



# [미니프로젝트] 1. 새가 날아든다



140자 이내 익명 인스턴트 메모 서비스입니다. 이번 시간을 통해 다음 내용을 배울 수 있습니다.

- 1) 익명으로 메모 작성하기
- 2) 최신 순으로 메모 정렬하고 타임라인에 보여주기
- 3) 24시간이 지난 데이터는 보여주지 않기
- 4) 메모 수정하기

## 1. [새가날아든다] - 프로젝트 세팅



VS Code 파일→폴더열기를 클릭해서,  
sparta > project > birdcomming 폴더를 열고 시작!

### ▼ 1) 완성작보기

- 링크를 통해 살펴보기

## ▼ 2) 새 프로젝트니까 - 가상환경 다시 잡기

- 터미널 → 새 터미널을 열어서, 아래를 입력!

### ▼ 윈도우

```
python -m venv myenv
```

Shell ▾

### ▼ 맥

```
python3 -m venv myenv
```

Shell ▾

- app.py 파일을 생성합니다. 내용은 없어도 돼요! (이름은 아무거나 정하셔도 괜찮아요!)
- 파이썬 파일을 연 채로 다시, 터미널 → 새 터미널을 열면, 공구함 리로딩 완료!
- 공구함 만든 김에, 필요한 패키지를 미리 설치해둡시다!

```
pip install flask pymongo
```

Shell ▾

## ▼ 3) 시작 코드

## ▼ index.html

```

<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <
name="viewport" content="width=device-width, initial-scale=1.0">
딩클럽 | 새가 날아든다</title> <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/boo
integrity="sha384-
Vkoo8x4CGs03+Hhvxv8T/Q5PaXtkKtu6ug5T0eNV6gBiFeWPGFN9MuhOf23Q9IfjH
crossorigin="anonymous"> <script src="https://code.jquery.com/jc
3.4.1.slim.min.js" integrity="sha384-
J6qa4849bLE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+r
crossorigin="anonymous"></script> <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper
integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAc
crossorigin="anonymous"></script> <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/boots
integrity="sha384-
wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl30g8ifwB6
crossorigin="anonymous"></script> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.m
<script> function openclose() { let status = $('#post-box').css(
(status === 'block') { $('#post-box').hide(); } else { $('#post-
} </script> <style> .area-edit { display: none; } .wrap { width:
10px auto; } #contents { width: 700px; } .area-write { display:
direction: column; align-items: flex-end; justify-content: cente
card { margin: 20px 0 20px 0; } #post-box { display: none; } </s
<body> <div class="wrap"> <div class="header"> <h1>새가 날아든다</h
onclick="openclose()" type="button" class="btn btn-outline-info
싶은 소식</button>을 입력해주세요. 24시간이 지난 뒤에는 사라집니다. </p> </di
id="post-box"> <div class="area-write"> <textarea placeholder="2
입력해주세요" name="contents" id="contents" cols="30" rows="10"></t
class="btn btn-danger" onclick="writePost()">작성하기</button> </d
id="cards-box" class="area-read"> <div class="card"> <header cla
header"> <p class="card-header-title"> Component </p> <a href="#"
header-icon" aria-label="more options"> <span class="icon"> <i c
angle-down" aria-hidden="true"></i> </span> </a> </header> <div
content"> <div class="content"> Lorem ipsum dolor sit amet, cons
adipiscing elit. Phasellus nec iaculis mauris. <a href="#">@bulm
href="#">#css</a> <a href="#">#responsive</a> <br> <time datetin
1">11:09 PM - 1 Jan 2016</time> </div> </div> <footer class="car
href="#" class="card-footer-item">Save</a> <a href="#" class="ca
item">Edit</a> <a href="#" class="card-footer-item">Delete</a> <
</div> </div> </body> </html>

```

HTML ▼

## ▼ app.py

```
from flask import Flask, render_template, request, jsonify
from pymongo import MongoClient from datetime import
datetime, timedelta app = Flask(__name__) # URL 별로 함수명이 같거
나, # route('/') 등의 주소가 같으면 안됩니다. client =
MongoClient('localhost', 27017) db = client.bird_comming
@app.route('/') def home(): return
render_template('index.html') if __name__ == '__main__':
app.run('0.0.0.0', port=5000, debug=True)
```

Python ▼

## 2. [새가날아든다] - 익명으로 메모 작성하기

## ▼ 4) 익명으로 메모 작성 기능 만들기



익명으로 메모를 작성하기 위해서는 다음 요소가 필요합니다.

**[클라이언트]**

- 1) 사용자가 작성한 메모 내용을 가져와서 올바른 입력인지 확인한다.
- 2) 랜덤으로 유저이름을 생성한다.
- 3) 유저이름과 메모 내용을 서버(/message)에 전달한다.

**[서버]**

- 1) 유저이름과 메모를 받아 DB에 저장하기

## ▼ (1) 클라이언트 코딩

- ▼ 1: 사용자가 작성한 메모 내용을 가져와서 올바른 입력인지 확인한다.

```
function isValidContents(contents) { // 1. 사용자가 입력을 하였
  는지 확인한다. if (contents == '') { alert('내용을 입력해주세요');
  return false; } // 2. 사용자의 입력 내용이 141자 이상인지 확인한다.
  if (contents.trim().length > 140) { alert('공백 포함 140자 이
  하로 입력해주세요'); return false; } // 3. 올바른 입력일 경우 true
  를 리턴한다. return true; } function writePost() { // 1. 사용
  자 입력 값을 가져온다. let contents = $('#contents').val(); //
  2. 사용자가 올바르게 입력했는지 isValidContents 함수를 통해 확인한다.
  if (isValidContents(contents) == false) { return; } }
```

JavaScript ▼

- ▼ 2: 랜덤으로 유저이름을 생성한다.

```
// 필요한 길이에 맞게 임의의 문자열을 만든다. function
  genRandomName(length) { // 돌려줄 임의의 문자열 let result = '';
  임의의 문자열이 될 수 있는 후보 let characters =
  'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456
  let charactersLength = characters.length; // 주어진 길이에 맞게 문자
  를 만든다. for (let i = 0; i < length; i++) { // 0과 같거나 그보다
  charactersLength 보다 작은 임의의 숫자를 뽑는다. let number =
  Math.random() * charactersLength; // number 가 소수라면 정수로 만
  (ex. 7.23 -> 7) let index = Math.floor(number); // index 위치
  는 문자열 하나를 result에 더한다. result += characters.charAt(index);
  // 임의의 문자열을 반환한다. return result; } function writePost()
  let contents = $('#contents').val(); if
  (isValidContents(contents) === false) { return; } // 길이 10의
  문자열을 뽑는다. let username = genRandomName(10); }
```

JavaScript ▼

## ▼ 3: 유저이름과 메모 내용을 서버(/message)에 전달한다.

```
function writePost() { let contents =
$('#contents').val(); if (isValidContents(contents) ===
false) { return; } let username = getRandomName(10);
$.ajax({ type: "POST", url: "/message", data: {
'username_give': username, 'contents_give': contents },
success: function (response) { if (response['result'] ==
'success') { // 서버에서 받아온 메시지를 보여준다.
alert(response['msg']); // 메시지를 저장한 후 창을 새로고침한다. ->
새롭게 조회하기 위해! window.location.reload(); } } }); }
```

JavaScript ▾

## ▼ (2) 서버 코딩

## ▼ 1: 유저이름과 메모를 받아 DB에 저장하기

```
@app.route('/message', methods=["POST"]) def
set_message(): # 클라이언트가 전달한 username, contents를 변수에
저장한다. username_receive = request.form['username_give']
contents_receive = request.form['contents_give'] # DB에 저
장할 대상을 만든다. doc = { 'username': username_receive,
'contents': contents_receive, # '지금 시각'을 created_at 이라
는 이름으로 함께 저장한다. 'created_at': datetime.now() } # DB에
메모를 저장한다. db.messages.insert_one(doc) # 성공 메시지를 반환한
다. return jsonify({'result': 'success', 'msg': '메시지 작성
에 성공하였습니다!'})
```

Python ▾

## ▼ (3) 완성코드

## ▼ index.html

```
<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8
name="viewport" content="width=device-width, initial-scale=1
딩클럽 | 새가 날아든다</title> <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css
integrity="sha384-
Vkoo8x4CGs03+Hhxv8T/Q5PaXtkKtu6ug5T0eNV6gBiFeWPGFN9MuhOf23Q9
crossorigin="anonymous"> <script src="https://code.jquery.co
3.4.1.slim.min.js" integrity="sha384-
J6qa4849bLE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJ
crossorigin="anonymous"></script> <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/
integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMf
crossorigin="anonymous"></script> <script
```

```

src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/b
integrity="sha384-
wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl30g8i
crossorigin="anonymous"></script> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jque
<script> function isValidContents(contents) { if (contents =
입력해주세요'); return false; } if (contents.trim().length > 140
140자 이하로 입력해주세요'); return false; } return true; } functi
genRandomName(length) { let result = ''; let characters =
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456
charactersLength = characters.length; for (let i = 0; i < le
number = Math.random() * charactersLength; let index = Math.
result += characters.charAt(index); } return result; } funct
let contents = $('#contents').val(); if (isValidContents(con
return; } let username = genRandomName(10); $.ajax({ type: "
"/message", data: { 'username_give': username, 'contents_giv
success: function (response) { if (response['result'] == 'su
alert(response['msg']); window.location.reload(); } } }); }
openclose() { let status = $('#post-box').css('display'); if
'block') { $('#post-box').hide(); } else { $('#post-box').sh
<style> .area-edit { display: none; } .wrap { width: 700px;
} #contents { width: 700px; } .area-write { display: flex; f
column; align-items: flex-end; justify-content: center; } .c
margin: 20px 0 20px 0; } #post-box { display: none; } </styl
<div class="wrap"> <div class="header"> <h1>새가 날아든다</h1> <
onclick="openclose()" type="button" class="btn btn-outline-i
싶은 소식</button>을 입력해주세요. 24시간이 지난 뒤에는 사라집니다. </p> <
id="post-box"> <div class="area-write"> <textarea placeholder
입력해주세요" name="contents" id="contents" cols="30" rows="10">
class="btn btn-danger" onclick="writePost()">작성하기</button>
id="cards-box" class="area-read"> <div class="card"> <header
header"> <p class="card-header-title"> Component </p> <a href="#">@
header-icon" aria-label="more options"> <span class="icon">
angle-down" aria-hidden="true"></i> </span> </a> </header> <
content"> <div class="content"> Lorem ipsum dolor sit amet,
adipiscing elit. Phasellus nec iaculis mauris. <a href="#">@
href="#">#css</a> <a href="#">#responsive</a> <br> <time dat
1">11:09 PM - 1 Jan 2016</time> </div> </div> <footer class=
href="#" class="card-footer-item">Save</a> <a href="#" class
item">Edit</a> <a href="#" class="card-footer-item">Delete</
</div> </div> </body> </html>

```

JavaScript ▾

## ▼ app.py

```

from flask import Flask, render_template, request, jsonify
from pymongo import MongoClient from datetime import

```

```

datetime, timedelta app = Flask(__name__) # URL 별로 함수명이
같거나, # route('/') 등의 주소가 같으면 안됩니다. client =
MongoClient('localhost', 27017) db = client.bird_comming
@app.route('/') def home(): return
render_template('index_start.html') @app.route('/message',
methods=["POST"]) def set_message(): username_receive =
request.form['username_give'] contents_receive =
request.form['contents_give'] doc = { 'username':
username_receive, 'contents': contents_receive,
'created_at': datetime.now() }
db.messages.insert_one(doc) return jsonify({'result':
'success', 'msg': '메시지 작성에 성공하였습니다!'}) if __name__
== '__main__': app.run('0.0.0.0', port=5000, debug=True)

```

Python ▾

### 3. [새가날아든다] - 최신순으로 메모 정렬하기



## ▼ 5) 최신 순으로 메모를 정렬하고 타임라인에서 조회하기

✓ 필요한 기능은 다음과 같습니다.

## [서버]

1) 메모 목록을 최신 순으로 정렬해서 돌려주기

## [클라이언트]

1) 사용자가 접속할 때 메모 목록을 요청하기

2) 전달 받은 메모 목록을 화면에 보여주기

## ▼ (1) 서버 코딩

## ▼ 1: 메모 목록을 최신 순으로 정렬해서 돌려주기

```
@app.route('/message', methods=["GET"]) def
get_messages(): # created_at 기준 내림차순으로 정렬한다.
messages = list(db.messages.find({}, {'_id':
False})).sort('created_at', -1)) # 결과를 messages 라는 key에
저장하여 돌려준다. return jsonify({'result': 'success',
'messages': messages})
```

Python ▼

## ▼ (2) 클라이언트 코딩

## ▼ 1: 사용자가 접속할 때 메모 목록을 요청하기

```
// 새로그침할 때마다 getMessages() 함수를 호출한다.
$(document).ready(function () { getMessages(); })
function getMessages() { // 기존 메모를 삭제한다. $('#cards-
box').empty(); // 메모 목록을 요청한다. $.ajax({ type: "GET",
url: "/message", data: {}, success: function (response) {
alert('메모를 요청하였습니다. ');
console.log(response['messages']); } }); }
```

JavaScript ▼

## ▼ 2: 전달 받은 메모 목록을 화면에 보여주기

```
$(document).ready(function () { getMessages(); })
function getMessages() { $('#cards-box').empty();
$.ajax({ type: "GET", url: "/message", data: {}, success:
function (response) { if (response['result'] ==
'success') { let messages = response['messages']; for
(let i = 0; i < messages.length; i++) { let message =
```

```

messages[i]; let username = message['username']; let
contents = message['contents']; let created_at =
message['created_at']; // addHTML 함수를 이용해 메모마다 HTML을
만들고 #cards-box에 붙인다. addHTML(username, contents,
created_at); } } else { alert('메시지를 불러오는데 실패했습니
다.');
```

JavaScript ▾

## ▼ (3) 완성 코드

## ▼ app.py

```

from flask import Flask, render_template, request, jsonify
from pymongo import MongoClient from datetime import
datetime, timedelta app = Flask(__name__) # URL 별로 함수명이
같거나, # route('/') 등의 주소가 같으면 안됩니다. client =
MongoClient('localhost', 27017) db = client.bird_comming
@app.route('/') def home(): return
render_template('index.html') @app.route('/message',
methods=["GET"]) def get_messages(): messages =
list(db.messages.find({}, {'_id':
False})).sort('created_at', -1)) return jsonify({'result':
'success', 'messages': messages}) @app.route('/message',
methods=["POST"]) def set_message(): username_receive =
request.form['username_give'] contents_receive =
request.form['contents_give'] doc = { 'username':
username_receive, 'contents': contents_receive,
'created_at': datetime.now() }
db.messages.insert_one(doc) return jsonify({'result':
```

```
'success', 'msg': '메시지 작성에 성공하였습니다!'})) if __name__
== '__main__': app.run('0.0.0.0', port=5000, debug=True)
```

JavaScript ▾

## ▼ index.html

```
<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8
name="viewport" content="width=device-width, initial-scale=1
딩클럽 | 새가 날아든다</title> <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css
integrity="sha384-
Vkoo8x4CGs03+HhXv8T/Q5PaXtkKtu6ug5T0eNV6gBiFeWPGFN9MuhOf23Q9
crossorigin="anonymous"> <script src="https://code.jquery.co
3.4.1.slim.min.js" integrity="sha384-
J6qa4849bLE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJ
crossorigin="anonymous"></script> <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/
integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMf
crossorigin="anonymous"></script> <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/b
integrity="sha384-
wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl30g8i
crossorigin="anonymous"></script> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jque
<script> $(document).ready(function () { getMessages(); }) f
getMessages() { $('#cards-box').empty(); $.ajax({ type: "GET
data: {}, success: function (response) { if (response['resul
let messages = response['messages']; for (let i = 0; i < mes
{ let message = messages[i]; let username = message['usernam
message['contents']; let created_at = message['created_at'];
contents, created_at); } } else { alert('메시지를 불러오는데 실패했
function addHTML(username, contents, created_at) { let tempH
makeMessage(username, contents, created_at); $('#cards-box')
} function makeMessage(username, contents, created_at) { ret
class="card custom-card"> <div class="card-body"> <textarea
textarea" class="area-edit" cols="80"></textarea> <h5 id="${
class="card-title">${contents}</h5> <h6 id="${username}-user
subtitle mb-2 text-muted">${username}</h6> <h6 class="card-s
muted">${created_at}</h6> </div> <footer class="card-footer"
id="${username}-edit" href="#" class="card-footer-item"
onclick="editPost('${username}')">수정</a> <a id="${username}
class="card-footer-item area-edit" onclick="hideEdits('${use
id="${username}-submit" href="#" class="card-footer-item are
onclick="submitEdit('${username}')">수정완료</a> </footer> </d.
isValidContents(contents) { if (contents == '') { alert('내용
return false; } if (contents.trim().length > 140) { alert('공
력해주세요'); return false; } return true; } function getRandom
```

```

result = ''; let characters =
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456
charactersLength = characters.length; for (let i = 0; i < le
number = Math.random() * charactersLength; let index = Math.
result += characters.charAt(index); } return result; } funct
let contents = $('#contents').val(); if (isValidContents(con
return; } let username = getRandomName(10); $.ajax({ type: "
"/message", data: { 'username_give': username, 'contents_giv
success: function (response) { if (response['result'] == 'su
alert(response['msg']); window.location.reload(); } } }); }
openclose() { let status = $('#post-box').css('display'); if
'block') { $('#post-box').hide(); } else { $('#post-box').sh
<style> .area-edit { display: none; } .wrap { width: 700px;
} #contents { width: 700px; } .area-write { display: flex; f
column; align-items: flex-end; justify-content: center; } .c
margin: 20px 0 20px 0; } #post-box { display: none; } </styl
<div class="wrap"> <div class="header"> <h1>새가 날아든다</h1> <
onclick="openclose()" type="button" class="btn btn-outline-i
싶은 소식</button>을 입력해주세요. 24시간이 지난 뒤에는 사라집니다. </p> <
id="post-box"> <div class="area-write"> <textarea placeholder
입력해주세요" name="contents" id="contents" cols="30" rows="10">
class="btn btn-danger" onclick="writePost()">작성하기</button>
id="cards-box" class="area-read"> <div class="card"> <header
header"> <p class="card-header-title"> Component </p> <a href="#">@
header-icon" aria-label="more options"> <span class="icon">
angle-down" aria-hidden="true"></i> </span> </a> </header> <
content"> <div class="content"> Lorem ipsum dolor sit amet,
adipiscing elit. Phasellus nec iaculis mauris. <a href="#">@
href="#">#css</a> <a href="#">#responsive</a> <br> <time dat
1">11:09 PM - 1 Jan 2016</time> </div> </div> <footer class=
href="#" class="card-footer-item">Save</a> <a href="#" class
item">Edit</a> <a href="#" class="card-footer-item">Delete</
</div> </div> </body> </html>

```

JavaScript ▾

#### 4. [새가날아든다] - 24시간 이내 데이터만 보여주기

## ▼ 6) 24시간이 지난 데이터는 보여주지 않기

✓ 필요한 기능은 다음과 같습니다.

## [서버]

1) 최근 24시간 이내 메모만 가져와서 최신 순으로 정렬 후 반환하기

## ▼ (1) 서버 코딩

```
@app.route('/message', methods=["GET"]) def get_messages(): #
    지금 시각 구하기 date_now = datetime.now() # 24시간 전 구하기
    date_before = date_now - timedelta(days=1) messages =
    list(db.messages.find({'created_at': {'$gte': date_before,
    '$lte': date_now}}, {'_id': False}).sort('created_at', -1))
    return jsonify({'result': 'success', 'messages': messages})
```

Python ▼

## ▼ (2) 완성 코드

```
from flask import Flask, render_template, request, jsonify
from pymongo import MongoClient from datetime import
datetime, timedelta app = Flask(__name__) # URL 별로 함수명이 같거
나, # route('/') 등의 주소가 같으면 안됩니다. client =
MongoClient('localhost', 27017) db = client.bird_comming
@app.route('/') def home(): return
render_template('index.html') @app.route('/message', methods=
["GET"]) def get_messages(): date_now = datetime.now()
date_before = date_now - timedelta(days=1) messages =
list(db.messages.find({'created_at': { '$gte': date_before,
'$lte': date_now}}, {'_id': False}).sort('created_at', -1))
return jsonify({'result': 'success', 'messages': messages})
@app.route('/message', methods=["POST"]) def set_message():
username_receive = request.form['username_give']
contents_receive = request.form['contents_give'] doc = {
'username': username_receive, 'contents': contents_receive,
'created_at': datetime.now() } db.messages.insert_one(doc)
return jsonify({'result': 'success', 'msg': '메시지 작성에 성공하였
습니다!.'}) if __name__ == '__main__': app.run('0.0.0.0',
port=5000, debug=True)
```

Python ▼

## 5. [새가날아든다] - 메모 수정하고 완성하기

## ▼ 7) 작성한 메모 수정 기능 만들기

✓ 필요한 기능은 다음과 같습니다.

## [클라이언트]

- 1) 수정 버튼을 누르면 (취소, 수정) 버튼이 나타나고, 기존 내용을 입력 칸에 담아 두기
- 2) 취소 버튼을 누르면 다시 원래대로 돌아가기
- 3) 수정완료 버튼을 누르면 메모 내용, 유저이름을 서버로 전달하기

## [서버]

- 1) 유저 요청에 따라 메모 내용을 수정하고, 생성 시각을 업데이트하기

## ▼ (1) 클라이언트 코딩

- ▼ 1: 수정 버튼을 누르면 (취소, 수정) 버튼이 나타나고, 기존 내용을 입력 칸에 담아 두기

```
// 수정 버튼을 누르면 editPost 함수가 호출된다. // 어떤 메모인지 알기
// 위해 username 전달 받는다. function editPost(username) { //
// username이 일치하는 메모에 수정 입력칸을 보여준다.
showEdits(username); // 기존 메모 내용을 가져온다. let contents
= $('#${username}-contents').text(); // 수정 입력 칸에 기존 메모
내용을 넣어둔다. $('#${username}-textarea').val(contents);
} function showEdits(username) { // 수정 입력 칸, 수정완료 버튼,
취소 버튼을 보여준다. $('#${username}-textarea').show();
$('#${username}-submit').show(); $('#${username}-cancel').show(); // 메모 내용, 수정 버튼을 숨긴다.
$('#${username}-contents').hide(); $('#${username}-edit').hide(); }
```

JavaScript ▼

- ▼ 2: 취소 버튼을 누르면 다시 원래대로 돌아가기

```
// 취소 버튼을 누르면 hideEdits 함수가 호출된다. // 어떤 메모인지 알기
// 위해 username 전달 받는다. function hideEdits(username) { //
// 수정 입력 칸, 수정완료 버튼, 취소 버튼을 숨긴다. $('#${username}-textarea').hide(); $('#${username}-submit').hide();
$('#${username}-cancel').hide(); // 메모 내용, 수정 버튼을 보여
준다. $('#${username}-contents').show(); $('#${username}-edit').show(); }
```

JavaScript ▼

- ▼ 3: 수정완료 버튼을 누르면 메모 내용, 유저이름을 서버로 전달하기

```
// 수정완료 버튼을 누르면 submitEdit 함수가 호출된다. // 어떤 메모인지
알기 위해 username 전달 받는다. function submitEdit(username) {
// 수정된 메모 내용을 가져오고 올바른지 검사한다. let contents =
$('#${username}-textarea').val(); if
(isValidContents(contents) == false) { return; } // 메모 내
용 수정을 서버에 요청한다. $.ajax({ type: "POST", url:
"/message/edit", data: { username_give: username,
contents_give: contents }, success: function (response) {
if (response['result'] == 'success') {
alert(response['msg']); // 변경된 메모를 보기 위해 창을 새로고침한
다. window.location.reload(); } else { alert('메시지 변경에 실
패했습니다. '); } } }); }
```

JavaScript ▾

## ▼ (2) 서버 코딩

### ▼ 1: 유저 요청에 따라 메모 내용을 수정하고, 생성 시각을 업데이트하기

```
@app.route('/message/edit', methods=["POST"]) def
edit_message(): username_receive =
request.form['username_give'] contents_receive =
request.form['contents_give'] # username 기준으로 메시지를 찾아
내용과 생성 시각을 업데이트합니다.
db.messages.update_one({'username': username_receive}, {
'$set': {'contents': contents_receive, 'created_at':
datetime.now()}}) return jsonify({'result': 'success',
'msg': '메시지 변경에 성공하였습니다!'})
```

Python ▾

## ▼ (3) 완성 코드

### ▼ index.html

```
<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8
name="viewport" content="width=device-width, initial-scale=1
딩클럽 | 새가 날아든다</title> <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css
integrity="sha384-
Vkoo8x4CGs03+Hhvxv8T/Q5PaXtkKtu6ug5T0eNV6gBiFeWPGFN9MuhOf23Q9
crossorigin="anonymous"> <script src="https://code.jquery.co
3.4.1.slim.min.js" integrity="sha384-
J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJ
crossorigin="anonymous"></script> <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/
integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMf
```



```

crossorigin="anonymous"></script> <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/b
integrity="sha384-
wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl30g8i
crossorigin="anonymous"></script> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jque
<script> $(document).ready(function () { getMessages(); }) f
getMessages() { $('#cards-box').empty(); $.ajax({ type: "GET
data: {}, success: function (response) { if (response['resul
let messages = response['messages']; for (let i = 0; i < mes
{ let message = messages[i]; let username = message['usernam
message['contents']; let created_at = message['created_at'];
contents, created_at, i); } } else { alert('메시지를 불러오는데 실패
} function addHTML(username, contents, created_at, i) { let
makeMessage(username, contents, created_at); $('#cards-box')
} function makeMessage(username, contents, created_at, i) {
class="card custom-card"> <div class="card-body"> <textarea
textarea" class="area-edit" cols="80"></textarea> <h5 id="${
class="card-title">${contents}</h5> <h6 id="${username}-user
subtitle mb-2 text-muted">${username}</h6> <h6 class="card-s
muted">${created_at}</h6> </div> <footer class="card-footer"
id="${username}-edit" href="#" class="card-footer-item"
onclick="editPost('${username}')">수정</a> <a id="${username}
class="card-footer-item area-edit" onclick="hideEdits('${use
id="${username}-submit" href="#" class="card-footer-item are
onclick="submitEdit('${username}')">수정완료</a> </footer> </d
isValidContents(contents) { if (contents == '') { alert('내용
return false; } if (contents.trim().length > 140) { alert('공
력해주세요'); return false; } return true; } function getRandom
result = ''; let characters =
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456
charactersLength = characters.length; for (let i = 0; i < le
number = Math.random() * charactersLength; let index = Math.
result += characters.charAt(index); } return result; } funct
let contents = $('#contents').val(); if (isValidContents(con
return; } let username = getRandomName(10); $.ajax({ type: "
"/message", data: { 'username_give': username, 'contents_giv
success: function (response) { if (response['result'] == 'su
alert(response['msg']); window.location.reload(); } } }); }
openClose() { let status = $('#post-box').css('display'); if
'block') { $('#post-box').hide(); } else { $('#post-box').sh
editPost(username) { showEdits(username); let contents = $(`
contents`).text(); $('#${username}-textarea').val(contents);
showEdits(username) { $('#${username}-textarea').show(); $(`
submit').show(); $('#${username}-cancel').show(); $('#${user
contents').hide(); $('#${username}-edit').hide(); } function
hideEdits(username) { $('#${username}-textarea').hide(); $(`
submit').hide(); $('#${username}-cancel').hide(); $('#${user

```



```

contents`).show(); $('#${username}-edit').show(); } function
submitEdit(username) { let contents = $('#${username}-textare
(isValidContents(contents) == false) { return; } $.ajax({ ty
"/message/edit", data: { username_give: username, contents_g
success: function (response) { if (response['result'] == 'su
alert(response['msg']); window.location.reload(); } else { a
했습니다.');
```

JavaScript ▾

## ▼ app.py

```

from flask import Flask, render_template, request, jsonify
from pymongo import MongoClient from datetime import
datetime, timedelta app = Flask(__name__) # URL 별로 함수명이
같거나, # route('/') 등의 주소가 같으면 안됩니다. client =
MongoClient('localhost', 27017) db = client.bird_comming
@app.route('/') def home(): return
render_template('index.html') @app.route('/message',
methods=["GET"]) def get_messages(): date_now =
datetime.now() date_before = date_now - timedelta(days=1)
messages = list(db.messages.find({'created_at': { '$gte':
date_before, '$lte': date_now}}, {'_id':
False})).sort('created_at', -1) return jsonify({'result':
'success', 'messages': messages}) @app.route('/message',
methods=["POST"]) def set_message(): username_receive =
request.form['username_give'] contents_receive =
request.form['contents_give'] doc = { 'username':
```

```
username_receive, 'contents': contents_receive,
'created_at': datetime.now() }
db.messages.insert_one(doc) return jsonify({'result':
'success', 'msg': '메시지 작성에 성공하였습니다!'})
@app.route('/message/edit', methods=["POST"]) def
edit_message(): username_receive =
request.form['username_give'] contents_receive =
request.form['contents_give']
db.messages.update_one({'username': username_receive}, {
'$set': {'contents': contents_receive, 'created_at':
datetime.now()}}) return jsonify({'result': 'success',
'msg': '메시지 변경에 성공하였습니다!'}) if __name__ ==
'__main__': app.run('0.0.0.0', port=5000, debug=True)
```

Python ▾

## ▼ 8) 완성 코드

## ▼ index.html

```

<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <
name="viewport" content="width=device-width, initial-scale=1.0">
딩클럽 | 새가 날아든다</title> <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/boo
integrity="sha384-
Vkoo8x4CGs03+Hhvxv8T/Q5PaXtkKtu6ug5T0eNV6gBiFeWPGFN9MuhOf23Q9IfjH
crossorigin="anonymous"> <script src="https://code.jquery.com/jc
3.4.1.slim.min.js" integrity="sha384-
J6qa4849bLE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+r
crossorigin="anonymous"></script> <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper
integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAc
crossorigin="anonymous"></script> <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/boots
integrity="sha384-
wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl30g8ifwB6
crossorigin="anonymous"></script> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.m
<script> $(document).ready(function () { getMessages(); }) funct
getMessages() { $('#cards-box').empty(); $.ajax({ type: "GET", u
data: {}, success: function (response) { if (response['result']
let messages = response['messages']; for (let i = 0; i < message
{ let message = messages[i]; let username = message['username'];
message['contents']; let created_at = message['created_at']; add
contents, created_at, i); } } else { alert('메시지를 불러오는데 실패했
} function addHTML(username, contents, created_at, i) { let temp
makeMessage(username, contents, created_at); $('#cards-box').app
} function makeMessage(username, contents, created_at, i) { retu
class="card custom-card"> <div class="card-body"> <textarea id="
textarea" class="area-edit" cols="80"></textarea> <h5 id="${user
class="card-title">${contents}</h5> <h6 id="${username}-username
subtitle mb-2 text-muted">${username}</h6> <h6 class="card-subti
muted">${created_at}</h6> </div> <footer class="card-footer"> <a
id="${username}-edit" href="#" class="card-footer-item"
onclick="editPost('${username}')">수정</a> <a id="${username}-can
class="card-footer-item area-edit" onclick="hideEdits('${username
id="${username}-submit" href="#" class="card-footer-item area-ec
onclick="submitEdit('${username}')">수정완료</a> </footer> </div>`
isValidContents(contents) { if (contents == '') { alert('내용을 입
return false; } if (contents.trim().length > 140) { alert('공백 포
력해주세요'); return false; } return true; } function getRandomName
result = ''; let characters =

```

```
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789'
charactersLength = characters.length; for (let i = 0; i < length
number = Math.random() * charactersLength; let index = Math.floa
result += characters.charAt(index); } return result; } function
let contents = $('#contents').val(); if (isValidContents(content
return; } let username = genRandomName(10); $.ajax({ type: "POST
"/message", data: { 'username_give': username, 'contents_give':
success: function (response) { if (response['result'] == 'succes
alert(response['msg']); window.location.reload(); } } }); } func
openClose() { let status = $('#post-box').css('display'); if (st
'block') { $('#post-box').hide(); } else { $('#post-box').show()
editPost(username) { showEdits(username); let contents = $('#${u
contents').text(); $('#${username}-textarea').val(contents); } f
showEdits(username) { $('#${username}-textarea').show(); $('#${u
submit').show(); $('#${username}-cancel').show(); $('#${username
contents').hide(); $('#${username}-edit').hide(); } function
hideEdits(username) { $('#${username}-textarea').hide(); $('#${u
submit').hide(); $('#${username}-cancel').hide(); $('#${username
contents').show(); $('#${username}-edit').show(); } function
submitEdit(username) { let contents = $('#${username}-textarea')
(isValidContents(contents) == false) { return; } $.ajax({ type:
"/message/edit", data: { username_give: username, contents_give:
success: function (response) { if (response['result'] == 'succes
alert(response['msg']); window.location.reload(); } else { alert
했습니다.');
```

JavaScript ▾

▼ app.py

```

from flask import Flask, render_template, request, jsonify
from pymongo import MongoClient from datetime import
datetime, timedelta app = Flask(__name__) # URL 별로 함수명이 같거
나, # route('/') 등의 주소가 같으면 안됩니다. client =
MongoClient('localhost', 27017) db = client.bird_comming
@app.route('/') def home(): return
render_template('index.html') @app.route('/message', methods=
["GET"]) def get_messages(): date_now = datetime.now()
date_before = date_now - timedelta(days=1) messages =
list(db.messages.find({'created_at': { '$gte': date_before,
'$lte': date_now}}, {'_id': False}).sort('created_at', -1))
return jsonify({'result': 'success', 'messages': messages})
@app.route('/message', methods=["POST"]) def set_message():
username_receive = request.form['username_give']
contents_receive = request.form['contents_give'] doc = {
'username': username_receive, 'contents': contents_receive,
'created_at': datetime.now() } db.messages.insert_one(doc)
return jsonify({'result': 'success', 'msg': '메시지 작성에 성공하였
습니다!'}) @app.route('/message/edit', methods=["POST"]) def
edit_message(): username_receive =
request.form['username_give'] contents_receive =
request.form['contents_give']
db.messages.update_one({'username': username_receive}, {
'$set': {'contents': contents_receive, 'created_at':
datetime.now()}}) return jsonify({'result': 'success', 'msg':
'메시지 변경에 성공하였습니다!'}) if __name__ == '__main__':
app.run('0.0.0.0', port=5000, debug=True)

```

Python ∨