-COURSEWORK ASSESSMENT 3: GROUP PROJECT-
-Functional programming-
-(Team 23)-

- Awais Patekari
- Mariia Ignashina
- Uttkarsh Raj
- Zhengwu Ren

-Introduction-

⇨ In this group project we have developed a real-world Haskell app that performs HTTP requests and parses an input from string to a custom Haskell data type and stores it into a database.

⇨ The database we have picked is the UK food hygiene rating database. We chose it because there is a lot of useful data in this source, but it's not structured in a very user-friendly fashion. We are also interested in deriving generalisations and statistics that could help people navigate through local food scenes.

⇨ The Food Hygiene Rating Scheme (FHRS) and Food Hygiene Information Scheme (FHIS) data is published at www.food.gov.uk/ratings

⇨ The **FHRS** is run in England, Wales and Northern Ireland, and the **FHIS** is run in Scotland.

⇨ The rating value is the result of an inspection for a business where 'FHRS' uses a rating between 0 and 5 and 'FHIS' uses 'Pass', 'Pass and Eat safe' or 'Improvement Required'

Extracting the Information

⇨ We are extracting the information from the web source and from there we are putting it into our database.
⇨ We also check the URL in the database, if it is present then we are not downloading the data.
⇨ We are initialising the database and saving the establishment.
⇨ We also maintain the error handling for the modules.

How to run the application ?

=> To run the application first we will go to the project directory by running the command.

```
cd haskell-project
```

=> Next, we will run the stack command:

```
stack build
```

=>Stack needs a GHC to build the project. If stack discovers that it is missing then it will automatically install for it. We can even do this manually by running the stack setup command.
=> Now to run the application we will have to run the following command:

```
stack run
```

- ⇨ It's a shorthand for stack build && stack run executable.

Once the application runs it will show the following options to select from, shown in the figure given below. Basically this is the UI for the app and the welcome page.

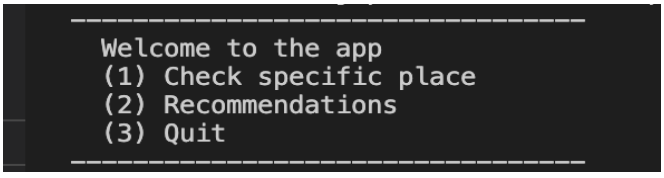

Fig1.1

Note: Here the API is using Desc_Rating which returns the results in the order of Rating in descending order (0,1,2,3,4,5, Improvement Required, Pass, Pass and Eat safe, Awaiting Publication, Exempt, Awaiting Inspection)

- ⇨ For Example: If we want to sort the data with top 10 places with the address, in descending order. We are doing it in a following manner.

```
-- top 10 places with this name
getInfoNameLink :: String -> String
getInfoNameLink name = "https://ratings.food.gov.uk/enhanced-search/en-GB/" ++ name ++
"/^/rating/0/^/1/10/json"
```