**Project Dashboard**

**Introduction:**

To create a Website with Multiple Widgets and Dashboards using HTML, CSS, Java-Script, JSON and Power Bi..

**Project Dashboard Overview:**

The Project Dashboard web application has 6 Functional Buttons, One live Dashboard that collects live information created using Power Bi.

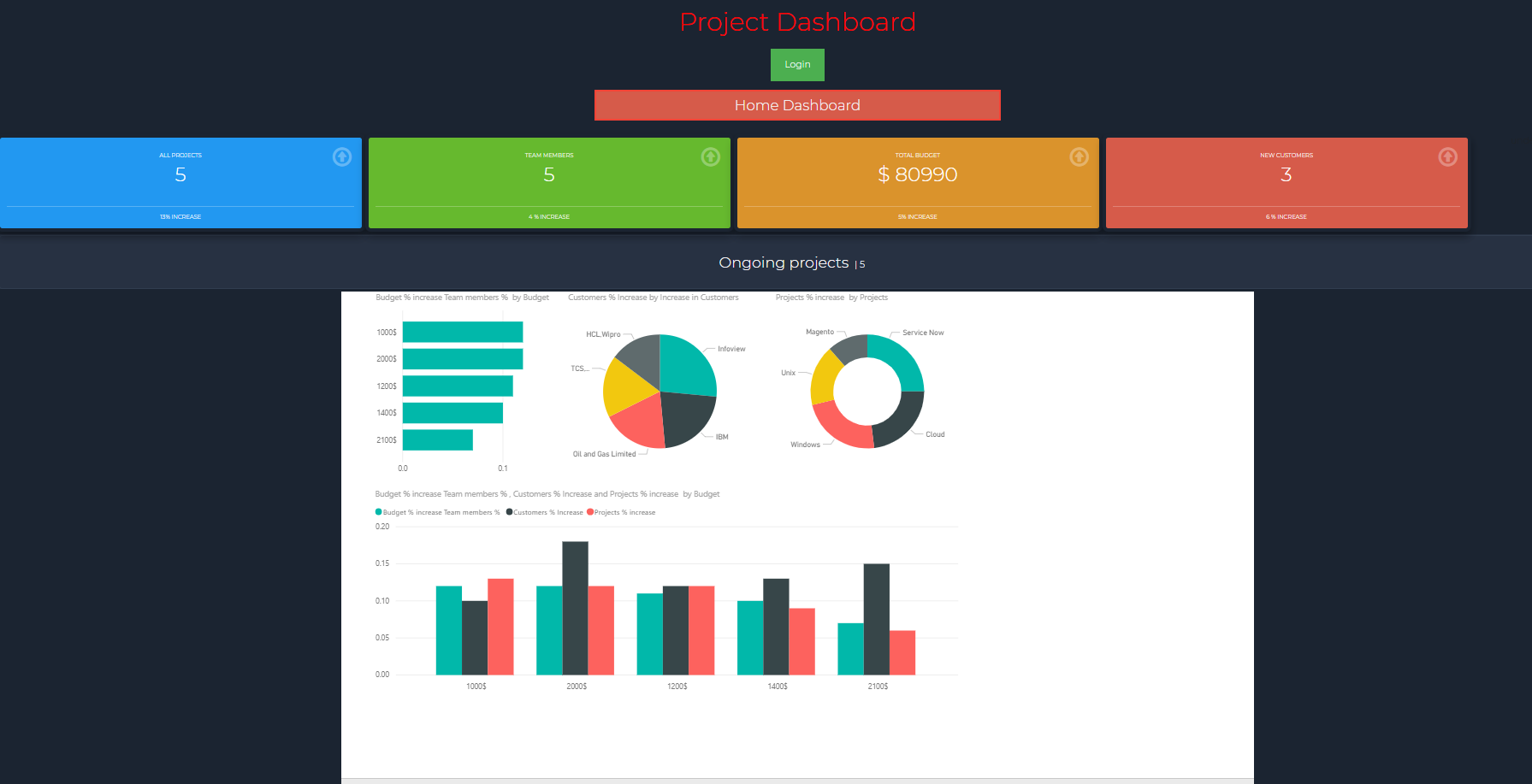
4 Widgets that collects live information from a Server/Database (Here we use JSON Data as live server data).

Out of the 6 Functional Buttons, The Home Dashboard Button displays a Dashboard/Table of live data( again using JSON as we don’t have a Database server to server data live using ajax calls).

The remaining 4 buttons are associated with the widgets/cards that display required information.

Below is the Website outlook.

**Project Dashboard Front end page:**



The actual JSON file is the one that actually servers with input data in the place of a server/database.

**Overview of Project Dashboard:**

**Data:** The data of the widget, coming from Dashboardcards.json file located in the Model part of the Website Folder.

**HTML:** The page which hosts the widgets,buttons and Dashboard. Located in the View part of the Website Folder.

**JavaScript:** The main functionality of parsing and displaying data in the website is done by 6 Java scripts stored in the Model part of the Website Folder.

**Images and Style:** Presentation of the overall website done by style.css stored in the view part of the website Folder.

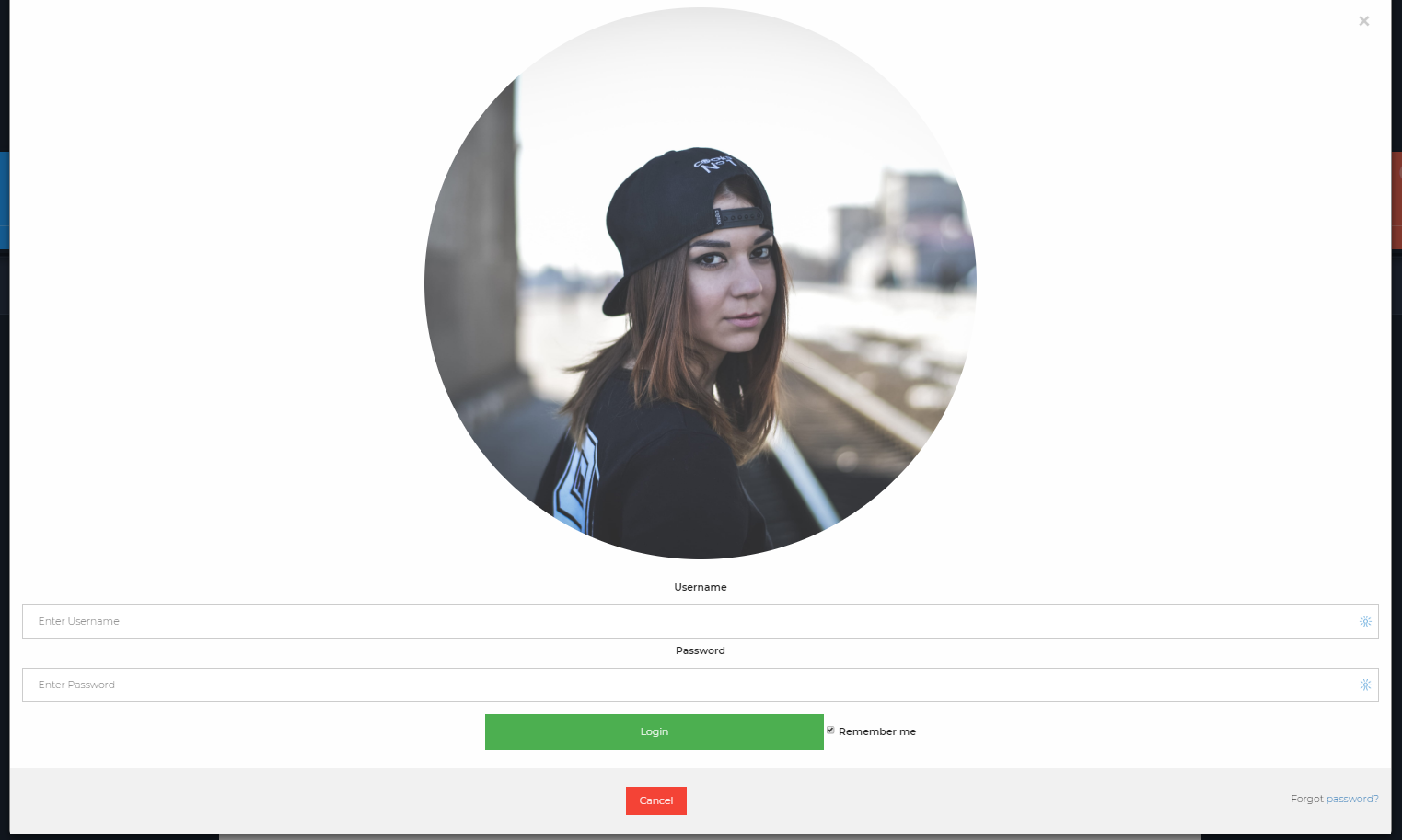
**Live Dashboard:** The Live Dashboard is created using Power Bi Tool and hosted on the Power Bi Trial account and Imported using the Power Bi API Call.

**App Requirements of the Project Dashboard :**

* A Web page (index.html)
* Bootstrap JavaScript file (bootstrap.js) which instantiates the Styling of the widgets and cards.
* The core functionality of the Website which takes the data and displays it is done using Multiple Java Scripts.
* JSON data file which provides the live data.
* The presentation of the Widget is done using CSS (styles.css)
* Power Bi Trial Version to create a Dashboard and use it by using an API call to Display it.

**Functionality:**

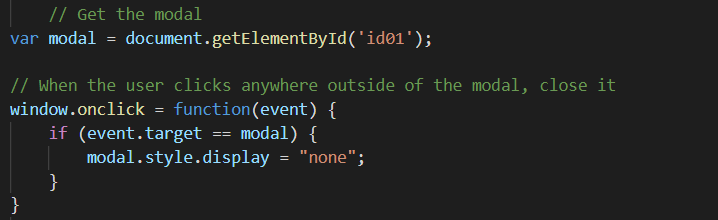
**Login Page :** The Web page has a built in login portal that we can use to login into the Website.



We can give in a username and password to login.

In This Website I would not be able to store Username and Password anywhere so this is a dummy login page with no actual functionality other than a display.

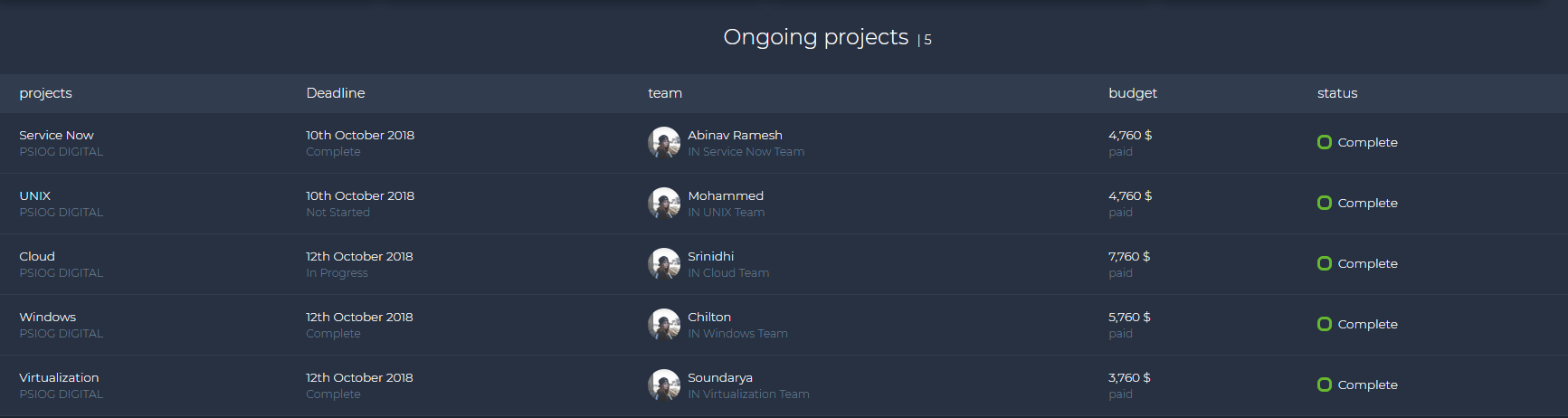
This Modal Function utilized a Java Script function to call as on click.



**Home Dashboard :**

The Web Page has a Home Dashboard Button that shows all the live data feed collected from the Json File.

Below is the Image that shows the Home Dashboard.



The Home Dashboard shows data like Projects,Deadline,Team,Budget and Status of the Project.

We use a for loop based Java Script Function to display and call data to display into the table.

Which collects live data from a JSON File.

**Cards/Widgets:**

Here we have written a HTML code as Div element that will display a card that displays the Projects and the percentage increase.



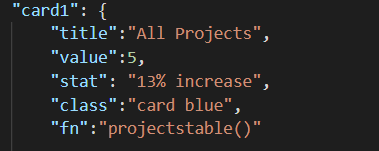
The id ‘cards ‘ calls for a Java Script Function that creates a table again using for loop function.



We use 4 different java script functions to call data from table.json and use a for loop to display certain live feed information.

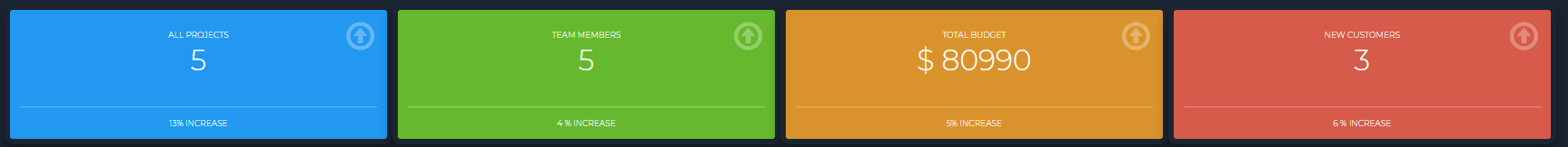


All the 4 cards are created using for loop and this can be reused by just feeding data into the json file.



This enables reuse ability as we can add another input into JSON using the same above parameters as an array and the Java Script function will automatically create a new card and add it there.

The actual cards/widgets are shown below.

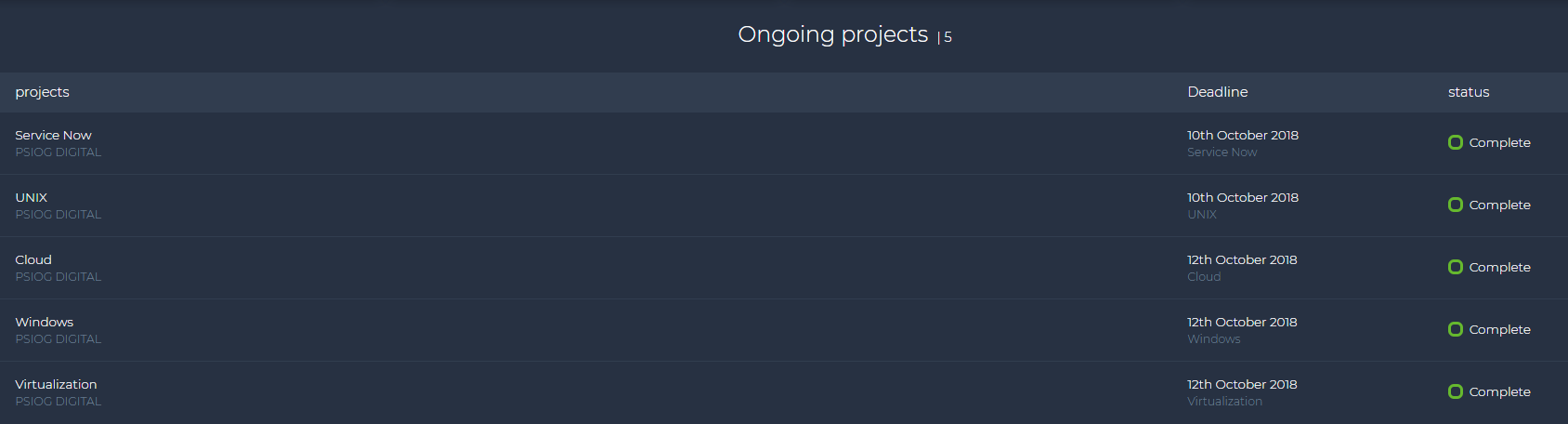


**Widgets on click functions:**

The main dashboard can show a lot of data. In case if we need linear data we had created a on click function to reduce and show precise information.

The On click function is also integrated along with the cards JSON file as “fn”.

4 functional JS Scripts are shown above as well.

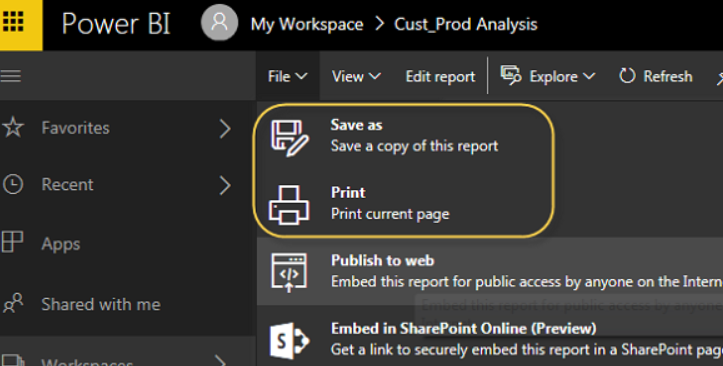


Clicking on the Projects card widget will call for projectstable() function which is integrated in the projectscard.js script. And as shown above displays certain information as requested.

**Power Bi Dashboard:**

I had used a table.txt file created using note pad, installed power BI, created a new account and imported the Text Document to create a Dashboard. This Power Bi is imported into my test account.

Now Login into my web account and click on Workspace.



File > Publish to web



Now I had added the iFrame element to my Index.HTML and it is being displayed.

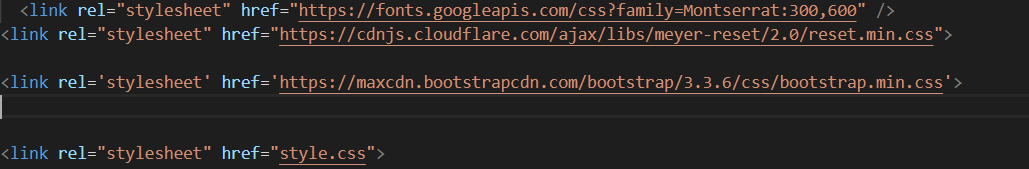


**Display of all the Data:**

Now this is done completely using CSS and Bootstrap.

**Bootstrap:**

I had imported Boot Strap and it’s functions to make it easy and lightweight for Dashboards.

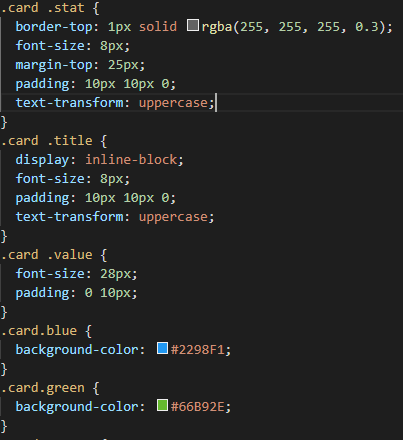


We Import Bootstrap templates using the same format as above.

**Presentation with HTML, CSS and Images:**

The first thing the bootstrap code loads is the style sheet because it should be loaded before we use it in the widget code. The presentation of the widget consists of both style sheet and the images which are used to display widget data. For our widget, let’s create a style.css and image(s) to be used for widget and background.

Create a new file Style.css and add it to your project. A separate stylesheet file allows you to change the look and feel of the widget without touching the widget functionality.



What happens is we had applied class as card on the DIV Element.

Card is parent class and blue is child class

And inside the DIV Element we have title, value and stat. .card .value and so on are applied as css styles in all the elements.

**JavaScript Object Notation (JSON) Data:**

* We use 2 JSON Files here to provide live data.
* We are using this mainly because we don’t have a live server/database to serve or provide live inputs.

**Concepts Learned /Implemented :**

* HTML
* CSS
* JAVA Script
* JSON
* FOR Loops
* Bootstrap
* Power Bi
* On Click functions

**Summary:**

In this White Paper I had given a brief overview of overview of the app requirements and functionality, concepts learned and implemented.

**License:**

This article, along with any associated source code and files, is licensed under **The Code Project Open License (CPOL).**

**Declaration:**

I hereby declare that the Project work entitled “Working with Widgets” is a record of an original work done by me under the guidance of Srinidhi Ramani, Mentor, Psiog Digital Pvt.ltd.

Abinav Ramesh

　　Psiog Digital