*“Children are our greatest treasure and future of the country”*

- Nelson Mandela

# Problem statement

The unprecedented situation due to COVID-19 has many parents apprehensive about the pandemic's negative impact on their child's education ; 42% of parents worry COVID-19 will affect child's education.

Adding to the anxieties, is the unavailability of a communication channel to connect to and solicit answers for their important queries or even the trivial doubts. With the schools completely shut down and school staff off work , calling the school helpdesk is no longer a viable option.

The texts on parent’s whatsapp group only adds fuel to the already disturbed mind, leaving the parents grappling with the thoughts about their child’s future

# Solution

Eurobot is a mindful and smart conversation solution envisaged to address the above problem. It is a solution leveraging cloud technology; answering parents queries 24x7, sharing reliable information, extracting individual student specific data from cloud database.

The motto therefore is to setup a conducive communication environment and build trust between school and parents.

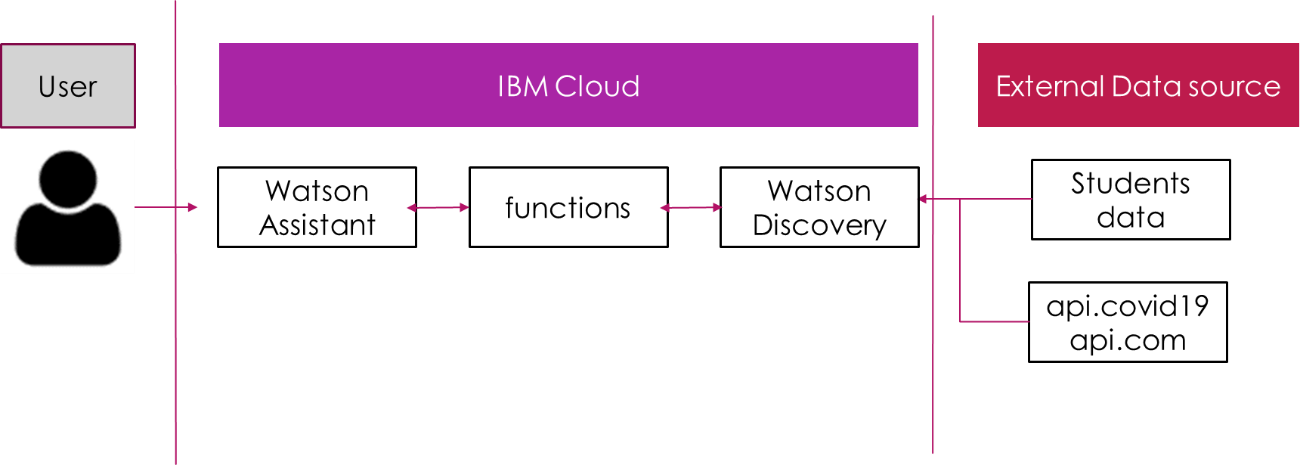
# Eurobot overview

* Eurobot is an AI bot to assist parents/guardian/students with answers for their queries or concerns on the academics or other logistics amidst covid-19 pandemic or otherwise too.
* The bot has a simple Interface where the user will have access to information around different functions.
* Connects to student’s data hosted on IBM cloud data service – Watson Discovery.
* Integrates with cloud data sources like COVID-19 stats and shares information instantly.
* The more you interact the smarter it will get.

# Technologies

* IBM Watson Assistant
* Conversation AI platform that can be deployed in any cloud or on-premises environment.
* Connects to Webhook to send HTTP requests to a single URL when one of the dialog nodes is configured to make a callout. This lets to pull in user information to personalize answers, post transactions to internal systems etc.
* IBM Watson Discovery
* Discovery is the IBM Cloud service to help you uncover connections in data by combining automated ingestion with advanced AI functions.
* Mapping external data to the discovery collection, the data set can be accessed from different applications with the help of discovery – API key, URL, collection id, environment id.
* Enables cognitive search and content analytics engine to applications.
* IBM Functions
  + Functions is a polyglot functions-as-a-service (FaaS) programming platform for developing lightweight code that scalably executes on demand.
  + It integrates with Watson Discovery.

# Architecture



# Access link

* For Eurobot interface follow the url: [https://ibm.co/2MJBan3](https://ind01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fibm.co%2F2MJBan3&data=02%7C01%7Crkotian%40techmahindra.com%7C7fdf6125eb0548c4887308d809595195%7Cedf442f5b9944c86a131b42b03a16c95%7C0%7C1%7C637269626510583187&sdata=vvjm9spHxzBSzX50JE1IXGdru3rnnchQnGp77Uagp7U%3D&reserved=0)
* Do keep the student GRID handy. It is the unique ID allotted to our child at the time of admission procedures.
* Watson Discovery parameters:
  + api\_key - iBROwMsUI1X8fnAzxgRKz2MuDUNF3tW0EJa1KmEf8f81
  + url - https://api.eu-gb.discovery.watson.cloud.ibm.com/instances/6ce6e59b-3b9a-484b-8040-af6bc53811d1
  + collection\_id - 07a8a07c-5600-4d66-830a-d0f6903bc1e6
  + env\_id- ac60e08e-0482-47bb-acb4-ca18963b6a3e
  + config\_id- 55baa341-3ed8-49bb-96e0-d786693c4701
* Function (Webhook api) :
  + <https://us-south.functions.cloud.ibm.com/api/v1/web/bfad874c-56f8-4644-9f54-8ba870097a5f/default/fnschooljs.json>

# Data Source

Both the data set here are hosted on Watson discovery as two collections.

* Student’s Data in json format (student.json)

Eg: {

"name": "Geeta",

"grid": "111",

"currentacad": "8A",

"rollno":"1",

"bookscollection": "2020-june-08"

}

* External data - <https://api.covid19api.com/summary>

# Function node.js code

|  |
| --- |
| /\*  Team TechApetite - Project Eurobot  \*/  /\*variable initialize\*/  var request = require("request-promise");  const DiscoveryV1 = require("watson-developer-cloud/discovery/v1");  /\*const {IamAuthenticator} = require("ibm-watson/auth");\*/  function getRandomInt(max) {  return Math.floor(Math.random() \* Math.floor(max));  }  /\*Main function\*/  async function main(params) {  /\* start - For covid stats connect to api.covid19 website\*/  if (params.type === "API") {  const summary = await request({  method: "GET",  uri: "https://api.covid19api.com/summary",  json: true,  });  let location= 'india'  for (var i = 0; i < summary.Countries.length; i++) {  if (  summary.Countries[i].Country.toLowerCase() ===  location.toLowerCase() ||  summary.Countries[i].CountryCode.toLowerCase() ===  location.toLowerCase()  ) {  return {  result: `Total Cases: ${summary.Countries[i].TotalConfirmed}\nTotal Deaths: ${summary.Countries[i].TotalDeaths}\nTotal Recovered: ${summary.Countries[i].TotalRecovered}\n\nSource: Johns Hopkins CSSE`,  };  }  }  }  /\* End - For covid stats connect to api.covid19 website\*/  /\* start - For students data extraction \*/  const discovery = new DiscoveryV1({  version: "2019-04-30",  iam\_apikey: params.api\_key,  url: params.url,  });  const offset = getRandomInt(50);  let inputpara = params.grid;  const queryParams = {  environment\_id: params.env\_id,  collection\_id: params.collection\_id,  config\_id:params.config\_id,  /\*natural\_language\_query:"",\*/  natural\_language\_query: inputpara.toString(),  };  data = await discovery.query(queryParams);  /\* if GRID do not exists in database report an error \*/  if (data.matching\_results===0 )  {  return { result: "GRID not found. Please enter a valid GRID to help you with details." };  };  /\* if data matching to GRID exists then return the details \*/  let studentname = data.results[0].name;  let studentrollno = data.results[0].rollno;  let studentclass = data.results[0].currentacad;  let studentbooks = data.results[0].bookscollection;  return {  result: `\ngrid value:${inputpara}\nname: ${studentname}\nRollNo: ${studentrollno}\nCurrent Academic Year: ${studentclass}\nBooks collection: ${studentbooks}`,  };  /\* End - For students data extraction \*/  } |

# Eurobot Interface

|  |  |
| --- | --- |
|  |  |

Use of Watson discovery collection to map external data like the student.json and covid website <https://api.covid19api.com/summary>

|  |  |
| --- | --- |
| Watson Discovery collection mapped to external json data file containing the students data | Watson Discovery collection mapped to external website <https://api.covid19api.com/summary> |

# Benefits

* Around the clock support
  + 24 X 7 support for all school related queries; whenever, wherever.
  + Fills in for the physical staff absence, in adverse situation like covid-19 lockdown that enforced full/partial movements or travel curbs.
* Keeping up with the trend
  + Integrating the chatbot into one of the popular platforms, that the parents from all demography use daily (eg: whatsapp), is cheaper and easier than alternate helpdesk arrangement.
* Improved communication
  + Aids to improve communication and minimize ambiguity from interactions.
* Improve staff productivity
  + Implementing a full functioning chatbot, is much cheaper and faster and time efficient, invariably spares the school staff to devote their time to other tasks if not answering parents query.

# Roadmap

September -2020

October -2020

6th- June-2020

August -2020

# References

Following artefacts uploaded to github repository <https://github.com/roshnihirlekar/withackathon_june2020>

* Watson assistant - skill-Eurobot\_skill.json
* Watson Discovery (students data) – student.json
* Function code (web hook api) - schoolnodejs\_code.js
* Video - YouTube Link to the Demo Video - <https://youtu.be/A4ddjet-O0c>
* Project Document – Team-TechApetitie\_Solution-Eurobot.docx

Happy Coding