Task

Build a web application that

- shows a list of promotional campaigns for products
- allows adding new campaigns

We understand that all of us have a life and probably a full-time job as well. That's why we do not expect investing more than 4-5 hours on the solution. We are more interested in your approach and way of thinking, how you structure the application in the given time constrains, what technological choices and compromises you make, etc. The solution needs to be pragmatic that can be evolved into a proper PR. A solution that has all features but is a big untestable mess, is strongly discouraged.

Depending upon the applied role, you may choose the programming language, tech stacks, etc to showcase some of the skills that are more apt for your role. Optionally you may also choose to skip some features/sections that are not relevant for your role.

The `Bonus Points` in each of the sections are just for reference. They are not mandatory at all. Focus on the task at hand and what you would like us to know about your skills. Be creative.

Front-end

- For display, choose any layout that you prefer; table, cards etc.
- Filter by active/inactive campaigns.
- Styling can be minimal, but the markup should be semantic.
- You may use a dummy endpoint with static JSON responses.
- Using reactjs is preferred but not mandatory.
- Using a state management library is not mandatory. However you may choose one if you want; redux, mobx or whatever else you prefer.

Bonus points

- Pagination setup
- Routing setup
- Some basic validation setup
- Base unit tests setup

API

- Create DB schema that roughly defines the structure, the choice of DB is up to you.
- Create RESTful and/or GraphQL endpoints for `campaign` and `product` entities.
- Use standard methods and formats for the RESTful resources.
- While both `campaign` and `product` entities are `read/write`, implementing `write` for `product` entity is optional.

Bonus Points

- Pagination support through query params

- Some basic validation and data sanitization
- Migration/seeding setup
- Deployment container setup
- Base unit tests setup

Full-stack

- Pick one feature each from the Front-end and the API section that completes an end-to-end feature.
- Splitting Front-end and API in their own separate installations, isn't mandatory.

Bonus Points

- Sever Side Rendering (SSR) setup
- Basic caching setup

Cues

Below are some data and visual pointers for you to get started. It's not meant to be an exhaustive setting stone and you may choose to take your creative liberty in line with what you would like us to know about your skills :)

Visual

Campaign List Page

□ Show Active

Name	Product	Start	End
Coca Cola Campaign	Coca Cola	15.08.2018	31.12.2018
Coca Christmas Campaign	Coca Cola	01.12.2018	31.12.2018
Nike New Year Campaign	Nike	01.12.2018	31.12.2018
...	...	...	...

[1] [2] [3] ... [10] Pages

Add New Campaign Page

Label	Type
Campaign Name:	[TEXT]
Product:	[DROPDOWN]
Start:	[DATETIME]
End:	[DATETIME]

[Discard Changes] [Create Campaign]

Data

```
{
    "data": [{
        "id": "71b35f7e-956f-4dbe-bd0c-b627ce0bb6c5", // String | Campaign ID
        "name": "Coca Cola Christmas Campaign", // String | Campaign Name
        "product": {
            "id": "445277f3-9552-4a74-bd35-7a7c4c135a47", // String | Product ID
            "name": "Coca Cola", // String | Product Name
        },
        "start": "1534345609", // String | UnixTimestamp
        "end": "1565881609", // String | UnixTimestamp
        "isActive": true // Boolean
    },
    ...
    ]
}
```

Recommendations

We recommend the following based on the roles. Note, it's just a recommendation after all and you are still free to showcase skills that you want us to know :) Just be wary of the time, we wouldn't want you to invest a lot of your precious time on this.

- Javascript Developer -- Front-end
- PHP Developer -- API
- .NET Developer -- Full-stack