

HUL Dashboard Analytics

Version 1.0

Atos – For Internal Use
Copyright © 2016 Atos India Private Limited.

All rights reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner.

Document control

Issue control

Author	Role	Version/Date
Kiran Joshi	Sr. Angular Js (Front-end UI)Developer	1.0 on 05/05/216

Reviewers	Role	Release Date

Version history

Version	Date	Comment
1.0	05/05/2016	Initial version With Level Management

Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope of this document	4
1.3	Version & Technology Used	4
1.4	Definitions, Acronyms and Abbreviations	4
2.	Project Introduction	Error! Bookmark not defined.
2.1	Introduction	Error! Bookmark not defined.
3.	Login	6
4.	Regional Trend	9
5.	India Compliance Monitoring	15
6.	Pan India Compliance	16
7.	Temperature Trend	17
8.	Appendix	18
8.1	Appendix A – Regional & Temperature Trend	18
8.2	Appendix B – Pan India Compliance	18
8.3	Appendix C – India Compliance Monitoring	18

1. Introduction

1.1 Purpose

This **Document** helps the user to understand the business discovery with the help of temperature values & it Covers HUL Dashboard Analytics Tool in Below Trend

- Regional Trend
- Pan India Compliance Trend
- India Compliance Monitoring
- Temperature Trend

1.2 Scope of this document

The Document provides an overview of HUL Dashboard Analytics with different trend Types. It Contains Flow Of HUL Dashboard & also It Contains How Internally This Analytics Tool is Working

1.3 Version & Technology Used

No	Technology	Version/Date
1.	Angular JS	1.4.7
2.	Bootstrap	3.3.2
3.	Google Chart	-
4.	Spring Framework	3.2.2

1.4 Definitions, Acronyms and Abbreviations

No	Abbreviations	Description
1.		
2.		

2. Project Introduction

2.1 Introduction

Atos is providing a web Solution that endures the business discovery for HUL through temperature values. Atos has installed sensors in HUL cold storage (one sensor per device) and these sensors are programmed to send temperature values to Atos cloud, at specified intervals.

This data is stored in the Internet of Things (IoT) database and is used by Analytics Tool – Angular JS & Java Spring We are Use d& Provide One Web Solutions To Analytics & generate reports.

3. Login

To login to the system, enter the following URL in the address bar and click **Enter**

<http://115.112.94.53:11023/huldashboard/#/login> **UAT**

<http://feetonstreet.atosicloud.com/huldashboard/#/login> **Production**

When any One Of Above URL Hits

Firstly it checks in web.xml to fetch the corresponding page

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xmlns="http://java.sun.com/xml/ns/javaee"
4   xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
5   version="2.5">
6   <display-name>HulDashboardApp</display-name>
7   <servlet>
8     <servlet-name>spring</servlet-name>
9     <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
10    <load-on-startup>1</load-on-startup>
11  </servlet>
12  <servlet-mapping>
13    <servlet-name>spring</servlet-name>
14    <url-pattern>/</url-pattern>
15  </servlet-mapping>
16  <filter>
17    <filter-name>charEncodingFilter</filter-name>
18    <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
19    <init-param>
20      <param-name>encoding</param-name>
21      <param-value>UTF-8</param-value>
22    </init-param>
23    <init-param>
24      <param-name>forceEncoding</param-name>
25      <param-value>true</param-value>
26    </init-param>
27  </filter>
28  <filter-mapping>
29    <filter-name>charEncodingFilter</filter-name>
30    <url-pattern>*</url-pattern>
31  </filter-mapping>
32 </web-app>
  
```

So here it finds that we have one more xml(spring-servlet.xml) which handles the page redirection. Here in spring-servlet.xml it finds :

- 1) `<context:component-scan base-package="net.hul.dashboard" />`
In all controllers it will search for method having `RequestMapping("/")`.
We have written this requestMapping in `BasicDashBoardController.java`

```
@RequestMapping(value="/", method = RequestMethod.GET)
public String loginPage(ModelMap map) {
    map.addAttribute("name", "HUL");
    return "main";
}
```

2) Once it gets which jsp page to be redirected, it finds path where the jsp can be found again from the spring-servlet.xml

```
<bean id="jspViewResolver"
    class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="viewClass"
        value="org.springframework.web.servlet.view.JstlView" />
        <property name="prefix" value="/jsp/" />
        <property name="suffix" value=".jsp" />
    </bean>
```

So it fetches the main.jsp page on the screen

This `main.jsp` is the basic(main)page which contains entries of all the css files .js files used in the project.

In this jsp we have “`<div ng-view></div>`”

This by default first redirects to “/login”

For this redirection we have made an entry in app.js file

```
var myApp = angular.module('hul', ['am.multiselect', 'ngRoute', 'Authentication',
    'ngCookies', 'myApp.directives']).config(['$routeProvider', '$httpProvider', function ($routeProvider,
    $httpProvider) {
        $routeProvider.
            when('/login', {templateUrl: tempContextPath+ '/jsp/login.jsp'}).
            when('/index/:trendType', {templateUrl: tempContextPath+ '/jsp/index.jsp', controller: "charts"}).
            otherwise({redirectTo: '/login'});
    }]);
```

This redirects to `login.jsp` and this page is displayed on browser.

Please sign in

Baasit Shukri

☐ Remember me

Sign in

LoginPage:

Both fields “Email Address” and “Password” are mandatory.

After clicking “Sign In”, control comes to loginController.js, here it sends the emailId and password to authenticationService.js which finally call webservice of backend

```
$http.post(tempContextPath+"/submitLogin",selectedValues).success(function
(response) {
```

```
    if(response.status == "success"){
```

```
    }else{
```

```
        response.message = 'Username or password is incorrect';
```

```
    }callback(response);});
```

This (“/submitLogin”) requestMapping is present in **BasicDashBoardController.java**

1) If this method returns failure, User has to fill in the form once again.

2) When emailId and password is matched it fetches the access_zone, access_factory, access_pv, access_depot, access_sv, access_rs_cold_room, access_rsv values and are set in session. Now success is returned to the frontend and here it redirects to index.jsp where-in according to user's region and cold-chain access, data is sent from backend and accordingly it is replicated in the google charts.

After The Successfully Login on to The System It redirects To Default Trend: **Regional Trend**

Note : We are Providing Analytics In For Trend As Mentioned Below.

- Regional Trend
- Pan India Compliance Trend
- India Compliance Monitoring
- Temperature Trend

4. Regional Trend



Here from frontend, in order to display the data on this page, it calls 2 backend web services:

1) roleBasedDropDownData:

This web-service fetches data to be shown in LHS dropdown

Regional Trend

Apply Filter

From date : 30-04-2016

To date : 02-05-2016

Region

Cold Chain

City

Branch Name

Vendor

Cold Room

Device ID

As Mentioned In Below Image function Code which is used to fill dropdown is Exist in interactive_google_charts.js

```
$scope.fetchDropdownValues = function () {
    $scope.openModal();
    $http.get(tempContextPath + "/data/roleBasedDropdown.json").success(function (data) {
        // $http.post(tempContextPath + "/roleBasedDropDownData").success(function (data) {
        // $scope.configParam.allUILevel = data.uLevelList;//countryName
        $scope.configParam.allRegion = data.regionsList;//countryName
        $scope.configParam.allColdChain = data.coldChainList;//countryName
        $scope.allRegion = data.regionsList;//countryName
        $scope.allColdChain = data.coldChainList;//countryName
        $scope.allDropdowndata = data.tablesJoinList;
        $scope.displayIndiaComplianceMonitoring();
        $scope.filterLocationData($scope.allDropdowndata, false, false, true, 0);
        $scope.getChartConfig();
    });
};
```

In Above code data.tablesJoinList Content All Data Based On That User (Currently Logged)

This Data passed in filterLocationData Function to set Scope For All Dropdown Based On Their Level

One thing TO Note we are Managing Level Over Here .Means Below Dropdown Only Open If Any Only if Above One Is Selected..

If You Want TO know about the flow you have 2 be clear About Level Management .There are 2 Types Of Level We Are Managing

- 1) Graph Level management
- 2) LHS dropdown Level Management

As per below image Level Define in \$scope.init() Function

```
$scope.init = function () {
    $scope.level = 1;
    $scope.initscope =
    [
        {
            "level": 0,
            "scopename": "allRegion",
            "selectedscope": "selectedRegion",
            "isDisabled": false
        },
        {
            "level": 1,
            "scopename": "allColdChain",
            "selectedscope": "selectedColdChain",
            "isDisabled": true
        }
    ],
```

Here \$scope.level=1 it indicates Graph Level & level inside \$scope.initscope is LHS dropdown Level which is managed by isDisabled parameter, it while change one after other when dropdown is selected

Graph Level:

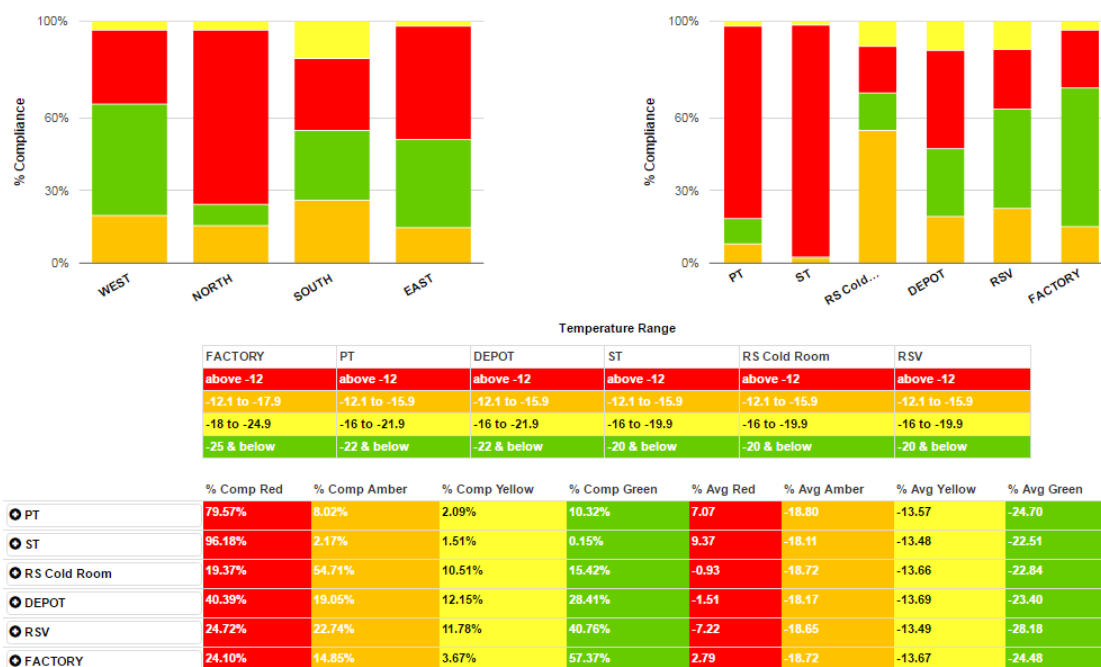
\$scope.level=1 Means Region & Cold chain Graph Will Be Visible

\$scope.level=2 Means Region & City Graph Is Visible

\$scope.level=3 Means Region & Device Graph is Visible

2)"/fetchDefaultValues":

This web-service fetches data to be shown in RHS for the two charts and data to be shown in table below.



In **webapp/data** folder just open finaloutput.json.. this JSON will come Back from Database when this service is called.

Note : based On webapp/data/temperature.json Which Contains All Temperature Range For All Types Of Trend (ex, Regnal India Complaints , etc..) This File Will be load on Init & based On Trend Type It Will Set the Temperature Range & graph is displayed

Other Config JSON file webapp/data/googlecharts.json Also Load In It Which Contain All Google Graph Related Configuration

APPLY Filter Functionality (For All Trend it will same):

When User select values from dropdown or Changes Dates & apply Filter that \$scope.applyFilter() Function Is called & in This function First it Which Check Which Level Currently it Is & set Filter data in \$scope.selectedFilter in \$scope.setSelectedFilter() function After that fetchChartValues Web Service Is called in that in \$scope.selectedFilter is passed & it will return Data as sample in webapp/data/finaloutput.json

NOTE : When First Time Page Loads at that Time

requestName= fetchDefaultValues & when Click On Apply Filter Button requestName= fetchChartValues

WebService: \$http.post(tempContextPath + "/" + requestName, \$scope.selectedFilter).success(function (data) {

```

$scope.setSelectedFilter = function () {
    $scope.selectedFilter = {
        "selectedUILevel": ($scope.configParam.selectedUILevel.length ? ""
        "selectedRegion": ($scope.configParam.selectedRegion.length ? "" +
        "selectedColdChain": ($scope.configParam.selectedColdChain.length ?
        "selectedCity": ($scope.configParam.selectedCity.length ? "" + $sc
        "selectedBranchName": ($scope.configParam.selectedBranchName.length
        "selectedVendor": ($scope.configParam.selectedVendor.length ? "" +
        "selectedColdRoom": ($scope.configParam.selectedColdRoom.length ? "
        "selectedDeviceID": ($scope.configParam.selectedDeviceID.length ? "
        "selectedFromDate": $scope.configParam.fromdate,
        "selectedToDate": $scope.configParam.todate,
        "type": $scope.trendType,
        "temperature": $scope.temperatureRange
    };
}

```

Graph Level Management for Regional trend:

1) **\$scope.level = 1** : you will see Region & cold Chain Graph

In More You will show city when you click on Cold Chain in table Filter as show in image & when You click On that city \$scope.drawSelectdDropdownChartByCity function will be called & Graph will be displayed Based on that city (Note : it will not change the Level but just displayed City Graph)

In Process **fetchDrilledValues** Web Service is called

\$http.post(tempContextPath + "/fetchDrilledValues", \$scope.selectedFilter).success(function (data) {

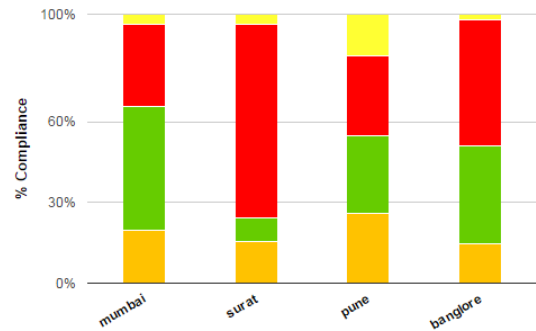
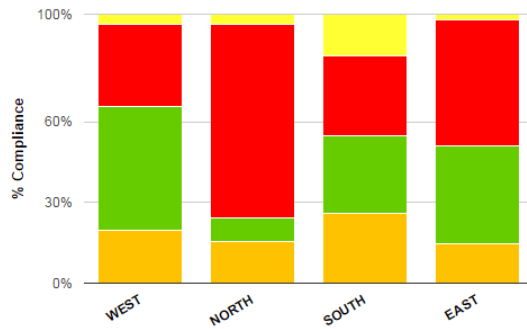
& will be return city data **data/city-deviceid-map.json**

	% Comp Red	% Comp Amber
<div>PT</div> <div>Search City</div> <div>Raigad</div> <div>Nashik</div>	79.57%	8.02%
ST	96.18%	2.17%
RS Cold Room	19.37%	54.71%
DEPOT	40.39%	19.05%

2) \$scope.level = 2 : you will see Region & City Based Graph

Note Whenever You click on Any Cold Chain in First Level : Level is changed to 2 & you are able to see below graph of city

PT ✖



In Process **fetchDrilledValues** Web Service is called

```
$http.post(tempContextPath + "/fetchDrilledValues", $scope.selectedFilter).success(function
(data) {
```

& will be return city data **data/city-diviceid-map.json**

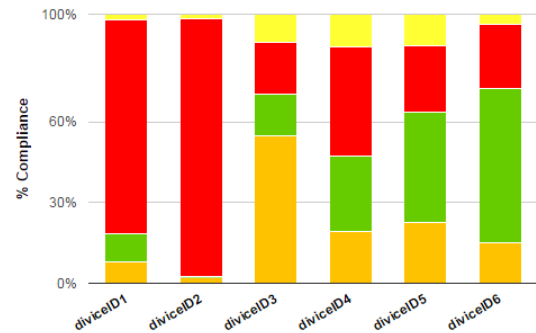
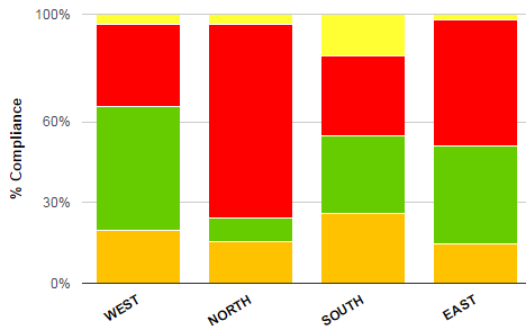
In More You will show Device ID when you click on City in table Filter as show in image & when You click On that city `$scope.drawSelectdDropDownChartByDevice` function will be called & Graph will be displayed Based on that Device (Note : it will not change the Level but just displayed Device Graph)

	% Comp Red	% Comp Amber
<div><div><div><div><div><div></div><div>mumbai</div></div></div><div><div><div><div></div><div>Search Devices</div></div><div><div>deviceID1</div><div>deviceID2</div></div></div></div></div></div></div>	30.70%	19.86%
<div><div><div><div></div><div>surat</div></div></div></div>	72.15%	15.35%
<div><div><div><div></div><div>pune</div></div></div></div>	29.65%	26.12%
<div><div><div><div></div><div>banglore</div></div></div></div>	46.75%	14.71%

3) \$scope.level = 3 : you will see Region & Device Based Graph

Note: Whenever You click on Any City in Second Level : Level is changed to 3 & you are able to see below graph of Device

PT ✕ mumbai ✕



In Process fetchDrilledValues Web Service is called
`$http.post(tempContextPath + "/fetchDrilledValues", $scope.selectedFilter).success(function (data) {`

& will be return city data [data/city-diviceid-map.json](#)

In More You will show all Device ID

Search Devices	% Comp Red
deviceID1	79.57%
deviceID2	96.18%
deviceID3	19.37%
deviceID4	40.39%
deviceID5	24.72%
deviceID6	24.10%

5. India Compliance Monitoring

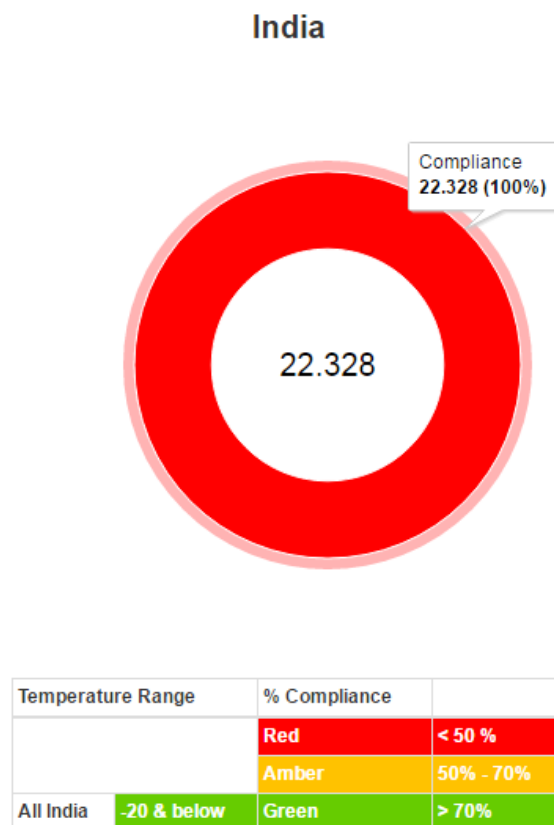
If User Has access Of all **4 Regions** Then This Trend Is Visible At right end side of the page...

Filter Will work same as Mention In Regional Trend

```
$http.post(tempContextPath + "/" + requestName,
$scope.selectedFilter).success(function (data) { web service will be called
```

```
$scope.indiaComplianceMonitoringData = data;// Hold The Backend response data
```

& letter \$scope.indiaComplianceMonitoring(); Function is called to Display Graph As Mention In Below Image



6. Pan India Compliance

Filter Will work same as Mention In Regional Trend

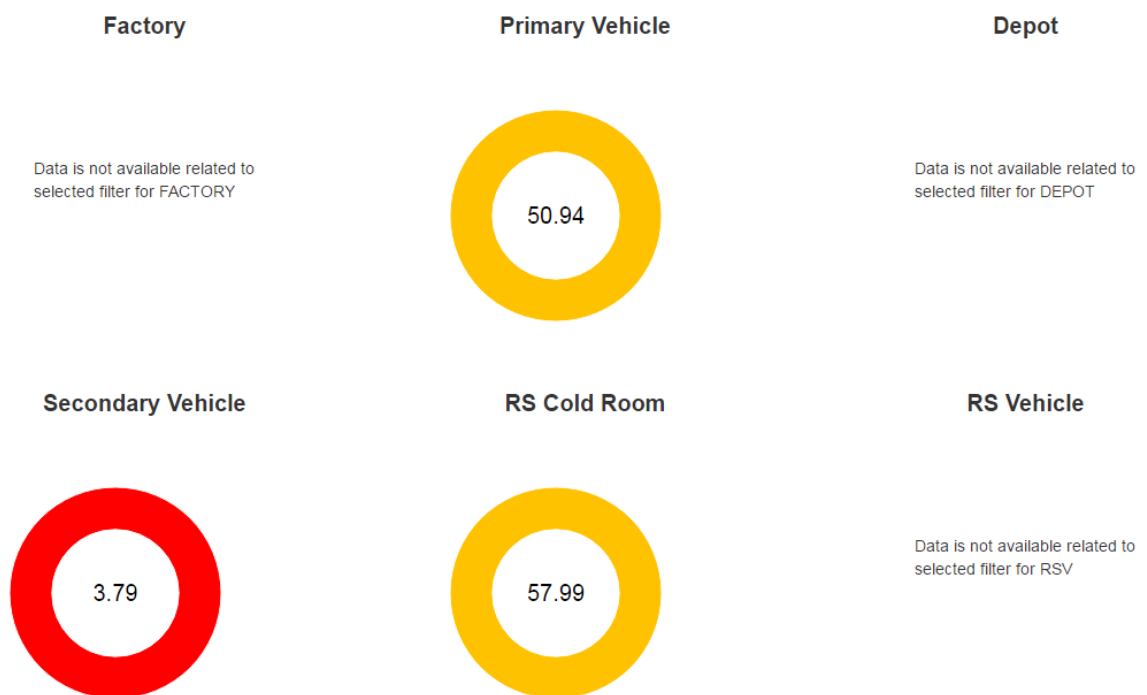
\$http.post(tempContextPath + "/" + requestName,
\$scope.selectedFilter).success(function (data) { web service will be called

\$scope.panComplianceMonitoringData = data.compliance;// Hold The Backend response
 data :

Sample response

\$scope.panComplianceMonitoringData = {"FACTORY": 12,"RS Cold Room": 57,"DEPOT":
29,"PT": 49.58,"ST": 23,"RSV": 89,"FACTORYANT": 5,"DEPOTANT": 72}

& letter \$scope. panIndiaComplianceMonitoring (); Function is called to Display Graph As
 Mention In Below Image



7. Temperature Trend

Filter Will work same as Mention In Regional Trend

`$http.post(tempContextPath + "/" + requestName, $scope.selectedFilter).success(function (data) { web service will be called`

`$scope.temperatureData = data.temperature;;// Hold The Backend response data`

& letter `$scope.temperatureTrend();` Function is called to Display Graph As Mention In Below Image



8. Appendix

8.1 : Appendix A – Regional & Temperature Trend.

Below is the Temperature Range & colour For Regional & Temperature Trend.

Temperature Range & Colour Wise data You will get in data/temperature.json File

That we are passing from Front-end to web service Based On that data Query is Executed & data Will be return.

This colour array we are ['Amber', 'Green', 'Red', 'Yellow'] displaying in graph based on condition in temperature json .you can clearly show this in below image

Temperature Range					
FACTORY	PT	DEPOT	ST	RS Cold Room	RSV
above -12	above -12	above -12	above -12	above -12	above -12
-12.1 to -17.9	-12.1 to -15.9	-12.1 to -15.9	-12.1 to -15.9	-12.1 to -15.9	-12.1 to -15.9
-18 to -24.9	-16 to -21.9	-16 to -21.9	-16 to -19.9	-16 to -19.9	-16 to -19.9
-25 & below	-22 & below	-22 & below	-20 & below	-20 & below	-20 & below

8.2 : Appendix B – Pan India Compliance

For Pan India Graph Color Will be Decide as Displayed In below Image

Temperature Range		% Compliance PV,Depot,SV,RSC,RSV		% Compliance Factory	
		Red	< 50 %	Red	< 70 %
RSC,RSV	-18 & below	Amber	50% - 70%	Amber	70% - 94%
Factory,PV,Depot,SV	-20 & below	Green	> 70%	Green	> 94 %

8.3 : Appendix C – India Compliance Monitoring

For Pan India Graph Color Will be Decide as Displayed In below Image

Temperature Range		% Compliance	
		Red	< 50 %
		Amber	50% - 70%
All India	-20 & below	Green	> 70%