Doctor's Office Application (We Care)

Harsh Patel (Author)
Dept. of Computer Engineering

San Jose State University
San Jose, USA
harshpankajkumar.patel@sjsu.edu

Pranali Bhavsar
Dept. of Computer Engineering

San Jose State University
San Jose, USA
pranalijigneshbhai.bhavsar@sjsu.edu

Aakash Thakkar
Dept. of Computer Engineering

San Jose State University
San Jose, USA
aakashmanishbhai.thakkar@sjsu.edu

Yesha Dave
Dept. of Computer Engineering

San Jose State University
San Jose, USA
yesha.dave@sjsu.edu

***Abstract* — We Care is an doctor office management software with feature of maintaining patient record for doctor, generating patient demographics and voice enabled form filling system. Our application help to reduce time in searching for patient data and easy form fill up by patient.** ***Index Terms*— Conversational EHR, Natural Language Processing, Node.js, React.js, MongoDB, Kafka, AWS**

I. INTRODUCTION

People go to see doctor to mitigate their pain and getting healed. When they visit doctor before getting their treatment started they have to fill out bunch of form, on an average 10-11 forms. This is really sore full for patient. Many times patient gets annoyed. Moreover many patients are not even that condition to fill out such long forms. Suppose, patient is right handed and his right hand got fractured. It gets really hard for that patient to fill up form. They need to ask for help from other and they don't feel to share their personal information with other. Major part of patient are old age people, USA has some many immigrant came over here. Older generation of these immigrant are not comfortable with English and find out hard to fill out forms in english. Many of them do not even know how to write or read English. All this things really creates problem for patient.

On the other side of story, doctors have to maintain these forms filled by patient for at least 10 years according to HIPAA compliance. These data should be kept secured as per US government rules. On an average a doctor in US gets about 20 patients a day. So doctor gets 4800 patient per year approximately. It gets really hard to manage data of such large numbers of patients for 10 years and that too secured. Also, when patient visits next time, doctor needs to retrieve data from such large amount of content making their job of treating patient delayed. There are many big companies providing EHR, but they are really expensive for low income doctors. They prefer to keep this hassle of work than affording those software.

So, we have come up with conversional EHR with low affording price, reducing pain of both doctors and patients.

II. SYSTEM ANALYSIS

A. Problem Statement

When patient visits a doctor they have to fill out bunch of forms and on other side doctors have to maintain those records securely. Both of these tasks are irksome to do it manually. Our application helps to reduce these problems faced and makes process easy.

B. Proposed Study

This application, help patient to fill out form easily. Patient will not have to struggle filling up forms. System will ask patient for question to fill up forms details. Answers said by patients are filled automatically so they don't have to struggle writing down paper. It will also reduce paper work maintenance for office staff. Doctor office staff will not have to go through hassle of maintaining papers. They can retrieve patient data with one click and seconds of response. Doctor can know demographics of his clinic. He could easily have any idea, how he's treatment is going on. It can know about patient's treatment record by looking at graph.

III. FRONTEND & DESIGN

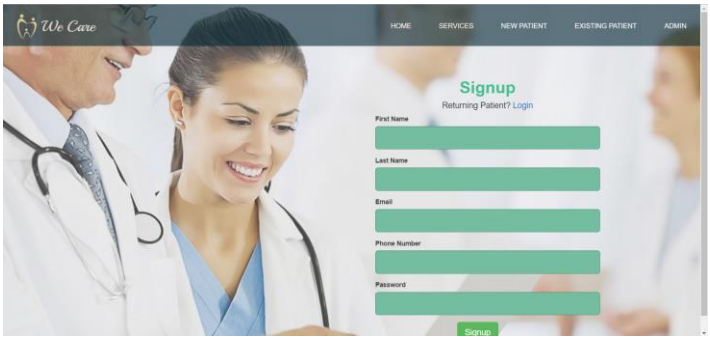
We implemented our application's frontend in React.js. This is the Homepage of our application.

1. Homepage

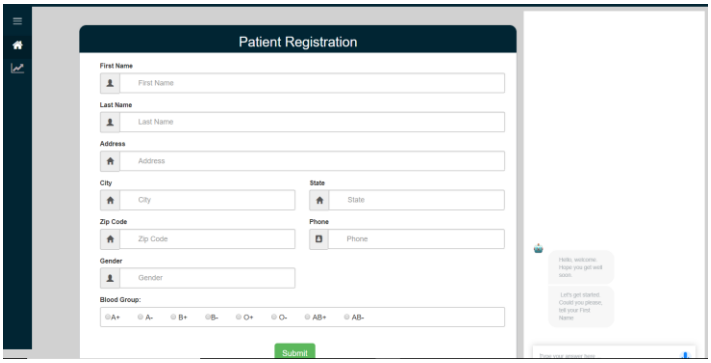
It has the following tabs:

1. Services: It gives information about all features of this application in detail and how to start using the application.
2. New Patient: In this page, patient can create new account.
3. Existing Patient: In this page, existing patient can login. and view, update and fill up the required forms.
4. Admin: Doctor and doctor's staff can login in this page. and

access the admin side feature.

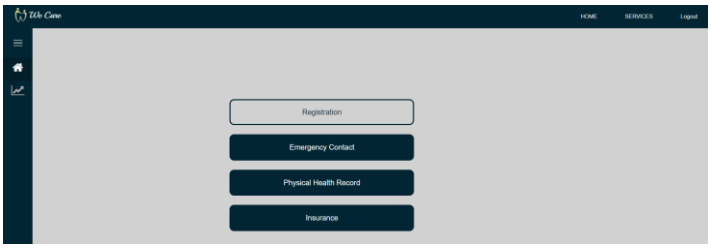


2. Patient Signup

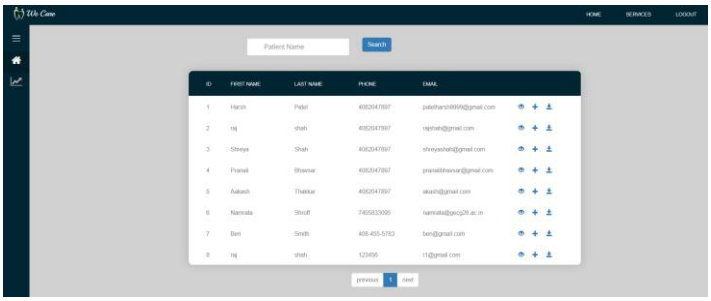


3. Registration Form(Form-1)

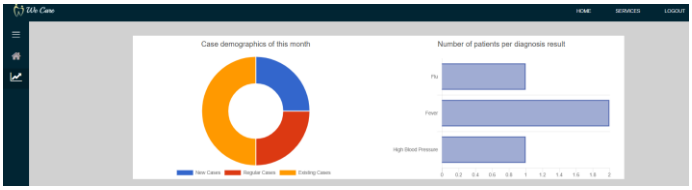
When patient creates account, they will be redirected to the registration form page, where they can fill up the form using chatbot or normal form fields. In dashboard of patient, patient can view the forms.



4. Patient Dashboard



5. Doctor Dashboard



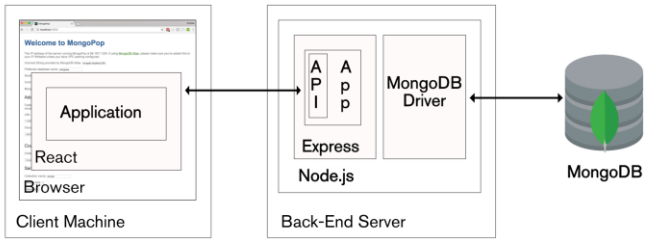
6. Analysis

When doctor logins to his account, he will be redirected to the dashboard. In dashboard, doctor can search for the patient and view, edit and download patient's form. Also he can go to analysis page. In analysis page, doctor can see some graphs like how many patients were diagnosed of which health problems.

7. Downloaded PDF of Patient's Data

IV. BACKEND AND DEPLOY

For developing backend, we used Node.js for creating apis and we used MongoDB as a database. We encrypted the data of patient using MD5 algorithm. Also, we used Kafka as a message broker. We deployed our web application on the AWS. We created five collections in MongoDB. One is for patients, one is for doctors and other three collections are used for three forms. We used patientid od patient collection as a foreign key in three collections which has forms data.



A. Understanding the Objective

The first step in developing a project is to understand the objective which involves an understanding of the intent and essentials of a system. This comprehension is used as a problem description and a preparatory system to accomplish the expectations. The objective of our project is to reduce workload of doctors so they can give more time towards his patients.

B. Creating Data

Once the understanding of the objective is over, the next step is to create the data. We use mongoDB as our Database. Most of all organizations are adopting MongoDB because it enables them to build applications faster, handle highly diverse data types, and manage applications more efficiently at scale. With the help of it. We can remove complex object- relational mapping (ORM) layer that translates objects in code to relational tables. It also has scalability, it can be scaled within a across multiple distributed data centers, providing new levels of availability and scalability previously unachievable with relational databases like MySQL. As your deployments grow in terms of data volume and

throughput, MongoDB scales easily with no downtime, and without changing your application.

V. CONCLUSION

Thus, Our Web application will help patients who faces problem in filling out number of forms while visiting the doctor by using Mozilla's speech to text API in which the patient will describe himself and the problems he is facing. This will reduce the load that patient faces while visiting the hospital. On the other hand our application will help the doctor to save the record of all the patients who are visiting the hospital so that he can track and maintain the record of the patient if the patient re visits the hospital. Apart from this, our application will also provide all the details in a graph format where the doctor can see how many new patients visited the hospital and what kind of symptoms the patients face. Our application is much cheaper than other system already in the market as we use all the open source technology. Thus, the hospitals with small capital can use our application and can reduce the paperwork that they face now a days.

VI. REFERENCES

[1] Statistics Data: https://www.statista.com/statistics/613959/us-physicans-patients-seen-per-day/

http://www.governing.com/topics/health-human-services/gov-doctors-electronic-health-records.html

[2] React JS | https://reactjs.org/

[3] Node JS | https://nodejs.org/en/

[4] Web Speech API: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API