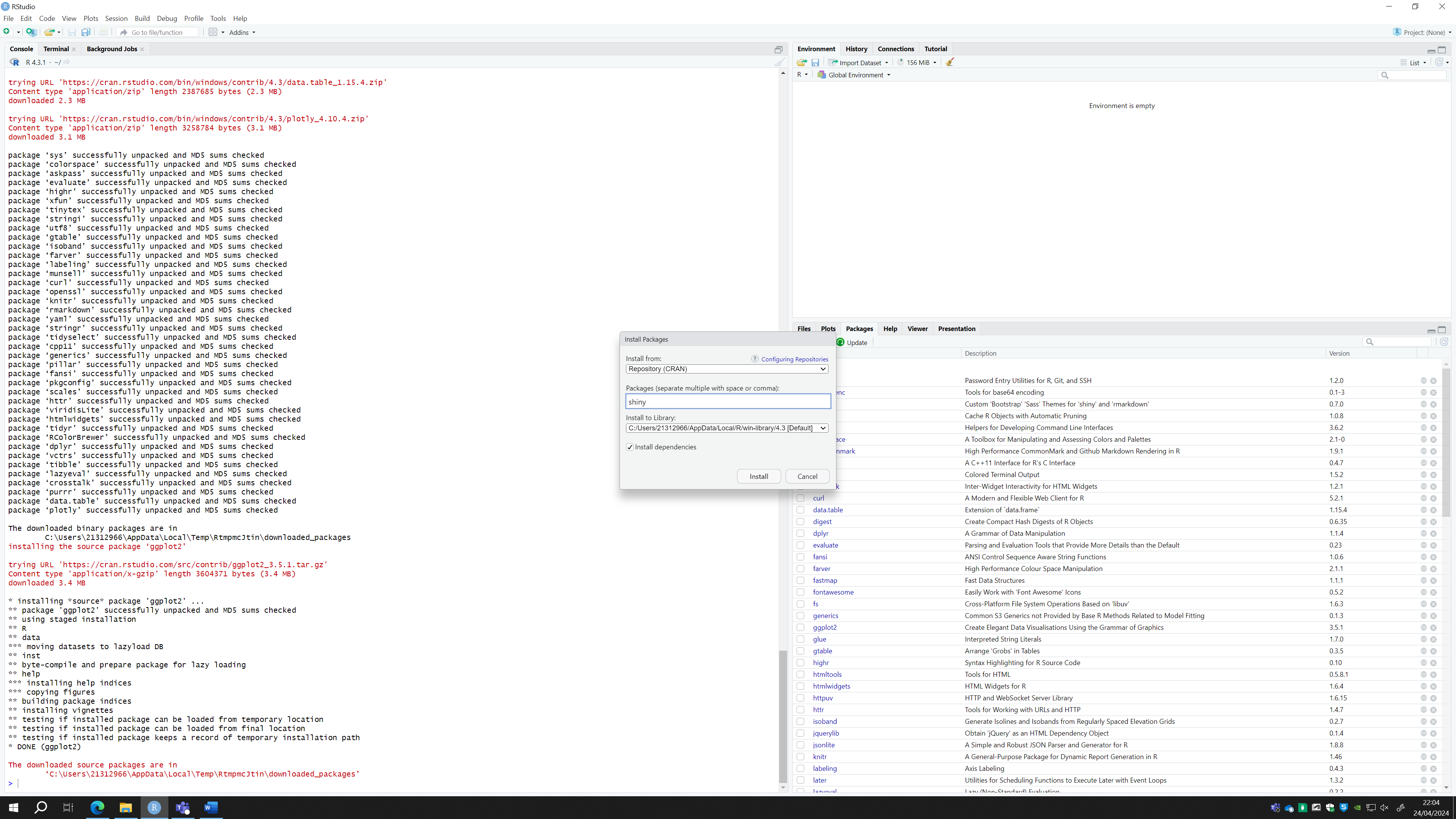
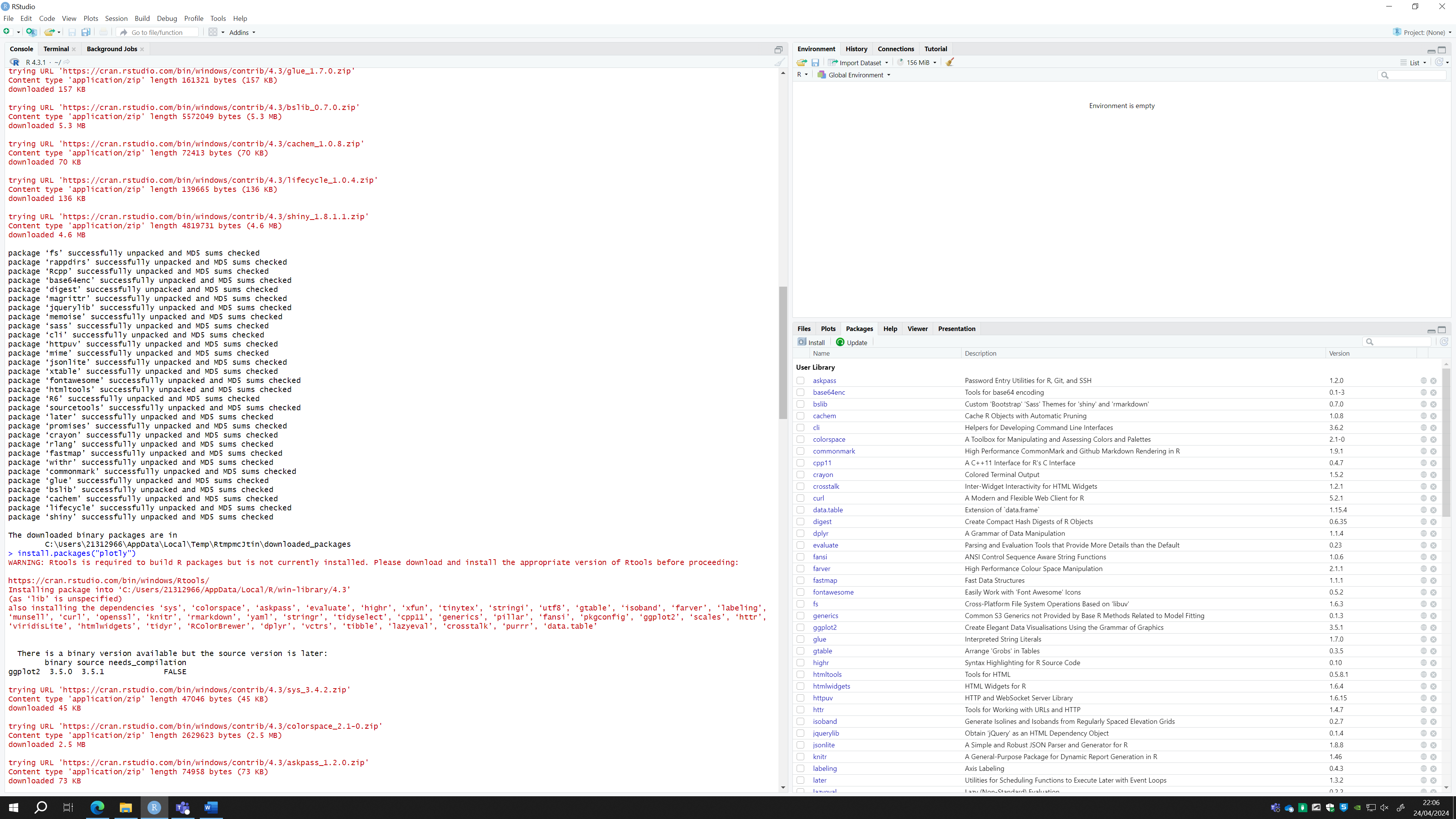
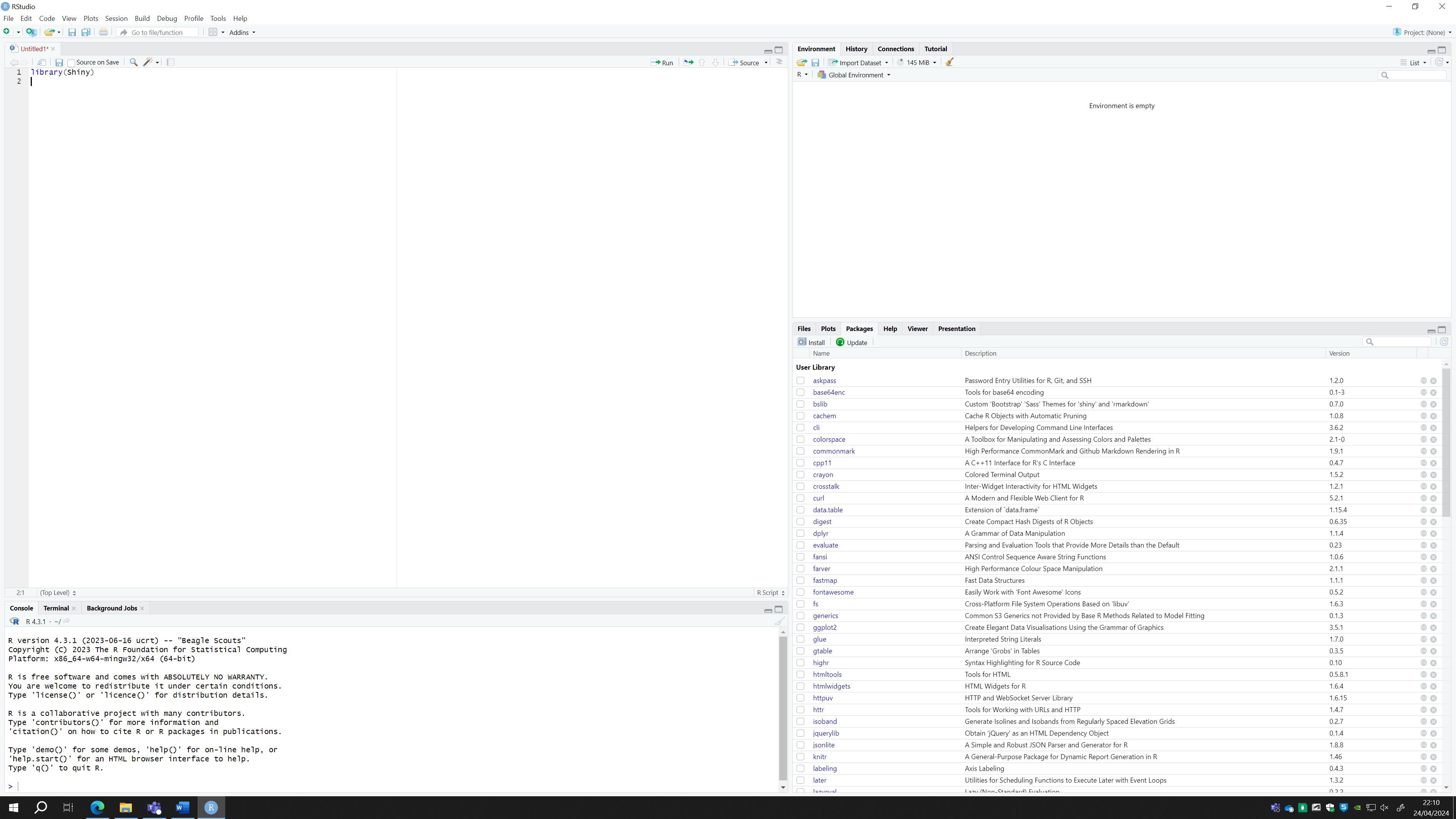
Steps in Implementation - Starting with Basic Operations Required for the Calculator to Work:

**1)** The procedure began with the R Shiny library package being installed in the RStudio Desktop environment. To enable the following development steps, this required sifting through the R environment's package repository and installing the necessary package.

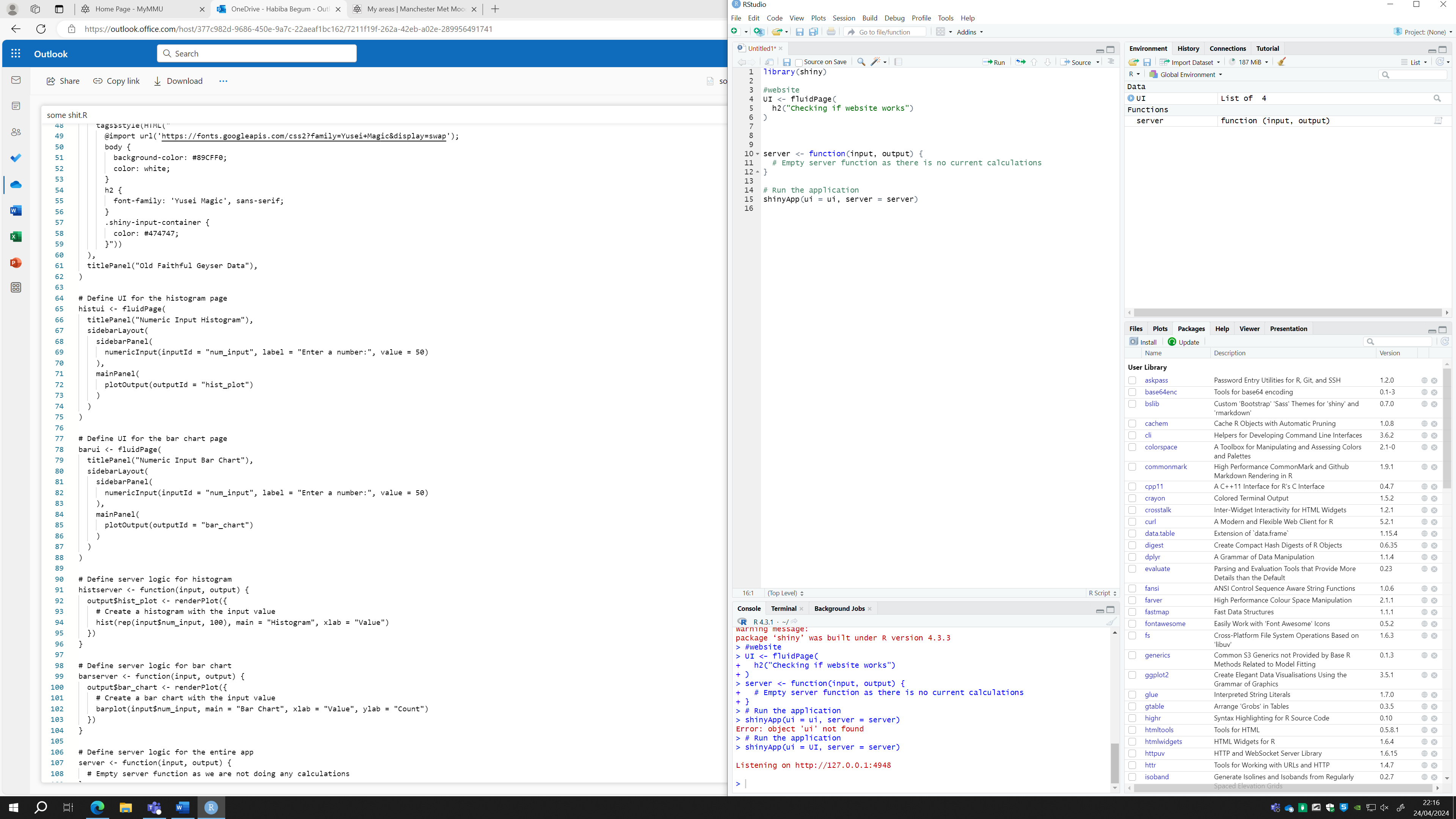


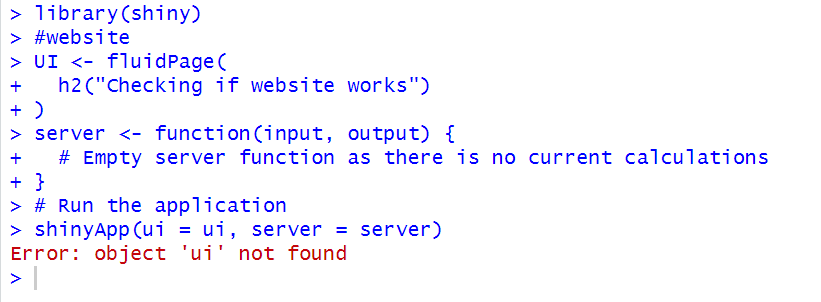


**2)** To make sure the Shiny library was available and prepared for use in further development tasks, within the R script, R Shiny was called after the installation was successful.

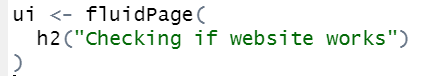


**3)** Then followed the development of a fluid page and the server that contained the website. This initial assessment was carried out to confirm the page was functioning and the library's proper installation was an essential step in this phase.

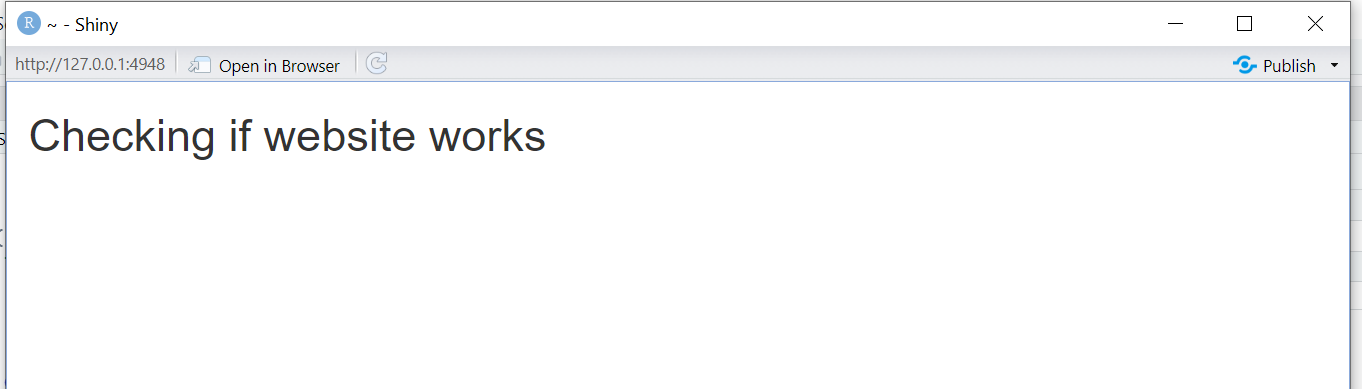




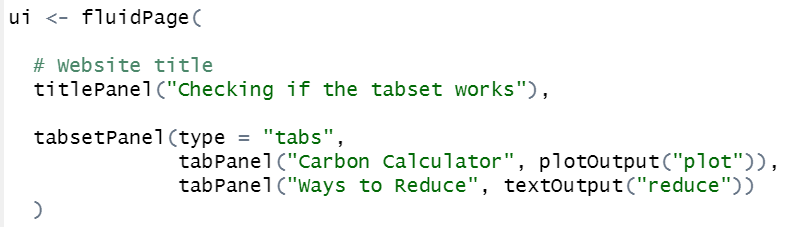
A thorough code study was initiated by an error message that was found and indicated that there was no 'ui'. To fix the code with correct syntax and functionality, the reference to 'UI' was changed to 'ui', therefore resolving the error.



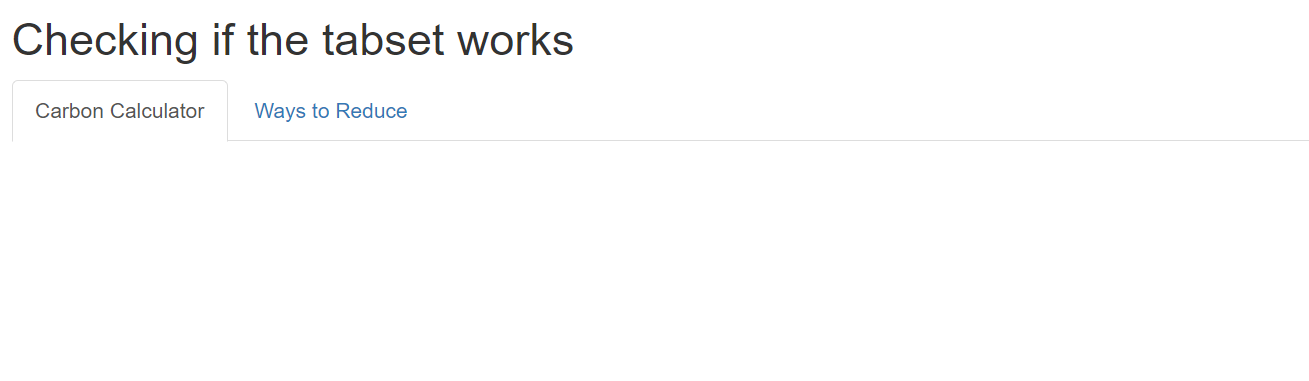
The code was executed, and the webpage was displayed.



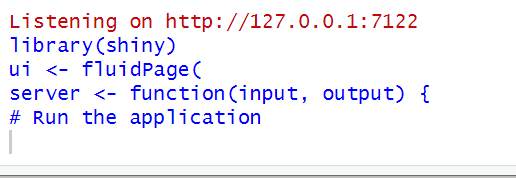
**4)** Moving on to the tab panel development stage, I carefully structured the interface using the complete Shiny instructions, particularly the Tabsets tutorial supplied by (*Tabsets* 2014).



Currently, just two tabs are available. The first tab contains the carbon calculator, which produces output in the form of graphs that can be shown visually. The second tab, labelled "Ways to Reduce," displays textual output containing information on how to reduce one's carbon footprint.

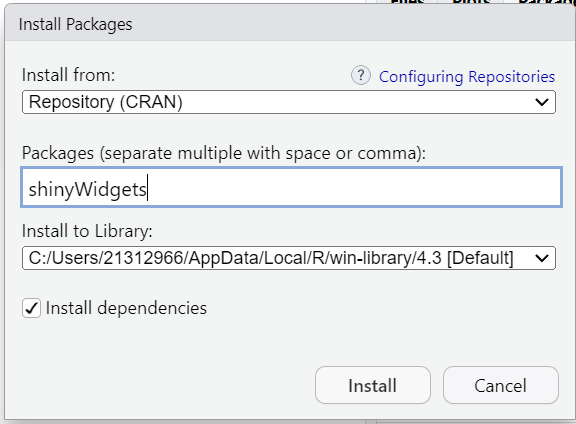


**5)** I came across an issue in which R Shiny failed to properly execute the code, instead running only the comments and skipping code blocks. Despite lengthy study to determine the core source of this anomaly, my efforts were unsuccessful, since forums addressing similar concerns went unanswered.

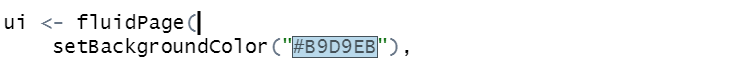


I determined that the problem was caused by the website running concurrently in the background, which I failed to quit before re-executing the code.

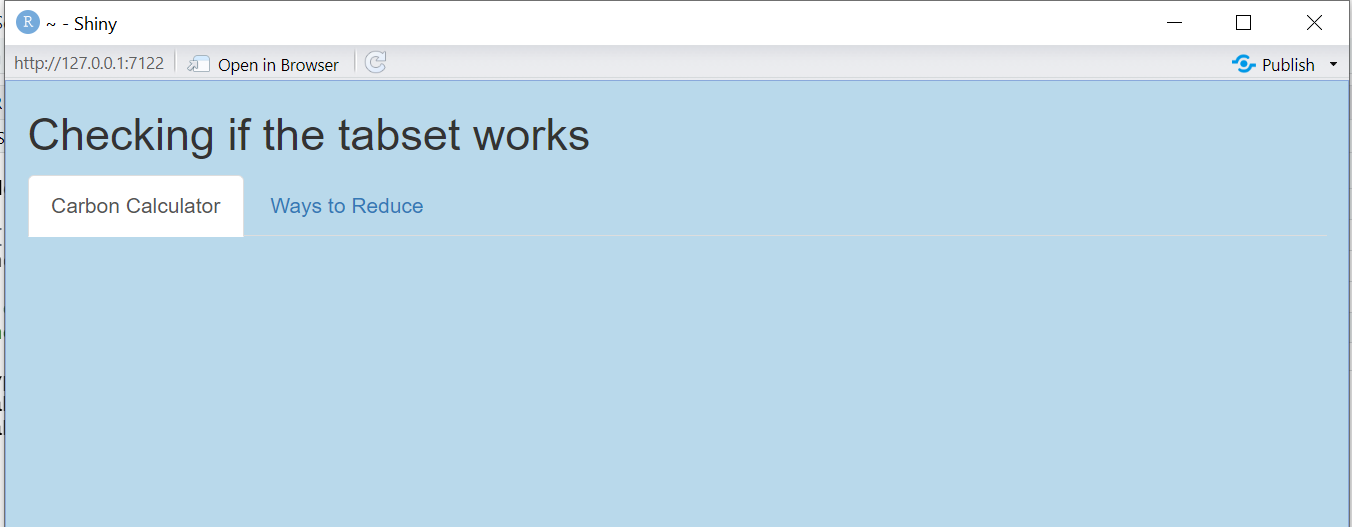
**6)** I proceeded along with the task of changing the background colour of the webpage. To accomplish this, I started by installing the ShinyWidget package. This package allows for considerable customisation possibilities, such as specifying and adjusting the website's colour palette.



I installed the previous package and included its library into the code. Following that, I set the backdrop colour to reflect the designated hue of Columbia Blue, as per the design elements.

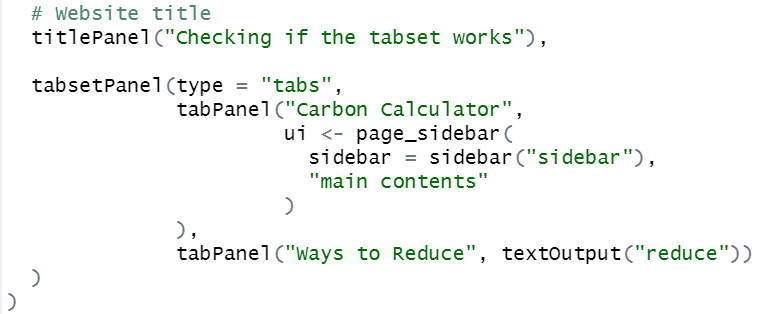


Subsequently, I ran the code to ensure that the website's backdrop adopts the designated Columbia Blue hue as intended.

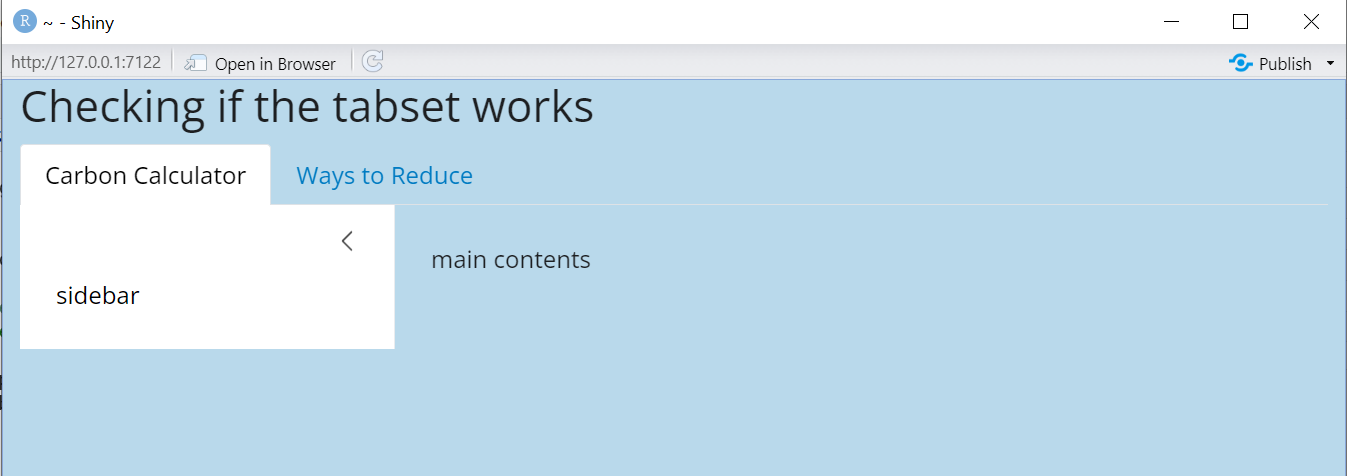


**7)** I started the design process by working on the Carbon Calculator page. Initially, I divided the page into two separate sections: one for input parameters and another for showing graphical output.

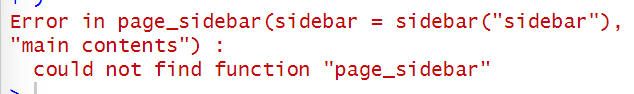
I created the sidebar layout using resources from the R Shiny help website. (*Build a user interface* 2014).



I executed the code to check it works correctly.



**8)** I obtained an error while switching from the University RStudio environment to my personal RStudio environment.



I was unsure about the origin of this error, so I checked the website from which I got the instructions for developing the side panel layout.

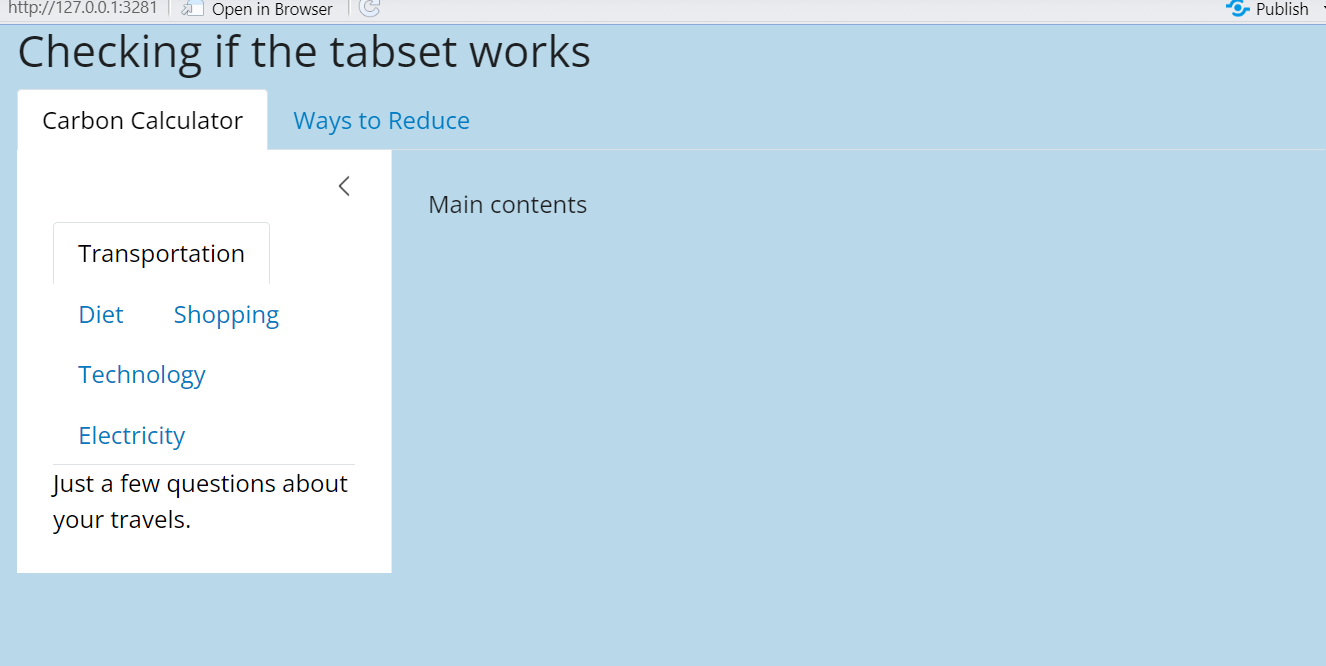


I forgot to include the bslib library, rendering RStudio unable to execute the code because it relies on utilities given by this library.

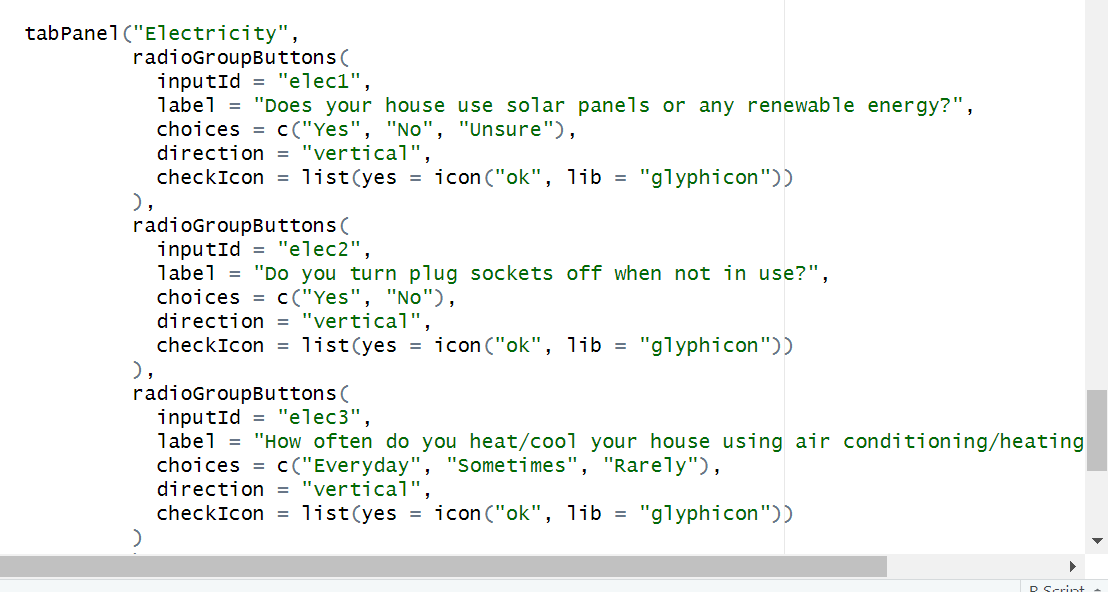
**9)** I began with the task of separating different areas of inputs by adding tab panels to the side panel.

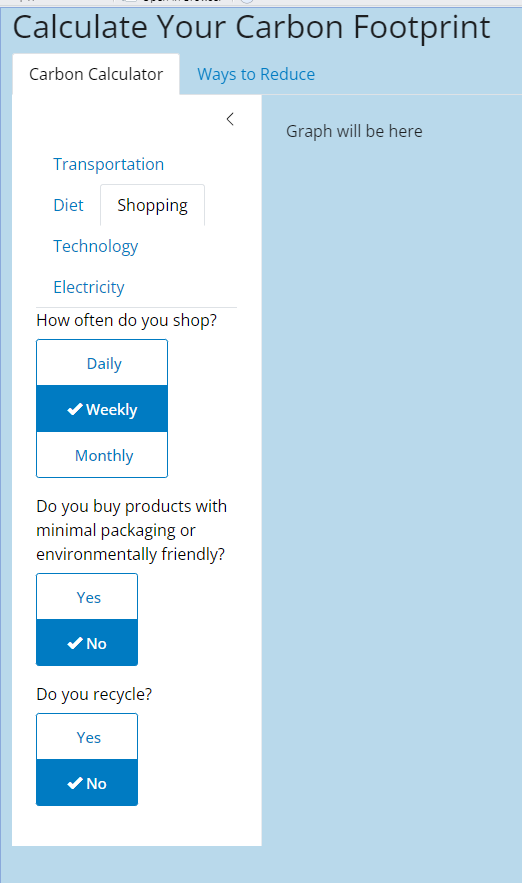
A computer screen shot of text

Description automatically generated



**10)** I incorporated the relevant questions along with the related radio button responses to the appropriate tabs. For this, I utilised the vertical radio button code provided by (*Shinywidgets overview* 2019).





**11)** I proceed to collect the user's inputs and assign numerical values to each based on their decisions. For this task, I used the approach provided at (Ben, 2022), where each input adds to a numerical value between 0 and 2.

A screenshot of a computer program

Description automatically generated

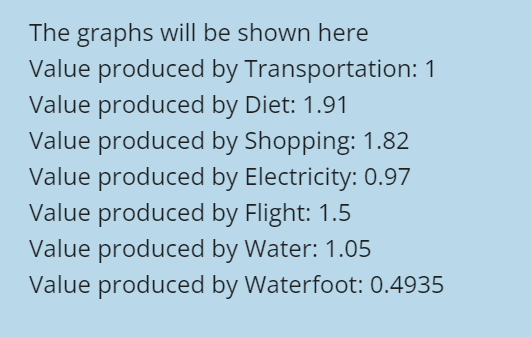
A screenshot of a computer

Description automatically generated

**12)** Using the user-selected inputs, I calculated the individual's carbon footprint, suggested as X(CP). The combined results from the questionnaire responses establish the value of X, which ranges between 0 and 2. Then, the method includes multiplying X by a constant assigned as C, which represents the UK's per capita carbon emissions (4.7). In addition, the outcome is adjusted by the percentage weights assigned to several categories, which are: transportation (19%), diet (20%), shopping (7%), home energy (19%), flights (15%), and water usage (10%).

A computer code with black text

Description automatically generated with medium confidence



**13)** Create graphical visualisations for each section of the carbon calculator with the plotly library, following the instructions at (*Bar charts in R* 2013).

A close-up of a computer screen

Description automatically generated

To check that the graphs are responsive to user inputs, I will run a test to confirm that the visualisations dynamically adapt based on the new input values. I've included the basic graph below, before any input changes were made.

A graph of a bar chart

Description automatically generated with medium confidence

Following input changes, the updated values are shown in the graph below.

A screenshot of a graph

Description automatically generated

**14)** I created a distinct bar chart to compare the individual's entire carbon footprint to the average carbon footprint in the UK. To begin, I computed the individual's carbon footprint by adding the results from each step of the calculator.

A close-up of a computer screen

Description automatically generated

As a result, I used the average carbon footprint per capita in the UK, derived from extensive literature research, as an average for comparison. Using this information, I created a graph to illustrate the difference between the individual's carbon footprint and the established average for the UK population.

A computer screen shot of a program

Description automatically generated

I ran validation tests by gradually modifying the input settings to see how responsive the graph was to these changes.

The initial graph is shown below:

A screenshot of a computer screen

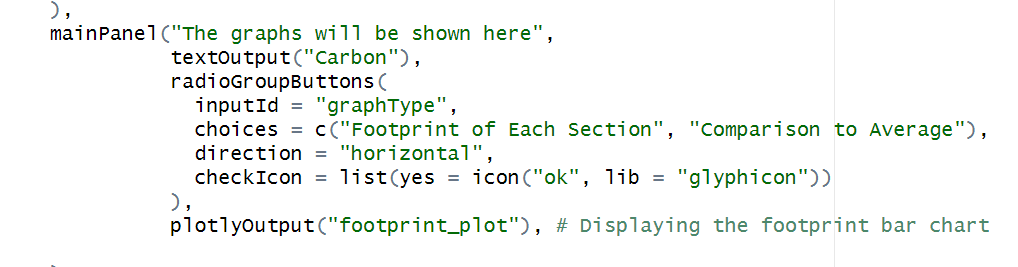
Description automatically generated

Following changes to the inputs, the graphs below show the updated parameters:

A screenshot of a computer

Description automatically generated

**15)** Recognising the potential for user confusion caused by the presence of two graphs on a single page, I designed a solution that included an additional radio button. This allows users to choose their preferred graph for visualisation. I used the input radio button component from <https://shinyapps.dreamrs.fr/shinyWidgets/> to add the user choice option into the main panel interface.



Then, within the server component, I added an if statement to determine the user's preferred output. This conditional statement permits the display of the appropriate graph based on the user's selection.

A screenshot of a computer program

Description automatically generated

After making these changes, I tested thoroughly to confirm that both graphs dynamically adjust to user input, resulting in successful functionality validation.

**16)** Recognising the weak interpretation of the simple graphs, particularly when determining the quality of a user's carbon footprint, I decided to improve their visual appeal. To accomplish this, I created a colour differentiation approach in which the footprints linked with each topic were coloured differently if their respective value surpassed the value, one. Additionally, for the comparative graph, I used a similar colour scheme to emphasise carbon footprints that exceeded the average value.

relying on resources such as the practical guide (*Plotting 2.0: Shiny* 2018), and referencing insights from the book, (Wickham, 2021).

I used appropriate techniques for incorporating colour modifications into the graphs.

A screenshot of a computer program

Description automatically generated

Following that, I undertook extensive testing to ensure that the colour changes implemented were effective, and that the graphs appropriately reflected the changes in carbon footprint values.

A screenshot of a graph

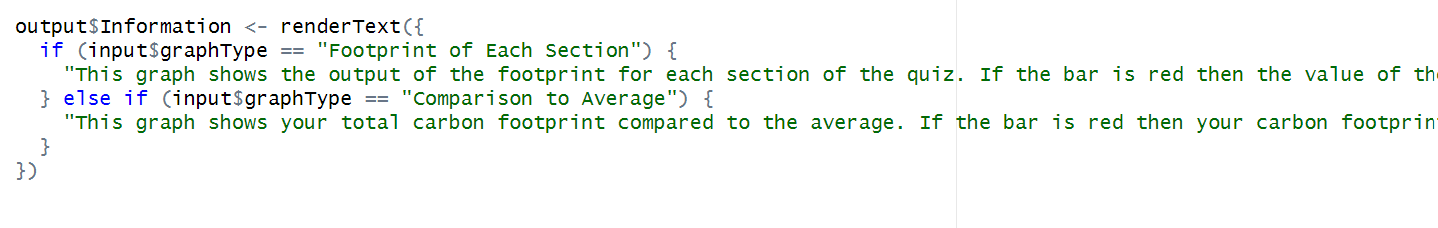
Description automatically generated

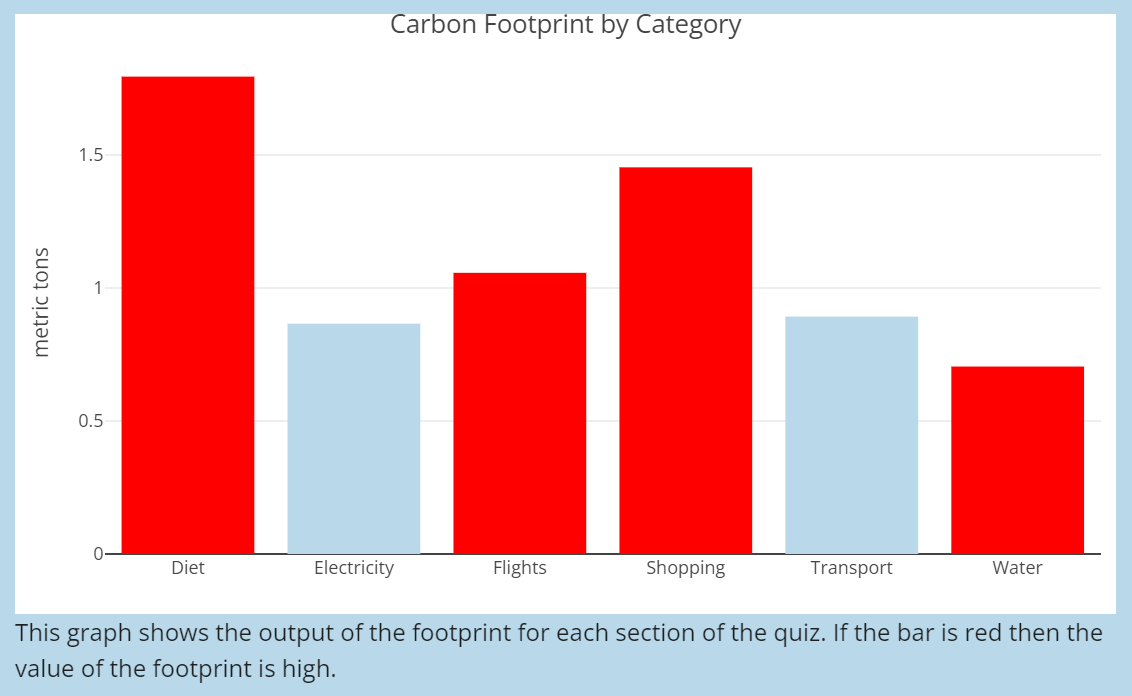
A screenshot of a computer

Description automatically generated

**17)** Recognising that the graphs lacked context without accompanying information, I included descriptive text underneath each graph. This paragraph explains the graph's purpose and the meaning of the colours used.

By involving this instructive information, readers can better grasp the visual representations in the graphs, allowing for a deeper understanding of their carbon footprint inquiry.

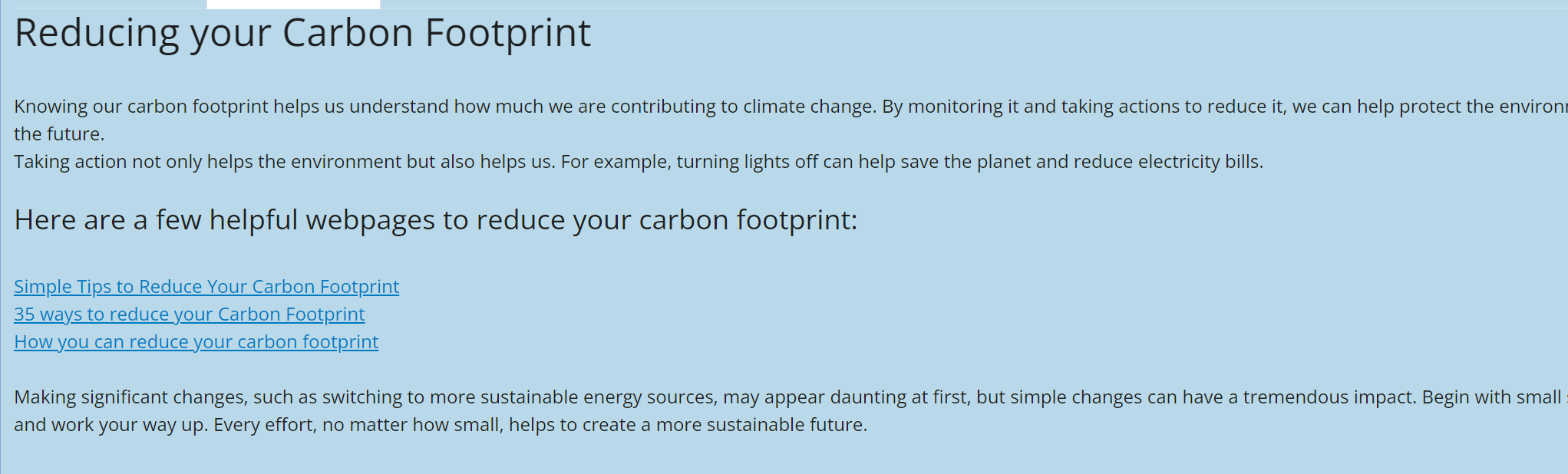




**18)** Incorporating data within the "Reduce Footprint" tab to assist users

A close-up of a computer screen

Description automatically generated



**19)** Acknowledging the source information is essential to maintaining integrity and giving due credit to the original creators. As a result, I have added references to the used data beneath the graphs, where it was incorporated. This provides transparency and adheres to ethical standards in information transmission.

A computer code with green and white text

Description automatically generated

