Hack Challenge 2021

by Smart Internz

Project Report

AI-Assisted farming for Crop Recommendation and Farm Yield Prediction Application

Team Name : 4Eco

Members :

Kamya H

Gayathri S

Geethika P

Kruthi R

Students

of

SSN College of Engineering

Kalavakkam 603 110

INDEX

1. Introduction:

1.1. Overview:

The world's population is assumed to be nearly 10 billion by 2050, so boosting agricultural order is the need of the hour. At present, about 37.7% of total land surface is used for crop production. It contributes a significant portion in the economic prosperity of developed nations and it plays an active part in the economy of the developing countries. For countries like India, the agricultural sector accounts for 18% of GDP and provides employment to 50% of the country's workforce. Thus development in the agriculture sector will boost the economy.

Artificial intelligence has grown very popular in today's world and is prevalent in almost all domains, even in agriculture. AI is based on the principle that human intelligence can be defined in a way that a machine can easily mimic it and execute tasks, from the simplest to those that are even more complex. AI has been used to develop applications and tools which help farmers with accurate and controlled farming by providing them proper guidance about water management, crop rotation, timely harvesting, type of crop to be grown, optimum planting, pest attacks, nutrition management,etc.

1.2. Purpose:

The aim of this project is to develop models using IBM Cloud Services for AI assisted farming. This model is integrated to an user interface, which is developed using Node Red. This application recommends the best crop in accordance with climatic and soil conditions.It also predicts the appropriate Fertilizer for their crop. Hence the revenue can also be predicted by knowing the costs of production, cultivation and the minimum support price. For the ease of accessibility the application has been implemented in various regional languages. Text to speech is also implemented to make it comprehensible.

2. Literature Survey:

2.1. Existing problem

Listing down general challenges that exist in the agricultural domain:

- In farming climatic factors such as rainfall, temperature and humidity play an important role in the agriculture lifecycle. Increasing deforestation and pollution result in climatic changes, so it’s difficult for farmers to make decisions to prepare the soil, sow seeds, and harvest.
- Every crop requires specific nutrition in the soil. There are 3 main nutrients nitrogen(N), phosphorus(P) and potassium(K) required in soil. The deficiency of nutrients can lead to

poor quality of crops.

Based on research, the most popular applications of AI in agriculture appear to fall into three major categories:

- Agricultural Robots – Companies are developing and programming autonomous robots to handle essential agricultural tasks such as harvesting crops at a higher volume and faster pace than human laborers.
- Crop and Soil Monitoring – Companies are leveraging computer vision and deep-learning algorithms to process data captured by drones and/or software-based technology to monitor crop and soil health.
- Predictive Analytics – Machine learning models are being developed to track and predict various environmental impacts on crop yield such as weather changes.

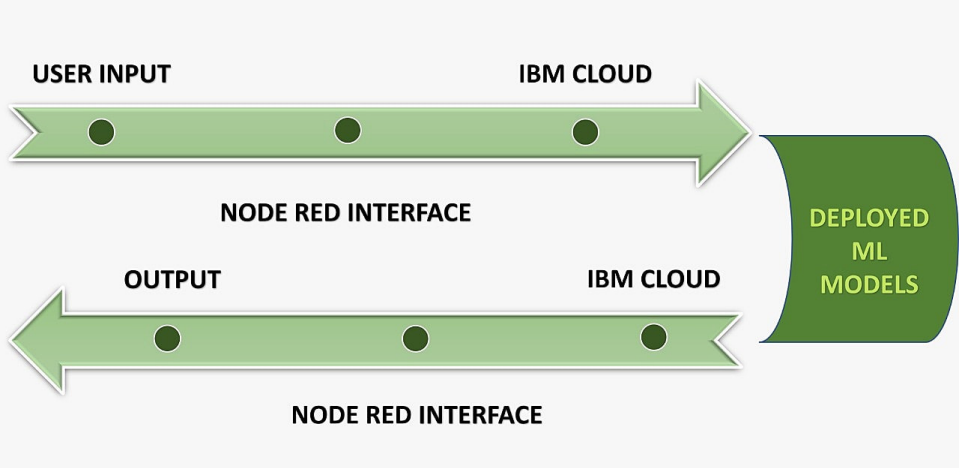
2.2. Proposed solution

Using AI systems to improve harvest quality and accuracy is a management style known as *precision agriculture*(PA). PA uses AI technology to aid in detecting diseases in plants, pests and poor plant nutrition on farms.

In this project we will be building a model that recommends crops and fertilizers as well as predicts the revenue and yield which will aid the farmers to maximize profits. This model will be integrated to an user Interface which will make it easily accessible.The main features of our project include, a Good User Interface, easy data visualization, multilingual capabilities (particularly regional languages) and Speech-text recognition.

3. Theoretical Analysis:

3.1. Block Diagram



3.2.Hardware/ Software Designing

The following softwares in IBM cloud platform have been used for various purposes to achieve the desired results:

- IBM Watson Studio:

  It is IBM's software platform for Data Science. It helps in creating a project with collaborators, bringing many open source tools like Python, Spark and RStudio under an integrated environment. In this project we have performed data binning and manipulation on the datasets for Crop Recommendation, Fertilizer Recommendation and Revenue Prediction using this software.

- Auto AI:

  It helps in automating the procedure for building, training and testing machine learning models. It has assisted us in building high quality predictive models for Crop Recommendation, Fertilizer Recommendation and Revenue Prediction.

- Watson Machine Learning:

  IBM Watson Machine Learning is a service offered by IBM Cloud that makes it easy to integrate predictive capabilities with desired applications. In this project Watson Machine Learning Service has been used with Auto AI facility from IBM Watson Studio to build 3 models for our application.

- Node Red:

  Node Red is a programming tool used to integrate APIs and online services. It provides a browser based editor that makes it easy to create flows using various available nodes. In this project we have used Node Red to create a web application which takes into consideration the linguistic diversity of India and provides the service of Crop Recommendation, Fertilizer Recommendation and Revenue Prediction in 8 different languages.

4. Experimental Investigation:

4.1.Backend:

Initially, data is collected from various sources and the raw data is sorted into datasets. Here 3 datasets are needed, one each for crop recommendation, fertilizer recommendation and revenue prediction.

A new project is created in IBM Watson Studio and then datasets are loaded into it. These datasets are then checked using auto AI to avoid duplications. Auto AI is also run on the dataset to find the appropriate pipeline.

Crop Prediction: This uses a multiclass classification (Extra Tree Classifier) algorithm. With inputs of Humidity, Moisture, N, P, K,Temperature, Soil type and Crop type it predicts the appropriate crop.

Fertilizer Recommendation: This uses Decision Tree Classification to recommend the suitable fertilizer for the crop by taking the values of N,P,K levels of the soil, temperature, humidity, moisture of the surroundings, type of the Soil and the crop.

Revenue Prediction: This uses Extra Trees Regression to Predict the Yield using the Inputs given for Cost of cultivation (A2+FL) and C2, Cost of Production (C2), Minimum Support Price, State and Crop.

This pipeline is then chosen in order to be deployed. In the deployment space the value to be predicted is chosen and the deployment is run to check if the correct value is being predicted. Once the deployment space is created an endpoint is generated which is to be stored separately.

4.2.User Interface:

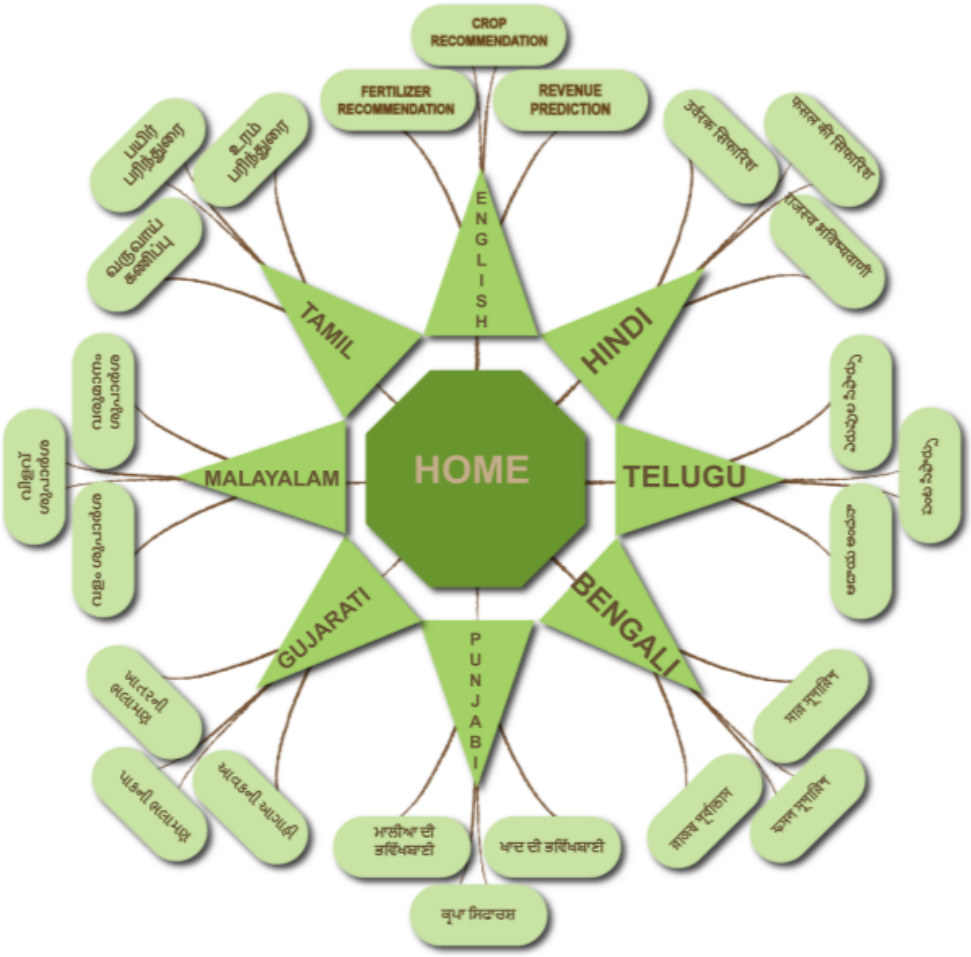
For the user interface, Node Red is used. A Node Red instance is created from the IBM Cloud. In Node Red various types of nodes like form node,dropdown node, button node, function node, etc., are used to design the dashboard.

Once the basic layout is created it has to be linked with the deployed model to create the web application. Values are acquired from the UI by setting the necessary attributes of the dataset as global variables. API keys are generated from the IBM Cloud. Access tokens are made available with the help of API keys and http request. The function nodes are used to do the same. The endpoint and the API Key are put in the respective function nodes to link the machine learning model to the web application.The web application is then deployed to predict the required parameters.

The User interface of this project is made Multi-lingual using the Language Translator Node, making our application easily accessible and comfortable for the users.

A Text-to-Speech conversion is also implemented with the help of Text-speech Node available in Node Red, aiding the Visually Impaired People to use our Web Application. Other Languages can be accessed by Navigating to Respective Tabs and the specific Agenda by navigating through Groups in their selected Languages . These navigations are done using buttons and the UI Control Node

5. Flow Chart:



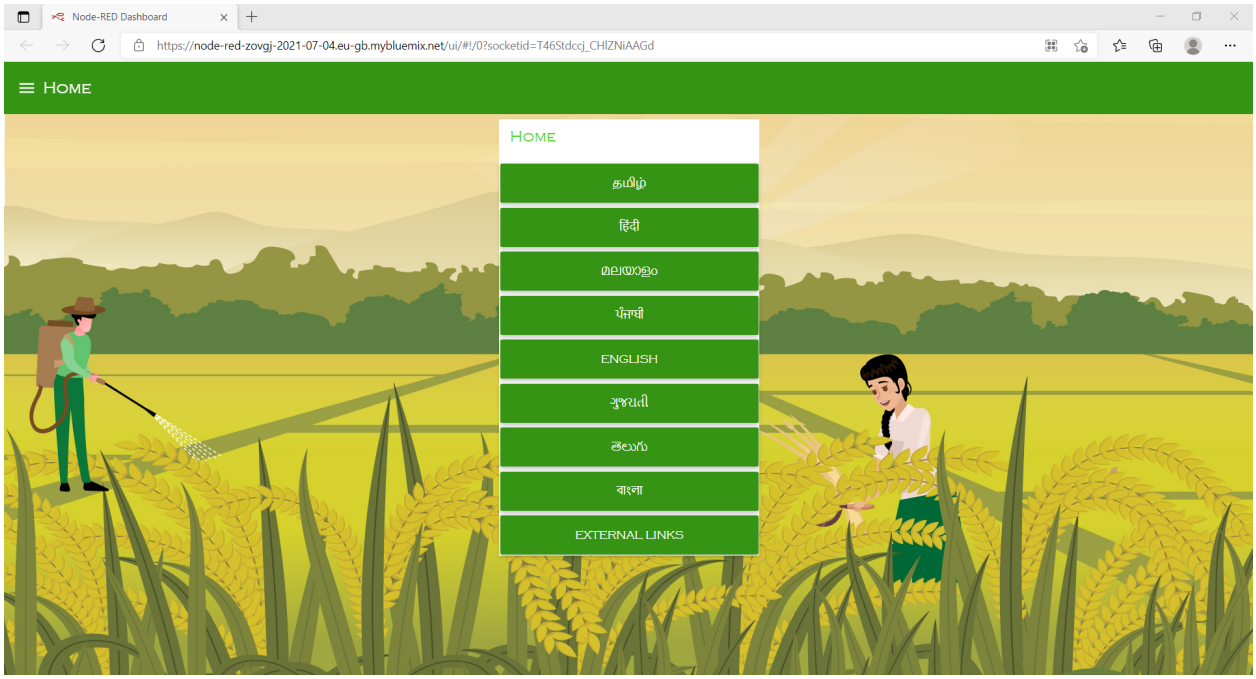
6. Result:

In this user friendly application:

- Crops can be recommended.
- Revenue and yield values can be determined.
- Fertilizers can be predicted.
- All of the above can be done in 8 different languages.
- The output of crop recommendation can be given out as an audio message(text-to-speech implementation).
- Helplines to provide timely help for farmers have been included.

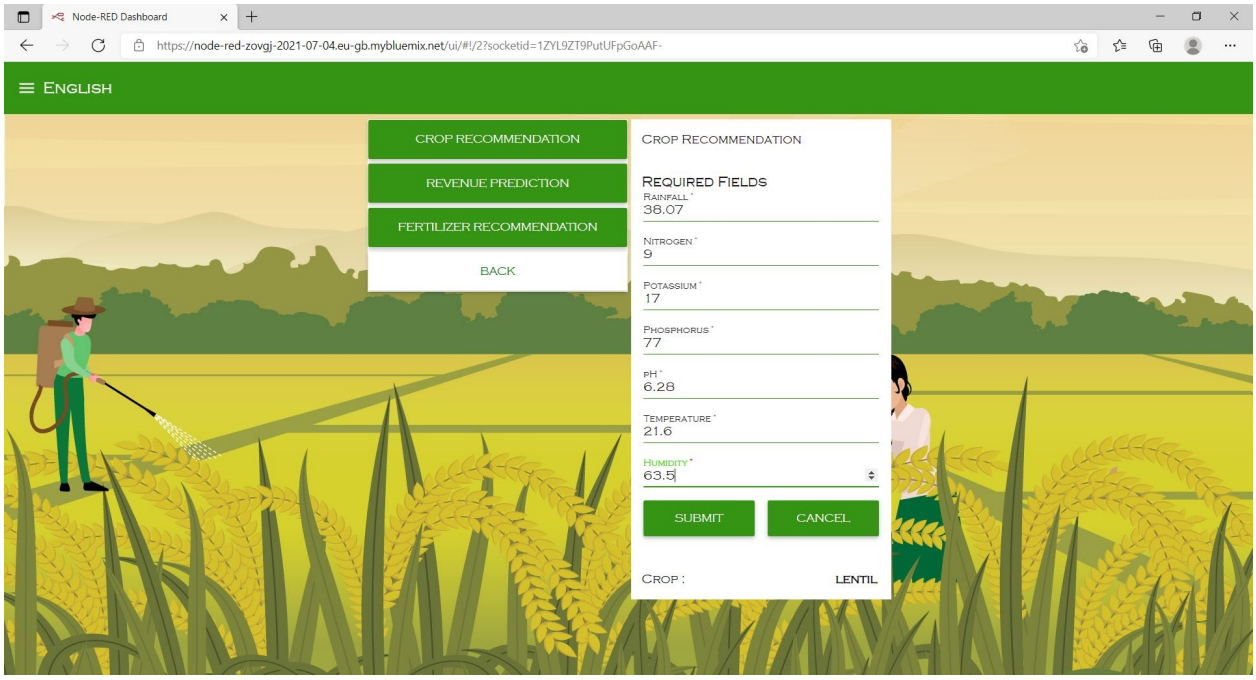
Output screenshots:

Home Page:

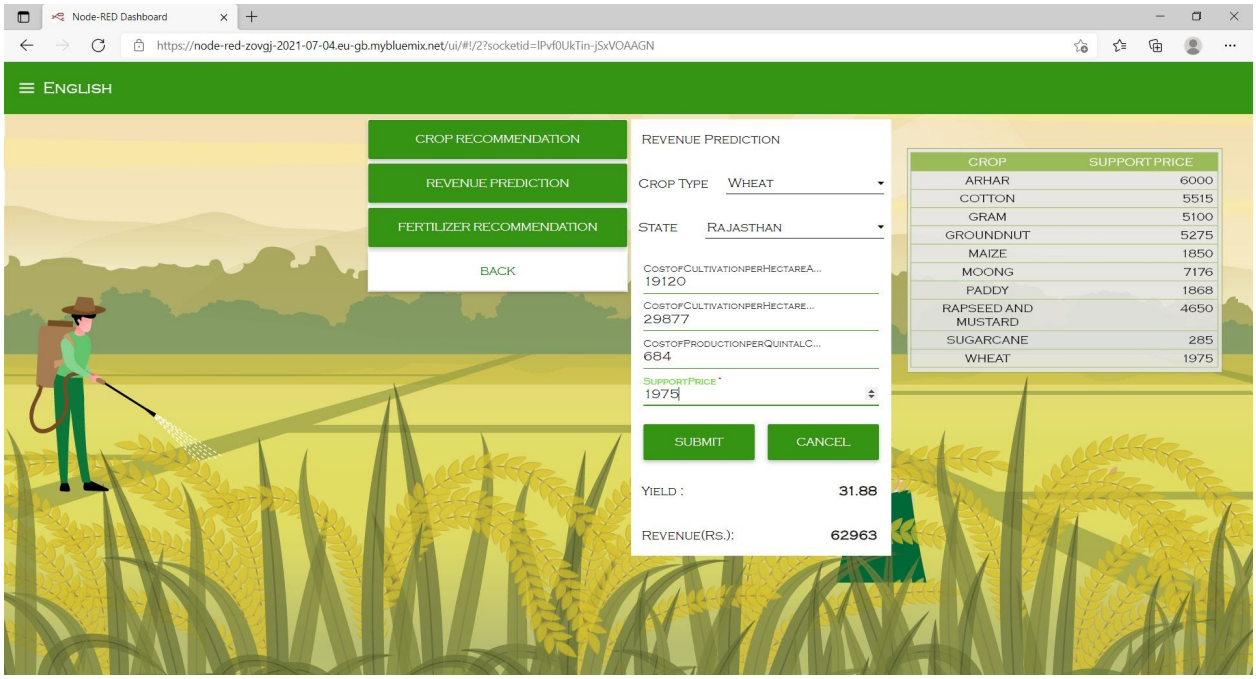


In English:

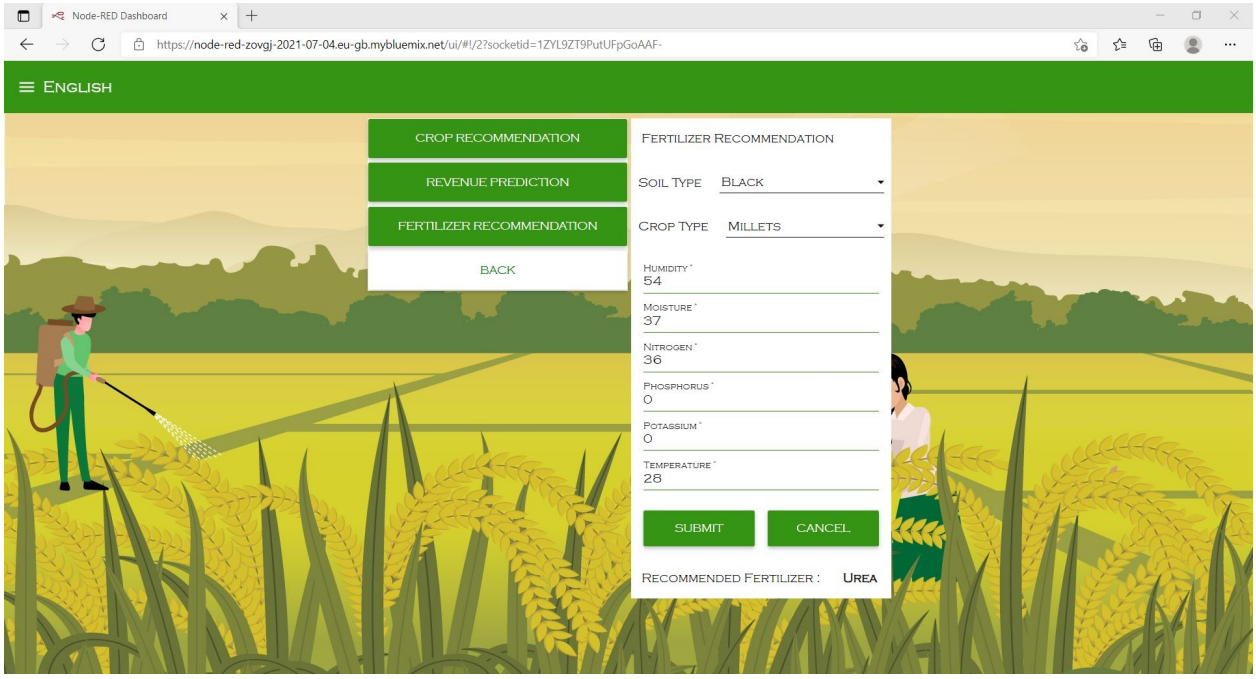
Crop Recommendation:



Revenue Prediction

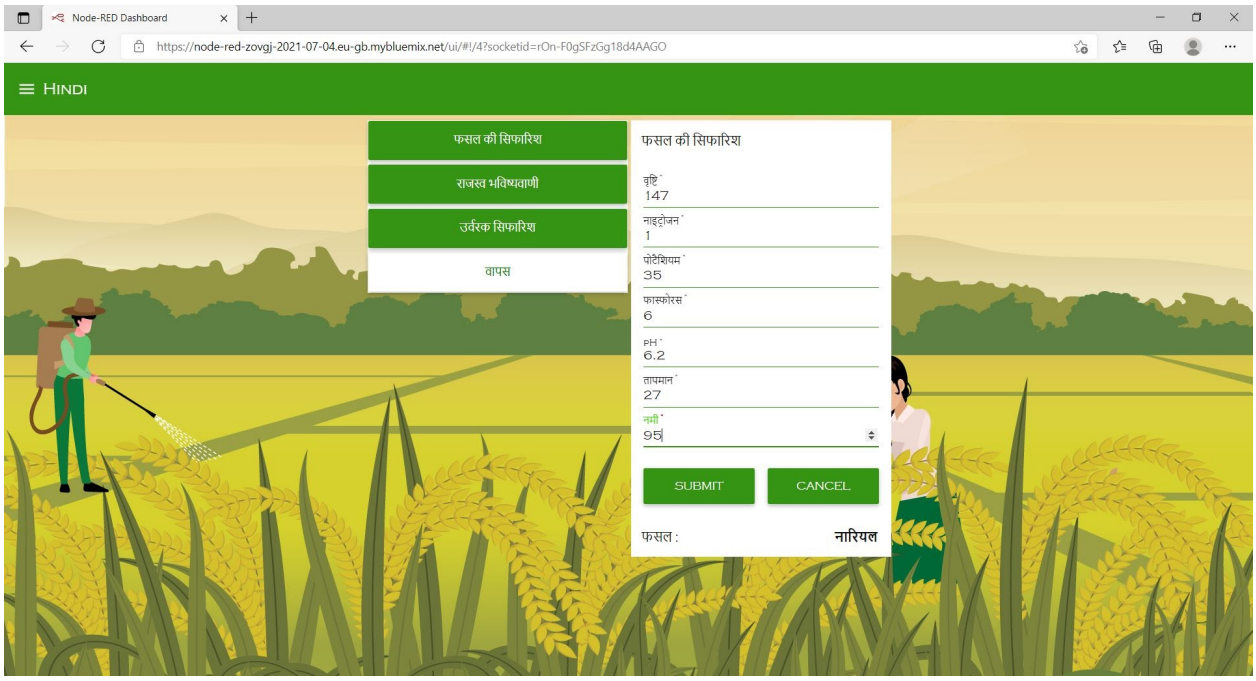


Fertilizer Recommendation

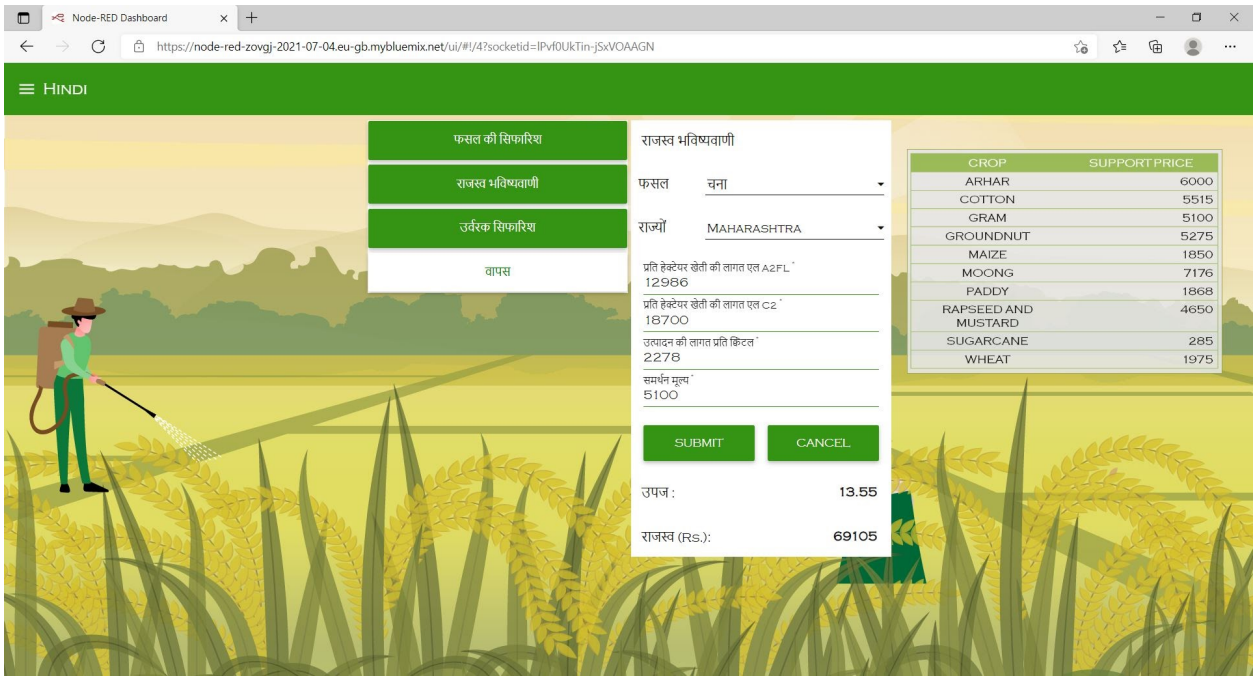


In Hindi

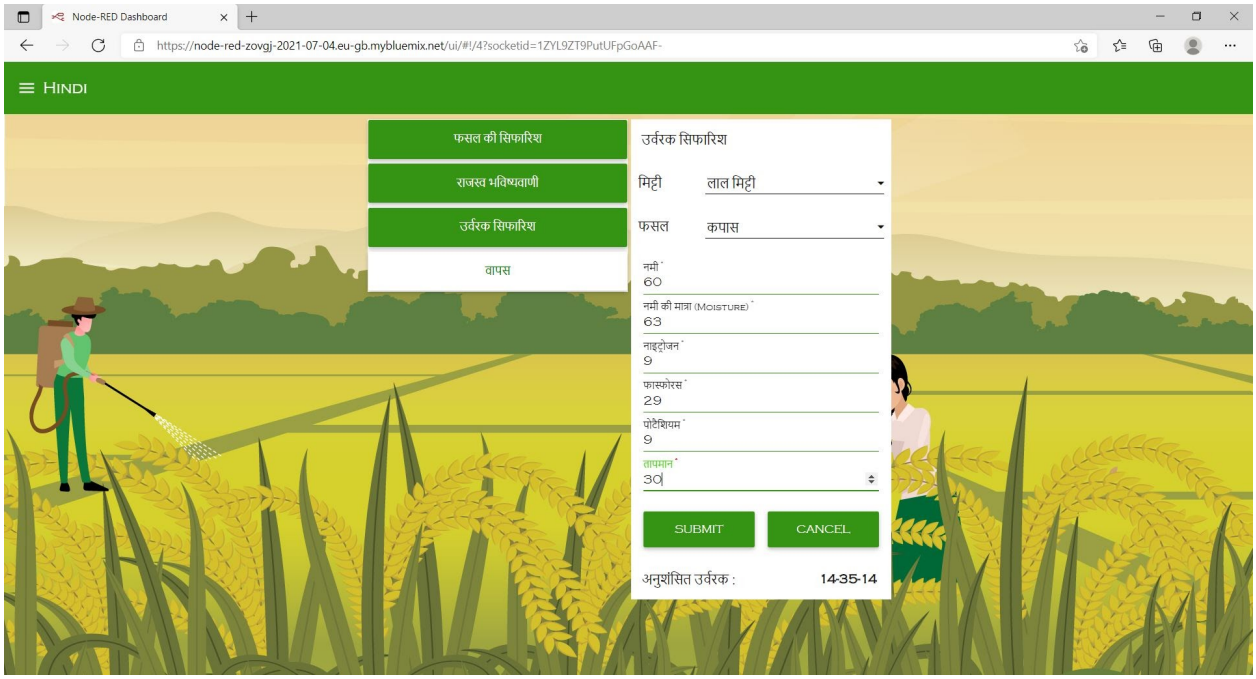
Crop Recommendation



Revenue Prediction:



Fertilizer Recommendation



More output screenshots can be found in the drive:
https://drive.google.com/drive/folders/1trBYtaqx875IMXPs4DKBtHF19oX1BNiw?usp=sharing

7. Advantages & Disadvantages:

7.1.Advantages:

- Multilingual implementation to have a broad reach.
- Text to speech conversion to aid the physically disabled.
- Fertilizers can be predicted.

7.2. Disadvantages:

- Not all crops can be predicted.
- Crops cannot be predicted in all kinds of climatic conditions.
- Better UI can be provided and can be made more automated.

8. Applications:

This handy web application helps the farmers by recommending the best crops and fertilizers for their field taking into account the soil and climatic conditions. It also helps the farmers by forecasting the yield and revenue thereby making planning and managment easier. This application provides service in 8 different languages. Thus the farmers can comfortably use the application in their own regional language. Text-to-speech provision is also available in the English language.

9. Conclusion:

Artificial Intelligence in agriculture not only helps farmers to automate their farming but also enables them to shift towards precise cultivation for higher crop yield and better quality while using fewer resources.

Companies involved in improving machine learning or Artificial Intelligence-based products or services like training data for agriculture, drones and automated machine making will get technological advancement in the future and will provide more useful applications to this sector helping the world deal with food production issues for the growing population.

10. Future Scope:

Our solution to the problem can be enhanced by adding a few more features:

- Minimum Support Price can be dynamically updated.
- Data on availabitlity of different storage centres can be provided to make transportation of the produce easy. We can avoid the wastage of grains due to climate or insufficient space by implementing this model.

11. Bibliography:

https://nodered.org/docs/tutorials/
https://youtube.com/playlist?list=PLyNBB9VCLmo1hyO-4fIZ08gqFcXBkHy-6
https://flows.nodered.org/node/node-red-dashboard
https://discourse.nodered.org/t/inject-node-with-input/22662
https://youtu.be/0nKGHGJcvls

Appendix

A.Source Code

For Crop Recommendation

```
import requests

# NOTE: you must manually set API_KEY below using information retrieved from      your IBM Cloud account.
API_KEY = "<your API key>"
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}

# NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data": [{"fields": [array_of_input_fields], "values":
[array_of_values_to_be_scored, another_array_of_values_to_be_scored]}]}

response_scoring =
requests.post('https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/17afdefa-1f18-45dd-a5af-0b909
0e2f2f8/predictions?version=2021-07-14?version=2021-07-14', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
print(response_scoring.json())
```

For Fertilizer Prediction

```
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your
IBM Cloud account.
API_KEY = "<your API key>"
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}

#NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data": [{"fields": [array_of_input_fields], "values":
[array_of_values_to_be_scored, another_array_of_values_to_be_scored]}]}
```

```
response_scoring =
requests.post('https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/2644b905-de22-41a9-8d65-7bac9cf9f504/predictions?version=2021-08-18?version=2021-08-18', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
print(response_scoring.json())
```

For Revenue Prediction:

```
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "<your API key>"
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}

# NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data": [{"fields": [array_of_input_fields], "values":
[array_of_values_to_be_scored, another_array_of_values_to_be_scored]}]}

response_scoring =
requests.post('https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/e5438216-3d56-496f-84a3-6697479a0b62/predictions?version=2021-08-24?version=2021-08-24', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
print(response_scoring.json())
```