<GREEN CO2DING />

SE TEAM 3

Contents

0

Key Points

Key Points

Global Participation

Code Sharing Platform

Algorithm Improvement & Addition

1

Algorithms

1st Alg. Improvements #1

```java
ArrayList<String> arr = new ArrayList<>();
arr.add("a");
arr.add("b");
…
arr.add("f");
arr.add("g");
for(int i=0; i<arr.size(); i++) {
    System.out.println("Hello");
```

Before

```java
ArrayList<String> arr = new ArrayList<>();
arr.add("a");
arr.add("b");
…
arr.add("f");
arr.add("g");
int arrSize = arr.size();
for(int i=0; i<arrSize; i++) {
    System.out.println("Hello");
```

After

1st Alg. Improvements #2

```
ArrayList<String> arr = new ArrayList<>();
arr.add("a");
arr.add("b");
…
arr.add("f");
arr.add("g")
int k = arr.size();
for(int i=0; i< k; i++) {
        System.out.println("Hello");
```

Before

```
ArrayList<String> arr = new ArrayList<>();
arr.add("a");
arr.add("b");
…
arr.add("f");
arr.add("g");
int arrSize = arr.size();
int k = arrSize;
for(int i=0; i< k; i++) {
        System.out.println("Hello");
```

After

1st Alg. Improvements #3

```java
ArrayList<String> arr = new ArrayList<>();
for(int i = 0; i < 1000000; i++){
    arr.add("a");
}
int i = 0;
while( i < arr.size()){
    System.out.println("Hello");
    i++;
```

Before

```java
ArrayList<String> arr = new ArrayList<>();
for(int i = 0; i < 1000000; i++){
    arr.add("a");
}
int i = 0;
int arrSize = arr.size();
while( i < arrSize){
    System.out.println("Hello");
    i++;
```

After

New Algorithm #1

Convert String concatenation to StringBuilder

```
String a = "";
String b = "i'm b";
for(int i = 0; i < 10000; i++){
        a += b;
}
System.out.println(a);
```

Before

```
StringBuilder a = new StringBuilder();
String b = "i'm b";
for(int i = 0; i < 10000; i++){
        a.append(b.toString());
}
System.out.println(a.toString());
```

After

New Algorithm #2

Convert FileReader/Writer to BufferedReader/Writer

```
FileWriter writer = new FileWriter(filePath);
ArrayList<String> arr = new ArrayList<>();
for(int i = 0; i < 1000; i++){
        arr.add("Line"+i+"\n");
}
for(int i = 0; i < arr.size(); i++) {
        writer.write(arr.get(i));
}
```

Before

```
BufferedWriter writer = new
BufferedWriter(new FileWriter(filePath));
ArrayList<String> arr = new ArrayList<>();
for(int i = 0; i < 1000; i++){
        arr.add("Line"+i+"\n");
}
int arrSize = arr.size();
for(int i = 0; i < arrSize; i++) {
        writer.write(arr.get(i));
}
```

After

2

Frontend

Cover Page #1

Title, Total submitted codes #, Total CO2 reduction

GREEN
CO_2DING|

Total Submitted Codes

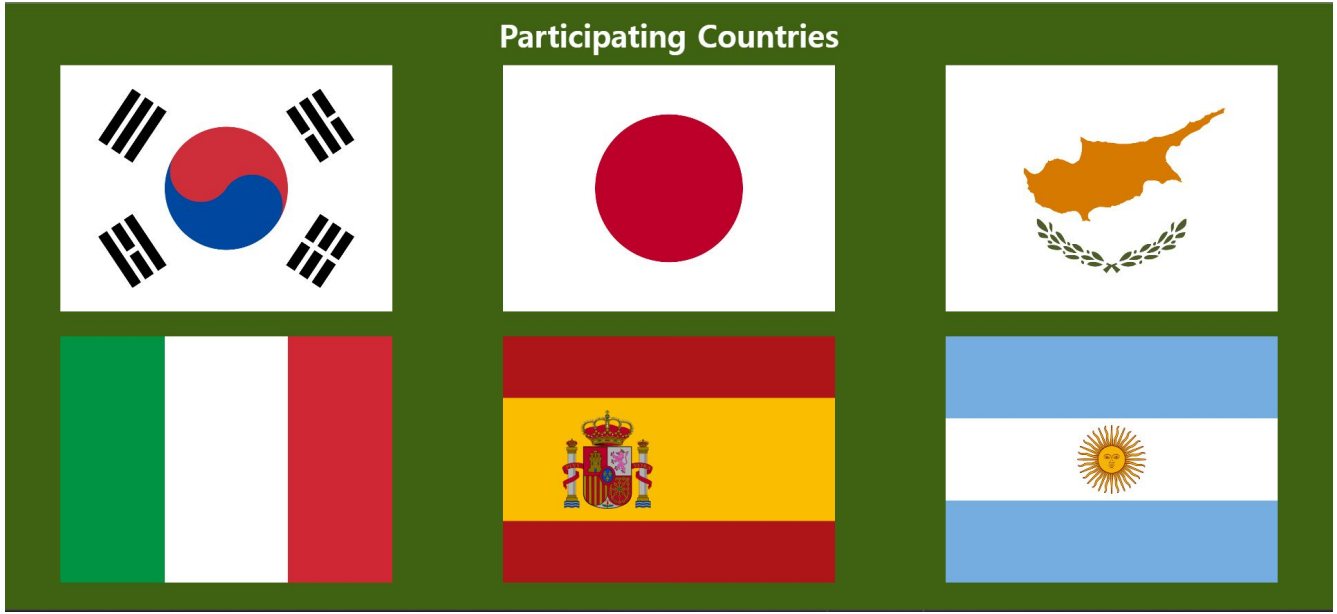
493

Total CO2 Reduction

6,405

Cover Page #2

Participating countries



Main Page #1

Header, Input & Output code editor, Country select

GREEN CO_2DING

Bulletin →

INPUT

```
1  import java.lang.reflect.Array;
2  import java.util.ArrayList;
3
4  public class Temp {
5      public static void main(String[] args) {
6          ArrayList<String> arr = new ArrayList<>(64);
7          for(int i = 0 ; i < 64; i++)
8              arr.add("o");
9          for(int i=0; i<arr.size(); i++) {
10             arr.get(i);
11         }
12     }
13 }
```

OUTPUT

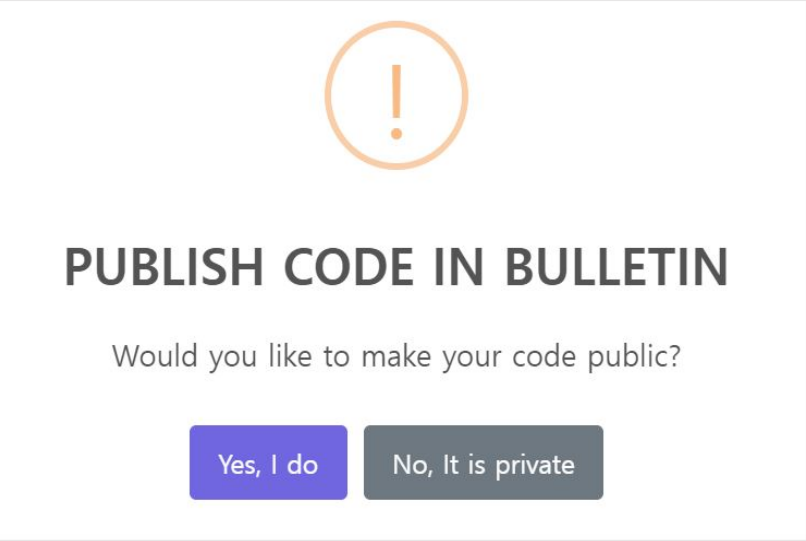
```
1  import java.lang.reflect.Array;
2  import java.util.ArrayList;
3
4  public class Temp {
5      public static void main(String[] args) {
6          ArrayList<String> arr = new ArrayList<>(64);
7          for(int i = 0 ; i < 64; i++)
8              arr.add("o");
9          int arrSize = arr.size();
10         for(int i=0; i<arrSize; i++) {
11             arr.get(i);
12         }
13 }
14 }
15
```

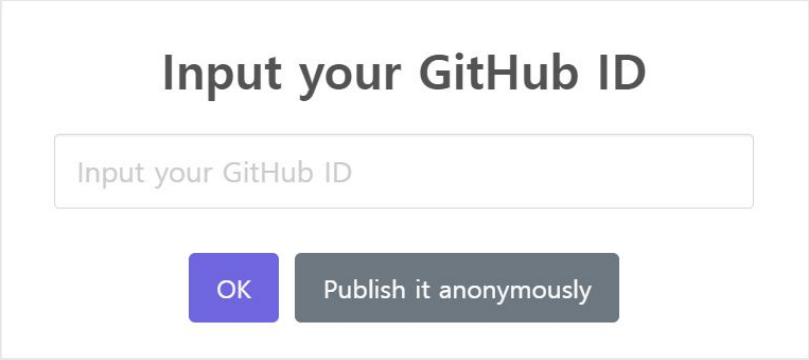
Belgium

submit

Main Page #2

Modal to publish input & output code on bulletin





Main Page #3

Server info, CO_2 emissions amount, World ranking

Server info.

FastAPI	Calculate Green-algorithm-based Carbon Emission Logic
Architecture	x86_64
CPU	Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz
Cores	1
Hypervisor	Xen
L3 Cache	30 MiB
Vulnerabilities	Spec store bypass (Vulnerable), Spectre v1 (Mitigation), Spectre v2 (Mitigation)

Input your code and Be green!

TOP Carbon Saving COUNTRY / Carbon(g)

Korea	6390.6360
Japan	3.3820
Cyprus	3.2234
Italy	3.1414
Spain	2.6502
Argentina	2.0057

Main Page #4

Visualize Carbon Emission

INPUT carbon emission (g)	OUTPUT carbon emission (g)	Effect
60.0	59.2	Carbon Emission **0.8g** is saved. **= Shower 0.1 minute(s)** 

Effect Standard (unit: CO_2 g)

~100

100~300

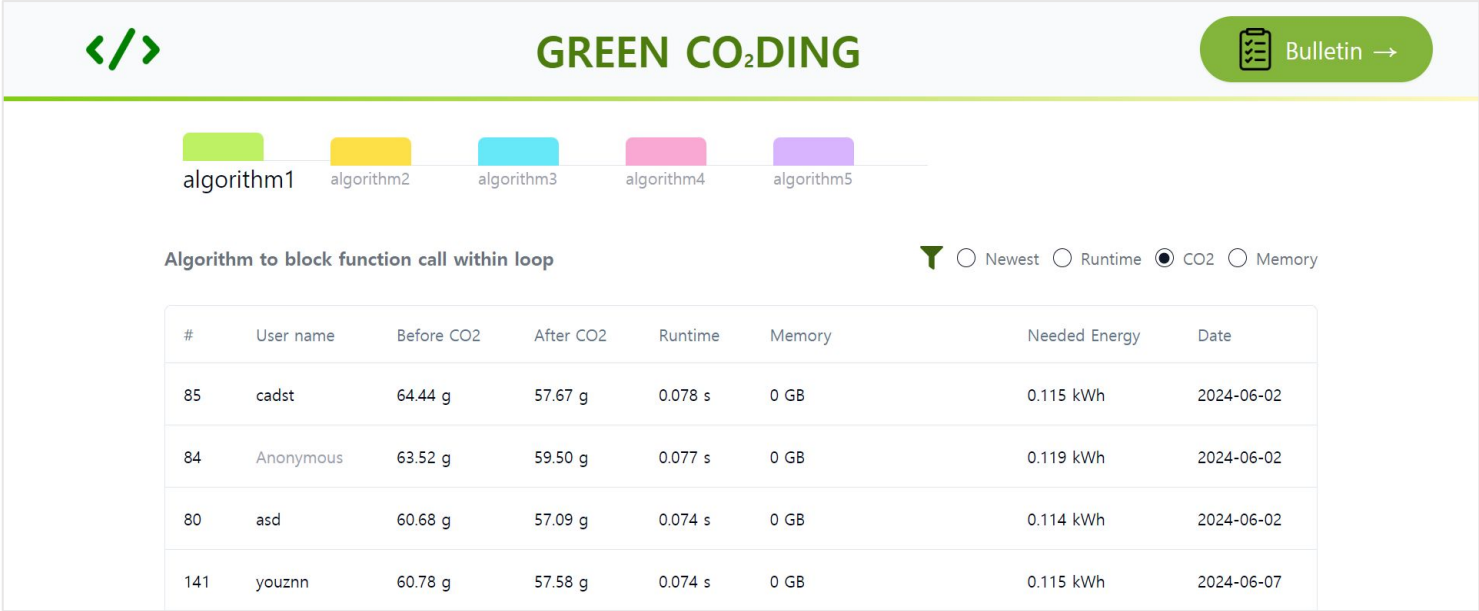
300~1000

1000~2000

2000~

Bulletin Page

Header, Algorithm tap, Radio group, Posts table



#	User name	Before CO2	After CO2	Runtime	Memory	Needed Energy	Date
85	cadst	64.44 g	57.67 g	0.078 s	0 GB	0.115 kWh	2024-06-02
84	Anonymous	63.52 g	59.50 g	0.077 s	0 GB	0.119 kWh	2024-06-02
80	asd	60.68 g	57.09 g	0.074 s	0 GB	0.114 kWh	2024-06-02
141	youznn	60.78 g	57.58 g	0.074 s	0 GB	0.115 kWh	2024-06-07

Detail Page

Header, Input & Output code editor, Contributor info

GREEN CO_2DING

Bulletin →

141

INPUT

```java
import java.lang.reflect.Array;
import java.util.ArrayList;

public class Temp {
    public static void main(String[] args) {
        ArrayList<String> arr = new ArrayList<>(64);
        for(int i = 0 ; i < 64; i++)
            arr.add("o");
        for(int i=0; i<arr.size(); i++) {
            arr.get(i);
        }
    }
}
```

OUTPUT

```java
import java.lang.reflect.Array;
import java.util.ArrayList;

public class Fixed {
    public static void main(String[] args) {
        ArrayList<String> arr = new ArrayList<>(64);
        for(int i = 0 ; i < 64; i++)
            arr.add("o");
        int arrSize = arr.size();

        for(int i=0; i<arrSize; i++) {
            arr.get(i);
        }
    }
}
```

CONTRIBUTOR INFO

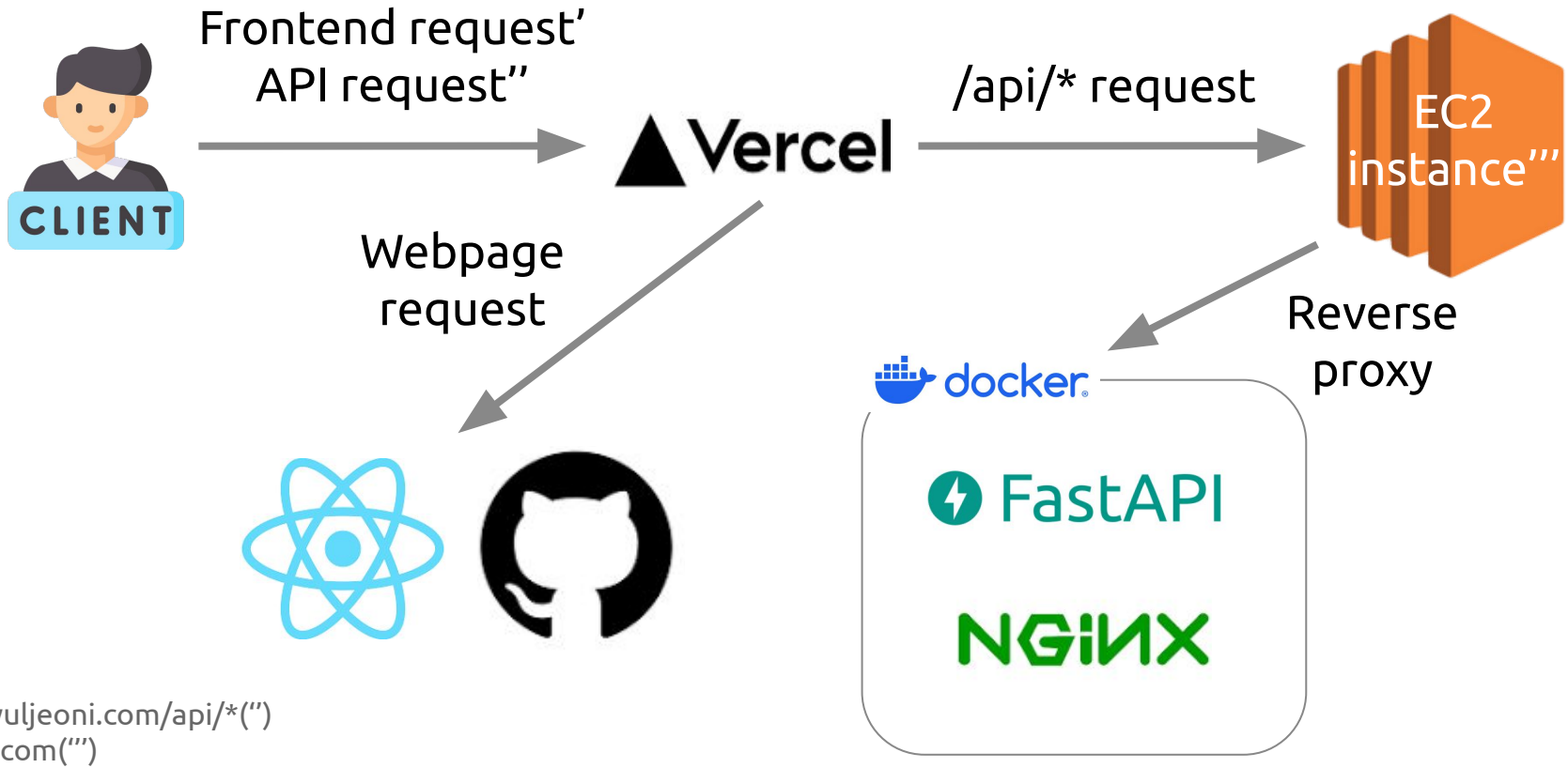
youznn
https://github.com/youznn

runtime	memory	before carbon emission	after carbon emission	needed energy
0.074s	0GB	60.8g	57.6g	0.115KWh

3

Backend & Deployment

System Structure



https://yuljeoni.com/api/*('')
yuljeoni.com(''')

API Specification

GET	/ Test	
GET	/api/default Cover	CO_2 reduction by country, Total submitted codes #
POST	/api/code Code	CO_2 emissions calculation for code, Code refactoring
POST	/api/sharing Sharing	Input & Output code posting on bulletin
GET	/api/bulletin/{algorithm_type} Bulletin	Posts table with alg. type retrieving
GET	/api/detail/{code_id} Detail	Post detail retrieving

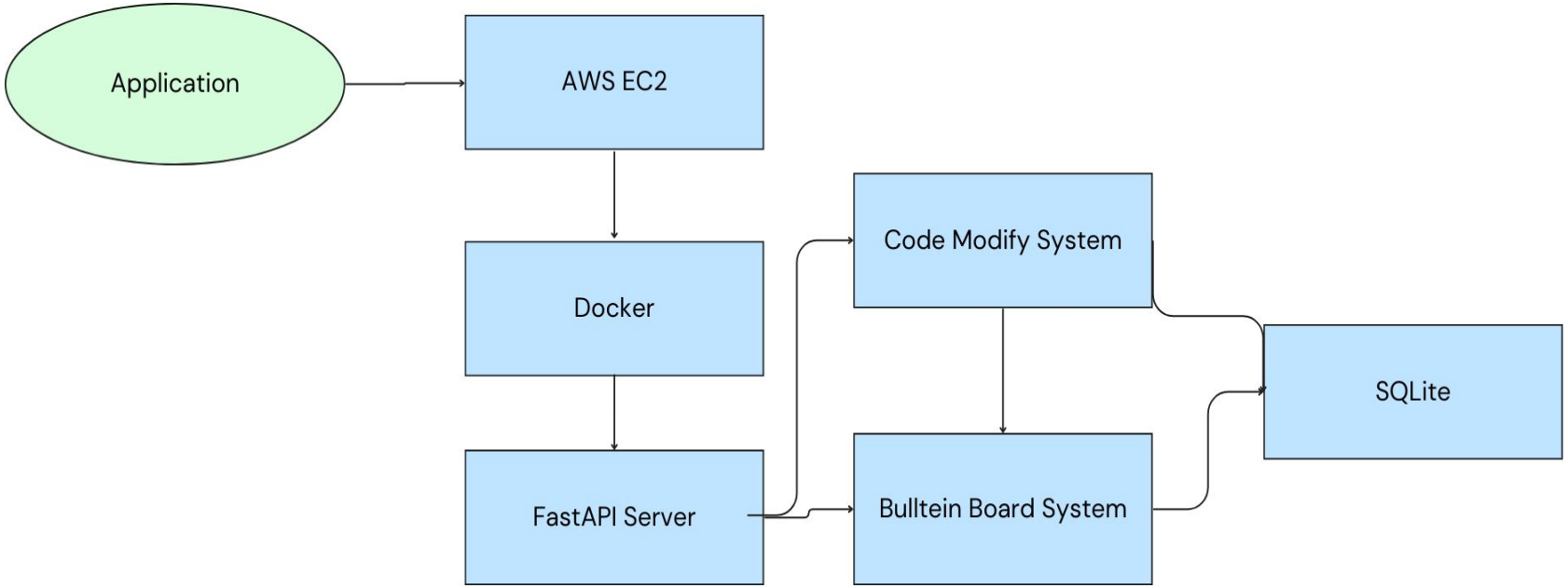
Carbon Emissions

Carbon footprint = E needed x carbon intensity
E needed = runtime x (power draw for cores x usage
+ power draw for memory) x PUE x PSF

carbon intensity: data/v2.1/CI_aggregated.csv
PUE: data/v2.1/default_PUE.csv
PSF: blob/master/app.py#L399

Thank you

Backend Structure



Overall System Structure