Documentation of project Time Spent Heat Map

CS326

Hang Weng

A screenshot of a computer

Description automatically generated

This project used to call canvas, but it only use the part of concept which the final map was using grids display.

To run this project, the user need to open a terminal(power shell or through VS code), and under the /canvas, run some command.

for expression.

And

for database connection.

After that it should be able to run locally.

If there is no npm or node.js environment please install it. It can find the method easily by searching GOOGLE.

And then run 

A screen shot of a computer

Description automatically generatedthey will jump out, that means the server is ready to be connected.

A close-up of a website

Description automatically generatedtype this line to the browser, the UI should showed now.

A screenshot of a computer screen

Description automatically generated

Like this. If it is not like this, Check the environment and the terminal again.

The database is already set up and connect to my default.

You can read the map create by me(author).

A screenshot of a computer screen

Description automatically generated

testUser, week 1

The blocks represent the hours of a week, not only about rime, but also add notes on it as this picture showed.

The note only shows when you hover on it.

In this UI you can create your own records, color your map, and commit it to the database. The database is connected to elephant SQL, and the key is listing on the server-side code. Once create your own database, they will not lose.

The reset button only works on the UI, so if a mistake was updated to the database, just delete it.

The buttons of create, read, update and delete using fetch API and REST API to communicate with server and read/write data to the database.

CODE STRUCTURE:

It was separated to client side and server side; they are in different folder.

On the client folder, it has main.js that hooks the front end and the connect.js to help send out data. And execute some local operations.

The connect.js file has all functions the help to connect front end and back end, using fetch api.

The utilities contain the heat map class which is the main class of this project, construct the data and also render the page.

Data shows on the UI will restore as an array containing the focus level and the note text data, this long array will send to database and keep as a JSON string. When reading and changing it, the class of heatmap will render this page again based on the data array.

For example, update a record instance,

A screen shot of a computer code

Description automatically generated

Click the button, and it will send the data with name and week to the function from connect.js, and then,

A computer screen shot of code

Description automatically generated

This function will using fetch api to send request and data body to the server.

On the server side, I have express tech that catch the specific request like /client to visit, or like /update to change data on the database.

A screen shot of a computer program

Description automatically generated

Based on the request method and keyword, it will call the database function,

A screen shot of a computer program

Description automatically generated

It helps create sql query and send to the database to change the record. After changing, it will brings back the status and notice the user by response status code.

This is how an execution of update instance. It goes from the front web page to the cloud database.

The database information is connected to a database I created on elephant SQL,

A computer screen shot of text

Description automatically generated

It connects to this database, and the database looks like this in their web: A screenshot of a computer

Description automatically generated

This database has already created a table call heatmap so you don’t need to create, if you plan to have your own database, open a new instance at the elephantSQL, fill in the area simmilar to the image showed above.

And create a table also called heatmap using this query:

CREATE TABLE heatMap (  
    user\_name varchar(255),  
    week\_number int,   
    mapdata text  
);

This part can be handier but I don’t have a lot of time here. So the created table is not changeable.

Now you should be able to play on this project.

Key features:

The webpage is persistent.

The heat map can use up to 5 colors to represent focus levels.

The heatmap can also be treated as a canvas with limitations.

The data of the map can be create/read/update/delete through buttons connect to the database.

The database can be also customized by the user just follow the part of instruction, and user will have their own secreted database.

The database can restore up to 200 rows of data for free.

The page is beautiful.

Thanks for reading.