

BE GREEN!



**Web service that measures and optimizes
carbon footprint in source code**

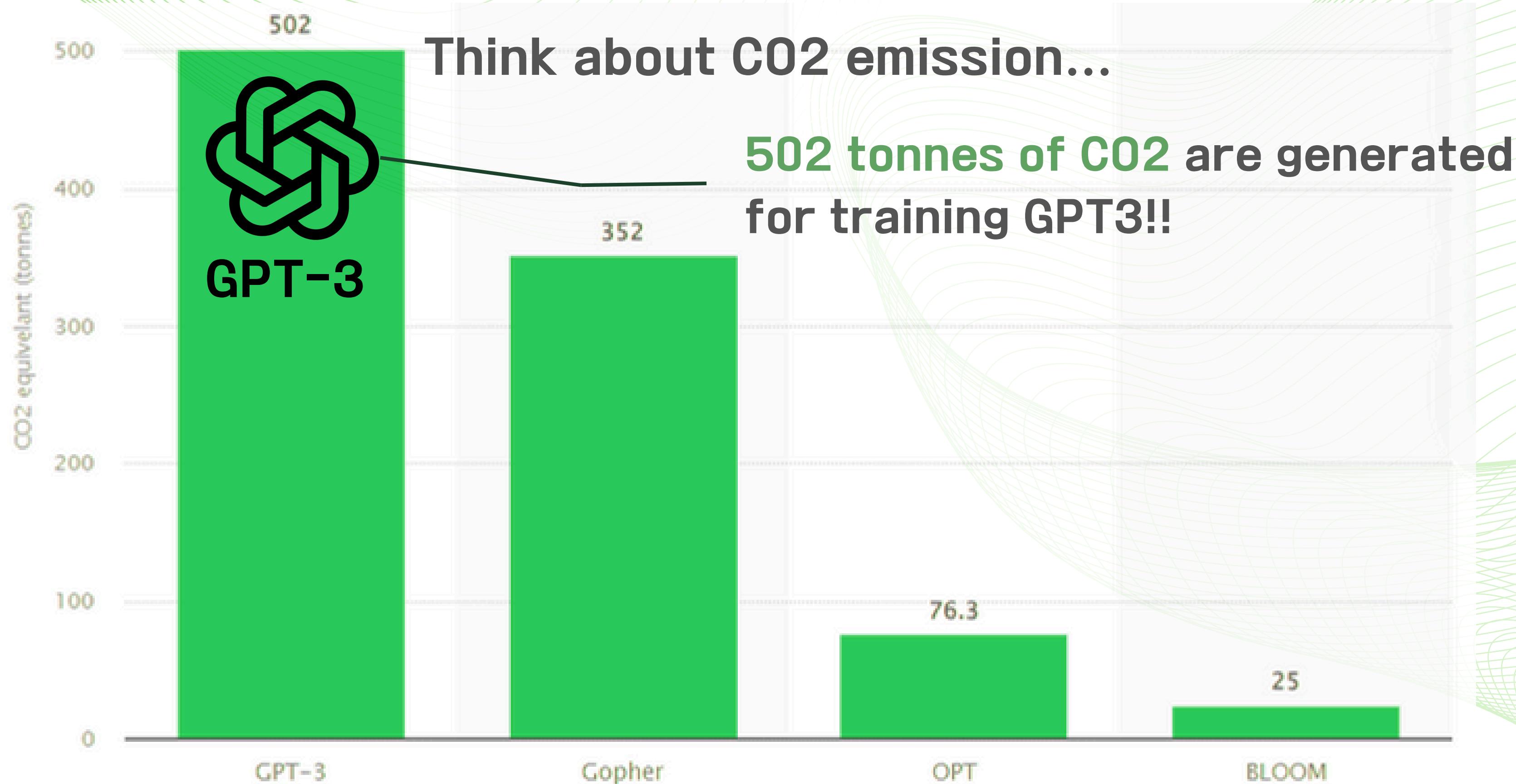
김도현
김민서
김지환
박진호
안현기
이재원

TABLE OF CONTENTS

-  **Introduction & Goals**
-  **Implementation**
 - Front-End
 - Back-End
-  **Team**
-  **Green Patterns & Carbon Emission**
-  **Why BEGREEN ?**
-  **DEMO**

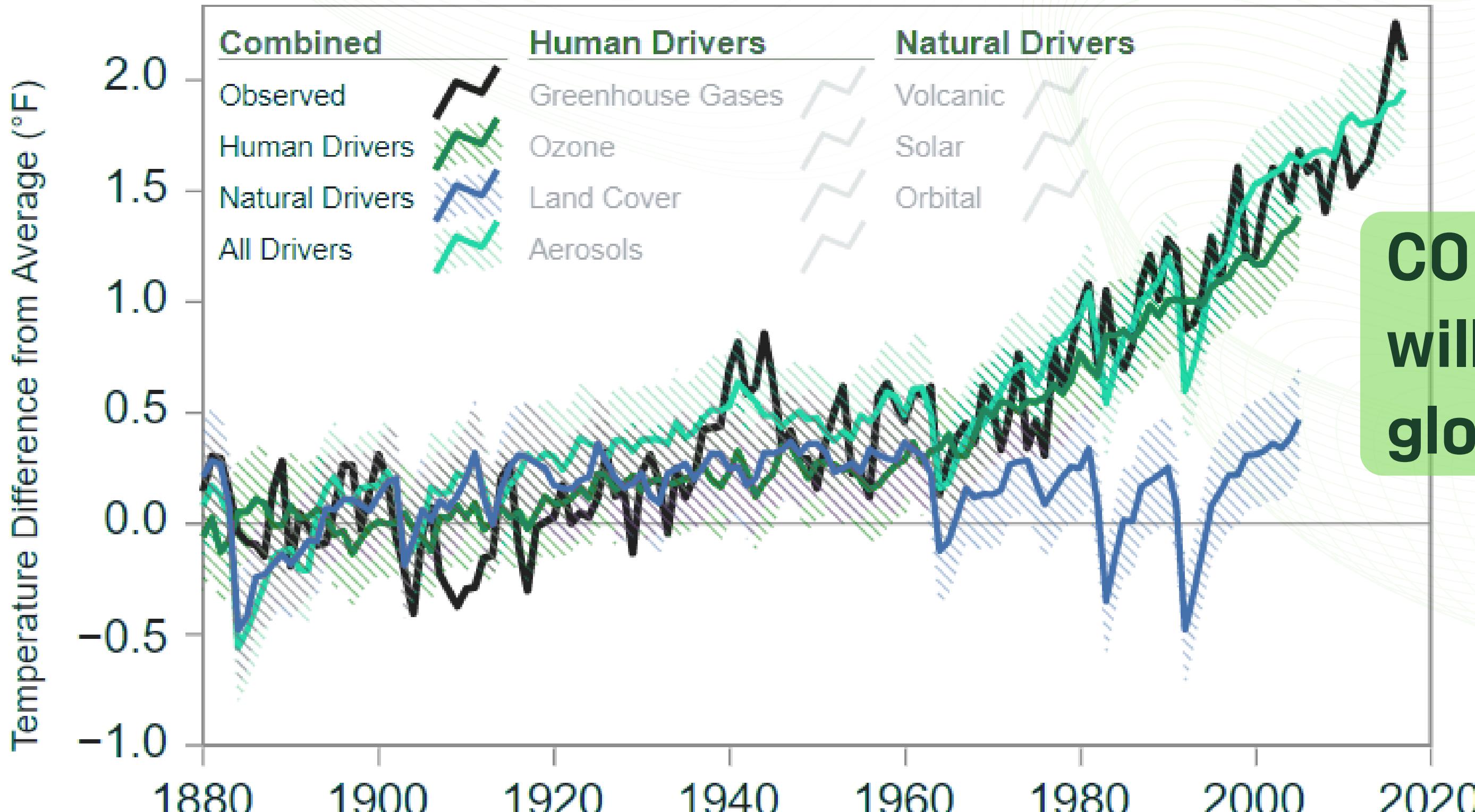


INTRODUCTION



INTRODUCTION

Human and Natural Influences on Global Temperature



CO2 emission...?
will accelerate
global warming!

INTRODUCTION



Need to measure carbon footprint
for code written by developer !!
&
ESG
(Environmental, Social, Governance)



GOALS

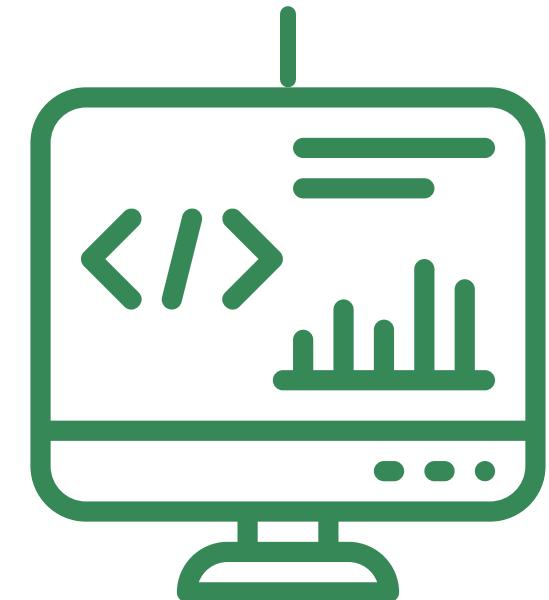
BE GREEN!



Raising
awareness of
carbon emission



Providing
optimization
instructions



Implementation of
carbon footprint measurement code

GOALS

"Web service that estimates carbon footprint of user's input codes, detects carbon waste pattern, provides improved code with greening pattern"



IMPLEMENTATION



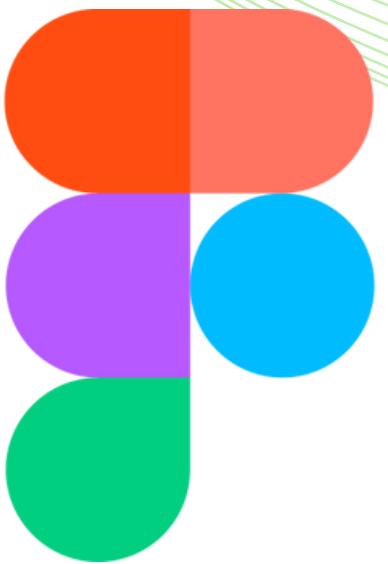
Front-End



Back-End

IMPLEMENTATION

- Front-End



Figma



Front-End

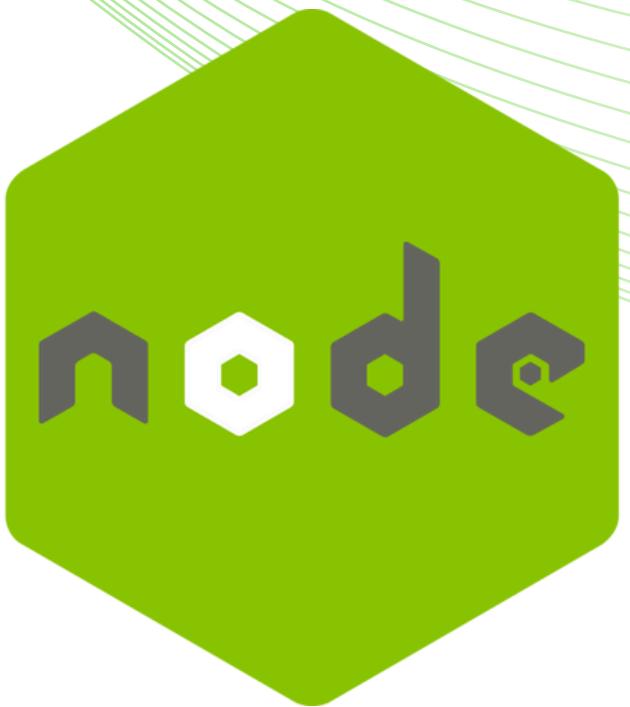


HTML, CSS,
JavaScript

- UI/UX
- More Intuitive, User-friendly!!

IMPLEMENTATION

- Back-End



Node.js



Back-End



Nunjucks

- Server/Database/API
- Implementing code analysis algorithms

TEAM



총괄

Front-End

Back-End

박진호

김도현, 안현지

김민서, 김지환, 이재원

GREEN PATTERNS

1. 중복된 함수 호출 (Duplicate Function Call Pattern)

`arr.size()`와 같이 반복적으로 호출되는 함수를 변수에 저장한다.

이를 통해 불필요하게 호출되는 함수의 횟수를 줄여 메모리 사용량과 실행시간을 줄인다.

2. 다중 If문 사용 (Nested If Pattern)

다중 If문은 `&&`를 통해서 하나의 If문으로 나타낼 수 있다.

이를 통해 컴파일러가 If문을 최적화하는 과정에서 메모리 사용량과 실행시간을 줄인다.

GREEN PATTERNS

3. 중복된 객체 선언 (Duplicate Object Decelaration Pattern)

for문 내에서 객체를 불필요하게 반복적으로 선언하는 패턴을 for문 밖에서 객체를 선언한 뒤 객체의 함수만 반복해서 사용한다.

불필요한 반복적인 객체의 선언을 줄임으로써 메모리 사용량과 객체 선언으로 발생하는 실행시간을 줄인다.

4. 다중 If Else If문 사용(Switch Pattern)

If Else If를 6번 이상 사용하는 경우에 Switch를 사용해준다.

컴파일러의 최적화 과정에서 발생하는 메모리 사용량과 실행시간을 줄인다.

CARBON EMISSION

$$C = t * (n_c * P_c * u_c + \text{memory} * P_m) * \text{PUE} * CI * 0.001$$

References :

Green Algorithms: Quantifying the Carbon Footprint of Computation

```
function calc_CE(t, memory){  
    t = parseFloat(t)  
    memory = parseFloat(memory)  
    uc = 0.3  
    Pc = 50  
    Pm = 5  
    PUE = 1.1  
    cores = 4  
    CI = 0.5  
    E = t * (cores * Pc * uc + memory * Pm) * PUE * 0.001  
    E = parseFloat(E).toFixed(2);  
    return E * CI  
}
```

Example)

$t = 0.059$

$\text{memory} = 28544$

$\rightarrow C = 7.07\text{kg}$

BEGREEN!

Three main competitive advantages of our service

1. Additional green pattern

Three existing green patterns + additional green pattern.

Perform code refactoring through our own algorithms, without using LLM.

→ In line with the purpose of service development.

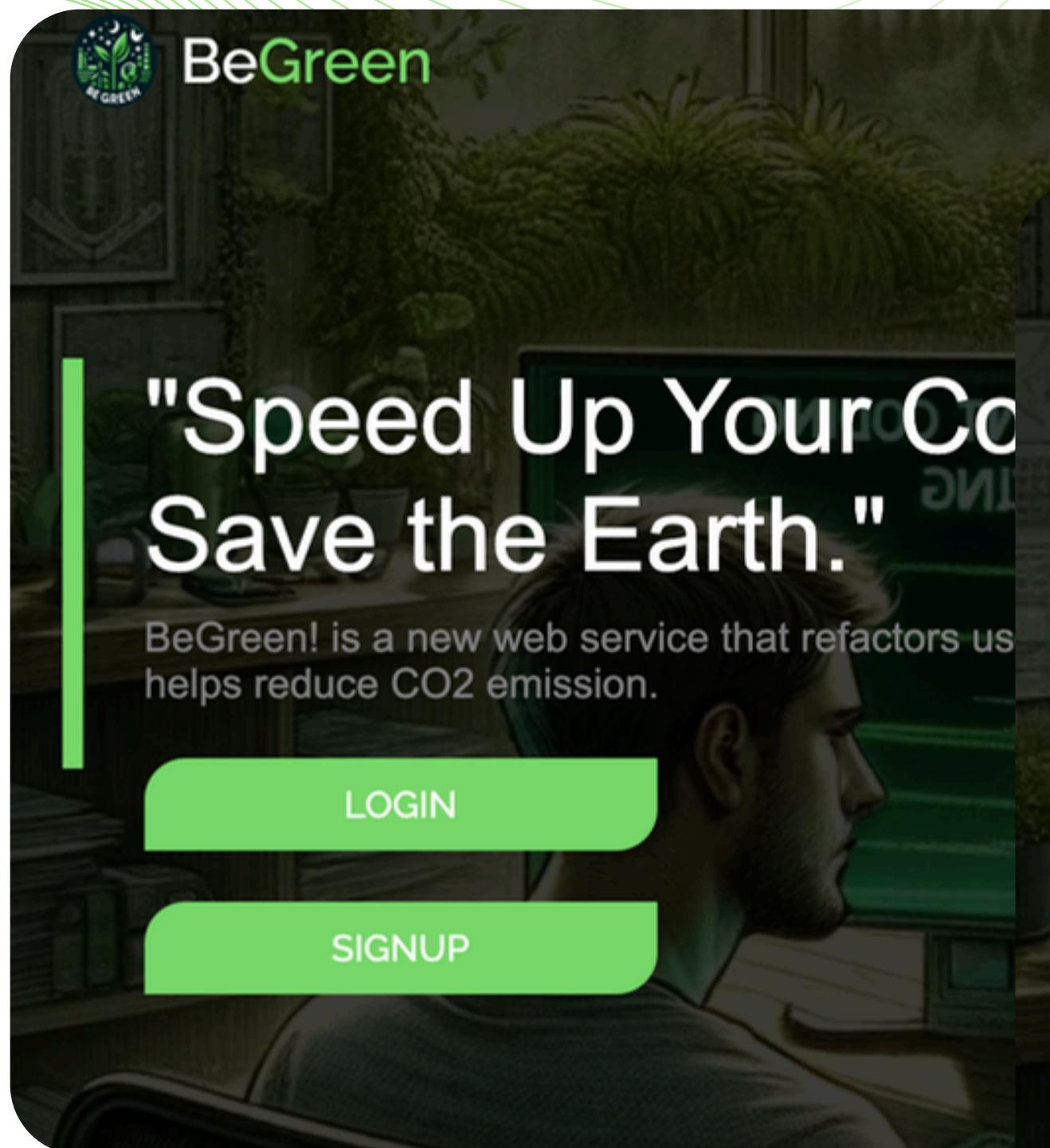
2. Outstanding UI/UX

**Responsive design & clean and intuitive layout & sophisticated color,
User-Centric Interface, designed an intuitive navigation and clear button
placement, allowing users to access service easily.**

3. More Features

**In addition to the code refactoring function,
Personal Info & Calendar & Statistics & Contact...**

DEMO



The dashboard shows a navigation menu on the left with "Overview", "Personal Info", "Input Code", "Calendar", "Statistics", and "Contact". The main area has a welcome message, an overview of the week's goals, and a CO2 reduction summary. A footer section discusses the importance of reducing carbon emissions.

Why Reducing Carbon Emission is Important?

Reducing carbon emission in programming isn't just about adhering to environmental



Thank you !

References :

- [1] <https://www.statista.com/statistics/1384418/co2-emissions-when-training-llm-models/>
- [2] <https://www.epa.gov/climatechange-science/causes-climate-change>
- [3] <https://arxiv.org/abs/2007.07610>