

Hello!

So you want to work with [Putler](#) / [Store Apps](#) / [Icegram](#)?

Of course you do, why else would you be looking at this task sheet!!

We're looking for the right people to join our team, ideally you'll already have some solid relevant experience, but it's not required. We also want to gauge how great your skills are, so here are some quick tasks you'll need to complete and submit with your application.

Complete the technical task below. Email your submission (attach your code as a .zip file) back to us.

Don't put your submission on Github or similar platform. In short, don't make this public. If you do, we have to create another task...

In your email, also **share with us about your experience completing the task.**

Thank you for your interest!

Whether we get to work together or not, we hope these tests are a fun & learn experience for you!

Frontend - ReactJS - Technical Task - Basic App

Scenario: You have to create a React app "Product Analytics". The app will have 3 dashboards or views/screens – Login view, List (or Cards) view and Details view (*Optional*).

Use below

- `dummyJSON` API for request-response and for populating the data.
 - APIs: <https://dummyjson.com>
 - Docs: <https://dummyjson.com/docs/>
- Tailwind CSS
 - <https://tailwindcss.com/>

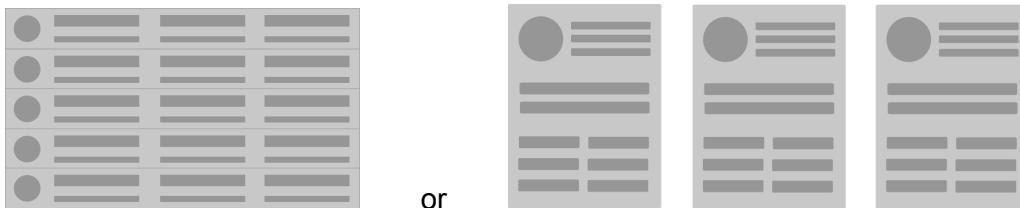
- If you are new to Tailwind CSS or if this seems complex to use, you can choose to skip it and use your own styling.

1. Login view



- This is the landing screen and will ask the user to enter their login credentials – `username` and `password`
- API Documentation for `auth`: <https://dummyjson.com/docs/auth>
- Valid credentials should take you to the list/cards view..
 - Valid credentials will return a `token` in the API response, which should be used as a mode of authentication for all the other API calls hereafter.
- Invalid credentials should show a basic error message.

2. List (or Cards) view



- Show the products in a list (*tabular format*) or cards view.
 - Do not show all the details right away.
 - You can choose from the list of fields from the API response.
- API Documentation for Products: <https://dummyjson.com/docs/products>
- Try to use images, wherever applicable.
- Clicking on any product from the list should open up a detailed view for the product.

3. Details view



- Get details about the selected product.
 - This can be shown in a sidebar, modal window or a `details` dashboard/view.
- Since the list / cards view above won't have all the information/details/fields related to the product, include everything (*from the API response*) in the details view with some additional information.
- In addition to the product detail, show a random todo and a random quote as well.
 - Get details/info for a particular product.
 - <https://dummyjson.com/docs/products>
 - Get todo
 - <https://dummyjson.com/docs/todos>
 - Get quotes
 - <https://dummyjson.com/docs/quotes>
- Try to use images, wherever applicable.

Please note:

- Read the documentation carefully so that you choose the right APIs.
 - APIs: <https://dummyjson.com>
 - Docs: <https://dummyjson.com/docs/>
- Make sure to include `/auth/` for all API calls since we have to use the `token` with every call. E.g. below.
 - Incorrect call: <https://dummyjson.com/quotes>
 - Correct call: <https://dummyjson.com/auth/quotes>
 - *Though the API calls works without the `auth` as well, we would want to use the token retrieved after successful login.*
- You can choose to design the app as per your choice and preferences.

- We encourage you to pay attention to the UI/UX and to follow the basic design principles.
- Code structuring, building components, code portability, reusability and scalability also plays an important role in App development. And you may want to consider them while building the app.
- Try to cover the basic test cases.
- Optional: If you are enjoying it, you can also incorporate additional APIs from User, Products, Carts, Categories, Quotes, To-dos, Comments, Posts, Image, Search.
 - But please make sure to include the default workflow (1 – 2 – 3) and to include the API calls as described in it.
- Bonus points – You can try incorporating any/all from below:
 - Details view as suggested above.
 - Use [Webpack](#) to configure your project/app.
 - Provision for Search
 - Documentation for search: <https://dummyjson.com/docs/products#search>