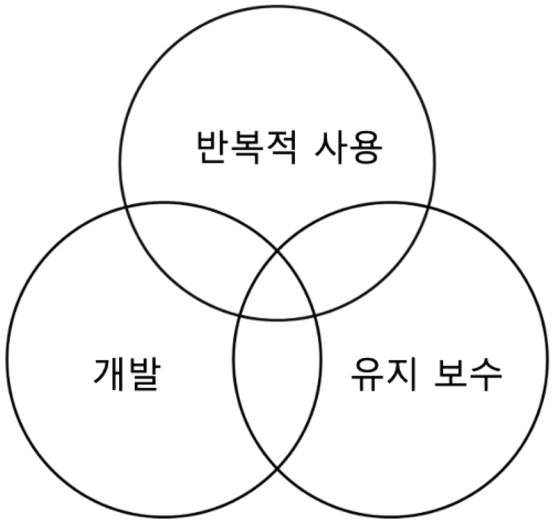
소프트웨어공학개론
6조 최종 발표

김선우 설현원 이찬구 이현민 조재범 한수민

목차

과제 목표

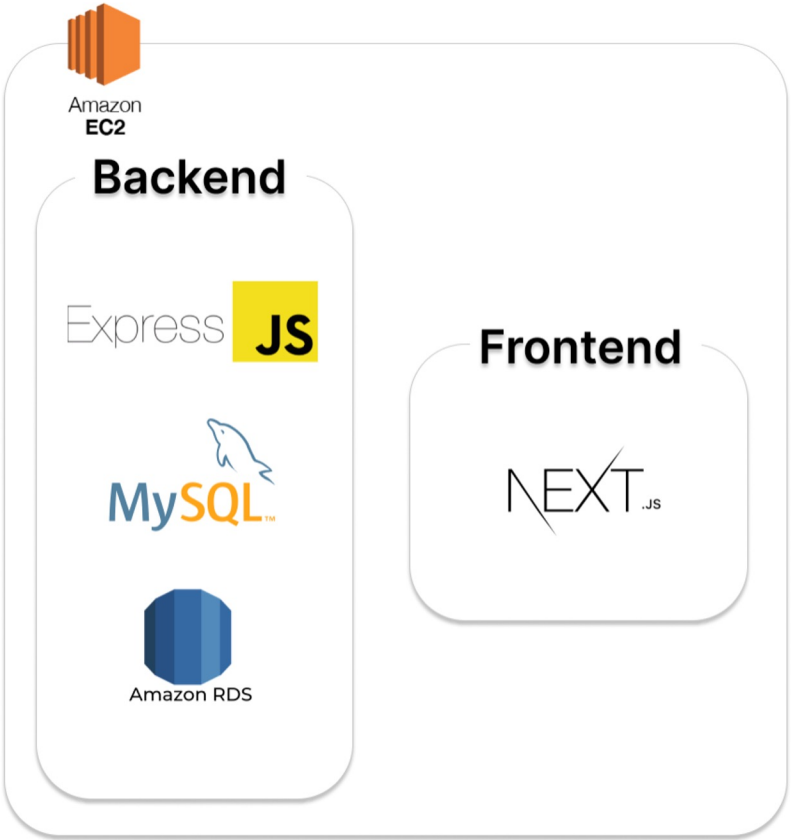
소스코드 탄소 배출량 측정 도구 개발



소프트웨어의 **탄소 배출**

소스코드에 대한
탄소 발생량 측정 필요

프로젝트 기술: Framework



- Version Control: Github
- Document: Notion
- CI/CD: Github action

프로젝트 기술: DB Schema

tb_user

Column	Description
name	사용자 이름

tb_sample

Column	Description
name	예시 이름
figure	예시 수치
description	예시 설명

tb_carbon

Column	Description
user_id	tb_user table 의 id
carbon_emission	탄소 배출량
code	사용자 입력 코드
core_num	서버의 core 개수
cpu_power	서버의 cpu power
memory	서버의 memory
memory_power	서버의 memory power
location	서버의 위치
runtime	코드 실행 시간
PUE	PUE
PSF	PSF
carbon_intensity	국가별 탄소 배출량 강도
provider	클라우드 환경에서의 aws 여부

프로젝트 설명: 서비스 흐름도

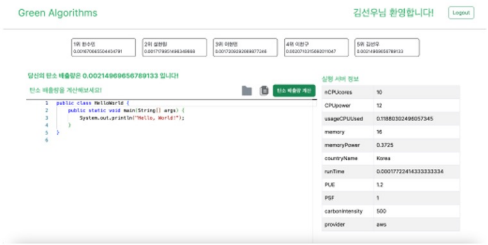
Progammer

소스코드에 따른
탄소 배출량을 알고 싶은 **개발자**

소스코드 제출
탄소 배출량 계산

Frontend

소스코드에 따른 탄소 배출량 계산 **Web**



REST API

Database

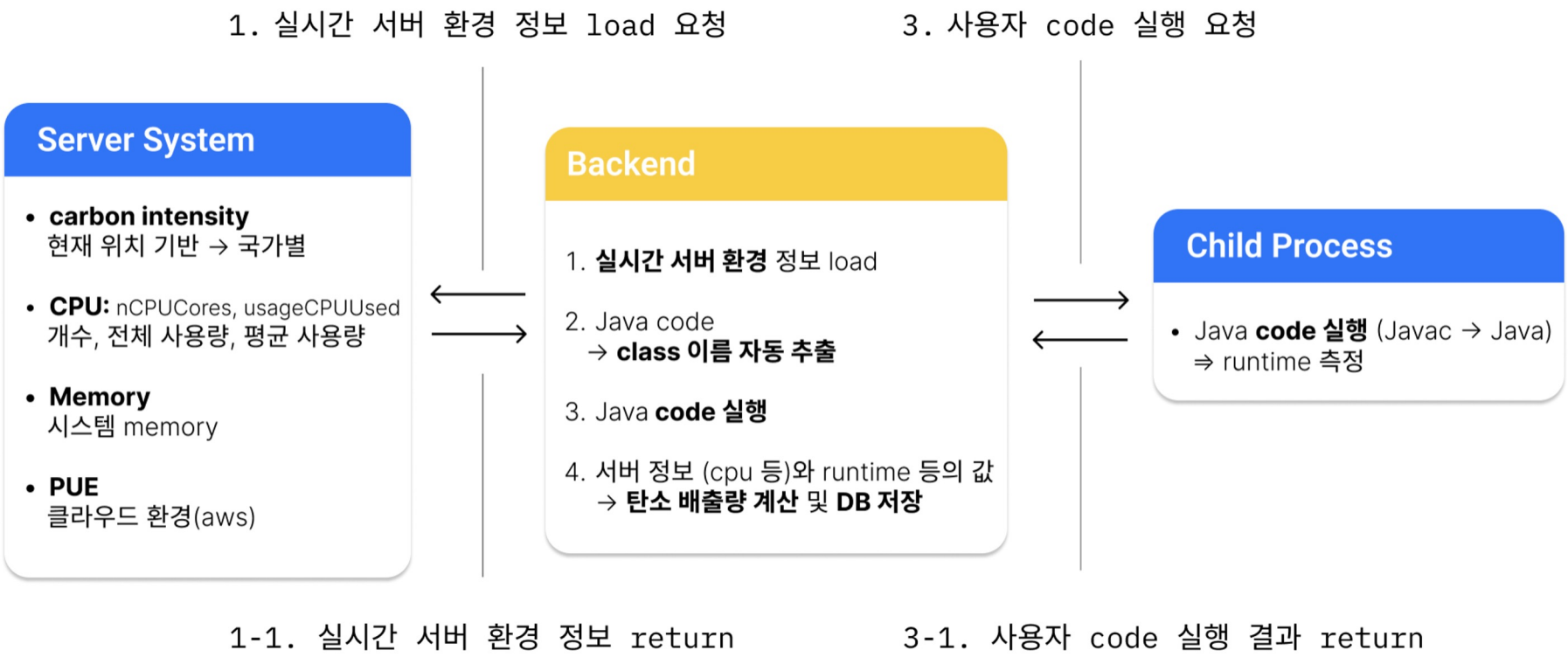
탄소 배출량 정보 **저장**
(서버 환경, sample 데이터 등)

Backend

Web에 필요한 **API**

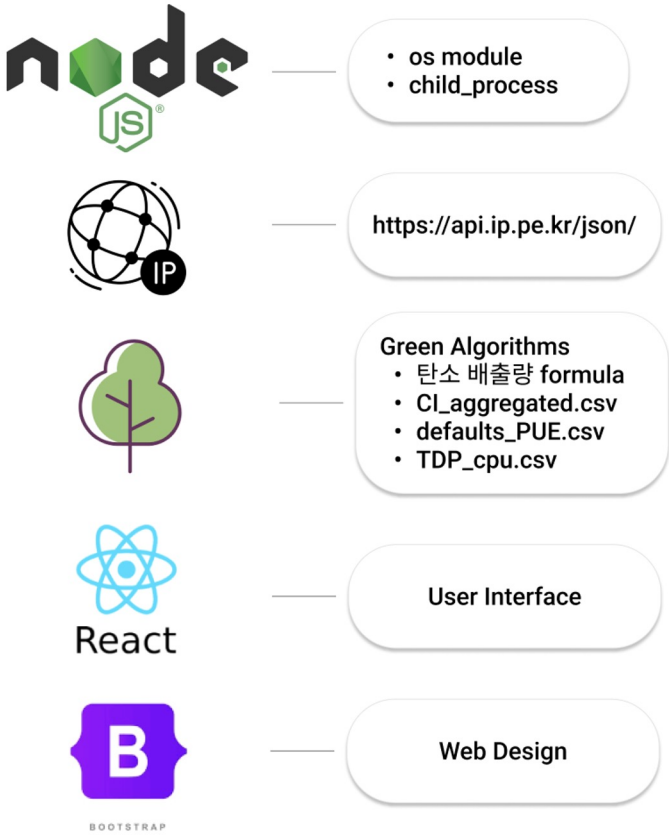
- 서버 환경 **load**
- 탄소 배출량 **계산**
- 탄소 배출량에 따른 **sample data**
- 사용자 **로그인**
- 사용자별 탄소 배출량 **ranking 시스템**

프로젝트 설명: 탄소 배출량 계산



프로젝트 설명: Open Source

- 실행 서버 환경 정보의 실시간 반영
  - os module, api.ip.pe.kr
- 사용자 Java Code 실행
  - child_process
- 웹 디자인
  - Bootstrap
- 유저 인터페이스 개발
  - React



프로젝트 설명: 구현 결과물 - 로그인, 랭킹

유저 이름을 이용한 간단한 로그인

프로젝트 설명: 구현 결과물 - 탄소 배출량 계산, 실행 서버

입력된 코드의 탄소 배출량

당신의 탄소 배출량은 0.00214969656789133 입니다!

탄소 배출량을 계산해보세요!

1. 자바 파일 업로드 버튼
2. 코드 복사 버튼
3. 탄소 배출량 계산 버튼


```
1  public class HelloWorld {
2      public static void main(String[] args) {
3          System.out.println("Hello, World!");
4      }
5  }
6
```

코드 입력 칸

탄소 배출량 계산 시 사용된 실행 서버 정보로 유동적으로 변화

실행 서버 정보

nCPUcores	10
CPUpower	12
usageCPUUsed	0.11880302496057345
memory	16
memoryPower	0.3725
countryName	Korea
runTime	0.00017722414333333334
PUE	1.2
PSF	1
carbonIntensity	500
provider	aws

프로젝트 설명: 구현 결과물 - 탄소 배출량 분석

앞서 계산된 탄소 배출량에 따른 활엽수, 산불, GPT, 쓰레기, 로켓, 넷플릭스, 보일러, 사람, 여객기, 화력 발전소에 의해 발생하는 탄소와의 비교

탄소 배출량 분석

- 0.2149696567891 33 년 동안 활엽수가 흡수하는 탄소
- 2023년 캐나다 산불이 배출한 탄소의 0.00214969656789133 배
- GPT-4 가 한번 학습할 때 배출하는 탄소의 0.00214969656789133 배
- 쓰레기를 0.00214969656789133 톤 연소시킬 때 배출되는 탄소
- 로켓을 한 번 쏘아 올릴 때 배출되는 탄소의 0.00214969656789133
- 넷플릭스를 0.00214969656789133 시간 이용할 때 배출되는 탄소
- 건물용 보일러를 0.00214969656789133 시간 이용할 때 배출되는 탄소
- 한국인 한 명이 0.05159271762939192 시간동안 배출하는 탄소
- 여객기의 승객 1명이 0.00214969656789133 km 이동할 때 배출되는 탄소
- 화력발전소가 0.00214969656789133 kWh당 배출하는 탄소

그린화 패턴: 수집 기준

Carbonfootprint = energy needed x carbonintensity

energy needed = runtime x (powerdrawforcores x usage + powerdrawformemory) x PUE x PSF

runtime : 탄소 배출량을 결정하는 요인 중 **코드와 관련된 요소**

-> 그린화 패턴의 판단 기준을 **runtime** 으로 결정

그린화 패턴: 수집 방법

문헌 조사, 커뮤니티 참여, 직접 개발을 통해 그린화 패턴을 수집

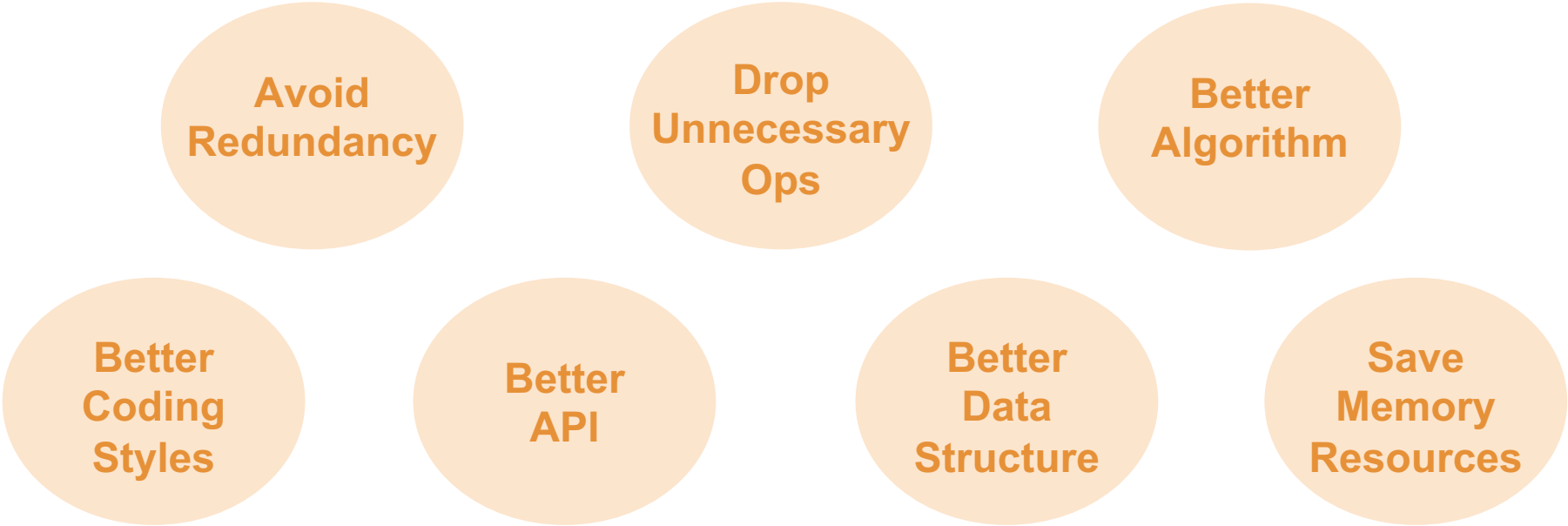
환경 친화적인 코딩 방법에 관한 학술 논문, 보고서, 책 등의 자료를 검색

개발자 커뮤니티와 포럼에 참여하여 코드 분석

패턴으로 사용될 수 있는 여러 방법을 구상 후 직접 개발

그린화 패턴: 수집 패턴 체계화

수집한 패턴을 탄소 배출량이 줄어드는 원리에 따라 카테고리화하여 **패턴 범용성**을 확보



그린화 패턴: 수집한 패턴 검증 자동화

자동화된 검증 툴을 제작하여 쉽게 빠르게 패턴을 추가하고 분석할 수 있게 함

Name	Last commit message	Last commit date
..		
binary_search	[edit] submission	2 weeks ago
pattern_template	[add] change directory, add submission automation fea...	2 weeks ago
patterns	[edit] submission	2 weeks ago
submission	[edit] submission	2 weeks ago
.gitignore	INIT green patterns	last month
checkSubmission.py	[edit] test automations	2 weeks ago
compileAll.py	[add] change directory, add submission automation fea...	2 weeks ago
compressAll.py		2 weeks ago
evaluateAll.py		2 weeks ago
readme.md		2 weeks ago
submitAll.py		2 weeks ago
submitOne.py	[edit] test automations	2 weeks ago
testOnePattern.py	[edit] test automations	2 weeks ago

자동화된
패턴 테스트 툴

compile, runtime 체크하고 jar파일 압축하기!!

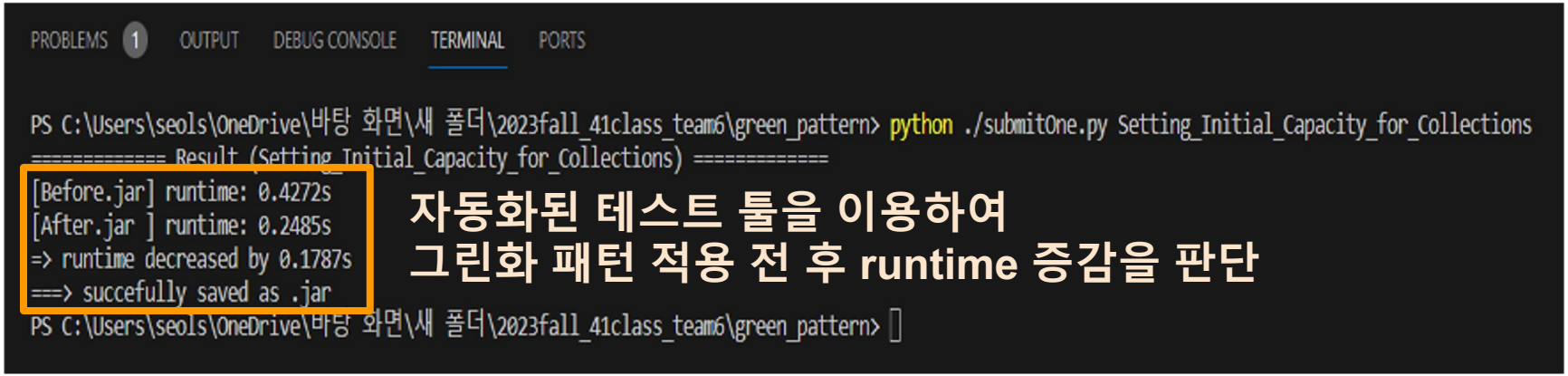
1. pattern 파일을 patterns 폴더에 넣기!
2. \2023fall_41class_team6\green_pattern > python ./submitOne.py {yourPatternName}

==========

다른 파이썬 파일들 (신경안쓰셔도 됩니다!!)

0. testOnePattern.py: patterns/ 폴더에 있는 하나의 패턴 테스트 usage: python ./testOnePattern.py {yourPatternName}
1. compileAll.py : patterns/ 폴더에 있는 모든 패턴 컴파일
2. compressAll.py : patterns/ 폴더에 있는 모든 패턴 압축 (.jar)
3. evaluteAll.py : patterns/ 폴더에 있는 모든 jar 파일 테스트
4. submitAll.py : patterns/ 폴더에 있는 모든 jar 파일 submission/ 폴더로 이동
5. checkSubmission.py: submission/ 폴더에 있는 모든 jar 파일 테스트

그린화 패턴: 수집한 패턴 검증 자동화



Avoid Redundancy	Drop Unnecessary Ops	Better Algorithm	Better Coding Styles	Better API	Better Data Structure	Save Memory Resources	Total
6	5	5	8	8	5	7	**44**

프로젝트 차별점

1. **실시간 서버** 정보 반영
- 다양한 환경에 대한 유연성 및 강건성, 확장성을 통하여 오픈소스로서의 가치 높임

2. 사용자 **랭킹** 시스템
- 다른 사용자와의 비교를 통해 더 적은 탄소배출량을 지닌 코드 작성을 유도

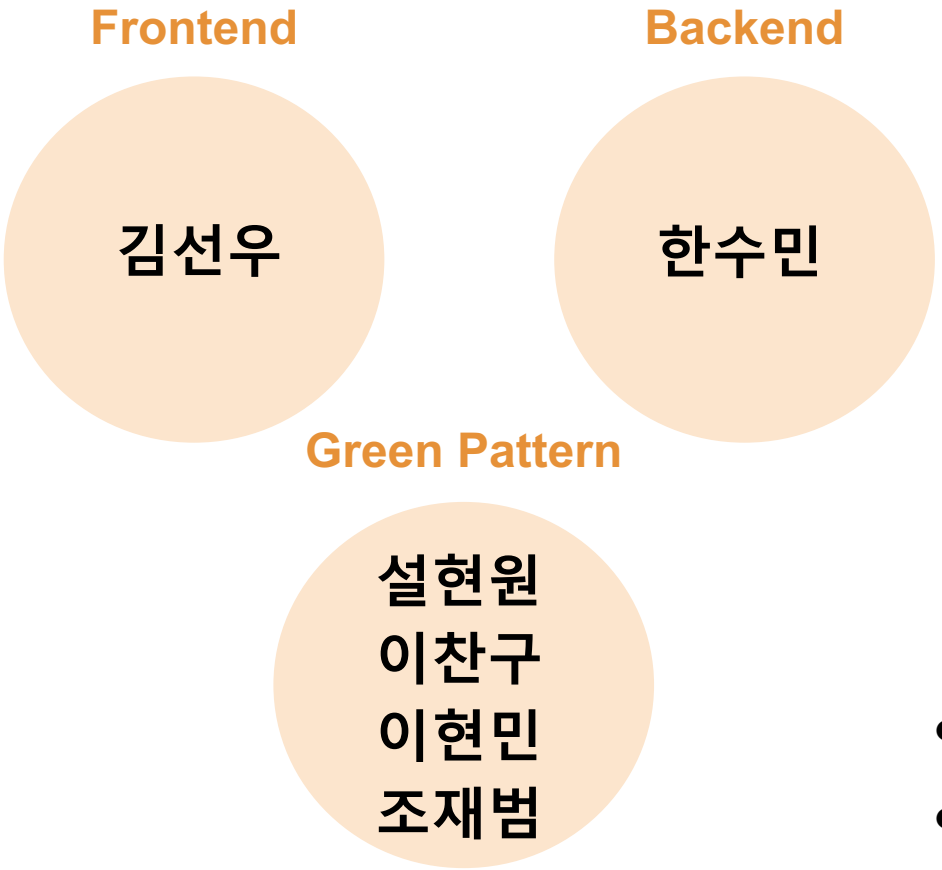
3. 코드 **입력의 다양성**
- 코드 입력창에 직접 작성하는 방식 & 자바 소스 코드 업로드 방식 모두 가능하여 사용자의 편의성을 높임
- 자바 코드 내 class명 자동 추출 -> 사용자의 자바 소스 코드 작성 방식의 제한 X

4. 탄소 배출량 **예시** 정보
- 단순히 탄소 배출량 자체만 정량적으로 제공X -> 실생활 예시 정보를 함께 반환해 사용자의 이해도를 높임

5. 그린화 **패턴의 다양성**
- 총 44개의 다양한 그린화 패턴을 발견하여 보다 많은 경우에 적용이 가능하도록 함

6. **자동화된 테스트** 툴 활용
- 그린화 패턴 검증을 위한 자동 테스트 툴을 사전에 개발하여 빠르고 정확한 패턴 발견이 가능하도록 함

팀 프로젝트 진행 방식: 역할 분담



- Notion: 개발 문서 공유
- Github: 개발 협업

팀 프로젝트 진행 방식: 진행 일정

주차	6	7	8	9	10	11	12	13	14	15
요구 사항 명세서										
디자인 명세서										
API 명세서										
개발										
통합										
테스트										
배포										