

Full Stack Developer Interview Test

Server: .Net
Client: React, CSS3

Server Side

1. Create a new .NET Web API Project (.NET Core, .NET 5, or .NET 6)
You may use Visual Studio, or terminal (for example, the following command:)

```
dotnet new webapi -o MyWebApp --no-https
```

2. Add new **DataController.cs**
3. Add a new action with the following URL: **/data/page**
This action will be used for pagination of JSON data. It will get 2 parameters from the client side:
 - **from**
 - **itemsPerPage**Then it should make HTTP GET request to the <https://jsonplaceholder.typicode.com/photos> endpoint and receive JSON data.

This JSON contains a list of 5000 items with the following structure:

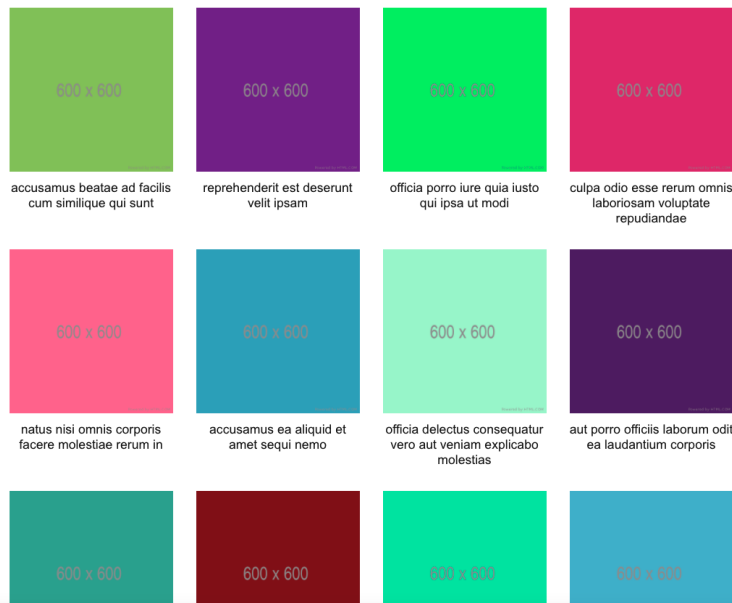
```
[ {  
    "albumId": 1,  
    "id": 1,  
    "title": "accusamus beatae ad facilis cum similique qui  
sunt",  
    "url": "https://via.placeholder.com/600/92c952",  
    "thumbnailUrl": "https://via.placeholder.com/150/92c952"  
},  
... ]
```

Return the JSON that contains only one page of data:

For example, if **itemsPerPage** = 10, and **from** = 0, then return first ten elements.
If **itemsPerPage** = 20, and **from** = 100, return 20 items starting from the element with index 100.

Client Side

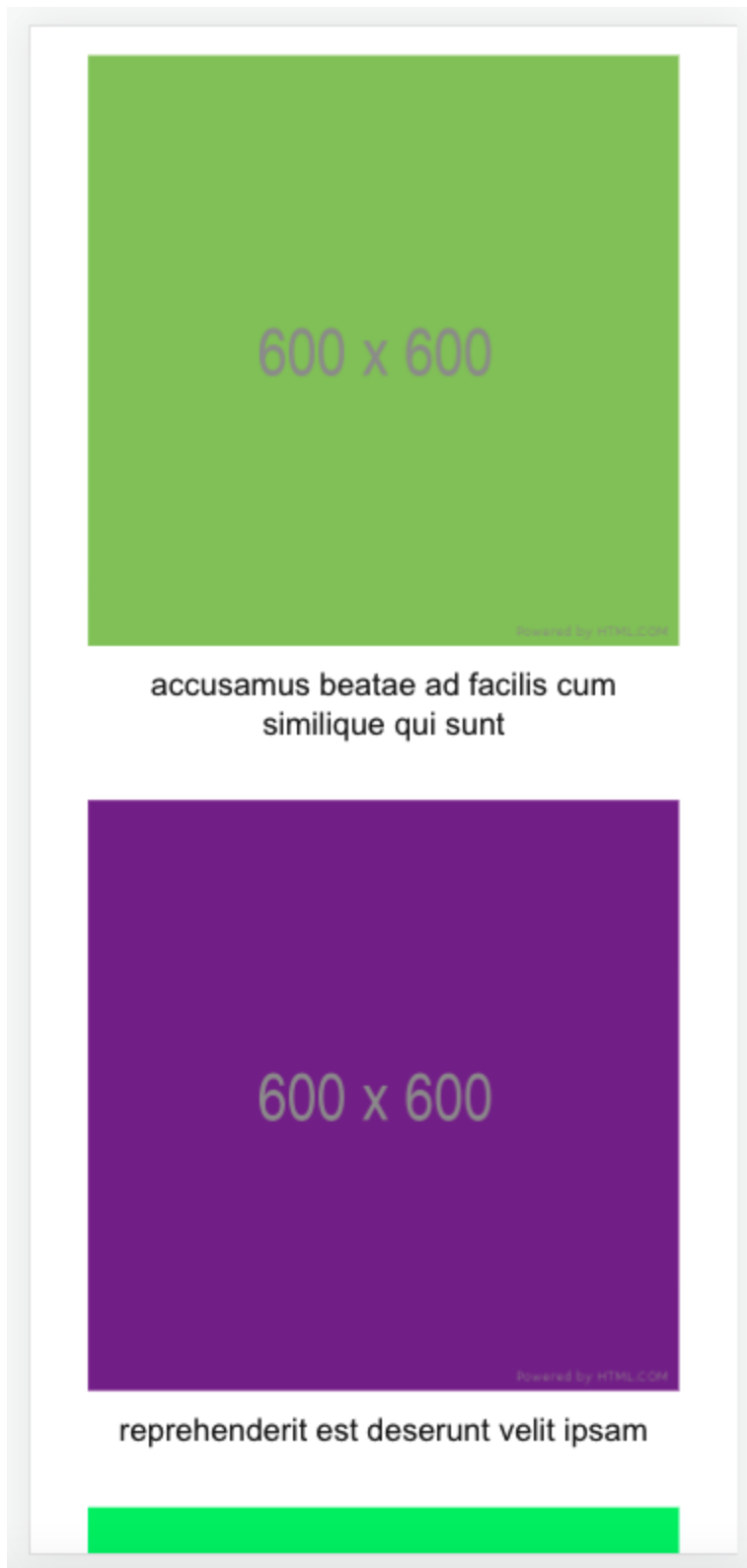
1. Create a new React application using "Create React App" - <https://create-react-app.dev/>
2. Make AJAX request to the **/data/page** endpoint you created in the "server" section, providing **from** and **itemsPerPage** parameters.
3. Render all received items on the page. Each item should contain image and 'title' from the received JSON.
4. Use modern CSS3 to design the page (flexbox, grid, etc.)
 - a. **Desktop:** Create a grid of 4 items per row.



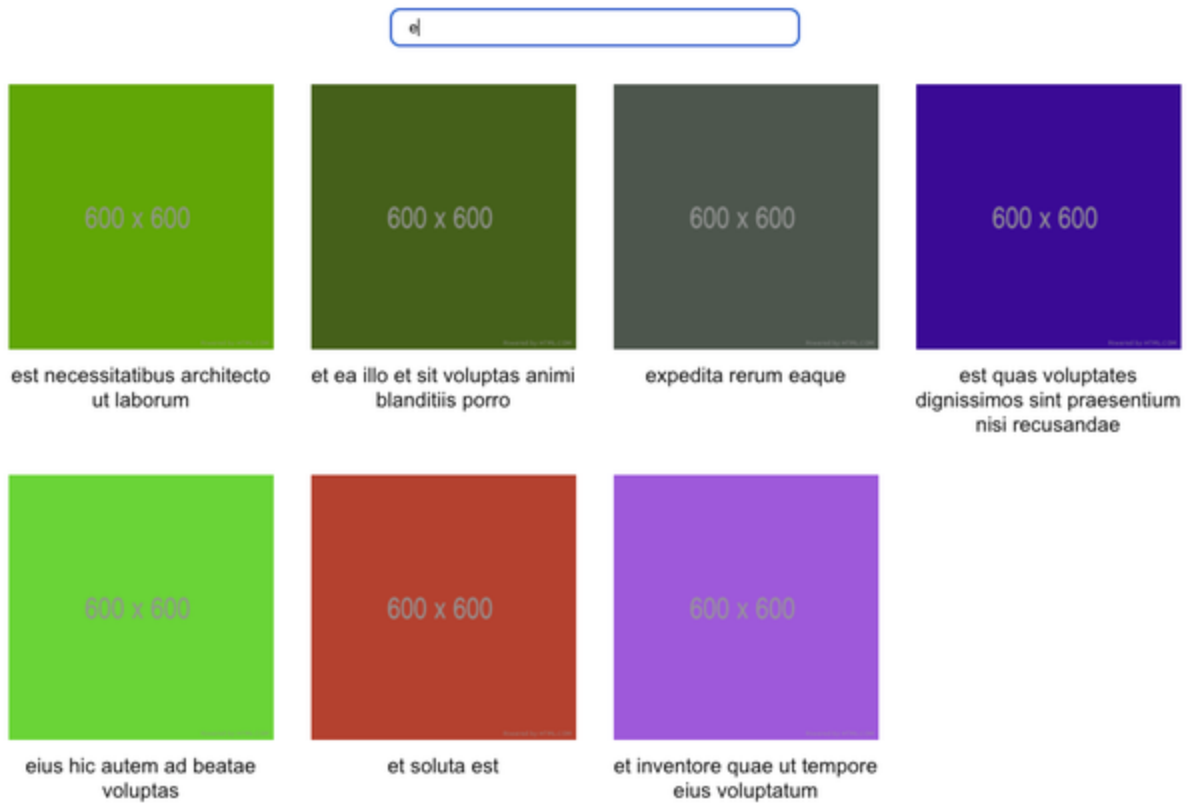
b. **Tablet:** Create a grid of 2 items per row.



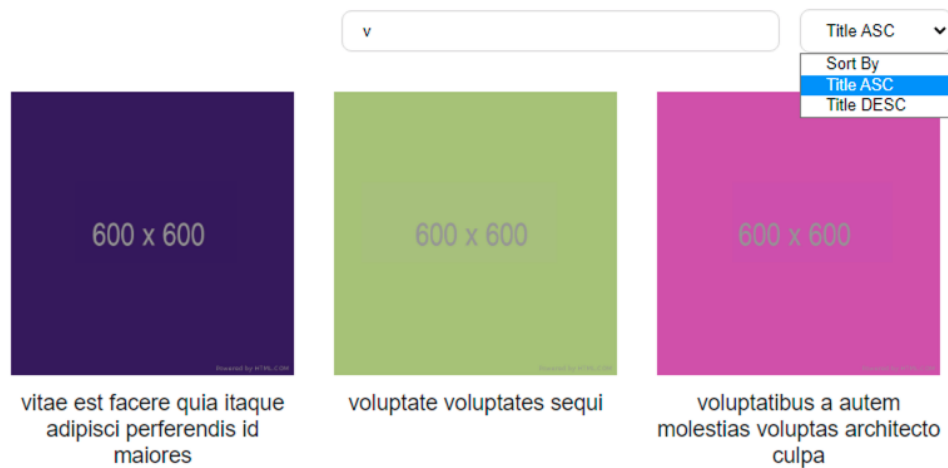
c. **Mobile:** Create a grid of 1 item per row.



5. **Add infinite scrolling.** Once user reaches the bottom of the page, make additional AJAX request to the **/data/page** entry point, and load the data of next page.
6. Add **textbox filter** above the grid. Grid items should be filtered by title on input event. For example, the user enters text 'abc'. Only items with title that starts from 'abc' text should be shown in the grid.



7. Add **sort by title** dropdown. The dropdown should have the following options:
- Sort By - the initial sort
 - Title ASC - sort by title in the ascending order
 - Title DESC - sort by title in the descending order



8. Make sure that filter and sort play nice together.
9. Make sure the the top panel is responsive. When page is scrolled, the panel should always be on top of the page and visible (sticky).

Search

Sort By



600 x 600

Powered by WOTW.COM

accusamus beatae ad facilis cum
similique qui sunt

600 x 600