

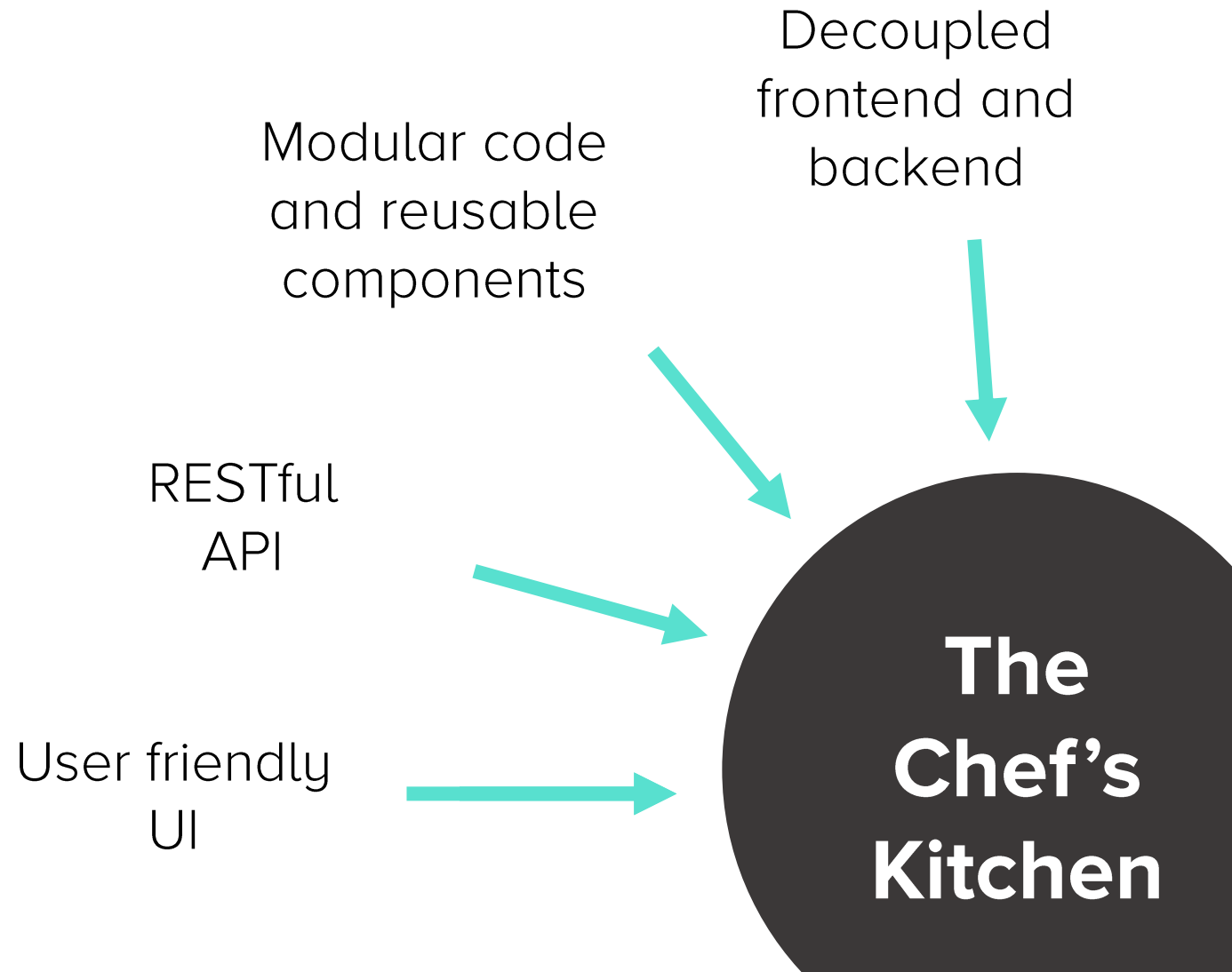


The Chef's Kitchen

Introduction

The Chef's Kitchen is a web-application built using ReactJS and the Slim framework that acts as a contest arena for CodeChef.

It satisfies all the requirements stated in the software requirements specification, provides a pleasant UI and has maintainable code.



Running the App

1. Clone the repository and change directory into the source code.

2. Install dependencies

```
cd backend && php composer.phar install  
cd ../frontend && npm i
```

3. Start the backend php server

```
cd ../backend  
php composer.phar start
```

OR

```
cd ../backend  
php -S localhost:8080 -t public public/index.php
```

4. Start the frontend react development server

```
cd ../frontend  
npm start
```

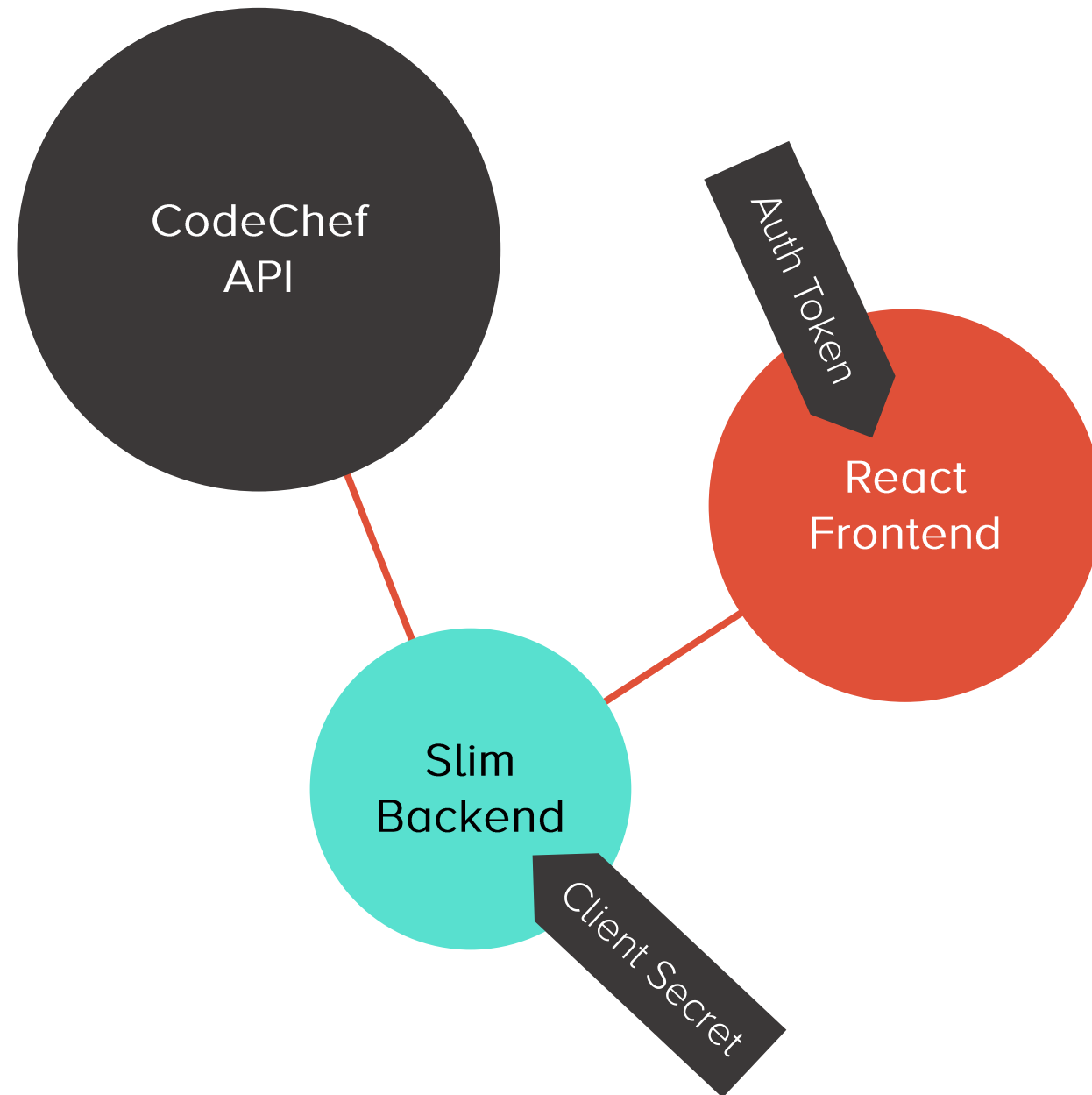
5. Navigate to localhost:3000/ in your web browser

Things to note

- Ensure composer and npm are installed. They are used for package management.
- While authorising, select the scopes 'public', 'private' and 'submissions'. This is an arbitrary decision, and can be easily changed as per the actual requirements.
- The app only lists the currently running contests, as it provides a live contest arena. This is again a decision that can easily be modified if required without affecting the rest of the application.
- Access to the internet (apart from the CodeChef API) is required for loading fonts and some libraries from content delivery networks.



The Architecture



The React frontend only communicates to the backend server, which communicates to the CodeChef API.

This allows the backend to make requests to one or more endpoints as required, process the data into the right format and return it to the frontend for rendering.

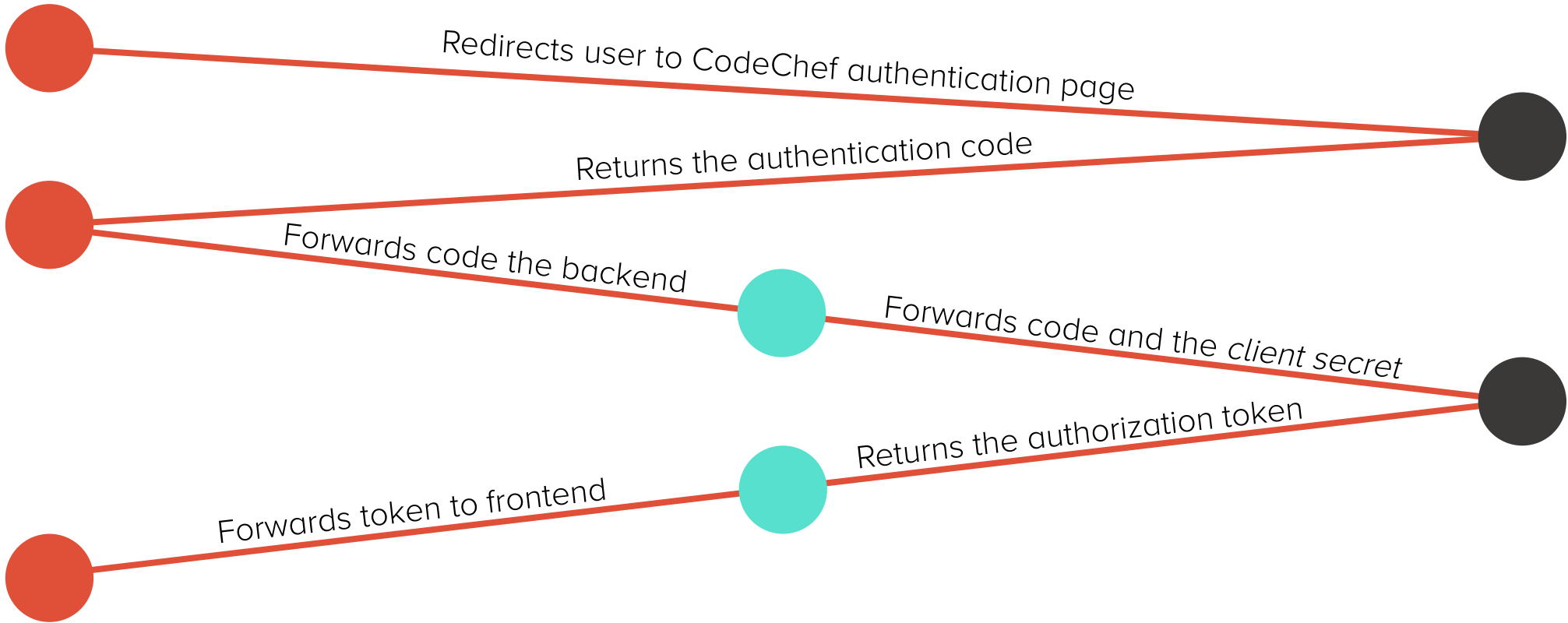
The *client secret* safely stays with the backend, and is not exposed to the user.

The authorisation token however, is stored with the frontend, and is forwarded by the backend while making API calls. Thus the backend does not need to maintain a database of each logged in user's token.

Frontend

Backend

CodeChef API



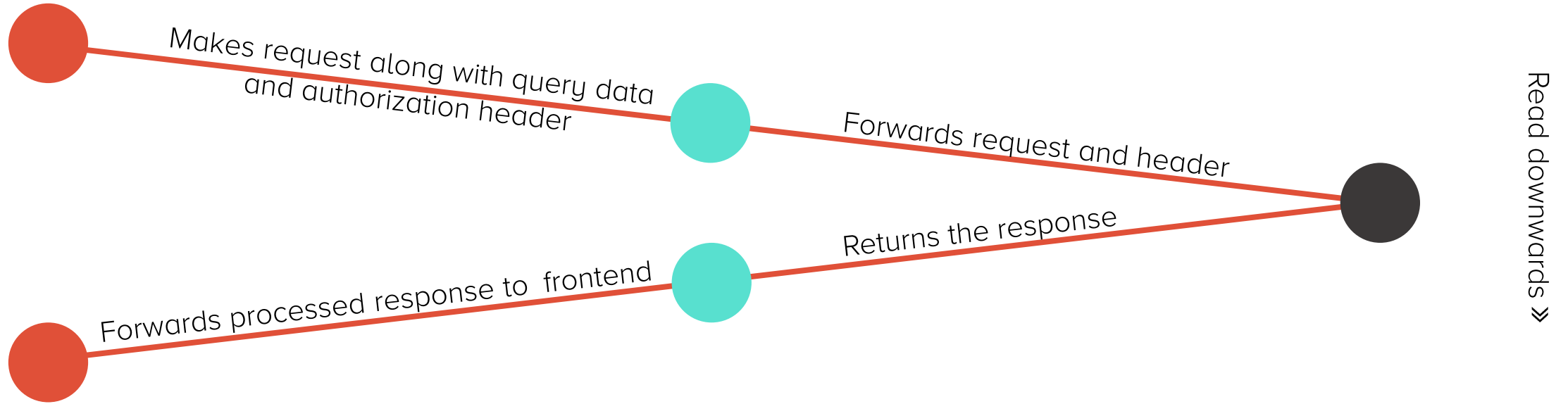
Read downwards »

Logging In Process

Frontend

Backend

CodeChef API



Read downwards »

The backend may make calls to several CodeChef APIs for a single request from the frontend.

The backend processes the data and handles errors while forwarding requests and responses in either direction

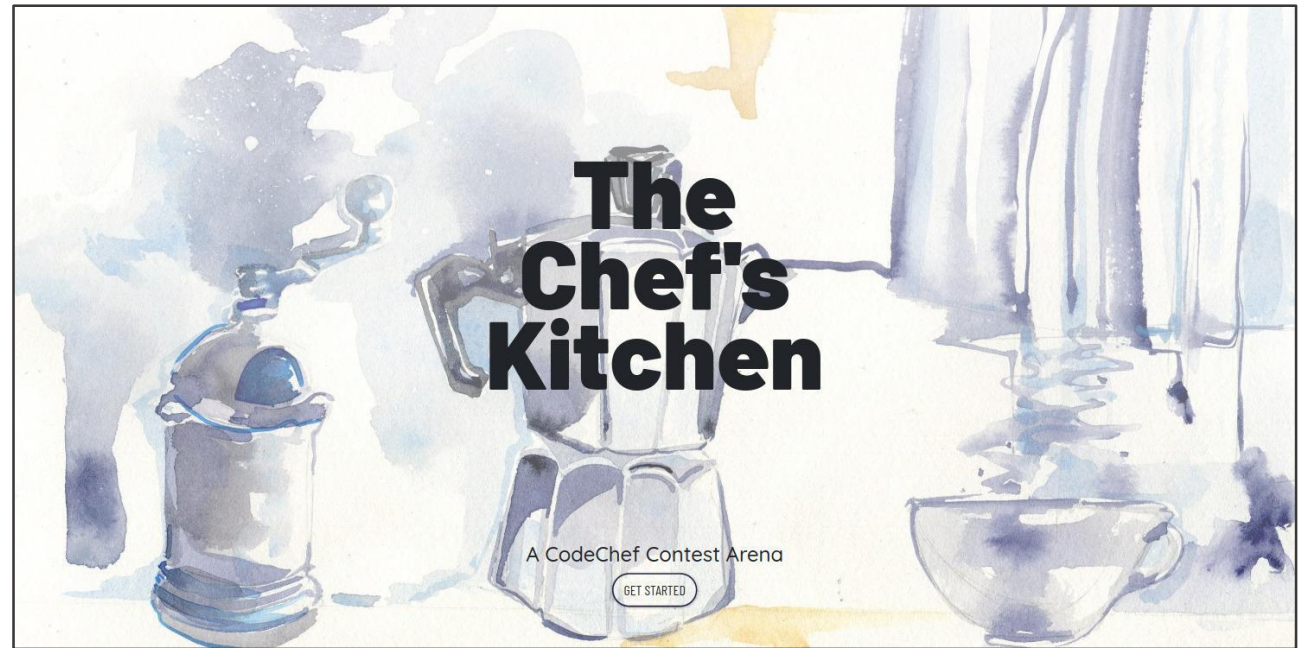
The Usual API Call



**In
Action**

Landing Page

The landing page provides a single button to get started, without any distractions.



Logging In


The user is then presented with a page to select the contest, again free of distractions.

The input box has autocomplete and makes sure the user has selected a contest before allowing them to proceed.

The Chef's Kitchen

Hi jiviteshjain [LOGOUT](#)

Choose a **Contest**



60

Loading and Error Pages

Illustrated loading and error pages keep the user involved while the components load and await responses from the backend.

Food is almost ready...



Please wait while we load this page.
If this is taking way too long, try refreshing the page or logging out and back in.

Uh Oh! We smell smoke :(



Looks like something is wrong.
Try refreshing the page or logging out and back in.

Contest Pages

The Contest Pages provide all the required information in well-organised components

The Chef's Kitchen

Hi jviteshjain LOGOUT

Contest IARCSJUD

IARCSJUD

IARCS OPC Judge Problems

556.03:25:07

Starts: October 9th 2019, 12:00:00 am

Ends: October 8th 2021, 3:00:00 am

Problems

METRO

Successful Submissions: 25

Accuracy: 84.84848484848484

TRY NOW

GREATESC

Successful Submissions: 177

Accuracy: 46.85466377440347

Ranklist

#	Username	Country	Total Score
1	rihar2006	India	4400.000
2	arsh_16	India	4400.000
3	trsteel	China	4400.000
4	codelegend	India	4400.000
5	raja8946	India	4400.000
6	pakzan	Singapore	4400.000
7	soham7	India	4250.000
8	manavspg2	India	4200.000
9	valentin_e	Russia	4100.000
10	viciouscoder	India	4000.000

Submissions

Problem	Username	Verdict
DIVSEQ	coder_209	WA
WORDHOP	codecunt7	AC
DIVSEQ	coder_209	WA
LONGPALI	sourcestrong	AC
WORDLIST	singhaltushar	AC
WORDHOP	codecunt7	WA
WORDLIST	singhaltushar	AC
WORDHOP	codecunt7	WA
WORDLIST	muski12	WA
IAREVERS	gaurav_512	AC

Problem Pages

The Problem Pages provide the *HTML* formatted problem statements, extra information as well as options to run and submit code.

The Chef's Kitchen

[Hi Jitesh Jain](#) [Logout](#)

Problem **METRO**

Srushti Metro

The city of Srushti is impeccably planned. The city is divided into a rectangular array of cells with M rows and N columns. Each cell has a metro station. There is one train running left to right and back along each row, and one running top to bottom and back along each column. Each train starts at some time T_i and goes back and forth along its route (a row or a column) forever.

Ordinary trains take two units of time to go from one station to the next. There are some fast trains that take only one unit of time to go from one station to the next. Finally, there are some slow trains that take three units of time to go from one station to the next. You may assume that the halting time at any station is negligible.

Here is a description of a metro system with 3 rows and 4 columns:

```
$ $
S(1) F(2) O(2) F(4)
F(3) .....
S(2) .....
O(2) .....
$ $
```

###Sample Output

10

Specifications

Author
admin2

Challenge Type
subtask

Source Size Limit
50000 bytes

Time Limit
2 seconds

Languages
C++17, C, C++, C#, F#, Go, Java, JavaScript, Kotlin, Lua, Objective-C, OCaml, Perl, PHP, Python, R, Rust, Scala, Swift, TypeScript, VB.NET, Verilog, Zig

Tags

Submissions

Username	Verdict	Time (s)	Memory (bytes)
codechef	AC	0.05	18368
riteshdixit	AC	0.34	177056
vishwanath	AC	0.4	18368
aditya_149	WA	0	0
amder1111	AC	0.4	18368
monish_124	AC	0.4	18368
amder1111	AC	0.4	18368
hailhail	AC	0.39	18368
pakistan	AC	0.4	18368
AJN	AC	0	15248

IDE

Run

Submit

Problems Faced

- One of the problems I faced was the lack of familiarity with the Slim Framework and the CodeChef API. Thus building this application involved reading a lot of documentation. But since the concepts are the same, I was able to manage.



@jiviteshjain
Jivitesh Jain