

Call Workboard

Nag Varun kondreddy
RA1811003010692

Computer Science and Engineering
SRM Institute of Science and Technology,
Kattankulathur

Gunda Sumanth
RA1811003010701

Computer Science and Engineering
SRM Institute of Science and Technology,
Kattankulathur

Abstract:- In this paper, we present work on Creating a call workboard web application where the customer name ,ID and due payments are displayed. So that the collection analyst can pick a customer or search the customer details and call the customer or send the correspondence. the

excel sheet was very crucial for analysts but there are problems such as if the sheet consists of thousands of records it would be difficult to search each customer and analyze it our research aims in making the work easier with efficiency and productivity by saving previous time.

1	Name	Title	Department	Office Address	Start Date	Email	Phone	Birthdate	Image
2	Chenabai Ganer	Research Nurse	Services	Bellevue, WA, 98004	5/18/2017	aganer@iames.com	362-403-2802	11/18/1984	http://www.callworkboard.com/employees/21.jpg
3	Yacine Taf	Health Coach IV	Research and Devlop	Bellevue, WA, 98004	8/7/2018	mtaf@iats.com	779-430-4217	10/2/1995	http://www.callworkboard.com/employees/22.jpg
4	Adriane Lindeman	Recruiter	Legal	Bellevue, WA, 98004	12/29/2017	aylindem@iames.com	807-402-4424	7/6/1978	http://www.callworkboard.com/employees/23.jpg
5	Tanner Canzo	CIS Technical Architect	Business Development	Bellevue, WA, 98004	7/5/2018	tcanzo@iames.com	201-402-7082	1/30/1987	http://www.callworkboard.com/employees/24.jpg
6	Ulrika Quillen	Developer I	Legal	Bellevue, WA, 98004	1/14/2017	uquillen@iames.com	853-214-0775	7/23/1987	http://www.callworkboard.com/employees/25.jpg
7	Seyner Longstone	Administrative Officer	Business Development	Bellevue, WA, 98004	12/1/2018	slongston@iames.com	434-572-2011	2/9/1979	http://www.callworkboard.com/employees/26.jpg
8	Orsola Winkler	Analyst Circuit Design manager	Human Resources	Bellevue, WA, 98004	8/7/2017	owinkler@iames.com	534-329-4015	8/27/1960	http://www.callworkboard.com/employees/27.jpg
9	Wasa Colby	Software Consultant	Human Resources	Bellevue, WA, 98004	8/25/2017	wcolby@iames.com	391-429-4054	6/15/1970	http://www.callworkboard.com/employees/28.jpg
10	Larry Lyndell	Junior Executive	Engineering	Bellevue, WA, 98004	1/4/2017	lyndell@iames.com	857-812-8288	8/27/1967	http://www.callworkboard.com/employees/29.jpg
11	Barbel Lantz	VP Sales	Marketing	Bellevue, WA, 98004	5/29/2017	blantz@iames.com	425-546-4759	8/3/1962	http://www.callworkboard.com/employees/30.jpg
12	Albert Kulu	Geologist I	Sales	Bellevue, WA, 98004	3/13/2018	akulu@iames.com	324-195-3079	11/17/1978	http://www.callworkboard.com/employees/31.jpg
13	Delaney Shremes	Biostatistician II	Legal	Bellevue, WA, 98004	7/17/2018	dshremes@iames.com	729-696-5448	12/18/1981	http://www.callworkboard.com/employees/32.jpg
14	David Winkler	Research Associate II	Business Development	Bellevue, WA, 98004	3/15/2019	dwinkler@iames.com	363-743-5846	8/28/1987	http://www.callworkboard.com/employees/33.jpg
15	Colleen Fisher	Systems Administrator III	Services	Bellevue, WA, 98004	3/17/2018	cfisher@iames.com	202-245-3205	10/21/1983	http://www.callworkboard.com/employees/34.jpg
16	Samuel Escobedo	Executive Director	Legal	Seattle, WA, 98103	8/15/2017	sescobedo@iames.com	206-413-8075	3/3/1965	http://www.callworkboard.com/employees/35.jpg
17	Wami Sankhla	Senior Financial Analyst	Marketing	Seattle, WA, 98103	2/22/2017	wsankhla@iames.com	866-488-1728	1/29/1989	http://www.callworkboard.com/employees/36.jpg
18	Ahmed Elshahhat	Programmer	Legal	Seattle, WA, 98103	12/11/2016	aelshahhat@iames.com	866-294-8424	1/22/1987	http://www.callworkboard.com/employees/37.jpg
19	Ingenier Elshahhat	Developer II	Research and Devlop	Seattle, WA, 98103	4/3/2018	ielshahhat@iames.com	534-819-6108	12/17/1978	http://www.callworkboard.com/employees/38.jpg
20	Hafsa Gul	Social Worker	Accounting	Seattle, WA, 98103	8/27/2017	hgul@iames.com	185-807-6425	7/16/1976	http://www.callworkboard.com/employees/39.jpg
21	Shobha Laxmanan	Media Manager IV	Services	Seattle, WA, 98103	4/13/2019	slaxmanan@iames.com	471-825-8811	10/14/1984	http://www.callworkboard.com/employees/40.jpg
22	Johna Balaban	Char. Design Engineer	Product Management	Seattle, WA, 98103	12/25/2017	jbaldaban@iames.com	443-586-4137	3/15/1978	http://www.callworkboard.com/employees/41.jpg
23	Gary Palmer	Professor	Support	Seattle, WA, 98103	1/8/2017	gpalmer@iames.com	144-871-8511	10/25/1970	http://www.callworkboard.com/employees/42.jpg
24	Sophia Prastar	Account Representative IV	Business Development	Seattle, WA, 98103	10/10/2016	sprastar@iames.com	857-743-3403	5/19/1984	http://www.callworkboard.com/employees/43.jpg
25	Orchard Shrestha	CIS Technical Architect	Sales	Seattle, WA, 98103	8/17/2018	oshrestha@iames.com	251-547-4280	5/25/1977	http://www.callworkboard.com/employees/44.jpg
26	Cindy Warner	Information Systems Manager	Accounting	Seattle, WA, 98103	10/25/2017	cwarner@iames.com	733-405-2303	8/11/1987	http://www.callworkboard.com/employees/45.jpg
27	Leonid Simenok	Geologist IV	Support	Seattle, WA, 98103	4/11/2019	lsimenok@iames.com	428-564-7488	6/28/1983	http://www.callworkboard.com/employees/46.jpg
28	Harold Christiansen	Software Test Engineer II	Engineering	Seattle, WA, 98103	7/13/2016	hchristiansen@iames.com	724-847-4312	6/19/1957	http://www.callworkboard.com/employees/47.jpg

Fig.1. Existing Method

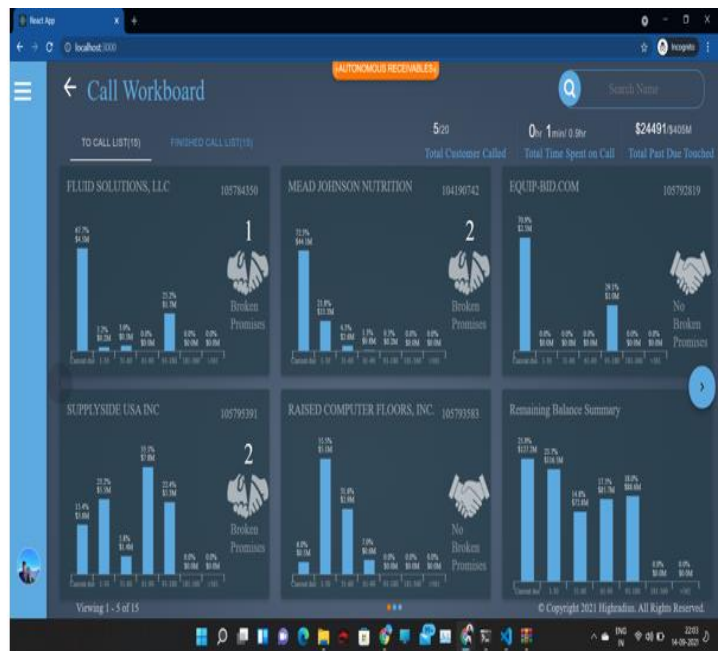


Fig.2. New Method

I. INTRODUCTION

It is very important for the data to be read and analyzed carefully so that the analysts can contact the customers. In the existing method the analyst uses to work with excel sheets where the data was not highly readable. In this project the visualization of data makes it easy for the analyst to understand the data and work efficiently and fast. Analyst can search the customer details using customer name and if it is present in the database it will displayed with customer id and the histogram of due payments.

The collector will be able to see the worklist for the highlights on the Call Workboard Screen. Basically, the tab

consists of all the cards which are known as customer cards, Balance of the remaining summary card, Navigation icon and the copyright logo of the company.

The estimates of the total customers to be called should be calculated so that there will be a track of total number of customers and the total number of customers to be called based on the data where the customers have due to be paid. Also, the total time which they spent on calls with customers is to be noted. The past due amounts should be summed for all the customers where they are called and then should be checked accordingly with the due amounts of all the customers to be called.

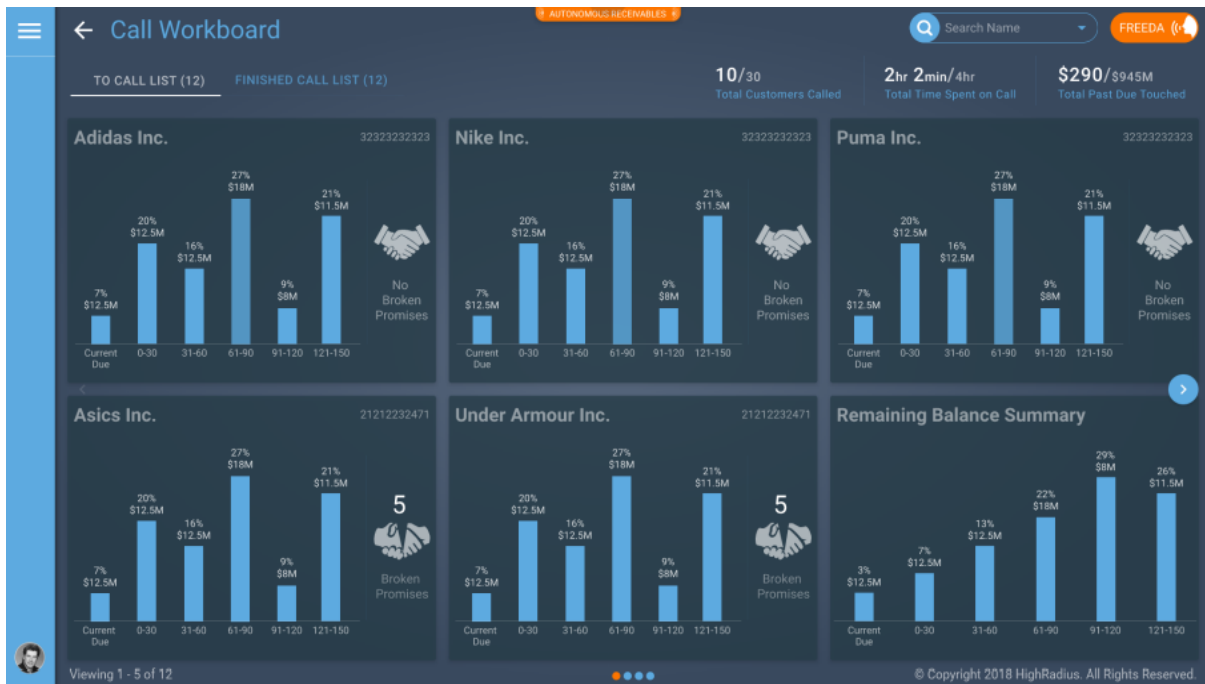


Fig.3. Dashboard

II. PROPOSED WORK

The screen will have the first tab as TO CALL LIST followed by the count of customers to be called. E.g. TO CALL LIST (10), if there are 10 customers to be called.

Formulas for all these details are as follows:
 Total Customers Called = Number of customers called / Total number of customers to be called
 E.g., 10 / 30 Total Customers Called

Total time spent on Calls = Time spent on the calls completed / Estimated Duration of Total Calls
 E.g., 2hr 2min / 4hr Total Time Spent on Call
 Total Past Due Touched = Past Due Amount total of the

customers called / Total Past Due Amount of all the customers to be called
 E.g., \$290M / \$945M Total Past Due Touched

Each customer card will have the following information which are configured for the account like

- Customer Name
- Customer Number
- Aging Buckets
- ✓ Each bucket should have the aggregated amount of invoices in the bucket at the top
- ✓ Each bucket should have the aggregated amount of invoices in the bucket at the top
- Broken P2P count

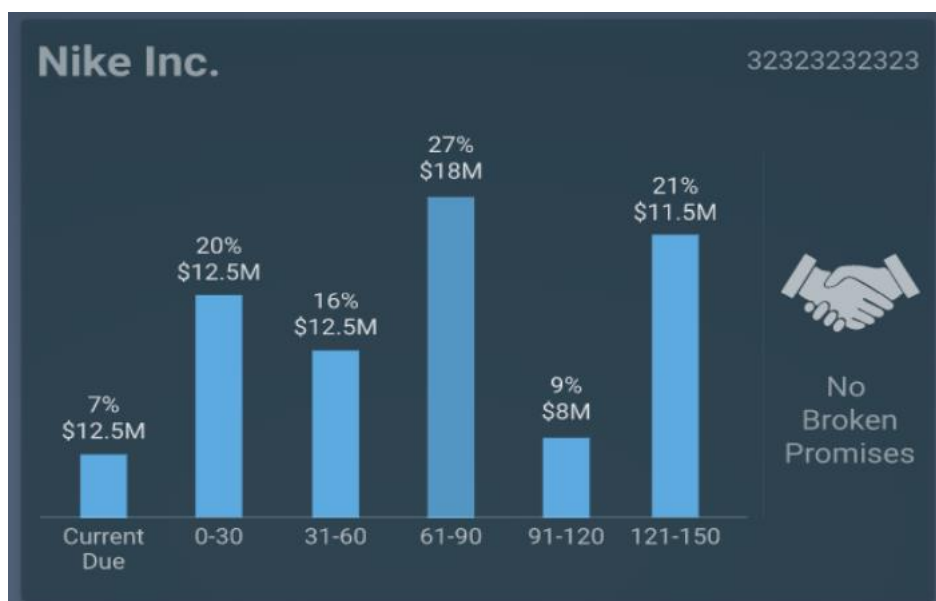


Fig.4. Customer Card

Remaining Balance summary card is the last card on the screen gives a summary of the next set of customers where

the card has aging buckets configured for the account with amounts in buckets of the next set of customers.

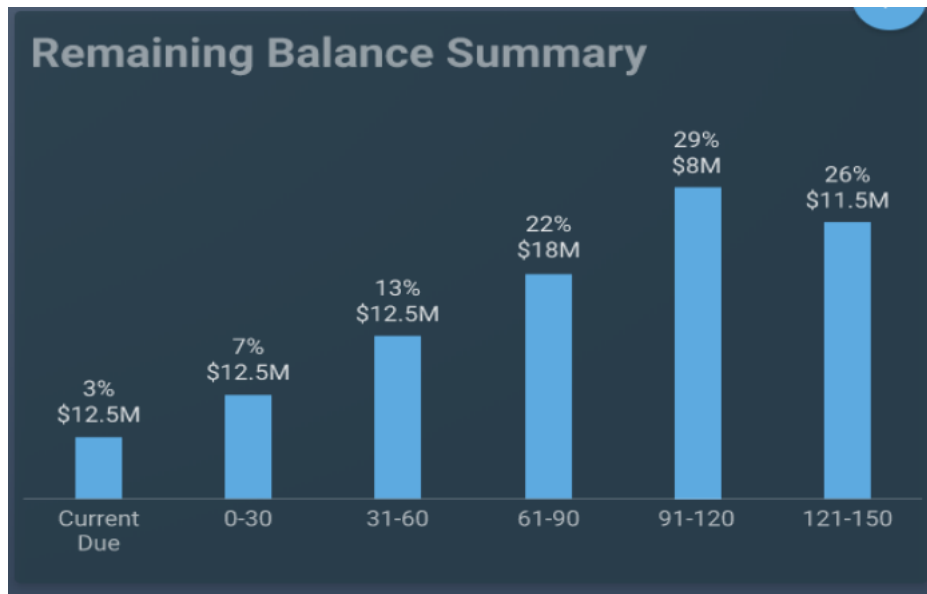


Fig.5. Remaining Balance Summary Card

The navigation icon will be displayed beside the customer cards.

- 1)The left icon will be disabled on the first screen (as no customers are present before that).
- 2). The right icon will be disabled on the last screen (as no customers are present after that).
- 3)The icon highlights in blue if the collector hovers over it.
- 4)When clicked, the icon should bring up the next set of customers on the list.

Viewing x-y of z footer count:

- 1)The footer should show the count of the set of customer

cards being displayed on the screen.

2)If the collector views the next set of customers, the counts should get updated accordingly.

E.g., Viewing 1 - 5 of 19

If no customers are available for calling the message displayed will be:

“No Customers available for calling

The collector can reach the Search Results screen by entering characters (minimum 3) on the search bar in the Call Workboard.



Fig.6. Search Results Screen

- Header: SEARCH RESULTS
- As soon as the user enters the first three characters of the customer’s name or customer number, the list of matching customers starts popping up.
- A dynamic tab should come up as soon as a string is entered.
- The tab should have the count of customers matching the string entered in the search bar, beside the header.
- The tab should have a cross icon beside the header. Clicking on the cross icon should take the user to the To

Call List Tab.

- The search tab will also have a cross icon. Clicking on the cross icon should take the user to the To Call List Tab.
- By clicking on the search button (magnifying glass icon) or any place in the text box, the text box gets enabled and the user can start entering the details to be searched.
- The customer names in the list must highlight the text which was typed in the search box (as can be seen in the screenshot above).



Fig.7. Customer Names in search results

- The user is allowed to type in more characters as the search results are displayed, i.e., it will be a dynamic search.
- The navigation arrows should appear in case more than 6 matching results are found
- The customer cards will be the same as in To Call List.
- If no customers are available for calling the message

displayed will be:

“No Matching Results Found”

The collector can open the Freeda screen by clicking on the Freeda button in Call Work



Fig.8. Freeda Assistance

- The Freeda window will be persistent on the screen.
- The screen should be responsive for all resolution and readjust as required.
- The header for Freeda window will be: FREEDA
- The cross icon beside the header should close the Freeda window and make the screen return to the default state.

While data loads on the screen, Skeleton Structure will appear on the screen for:

- 1) To Call List
- 2) Search Results

III. IMPLEMENTATION

This project contains two parts frontend and backend. The frontend is created using React js and Material UI.

The UI has four components they are;

- Header
- Drawer
- Card Grid
- Footer

In backend three API calls should be handled

- getGridData
- getSearchResults
- getSummary

These API's will fetch data accordingly and gives accurate results.

A. Front end:

In the front end after installing the packages there are default package-lock.json and package.json files.

In the SRC folder in frontend a new folder API is created for all API calls

getSearchResult.js is used for searching from the input provided by the user and the code is as follows:

```
import axios from "axios";

const getSearchResults = async (input) => {
  const response = await axios({
    method: "post",
    url: `http://localhost:4000/esCustomerSearch.do?`,
    data: {
      customerName: input,
    },
  });
};

return response.data;
};

export default getSearchResults;
```

For getting the summaries of upcoming API calls the API get getUpcomingSummaryAPI.js is created as follows:

```
import axios from "axios";

const API_URL = "http://localhost:4000/getUpcomingSummary.do";

const getUpcomingSummary = async () => {
  const response = await axios.post(API_URL);
  return response.data;
};

export default getUpcomingSummary;
```

For getting the user call workbook data for each individual user in-order to check the whether the call is made is or not regarding the past dues.

```
import axios from "axios";

const getUserCallWorkbookData = async (pageNum) => {
  const API_URL = "http://localhost:4000/getUserCallWorkbook.do?pageNumber=${pageNum}&pageSize=5";
  const response = await axios.post(API_URL);
  // console.log(response.data);
  return response.data;
};

export default getUserCallWorkbookData;
```

Components included are ;

Call workboard:

In the call workboard the following js files are used Callworkboard.js, FooterSection.js, HeaderSection.js, TabSummary.js

Drawer Section:

The drawer section has two files where FreedaDrawer.js is included

Homepage:

The homepage includes the cards numbered accordingly and the overall chart for this js files are used such as cards with respect to their numbers and chart.

Using CSS, the styling is done for dashboard to look user friendly and freedadrawer is also used.

In the APP.js the functions drawer section and call work board is included

B. Back end:

In the Back end after installing the packages there are default package-lock.json and package.json files. In the backend api folder is created where the routes are connected to API's each route folder has two files of js and json files types.

In Customer search file the index.js file filters the customer data individually and the data.length, customer, and bucket names are sent

In the index.json file total number of customers is created where each customer has following details:

Bucketnames, company code, total current open amount, past due bucket document amount, total broken promises, customer number, past due, customer name, processed id, pkcustomer map id.

In fetch customer file the index.js file fetches the customer data if customerData.id==customerid

In the index.json file there are following details such as id,customer name,customer number,company code,risk class,payment behaviour,currency and reasons which include broken promises as well as no broken promises.

The account details consist of actual credit limit, credit utilized, total actual open amount, total current open amount, total current invoice count, total past due invoice count, total past due amount, total past due current, total disputed amount, past due bucket document count, past due bucket document amount and bucket names consist of the ranges 1-30, 31-60, 61-90, 91-180, 181-360, credit limit utilization percent, currency type and document currency type.

In finished calls page the index.js file has a counter which counts the call where as the index.json file has an overview key called finished calls page where the first value is finished calls overview which consist of total calls made, total time spent, total promises created, voice mail count, total amount received also there is another key called finished calls which has the following information customer id, customer number,customer name, communication time, call duration, call outcome summary, called number, and is worklist true or false, and finally call state which mentions incoming, rejected, outgoing, misses based on the circumstances.

In the get upcoming summary the upcoming past due amount, upcoming open amount, upcoming calling minutes, upcoming customer count, upcoming past due bucket document count, upcoming past due bucket document amount are created.

Get user call workbook has an overview which consist of id, total open amount, total past due amount, total past due processed, expected calling minutes, completed calling minutes, processed customer count, total customer count, total promises count, total promises amount, pending past due bucket document count and pending past due bucket document amount.

Another key included is workbook items which has id, customer name, customer number, currency, total broken promises, past due bucket document count, past due bucket document amount, bucket names, which includes categories 1-30, 31-60 as follows, total current open amount.

The get user overview contains the keys in the dictionary as id, total open amount, total past due amount, total past due processed, expected calling minutes, completed calling minutes, processed customer count, total customer count, total p2p count, total p2p amount, pending past due bucket document count, pending past due bucket document amount.

For the profile details of a user id, account name, account id, email id, first name, last name, phone and the virtual number has the following details virtual number, full name, sip user, sip token, incoming number also the worklist has different ids in accordance with customer names which are included in a list.

The save or update contact has customer id and contacts where each contact has id, isprimary, type, first name, last name, city, state, country, data type id, email, mobile number, phone number, title, fax number, address, extension, zip, role, and call logs in each call log there are following details id, customerid, calling number, extension, start time, end time, duration, communication time, status, call summary, total open amount, total past due amount, note has id, customer id, note items, subject, created time, created user.

The contact key has id, is primary, type, first name, last name, city, state, country, data type id, email and mobile number.

Communication reason, voice template, unique id and call type are also mentioned in call logs. Data.json file has defined dictionaries and all the data is stored in index.json file in save or update contact folder.

IV. TEST CASES

A. Negative Test Cases

If no customers are available on the To Call List Tab, the following message should come up on the screen:

“No Customers available for calling”.

After entering the first three characters in the text box, if the system does not find a match for a corresponding customer name or customer number, the system should prompt with the:

“No Matching Results Found”.

B. Acceptance Test Cases

- To Call List must have the count of customers to be called.
- The number of customer cards on the screen should match the count on the tab header.
- On clicking on the navigation arrows, the set of customers should come up.
- The count at the bottom of the screen should match the count of cards.
- Under the following test cases the system should accept the input in the search bar:
 - ✓ Customer Name(Alpha)
 - ✓ Customer Number(Numeric)
 - ✓ Account Number(Alphanumeric)
- After typing in the first 3 letters, the list should come up.

- Clicking on the cross icon should remove all the results.
- UI should be performant even if a large list of customers is rendered on the screen.

V. RESULTS DISCUSSION

In this article we have presented an analysis of learning dashboards that have been deployed with react and the dashboard helps the analysts to pick a customer and call the customers in correspondence with due payments at ease.



Fig.9.0. Call workboard



Fig.9.1. Freeda Assistance for opening customer cards



Fig.9.2. Call list created based on Analysis

VI. CONCLUSION

The problem statement which we took is now-a-days companies are having lot of issues while maintaining the client’s data. The existing method followed is that they are using excel sheets which causes many issues for handling and analyzing data. So, the solution which we found is to make a dashboard which we named as call workboard. In this dashboard we are maintaining all the client’s details in the form of the cards so it becomes easier to handle and analyze data with multiple features included so that the analysts can do the work easier by checking the cards and contacting them for clearing their dues. In future more freeza assistance can be upgraded for contacting the clients automatically and making it more friendly by saving time.

REFERENCES

- [1]. Learning dashboards an overview an overview and future research opportunities. <https://core.ac.uk/download/pdf/34593851.pdf>
- [2]. Data visualization in react application <https://codeburst.io/data-visualization-in-react-application-f033858c71f3>
- [3]. Data visualization with React <https://reactresources.com/topics/data-visualization>
- [4]. creating an analytical dashboard using react <https://www.fusioncharts.com/blog/analytical-dashboard-react-graphql/>