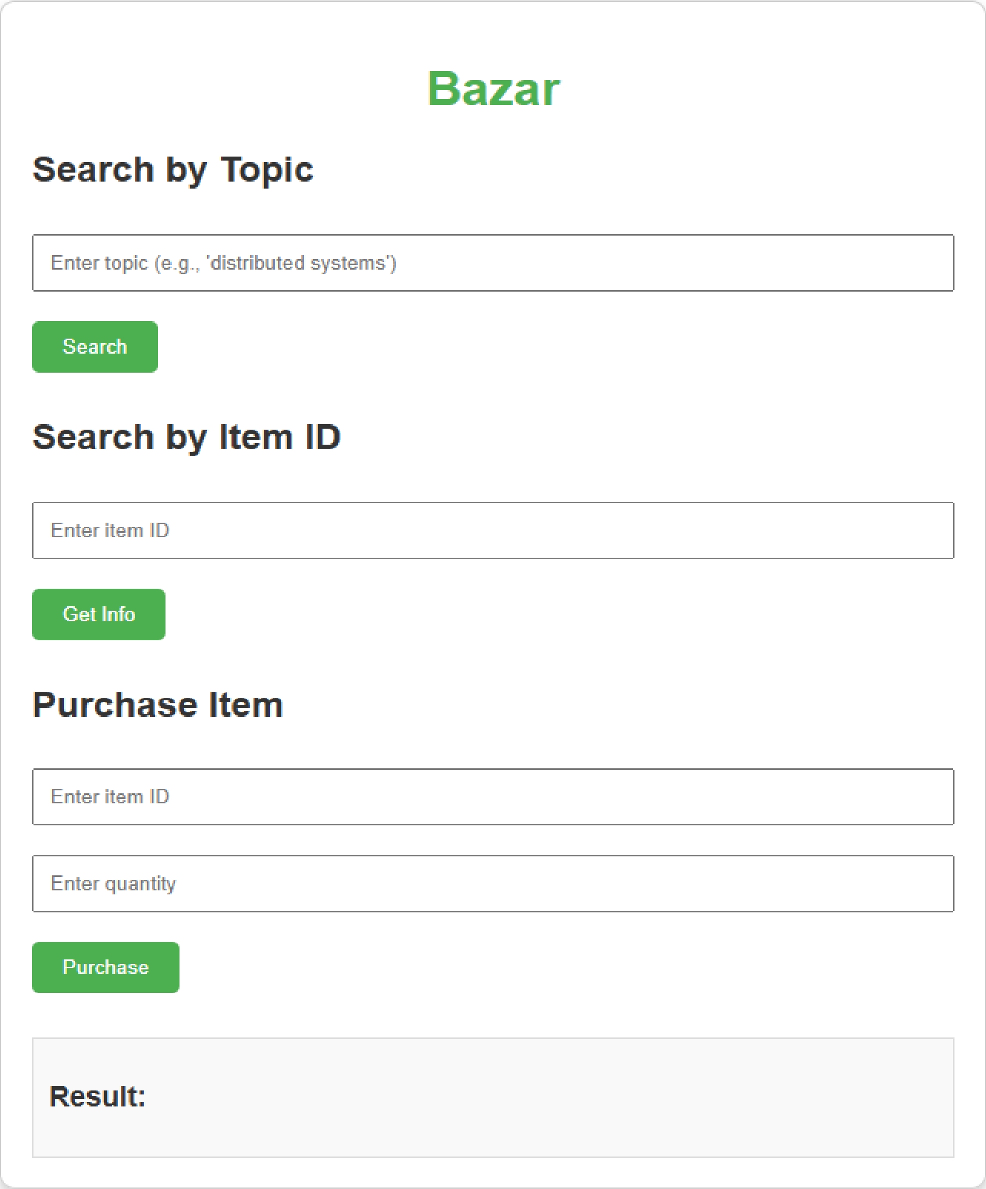
Project Output

- There are many ways to test the API end points, like postman for ex, but we created a user friendly page to ease the user interaction and testing.
- First, this is the page that shows on http://localhost:3002/ when the project is running.



Bazar

Search by Topic

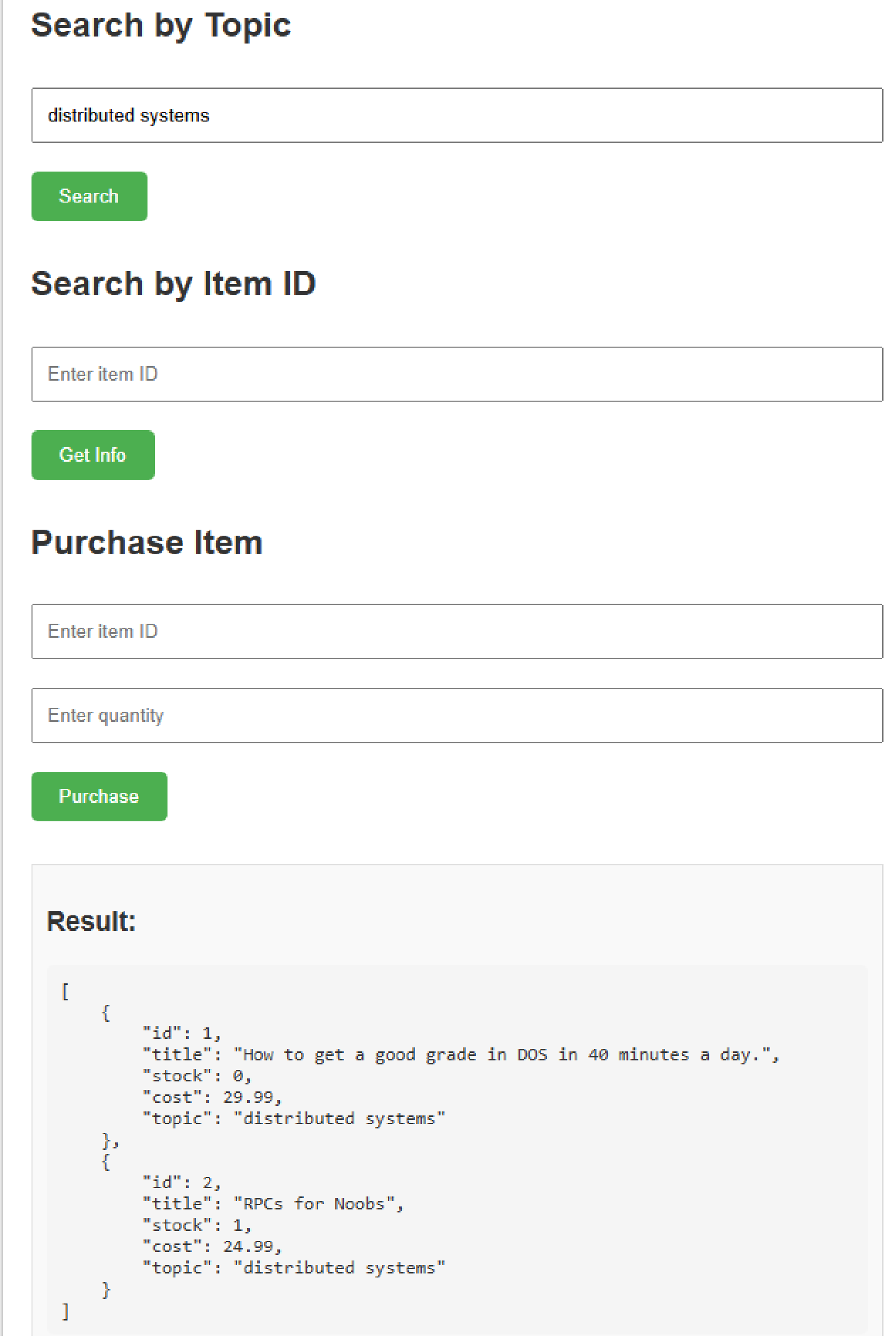
Search by Item ID

Purchase Item

Result:

1) Search By Topic

- For example let’s search for distributed systems topic :



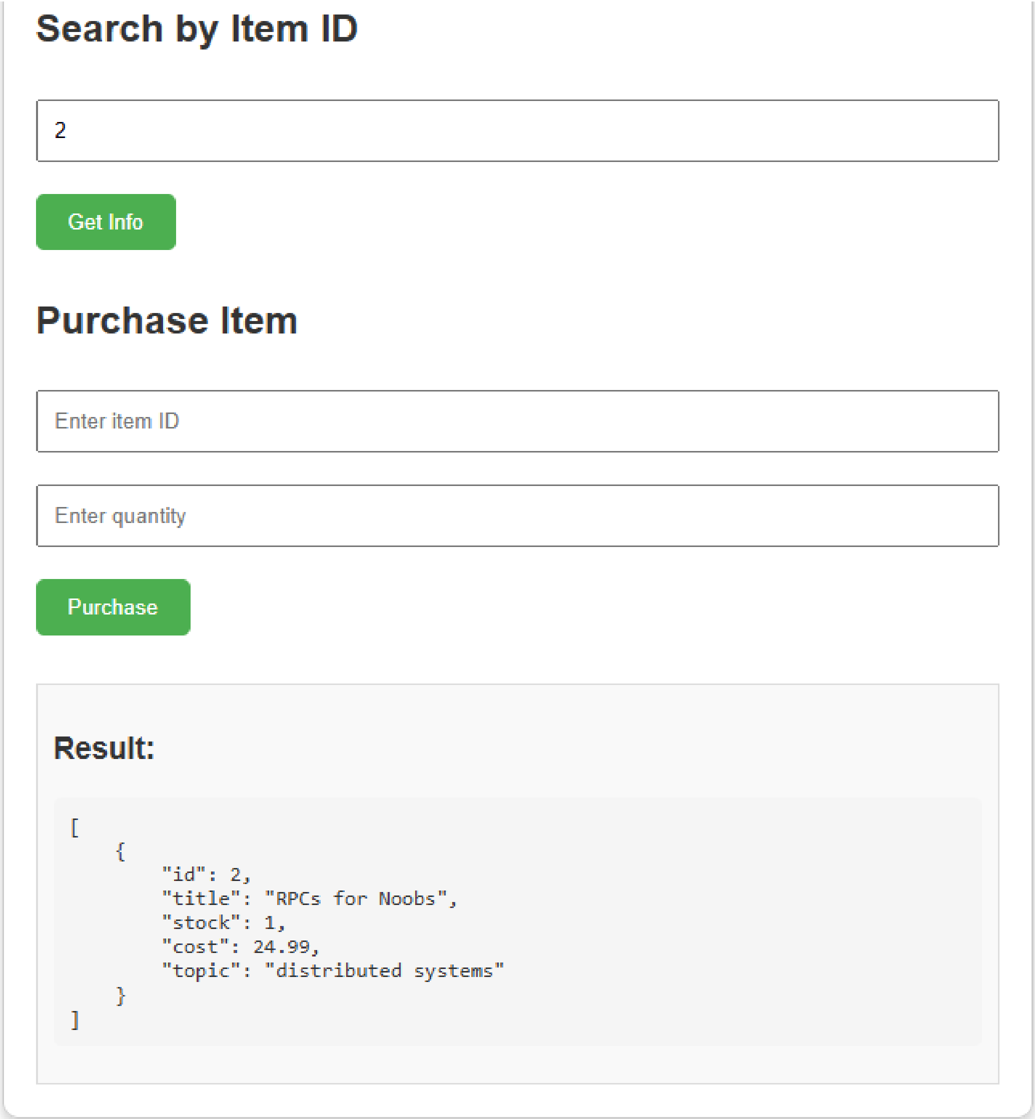
distributed systems

```
[
    {
        "id": 1,
        "title": "How to get a good grade in DOS in 40 minutes a day.",
        "stock": 0,
        "cost": 29.99,
        "topic": "distributed systems"
    },
    {
        "id": 2,
        "title": "RPCs for Noobs",
        "stock": 1,
        "cost": 24.99,
        "topic": "distributed systems"
    }
]
```

- The result is the books under this topic.

2) Search By item ID

- Now let’s search for a specific item, item with id 2 :



Search by Item ID

2

Purchase Item

Result:

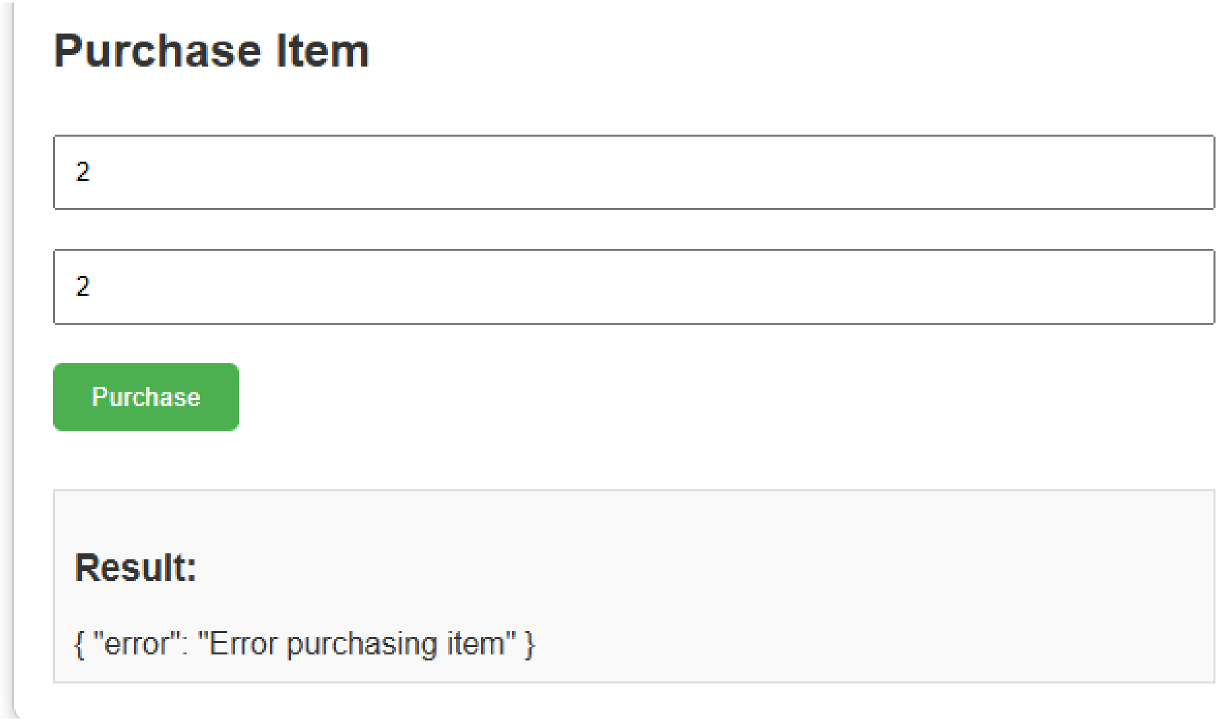
```
[
    {
        "id": 2,
        "title": "RPCs for Noobs",
        "stock": 1,
        "cost": 24.99,
        "topic": "distributed systems"
    }
]
```

- We get the info of that book as shown.

3) Purchase Item

- We have two cases :

1) There’s not enough stocks or quantity. Notice from the previous request that the stock for book 2 is 1.



Purchase Item

2

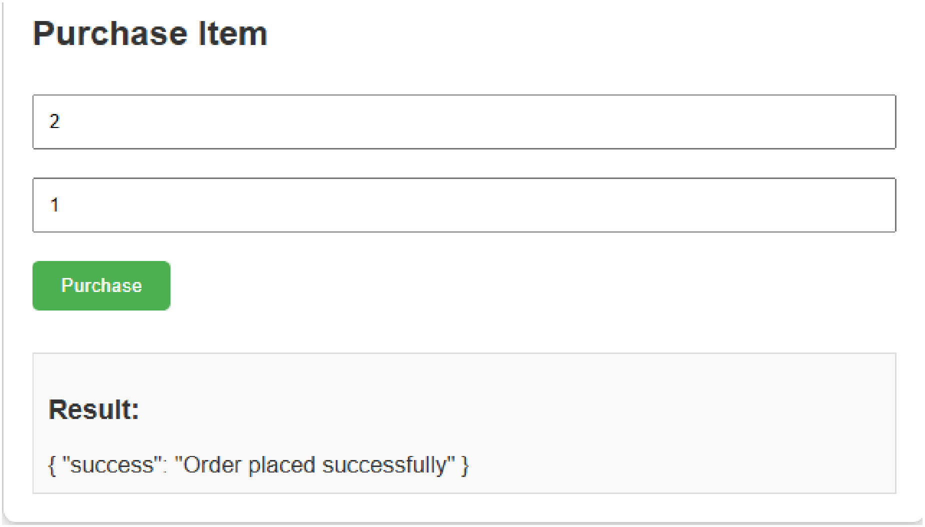
2

Result:

```
{ "error": "Error purchasing item" }
```

- We tried to purchase 2 items of book 2, so we got an error from the order server.

2) When there’s enough stocks.


Purchase Item

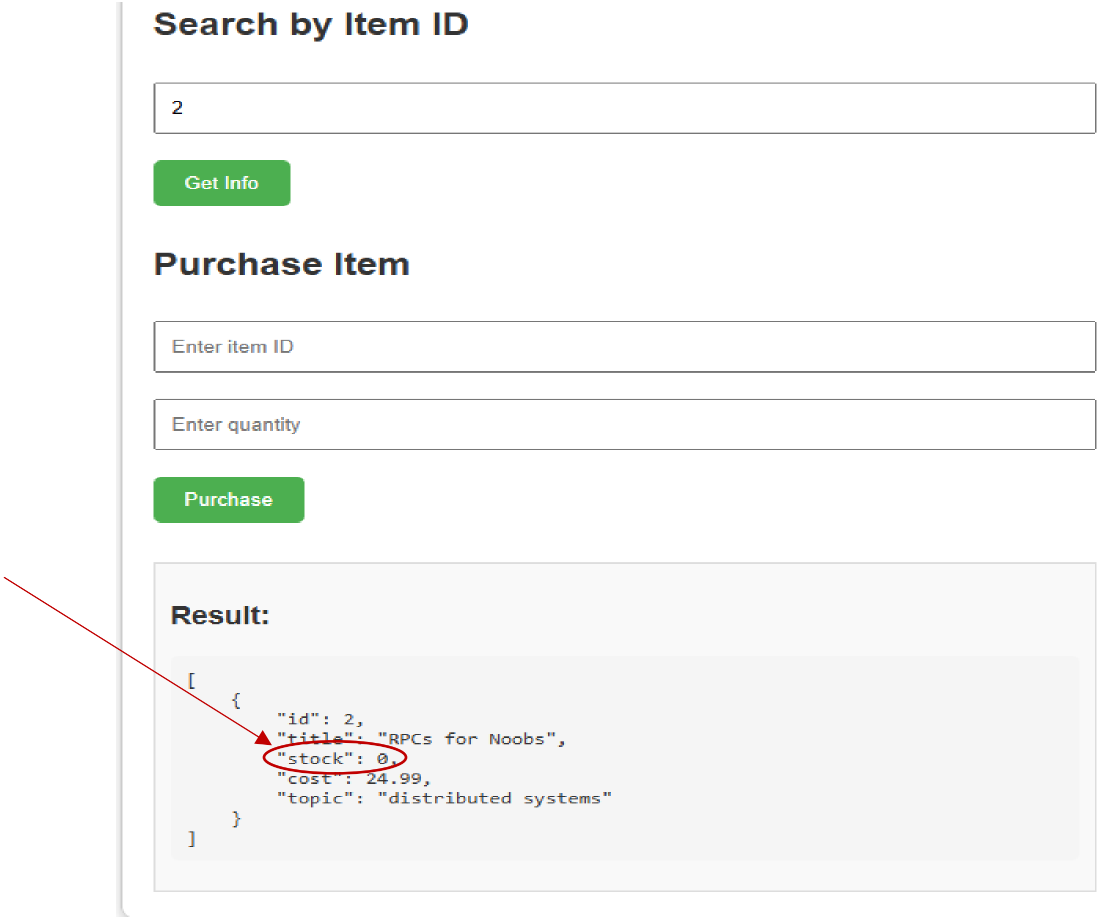
2

1

Result:

{ "success": "Order placed successfully" }

- The transaction is successful, and the stock is decremented by 1, so if we run the /info/2 we will get :


Search by Item ID

2

Result:

```
[
    {
        "id": 2,
        "title": "RPCs for Noobs",
        "stock": 0,
        "cost": 24.99,
        "topic": "distributed systems"
    }
]
```