# **Interactive Web Portal**

# 1 Introduction

## 1.1 Purpose

The purpose of this document is to outline the technical design of the interactive web portal and provide an overview for the interactive web portal implementation.

Its main purpose is to -

* Provide a basis for the Interactive web portal’s design and development
* Provide the link between the Functional Specification and the detailed Technical Design documents
* Detail the functionality which will be provided by each component or group of components and show how the various components interact in the design

## 1.2 Scope

The Application Design outlined in this document builds upon the scope defined in the requirements phase.

# 2 Guiding Principles

Guiding principles provide a foundation upon which to develop the target architecture for the interactive web portal, in part by setting the standards and measures that the tool must satisfy.

Some of the guiding principles that will be followed during the interactive web portal design and development are outlined below.

* **Scalable**

The Interactive web portal is designed to support distribution to multiple servers.

* **Open Source**

Tools and technologies adopted in the development of the Interactive web portal are all opensource and require no licenses.

# 3 Development Infrastructure

The Interactive web portal has been developed using the following

* React: Used for frontend development of the app because of its overall simplification of the process of scripting components and quicker rendering, among others.
* Yarn
* Highcharts: Used for creating data visualization.
* Leaflet: Used for creating data visualization as maps
* Material UI: Used for styling the app user interface
* Material Design: Used for styling the app user interface
* DHIS2 d2 Library: Provides DHIS2 applications functions

# 4 Installation and development

To install and deploy the interactive web portal, download the source code and make a copy to your local machine.

In the project directory, you can run:

yarn start

Runs the app in the development mode. Open [http://localhost:3000](http://localhost:3000/) to view it in the browser. The page will reload if you make edits. You will also see any lint errors in the console.

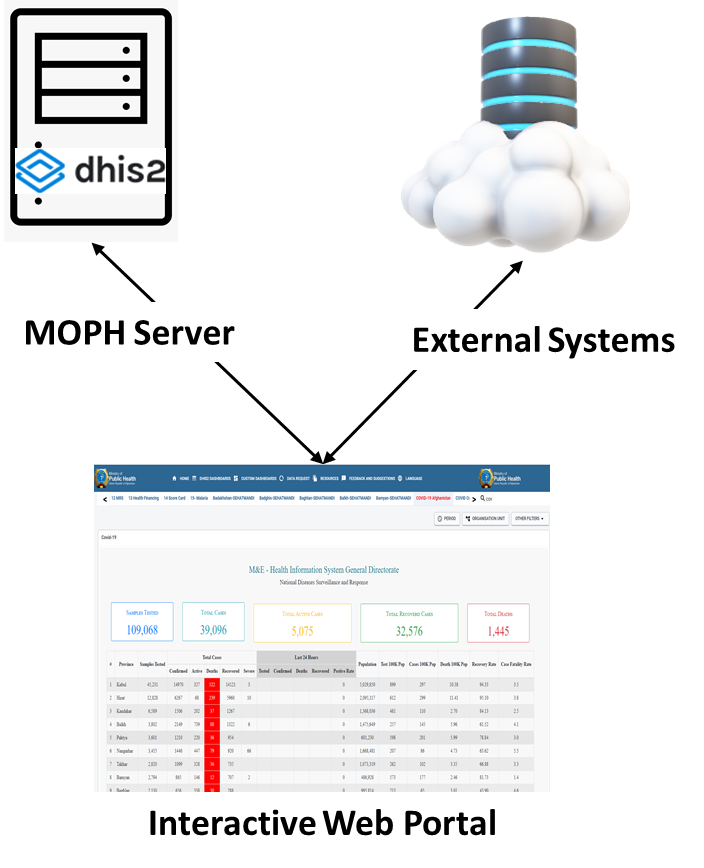
To deploy the application, you can run:

yarn build

Builds the app for production to the build folder. It correctly bundles React in production mode and optimizes the build for the best performance. The build is minified, and the filenames include the hashes.

Your app is ready to be deployed!

# 5 Topology Diagram

The diagram below provides an illustration of the System Architecture along with various system components that will be used in architecting the Interactive web portal.

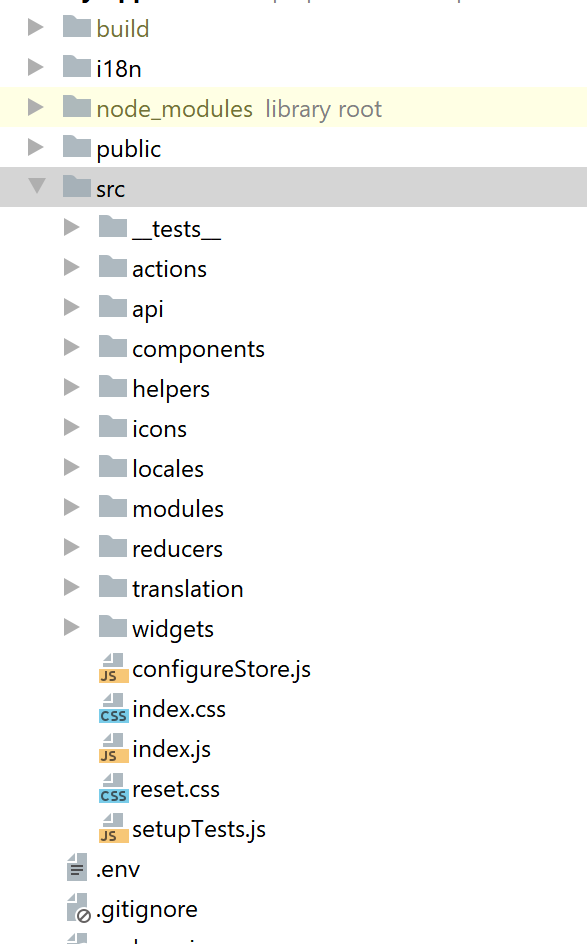
Interaction of software components along with its responsibilities is explained below –

* **MOPH DHIS2 Server**
  + Provides data and dashboards
  + Stores the Interactive web portal custom dashboards configurations
* **External Systems**
  + External systems provide the custom dashboard with data.
* **Interactive web portal**
  + Visualization of the data.

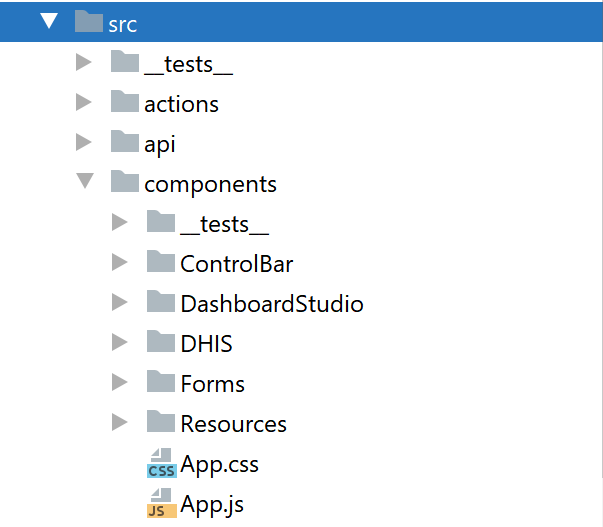
The Interactive web portal App follows the ReactJS system architecture, so the following are the existing components.

* **Package (.json)**: Specifies the application dependencies which are imported or installed using the NPM or YARN package manager
* **Index (.js)**: Starting point of the **application**, which calls the app.js
* **App (.js)**: Implements the navigation bar using the BrowserRouter class component for client-side routing with URL segments - This calls the navigation bar component with specific url links to the different pages that will be navigated.
* **Manifest (.webapps**): DHIS2 Apps need the manifest file to describe the contents of the app. It contains information about the type of DHIS2 app (Standard app, Dashboard Widget or Tracker Widget), version information, icon for the app and the developer information.
* **Components (.jsx)**: Building blocks of a React app which can be a JavaScript class or function that optionally accepts inputs i.e. properties (props) and returns say, a React element describing how a section of the UI (User Interface) should appear.

# 6 Code Structure

The Interactive web portal file structure is as shown below

## Components

Components are grouped into 5 folders

* Control Bar Folder: Contains components for the definitions of the application header menu
* DHIS Folder: Contains components for the visualization of the DHIS2 dashboards
* Dashboard Studio Folder: Contains components for the
* Forms Folder: Contains components defining the user feedback and suggestions form and the user data request form
* Resource Folder: Contains component for the visualization of the resources from DHIS2

## The Helper File

**ApiHelper (.js)**: A helper javascript file was added to perform the following functions:

1. **chartDataFetchHelperCsv(url,source,headers):** A user defined function for fetching data.
2. csvtojson (data, delimiter = ','): A user defined function to convert data fetched from csv to JSON.

## The Component File

**Components (.jsx)**: Three components were developed for the Interactive web portal app. These included: Navigation (HeaderMenu.js), DHIS2 (Dashboard.jsx) and Custom Dashboards (Chartstudio.jsx). They are described as follows:

1. Navigation component - It is a class component, called ‘HeaderMenu’, containing the header menu design with two menuitems. It implements the click functionality for opening the respective visualizations when a specific menu item is clicked by a user.
2. DHIS Component: It is a class component and it includes the following components:
   1. ViewDashboard: user defined class component that returns DHIS2 dashboards
   2. HorizontalMenu: user defined class component that returns the DHIS2 dashboards as menu items
3. Custom Dashboards: It is a class component and it includes the following components
   1. ChartStudio: user defined class to visualize the custom defined charts
   2. Designer: user defined class component for the development of the custom dashboard

# 7 User interfaces

## Landing Page

After loading the application, the user is directed to the Interactive web portal homepage. The landing page displays the DHIS2 dashboards.

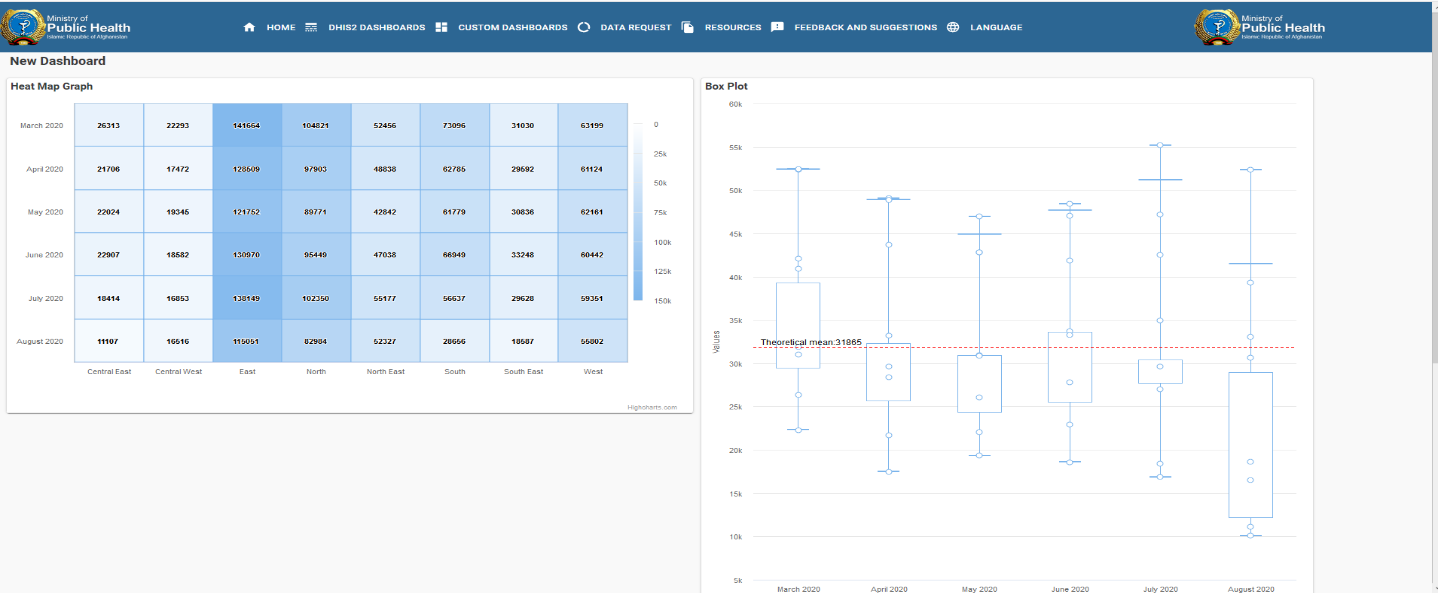
The landing page has the features below

* Horizonal navigation menu to allow user to switch between the available DHIS2 Dashboards
* Search bar: This allows for user to search through the available dashboards
* Filters: The filter buttons provide an interface in which users can select preferred parameter, effecting the data displayed by the dashboard



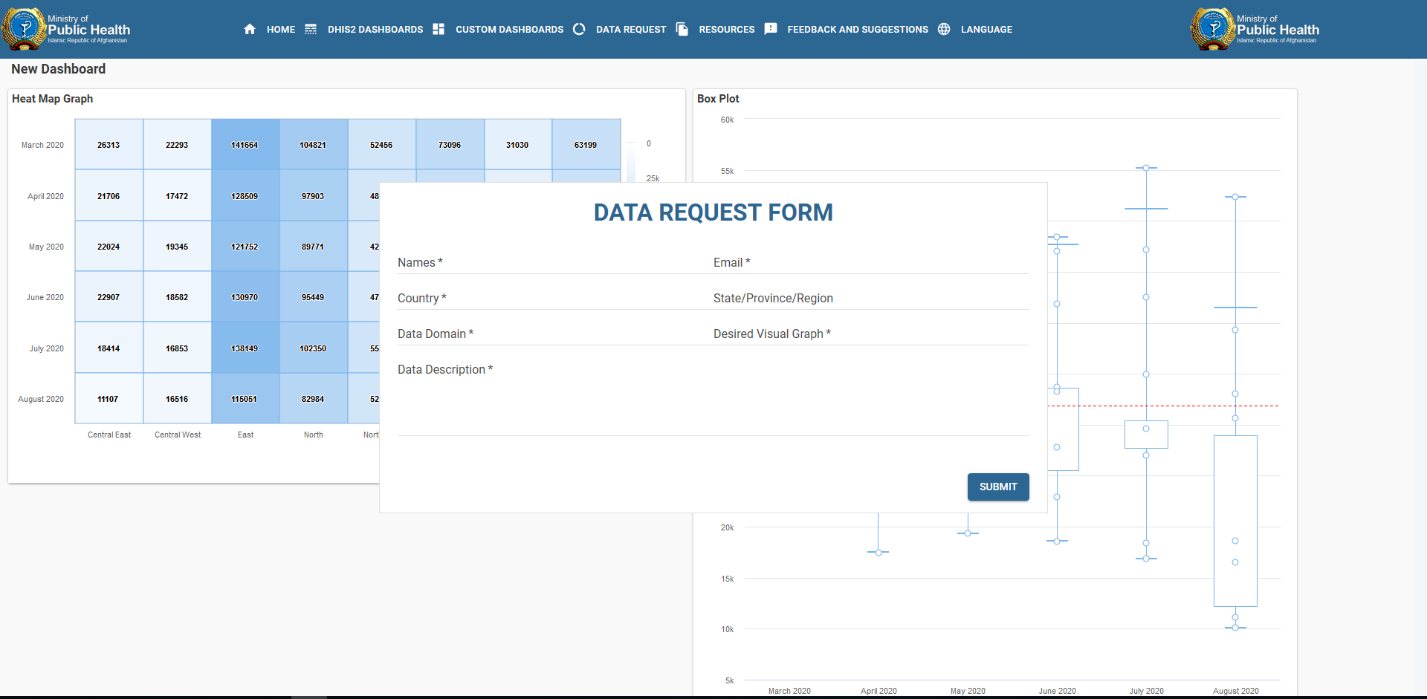
## Custom Dashboard

The custom dashboard menu item provides navigation to available customized dashboards available to the user to view.



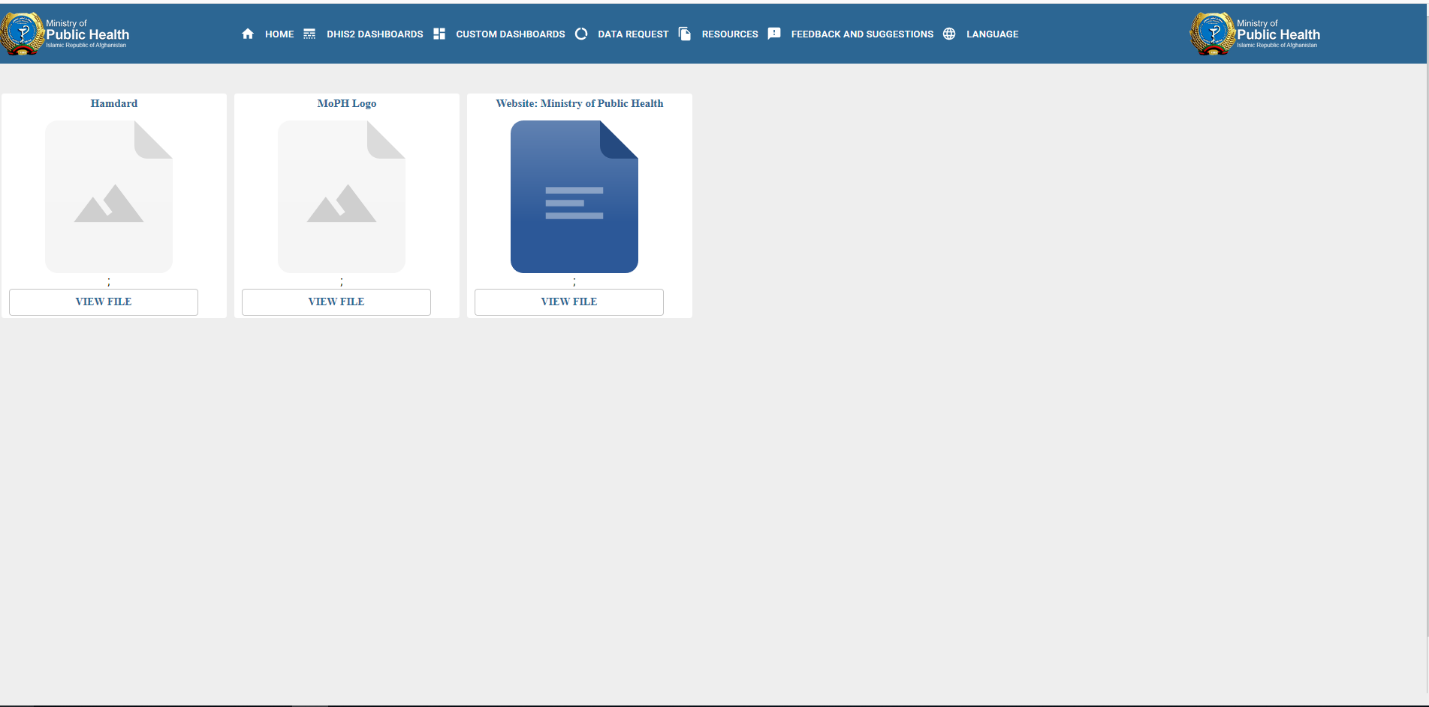
## Data Request Form

The data request menu item opens a window where user can make data request to the MOPH team by filling in the form and submitting the form.



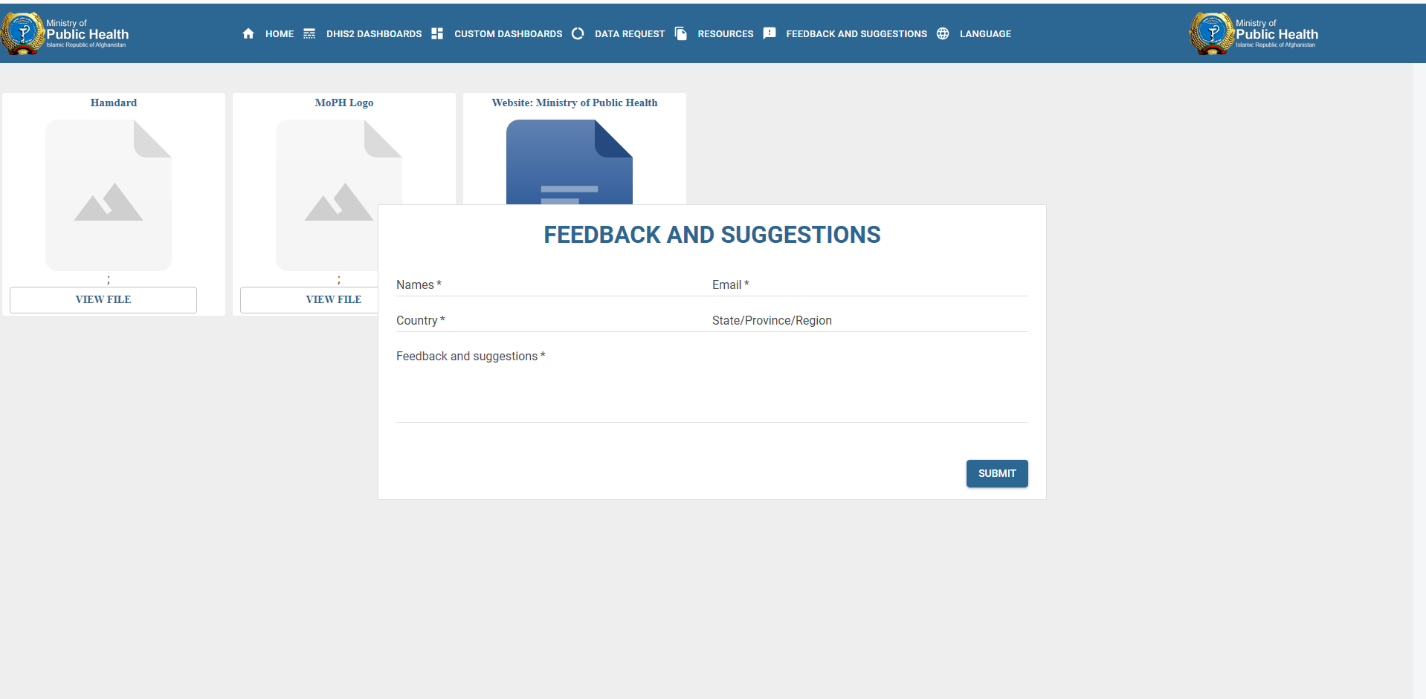
## Resource Page

The resource provides and interface where users can download, or view resources shared by the MOPH.



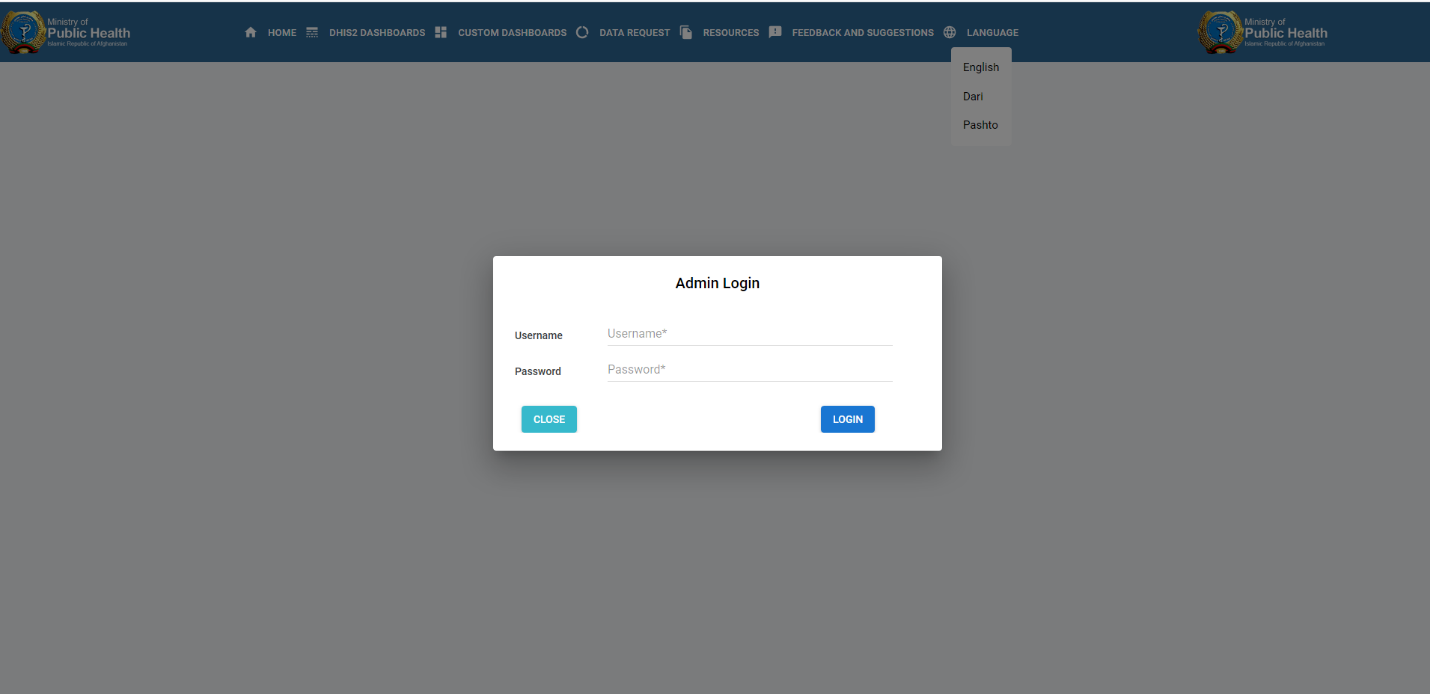
## Feedback and Suggestions Form

The Feedback and Suggestion menu item opens a window where user can share their feedback or make suggestion to the MOPH team by filling in the form and submitting the form.



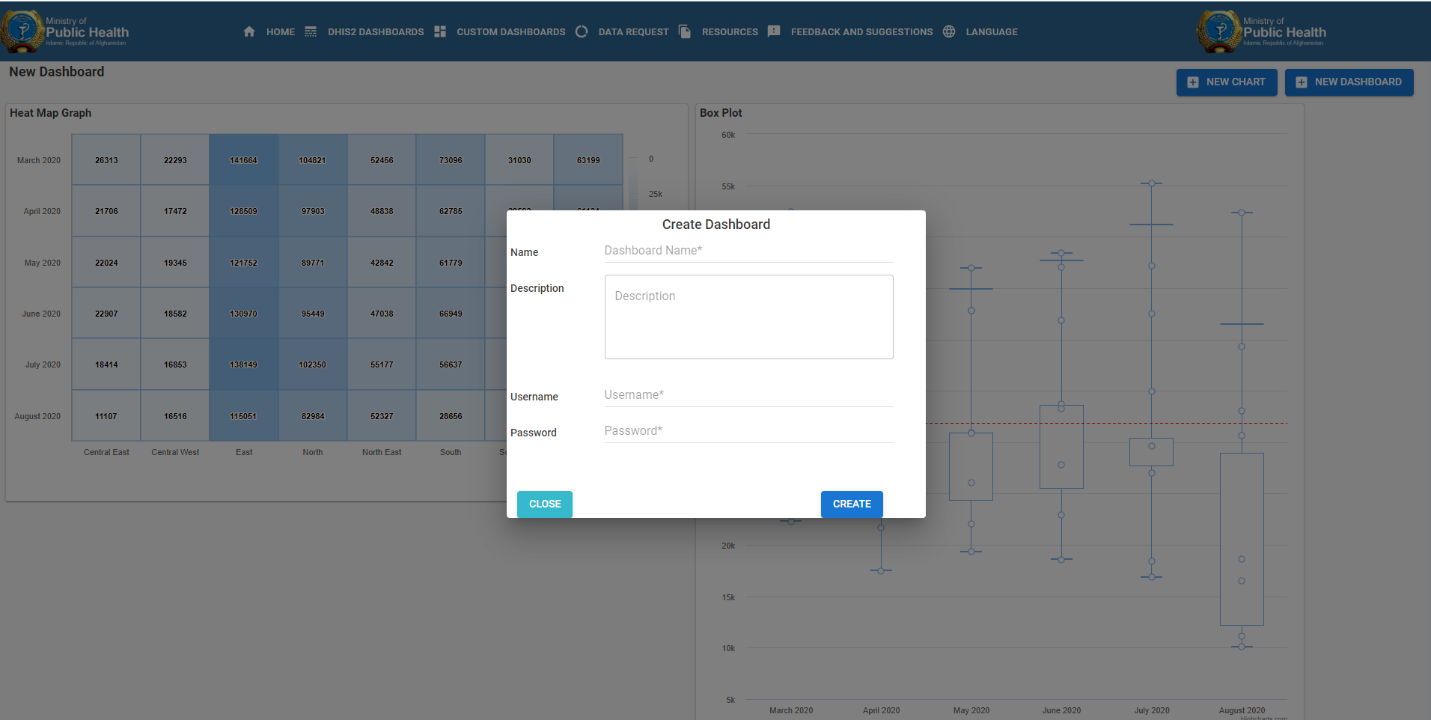
## Admin Login

The admin login provides an interface to login to the portal, this allows the user to create new dashboards or add new charts to a dashboard.



## New Dashboard

Once logged into the admin portal the user can view New Dashboard button. This opens a window where admin user can fill in details for the new dashboard and initiate the creation of the new dashboard.



## New Chart

Once logged into the admin portal the user can view New Chart button. This opens a window where admin user can fill in details for the new chart and initiate the creation of the new chart by clicking on the save button.

