



Facit | React Admin Dashboard Template

 Thank you for purchasing my theme. If you have any questions that are beyond the scope of this help file, please feel free to email via my user page contact form  [here](#). Thanks so much!

Project Title: Facit | React Admin Dashboard Template

Version: 2.0.0

Created: 2021-08-10

By: [OmtankeStudio.com](#)

Email:  info@omtankestudio.com

Note: Node version must be 16 or higher.

While explaining these documents, it is assumed that you have a basic knowledge of JS, HTML, CSS and ReactJS. You can learn more about ReactJS at <https://reactjs.org/>.

If you do not have nodejs installed on your computer, you can download versions 16 and above from <https://nodejs.org/>. When you install node js, npm is also installed with it.

If you want to use Yarn you can follow the instructions here <https://classic.yarnpkg.com/en/docs/install/>.

You are now ready to run the project.

Phase 1: Initiation

Run Demo

Enter Into Folder: `cd facit`

Install Dependencies

For **NPM**: `npm install`

or

For **YARN**: `yarn install`

Phase 2: Running

For **NPM**: `npm start`

or

For **YARN**: `yarn start`

Runs the app in development mode.

Open <http://localhost:3000> to view it in the browser.

Run the following code to run the Storybook. We recommend reading the documentation at <https://storybook.js.org/docs/react/get-started/introduction> to edit it.

Plain text

```
1 start-storybook -p 6006 -s public
```

Phase 3: Project Structure

Plain text

```
1  facit
2  |— .storybook
3  |— node_modules
4  |— public
5  |— src
6  |   |— App
7  |   |   |— App.js
8  |   |— assets
9  |   |— common
10 |   |— components
11 |   |— contexts
12 |   |— helpers
13 |   |— hooks
14 |   |— layout
15 |   |— pages
16 |   |— routes
17 |   |— stories
18 |   |— styles
19 |   |   |— components
20 |   |   |— custom
21 |   |   |— elements
22 |   |   |— generic
23 |   |   |— objects
24 |   |   |— settings
25 |   |   |— tools
26 |   |   |— utilities
27 |   |   |— vendors
28 |   |   |— style.scss
29 |   |— i18n.js
30 |   |— index.js
31 |   |— lang.js
32 |   |— menu.js
33 |   |— reportWebVitals.js
34 |   |— setupTests.js
35 |   |— steps.js
```

```
36 |— SvgIcons
37 |   └─ custom-icon.svg
38 |— .env
39 |— .env.development
40 |— .env.production
41 |— .eslintignore
42 |— .eslintrc
43 |— .gitattributes
44 |— .gitignore
45 |— .prettierrc
46 |— .stylelintignore
47 |— .stylelintrc
48 |— .svgrrc
49 |— package.json
50 |— README.md
51 └─ yarn.lock
```

Phase 4: Customize SCSS Variables

We have overwritten some variables of the bootstrap and added new variables.

Variables Used: `src/styles/settings/_index.scss` to update used variables.

Overwrite Bootstrap Variable:

https://github.com/twbs/bootstrap/blob/v5.1.3/scss/_variables.scss, you can copy the name of all variables here, type new value and overwrite it. Also don't forget to update the colors in the `src/.env` file.

Phase 5: New SVG Icon

Put the svg file you want to use as an icon in the "SvgIcons" folder and update its name as custom-NAME.svg. then run the `yarn icon` command. You can use the name as PascalCase as CustomName for components that support icons.

Phase 6: Component Structure

6. 1: index.js

JavaScript

```
1  // src/index.js
2  <Router>
3    <React.StrictMode>
4      <ThemeProvider>
5        <App />
6      </ThemeProvider>
7    </React.StrictMode>
8  </Router>
```

6. 1. 1: App.js

JavaScript

```
1  // src/App/App.js
2  <ThemeProvider>
3    <div className='app'>
4      <Routes>
5        {withoutAsidePages.map((path) => (
6          <Route key={path} path={path} />
7        ))}
8        <Route path='*' element={<Aside />} />
9      </Routes>
10    </div>
11    <Portal id='portal-notification'>
12      <ReactNotification />
13    </Portal>
14  </ThemeProvider>
```

❗ Add paths to the array (`withoutAsidePages`) that you don't want to be "Aside".

6.1.1.1: Aside.js

JavaScript

```
1  // src/layout/Aside/Aside.js
2  <aside className='aside'>
3    <div className='aside-head'>
4      <Brand {...props}/>
5    </div>
6    <div className='aside-body'>
7      <Navigation {...props} />
8      <NavigationLine />
9      <Navigation {...props} /> // You can use multiple
10     // ...
11   </div>
12   <div className='aside-foot'>
13     <User />
14   </div>
15 </aside>
```

6.1.1.2: Wrapper.js

JavaScript

```
1  // src/layout/Wrapper/Wrapper.js
2  <WrapperContainer>
3    <HeaderRoutes />
4    <Content />
5    <FooterRoutes />
6  </WrapperContainer>
7  <WrapperOverlay />
```



NOTE: React-Router-Dom 6 is used for routing. For detailed information <https://reactrouter.com/docs/en/v6/getting-started/overview>.

6.1.1.2.1: HeaderRoutes.js

JavaScript

```
1  // src/layout/Header/HeaderRoutes.js
2  import React from 'react';
3  import { Route, Routes } from 'react-router-dom';
4  import headers from '../routes/headerRoutes';
5
6  const HeaderRoutes = () => {
7      return (
8          <Routes>
9              {headers.map((page) => (
10                  <Route key={page.path} {...page} />
11              ))}
12          </Routes>
13      );
14  };
15
16  export default HeaderRoutes;
```

⚠ You can intervene here or set "src/routes/headerRoutes.js" here.


JavaScript

```
1  // src/routes/headerRoutes.js
2  const headers = [
3      {
4          path: 'path',
5          element: <Component />,
6          // Set it to "null" if you don't want it to happen
7          exact: true
8      },
9      // ADD MORE
10 ];
```

Example Header Component

JavaScript

```
1 <Header>
2   <HeaderLeft>...</HeaderLeft>
3   <HeaderRight>...</HeaderRight>
4 </Header>
```

 You can use any of the existing components within the HeaderLeft and HeaderRight components, or by changing the examples, to use whatever you need.

6.1.1.2.2: Content.js

JavaScript

```
1 // src/layout/Content/Content.js
2 <main className='content'>
3   <Suspense fallback={'Loading...'}>
4     <ContentRoutes />
5   </Suspense>
6 </main>
```

6.1.1.2.2.1: ContentRoutes.js

JavaScript

```
1 // src/layout/Content/ContentRoutes.js
2 import React, { lazy } from 'react';
3 import { Route, Routes } from 'react-router-dom';
4 import contents from '../..//routes/contentRoutes';
5
6 const PAGE_404 = lazy(() => import(
7   '../..//pages/presentation/auth/Page404'));
8 const ContentRoutes = () => {
9   return (
```



```

9      <Routes>
10        {contents.map((page) => (
11          <Route key={page.path} {...page} />
12        ))}
13        <Route path='*' element={<PAGE_404 />} />
14      </Routes>
15    );
16  };
17
18  export default ContentRoutes;

```

⚠ You can intervene here or set "src/routes/contentRoutes.js" here.

JavaScript

```

1  // src/routes/contentRoutes.js
2  const content = [
3    {
4      path: 'path',
5      element: <Component />,
6      // Set it to "null" if you don't want it to happen
7      exact: true
8    },
9    // ADD MORE
10 ];

```

Example Page Component

JavaScript

```

1  <PageWrapper {...props}>
2    <SubHeader>
3      <SubHeaderLeft>...</SubHeaderLeft>
4      <SubHeaderRight>...</SubHeaderRight>
5    </SubHeader>
6    <Page>
7      // PAGE CONTENT HERE

```

```
8     </Page>
9   </PageWrapper>
```

6.1.1.2.3: FooterRoutes.js

JavaScript

```
1  // src/layout/Footer/FooterRoutes.js
2  import React from 'react';
3  import { Route, Routes } from 'react-router-dom';
4  import footers from '../../routes/footerRoutes';
5
6  const FooterRoutes = () => {
7    return (
8      <Routes>
9        {footers.map((page) => (
10          <Route key={page.path} {...page} />
11        ))}
12      </Routes>
13    );
14  };
15
16  export default FooterRoutes;
17
```

❗ You can intervene here or set "src/routes/footerRoutes.js" here.

JavaScript

```
1  // src/routes/footerRoutes.js
2  const footers = [
3    {
4      path: 'path',
5      element: <Component />,
6      // Set it to "null" if you don't want it to happen
7      exact: true
8    },
9  ]
```

```
9      // ADD MORE
10   ];
```

Phase 7: Use of Components

Detailed use of components is explained at <https://facit.omtankestudio.com/> and <https://facit-story.omtankestudio.com/>.

Example for Button

JavaScript

```
1   <Button
2       className={ String }
3       color={ String } // 'primary' || 'secondary' ||
    'success' || 'info' || 'warning' || 'danger' || 'light' ||
    'dark' || 'link'
4       isOutline={ Boolean }
5       isLink={ Boolean }
6       isLight={ Boolean }
7       isActive={ Boolean }
8       isDisable={ Boolean }
9       size={ String } // 'sm' || 'lg'
10      rounded={ String || Number } // 'default' || 0 || 1 || 2
    || 3 || 'bottom' || 'top' || 'circle' || 'end' || 'start' ||
    'pill'
11      shadow={ String } // 'none' || 'sm' || 'default' || 'lg'
12      hoverShadow={ String } // 'none' || 'sm' || 'default' ||
    'lg'
13      tag={ String } // 'button' || 'a' || 'input' || 'link'
14      type={ String } // 'button' || 'submit' || 'reset'
15      to={ String }
16      href={ String }
17      target={ String } // '_blank' || '_self' || '_parent' ||
    '_top'
18      {...props}>
19      ...
```

You can review the sample usage [here](#) , make changes, and see instant previews. If you want, you can share the link with your team.

⚠ You can go to the docs tab from the top left and click the "**Show Code**" button at the bottom left of the preview to copy the state you will use in your project and quickly paste it to the place you will use in your project.

Phase 8: Classic Design

The "modern" design is set by default, you can set the REACT_APP_MODERN_DESGIN variable in the .env file to "false" to switch to the "classic" design.

Credits

Free purchases of used 3D shapes at <https://www.shapefest.com/>.

Free purchases of used 3D shapes at <https://iconscout.com/3d-pack/abstract-shapes/>.

The 3D human used can be purchased at <https://www.shapefest.com/>.