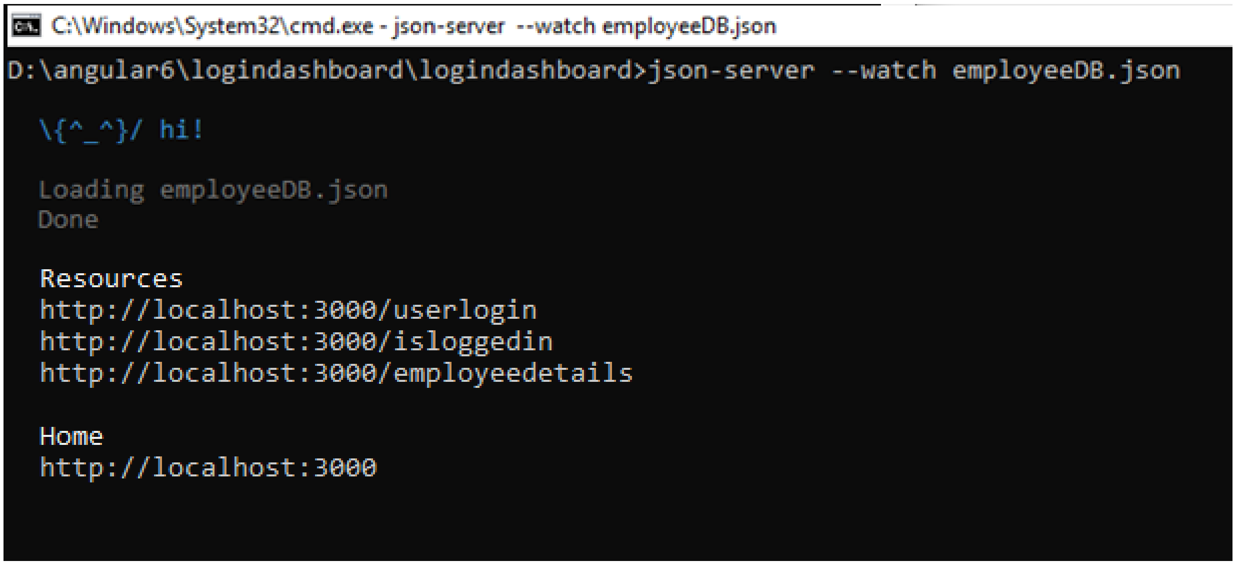
Login POC Handbook

Contents

JSON Server to initialize fake Database.

To run fake Database and API execute command “**json-server –watch employeeDB.json**” on terminal. Fake DB file “employeeDB.json” is placed along with “package.json” and “angular.json” in logindashboard” folder.

Execute command - json-server –watch employeeDB.json



C:\Windows\System32\cmd.exe - json-server --watch employeeDB.json

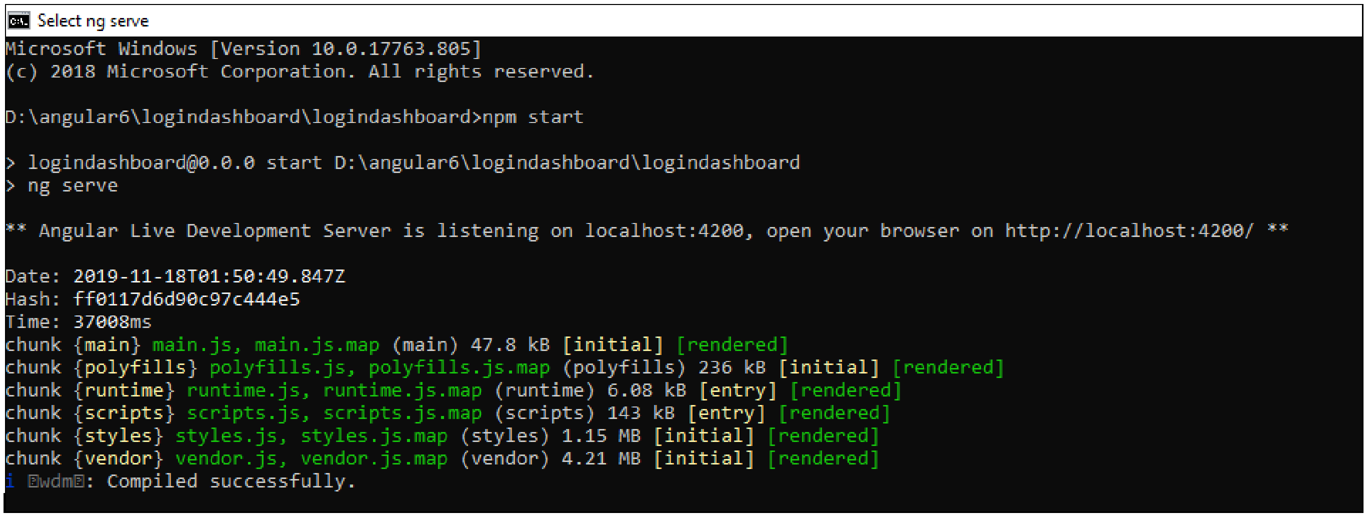
```
D:\angular6\logindashboard\logindashboard>json-server --watch employeeDB.json

  \{^_^}/ hi!

  Loading employeeDB.json
  Done

  Resources
  http://localhost:3000/userlogin
  http://localhost:3000/isloggedin
  http://localhost:3000/employeedetails

  Home
  http://localhost:3000
```

Fig: All the three APIs are executed on JSON server

Run npm

To execute angular 6 application, start the npm server.

Execute command – npm start

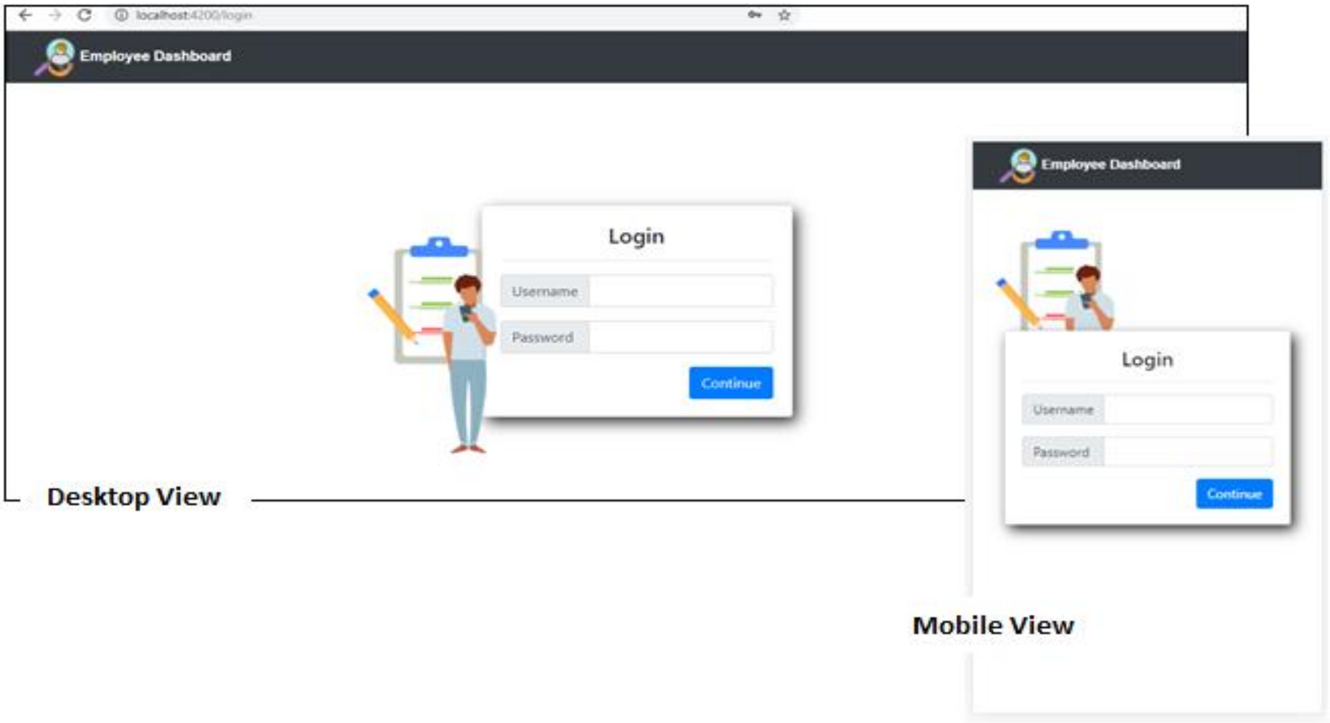


```
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

D:\angular6\logindashboard\logindashboard>npm start

> logindashboard@0.0.0 start D:\angular6\logindashboard\logindashboard
> ng serve

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

Date: 2019-11-18T01:50:49.847Z
Hash: ff0117d6d90c97c444e5
Time: 37008ms
chunk {main} main.js, main.js.map (main) 47.8 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 236 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.08 kB [entry] [rendered]
chunk {scripts} scripts.js, scripts.js.map (scripts) 143 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 1.15 MB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 4.21 MB [initial] [rendered]
i □wdm□: Compiled successfully.
```

Login screen

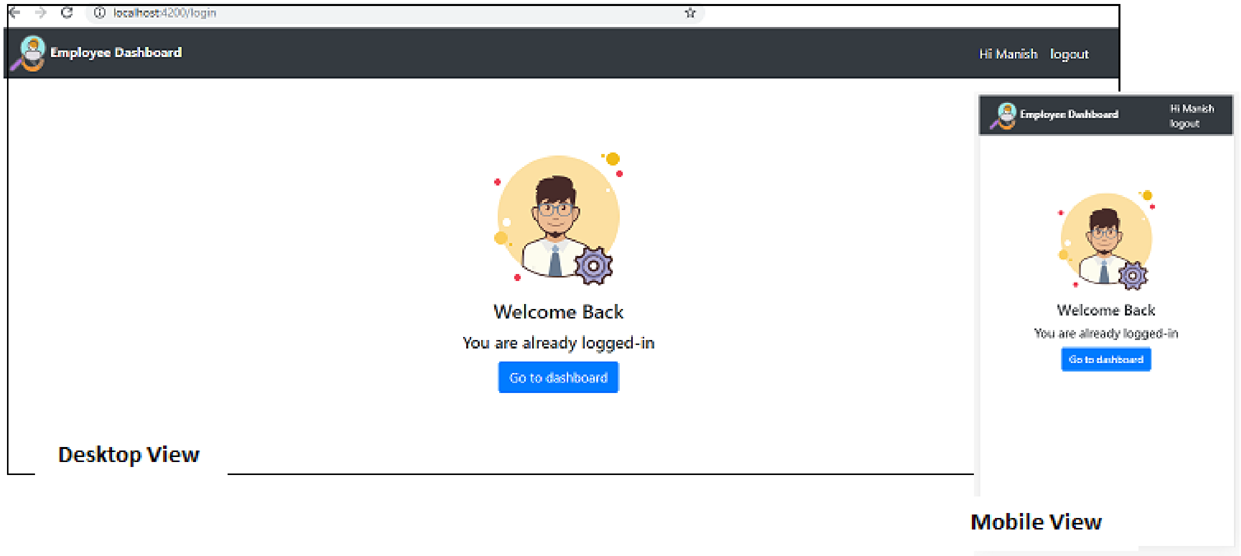
To access login screen, open URL http://localhost:4200 or with your local node JS port number.

It will open login screen where you need to fill your credentials to proceed further.



Already Logged-in Screen

In case, if you are already logged-in, it will display "Already logged in".



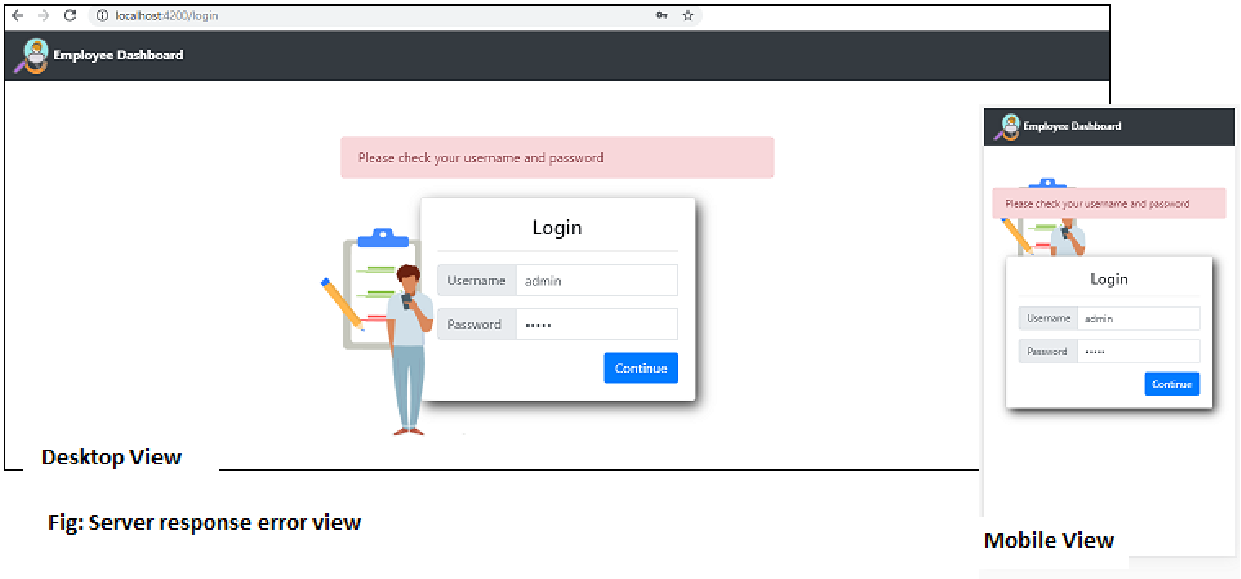
Welcome Back

You are already logged-in

Fig: Already logged-in Screen

Invalid Login

Server Response: If you will submit invalid credential, Server Error message will display on the top of the login screen.


localhost:4200/login

Employee Dashboard

Please check your username and password

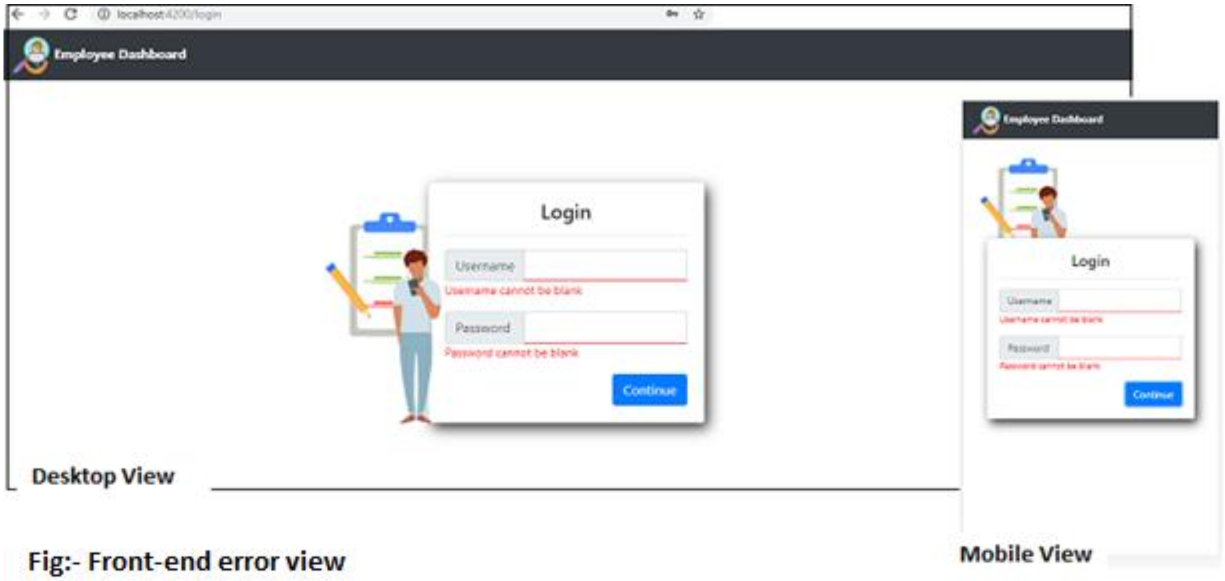
Login

Username: admin

Password: •••••

Fig: Server response error view

Front-end Error: If you will try to submit blank values then error message will display below each input box.


localhost:4200/login

Employee Dashboard

Login

Username

Username cannot be blank

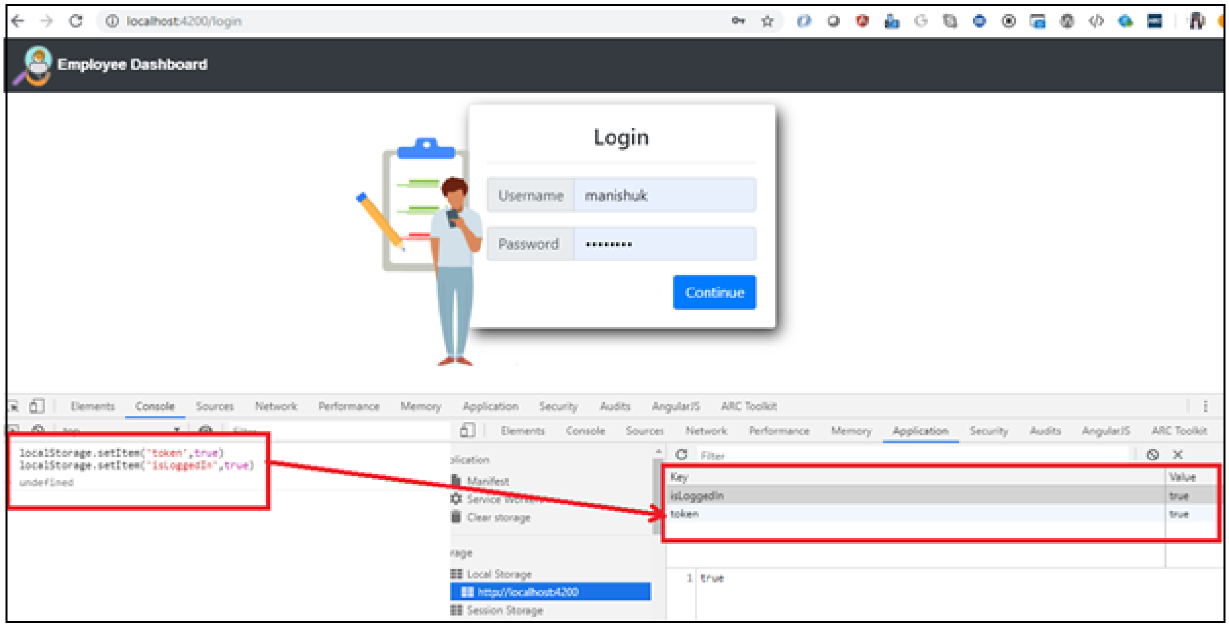
Password

Password cannot be blank

Fig:- Front-end error view

Login Authentication

I wrote a script that modifies the employeeDB.json on each login and logout along with local Storage. Idea of using and modifying fake DB JSON is to implement the replica of actual server functionality (where we modify values in DB). This increases the security and makes the login difficult to reproduce without proper authentication.



Username: manishuk
Password: •••••••••

```
localStorage.setItem('token',true)
localStorage.setItem('isLoggedIn',true)
undefined
```

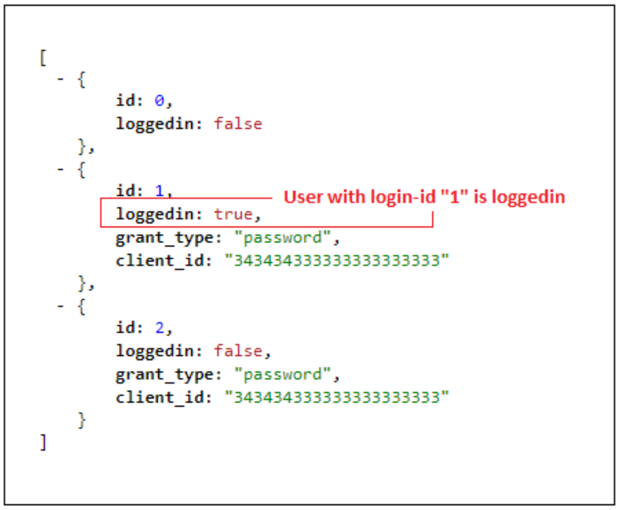
Key	Value
isLoggedIn	true
token	true

```
1 true
```

Fig: Manipulating stored values from console to access pages without login

If user tries modifying local storage values through console, then the code checks actual login status in “employeeDB.json”. If loggedin status is true in employeeDB then the user is allowed to proceed further else the user gets “You are not authorized” page.

After successful login, dashboard page will display.



```
[
  - {
        id: 0,
        loggedin: false
    },
  - {
        id: 1,
        loggedin: true,
        grant_type: "password",
        client_id: "34343433333333333333"
    },
  - {
        id: 2,
        loggedin: false,
        grant_type: "password",
        client_id: "34343433333333333333"
    }
]
```

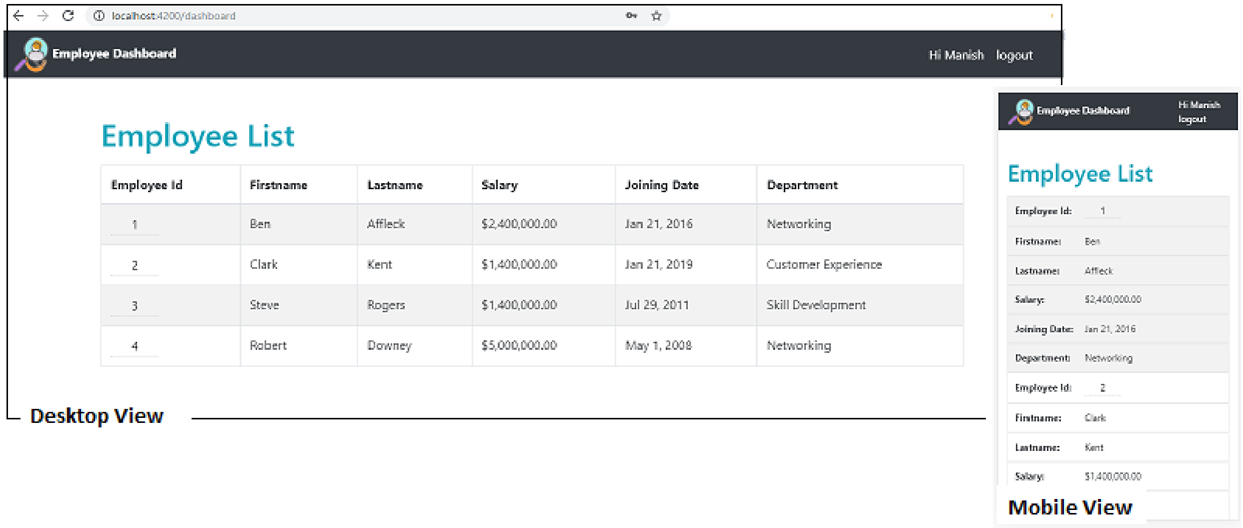
User with login-id "1" is loggedin

Fig: Authenticating user login from server file

Dashboard

In dashboard page, you can find one Employee list table and common header that includes “Username”, logout option and logo.

Employee dashboard will appear in table view in desktop and in Stack view in mobile with Username and logout option.



Employee List

Employee Id	Firstname	Lastname	Salary	Joining Date	Department
1	Ben	Affleck	$2,400,000.00	Jan 21, 2016	Networking
2	Clark	Kent	$1,400,000.00	Jan 21, 2019	Customer Experience
3	Steve	Rogers	$1,400,000.00	Jul 29, 2011	Skill Development
4	Robert	Downey	$5,000,000.00	May 1, 2008	Networking

Desktop View

Employee List

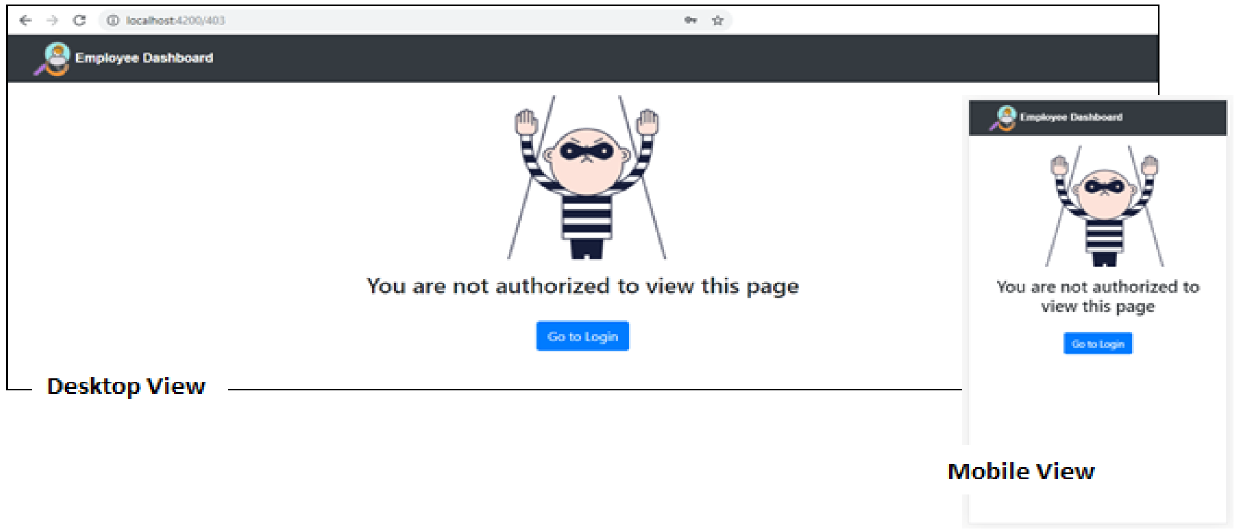
Employee Id: 1
Firstname: Ben
Lastname: Affleck
Salary: $2,400,000.00
Joining Date: Jan 21, 2016
Department: Networking
Employee Id: 2
Firstname: Clark
Lastname: Kent
Salary: $1,400,000.00

Mobile View

If user tries to access dashboard page without login, it will redirect to “You are not authorized” Page with option “Go to login”.

Unauthorized access control for pages

In case user tries to open any page without login, then the page will redirect to “You are not authorized” Page with option “Go to login”.

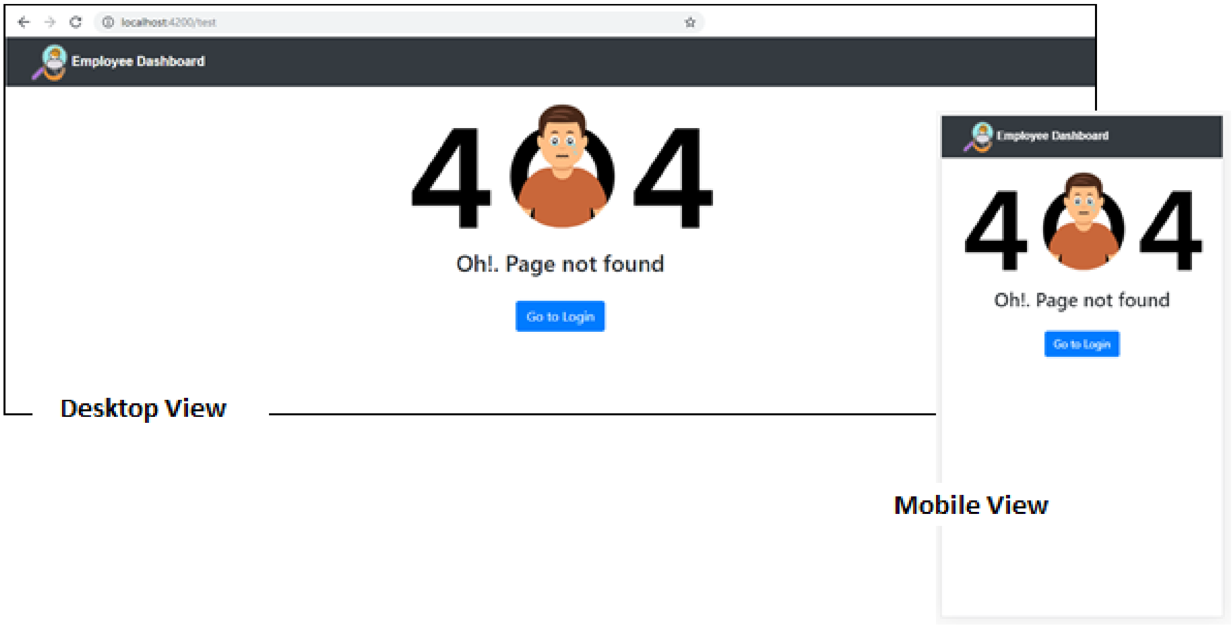


You are not authorized to view this page

You are not authorized to view this page

Page not found (404)

In case user tries to access unavailable page, it will redirect to “Page not found”



Oh!. Page not found

Oh!. Page not found