**Early Prediction of Diabetes Mellitus in Pregnant Women Using Machine Learning**

**Done by**

**Ms. M.K. Dharani**

[**dharani.cse@kongu.ac.in**](mailto:dharani.cse@kongu.ac.in)**,** [**mkdharanishka@gmail.com**](mailto:mkdharanishka@gmail.com)

**Assistant Professor**

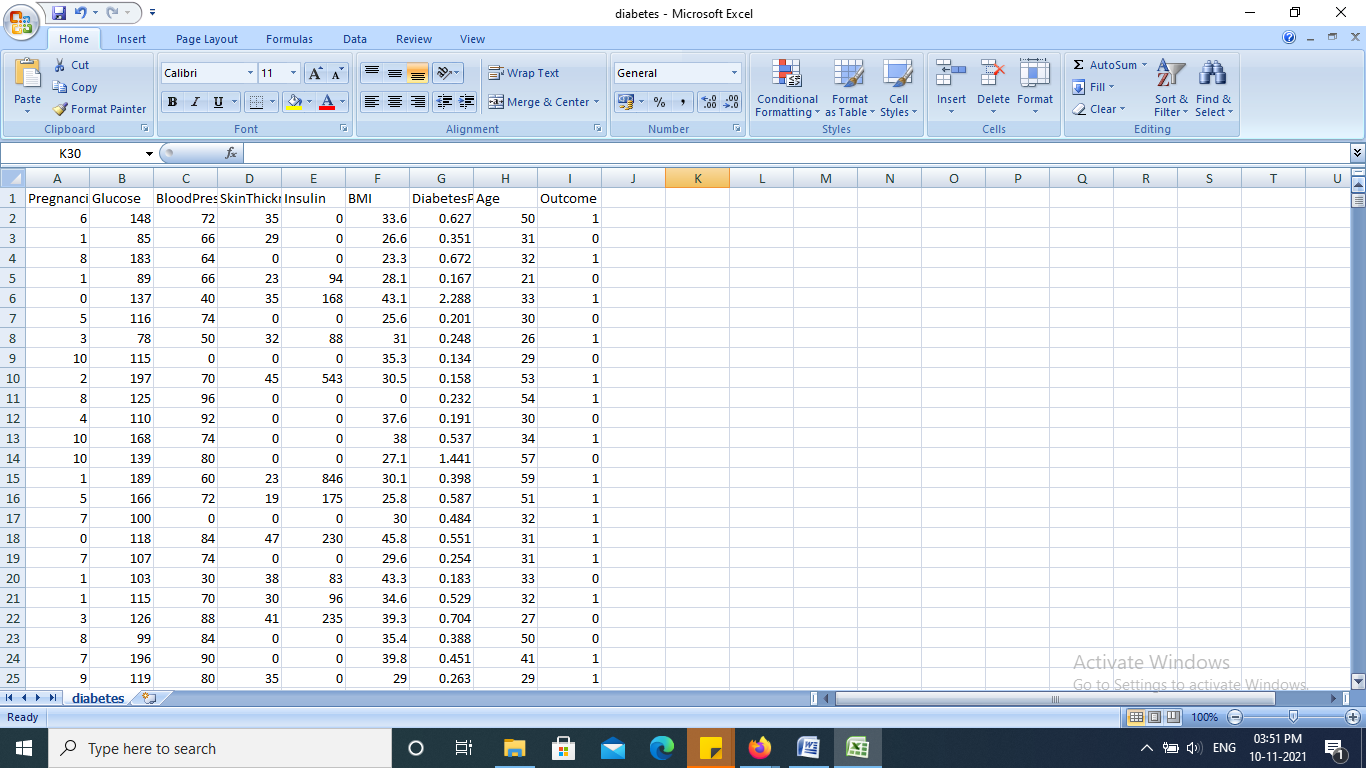
**Kongu Engineering College, Perundurai, Erode, TamilNAdu**

**Deployment of Model in Watson Using Auto AI**

Step-1: Collect the data set

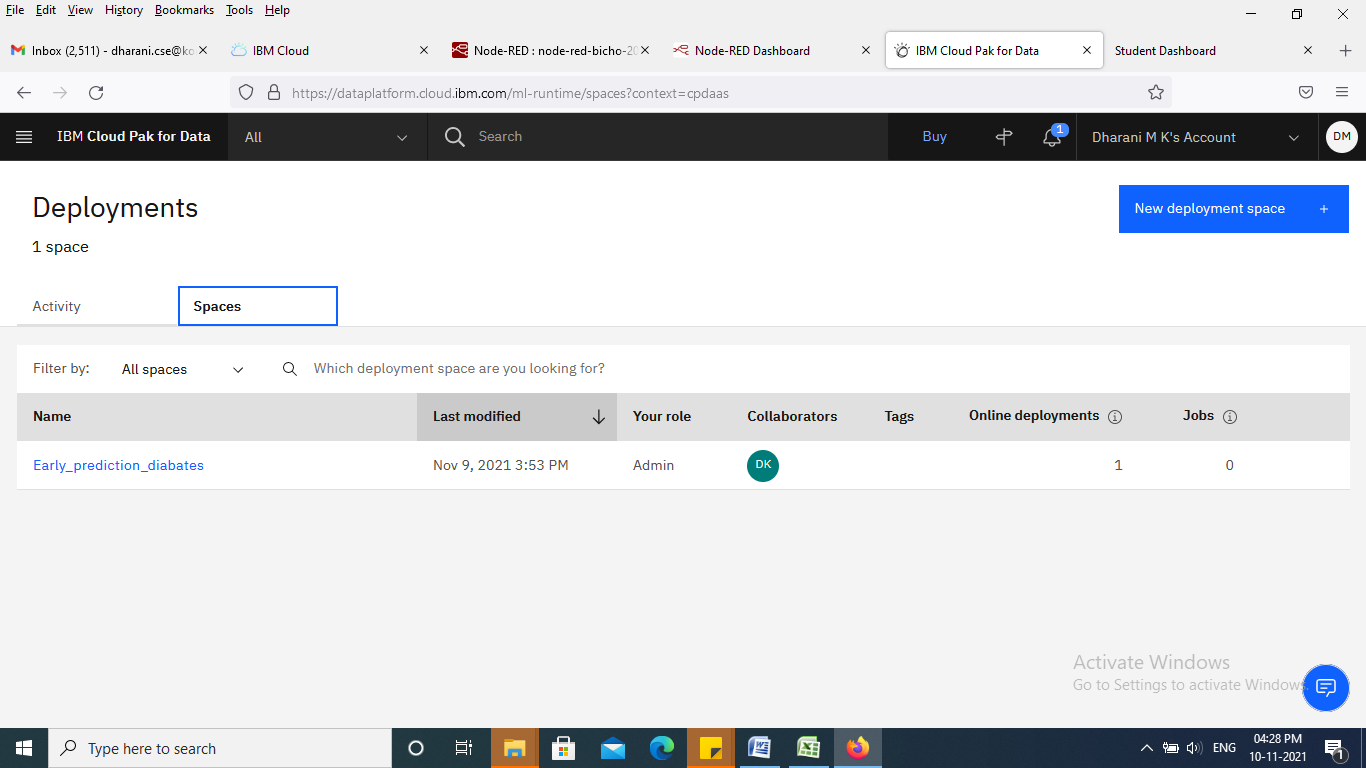
Dataset Description:

1. Pregnancies
2. Glucose
3. Blood Pressure
4. Skin Thickness
5. Insulin
6. BMI
7. Diabetes Prediction Function
8. Age
9. Outcome



Step -2: Create Watson Studio project

Separate space was created to deploy the project.



Step-3: Add Auto AI experiment

Select Auto AI in Watson cloud environment

Associate Machine learning in Auto AI

Upload the dataset in the Watson cloud space

Load your dataset and select the input variables and output variable.

Based on which it will show the given dataset is for Classification.

Where we can select the algorithms available

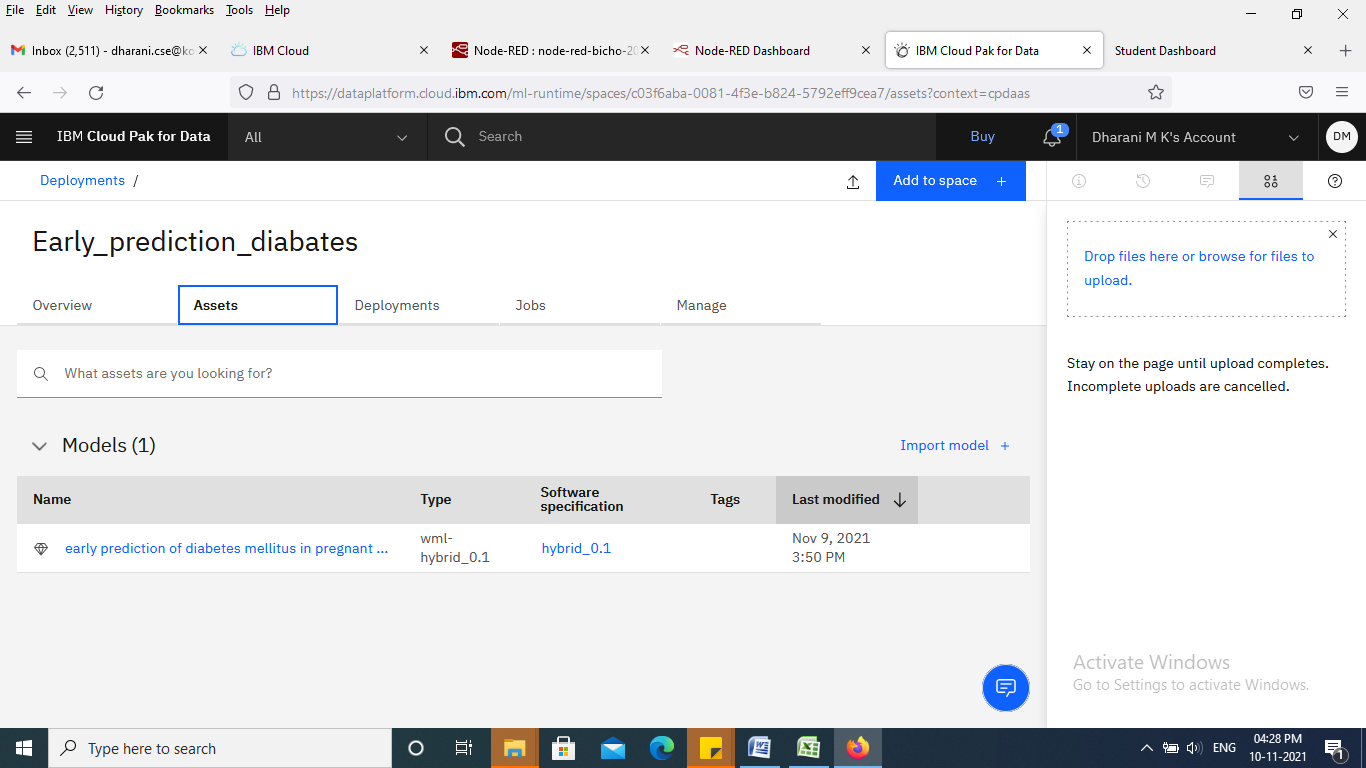
Step-4: Run AI Experiment

After choosing the algorithm run the experiment in which Auto AI selects the best algorithm by training various algorithm.

Best algorithm will be at the top rated with star.

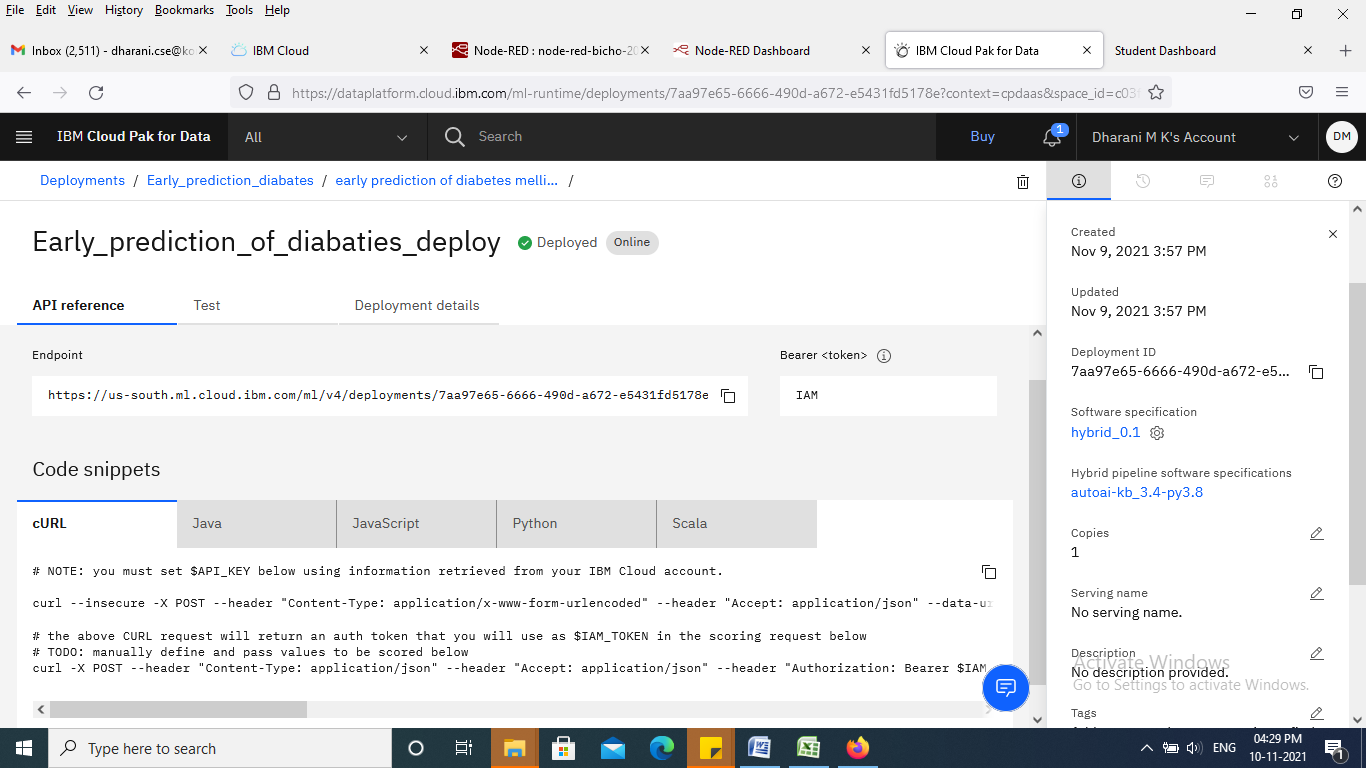
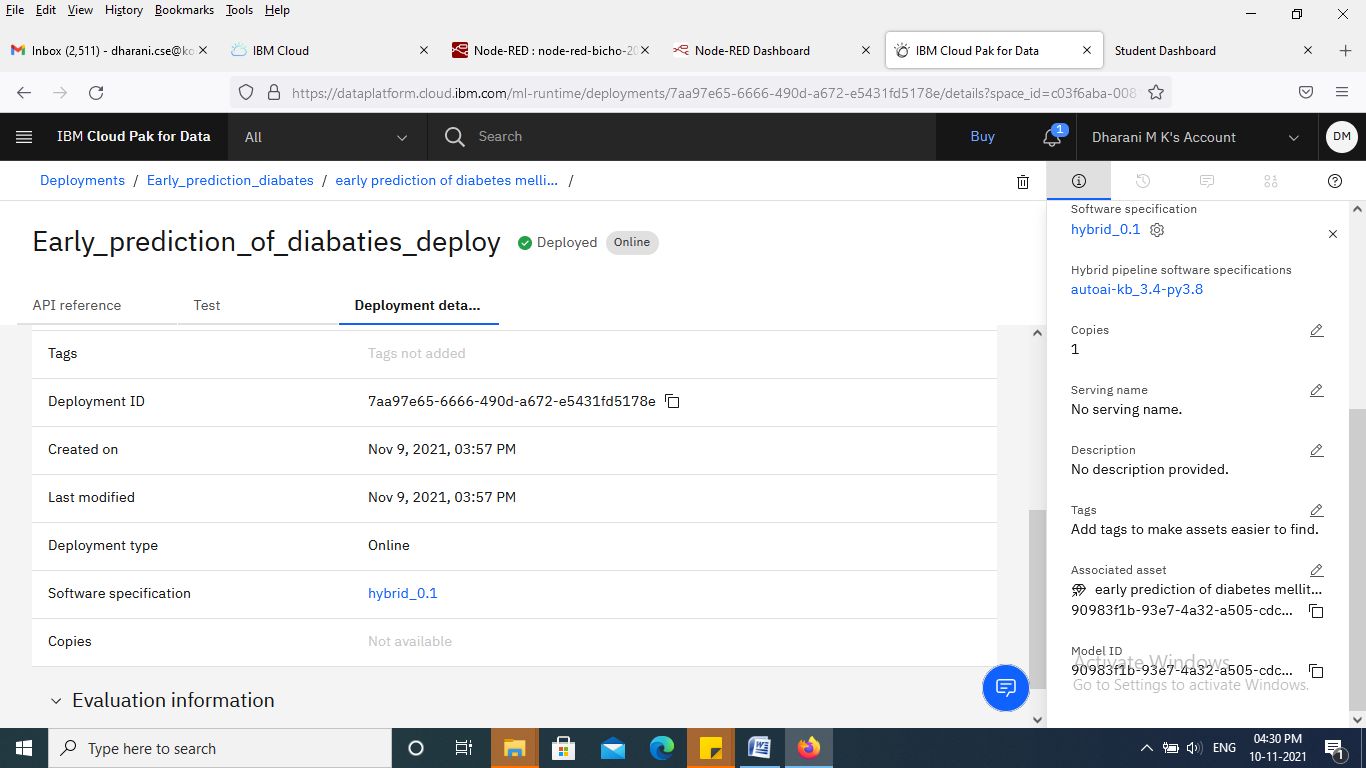
Step-5: Save the model

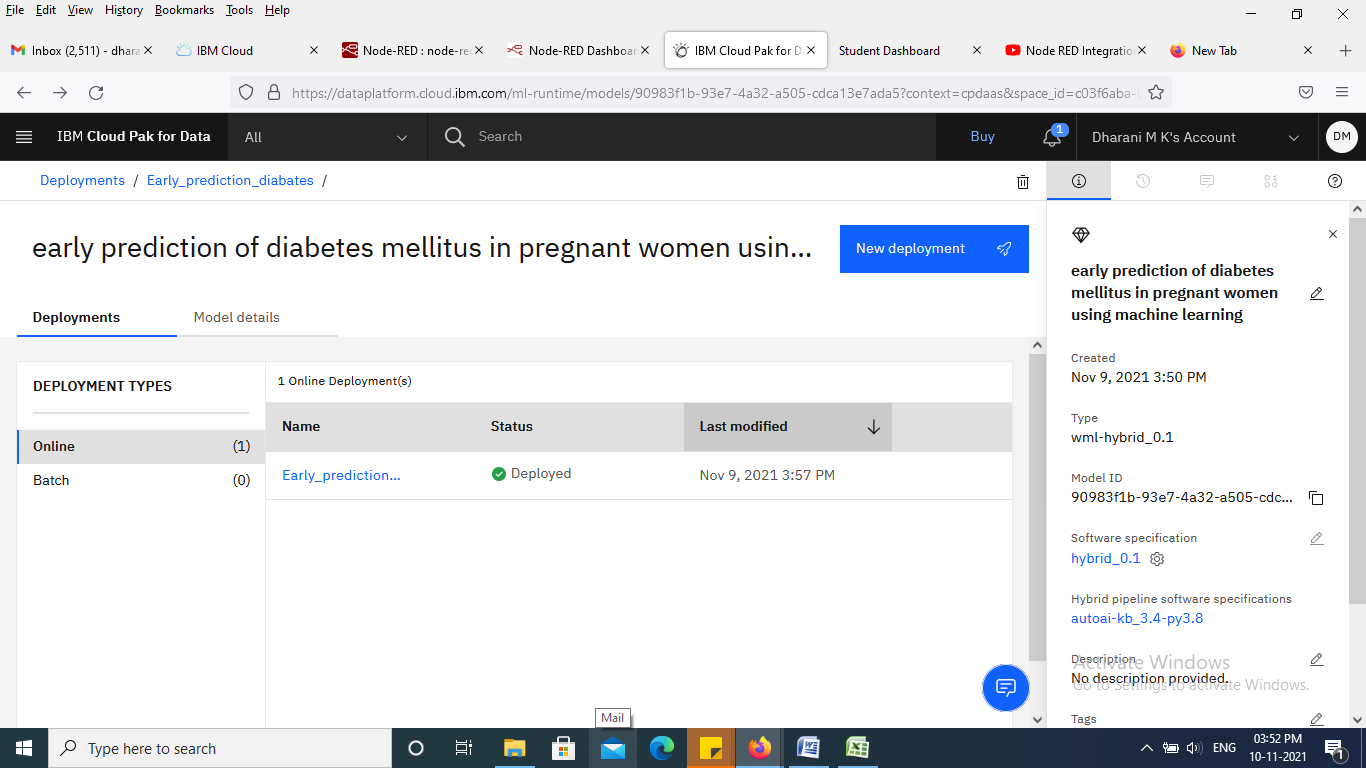
Select the best model and click save it would be saved in a cloud space.



Step-6: Deploy the model

Click on deploy option to deploy your model.

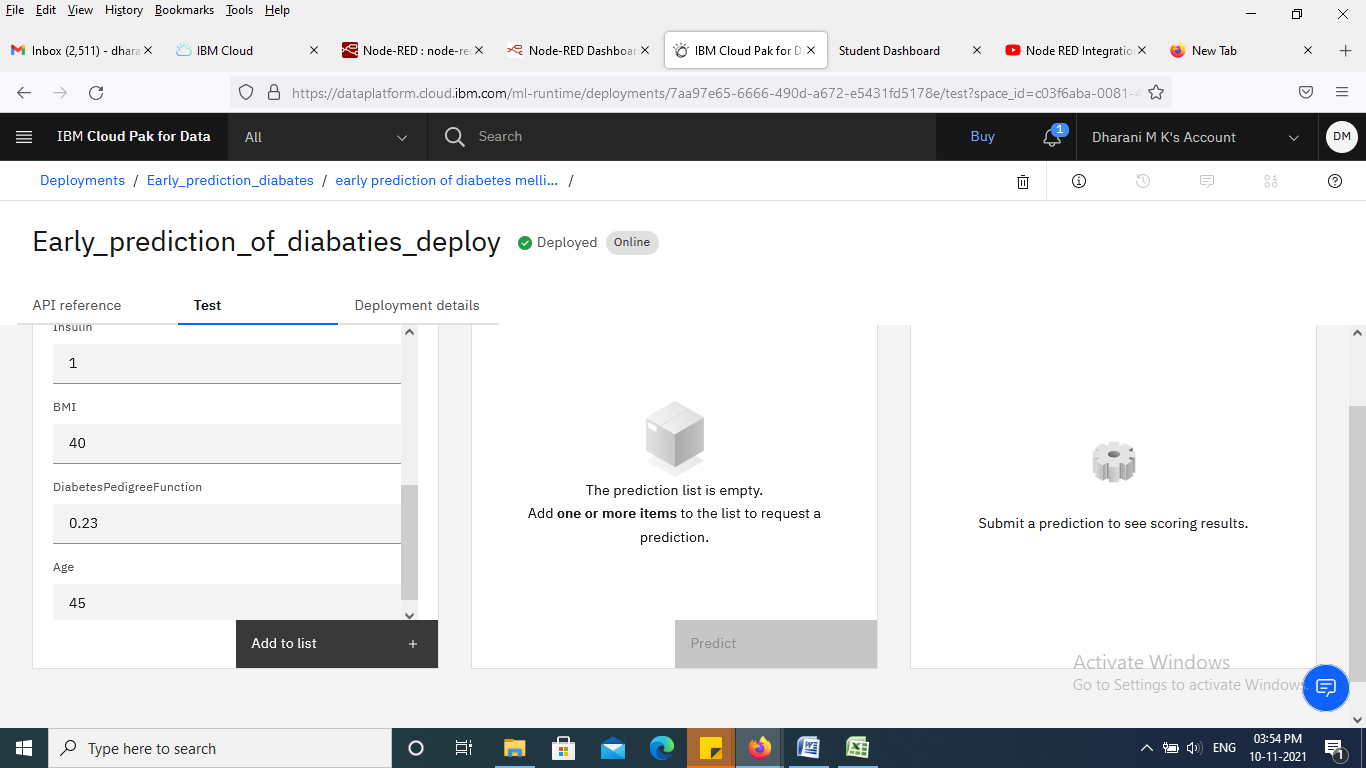
* 
* 



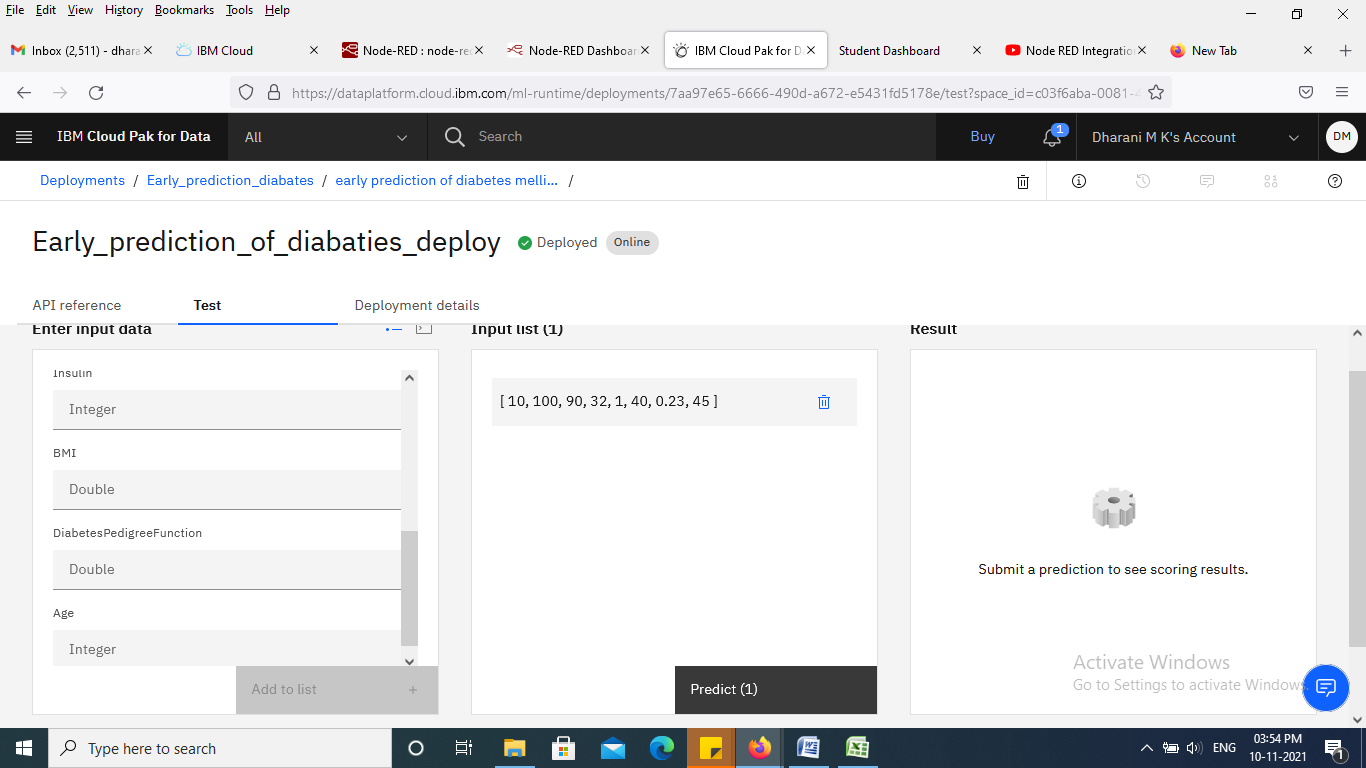
Test your Model:

Test your model by giving inputs and check the obtained output.

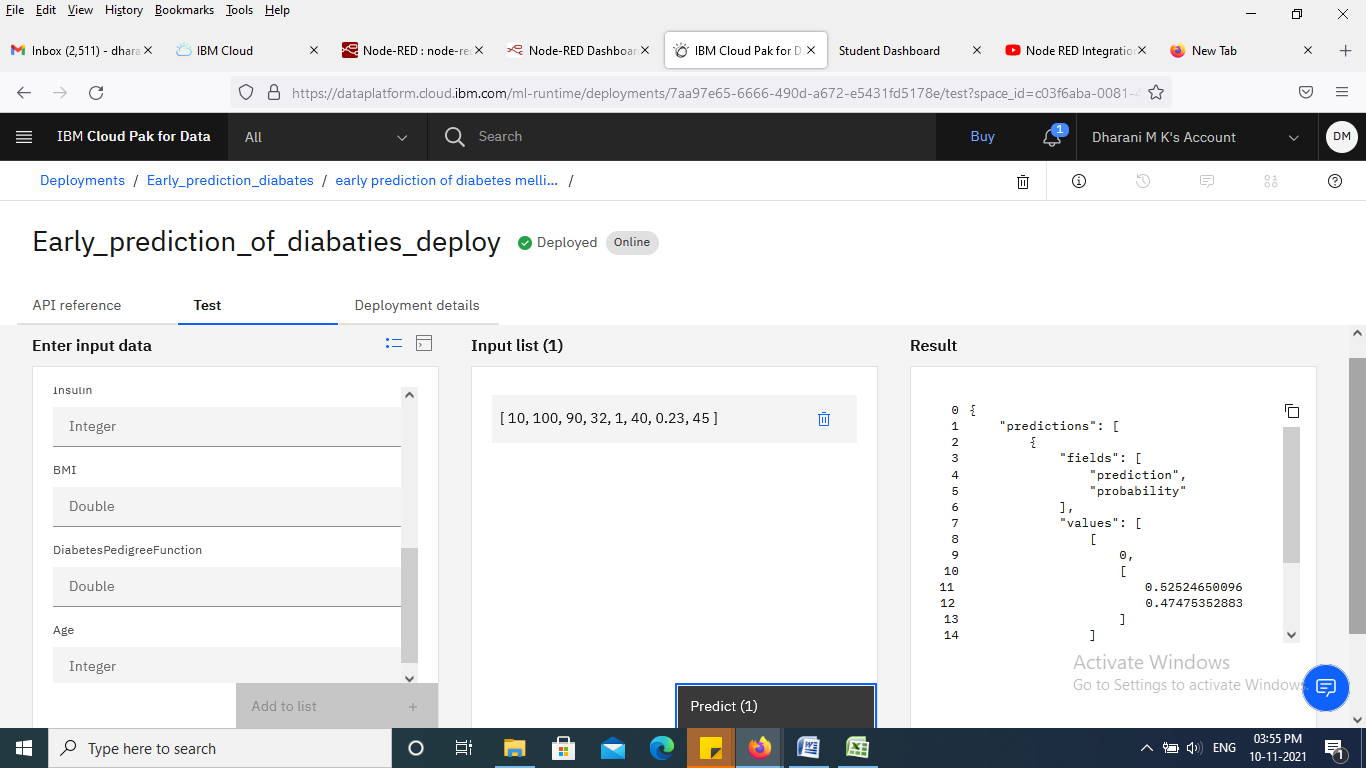
Loading inputs:



Verify the input by prediction

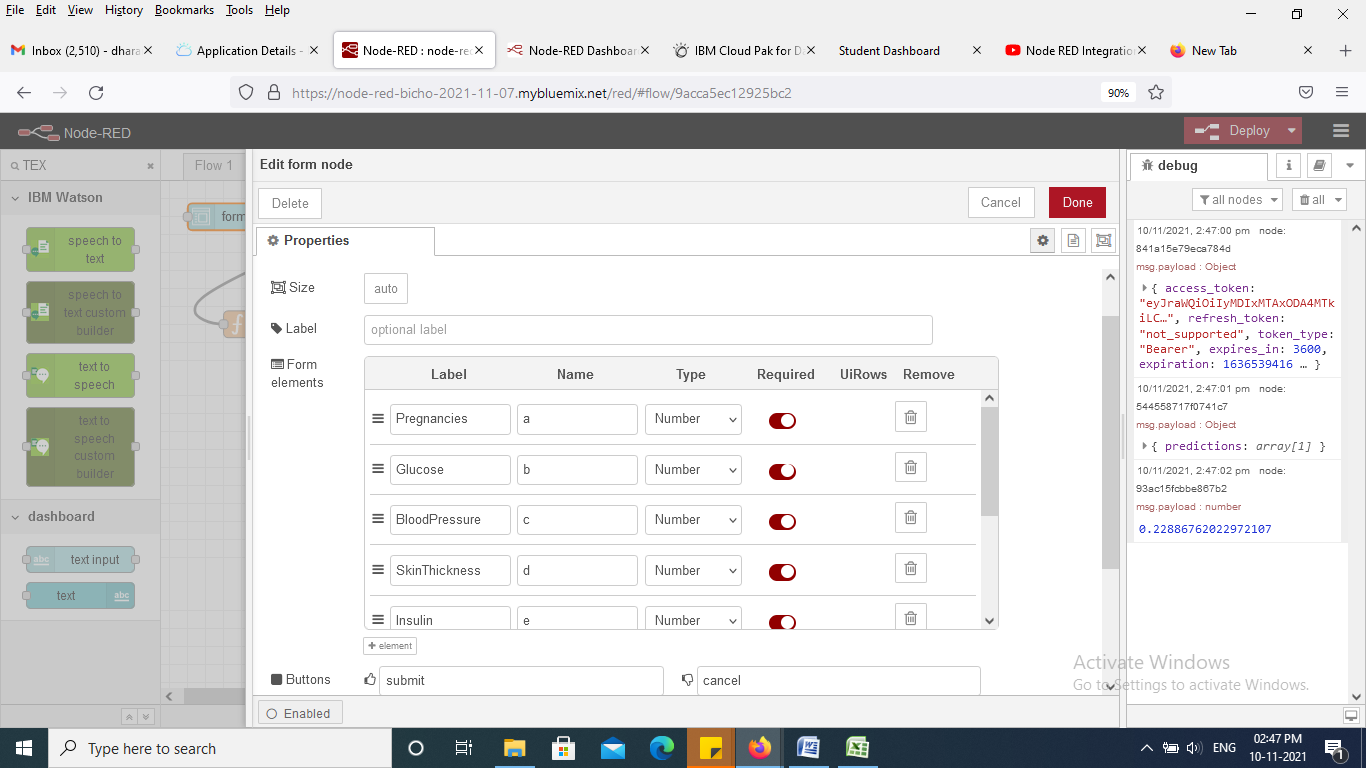


Output



**Create UI**

1. Deploy the app in Nodered and install dashboard in Nodered.
2. Select form and feed all the input labels in it



1. Grab the values from UI
2. Set the Global variables for UI
3. Define the global variables for all the input variables

**Code:**

global.set("Pregnancies",msg.payload.a)

global.set("Glucose",msg.payload.b)

global.set("BloodPressure",msg.payload.c)

global.set("SkinThickness",msg.payload.d)

global.set("Insulin",msg.payload.e)

global.set("BMI",msg.payload.f)

global.set("DiabetesPedigreeFunction",msg.payload.g)

global.set("Age",msg.payload.h)

var apikey="YXOz-Z1Sj9LnDZbMHV-1JyXlo2HV\_PiS\_svs4As1Dipb"

msg.headers={"content-type":"application/x-www-form-urlencoded"}

msg.payload={"grant\_type":"urn:ibm:params:oauth:grant-type:apikey","apikey":apikey}

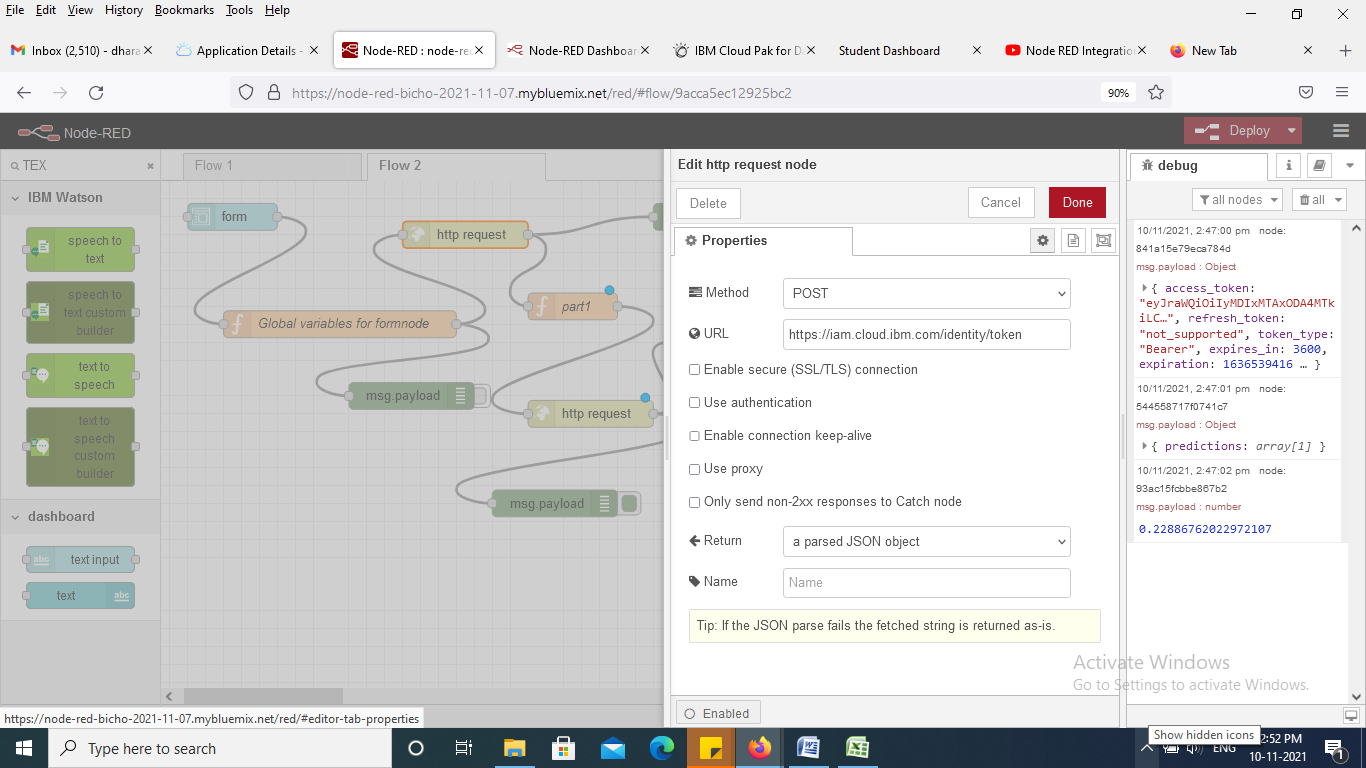
return msg;

1. Get access token with the help of HTTP request(Getting an OTP)

In the http request:

* select the method as POST
* url as <https://iam.cloud.com/identiy/token>

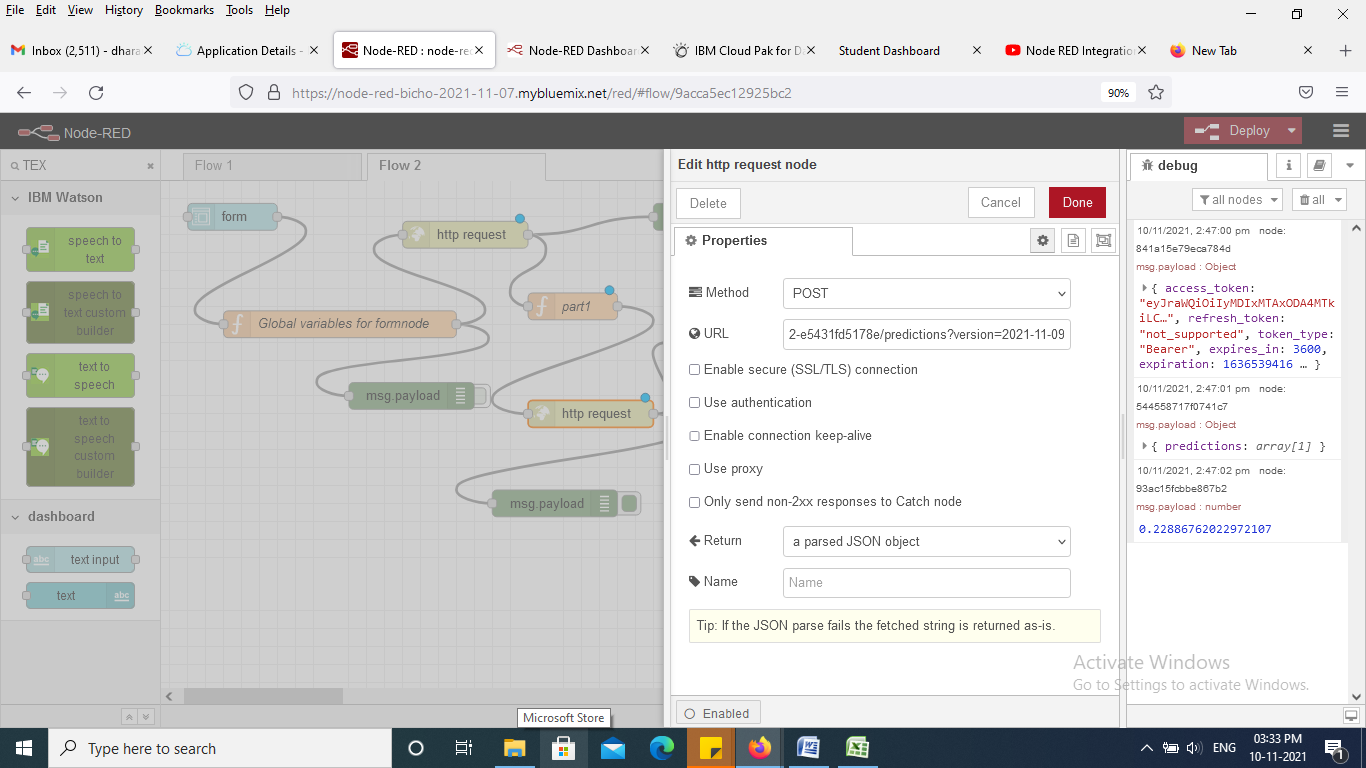
Generated Token: {"access\_token":"eyJraWQiOiIyMDIxMTAxODA4MTkiLCJhbGciOiJSUzI1NiJ9....","refresh\_token":"not\_supported","token\_type":"Bearer","expires\_in":3600,"expiration":1636539416,"scope":"ibm openid"}



1. Send the input values to scoring endpoint along with access token using http request

Link the node red with automated AI node created for Early diabeties prediction using the url:

<https://us-south.ml.cloud.ibm.com/ml/v4/deployments/7aa97e65-6666-490d-a672-e5431fd5178e/predictions?version=2021-11-09>



**Code:**

var Pregnancies1=global.set("Pregnancies")

var Glucose1 = global.set("Glucose")

var BloodPressure1 = global.set("BloodPressure")

var SkinThickness1= global.set("SkinThickness")

var Insulin1 = global.set("Insulin")

var BMI1= global.set("BMI")

var DiabetesPedigreeFunction1= global.set("DiabetesPedigreeFunction")

var Age1=global.set("Age")

var token=msg.payload.access\_token

msg.headers={'content-type':'application/json',"Authorization":"Bearer"+token,"Accept":'application/json'}

msg.payload={

"input\_data": [{"fields": ["Pregnancies","Glucose","BloodPressure","SkinThickness",

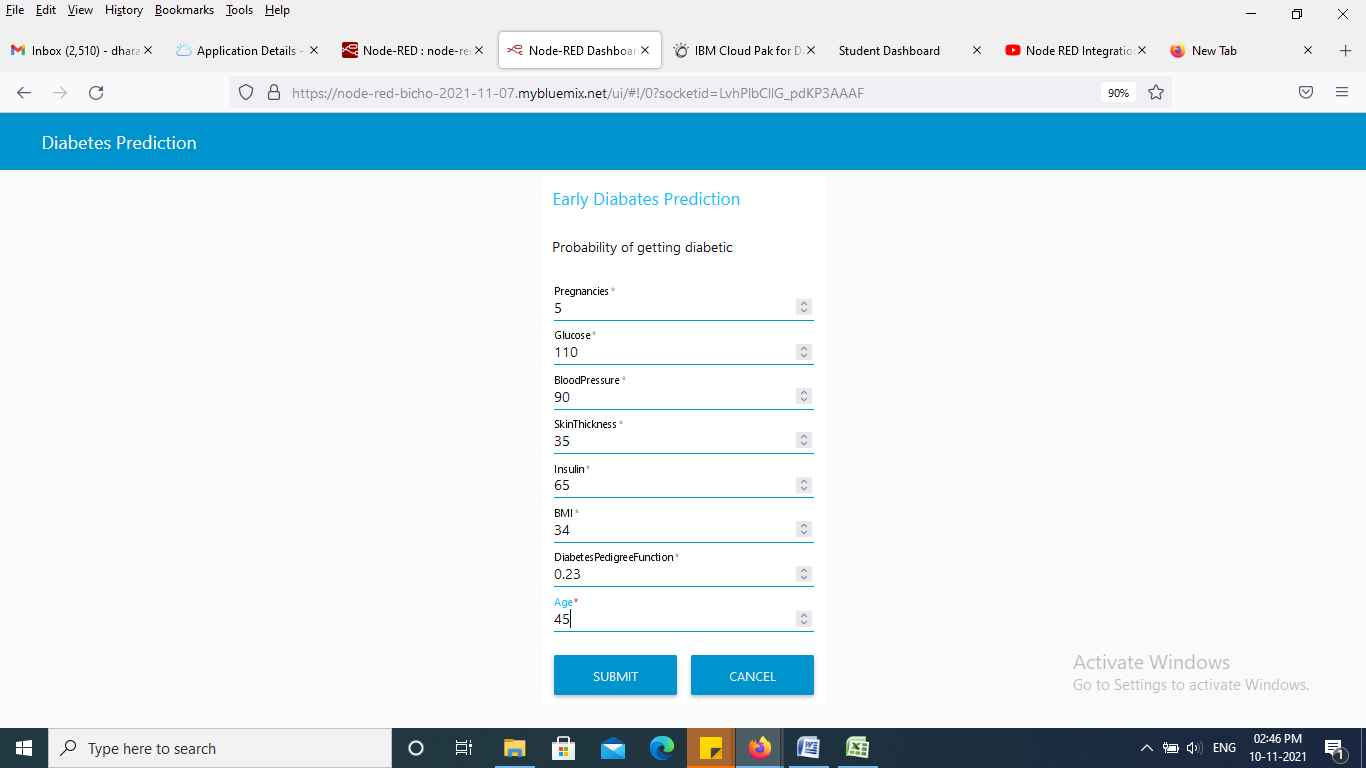
"Insulin","BMI","DiabetesPedigreeFunction","Age" ],"values": [

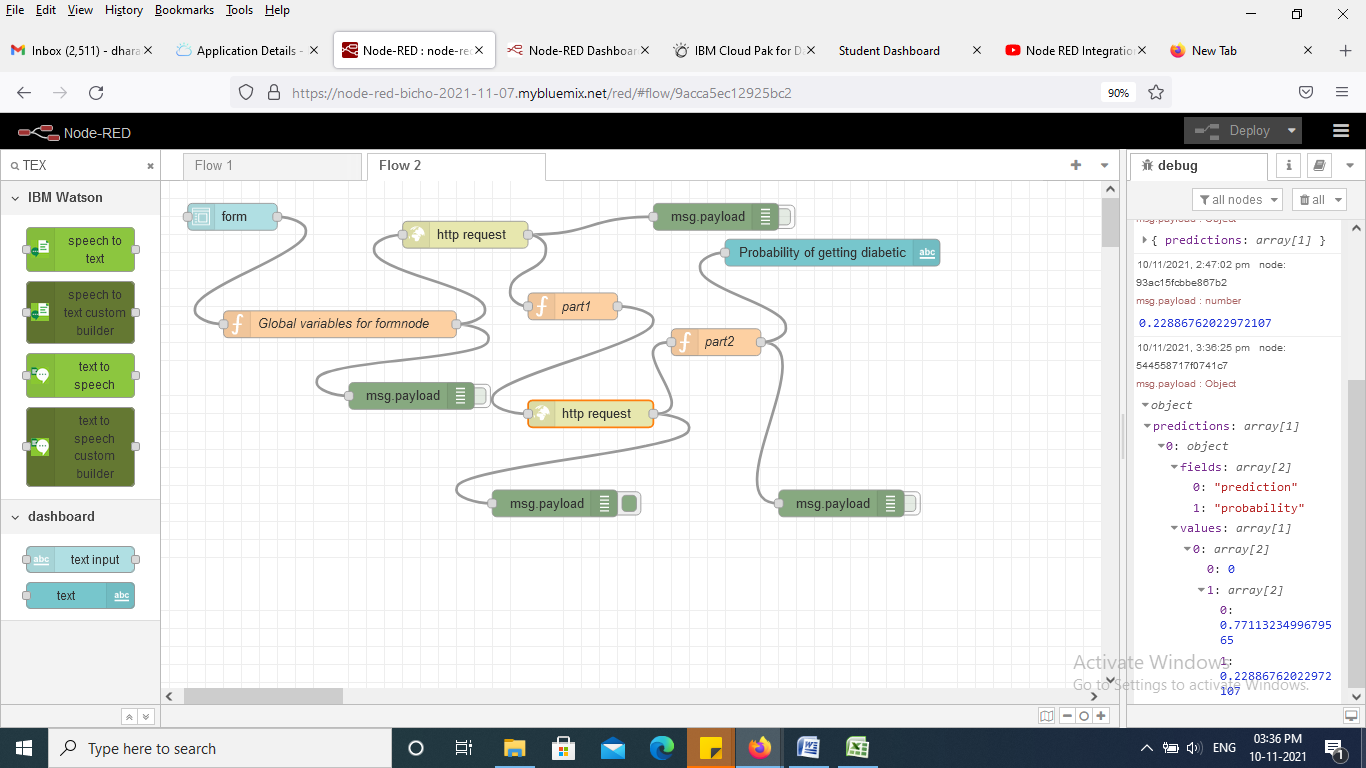
[Pregnancies1,Glucose1,BloodPressure1,SkinThickness1,Insulin1,BMI1,DiabetesPedigreeFunction1,Age1]]}]}

return msg;

1. Get the predicted output

Predicted output:



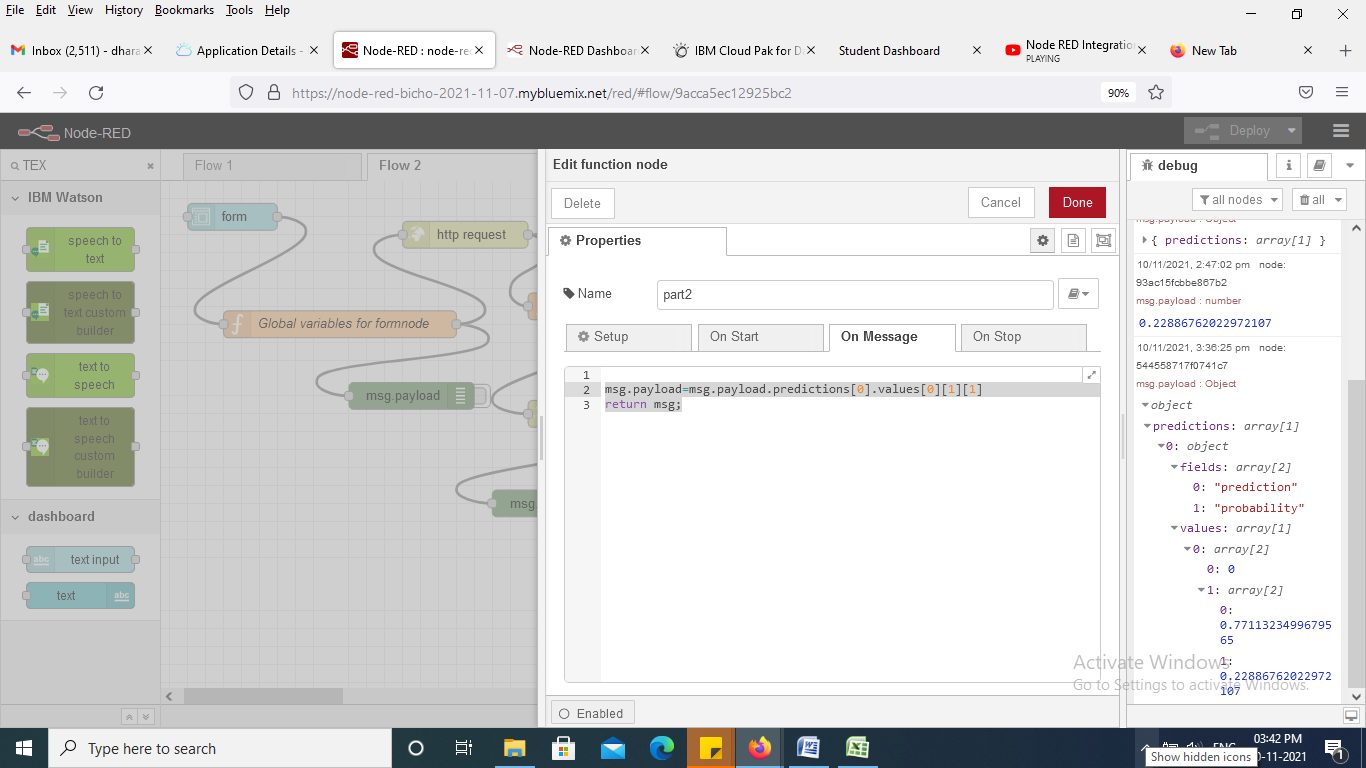


1. Parse the output

**Code:**

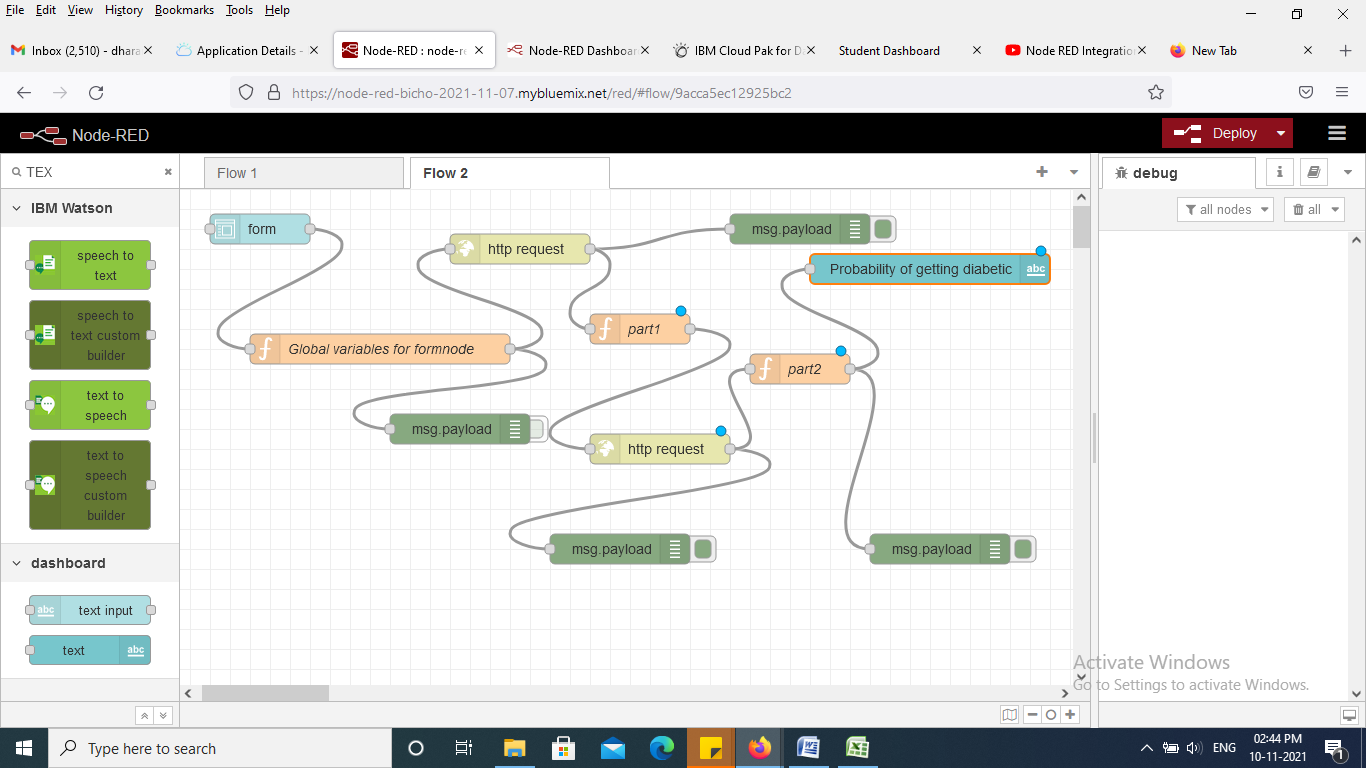
msg.payload=msg.payload.predictions[0].values[0][1][1]

return msg;

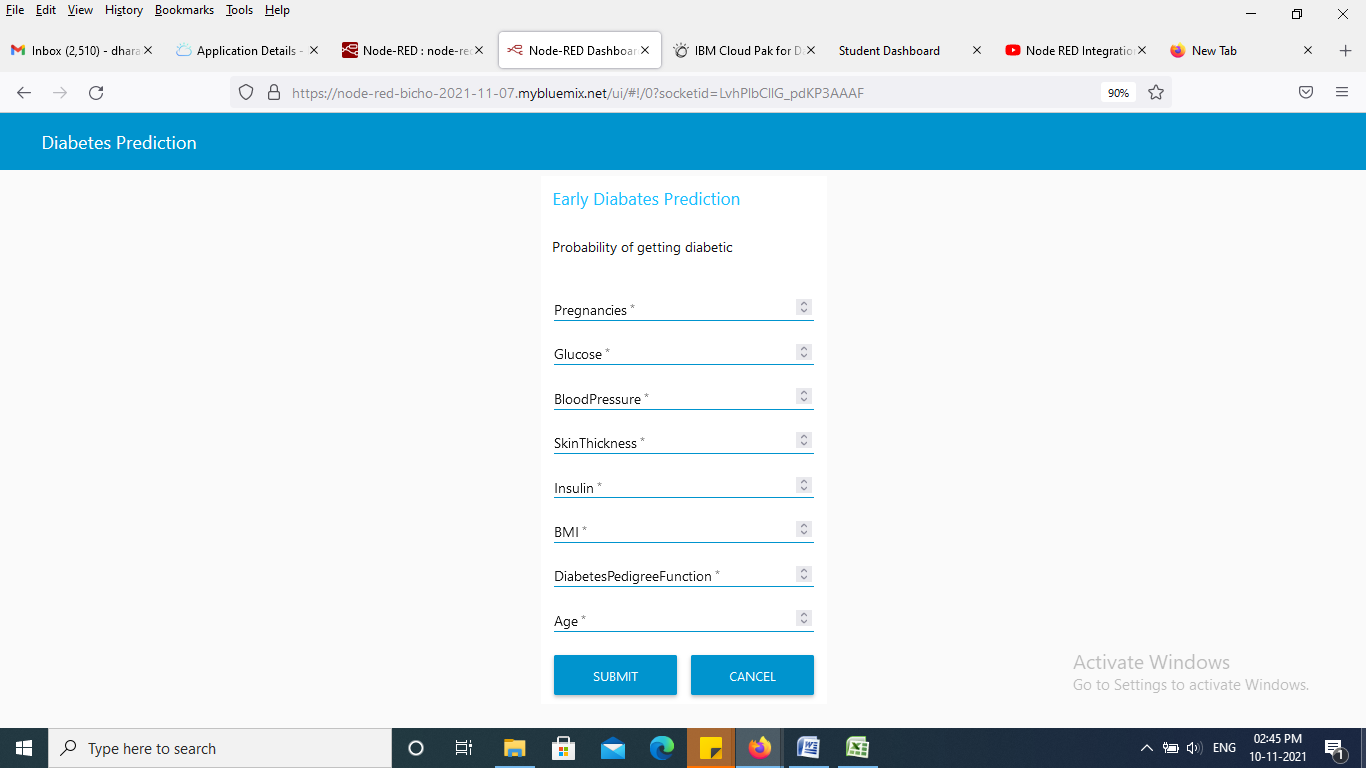


1. Showcase the output on UI

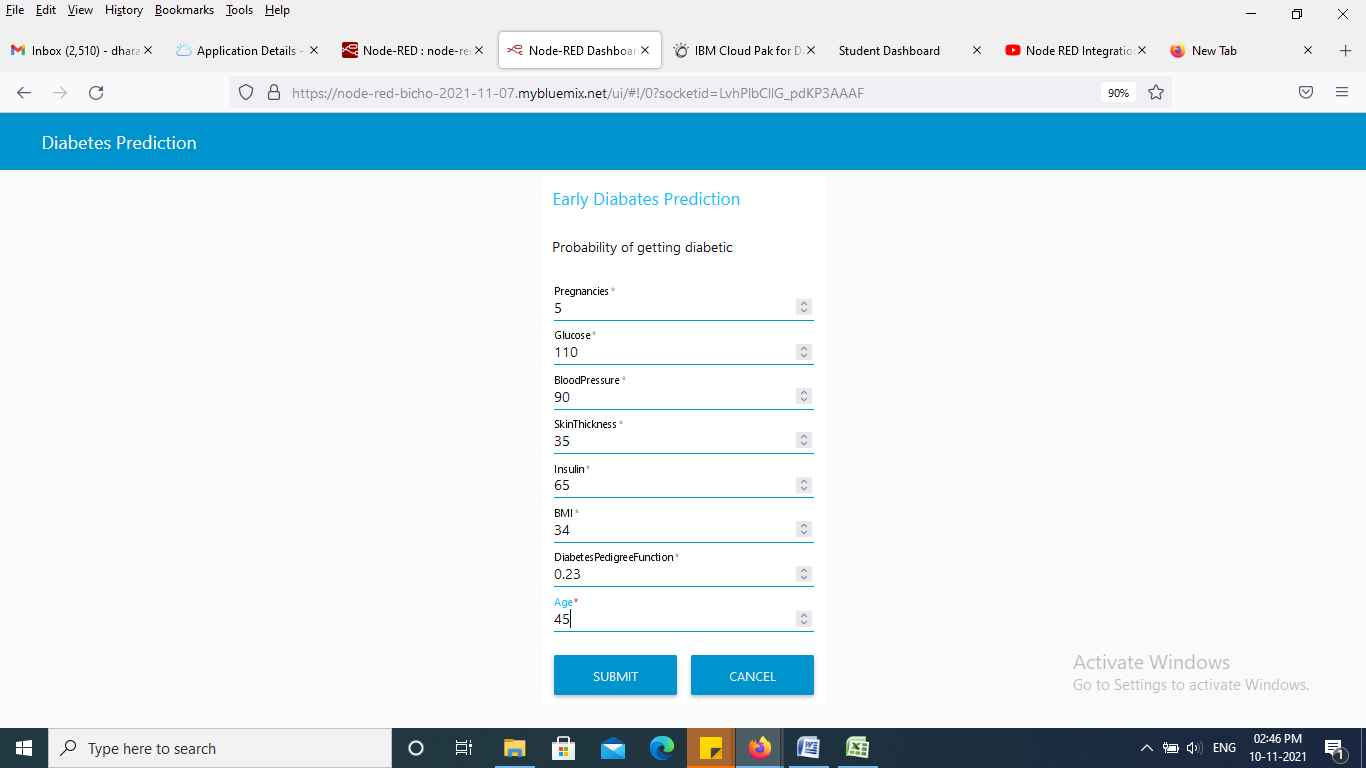
Complete UI Design



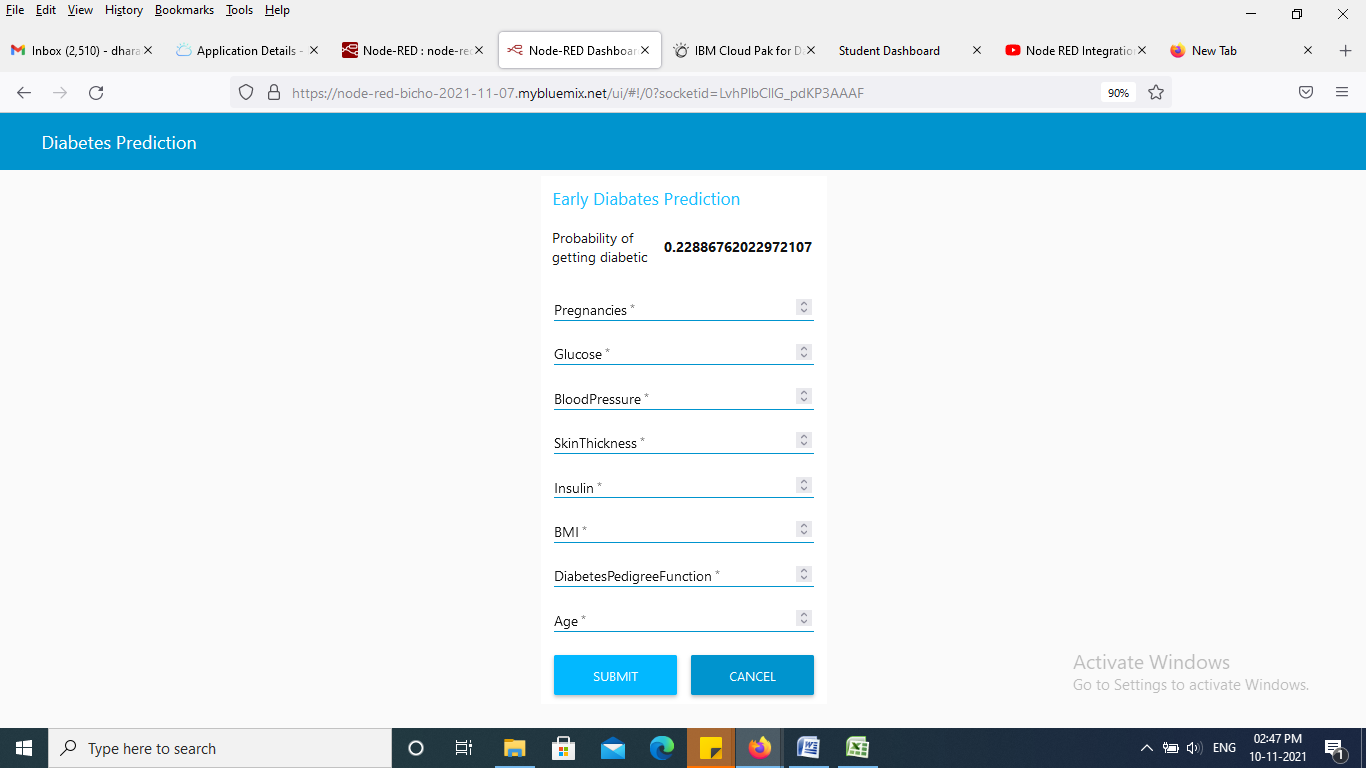
Input Screen



Feeding Input through Nodered UI



Predicted Output:



**Demo Video & Feedback:**

[**https://drive.google.com/file/d/1iw9Cp6qJdquQmR5USqPv7kGR0WsUlLMu/view?usp=sharing**](https://drive.google.com/file/d/1iw9Cp6qJdquQmR5USqPv7kGR0WsUlLMu/view?usp=sharing)