**Team Name - Regression**

**Team Leader -** Siddhant Dangi

**Email ID -** [f20170074@pilani.bits-pilani.ac.in](mailto:f20170074@pilani.bits-pilani.ac.in)

**Solution Brief Overview**

As the number of people testing positive is rising exponentially, we’re all witnessing that the treatment centers often have no availability of beds and are already working at full capacity. This leads to inefficiency and confusion in communication while transferring patients from a COVID-testing facility to these treatment centres. Our application provides a platform for the hospitals to share real-time statistics like number of beds available, presence of testing facilities, number of ventilators and other resources.

The application also provides live monitoring of the safety status of the locations around people which will help manage crowds and depict the safety scenario in an efficient manner. This would help the user make an aware decision before visiting a place after the lockdown relaxes.

**Solution Description**

We’re designing a web application to help everyone feel empowered to fight the multifaceted problems posed by this pandemic in a much better way - those who have been affected as well as those who haven't been yet.

As the number of Covid-19 cases continue to escalate rapidly, the number of people seeking treatment will overwhelm many hospitals. We’re already coming across news of **Mumbai running out of beds for critical Covid patients although only about 30% patients require hospitalised care** in a Dedicated Covid Hospital. The major reason behind this was concluded to be **inadequate resource-monitoring** leading to inefficient utilisation of critical care beds and equipment.

Having a handle on the local hospitals’ capacity and resource availability could help balance the load of Covid-19 patients requiring hospitalization across a region by allowing medical staff to send a patient to a facility where they are more likely to be treated quickly. But many states **lack real-time data** on their current capacity to treat Covid-19 patients.

To address the immediate need of a real-time tracking system for Covid-19 beds and hospital resources, our team designed a platform with a live **‘Hospital Resource Monitoring Dashboard’**. The treatment centres update the real-time statistics like number of beds available and presence of testing facilities on this dashboard. We are using the **IBM cloud service Cloudant** to design and manage our database for Hospital records.

Another issue that the society is collectively facing with the relaxation of lockdown restrictions is the question of stepping out of our homes to purchase necessities and getting things repaired or serviced on an urgent basis which could not happen earlier but **wanting to avoid crowded places** at the same time. There have also been several enforcements by the Delhi Govt like the odd-even scheme for opening markets and malls to monitor the gathering of people in the Red Zone.

Our application provides the user with a close real-time approximate of the **number of people present** andthe **percentage of people wearing masks** at any given location, which helps the user make an aware decision. With the help of deep learning, a mask detection model is also trained and integrated with the app which calculates the **safety score** of a place based on the percentage of people wearing masks, using the live footage from a video camera (CCTV camera, drones or any video camera system feasible depending on the location). **IBM Cloudant database** is used to store the camera location information and corresponding safety scores. For publicly crowded places like malls and markets, this application is furthermore crucial as the safety score can clearly convey the peril of visiting the place and allow efficient **safety monitoring** consequently **avoiding overflow of people in places at any point of time**.

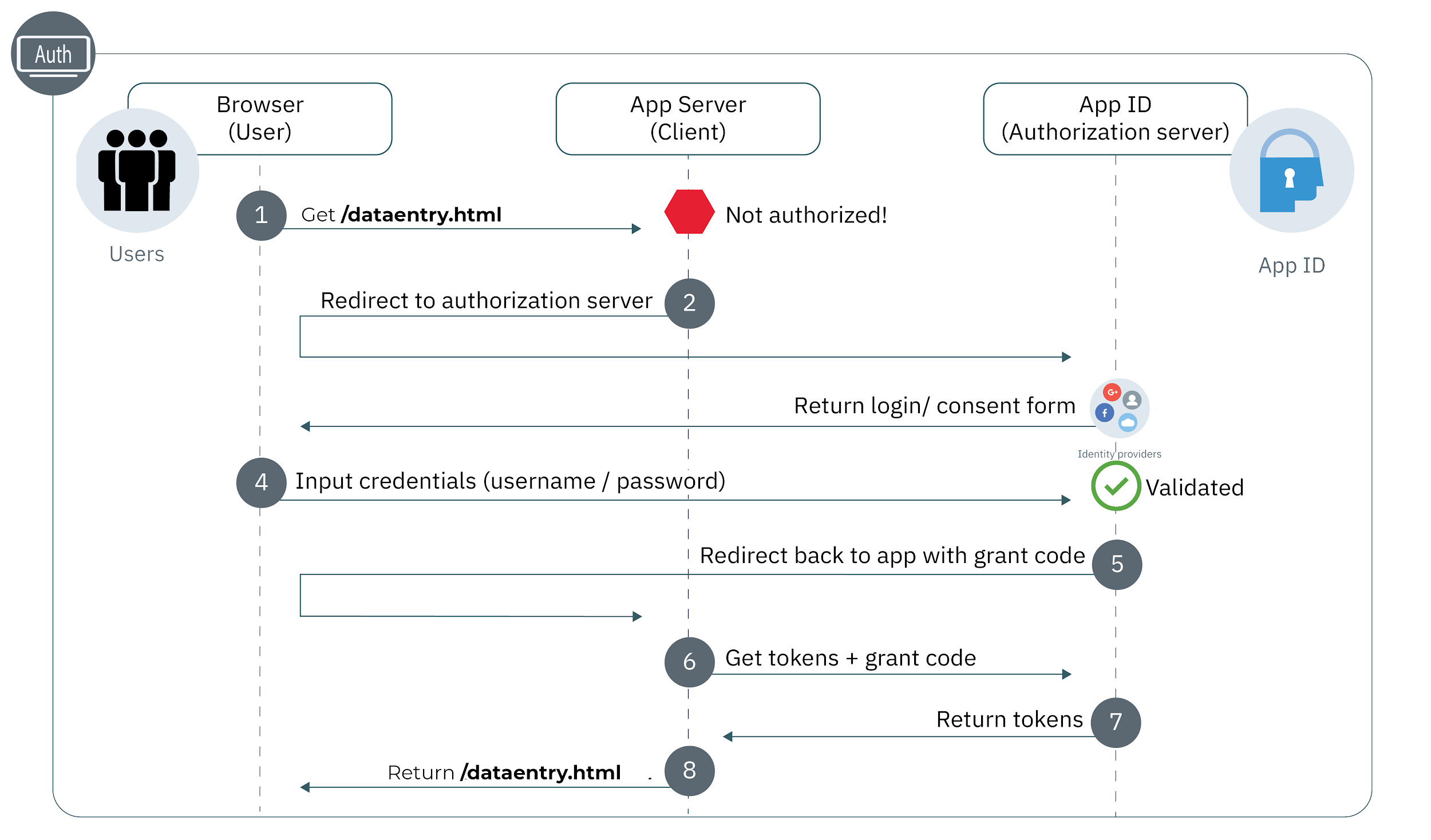
Our solution aims to solve the widespread and major issues posed by the COVID-19 crisis by providing a responsive, transparent and quick service for everyone. It stands out from existing solutions as it provides a unified platform to deal with problems affecting not any particular division of the society, but the society as a whole. It empowers the patients in distress with the ability to make a well-thought decision based on the real-time data from the nearest treatment centers. Likewise, the common public which will take to the streets after the lockdown relaxes are also presented with a statistic aid to take a calculated step. Further, the application can be scaled conveniently with the help of IBM services and thus offers an affordable and reliable solution to both the problems being faced at the present, as well as those which will be standing in front of us once the lockdown is completely lifted in near future. If the concerned authorities come onboard, this application can be resourceful to patients, doctors, emergency medical personnel and all the other users beyond measure.

**Solution Architecture**

The solution consists of modules which have been developed parallely and then integrated in different phases. The individual module architecture and their integration to the web application have been detailed below.

### **Integration of Hospital Monitoring Database with the web application**

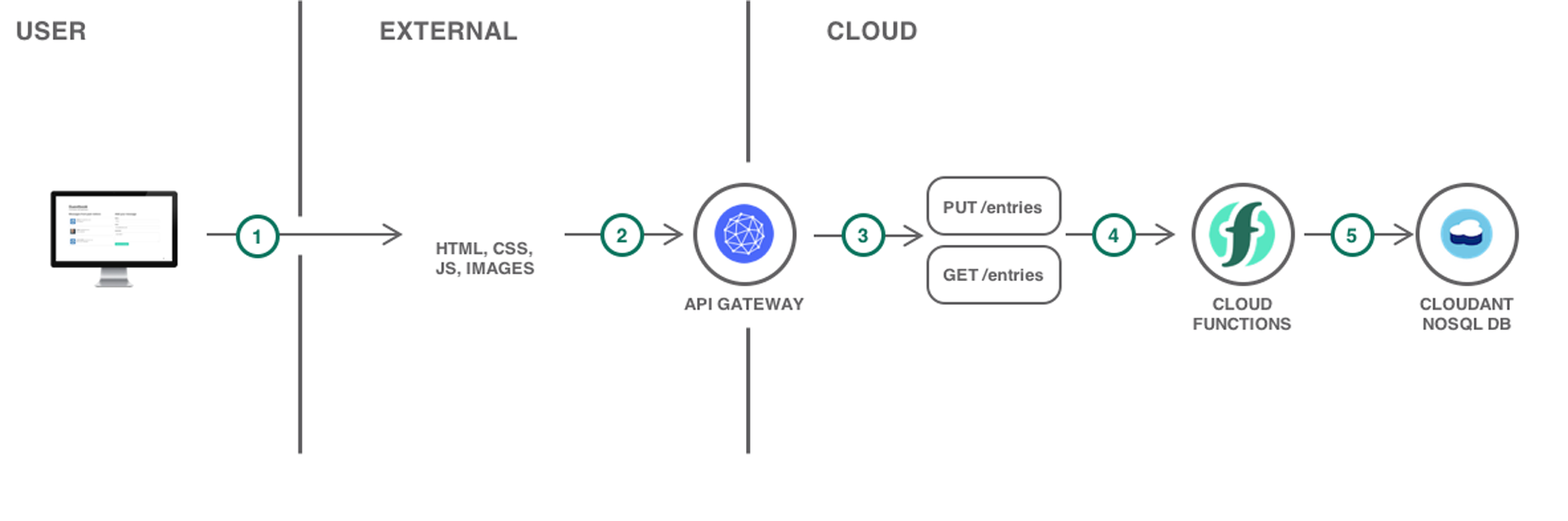
### **1. Authentication**

***App ID by IBM*** is used to monitor authentication for the login procedure in the app. Only hospitals will be authorised to input data into the form at dataentry.html. If an unauthorised person does so, an alert message shows up and the user is prompted to login. Their input will not be recorded until they are authorised. The workflow for the App ID looks as follows: -

***Fig 1. Workflow for App ID***

**2. Hospital Monitoring Database**

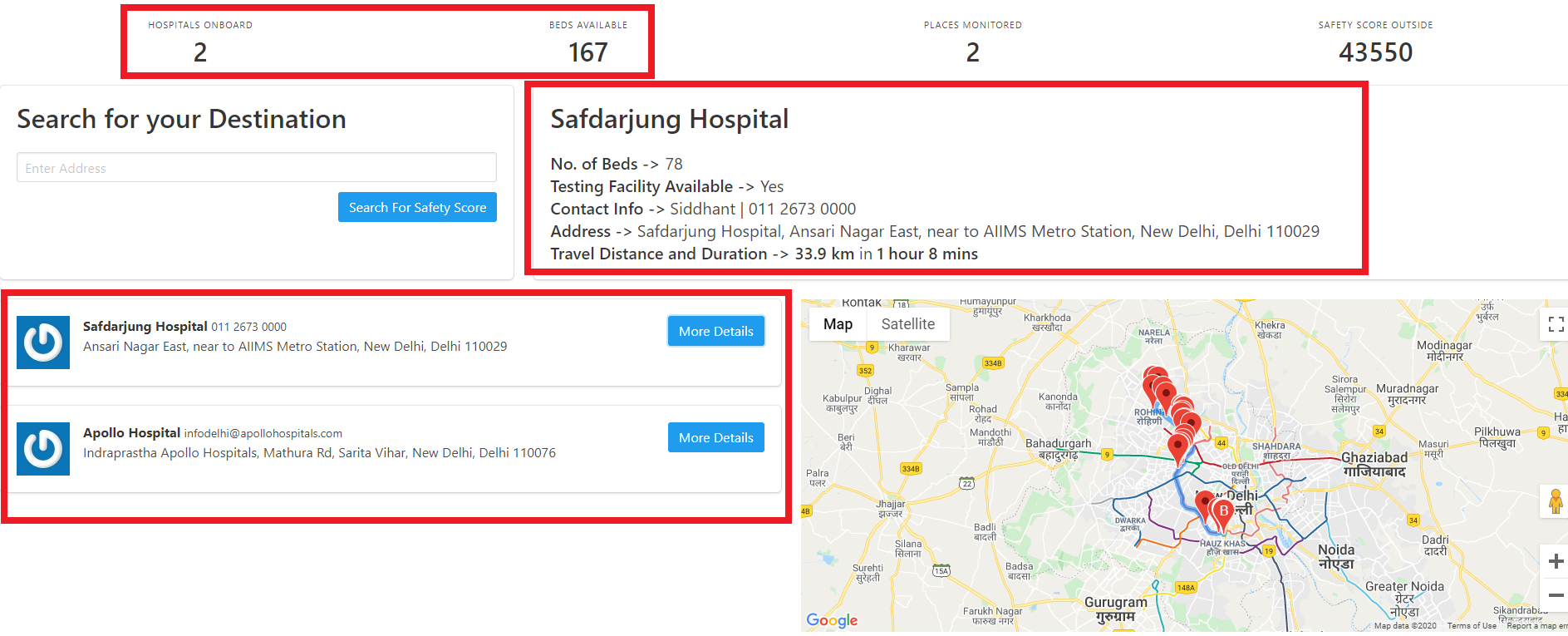
***IBM's cloud-service Cloudant*** is used to host the Hospital Monitoring Database containing real-time statistics on the presence of covid-19 test facility and number of beds available. Cloudant is used to setup a NoSQL Database which can then be used with a serverless web application.



***Fig 2. Workflow for IBM Cloudant***

The details filled by the hospital post login, are sent to the guestbook database (Hospital Monitoring Database) on click of the ***Submit Details*** button.

A GET request is also made to the database to retrieve the data regarding hospitals onboard which is displayed on the ***Currently Onboard*** tab of the web app. On clicking the ***More Details*** button, all the details for the particular hospital are retrieved and made visible to the user.



***Fig 3. Overview - Hospital Monitoring Dashboard***

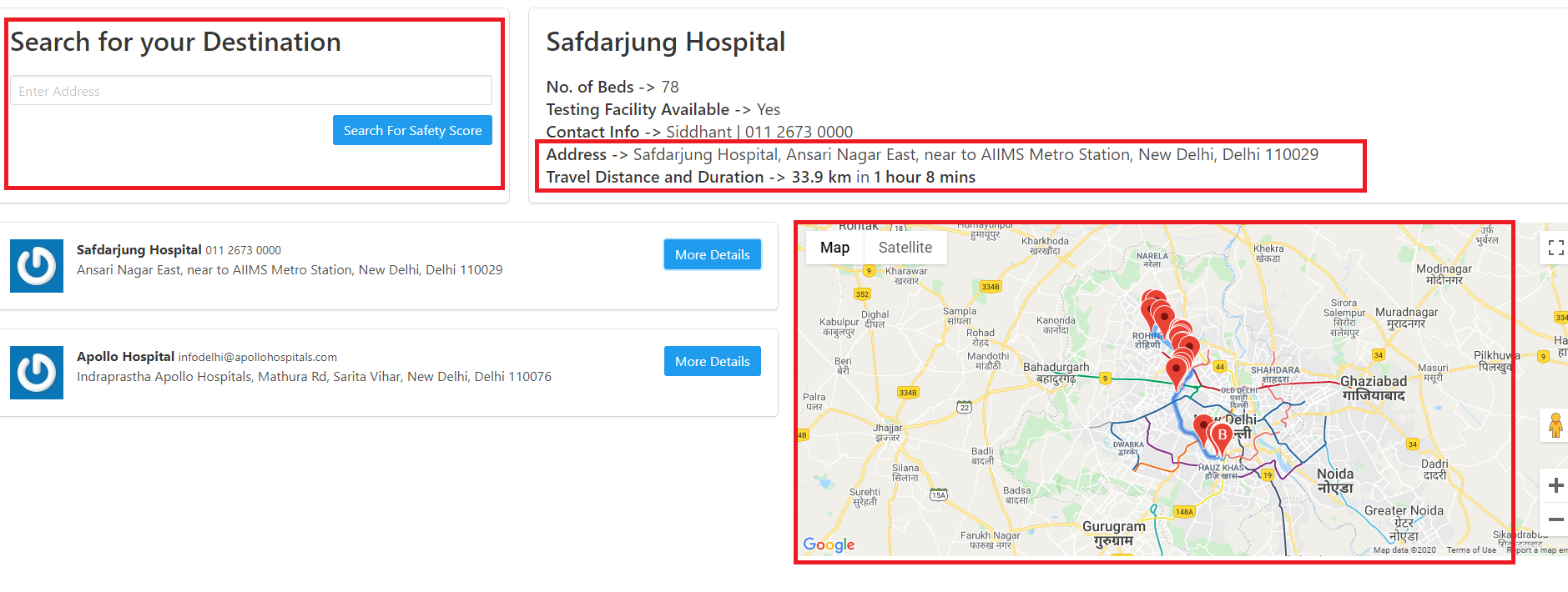
### 

### **Integration of the web application with Google Maps Platform**



***Fig 4. Integration of Google Maps Platform with app***

When the user inputs a specific location in Search for your Destination bar or clicks on the ***More Details*** button for a hospital, the location address string is retrieved and sent to the ***Geocoding API***, which finds the latitude and longitude for the place. Then using the ***Directions API***, the shortest route to the destination is computed and displayed on the map. The ***Distance Matrix API*** finds the distance and travel time required to reach the destination.



***Fig 5. Integration of map-based features with the app***

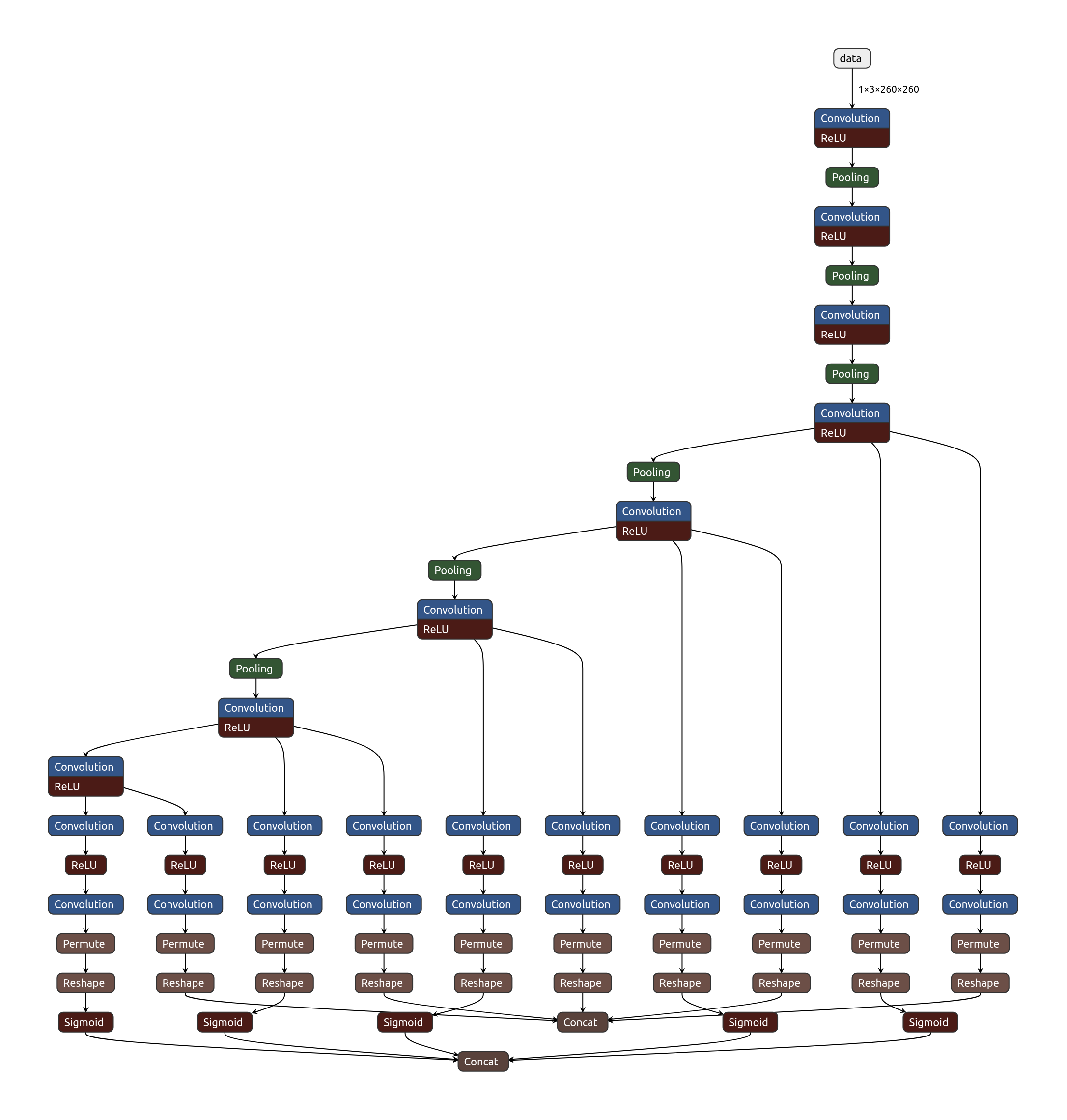
### 

### **Prediction of Safety score with Mask Detection Model**

### **1. Architecture**

We used the structure of SSD (Single Shot Detector) Object Detector. However, to enable inference on the device integrated with the CCTV camera with minimum latency, the backbone network is light. It is easier to deploy the model to embedded systems (Raspberry Pi, Google Coral, Jetson, Nano, etc.).

The total model has 1.01M parameters. Input size of the model is 260x260, the backbone network has 8 conv layers. Overall, the model has 24 layers with the location and classification layers included. We merge the BatchNormalization to Conv layers in order to accelerate the inference speed. The model architecture has been shown below



***Fig 6. Model Architecture***

We apply face detection to compute the bounding box location of the face in the image. Once the faces in the image are detected, we can extract the Regions of Interest (ROI). The facial landmarks are used to localize and represent salient regions of the face, such as - Eyes, Eyebrows, Nose, Mouth and Jawline. Leveraging these ***facial landmarks***, the model learns as to which facial features are covered by mask and which are not. Thus, we have a classifier ready which adopts the following flow for mask detection:

* Take an input image -> Detect faces -> Detect ROI -> Use Facial Landmarks to Localize -> Classify as ***mask*** or ***no\_mask***

### **2. Integration of live Video feed with Mask Detection model to predict Safety score**

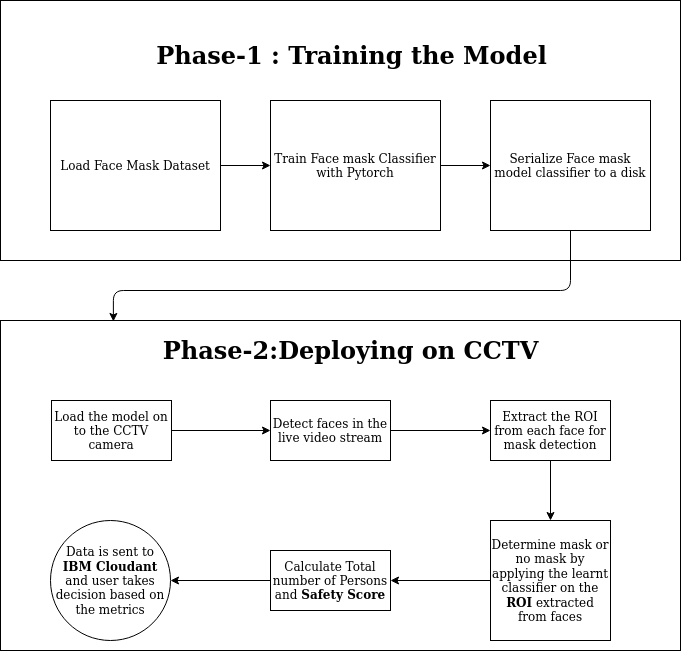


***Fig 7. Model Prediction***

The mask detection model can be employed on the CCTV cameras installed throughout the city. For the locations with no CCTV cameras in place, the model can be integrated with drones. As it is a light model with less number of parameters, live video stream captured can be directly inferenced on the connected computer or system.

Instead of processing the video stream on cloud, we are leveraging the edge computing facilities already available with the camera and sending only the corresponding numeric metrics calculated. This saves us from the overhead of sending the entire video feed to the cloud and avoids any privacy concerns which may arise due to uploading of live video feeds. This makes the solution even more lightweight and easily deployable.

***Safety score*** is calculated as the percentage of people wearing masks in a given frame. The live video feed is taken as input in intervals of 10 minutes and a corresponding safety score is output for each interval by averaging the safety score over all the frames. A 10 minute interval is selected to give the user a fair idea about the area he/she is planning to visit during that time. Thus, this will help us as a society, be prepared to handle the situation better by delaying outdoor visits if they are not urgent or taking necessary precautions and increasing Personal Protective equipment (PPE) otherwise. The workflow described is summarized in the flowchart below:-

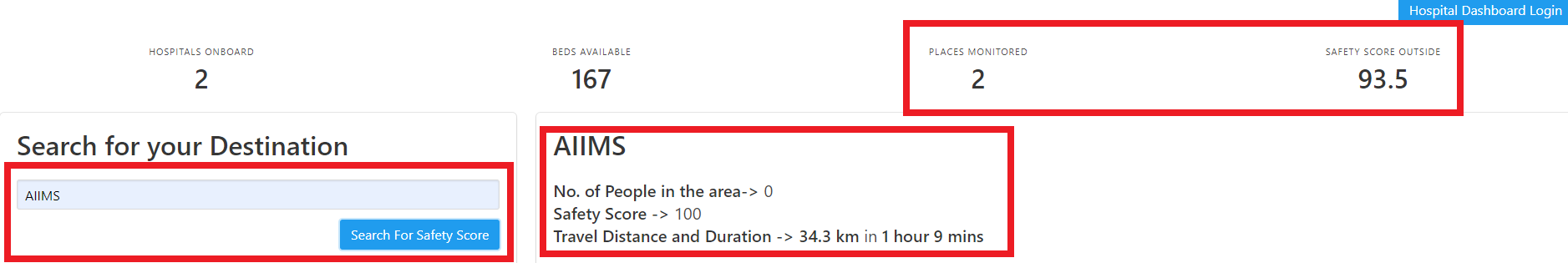


***Fig 8. Integration of Safety score with app***

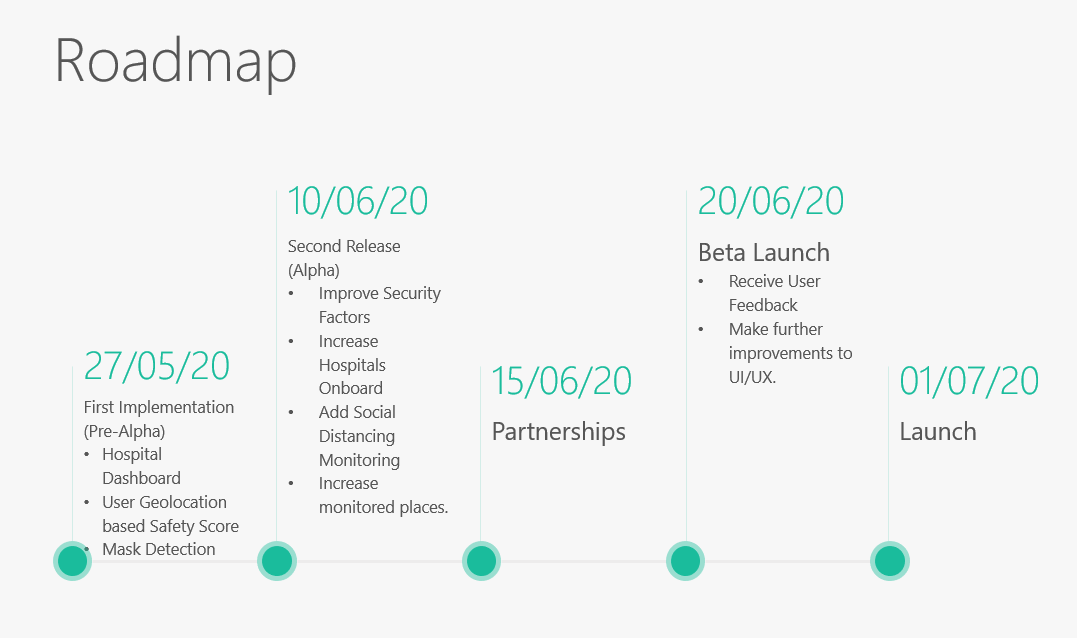
### 

### **Integration of location-based Safety score with the web application**

***IBM Cloudant*** is used to setup a NoSQL Database which can then be used with a serverless web application. The Python script uploads data points such as no of people in the camera feed, safety score (based on % of people wearing masks), latitude and longitude of the camera location to the Cloudant mapbook database. The data is retrieved via a GET request, and then based on the user's location entry in the input bar, the database is searched for an entry. If a camera feed is found in the region, the score associated with it is retrieved and displayed to the user. The web app also keeps a track of the average score in all the places that are being monitored.



***Fig 9. Safety score and number of people displayed corresponding to user’s entered location***



The solution has already implemented features such as mask detection and managing a database of hospitals, which provides the users with access to information such as testing facilities and bed availability, which both play a crucial role and will be of huge help to first responders in getting the person in need of the required medical assistance to the nearest hospital with those facilities available. A few security fixes which will not take more than a week will make the solution ready for deployment on a commercial scale. Our initial strategy will be to get some hospitals involved in the treatment of COVID patients on-board to share the required information, so that it can be displayed on the webapp. Our next step will be to collaborate with RWAs, and various malls and shops in crowded areas, so as to run the python script necessary for mask detection on their CCTV systems. For privacy and safety reasons we never gain direct access to the video feed, but the script only uploads the number of people, safety score and the coordinates of the camera on the server. The python script has been optimized so that it can also run with minimal computation requirements, and hence can also be used by the Police on their drones to monitor the situation in a place with much more ease.

For our monetization strategy, we will charge the RWAs, shops, the police, and the malls for installing and deploying our system in their camera systems. The pricing model will be decided after doing an alpha launch in a small market in Delhi. The nature of our solution allows us to even grow beyond the COVID pandemic, as quick access to healthcare, monitoring public health and maintaining social distancing norms has become a necessity and will be relevant and important in the post-COVID era. This makes our business proposition scalable, as this solution has a need in almost all the crowded markets of the place.

For funding requirements, we estimated an initial round ~100k USD, will help us bring the solution to scale and have >80% coverage (places monitored and hospitals) in metropolitan areas of India. ~50k USD will be required for continuously using the IBM and Google Maps Platform. 20k USD will be utilised to raise awareness regarding our solution through targeted marketing as a tool of economic customer acquisition. The rest of the 30k USD will be utilised for further R&D to make the webapp, more user friendly, add more functionality and later even evolve to take the form of a mobile application.

**IBM Cloud Services/Systems**

The IBM Cloud Services/Systems used in building the web application are:

* [IBM Cloudant](https://cloud.ibm.com/docs/Cloudant?topic=Cloudant-getting-started-with-cloudant)
* [IBM App ID](https://cloud.ibm.com/docs/appid?topic=appid-getting-started)
* [IBM Functions](https://cloud.ibm.com/docs/openwhisk?topic=openwhisk-getting-started)