CleverCare

(Reducing Hospital Readmissions)

Darshit Thesiya, Siddharth Daftari, Tanmay Bhatt, Vikas Miyani
Software Engineering
San Jose State University
San Jose, US
darshit.thesiya@sjsu.edu, siddharthrajesh.daftari@sjsu.edu, tanmay.bhatt@sjsu.edu, vikas.miyani@sjsu.edu
Advisor: Prof. Rakesh Ranjan

***Abstract*— One of the major contributors to healthcare costs is hospital readmissions. Research shows that 15-20% of people who are discharged from hospital are readmitted within 30 days or less. Many of these readmissions are preventable. There are some intervention techniques that can help address this issue. One of them is to use machine learning to predict the chances of readmission of a patient. The project facilitates out-patient care to take feedbacks at regular intervals from patient once he/she is discharged and calculating his/her chances of readmission using machine learning. The doctor can review these results and take appropriate action by providing some suggestions and educational content to the patient, and reschedule patient's next feedback.**

I. INTRODUCTION

Nowadays, computers can learn from experience through machine learning instead of programming it to do on how to do a task. Some of the examples of practical applications of machine learning are netflix's movie recommendations, product recommendations on ecommerce websites etc. Machine learning is applicable in a vast range of fields, for example in medical diagnosis, economics, bioinformatics, marketing etc. Machine learning in medical diagnosis has various applications. One such application is predicting the chances of readmission of a patient after he/she has been discharged.

Nearly 20% of all discharges had a readmission within 30 days. Among this, an estimated 12% of readmissions are avoidable. Even if 10% of these readmissions are prevented, it can save up to $1 billion which is a huge amount. Today, hospitals are charged penalty for excess readmissions of patients that could have been avoided otherwise. In order to prevent this, the hospitals are now employing intervention methods like improved discharge process, post-discharge follow ups, remote monitoring, improved patient education and self-management support etc. Predicting the chances of readmission of patients can help hospital authorities and staff take proper measures to prevent avoidable readmissions and thereby reducing the penalty incurred due to readmission of patients.

We tried to come up with a solution using machine learning that would benefit both the patients and hospitals and reduce the spending in health care due to avoidable hospital readmission. We implemented two of the above mentioned interventions: Predicting chances of readmission and improved patient education and self-management support.

II. TECHNOLOGIES

A. *Machine Learning:*

For machine learning, Apache Spark 2.0.1 version is used. The algorithm for classification used is Random Forest Classifier.

B. *Back-end server:*

Server side scripting language is node.js which enables improving performance of application through asynchronous programming.

C. *Client-side:*

Client side scripting language is angularJs which gives the ease of programming due to its two-way binding feature.

D. *Framework:*

Node-Express 4 is used for web application. Also, the machine learning prediction services for new data is accessed by exposing the service over REST api using Flask framework.

E. *Message Oriented Middleware(MOM):*

RabbitMQ is used for communication between client and server of the application. It helps decouple parts of application and also improves reliability when user requests scale up to large numbers.

F. *Database:*

MongoDb for easy scalability and schema less database.

G. *Object Relational Mapping(ORM):*

Mongoose is used as ORM which enables built-in data validation and abstraction of mongoDB code from rest of the application.

H. Deployment

The application is dockerized and deployed on AWS EC2 micro instance along with spark MLlib services. Using Docker will provide lightweight process level virtualization and you can have more than one containers running on the same machine to scale out entire application.

III. ARCHITECTURE

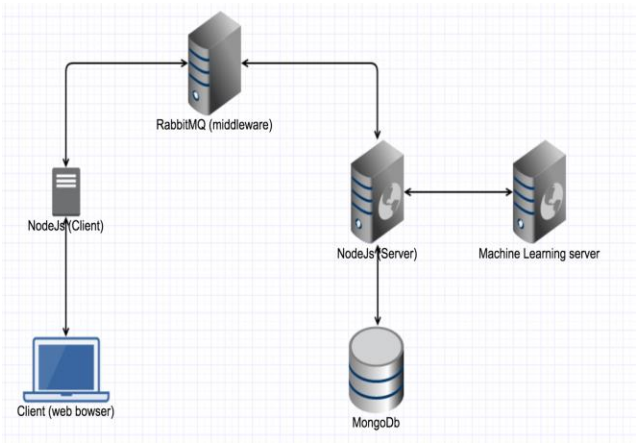

Figure 1

As shown in Figure 1, the different components of system are connected with each other. For example, when a user submits credentials in order to log into the system, the browser forwards this request to node.js client. Further, the node.js client communicates with node.js server using MOM-RabbitMQ. The node.js server validates the credentials against encrypted credentials stored in mongoDb database.

IV. PROJECT DESCRIPTION

Our project consists of two parts: web application and machine learning.

On the web application part, there are three types of users that will be using our application – doctor, out-patient care nurse and admin. We have developed a web application where the hospital's out-patient care staff (nurse) can take follow ups from patient after their discharge from hospital. Once the follow up is taken, the chances of readmission of patient are predicted using machine learning based on the inputs provided by the patient in the follow up. The nurse can also look at the list of patients whose follow ups are pending.

The doctor can review the patients file and take necessary action. While reviewing the file, the doctor can also see a graph that shows the trend of chances of readmission over different follow ups. As a part of intervention, the doctor can suggest some educational videos to patient for self-management of patient. The doctor can also upload new educational content on the portal for the patients. He/she can write some notes and send it to the patient via email. The doctor can also reschedule the next follow up for a particular patient.

The admin can view various statistics in the application like number of patients that fall under different classes of readmission chances, the number of reviews taken and missed by different doctors, etc. in form of graphs and charts. The admin can add doctor and nurse in the system.

The nurse and doctors can see the list of patients whose follow-ups and reviews are pending sorted in order of the priority. Also, they can see a dashboard where a donut chart of the pending cases till current date are shown so that they can plan the work for rest of the day accordingly.

On the machine learning part, we have used Random Forest Classifier which gives the prediction and probability of the prediction being true. Based on the probability and the accuracy of the algorithm, the percentage of readmission chances are calculated. We have used Apache spark as tool and python as programming language for machine learning.

V. FEATURES

A. Security:

a) It is secure against cross site scripting (XSS) attacks.

b) Web application is secure against SQL injection attacks.

c) Passwords are encrypted with bcrypt which uses hashing and salt to provide advanced security in the application.

d) Proper validation checks are provided at client as well as server side.

B. Functional:

a) Doctors and nurses can view the list of patients whose reviews/follow-ups are pending.

b) Nurse can take follow-up of patients and doctor review those follow-up.

c) Admin can view various statistics. Admin can add a doctor / nurse to the system.

d) Doctor can upload and send educational content to patients via email.

e) The chances of readmission of patients are predicted.

C. Memory and performance:

a) Proper care is taken so that garbage collection of objects is done and no situation of memory leaks arise.

VI. Challenges

A. The machine learning part was a bit challenging. Choosing the correct algorithm for prediction of readmission was challenging.

B. The learning curve was high for the project since team members were new to the MEAN stack technologies.

C. Dockerizing the applicatio.

VII. Conclusion

The cost incurred in healthcare domain because of hospital readmissions is huge. Using machine learning to address this problem can greatly reduce this cost and also prevent patients from the inconvenience caused by unnecessary hospitalization. By implementing the solution for the project, we gained a lot of knowledge both technical and domain specific (i.e. healthcare domain). We studied about the various intervention methods for reducing readmissions. From a technological point of view, we learned and got a chance to have a hands on experience with machine learning using spark.

Currently the system has been developed to prevent readmission for patients with diabetes disease. As an extension, for future development, the project has been developed generic enough to extend it for other diseases.

VIII. Github Link

https://github.com/SJSU272Lab/Fall16-Team3

References

[1] https://www.healthcatalyst.com/CMS-reporting-requirements-4-changes-2014

[2] http://www.chqpr.org/readmissions.html

[3] https://aws.amazon.com/blogs/big-data/readmission-prediction-through-patient-risk-stratification-using-amazon-machine-learning/

[4] https://spark.apache.org/docs/1.6.2/ml-classification-regression.html#random-forest-classifier

[5] https://www.npmjs.com/

[6] http://flask.pocoo.org/docs/0.11/quickstart/

SCREENSHOTS AND EXPLAINATION

Screens for user: out-patient care nurse

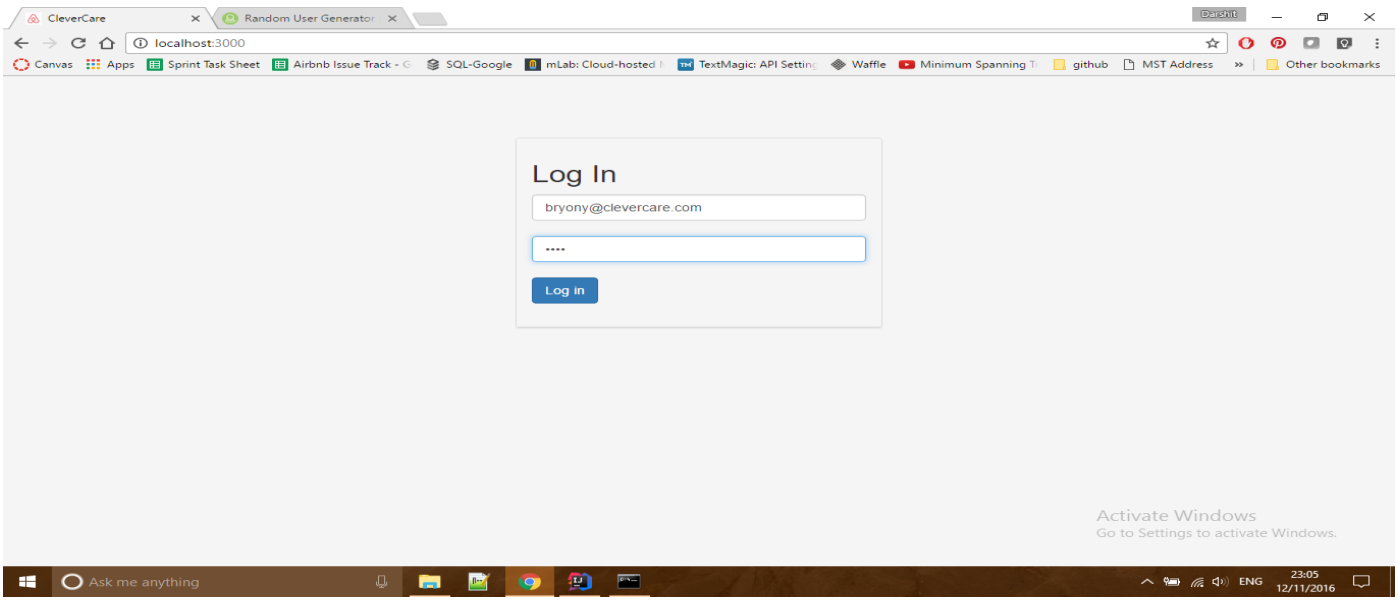

Figure 2.1

- Figure 2.1 displays the login screen of application.

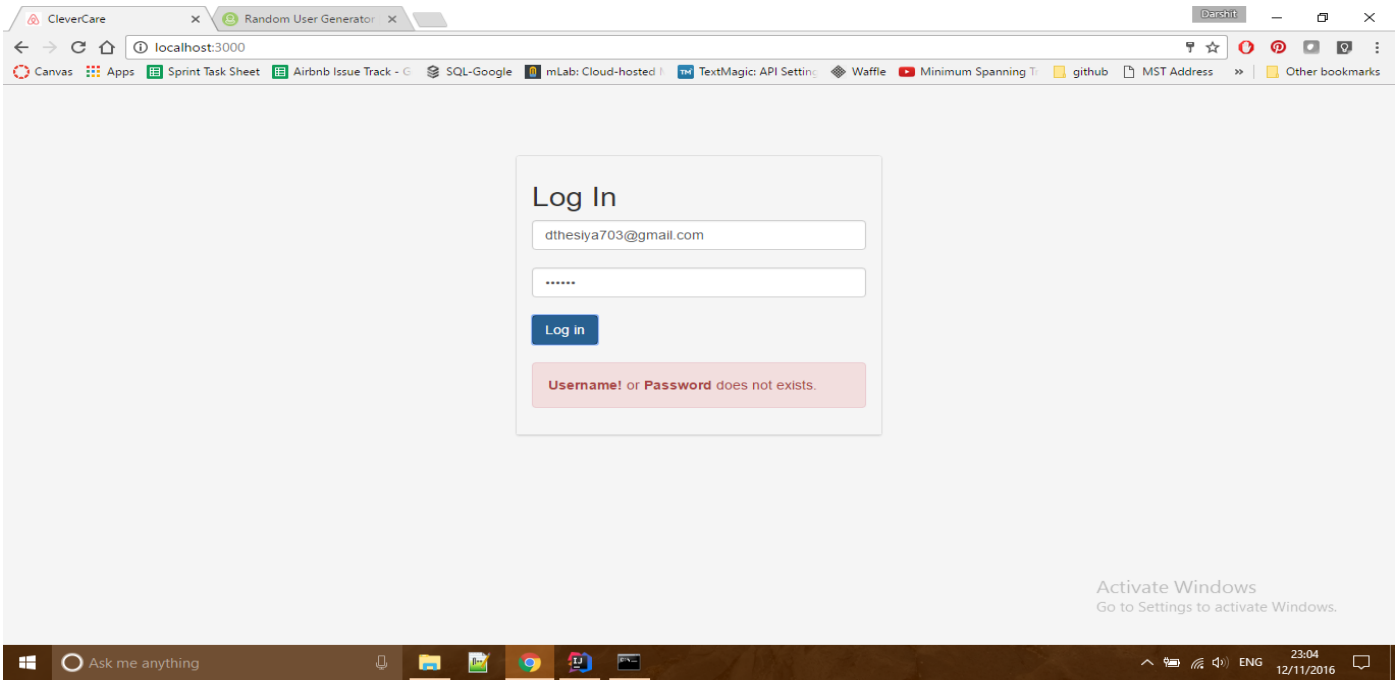

Figure 2.2

- Figure 2.2 displays the error message when the user enters wrong credentials.

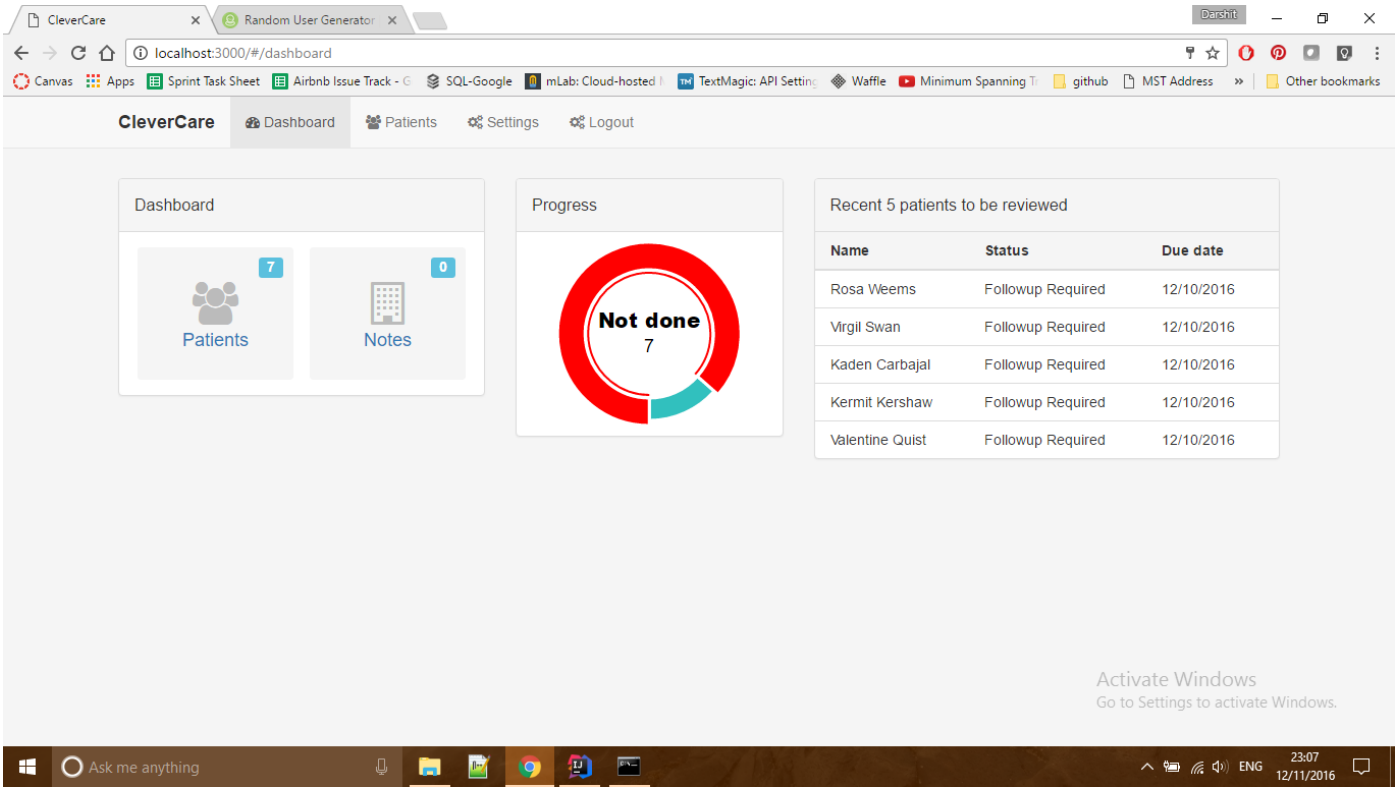

Figure 2.3

- Figure 2.3 shows the dashboard that nurse is redirected to when she logs in to the system.
- As shown in figure 2.3, there are few cards such as Dashboard, Progress, and Recent 5 patients to be reviewed.
- Dashboard card summarizes the total number patients whose follow up is pending. Clicking on patients and notes tab leads to list of patients whose follow up is pending and list of notes respectively.
- Progress card shows the number of patients whose follow ups are pending till today.
- Recent 5 patients to be reviewed shows the list of next 5 patients pending to be reviewed.

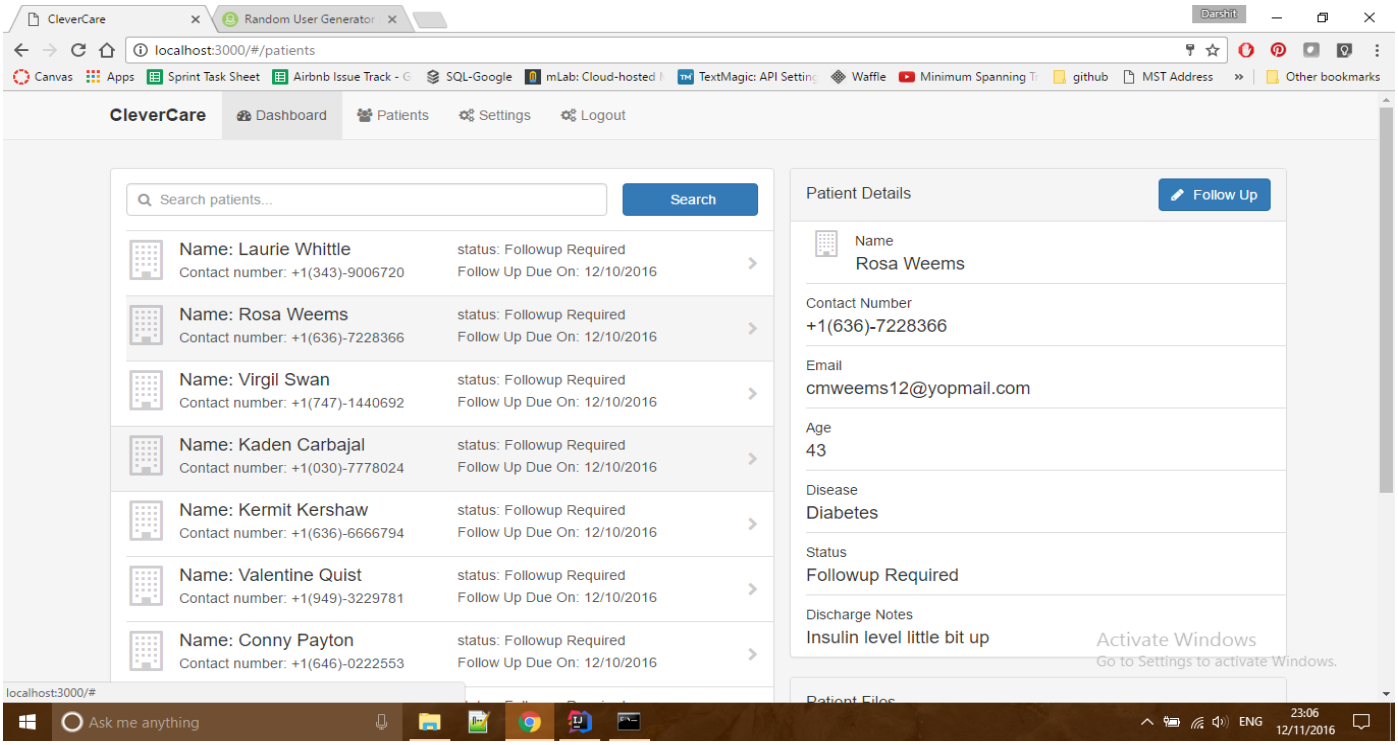

Figure 2.4

- Figure 2.4 displays list of patients sorted by due on date. The right pane on the screen displays currently selected patient.
- Nurse can search through the list of patients.

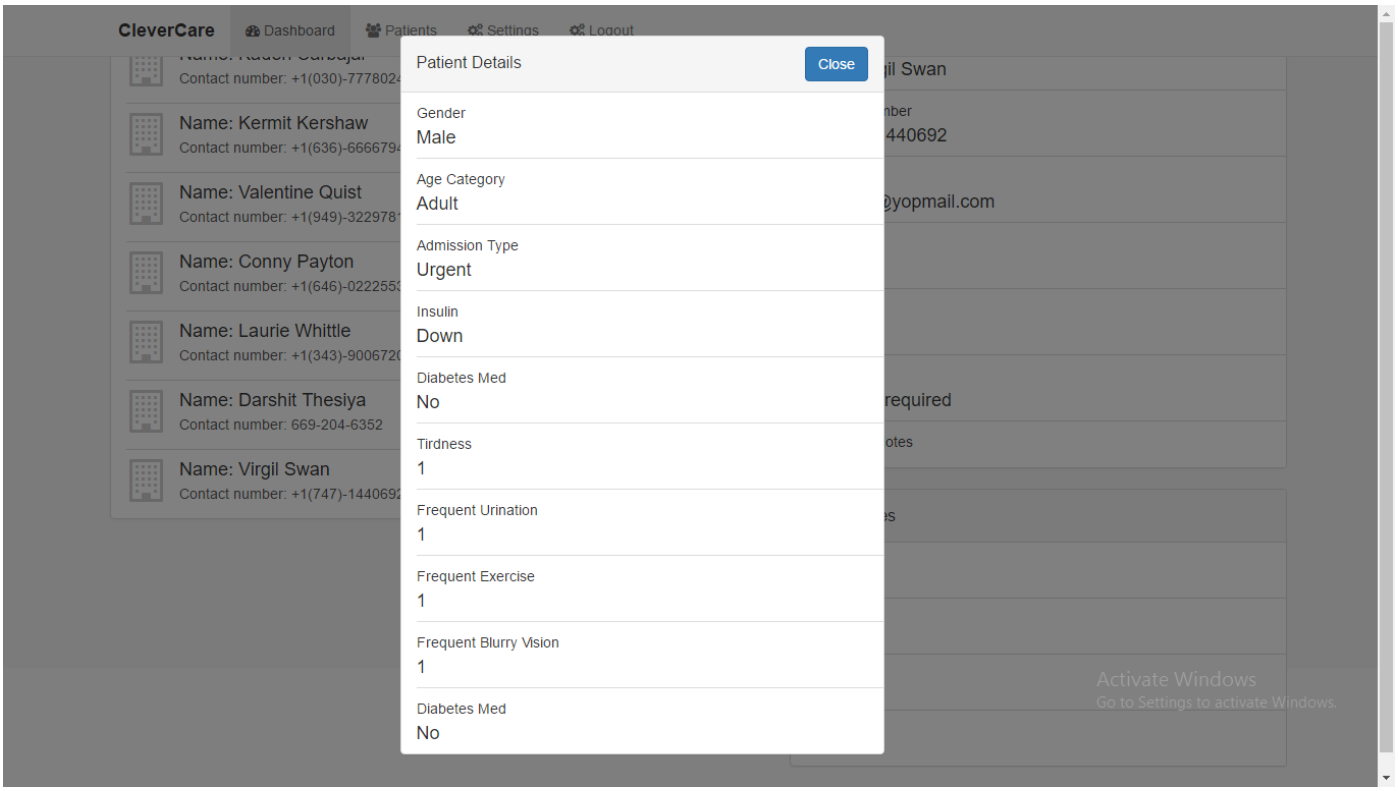

Figure 2.4.1

- Figure 2.4.1 shows the patients' previous follow up.

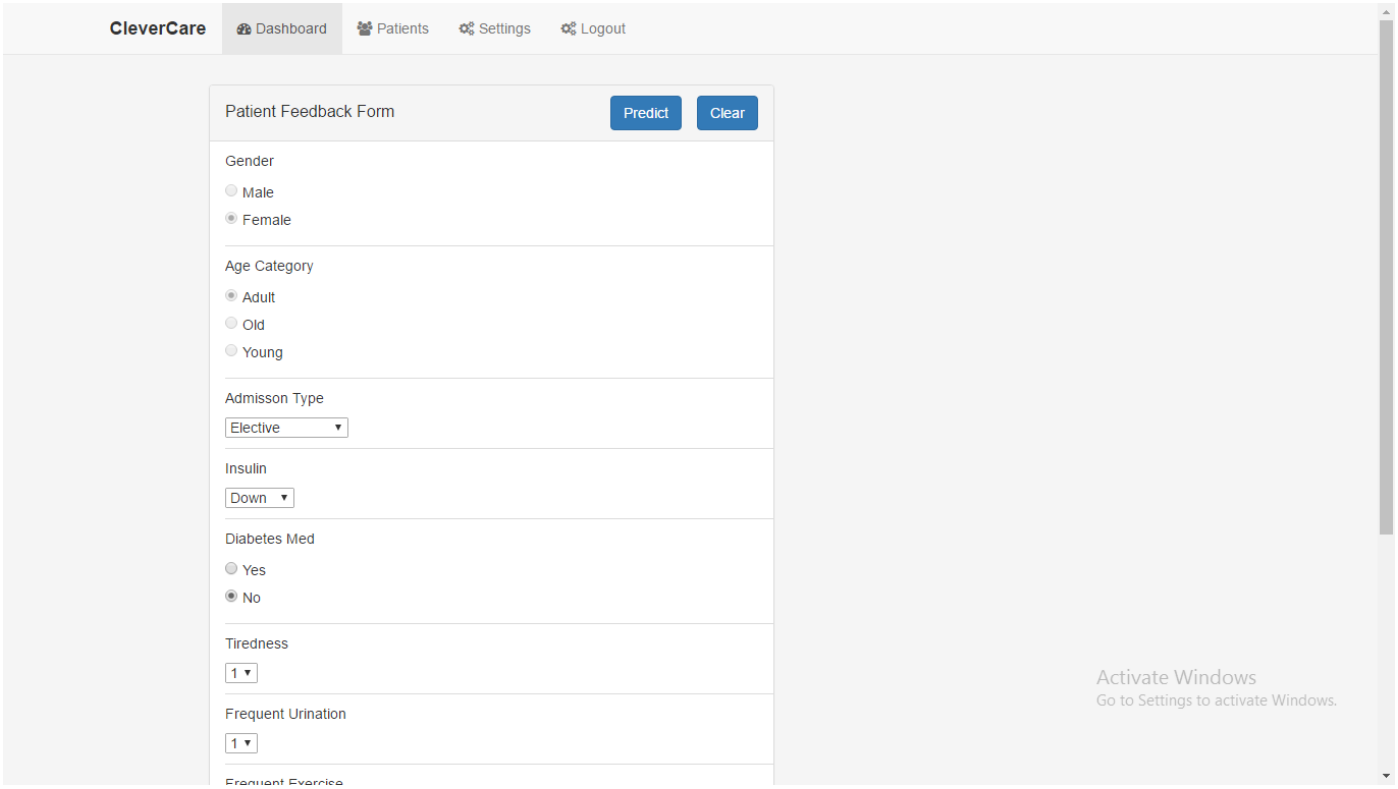

Figure 2.5

- Figure 2.5 shows the patient follow up form to be filled by nurse based on the input provided by patient.
- This screen is displayed when the nurse clicks on take follow up button shown in figure 2.4
- After finishing the follow up, the nurse clicks on predict button. This leads to a call to machine learning REST api which takes the form fields as input parameters and returns the percentage of readmission chances. This percentage is stored in the database.

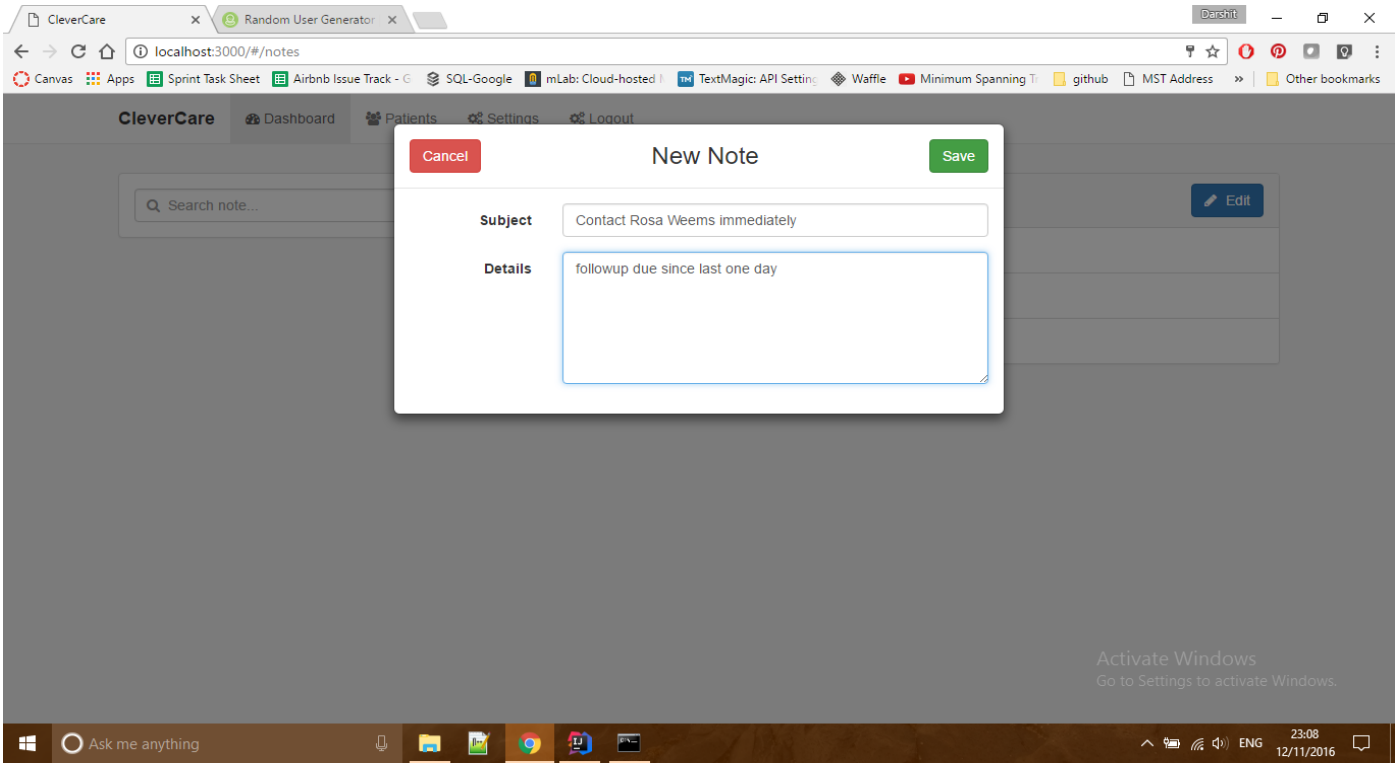

Figure 2.6

- When user clicks notes button shown in figure 2.3, it leads to notes page. Clicking on add button displays this pop up shown in figure 2.6. The nurse can add subject and details of notes and click save.

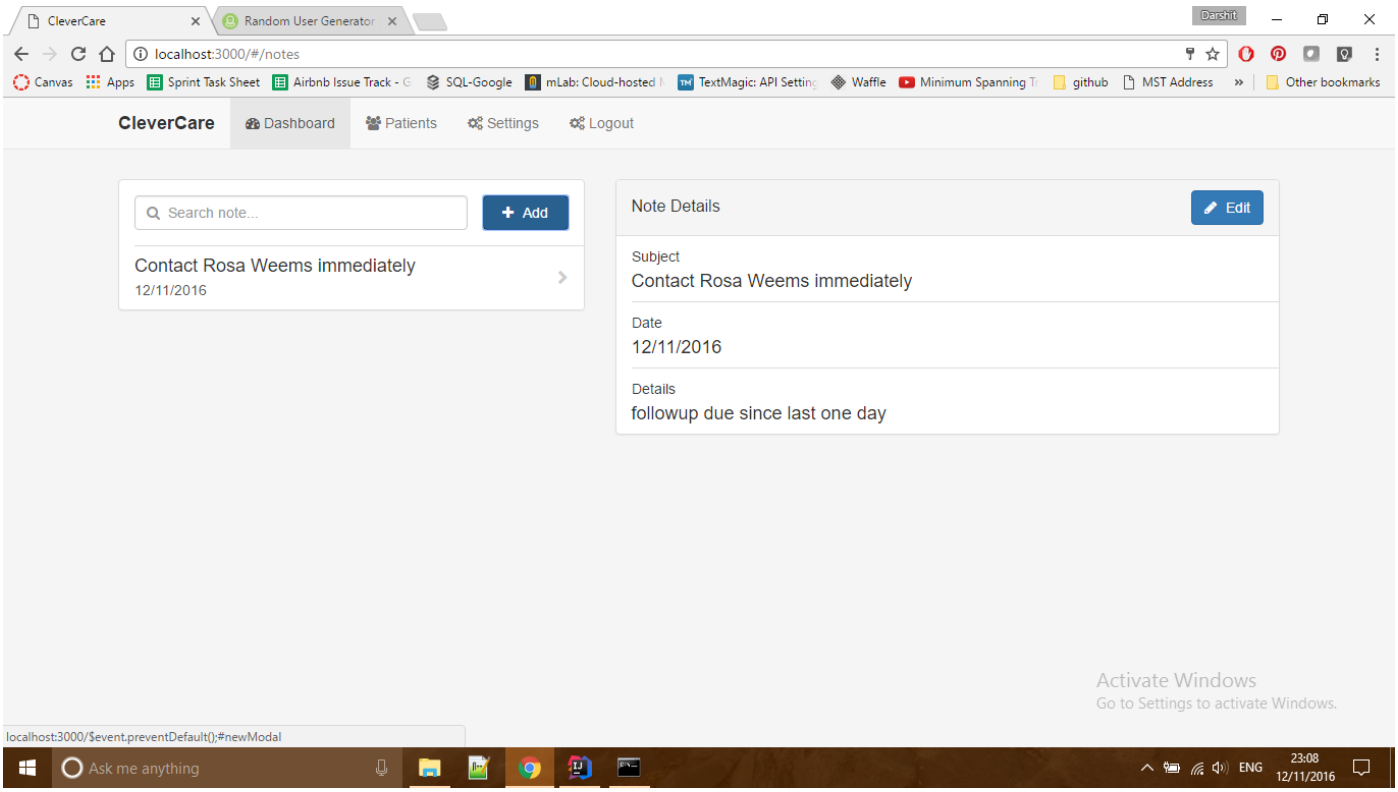

Figure 2.7

- On clicking save in figure 2.6, the nurse is redirected to page shown in Figure 2.7.

Screens for user: doctor

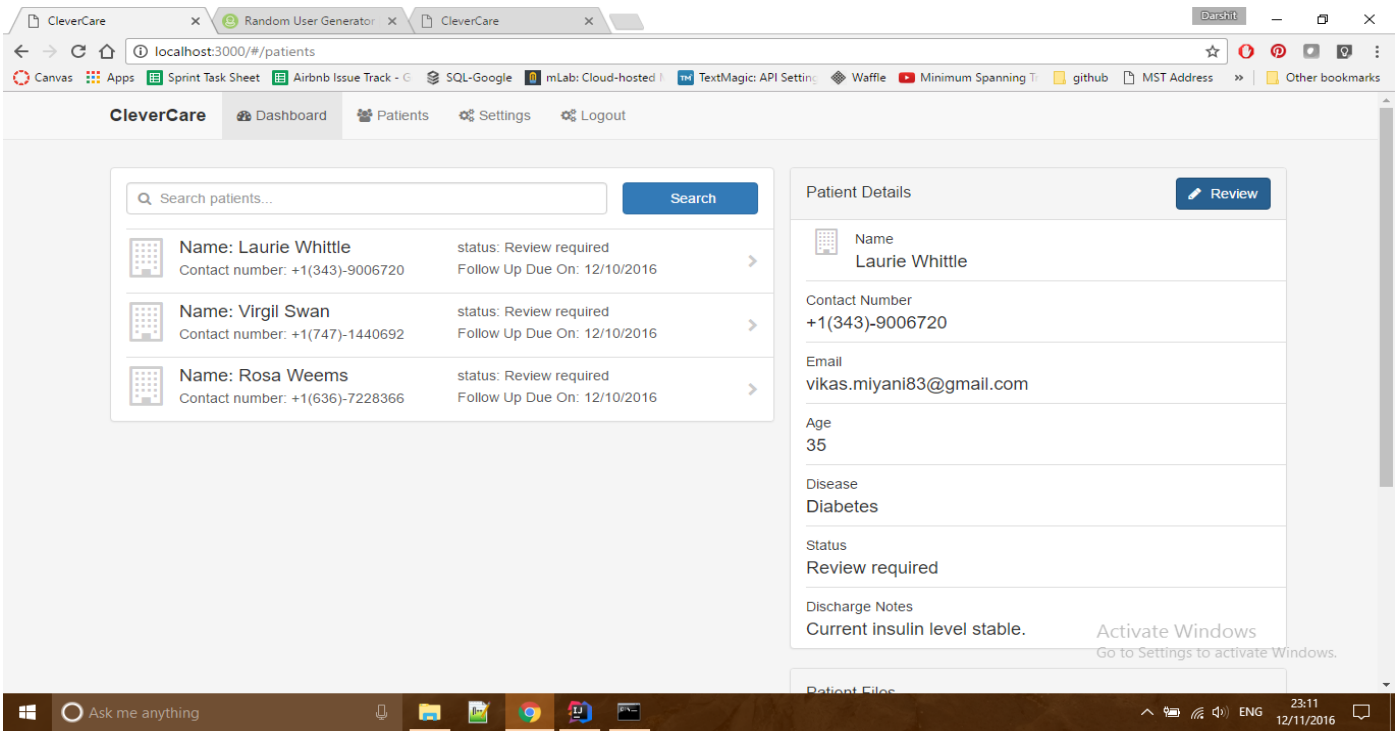

Figure 2.8

- Figure 2.8 shows the list of patients whose review is pending.
- Clicking on review button redirects doctor to screen shown in figure 2.9

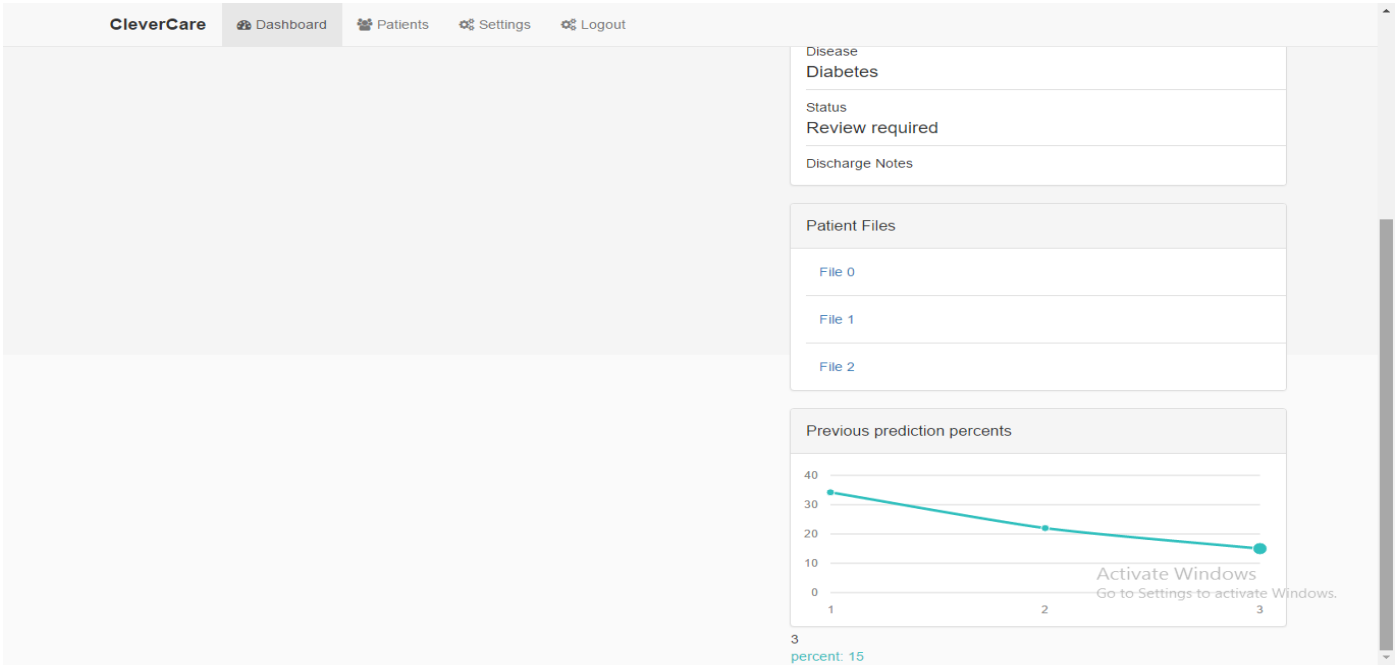

Figure 2.8.1

- Figure 2.8.1 shows the other half of the page shown in figure 2.8
- The patient files show the past review/follow-ups of selected patient.
- The previous prediction percent graphs shows the prediction (chances of readmission) across past reviews/follow ups.

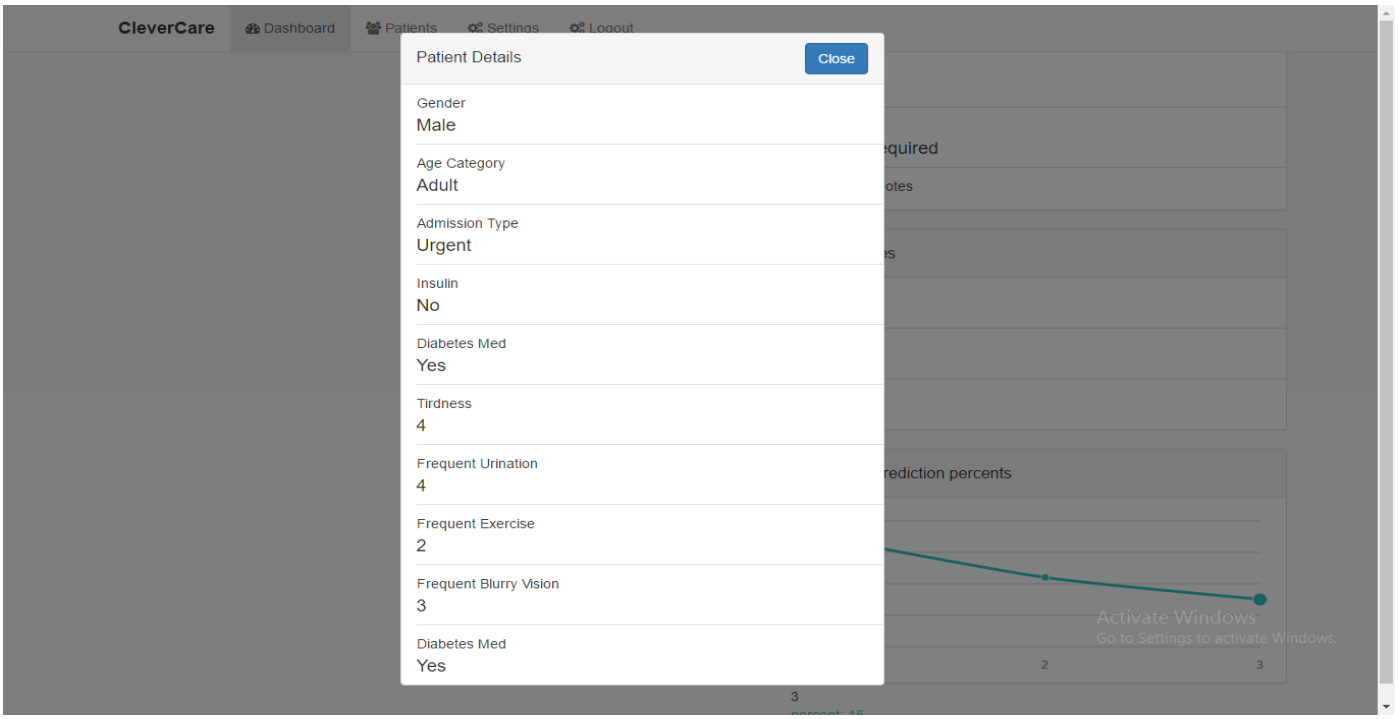

Figure 2.8.2

- Figure 2.8.2 shows the past patient review file.

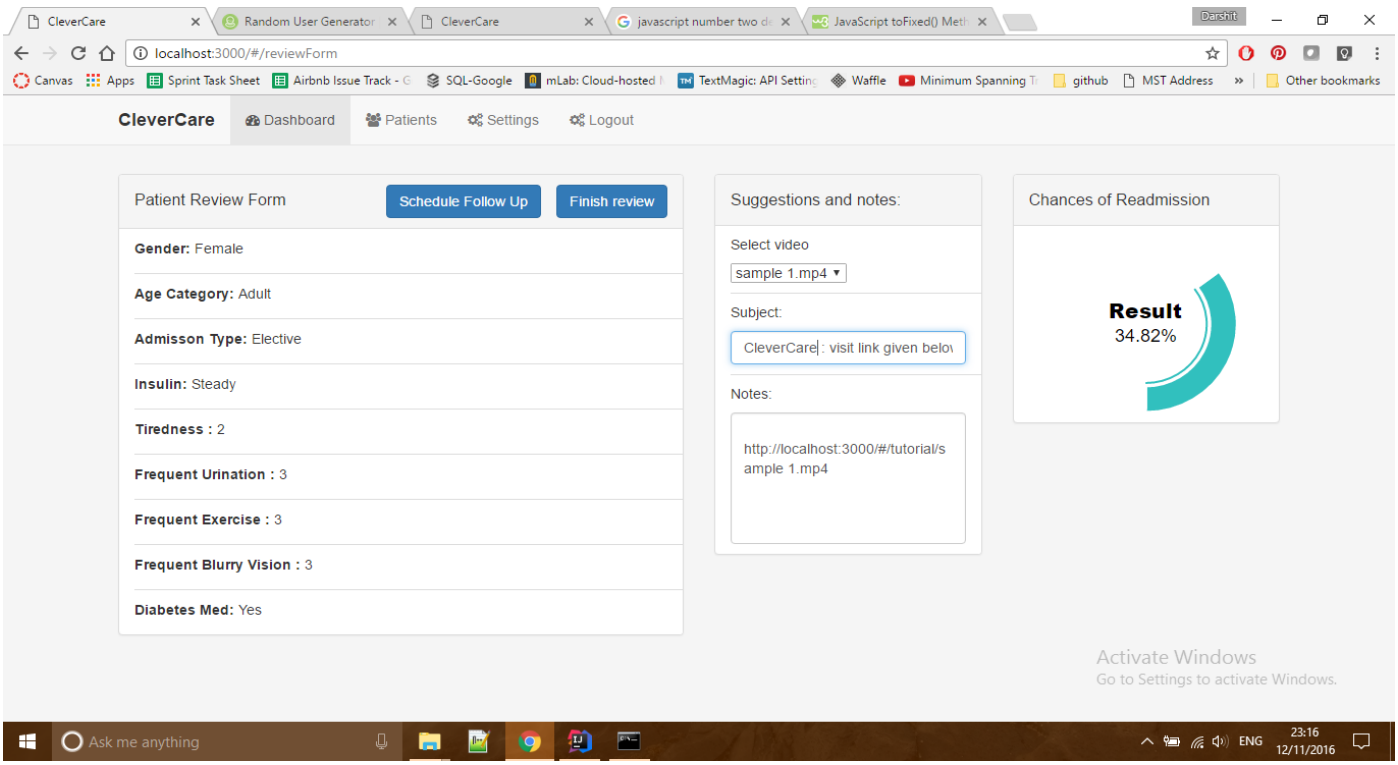

Figure 2.9

- Figure 2.9 shows the review form. Here the doctor can suggest some educational content to the patient via email.
- The Chances of Readmission card shown in figure displays the chances of readmission predicted by machine learning algorithm when the follow up is taken.
- The doctor can schedule another follow up by clicking on schedule follow up button.

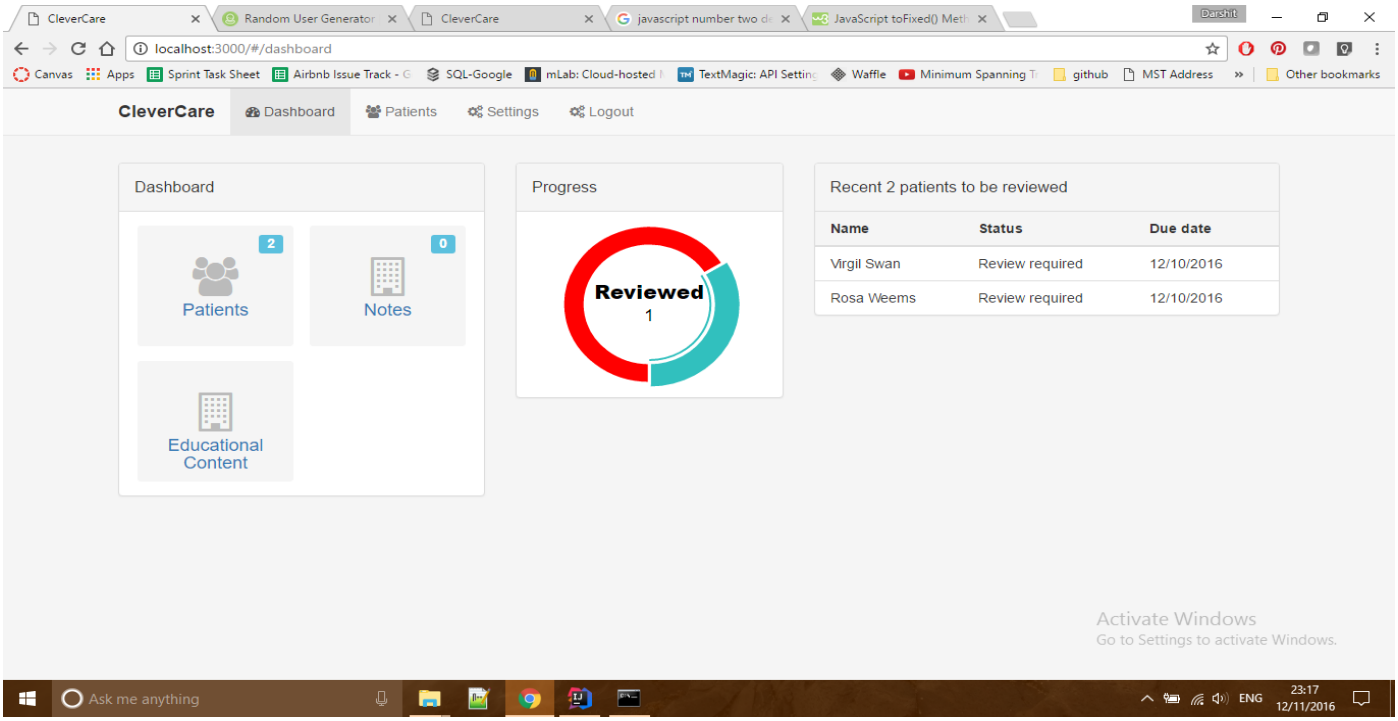

Figure 2.10

- Figure 2.10 shows the dashboard for doctor. Here the dashboard card contains one additional button labeled Educational Content. When doctor clicks it, he/she is redirected to a page where some educational content can be uploaded.
- The progress card shows the summary of number patients till date whose review is pending.

Screens for user: admin

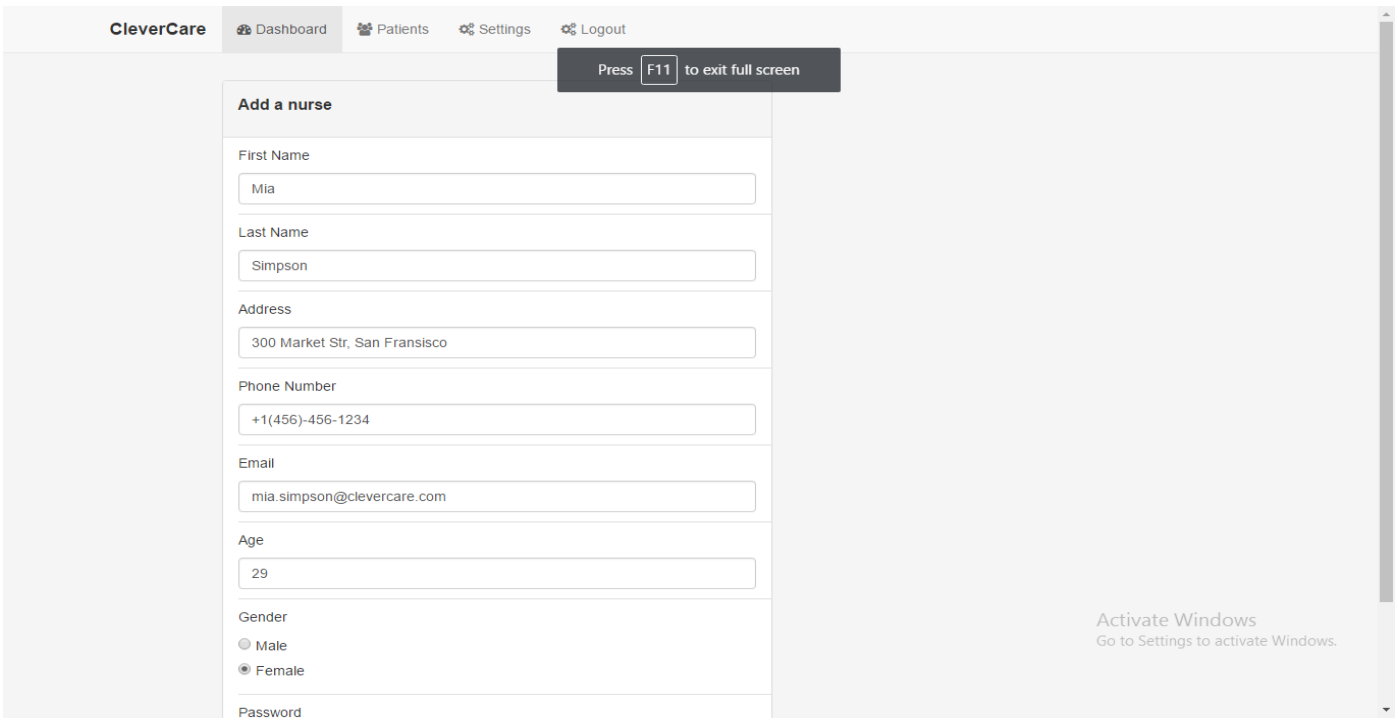

Figure 2.11

- Figure 2.11 shows the form provided to admin for adding a nurses' details to the system.

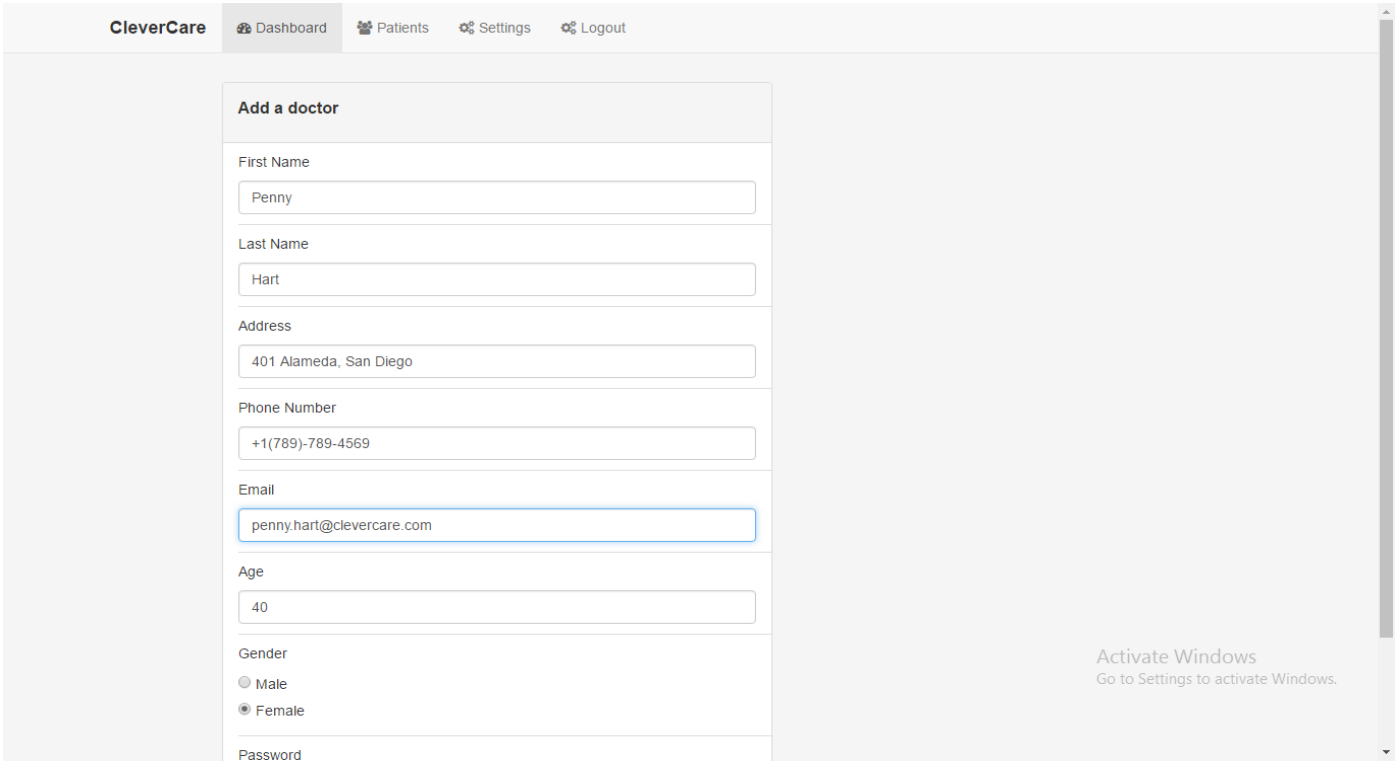

Figure 2.12

- Figure 2.12 shows the form provided to admin for adding a doctors' details to the system.

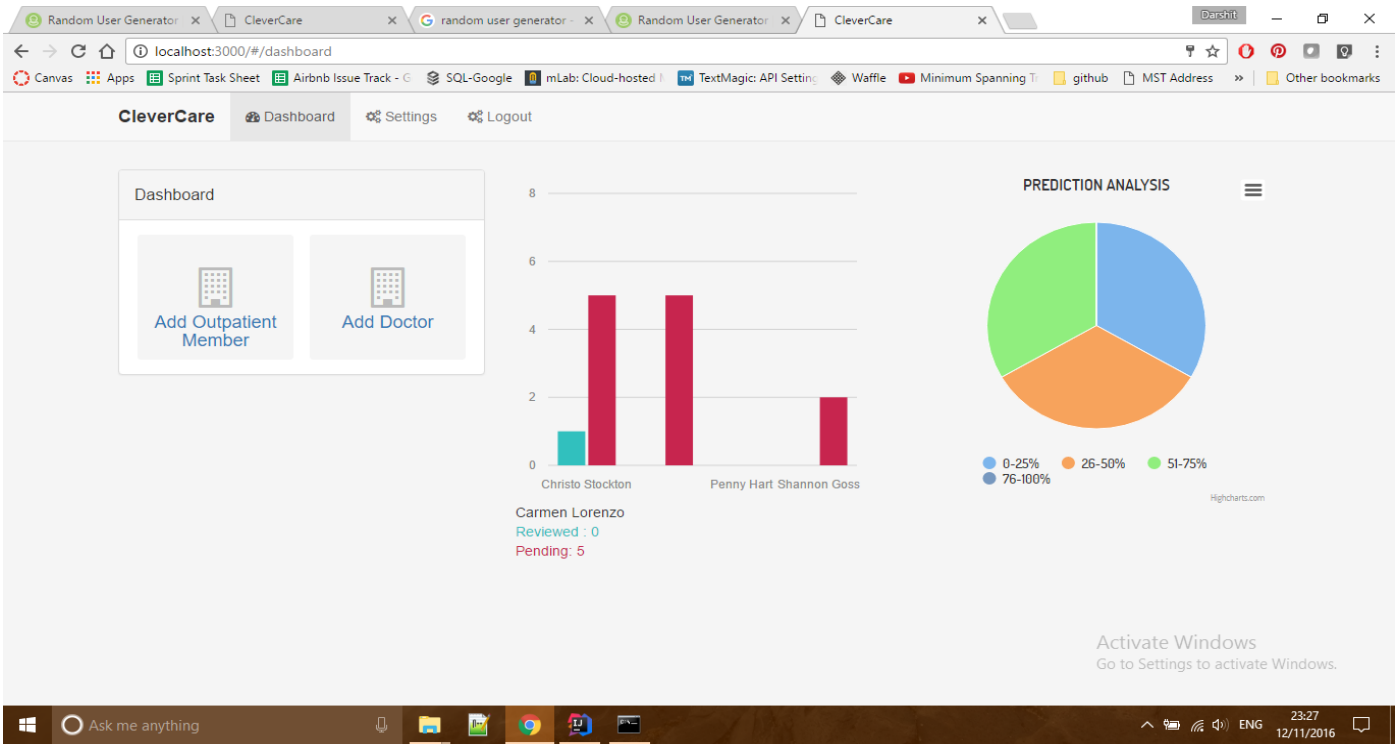

Figure 2.13

- Figure 2.13 shows the dashboard for admin. It comprises of various graphs and charts that provide analysis like the pending vs completed reviews by each doctor; a pie chart showing the number of patients falling under different ranges of readmission chances.