PLANT ME

A Project Report

Submitted in partial fulfillment of the Requirements

for the award of the Degree of

MASTER OF SCIENCE (INFORMATION TECHNOLOGY)

By

Mr. Vinay Singh

Mr. Mohan Yadav

Under the esteemed guidance of

Dr. Divya Premchandran

Assistant Professor

DEPARTMENT OF INFORMATION TECHNOLOGY

KERALEEYA SAMAJAM (REGD.)

DOMBIVLI'S

MODEL COLLEGE (AUTONOMOUS)

(A**ffi**liated to University of Mumbai)

DOMBIVLI, 421201

MAHARASHTRA

JAN 2021

ABSTRACT

Traditional systems do not have the descriptive ability to represent the information about which plants can survive at specific location. The project focuses on improving factor productivity and competitiveness of crops and plants by providing specific crops or plants for a specific geographical location.

This project is aimed to develop an information application for user to understand which crops or plants to plant as per the geographic location and present review is to highlight the possibilities of a sustainable crop or plant production in region, which is predicted to suffer from increasingly severe drought in the future due to climate changes.

System will save the information of variety of plant. System will predict which plant can be sown as per the weather, latitude, longitude, season, temperature, humidity, previous year data and other factors will be used to estimate which plants can be sown in a certain geographic region. Artificial intelligence will be used to anticipate the optimal produce for a plantation.

Acknowledgement

It gives us a pleasure to present our project on “PlantMe” This is my milestone in Master of Science (Information Technology). I would like to expressmy sincere thanks to all the teachers who helped us throughout the project. I would liketo acknowledge the help and guidance provided by our Guide Dr. Divya Premchandran in all places during the presentation of this project.

I am thankful to our honorable Principal Dr. Ravindra P Bambardekar. Onwards my project works, I am also thankful to the staff members of the computer department for their moral support towards the project.

Declaration

I hereby declare that the project entitled, PlantMe done at **Model College**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfilment of the requirements for the award of degree **MASTER OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Vinay Singh & Mohan Yadav

INDEX

TABLE OF FIGURES

1. Introduction

1.1.Background study:

During the six decades after independence, India's agriculture has seen a wide range of growth performance. Due to the subsistence character of farming in India and the sector's substantial reliance on monsoon and other environmental conditions, the fluctuation was particularly severe. Green revolution technology propelled the sector's expansion for about three decades in the early years after planned development's inception. As the twentieth century progressed, the impact of the green revolution began to fade.

With the expansion of the Indian economy, agriculture's contribution of GDP has decreased over time, and even now, the productivity of some agricultural practices has decreased.

1.2.Scope:

Agriculture may be the oldest profession, but with the threat of food insecurity rising, its importance has only grown. Artificial intelligence-powered technology is ensuring the long-term viability of high-quality agricultural production. By examining previous year data, AI can be used to choose high-yielding crops and filter out low-yielding crops. Landowners can get recommendations for which crops to plant based on user information such as planting time and location. The system will figure out which crops performed best and make recommendations. The system will include a map of the land field as well as crop information. Agricultural development will help not just farmers, but also a large section of the rural poor who are directly or indirectly involved in agriculture, such as consumers. More efficient production methods, stabilized pricing, and higher agricultural income would produce a more conjugative environment in the country for the overall development of the economy and improved agricultural income.

1.3.Objective:

This will be an information platform that will assist farmers in selecting crops to plant.The system will also provide static crop information. Information systems are becoming increasingly important in improving the efficiency of land administration by making information and services more accessible in support of land markets and urban and rural economic growth. GIS will be used as a technology tool for grasping geography and making intelligent decisions based on space and place since making decisions based on geography is fundamental to human thinking.

1.4. Purpose:

Artificial Intelligence (AI) has lately made an appearance in the agriculture industry. Improper soil treatment, disease and insect infestation, huge data needs, low production, and a knowledge gap between farmers and technology are just a few of the problems the industry faces in order to optimize its produce. The flexibility, high performance, accuracy, and cost-effectiveness of AI in agriculture are the core concepts.

Irrigation is one of the most labor-intensive agricultural operations. AI-trained machines that are aware of past weather patterns, soil condition, and crop types can automate irrigation and boost total production. Irrigation uses almost 70% of the world's fresh water supply; such automation can help farmers manage their water concerns while conserving water.

1.5. Applicability:

Agriculture is one of the major areas where AI can enable growth and greater inclusion. In agriculture, AI-enabled technologies aid farmers in increasing crop yield, monitoring soil health, optimizing pest, and weed control, managing irrigation, and releasing the best price possible.

AI can make crop selection easier and assist farmers in determining which product is the most profitable. Forecasting and predictive analytics can help farmers decrease mistakes in business operations and lower the chance of crop failure. It is feasible to automate harvesting and even anticipate the optimal time for it using AI.

2. Literature Survey:

Artificial Intelligence is revolutionizing agriculture by replacing inefficient methods with more effective ones, making the earth a better place.[1]

Agriculture's future depends on the application of cognitive solutions. Despite the fact that large-scale research is undertaken and certain applications are now available, the area remains underdeveloped. When it comes to dealing with real-world problems and finding answers using autonomous decision-making and predictive solutions, farming is still in its infancy.[2]

AI-based technologies aid in the improvement of efficiency in all fields and the management of difficulties faced by numerous businesses, including crop yield, irrigation, soil content sensing, crop monitoring, weeding, and crop establishment in the agricultural sector. In vast sectors of the

economy with poor productivity, such as agric-food, robotics and autonomous systems (RAS) are being implemented. [3]

The purpose of this article is to discuss various irrigation methods with the goal of building a system that uses less resources and is more efficient. Fertility meters and PH meters are set up on the field to detect the proportion of the primary ingredients of the soil, such as potassium, phosphorous, and nitrogen, to measure the fertility of the soil. Wireless technology is used to plant automatic plant irrigators on the field for drip irrigation. This strategy ensures the soil's fertility as well as the efficient use of water resources.[4]

Artificial intelligence (AI), machine learning (ML), and Internet of Things (IoT) sensors that offer real-time data for algorithms boost agricultural efficiencies, crop yields, and food production costs. According to UN population and hunger projections, the global population will grow by 2 billion people by 2050, necessitating a 60 percent increase in food productivity to feed them. According to the U.S. Department of Agriculture's Economic Research Service, cultivating, processing, and distributing food is a $1.7 trillion industry in the United States alone. AI and machine learning are already showing signs of being able to help overcome the gap in food needs for an additional 2 billion people by 2050.[5]

AI has the potential to transform the way we think about agriculture, allowing farmers to obtain better outcomes with less labor while also providing a slew of other advantages. AI, on the other hand, is not a self-contained technology. AI can enhance currently installed technology as the next stage on the path from traditional to innovative farming.[6]

Conclusion and Thoughts:

In our project has several obstacles, including a lack of appropriate irrigation systems, weeds, crop height-related issues with plant monitoring, and extreme weather conditions. However, with the help of technology, performance may be improved, and thus these issues can be resolved. It can be improved with AI-detecting the soil moisture content and it make more ability of the crops. Farmers' difficulty was that precision weeding techniques were able to offset the high number of crops lost during the weeding procedure. These predictions not only increase productivity, but they also cut the use of unneeded pesticides and herbicides as per the cultivate area. Aside from that, farmers may use AI generated prediction to successfully cultivate their farms, and plant monitoring is no longer a hassle. For starters, in agriculture difficulties, man-made brain power can be used to understand resource. In traditional tactics, a large amount of labor was necessary to get agricultural parameters such as plant height, soil texture, and content, Quick and non-damaging high through put phenol typing would be possible with the help of the various systems investigated, with the added benefit of flexible and favorable activity, on-demand access to information, and spatial goals.

3. **Requirement Analysis:**

3.1. **Problem Definition:**

Even though India is the world's second-largest irrigated country after China, barely one-third of the planted area is irrigated. In a tropical monsoon nation like India, where rainfall is unpredictable, inconsistent, and erratic, irrigation is the most critical agricultural input. India will not be able to make sustained development in agriculture unless and until more than half of the planted area is irrigated.

Low yield is one of the most pressing concerns confronting India's agricultural sector: India's agriculture production is 30-50 percent lower than that of industrialized countries.

Low agricultural production is caused by factors such as average farm size, insufficient infrastructure, a lack of utilization of farm technology and best farming practices, decreased soil fertility owing to excessive fertilization, and continued pesticide usage. Because Indian farms are tiny (70 percent are less than 1 hectare, with the national average being less than 2 hectares), they have limited access to financial services, credit (or lenders), support skills, educational services, and irrigation solutions.

Changing demand as a result of growing incomes, globalization, and more health awareness has an influence on production and will continue to have an impact in the future. Fruits and vegetables, dairy products, seafood, and meat will all see an increase in demand in the future.

There will be increased research, technological advancements, and protected growing of high-value greens and other crops. Processed and low-cost high-quality items will be in more demand.

3.2. **Problem Methodology:**

The AI-based project will help to maximize efficiency in all sectors and handle the issues encountered by various industries, including crop yield, irrigation, soil content sensing, and crop establishment in the agricultural sector. The agriculture industry is in a state of flux, but AI has the ability to provide a much-needed answer.

The project will include a map representation of a geographical region as well as calculations based on AI. For the project's development, we'll use the Agile technique.

Real-world AI solutions are frequently considerably more than a single Machine Learning model that solves the problem "end to end". They're composed of a variety of algorithms, including machine learning and traditional algorithms.

While machine learning models are likely to tackle most of the "toughest" problems, the downstream outcomes that the user should see are simply the result of the combination of all of these components.

3.3. **Survey of Technology:**

There are many interrelated variables that are relevant to arriving at a good understanding of the future of AI, in particular the path towards general AI, with varying degrees of ement Specify this year's survey, we asked respondents across the world and across different business functions about adoption, benefits and barriers of AI for each of activities where AI is used. The results suggest that AI is delivering meaningful value to companies and individuals, as well.

AI is a set of technologies that are applied to certain jobs, rather than a single technology. In order to create a glimpse of the future of AI in project management, the questionnaire attempted to capture collective viewpoints by spanning various geographies, individuals, sectors, and jobs.

Due to the success of various machine learning methods in the field of AI, many firms have been using AI and ML interchangeably in recent years. To be clear, machine learning refers to a program's ability to learn, whereas artificial intelligence includes learning as well as other capabilities. The utilisation of a significant amount of data by AI algorithms to alter their internal structure so that when new data is offered, it is categorised in accordance with the previous data is a crucial takeaway. Rather of merely following the categorization instructions contained in the code, we term this "learning" from the data. The most significant aspects of creating deep learning system specs, such as the CPU for general computation, the GPU (and GPU compute) for neural network primitives, and system memory for processing large datasets, will be discussed. We'll compare two hypothetical case studies with different approaches to similar domains and, ultimately, very similar prime directives to make things more tangible. These will be based on two alternative ways to launching a company focused on finding new pharmaceuticals to treat diseases that are now untreatable.

3.4. **Requirement Specification:**

Python was used to create this software. Python is a wonderful language for AI since it comes with built-in libraries like Numpy for scientific computation, Scipy for advanced computing, and Pybrain for machine learning (Python Machine Learning).

3.4.1. Functional Requirement:

We must exhibit our data in a map for this project, thus it is critical to display data in such a way that users can comprehend the best Corp for irrigation. Some of the project's

components include data visualisation in static form, which will serve as an educational tool for viewers.

Map Representation – This is the section where map of state will represent irrigation data. System Should Show Best Corps for irrigation.

Corp Static Information – This is the section of application where application will show static information of crop and other related information.

3.4.2. Software and Hardware Specification:

- **Hardware Requirement**
  - monitor
  - Keyboard and mouse
  - CPU
  - Etc.
- **Software Requirement**
  - **Python** – Python is a beautiful and powerful programming language that is simple to use (both to read and write) and allows you to connect your project to the real world using the Raspberry Pi.
  - **Python logo** - Python syntax is simple and straightforward, with a focus on readability and the usage of standard English vocabulary. Begin by launching IDLE from your desktop.
  - **IDLE-** The most straightforward way to learn Python is to use IDLE, a Python development environment. IDLE can be launched from the Desktop or the Applications menu:
  - Python in the applications menu –
    IDLE includes a REPL (Read-Evaluate-Print-Loop) prompt where you can type Python instructions. You can even receive the output of commands printed to the screen without using print because it's a REPL.
    Python is available in two versions: Python 2 and Python ement Specif
    Python 3 was originally launched in 2008, while Python 2 development came to an end with the release of Python 2.7 in 2010. Python 3 is the preferred version, however Python 2 is available for legacy applications that don't yet support Python 3.
  - **Java Script** - JavaScript (commonly abbreviated as JS) is a lightweight, interpreted, object-oriented programming language with first-class functions. It's best known as a scripting language for Web pages, although it's also used in a variety of non-browser applications. It's a multi-paradigm, prototype-based scripting language that supports object-oriented, imperative, and functional programming techniques.
  - JavaScript is a client-side scripting language that may be used to create and programme how web pages react when an event occurs. JavaScript is a popular scripting language for managing the behaviour of web pages. It is simple to learn and powerful.

- Despite widespread belief, JavaScript is not "Interpreted Java." In a word, JavaScript is a dynamic programming language that allows you to create prototype-based objects. To limit the number of new concepts necessary to learn the language, the basic grammar is designed to be similar to both Java and C++. If statements, for and while loops, switch and try... catch blocks, and other language constructs work the same way they do in these languages (or nearly so).
- **Django** - Django is a high-level Python web framework for building secure and maintainable websites quickly. Django is a web framework created by experienced developers that handles much of the heavy lifting so you can concentrate on designing your project rather than reinventing the wheel. It's open source and free, with a thriving community, comprehensive documentation, and a range of free and paid support options..
- Insanely fast - Django was created with the goal of assisting developers in getting applications from concept to completion as rapidly as feasible.
- Reassuringly secure - Django prioritizes security and assists developers in avoiding many common security pitfalls.
- · Extremely scalable - Django's ability to grow swiftly and flexibly is used by some of the busiest websites on the internet.
- **Visual Studio Code** - Microsoft's Visual Studio Code is a source-code editor for Windows, Linux, and macOS.
- Microsoft first launched Visual Studio Code at the 2015 Build conference on April 29, 2015. Shortly later, a preview build was released.

- Visual Studio Code's source code was released under the MIT License and made accessible on GitHub on November 18, 2015. Support for extensions was also announced. Visual Studio Code was released to the Web on April 14, 2016, after graduating from the public preview stage

3.4.2.1. Front End –

- • HTML- HTML, or Hyper Text Markup Language, is the standard markup language for web-based writings. Cascading Style Sheets (CSS) and computer languages like JavaScript can be of use.
- • Web browsers translate HTML documents received from a web server or locally saved files into multimedia web pages. HTML originally included visual clues for the document's appearance and logically represented the structure of a web page.
- The components that make up HTML pages are known as HTML elements. Using HTML techniques, images and other objects, such as interactive forms, can be incorporated in the final page. By specifying structural semantics for text components such as headers, paragraphs, lists, links, quotations, and other

elements, HTML allows you to create well-organized texts. HTML elements are separated by tags, which are written in angle brackets. Tags like image /> and input /> immediately add content to the page. Other tags, like p>, surround and provide information about document text, and may have sub-elements like other tags. Browsers do not show HTML tags, but they are used to read the content of the page.

- • CSS - Cascading Style Sheets (CSS) is a style sheet language for describing the appearance of a document created with a markup language such as HTML. CSS, like HTML and JavaScript, is an important part of the web.
- CSS (Cascading Style Sheets) is a style sheet that allows you to separate presentation from content, such as layout, colours, and fonts.
- The same markup page can be presented in several styles for different rendering techniques, such as on-screen, in print, by voice (through a speech-based browser or screen reader), and on Braille-based tactile devices, because formatting and content are separated. CSS also gives guidelines for different formatting if the material is viewed on a mobile device.If more than one style rule fits a specific element.

- ## Back End
- **Python –**

  Python is a high-level programming language that is dynamically semantic, interpreted, and object-oriented. It's ideal for Rapid Application Development and as a scripting or glue language for connecting existing components because of its high-level built-in data structures, dynamic type, and dynamic binding. Python's concise, easy-to-learn syntax is prioritised for readability, minimising programme maintenance costs. Modules and packages are supported by Python, which fosters programme modularity and code reuse. The Python interpreter and its substantial standard library are free to download and distribute in source or binary form for all major platforms. Because there is no compilation stage, the edit-test-debug cycle is extremely fast. Python programmes are easy to troubleshoot: a bug or incorrect input will never cause a segmentation fault. Instead, if the interpreter discovers a flaw, it raises an exception. If the programme fails to catch the exception, the interpreter outputs a stack trace. A source level debugger allows you to inspect local and global variables, execute arbitrary expressions, set breakpoints, walk through the code one line at a time, and so on. Python is used to create the debugger, exhibiting Python's introspective capabilities. Adding a few print lines to the source code, on the other hand, is typically the quickest way to debug a programme: the quick edit-test-debug cycle makes this simple approach highly successful.

4. System Planning

Gantt Chart –

A Gantt chart, which is widely used in project management, is one of the most popular and practical ways to depict activities (tasks or events) against time. A list of the activities may be found on the left side of the chart, and an appropriate time scale can be found along the top. Each action is represented by a bar, whose location and length indicate the activity's start, duration, and finish dates.

PlantMe Project Schedule

Gantt Chart Template © 2006-2018 by Vertex42.com.

SNI

Project Start Date: 16-08-2021 (Monday) Display Week: 1

WBS	TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS	Predecessor
1	**PlantMe**			-			-	
1.1	Future Corp Prediction		Mon 16-08-21	Tue 01-02-22	170	50%	122	
1.2	Requirment gathering		Mon 16-08-21	Sun 22-08-21	7	40%	5	1.1
1.3	planning		Wed 25-08-21	Wed 01-09-21	8	25%	6	1.2
1.4	Analysis and gathering		Mon 06-09-21	Mon 13-09-21	8	10%	6	1.3
2	**Module 1- Seach Module**			-			-	-
2.1	Requirment gathering		Wed 15-09-21	Fri 17-09-21	3	35%	3	
2.2	planning		Mon 20-09-21	Thu 23-09-21	4	45%	4	2.1
2.3	Analysis and gathering		Fri 24-09-21	Tue 28-09-21	5	20%	3	2.2
2.4	Coding		Fri 01-10-21	Thu 21-10-21	21	10%	15	2.3
2.5	Testing		Thu 21-10-21	Thu 28-10-21	8	10%	6	2.4
3	**Module 2-Information Module**			-			-	-
3.1	Requirment gathering		Wed 27-10-21	Thu 04-11-21	9	40%	7	
3.2	planning		Thu 04-11-21	Thu 11-11-21	8	20%	6	3.1
3.3	Analysis and gathering		Thu 11-11-21	Mon 22-11-21	12	15%	8	3.2
3.4	Coding		Mon 22-11-21	Sun 19-12-21	28	10%	20	3.3
3.5	Testing		Sat 18-12-21	Sun 26-12-21	9	0%	5	3.4
4	**Module 3- Forcast module**			-			-	-
4.1	Requirment gathering		Sun 26-12-21	Sun 02-01-22	8	0%	5	
4.2	planning		Mon 03-01-22	Thu 13-01-22	11	0%	9	4.1
4.3	Analysis and gathering		Fri 14-01-22	Wed 19-01-22	6	0%	4	4.2
4.4	Coding		Fri 21-01-22	Sun 06-03-22	45	0%	31	4.3
4.5	Testing		Fri 04-03-22	Sun 13-03-22	10	0%	6	4.4
4.6	Testing and Implitation		Sat 12-03-22	Wed 16-03-22	5	0%	3	4.5

Week 1 16 Aug 2021 | Week 2 23 Aug 2021 | Week 3 30 Aug 2021 | Week 4 6 Sep 2021 | Week 5 13 Sep 2021 | Week 6 20 Sep 2021 | Week 7 27 Sep 2021 | Week 8 4 Oct 2021

Fig. 1. Gantt Chart

Task for Project and Search Modules

TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS	Predecessor
PlantMe			-			-	
Future Corp Prediction		Mon 16-08-21	Tue 01-02-22	170	50%	122	
Requirment gathering		Mon 16-08-21	Sun 22-08-21	7	40%	5	1.1
planning		Wed 25-08-21	Wed 01-09-21	8	25%	6	1.2
Analysis and gathering		Mon 06-09-21	Mon 13-09-21	8	10%	6	1.3
Module 1- Seach Module			-			-	-
Requirment gathering		Wed 15-09-21	Fri 17-09-21	3	35%	3	
planning		Mon 20-09-21	Thu 23-09-21	4	45%	4	2.1
Analysis and gathering		Fri 24-09-21	Tue 28-09-21	5	20%	3	2.2
Coding		Fri 01-10-21	Thu 21-10-21	21	10%	15	2.3
Testing		Thu 21-10-21	Thu 28-10-21	8	10%	6	2.4

Fig. 1.1 Gantt Chart

Task for 2nd and 3rd Module

TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS	Predecessor
Module 2-Information Module			-			-	-
Requirment gathering		Wed 27-10-21	Thu 04-11-21	9	40%	7	
planning		Thu 04-11-21	Thu 11-11-21	8	20%	6	3.1
Analysis and gathering		Thu 11-11-21	Mon 22-11-21	12	15%	8	3.2
Coding		Mon 22-11-21	Sun 19-12-21	28	10%	20	3.3
Testing		Sat 18-12-21	Sun 26-12-21	9	0%	5	3.4
Module 3- Forcast module			-			-	-
Requirment gathering		Sun 26-12-21	Sun 02-01-22	8	0%	5	
planning		Mon 03-01-22	Thu 13-01-22	11	0%	9	4.1
Analysis and gathering		Fri 14-01-22	Wed 19-01-22	6	0%	4	4.2
Coding		Fri 21-01-22	Sun 06-03-22	45	0%	31	4.3
Testing		Fri 04-03-22	Sun 13-03-22	10	0%	6	4.4
Testing and Implitation		Sat 12-03-22	Wed 16-03-22	5	0%	3	4.5

Fig. 1.2 Gantt Chart

Gantt Chart for Project and Search Modules

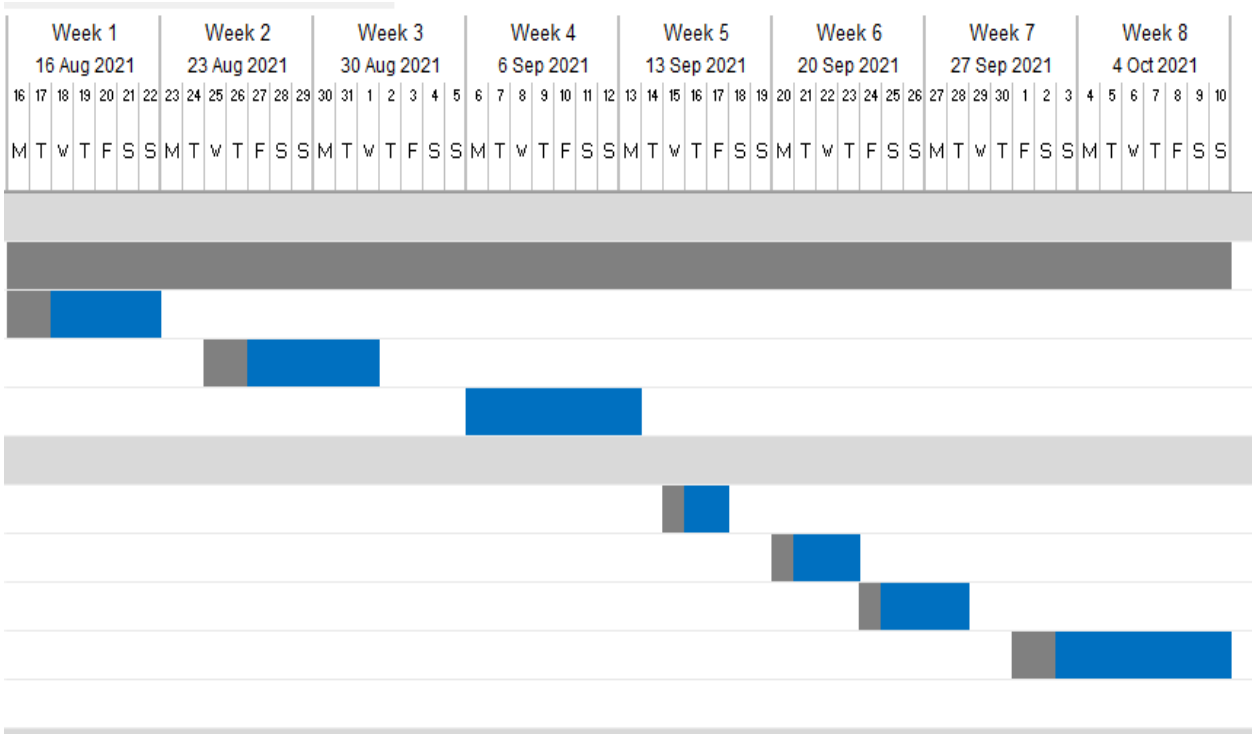


Fig. 1.3 Gantt Chart

Gantt Chart for Module 2

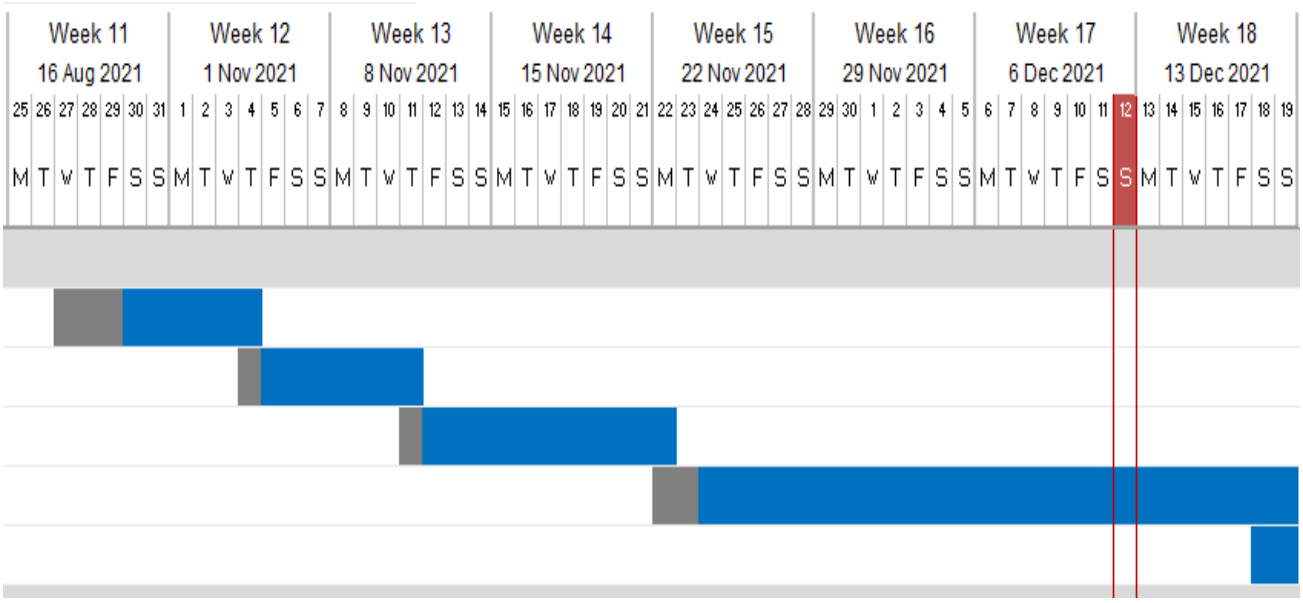


Fig. 1.4 Gantt Chart

Gantt Chart for Module 3

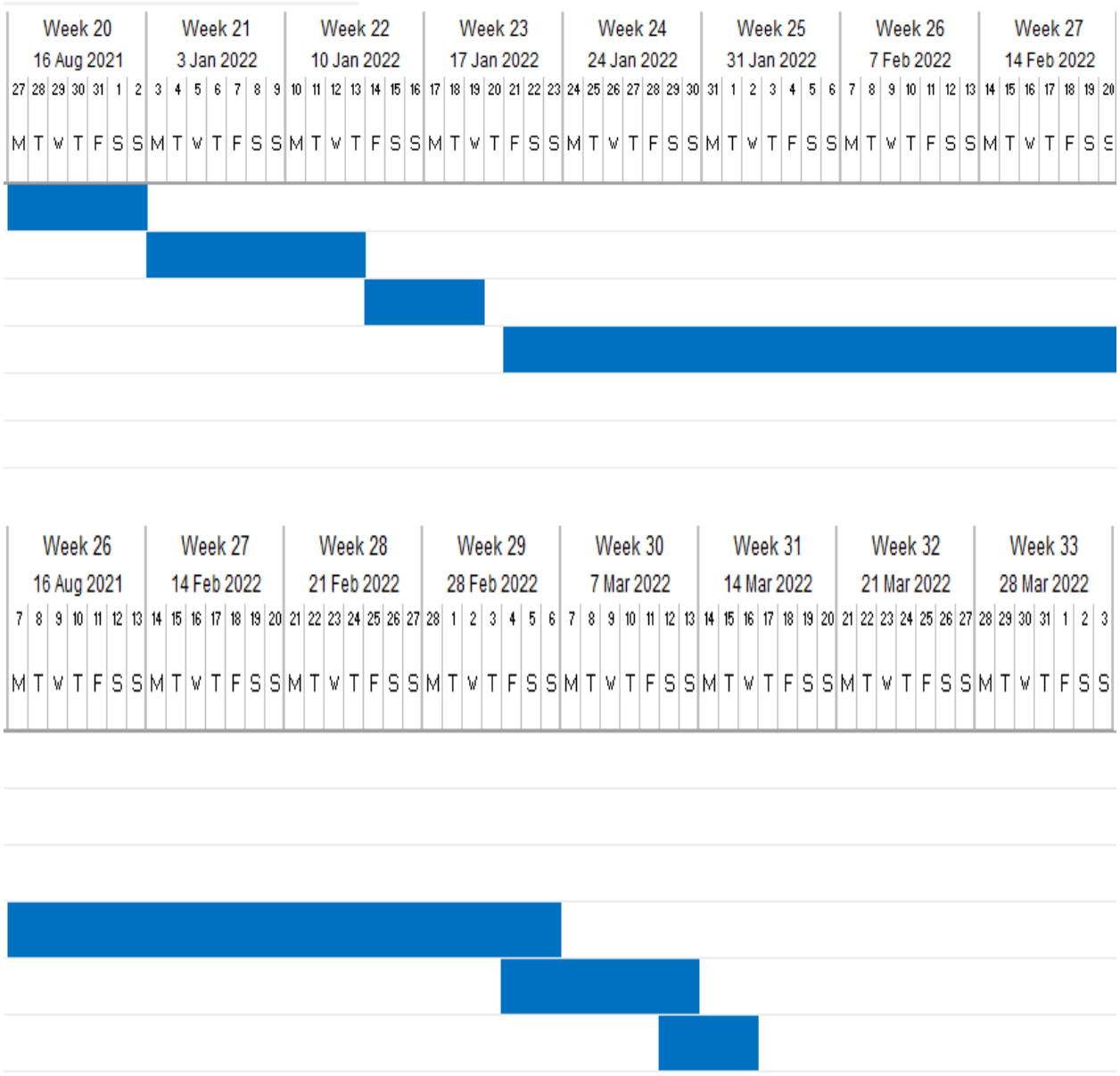


Fig. 1.5 Gantt Chart

Network Diagram-

Network diagram for Module 1:

A project network is a graph (weighted directed graph) that shows the order in which a project's terminal elements must be finished, as well as their dependencies. To reflect the timeline of the project, it is always drawn from left to right.

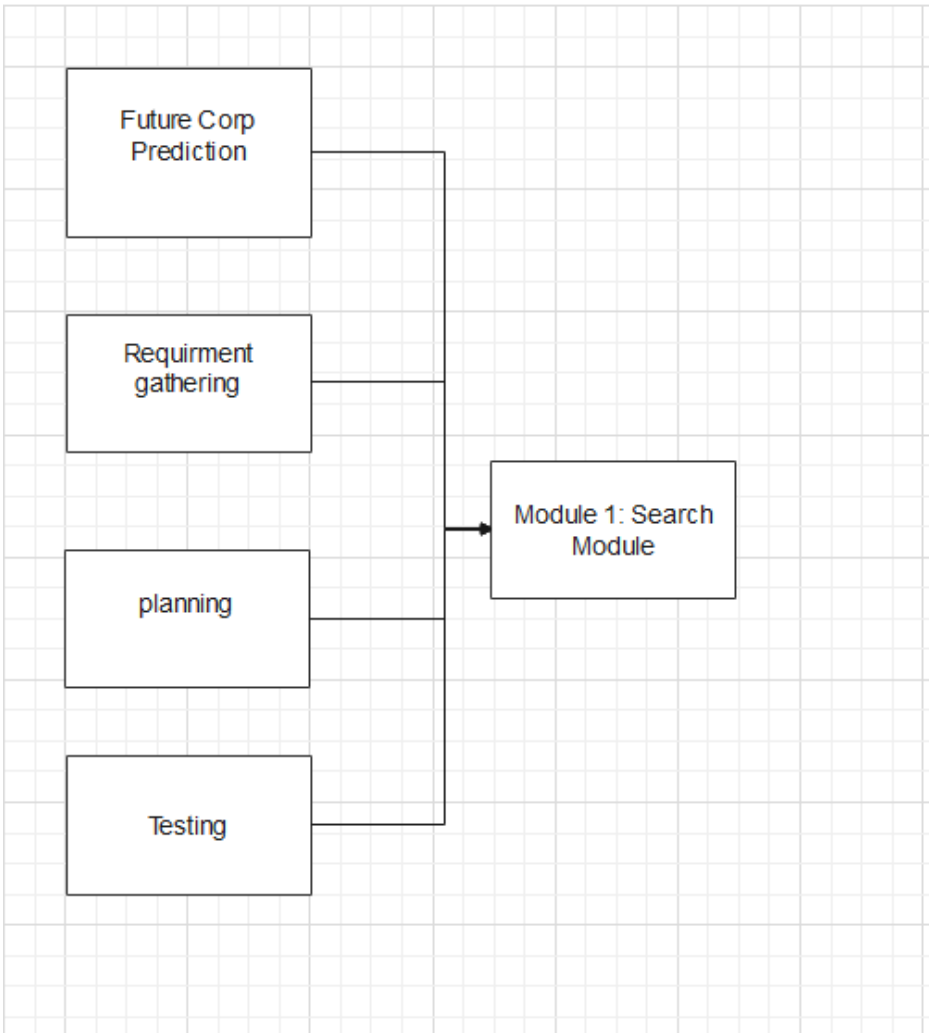


Fig. 2 Network Diagram

Network diagram for Module 2:

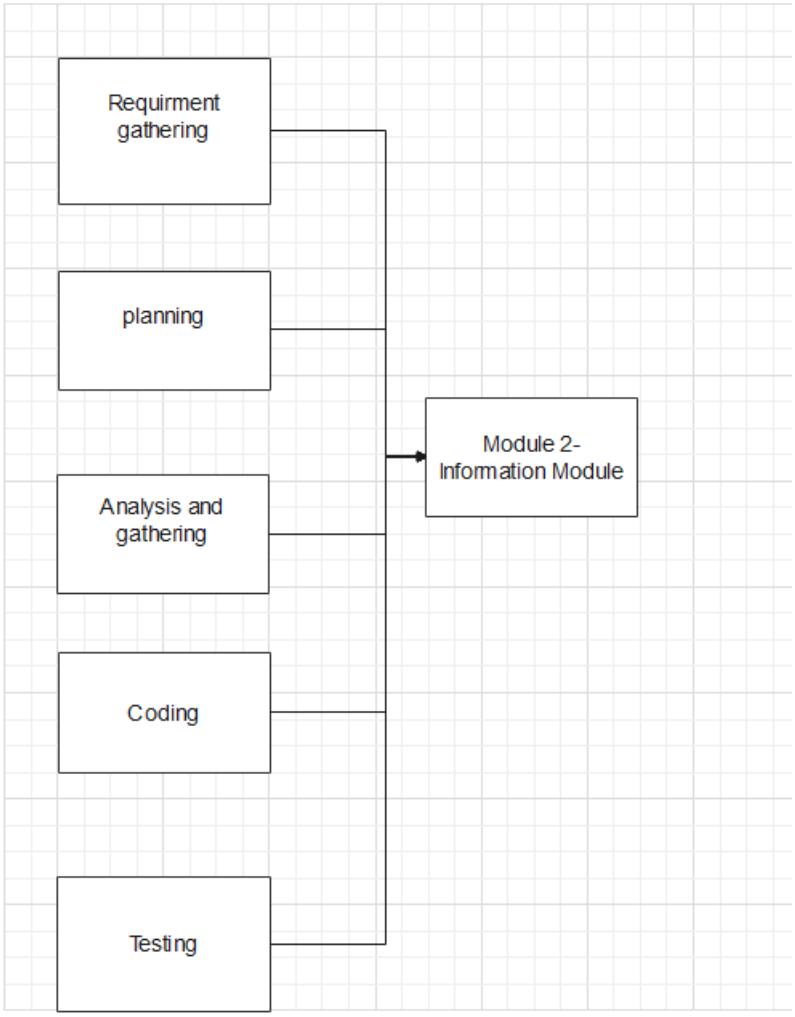


Fig. 2.1 Network Diagram

Network diagram for Module 3:

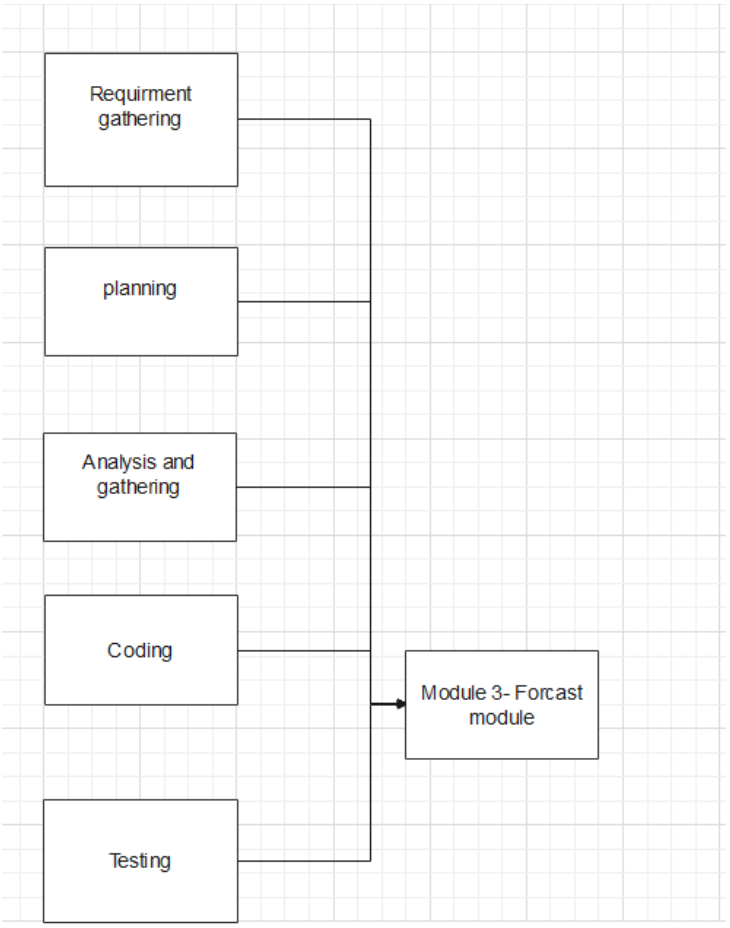


Fig. 2.2 Network Diagram

Testing and Implementation Task for Project Completion:

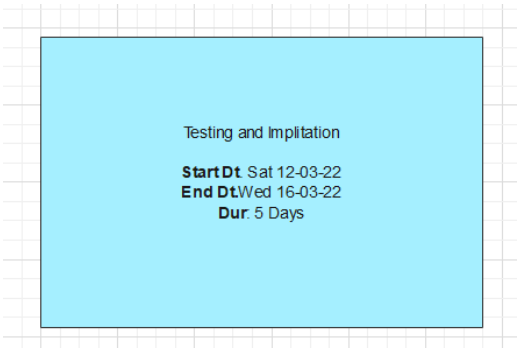


Fig. 2.3 Network Diagram

5. System Design
5.1. Procedural Design

Algorithm to Access Software
Step 1 :- Open Web Application on the Computer or Laptop.

Step 2 :- Select Region for Prediction.

Step 3 :- Select Sowing season

Step 4:- Display the forecast data

5.2. Conceptual Model:

Flowchart: -

A flowchart is a formalized graphic representation of a logic sequence, work or manufacturing
process, organization chart, or similar formalized structure. The purpose of a flow chart is to provide people with a common language or reference point when dealing with a project or process.

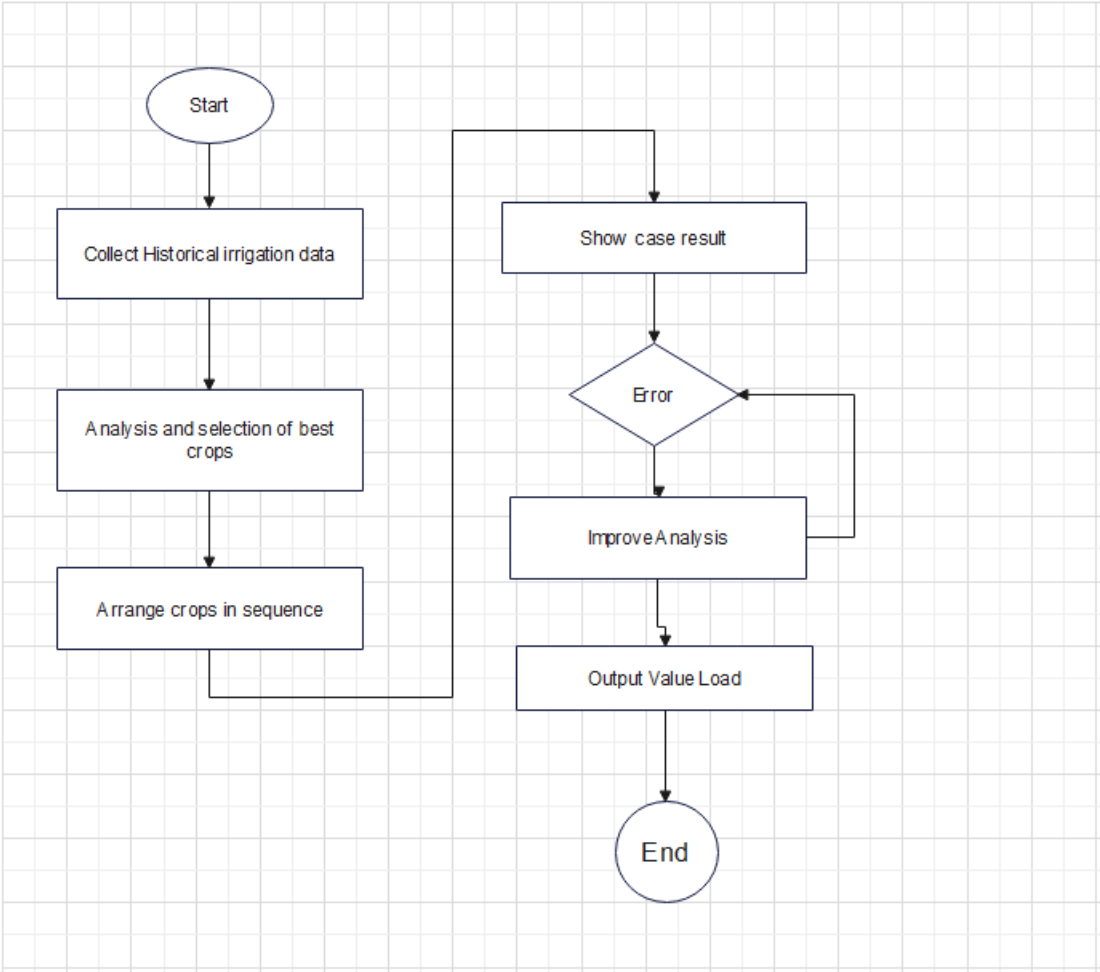


Fig. 3 Flowchart

5.3. Basic Modules-

There are 3 major modules. They are as follow.

Search Module

This is the module where user must provide input date based on which system will calculate the prediction. System will require information lake state name city name season of sowing seeds etc.

Information Module

This is the Module where user will be able to see static information and current affairs related to agriculture. It will display modern farming methods and schemes provided by government. This module will be educating farmers and will help them to understand modern methods of farming.

Forecasting Module

This is the Module which will be used to make calculation and prediction of best crops based on provided data by user. Application will predict and will sort the best sowing crops in sequence and system will rate them as well so user can understand how good it would be to plant crops.

5.4. Logic Diagram

Activity Diagram: -

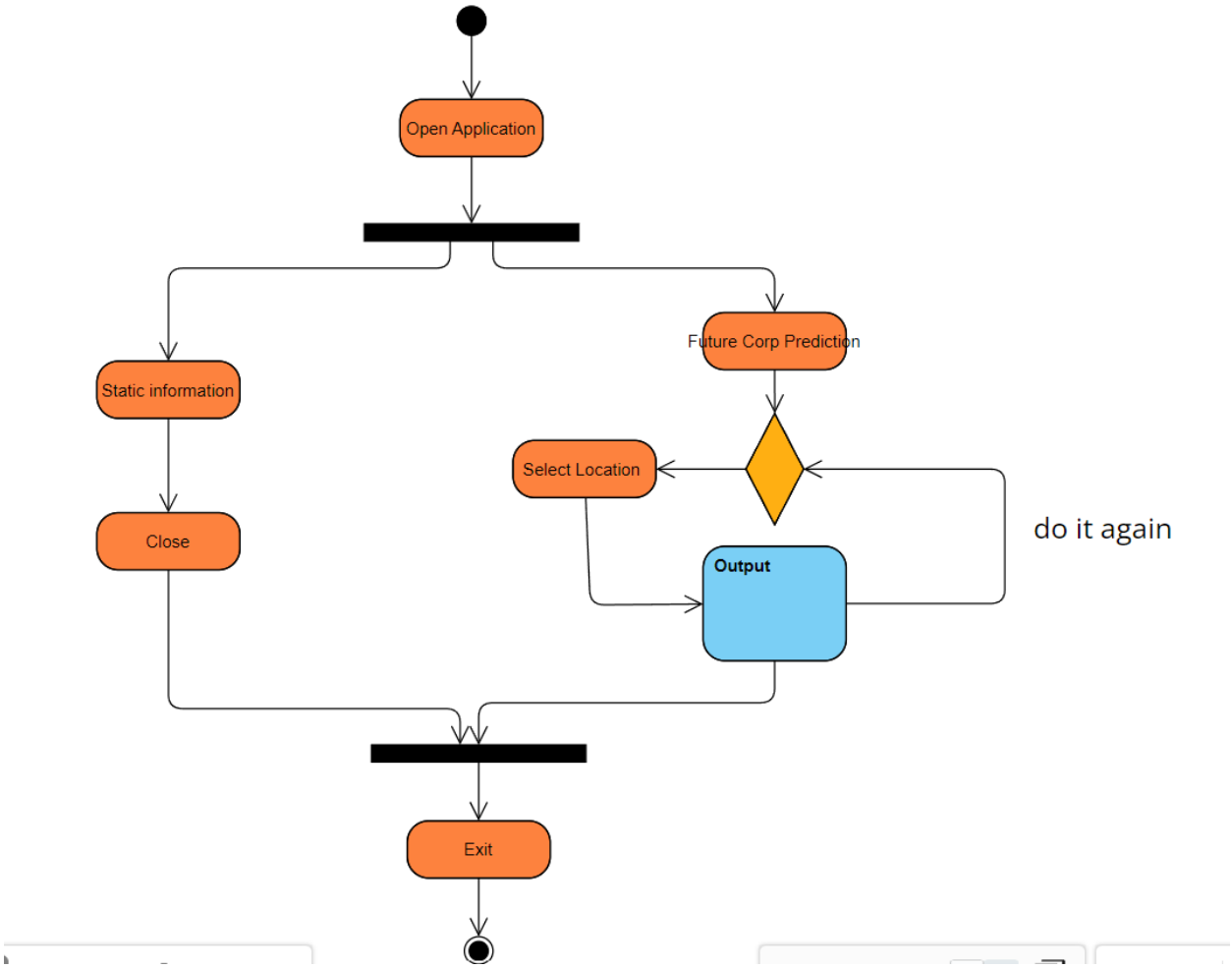


Fig 4 Activity Diagram

5.5. Security Issues

The application layer is responsible for providing services to end users, storing data, and making system choices. This layer's security concerns are specific to different applications and focus on preventing data theft and ensuring privacy. Some applications have a sub-layer that provides support for services and aids in resource allocation. Each program has its own set of characteristics, and it's hard to anticipate all potential flaws. As a result, the security concerns described below are some of the vulnerabilities that cloud-based applications and services may face.

Phishing: - It is a virtual bug that seeks to gain confidential user data, such as ID and password, through deception. Phishing typically obtains end-users through phishing emails or websites. An adversary with administrative access to the system can send fake commands to actuators and modify system settings. In some circumstances, the attacker may be able to disrupt decision-making or other internal processes. This type of attack is impossible to avoid, however secure access control mechanisms can help limit the damage. However, the most effective prevention would be for individuals to remain attentive while using the internet.

Malicious scripts: - Agricultural solutions that are connected to the Internet can interact with other online services and users. Because of this interaction, malicious programs such as Java applets, Active-X scripts, and cross-site scripting (XSS) might attack them. Customers can be misled, dangerous material can be injected, sensitive information can be accessed, and security mechanisms can be broken by malicious scripts. This type of attack is frequently carried out by cybercriminals for personal, financial, or political gain. Malicious scripts can cause service disruption or harm, as well as display unwelcome adverts and extort money.

Denial of Services: - By overwhelming the network traffic or flooding the service with repeated requests, this attack causes service interruptions. An opponent can start by exploiting security flaws this attack originates from the Internet or a subsystem. Such attacks deny legitimate users access to services, inhibit the appropriate processing or storing of non-persistent data, degrade the efficiency of vital systems (such as environmental controls in mushroom farms), and may even result in the entire shutdown of the system.

Because it comprises cloud-based apps and services, it is subject to all cloud security concerns. Because the cloud exposes apps and resources to Internet-based threats, it's critical to take security precautions. To safeguard the many sensitive data stored and processed in the cloud, security usually focuses on privacy and access control. However, it is critical to consider more than simply privacy and access control when implementing security measures to preserve the system's availability and integrity.

Smart farming systems are made up of a collection of gadgets that interact with one another and have varying degrees of limits. Many flaws arise as a result of device limitations, which make it hard to employ established tools and security procedures. Security is supported by technologies created for other systems, such as IoT or Industry 4.0, but their use necessitates processor and memory resources that some devices lack. It is, nevertheless, vital to be aware of existing vulnerabilities and to develop methods to mitigate the effects of occurrences. Then, at the highest layers, and on devices with the requisite resources, security measures can be implemented. To ensure fast and reliable operation, top-layer appliances with substantial computational capabilities should incorporate more rigorous security procedures.

In terms of ICT security, the rapid evolution and application of smart communication technologies, as well as the integration of IoT in addition to the digitalization and automation of business, introduces new threats and dangers in the worldwide market. In the dynamic and distributed cyber-physical environment, potential assaults on various smart agricultural systems can lead to major security challenges. Threats and attacks of this nature can cause severe disruptions to interconnected businesses. Furthermore, in the extensively mechanised environment of agriculture, smart technology and remote administration utilised in PA and smart farming are completely new to the industry's stakeholders, with the majority of new dangers in this domain being tightly linked to similar concerns in other industries. Cybersecurity, data integrity, and data loss are the most common dangers. Furthermore, because the PA industry employs heavy machinery that is linked to the internet, there are a slew of new vulnerabilities that could have severe effects.

The following paragraphs of this section take a more in-depth look at the security threats that modern agriculture faces. This section's broad literature analysis focuses on cyber security and IoT-related concerns in agriculture, which are still closely interwoven with PA, Agriculture 4.0, and smart farming.

6. Implementation and Testing:

6.1. Coding

Index Page:

```
{% extends 'base.html' %}
{% block content %}
<!doctype html>
<html lang="en">

<head>
  {% load static %}

  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Google Maps JavaScript library -->
  <script

src="https://maps.googleapis.com/maps/api/js?v=3.exp&libraries=places&key=AIzaSyC
dGv5cjpA0dMUCSolCf89tl_vgccGvsu0"></script>
  <!-- gmaps key call-->
  <script

src="https://maps.googleapis.com/maps/api/js?v=3.exp&libraries=places&key=AIzaSyC
6cmdeVLv2otcbqCAVSOTGX0f3rTkymok"></script>

  <!-- icon   -->
  <link rel='icon' href="{% static 'image/logo.png' %}">

  <!-- css call   -->
  <!-- <link rel="stylesheet" type="text/css" href="{% static
'dashboard/main.css' %}"> -->

  <!-- Bootstrap CSS -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet"
    integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
</head>
```

```
<body>


  <div class="wrapper">
    {% load static %}
    <div class="container-fluid">
      <h1>Map</h1>
      <div class="row">
        <div class="col-sm-6">
          {{m | safe}}
        </div>
        <div class="col-sm-3">
          <form action="." method="POST">
            {% csrf_token %}
            <div class="form-group">
              <label for="District">Enter District:</label>
              <select class="form-control" name="district" aria-
placeholder="Select District">
                <option value=""></option>
                <option value="102">Ahmednagar</option>
                <option value="114">Akola</option>
                <option value="115">Amarawati</option>
                <option value="108">Aurangabad </option>
                <option value="110">Beed</option>
                <option value="119">Bhandara </option>
                <option value="113">Buldhana </option>
                <option value="120">Chandrapur</option>
                <option value="100">Dhule </option>
                <option value="101">Jalgaon </option>
                <option value="107">Kolhapur</option>
                <option value="118">Nagpur</option>
                <option value="111">Nanded</option>
                <option value= "99">Nasik</option>
                <option value="112">Osmanabad </option>
                <option value="109">Parbhani</option>
                <option value="103">Pune</option>
                <option value= "97">Raigad </option>
                <option value= "98">Ratnagir</option>
                <option value="105">Sangli</option>
                <option value="104">Satara</option>
                <option value="106">Solapur</option>
                <option value= "96">Thane</option>
                <option value="117">Wardha</option>
                <option value="116">Yeotmal</option>
```

```
            </select>
          </div>
          <input type="submit" value="submit" class="btn btn-primary">
        </form>
        <br>
        <!-- <th scope="col">#</th> -->
        <div>
          <br>
          <h5>Yield Prediction </h5>
          <table class="table table-borderless">
            <thead>
              <tr>
                <!-- <th scope="col">#</th> -->
                <!-- <th scope="col">State Name </th> -->
                <th scope="col">Crop name</th>
                <th scope="col">Yield Prediction</th>
              </tr>
            </thead>
            <tbody>
              <tr>
                {% for row in compare_data %}
                <!-- <th scope="row">{{ state_id }}</th> -->
                <!-- <td>{{ row1.state_no   }}</td> -->
                <td>{{ row.crop_name  }}</td>
                <td>{{ row.yieldd     }}</td>
              </tr>
                {% endfor %}
            </tbody>
          </table>
        </div>
      </div>

      <div class="col-sm-3">
        <div id="carouselExampleControls" class="carousel slide" data-
ride="carousel">
          <div class="carousel-inner">
            <div class="carousel-item active">
              <img class="d-block w-100" src="media/{{first_news.image}}"
alt="First slide">
              <div class="carousel-caption">
                <div class="bg-success p-2 text-
light">{{first_news.title}}</div>
              </div>
            </div>
            <div class="carousel-item">
```

```
                <img class="d-block w-100" src="media/{{second_news.image}}"
alt="Second slide">
                <div class="carousel-caption">
                  <div class="bg-success p-2 text-
light">{{second_news.title}}</div>
                </div>
              </div>
              <div class="carousel-item">
                <img class="d-block w-100" src="media/{{third_news.image}}"
alt="Third slide">
                <div class="carousel-caption">
                  <div class="bg-success p-2 text-
light">{{third_news.title}}</div>
                </div>

              </div>
            </div>
            <a class="carousel-control-prev" href="#carouselExampleControls"
role="button" data-slide="prev">
              <span class="carousel-control-prev-i
              con" aria-hidden="true"></span>
              <span class="sr-only">Previous</span>
            </a>
            <a class="carousel-control-next" href="#carouselExampleControls"
role="button" data-slide="next">
              <span class="carousel-control-next-icon" aria-hidden="true"></span>
              <span class="sr-only">Next</span>
            </a>
          </div>
          <hr>
          <div>
            {% for news in three_news %}
            <!-- News Box -->
            <div class="card">
              <div class="row no-gutters">
                <div class="col-md-4">
                  <img src="media/{{news.image}}" class="card-img-top"
alt="{{news.title}}">
                </div>
                <div class="col-md-8">
                  <div class="card-body">
                    <h5 class="card-title">{{news.title}}</h5>
                    <p class="card-text"><a href="/detail/{{news.id}}" class="btn
btn-sm btn-primary">Read Full</a>
                    </p>
```

```
                </div>
              </div>
            </div>
          </div>
          <hr />
          {% endfor %}
        </div>
      </div>
    </div>



    <div>
      {% if chart %}
        <img src=" {{ chart | safe }}">
      {% endif %}
      <br><br><br><br><br><br>
    </div>


    {% endblock %}
    <!--jquery call-->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
      integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
      crossorigin="anonymous"></script>
    <!--jquery call-->
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <!-- Bootstrap CSS -->
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/js/bootstrap.bundle.min.js
"
      integrity="sha384-
Piv4xVNRyMGpqkS2by6br4gNJ7DXjqk09RmUpJ8jgGtD7zP9yug3goQfGII0yAns"
      crossorigin="anonymous"></script>
</body>

</html>
```

News.html

```
{% extends 'base.html' %}
{% block content %}
<!-- 14:25 / 43:04 -->
<title>{% block title %}News{% endblock %}</title>
<main class="container mt-3 mb-5">
    <!-- featured news -->
    <div class="row my-3">
        <!-- Left -->
        <div class="col-md-7">
            <div class="card mb-3">
              <img src="media/{{first_news.image}}" class="card-img-top"
alt="{{first_news.title}}">
              <div class="card-body">


                <h5 class="card-title">{{first_news.title}}</h5>
                <hr/>
                <p class="card-text"><a href="/detail/{{first_news.id}}"
class="btn btn-sm btn-primary">Read Full</a></p>
              </div>
            </div>
        </div>
        <!-- Right -->
        <div class="col-md-5" style="max-height: 450px; overflow: auto;">
            {% for news in three_news %}
            <!-- News Box -->
            <div class="card">
              <div class="row no-gutters">
                <div class="col-md-4">
                    <img src="media/{{news.image}}" class="card-img-top"
alt="{{news.title}}">
                </div>
                <div class="col-md-8">
                  <div class="card-body">
                    <h5 class="card-title">{{news.title}}</h5>
                    <p class="card-text"><a href="/detail/{{news.id}}" class="btn
btn-sm btn-primary">Read Full</a></p>
                  </div>
                </div>
              </div>
            </div>
            <hr/>
```

```
            {% endfor %}
        </div>
    </div>
    <!-- Category Wise News -->
    {% for category in three_categories %}
    <h3 class="border-bottom pb-1">{{category.title}} <a
href="/category/{{category.id}}" class="float-right btn btn-sm btn-dark">In this
Category</a></h3>
    <div class="row my-4">
        {% for news in category.news_set.all %}
            {% if forloop.counter < 5 %}
            <!-- News box -->
            <div class="col-md-3">
                <div class="card mb-3 shadow">
                <img src="/media/{{news.image}}" class="card-img-top"
alt="{{news.title}}">
                <div class="card-body">
                    <h5 class="card-title">{{news.title}}</h5>
                    <hr/>
                    <p class="card-text"><a href="#" class="btn btn-sm btn-
primary">Read Full</a></p>
                </div>
                </div>
            </div>
            {% endif %}
        {% endfor %}
    </div>
    {% endfor %}

    <!-- End Single Category -->
</main>
{% endblock %}
```

Views.py for index.html:

```
from django.shortcuts import render
import folium
import ee           # earth engine api
from folium import plugins
from requests import request
from dashboard.models import showCal,Feedback
from newsapp.models import News
from django.contrib import messages
from .utils import get_plot


# Create your views here.

#Add custom basemaps to folium

basemaps = {
    'Google Maps': folium.TileLayer(
        tiles = 'https://mt1.google.com/vt/lyrs=m&x={x}&y={y}&z={z}',
        attr = 'Google',
        name = 'Google Maps',
        overlay = True,
        control = True
    ),
    'Google Satellite': folium.TileLayer(
        tiles = 'https://mt1.google.com/vt/lyrs=s&x={x}&y={y}&z={z}',
        attr = 'Google',
        name = 'Google Satellite',
        overlay = True,
        control = True
    ),
    'Google Terrain': folium.TileLayer(
        tiles = 'https://mt1.google.com/vt/lyrs=p&x={x}&y={y}&z={z}',
        attr = 'Google',
        name = 'Google Terrain',
        overlay = True,
        control = True
    ),
    'Google Satellite Hybrid': folium.TileLayer(
        tiles = 'https://mt1.google.com/vt/lyrs=y&x={x}&y={y}&z={z}',
        attr = 'Google',
        name = 'Google Satellite',
        overlay = True,
        control = True
```

```
    ),
    'Esri Satellite': folium.TileLayer(
        tiles =
'https://server.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer/til
e/{z}/{y}/{x}',
        attr = 'Esri',
        name = 'Esri Satellite',
        overlay = True,
        control = True
    )
}

def add_ee_layer(self, ee_object, vis_params, name):

    try:
        # display ee.Image()
        if isinstance(ee_object, ee.image.Image):
            map_id_dict = ee.Image(ee_object).getMapId(vis_params)
            folium.raster_layers.TileLayer(
            tiles = map_id_dict['tile_fetcher'].url_format,
            attr = 'Google Earth Engine',
            name = name,
            overlay = True,
            control = True
            ).add_to(self)
        # display ee.ImageCollection()
        elif isinstance(ee_object, ee.imagecollection.ImageCollection):
            ee_object_new = ee_object.mosaic()
            map_id_dict = ee.Image(ee_object_new).getMapId(vis_params)
            folium.raster_layers.TileLayer(
            tiles = map_id_dict['tile_fetcher'].url_format,
            attr = 'Google Earth Engine',
            name = name,
            overlay = True,
            control = True
            ).add_to(self)
        # display ee.Geometry()
        elif isinstance(ee_object, ee.geometry.Geometry):
            folium.GeoJson(
            data = ee_object.getInfo(),
            name = name,
            overlay = True,
            control = True
        ).add_to(self)
        # display ee.FeatureCollection()
```

```
        elif isinstance(ee_object, ee.featurecollection.FeatureCollection):
            ee_object_new = ee.Image().paint(ee_object, 0, 2)
            map_id_dict = ee.Image(ee_object_new).getMapId(vis_params)
            folium.raster_layers.TileLayer(
            tiles = map_id_dict['tile_fetcher'].url_format,
            attr = 'Google Earth Engine',
            name = name,
            overlay = True,
            control = True
        ).add_to(self)

    except:
        print("Could not display {}".format(name))

# Add EE drawing method to folium.
folium.Map.add_ee_layer = add_ee_layer

# Create your views here.
def index(request):
    # Set visualization parameters.
    vis_params = {
    'min': 0,
    'max': 4000,
    'palette': ['006633', 'E5FFCC', '662A00', 'D8D8D8', 'F5F5F5']}

    #creating map object
    m= folium.Map(location=[20.5937,78.9629],zoom_start=4)

    # Add custom basemaps
    basemaps['Google Maps'].add_to(m)
    basemaps['Google Terrain'].add_to(m)
    basemaps['Google Satellite Hybrid'].add_to(m)


    # Add a layer control panel to the map.
    m.add_child(folium.LayerControl())

    # Add fullscreen button
    plugins.Fullscreen().add_to(m)
    #conversion in html
    m= m._repr_html_()

    #news renders
    first_news=News.objects.first()
    second_news=News.objects.order_by('image')[1]
```

```
    third_news=News.objects.order_by('image')[2]
    three_news=News.objects.all()[0:3]

    #AI renders
    #topfive=showCal.objects.all()
    state_id = request.POST.get("district")
    compare_data = showCal.objects.filter(state_no=state_id)
    x = [x.crop_name for x in compare_data]
    y = [y.yieldd for y in compare_data]
    chart =get_plot(x,y)



    context ={
        'm':m,
        #'topfive': topfive,
        'first_news':first_news,
        'second_news':second_news,
        'third_news' : third_news,
        'three_news':three_news,
        'state_id' : state_id,
        'compare_data' : compare_data,
        'chart' : chart,


    }
    return render(request, 'index.html',context)



def contact(request):
    if request.method =='POST':
        name=request.POST['name']
        email=request.POST['email']

        subject=request.POST['subject']
        message = request.POST['message']



        contact= Feedback.objects.create(
            name=name,
            email=email,
            subject=subject,
            message=message
```

```
        )
        messages.success(request,'Comment submitted!!!')
    return render(request, 'contact-us.html')
```

views.py for news.html:

```
from django.http import HttpResponse
from django.shortcuts import render
from .models import News,Category,Comment
from django.contrib import messages
from newsapp.models import News,Category,Comment


# Create your views here.



def news(request):
    first_news=News.objects.first()
    three_news=News.objects.all()[0:3]
    three_categories=Category.objects.all()[0:3]
    return render(request,'news.html',{
        'first_news':first_news,
        'three_news':three_news,
        'three_categories':three_categories
    })

def all_news(request):
    all_news=News.objects.all()
    return render(request,'all-news.html', {
        'all_news':all_news
    })

# Fetch all category
def all_category(request):
    cats=Category.objects.all()
    return render(request,'category.html',{
        'cats':cats
    })


# Fetch all category
def category(request,id):
    category=Category.objects.get(id=id)
```

```
    news=News.objects.filter(category=category)
    return render(request,'category-news.html',{
        'all_news':news,
        'category':category,
    })

# Detail Page
def detail(request,id):
    news=News.objects.get(pk=id)
    if request.method=='POST':
        name=request.POST['name']
        email=request.POST['email']
        comment=request.POST['message']
        Comment.objects.create(
            news=news,
            name=name,
            email=email,
            comment=comment
        )
        messages.success(request,'Comment submitted!!!')
    category=Category.objects.get(id=news.category.id)
    rel_news=News.objects.filter(category=category).exclude(id=id)
    comments=Comment.objects.filter(news=news,status=True).order_by('-id')
    return render(request,'detail.html',{
        'news':news,
        'related_news':rel_news,
        'comments':comments
    })


def Fungicides(request):
    return render(request,'Fungicides.html')

def Herbicides(request):
    return render(request,'Herbicides.html')

def Insecticides(request):
    return render(request,'Insecticides.html')

def Mothballs(request):
    return render(request,'Mothballs.html')

def Crop(request):
    return render(request,'Crop.html')
```

```
def aboutUs(request):
    return render(request,'about-us.html')

def contactUs(request):
    return render(request,'contact-us.html')
```

Model For Dashboard app:

```
from django.db import models
# Create your models here.
from tabnanny import verbose
from django.db import models
from matplotlib.pyplot import title
from django.utils.translation import gettext as _


class showCal(models.Model):
    #id= models.ForeignKey()
    state_no = models.FloatField(_("state_no"), max_length=255)
    crop_name= models.CharField(_("crop_name"), max_length=255)
    yieldd= models.FloatField(_("yield"), max_length=255)

    # def __str__(self):
    #     return self.state_no,self.yieldd,self.crop_name
    class Meta:
        verbose_name_plural='ShowCal'
    # def __str__(self):
    #     return self.state_no,self.yieldd,self.crop_name



class Feedback(models.Model):
    #news=models.ForeignKey(News,on_delete=models.CASCADE)
    name=models.CharField(max_length=100)
    email=models.CharField(max_length=200)
    subject=models.CharField(max_length=200)
    message=models.TextField()
    status=models.BooleanField(default=True)

    class Meta:
        verbose_name_plural='Feedbacks'

    def __str__(self):
        return self.subject
```

Model For News app:

```
from tabnanny import verbose

from django.db import models
from matplotlib.pyplot import title
from django.utils.translation import gettext as _



class Category(models.Model):
    title=models.CharField(max_length=200)
    category_image=models.ImageField(upload_to='imgs/')

    class Meta:
        verbose_name_plural='Categories'

    def __str__(self):
        return self.title

#news model creating here

class News(models.Model):
    category=models.ForeignKey(Category,on_delete=models.CASCADE)
    title=models.CharField(max_length=300)
    image=models.ImageField(upload_to='imgs/')
    detail=models.TextField()
    add_time=models.DateTimeField(auto_now_add=True)

    class Meta:
        verbose_name_plural='News'

    def __str__(self):
        return self.title

#cmt model
class Comment(models.Model):
    news=models.ForeignKey(News,on_delete=models.CASCADE)
    name=models.CharField(max_length=100)
    email=models.CharField(max_length=200)
```

```
    comment=models.TextField()
    status=models.BooleanField(default=True)


    class Meta:
        verbose_name_plural='Comments'

    def __str__(self):
        return self.comment
```

AI Prediction function 1:

```
import pandas as pd
import numpy as np
import csv
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
from math import sqrt

df=pd.DataFrame()


state_number=96
corp_no=2   # 2 =rice

def AI_write():
    state_number=96
    corp_no=2
    x = range(1, 121-96, 1)
    for corp_no in x:
        AI_Cal(state_number,corp_no)
        state_number += 1



def AI_Cal(state_number,corp_no):

    x = range(2, 32, 1)
    for corp_no in x:
```

```
        df = pd.read_csv('bk1.csv',index_col='Year')

        df=df[df.DistCode==state_number]

        df2=df.columns      #to get the column name
        df2=df2[corp_no]

        df=df.iloc[:,[0,corp_no]]
        df.head()
        df.plot(figsize=(12,8))


        df['1YearYield']=df[df2].shift(+1)
        df['2YearYield']=df[df2].shift(+2)
        df['3YearYield']=df[df2].shift(+3)

        df=df.dropna()


        lin_model=LinearRegression()

        model=RandomForestRegressor(n_estimators=100,max_features=3,
random_state=1)

        x1,x2,x3,y=df['1YearYield'],df['2YearYield'],df['3YearYield'],df[df2]
        x1,x2,x3,y=np.array(x1),np.array(x2),np.array(x3),np.array(y)
        x1,x2,x3,y=x1.reshape(-1,1),x2.reshape(-1,1),x3.reshape(-1,1),y.reshape(-
1,1)
        final_x=np.concatenate((x1,x2,x3),axis=1)


        X_train,X_test,y_train,y_test=final_x[:-3],final_x[-3:],y[:-3],y[-3:]
        model.fit(X_train,y_train)
        lin_model.fit(X_train,y_train)


        pred=model.predict(X_test)
        plt.rcParams["figure.figsize"] = (12,8)
        plt.plot(pred,label='Random_Forest_Predictions')
        plt.plot(y_test,label='Actual YIELD')
        plt.legend(loc="upper left")
        #plt.show()


        lin_pred=lin_model.predict(X_test)
```

```
        plt.rcParams["figure.figsize"] = (12,8)
        plt.plot(lin_pred,label='Linear_Regression_Predictions')
        plt.plot(y_test,label='Actual YIELD')
        plt.legend(loc="upper left")
        #plt.show()


        rmse_rf=sqrt(mean_squared_error(pred,y_test))
        rmse_lr=sqrt(mean_squared_error(lin_pred,y_test))

        print('Mean Squared Error for Random Forest Model is:',rmse_rf)
        print('Mean Squared Error for Linear Regression Model is:',rmse_lr)


        f= open("Yield.csv", "a", newline="")
        tup1 = (state_number,df2,rmse_rf,pred[0],rmse_lr,lin_pred[0])
        writer= csv.writer(f)
        writer.writerow(tup1)
        f.close()

#AI_write()
# call above function to execute script
```

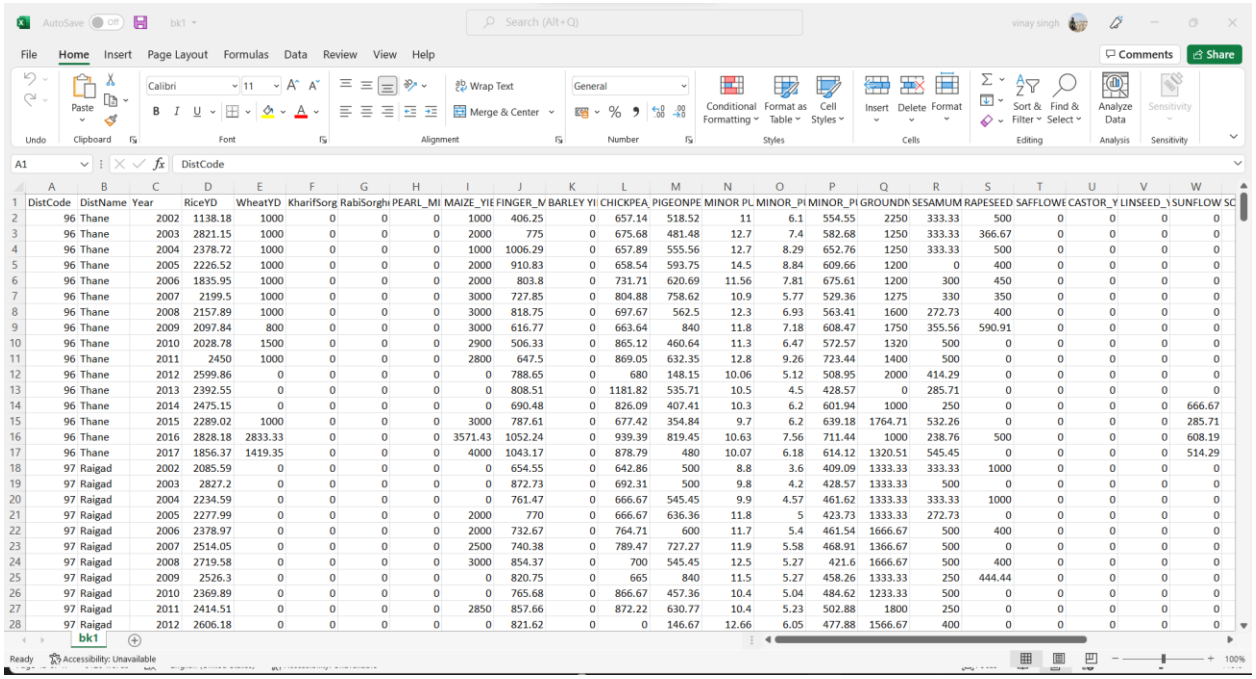
AI Prediction function 2:

```
import pandas as pd
import numpy as np
import csv
from pandas import read_csv
from matplotlib import pyplot
df=pd.DataFrame()

#given values
state_number=96
#state_name='Thane'
corp_no=2   # 2 =rice




def cal():
    x = range(96, 121, 1)
    for state_number in x:
        df = pd.read_csv('Yield.csv')
```

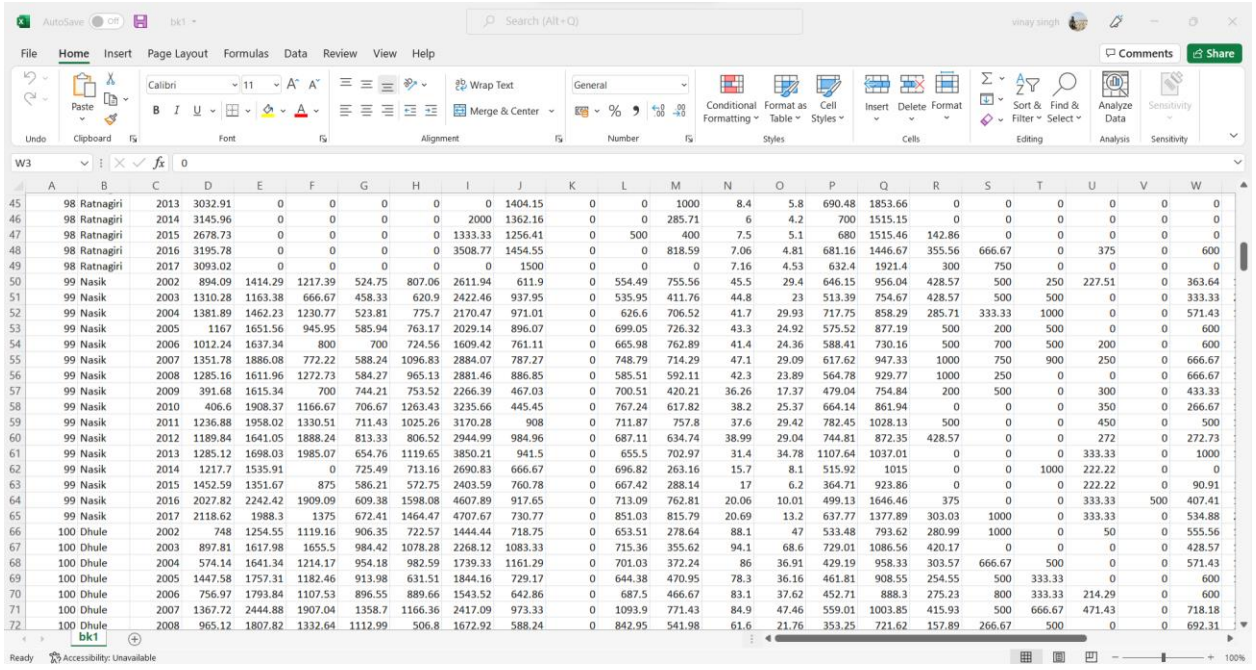
```
        df.RandomForest== df.RandomForest.astype(float)  #more accurate
        #df.LinearRegression== df.LinearRegression.astype(float) #not using
LinearRegrssion


        df=df[df.state_number==state_number]
        df2=df.sort_values(by='RandomForest',ascending=False)
        #print(df2)



        for i in range(1,6,1):
            df3=df2.iloc[[i]]
            df_state=df3['state_number'].to_string(index=False)
            df_crop=df3['crop_name'].to_string(index=False)
            df_RF=df3['RandomForest'].to_string(index=False)
            print(df_RF)
            print(df_state)

            f= open("Top5.csv", "a", newline="")

            #tup1 = (state_name,df2,rmse_rf,rmse_lr)

            tup1 = (df_state,df_crop,df_RF)
            writer= csv.writer(f)
            writer.writerow(tup1)
            f.close()
            state_number +=1


#cal()
#run this to get top 5 yields
```

6.2. **Screenshots**-

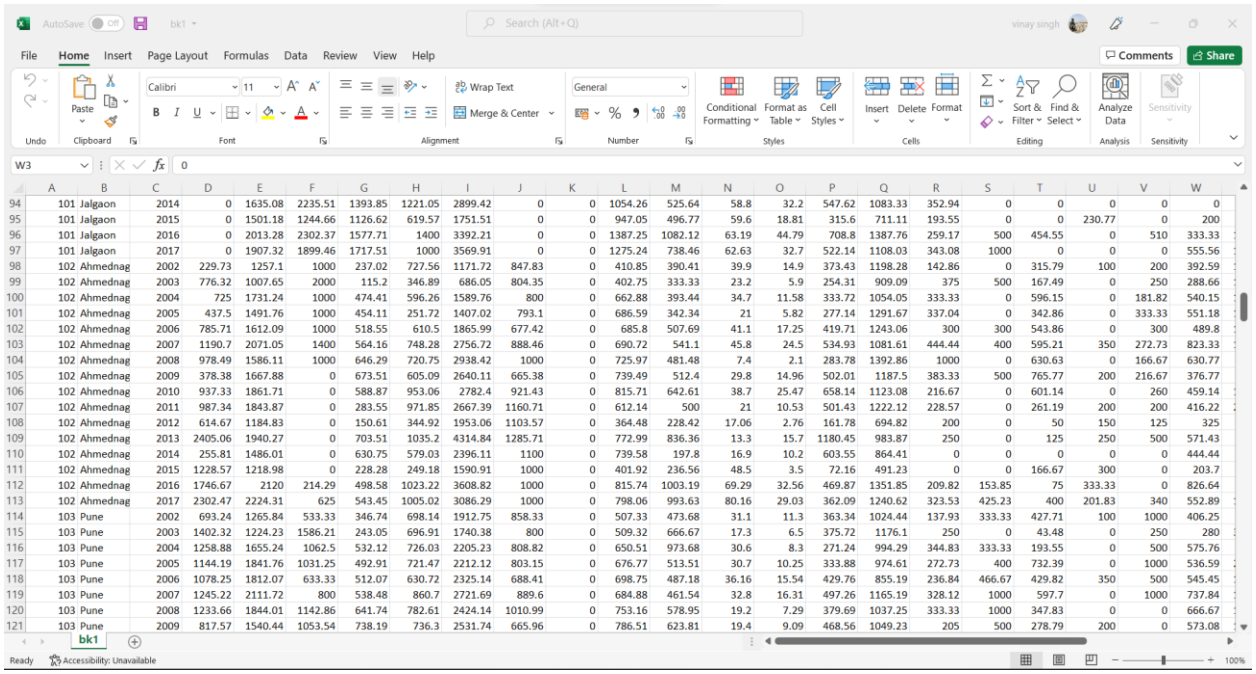
- Data set – (Training & Testing)-



DistCode	DistName	Year	RiceYD	WheatYD	KharifSorg	RabiSorghu	PEARL_MI	MAIZE_YIE	FINGER_M	BARLEY YI	CHICKPEA	PIGEONPE	MINOR PU	MINOR_PI	MINOR_PI	GROUNDN	SESAMUM	RAPESEED	SAFFLOWE	CASTOR_Y	LINSEED_Y	SUNFLOW	SC
96	Thane	2002	1138.18	1000	0	0	0	1000	406.25	0	657.14	518.52	11	6.1	554.55	2250	333.33	500	0	0	0	0	
96	Thane	2003	2821.15	1000	0	0	0	2000	775	0	675.68	481.48	12.7	7.4	582.68	1250	333.33	366.67	0	0	0	0	
96	Thane	2004	2378.72	1000	0	0	0	1000	1006.29	0	657.89	555.56	12.7	8.29	652.76	1250	333.33	500	0	0	0	0	
96	Thane	2005	2226.52	1000	0	0	0	2000	910.83	0	658.54	593.75	14.5	8.84	609.66	1200	0	400	0	0	0	0	
96	Thane	2006	1835.95	1000	0	0	0	2000	803.8	0	731.71	620.69	11.56	7.81	675.61	1200	300	450	0	0	0	0	
96	Thane	2007	2199.5	1000	0	0	0	3000	727.85	0	804.88	758.62	10.9	5.77	529.36	1275	330	350	0	0	0	0	
96	Thane	2008	2157.89	1000	0	0	0	3000	818.75	0	697.67	562.5	12.3	6.93	563.41	1600	272.73	400	0	0	0	0	
96	Thane	2009	2097.84	800	0	0	0	3000	616.77	0	663.64	840	11.8	7.18	608.47	1750	355.56	590.91	0	0	0	0	
96	Thane	2010	2028.78	1500	0	0	0	2900	506.33	0	865.12	460.64	11.3	6.47	572.57	1320	500	0	0	0	0	0	
96	Thane	2011	2450	1000	0	0	0	2800	647.5	0	869.05	632.35	12.8	9.26	723.44	1400	500	0	0	0	0	0	
96	Thane	2012	2599.86	0	0	0	0	0	788.65	0	680	148.15	10.06	5.12	508.95	2000	414.29	0	0	0	0	0	
96	Thane	2013	2392.55	0	0	0	0	0	808.51	0	1181.82	535.71	10.5	4.5	428.57	0	285.71	0	0	0	0	0	
96	Thane	2014	2475.15	0	0	0	0	0	690.48	0	826.09	407.41	10.3	6.2	601.94	1000	250	0	0	0	0	666.67	
96	Thane	2015	2289.02	1000	0	0	0	3000	787.61	0	677.42	354.84	9.7	6.2	639.18	1764.71	532.26	0	0	0	0	285.71	
96	Thane	2016	2828.18	2833.33	0	0	0	3571.43	1052.24	0	939.39	819.45	10.63	7.56	711.44	1000	238.76	500	0	0	0	608.19	
96	Thane	2017	1856.37	1419.35	0	0	0	4000	1043.17	0	878.79	480	10.07	6.18	614.12	1320.51	545.45	0	0	0	0	514.29	
97	Raigad	2002	2085.59	0	0	0	0	0	654.55	0	642.86	500	8.8	3.6	409.09	1333.33	333.33	1000	0	0	0	0	
97	Raigad	2003	2827.2	0	0	0	0	0	872.73	0	692.31	500	9.8	4.2	428.57	1333.33	500	0	0	0	0	0	
97	Raigad	2004	2234.59	0	0	0	0	0	761.47	0	666.67	545.45	9.9	4.57	461.62	1333.33	333.33	1000	0	0	0	0	
97	Raigad	2005	2277.99	0	0	0	0	2000	770	0	666.67	636.36	11.8	5	423.73	1333.33	272.73	0	0	0	0	0	
97	Raigad	2006	2378.97	0	0	0	0	2000	732.67	0	764.71	600	11.7	5.4	461.54	1666.67	500	400	0	0	0	0	
97	Raigad	2007	2514.05	0	0	0	0	2500	740.38	0	789.47	727.27	11.9	5.58	468.91	1366.67	500	0	0	0	0	0	
97	Raigad	2008	2719.58	0	0	0	0	3000	854.37	0	700	545.45	12.5	5.27	421.6	1666.67	500	400	0	0	0	0	
97	Raigad	2009	2526.3	0	0	0	0	0	820.75	0	665	840	11.5	5.27	458.26	1333.33	250	444.44	0	0	0	0	
97	Raigad	2010	2369.89	0	0	0	0	0	765.68	0	866.67	457.36	10.4	5.04	484.62	1233.33	500	0	0	0	0	0	
97	Raigad	2011	2414.51	0	0	0	0	2850	857.66	0	872.22	630.77	10.4	5.23	502.88	1800	250	0	0	0	0	0	
97	Raigad	2012	2606.18	0	0	0	0	0	821.62	0	0	146.67	12.66	6.05	477.88	1566.67	400	0	0	0	0	0	



98	Ratnagiri	2013	3032.91	0	0	0	0	0	1404.15	0	0	1000	8.4	5.8	690.48	1853.66	0	0	0	0	0	0	
98	Ratnagiri	2014	3145.96	0	0	0	0	2000	1362.16	0	0	285.71	6	4.2	700	1515.15	0	0	0	0	0	0	
98	Ratnagiri	2015	2678.73	0	0	0	0	1333.33	1256.41	0	500	400	7.5	5.1	680	1515.46	142.86	0	0	0	0	0	
98	Ratnagiri	2016	3195.78	0	0	0	0	3508.77	1454.55	0	0	818.59	7.06	4.81	681.16	1446.67	355.56	666.67	0	375	0	600	
98	Ratnagiri	2017	3093.02	0	0	0	0	0	1500	0	0	0	7.16	4.53	632.4	1921.4	300	750	0	0	0	0	
99	Nasik	2002	894.09	1414.29	1217.39	524.75	807.06	2611.94	611.9	0	554.49	755.56	45.5	29.4	646.15	956.04	428.57	500	250	227.51	0	363.64	
99	Nasik	2003	1310.28	1163.38	666.67	458.33	620.9	2422.46	937.95	0	535.95	411.76	44.8	23	513.39	754.67	428.57	500	500	0	0	333.33	
99	Nasik	2004	1381.89	1462.23	1230.77	523.81	775.7	2170.47	971.01	0	626.6	706.52	41.7	29.93	717.75	858.29	285.71	333.33	1000	0	0	571.43	
99	Nasik	2005	1167	1651.56	945.95	585.94	763.17	2029.14	896.07	0	699.05	726.32	43.3	24.92	575.52	877.19	500	200	500	0	0	600	
99	Nasik	2006	1012.24	1637.34	800	700	724.56	1609.42	761.11	0	665.98	762.89	41.4	24.36	588.41	730.16	500	700	500	200	0	600	
99	Nasik	2007	1351.78	1886.08	772.22	588.24	1096.83	2884.07	787.27	0	748.79	714.29	47.1	29.09	617.62	947.33	1000	750	900	250	0	666.67	
99	Nasik	2008	1285.16	1611.96	1272.73	584.27	965.13	2881.46	886.85	0	585.51	592.11	42.3	23.89	564.78	929.77	1000	250	0	0	0	666.67	
99	Nasik	2009	391.68	1615.34	700	744.21	753.52	2266.39	467.03	0	700.51	420.21	36.26	17.37	479.04	754.84	200	500	0	300	0	433.33	
99	Nasik	2010	406.6	1908.37	1166.67	706.67	1263.43	3235.66	445.45	0	767.24	617.82	38.2	25.37	664.14	861.94	0	0	0	350	0	266.67	
99	Nasik	2011	1236.88	1958.02	1330.51	711.43	1025.26	3170.28	908	0	711.87	757.8	37.6	29.42	782.45	1028.13	500	0	0	450	0	500	
99	Nasik	2012	1189.84	1641.05	1888.24	813.33	806.52	2944.99	984.96	0	687.11	634.74	38.99	29.04	744.81	872.35	428.57	0	0	272	0	272.73	
99	Nasik	2013	1285.12	1698.03	1985.07	654.76	1119.65	3850.21	941.5	0	655.5	702.97	31.4	34.78	1107.64	1037.01	0	0	0	333.33	0	1000	
99	Nasik	2014	1217.7	1535.91	0	725.49	713.16	2690.83	666.67	0	696.82	263.16	15.7	8.1	515.92	1015	0	0	1000	222.22	0	0	
99	Nasik	2015	1452.59	1351.67	875	586.21	572.75	2403.59	760.78	0	667.42	288.14	17	6.2	364.71	923.86	0	0	0	222.22	0	90.91	
99	Nasik	2016	2027.82	2242.42	1909.09	609.38	1598.08	4607.89	917.65	0	713.09	762.81	20.06	10.01	499.13	1646.46	375	0	0	333.33	500	407.41	
99	Nasik	2017	2118.62	1988.3	1375	672.41	1464.47	4707.67	730.77	0	851.03	815.79	20.69	13.2	637.77	1377.89	303.03	1000	0	333.33	0	534.88	
100	Dhule	2002	748	1254.55	1119.16	906.35	722.57	1444.44	718.75	0	653.51	278.64	88.1	47	533.48	793.62	280.99	1000	0	50	0	555.56	
100	Dhule	2003	897.81	1617.98	1655.5	984.42	1078.28	2268.12	1083.33	0	715.36	355.62	94.1	68.6	729.01	1086.56	420.17	0	0	0	0	428.57	
100	Dhule	2004	574.14	1641.34	1214.17	954.18	982.59	1739.33	1161.29	0	701.03	372.24	86	36.91	429.19	958.33	303.57	666.67	500	0	0	571.43	
100	Dhule	2005	1447.58	1757.31	1182.46	913.98	631.51	1844.16	729.17	0	644.38	470.95	78.3	36.16	461.81	908.55	254.55	500	333.33	0	0	600	
100	Dhule	2006	756.97	1793.84	1107.53	896.55	889.66	1543.52	642.86	0	687.5	466.67	83.1	37.62	452.71	888.3	275.23	800	333.33	214.29	0	600	
100	Dhule	2007	1367.72	2444.88	1907.04	1358.7	1166.36	2417.09	973.33	0	1093.9	771.43	84.9	47.46	559.01	1003.85	415.93	500	666.67	471.43	0	718.18	
100	Dhule	2008	965.12	1807.82	1332.64	1112.99	506.8	1672.92	588.24	0	842.95	541.98	61.6	21.76	353.25	721.62	157.89	266.67	500	0	0	692.31	



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
94	101	Jalgaon	2014	0	1635.08	2235.51	1393.85	1221.05	2899.42	0	0	1054.26	525.64	58.8	32.2	547.62	1083.33	352.94	0	0	0	0	0
95	101	Jalgaon	2015	0	1501.18	1244.66	1126.62	619.57	1751.51	0	0	947.05	496.77	59.6	18.81	315.6	711.11	193.55	0	0	230.77	0	200
96	101	Jalgaon	2016	0	2013.28	2302.37	1577.71	1400	3392.21	0	0	1387.25	1082.12	63.19	44.79	708.8	1387.76	259.17	500	454.55	0	510	333.33
97	101	Jalgaon	2017	0	1907.32	1899.46	1717.51	1000	3569.91	0	0	1275.24	738.46	62.63	32.7	522.14	1108.03	343.08	1000	0	0	0	555.56
98	102	Ahmednag	2002	229.73	1257.1	1000	237.02	727.56	1171.72	847.83	0	410.85	390.41	39.9	14.9	373.43	1198.28	142.86	0	315.79	100	200	392.59
99	102	Ahmednag	2003	776.32	1007.65	2000	115.2	346.89	686.05	804.35	0	402.75	333.33	23.2	5.9	254.31	909.09	375	500	167.49	0	250	288.66
100	102	Ahmednag	2004	725	1731.24	1000	474.41	596.26	1589.76	800	0	662.88	393.44	34.7	11.58	333.72	1054.05	333.33	0	596.15	0	181.82	540.15
101	102	Ahmednag	2005	437.5	1491.76	1000	454.11	251.72	1407.02	793.1	0	686.59	342.34	21	5.82	277.14	1291.67	337.04	0	342.86	0	333.33	551.18
102	102	Ahmednag	2006	785.71	1612.09	1000	518.55	610.5	1865.99	677.42	0	685.8	507.69	41.1	17.25	419.71	1243.06	300	300	543.86	0	300	489.8
103	102	Ahmednag	2007	1190.7	2071.05	1400	564.16	748.28	2756.72	888.46	0	690.72	541.1	45.8	24.5	534.93	1081.61	444.44	400	595.21	350	272.73	823.33
104	102	Ahmednag	2008	978.49	1586.11	1000	646.29	720.75	2938.42	1000	0	725.97	481.48	7.4	2.1	283.78	1392.86	1000	0	630.63	0	166.67	630.77
105	102	Ahmednag	2009	378.38	1667.88	0	673.51	605.09	2640.11	665.38	0	739.49	512.4	29.8	14.96	502.01	1187.5	383.33	500	765.77	200	216.67	376.77
106	102	Ahmednag	2010	937.33	1861.71	0	588.87	953.06	2782.4	921.43	0	815.71	642.61	38.7	25.47	658.14	1123.08	216.67	0	601.14	0	260	459.14
107	102	Ahmednag	2011	987.34	1843.87	0	283.55	971.85	2667.39	1160.71	0	612.14	500	21	10.53	501.43	1222.12	228.57	0	261.19	200	200	416.22
108	102	Ahmednag	2012	614.67	1184.83	0	150.61	344.92	1953.06	1103.57	0	364.48	228.42	17.06	2.76	161.78	694.82	200	0	50	150	125	325
109	102	Ahmednag	2013	2405.06	1940.27	0	703.51	1035.2	4314.84	1285.71	0	772.99	836.36	13.3	15.7	1180.45	983.87	250	0	125	250	500	571.43
110	102	Ahmednag	2014	255.81	1486.01	0	630.75	579.03	2396.11	1100	0	739.58	197.8	16.9	10.2	603.55	864.41	0	0	0	0	0	444.44
111	102	Ahmednag	2015	1228.57	1218.98	0	228.28	249.18	1590.91	1000	0	401.92	236.56	48.5	3.5	72.16	491.23	0	0	166.67	300	0	203.7
112	102	Ahmednag	2016	1746.67	2120	214.29	498.58	1023.22	3608.82	1000	0	815.74	1003.19	69.29	32.56	469.87	1351.85	209.82	153.85	75	333.33	0	826.64
113	102	Ahmednag	2017	2302.47	2224.31	625	543.45	1005.02	3086.29	1000	0	798.06	993.63	80.16	29.03	362.09	1240.62	323.53	425.23	400	201.83	340	552.89
114	103	Pune	2002	693.24	1265.84	533.33	346.74	698.14	1912.75	858.33	0	507.33	473.68	31.1	11.3	363.34	1024.44	137.93	333.33	427.71	100	1000	406.25
115	103	Pune	2003	1402.32	1224.23	1586.21	243.05	696.91	1740.38	800	0	509.32	666.67	17.3	6.5	375.72	1176.1	250	0	43.48	0	250	280
116	103	Pune	2004	1258.88	1655.24	1062.5	532.12	726.03	2205.23	808.82	0	650.51	973.68	30.6	8.3	271.24	994.29	344.83	333.33	193.55	0	500	575.76
117	103	Pune	2005	1144.19	1841.76	1031.25	492.91	721.47	2212.12	803.15	0	676.77	513.51	30.7	10.25	333.88	974.61	272.73	400	732.39	0	1000	536.59
118	103	Pune	2006	1078.25	1812.07	633.33	512.07	630.72	2325.14	688.41	0	698.75	487.18	36.16	15.54	429.76	855.19	236.84	466.67	429.82	350	500	545.45
119	103	Pune	2007	1245.22	2111.72	800	538.48	860.7	2721.69	889.6	0	684.88	461.54	32.8	16.31	497.26	1165.19	328.12	1000	597.7	0	1000	737.84
120	103	Pune	2008	1233.66	1844.01	1142.86	641.74	782.61	2424.14	1010.99	0	753.16	578.95	19.2	7.29	379.69	1037.25	333.33	1000	347.83	0	0	666.67
121	103	Pune	2009	817.57	1540.44	1053.54	738.19	736.3	2531.74	665.96	0	786.51	623.81	19.4	9.09	468.56	1049.23	205	500	278.79	200	0	573.08

We have total 30 Column and 400 rows of data

Prediction model output:

	A	B	C	D	E
1	model	id	state_no	crop_name	yieldd
2	newsapp.s	1	96	GROUNDNUT_YIELD	1305.15
3	newsapp.s	2	96	CHICKPEA_YIELD	978.3839
4	newsapp.s	3	96	MAIZE_YIELD	940
5	newsapp.s	4	96	FINGER_MILLET_YIELD	739.2224
6	newsapp.s	5	96	MINOR_PULSES_YIELD	594.5032
7	newsapp.s	6	97	GROUNDNUT_YIELD	2515.6663
8	newsapp.s	7	97	MAIZE_YIELD	1617
9	newsapp.s	8	97	OILSEEDS_YIELD	1443.3385
10	newsapp.s	9	97	FINGER_MILLET_YIELD	845.8276
11	newsapp.s	10	97	MINOR_PULSES_YIELD	570.9337
12	newsapp.s	11	98	MAIZE_YIELD	2358
13	newsapp.s	12	98	GROUNDNUT_YIELD	1625.5591
14	newsapp.s	13	98	FINGER_MILLET_YIELD	1372.1725
15	newsapp.s	14	98	OILSEEDS_YIELD	1275.7925
16	newsapp.s	15	98	SUGARCANE_YIELD	983.15165
17	newsapp.s	16	99	MAIZE_YIELD	3046.1514
18	newsapp.s	17	99	WheatYD	1694.099
19	newsapp.s	18	99	SOYABEAN_YIELD	1425.8121
20	newsapp.s	19	99	OILSEEDS_YIELD	1329.2556
21	newsapp.s	20	99	PEARL_MILLET_YIELD	1024.2851
22	newsapp.s	21	100	MAIZE_YIELD	2329.2042
23	newsapp.s	22	100	WheatYD	2019.4118
24	newsapp.s	23	100	KharifSorghumYD	1580.646
25	newsapp.s	24	100	FINGER_MILLET_YIELD	1405.3937
26	newsapp.s	25	100	SOYABEAN_YIELD	1376.0501
27	newsapp.s	26	101	MAIZE_YIELD	3156.8233
28	newsapp.s	27	101	WheatYD	2222.2066

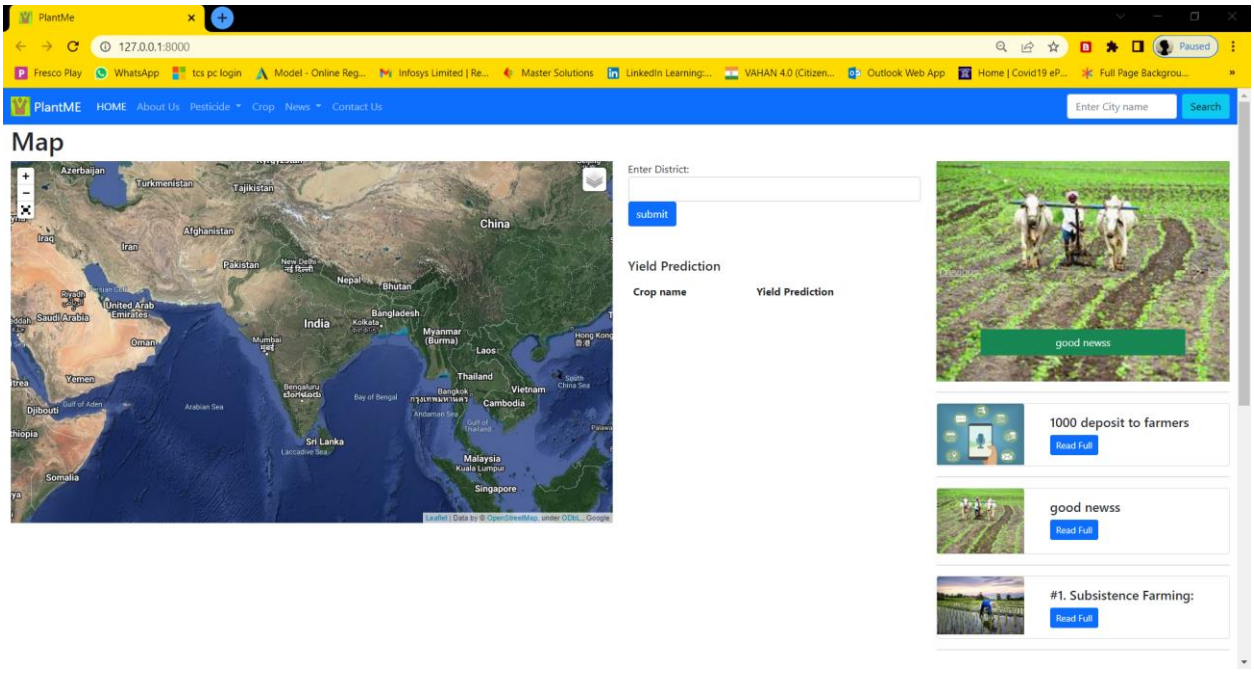
	A	B	C	D	E
29	newsapp.s	28	101	KharifSorghumYD	2138.3086
30	newsapp.s	29	101	SOYABEAN_YIELD	1824.0351
31	newsapp.s	30	101	OILSEEDS_YIELD	1529.3913
32	newsapp.s	31	102	MAIZE_YIELD	2605.2317
33	newsapp.s	32	102	WheatYD	1631.6691
34	newsapp.s	33	102	RiceYD	1628.0962
35	newsapp.s	34	102	SOYABEAN_YIELD	1518.6514
36	newsapp.s	35	102	FINGER_MILLET_YIELD	1199.5401
37	newsapp.s	36	103	MAIZE_YIELD	2875.9046
38	newsapp.s	37	103	SOYABEAN_YIELD	2445.5745
39	newsapp.s	38	103	WheatYD	1888.7594
40	newsapp.s	39	103	RiceYD	1693.6999
41	newsapp.s	40	103	FINGER_MILLET_YIELD	1204.3567
42	newsapp.s	41	104	MAIZE_YIELD	2309.1059
43	newsapp.s	42	104	WheatYD	1906.9069
44	newsapp.s	43	104	KharifSorghumYD	1696.2966
45	newsapp.s	44	104	RiceYD	1691.2587
46	newsapp.s	45	104	SOYABEAN_YIELD	1575.4818
47	newsapp.s	46	105	MAIZE_YIELD	2398.1645
48	newsapp.s	47	105	WheatYD	2039.4129
49	newsapp.s	48	105	RiceYD	1935.6354
50	newsapp.s	49	105	SOYABEAN_YIELD	1712.097
51	newsapp.s	50	105	OILSEEDS_YIELD	1282.2844
52	newsapp.s	51	106	MAIZE_YIELD	2253.8298
53	newsapp.s	52	106	WheatYD	1357.8324
54	newsapp.s	53	106	SOYABEAN_YIELD	1173.4478
55	newsapp.s	54	106	GROUNDNUT_YIELD	1150.1323
56	newsapp.s	55	106	OILSEEDS_YIELD	1030.6428

57	newsapp.s	56	107	RiceYD	2708.7159
58	newsapp.s	57	107	WheatYD	2442.8239
59	newsapp.s	58	107	SOYABEAN_YIELD	2267.2818
60	newsapp.s	59	107	MAIZE_YIELD	2062.7181
61	newsapp.s	60	107	KharifSorghumYD	2049.6494
62	newsapp.s	61	108	MAIZE_YIELD	2579.7356
63	newsapp.s	62	108	SAFFLOWER_YIELD	1640.4136
64	newsapp.s	63	108	WheatYD	1639.0618
65	newsapp.s	64	108	KharifSorghumYD	1395.7272
66	newsapp.s	65	108	SOYABEAN_YIELD	1200.266
67	newsapp.s	66	109	MAIZE_YIELD	1331.5928
68	newsapp.s	67	109	SOYABEAN_YIELD	1312.2299
69	newsapp.s	68	109	CHICKPEA_YIELD	1296.1645
70	newsapp.s	69	109	KharifSorghumYD	1141.6636
71	newsapp.s	70	109	WheatYD	1129.5812
72	newsapp.s	71	110	MAIZE_YIELD	1526.5777
73	newsapp.s	72	110	WheatYD	1024.9176
74	newsapp.s	73	110	KharifSorghumYD	920.1041
75	newsapp.s	74	110	GROUNDNUT_YIELD	892.6159
76	newsapp.s	75	110	PEARL_MILLET_YIELD	859.2157
77	newsapp.s	76	111	WheatYD	1448.6276
78	newsapp.s	77	111	MAIZE_YIELD	1193.4209
79	newsapp.s	78	111	SOYABEAN_YIELD	1141.6568
80	newsapp.s	79	111	KharifSorghumYD	1090.4878
81	newsapp.s	80	111	GROUNDNUT_YIELD	1079.1663
82	newsapp.s	81	112	SOYABEAN_YIELD	1698.8507
83	newsapp.s	82	112	MAIZE_YIELD	1504.0699
84	newsapp.s	83	112	WheatYD	1244.2197

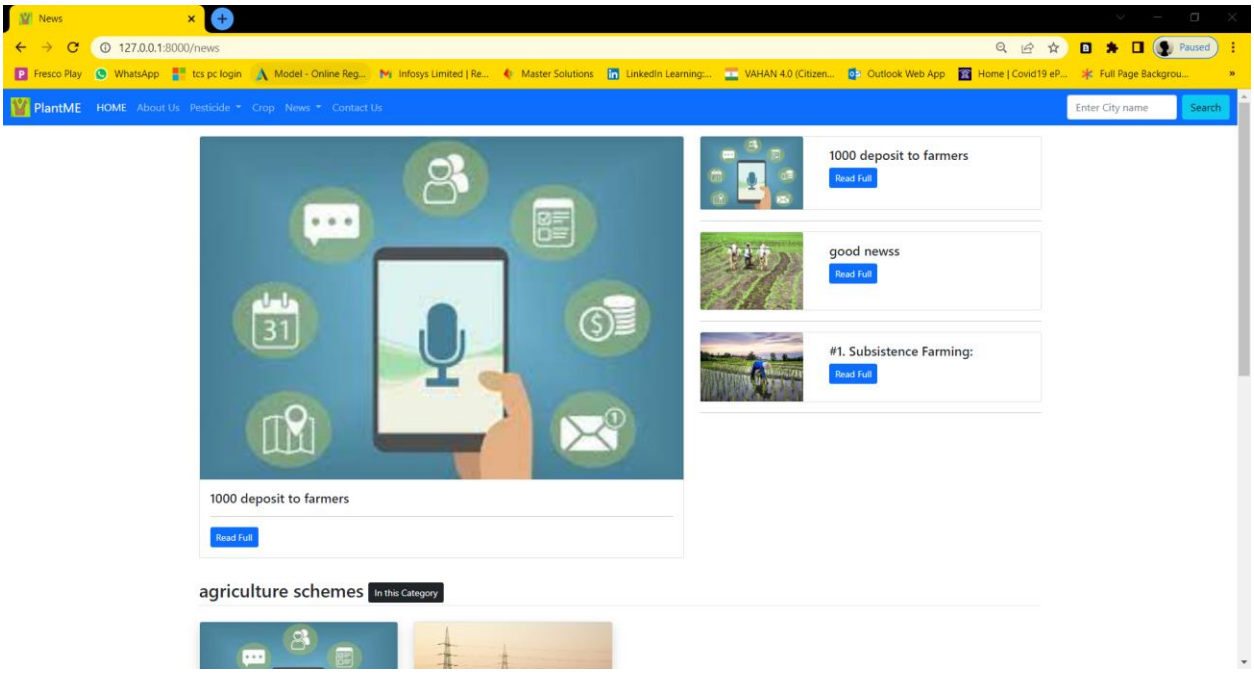
98	newsapp.s	97	115	WheatYD	1756.707
99	newsapp.s	98	115	SOYABEAN_YIELD	1263.6459
100	newsapp.s	99	115	OILSEEDS_YIELD	1250.6957
101	newsapp.s	100	115	CHICKPEA_YIELD	1134.0143
102	newsapp.s	101	116	MAIZE_YIELD	1791.9263
103	newsapp.s	102	116	GROUNDNUT_YIELD	1349.4873
104	newsapp.s	103	116	WheatYD	1223.5424
105	newsapp.s	104	116	SOYABEAN_YIELD	1110.1345
106	newsapp.s	105	116	OILSEEDS_YIELD	1049.8579
107	newsapp.s	106	117	GROUNDNUT_YIELD	1555.2608
108	newsapp.s	107	117	WheatYD	1514.5331
109	newsapp.s	108	117	PIGEONPEA_YIELD	937.4966
110	newsapp.s	109	117	SOYABEAN_YIELD	901.9301
111	newsapp.s	110	117	MAIZE_YIELD	820
112	newsapp.s	111	118	MAIZE_YIELD	2062.5
113	newsapp.s	112	118	WheatYD	1306.385
114	newsapp.s	113	118	RiceYD	1208.7046
115	newsapp.s	114	118	GROUNDNUT_YIELD	892.8778
116	newsapp.s	115	118	KharifSorghumYD	679.2247
117	newsapp.s	116	119	MAIZE_YIELD	2101.001
118	newsapp.s	117	119	RiceYD	1463.3241
119	newsapp.s	118	119	WheatYD	1098.6196
120	newsapp.s	119	119	GROUNDNUT_YIELD	799
121	newsapp.s	120	119	PIGEONPEA_YIELD	711.4168
122	newsapp.s	121	120	KharifSorghumYD	1367.5623
123	newsapp.s	122	120	GROUNDNUT_YIELD	1215.6667
124	newsapp.s	123	120	PIGEONPEA_YIELD	1047.3114

Website Interface:

Home page:



News app Page:



6.3. Testing-

Sr. No.	Test Case	Test Steps	Actual Result	Status
1	In Index page select district and transmit. Predicted values should be visible in form of table	Go to index page > Select district from drop-down and press submit	Output as expected	Pass
2.	In Index page select district and transmit. Predicted values should be visible in form of Chart	Go to index page > Select district from drop-down and press submit	Output as expected	Pass
4.	Login to Admin Page	Open Admin Page > Enter username and password and hit Transmit	Output as expected	Pass
3.	Check every page present in website should open without any crash or error	Do a sanity and open every page present in website	Output as expected	Pass
4.	Go to news and enter a comment.	Open a news and go to comment section of news > Add Name, Email and Comment. OK message should come	Output as expected	Pass
5.	Open database and check comment.	Login to Admin Page > Check Latest comment. Comment should be visible	Output as expected	Pass
6.	Delete comment from Admin page.	Login to Admin Page > Select comment and delete. Comment should not appear in news page	Output as expected	Pass
7.	Go To Contact Us page and submit feedback.	Open Contact us Page> Add your feedback. Ok message should come	Output as expected	Pass
8.	Open database and check Feedback.	Login to Admin Page > Feedback should appear in Admin Feedback section.	Output as expected	Pass

7. Conclusion-

The agricultural business has a number of obstacles, including a lack of appropriate irrigation systems, weeds, crop height-related concerns with plant monitoring, and harsh weather conditions. However, with the help of technology, performance may be improved, and therefore these issues can be resolved. It may be enhanced using AI-driven approaches such as remote sensors for detecting soil moisture content and GPS-assisted automatic watering. Farmers' difficulty was that precision weeding techniques were able to offset the high amount of crops lost during the weeding procedure. These self-driving robots not only increase productivity, but they also cut the use of unneeded pesticides and herbicides. Aside from that, farmers may use drones to successfully spray pesticides and herbicides on their farms, and plant monitoring is no longer a hassle. For starters, in agriculture difficulties, man-made brain power may be used to understand resource and employment shortages. In traditional tactics, a significant amount of effort was necessary to get agricultural parameters such as plant height, soil texture, and content, which necessitated manual testing, which was time-consuming. Quick and non-damaging high throughput phenotyping would be possible with the help of the different systems investigated, with the added benefit of flexible and favorable activity, on-demand access to information, and spatial goals.

8. Limitation and Future enhancement-

8.1. Limitation-

Farmers have a tendency to think of AI as something that only applies in the digital realm. They may be unable to see how it can assist them in working the real land. This isn't because they're fearful of the unknown or conservative. Their opposition stems from a lack of awareness of how AI technologies may be applied in the real world.

Because AgriTech vendors fail to adequately explain why their solutions are valuable and how they should be deployed, new technologies sometimes appear complex and unduly expensive. This is what occurs in agriculture when artificial intelligence is used. Although AI might be beneficial, technology companies must still do a lot of work to assist farmers in properly using it.

8.2. Future Enhancement-

We are planning to implement other modules which will consume other sorts of data and it will calculate prediction more accurately.
We are focused on improving enhancement of website performance.

9. Bibliography – references and links -

[1] https://www.analyticsvidhya.com/blog/2020/11/artificial-intelligence-in-agriculture-using-modern-day-ai-to-solve-traditional-farming-problems/

[2] https://www.mindtree.com/sites/default/files/2018-04/Artificial%20Intelligence%20in%20Agriculture.pdf

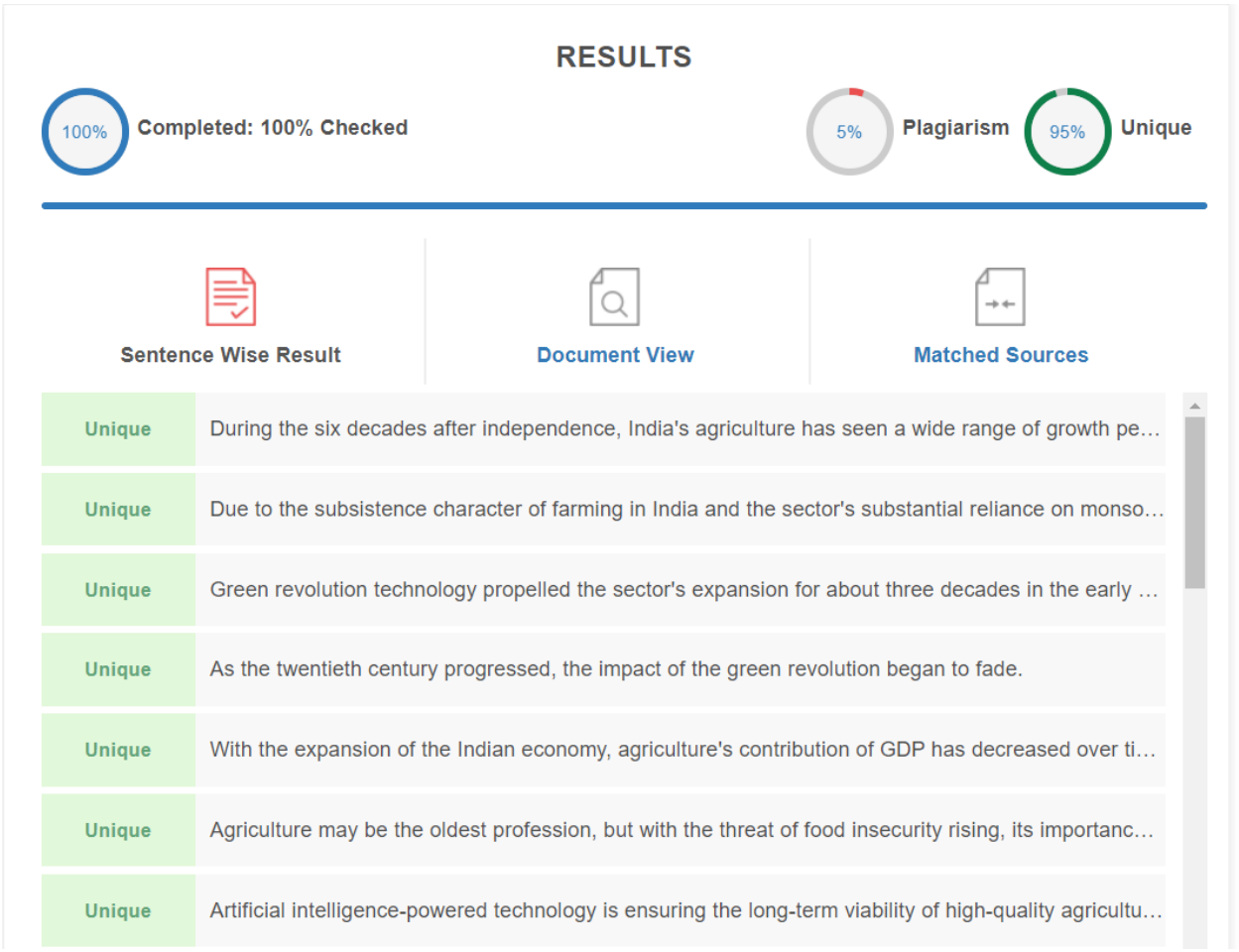
[3] https://www.sciencedirect.com/science/article/pii/S258972172030012X

[4] https://www.sciencedirect.com/science/article/pii/S258972172030012X#bbb0245

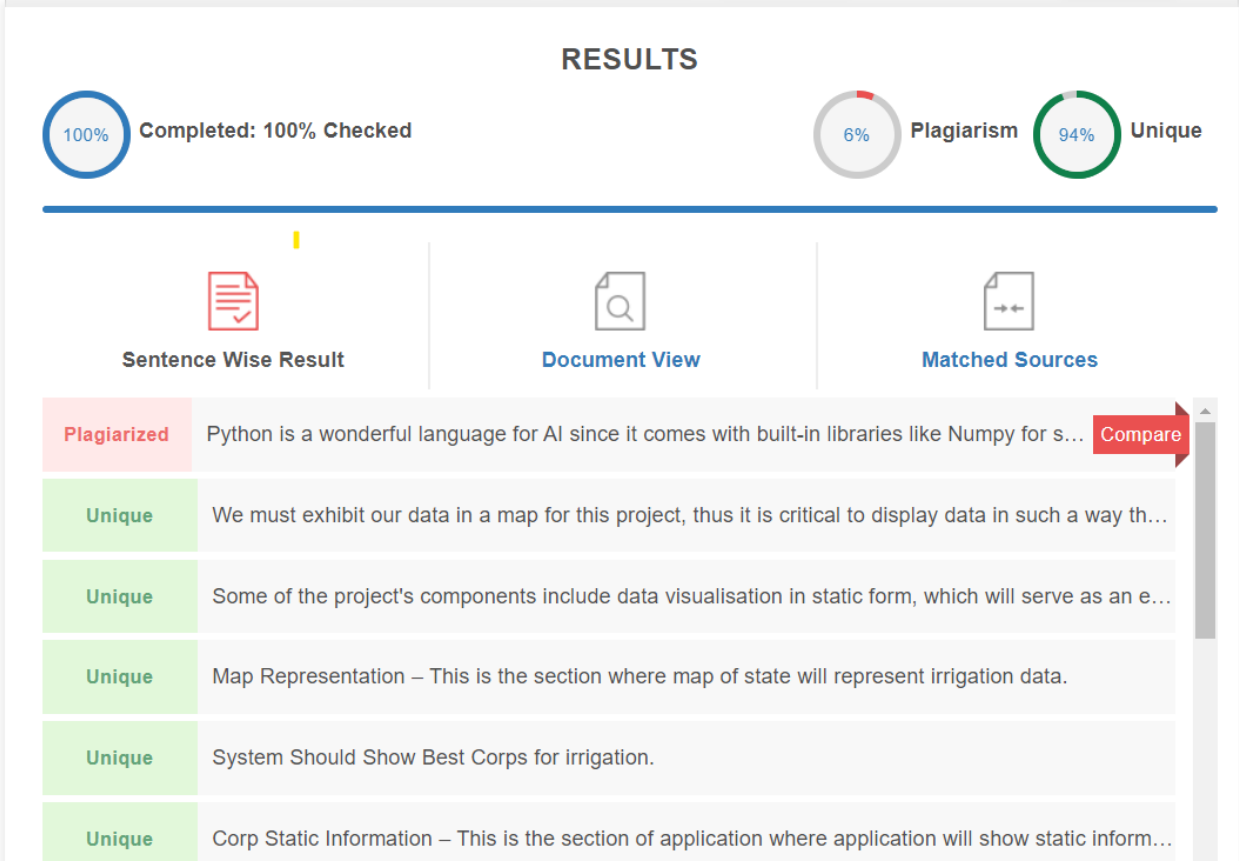
[5] https://www.forbes.com/sites/louiscolumbus/2021/02/17/10-ways-ai-has-the-potential-to-improve-agriculture-in-2021/?sh=16b50c157f3b

[6] https://intellias.com/artificial-intelligence-in-agriculture/

https://www.sciencedirect.com/science/article/pii/S2590005620300333


RESULTS
100%
Completed: 100% Checked
5%
Plagiarism
95%
Unique
Sentence Wise Result
Document View
Matched Sources
Unique
During the six decades after independence, India's agriculture has seen a wide range of growth pe…
Unique
Due to the subsistence character of farming in India and the sector's substantial reliance on monso…
Unique
Green revolution technology propelled the sector's expansion for about three decades in the early …
Unique
As the twentieth century progressed, the impact of the green revolution began to fade.
Unique
With the expansion of the Indian economy, agriculture's contribution of GDP has decreased over ti…
Unique
Agriculture may be the oldest profession, but with the threat of food insecurity rising, its importanc…
Unique
Artificial intelligence-powered technology is ensuring the long-term viability of high-quality agricultu…

2


RESULTS
100%
Completed: 100% Checked
6%
Plagiarism
94%
Unique
Sentence Wise Result
Document View
Matched Sources
Plagiarized
Python is a wonderful language for AI since it comes with built-in libraries like Numpy for s…
Compare
Unique
We must exhibit our data in a map for this project, thus it is critical to display data in such a way th…
Unique
Some of the project's components include data visualisation in static form, which will serve as an e…
Unique
Map Representation – This is the section where map of state will represent irrigation data.
Unique
System Should Show Best Corps for irrigation.
Unique
Corp Static Information – This is the section of application where application will show static inform…

RESULTS

100% Completed: 100% Checked

15% Plagiarism 85% Unique

 Sentence Wise Result

 Document View

 Matched Sources

Unique	• • HTML- HTML, or Hyper Text Markup Language, is the standard markup language for web-base…
Unique	Cascading Style Sheets (CSS) and computer languages like JavaScript can be of use.
Unique	• • Web browsers translate HTML documents received from a web server or locally saved files into …
Unique	HTML originally included visual clues for the document's appearance and logically represented the…
Unique	• The components that make up HTML pages are known as HTML elements.
Unique	Using HTML techniques, images and other objects, such as interactive forms, can be incorporated …
Unique	By specifying structural semantics for text components such as headers, paragraphs, lists, links, q…

Plagiarism Scan Report

Check Grammar | Make it Unique

Characters: 6576 **Words:** 972 **Sentences:** 20 **Speak Time:** 8 Min

GO PRO — No ads — Deep Search — Support — Accurate Reports

Step 1 :- Open Web Application on the Computer or Laptop. Step 2 :- Select Region for Prediction. Step 3 :- Select Sowing season Step 4:- Display the forecast data 5.2. Conceptual Model: Flowchart: - A flowchart is a formalized graphic representation of a logic sequence, work or manufacturing process, organization chart, or similar formalized structure. The purpose of a flow chart is to provide people with a common language or reference point when dealing with a project or process. Fig. 3 Flowchart 5.3. Basic Modules- There are 3 major modules. They are as follow. Search Module This is the module where user must provide input date based on which system will calculate the prediction. System will require information lake state name city name season of sowing seeds etc. Information Module This is the Module where user will be able to see static information and current affairs related to agriculture. It will display modern farming methods and schemes provided by government. This module will be educating farmers and will help them to understand modern methods of farming. Forecasting Module This is the Module which will be used to make calculation and prediction of best crops based on provided data by user. Application will predict and will sort the best sowing crops in sequence and system will rate them as well so user can understand how good it would be to plant crops. 5.4. Logic Diagram Activity Diagram: - Fig 4 Activity Diagram 5.5. Security Issues The application layer is responsible for providing services to end users, storing data, and making system choices. This layer's secu

88% Unique

13% Plagiarized

100%

View Plagiarized Sources

6% Plagiarized +

6% Plagiarized +