

Task1: (React)

Home / **Users**

5 ▾ Entries | 🔍 | Name ▾ Email ▾ Birth Date ▾ Gender ▾

FIRST NAME	LAST NAME	MAIDEN NAME	AGE	GENDER	EMAIL	USERNAME	BLOODGROUP	EYECOLOR
est	repudiandae	Chance Towne	50	Male	test@TestTest	Donavon Fay	Joanie Reichel	Mohammed Bartell
autem	voluptatem	Jerod Gleichner	70	Male	test@TestTest	Beulah Mitchell	Johathan Moore	Mohammed Bartell
deat_test2dd	Wayne Anderson	Jerod Gleichner	13	Female	test@TestTest	Beulah Mitchell	Austyn Marvin	Beulah Mitchell
laborum	quaerat	Hilario Spinka	18	Female	test@TestTest	Nikolas Cronin	Austyn Marvin	Johathan Moore
excepturi	voluptatem	Beulah Mitchell	27	Female	test@testTest	Beulah Mitchell	Johathan Moore	Donavon Fay

← 1 2 3 4 5 ... 8 →

- 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 combine the project files into one compressed file (.zip)

Task2: (Laravel) – Optional

- Create API by using Laravel, you need to make the following endpoints:
 - **GET** /companies -> will return the full list of companies
 - **GET** /companies/{id} -> will return the data of the selected company
 - **POST** /companies -> to create a new company
 - **POST** /company/update -> to update the company
 - **POST** /company/delete/{id} -> to delete the selected company

Column	Type
id	bigint(20) unsigned <i>Auto Increment</i>
name	varchar(255)
sector_id	int(10) unsigned
website	varchar(255) <i>NULL</i>
phone	varchar(255)
email	varchar(255)
country	varchar(255) <i>NULL</i>
city	varchar(255) <i>NULL</i>
address	varchar(255)

- sector_id is a foreign key for another table that stores the sectors (Accounting, Arts, Fashion, ...)
- Sectors table only have id and name.
- The response should be in JSON format.
- The first end point (**GET** /companies) should support filtering the data by the field value and supports pagination. i.e.

/companies?pageSize=10&page=1&name="abc"

- Please combine the project files into one compressed file (.zip)