Mars "viewer"

The goal of this assignment is to use different ejs templating and view features. I have give you some code that will access some NASA open api data. This data returns a list of different Mars Rovers and photos they have taken. You will create an application that will list the various Mars rovers and let the user select one. Once selected the user will then be given a list of photos taken by the rover. Photos are organized by the Martian year. The first page of photos listed will be the last set of photos taken. Also there will be buttons on the page to let the user go to different dates.

The first method I have provided is getRovers(callback). This method will call the server and get a list of rovers and then will call the callback function with the rover data array. The return value is an array where each rover data set contains the following fields:
id -> id of rover
name -> name of rover
max_sol -> maximum sol date of rover

The second function gets a list of pictures for a given rover and a give date. getMarsPhotos(rover,soldate,callback) will return an array of photos for the given rover and given soldate. Note: The rover MUST BE LOWERCASE.

The return data is an array. I used the following fields for each array entry:
id
earth_date
camera.full_name
img_src

Please organize your views into:
header
SpecificViewpage
footer.

In the footer place a copyright statement and a button/link back to the main page.

Pages:
/ -> this page should call and list the set of rovers. Each rover should be a link to /mars/:rover
Name the SpecificViewPage index.ejs

/mars:rover -> this page should list the photos available for the latest sol date of the rover. Name the SpecificViewpage mars.ejs. The set of photos should be displayed using an HTML table. At the botton of this file put links to the previous, next and last dates. Use ejs logic to only show the next and last dates if not displaying the last sol date.

/mars/:rover/:date -> this page should list photos for the given date. You should also use the mars.ejs as the SpecificViewPage for this page. It should have the prev,next,last buttons.

- Call this folder "Assignment-12
- Keep this server running on port 3012
- submit a link to your git-repository.
- Submit a link to the running code on your instance

Here is the code to make the calls to my model. Place this code in a file called api.js.

```
------------------------------------------------------------
/*
 * Scott Campbell
 * model for assignment 12
 * cse270e
 * */

const axios = require('axios');

const host = "http://campbest.cec.miamioh.edu:3000/api/v1";

function getRovers(cb) {
    var url = host + '/rovers';

    axios.get(url)
            .then(response => {
                        cb(response.data);
            })
            .catch(error => {
                        console.log(error);
                        cb(err);
            });
}
function getMarsPhotos(rover,soldate,cb) {
    var url = host + '/photos/' + rover + "/" + soldate;

    axios.get(url)
            .then(response => {
                        cb(response.data);
            })
            .catch(error => {
                        console.log(error);
                        cb(error);
```

```
        });
}

function getUsers(cb) {
  var url = host = '/users';

  axios.get(url)
        .then(response => {
               cb(response.data);
        })
        .catch(error => {
               console.log(error);
               cb(error);
        });

}

module.exports.getUsers = getUsers;
module.exports.getMarsPhotos = getMarsPhotos;
module.exports.getRovers = getRovers;
----------------------------------------------
```

you will use this in your code as:

```
var rest = require('api.js');
rest.getRovers(function(data) {
….
```

also you will need to install the package axios using npm.