

# React Firebase Login Flow

**Weight: 20**

**Due Date: Friday March 25th 2022**

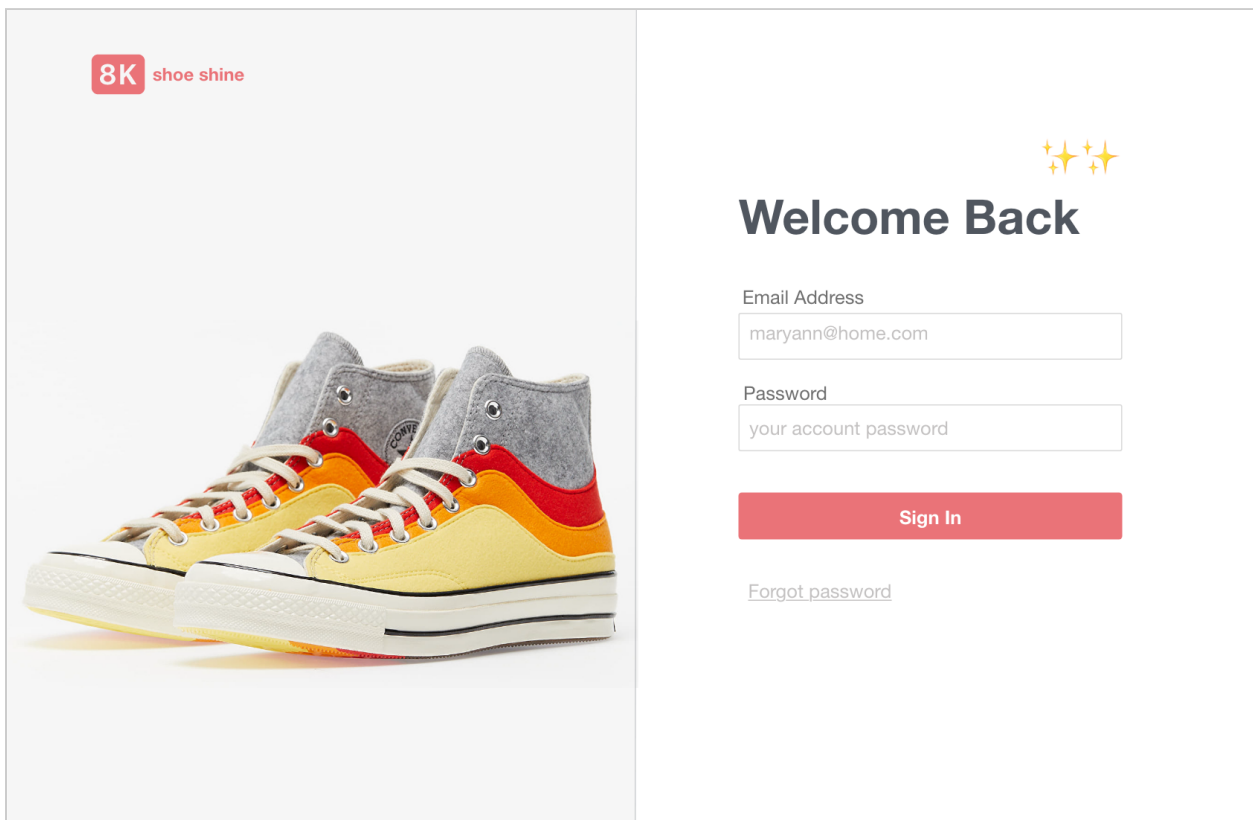
**Submit Before: 11:59pm**

## Introduction

In this assessment you are going to work through the videos provided in class and create the login flow to sign in a user and grant them access to the dashboard.

From the previous assessment your login page will look something like the example below. With the components completed you can now add the javascript code to your React application to complete the dashboard sign in flow using the Firebase authentication service.

Make sure to add the following user [jim@home.com](mailto:jim@home.com) using the password **123456** to your Firebase authentication service. This is the login that your instructor will use to mark this assessment.



Upon a successful sign in attempt the user should be redirected to the dashboard. Your instructor will check that the dashboard is a protected route. Meaning only an authenticated user will have access to the Dashboard.js component (Dashboard page). Due to time constraints form validation will be skipped but you can implement a simplified validation using toast messages shown in class.



# Build Requirements

## Login Component

The login component will use the method demonstrated in class to authenticate a user with Firebase using email and password. Add toast messaging using custom or default error codes from the return value to the firebase authentication method `signInWithEmailAndPassword`. Use the `react-router-dom` libraries, `useNavigate` to redirect a successful login to the `DashBoardPage` component. To do this you must set up the required routes in the `App.js` component.

## Protected Routes

The `DashBoardPage.js` component must only be accessible to an authenticated user. In class you were shown how to accomplish this using `react-router-dom` along with the `authStateChange()` method provided by the Firebase SDK. You are required to make the route `/dashboard` only accessible to a valid user of the app. Any try to access the `/dashboard` route should return the client to the `LoginPage` component.

## 404 Page

The custom 404 page that you provide will need to check for a valid user as well. In this component you will use conditional rendering. If there is a valid user signed in then the link provided will take the user back to the dashboard. For everyone else the 404 page link should return the user to the `LoginPage` component .

## Panels Nested Routes

You were shown how to add nested routes. In your application you are required to add this functionality. In the sidebar you should provide 3 links to each of the required panels. The View All Panel (default panel), Add New Product Panel and the Edit Panel. In the `App.js` you will have to create 3 nested routes. The View All Panel should be the default panel displayed when a user goes to the `/dashboard` route. To facilitate this use the `index` property of `react router dom` when creating nested routes under the `/dashboard` route.

## Product Card

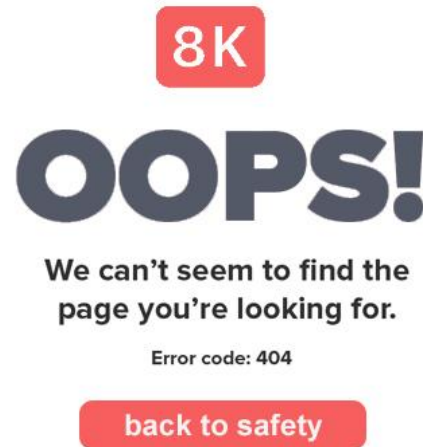
Create a component that will be used to display products in the View All Panel and in the Add and Edit Product panels. The card needs to have an image, product name, product price and a description. See the example images later in this handout for an example.

# Component Examples

## Examples 404 Page

Your 404 component should be styled and branded. You can have fun with this component search on google images for examples of 404 pages. There are some really great examples and you should use the one that feels right for you.

The back to safety button or get me out of here button will function differently depending if you have a logged in user or not. Valid users will return to the dashboard and everyone else will be redirected to the login page.



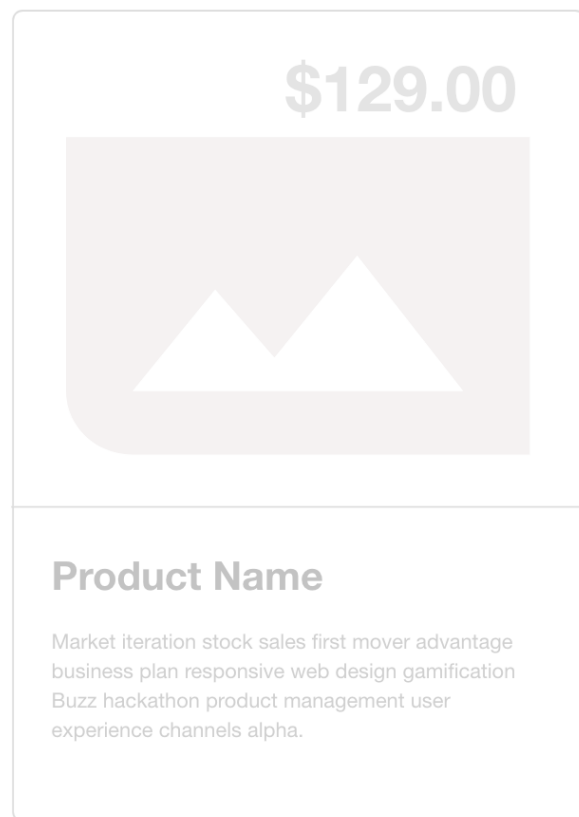
## Examples Product Card

The product card should have a display area for the following data

1. Product Price
2. Product Image
3. Product Name
4. Product Description.

You can add additional information if you wish but the list provided is the minimum requirements for the product card.

Make sure to have these four data elements as they will be required later when we set up the payment gateway.

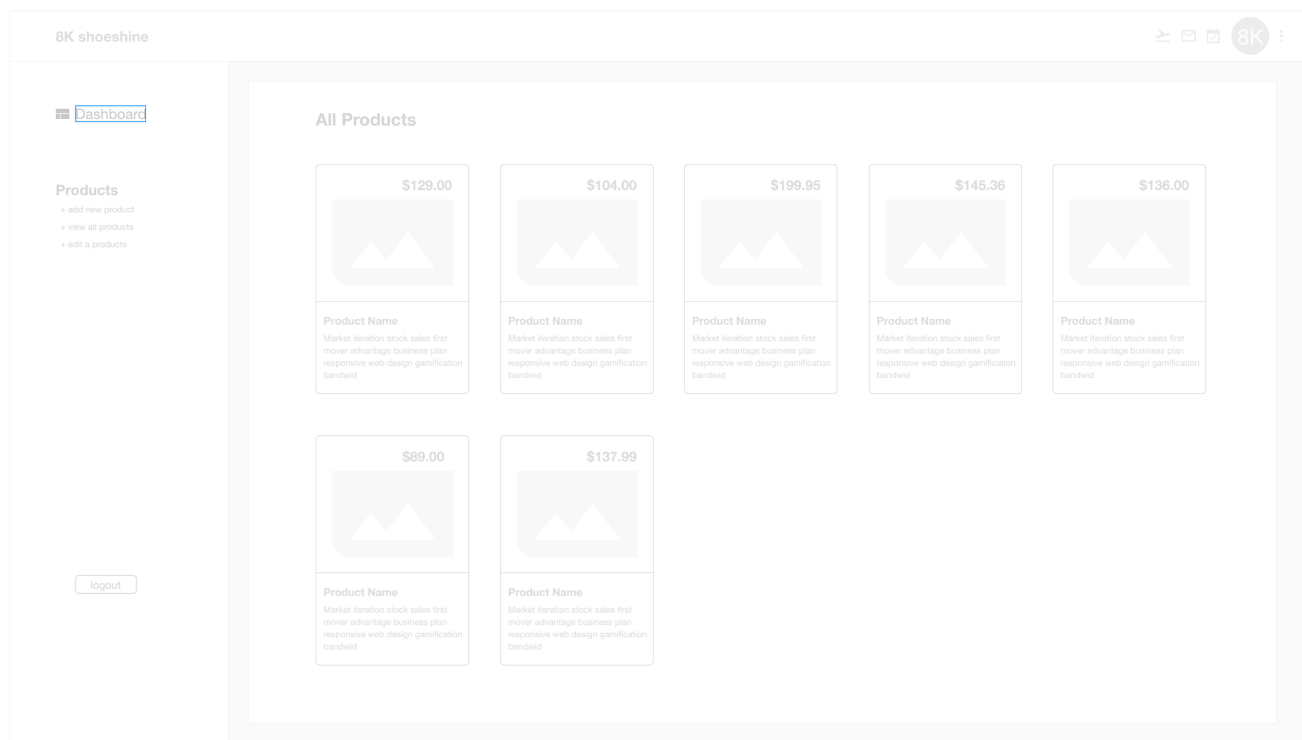


## Example Panel

In the last assessment you created a panel for as part of component development. Use your panel component to create the required 3 panels.

1. View All Products Panel
2. Edit Product Panel
3. Add New Product Panel

Below is an example of the view all panel which is the default panel displayed when a user gains access to the dashboard. The router will use the outlet component to load the view all panel when matching the /dashboard route path. This is accomplished by nesting routes within the dashboard route. There will be three nested routes to the panels listed previously.



For each of the panels you will have to use the Link component provided by react router dom. In order to remain consistent use the following names when you create the sidebar Links. These names should match up with the nested route paths that you create in the App.js component.

1. View All Panel Route Path: all
2. Edit Panel Route Path: edit
3. Add New Panel Route Path: add



## Deployment

As with the last two assessments 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. First looks are important so provide a detailed product description. You can add why you created the project, what you learned by doing this project, what you liked and didn't like. Describe the type of tech that went into creating the project often called the stack. Just try your best to make the Readme.md look as professional as possible.

## 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
<b>LoginPage Component + Protected Routes.</b> <b>Requirements</b> Form validation errors are handled using toast messages. Toast will present firebase default error codes or custom error codes. Firebase default user sign in redirect to dashboard.		5
<b>Nested Route + Panels</b> <b>Requirements</b> Three nested routes. View All Panel Route path all set as default route. Add New Product Panel route path add Edit Product Panel route path edit.		5
<b>404 Page</b> <b>Requirements</b> Implements authStateChange Uses conditional rendering based user authentication. Valid user returned to the dashboard page. Invalid user returned to login page. Custom styled page with branding/logo		5
<b>Product Card + Deployment</b> <b>Requirements</b> View all page provides and example of the Product card Product card contains a placeholder image, product price and description.. 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.		5
		20



# Marking Rubric

Marks	5 Marks Criteria
<b>5 [0]</b>	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.
<b>4 [-1]</b>	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.
<b>3 [-2]</b>	Task was completed satisfactorily. Some features are missing or incorrectly implemented. Shows a moderate level of understanding in the task with room for improvement.
<b>2 [-3]</b>	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.
<b>1 [-4]</b>	Some of the tasks were completed. Shows a lack of understanding in the subject matter and very poor execution.
<b>0 [-5]</b>	Not completed.

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

Marks	1 Marks Criteria
<b>1</b>	Task completed satisfactorily
<b>0</b>	Task was not completed

