Project Report Format

Source Code, GitHub & Project Demo Link

1. INTRODUCTION

1.1 Project Overview:

Crop production in India is one of the most important sources of income and India is one of the top countries to produce crops. Digital Farming and Precision Agriculture allow precise utilization of inputs like seed, water, pesticides, and fertilizers at the right time for the crop to maximize productivity, quality, and yields. Most farmers practice traditional farming patterns to decide on crops to be cultivated in a field. Based on analytics farmers can take better decisions for healthy crop production.

1.2 Purpose:

To build a system that helps Farmers to make decisions on what crop to sow in a particular season to make maximum yield.

2. LITERATURE SURVEY

2.1 Existing Problem:

Weather-Based Crop Prediction in India Using Big Data Analytics worked on collecting and analyzing data, which will help the farmers improve the productivity of their crops. Firstly, Akhilesh Kumar pre-processed the data and used the MapReduce framework then K means Clustering is employed. The design constraints a program's ability to process smaller data items.

Assimilating remote sensing data into a crop model improves winter wheat yield estimation based on regional irrigation data study involves analysis regarding different assimilation state variables and algorithms. The PSO produced the highest accuracy for yield estimation.PSO algorithms often converge to some local optimization.

In this Estimation of crop production using machine learning techniques Historical production and meteorological data were collected and processed for analysis and applying ML algorithms. The ML Techniques used were decision trees, random forest, support vector regressor, and gradient boosting. It will take more time for training-time complexity as the input increases.

Estimation of Crop Yield From Combined Optical and SAR Imagery Using Gaussian Kernel Regression study proposed Gaussian kernel regression for rice yield estimation. The performance depends on the amount of available ground truth data. Yeshanbele loses efficiency in high-dimensional spaces.

Crop Yield Estimation and Interpretability With Gaussian Processes introduced the use of Gaussian processes (GPs) for the evcThe proposed methodology combines information on the canopy, biomass, soil, and plant water with the atmospheric variables. Makes use of the whole features/samples.

2.2 References:

1. Gupta, R., Sharma, A.K., Garg, O., Modi, K., Kasim, S., Baharum, Z., Mahdin, H. and Mostafa, S.A., 2021. WB-CPI: Weather-based crop prediction in India using big data analytics. *IEEE Access*, *9*, pp.137869-137885.

2. Jin, N., Tao, B., Ren, W., He, L., Zhang, D., Wang, D. and Yu, Q., 2022. Assimilating remote sensing data into a crop model improves winter wheat yield estimation based on regional irrigation data. *Agricultural Water Management*, *266*, p.107583.

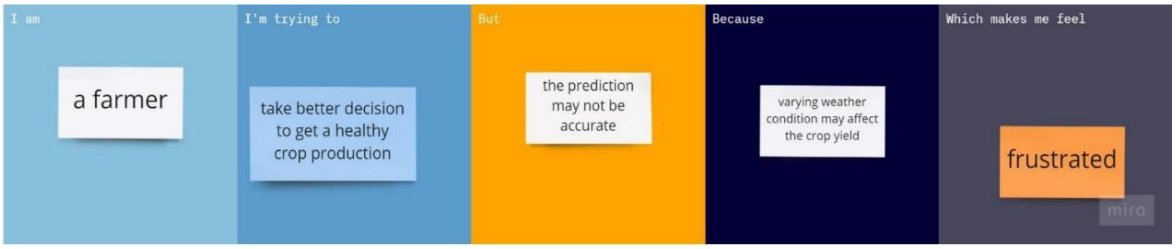
3. Mahajan, J., Banal, K. and Mahajan, S., 2021. Estimation of crop production using machine learning techniques: a case study of J&K. *International Journal of Information Technology*, *13*(4), pp.1441-1448.

4. Alebele, Y., Wang, W., Yu, W., Zhang, X., Yao, X., Tian, Y., Zhu, Y., Cao, W. and Cheng, T., 2021. Estimation of Crop Yield From Combined Optical and SAR Imagery Using Gaussian Kernel Regression. *IEEE Journal of Selected Topics in Applied EarthObservations and Remote Sensing*, *14*, pp.10520-10534.

5. Martínez-Ferrer, L., Piles, M. and Camps-Valls, G., 2020. Crop yield estimation and interpretability with Gaussian processes. *IEEE Geoscience and Remote Sensing Letters*, *18*(12), pp.2043-2047.

2.3 Problem Statement Definition:

To Analyse the past harvesting yield which enables the farm practices to be modified throughout the growing season, with the potential to increase the final yield. Thus, assisting the cultivators in prior to make valuable decisions and achieve maximum yield.

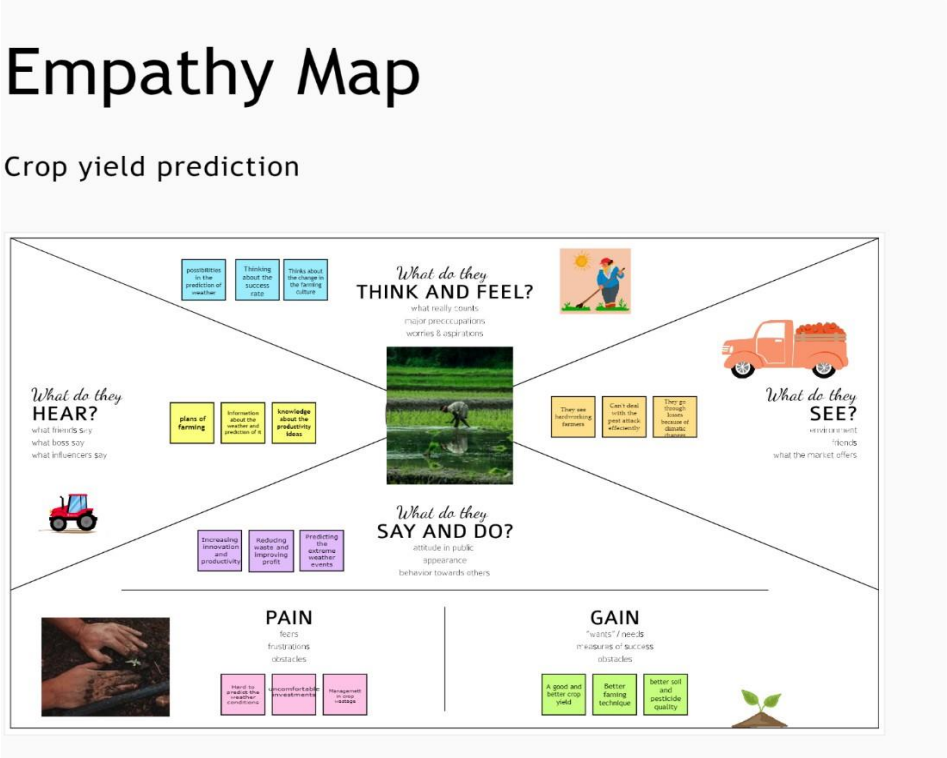


Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A farmer	Take better decision to get a healthy crop production	The prediction may not be accurate	Varying weather condition may affect the crop yield	frustrated

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas:

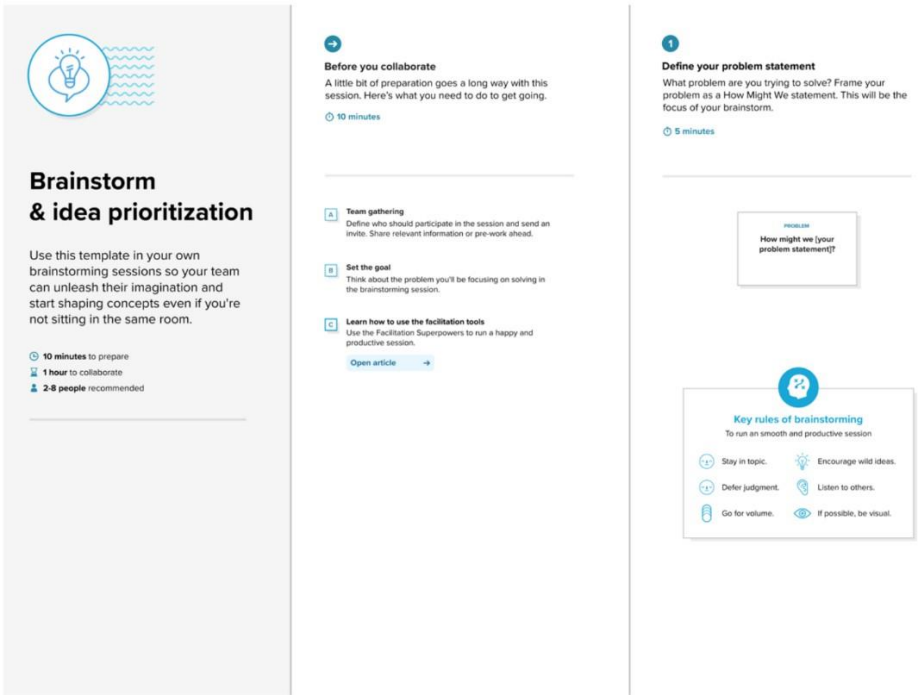
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes. It is a useful tool to help teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



3.2 Ideation and Brainstorming:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem-solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon. All participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-3: Idea Prioritization

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

3.3 Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To assist the farmers in taking better decision in order to acquire healthy crop production by applying data mining.
2.	Idea / Solution description	Applying data mining methods for predicting the crop production across various areas let us to estimate the optimal crop production assisting the framers to benefit from the forecast.
3.	Novelty / Uniqueness	To visualize the past crop yield data and to list out the crops that may yield poor production leading to loss of invested revenue and identify suitable areas for their production.
4.	Social Impact / Customer Satisfaction	Extreme weather conditions such as high temperature, heavy storms or droughts can severely disrupt crop production.
5.	Business Model (Revenue Model)	Increased amount of waste produced from the crop production may lead to a degrade of profit margin
6.	Scalability of the Solution	The acquired insights from the visualization of crop yield data must be durable in such a way that the production is fairly stable even in sudden change of conditions.

3.4 Problem Solution Fit:

4. REQUIREMENT ANALYSIS

4.1 Functional Requirement:

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form
FR-2	User Confirmation	Confirmation via Email
FR-3	User Authentication	Authenticate the user's attempt to login using the database.
FR-4	Yield Forecasting	Crop yield analysis by area, season, climate.
FR-5	Farmer Management	Validating and managing the registered farmer details
FR-6	Crop Management	Add/Delete different types of crop details to the system

4.2 Non-Functional Requirement:

Non-functional Requirements:

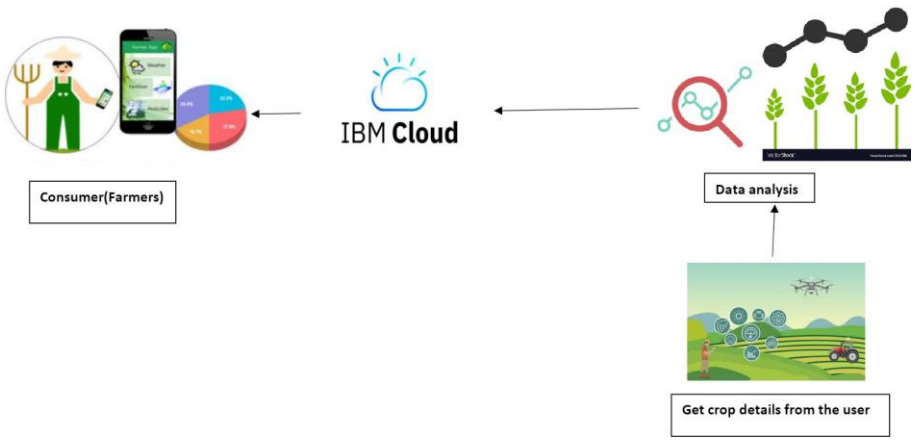
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system must be user friendly allowing the users to ease navigation within the system and also the system must be more interactive.
NFR-2	**Security**	The users must be ensured of data security as the user enters sensitive data into the system.
NFR-3	**Reliability**	The system must be less prone to error.
NFR-4	**Performance**	The performance of the system must assist the system's quality
NFR-5	**Availability**	The system's response time must be less in order to satisfy the user expectations
NFR-6	**Scalability**	The system must able to handle an increase in workload without performance degradation.

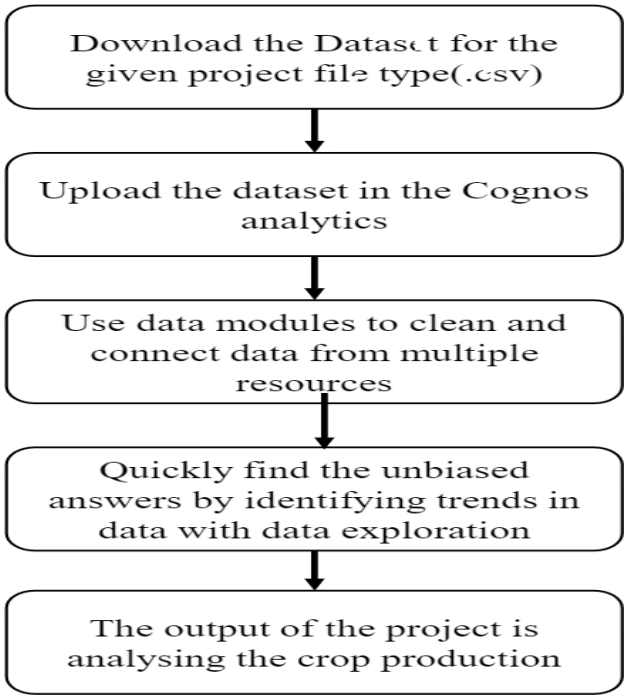
5. PROJECT DESIGN

5.1 Data Flow Diagrams:

Data Flow Diagram for Crop Yield Estimation:



5.2 Solution Architecture:



Technological Architecture:

Technological architecture for crop yield estimation:

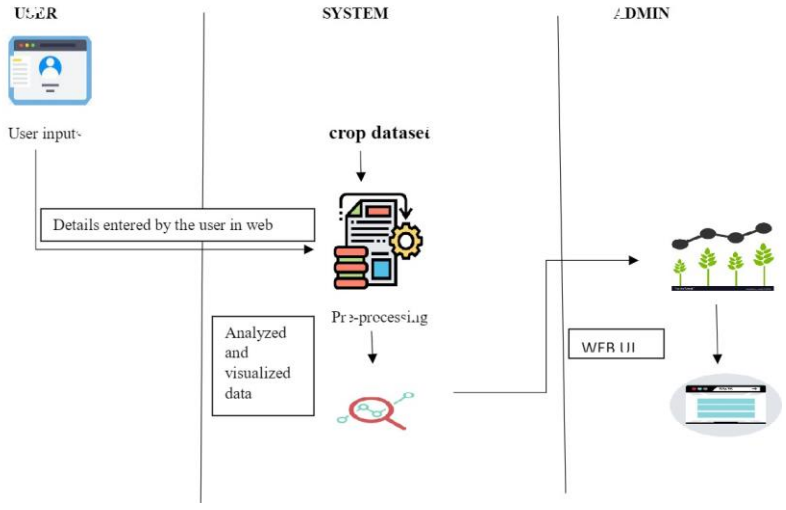


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-253, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used

5.3 User Stories:

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer(Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application.	I can register & access the dashboard	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	Succesfully Registered	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	Successfully Login	High	Sprint-1
	Dashboard	USN-6	As a user, I can access the dashboard.	I can enter the Value in dashboard.	High	
	Analysis	USN-7	As, a user I can analyze the crop	estimation of crop	High	
Administrator	Managing all the Users.	USN-8	As a administrator, I can access the database	add,delete or update the users information.	High	

6. PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Registration	1	As a user(farmer), I can register for by entering my farmer ID.	2	High
		2	As a user, I can register for the application by entering my email and password.	2	Medium
	Login	3	As an already registered user, I can enter into the dashboard after user authentication process.	2	High
	Working with the Dataset	4	To work on the given dataset and to analyse the data , Understand the Data set .	2	High
		5	Load the dataset to Cloud platform then Build the necessary Visualizations on the uploaded dataset.	10	High
Sprint-2	Data Visualization Chart	6	Using the Crop production in Indian dataset, create various graphs and charts to highlight the insights and visualizations. 1)Build a Visualization to showcase Average Crop Production by Seasons.	4	Medium
			2)Showcase the Yearly usage of Area in Crop Production.	4	Medium

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
			3)Build a visualization to show case top 10 States in Crop Yield Production by Area.	4	Medium
			4)Build the required Visualization to showcase the Crop Production by State.	4	Medium
			5)Build Visual analytics to represent the Sates with Seasonal Crop Production using a Text representation.	4	Medium
Sprint-3	Creating The dashboard	7	Create a dashboard that displays all the visualizations.	20	High
Sprint-4	Export The Analytics	8	Export the created Dashboard	20	High

Project Tracker, Velocity & Burndown Chart:

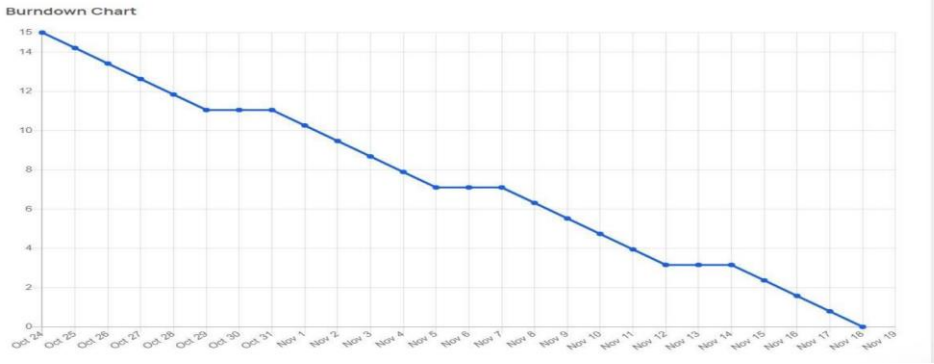
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

We have a 24-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \text{Sprint Duration} / \text{Velocity} = 24 / 20 = 1.2$$

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



6.2 Sprint Delivery Schedule:

Activity Name	Activity Number	Activity Description	Status
Preparation Phase	1	a) Access the resources in project dashboard b) Explore the dataset provided in workspace c) Create GitHub account & collaborate with Project Repository in project workspace d) Set-up the prerequisites for the project	Completed
Ideation Phase	2	a) Literature survey relevant to the selected project and information gathering. b) Preparation of Empathy Map to identify the user pros and cons c) List the ideas by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance	Completed
Project Design Phase-	3		

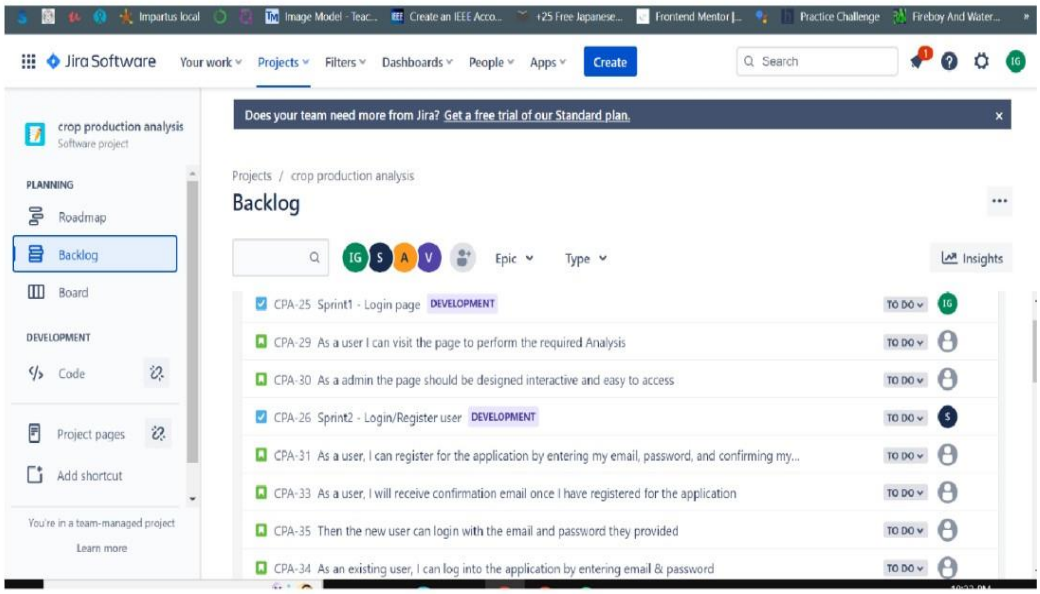
I			
Proposed Solution	3.1	Preparation of proposed solution document, which includes the novelty, feasibility of idea, business model, social impact and solution scalability solution	Completed
Problem Solution Fit	3.2	Prepared problem solution fit which provides effective solutions for the problem	Completed
Solution Architecture	3.3	Develop effective architecture for the proposed solution	Completed
Project Design Phase-II	4		
Requirement Analysis	4.1	Identify the Functional and Non-Functional requirements	Completed
Customer Journey	4.2	Preparation of customer journey map to understand the user interactions & experiences with the application from the entry level to exit level	Completed
Data Flow Diagram and User stories	4.3	Generate Data flow diagram of the project	Completed
Technical Architecture	4.4	Develop effective technical architecture for the proposed solution	Completed
Project Planning Phase	5		
Milestones & Activity List	5.1	Prepare Milestone and Activity list of the project	Completed
Sprint Plan	5.2	Prepare Sprint Delivery plan of the project	Completed
Project Development	6		

Phase			
Delivery of Sprint-1	6.1	Implement the coding phase of Sprint-1	In Progress
Delivery of Sprint-2	6.2	Implement the coding phase of Sprint-2	In Progress
Delivery of Sprint-3	6.3	Implement the coding phase of Sprint-3	In Progress
Delivery of Sprint-4	6.4	Implement the coding phase of Sprint-4	In Progress

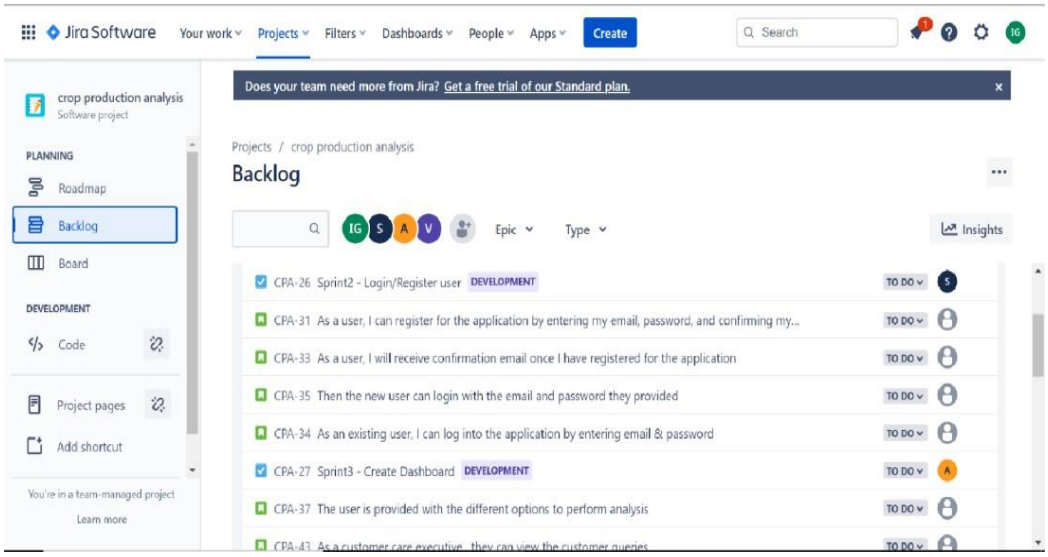
6.3 Reports from JIRA:

SPRINT DELIVERY PLAN

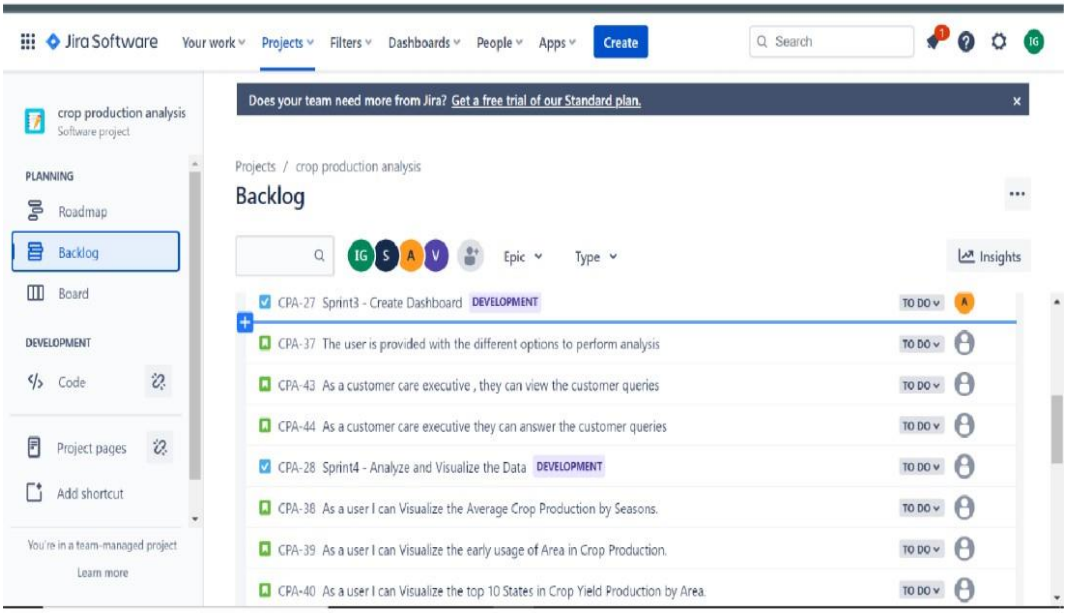
Sprint1 :



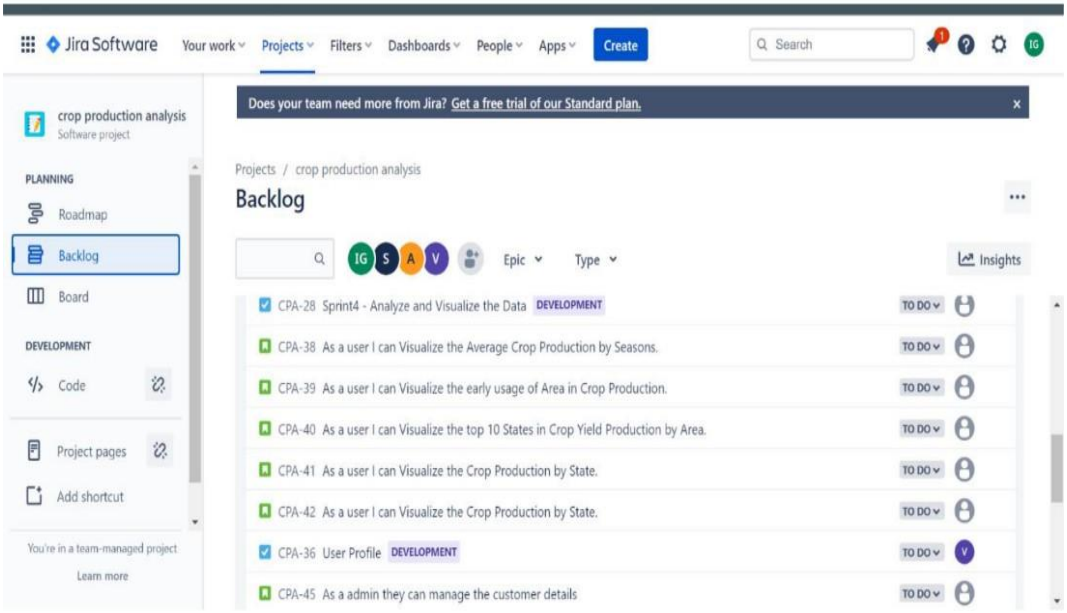
Sprint2:



Sprint3:



Sprint4:



7. CODING & SOLUTIONING

7.1 Feature 1

The farmer can easily visualize the particular features by selecting their area, district, state, and crops.

Crop Yield Analysis and Visualization code

```
import streamlit as st
import numpy as np
import pandas as pd
#import plotly.figure_factory as ff import
plotly.express as px import matplotlib.pyplot as plt
import seaborn as sns
from pandas_profiling import ProfileReport
from streamlit_pandas_profiling import
st_profile_report
import pickle
from pathlib import Path
import streamlit_authenticator as stauth # pip install streamlit-authenticator

def main():
    st.title("Crop Yield Estimation")
     df=pd.read_csv("crop.csv.csv")
     st.dataframe(df)
     fig1=plt.figure(figsize =(10, 4))
     st.title("Visualizaiton to showcase Average Crop Production by Seasons.")
     sns.barplot(x="Season",y="Production",data=df)
     st.pyplot(fig1)
       grouped_single.sort_values(("Area", "sum"))
last=grouped_single.sort_values(("Area","sum")).tail(10)
     fig2=plt.figure(figsize =(10, 4))
     courses=["Punjab","Bihar","Andhra
Pradesh","Gujarat","Karnataka","West
Bengal","Rajasthan","Maharashtra","Madhya Pradesh","Uttar Pradesh"]
values=[4.336316e+08,3.298131e+08,3.222062e+08,2.
720249e+08,2.154052e+08,2.029101e+08,1.549440e+
08,1.315458e+08,1.282720e+08,1.267256e+08]
     st.title(" visualization to show case top 10 States in Crop Yeild Production by Area.")
     sns.barplot(x=courses,y=values) #plt.title("Top 10 States With Most Area",fontsize=20)
     st.pyplot(fig2)
     fig3 = plt.figure(figsize =(10, 4))
       sns.lineplot(df['Crop_Year'],df['Production'])
       st.title("Yearly usage of Area in Crop Production.")
       st.pyplot(fig3)
```

```
fig4 = plt.figure(figsize =(10, 4))
grp =df.groupby("Crop_Year")["Area"].sum().sort_index(asc ending=True)
grp.plot(kind = 'area')
st.title(" Area plot.")
plt.xlabel("Year",fontsize=20)
plt.ylabel("Area",fontsize=20)
st.pyplot(fig4)
fig5 =px.sunburst(df, path=['State_Name', 'Crop'], values='Production')
plt.figure(figsize =(10, 4))
st.plotly_chart(fig5)
plt.title("State With Crop Production",fontsize=20)
```

7.2 Feature 2:

User authentication is provided by maintaining the database. If it is a valid user then theapplication will authenticate the user and enable the user to access it. If it is not a valid user itwill not allow the user to access the application.

Database.py

```
import os
import database as db
from deta import Deta # pip install deta
from dotenv import load_dotenv # pip install python-dotenv
#Load the environment variables
load_dotenv(".env")
DETA_KEY = os.getenv("DETA_KEY")
#  Initialize with a project key
deta = Deta(DETA_KEY)
#This is how to create/connect a database db =deta.Base("crop")
def insert_user(username, name, password):
    """Returns the user on a successful user creation, otherwise raises and error"" return
db.put({"key": username, "name": name, "password": password})
insert_user("pparker","Peter Parker", "abc123")
def fetch_all_users():
     """Returns a dict of all users"""
     res = db.fetch()
return res.items
```

upload.py

```
import streamlit_authenticator as stauth
import database as db
usernames = ["pparker", "rmiller"]
names = ["Peter Parker", "Rebecca Miller"]
passwords = ["abc123", "def456"]
hashed_passwords = stauth.Hasher(passwords).generate()
for (username, name, hash_password) in zip(usernames, names,
hashed_passwords):
    db.insert_user(username, name, hash_password)
```

generate_keys.py

```
import pickle
from pathlib import Path
import streamlit_authenticator as stauth names =["Peter Parker", "Rebecca Miller"]
usernames = ["pparker", "rmiller"] passwords = ["abc123", "def456"]
hashed_passwords = stauth.Hasher(passwords).generate() file_path =
Path(_file_).parent / "hashed_pw.pkl" with file_path.open("wb") as
file:pickle.dump(hashed_passwords, file)
```

8. TESTING

8.1 Test Cases:

A1 fx Test case ID

	A	B	C	D	E	F	G	H	I	J	K
1	Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comm
2	Main Page	UI	Home Page	User can explore the Web App and can login if they want to		Visit the web page URL and click GO		Login/Signup button should display	Working as expected	Pass	
3	LoginPage_TC_001	Functional	Home Page	Verify the UI elements in Login/Signup popup		1.Click on the login button displayed on the top of the application		Application should show below UI elements: a.User Name b.password text box	Working as expected	Pass	
4	LoginPage_TC_002	Funtional	Home Page	When incorrect login details are enterd.		1.Enter URL and click go 2.Click on My Account dropdown button		Application should show Incorrect email or password validation message	Working as expected	Pass	
5	DashBoard_TC_001	Functional	Second page	Once the user logged in the user can see the sample visualizations .For the default dataset		1.Enter the correct password. 2.click the login button. 3.Can see the sample visualizations	Username: peterparker password: Testing123	User should be navigated from the loginpage to the dashboard. The Dashboard displayes the User Name.	Working as expected	Pass	
6	DashBoard_TC_002	Functional	Second page	Here the User can upload the dataset and get an insight about the dataset.		Upload the dataset in the side panel. The UI Elements are . 1.Upload Your CSV Data. 2.Select the Dataset From your device. 3.Wait for your dataset to be loaded. 4.Can see the description of your dataset.	Upload Your CSV Data: Crop_Production.csv	Dataset has been uploaded and shows the basic descrptions of it Successfully	Working as expected	Pass	
				Once the datset is uploaded		Click on the side panel. The UI Elements. 1.Please select your filter.	Upload Your CSV Data: Crop_Production.csv	The Visualizations has been done successfully			

8.2 User Acceptance Testing:

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the crop yield estimation through data analysis project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	7	5	3	2	17
Duplicate	1	0	2	0	3
External	3	2	0	1	6
Fixed	11	3	5	15	34
Not Reproduced	0	0	0	1	1
Skipped	0	1	0	1	2
Won't Fix	0	3	5	1	9
Totals	22	14	15	21	72

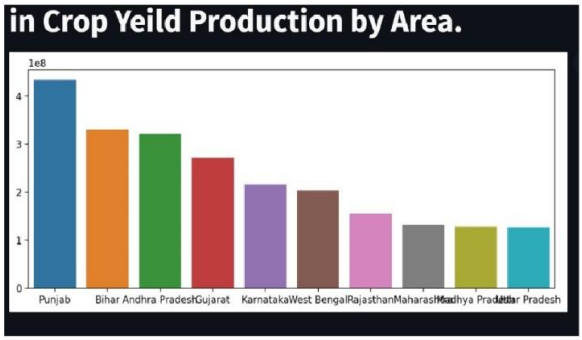
3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	6	0	0	6
Client Application	40	0	0	40
Security	2	0	0	2

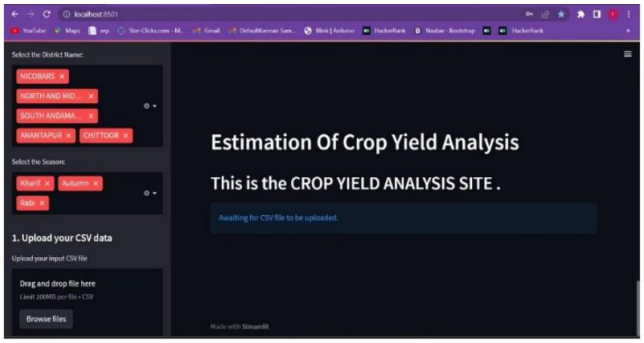
9. RESULTS

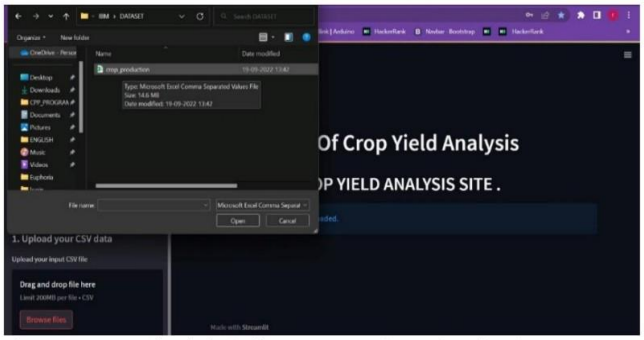
9.1 Performance Metrics:



The user can see the sample visualization.

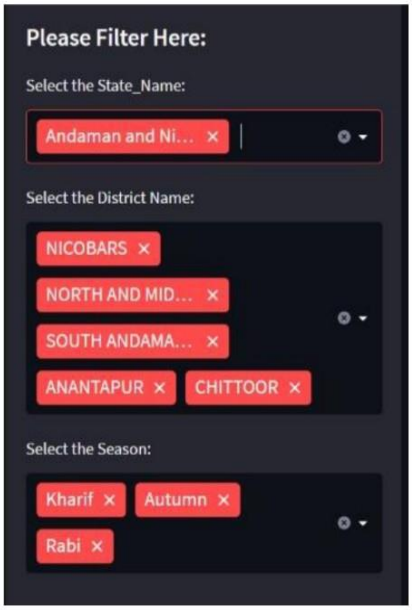
2.Data Upload





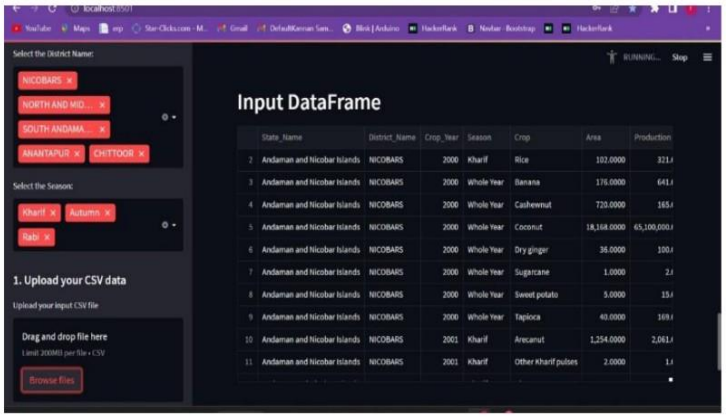
The User can upload their dataset to perform visualization.

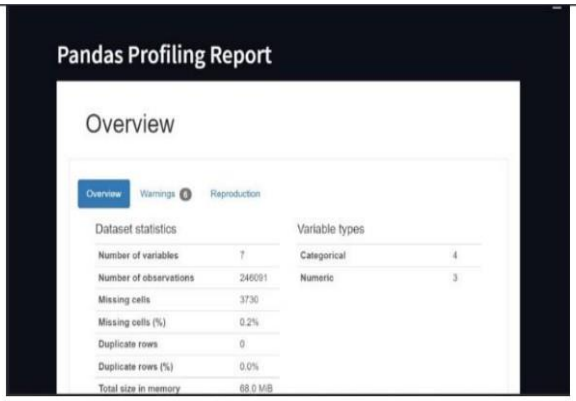
3.Filters

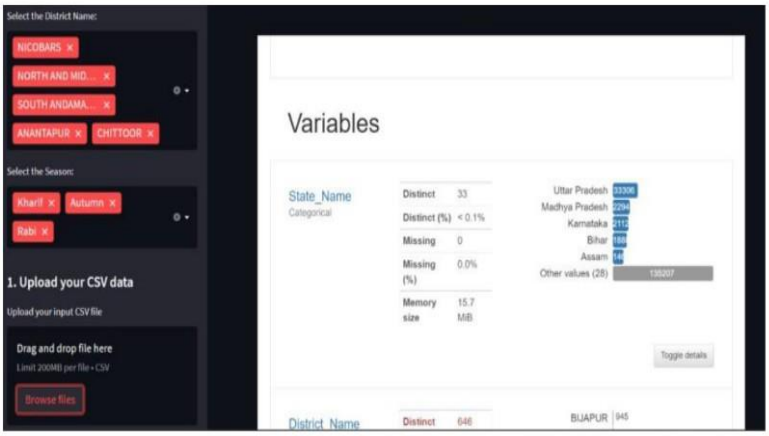


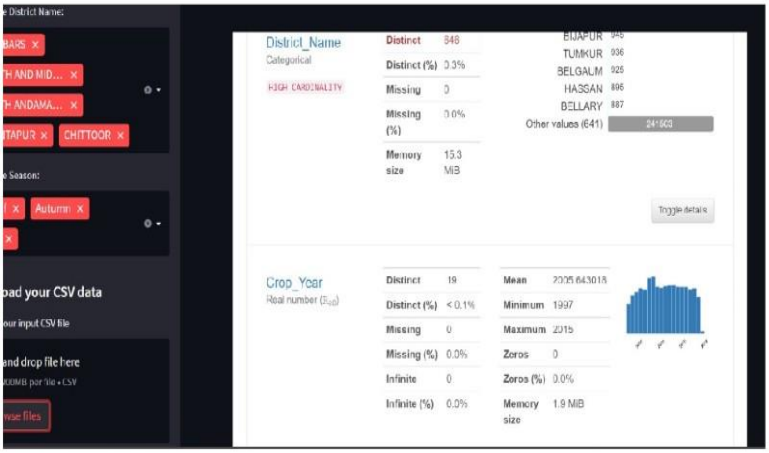
The user can select the filters to perform the respective visualization.

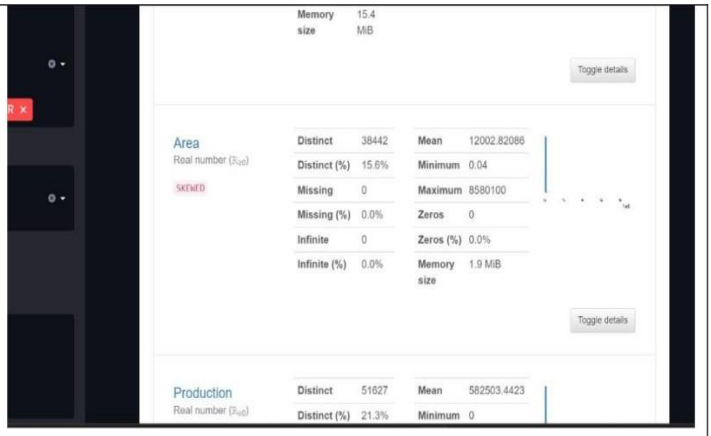
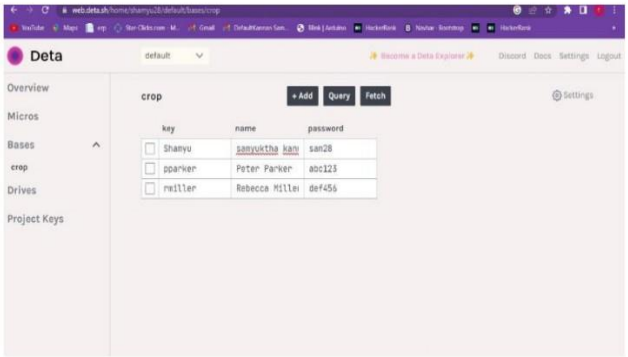
Once the dataset is uploaded we can see the description of the dataset.

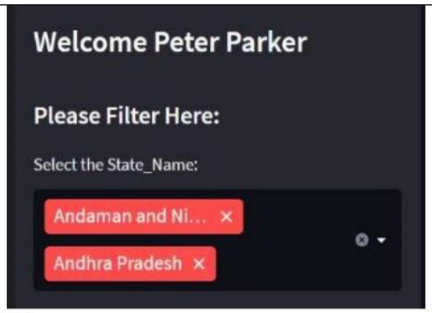
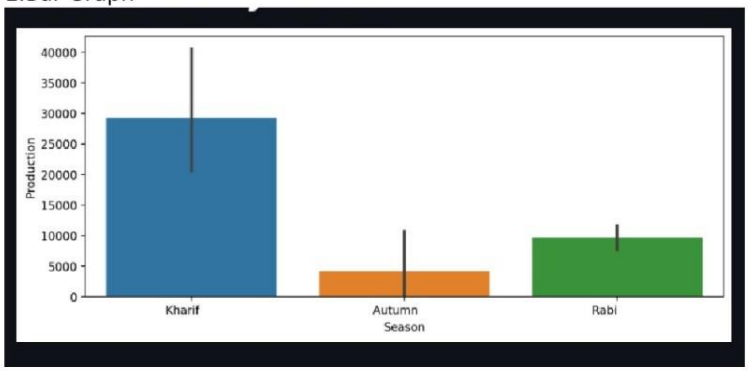



Pandas Profiling Report
Overview
Overview
Warnings 6
Reproduction
Dataset statistics
Number of variables 7
Number of observations 246091
Missing cells 3730
Missing cells (%) 0.2%
Duplicate rows 0
Duplicate rows (%) 0.0%
Total size in memory 68.0 MiB
Variable types
Categorical 4
Numeric 3


Select the District Name:
NICOBARS
NORTH AND MID...
SOUTH ANDAMA...
ANANTAPUR
CHITTOOR
Select the Season:
Kharif
Autumn
Rabi
1. Upload your CSV data
Upload your input CSV file
Drag and drop file here
Limit 200MB per file • CSV
Browse files
Variables
State_Name
Categorical
Distinct 33
Distinct (%) < 0.1%
Missing 0
Missing (%) 0.0%
Memory size 15.7 MiB
Uttar Pradesh
Madhya Pradesh
Karnataka
Bihar
Assam
Other values (28)
Toggle details
District_Name
Distinct 646
BIJAPUR 945

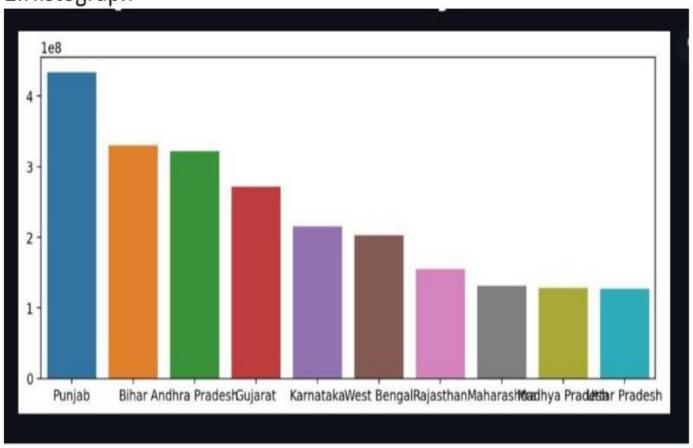

District_Name
Categorical
HIGH CARDINALITY
Distinct 848
Distinct (%) 0.3%
Missing 0
Missing (%) 0.0%
Memory size 15.3 MiB
BIJAPUR 945
TUMKUR 936
BELGAUM 925
HASSAN 896
BELLARY 887
Other values (641)
Toggle details
Crop_Year
Real number (ℝ≥0)
Distinct 19
Distinct (%) < 0.1%
Missing 0
Missing (%) 0.0%
Infinite 0
Infinite (%) 0.0%
Mean 2005.643018
Minimum 1997
Maximum 2015
Zeros 0
Zeros (%) 0.0%
Memory size 1.9 MiB

		 
3.	Amount Data to Rendered (DB2 Metrics)	The User's login details are maintained in the Database Which is accessed only by the admin  The visualization is based on the data uploaded and the filters selected.
4.	Utilization of Data Filters	The Filters are based on 1.State Name

		 2.District Name  3.Season 
5.	Effective User Story	No of Scene Added - 5
6.	Descriptive Reports	No of Visulizations - 5 1.Bar Graph  **Crop production based on the season** The following graph shows when the most crops are produced based on the seasons from the graph we can deduce that most crops are produced in the Kharif season followed by rabi and

autumn. The attributes considered are Production and season from the dataset.

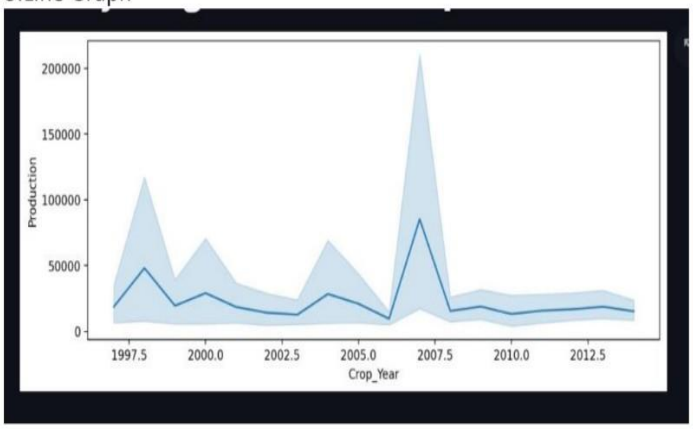
2.Histograph



Crop production based on the district

This graph depicts crop production based on the location and the production from the dataset and infers that Punjab produces the maximum yield

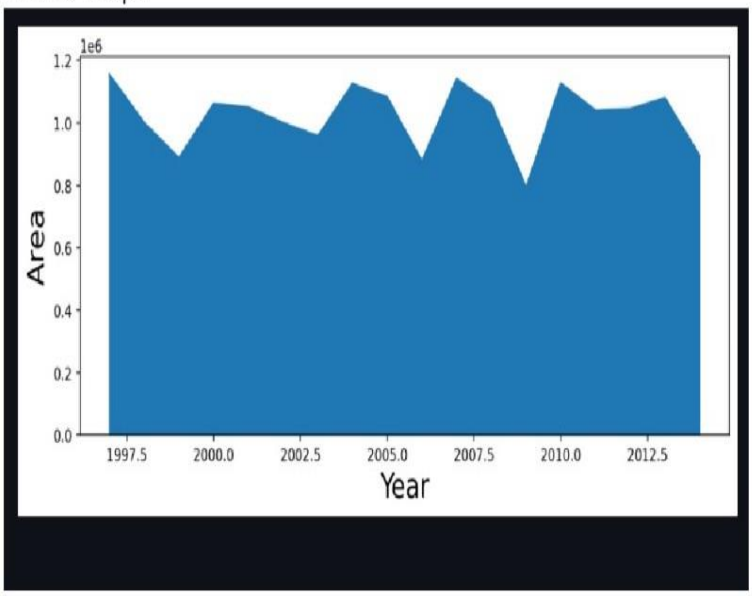
3.Line Graph



Yearly Usage of area in crop production

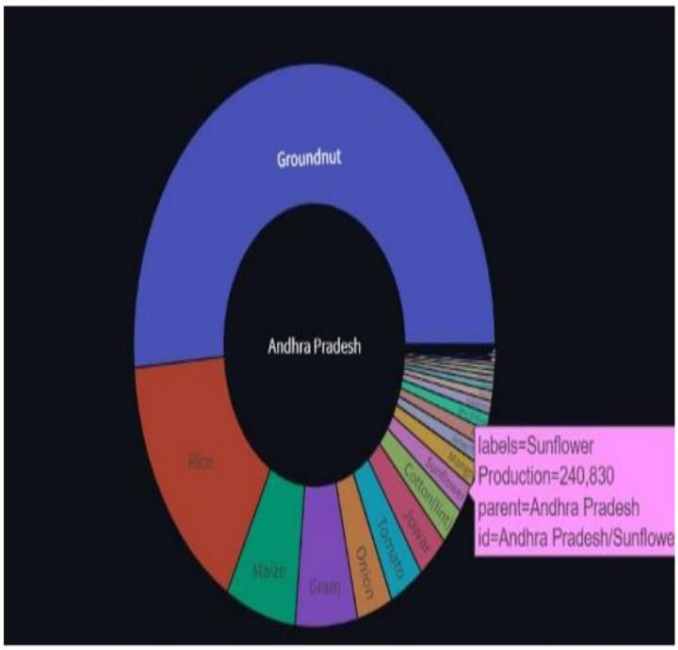
From the graph, we can conclude that the maximum area consumed for agriculture is in 2007

4.Area Graph



Area consumption by Year

5.Pie Chart



Crop Production Based on the location

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

This application makes farmers aware of the riskiness they might face in the future by providing them with real-time weather information and weekly forecast, market prices, and agricultural updates.

This application will help Farmers to make decisions on what crop to sow in a particular season to make maximum yield.

This will reduce the waste generated and improve the profit of the farmer in a digitalized way.

DISADVANTAGES:

Some uneducated farmers cannot able to use this application.

11. CONCLUSION

The agricultural sector is of vital importance to the region. It is undergoing a process of transition to a market economy, with substantial changes in the social, legal, structural, productive, and supply setups, as is the case with all other sectors of the economy. It delivers a well-friendly graphical UI and gives proper access to approved users depending upon their approvals. It successfully overcomes the delay in communications. Modern agriculture uses planned technology and emphasizes management practices of conservation and renewability of resources.

12. FUTURE SCOPE

The application is planned so that future changes can be effectively done. Further, this website can be enhanced for making payments between the farmers and customers. To open agriculture and help farmers from all over the world available in many languages.

13. APPENDIX

SOURCE CODE:

App.py

```
import pickle
from pathlib import Path
import database as db
import numpy as np
import database as db
import matplotlib.pyplot as plt
import seaborn as sns
from pandas_profiling import ProfileReport
from streamlit_pandas_profiling import st_profile_report
import pickle
from pathlib import Path
import streamlit_authenticator as stauth # pip install streamlit-authenticator
import pandas as pd # pip install pandas openpyxl
import plotly.express as px  # pip install plotly-express
import streamlit as st # pip install streamlit
st.set_page_config(page_title="Crop Yield Estimation", page_icon=":bar_chart:",
layout="wide")
# --- USER AUTHENTICATION ---
names = ["Peter Parker", "Rebecca Miller"]
usernames = ["pparker", "rmiller"]

# load hashed passwords
file_path = Path(_file_).parent / "hashed_pw.pkl"
with file_path.open("rb") as file:
    hashed_passwords = pickle.load(file)

authenticator = stauth.Authenticate(names, usernames, hashed_passwords,
    "crops_dashboard", "abcdef", cookie_expiry_days=30)

name, authentication_status, username = authenticator.login("Login", "main")
```

```
if authentication_status == False:
    st.error("Username/password is incorrect")

if authentication_status == None:
    st.warning("Please enter your username and password")

if authentication_status:
    # ---- READ EXCEL ----
    @st.cache
    def get_data_from_excel():
        df = pd.read_excel(
            io="supermarkt_sales1.xlsx",
            engine="openpyxl",
            sheet_name="Sheet1",
            skiprows=3,
            usecols="B:H",
            nrows=1000,
        )
        # Add 'hour' column to dataframe
        #df["hour"] = pd.to_datetime(df["Time"], format="%H:%M:%S").dt.hour
        return df

    df = get_data_from_excel()

    # ---- SIDEBAR ----
    authenticator.logout("Logout", "sidebar")
    st.sidebar.title(f"Welcome {name}")
    st.sidebar.header("Please Filter Here:")

    city = st.sidebar.multiselect(
        "Select the State_Name:",
        options=df["State_Name"].unique(),
        default=df["State_Name"].unique()
    )

    customer_type = st.sidebar.multiselect(
        "Select the District Name:",
        options=df["District_Name"].unique(),
```

```
    default=df["District_Name"].unique(),
  )

  gender = st.sidebar.multiselect(
    "Select the Season:",
    options=df["Season"].unique(),
    default=df["Season"].unique()
  )
 df = df.query(
    " State_Name == @city & District_Name==@customer_type & Season ==@gender"
  )
  st.title(":bar_chart: Crop Yield Estimation Dashboard")
  st.markdown("##")
  fig1=plt.figure(figsize =(10, 4))
  st.title("Visualizaiton to showcase Average Crop Production by Seasons.")
  sns.barplot(x="Season",y="Production",data=df)
  st.pyplot(fig1)
  fig2=plt.figure(figsize =(10, 4))
                    courses=["Punjab","Bihar","Andhra      Pradesh","Gujarat","Karnataka","West
Bengal","Rajasthan","Maharashtra","Madhya Pradesh","Uttar Pradesh"]

values=[4.336316e+08,3.298131e+08,3.222062e+08,2.720249e+08,2.154052e+08,2.029101e+0
8,1.549440e+08,1.315458e+08,1.282720e+08,1.267256e+08]
  st.title(" visualization to show case top 10 States in Crop Yeild Production by Area.")
  sns.barplot(x=courses,y=values)
  #plt.title("Top 10 States With Most Area",fontsize=20)
  st.pyplot(fig2)
  fig3 = plt.figure(figsize =(10, 4))
  sns.lineplot(df['Crop_Year'],df['Production'])
  st.title("Yearly usage of Area in Crop Production.")
  st.pyplot(fig3)
  fig4 = plt.figure(figsize =(10, 4))
  grp = df.groupby("Crop_Year")["Area"].sum().sort_index(ascending=True)
  grp.plot(kind = 'area')
  st.title(" Area plot.")
  plt.xlabel("Year",fontsize=20)
  plt.ylabel("Area",fontsize=20)
  st.pyplot(fig4)
  fig5 =px.sunburst(df, path=['State_Name', 'Crop'], values='Production')
```

```
    plt.figure(figsize =(10, 4))
    st.plotly_chart(fig5)
    plt.title("State With Crop Production",fontsize=20)
    st.markdown('''
    # **Estimation Of Crop Yield Analysis**
    This is the **CROP YIELD ANALYSIS SITE** .---''')
    with st.sidebar.header('1. Upload your CSV data'):
        uploaded_file = st.sidebar.file_uploader("Upload your input CSV file", type=["csv"])
    if uploaded_file is not None:
        @st.cache
        def load_csv():
            csv = pd.read_csv(uploaded_file)
            return csv
        df = load_csv()
        pr = ProfileReport(df, explorative=True)
        st.header('**Input DataFrame**')
        st.write(df)
        st.write('---')
        st.header('**Pandas Profiling Report**')
        st_profile_report(pr)
    else:
        st.info('Awaiting for CSV file to be uploaded.')
```

Database.py

```
import os
import database as db
from deta import Deta # pip install deta
from dotenv import load_dotenv # pip install python-dotenv



# Load the environment variables
load_dotenv(".env")
DETA_KEY = os.getenv("DETA_KEY")
# Initialize with a project key
deta = Deta(DETA_KEY)
# This is how to create/connect a database
db = deta.Base("crop")
def insert_user(username, name, password):
    """Returns the user on a successful user creation, otherwise raises and error"""
```

```
    return db.put({"key": username, "name": name, "password": password})

insert_user("pparker","Peter Parker", "abc123")
def fetch_all_users():
    """Returns a dict of all users"""
    res = db.fetch()
    return res.items
def get_user(username):
    """If not found, the function will return None"""
    return db.get(username)
def update_user(username, updates):
    """If the item is updated, returns None. Otherwise, an exception is raised"""
    return db.update(updates, username)
def delete_user(username):
    """Always returns None, even if the key does not exist"""
    return db.delete(username)
```

Generate_keys.py

```
import pickle
from pathlib import Path
import streamlit_authenticator as stauth
names = ["Peter Parker", "Rebecca Miller"]
usernames = ["pparker", "rmiller"]
passwords = ["abc123", "def456"]
hashed_passwords = stauth.Hasher(passwords).generate()
file_path = Path(_file_).parent / "hashed_pw.pkl"
with file_path.open("wb") as file:
    pickle.dump(hashed_passwords, file)
```

Upload_to_database.py

```
import streamlit_authenticator as stauth

import database as db

usernames = ["pparker", "rmiller"]
names = ["Peter Parker", "Rebecca Miller"]
passwords = ["abc123", "def456"]
hashed_passwords = stauth.Hasher(passwords).generate()
```

```
for (username, name, hash_password) in zip(usernames, names, hashed_passwords):
    db.insert_user(username, name, hash_password)
```

GITHUB AND PROJECT DEMO LINK:

Github Link:

https://github.com/IBM-EPBL/IBM-Project-29643-1660127930

Project Demo Link:

https://drive.google.com/file/d/1A9UdlPrLG9BaE_OM3gG0kYZ5IKT8nebm/view?usp=sharing