## Introduction

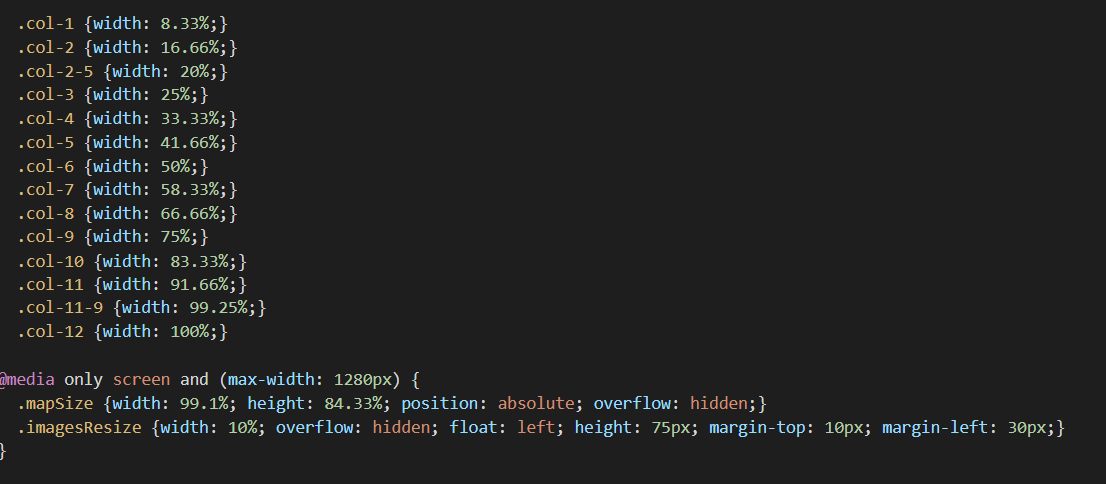
The front-end, which is in regards to the forest monitoring project, is a web-based application in Electron JS (which translates it into a desktop application) made to monitor the afforestation of several regions. It is done using HTML, CSS and JavaScript, and this README will explain the relevant parts of each of these three components.



*Fig: An overall overview of the application*

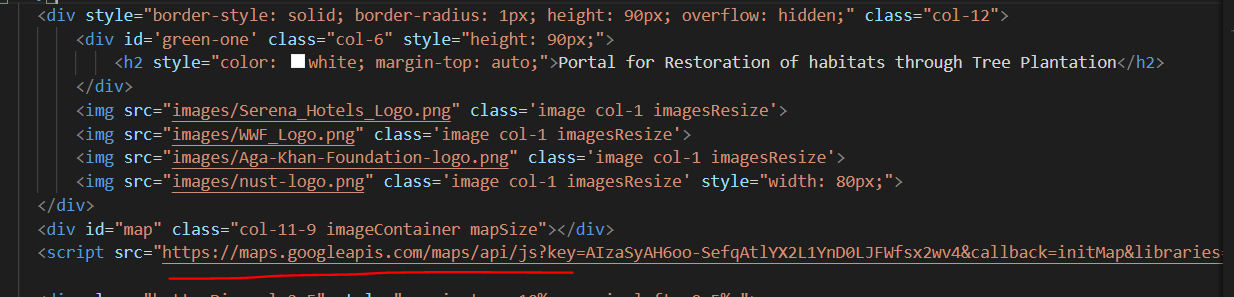
## The HTML and CSS

The HTML portion is fully responsive. However, instead of using Bootstrap for responsiveness, we created our col and row classes manually in CSS.



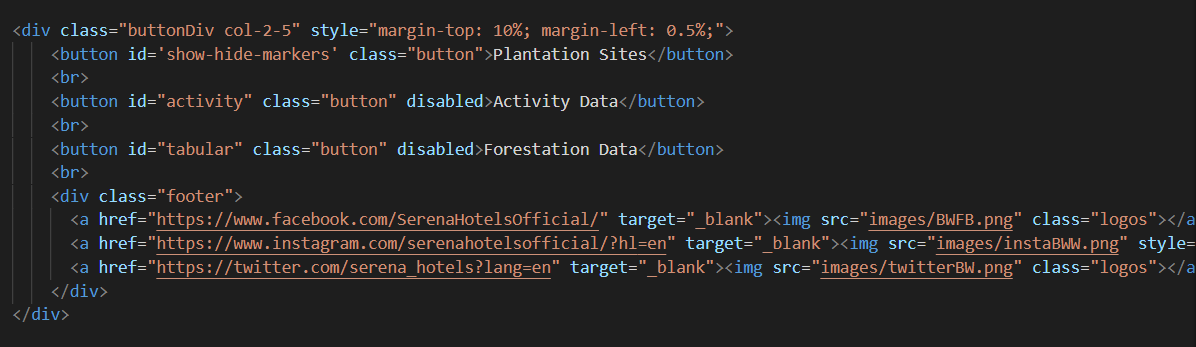
The header of the page is quite simple. One the left side, there is a div with the title in it; “Portal for Restoration of Habitats through Tree Plantation”, and on the right, there are divs that contain the four logos of the companies in collaboration of this project. The image dimensions and such are adjusted so they look natural even in windowed mode.

Using an electro property, the windowed mode goes to a minimum resolution of 1280x720.

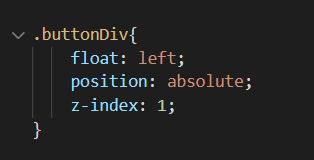


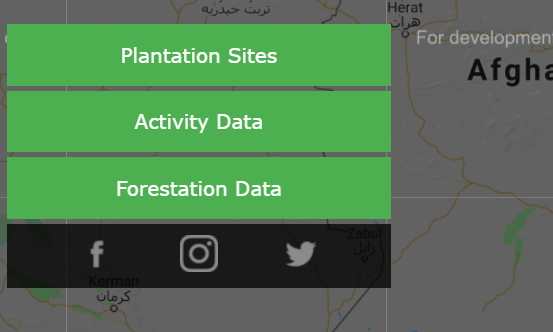
In the above image, the import of Google Maps API can clearly be seen. This will be elaborated more in the JavaScript section.

Next is the buttonDiv. This div has some special properties that keep the buttons fixed in place above the “footer”, which contains social media links to Serena Hotels.





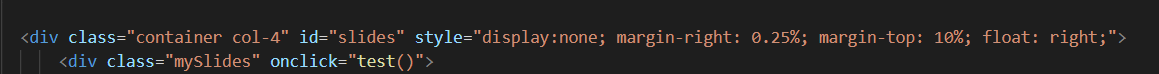




The “slideshow” is something we use for the functionality of the button “Activity Data”. What this button does is simple; it changes the state to “activity”, wherein, if we click on any marker, it will bring up satellite imagery of the location with different time-gaps. This imagery shows the difference in the forestation of the area over time.



The slides are placed in a container with a relative position, and then we define different classes for the captions and the images themselves. The images are all of the same size, and the position of the container is always to float on the right with specific percentage margins.



OnClick, the function that executes allows us to go into modal view (more on that later). On the slides, there are two arrow keys, which also, via a JavaScript onClick function, move the images, allowing us to go back and forth.



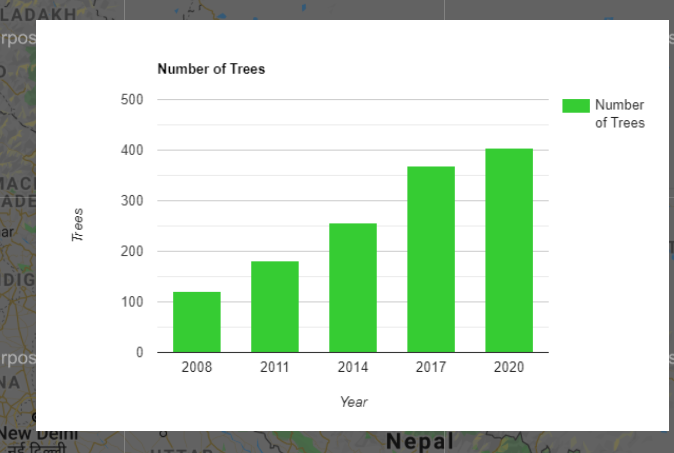
The visibility of the whole thing is hidden by default, and only when we click on the respective marker, will it become visible.

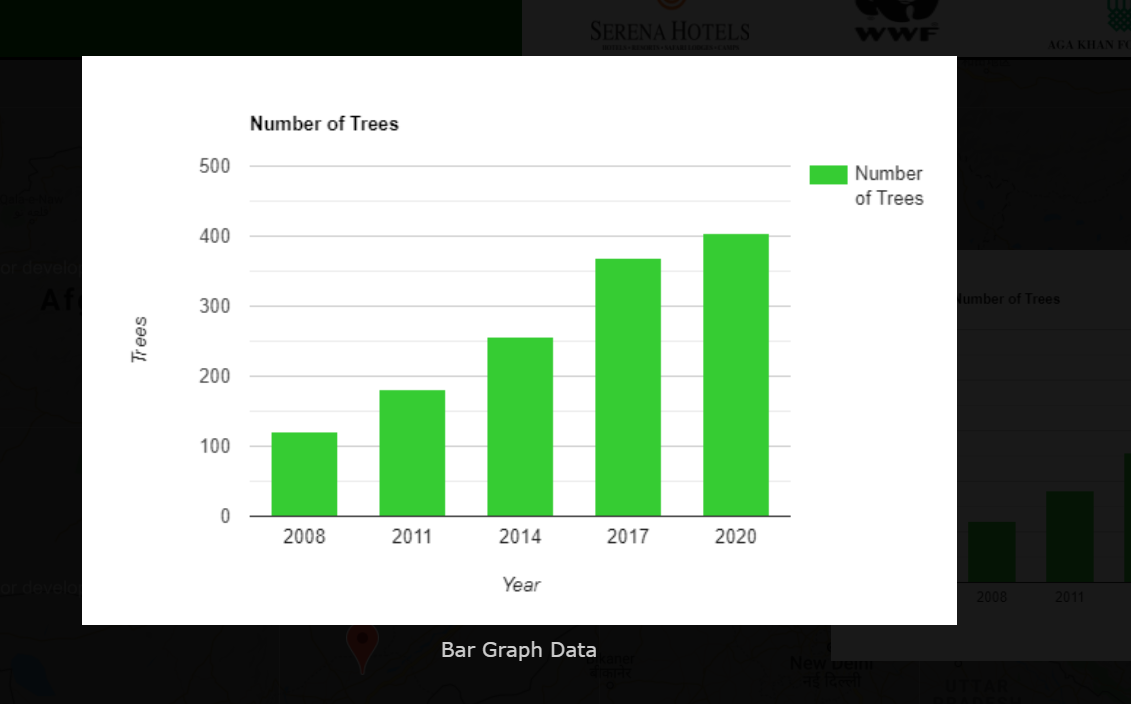
Finally, we come to modal view. Via a function, we implemented a “zoom” effect. The images on each slide are clickable, and when clicked, they will be zoomed in on, and become visible on a higher z-axis plane.



Next is the third button, “Forestation Data”. This shows tabular/graphical data, using the exact same logic as the slide-show. The only difference is the state of the application. When clicking on this button, it changes to graphical state, and the slide-show of satellite images is instead replaced by graphical data.

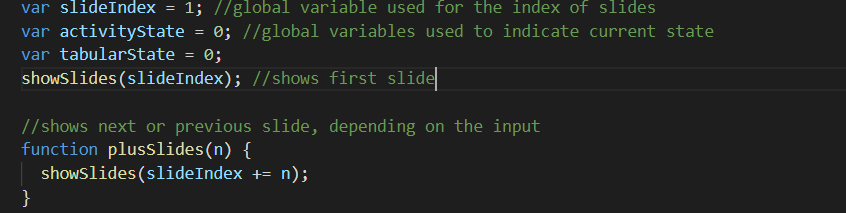
This data also has a modal view, again, using the exact same logic and functions.



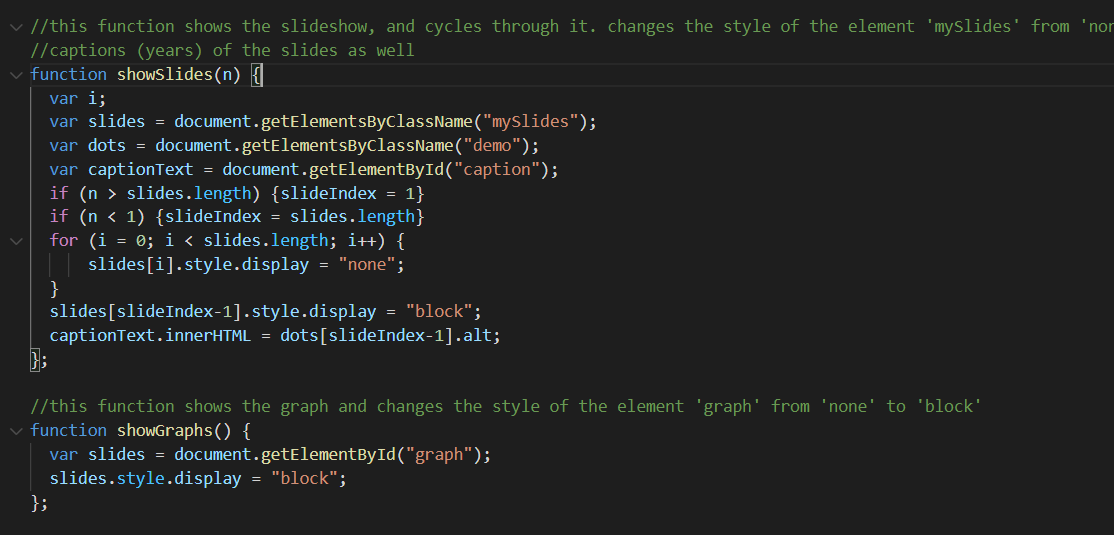


## The JavaScript

The most important part of the application is the JavaScript running it. As such, there are two files; index.js and main.js. main.js is the Electron JS main file, which is responsible for linking the application to electron, and creating the electron window. This is all very standard code, with no “logic”, or at least of our own. It is simply a default file that must exist for the program to run in electron. There are some configurations possible, such as hiding the electron menu, creating menus and submenus, toggling dev-tools, and fixing the window minimum resolution to 720p.

In index.js, the first few functions are for implementing the slide-show of satellite imagery. 

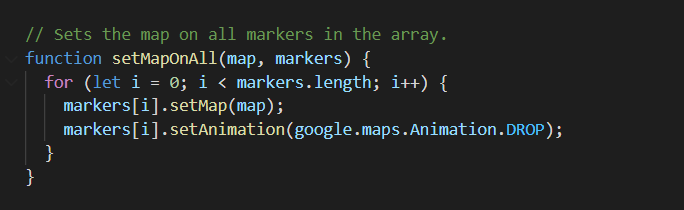
The global variables act as described. We use an index for the images, starting from 1 and going to n (where n is the number of images). The plusSlides function calls to showSlides, which is the standard slide’s function. Clicking on the arrow keys will either add or subtract the index. Their functionality is implemented in the HTML section.

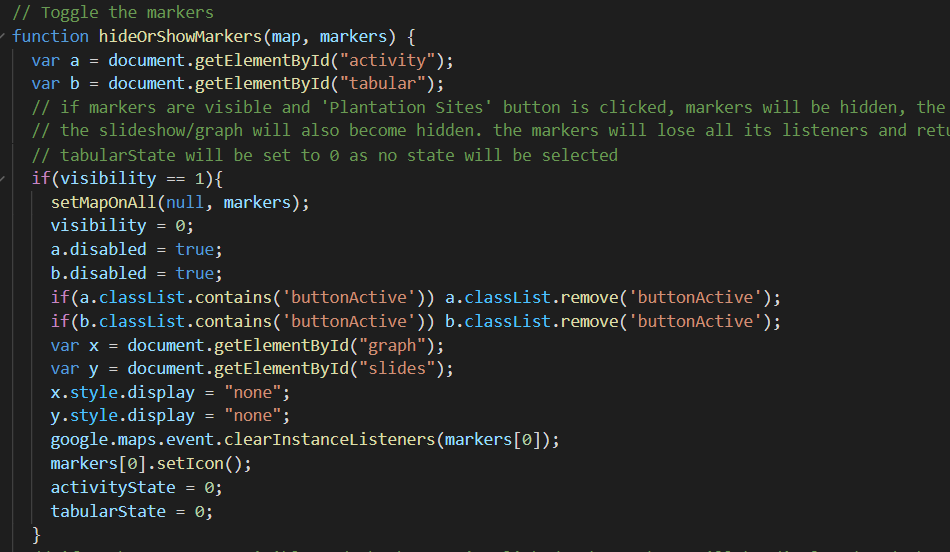


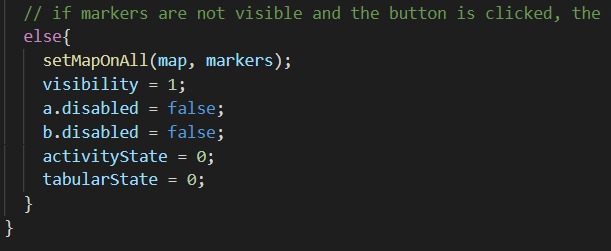
ShowSlides and showGraphs use the same logic. By default, both the slide and graph images are hidden, and these functions are responsible for changing them to “block” if they are hidden. The IDs previously assigned to these divs in the HTML section are called now.

After this, the remaining functions are all associated with, and use the Google Maps API. The variables and such are explained in the comments, as are the functions, though some information will be added here. First, we give a function to the first button. The map is toggled to on at runtime, and its coordinates are set so that it, by default, focuses on Pakistan at a certain zoom level by default.

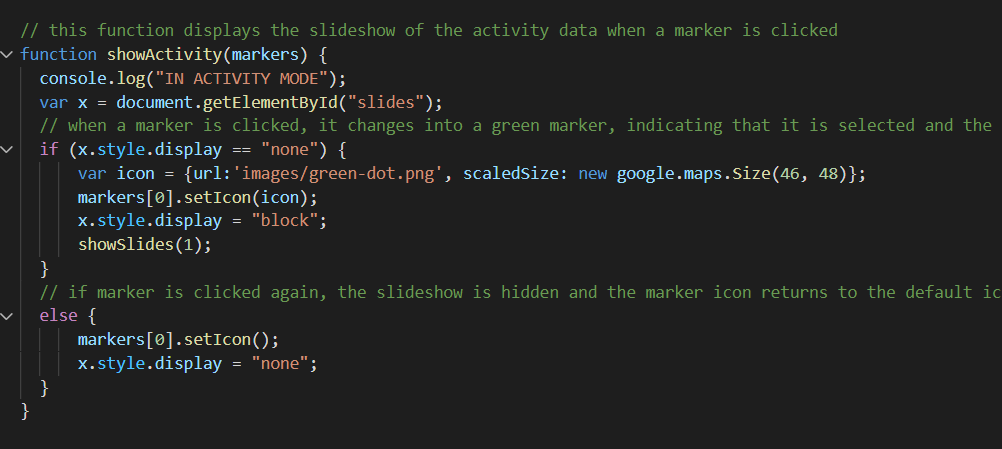
The “Plantation Sites” button, when clicked, toggles the markers on and off. These markers are also from the API, and we added a drop animation when they are toggled on. This applies each time they are toggled on, not just the first time.



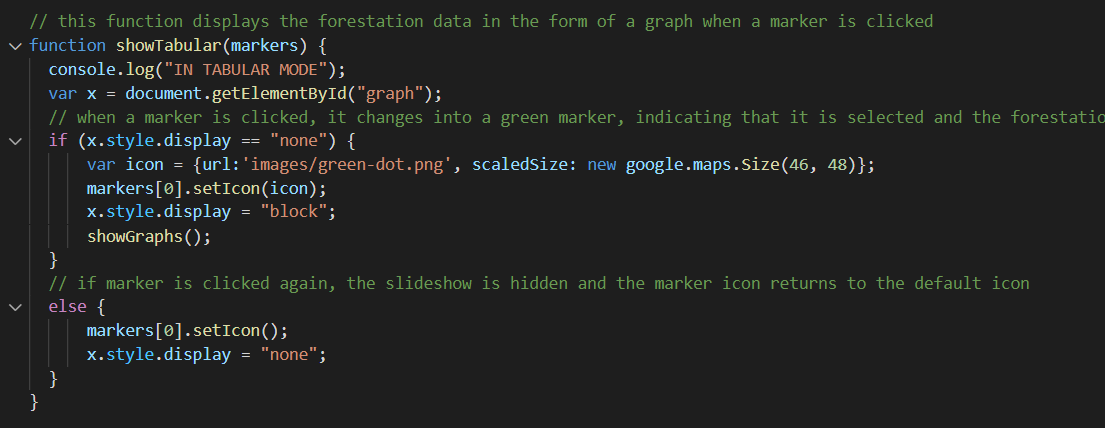




After this, the rest is functionality for the remaining two buttons. The logic here is that these buttons (Activity Data and Forestation Data) act as “states”. When the program is in activity state, the button is highlighted, and clicking on the marker toggles on and off the slide-show. Similarly, when the user clicks on Forestation Data, the state changes to instead show a bar graph with the trend of afforestation when the marker is clicked. Again, the marker toggles on and off the image. Clicking on Plantation Site resets both states and toggles the markers off.

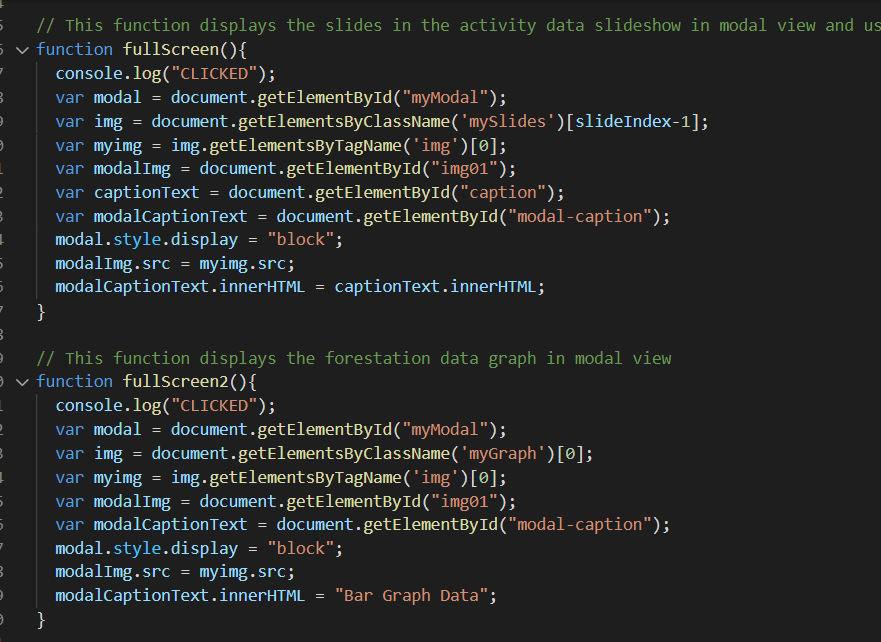


Both of these functions follow the above logic.

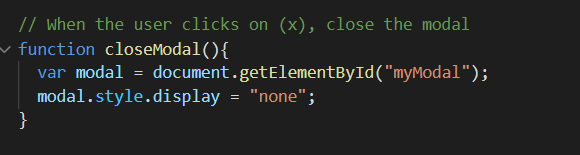


These functions are called within the functions that define the state (activityMode and tabularMode), and are used to toggle the images on and off by clicking the marker.

Finally, the last two functions are implemented for the modal image. These essentially turn the image, when clicked on, into a full zoom and shifts its z-axis by 1, darkening the layer of display below it to bring it into focus.



The final function is also related to modal mode. This function implements a click on the “X” symbol on the top right when we are in full-zoom mode. When clicked, modal is exited.



## How to run

To run, simply navigate to ForestMonitoring via terminal, after opening an environment in ForestProject. Once there, type “npm start” in terminal to open the Electron JS application. Remember that Node JS and npm must be installed beforehand, and the packages files must be present to run.