



# 스파르타 내일배움캠프 팀(6 CAN DO IT!) 소개 미니 프로젝트 결과물

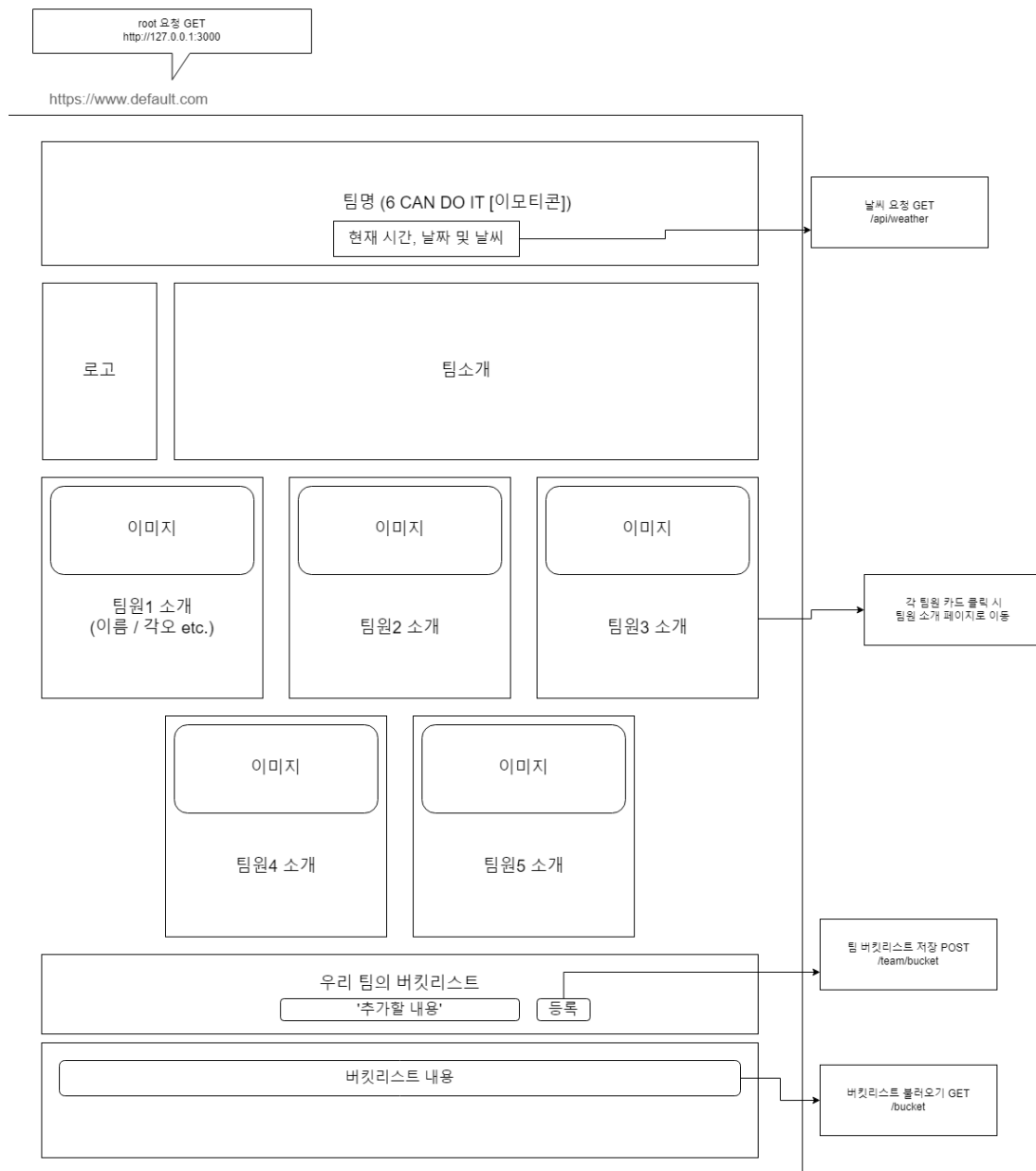
## 1. 프로젝트 제목 및 설명

- 프로젝트 제목: 6캔두잇(6(=YOU) CAN DO IT)
- 설명: 개발의 'ㄱ'자도 모르던 사람들이 '검색만 하면 코딩할 수 있어'라는 마음가짐으로, 자기주도적 학습을 통해 훌륭한 개발자로 성장하고 싶다는 포부를 소개하는 페이지
- 총 개발 인원: 5명 (우태현(팀장), 황민주, 한동주, 김형섭, 이동호)
- 개발 기간: 2022.11.14~2022.11.18 (총 5일)

## 2. 와이어 프레임

- 총 1페이지로 구성
- 팀원 소개는 각 팀원 카드를 눌렀을 때 모달창으로 구현

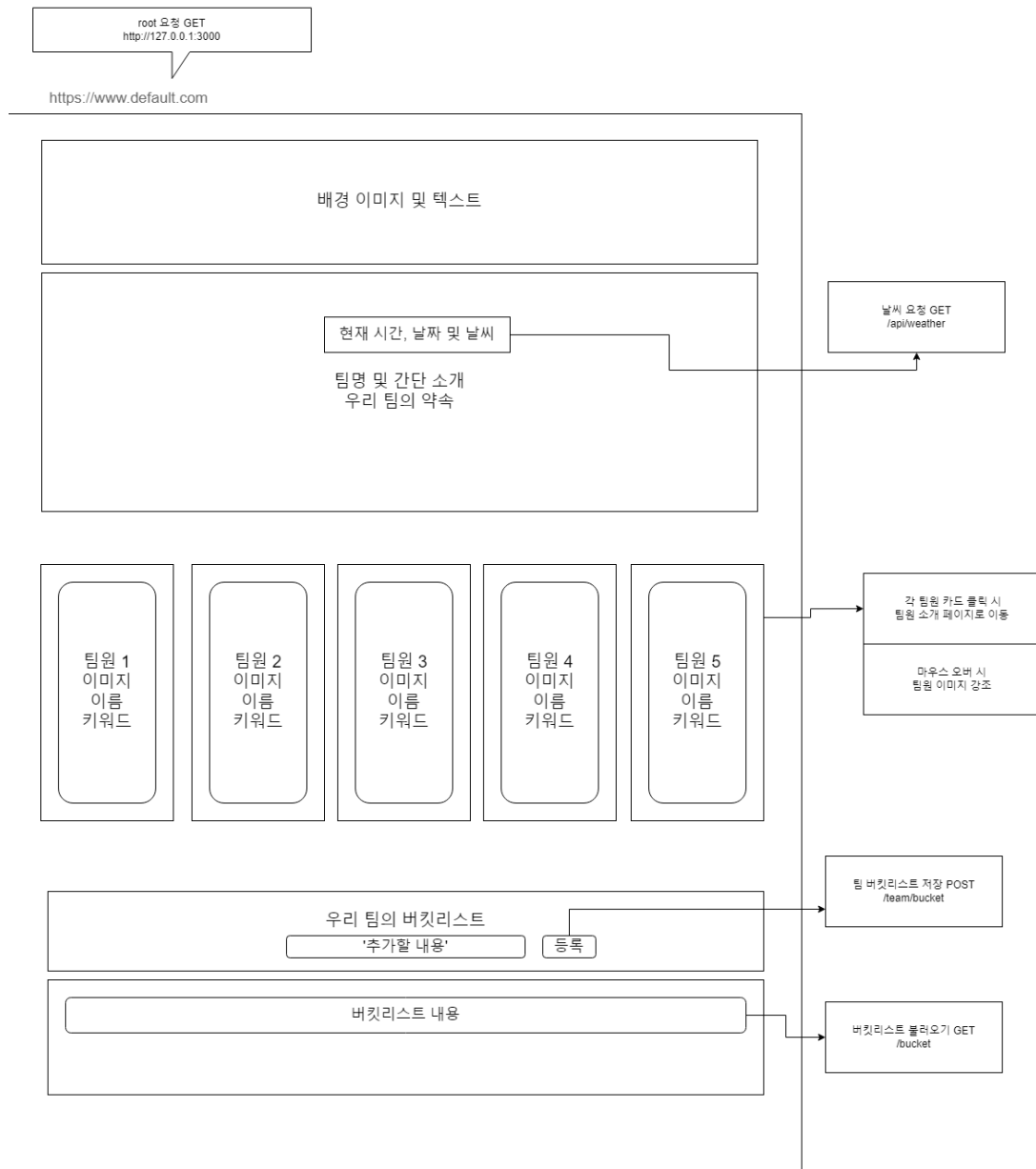
### [1. 팀 소개 페이지 (1차 초안)]



#### (구성 내용)

- 팀명
- 우리 팀만의 특징과 추구하는 궁극적인 목표
- 우리 팀의 약속
- 팀원 소개
- 우리 팀의 버킷리스트

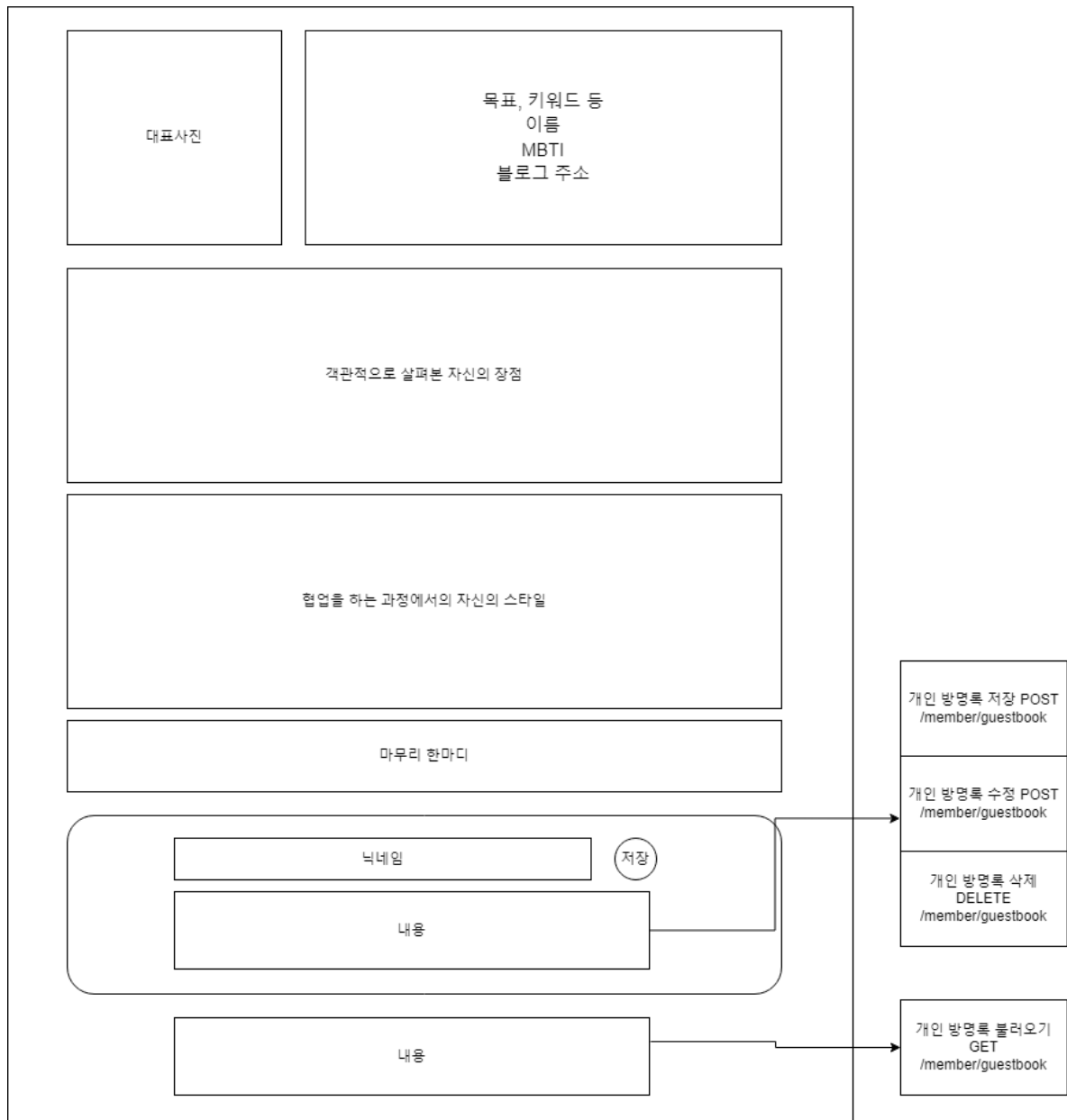
#### [1. 팀 소개 페이지 (최종)]



#### (변경 사항)

- 팀명
- 우리 팀만의 특징과 추구하는 궁극적인 목표
- 우리 팀의 약속
- 팀원 소개 → 팀원 카드의 디자인을 기존 2단 → 1단으로 배치하고 마우스 오버 시 팀원 이미지가 강조되는 효과를 추가
- 우리 팀의 버킷리스트

#### [2. 모달창 (팀원 카드 클릭 시 이동하는 페이지)]



#### (구성 내용)

- 사진
- 목표, 키워드, 이름, MBTI, 블로그 주소
- 객관적으로 살펴본 나의 장점
- 협업을 하는 과정에서의 자신의 스타일
- 마무리 한마디
- 개인 방명록 저장 및 불러오기 → 1차 완성 후 방명록 수정/삭제 기능을 추가 → 개인 방명록을 팀 방명록으로 변경

### 3. 개발해야 하는 기능들

기능	Method	URL	Request	Response
----	--------	-----	---------	----------

기능	Method	URL	Request	Response
날씨 불러오기	GET	api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={API key}	-	{ "coord":{"lon":126.9068,"lat":37.4105}, "temp":283.23,"feels_like":282.4,"speed":0.51,"deg":240},"clouds":{"type":1,"id":8105,"country":"KR"}}
버킷 리스트 저장	POST	/bucket	{ 'num' : count, 'bucket':bucket_receive, 'done' :0 }	{ 'msg': '등록 완료!' }
버킷 리스트 불러오기	GET	/bucket	find({}, {'_id': False})	{'buckets': bucket_list}
버킷 리스트 완료하기	POST	/bucket/done	{ 'num': int(num_receive), '\$set': {'done': 1} }	{ 'msg': '버킷 완료!' }
팀 방명록 저장하기	GET	/member/guestbook	{ 'name': name_receive, 'comment': comment_receive }	{ 'msg': '응원댓글 작성 완료' }
팀 방명록 불러오기	POST	/member/guestbook	find({}))	{ 'comments': comment_list }
팀 방명록 수정하기	PUT	/member/guestbook/<id>	{ '_id': ObjectId(id), '\$set': {'name': name_receive} } { '_id': ObjectId(id), '\$set': {'comment': comment_receive} }	{ 'msg': '수정 완료!' }
팀 방명록 삭제하기	DELETE	/member/guestbook/<id>	{ '_id': ObjectId(id) }	{ 'msg': '삭제 완료!' }

## 4. 사용 기술 스택

- Front : HTML, CSS, Javascript, Bootstrap
- Back : JavaScript, Python, Flask, MongoDB
- Server : AWS, Linux, EC2

## 5. 작업 히스토리

- 2022.11.14
  - 와이어프레임 1차 작성 완료
  - SA 작성 완료
  - 팀 소개(메인) 페이지 작업 시작
- 2022.11.15
  - 팀 소개(메인) 페이지 작업 완료
  - 팀원 소개 페이지(모달창) 작업 시작
- 2022.11.16
  - 팀원 소개 페이지 작업 완료
- 2022.11.17
  - 개인 방명록 수정/삭제 기능 추가 작업 시작 → 팀 방명록으로 변경
  - 접속한 사람의 현재 위치 기준으로 날씨 정보를 불러오기 완료
  - 방명록 수정/삭제 기능 추가 완료

- 2022.11.18
  - 버킷리스트, 방명록 공백 입력 시 얼럿창 노출 추가 완료
  - 팀 소개 페이지 완성

## 6. 회고

### 1) 개발을 진행하면서 어려웠던 점

- 접속한 사람의 현재 위치 기준으로 날씨 정보를 불러오기
- 방명록 수정/삭제 기능 추가
- 버킷리스트, 방명록 공백 입력 시 얼럿창 노출 추가
- 각 팀원 소개 모달창에서 개인 방명록을 저장하고 불러오기 기능 구현 시, Restful한 API를 작성

### 2) 해결한 내용

- 접속한 사람의 현재 위치 기준으로 날씨 정보를 불러오기

```
//발전사항 : 서울 API만 받아오던것이 사용자의 위치를 기반으로 위도와 경도를 받아온 후
//          HTML에 위도와 경도(두번째 자리까지), 날씨 아이콘(날짜에 따라 변함), 위치(영어로 출력),
//          현재 위치 온도(화씨와 섭씨로 출력, 소숫점 2번째 자리까지, 섭씨는 1번째자리까지)
//날씨 정보 얻기
//현재 위도 경도로 위치잡아주고 기반으로 다른 값 출력
function showLocation(event) {
  var latitude = event.coords.latitude
  var longitude = event.coords.longitude
  document.querySelector("#latitude").textContent = Math.round(latitude * 100) / 100
  document.querySelector("#longitude").textContent = Math.round(longitude * 100) / 100

  //위도와 경도를 추가하여 weatherUrl이 바뀌므로 사용자의 지역에 따라 서로 다른 값 출력
  //위도와 경도에 따른 종속변수 #앞의 값에따라 변하는 변수들
  let apiKey = "c9094277ce42cae0392ac8222a2dd1f2"
  let weatherUrl = "https://api.openweathermap.org/data/2.5/weather?lat=" + latitude
    + "&lon=" + longitude
    + "&appid=" + apiKey;
  //api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={API key}
  let options = { method: 'GET' }
  $.ajax(weatherUrl, options).then((response) => {
    console.log(response)
    let icon = response.weather[0].icon
    let iconUrl = "http://openweathermap.org/img/wn/" + icon + "@2x.png"
    let img = document.querySelector("#wicon")
    let name = response.name
    img.src = iconUrl
    document.querySelector("#temp").textContent = /*(" + */Math.round((response.main.temp - 273)*10) / 10 + "°C"/+ " )"*/ /* (response.main.temp)+"°F" 화씨까지 띄우기 // 현재 온도
    document.querySelector("#name").textContent = name
  }).catch((error) => {
    console.log(error)
  })
}

function showError(event) {
  alert("위치 정보를 얻을 수 없습니다.")
}

window.addEventListener('load', () => {
  if(window.navigator.geolocation) {
    window.navigator.geolocation.getCurrentPosition(showLocation, showError)
  }
})
```

- 방명록 수정/삭제 기능 추가

```
@app.route("/member/guestbook/<id>", methods=["PUT"])
def member_update(id):
    name_receive = request.form['name_receive']
    comment_receive = request.form['comment_receive']
    db.guestbook.update_one({'_id': ObjectId(id)}, {'$set': {'name': name_receive}})
    db.guestbook.update_one({'_id': ObjectId(id)}, {'$set': {'comment': comment_receive}})
    return jsonify({'msg': '수정 완료!'})
```

```
@app.route("/member/guestbook/<id>", methods=["DELETE"])
def member_delete(id):
    db.guestbook.delete_one({'_id': ObjectId(id)})
    return jsonify({'msg': '삭제 완료!'})
```

```
function update_member(id) {
    //alert(id) 정상처럼 받아오면 JS에는 문제가 없는것
    let name = $('#name').val()
    let comment = $('#comment').val()
    $.ajax({
        type: "PUT",
        url: `/member/guestbook/${id}`,
        data: {
            name_receive: name,
            comment_receive: comment,
        },
        success: function (response) {
            alert(response["msg"])
        }
    });
    window.location.reload()
}

function delete_member(id) {
    $.ajax({
        type: "DELETE",
        url: `/member/guestbook/${id}`,
        data: {},
        success: function (response) {
            alert(response["msg"])
        }
    });
    window.location.reload()
}

}
```

- 버킷리스트, 방명록 공백 입력 시 얼럿창 노출 추가

```
function save_bucket() {
    let bucket = $('#bucket').val()
    bucket = bucket.replaceAll(" ", '')
    if (bucket == "") {
        return alert("공백은 허용 되지 않습니다.")
    }
}
```

```
function save_comment() {
    let name = $('#name').val()
    let comment = $('#comment').val()
    name = name.replaceAll(" ", '')
    if (name == "" || comment == "") {
        return alert("공백은 허용 되지 않습니다.")
    }
}
```

### 3) 해결하지 못한 것

- 각 팀원 소개 모달창에서 개인 방명록을 저장하고 불러오기 기능 구현 시, Restful한 API를 작성하지 못함 → 개인 방명록이 아닌 팀 방명록으로 대체