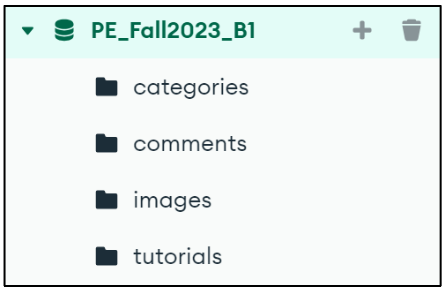
LAB 4

Requirements:

- Using **MongoDB Compass**, create a database named “**PE_Fall2023_B1**”. Create collections corresponding to the data in the **given** directory. Then, import data into the collections created in the database.



PE_Fall2023_B1

categories

comments

images

tutorials

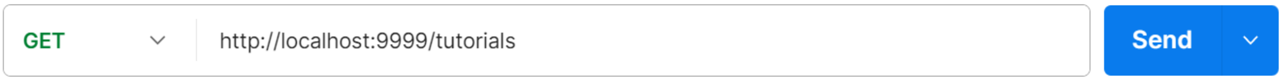
Question 1. (5 points)

In this question, you are asked to write an application (**back-end** given folder), that provide some API to manage: **Categories**, **Tutorials** and **Comments**.

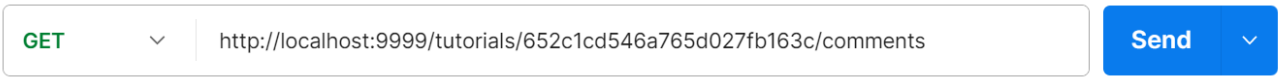
Note - Scored 0 points if:

- Not using database connection string in the **.env** file
- Not configuring the root path of API Web App at: **http://localhost: 9999**

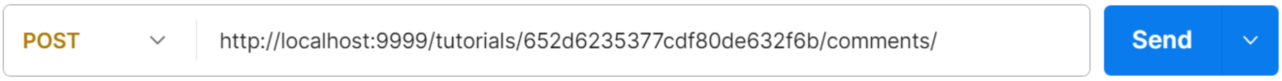
You are asked to build three APIs, as following:



GET http://localhost:9999/tutorials



GET http://localhost:9999/tutorials/652c1cd546a765d027fb163c/comments



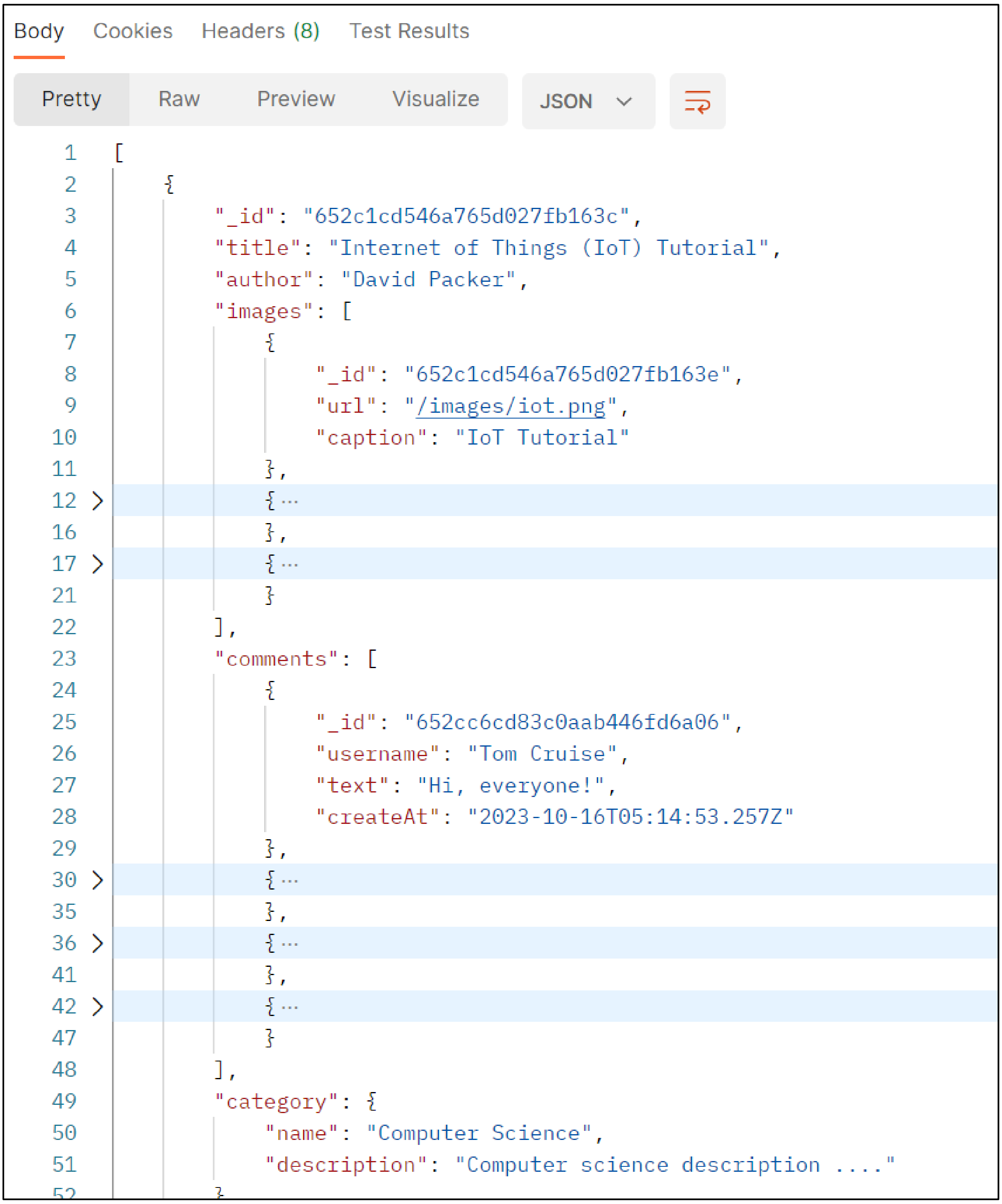
POST http://localhost:9999/tutorials/652d6235377cdf80de632f6b/comments/

1.1. (1.5 points)

The API at url: http://localhost:9999/tutorials return information of all **tutorials** in database, using GET method.

Each **tutorial** requires the information shown in Figure 1. Note that:

- [**images**]: Is an embedded attribute from the **Images** entities
- [**comments**]: is a reference attribute of the **Comments** entities
- [**category**]: is a reference attribute of the **Categories** entities



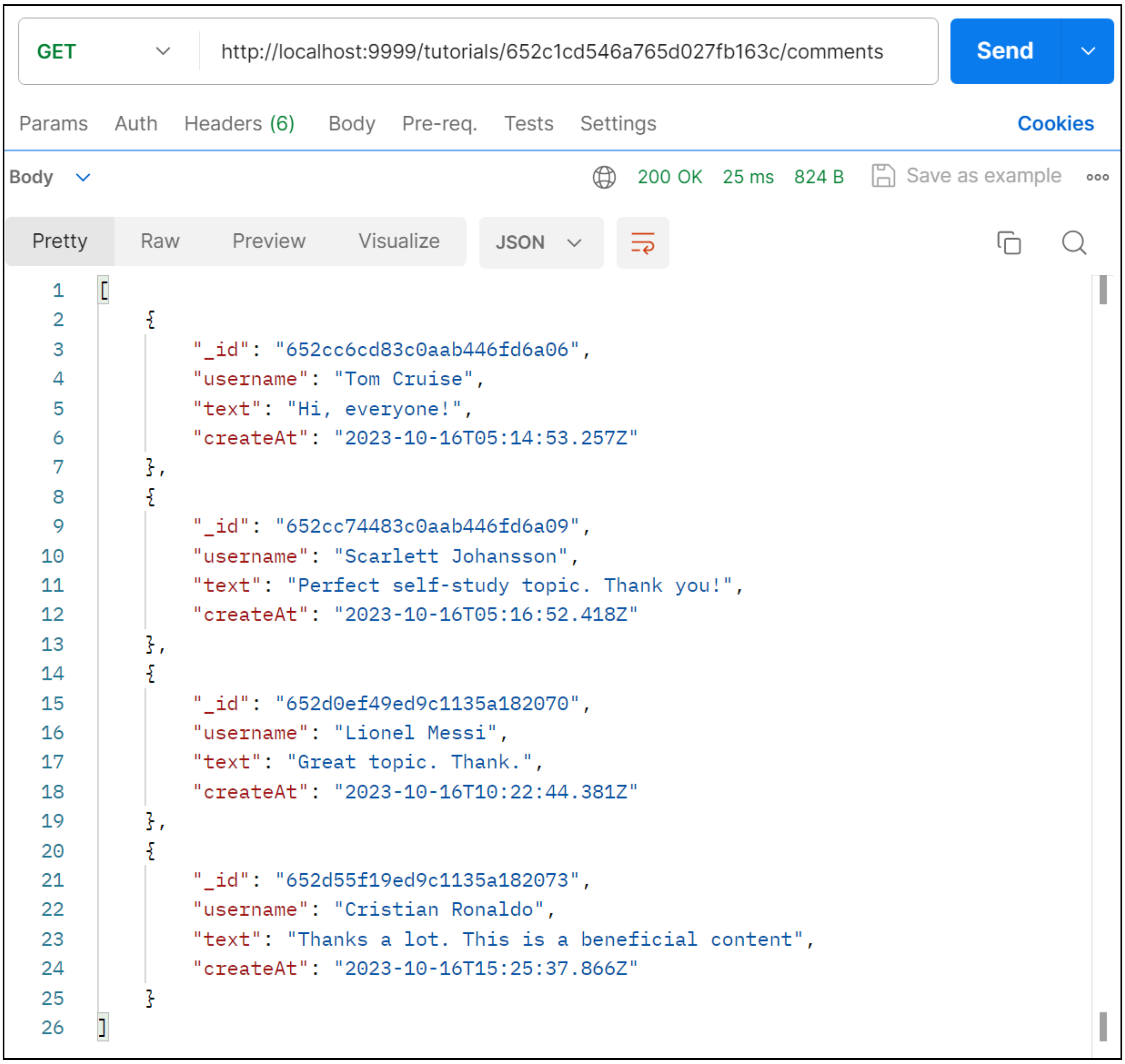
```
[
    {
        "_id": "652c1cd546a765d027fb163c",
        "title": "Internet of Things (IoT) Tutorial",
        "author": "David Packer",
        "images": [
            {
                "_id": "652c1cd546a765d027fb163e",
                "url": "/images/iot.png",
                "caption": "IoT Tutorial"
            },
            {…
            },
            {…
            }
        ],
        "comments": [
            {
                "_id": "652cc6cd83c0aab446fd6a06",
                "username": "Tom Cruise",
                "text": "Hi, everyone!",
                "createAt": "2023-10-16T05:14:53.257Z"
            },
            {…
            },
            {…
            },
            {…
            }
        ],
        "category": {
            "name": "Computer Science",
            "description": "Computer science description ...."
        }
```

Figure 1 – The result of API at http://localhost:9999/tutorials (using method GET)

1.2. (1.5 points)

The API at url: http://localhost:9999/tutorials/:id/comments (**:id** - is an ObjectId of the Tutorial) return information of all comments by Id of the tutorial in database, using GET method.

See Figure 2 for more detail.

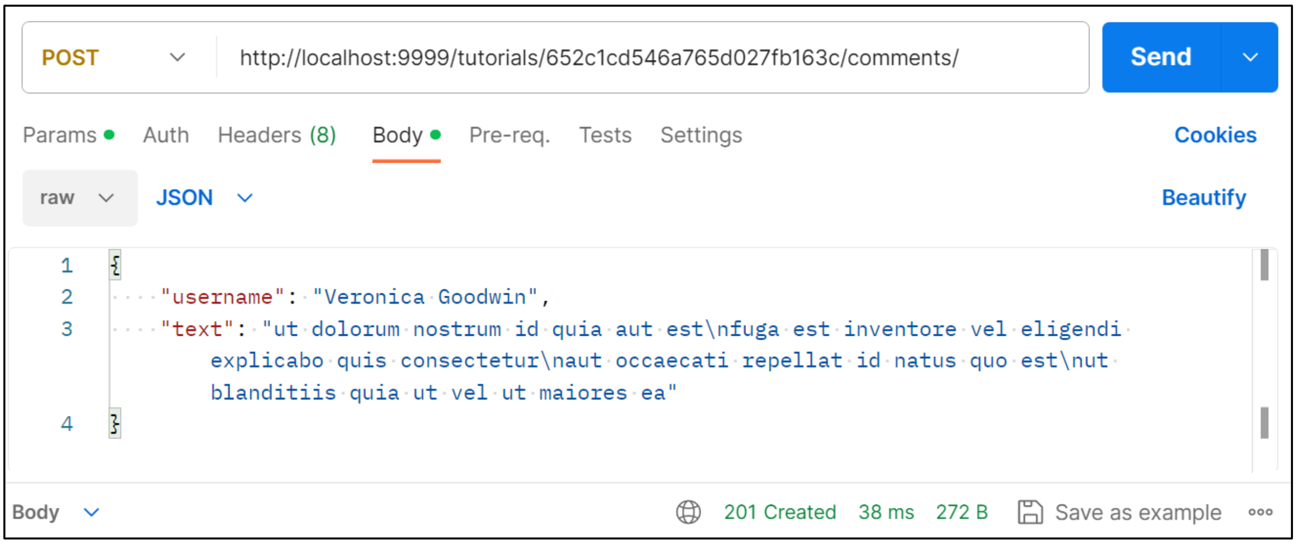


```
[
    {
        "_id": "652cc6cd83c0aab446fd6a06",
        "username": "Tom Cruise",
        "text": "Hi, everyone!",
        "createAt": "2023-10-16T05:14:53.257Z"
    },
    {
        "_id": "652cc74483c0aab446fd6a09",
        "username": "Scarlett Johansson",
        "text": "Perfect self-study topic. Thank you!",
        "createAt": "2023-10-16T05:16:52.418Z"
    },
    {
        "_id": "652d0ef49ed9c1135a182070",
        "username": "Lionel Messi",
        "text": "Great topic. Thank.",
        "createAt": "2023-10-16T10:22:44.381Z"
    },
    {
        "_id": "652d55f19ed9c1135a182073",
        "username": "Cristian Ronaldo",
        "text": "Thanks a lot. This is a beneficial content",
        "createAt": "2023-10-16T15:25:37.866Z"
    }
]
```

Figure 2 - The result of API at ***/tutorial/652c1cd546a765d027fb163c/comments*** *using GET method*

1.3. (2 points)

The API at url: http://localhost:9999/tutorials/:id/comments (**:id** - is an ObjectId of the Tutorial), using POST method to create a new comment. At the same time, update the **_id** of the comment to the Tutorial entity. See Figure 3 for more detail.



```
http://localhost:9999/tutorials/652c1cd546a765d027fb163c/comments/
```

```
{
    "username": "Veronica Goodwin",
    "text": "ut dolorum nostrum id quia aut est\nfuga est inventore vel eligendi explicabo quis consectetur\naut occaecati repellat id natus quo est\nut blanditiis quia ut vel ut maiores ea"
}
```

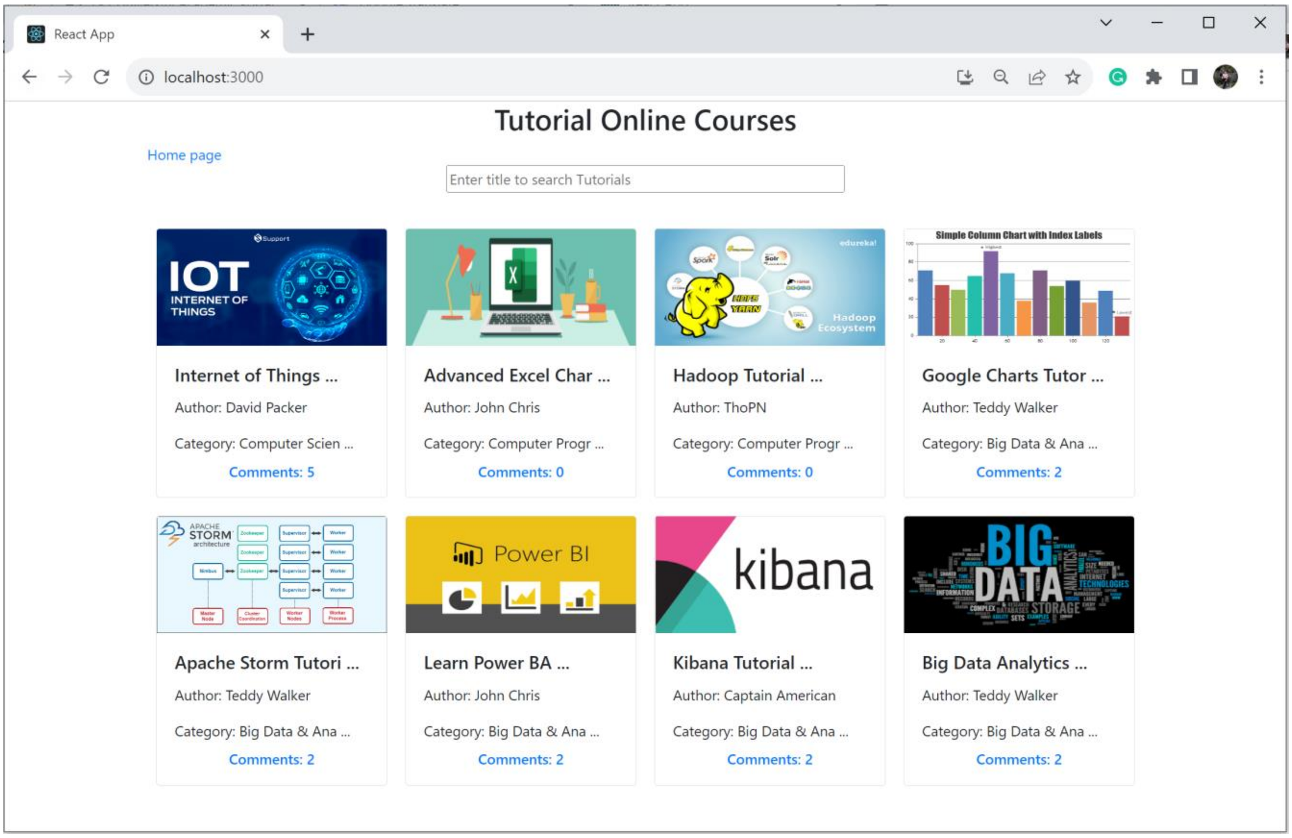
```
201 Created   38 ms   272 B
```

Figure 3 - Using POST method to create a new Comment at url:
http://localhost:9999/tutorials/652c1cd546a765d027fb163c/comments/

Question 2. (5 points)

In this question, you are asked to write a React web application (in **front-end** given folder), that using APIs of Question 1 to manage **Tutorials** and **Comments**.

Note: using the root path **http://localhost:3000** to call APIs.



Tutorial Online Courses

Home page

Enter title to search Tutorials

Internet of Things ...
Author: David Packer
Category: Computer Scien ...
Comments: 5

Advanced Excel Char ...
Author: John Chris
Category: Computer Progr ...
Comments: 0

Hadoop Tutorial ...
Author: ThoPN
Category: Computer Progr ...
Comments: 0

Google Charts Tutor ...
Author: Teddy Walker
Category: Big Data & Ana ...
Comments: 2

Apache Storm Tutori ...
Author: Teddy Walker
Category: Big Data & Ana ...
Comments: 2

Learn Power BA ...
Author: John Chris
Category: Big Data & Ana ...
Comments: 2

Kibana Tutorial ...
Author: Captain American
Category: Big Data & Ana ...
Comments: 2

Big Data Analytics ...
Author: Teddy Walker
Category: Big Data & Ana ...
Comments: 2

Figure 4 – The page when first loaded

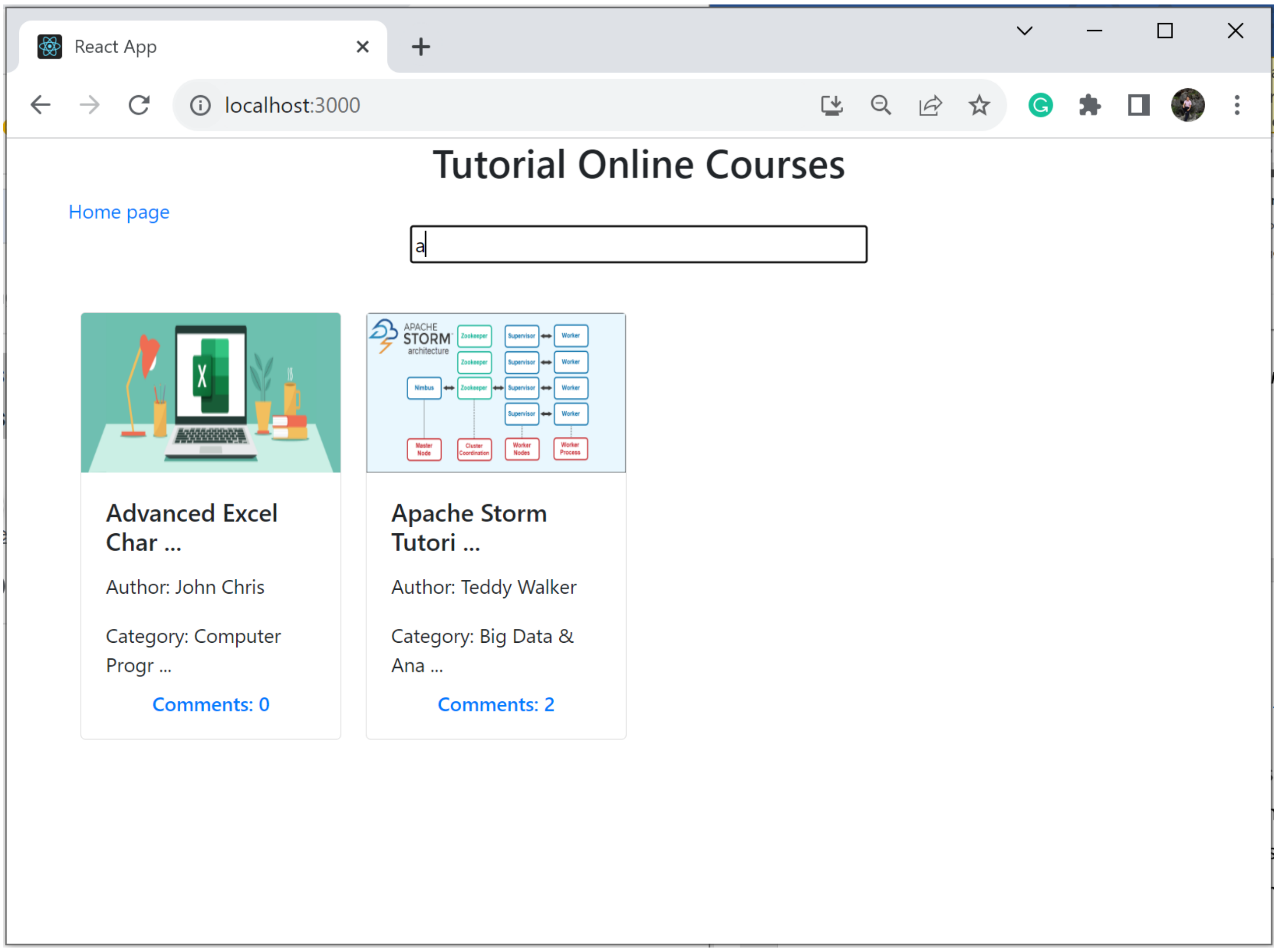
1.1. (2.5 points)

When user open http://localhost:3000/ , load all Tutorials from database like Figure 4.

- Reduce **0.5 points** if the design does not have the correct layout, as shown in Figure 4
- Reduce **0.5 points** if the number of comments for each Tutorial is not counted

1.2. (1 points)

When users enter search keywords on the search form. The list of Tutorials will be filtered by Title starting with the keyword to search for. See Figure 5:



Tutorial Online Courses

Home page

a

Advanced Excel Char ...

Author: John Chris

Category: Computer Progr ...

Comments: 0

Apache Storm Tutori ...

Author: Teddy Walker

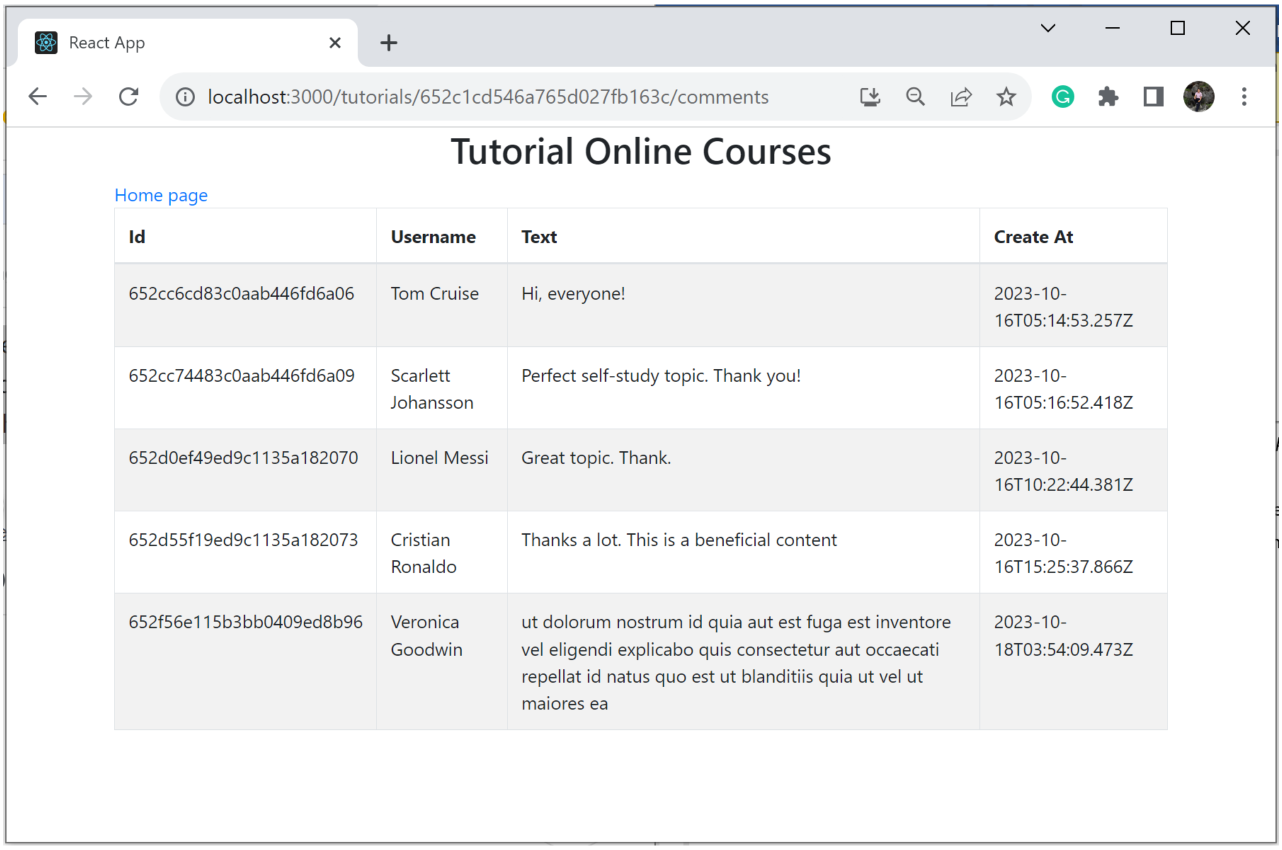
Category: Big Data & Ana ...

Comments: 2

Figure 5 - The search result with search keyword: “a”

1.3. (1.5 points)

When the user clicks on the "comments" link, navigate to the URL: http://localhost:3000/posts/:id/comments . At the same time, display the list of corresponding Tutorial Comments, as shown in Figure 6



Tutorial Online Courses

Home page

Id	Username	Text	Create At
652cc6cd83c0aab446fd6a06	Tom Cruise	Hi, everyone!	2023-10-16T05:14:53.257Z
652cc74483c0aab446fd6a09	Scarlett Johansson	Perfect self-study topic. Thank you!	2023-10-16T05:16:52.418Z
652d0ef49ed9c1135a182070	Lionel Messi	Great topic. Thank.	2023-10-16T10:22:44.381Z
652d55f19ed9c1135a182073	Cristian Ronaldo	Thanks a lot. This is a beneficial content	2023-10-16T15:25:37.866Z
652f56e115b3bb0409ed8b96	Veronica Goodwin	ut dolorum nostrum id quia aut est fuga est inventore vel eligendi explicabo quis consectetur aut occaecati repellat id natus quo est ut blanditiis quia ut vel ut maiores ea	2023-10-18T03:54:09.473Z

Figure 6 - List of comments by tutorialId at url: *http://localhost:3000/tutorials/652c1cd546a765d027fb163c/comments*