## Task1: (React)

Table

Description automatically generated

* Create two pages /**users** and /**products**, these two pages will have the same re-usable components.
* https://dummyjson.com/ is the API that you will use, you can check the docs from this link https://dummyjson.com/docs to see what parameters you should send for filters and pagination.
* Use the **Axios** library for fetching the data.
* Use **Context API** to manage the state.
* Use the following colors:
  + Black: #322625
  + Grey: #ebebeb
  + Blue: #c0e3e5
  + Yellow: #fdc936
* **Filters:** 
  + **Page size:** the default value is 5, when click, it will open a dropdown that has the options: 5,10,20,50. When changing the value, it will send a request to the API and fetch the data based on the value that has been set. The pagination below the table will change based on that too.
  + **Search icon:** on click, it will show a text input beside it to enter the value, this will filter that result on the client side (no need to send a request to the API to filter the data). i.e. if you enter that value “19”, it will filter the data and show only the row that contains the value “19”
  + **Other filters:** will send a request to the API to fetch the data based on the value, the API doesn’t filter more than one field so when typing a value inside one of the filters, please reset the others.
  + **Data Table:** please fill the table with the data fetched (12 columns are enough, please include the fields mentioned in the filters)
* **Pagination:** when click on it, it will send a request to the API and fetch the data based on that.
* Please include the font family: Neutra Text.
* Please repeat the steps for the products page (filters are: **Title**, **Brand**, and, **Category** and tabs are: **ALL** and **Laptops**)
* Please upload your work on GitHub and share the repository link.

## Task2: (Laravel)

By using Laravel, please do the following:

* Create the following models: User, Project, Timesheet: the user Should have: first name, last name, date of birth, gender, email, password. The project should have: name, department, start date, end date, status. Timesheet should have: task name, date, hours.
* You need to create the proper relationships between the models, user can be assigned to more than one project, and each project can have many users, every user can log his timesheets to many projects, but each timesheets record should be linked with only one project and only one user.
* Create API endpoints for each model:
  + Create: POST /api/{model\_name}, you should pass the data of that model.
  + Read a record: GET /api/{model\_name}/{id}, it brings all the data from the select record.
  + Read all records: GET /api/{model\_name}, it brings all the records of that model
  + Update: POST /api/{model\_name}/update, it updates the record, you should pass the record id and updated data.
  + Delete: POST /api/{model\_name}/delete, you should pass the id of the record you need to delete, it should delete the timesheets records related to him.
* We need to have filtering system in “Read all records” endpoint, you need to send the filter value with the request, it should include the fields you need to filter with “AND” operation, so you may filter users by first name AND gender AND date.
* All endpoints cannot be accessible by public, you should build authentication for the users after login (please create the needed endpoints for register, login and logout) and if the user is not logged-in then cannot access the API.
* Please upload your work on GitHub and share the repository link.
* Please attach the database in SQL format.
* Please share the username, passwords, instructions to access the API.