

**Enlace Video Youtube:** <https://youtu.be/oz_9t8phK_I>

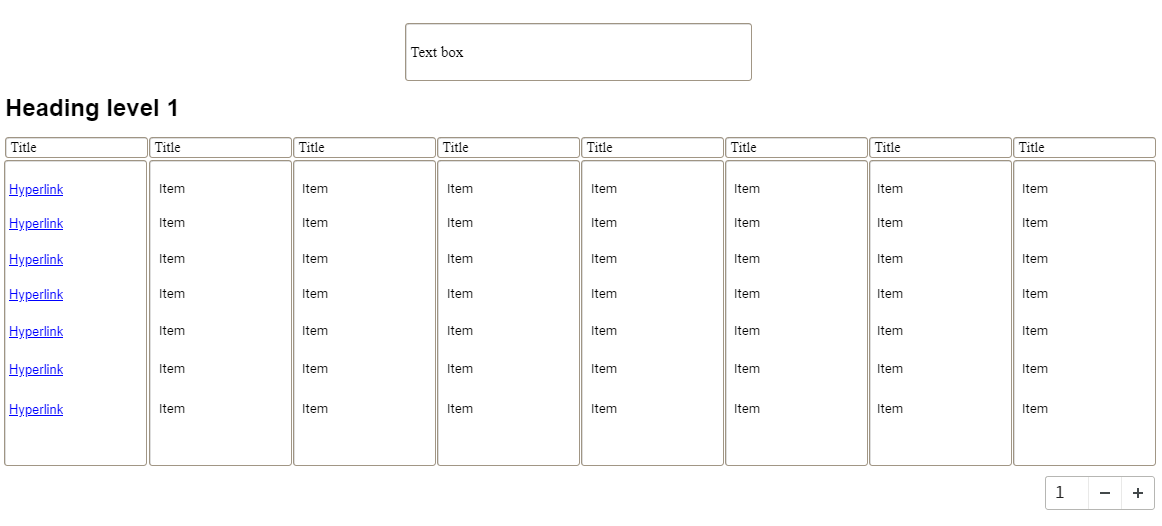
**Enlace Git-Hub:** <https://github.com/RommelZambrano/ESPE202205-T6-EuroByte.git>

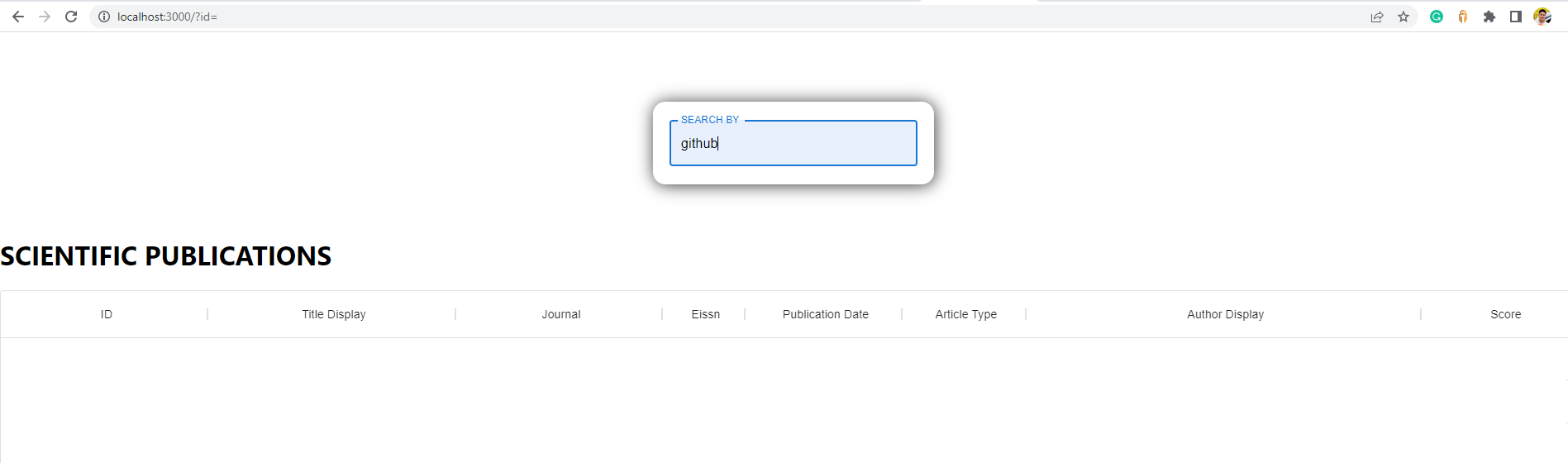
**1) PUBLIC API CONSUMPTION**

You have to implement one web page where the user will enter the search term. The user must select how they want to view the data. The developer is free to design the user interface of these five web pages, but you must use a framework to implement it later. The form (and its results) and the four pages are dynamically created with the retrieved data as follows:

1. **This page shows the tabulated documents (do not include the abstract) of each of the articles returned by URI 1, the titles of the columns are:**

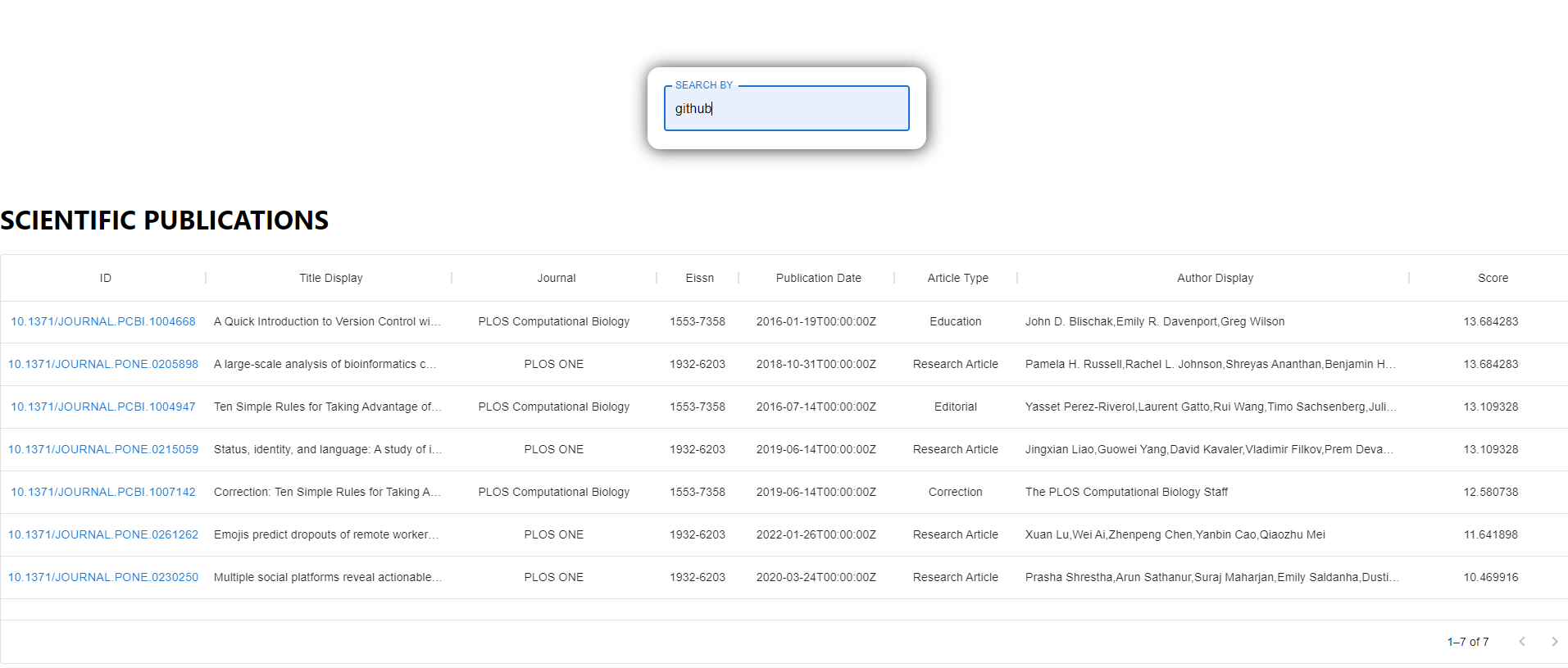
* id,
* journal,
* eissn,
* publication\_date,
* article\_type,
* author\_display,
* title\_display,
* score



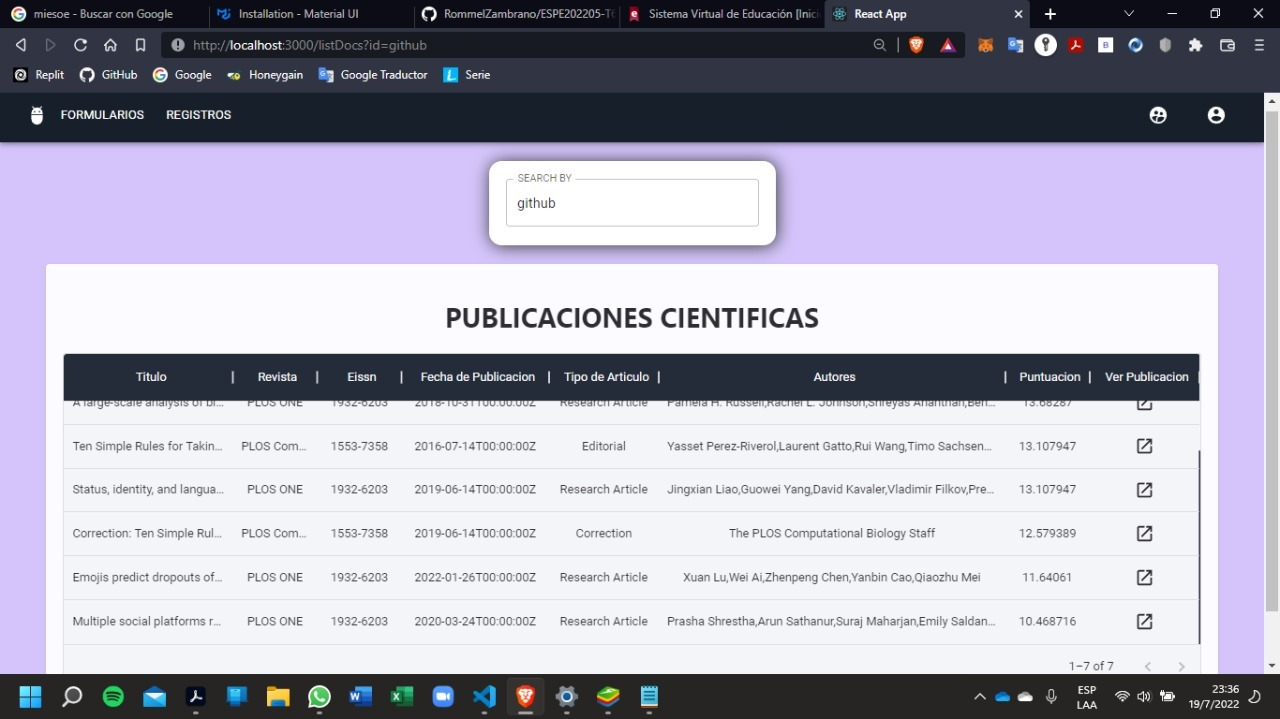


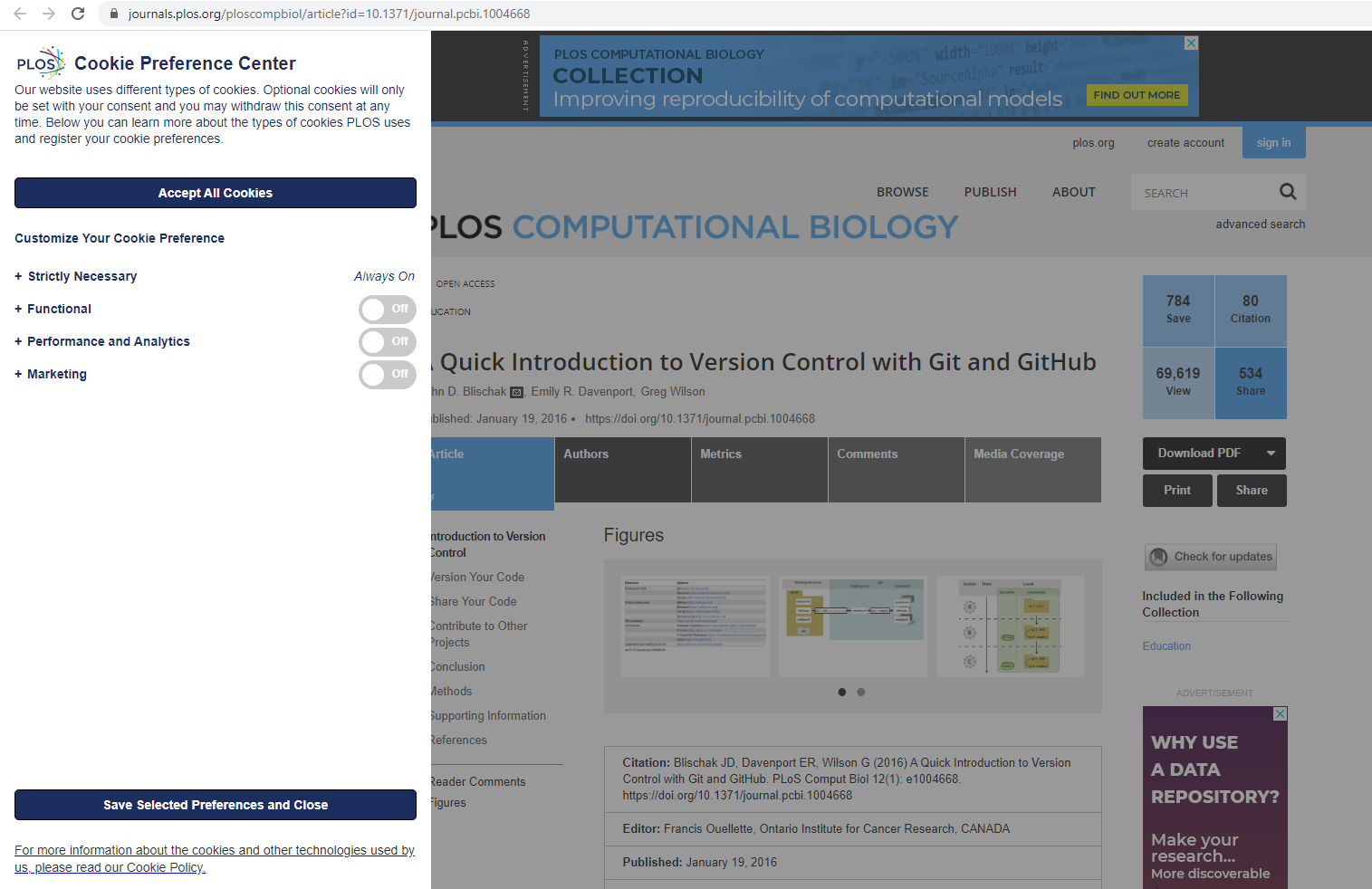
1. **On the previous page, for each title of the papers, program a link that displays the abstract for that paper on a new page as read from the URI.**

In the same screen the links for each article title were programmed, as well as the respective id to visualize the complete information. In this way, we comply with paragraphs one, two and three respectively.

****

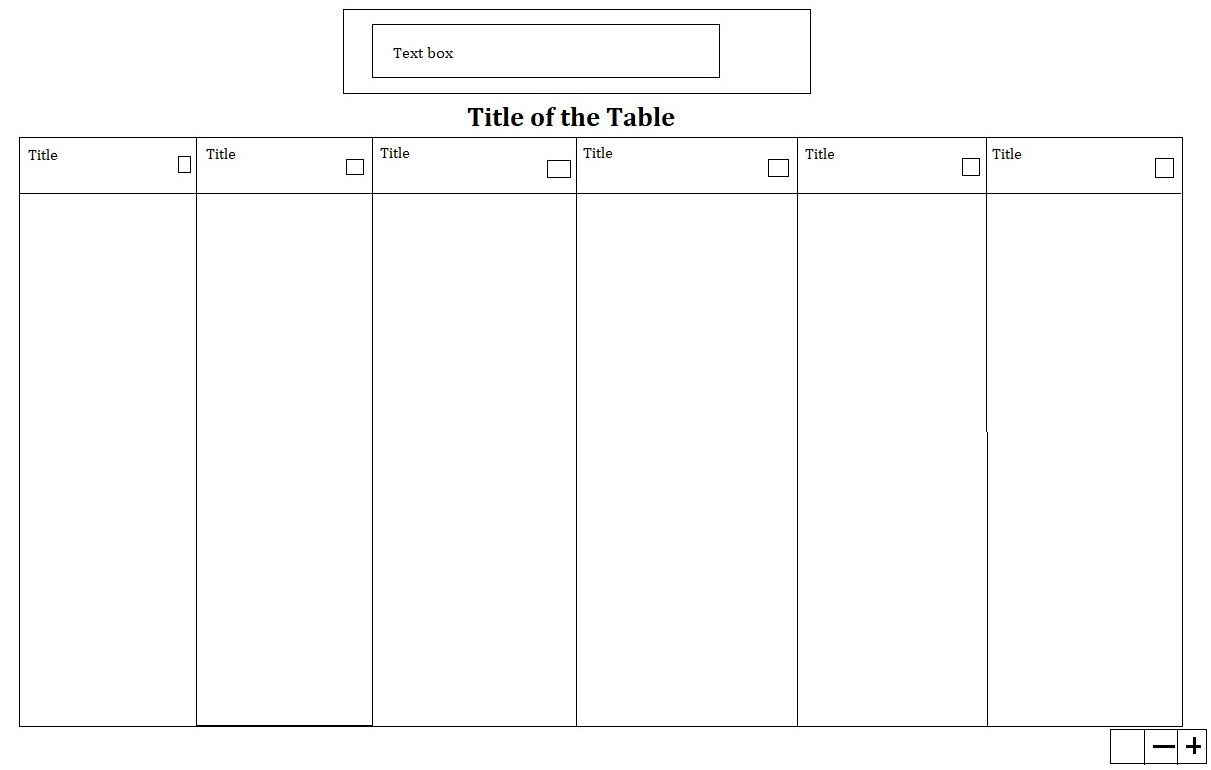
1. **A new page shows a table of every document identifier, with their respective id. From each id, a URL must be created so that the entire document can be viewed in a new tab (a new web browser tab), not in the same web application you are developing.**

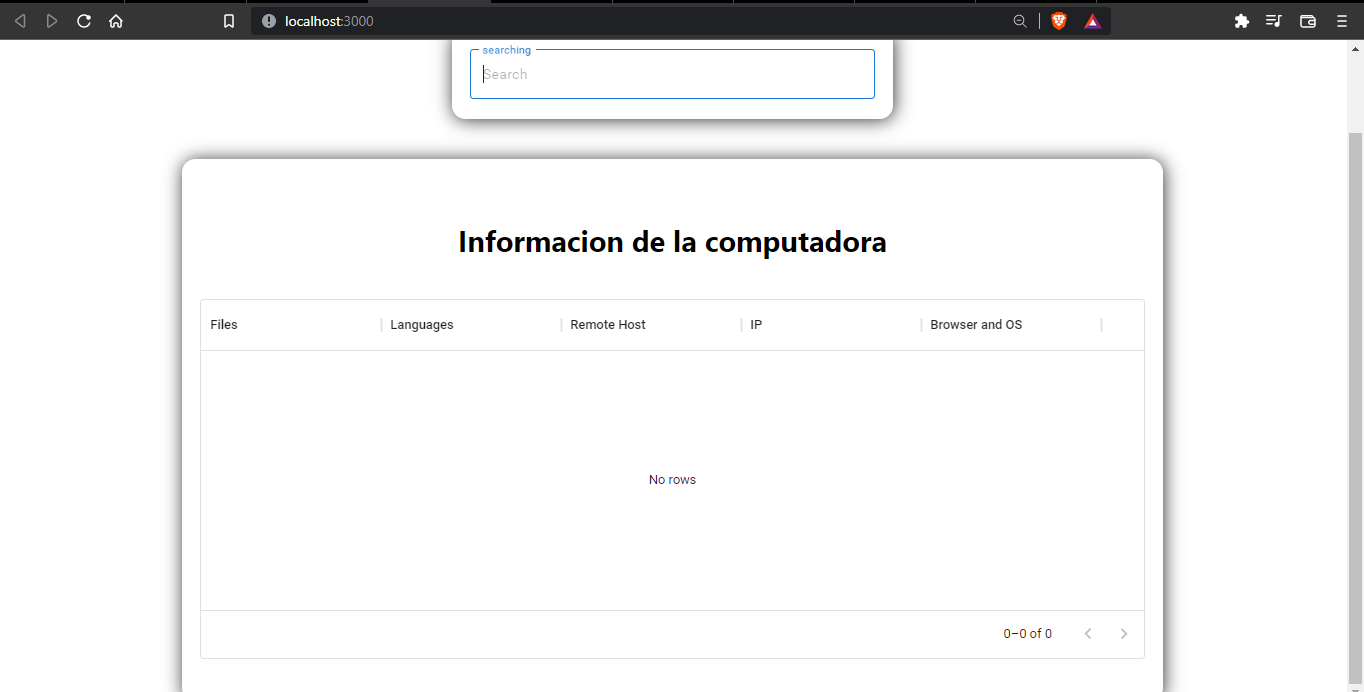




1. **You must also create a page that shows the information of the client computer from where you are accessing the API. For that, the public API that you are using provides the following URI https://httpbin.org/get. That URI returns the data of the client machine in JSON format. The data to be submitted are (in a web page, i.e., the data must be formatted as a web page, not JSON):**

* The files that are accepted
* The language
* The remote host
* The IP that originated the request
* The name of the browser
* The Operating System of the local machine





**2) USER INTERFACES FOR YOUR PROJECT**

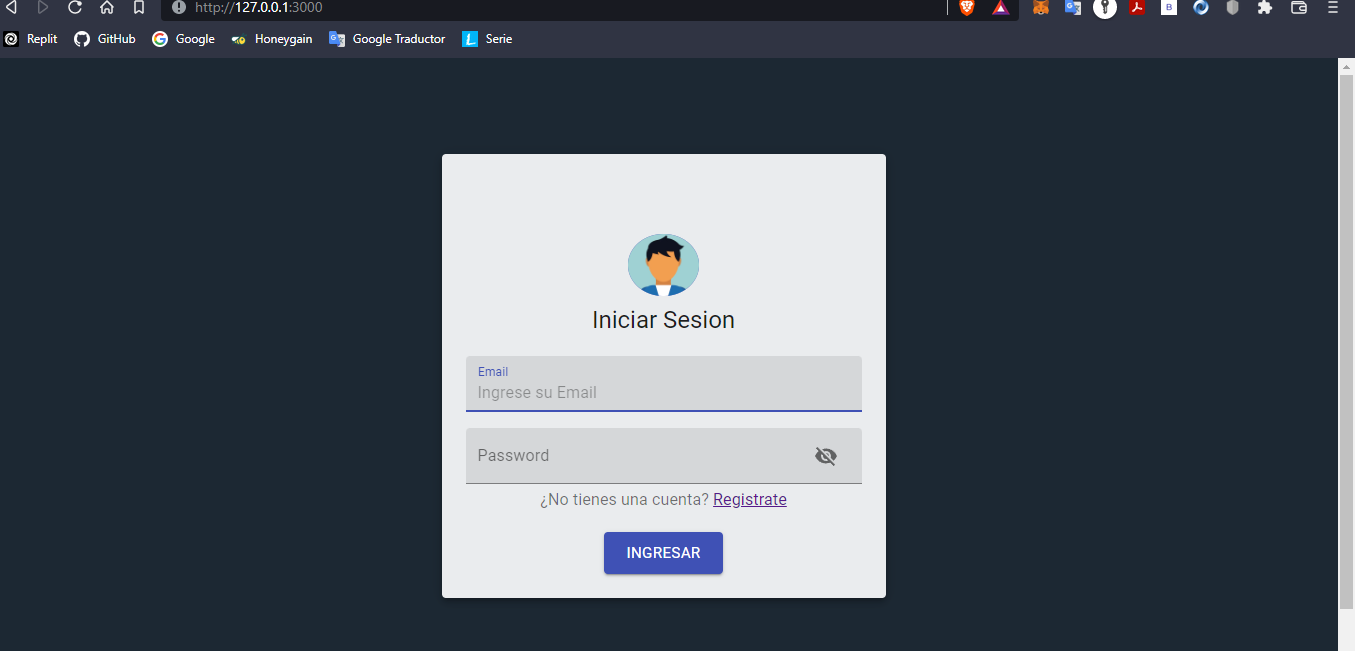
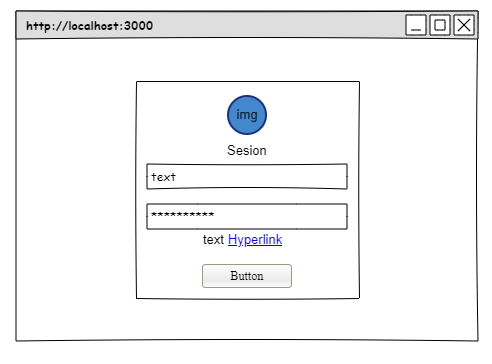
You must design the user interfaces for your project (your topic) using React, if you wish you can use another library or framework for Front-End Web design. From the designed and implemented pages, at least one of them must consume information from a GET type URI. Also, in this part of the implementation, you should learn the use of sessions and implement them at both the back-end and the front-end layers. To verify that the sessions were correctly implemented, you must open a URL and a URI of your project in a browser where you have not yet entered (logged in) to the system. If sessions are correctly implemented, the page and the URI should not display the requested information.

The development must be done as a team, this can be validated in the GitHub repository. The authorship will be written in every file, also validation questions about how each student did their part will be asked. In summary, each member of the team must program at least one of the user interfaces requested here.

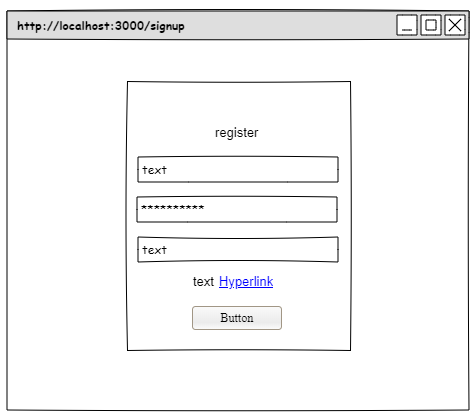
As a team, you must integrate each developer's solution without node\_modules (libraries) into the 06-Code folder.

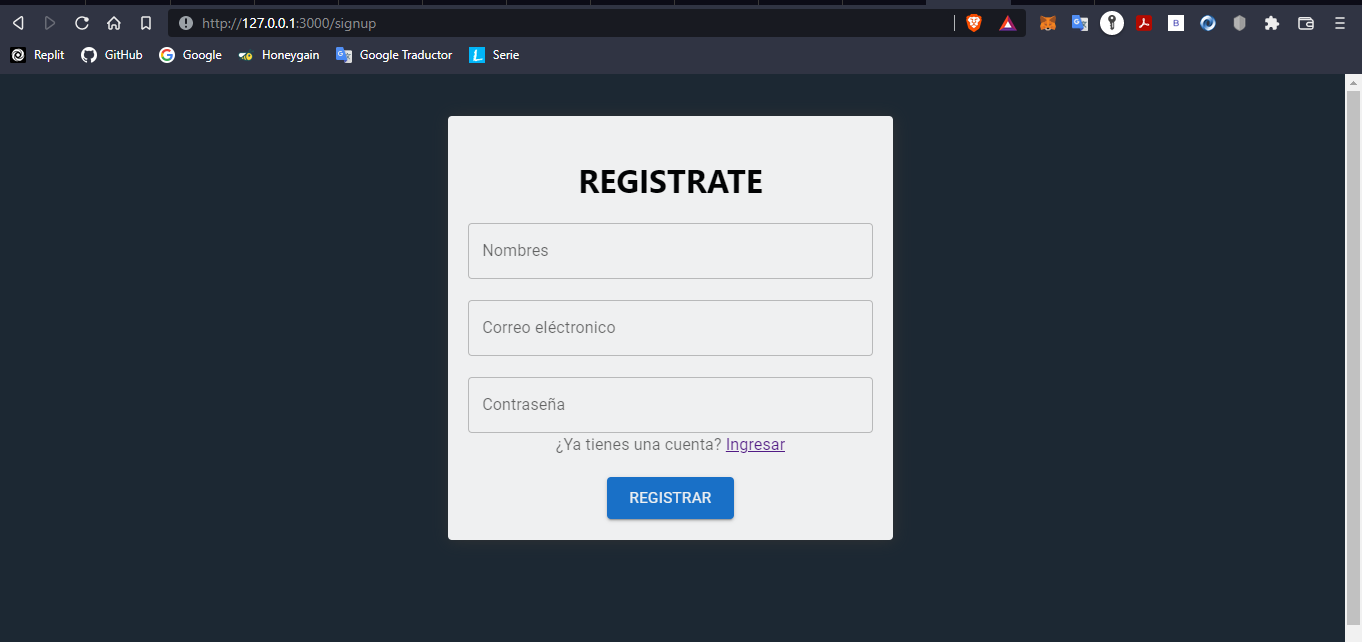
The execution of the web pages will be recorded in a video on YouTube by all the team members, the functionality is described while the corresponding searches of the public API are executed and when the data of the REST service is printed out, It should also show the data in JSON format produced by the URI, and the data as it is updated in the MongoDB Atlas database, you can use MongoDB Compass. At the beginning of the video, the names of the members, the name of the team, and the topic should be mentioned.

* **Login Page**

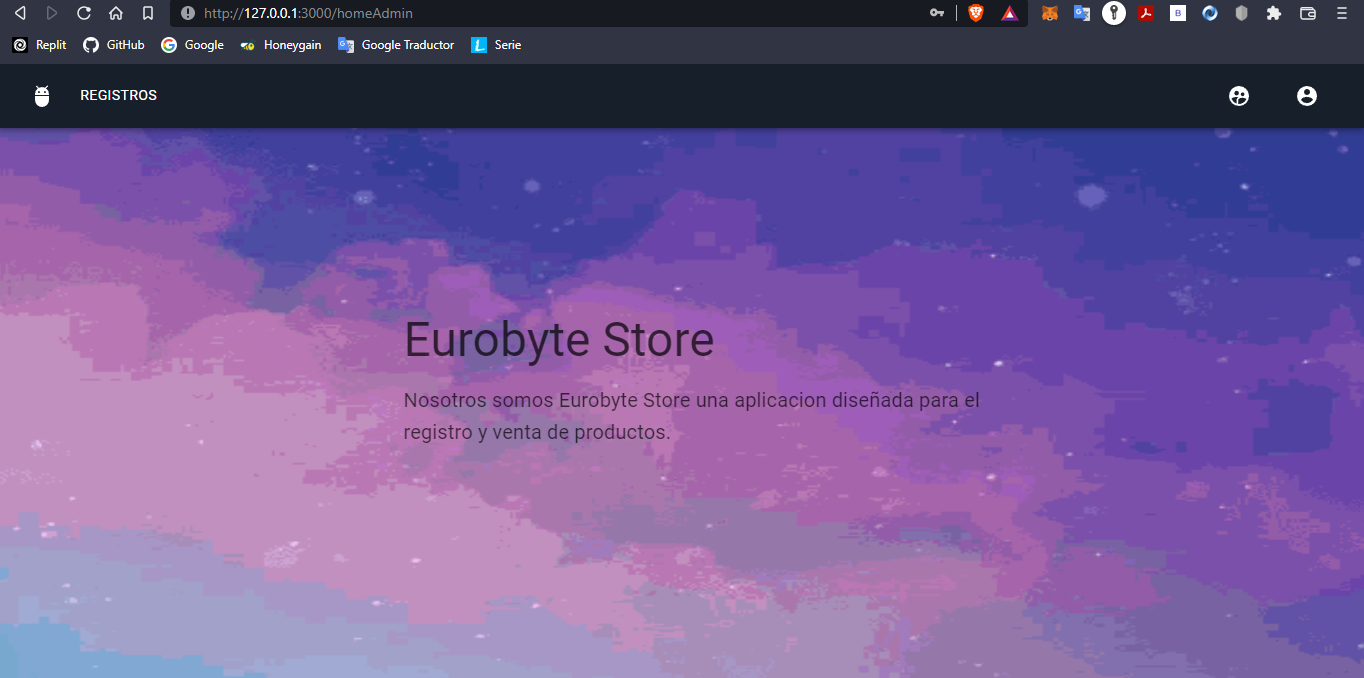
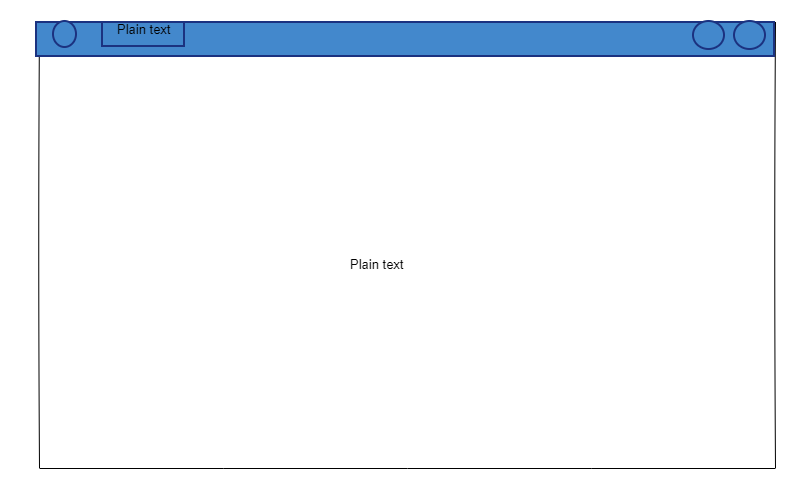


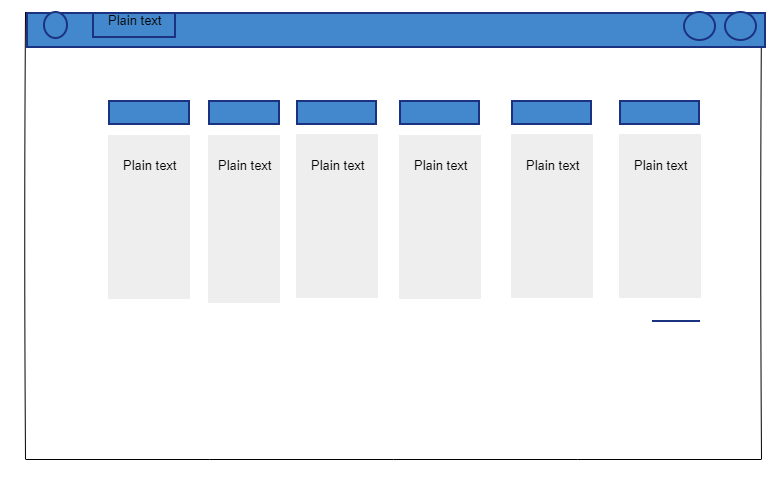
* **Register Page**

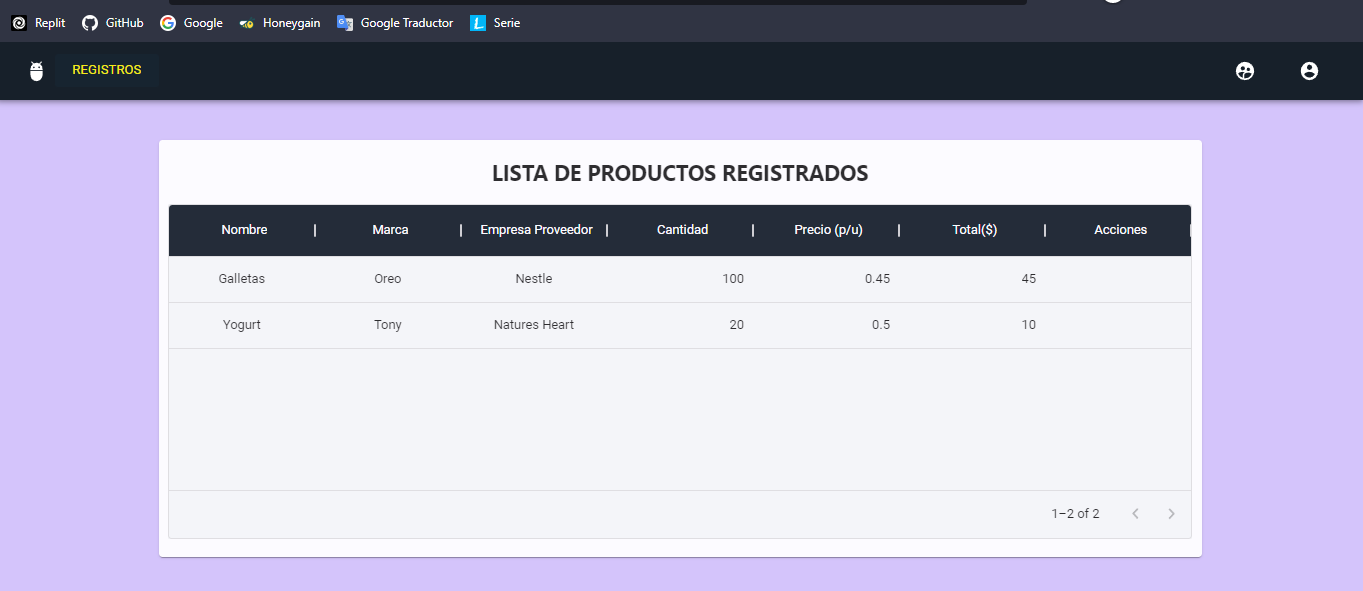




* **Home Page**







* **Consumer Api get product and users**

