AI Assisted Farming for Crop Recommendation & Farm Yield Prediction Application

TEAM NAME: ELECTRIC GUEST
WRITTEN BY: ROHIT D

1. INTRODUCTION

1.1 Overview

This project is a culmination of integration of various IBM services such as IBM Watson Studio, Cloud Foundry Apps and IOT Platform Service. The Watson Studio was used to create an Auto AI model which can be used to predict the optimal crop. Cloud Foundry Apps was used to host the Django Website. The IOT Platform Service was used to collect soil conditions using Sensor Data and feed it to the Dashboard and ML model to predict the optimal crop.

1.2 Purpose

To build an application that can be used by farmers throughout India to get accurate information on which crops to be planted. The crop recommendation will be based on various parameters such as nutrient levels of soil, rainfall in the region, temperature and other climate conditions, which when fed to an ML Model, gives an accurate output. By using various IBM services, we can help farmers get the best yield and revenue from their hard-work and land.

2. LITERATURE SURVEY

2.1 Existing Problem

Farming is a profession, which involves dealing with life and nature. Nature as we know is very unpredictable. Thus, farmers are at the mercy of nature. The risk factor is very high as it might rain properly year, or it might be a drought or a flood. They can put in as much hard work as they can and still not reap the benefits of their hard work. Thus, their hard work can go to waste if nature decides otherwise.
Now, the population of the earth is 7.2 Billion. In a few decades, it is going to touch 10 Billion. In order to feed 10,000,000,000 people, farming has to modernize. Traditional methods is not going to cut it anymore where the crops grown in a land can be sub-optimal to those conditions. As the amount of fertility in the soil is limited, it has to be constantly monitored and maintained.

2.2 Proposed Solution

AI and IOT Solutions can be a game changer to the farming industry. These solutions can reduce the mental and physical workload farmers face, which can improve results with less effort and bring about better changes such as:

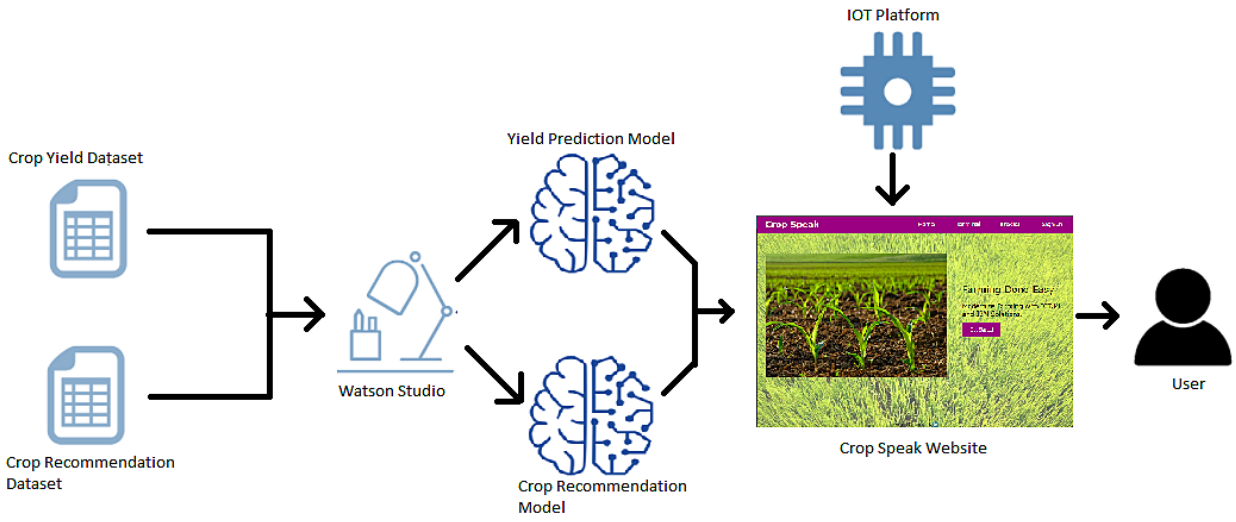
- Automate Crop harvesting and Plantation using ML Models which can give accurate timings for these tasks.
- Give Information on Revenue and Cost of Production for crops to farmers using the Dataset.
- Simplify Crop Selection for a particular land and find most profitable crop using ML Models.
- Real-Time Predictive Analysis is the next big thing in the IOT Sphere. Farmers can collect the data and determine the optimal use of fertilizers, irrigation and plantation systems. They can take action in real time with IOT Solutions.

Thus, the solution can help farmers minimize the risk in their profession and get the best results with minimal efforts. The idea can be split into 3 parts:

- Auto AI Model to give crop recommendation to farmers and give information on revenue
- IOT Device and Platform to give Real-Time Predictive Analysis
- Django Web Application as a portal for farmers to access these Services.

3. THEORETICAL ANALYSIS

3.1 Block Diagram



3.2 Hardware/Software Designing

HARDWARE DESIGN(Crop Sense IOT Device):

- N,P,K Soil Sensor
- pH, Humidity Sensor for Soil
- Temperature Sensor

SOFTWARE DESIGN(Crop Speak App):

- Auto AI Model using IBM Watson Studio.
- IOT Platform Service to connect Crop Sense Device to the Auto AI and Cloud Foundry App.
- Cloud Foundry Django App for users to see the dashboard.

4. EXPERIMENTAL INVESTIGATION

4.1 IOT Solution

In India, farms are smaller compared to the western world. Thus, the villages are small and the people are a tight knit community. Thus a solution has to be designed which can be accepted by the society as a whole. The solution should involve the community as a whole instead of target individual farmers or users. The IOT Solutions can be integrated for each and every farmer in the community so that there is a collective upbringing of farmers and development to the community.
The solution is to integrate the sensor values of the IOT devices of all users in the community and let the users in the community see the crops grown in their community. This way, users get a better idea of what crops to grow and how much fertilizer/manure has to be added by comparing their yield with their neighbors.

4.2 Watson Studio 20CUH

As Watson Studio only allows Lite users to use up to 20 capacity unit hours of CPU time per month, I had already exhausted the 20CUH in the first 3 days of testing and building the project. Thus I now have to improvise and output random crop samples in the web app as the Deployed Space does not run the model on a http request after exhausting the 20CUH.

4.3 Cloud Foundry App

Cloud Foundry Deployment is not Available to Lite Users in the App Development Portal in the IBM Services. Thus deploying using Node-red was not possible. Thus, I had to create a Django app using python and push it to the Cloud Foundry App Service via CLI interface.

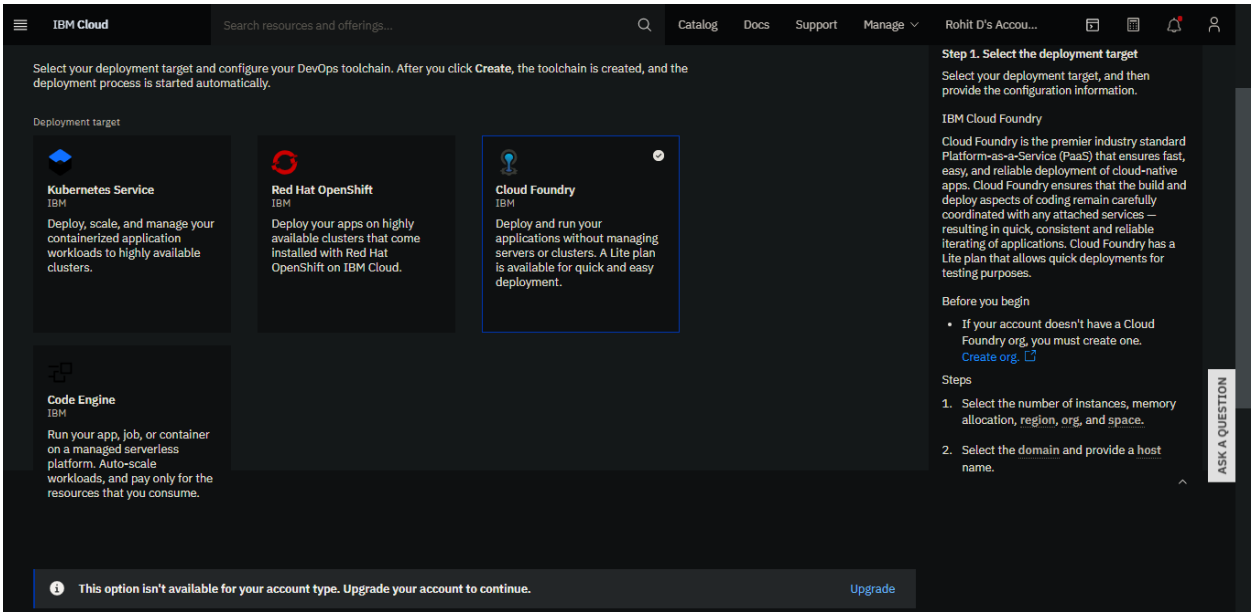
4.4 IOT Platform

Watson IOT Platform Requires Cloud Foundry Deployment Space to transmit the simulated sensor values to the Cloud. As Cloud Foundry is not Available to Lite Users, the sensor data could not be transmitted as it was not deployed. Thus, random numbers were generated in python script to simulate the IOT Sensor Values.

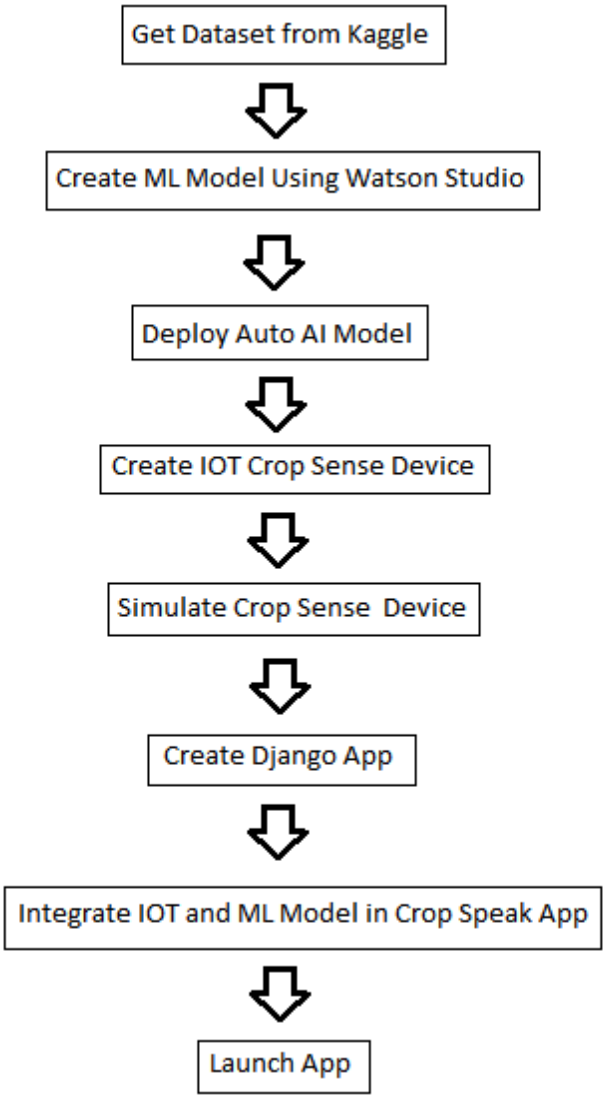
4.5 Crop Sense Device

This IOT Device requires constant internet connection to be able to provide real time analysis. This might not be possible until villages in India have constant power and internet connection. This device can also be costly for farmers who do not have enough land.

Thus, adding a backdoor where farmers can manually enter Soil parameters after getting the soil tested in a nearby government soil testing facility would be a good feature. Thus, in the account tab in the website, farmers can manually enter the Soil and Climate Parameters.



5. FLOWCHART



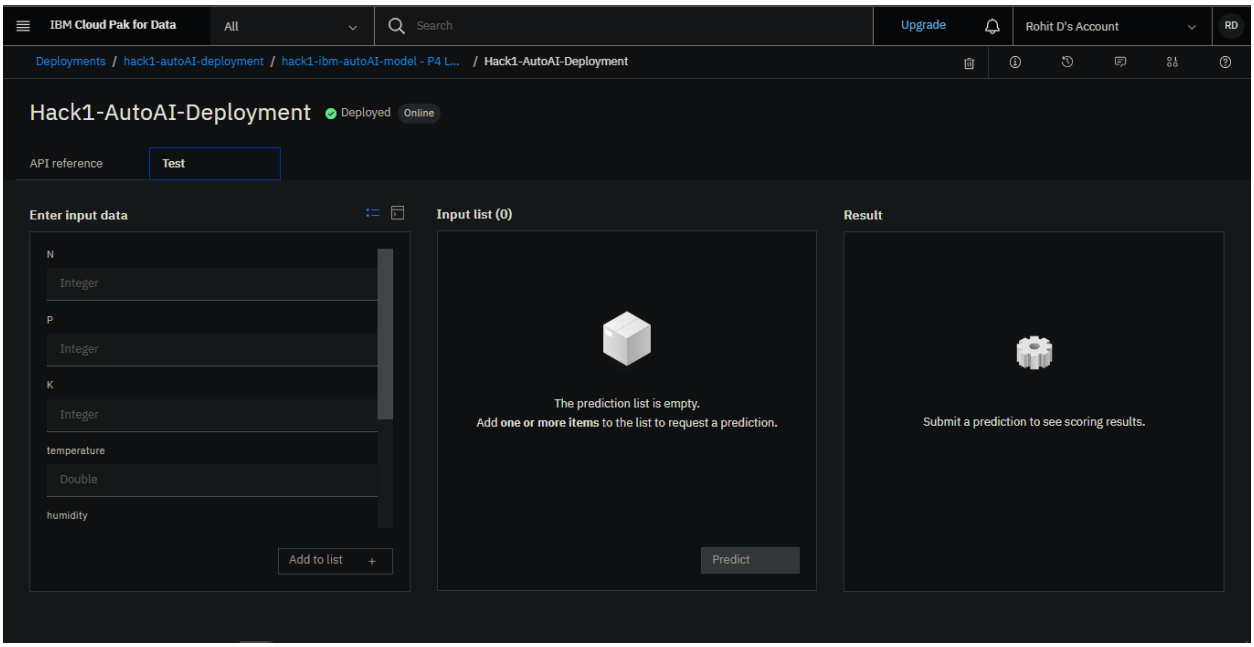
6. RESULT

6.1 Results Achieved

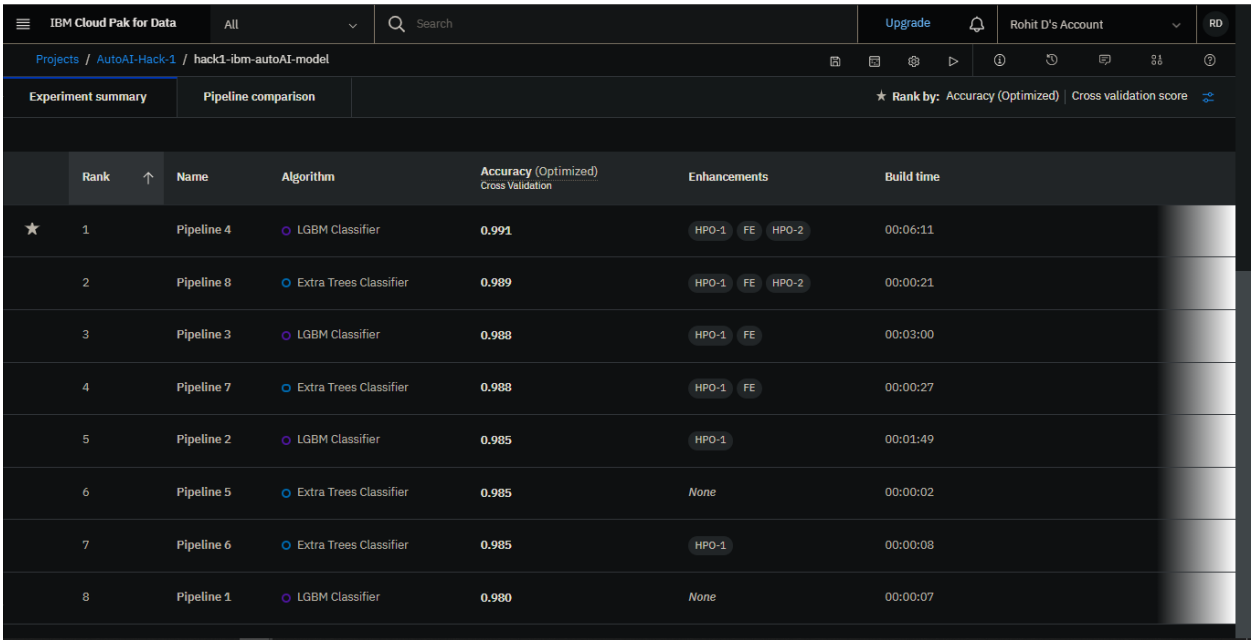
- Auto AI Model successfully built using the datasets with an accuracy of 99.1%.
- Django Application launched using Cloud Foundry Apps
- IOT Platform Service integrated with Cloud and Website

6.2 ScreenShots

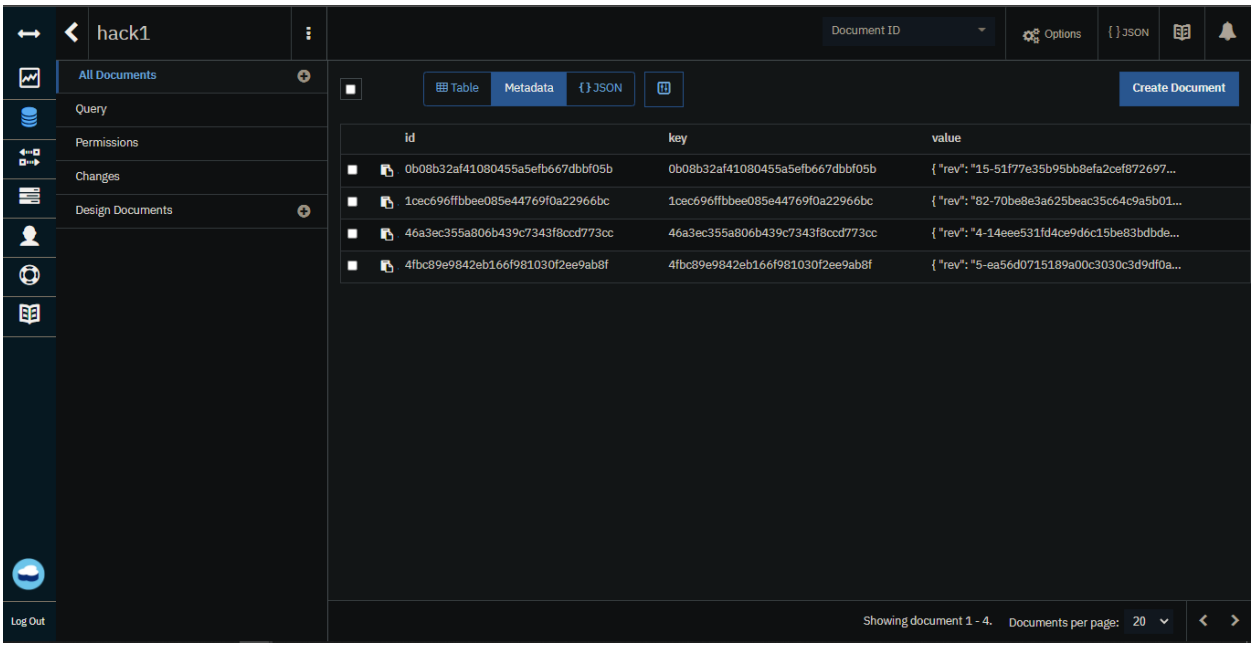
DEPLOYMENT SPACES



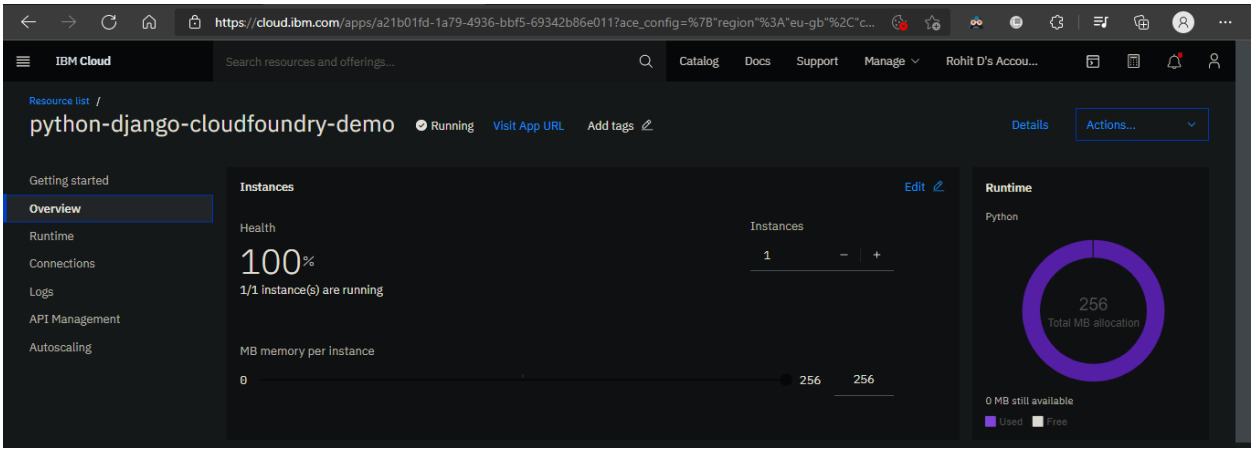
1. **Watson Studio Deployment Space**



2. **LBGM Classifier Chosen as Model to Deploy**

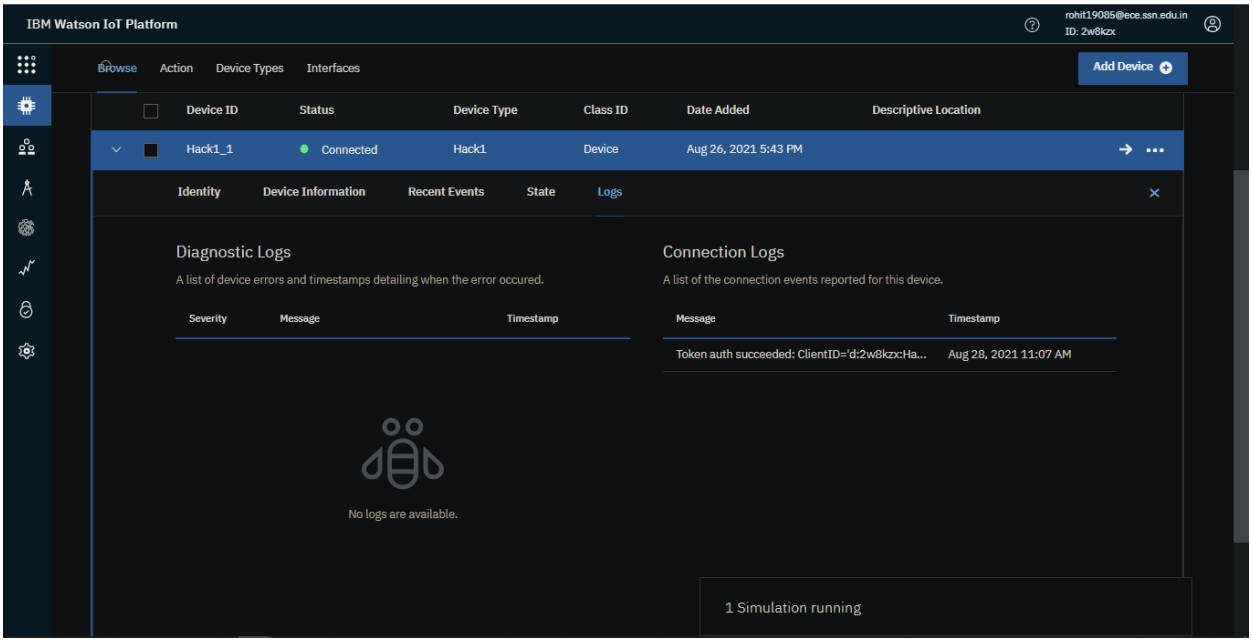


3. Cloudant Database Dashboard

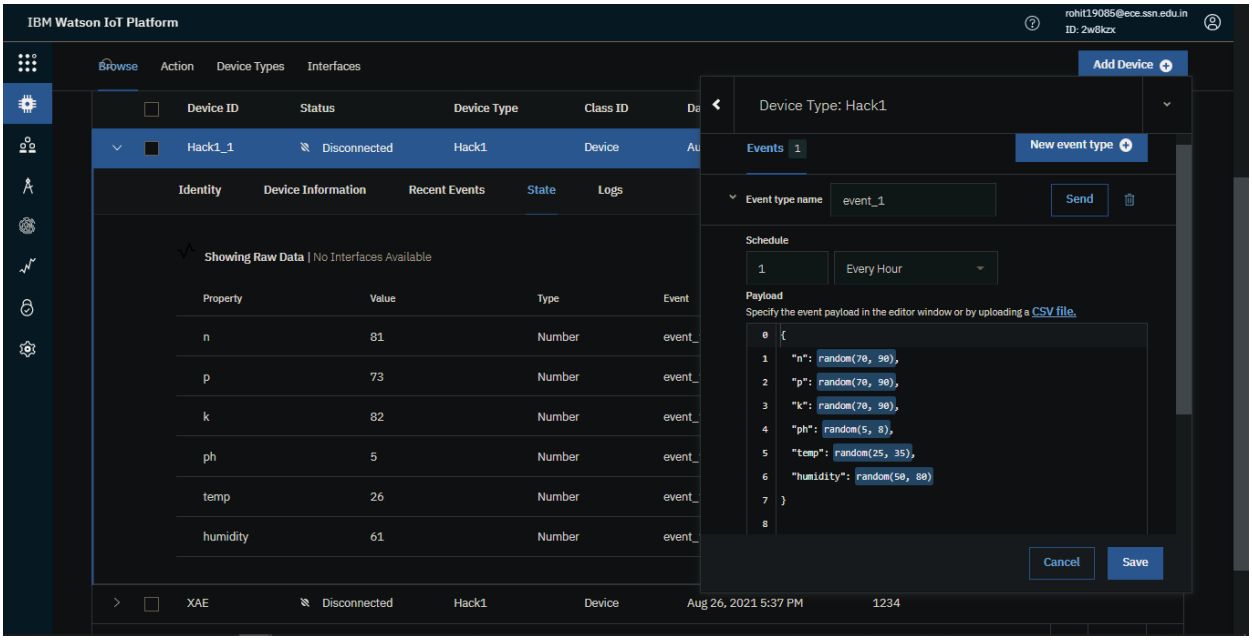


4. Django App Deployment Space

WATSON IOT PLATFORM

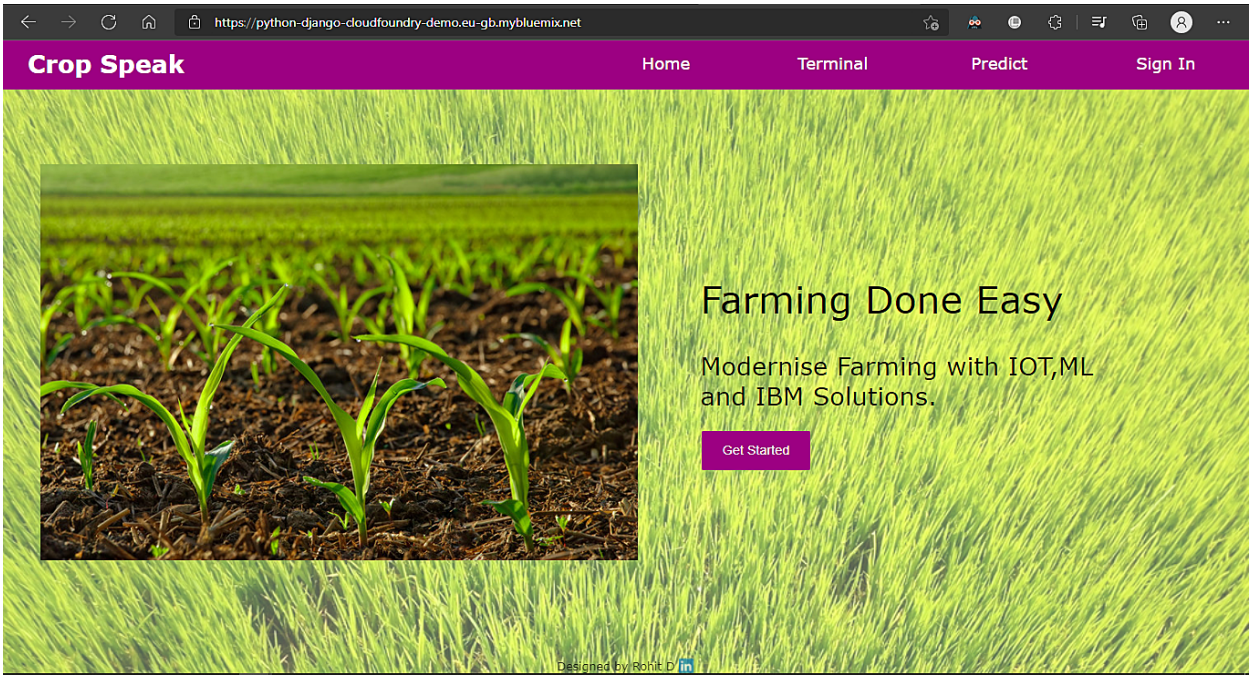


5. Crop Sense Device

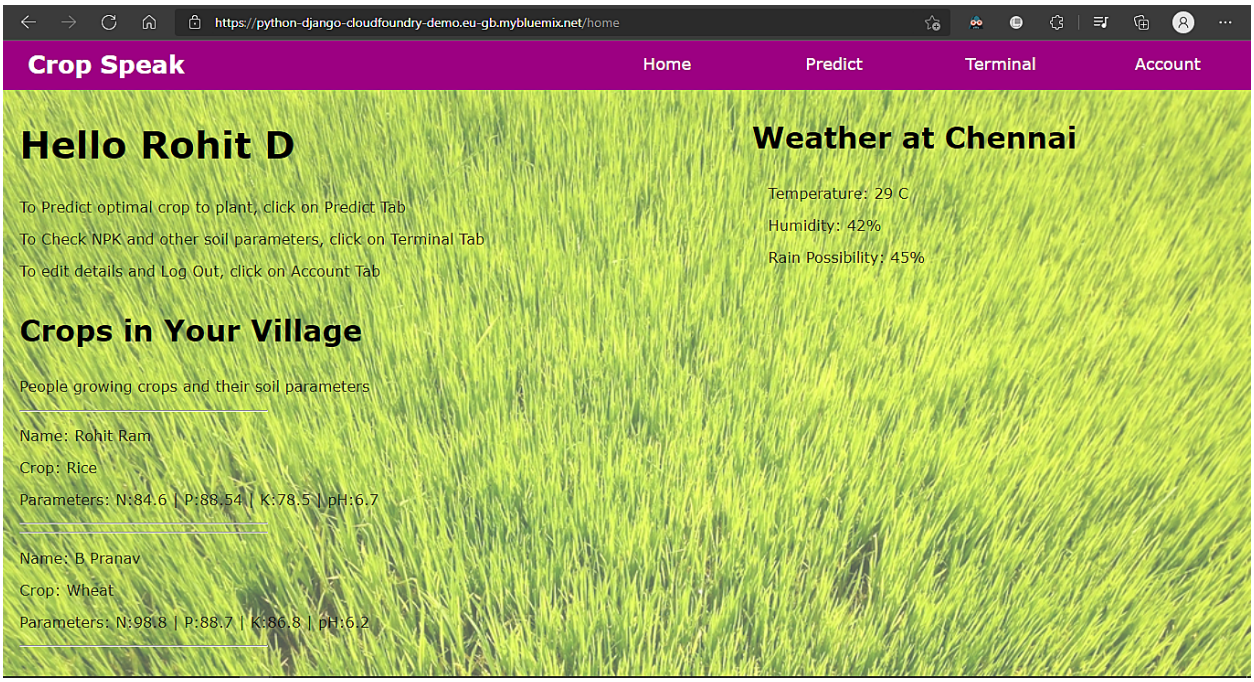


6. Crop Sense Device Settings

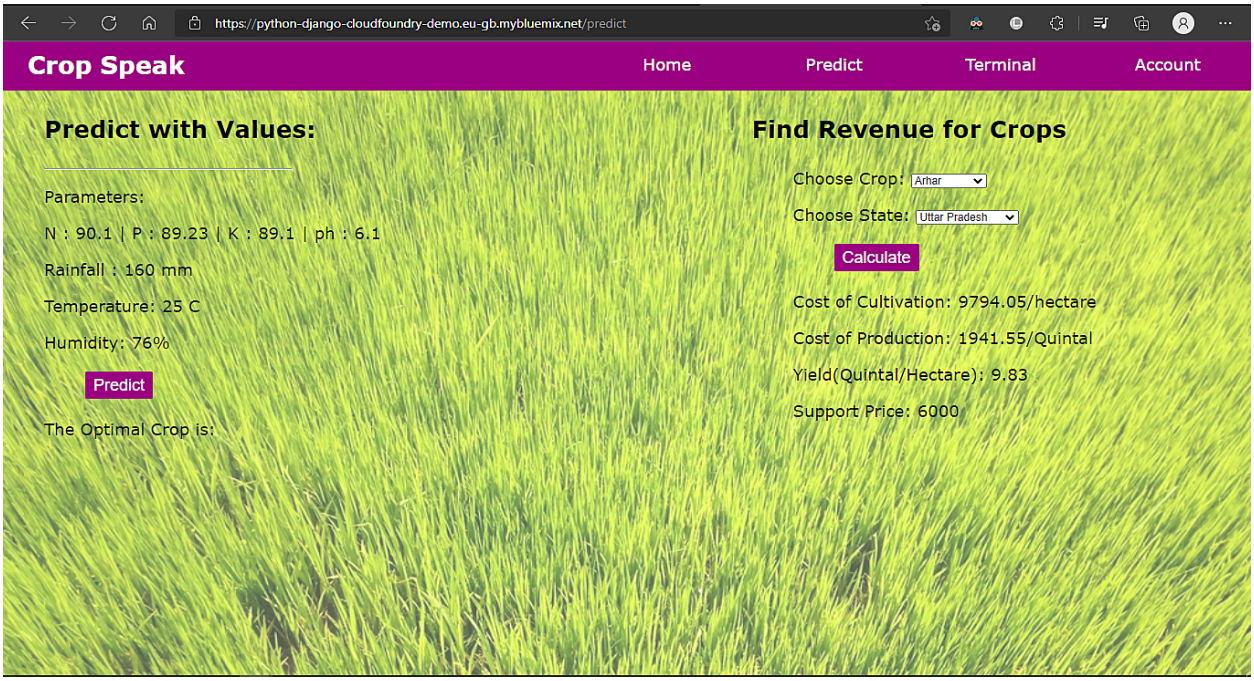
CROP SPEAK WEBSITE



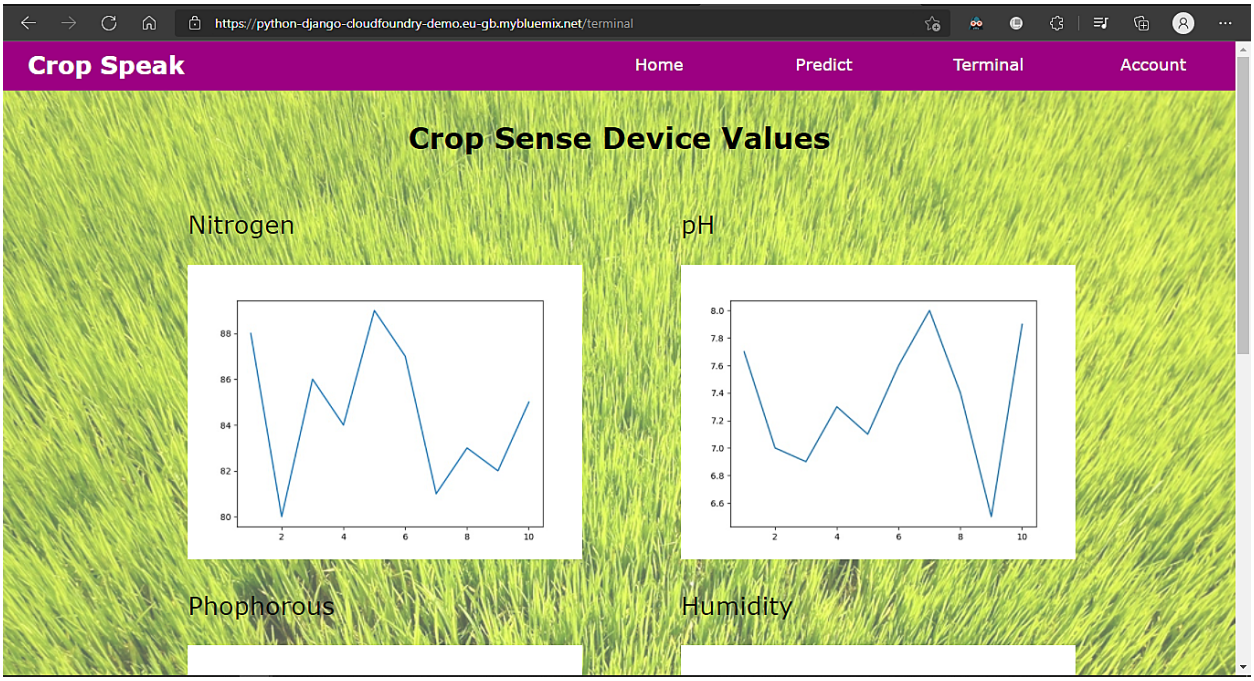
7. Crop Speak Landing Page



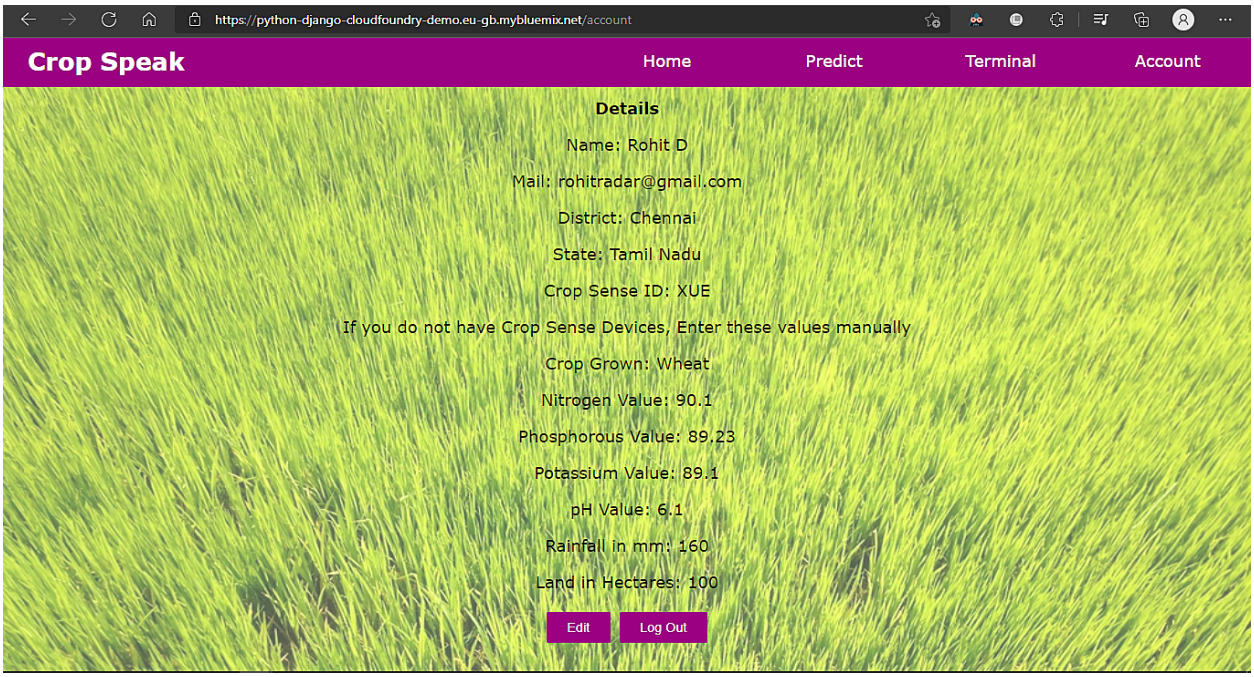
8. Crop Speak Home Page



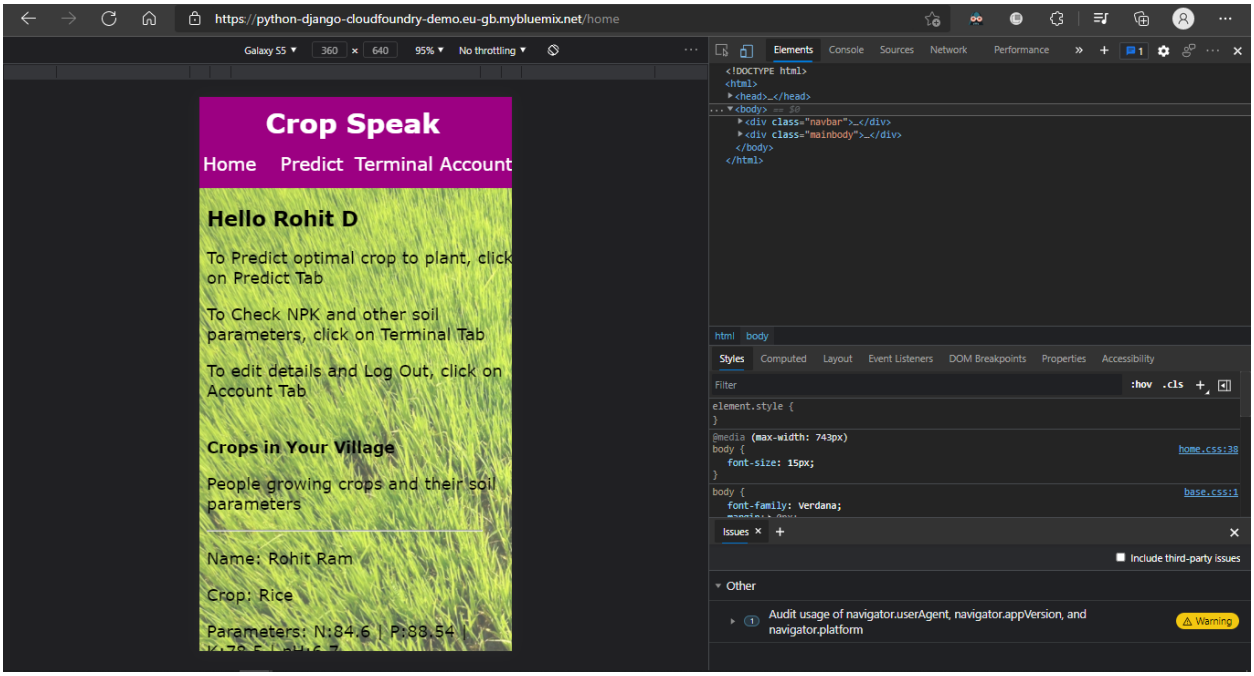
9. Yield and Prediction Page



10. IOT Platform integration to Website



11. Manual Crop Parameter Entry Page



12. Responsive Mobile Layout

7. ADVANTAGES AND DISADVANTAGES

7.1 Advantages

- Accurate Recommendations achieved using Watson Studio - 99.1%
- IOT Sensors give real time predictive analysis, which is displayed on the dashboard
- Interactive Website where users can input their sensor value data, if they do not have a crop sense sensor.
- The Website is free of cost to use.

7.2 Disadvantages

- Crop Sense Sensors require Internet services to be able to connect to the Cloud. This might not be available in remote regions of India
- Crop Sense Sensors are an added cost to the AI Assisted Solution, which might not be affordable by farmers who have smaller lands.

8. APPLICATIONS

- The Crop Speak and Crop Sense Systems can be used both in small scale and large scale farms. The Crop Sense IOT Device can be used even in gardens in the city. Irrigation Systems can be automated using the Crop Sense Device. Automatic Watering of plants is possible by combining Hardware and Software Solutions via the IBM IOT Platform.
- This App can help new comers in gardening or farming, helping them understand the planting and harvesting periods for crops.
- The modernization of farming can improve the society as a whole. Standing in the sun for hours in a day is no longer necessary if irrigation and and fertilizer systems are automated.
- Automated Farming can be an upcoming industry which can create new jobs and increase the GDP and productivity of the nation. Crops can be grown in surplus and exported to other countries if rapid modernization can be implemented to this industry.

9. CONCLUSION

To conclude, The Crop Speak and Crop Sense IOT Solutions is an integral part of automating the farming Industry. This solution can benefit the nation as a whole and the individual communities present in it. Being a cloud based solution, it can be implemented in a rapid scale and be integrated with multiple hardware solutions. Thus, the farming industry can be stepped upto industry 4.0 standards in the next coming years.

10. FUTURE SCOPE

Apart from the Crop Sense Device, the IOT Sphere can be improved vastly. Solutions such as

- Automated Irrigation System using motors,pumps and sprinklers as hardware when the
- Automated Manure/Fertilizer Dispenser using Sprinklers as Hardware when the soil nutrient levels are low.
- Pest/Disease Detection using Cameras and Surveillance systems for crops so that farmers can be alerted on the presence of rodents/pests on their farms. This way, the damage caused by pests/diseases can be prevented before it happens.

11. BIBLIOGRAPHY

1. Dataset for Crop Recommendation: https://github.com/https://www.kaggle.com/siddharthss/crop-recommendation-dataset
2. Dataset for Crop Yield: https://github.com/shreyzo/Crop-yield-and-profitability-prediction/blob/main/datafile.csv
3. Django App Deployment in Cloud
   How to Deploy Django Application | Cloud Foundry Foundation (medium.com)

11.2 APPENDIX

SOURCE CODE:

smartinternz02/SBSPS-Challenge-5506-AI-Assisted-Farming-for-Crop-Recommendation-Farm-Yield-Prediction-Application (github.com)

PYTHON REQUEST CODE:

```
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "hZYCe5CcFdL6-9Sjpps65rlWv2ZXIyxZyDPqyKlaS02t"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
fields=['N','P','K','temperature','humidity','ph','rainfall']
field_values=[10,10,10,10,10,10,10]
payload_scoring = {"input_data": [{"fields":[fields], "values":[field_values]}]}

link='https://us-south.ml.cloud.ibm.com/ml/v4/deployments/c58f94cf-d066-407b-a162-d8e2346d129c/predictions?version=2021-08-15'
response_scoring = requests.post(link, json=payload_scoring, headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
print(response_scoring.json())
```