

# Mini Project

**Team : 조장은너가해조**

👤 오영환, 이정기, 배진영, 박경진, 김호진

📅 11.22 - 12.09

## ▼ 프로젝트 사전 계획

- ✓ 22일 (화) : 자료구조 알고리즘, JavaScript 올인원
- ✓ 23일 (수) : 자료구조 알고리즘, JavaScript 올인원
- ✓ 24일 (목) : 자료구조 알고리즘, JavaScript 올인원
- ✓ 25일 (금) : 자료구조 알고리즘\_4주차, JavaScript 올인원 1주차



- ✓ 28일 (월) : 자료구조 알고리즘, JavaScript 올인원
- ✓ 29일 (화) : 복습, git 활용 기초
- ✓ 30일 (수) : 복습, git 활용 기초
- ✓ 01일 (목) : 복습, git 활용 기초, api 작성, git 정보 공유

## 할일 목록

이름	☑ 22일(화)	☑ 23일(수)	☑ 24일(목)	☑ 25일(금)	☑ 28일(월)	☑ 29일(화)	☑ 30일(수)	☑ 01일(목)
오영환	☑	☑	☑	☑	☑	☑	☐	☐
이정기	☑	☑	☑	☐	☐	☐	☐	☐
김호진	☑	☑	☑	☑	☑	☑	☐	☐

이름	<input checked="" type="checkbox"/> 22일(화)	<input checked="" type="checkbox"/> 23일(수)	<input checked="" type="checkbox"/> 24일(목)	<input checked="" type="checkbox"/> 25일(금)	<input checked="" type="checkbox"/> 28일(월)	<input checked="" type="checkbox"/> 29일(화)	<input checked="" type="checkbox"/> 30일(수)	<input checked="" type="checkbox"/> 01일(목)
박경진	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
배진영	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

▼ Git 셋팅

▼ 미니프로젝트

▼ Starting Assignments

1. 프로젝트 제목/간단 설명

<제목>

- 생존신고월드

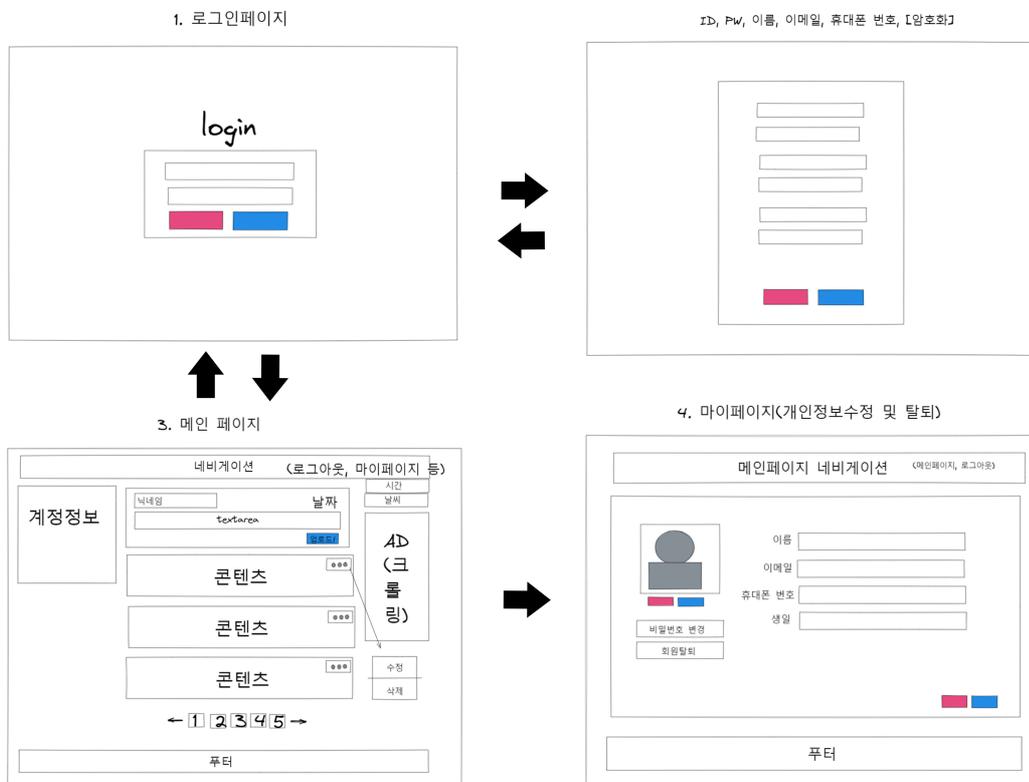
<설명>

- 팀원들이 알고리즘 코딩하다 죽지 않았는지 확인하는 사이트를 만들어 봅시다!

2. 기술 스택

역할	사용 툴
프론트엔드	HTML, CSS, Bootstrap
백엔드	Flask (Python), Ajax (Javascript)
협업툴	Github, Sourcetree, Notion
DB	MySQL

3. 와이어 프레임



시간 가능할 시 추가할 기능:

- 다크모드
- 크롤링
- 

#### 4. 개발해야 하는 기능들

### API 명세

기능	Method	URI	Request	Response	설명
<b>메인 페이지</b>					
게시글 조회	GET	/api/feed		feed 개체: {'uniqueId': uniqueId, 'feedId': feedId, 'feedComment' : feedComment, 'feedDate': feedDate} account 개체: {'uniqueId': uniqueId, 'name': name, 'img', img}	feed 개체에서 정 보를 불러오고 'uniqueId' 속성으 로 account 개체 랑 left join해서 'name'과 'img' 속 성 불러옴
게시글 작성	POST	/api/feed/write	{'uniqueId' : uniqueId, 'feedId': feedId, 'feedComment' : feedComment, 'feedDate': feedDate, 'commentPw': commentPw}		새 게시글을 feed 개체에 저장
게시글 수정	PUT	/api/feed/<feedId>/edit	{'commentPw': commentPw}	{'commentPw': commentPw}	feed 개체에서 게 시글 내용 업데이 트
게시글 삭제	DELETE	/api/feed/<feedId>/delete	{'commentPw': commentPw}		feed 개체에서 게 시글 삭제
<b>회원가입 페이지</b>					
회원가입	POST	/api/user	{'uniqueId': uniqueId, 'id': id, 'pw': pw, 'name': name, 'email': email, 'phoneNum': phoneNum, 'img': img}		회원가입 정보를 account 개체에 저장
ID중복확인	POST	/api/user/<id>/check	{'id': id}		account 개체에서 중복된 id인지 확 인
<b>로그인 페이지</b>					
로그인	POST	/api/user	{'id': id, 'pw': pw}		로그인 정보가 맞 는지 account 개 체에서 확인

기능	Method	URI	Request	Response	설명
로그아웃	POST	/api/user	{'id': id, 'pw': pw}		로그아웃 할 경우 로그인 페이지로 이동 및 뒤로 가기로 진입 방지
<b>마이페이지</b>					
회원정보 조회	GET	/api/user/<uniqueId>		{'uniqueId': uniqueId, 'id': id, 'name': name, 'email': email, 'phoneNum': phoneNum, 'img': img}	account 개체에서 회원 정보 불러오기
회원정보 수정	PUT	/api/user/<uniqueId>/edit	{'uniqueId': uniqueId, 'id': id, 'pw': pw, 'name': name, 'email': email, 'phoneNum': phoneNum, 'img': img}	{'uniqueId': uniqueId, 'id': id, 'pw': pw, 'name': name, 'email': email, 'phoneNum': phoneNum, 'img': img}	account 개체에서 회원정보 업데이트
회원 탈퇴	DELETE	/api/user/<uniqueId>/delete	{'pw': pw}		account 개체에서 회원정보 삭제

NodeJS => /api/user/:userId (userId => 1)

flask => /api/user/<userId> (userId => 1)

명세서 작성 참고자료 : <https://github.com/ChickenPaella/Sloot/wiki/API-%EB%AA%85%EC%84%B8%EC%84%9C>

#### ▼ Development

##### 개발진행 일정 및 간단한 회의록

Aa 기능	날짜
<u>프로젝트 기초작업/ 작업순서 확인</u>	@2022년 12월 2일 오전 9:30-오후 6:15

#### 업무 담당자

##### ▼ 담당자

이름	직책	역할	업무
오영환	팀장		
박경진	팀원		
배진영	팀원	백엔드	DB 구축
이정기	팀원	프론트엔드	HTML, CSS 수정
김호진	팀원		

#### 미니프로젝트 구현해야 하는 화면

- 메인페이지 / 로그인했을때 보이는 페이지
- 회원가입페이지
- 마이페이지 -> 프로필 수정

#### 필수 기능 요구사항 체크리스트

▼ 필수 기능 요구사항

• 회원 기능

- 회원 가입, 로그인 구현
- 마이 페이지 프로필 수정 구현
- 비밀번호 DB에 암호화

• DB

- MySQL 정규화 작업
- MySQL Join 쿼리문을 활용 하여 데이터 전달

• CRUD

- 게시물 쓰기 기능 구현
- 게시물 글 보여주기 구현
- 게시물 글 수정 구현
- 게시물 글 삭제 구현

• GIT

- git add / commit / push 활용
- git pull / merge 활용

• 기타 기능

- Logging 데이터 확인 가능
- Pagination 정상 동작
- 세션 관리