| **BLOG-BYTE Progress File-5** |
| --- |

# **Protected Routes for Admin**

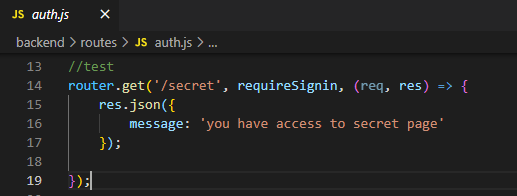
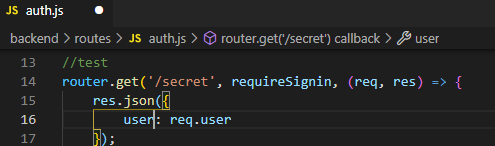
1. **Auth and admin middlewares - Backend**
2. **Admin and user dashboard**
3. **Protecting routes for auth and admin user**
4. **Show dashboard link in nav based on user role**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

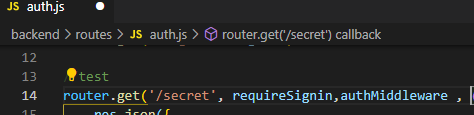
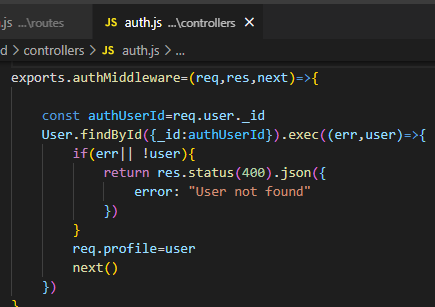
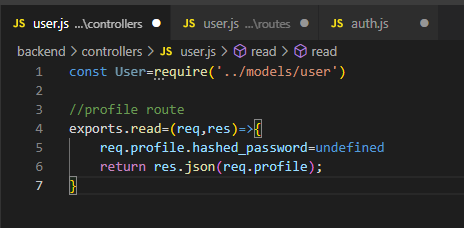
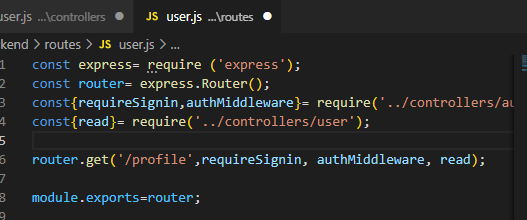
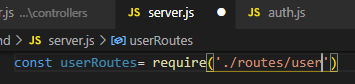
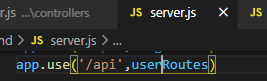
# **TASK-1**

# **Auth and admin middlewares - Backend**

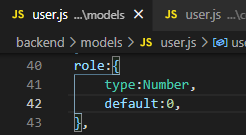
**STEPS :**

* Redirecting users to dashboard on signin.
* Need to create 2 middlewares in backend, one for authenticated users and one for admin users.
* In **backend** folder, got to **routes** and open **auth.js** file.
* 
* Change the message to req.user.
* 
* It will make user available in request object.
* Go to postman, signin , copy the token from response and paste in authorization header.
* Now make a get request to **secret page**. We can see user response .

|  |
| --- |

* Adding auth middleware, to make requested user available.
* 
* In **controllers** folder, go **auth.js** file and create middlewares here.
* Based on id we created database, given the user based on that id, and making that user available in profile.
* 
* Now creating middleware for admin.
* 
* Reading user from profile.
* In **controllers** folder, create a new file **user.js**. Adding user read method.
* 
* In **routes** folder, create a new file **user.js**.
* Creating route for profile.
* 
* In **backend** folder, go to file **server.js**.Add user routes.
* 
* Applying user middlewares.
* 
* Now make a get request to profile using postman, we will get user profile in response.

|  |
| --- |

* In **models** folder, go to file **user.js**. Change role.
* 
* Now apply admin middleware, access denied because user is not admin.

|  |
| --- |

* Updating role manually from Mongo Atlas. **(role: 0 =>for user , 1=>for admin)**

|  |
| --- |

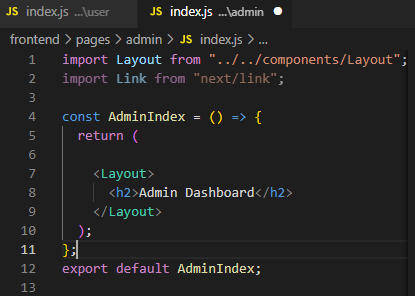
* Now if we will access profile as admin we can get response.

|  |
| --- |

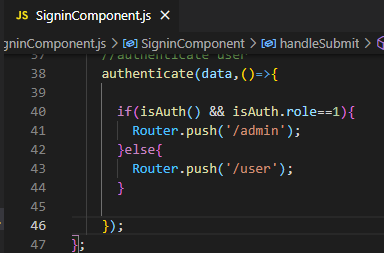
**=>Auth and admin middlewares working, Task Completed… :)**

# **TASK-2 Admin and user dashboard**

**STEPS :**

* In **frontend** folder, go to **pages** create two new folders called **user** and **admin**.
* Inside **user** create a new file index.js.
* Create user dashboard page. And admin as same.
* 
* On sending request to user/admin page, we will get user/admin dashboard.

|  |  |
| --- | --- |

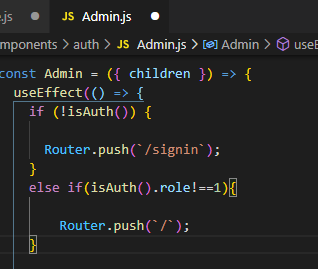
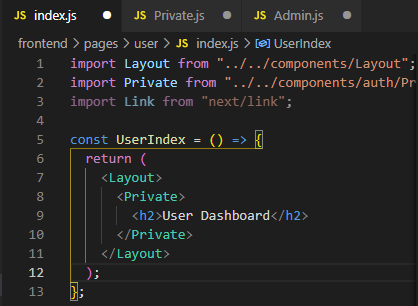
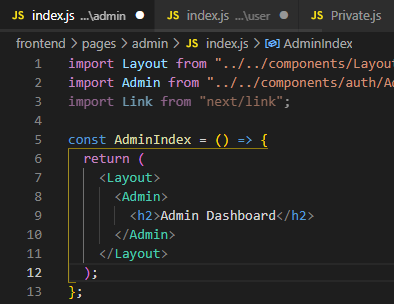
* Redirecting signed users to user dashboard.
* In **components**, go to **SigninComponent.js.**
* 
* Now protecting routes for both pages.

**=>Admin and user dashboard working, Task Completed… :)**

# **TASK-3**

# **Protecting routes for auth and admin user**

**STEPS :**

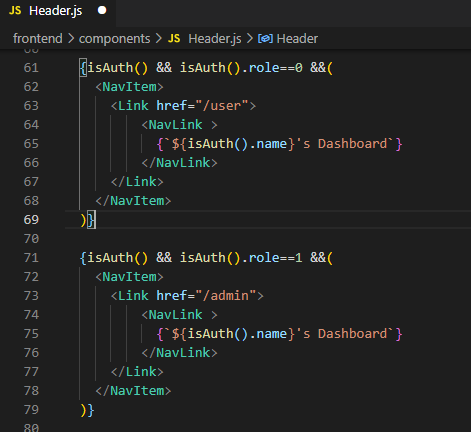
* In **frontend**, go to **components, auth** and create 2 components named **Private.js** and **Admin.js**.
* Creating Private component. If user not authenticated, redirect to signin page,otherwise return children props.
* 
* Admin component with auth check and also checking role of amin.
* 
* Now go to **pages** and apply changing in **user** **index.js**.
* Applying layout inside private component, so only logged user can access it.
* 
* Now go to **pages** and apply changing in **admin** **index.js**.
* 
* Now run the server [**http://localhost:3000**](http://localhost:3000/signin) If not singed in, try accessing user Dashboard using <http://localhost:3000/user> , it will redirect to <http://localhost:3000/signin>. (not giving access to unauthorized users.)
* If we sign in as **User** we can access user <http://localhost:3000/user>but we can't access admin <http://localhost:3000/admin>, we will be redirected to home.
* But if we sign in as **Admin** we can access admin <http://localhost:3000/admin> as well as user <http://localhost:3000/user>.

**=>Admin and user Routed Protected, Task Completed… :)**

# **TASK-4**

# **Show dashboard link in nav based on user role**

**STEPS :**

* Lets create a link that will take user to dashboard.
* In **frontend**, go to **components, auth** and change **Header.js**.
* 
* Now sign in and see dashboard links.

|  |
| --- |

**=>Dashboard links displayed , Task Completed… :)**

# 