React Firebase Login Flow

Weight: 20%

Due Date: Friday April 8th 2022

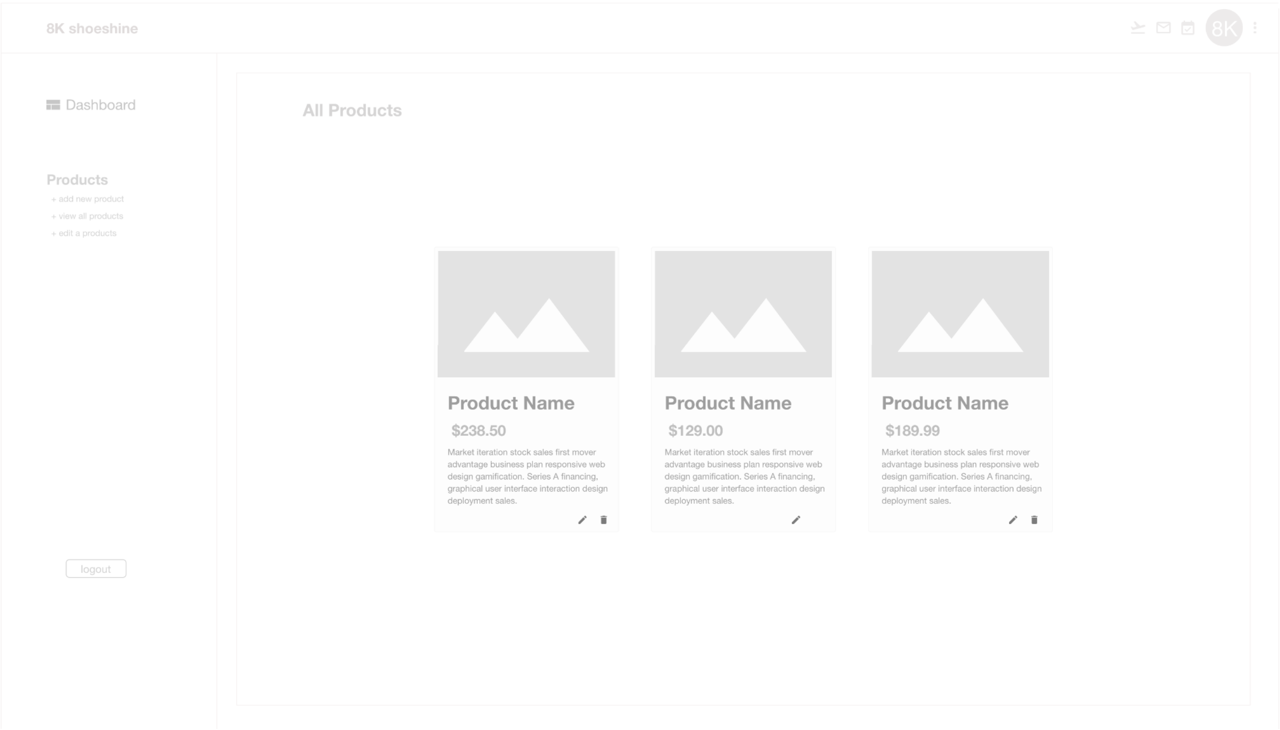
Submit Before: 11:59pm

Introduction

For the last couple of weeks you have been working on your dashboard application creating the product editor component. In this assessment you are going to complete the product write functionality and implement the read functionality for the view all products pane.

From the previous assessment your login page will only allow access to validated users. Later you will add rules to the Firebase Real Time Database to restrict read and write access to the product data

Make sure to add the have created a user with the email address of ***jim@home.com*** and a password of ***123456.*** Check that you have enabled sign in with email and password in the Firebase dashboard Authentication panel.



Product Name

$238.50

Market iteration stock sales first mover advantage business plan responsive web design gamification. Series A financing, graphical user interface interaction design deployment sales.

Product Name

$129.00

Market iteration stock sales first mover advantage business plan responsive web design gamification. Series A financing, graphical user interface interaction design deployment sales.

Product Name

$189.99

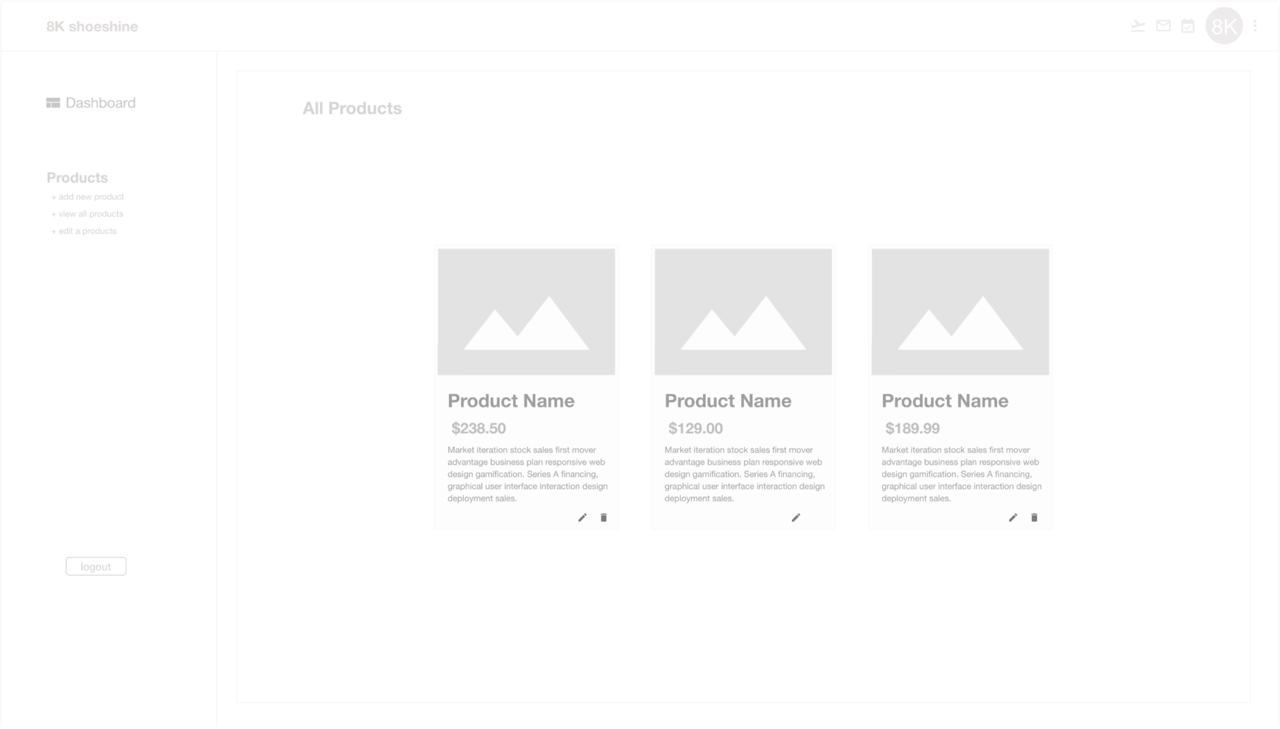
Market iteration stock sales first mover advantage business plan responsive web design gamification. Series A financing, graphical user interface interaction design deployment sales.

For this assessment you will be developing the default dashboard panel, which will display all products to the user. Your product card will display the product image, the product price, the product description along with an edit and delete button.

Build Requirements

The View All Panel

The view all panel should be the default nested route when the user is directed to the dashboard from the login screen. The panel should show a minimum of 3 products. You will be using these three products later when you build the storefront in Nextjs. To read the products from the Firebase RealTime Database you will use the get method provided to you in the firebase/database package.



Product Name
$238.50
Market iteration stock sales first mover advantage business plan responsive web design gamification. Series A financing, graphical user interface interaction design deployment sales.

Product Name
$129.00
Market iteration stock sales first mover advantage business plan responsive web design gamification. Series A financing, graphical user interface interaction design deployment sales.

Product Name
$189.99
Market iteration stock sales first mover advantage business plan responsive web design gamification. Series A financing, graphical user interface interaction design deployment sales.

useGetAllProducts.js

Inside your hook’s folder create a file called useGetAllProducts.js. Inside this file create the logic for reading product data from the RealTime Database. Remember that fetching data is asynchronous so you will want to use the promise api or async await. Your hook should pass back the data to the view all products panel. So you need some way to check when this data becomes available to the component and when the component has data you then must return the ui.

Display All Products

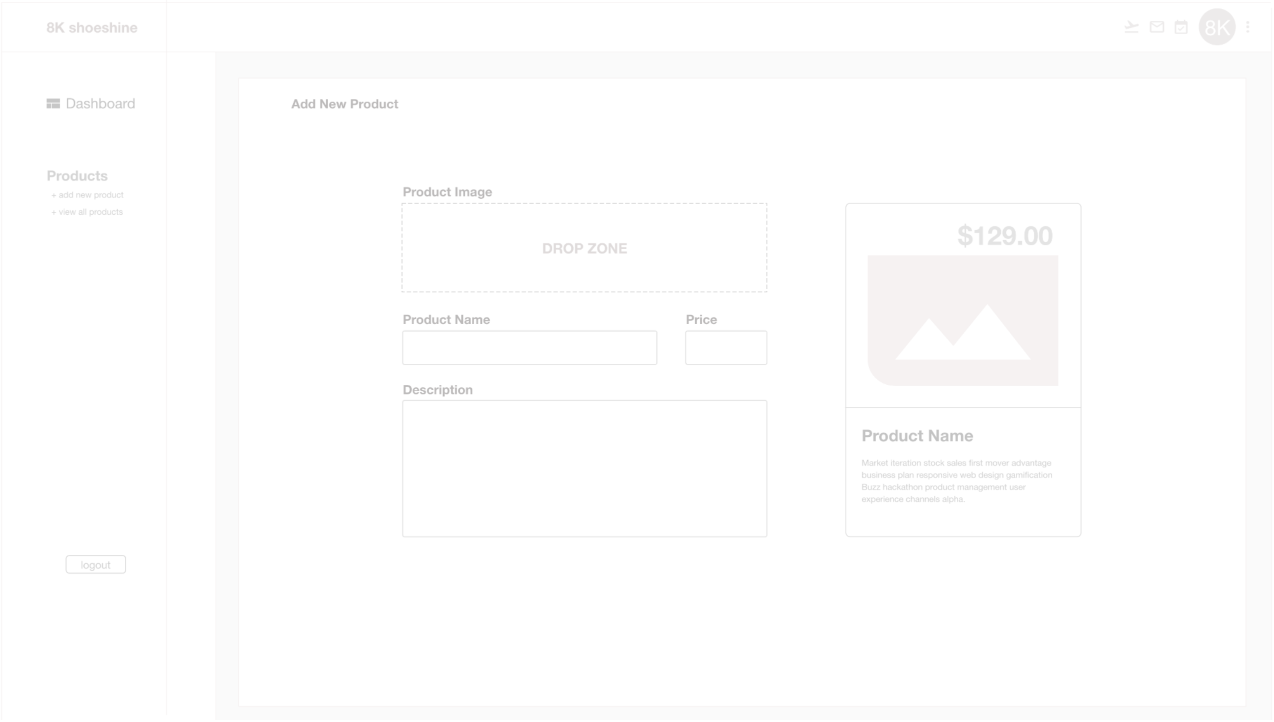
When the data is returned to the view all products component you can then start to the next step.
You have to display a minimum 3 unique products and to do this you will have to do a conditional render. If there is data then create the UI if not then return null in your component logic.
Once the data is available you will have to loop through the data and pass each data object to your ProductCard component as props. You must show the product image, product price, product name, and the product description along with an edit and delete button.

Build Requirements

The Product Editor Panel

A user of the dashboard should be able to add a new product. To access the product editor provide a link in the sidebar to the Add New Product Panel. This panel should show the loading interface for a new product along with a live preview of the product as it being created.

The editor should contain a control for loading the product image, the product price, the product name and the description of the product. Whenever one of the controls updates the new state should be displayed in the product preview component.



Market iteration stock sales first mover advantage business plan responsive web design gamification Buzz hackathon product management user experience channels alpha.

Provide the user with a submit button that will write the data to the RealTime Database and store the image in Firebase Storage. As soon as the submit button is clicked switch the UI using conditional rendering to show the upload/write status. When the data has been saved, update the UI to confirm to the user that data was successfully written and provide your user the option of returning to the dashboard route (which will show all products) or allowing the user to add another product.

Github And Deployment

You are required to have the source code for your application accessible from GitHub. Please take the time to provide a well organized and professional looking readme document. Make sure that your Firebase API keys are not accessible via Github. To do this add the .env and or the .env.local to the .gitignore file.

Provide the URL to Netlify where your project is hosted. Your project must contain a custom 404 page and not display the Netlify 404 page.

Moodle Submission Requirement

Link to your GitHub repo for your project
Netlify site deployment URL.
Late Assignments will not be graded.
Any site that displays a 404 error will receive a grade of zero

NOTE

Custom 404 Page on Netlify needs the _redirect file in the public folder. Without it you will get the Netlify page not found displayed.

NOTE

Your assessment will be checked using the default user.
user email: **jim@home.com**
password: **123456**

Marking Key

Task		Task Value
View All Products Component **Requirements** Custom Product Card Component Conditional rendering of the product card UI. Component card display logic. Completion of the firebase read functionality		**8**
Product EditorComponent **Requirements** Editor component custom loading interface and product preview. Preview component update functionality. Preview component write functionality. Feedback component functionality. Feedback option functionality.		**10**
Deployment **Requirements** GitHub repo is well organized and contains a professional looking Readme.md file. (No assignment 3 naming) Naming should be branded to the company/organization that is selling the product. Netlify deployment working and displaying custom 404 pages.		**2**
		20

Marking Rubric

Marks	5 Marks Criteria
5 [0]	Task was completed with the highest proficiency, adhering to best practices. The tasks demonstrate a high level of proficiency implementing the subject matter and the presentation of the content is to a professional standard.
4 [-1]	Task was completed well, with some minor mistakes. Well above average work, shows good understanding of the implementation details of subject matter. Tasks show a high degree of competence.
3 [-2]	Task was completed satisfactorily. Some features are missing or incorrectly implemented. Shows a moderate level of understanding in the task with room for improvement.
2 [-3]	Task completion is below average, the task was poorly completed. Shows understanding of the task and the requirements to implement, but implementation was poorly executed.
1 [-4]	Some of the tasks were completed. Shows a lack of understanding in the subject matter and very poor execution.
0 [-5]	Not completed.

Marks	3 Marks Criteria [minus]
3 [0]	Task was completed well, adhering to best practices, and shows a high degree of proficiency with the subject matter guidelines.
2 [-1]	Task was completed satisfactorily. Some features are missing or incorrectly implemented. Shows a moderate level of understanding in the task with room for improvement.
1 [-2]	Some of the tasks were completed. Shows a lack of understanding in the subject matter and very poor execution.
0 [-3]	Shows a little to no degree of competence in completing the task; not completed.

Marks	1 Marks Criteria
1	Task completed satisfactorily
0	Task was not completed