CSE3999 - Technical Answers for Real World Problems (TARP)

Project Report

Title

By

18BCE1140 Masooma Suleman
18BCE1047 Aditya Prasad
18BCE1043 Kaustubh Jha
18BCE1016 Soham Sarkar
18BCE1340 Harsh Kailash
18BCE1218 Udayan Chaudhary

B.Tech. Computer Science and Engineering

Submitted to

Dr. G.Malathi

School of Computer Science and Engineering

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

June 2021

DECLARATION

I hereby declare that the report titled "**Smart Irrigation System"** submitted by our team to VIT Chennai is a record of bona-fide work undertaken by me under the supervision of **Dr G.Malathi**, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai.

(Aditya Prasad,Masooma Suleman, Kausutbh Jha,Soham Sarkar,Harsh Kailash,Udayan Chaudhary)

Reg. No.
18BCE1047,18BCE1140,18BCE1043,18BCE1016,18BCE1340,18BCE1218

<u>CERTIFICATE</u>

Certified that this project report entitled “**Smart Irrigation System”** is a bonafide work of **Masooma Suleman(18BCE1140), Aditya Prasad(18BCE1047), Kaustubh Jha(18BCE1043), Soham Sarkar(18BCE1016), Harsh Kailash(18BCE1340), Udayan Chaudhary(18BCE1218)** and they carried out the Project work under my supervision and guidance for CSE3999 - Technical Answers for Real World Problems (TARP).

Dr G.Malathi
SCOPE, VIT Chennai

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr G.Malathi**, School of Computer Science and Engineering for his/her consistent encouragement and valuable guidance offered to us throughout the course of the project work.

We are extremely grateful to **Dr. R. Jagadeesh Kannan**, Dean and **Dr. S. Geetha**, Associate Dean, School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, Chennai, for extending the facilities of the School towards our project and for their unstinting support.

We express our thanks to **Dr. Justus S,** Head of the Department, B.Tech. Computer Science and Engineering for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the courses.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

(Aditya Prasad,Masooma Suleman, Kausutbh Jha,Soham Sarkar,Harsh Kailash,Udayan Chaudhary)

Reg. No. 18BCE1047,18BCE1140,18BCE1043,18BCE1016,18BCE1340,18BCE1218

ABSTRACT

The real-world problem we identified is linked to the agricultural field. We started off with development of a smart irrigation system but later expanded our solution to providing a one-step solution to the Indian farmers.

An android application accompanied with a corresponding website where he/she can avail the following facilities.

1)Fertilizer Recommendation System: - Suggests the farmer which fertilizer he/she should use for their crops using Machine Learning Models.

2)Crop Recommendation System: - It recommends which crops a farmer must grow according to the quality of the soil and its nutritional values.

3)Plant Disease Detection System: - It helps in identifying the plant health and predicts if any of the crops is suffering from any disease.

The technologies involved are: -

1)Machine Learning.

2)Android Development (Flutter, Firebase).

3)Deep Learning.

4)Internet of Things.

5)Web Development and deployment.

CONTENTS

1. Introduction

1.1 **Objective and goal of the project**

The main objective of our project is to provide a one-stop farming solution to the farmers. A technological platform equipped with latest machine learning and deep learning techniques to provide agricultural solutions.

1.2 **Problem Statement**

To automate agricultural processes such as plant disease detection, crop recommendation using advance technologies such as deep learning.

2. Literature Survey

RESEARCH PAPERS:

1. Smart Farming: IoT Based Smart Sensors Agriculture Stick for Live Temprature and Moisture Monitoring using Arduino, Cloud Computing & Solar Technology

   Link:
   https://www.researchgate.net/publication/313804002_Smart_farming_IoT_based_smart_sensors_agriculture_stick_for_live_temperature_and_moisture_monitoring_using_Arduino_cloud_computing_solar_technology

   Author:
   Anand Nayyar:-
   Assistant Professor, Department of Computer Applications & IT KCL Institute of Management and Technology, Jalandhar, Punjab
   Er. Vikram Puri:-
   M.Tech(ECE) Student, G.N.D.U Regional Center, Ladewali Campus, Jalandhar

   Conference: The International Conference on Communication and Computing Systems (ICCCS-2016)

   Dataset used:
   Data of Soil Moisture
   Data of Temperature
   Data of Solar Power Plates powering the entire Agriculture IoT Stick

   Methodology used:
   Smart Farming Based Agriculture IoT Stick is regarded as IoT gadget focusing on Live Monitoring of Environmental data in terms of Temperature, Moisture and other types depending on the sensors integrated with it. Agricultural IoT stick provides the concept of "Plug & Sense" in which farmers can directly implement smart farming by as such putting the stick on the field and getting Live Data feeds on various devices like Smart Phones, Tablets etc. and the data generated via sensors can be easily shared and viewed by agriculture consultants anywhere remotely via Cloud Computing technology integration. IoT stick also enables analysis of various sorts of data via Big Data Analytics from time to time.

   Performance:

Conclusion:
In this Research Paper, a Novel Smart Farming Enabled: IoT Based Agriculture Stick for Live Monitoring of Temperature and Soil Moisture has been proposed using Arduino, Cloud Computing and Solar Technology. The stick has high efficiency and accuracy in fetching the live data of temperature and soil moisture. The Agriculture stick being proposed via this paper will assist farmers in increasing the agriculture yield and take efficient care of food production as the stick will always provide helping hand to farmers for getting accurate live feed of environmental temperature and soil moisture with more than 99% accurate results.

FUTURE SCOPE

Future work would be focused more on increasing sensors on this stick to fetch more data especially with regard to Pest Control and by also integrating GPS module in this IoT Stick to enhance this Agriculture IoT Technology to full-fledged Agriculture Precision ready product.

2. A RESEARCH PAPER ON SMART AGRICULTURE USING IOT

Link: https://www.irjet.net/archives/V7/i7/IRJET-V7I7479.pdf

Author:
Ritika Srivastava, Vandana Sharma, Vishal Jaiswal, Sumit Raj(Students of B.Tech (CSE)Krishna Engineering College, Ghaziabad, Uttar Pradesh)

Conference:
International Research Journal of Engineering and Technology (IRJET)

Dataset used:
Data of Soil Moisture
Data of water level

Methodology used:
This paper describes Internet of Things (IOT) technology has brought revolution to each and every field of common man's life by making everything smart and intelligent. IOT refers to a network of things which make a self-configuring network. The development of Intelligent Smart Farming IOT based devices is day by day turning the face of agriculture production by not only enhancing it but also making it cost-effective and reducing wastage. The aim /

objective of this paper is to propose a Novel Smart IOT based Agriculture assisting farmers in getting Live Data (Temperature, Soil Moisture) for efficient environment monitoring which will enable them to do smart farming and increase their overall yield and quality of products.

Performance:
Based on the results obtained in the analysis of the articles considered for this study, it was possible to observe a growth trend in the number of publications related to IoT and smart farming since 2011, with special emphasis from 2016 onwards

Conclusion:
We have designed automated Smart Agriculture system which reduces the time and resources that is required while performing it manually. This system uses the technology of Internet of Things. The system also measure moisture of soil and level of water in fields. This system works well in the ideal conditions and further improvement can be made when the conditions are not ideal like proper illumination or lightning.

FUTURE SCOPE:
In future we try design this system much smarter where we can even apply food nutrients to the root of plant and crop by air mixed with very less water. And also try to give some advanced features such as detection of broken pipes and identification of sensor damaged position. Detection damaged sensor positions provides help to farmer for easily access and replace particular sensors. This future will make it more efficient and useful

3. On the Analysis of Cascading Style Sheets

Link:
https://www.researchgate.net/publication/254008906_On_the_Analysis_of_Cascading_Style_Sheets

Authors:
Pierre Genevès- pierre.geneves@inria.fr
Nabil Layaïda- nabil.layaida@inria.fr
Vincent Quint- vincent.quint@inria.fr

Conference: Proceedings of the 21st international conference on World Wide Web

Dataset used:

This research paper is on Cascading Styling Sheet(CSS). No such dataset is used

Methodology used:

We use a tree logic capable of capturing the semantics of CSS selectors as well as schemas. Schemas we consider are regular tree grammars which capture most of the XML Schemas, Relax NG schemas, and DTDs. Our approach consists in modeling element selection performed by CSS selectors and structural constraints described by schema information into the tree logic. We then use an algorithm to check satisfiability of formulas of the logic. Such an algorithm defines a partition of the set of logical formulas: satisfiable formulas (for which there exist at least one tree, among those defined by the schema, that satisfies the constraints expressed by the formula) and remaining formulas which are unsatisfiable (no tree satisfies the formula). Alternatively (and equivalently), formulas can be divided into valid formulas (formulas which are satisfied by all trees) and invalid formulas (formulas that are not satisfied by at least one tree). The use of a satisfiability-testing algorithm allows proving validity of a given logical statement P by testing its negation (¬P) for unsatisfiability.

Performance:

The result of the analysis corresponds to two situations: either the formula is found unsatisfiable (meaning that the checked property holds for any tree), or it is satisfiable. In this case, the solver generates a counter-example document satisfying the formula

Conclusion:

In this paper, we introduce the concept of static analysis for CSS style sheets. To the best of our knowledge, this is the first attempt at statically analyzing CSS style sheets. We propose an original tool based on recent advances in tree logics. The tool is capable of statically detecting a wide range of common errors, as well as proving properties related to sets of documents, such as coverage of styling information, in the presence or absence of schema information

Future Scope:

There are several directions for future work. One is to characterize erroneous patterns in the box model and in positioning. As seen in the examples, such errors may also be captured by logical descriptions regardless of values such as sizes, paddings, etc. Another direction is to extend the analysis to complex and very large style sheets such as those for Docboo

4. An Empirical study on Drip Irrigation

Link:
https://www.researchgate.net/publication/321137448_DRIP_IRRIGATION-full_paper

Authors:

E.Sathyapriya[1*] M.R.Naveenkumar[2] and V.Dhivya[3]

[1&2] Ph.D. Scholars, Department of Agricultural Extension and Rural Sociology,

[3] Ph.D. Scholars, Department of Agricultural Entomology, TNAU, Coimbatore.

Mail: sathyasree321@gmail.com

Dataset used:

Data of water dripped into the field using the system produced

Methodology:

The present study was conducted Dindigul district of Tamilnadu. The expost - facto research design was used for the study. A sample consisting of 30 drip irrigation farmers were selected randomly from the purposively selected three villages of R.P. Pudhur, Manjanaickenpatty and chatrapatti in Oddanchatram taluks, where in maximum area of horticultural crops is irrigated by drip method. The questionnaire was developed keeping the objectives of the study in the background, presented in non - sampling area and then employed for collecting the required data from the respondents.

Performance:

The results revealed that majority of the respondents opined that saving of water (93.33 %) is major benefit of drip irrigation and followed by Uniform application (90.00 %), Easy method of irrigation (86.67 %), Increased crop yield(76.67 %), Saving of labour cost for irrigation(73.33 %), Decreased weed growth(70.00%) and Improved quality of produce (66.67 %).

Conclusion:

The Benefits encountered by the farmers are saving of water, uniform application and easy method of irrigation and the constraints are problem of non-availability of quality material and no follow up services by drip agencies. It is clear from the study that the drip irrigation agencies, financing institutions and others to supply adequate standard spare parts and other appropriate measures to ensure the satisfactory situation for proper adoption of drip irrigation method.

Future Scope:
Drip irrigation will definitely grow in the near future because it uses less water and increases crop yield. This is important because water is quickly becoming a rare resource. The increasing population is putting pressure on the agriculture industry to increase yield. This is why drip irrigation has a strong secure future.

5. Node.js Challenges in Implementation

   Link:
   https://www.researchgate.net/publication/318310544_Nodejs_Challenges_in_Implementation

   Author:
   By Hezbullah Shah & Tariq Rahim Soomro

   Conference: National conference on Micro Irrigation At: TNAU, Coimbatore

   Dataset used:
   The survey was created on Google forms and ran for 1 month from the following link:
   https://docs.google.com/forms/d/e/1FAIpQLSc4GhroqubE5jQSnzmPOXEWuXzD8IcpRCtFPbSfFYscszX SVg/viewform?hl=en

   Methodology:
   This study gains knowledge from the Node.js domain experts through intensive literature review. This study also gains knowledge from a surveys conducted from the professional developers. The survey was created on Google forms and ran for 1 month. The survey was targeted to the developers groups on the LinkedIn, Facebook and also shared with professional developers

Performance:

The Literature Review concluded that Node.js can be useful and should be implemented in any place where there is processing of large files or requires large network load. Below are discussed some results in the same context. Due to Node.js a developer can easily become Full Stack Developer where he as a developer does not require separate colleagues for server side development and database development. Also the employers can reduce their cost by adopting Node.js as they will find a single developer taking care of performing all task at server side as well as at client side.

Conclusion:

The study got the finding about the implementation of Node.js. The Node.js have made Full Stack Developers’ job a dream come true. In absence of Node.js it was hard for a developer to learn several different languages and environments to manage the complete system at server side and client side. Organizations and developers can now with the invent of Node.js build highly load bearable and faster applications and by using Single Page Applications (SPA) now the server calls are reduced and the applications are more user friendly and faster.

Future scope:

As a result of the survey, a challenge comes to front is that most organizations are not ready to adopt the new technologies like Node.js over their existing ones like PHP, .Net, etc. Also there is a lack of market awareness which is causing a barrier to adopt Node.js for implementation. At a developer level, there is a challenge which is seen from the survey results that they are not feeling it easy to learn the database working and using of the JavaScript environment. And there also seems a lack of enough knowledge among the developers as from the survey results a reasonable respondent to the questions on the important features like event-driven, non-blocking I/O and asynchronous processing is making the decision about this

3 Requirements Specification

3.1 **Hardware Requirements**

3.1.1 Laptops with Intel Processors.

3.2 **Software Requirements**

3.2.1 Android Development

3.2.1.1 Flutter

3.2.1.2 Firebase

3.2.2 Machine Learning

3.2.2.1 Jupyter Notebook

3.2.2.2 Anaconda Navigator

3.2.2.3 Flask

3.2.3 Web Development

3.2.3.1 Microsoft Visual Studio Code

3.2.3.2 PHP

3.2.3.3 JavaScript

3.2.3.4 HTML5

3.2.3.5 Cascading Style Sheets

4 System Design

Our project is modular and we identified three main modules namely: -

4.1 Android Application and Internet of Things

- 4.1.1 We designed the different pages/screens of the app.
- 4.1.2 Index Page.
- 4.1.3 Home Page.
- 4.1.4 Crop Details Page.
- 4.1.5 Profile Page.

4.2 Machine Learning and Image Processing

- 4.2.1 Data Acquisition from various websites such as Kaggle, Github.
- 4.2.2 Exploratory Data Analysis setup in Jupyter Notebooks.

5 Implementation of System

(a) Android App Development

Agrofy (our app) is developed using flutter. Flutter is Google's SDK for crafting beautiful, fast user experiences for mobile, web, and desktop from a single codebase. Flutter works with existing code, is used by developers and organizations around the world, and is free and open source. It combines the quality of native apps with the flexibility of cross-platform development. Flutter renders the same look as a native app: it draws the UI from scratch, not acts as a wrapper on top of native UI components like other frameworks do.

Registration to Agrofy is a 2-step procedure. One can sign up into the app using mobile number or email address. In the first step of registration, one is asked to enter their basic details like name, profession, phone number, organization and address. In the next step, we ask for the experience of the user and any other related achievements.

Once the user has signed in successfully, he/she will land on the home page from where they can navigate to their profile page, field detail page and related FAQs and news page.

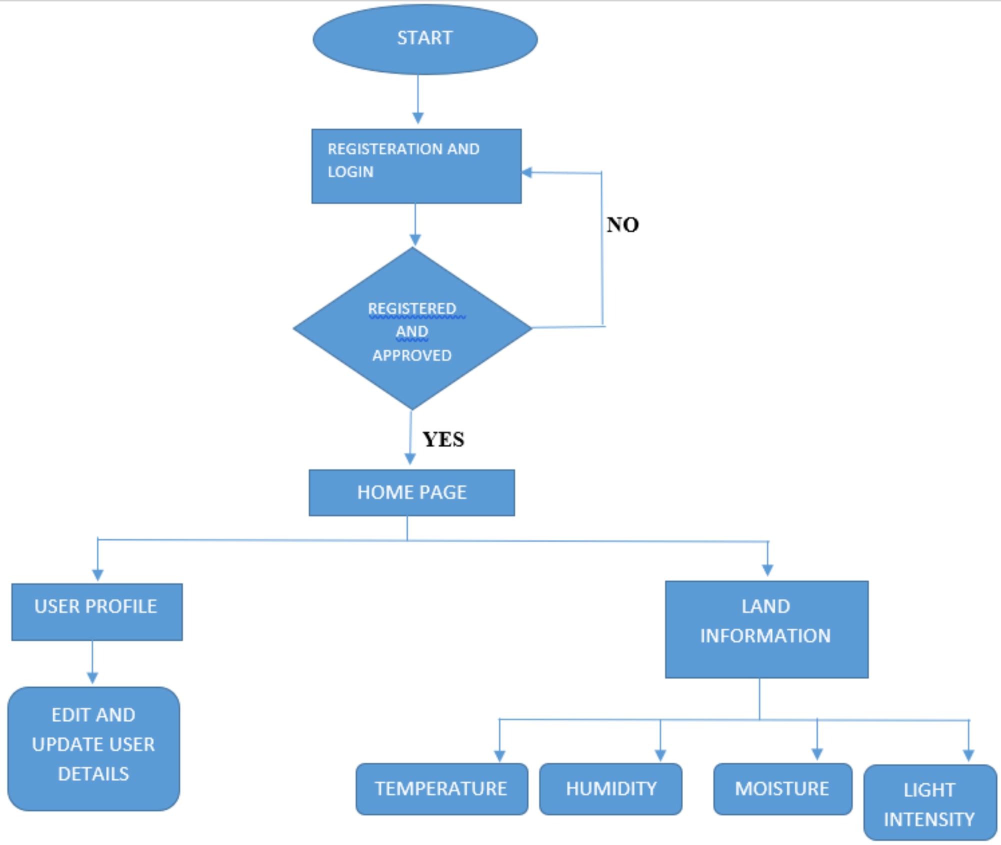
The field detail page consists soil information and displays real time data which is acquired from the IOT sensors deployed on the user's field. This data is accessed from the backend (firebase) whose end point is connected to nodeRed (explained in IOT section). The user can visualize the important data concerning to his farm/field namely

1. Soil moisture content
2. Humidity and temperature of the field
3. Plant details and growth rate
4. Light intensity in the field

WORKFLOW

1) Identification of number of pages and the navigation flow of the App.
2) Backend API creation and connection with node Red
3) Creation of User Authentication and login API
4) Backend integration (Firebase) with flutter project

Testing the App



(b)Internet of Things

WORKFLOW:

The Internet of Things aspect of the project will be handled in the following manner:

1) Identifying the sensors: We need to identify that what sensors do we need to deploy on the field to proceed further.

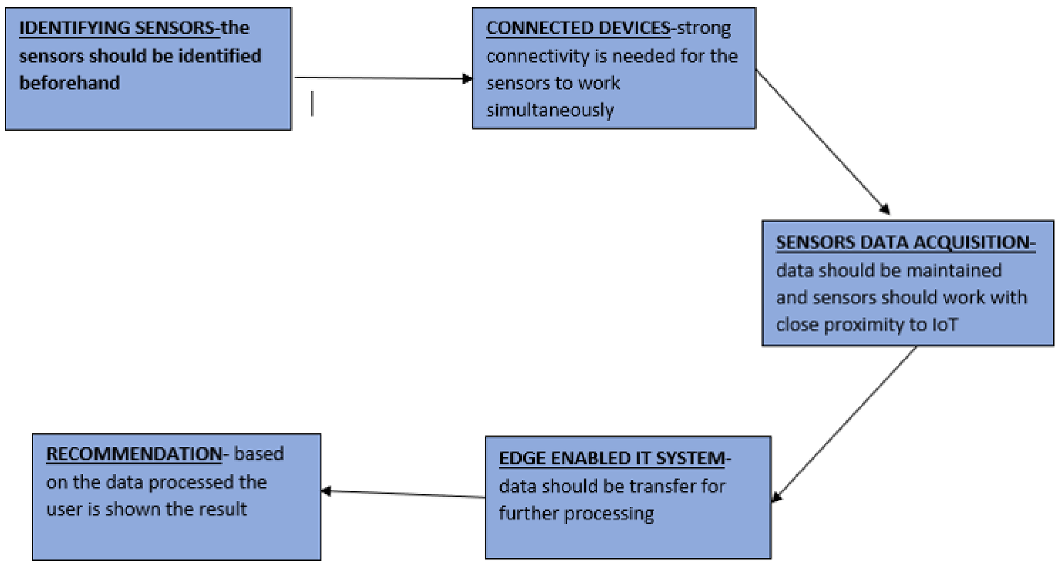
2) connected devices: The best thing about sensors is that it can convert the information it senses into a set of data which we can process further for analysis. Alternatively, it's important to start including sensors in the early stages of IoT architecture framework to get information that we need to process.

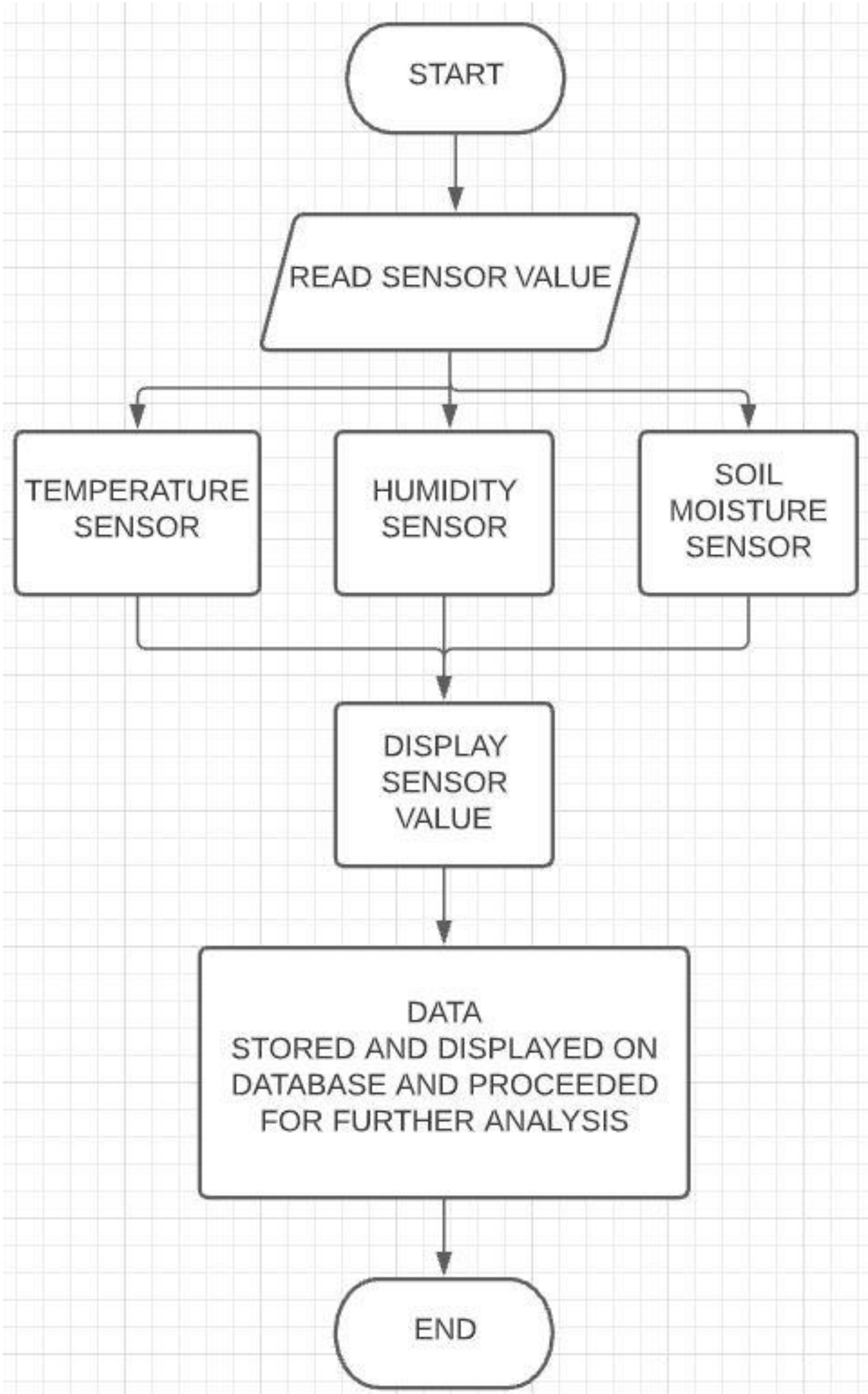
3) Sensor Data Acquisition: We understand at this stage that IoT deals with working with sensors and actuators in close proximity. Internet gateways and Data Acquisition Systems (DAS) plays an important role here as well. DAS aggregates output by connecting to the sensor network. On the other hand, Internet gateways work with Wi-Fi, wired LANs and perform further processing.
Basically, this stage helps to make data aggregated and digitized.

4) The appearance of edge enabled IT systems: Here, in this stage, we transfer the data that we prepared in stage 2 and expose them to the IT world. To be precise, the edge IT system performs enhanced analytics here along with pre-processing. Particularly, machine learning and visual representation.

5) Recommendations: Based on the gathered data, the project will recommend the necessary actions to the user.

FLOW CHART:




START
READ SENSOR VALUE
TEMPERATURE SENSOR
HUMIDITY SENSOR
SOIL MOISTURE SENSOR
DISPLAY SENSOR VALUE
DATA STORED AND DISPLAYED ON DATABASE AND PROCEEDED FOR FURTHER ANALYSIS
END

(c) Image Processing and Web Development

Dataset Resizer

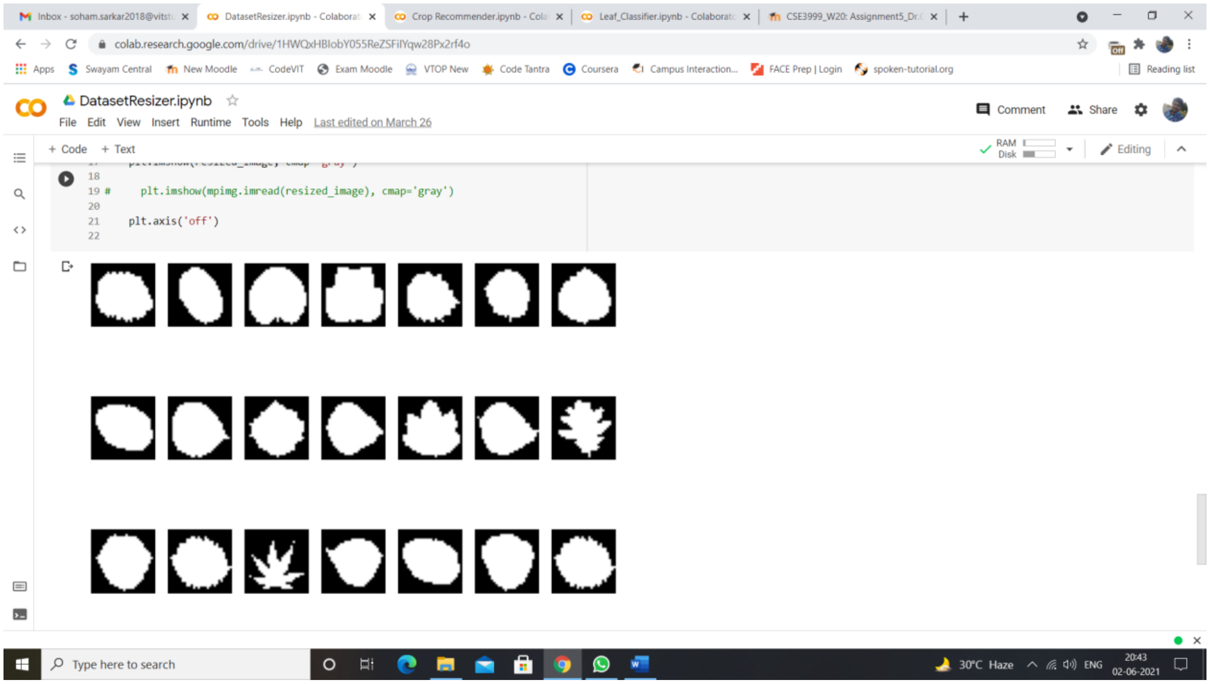
For this part out target was to create an image dataset of uniform pixel size from an unequal one, so that the uniformity of size helps while training the model in the subsequent part. The idea was that the user can set various cameras to different pixels (in order to conserve power) and still the dataset would be produced of same pixel size.

The dataset used here comprises of different leaf shapes in black-and-white. The images are of differing sizes. The pandas, numpy and matplotlib packages have been utilized for reading image files and printing them. The resize function of cv2 module has been used to resize the images to 28X28 size.

Original size

Resized images

This process works for differently shaped leaves but similarly shaped ones become too similar when resized. Also, when a coloured dataset is utilized the processing time becomes quite a lot.



Leaf classifier

We wanted to create a system where the installed cameras would take periodic photos of the various crops planted and then what species they belonged to would be automatically identified by our program. Thus, the user would not have to manually enter that information for each crop.
For this purpose, we utilized a dataset from Kaggle that had 4750 images of leaves belonging to 12 species of plants- ‘Fat Hen', 'Scentless Mayweed', 'Common wheat', 'Common Chickweed', 'Small-flowered Cranesbill', 'Maize', 'Loose Silky-bent', 'Sugar beet', 'Shepherds Purse', 'Cleavers', 'Charlock' and 'Black-grass'. This was the training set and there were 794 testing images separate.

An image of a leaf belonging to ‘Cleavers’ species

Mainly the pandas and tensorflow modules were used for the training and testing methods. The dataset was uploaded to Google drive and accessed through Google colab so that we could utilize its GPU to decrease training time. The neural network used for training to identify the images was a deep residual network: ResNet-50. We kept the number of hidden layers as 4, the learning rate as 0.001 and activation function as ReLU. The number of epochs was kept at 8 so that the training time did not exceed 3.5 hours. After training, a training accuracy of about 62% and validation accuracy of 76% came across 5 iterations. If we had increased the epochs to let us say 15, a better accuracy could have been achieved albeit at a training time of 7+ hours.

PLANT DISEASE IDENTIFICATION SYSTEM:

A major issue faced by farmers maintaining huge pieces of land is to identify ill crops so that other crops do not get infected and the yield can be saved. To do so manually is a very difficult task and requires lot of manpower and skill to identify and weed out such plants or to treat them.

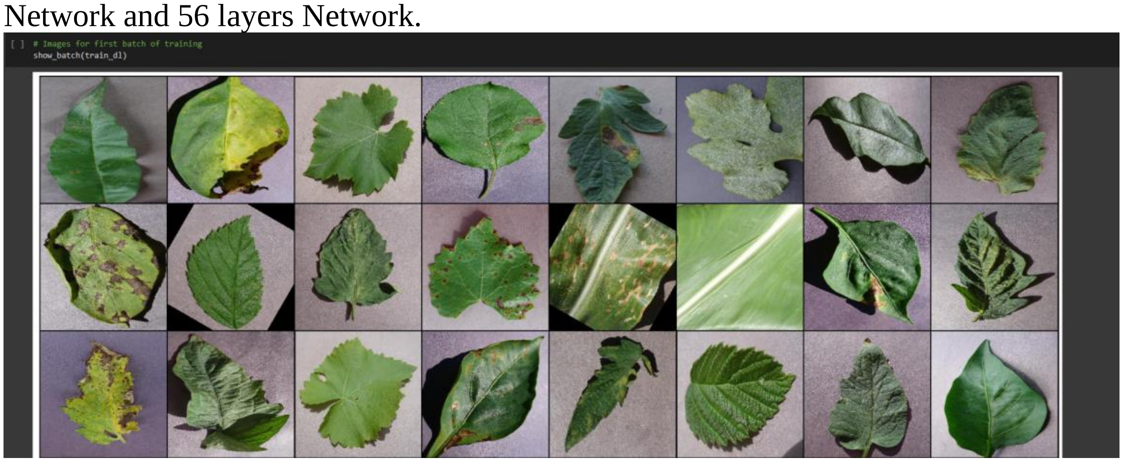
We took a huge dataset containing approximately 86000 images containing images of healthy plants and diseased plants.

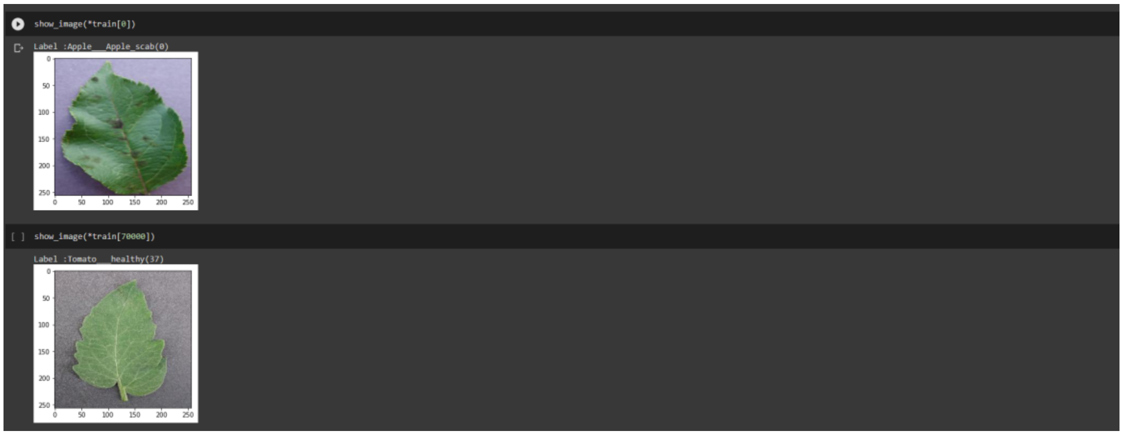
We used ResNet to carry out the training of our model

ResNet, short for Residual Network is a specific type of neural network that was introduced in 2015 by Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun in their paper "Deep Residual Learning for Image Recognition".

Need for ResNet:

Mostly in order to solve a complex problem, we stack some additional layers in the Deep Neural Networks which results in improved accuracy and performance. The intuition behind adding more layers is that these layers progressively learn more complex features. For example, in case of recognizing images, the first layer may learn to detect edges, the second layer may learn to identify textures and similarly the third layer can learn to detect objects and so on. But it has been found that there is a maximum threshold for depth with the traditional Convolutional neural network model. Here is a plot that describes error% on training and testing data for a 20-layer Network and 56 layers Network.





Residual Block

This problem of training very deep networks has been alleviated with the introduction of ResNet or residual networks and these Resnets are made up from Residual Blocks.

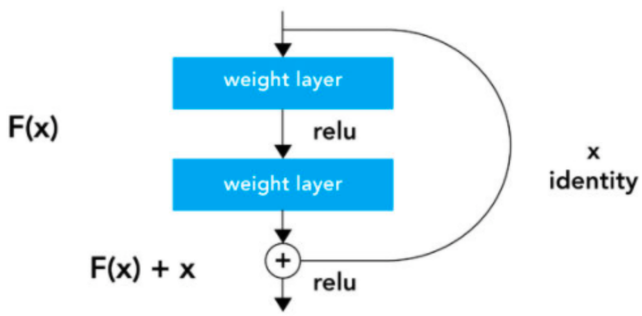


Figure 2. Residual learning: a building block.

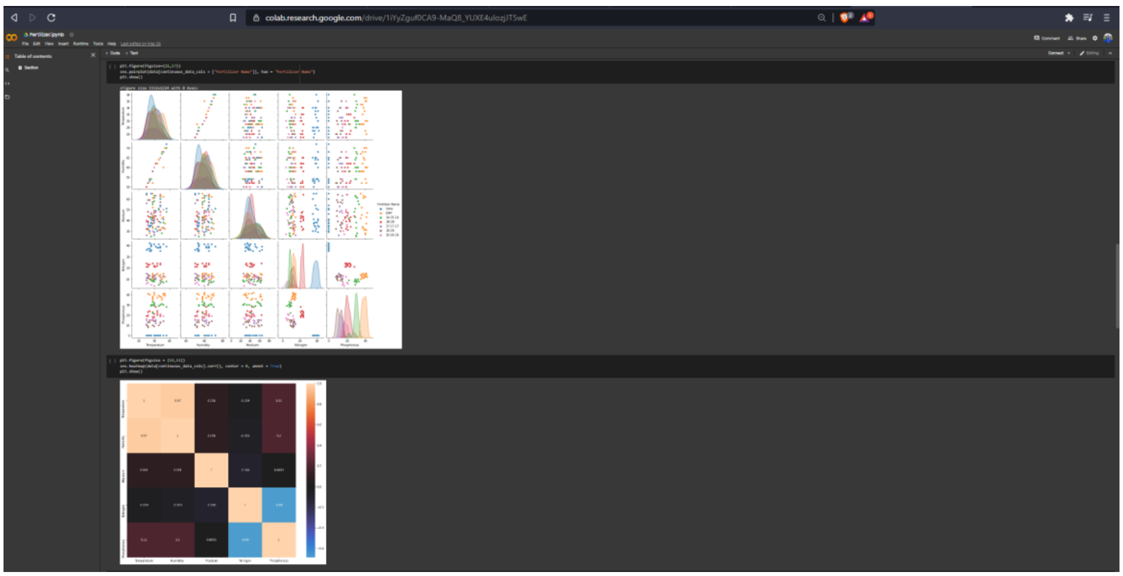
The very first thing we notice to be different is that there is a direct connection which skips some layers(may vary in different models) in between. This connection is called 'skip connection' and is the core of residual blocks. Due to this skip connection, the output of the layer is not the same now. Without using this skip connection, the input 'x' gets multiplied by the weights of the layer followed by adding a bias term.

Fertilizer Recommendation System:

The model which we were finally able to implement in our website was the fertilizer recommender. It uses a dataset which consists of values like – nitrogen, phosphorus and potassium and the crop the farmer desires to grow.

```
[ ] data.head()
```

	Temparature	Humidity	Moisture	Soil Type	Crop Type	Nitrogen	Potassium	Phosphorous	Fertilizer Name
0	26	52	38	4	3	37	0	0	6
1	29	52	45	2	8	12	0	36	5
2	34	65	62	0	1	7	9	30	1
3	32	62	34	3	9	22	0	20	4
4	28	54	46	1	6	35	0	0	6



We tried many different machine learning models such as an SVM classifier, XGBoost classifier, KNN Classifier.

These are the accuracies:

1)KNN : 76.4356%

2)SVM : 97.56334%

3)XGBoost : 98.70129%

We chose XGBoost and created its pickle file and uploaded in our model.

(d) Web Development and deployment

Methodology:

Harvestify is a web portal developed by us through which farmers can monitor their crops and use machine learning tools to find which crop is ideal for their field. The website uses machine learning algorithms and tools to identify which crop is ideal for the farmer to grow and under which condition. When the farmer visits the web portal, by inputting the nitrogen, phosphorus and potassium content of his field he can find out which crop is ideal for him to sow. Along with the chemical composition of the soil, he also has to input the pH level, rainfall in mm and the region of the country in which he lives.

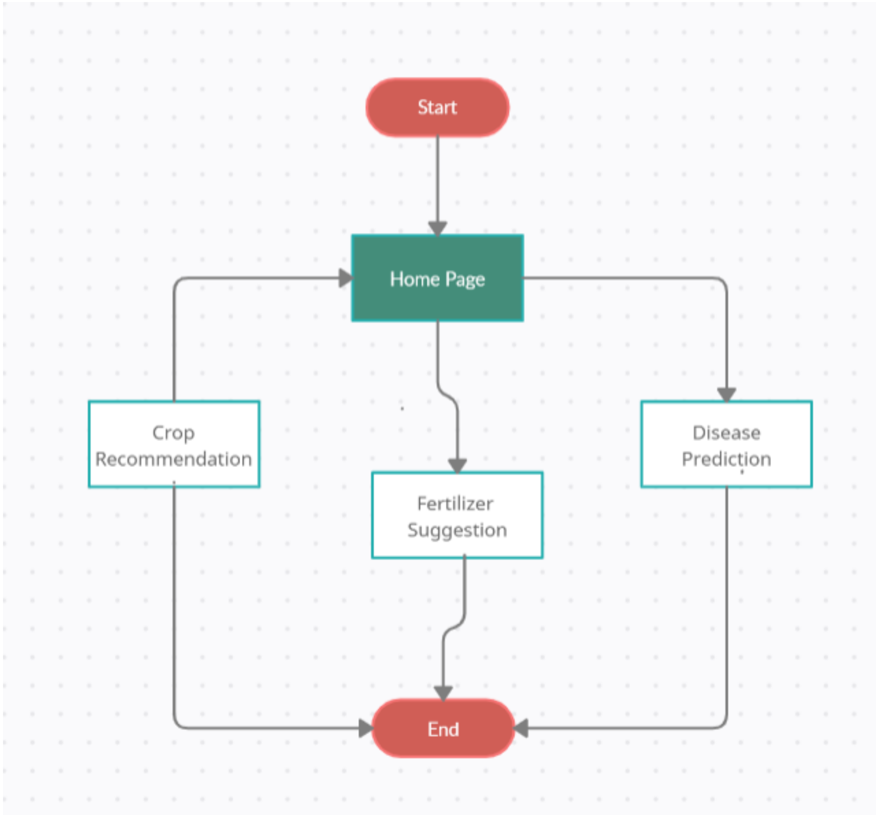
The use of the web portal is not just limited to monitoring the field and recommending the crop. Farmers can also use our application for finding which and how much fertilizers they need to use in order to yield an optimal crop. For all this they just have to input the nitrogen, phosphorus and potassium content of the soil along with the crop they are planning to grow.

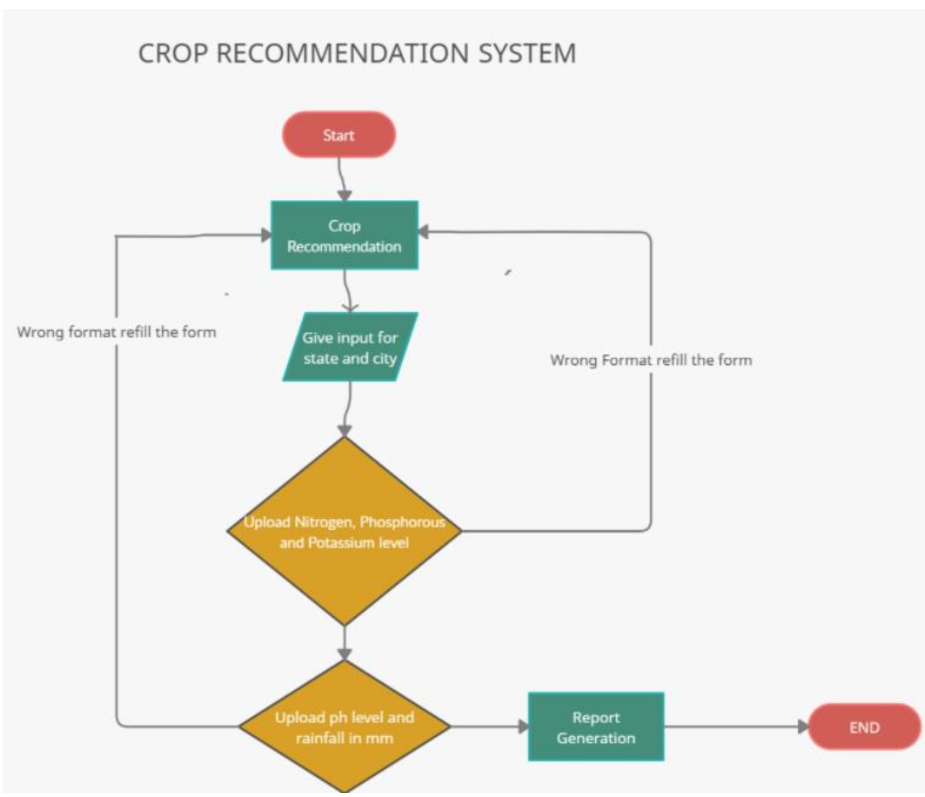
Functionalities till now include helping farmers to improve their yield. But these are not the only features of our portal. We also have a section under the name “Disease” where one can upload any kind of crop or fruit leaf and the website in turn tells about whether the plant is infected by any kind of disease. If so then we provide the proper information about the disease, it’s type and it’s cure.

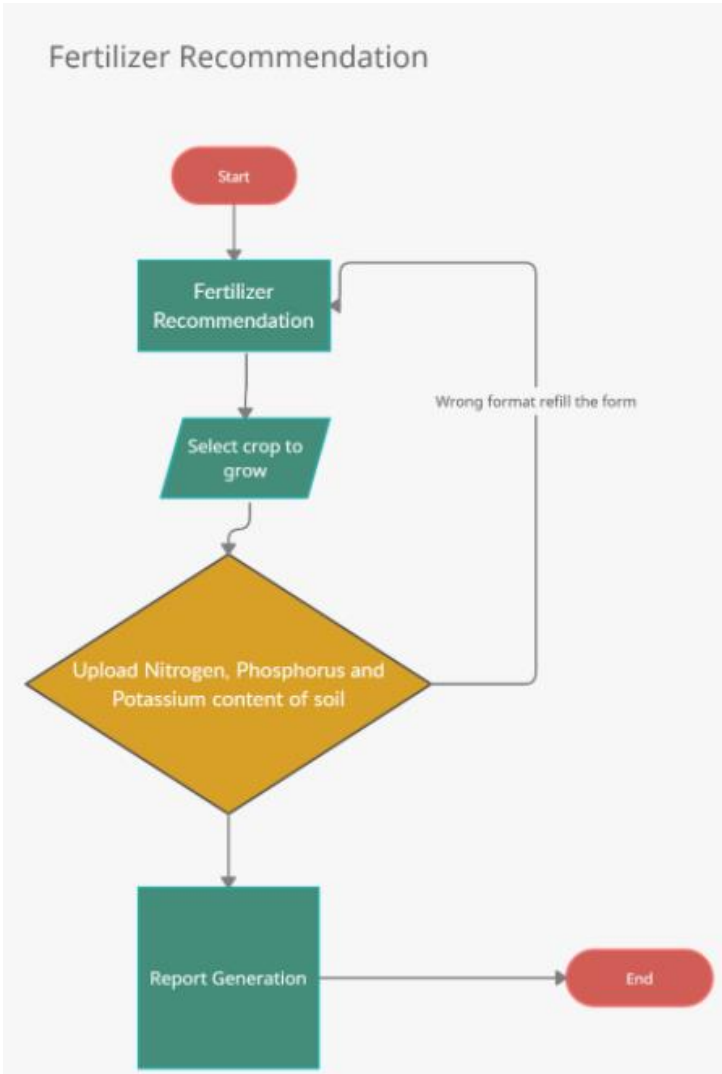
We have used HTML ,CSS for designing the front-end of our website and Flask , JAVASCRIPT to connect the backend.

Functionality In the context of the Website

- They allow the user to send a request for information to the server.
- They format the request so that the server can understand it.
- They format the response from the server in a way that the user can read it.


Start
Home Page
Crop
Recommendation
Fertilizer
Suggestion
Disease
Prediction
End


CROP RECOMMENDATION SYSTEM
Start
Crop
Recommendation
Wrong format refill the form
Give input for
state and city
Wrong Format refill the form
Upload Nitrogen, Phosphorous
and Potassium level
Upload ph level and
rainfall in mm
Report
Generation
END



Fertilizer Recommendation
Start
Fertilizer Recommendation
Wrong format refill the form
Select crop to grow
Upload Nitrogen, Phosphorus and Potassium content of soil
Report Generation
End

6. Results and Discussion

We successfully made a fully functional android app accompanied by a website with pickle machine learning models inbuilt.
The app has a profile page, home page, index page and displays the current Temperature, weather and other important details.
The website allows us to upload picture of a crop and get the result whether the crop is healthy or diseased.

The different models have been trained to perform the following tasks: -

1) Fertilizer Recommendation System.
2) Crop Recommendation System.
3) Plant Disease Detection.

CROP RECOMMENDATION SYSTEM

We divided the dataset into 80% training and 20% testing data and tested various models on it-

- Decision tree - 90% accuracy
- Naïve Bayes - 99.1% accuracy
- Random Forrest – 99.1% accuracy
- Logistic Regression – 95.3% accuracy

We chose Random Forrest as the model to implement finally based on the accuracy and created its pickle file.

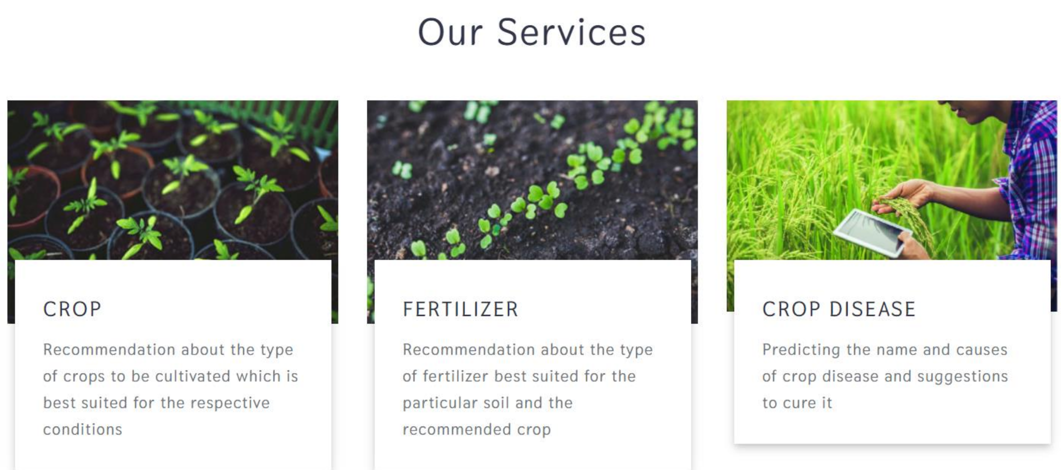
PLANT DISEASE

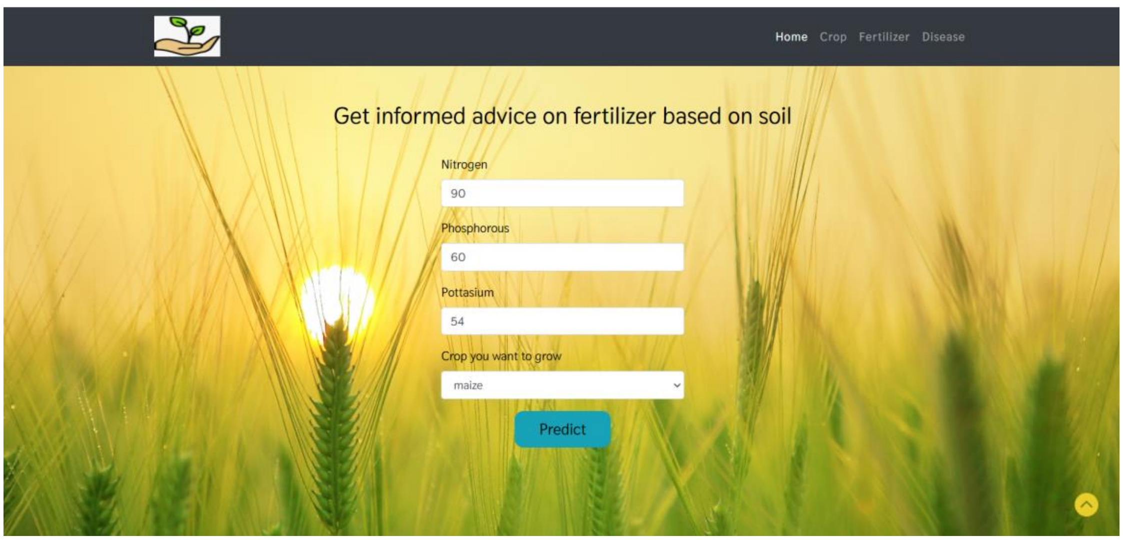
The model predicts with an amazing accuracy 98.92512%. We divided the images into batches of 32 images. Then we trained the model and tried different combinations of correction measures and epochs.
Finally, we found the optimum values and loaded the pickle file for deployment

About Us

IMPROVING AGRICULTURE, IMPROVING LIVES, CULTIVATING CROPS TO MAKE FARMERS INCREASE PROFIT.

We use state-of-the-art machine learning and deep learning technologies to help you guide through the entire farming process. Make informed decisions to understand the demographics of your area, understand the factors that affect your crop and keep them healthy for a super awesome successful yield.


Our Services
CROP
Recommendation about the type of crops to be cultivated which is best suited for the respective conditions
FERTILIZER
Recommendation about the type of fertilizer best suited for the particular soil and the recommended crop
CROP DISEASE
Predicting the name and causes of crop disease and suggestions to cure it


Home
Crop
Fertilizer
Disease
Get informed advice on fertilizer based on soil
Nitrogen
90
Phosphorous
60
Pottasium
54
Crop you want to grow
maize
Predict

The K value of your soil is high.

Please consider the following suggestions:

1. *Loosen the soil* deeply with a shovel, and water thoroughly to dissolve water-soluble potassium. Allow the soil to fully dry, and repeat digging and watering the soil two or three more times.
2. *Sift through the soil*, and remove as many rocks as possible, using a soil sifter. Minerals occurring in rocks such as mica and feldspar slowly release potassium into the soil slowly through weathering.
3. Stop applying potassium-rich commercial fertilizer. Apply only commercial fertilizer that has a '0' in the final number field. Commercial fertilizers use a three number system for measuring levels of nitrogen, phosphorous and potassium. The last number stands for potassium. Another option is to stop using commercial fertilizers all together and to begin using only organic matter to enrich the soil.
4. Mix crushed eggshells, crushed seashells, wood ash or soft rock phosphate to the soil to add calcium. Mix in up to 10 percent of organic compost to help amend and balance the soil.
5. Use NPK fertilizers with low K levels and organic fertilizers since they have low NPK values.
6. Grow a cover crop of legumes that will fix nitrogen in the soil. This practice will meet the soil's needs for nitrogen without increasing phosphorus or potassium.

7. Conclusion and Future Work

Our portal gives an accurate prediction about which crop should be grown and what fertilizer should be used in a particular field upon being given the nitrogen, phosphorus, sulphur, temperature, humidity, ph and rainfall values for a region. It also detects the type and cure of a disease upon uploading the image of the leaf of the crop.

FUTURE WORK:
1) We can create a custom dataset of local Indian crops to make the model more indigenous.
With better hardware one can expect much more faster results and smoother flow of data and operations.
2) This is just a prototype that we have prepared on a small scale level. We can expand the project by deploying real time sensors and cameras in a test field which will maintain a day to day data collection and analysis of data. This is the next step of our prototype. Farmers won't have to input the data but the sensors will send the data to the system (i.e. nitrogen/phosphorus/potassium content and also monitor crops via cameras) and the farmer will be notified in case if there is an action to be taken.

8. REFERENCES

[1] Smart Irrigation and Crop Suggestion Using Raspberry-Pi. International Journal of Scientific Research in Science and Technology- Kakade, K.R., Pisal, A.R., Chavanss, A.V. and Khedkar, S.B. (2017)

[2] Smart Irrigation System. International Journal of Scientific Research in Science and Technology- Akubattin, V.L., Bansode, A.P., Ambre, T., Kachroo, A. and SaiPrasad, P. (2016)

[3] Arduino Based Smart Drip Irrigation System Using Internet of Things. International Journal of Engineering Science and Computing- Parameswaran, G. and Sivaprasath, K. (2016)

[4] Smart irrigation system for smart farming- Pedro Alexander Tenezaca Sari, Christian David Piedra García, Alberto Steven Godoy Mendía, Daniel Felipe Merchán Piedra, Edisson Fernando Patiño Zaruma, Irene Priscila Cedillo Orellana

[5] A Study on Smart Irrigation System Using IoT for Surveillance of CropField - Ashwini B V

[6]Smart Irrigation System: A Water Management Procedure - Olugbenga Kayode Ogidan, Abiodun Emmanuel Onile, Oluwabukola Grace Adegboro

[7] Development of Smart Irrigation System- Aashu Bedrae R. , Jayalakshmi , Nayana , Swetha , Shridhara

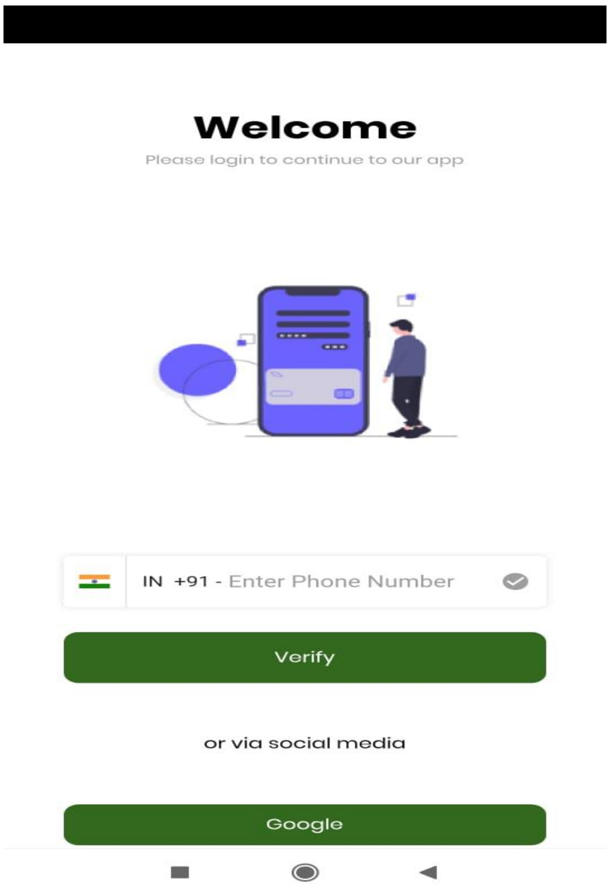
[8]IOT based Smart Irrigation System International Journal of Computer Applications (0975 – 8887) Volume 159 – No 8, February 2017- Srishti Rawal, Department of Computer Science, VIT University

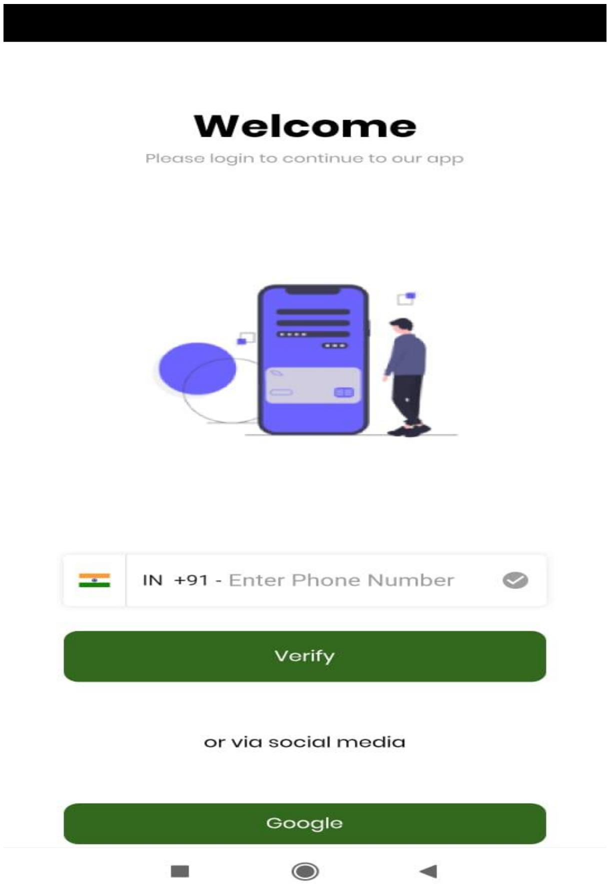
[9]Smart Irrigation Control System International Journal of Environmental Research and Development ISSN 2249-3131 Volume 4, Number 4 (2014) -Mr. Deepak Kumar Roy and Mr. Murtaza Hassan Ansari Amity School of Engineering and Technology, Noida

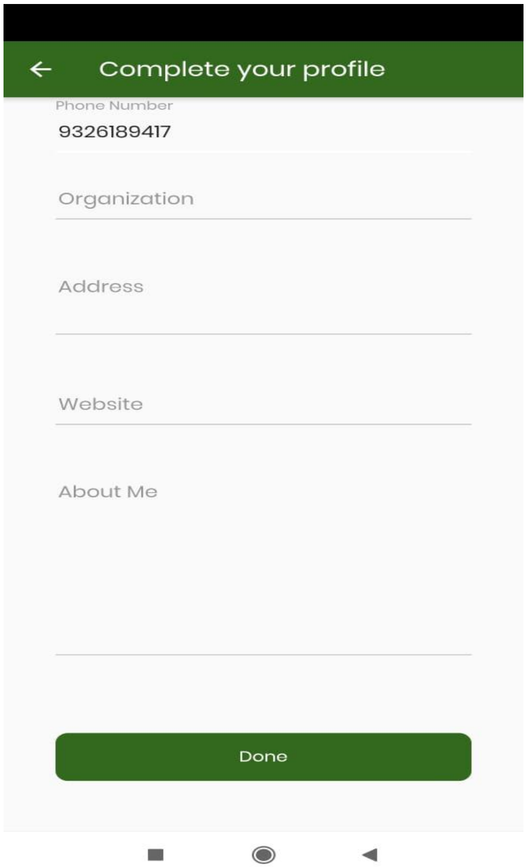
[10]Soil moisture monitoring using IoT enabled arduino sensors with neural networks for improving soil management for farmers and predict seasonal rainfall for planning future harvest in North Karnataka - India 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) - Suhas Athani, CH Tejeshwar, Mayur M Patil, Priyadarshini Patil, Rahul Kulkarni

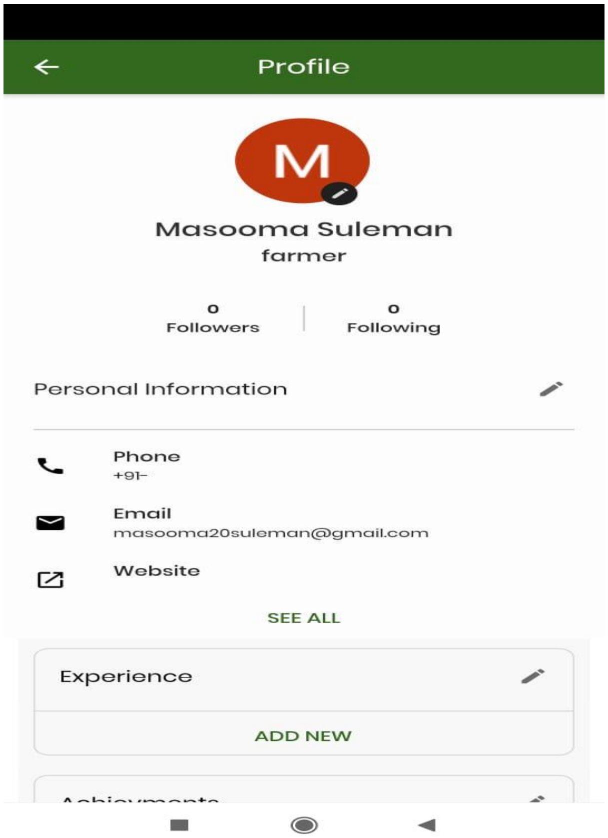
APPENDIX <Sample code, snapshot etc.>

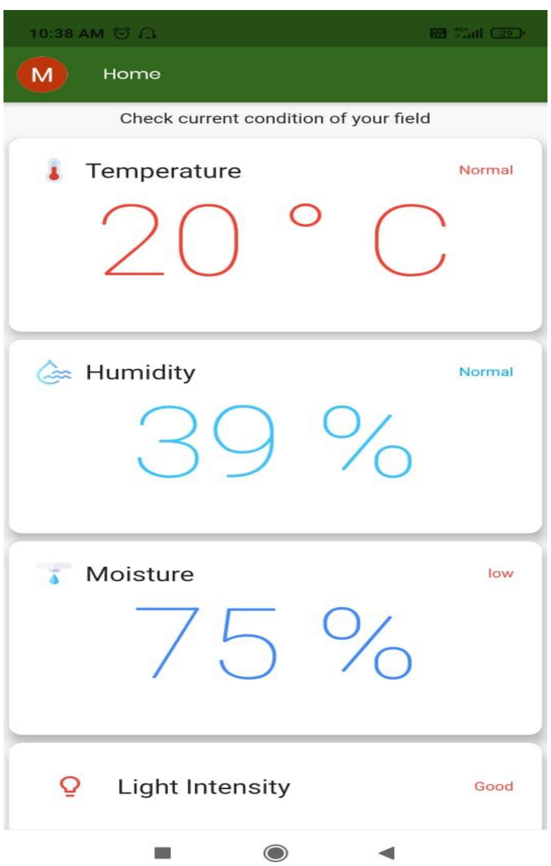
ANDROID APP:

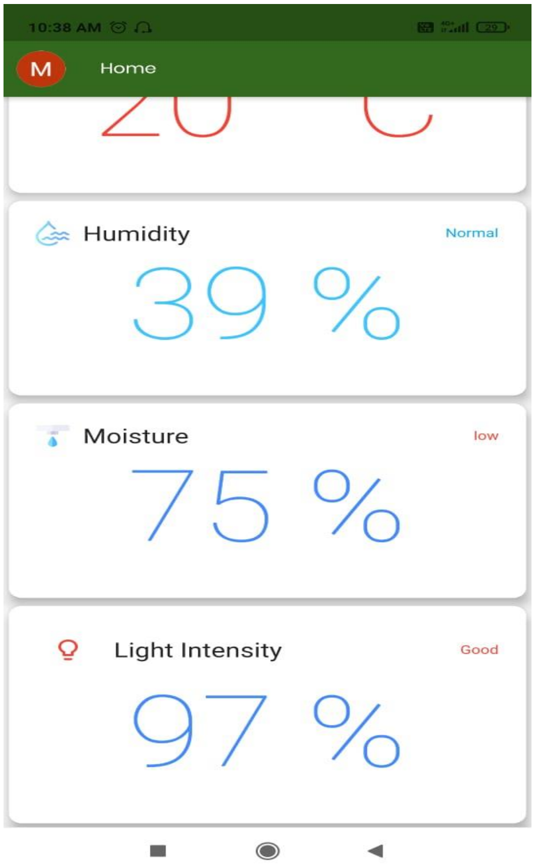



Welcome
Please login to continue to our app
IN +91 - Enter Phone Number
Verify
or via social media
Google

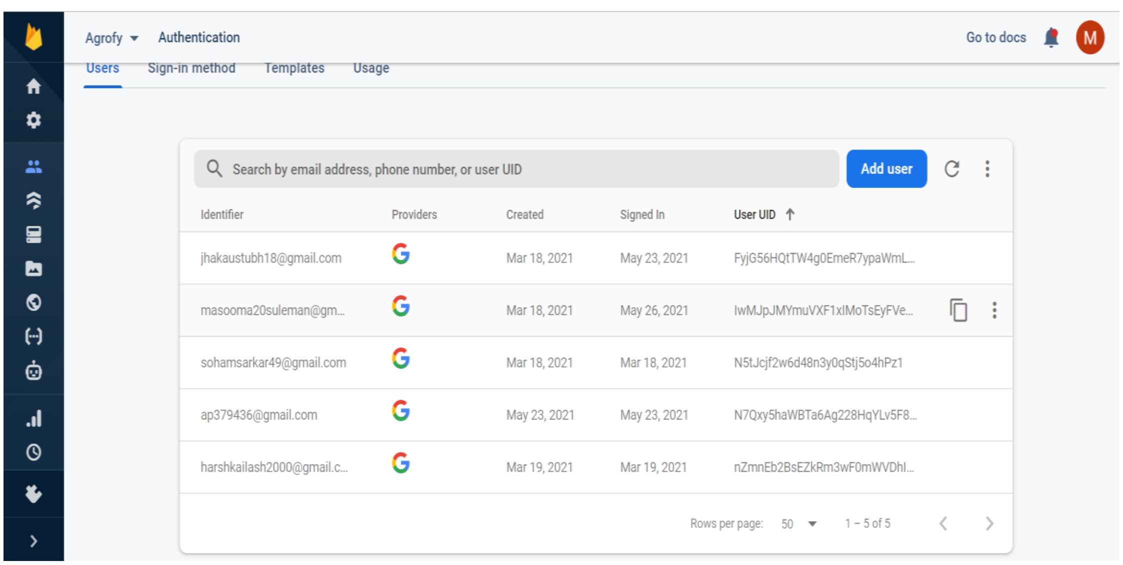

Complete your profile
Phone Number
9326189417
Organization
Address
Website
About Me
Done

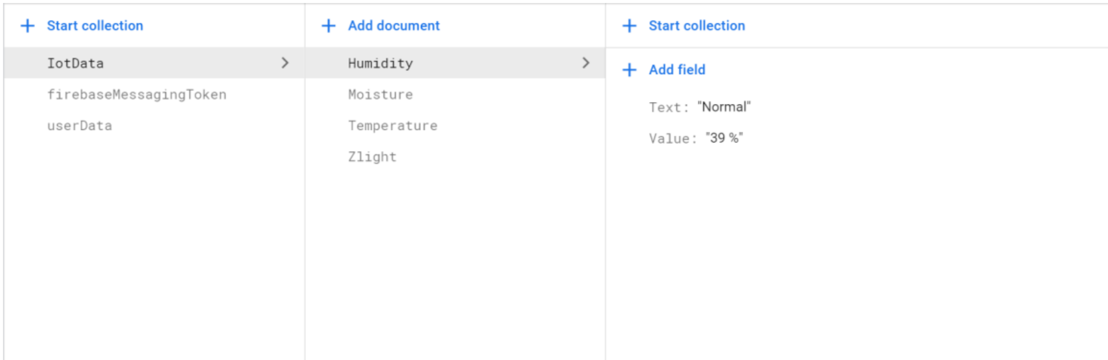

Profile
M
Masooma Suleman
farmer
0
Followers
0
Following
Personal Information
Phone
+91-
Email
masooma20suleman@gmail.com
Website
SEE ALL
Experience
ADD NEW


10:38 AM
M
Home
Check current condition of your field
Temperature
Normal
20 ° C
Humidity
Normal
39 %
Moisture
low
75 %
Light Intensity
Good

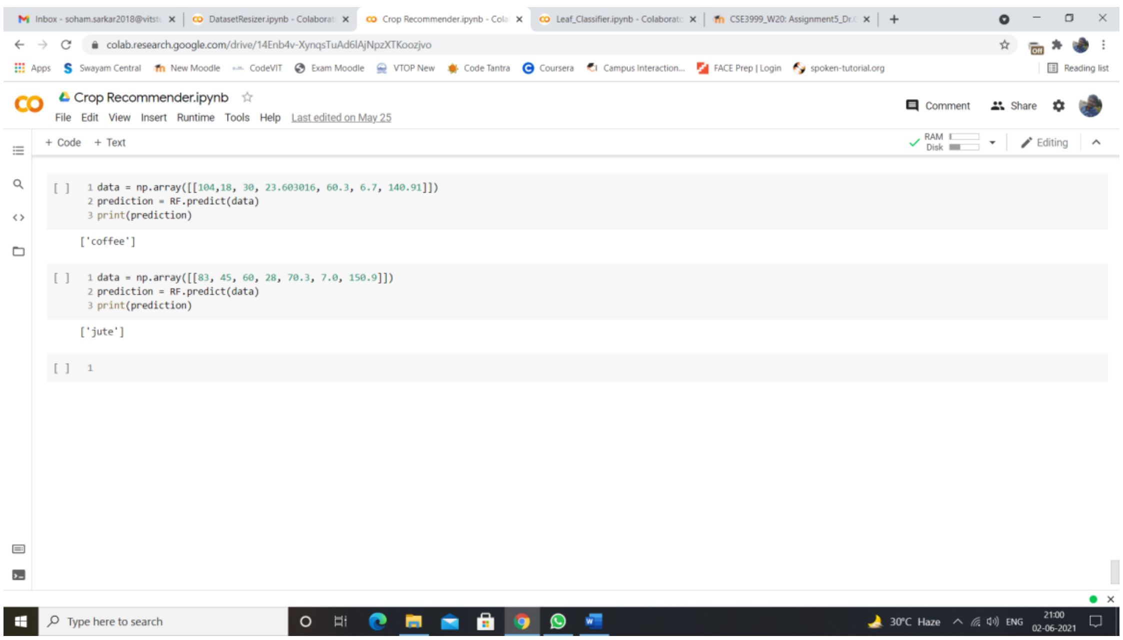

10:38 AM
M
Home
Humidity
Normal
39 %
Moisture
low
75 %
Light Intensity
Good
97 %

BACKEND:


Agrofy
Authentication
Go to docs
M
Users
Sign-in method
Templates
Usage
Search by email address, phone number, or user UID
Add user
Identifier
Providers
Created
Signed In
User UID
jhakaustubh18@gmail.com
Mar 18, 2021
May 23, 2021
FyjG56HQtTW4g0EmeR7ypaWmL...
masooma20suleman@gm...
Mar 18, 2021
May 26, 2021
IwMJpJMYmuVXF1xIMoTsEyFVe...
sohamsarkar49@gmail.com
Mar 18, 2021
Mar 18, 2021
N5tJcjf2w6d48n3y0qStj5o4hPz1
ap379436@gmail.com
May 23, 2021
May 23, 2021
N7Qxy5haWBTa6Ag228HqYLv5F8...
harshkailash2000@gmail.c...
Mar 19, 2021
Mar 19, 2021
nZmnEb2BsEZkRm3wF0mWVDhl...
Rows per page:
50
1 – 5 of 5



SAMPLE CODE FOR MACHINE LEARNING:



```
data_dir = "/content/drive/MyDrive/Dataset"
train_dir = data_dir + "/train"
valid_dir = data_dir + "/valid"
diseases = os.listdir(train_dir)
```

```
print(diseases)
```

```
['Tomato___Tomato_Yellow_Leaf_Curl_Virus', 'Tomato___Tomato_mosaic_virus', 'Tomato___Septoria_leaf_spot', 'Tomato___Early_blight', 'Strawberry___Leaf_scorch', 'Tomato___healthy', 'Tomato___Leaf_Mold', 'Strawberr
```

```
print("Total disease classes are: {}".format(len(diseases)))
```

```
Total disease classes are: 38
```

```
plants = []
NumberOfDiseases = 0
for plant in diseases:
    if plant.split('___')[0] not in plants:
        plants.append(plant.split('___')[0])
    if plant.split('___')[1] != 'healthy':
        NumberOfDiseases += 1
```

```
# unique plants in the dataset
print(f"Unique Plants are: \n{plants}")
```

```
Unique Plants are:
['Tomato', 'Strawberry', 'Pepper,_bell', 'Potato', 'Raspberry', 'Squash', 'Soybean', 'Peach', 'Corn_(maize)', 'Grape', 'Cherry_(including_sour)', 'Orange', 'Blueberry', 'Apple']
```

```
print("Number of plants: {}".format(len(plants)))
```

```
Number of plants: 14
```

```
# Number of images for each disease
nums = {}
for disease in diseases:
    nums[disease] = len(os.listdir(train_dir + '/' + disease))

# converting the nums dictionary to pandas dataframe passing index as plant name and number of images as column
```

```
# for calculating the accuracy
def accuracy(outputs, labels):
    _, preds = torch.max(outputs, dim=1)
    return torch.tensor(torch.sum(preds == labels).item() / len(preds))


# base class for the model
class ImageClassificationBase(nn.Module):

    def training_step(self, batch):
        images, labels = batch
        out = self(images)                  # Generate predictions
        loss = F.cross_entropy(out, labels) # Calculate loss
        return loss

    def validation_step(self, batch):
        images, labels = batch
        out = self(images)                   # Generate prediction
        loss = F.cross_entropy(out, labels)  # Calculate loss
        acc = accuracy(out, labels)          # Calculate accuracy
        return {"val_loss": loss.detach(), "val_accuracy": acc}

    def validation_epoch_end(self, outputs):
        batch_losses = [x["val_loss"] for x in outputs]
        batch_accuracy = [x["val_accuracy"] for x in outputs]
        epoch_loss = torch.stack(batch_losses).mean()       # Combine loss
        epoch_accuracy = torch.stack(batch_accuracy).mean()
        return {"val_loss": epoch_loss, "val_accuracy": epoch_accuracy} # Combine accuracies

    def epoch_end(self, epoch, result):
        print("Epoch [{}], last_lr: {:.5f}, train_loss: {:.4f}, val_loss: {:.4f}, val_acc: {:.4f}".format(
            epoch, result['lrs'][-1], result['train_loss'], result['val_loss'], result['val_accuracy']))
```

```
# Architecture for training

# convolution block with BatchNormalization
def ConvBlock(in_channels, out_channels, pool=False):
    layers = [nn.Conv2d(in_channels, out_channels, kernel_size=3, padding=1),
             nn.BatchNorm2d(out_channels),
             nn.ReLU(inplace=True)]
    if pool:
        layers.append(nn.MaxPool2d(4))
    return nn.Sequential(*layers)


# resnet architecture
class ResNet9(ImageClassificationBase):
    def __init__(self, in_channels, num_diseases):
        super().__init__()

        self.conv1 = ConvBlock(in_channels, 64)
        self.conv2 = ConvBlock(64, 128, pool=True) # out_dim : 128 x 64 x 64
        self.res1 = nn.Sequential(ConvBlock(128, 128), ConvBlock(128, 128))

        self.conv3 = ConvBlock(128, 256, pool=True) # out_dim : 256 x 16 x 16
        self.conv4 = ConvBlock(256, 512, pool=True) # out_dim : 512 x 4 x 44
        self.res2 = nn.Sequential(ConvBlock(512, 512), ConvBlock(512, 512))
```

TRAINING THE MODEL

```
# for training
@torch.no_grad()
def evaluate(model, val_loader):
    model.eval()
    outputs = [model.validation_step(batch) for batch in val_loader]
    return model.validation_epoch_end(outputs)


def get_lr(optimizer):
    for param_group in optimizer.param_groups:
        return param_group['lr']


def fit_OneCycle(epochs, max_lr, model, train_loader, val_loader, weight_decay=0,
                grad_clip=None, opt_func=torch.optim.SGD):
    torch.cuda.empty_cache()
    history = []

    optimizer = opt_func(model.parameters(), max_lr, weight_decay=weight_decay)
    # scheduler for one cycle learniing rate
    sched = torch.optim.lr_scheduler.OneCycleLR(optimizer, max_lr, epochs=epochs, steps_per_epoch=len(train_loader))


    for epoch in range(epochs):
        # Training
        model.train()
        train_losses = []
        lrs = []
        for batch in train_loader:
            loss = model.training_step(batch)
            train_losses.append(loss)
            loss.backward()

            # gradient clipping
            if grad_clip:
                nn.utils.clip_grad_value_(model.parameters(), grad_clip)

            optimizer.step()
            optimizer.zero_grad()

            # recording and updating learning rates
            lrs.append(get_lr(optimizer))
            sched.step()


        # validation
        result = evaluate(model, val_loader)
        result['train_loss'] = torch.stack(train_losses).mean().item()
        result['lrs'] = lrs
        model.epoch_end(epoch, result)
        history.append(result)

    return history
```

Code snippet-

1)crop recommendation

tate_Name	District_Name	Crop_Year	Season	Crop	Area	Produ
ndaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254.00	2000.
ndaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2.00	1.00
ndaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102.00	321.0
ndaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176.00	641.0
ndaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720.00	165.0
ndaman and Nicobar Islands	NICOBARS	2000	Whole Year	Coconut	18168.00	65100
ndaman and Nicobar Islands	NICOBARS	2000	Whole Year	Dry ginger	36.00	100.0
ndaman and Nicobar Islands	NICOBARS	2000	Whole Year	Sugarcane	1.00	2.00
ndaman and Nicobar Islands	NICOBARS	2000	Whole Year	Sweet potato	5.00	15.00
ndaman and Nicobar Islands	NICOBARS	2000	Whole Year	Tapioca	40.00	169.0
ndaman and Nicobar Islands	NICOBARS	2001	Kharif	Arecanut	1254.00	2061.
ndaman and Nicobar Islands	NICOBARS	2001	Kharif	Other Kharif pulses	2.00	1.00
ndaman and Nicobar Islands	NICOBARS	2001	Kharif	Rice	83.00	300.0
ndaman and Nicobar Islands	NICOBARS	2001	Whole Year	Cashewnut	719.00	192.0
ndaman and Nicobar Islands	NICOBARS	2001	Whole Year	Coconut	18190.00	64430
ndaman and Nicobar Islands	NICOBARS	2001	Whole Year	Dry ginger	46.00	100.0
ndaman and Nicobar Islands	NICOBARS	2001	Whole Year	Sugarcane	1.00	1.00
ndaman and Nicobar Islands	NICOBARS	2001	Whole Year	Sweet potato	11.00	33.00
ndaman and Nicobar Islands	NICOBARS	2002	Kharif	Rice	189.20	510.8
ndaman and Nicobar Islands	NICOBARS	2002	Whole Year	Arecanut	1258.00	2083.
ndaman and Nicobar Islands	NICOBARS	2002	Whole Year	Banana	213.00	1278.
ndaman and Nicobar Islands	NICOBARS	2002	Whole Year	Black pepper	63.00	13.50
ndaman and Nicobar Islands	NICOBARS	2002	Whole Year	Cashewnut	719.00	208.0
ndaman and Nicobar Islands	NICOBARS	2002	Whole Year	Coconut	18240.00	67490
ndaman and Nicobar Islands	NICOBARS	2002	Whole Year	Dry chillies	413.00	28.80

loading crop recommendation model

```
# Loading crop recommendation model

crop_recommendation_model_path = 'models/RandomForest.pkl'
crop_recommendation_model = pickle.load(
    open(crop_recommendation_model_path, 'rb'))
```

B.to take values from html page and use it for prediction by inserting into our saved ml model

```
@ app.route('/crop-predict', methods=['POST'])
def crop_prediction():
    title = 'agrofy - Crop Recommendation'

    if request.method == 'POST':
        N = int(request.form['nitrogen'])
        P = int(request.form['phosphorous'])
        K = int(request.form['pottasium'])
        ph = float(request.form['ph'])
        rainfall = float(request.form['rainfall'])

        # state = request.form.get("stt")
        city = request.form.get("city")

        if weather_fetch(city) != None:
            temperature, humidity = weather_fetch(city)
            data = np.array([[N, P, K, temperature, humidity, ph, rainfall]])
            my_prediction = crop_recommendation_model.predict(data)
            final_prediction = my_prediction[0]

            return render_template('crop-result.html', prediction=final_prediction, title=title)

        else:

            return render_template('try_again.html', title=title)
```

2.fertilizer

taking values from html form for prediction

```
@ app.route('/fertilizer-predict', methods=['POST'])
def fert_recommend():
    title = 'agrofy - Fertilizer Suggestion'

    crop_name = str(request.form['cropname'])
    N = int(request.form['nitrogen'])
    P = int(request.form['phosphorous'])
    K = int(request.form['pottasium'])
    # ph = float(request.form['ph'])

    df = pd.read_csv('Data/fertilizer.csv')

    nr = df[df['Crop'] == crop_name]['N'].iloc[0]
    pr = df[df['Crop'] == crop_name]['P'].iloc[0]
    kr = df[df['Crop'] == crop_name]['K'].iloc[0]

    n = nr - N
    p = pr - P
    k = kr - K
    temp = {abs(n): "N", abs(p): "P", abs(k): "K"}
    max_value = temp[max(temp.keys())]
    if max_value == "N":
        if n < 0:
            key = 'NHigh'
        else:
            key = "Nlow"
    elif max_value == "P":
        if p < 0:
            key = 'PHigh'
        else:
            key = "Plow"
    else:
```

```python
    else:
        if k < 0:
            key = 'KHigh'
        else:
            key = "Klow"

    response = Markup(str(fertilizer_dic[key]))

    return render_template('fertilizer-result.html', recommendation=response, title=title)

# render disease prediction result page
```

disease prediction

```python
disease_classes = ['Apple___Apple_scab',
                   'Apple___Black_rot',
                   'Apple___Cedar_apple_rust',
                   'Apple___healthy',
                   'Blueberry___healthy',
                   'Cherry_(including_sour)___Powdery_mildew',
                   'Cherry_(including_sour)___healthy',
                   'Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot',
                   'Corn_(maize)___Common_rust_',
                   'Corn_(maize)___Northern_Leaf_Blight',
                   'Corn_(maize)___healthy',
                   'Grape___Black_rot',
                   'Grape___Esca_(Black_Measles)',
                   'Grape___Leaf_blight_(Isariopsis_Leaf_Spot)',
                   'Grape___healthy',
                   'Orange___Haunglongbing_(Citrus_greening)',
                   'Peach___Bacterial_spot',
                   'Peach___healthy',
                   'Pepper,_bell___Bacterial_spot',
                   'Pepper,_bell___healthy',
                   'Potato___Early_blight',
                   'Potato___Late_blight',
                   'Potato___healthy',
                   'Raspberry___healthy',
                   'Soybean___healthy',
                   'Squash___Powdery_mildew',
                   'Strawberry___Leaf_scorch',
                   'Strawberry___healthy',
                   'Tomato___Bacterial_spot',
                   'Tomato___Early_blight',
                   'Tomato___Late_blight',
                   'Tomato___Leaf_Mold',
```

Dictionary to match disease class

```
disease_dic = {
    'Apple___Apple_scab': """ <b>Crop</b>: Apple <br/>Disease: Apple Scab<br/>
        <br/> Cause of disease:

        <br/><br/> 1. Apple scab overwinters primarily in fallen leaves and in the soil. Disease development is favored b

        <br/> 2. Fungal spores are carried by wind, rain or splashing water from the ground to flowers, leaves or fruit._
        <br/><br/> How to prevent/cure the disease <br/>
        <br/>1. Choose resistant varieties when possible.

        <br/>2. Rake under trees and destroy infected leaves to reduce the number of fungal spores available to start the

        <br/>3. Water in the evening or early morning hours (avoid overhead irrigation) to give the leaves time to dry ou
        <br/>4. Spread a 3- to 6-inch layer of compost under trees, keeping it away from the trunk, to cover soil and pre

    'Apple___Black_rot': """ <b>Crop</b>: Apple <br/>Disease: Black Rot<br/>
        <br/> Cause of disease:

        <br/><br/>Black rot is caused by the fungus Diplodia seriata (syn Botryosphaeria obtusa).The fungus can infect de
        <br/><br/> How to prevent/cure the disease <br/>
        <br/>1. Prune out dead or diseased branches.

        <br/>2. Prune out dead or diseased branches.

        <br/>3. Remove infected plant material from the area.
        <br/>4. Remove infected plant material from the area.
        <br/>5. Be sure to remove the stumps of any apple trees you cut down. Dead stumps can be a source of spores.""",

    'Apple___Cedar_apple_rust': """ <b>Crop</b>: Apple <br/>Disease: Cedar Apple Rust<br/>
        <br/> Cause of disease:

        <br/><br/>Cedar apple rust (Gymnosporangium juniperi-virginianae) is a fungal disease that depends on two species
```

taking input for disease prediction

```python
@app.route('/disease-predict', methods=['GET', 'POST'])
def disease_prediction():
    title = 'agrofy - Disease Detection'

    if request.method == 'POST':
        if 'file' not in request.files:
            return redirect(request.url)
        file = request.files.get('file')
        if not file:
            return render_template('disease.html', title=title)
        try:
            img = file.read()

            prediction = predict_image(img)

            prediction = Markup(str(disease_dic[prediction]))
            return render_template('disease-result.html', prediction=prediction, title=title)
        except:
            pass
    return render_template('disease.html', title=title)
```