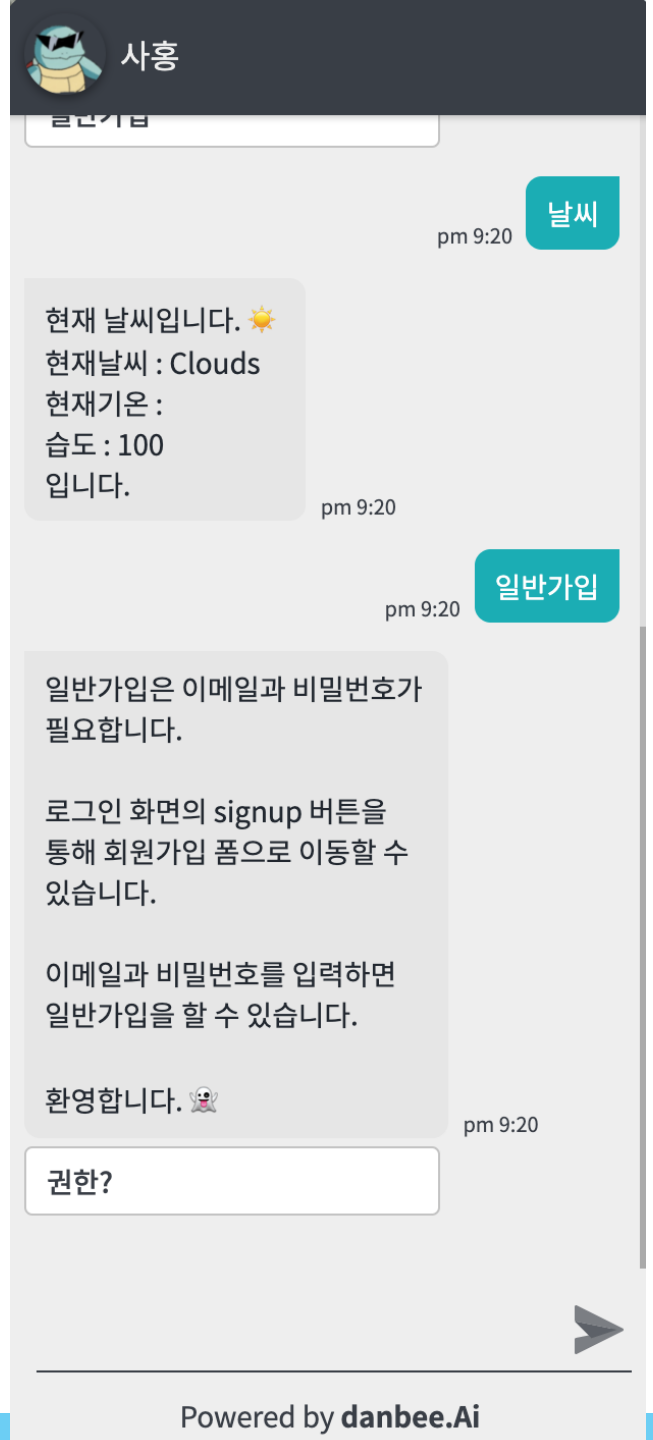


소스코드는 공간상 생략하였습니다.

2 기능 소개 – 챗봇

챗봇





소스코드는 공간상 생략하였습니다.

QnA