**Requirements Document**

General Info about the theme (will be provided):

<https://material-ui.com/store/previews/minimal-dashboard/>

<https://docs-minimals.vercel.app/introduction>

Info About the React Library we use:

<https://reactflow.dev/>

<https://reactflow.dev/docs/>

<https://reactflow.dev/examples/>

<https://github.com/wbkd/react-flow>

**Pages:**

We will keep the following pages:

1. First user starts at Home Page from theme (<https://minimals.cc/> ) 🡪 no customization needed now
2. Second we keep Pricing Page (<https://minimals.cc/pricing> ) --> no customization needed now
3. Once User clicks on any of the Pricing Pages he lands on

Payment Page (<https://minimals.cc/payment> ) 🡪 customization needed

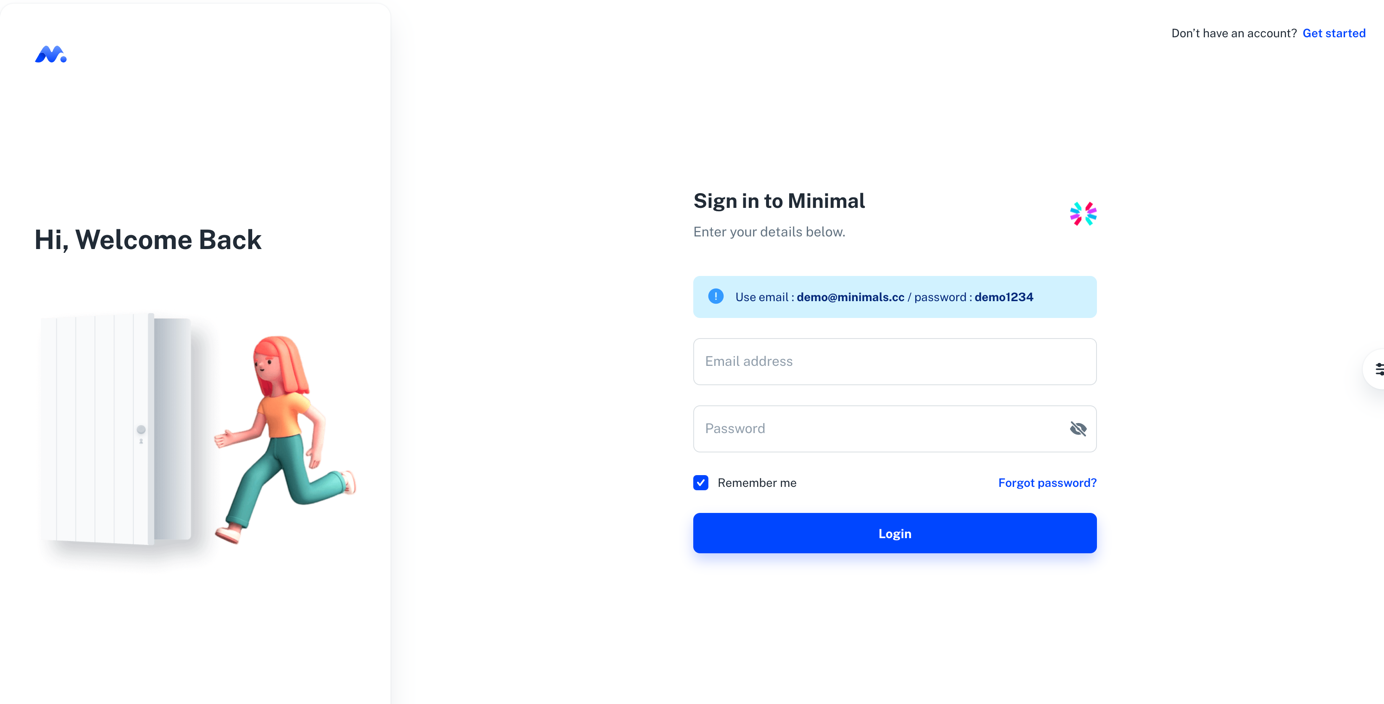


If he clicks on Create my account he will be guided to <https://minimals.cc/auth/register-unprotected> . Once he clicks register he will be sent back to the Payment page but this time a button appears like that:



If he clicks on already have an account he will be forwarded to this the login page of the theme: <https://minimals.cc/auth/login-unprotected> . Remove the use email demo login data from login page. Include AWS Cognito for User Management. This is compatible with the theme and does not need programming.

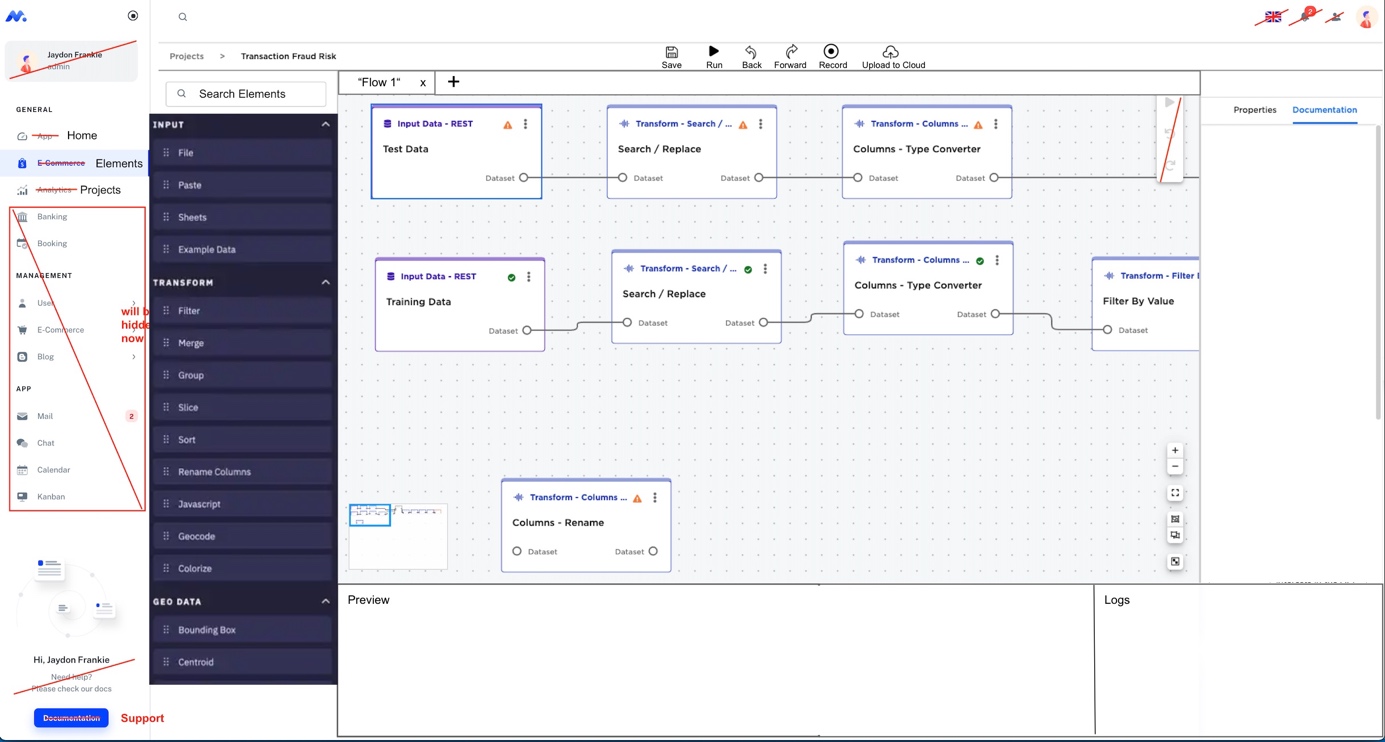
1. The most important page is the Dashboard page where the user gets into once logged in. This page will be heavily customized. Details are below.
2. The Home page inside the Dashboard is the main dashboard site with the workflow area and the side bars. The Home page outside is referred to this page <https://minimals.cc/> and is the landing page.
3. Keep Login Page like in template



1. After Login we see the Dashboard area
2. In Dashboard area should be customized to this Image using Reactflow open source Library ( <https://reactflow.dev/>) . The Workflow window from ReactFlow should be embedded into the provided the Material UI template. For inspiration also look at the following link:

<https://webkid.io/blog/datablocks-node-based-editor-data-processing-visualization/>

Below is the UI we want to achieve and get.



When you click on Elements in Menu Bar, the elements side bar will open up (blue field). Make this blue field also the same black white color as the other ones.

There is the left Menu Bar after there comes the Element Bar when you click on Elements. The Home Menu item if you click it it will show the dashboard as shown in the screenshot.

The grey area is the Workflow area. It contains the connected Blocks. The functionality of Zoom + - or the small overview window is already implemented in the React Flow Library and needs to be added. A lot of functionalities can be taken out of the React Flow Library.

Include a Path above the Workflow window.

Include the Save, Run Icons and Text and so on above the Workflow window.

Select the right icons for the Menu items Home, Elements and Projects

For the Workflow Elements put the 6 dots to the right side instead of the left side of the box.

Leave out the headlines for now (Input, Transform ..).

Make a total of 14 Element Boxes (Drag & Drop Elements in the Blue Box) inside the Elements bar. Name them Element 1, Element 2, Element 3, Element 4 and so on.)

On top of the workflow area there are tabs. Flow 1 is shown. If you click on + a new tab will be added to the window and a clean Workflow Area will be opened up.

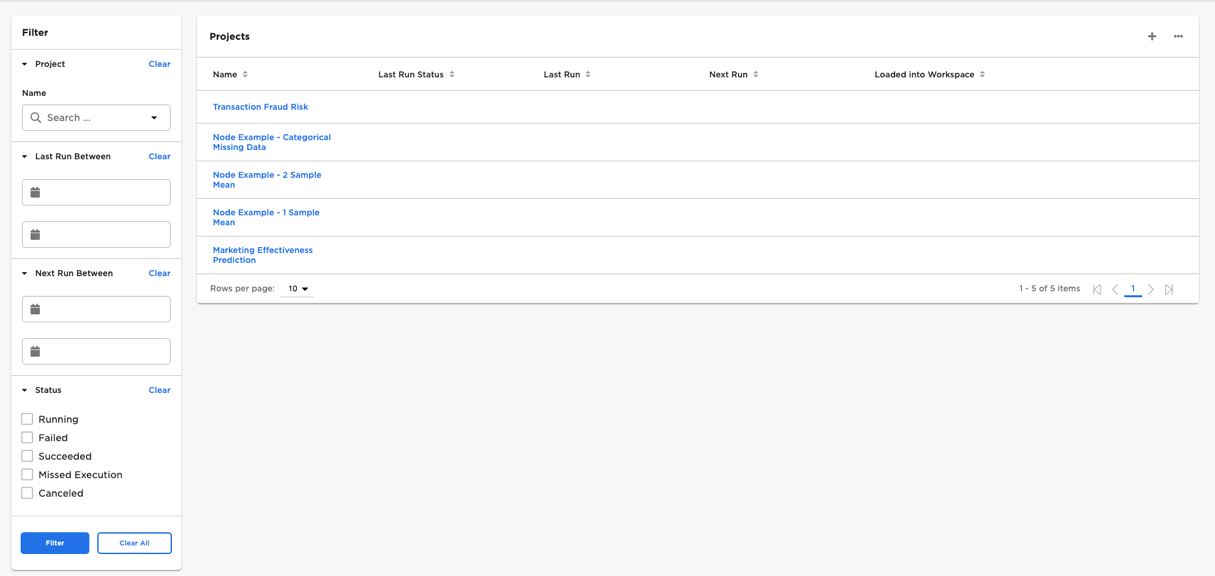
Hide anything with a crossed line.

Replace the text in the Menu bar with the one shown in Red, but use a black font color for the new Text.

Do not delete Pages or code, just comment out or Hide them in case we are going to use it.

1. Project Menu Page

The Ui should be implemented like that:

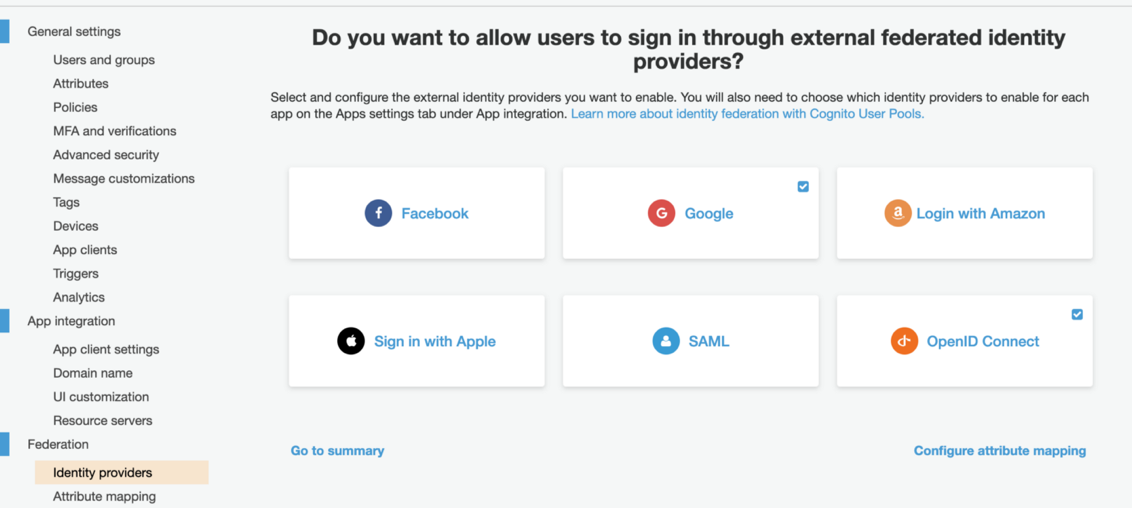


1. Other Pages

Just hide them instead we need to use them.

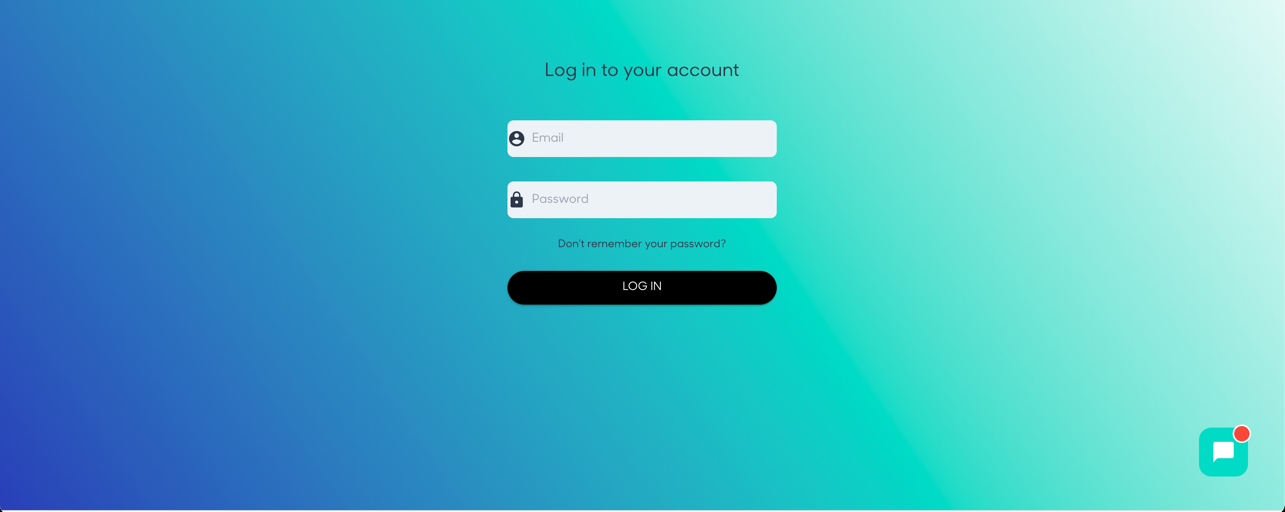
1. Connect MangoDB Atlas. We need to be able to save the Workflows and open them if needed. Each Workflow will contain Javascript Code later. It has to be stored as JSON File in MangoDB. Please build the Database and Structure. We also want to store the order of Blocks. Each Block will contain a specific JavaScript Code later. The projects and the flows need to be stored in the database.
2. Connect AWS Amplify with Github and all components so we can see the App hosted on AWS. Also connect Cognito and configure it for Login with the template Material UI I provided you.

Let users also sign in with other services like Google, Apple or Facebook. (see image)  
(Cognito Identity Provider Configuration)



1. Enable Dark/Light Mode (already integrated into Theme)
2. Check responsiveness (already implemented into theme)
3. Add Stripe Payment only leave out the API Keys. Use dummy data from stripe. There is a payment page where you will integrate it ( <https://minimals.cc/payment> ).
4. Please document the code so anyone can understand what is happening.
5. Design guidelines. All the icons and design should be aligned with Material UI guidelines.
6. Colors are Black, White (use the Light Mode as default), #00dac5, # 3700b3

An Example for the color usage:



1. For Fonts you will receive the fonts which you can use.
2. In end you will have to show me where to include the JavaScript code for the Flow Blocks.
3. There will be 1 or 2 revisions once reviewed on a day by day basis. This will not include heavy coding or new work. It includes small adjustments and changes which has to be made on the fly and are necessary. It is our goal to finish this project fast and accurate.