

**SS - Technical Coding Assignment - 2022**

1. **Create an application to parse the JSON from the API mentioned below.**  
   <https://s3.amazonaws.com/open-to-cors/assignment.json>

The JSON file contains 1000 records of products. Each product has the following 4 attributes

* Subcategory
* Title
* Price
* Popularity

**Task:**

1. Fetch the JSON file programmatically and store the data in the data structure of your choice.
2. Display the data in the presentation of your choice with Title, Price ordered based on the descending popularity.

**Deliverables:**

* Hosting URL on Heroku / Github pages
* Github repository link to your solution.
* Time taken to complete this

Solution:

{

    "Prdouct Name": "Boat",

      },

    "Title": [

        {

            "type": "Airpods",

            "model number": "XYZ-!@#123-ABC"

        },{

            "type": "speaker",

                    }

    ]

{

“Price”: “599.00 inr”,}

{

“Popularity”: “4.7”,}

}

}

2. **Create an application to parse the JSON from the API mentioned below.**

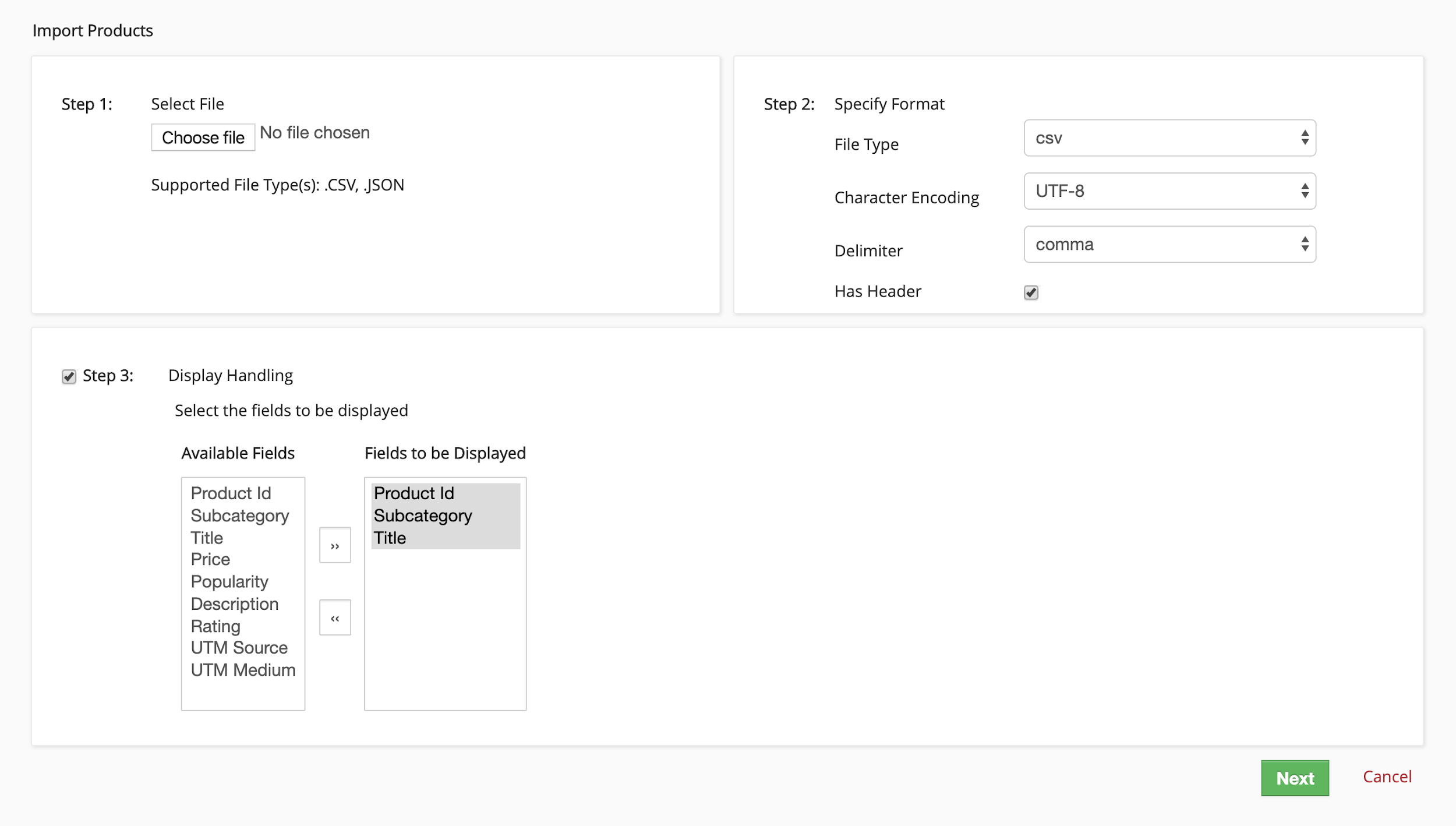
<https://s3.amazonaws.com/open-to-cors/assignment.json>

The JSON file contains 1000 records of products. Each product has the following 4 attributes

* Subcategory
* Title
* Price
* Popularity

**Task:**

1. Download the JSON file create the following UI to import the data on the browser and display the data (file : [Import\_products\_Screen\_1.png](https://open-to-cors.s3.amazonaws.com/Import_products_Screen_1.png)).



1. The display handling option controls the display of columns in the table in which data is displayed. It allows a multi-select functionality and clicking on the “>>” and “<<” buttons adds and removes the selected option from the Available Fields List to Fields to be displayed List and Vice Versa
2. Display the data in table format of your choice with Title, Price ordered based on the descending popularity.

**Deliverables:**

* Hosting URL on Heroku / Github pages
* Github repository link to your solution.
* Time taken to complete this.

3. **Create a basic frontend of a login dashboard with the below mentioned functionality**

* The login page should look like below : (file : [login\_page.png](https://open-to-cors.s3.amazonaws.com/login_page.png)).



* The user name field should only accept an email format. The validation should happen through Javascript.
* The password field must be a masked field i.e. should not reveal what's being entered.
* The password field should not accept any special character other than @ and must contain an uppercase letter and a number.

**Deliverables:**

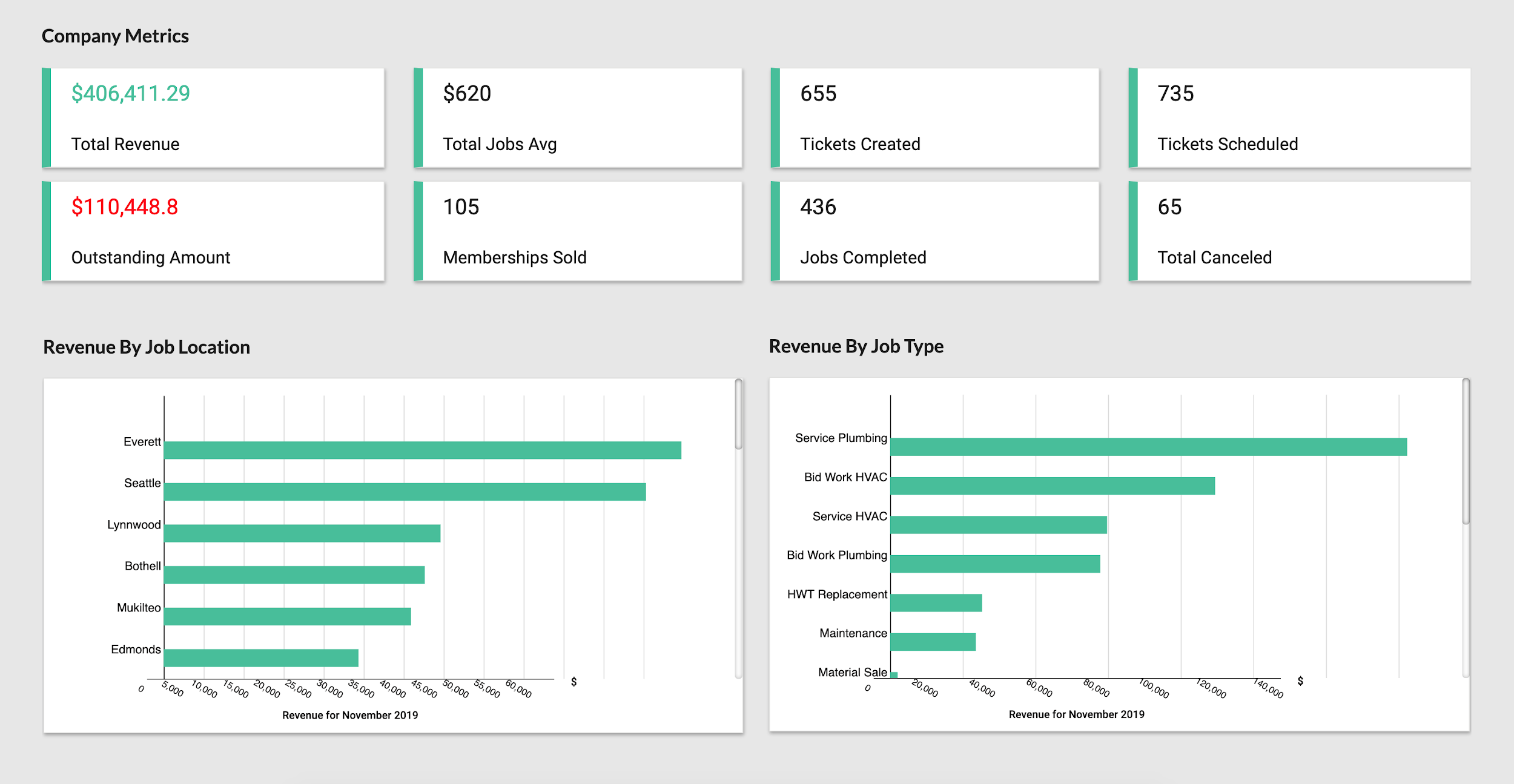
* Hosting URL on Heroku / Github pages
* Github repository link to your solution.
* Time taken to complete this.

**4. Create a basic frontend of a login dashboard with the below mentioned functionality**

* The login page should look like below : (file : [login\_page.png](https://open-to-cors.s3.amazonaws.com/login_page.png)).



* The user name field should only accept an email format. The validation should happen through Javascript.
* The password field must be a masked field i.e. should not reveal what's being entered.
* The password field should not accept any special character other than @ and must contain an uppercase letter and a number.
* Entering the password as - SmartServTest@123 should redirect the user to a dashboard page which must look like below. (file : [dashboard2.png](https://open-to-cors.s3.amazonaws.com/dashboard2.png)). Any other password should throw an error.



* Clicking on the Forgot your password link should open any email client to send an email to support@smartserv.io for resetting the password.
* All the components on the dashboard page must be actual components like charts and dropdowns. No image use is allowed.

**Deliverables:**

* Hosting URL on Heroku / Github pages
* Github repository link to your solution.
* Time taken to complete this.

***Feel free to reach out to us if you have any questions.***