탄소 배출량 계산 웹 서비스 개발

Team 9

목차

Goals

01

CALCULATE

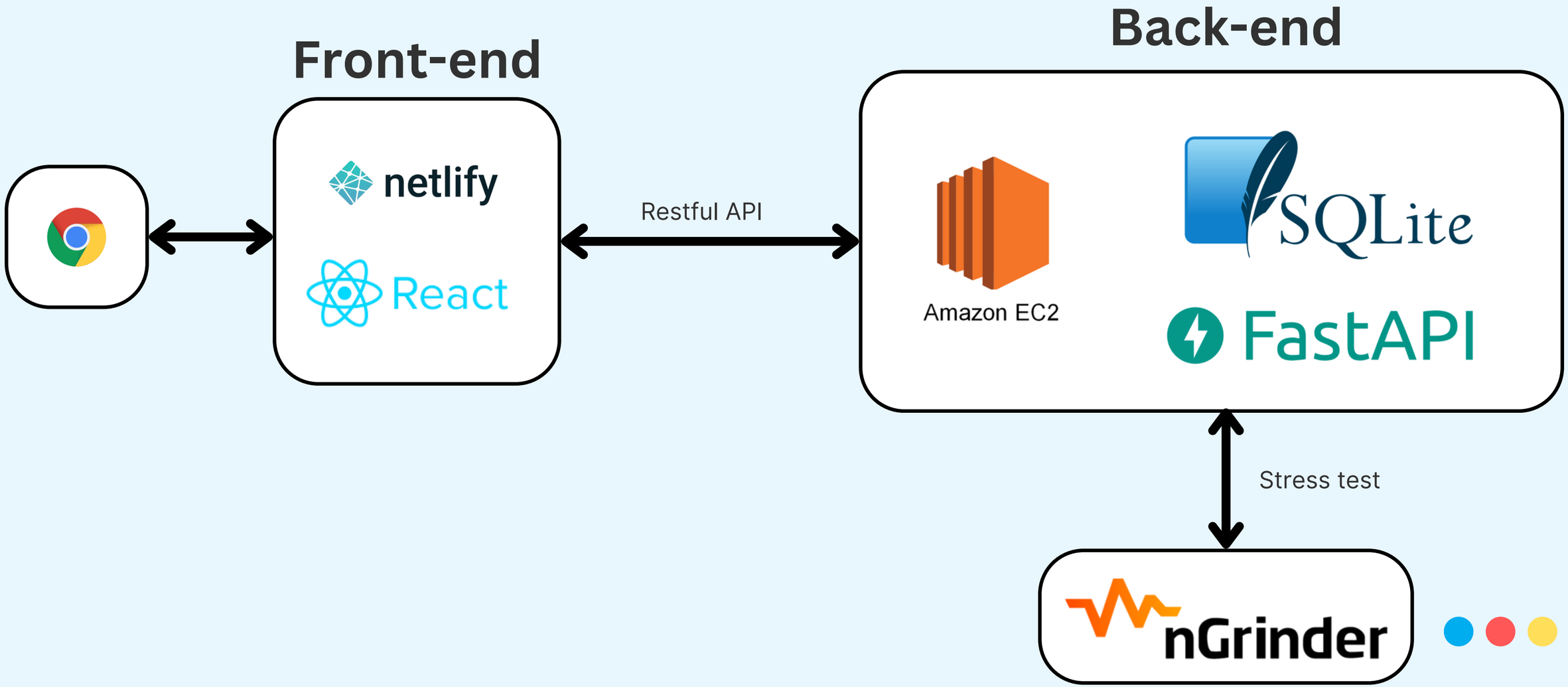
Java 코드의 탄소 배출량을 계산하는
웹 서비스 개발
기존 탄소 배출량 계산 사이트에 없는
코드 컴파일 기능 구현

02

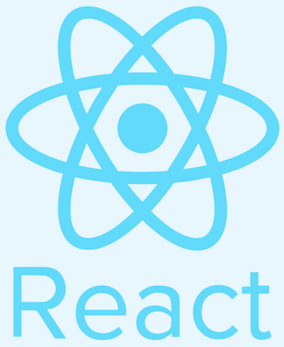
COMPAIR

사용자가 입력한 코드를 기록하여
탄소 배출량을 비교할 수 있는 기능 구현
최대 5개의 코드 비교 가능

Overview


Front-end
netlify
React
Restful API
Back-end
Amazon EC2
SQLite
FastAPI
Stress test
nGrinder

Used Tools



Used Tools

Open sources

React

SQLite

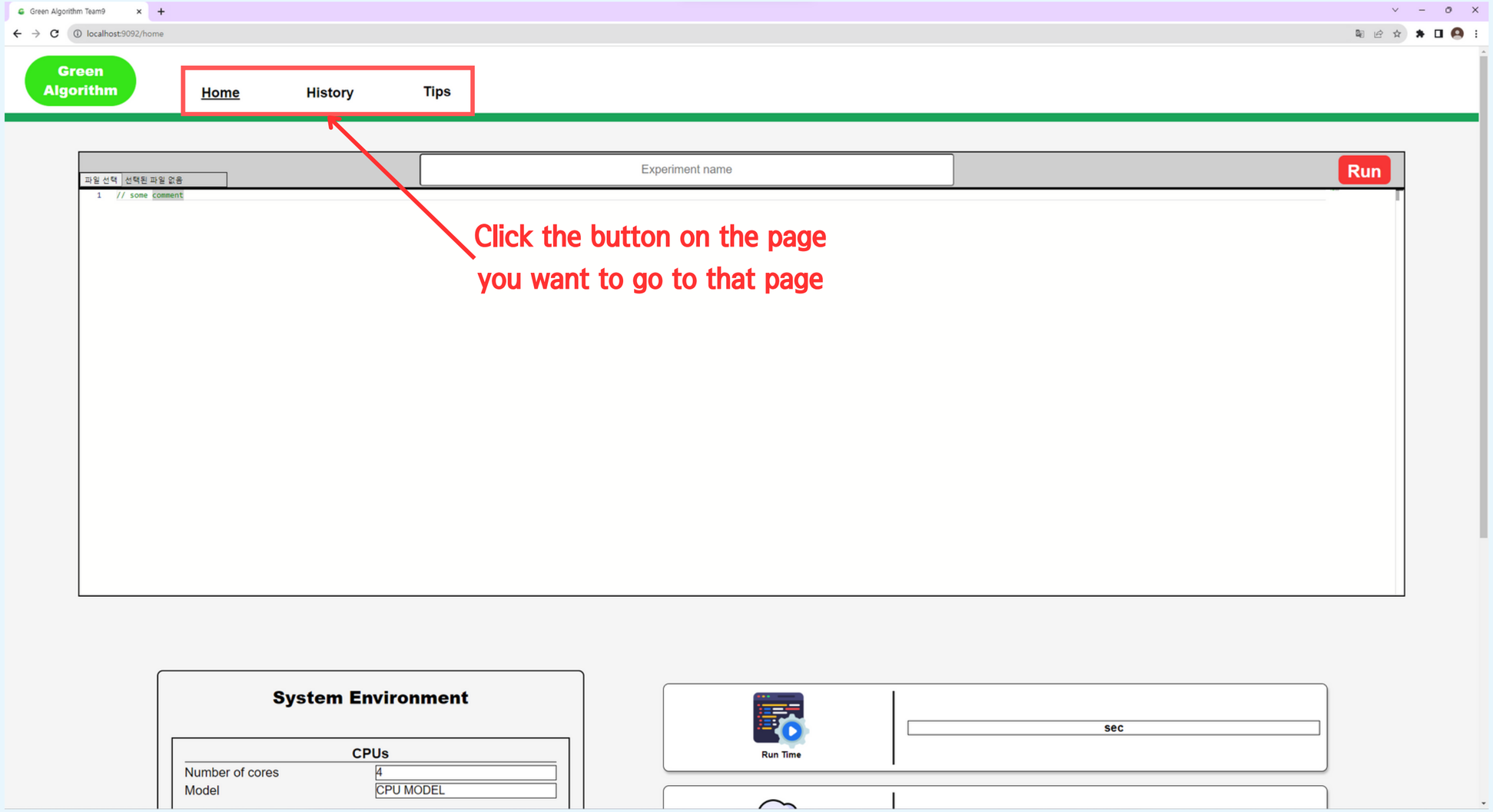
FastAPI

nGrinder

netlify

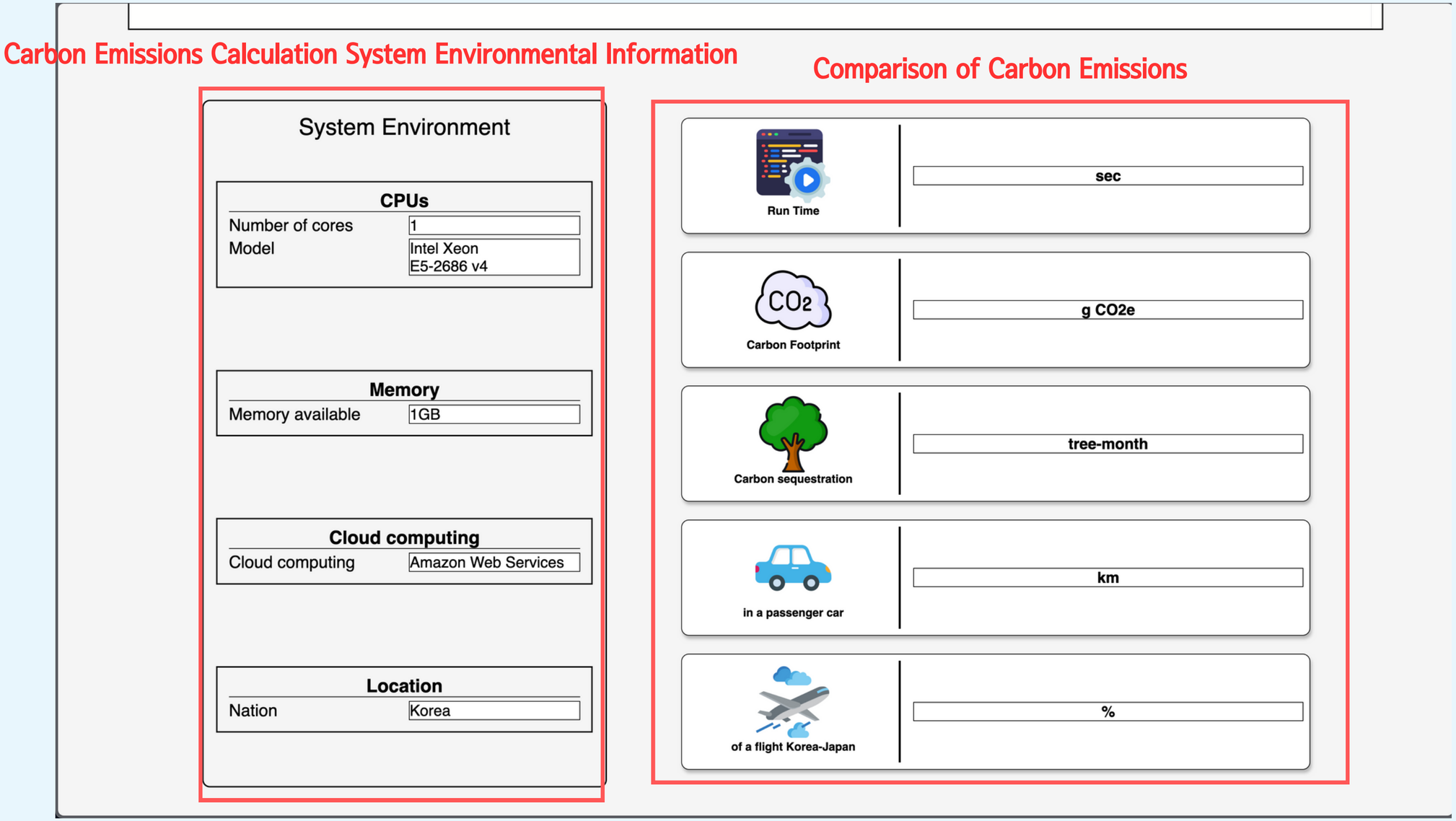
EC2

UI/UX Scenarios



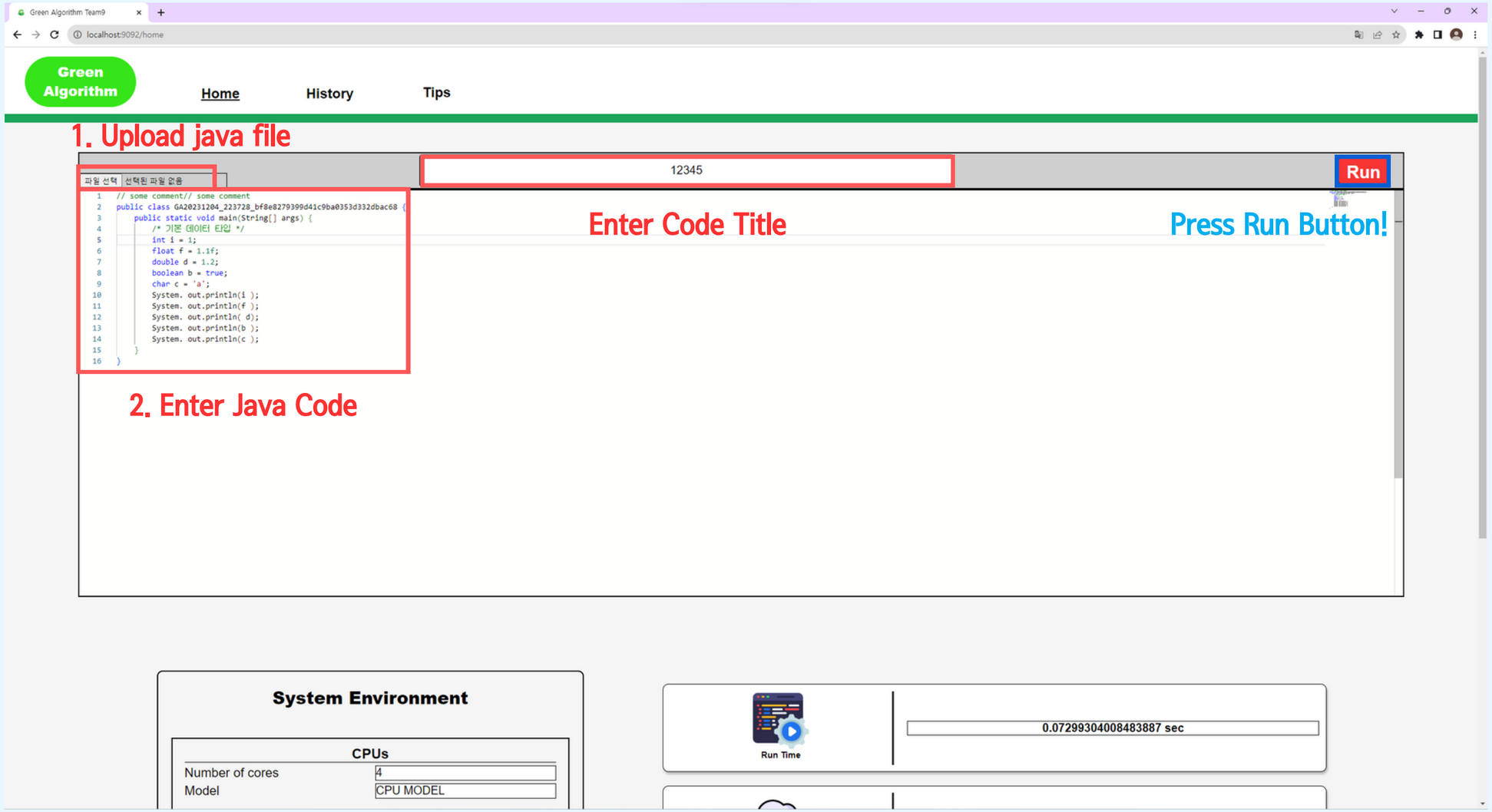
HOME PAGE

UI/UX Scenarios



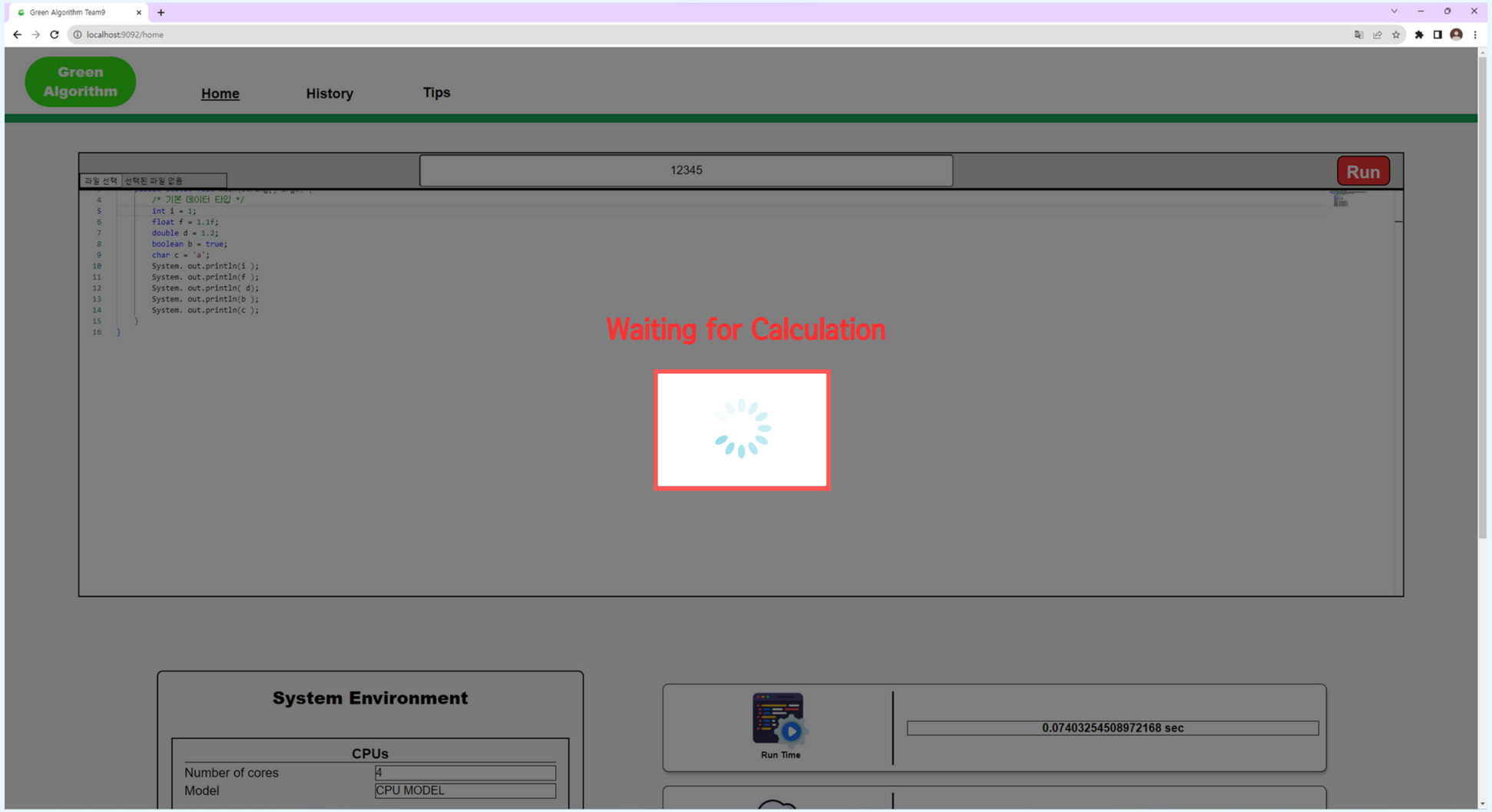
HOME PAGE

UI/UX Scenarios



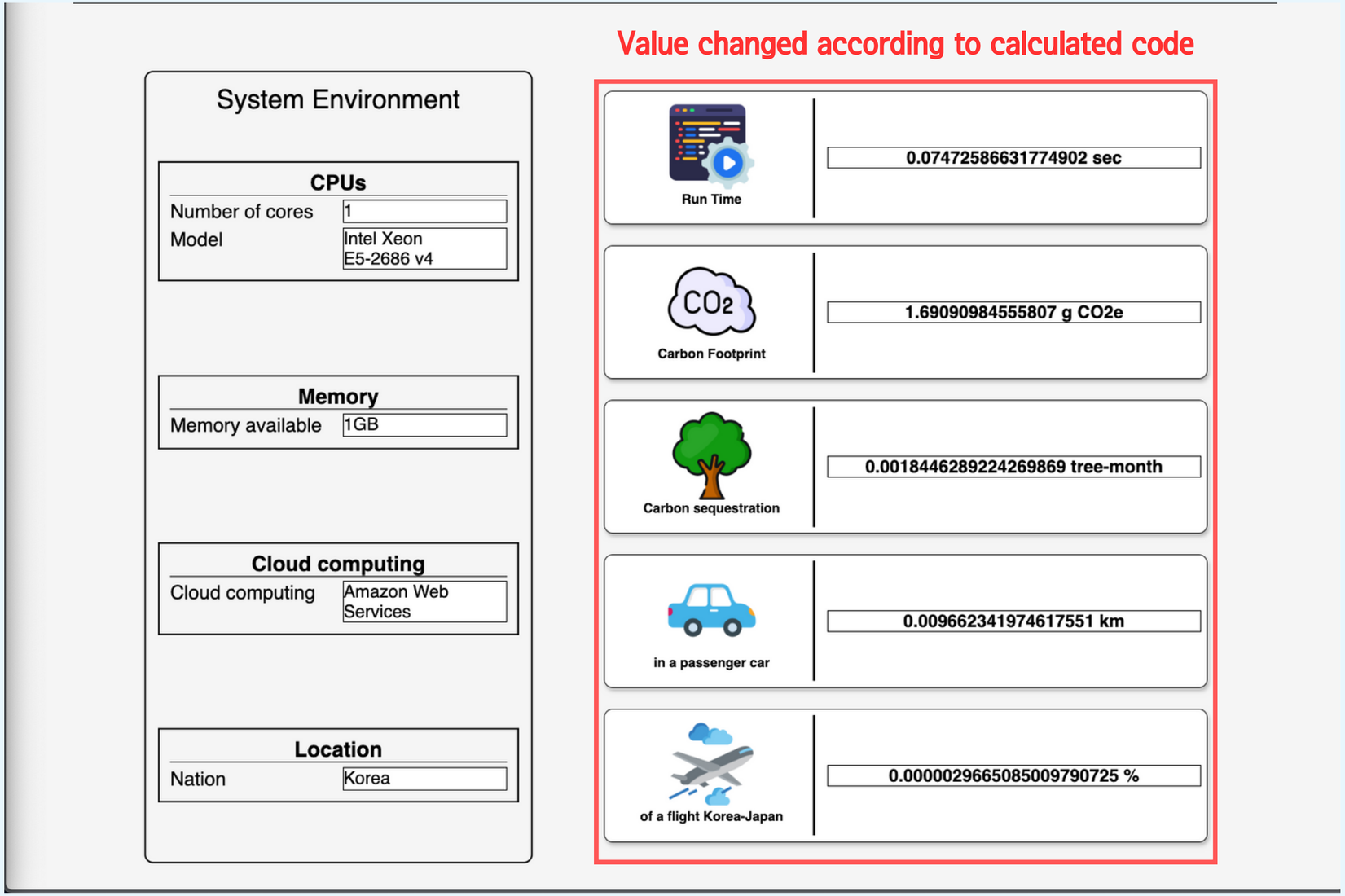
HOME PAGE

UI/UX Scenarios



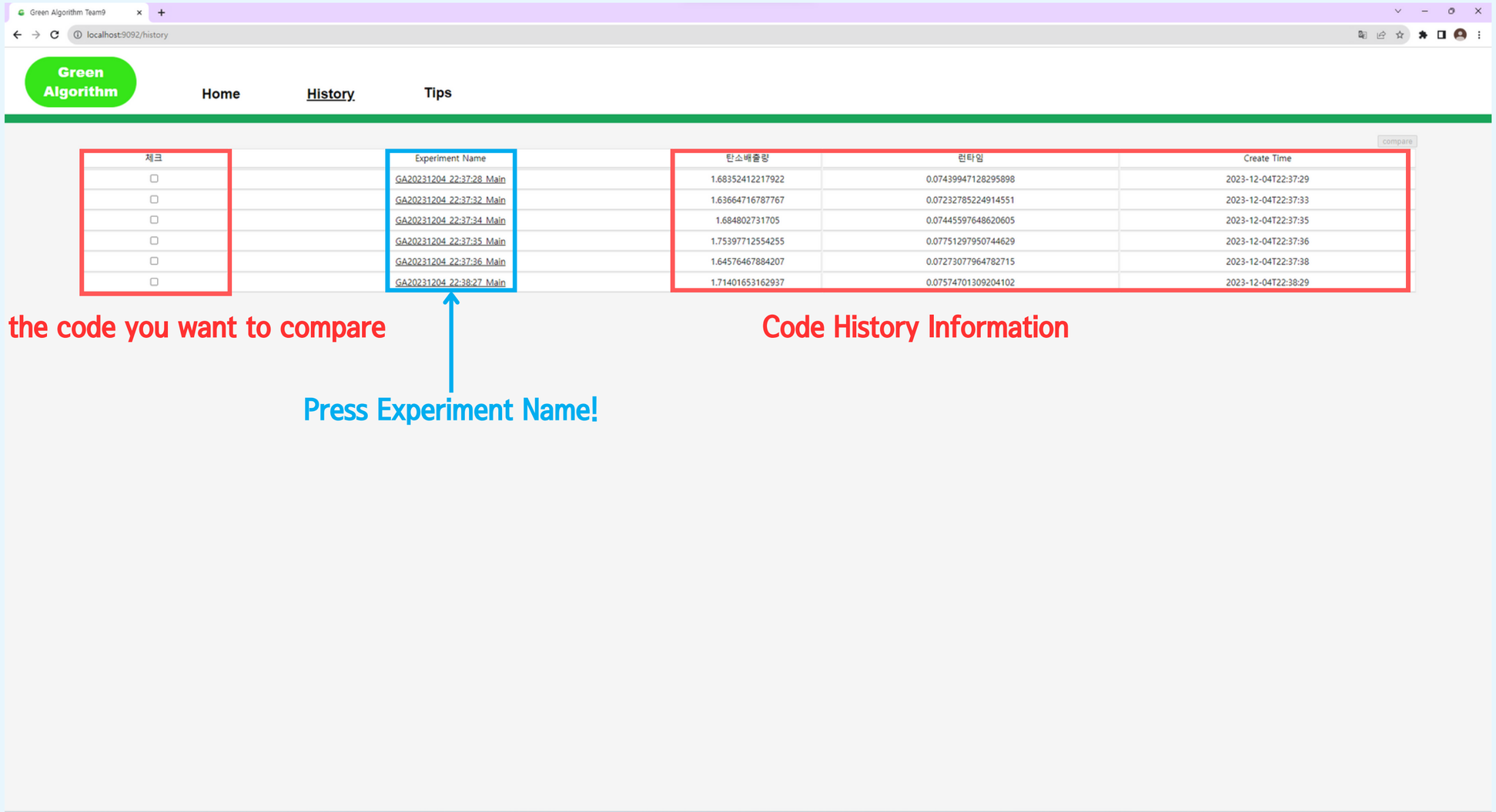
HOME PAGE

UI/UX Scenarios



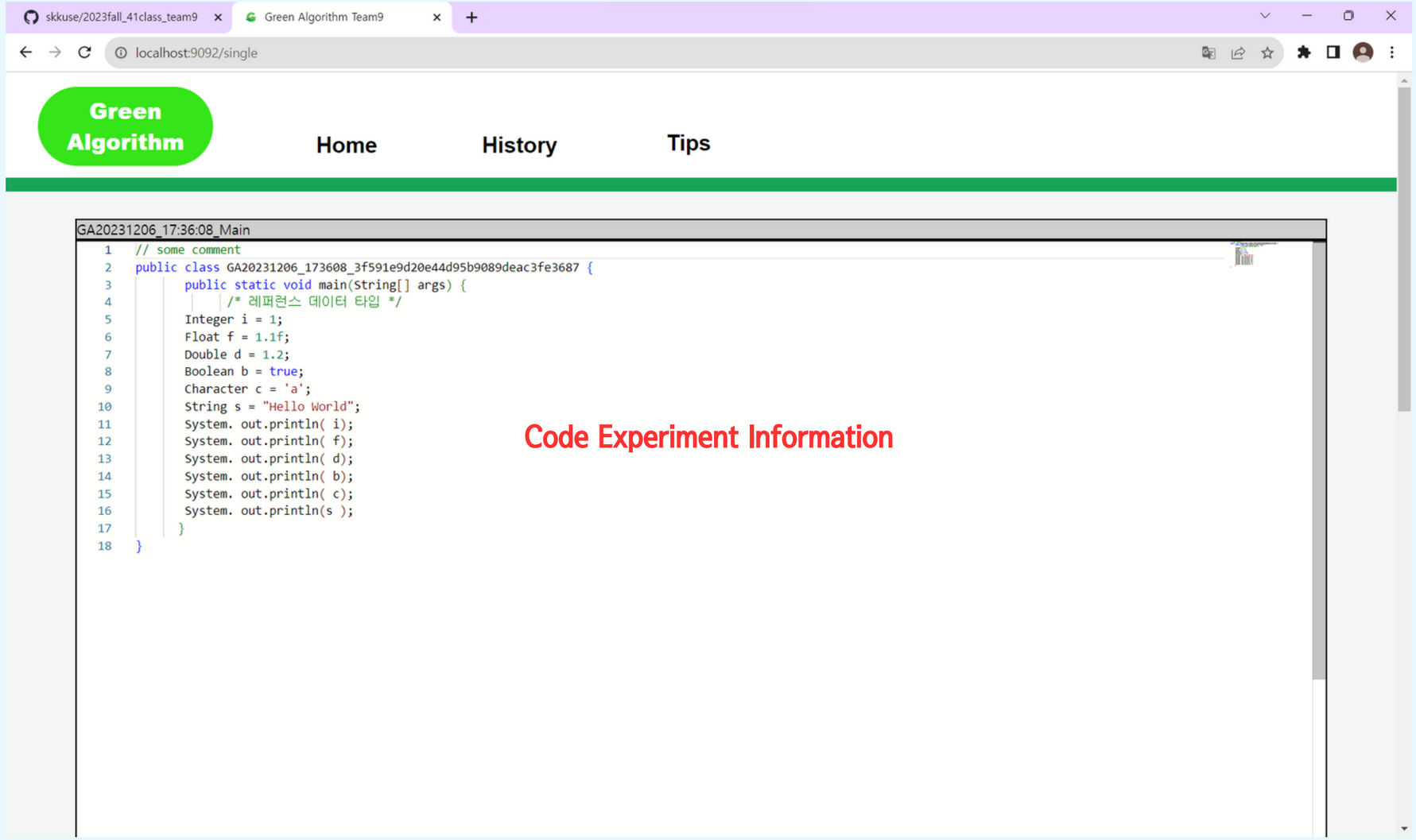
HOME PAGE

UI/UX Scenarios



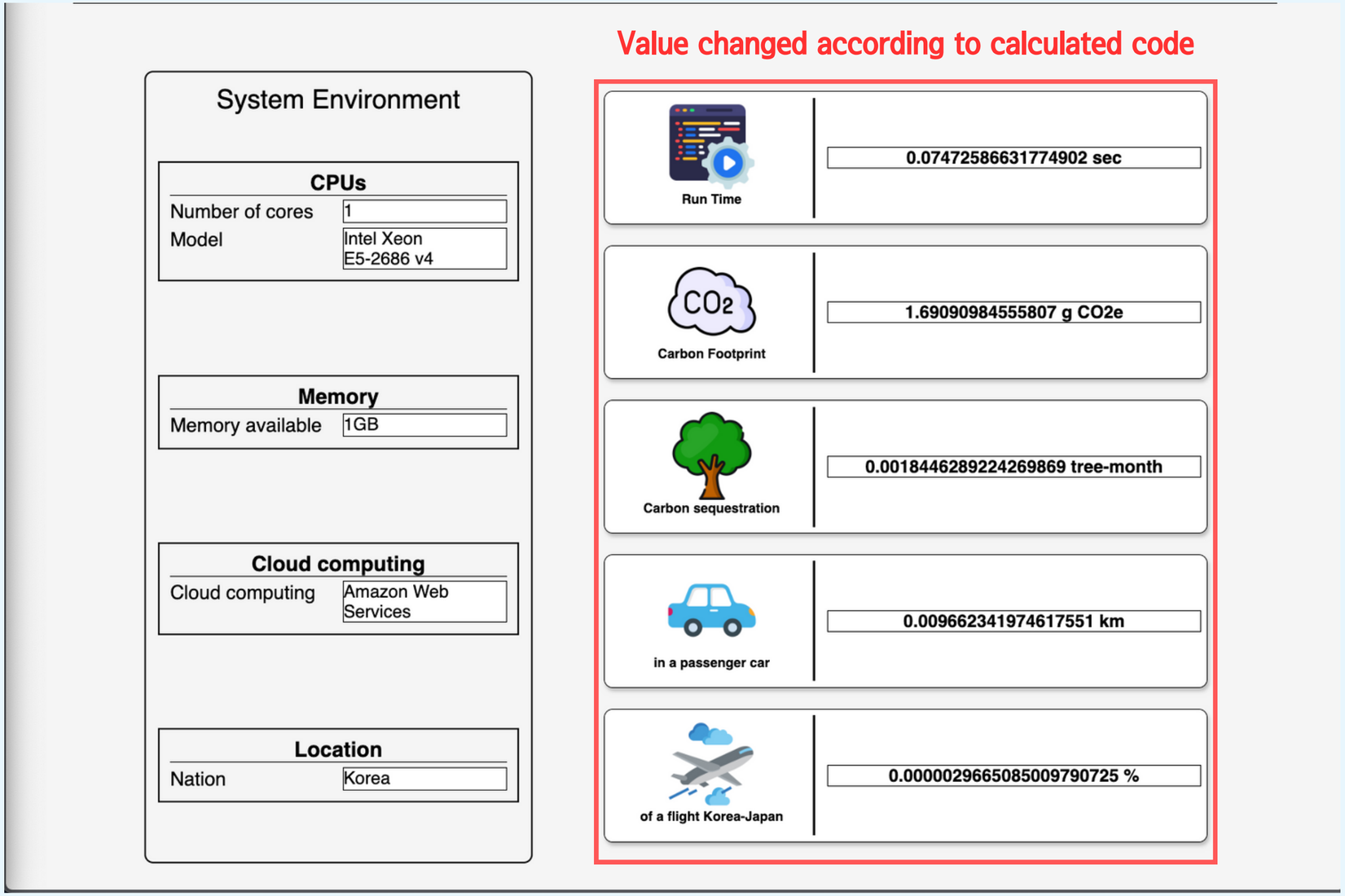
HISTORY PAGE

UI/UX Scenarios



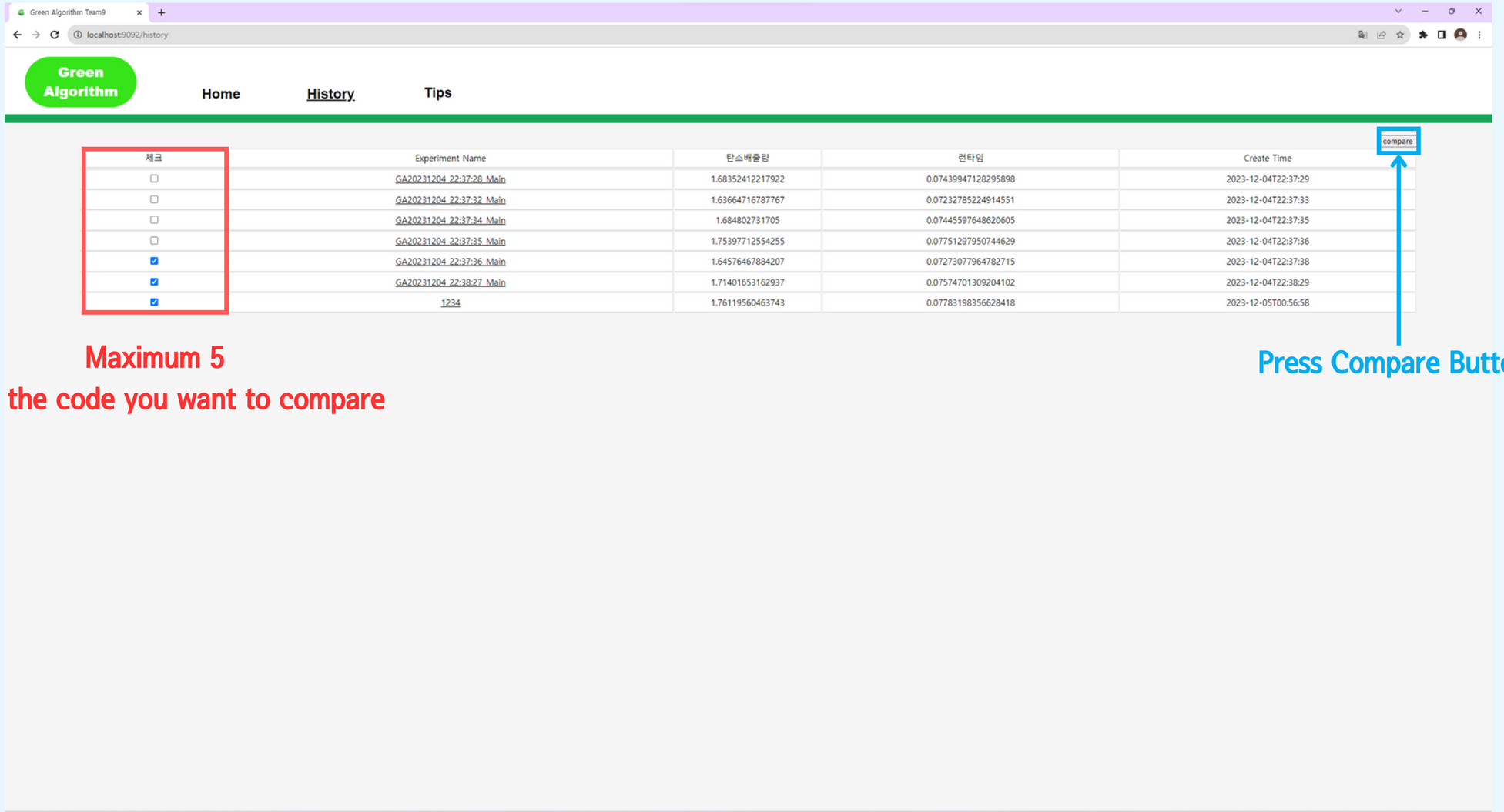
HISTORY PAGE

UI/UX Scenarios



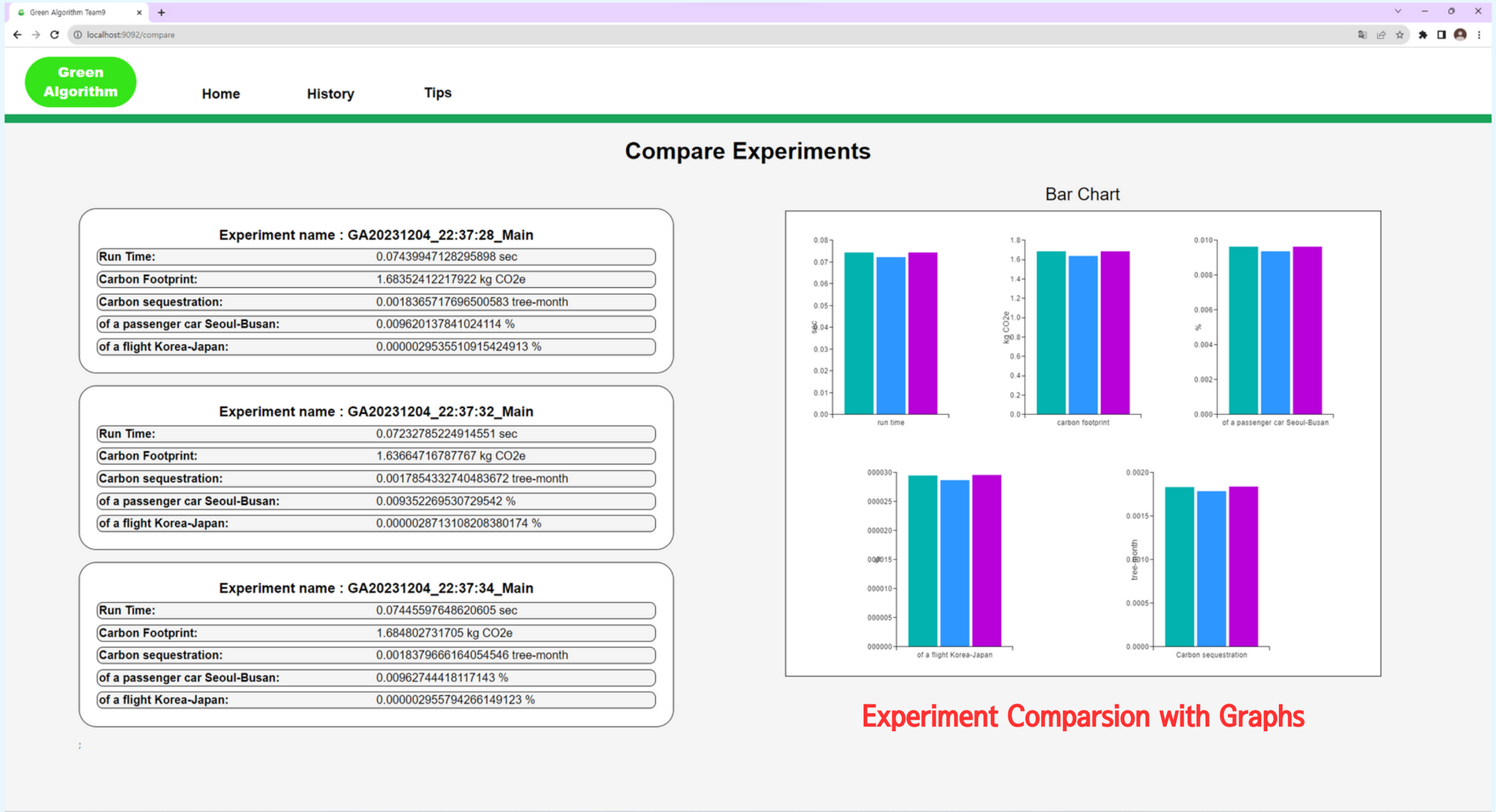
HISTORY PAGE

UI/UX Scenarios



HISTORY PAGE

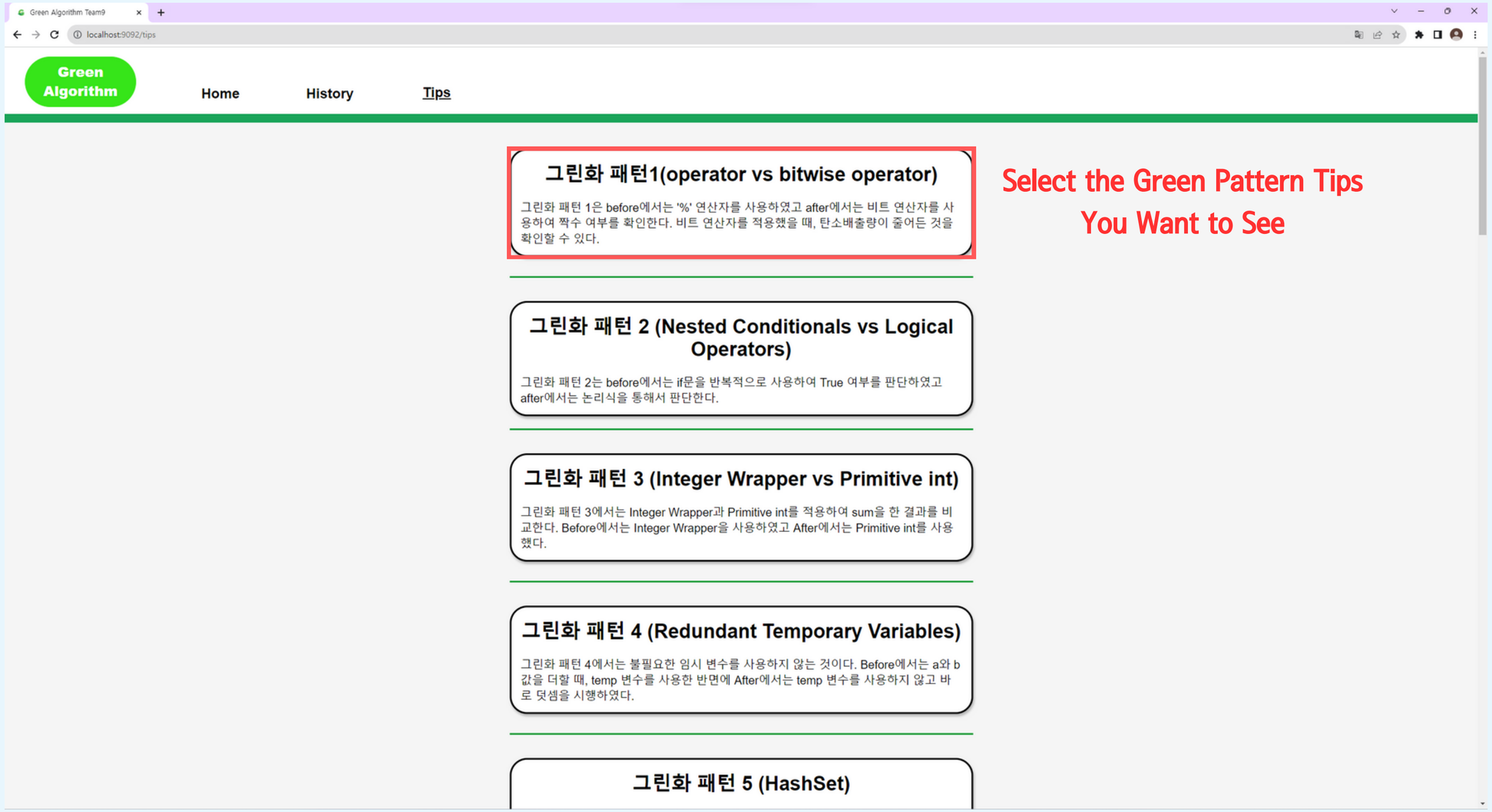
UI/UX Scenarios



Experiment Comparsion with Graphs

HISTORY PAGE

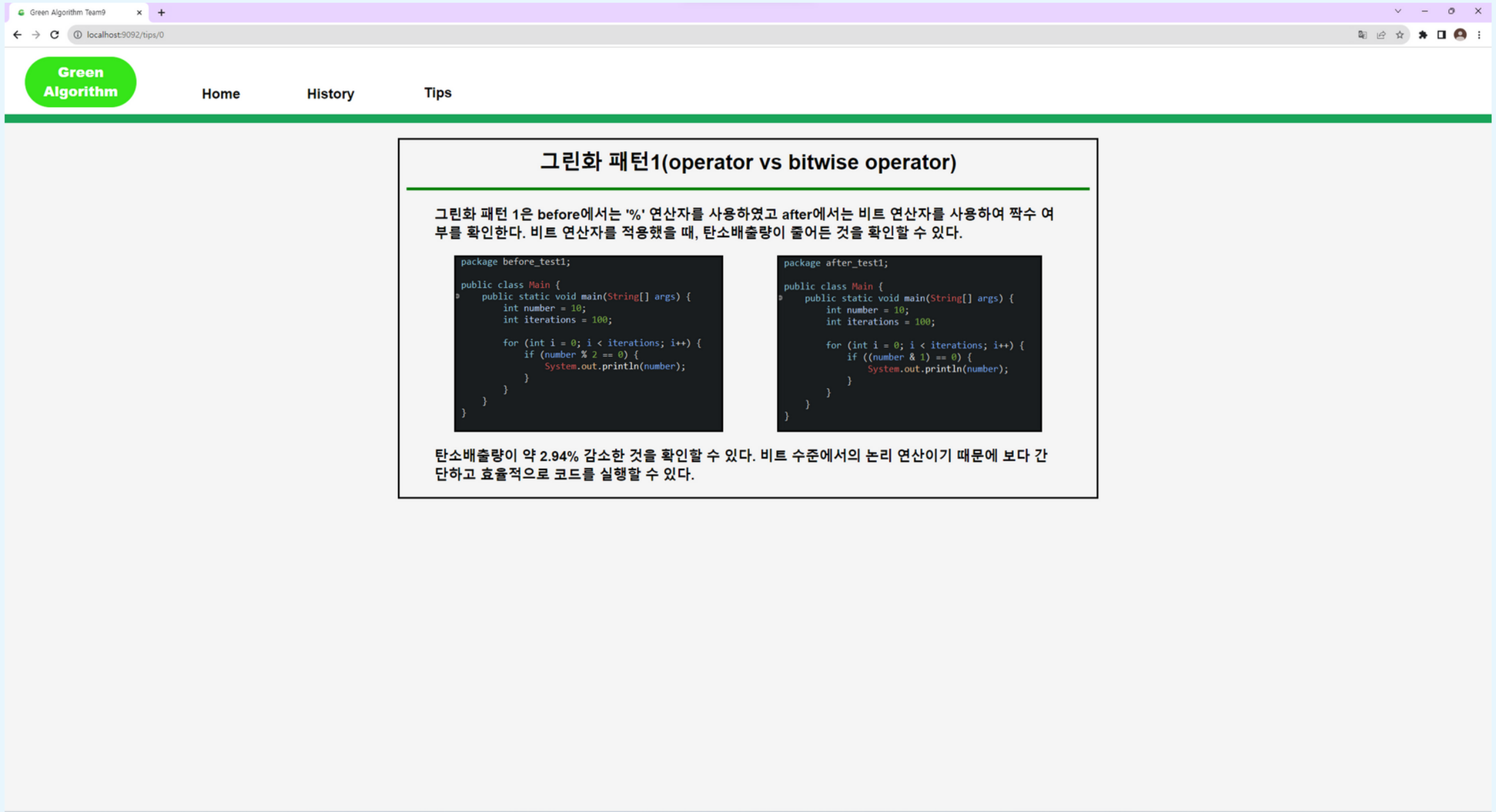
UI/UX Scenarios



Select the Green Pattern Tips
You Want to See

TIPS PAGE

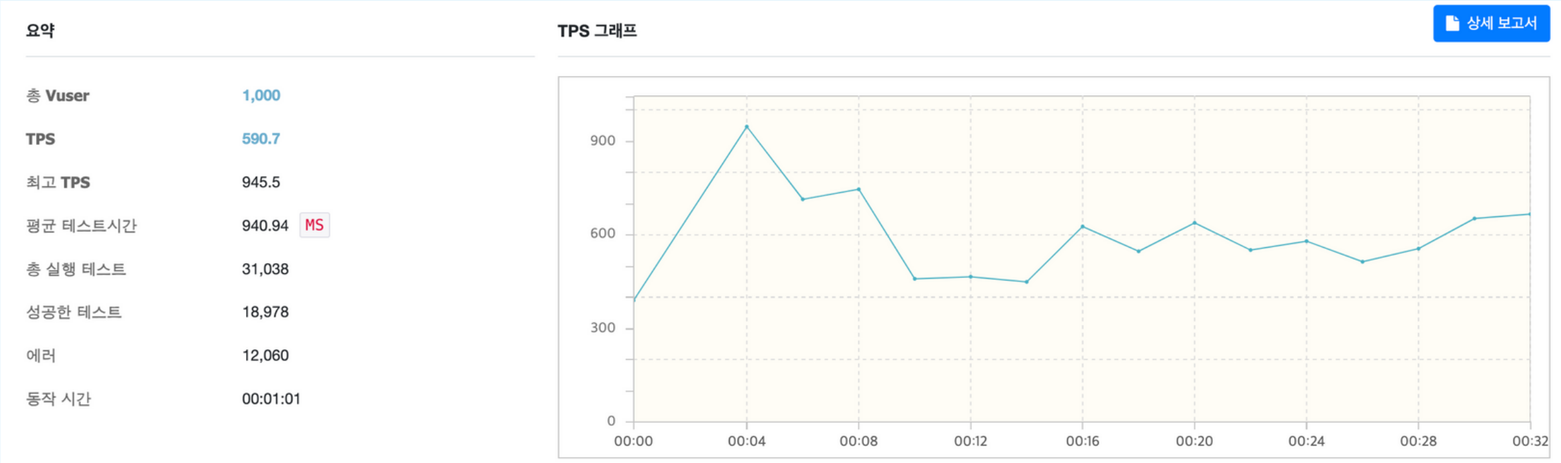
UI/UX Scenarios



TIPS PAGE

Testing & Challenge

nGrinder Test 결과



	일반 요청	실험 생성
가상 유저수	1000명	10명
최고 TPS	945	5
평균 TPS	590	1

Scale out

- Load balancing
- Docker container

Scale up

- Better CPU, More memory

Green Patterns 수집 방법

Methods of Collecting Green Patterns

BOOKS

Java Performance Tuning
Java Optimization

THESIS

How Green Are Java
Best Coding Practices

SEARCH

Java Optimization
Java Perforamce
Java Memory
Java Runtime

Green Patterns 적용 전과 후의 탄소배출량에서 중요한 비교요소

RUNTIME

Green Patterns 수집 방법

Methods of Collecting Green Patterns

Carbon footprint =
Runtime x (power draw for cores x usage + power draw for memory)
x PUE x PSF x Carbon intensity

테스트 환경

Location: South Korea
TDP: 15(W)
Memory available: 11(GB)

Green Patterns Test

```java
package run_time;

public class Main {
    public static void main(String[] args) {
        String beforejarPath = "C:\\Users\\User\\Desktop\\jar file\\before_test20.jar";
        String afterjarPath = "C:\\Users\\User\\Desktop\\jar file\\after_test20.jar";

        int numExecutions = 100;
        long beforetotalRuntime = 0;
        long aftertotalRuntime = 0;

        for (int i = 0; i < numExecutions; i++) {
            long beforestartTime = System.currentTimeMillis();


            try {
                ProcessBuilder beforeprocessBuilder = new ProcessBuilder("java", "-jar", beforejarPath);
                Process beforeprocess = beforeprocessBuilder.start();
                beforeprocess.waitFor();

                long beforeendTime = System.currentTimeMillis();
                long beforeexecutionTime = beforeendTime - beforestartTime;
                beforetotalRuntime += beforeexecutionTime;
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

        double beforeaverageRuntime = (double) beforetotalRuntime / numExecutions;
        System.out.println("Before 평균 실행 시간: " + beforeaverageRuntime + "ms");

        for (int i = 0; i < numExecutions; i++) {
            long afterstartTime = System.currentTimeMillis();


            try {
                ProcessBuilder afterprocessBuilder = new ProcessBuilder("java", "-jar", afterjarPath);
                Process afterprocess = afterprocessBuilder.start();
                afterprocess.waitFor();

                long afterendTime = System.currentTimeMillis();
                long afterexecutionTime = afterendTime - afterstartTime;
                aftertotalRuntime += afterexecutionTime;
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

        double afteraverageRuntime = (double) aftertotalRuntime / numExecutions;
        System.out.println("After 평균 실행 시간: " + afteraverageRuntime + "ms");


        System.out.println((beforeaverageRuntime/3600000)* (15*1+4.0975)*1*100*500*0.001);
        System.out.println((afteraverageRuntime/3600000)* (15*1+4.0975)*1*100*500*0.001);
    }
}
```

그린화 패턴 테스트 과정

1. 그린화 패턴이 적용되기 전후의 **jar 파일 로드**
    a. before_test#.jar
    b. after_test#.jar
2. 각 파일의 **런타임 평균과 총합 계산**
    a. 테스트의 객관화를 위해 **100번 측정**
3. 평균 런타임을 통해 **탄소배출량 계산**
4. 탄소배출량 **감소 비율 계산**

Team Works & Schedule

Frontend	Backend	Green & Doc
이진혁 이한준	박태현 한상안	김한얼 한채윤

감사합니다

Web Service Development

Calculation of carbon emissions

TEAM 9