GROUP - 27

Functional Programming Group Coursework

Public Holidays of the World

Student IDs:
Hassan Al-nasrawi 200819101
Safiah Mohammed Alamr : 180831904
Franklin Antony : 200626510
Abdualmajeed Ali M Alzhrani::190625997

Aim

The main purpose of this program is to

- Download the details of the public holidays from an open source library for Dates and Public holidays which in JSON format
- Parse the JSON data into a haskell data type
- Save this data in DB (using INSERT)
- Perform set of tasks involving DB functions such as SELECT, JOIN etc based on users choice
- Save (or write) the details into JSON file after querying it from the DB

How to compile the code

```
stack build
```

How to run the code

```
stack exec Public-Holidays-of-the-World-exe
```

How to generate the haddock documentation

```
stack haddock
```

Using the program (control flow)

1. The user will be asked to select a country ID and a year which will then store all of the local holidays relating to the chosen country and calendar year .
2. The user will then be given 3 options from which they must select one of them:
   a. The first option will output all of the data stored within the database i.e.country holidays for the selected year.
   b. The second option will allow the user to input two dates and the output will give all of the holidays that transpire between those two dates for the selected country and year.
   c. The third option will output the local and international holidays stored within the database for the selected country.
3. The user will then be asked if they would like to end the program or not if ‘NO’ then the program will repeat from point 1

How does it work (Explanation)?

The web source we use to obtain the data is the open source library called (https://date.nager.at). We use the API to get the data depending upon the user's choice.

Example : When the user chooses a country by giving country code and the calendar year as arguments. Let's take 'GB' - for Great Britain and '2019' as the year. The API structure can be made with those like shown below -
`https://date.nager.at/api/v2/publicholidays/`**`2019`**`/`**`GB`**

The data is now parsed and converted into the Haskell data type (HolidayRecord) and stored in the DB using the INSERT statement. For using the DB we use SQLite3 and we use the Database.HDBC.Sqlite3(have to be defined separately) module to achieve the same.

We imported the Database.HDBC and Database.HDBC.Sqlite3 into our module and parse library because we will make use of the record data type. The function of initializing the Database is using the connectSqlite3 process in the Sqlite3 module, which takes a filePath. It performs as an IO action, opening that file and returning a connection between that database file (conn). The run function is part of the HDBC module, and it runs any statement that we have prepared as String, so it takes a connection, statement, and some parameters, then it will perform that action. It will then return the number of rows into the database that has been modified by that statement.
Three tables are stored in the database which will have SQL data stored within them. We saved HolidayRecord to convert from Haskell data type to SQL data type that would then be stored within the database to their corresponding tables.

In this next part we added the various queries for the user to find certain information using SELECT and JOIN in which the user can select between different 'CountryCodes' and then a selected 'date' which would output the corresponding holidays for the selected values, also the JOIN query will allow for joint values to be outputted where in the case of this program it is both local and global holidays that are recorded in the database.

Within this section, we had to complete the opposite of the fetch and conversion within the first part we fetched data from the database then converted it into Haskell before converting it into JSON and then storing it within a JSON file with the assistance of the 'Aeson' library.
In addition to this we also had to change the type of data used as fetching and converting into haskell gives us data as a type of 'Bytestring' which we needed to change into 'String' so that it can be converted into text and stored into the JSON file.