FARMER CONNECT EXPERT USING MACHINE LEARNING

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

Submitted by

P. SRIKOWNDINYA

Roll No. 23551F0048

Under the Supervision of

Mr. MD. Basheer Ali

Assistant Professor, MCA Dept.

DEPARTMENT OF COMPUTER APPLICATIONS

GODAVARI INSTITUTE OF ENGINEERING ANDTECHNOLOGY (A)

CHAITANYA KNOWLEDGE CITY, NH-16, RAJAMAHENDRAVARAM, AP

Jawaharlal Nehru Technological University, Kakinada, AP, India

March 2025

GODAVARI INSTITUTE OF ENGINEERING ANDTECHNOLOGY (A)

CHAITANYA KNOWLEDGE CITY, NH-16, RAJAMAHENDRAVARAM 533296,
AP, India

BONAFIDE CERTIFICATE

Certified that this project report "**FARMER CONNECT EXPERT USING MACHINE LEARNING**" is the Bonafide work of "**P. SRIKOWNDINYA (23551F0048)**", who carried out the project work under my supervision during the year 2024 - 2025 towards partial fulfillment of the requirements of the Degree in **MASTER OF COMPUTER APPLICATIONS** as administered under the Regulations GRCA-20 of **GODAVARI INSTITUTE OF ENGINEERING AND TECHNOLOGY, RAJAMAHENDRAVARAM**, AP, India and award of the Degree from Jawaharlal Nehru Technological University, Kakinada. The results embodied in this report have not been submitted to any other University for the award of any degree.

Signature of the Head of the Department

Dr. SHRIJA MADHU
HEAD OF THE DEPARTMENT
Department of Computer Applications
Godavari Institute of Engineering and
Technology(A)
NH-16, Chaitanya Knowledge city,
Rajamahendhravaram, AP, India.

Date:

Signature of the Supervisor

Mr. MD. BASHEER ALI
SUPERVISOR
Assistant Professor,
Department of Computer Applications
Godavari Institute of Engineering and
Technology(A)
NH-16, Chaitanya Knowledge city,
Rajamahendhravaram, AP, India.

External Viva voce conducted on ________________

Internal Examiner

External Examiner

GODAVARI INSTITUTE OF ENGINEERING ANDTECHNOLOGY (A)
CHAITANYA KNOWLEDGE CITY, NH-16, RAJAMAHENDRAVARAM 533296,
AP, India

CERTIFICATE OF AUTHENTICATION

I solemnly declare that this project report "**FARMER CONNECT EXPERT USING MACHINE LEARNING"** is the bona-fide work done purely by me, carried out under the supervision of **Mr. MD. BASHEER ALI,** towards partial fulfillment of the requirements of the Degree in **MASTER OF COMPUTER APPLICATIONS** as administered under the Regulations GRCA-20 of **GODAVARI INSTITUTE OF ENGINEERING AND TECHNOLOGY, RAJAMAHENDRAVARAM, AP, INDIA** and award of the Degree from Jawaharlal Nehru Technological University, Kakinada during the year 2024 – 2025.

I also declare that no part of this document has been taken up verbatim from any source without permission from the author(s)/publisher(s). Wherever few sentences, findings, images, diagrams or any other piece of information has been used for the sake of completion of this work, I have adequately referred to the document source. In the event of any issue arising hereafter about this work, I shall be personally responsible.

It is further certified that this work has not been submitted, either in part of in full, to any other department of the Jawaharlal Nehru Technological University Kakinada, or any other University, institution or elsewhere, in India or abroad or for publication in any form.

Signature of the Student
P. SRIKOWNDINYA (PIN: 23551F0048)

Date:

ACKNOWLEDGEMENT

I feel immense pleasure to express my sincere thanks and profound sense of gratitude to all those people who played a valuable role in the successful completion of my project.

I would like to express our gratitude to **Shri. K.V.V. SATYANARAYANA RAJU**, chairman, Chaitanya group of Institutions, **Shri. K. SASI KIRAN VARMA**, Vice Chairman **GIET** group of Institutions, for providing necessary infrastructure.

I am thankful to Principal **Dr. T. Jayananda Kumar** for permitting and encouraging us in doing this project.

My special thanks to **Dr. N. LEELAVATHY**, Vice Principal (Academics) for her constant guidance and motivation and for providing an excellent environment. We are truly grateful for her valuable suggestions and advice.

I am proudly grateful to express my deep sense of gratitude and respect towards **Dr**. **SHRIJA MADHU,** Head of the Department, MCA, whose motivation and constant encouragement has led to pursuing a project in the field of software development. I am very fortunate, for having her to enlighten us in all the aspects of life and transforming me to be an ideal individual in this field.

I am much obliged and thankful to our internal guide **Mr. MD. BASHEER ALI,** Assistant Professor, Department of MCA, for providing this opportunity and constant encouragement given by him during the course. I am grateful to his valuable guidance and suggestions during my project work.

My heart full thanks to my parents for putting me ahead of themselves, because of their hard work and dedication I have opportunities beyond my wildest dreams.

Finally, I express my thanks to other professors, classmates, friends and non- teaching staff who helped me for the completion of my project without infinite love and patience, this would never have been possible.

DECLARATION

I **P. SRIKOWNDINYA, (23551F0048),** declare that the project titled "**FARMER CONNECT EXPERT USING MACHINE LEARNING**" is a Bonafide work carried out by me and has not been submitted to any other university or college for the award of Degree.

(P. SRIKOWNDINYA)
(23551F0048)

TABLE OF CONTENTS

ABSTRACT

Farmer Connect Expert is a huge, innovative platform that tries to address the various needs of farmers and comes up with comprehensive solutions for common issues in agriculture. Inspired by the need to have easy access to relevant information on market trends and agriculture assistance, **Farmer Connect Expert** comes with various features aimed at making farmers optimize their productivity and make informed decisions. This includes providing soil fertility and weather-based personalized crop recommendations through API integration with real-time market trends and prices, as well as multilingual support in Telugu, Hindi, and English—all these will be further extended. The important features also include an AI-driven multi-lingual chatbot. Social interactions are possible, and there would also be experience-based sharing among farmers; marketplace facilities for online transactions between farmers and retailers would also be made available. Built on top of the MERN stack, **Farmer Connect Expert** leverages machine learning to offer intelligent crop recommendations. Key challenges included API integration, UI/UX design with multilingual support, and ensuring platform scalability. Some of the notable accomplishments include the development of personalized crop recommendations, multilingual chatbot functionality, and a community-driven social platform.

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The sector is twirling around agriculture. Today, farmers find that the crowd is enormous in every way-farming itself, the market tendencies, and the most recent updates the technologies have to offer in farming. Plus, there are many factors with which keeping farmers abreast is always challenging to have timely and precise up-dates concerning the conditions of the soil, weather, and crop management techniques. That is where Farmer Connect Expert Paper comes into play as a holistic platform to help farmers with making informed decisions and improving agricultural practices.

As it is, Farmer Connect Expert Paper uses the best usually brought by IoT technology to provide individual farm solution and therefore crop advice based on real-time data rather than soil fertility and weather reports. Smart irrigation and precision farming, both of which are integrated into the platform, help optimize farming operations while minimizing waste of resources and improving the yields from crops. Ac-cording to Kumar et al. (2022), IoT-based smart irrigation systems are being defined with most high performance by monitoring soil moisture and weather conditions by itself to contract irrigation in these most efficient manners to gain maximum water use with even crop growth. Thus, according to Singh et al. (2021), IoT technologies cater to many opportunities in agriculture-from crop management to resource allocation based on data-driven decision-making.

In addition, real-time access to updates about market tendencies and prices helps farmers make better decisions about how and when to sell their produce. The open APIs thereby provide market information, allowing farmers to remain competitive and maximize their profits in the market. Akhtar and Shafique (2023) provide an example of the integration of market.

1.1 Problem Statement

Farmers in the developing world are often unable to eliminate impediments that may exist under limited access to relevant information, fluctuating market trends, and lack of individualized support. They optimize their produce but never know the prices they need to furnish for different crops in an appropriate way or receive timely agricultural information. Language barriers and fragmented services inhibit Agri-entrepreneurial decision-making. There are no platforms that offers personalized customer service, timely market information, and community support to such a large extent.

The following major subjects are connected to this study's research goals:

1. **Aggrotech:** This sector applies technology and innovation to agriculture as a way of improving farming practices. It includes the developing and usage of platforms like Farmer Connect Expert which give farmers the tools for optimizing yield, managing resources, and enhancing their decision-making skills.

1. **Machine Learning and Artificial Intelligence:** The nature of machine learning techniques on crop recommendation systems and AI chatbots dovetails with the notion of personalizing assistance to farmers from the platform. This fact also entails supervised learning, reinforcement learning, and natural language processing (NLP) to advance decision support in agriculture.

2. **Data Science and Big Data:** The subject examines the big agricultural-scaled data of weather reports, soil fertility, market trends, and so forth, to provide insights that farmers can directly use. The understanding of data collects, preprocess, and analysis is a critical link in optimizing agricultural outputs.

3. **Mobile and Web Development:** The MERN stack (MongoDB, Express.js, React.js, Node.js) used for building this platform is one front that involves research on how to design, develop and deploy scalable web applications to serve diverse users in the agriculture sector.

4. **Multilingual Systems and Localization:** This shall include research in multilingual support, translation models, and UI/UX design for a diverse, multilingual user base to reach farmers in various languages.

5. **E-commerce and Marketplaces:** The feature for connecting farmers with retailers and services like machinery rentals and pesticide sales would address the areas of e-commerce platforms, online transactions, and digital marketplace models within agriculture.

1.2 Motivation

The Farmer Connect Expert project was conceived and grounded in the pressing need for a complete response to the multiple challenges farmers face in this fast-changing climate of agriculture. Farmers in many developing countries have an urgent need for instant information on market prices, flexible market access, and data-driven evaluations that would otherwise yield higher productivity and sustainability on their part. With climate changes, erratic weather conditions, and pest outbreaks, theirs has become a harder task rather than an easier one.

There is an imminent necessity for an interactive and accessible platform that would provide individual support to farmers against those challenges. Farmer Connect Expert serves this purpose by unifying solutions that offer the farmer tools for data-based decision-making for real-time market trends and localized advice on various aspects such as soil health, weather patterns, etc.

The agro-motivations behind this platform include improved productivity and sustainability in agriculture. By applying machine learning and artificial intelligence, Farmer Connect Expert provides farmers with evidence-based crop recommendations and timely information; while also hatching community life and knowledge transfer among farmers, a quality often absent in rural settings.

Along with this, the motivation for multilingualism is to broaden accessibility of the platform for a wide range of farmers from different languages, thus ensuring the influence of technology is spread among varied people.

Farmer connect Expert’s long-term vision is to create a major transformation in agriculture by improving farmers' livelihoods, their access to markets, and ultimately food security and sustainable farming practices across the globe. It is the conviction of this project that technology can be a major facilitator of change, bringing together the best of innovation and agriculture to improve the lives of farmers and their communities.

1.3 Objectives

The objective of the Farmer Connect Expert project are to developing an intelligent system for personalized crop recommendation with respect to local soil fertility, weather, and environmental factors that could help farmers optimize farming practices and yield maximization. The platform will offer real-time market trends and prices through various integrated APIs so farmers can make informed choices about when and where to sell produce for maximum returns.

However, the project also seeks to make the platform as accessible as possible to all users through multilingual support - Telugu, Hindi, and English - and provisions for further languages in the future. This is not all as another of the major objectives is to implement an OpenAI-based AI chatbot. Thus, this would provide real-time, relevant, and helpful information to farmers, helping them to enhance user experience and accessibility to share useful information in several languages.

Farmer Connect Expert is interested in creating an association of farmers will serve as a platform for farmers to discuss experiences, get advice, and even work together to share knowledge. Building a transparent agricultural marketplace - where farmers could list their products and where retailers could rent agricultural machinery, tools, and pesticides - is the goal of the project.

This platform is built to ensure scalability and high performance to meet the user demands as well as services they require. It also seeks to improve user engagement through initiatives like forums, expert Q&A sessions, and knowledge transfer among the community in promoting continual learning among the farming community.

The last thing to do is to create a digital platform for facilitating easy access for farmers by which they can realize the benefits of efforts made into harvesting productivity and sustainability through data-driven insights and resource usage. The long-term future vision is to take the platform into all types of communities across the globe for its offerings, language support, and market capabilities.

2. LITERATURE SURVEY

2.1 Introduction

Farmer Connect Expert Project has a goal to create a comprehensive, easy-to-use platform that empowers farmers through personalized guidance, access to markets, and a vibrant community. Platform uses features like personalized crop advisory, local farm produce and price trends and indoor weather prediction and requirements to touch the agricultural sector and help farmers improve their yield.

Kumar et al. (2022) Present a review of IoT-based smart irrigation systems showing that these technologies can strongly enhance water management through minimization of wasting and non-optimal Crops irrigation. Their review emphasizes the use of IoT sensors that can continuously track soil moisture, weather, and plant health, allowing farmers to respond swiftly and knowledgeably. Efficient irrigation practices of crops are crucial for sustainable agriculture, which is especially beneficial in water-scarce areas [1]. Likewise, Mehta and Rao (2023) introduce a (Internet of Things) IoT-based smart irrigation system, which automatically distributes water based on real-time surrounding conditions, leading to improved crop yield and water saving [10].

One more area where IoT is making a difference in agriculture is in precision farming. Akhtar and Shafique (2023) discuss how IoT can optimize the best farming practices according to the data of soil condition, plant health, and the level of nutrients, which enables farmers to apply fertilizers and pesticides in an optimal way. Zhang et al. (2020) support this with an overview of IoT-based smart agriculture systems, including sensor networks and cloud computing, which provide real-time data that can be used to improve decision-making across a range of aspects of farming [4].

Singh et al. (2021) elaborates more on the general implications of IoT in agriculture, highlighting how IoT technologies not only boost resource management but also aid in increasing productivity and sustainability. Their review reflects on the future of IoT in agriculture by identifying areas such as crop health monitoring, automated systems for irrigation and fertilization, which would lead to increased crop production with minimal input costs [2]. In developing this perspective, Gupta and Sharma (2022) highlight the design of an IoT-based smart farming system where sensors for soil moisture, temperature, and other environmental factors are integrated for the better management of resources and increased productivity of crops [5].

Crop monitoring systems are another prominent use of IoT in modern agriculture. According to Sharma and Choudhary [8], the IoT based crop monitoring and irrigation control systems can monitor multiple parameters in the environment such as soil moisture, humidity, temperature. It ensures the irrigation system for crops with the amount of vital water and nutrients required for good yield and reduces the excessive use of resources. Integration of IoT for crop monitoring has been one of the developments contributing to sustainable farming practices [7].

The several challenges that also come with the adoption of IoT technologies by farmers include the availability of a reliable connectivity network and technical expertise. In this regard, Kumar and Reddy (2021) analyze the implementation of IoT in smart agriculture and the challenges facing it; they note that infrastructure limitations and the lack of technical knowledge among farmers may affect widespread adoption of these emerging technologies. The study again highlights the need to enable farmers with the required training and resources for successful implementation of IoT solutions in agriculture [9].

With the increasing adoption of IoT, it is expected to contribute much toward the transformation of agriculture, leading to a more sustainable, efficient, and productive agricultural ecosystem [6][8].

2.2 Existing System

Farmers are dependent today on traditional methods, fragmented information sources, and limited access to technology for their requirements in agriculture. Such systems include word-of-mouth communication, basic weather forecasts, and local agricultural extension services. In some cases, farmers would have access to agricultural advisory services or market information, but most are general and neither personalized nor real-time.

Furthermore, the farmer has been facing language barriers and limited access to the digital tools that could enable him/her to make data-driven decisions. Many of the existing platforms or tools are too complicated or not farmer-friendly enough for people with underdeveloped digital literacies.

Disadvantages of Existing System:

1. **Scattered Information:** Information on crops, pests, market prices, and weather is often dispersed in different platforms and makes it difficult for farmers to get one unified source of knowledge.

2. **Technology Access Restriction:** Access to modern digital tools is limited for many farmers, which prevents them from making decisions based on data.

3. **Language Barriers:** Most agricultural platforms or advisory services are not multilingual, so most farmers who cannot speak English, Hindi, etc., cannot enjoy their services.

4. **Limited Market Access:** Farmers are unable to reach markets that afford fair prices to their produce or get paid lesser prices through the networks of middlemen, thereby earning very little.

5. **Low Adoption AI:** There is little use of artificial intelligence and machine learning in existing systems to provide intelligent and automated support to farmers.

2.3 Summary

They are outdated, uncoordinated, and ineffective systems that allow farmers little opportunity for productivity enhancement, timely market access, or even companionship with other farmers. There is a need for an integrated platform that would bring all aspects of personalized help, real-time data, and community support into one space that is both accessible and user-friendly.

3. PROPOSED SYSTEM

3.1 Description

Farmer Connect Expert serves as a holistic solution to these gaps and incorporates personalized crop recommendations, market insights, multilingual support, and community features. This will leverage machine learning algorithms and provide the farmers with data-based insights and recommendations applicable to the local situation in terms of soil fertility, weather, and climate patterns. Also, Farmer Connect Expert will use open APIs to provide current market prices and trends and act as a social platform for farmers to network, share experiences, and learn from one another. Finally, a very transparent marketplace for all agricultural products and services, making it very easy to connect with suppliers and retailers, will finally be put into place within the site. It will also include an AI-powered chatbot to respond to farmer queries in multiple languages and enhance user experience.

Advantages of Proposed System:

1. **Local Customized Assistance:** It provides a local-level recommendation based on environmental factors to help farmers optimize crop production and farming practices.
2. **Real-time Market Data:** It includes getting the latest market trends and prices so that a farmer can make an informed decision regarding selling the produce at the best price.
3. **Multilingual Support:** Farmer Connect Expert has multi-language support, hence ensuring coverage for many farmers across the regions.
4. **AI-Irrational Chatbot:** An AI chatbot provides real-time relevant responses for queries of farmers thus making inquiry and support in many languages much easier.
5. **Community Participation:** Brains can be developed into a community farm online that will have farmers share experiences and challenges of farming with one another.
6. **Transparent Marketplace:** An expressly-catered marketplace allows farmers to list their availabilities in products while renting in agricultural implement tools and pesticides for fair and transparent transactions.
7. **Scalability Performance:** Built on a scalable architecture, the system can handle demand from users when it increases and still give reliable performance with increasing features.
8. **User-Friendly:** The platform has trails, thus, easily accessible by farmers who do very baseline and above up to better technological knowledge.

3.2 Implementation

3.2.1 System Architecture

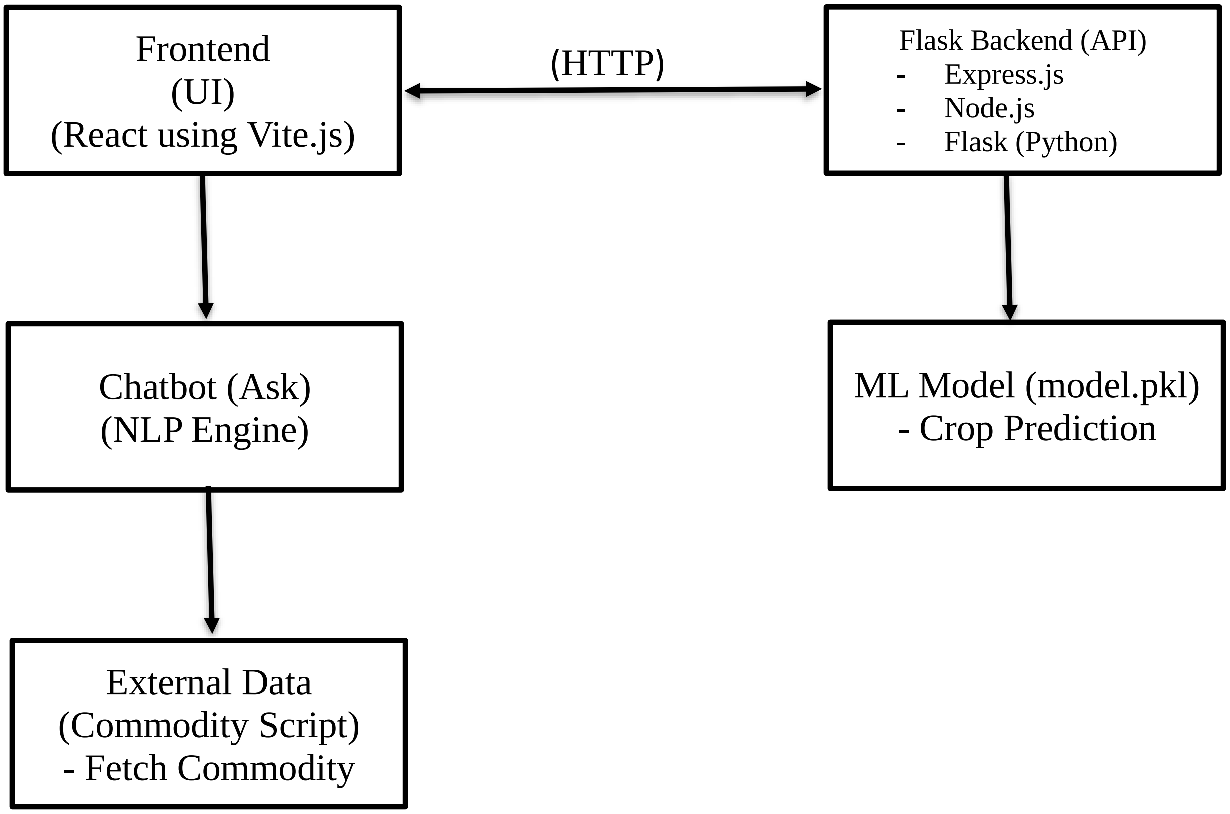


Fig 3.2.1 System Architecture

3.2.2 System Development Environment

Python:

The profusion of available libraries coupled with the sheer elegance of its syntactical formations makes Python not only one of the best interpreted languages but also interpreted object-oriented scripting languages, interactive and thus very much executable besides pure interpreted ones. Python was designed with emphasis on clarity of code; it uses more English words than punctuation and essentially less syntactical constructs compared to other languages.

Python is versatile: While you are executing Python, it works through the interpreter. Your code can thus be run without compilation. The same general area as PHP or PERL.

Python is Dynamic: You can converse with the interpreter and build your program by typing commands in a Python prompt.

Object-oriented programming in Python: The Object-oriented support of Python implies that the programmed can be packaged as objects.

Python is an excellent first language-Python is the great language for a beginner since it can be used to develop varieties of applications, from text editors to web browsers to games.

Background of Python

Guido van Rossum in the late 1980s and early 1990s at the National Research Institute for Mathematics and Computer Science in The Netherlands developed the language Python. Also, Python picked a number of ideas from other languages such as Unix shell, ABC, Modula-3, C, C++, Algol-68, Small Talk, etc.

The source codes of Python are copyright protected. Like the Perl source code previously, it is now available on a GNU General Public License (GPL).

Though there exists a core development team at the university now to take care of its maintenance, Guido van Rossum still plays an important part in holding the steering towards the development of Python.

Python Features

Some of the features built into Python are the API calls and libraries, which function on Windows-based platforms such as Windows MFC. Some of the other features are:

- Easily learned
- Easy to read
- Easy maintenance
- Extensive standard libraries
- Interactivity
- Portability
- Extendable
- Computer databases. Python helps to access virtually all commercial database systems.
- GUI programming. Python simplifies development and deployment of GUI applications across diverse platforms, including Mac, as well as the UNIX X Window system.
- Scalability. Python offers more organization and guidance while dealing in giant projects compared to shell programming.
- Sharing is another constructive aspect of Python, which has many important features.
- Examples include object-oriented programming (OOP) as well as functional and structured programming.
- Build larger systems or smaller programs interpreted to byte code.

- Supports dynamic checking with complex dynamic data types.
- Implement garbage collection.
- Integration with Java, C, C++, COM, ActiveX, and CORBA is smooth.

Python is available on diverse operating systems, including Linux and Mac OS X. Let's go through a Python setup process.

Getting Python

According to detailed literature, Python's latest source code, binaries, documentation, news, and many reside at the official website https://www.python.org.

Setup of Windows

- This document will detail all the steps that you need to take to download and install Python on your PC that runs on Windows.
- After having opened your web browser, you need to go to the website - https://www.python.org/downloads/.
- To access the installer for Windows, go to the python-XYZ.msifile link where XYZ will be the actual version you are about to install.
- This Python-XYZ.msi installation will require Microsoft's installation 2.0 which is supported only by Windows whether or not you are sure your machine supports MSI just download and run it locally.
- Read the document you just saved. This will lead into an ordinary python installation procedure. Do not change anything from its default state and wait for the finish of installation.
- Syntax in Python resembles that of Perl, C and Java. There are, however, many differences from those languages.

MERN STACK:

MERN stack is an online application development technology. It includes MongoDB, Express.js, React.js, and Node.js simultaneously. These four technologies provide a complete end-to-end solution for web development-from database management to front-end user interfaces.

Components of the MERN Stack:

MongoDB:

MongoDB is a type of database that has been developed with NoSQL features. It stores data in a flexible and JSON-like format called BSON (Binary JSON). It is highly capable for horizontally scaling and can also handle large amounts of unstructured data.

In a MERN stack application, MongoDB is concerned with the database layer of your application. All application data, such as user profiles and product details, is stored there.

MongoDB has the potential to cater to highly scalable applications of complex queries and hence is a great choice when building modern day web applications.

Example of MongoDB document (in BSON format):

```
{
  "_id": "643216",
  "name": "cgchgch",
  "email": "gchtfjgbkj@kghj.com",
  "age": 30,
  "address": {
    "street": "123 Main St",
    "city": "New York"
  }
}
```

React:

React is an open-source JavaScript library for creating user interfaces, especially for single-page applications (SPAs)-fast, interactive experiences. It develops at Facebook, and now it is supported by Facebook with a lot of other developers in the revolutionary process.

React allows developers to write web applications that can change data without reloading the page. It works with the philosophy that reusable components handle their own state, and then compose them to build complex UIs. It is mostly defined as a view in MVC or Model View Controller, meaning the main focus is rendering the UI and managing state.

React is a widely-used JavaScript library for developing user interfaces mainly for single-page applications. It was originally created and is maintained by Facebook, with many contributions from the open-source community.

Concepts of React

To solve the complexity of UIs, React came up as a solution by splitting UIs into reusable components. These components are responsible for rendering UI elements and holding their state and logic.

Here are some of the key concepts and features of React:

1. Components

A React app is made out of components. Components are reusable pieces of UI that can manage their own state and behavior. Each component can be considered a function or class written in JavaScript that returns HTML-like syntax called JSX (we'll talk more about JSX shortly).

Components are of two types in React:

Functional components-simplest, and often for purely presentational purposes.

Class components-more complex component, holding internal state and lifecycle methods. Their use has diminished in modern React, with functional components being preferred-functional components with hooks.

Example of a Functional Component:

```
import React from 'react';

function Greeting() {
  return <h1>Hello, welcome to React!</h1>;
}

export default Greeting;
```

2.JSX (JavaScript Extensible Markup Language)

It is a syntax extension that React uses in order for developers to be able to write their HTML-like codes within the JavaScript functions. JSX is an effortless way to program a UI while actually using JavaScript.

Example of JSX:

```
const element = <h1>Hello, world!</h1>;
```

3.State and Props:

Props (properties): In the inputs of a component. It allows the data to be transferred from a parent component to a child one. Props are read-only and cannot be changed by the child component.

State: A changing state can be internal to an individual component. The component rerenders to effect the change in UI since a change in state means change in the UI.

Example of a simple component with state:

```
function Counter() {
    const [c, setCount] = useState(0);
    return (
        <div>
            <p>{c}</p>
            <button onClick={() => setCount(c + 1)}>Increment</button>
        </div>
    );
}
```

4.Virtual DOM:

The Virtual DOM represents the DOM elements in memory, a lightweight version of it, React maintains that whole. It uses this Virtual DOM to optimize updating the actual DOM. When the state of a component changes, React first updates the Virtual DOM and reconciles it with the previous version. Finally, it updates the real DOM where necessary, improving performance.

5.Event Handling:

React is focused on a more declarative management of events. Instead of resorting to the direct attachment of event handlers in plain JavaScript, it allows for React's own event handling system to be employed.

```
function Button() {
   const handleClick = () => {
      alert('Button clicked!');
   };
   return <button onClick={handleClick}>Click Me</button>;
}
```

Advanced Concepts in React:

1.Hooks:

React Hooks are functions that give you access to state and other React features from functional components. Hooks were introduced in React 16.8 as an alternative to class components for state management, side effects, and similar tasks.

The common hooks are:

useState: Used for managing local state in functional components.

useEffect: Used to perform side effects such as data fetching, subscriptions, or DOM manipulations.

useContext: Used for consuming context inside a component in a way that allows you to pass data down the component tree without having to pass props manually.

Example of using useState and useEffect:

```
import { useState, useEffect } from 'react';
function FetchData() {
  const [data, setData] = useState(null);
  useEffect(() => {
    fetch('https://api.example.com/data')
      .then(response => response.json())
      .then(data => setData(data));
  }, []); // Empty dependency array means it runs once on mount

  if (!data) return <div>Loading...</div>;
  return <div>{JSON.stringify(data)}</div>;
}
```

2.Context API:

React, use this context API to set up what will mean sharing values—such as global data-between components without needing to pass props down the component tree. This might be good for managing state across different dimensions; themes may define user authentication and language preferences.

3.React Router:

Forwarding navigation and URLs routing within an app allows access to develop a single-page application containing different views, or so-called "pages." This includes React Router allowing users to traverse varying components based on the URL.

4.React Redux:

The Redux library refers to a library that is normally used with React in managing the state's application in a centralized store. On most occasions found utilized in larger applications where state needs to be shared by so many components, this requires lots of boilerplate code than the regular state management built into React alone.

The React library for modern web applications is very powerful. The component architecture, fast update mechanisms, and vast ecosystem make it a preferred choice among developers. Small or large-scale enterprise, React empowers developers with tools and flexibility to build dynamic, high-performance user interfaces.

EXPRESS:

Express is one such lean, lightweight, and minimalistic framework for web applications that runs on Node.js. It was designed to revolutionize application server development and APIs with its features that make all the complicated tasks of handling HTTP requests, routing, middleware, etc. simplified. Express is amongst the most popular frameworks that can be used to build server and API applications in Node.js.

Express Core Concepts

1.Middleware:

Middleware functions lie at the heart of Express applications, acting on requests before they reach the route handlers or responses after the response has been sent to the client. The functions carry out a range of tasks such as:

Handling request bodies (for example, express.json() handles JSON payloads).

Checking for user authentication.

Logging requests.

Error processing.

Setting HTTP headers.

The middleware functions defined on the Express app will be executed in the order they were defined and can operate on the request (req) or response (res) objects before transferring control to the next middleware or route handler.

Example:

```
app.use(express.json());  // Middleware for parsing JSON bodies
```

2.Routing:

Routing is Express's ability to connect an HTTP request with a corresponding function or route handler. An operation is one of the two or more that occurs when an endpoint (URL) specific for that operation is matched against a certain HTTP method, for instance: GET, POST, PUT, DELETE, etc.

Example:

```
ap.get('/', (rq, re) => {
   re.send('Hello, World!');
});
ap.post('/user', (rq, re) => {
   // Handle POST request to create a new user
});
```

3.Request and Response Objects:

The Express framework has given two entities for handling incoming requests and outgoing responses which are request(req) and response(res) objects. It contains all the information pertaining to the HTTP request in req including headers, body, query parameters, etc., and the res object has methods in it to send required response back to the client.

Some of the frequently used res objects methods are:

res.send(): Send any type of response.

res.json(): Send a response in the JSON format.

res.status(): Set HTTP status code for the incoming response.

res.redirect(): Redirect the client to another URL.

Example:

```
ap.get('/he', (rq, re) => {
   re.stat(20).json({ message: 'Hel, Exp!' });
});
```

4.Static File Serving:

The express.static middleware can be used to serve static files(like images, a CSS or JavaScript files) directly to the client. That is how it is usually done in production environments.

Example:

```
ap.use(express.static('public'));
```

Express Function

Express works in a very simple way; it uses an event-driven architecture, which runs non-blocking actions, just the way Node.js facilitates asynchronous methods. Here's a very simple request-response cycle in Express:

1.A client sends an HTTP request to the server.

2.The request is matched against a route by Express based on the HTTP method (GET, POST, etc.) that has been used with URL path.

3.Middleware functions are applied to the request before it gets passed to the final route handler.

4.The route handler then processes the request (for example, fetching data from a database) and prepares a response for the client's use.

5.The latter sends the response back to a client, either as an HTML page, data in JSON format, etc.

Example of an Express App:

```
const exp = require('exp');

const ap = exp();

const port = 3001;


// Middleware

ap.use(exp.json());


// GET

ap.get('/', (rq, re) => {

   re.send('Welcome to Exp!');

});


// Route to handle POST requests

ap.post('/submit', (rq, re) => {
  const { nam, em } = rq.body;
  re.json({ message: `Recei data for ${nam}, ${em}` });
});

// Error handling middleware
ap.use((er, rq, re, next) => {
  console.error(er);
  re.status(501).send('Some went rong!');
});

ap.listen(port, () => {
  console.log(`Serv run at http://localhost:${port}`);
});
```

Express is an outrageous and adaptive minimalistic web framework that helps to set up web applications and APIs in Node.js very easily. In fact, everything is done for you, including handling complexity caused by HTTP, routing, and middleware configuration, thus allowing you to focus on feature implementation. With a great crowd and ecosystem behind, Express is still one of the most popular-attended frameworks in the world of Node.js.

Node.js

Node.js will works without one's unease. Node js is the run time environment open-source for cross-platform operation execute JavaScript code outside web browser. The engine, in this case, is based on V8 JavaScript engine (the same engine that runs Google's Chrome browser). Model has an event- and I/O non-blocking, which makes it lightweight and very efficient. Its primary use is in application development done server-side for I/O-intensive operations such as web servers, APIs, real-time applications, etc. It is highly regarded as among the most effective tools in the backend for scalable and high-performance application development.

Concepts of Node.js:

1.Non-blocking I/O:

Node. j s fundamentally would cost some part in making a model non- blocking and event-driven processing, i.e. any work done from this would wait for its completion before proceeding to the subsequent activities (reading files, querying a database, etc.). It can easily manage the asynchronous I/O operations all because of Event loop which doesn't block I/O when code attempts to finish that operation. Thus the nodejs becomes possible to perform many activities at the same time which turns out to be really helpful for applications known to be real time, speed-center high responsiveness.

Example:

```
con f = req('f');

// Non-blocking code (asynchronous file read)

f.readFile('fi.txt', 'ut5', (e, d) => {

  if (e) throw e;

  console.log(d); // This will print the file content once it's read
});
console.log('This mes appears first, because readFile is non-blocking');
```

2.Event Loop:

Node.js works on a single-threaded event loop model. The event loop keeps looking at the queue of tasks and executes them one at a time. Even though JavaScript in Node.js runs on a single thread, the event loop lets it seem it is servicing multiple requests concurrently. I/O operations are delegated to the system, which runs them asynchronously.

Single-Threaded: One thread executes the JavaScript code, but many connections are handled concurrently by offloading the I/O operations.

Asynchronous: Reading files, accessing databases, or making network calls will not block the execution.

Event Loop Phases:

Timers: It executes callbacks for timers (setTimeout() and setInterval() are included).

I/O Callbacks: It executes callbacks for I/O-like activities (like: read files, access databases) operations.

Poll: The event loop polls for new events, executes them, and processes I/O events.

Check: setImmediate() callbacks are executed.

Close Callbacks: Closes callbacks (like: socket.on('close')).

3.V8 JavaScript Engine:

Node.js is built on the V8 JavaScript engine, designed by Google to execute JavaScript inside Google Chrome. The V8 compiler translates JavaScript into native machine code, giving some execution performance gains by executing JavaScript at a more primitive level (close to the hardware).

This is one of the reasons for the fast and efficient Node.js server-side JavaScript code execution.

4.Node Package Manager (npm):

npm is Node.js's default package manager and enables developers to manage their application external libraries and dependencies. It hosts the world's largest ecosystem for open-source libraries: npm packages, which can be installed and used in Node.js applications with ease.

NPM can be used to install packages, run scripts, and manage definitions of scripts.

For example, here is a command to install express, HTTP server library:

```
npm install express
```

These are included in your Node.js code like this:

```
con exp = require('exp');

con ap = exp();
```

A powerful environment for server-side JavaScript is called Node.js. It cooperates with all events and non-blocking I/O operations that use the V8 engine so that you can create fast scalable applications with much high concurrency. There is a wide npm ecosystem, which easily allows for enhancement of Node.js with libraries and frameworks thus presents itself for APIs, web servers, real-time and many other microservices.

The MERN stack is great as it offers a completely comprehensive and pragmatic solution to the problem of full-stack web development in pure JavaScript for client and server. MongoDB, Express.js, React.js, and Node.js together create a perfect environment to build high-performance, scalable applications because as a developer, you find yourself using the same JavaScript from the start to the end of your development experience. That is why MERN is the reigning technology today for state-of-the-art web development, especially single-page applications, real-time applications, and APIs.

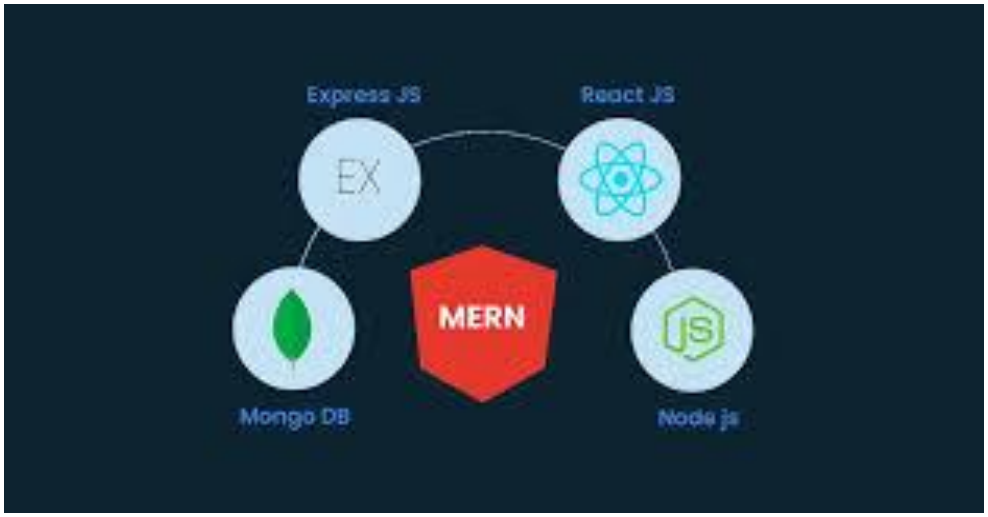


Fig 3.2.2 Mern Stack

MACHINE LEARNING

Machine learning is a set of computer algorithms that draw inferences from events and improve themselves without explicit programming. This technic falls under the aspect of artificial intelligence, whereby the statistical methods and data are used to predict an outcome that could result in some actionable insights.

The basis of this innovation is that the computer can generate correct outcomes simply by learning from the data (i.e., examples). Ironically, data mining and Bayesian predictive modeling are interlinked with machine learning. Data is entered into the computer using input; the computer interacts with them using an algorithm in return.

Recommendation is a common instance of machine learning. All Netflix suggestions for users having accounts are based on the previous viewing history of the user. IT companies are harnessing unsupervised learning in improving user experience through personalization.

Another utilization of machine learning is to increase the automation of tasks such as fraud detection, predictive maintenance, portfolio optimization, and so on.

Traditional programming vs. machine learning

Essentially, programming is the process of coding every one of the rules, with input from someone well versed in the particular field for which the software is being developed. Each of those rule’s logic that the computer understands; it does the outcome put forth by that logical assertion. More of those rules will need to be created as the system becomes more and more complicated, and before long, it may become practically impossible to maintain.

Machine learning is quite different from conventional programming. in standard programming. More rules must be created as the system becomes more complicated. Maintaining it might soon become impossible.

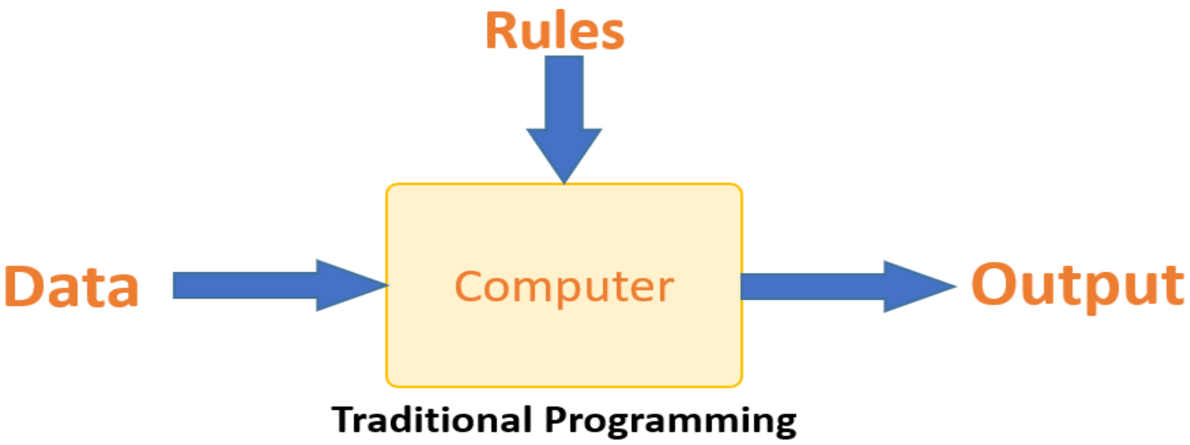


Fig:3.2.3 Traditional Programming

This problem is supposed to be tackled by machine learning. The machine gets trained to know how the input data is related to the output and then forms a rule to predict the output. Every time fresh data comes, programmers will no longer need to write new rules. The algorithm adapts of given new information and experience, improving and becoming more effective with time.

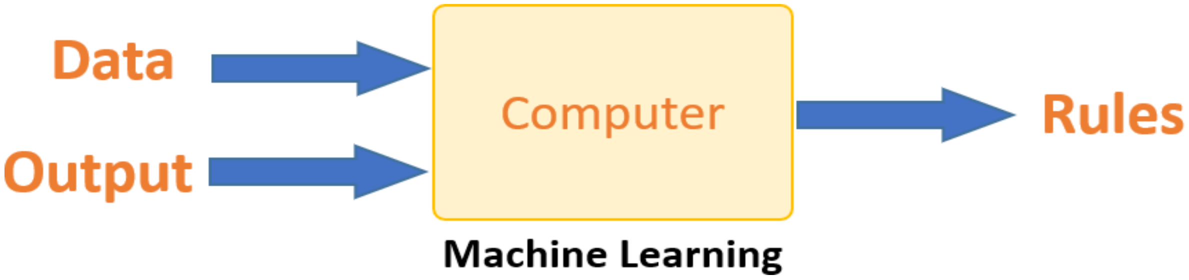


Fig:3.2.4 Machine Learning

Implementation of Machine Learning

Machine learning basis is the place where all learning occurs. A machine's learning is akin to a human. Human beings learn by experience. The more we know, the more we can predict correctly. When confronted with one unknown, our chances of success are lower than they would be in a known situation. But unlike a human, a computer would also have some difficulty in predicting the outcome when given a new sample.

Learning and inference are two important aspects of machine learning. The first way a computer learns is by pattern recognition. The other way to look at it is that data allowed to make that finding. It is one of the most important skills of a data scientist to choose the data very carefully to feed the computer.

These clustering algorithms then simplify this world; in other words, this finding becomes a model. Hence, model learning describes the data and summarizes it into a model.

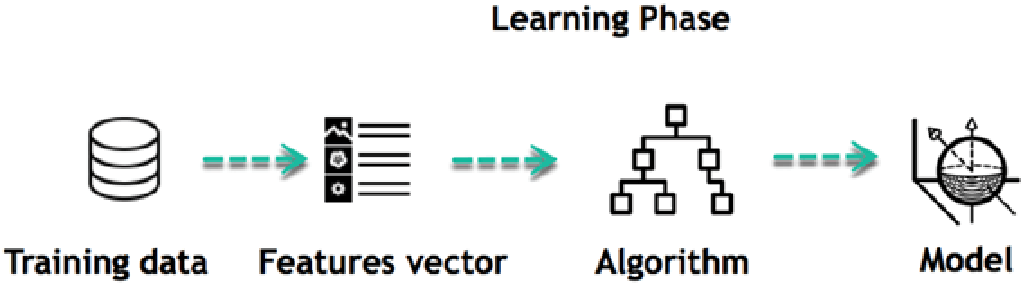


Fig:3.2.5 Learning Phase

The computer attempts to ascertain the correlation between a person's income and the likelihood that they will dine in an upscale establishment. Turns out, to the computer, income and fine dining are correlated positively: take an example. Inferring Post model creation, it is possible to validate the performance on unseen data. The new data is converted into features vector, it undergoes processing by the model, and then output as prediction. That is what makes machine learning unbelievably great: there is no need to retrain the model or change the rules. The already trained model is able to make inference with fresh data.

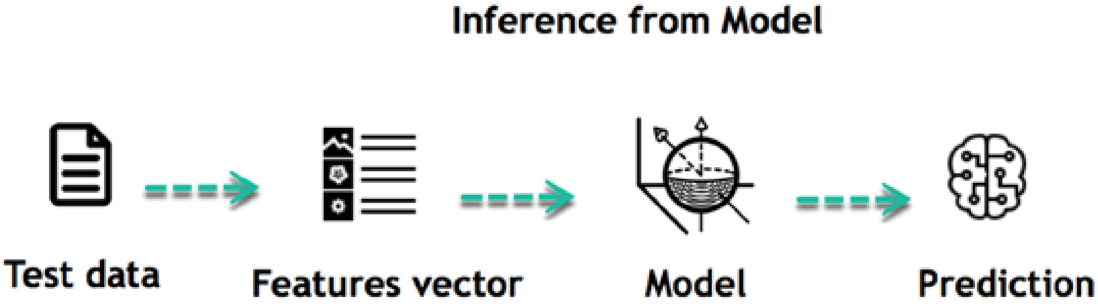


Fig:3.2.6 Interface from Model

The following bullet points sum up the basic nature of machine learning programs:

1. Formulate question
2. Collect data
3. Visualize data
4. Build algorithm
5. Validate algorithm
6. Collect feedback
7. Improve algorithm
8. Repeat process 4-7 till satisfaction
9. Prediction on the model

As soon as the algorithm learns how to make the correct conclusions, it applies this knowledge to fresh sets of data.

Supervised education

An algorithm works by establishing a connection between inputs provided to it and an output through training data and feedback from individuals. For example, a practitioner might be able to forecast sales from the accompanying input marketing expenses and weather predictions.

Let's say, in terms of output data known, this is an example of supervised learning type. The program will predict the new data.

There are two types of supervised learning:

- A classification task
- Regression task

Regression

If the output is a continuous value, the work becomes called regression. "Predict the stock price at any time necessary according built on many specified factors such as equity and performance in stocks earlier, performance of the macroeconomics index," the financial analyst may have to say. Essentially, you train the system to get stock prices as less error-prone as possible.

Unsupervised learning

In unsupervised learning, the algorithm analyzes the input data without being told what to expect as an output variable itself (e.g., cluster customer demographic information to find patterns).

It can be used when there is no prior indication of how to categorize the data and an algorithm can find trends and categorize it itself.

MACHINE LEARNING ALGORITHMS

Algorithm Name	Description	Type
Linear Regression	The output of continuous target variable is predicted through one or more continuous input features which use the regression algorithm. The model assumes a linear relationship.	Supervised (Regression)
Logistic Regression	It's an example of a classification algorithm which predicts a binary outcome (0 or 1). Another way of putting this would be that it models the probability that an instance belongs to a class.	Supervised (Classification)
Decision Trees	It employs the creation of a tree-like structure by splitting the data into subsets from one another on the basis of values of different features. It is a supervised learning algorithm used for either classification or regression tasks.	Supervised (Classification/Regression)
Random Forest	The ensemble method entails the creation of multiple decision trees and the combination of their predictions. It is applicable in either classification or regression tasks.	Supervised (Classification/Regression)
K-Nearest Neighbors (KNN)	An example of a non-parametric classification algorithms where the class of a data point is identified using the majority class value from the class of its nearest neighbors.	Supervised (Classification)
Support Vector Machine (SVM)	An algorithm whose purpose is to find the hyperplane which divides the data into two classes with the maximize distance between that hyperplane and the nearest data sample.	Supervised (Classification)
Naive Bayes	The Bayes classifier is statistical probabilistic model that uses Bayes theorem and assumes feature independence. It is most commonly used for text classification (like spam detection).	Supervised (Classification)
K-Means Clustering	In this article, algorithms that learn unsupervised cluster of k-group data by optimizing the total variance within each cluster are introduced.	Unsupervised (Clustering)

Fig 3.2.1 Machine Learning Algorithms

How to Choose Machine Learning Algorithm

3.2.3. Machine Learning (ML) algorithm:

There are many algorithms in machine learning. The selection of the algorithm depends mainly on the specific goal. One of the machine learning examples has the challenge of predicting one of the three variations of a given flower. With the length and breadth of the petals, the predictions will be made. The results from 10 different algorithms are shown in the image. The dataset image is displayed on the top left by itself. The following images show different algorithms and their attempts in categorizing the data.

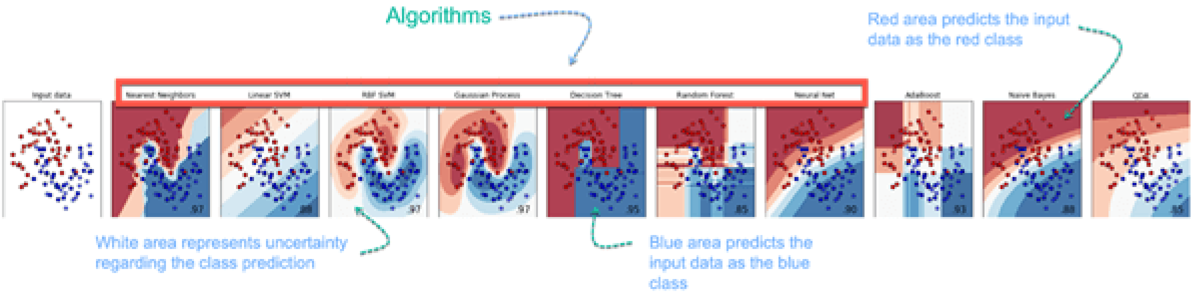


Fig 3.2.3.1 Machine Learning

Challenges and Limitations of Machine Learning

The main challenge in machine learning is an absence or sparse variety in the dataset. When there is no data at all, machines find it impossible to learn. Even when a dataset is not sufficiently variable, it will still challenge a machine. For insightful knowledge from the computer between a few domains' heterogeneity is required. If there are not many or no variations at all, it is rare for any algorithm to extract information. For a machine to learn, it is advisable for everybody to have at least 20 observations.

The use of machine learning

Augmentation:

Machine learning is a method by which a computer uses algorithms to support a person with a limited amount of control over the output. Machine learning has various applications, such as in software programs, data analysis, and virtual assistance. Another application is for the reduction of errors that are likely to occur due to bias.

Automation:

To better illustrate: imagine robots performing important operation stages in production plants.

That's completely independent machine learning from anything in any industry; it doesn't need human input.

Finance Sector

The financial industry is at the forefront of machine learning. Majorly, for the identification of trends in data, banks now deploy ML so as to prevent fraud.

Governmental institution

The management uses machine learning for monitoring and viewing utilities and public safety. Take for instance China, which has an advanced facial recognition system. To keep people from jaywalking, the government employs artificial intelligence.

It was in very early stages that the healthcare industry embraced machine learning for the purpose of image identification.

Marketing

As a result of the easy access to a lot of data, AI is now used extensively in marketing. Before mass data, researchers looked into applied advanced mathematics like Bayesian analysis for some estimate of what a client might be worth. In the event of the explosion of data, AI enhances methods of relationship building with consumers and campaigns for marketing within the marketing department.

Supply Chain Machine Learning Application Example

Machine learning gives fantastic outputs for visual pattern detection making it possible use of this technology throughout supply chains for physical inspection and maintenance purposes.

In an unsupervised manner, it is possible to make quick exploratory searches of a heterogeneous dataset looking for similar patterns. It may also enable system monitoring for wear and damage to shipments as they move through the logistics hub.

For example, Watson-a platform by IBM-can recognize that damage has been done to shipping containers, while having systematic-data and visual-data integration-monitor, report, and recommend actions in real-time.

This approach was widely used by stock managers last year to assess and predict the inventory. When big data is mixed with machine learning, sales will increase by 2 to 3 percent on the back of anticipated reductions in inventory costs.

A Google Car Example of Machine Learning

Take, for example, the Google car. This automobile is covered with too many laser roofs, constantly providing location information about the car itself. Radar is mounted on the front of the car, allowing it to detect the relative speeds and positions of all other vehicles in its vicinity. Almost one gigabit of data is processed by the automobile.

Analyzing anomalies

A very general definition of anomaly detection is outlier discovery, which usually means the methods for locating outlier events or observations that deviate from the norm and raise a question. Anomalous things usually mean trouble: bank fraud, structural flaws, health issues, and text faults. The terms outliers, novelties, noise, deviations, and exceptions are all synonymous with "anomalies." A perturbation, say detection of intrusion in network or misuse detection usually happens to be more interesting than some of the rare items. One could even map this scenario to micro-clusters formed by the patterns through cluster analysis.

Anomaly detection methods can be divided into three broad categories. The first category consists of unsupervised anomaly detection methods, which appointee some locality that intrinsically least resembles the entire data set to define an anomaly in an unlabeled test data set with an assumption that most of the instances considered are normal. Semi-supervised approaches use a set of normal training data to define normal behavior and then reject test cases depending on how likely they are to be produced by a model of that normal behavior.

Robot education

In developmental robotics, robots program their own learning curricula or sequences of learning events in order to gradually develop new abilities through autonomous exploration and social contact with humans. The learning guiding techniques for these robots are imitation, motor synergy, active learning, and maturation.

Rule of association

Associative rule learning is a rule-based machine learning method that attempts to find correlations between the variables of a large dataset. The idea is to use some measure of "interestingness" to find strong rules that have been found in databases. All rule-based machine learning encompasses any machine learning whereby a set of rules is either detected, learned, or generated to store, transform, or apply knowledge. Thus, a rule-based machine-learning algorithm can be defined as one that finds a number of relational rules to describe all the knowledge that has been learned by the system.

Association rules concerning the idea of strong rules were established by Rakesh Agrawal, Tomasz Imieliski, and Arun Swami to compensate for the missing patterns among items. For example, given that a rule "potatoes, onions" right arrow "hamburger meat" has been learned from a supermarket's sales data, basically, if customers purchase potatoes and onions at the same time, they are also likely to purchase hamburger meat.

Learning classifier systems (LCS) are a subfamily of rule-based machine learning algorithms, which combine a learning component performing supervised learning, reinforcement learning, or unsupervised learning with a discovery component that is often a genetic algorithm.

Inductive logic programming (ILP) is a technology for rule learning that uses logic programming as its standard form for input instances, prior knowledge, and hypotheses. An ILP system will output logic programs consisting of all positive and none of the negative examples if given a representation of some known background information along with a number of examples represented as a logical database of facts.

The fields of natural language processing and biomedicine are benefitting tremendously from inductive logic programming. The fundamental theoretical framework for inductive machine learning in the context of logic was set up by Gordon Plotkin and Ehud Shapiro, where it is inductive reasoning by the property shown to each member of the well-ordered set as getting different with mathematical induction.

Models

Machine learning is the process of developing a model such that when it is trained on data, it is able to analyze much more data and make predictions. There are many different models that have been researched and applied in machine learning systems.

Networks of artificial neurons

An artificial neural network is like an extensive network of neurons in the brain made of groups of nodes linked together. Each node symbol represents an artificial neuron in this scheme, while arrows indicate how the output from one artificial neuron feeds into another's input. Running microcircuits are usually modularly structured and constitute a connectionist artificial neural network. Such systems "learn" to execute tasks by taking into account examples, often without having any task-specific rules written into them. An artificial neural network is what a model calls, which has been built from a series of interconnected "artificial neurons" designed to mimic neurons in a human brain. Just like the synapses in a human brain, a "signal" is passed from one artificial neuron to another through each of these connections. In conventional ANN implementations, the output of each artificial neuron is calculated by some non-linear function of the sum of its inputs, and the signal at a link between artificial neurons is a real number. "Edges" are the connections between synthetic neurons. Artificial neurons are grouped into layers most times. Initially, the aim of ANN was to solve problems in a manner similar to that of the human brain.

However, over the years, the focus has shifted towards performing certain very specific affairs in ways that biologically deviated. For example, applications of artificial neural networks include computer vision, speech recognition, machine translation, social network filtering, playing video and board games, and medical diagnosis. This is called deep learning by an artificial neural network when it has multiple hidden layers. This simulates the brain's transformation of light into vision and sound into hearing. Successful applications in these areas of deep learning include computer vision and voice recognition.

A decision tree

Decision tree learning uses a predictive model in the form of a decision tree to go from observations concerning an item (presented in branches) to inferences about the item's target value (expressed in leaves). This is a predictive modeling technique used in data mining, statistics, and machine learning. Classification trees are those tree models where the target variable is discrete in nature; their leaves represent class labels and the branches correspond to the attributes that together define those class labels. In data mining, a decision tree summarizes the data, while a resultant classification tree is used as input for decision-making.

Machines that support vectors

Support vector networks, better known as support vector machines, form a family of similar supervised learning techniques that fall under the category of regression and classification. An SVM training algorithm is a relatively straightforward model for determining whether a new example falls into one of two categories conceived from the training examples of which each has been labeled by one of the two categories. They can also do even nonlinear classification effectively as well as linear classification by implicitly mapping their inputs into very high-dimensional feature spaces, which has been called the kernel trick.

SVM or support vector networks, have evolved to form a family of similar supervised learning techniques for regression and classification. An SVM training algorithm is a model that predicts whether a new example falls into one of two specified categories given a collection of training examples, each labeled as belonging to one of the two categories. They have the ability to even classify nonlinearly along with linear classification by implicitly mapping their inputs in a high-dimensional feature space, this technique has been popularly known as the kernel trick.

Analysis of regression

Regression analysis comprises a vast set of statistical techniques meant to measure the strength of association between input variables and the characteristic to which they are related. It is possible to specify a line that becomes the best fit with respect to the given data according to some mathematical criterion such ordinary least squares in its most popular version, linear regression. The latter is usually enlarged by regularization techniques to overcome overfitting and reduce bias, as ridge regression does. Commonly employed models in non-linear problem type are logistic regression (often found in scalar classification), polynomial regression, and kernel regression, which adds non-linearity by taking advantage of the kernel trick and implicitly mapping input variables to higher dimensional space.

Boasian networks

A simple Bayesian network would involve rain affecting whether the sprinkler is on, and both would affect the wetness of the grass. A DAG (Directed Acyclic Graph) comprises a collection of random variables or their conditional independence in a Bayesian network, a belief network, or a directed acyclic graphical model. This can be as simple as how certain illnesses are connected to one another by symptoms, where the network can be used to calculate how likely it would be that some specific set of illnesses would be present given a set of symptoms. Algorithms are used effectively for inference and learning purposes. Dynamic Bayesian networks may show real-time sequences, similar to voice signals or protein sequences. Influence diagrams are generalizations of Bayesian networks that may condition so as to express and resolve to some extent decision problems under uncertainty.

Gene-based algorithms

A genetic algorithm (GA) is a search technique and heuristic that emulates natural selection in that it perturbs new genotypes by mutation and crossover in search of better solutions for a given problem. Application of genetic algorithms to the area of machine learning started in 1980s and 1990s. On the contrary, machine learning concepts have improved the working of genetic algorithms and evolutionary algorithms.

Modeling exercises

It is often seen that machine models are starving too much for data before perfection. Most commonly, a huge, representative chunk of the training set data is required to train any machine learning model. This can go from corpus of texts, assortments of images to data collected from certain users of a service. Sometimes, while training a machine learning model, one should avoid the case of overfitting.

3.2.4 SYSTEM REQUIREMENTS

Software Requirements

➢ Operating system (OS)	:	Windows 10 or above.
➢ Coding Language	:	Python, MongoDB, Express.js, React.js, Node.js
➢ Framework	:	Flask

Hardware Requirements

➢ Processor	:	i5 or above.
➢ Hard Disk	:	500 GB.
➢ Monitor	:	15’’ LED
➢ Input Devices	:	Keyboard, Mouse
➢ Ram	:	4 GB or above

3.2.5 Module Description

Data Collection:

For an agricultural study on rice cultivation, a variety of environmental and soil factors will have to be monitored and recorded. Soil samples are collected from different sites in the field, getting analyzed in the laboratory for the nutrient ranges of N, P, K, and acidity levels. Environmental conditions taken as temperature, humidity, and rainfall are recorded by automatic weather stations, thermometers, hygrometers, and rain gauges, and collected daily or weekly. The crop (rice), and their growth phases are also recorded including yield data at harvest. Data is collected periodically during the entire growing season starting with soil testing prior to planting and ending when the growing cycle has reached its conclusion. All of this data is typically organized in a systematic record often consisting of a computer database or spreadsheet in which each entry comprises the date and place of retrieval along with all associated environmental and soil parameters. Data validation also ensures that the data collected is structurally consistent. This data could then be used to analyze the relationship between soil nutrients, environmental conditions, and rice growth, which will help optimize farming practices while being able to predict future yield rates.

Dataset:

Multivariate dataset

There are 2,200 number of records

The dataset includes information is given below:

N (Nitrogen): In our dataset, the relevance of nitrogen in the soil, which is an important nutrient for the growth of plants, ranges from 60 to 94 in value. It is essential for photosynthesis and protein synthesis in plants.

P (Phosphorus): Phosphorus has been shown to be efficient in transferring energy; it plays a role in photosynthesis and in transport of nutrients in plants. The measure found in soil as phosphorus is numerically identified here between 35 and 58.

K (Potassium): Potassium is one of the essential macronutrients and is very essential for the metabolism of plants, activating enzymes and resistance to stresses. The ranges of potassium are 36 to 44.

Average temperature: The average temperature in the environment described in degrees Celsius. How much it varies from place on earth also influences metabolism and plant growth. The ambit goes from 20.1°C to 26.8°C. It suggests tropical environment or close to subtropical temperatures.

Moisture: Moisture in the air is what humidity really means. It can have a bearing on transpiration and availability of water for plants. The percentage values vary between 80 and 83, which show the level of humidity with which the moisture is contained within the given environment.

pH: Soil pH. Nutrient availability is influenced by soil pH, as is microorganism activity. In the present case, pH values between 5.1 and 7.8 define a slightly acidic to neutral soil environment.

Rainfall: This is that column ranging rainfall in millimeters. Rain provides water sources for plants. Hence all that is given ranges from 185.3 mm to 284.4 mm, reflecting moderate to maximum rainfall conditions found.

Label: The label rice is attached to this dataset, indicating that it is all about the cultivation of rice.

The data seem to be part of a larger analysis of possible machine learning or statistical modeling to identify some optimal conditions for rice growth or even predict rice yield in different environmental conditions.

Data Loading and Inspection:

The dataset is loaded using a CSV file (df=pd.read_csv('dataset.csv')). This data set is then going to be inspected with df to understand its structure including the label column which designates the target variable, the type of crop.

Feature Selection:

The 'N', 'P', 'K', 'temperature', 'humidity', 'ph', and 'rainfall' are the features used for prediction. These columns are representative of soil and environmental factors that are believed to affect crop growth.

The label variable is that df['label'] which contains types of crops, e.g. rice.

Data Splitting:

In training and testing datasets, the entire dataset is divided into train and test datasets and uses train_test_split from sklearn.model_selection: The model trains through such data in training dataset and checks its possible performance with test dataset. Now the test_size=0.2 parameter defines that 20% of the data gets used for testing, while 80% gets used for training.

Naive Bayes Model:

Naive Bayes (GaussianNB) the first machine learning model for this project. It is a classifier and probabilistic classifier. Training is done from training data (model.fit(Xtrain, Ytrain)) prediction through the test set (model.predict(Xtest)).

The accuracy of the model is calculated with metrics.accuracy_score(Ytest, predicted_values) and printed as "Naive Bayes's Accuracy".

The storage of the Naive Bayes Model:

Storing the trained Naive Bayes model using Python's pickle library to serialize and save it as a file (model.pkl). In that way, the model can be reloaded and used without retraining.

Random Forest Classifier:

Both model and target were trained using Random Forest classifier (RandomForestClassifier) with the same features and target.

The evaluation and printing of the trained (RF.fit(Xtrain, Ytrain)) and tested accuracy of the model took place.

Decision Tree Classifier:

A Decision Tree classifier (DecisionTreeClassifier) is also termed to train the model. The nodes were split based on the criterion of entropy, which estimates the dataset impurity. The parameter max_depth=5 means that the tree will never be deeper than five levels.

After training the decision tree model (DecisionTree.fit(Xtrain, Ytrain)), its accuracy is calculated and printed.

Label Encoding:

The labels target of crop type are, therefore, categorical and are converted into numbers through Label Encoding (LabelEncoder). This is necessary for models such as XGBoost that require that input numeric values become input for target labels.

Using le.transform(df['label']), the labels are encoded and again split between train and test datasets.

XGBoost Model:

The XGBoost classifier (XGBClassifier) is trained on the features and encoded target labels. XGBoost itself is a fast and highly efficient algorithm within the gradient-boosting framework. Model accuracy is computed and printed post-training.

Summary of Model Training:

The code contains implementations of four different algorithms: Naive Bayes, Random Forest, Decision Trees, and XGBoost.

Naive Bayes, Random Forest, Decision Trees, and XGBoost have been implemented in the code with four different algorithms.

They are each trained using the same feature set, namely ('N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall'), to predict a crop label ('rice', and so forth).

The models are evaluated on the basis of how truly they could predict cuts based on environmental factors and soil conditions. This is the Naive Bayes model, which is stored for future use, while accuracy scores for all the other models are printed for their performance comparison.

Output:

Herein, printed for each model, the accuracy on the test set will show how well each of the models performs in predicting the crop type.

We achieved a test set accuracy of 98.1%.

Saving the Trained Model:

Whenever you are ready for a production-like situation, the first task to do is to persist your trained and tested model into a .h5 or .pkl file, with libraries such as pickle. Ensure that Pickle is installed in your environment. Next, import the model into the module and dump it as a .pkl file.

3.2.6 UML Diagrams Description

However, the use of UML, a diagrammatic language, will serve to just lay down a pattern for possible operations on any intricate system. A photo or drawing will depict what an intricate framework would resemble. Such diagrams have an early bearing on the manner in which we understand such intricate matters.

- Class Diagram
- Use Case Diagram
- Activity Diagram
- Collaboration Diagram
- Object Diagram
- Data Flow Diagram

Class diagram

An orientation about the interconnections or source articles regarding the instruction in the Unified Modeling Language (UML). In a special case, the class denotes methods or attributes of an object which is a particular instance of a program but one in code-has a say about that kind of substance. The class diagram is a valuable tool in nearly all realms of object-oriented programming. It has been historically very ancient but has a refinement-OOP showing perfect models have developed.

Thus, it became the famous design to develop the software applications. The class diagram can be represented by attributes and methods. The purpose of the class chart is to communicate static views for the entire system.

Classes are arranged from the groups so much to share the main attributes. Hence a flowchart takes on between which classes are depicted as boxes, each box dividing into three compartments. The first rectangle is name of the class while the second rectangle holds that class file's properties. The third square holds about the methods, also known as operations, of the class. These boxes are being connected by lines having arrows at the one or both ends. The lines signify associations, namely called relationship, between the classes.

Use Case diagram

These are considered due to the experimental framework requirement of paranormal state. Cases are caught alive, with functionalities supplying modified cast conditions on a mold owing to the workings. Therefore, these expect utterances use instances because those belong solely to the regular skeleton functionalities, written in an abroad sorted way. Second, the most important regarding the past use cases is the characters. Entertainers may remain characterized as like something up to expectation interfaces with the framing. It should be spoken by the entertainers, use cases and their connections. The main use of this visual case diagram is to show all the systems in a graphic form. The use case diagram would be- capturing the dynamic parts of the system.

The characters on the screen can be a human client, some interior applications or some external applications. So in a concise when they are intending to draw an utilization case outline to have the accompanying things recognized.

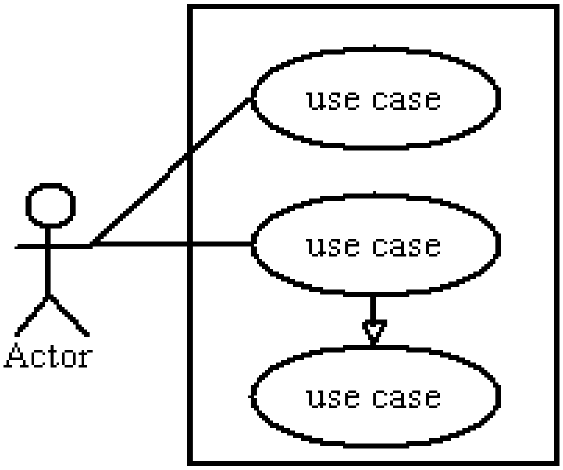
- Actors
- Functionalities in imitation of stay spoken in imitation of so a utilization suit
- Relationships among the usage cases and actors

Use case graphs are drawn up according to trap the functional necessities concerning a system. So, the differentiate on identifying the aforementioned topic wants to follow up on the secondary directives before making a unique use case scenario graph. The recognition about the use case is very important, so the recognition would be the task for the test case to map into this way that much differentiate to the functionalities done up. Provide a realistic name for actors on-screen. Explicitly show relationships or conditions in reference to the map.

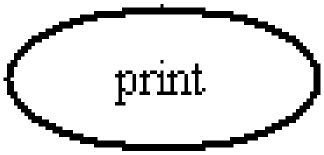
Do not operate such long subjects. Because the main writing behind the diagram is regarding understanding the requirements.

The primary purpose is to show the on-screen actor for whom the mold function is done based on the use case diagram.

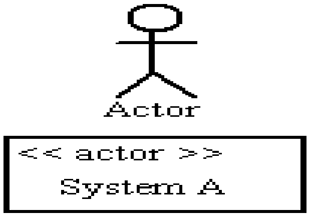
- **Symbols and Notations of Use Case Diagram System**



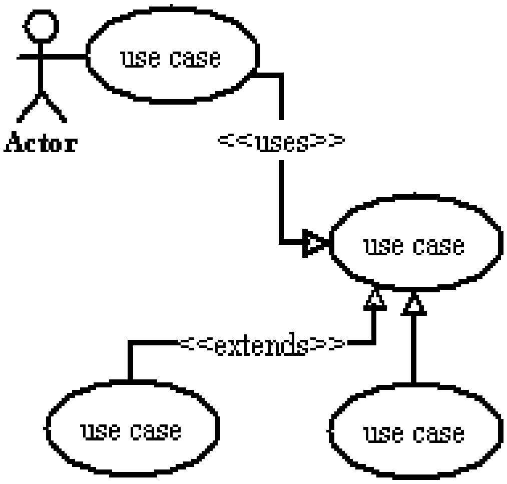
Use Case



Actor



Relationships



Sequence Diagram

Now that we are going to talk about sequence diagrams, we need to mention the concept of UML. UML is actually a toolbox for illustrating and constructing various diagrams-behavioral and structural to interaction design.

A sequence diagram is viewed as an external type diagram because it illustrates the interaction between a number of objects in a particular order. These types of diagrams help to capture the requirements for a new system or document a particular software designer/engineering process. Sequence charts are popularly known as event charts or event scenarios.

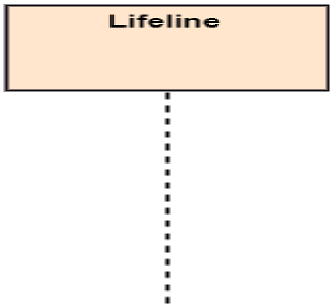
We find two types of sequence diagrams: UML diagrams and code-based diagrams. For reporting purposes, code is the last in the list and not included herein. Everything needed to model structure and behavior for both types can be found in the UML diagram software by Lucid chart. For companies and other organizations, the sequence diagram provides some useful references. Try drawing a diagram for:

- Display the UML Use case details.
- Modelling of complex processes, functions, or operational logic.
- To view inter-relationships between objects and components to complete a process.
- Detailing some functionality of an existing or planned scenario and grasping it.

Sequence graph notations

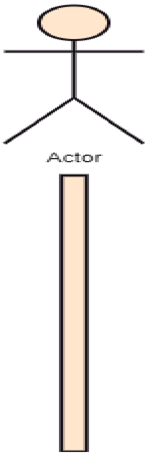
Lifeline

When taken with the amidships, the life line appeared to be a sole participant of a sequence diagram on the chart right at the top.



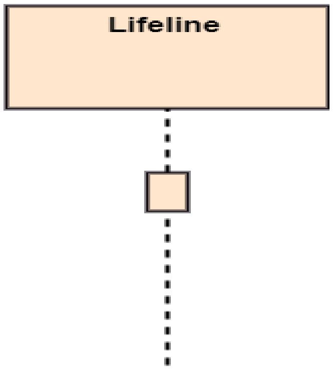
Actor

Actors are entities interacting with a subject. It lies outside the system scope. This is the role played by human users and externals. Actors may be used to represent physical entities or not, but all that matters is what is being acted out by the role of the actor. An actor may actually belong to different roles or the other way around.



Activation

A slender rectangular figure lies on the lifeline, which refers to the period of activity of an element during which the upper edge and lower edge are connected to the time of start and time of end respectively.

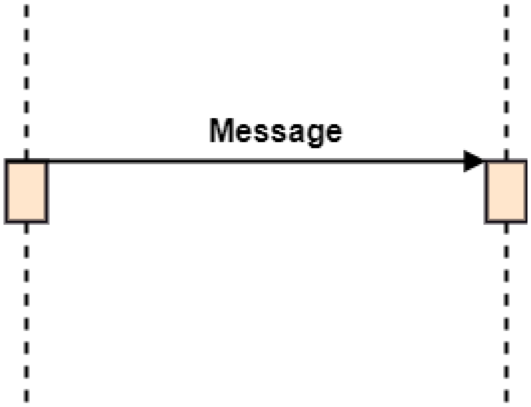


Messages

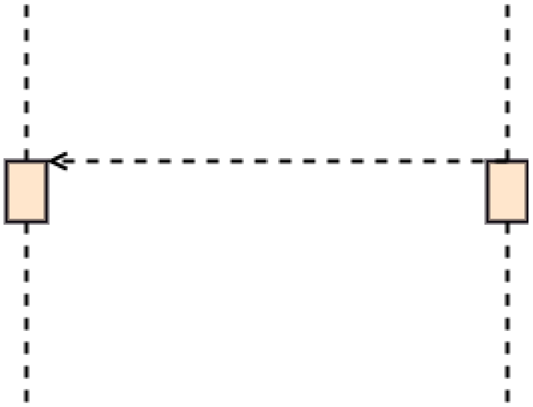
The messages express the interaction between the objects and are represented in form of arrows. They appear sequentially over the lifeline. Messages and lifelines are the main components of a sequence diagram.

The kinds of messages are as follows:

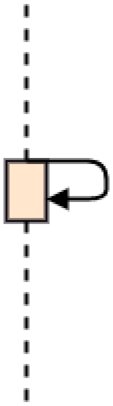
Caller ID: This indicates a particular method of interaction between the lifelines, which represent an operation invoked by the target lifeline.



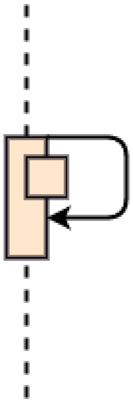
- **Return Message:** It refer to a certain kind of interaction across the interactive lifelines that represent the information flow from the receiver.



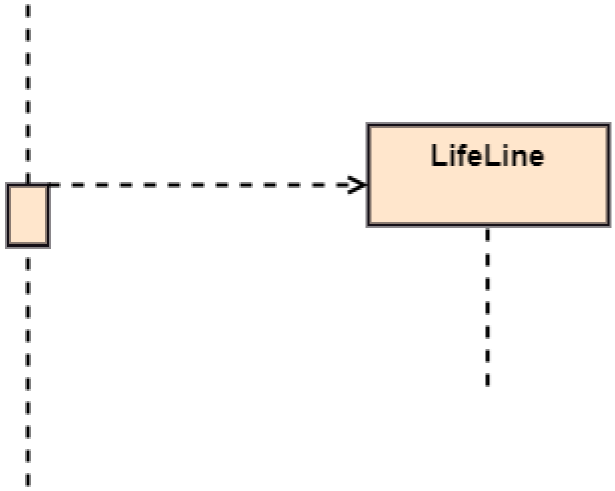
- **Self-Message:** This is an explanation of a communication triggered within lines of interaction and would represent another message within this same line of interaction.



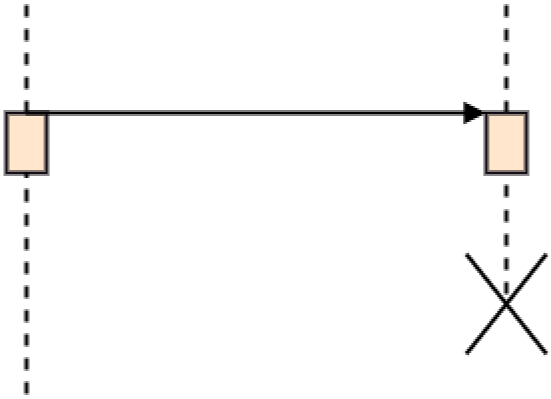
- **Recursive Message:** A separate message important for recursion is sometimes referred to as a self-message. In considerable detail, one may argue that the recursive message is, in fact, a special case.



- **Create Message:** It symbolizes a dialogue especially between the two life lines that delineate the threshold of the target.



- **Destroy Message:** This depicts a conversation about life talks and affordability with a resolution of killing the target.



Collaboration diagram

This is called a similarity outline or connection chart. Copies of the ties and collaborations realized in the applications in Additional or Unified Modeling Language (UML). The concept becomes older than ten years because much has been polished like this modelling of the ideal model in the developing years.

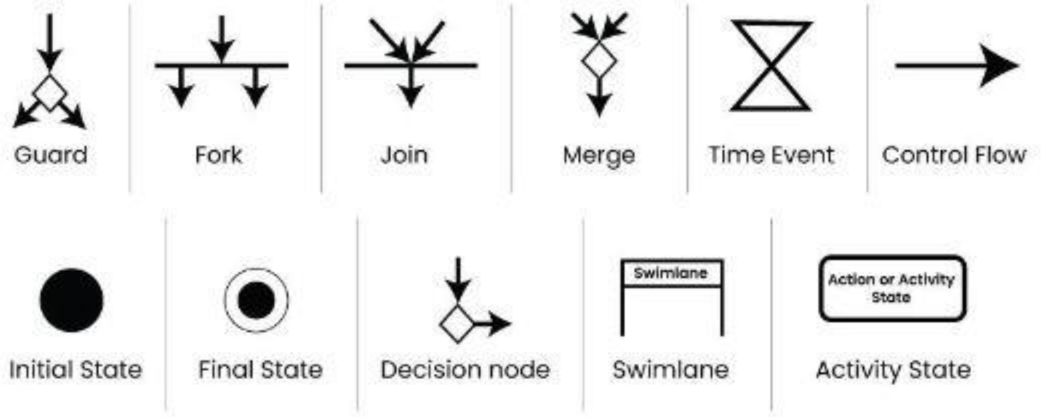
It has a flowchart which represents all the jobs, functions and procedures involving a standalone entity as the overall simulation of skeletons progressively. Things are generally in squares with names in them. These names have taken long so that they are hidden by colons. The relationships between the objects appear as lines that are interacting with the square forms.

Coordinated efforts charts are only too limited in concerning many other things including those which do not actually include common views on the main relationships of very short sections. As the numbers of objects and messages rise, a coalition concern might become very hard to read. Some vendors sell the programming just for building and also changing junction effort charts.

Activity Diagram

It is used as a flowchart and certainly holds some greater capabilities. An activity can be seen as a presentation given by the system. An activity diagram is utilized to display the relationship of the system with the primary activity and the subsequent activities.

An activity is something that has a characteristic of the system. Activity diagrams are now not only used for visually representing the architecture but are also actively involved in building the executable model via the previous or defining methods. The main drawback of the activity diagram is the information section. It is like half a generation space that is regarded as the flowchart. In spite of the fact that it has some resemblance to an activity diagram, it is not one. It shows unique flow types-parallel, spread, at one time but one.



Object diagram

An object diagram is a type of diagram featured in Unified Modeling Language (UML) that shows the structure of an object system at a moment in time. It represents the instances of classes--called objects in UML terminology--and describes their current relationships, which may be quite different from those defined in the relatively permanent class diagram describing the system blueprint.

Object diagram is primarily used to show instances of objects and how they link together through association, aggregation, and composition. These helps visualize the state of the system by capturing the values of the attributes of the objects at a specific time. Objects are represented in the diagram as rectangles similar to class-diagram representations but have instance names underlined to differentiate them from class names. Solid lines form links between objects associated with each other; role-name or multiplicities may or may not be specified on these links in order to indicate the relationship.

An object diagram is a tool that gives a better understanding of the dynamic behavior of the system, since it captures the relationship that exists between various objects in a scenario or use case. It allows representing a very complex relationship and is especially useful for debugging or browsing the design of the system developed. Although object diagrams and class diagrams are very close in kind, object diagrams particularize and ground the abstract structures of classes to concrete instances and their states at a present moment.

Dataflow diagram

A data flow diagram, or DFD, is a kind of graphical interface that illustrates how data flows in a system-that is, through what input is processed to produce an output. In essence, the information flow and transformation image provide a better understanding of the function of the system. A DFD possesses the following basic elements: processes-that transform data by activities or functions; data flows-that depict the movement of data among processes, data stores, and external entities; data stores-that house the data within the system; and external entities-that send or receive data relating to the system. Within degrees of abstraction from Context Diagram Level 0 to Level 1 and higher, the DFDs level can be defined from very broad to very narrow. The DFD indicates that at Level 0 the system appears as a single process interacting with external entities to produce inputs and outputs, whereas Level 1 and beyond will cut processes into sub-processes. In summary, DFDs facilitate and clarify system analysis and design by defining information flow so as to detect inefficiencies and serve as a communicating medium for various stakeholders engaged in the development process. Beyond that, they offer the assurance that the entire system is well understood and correctly designed.

The Major Characteristics of a DFD:

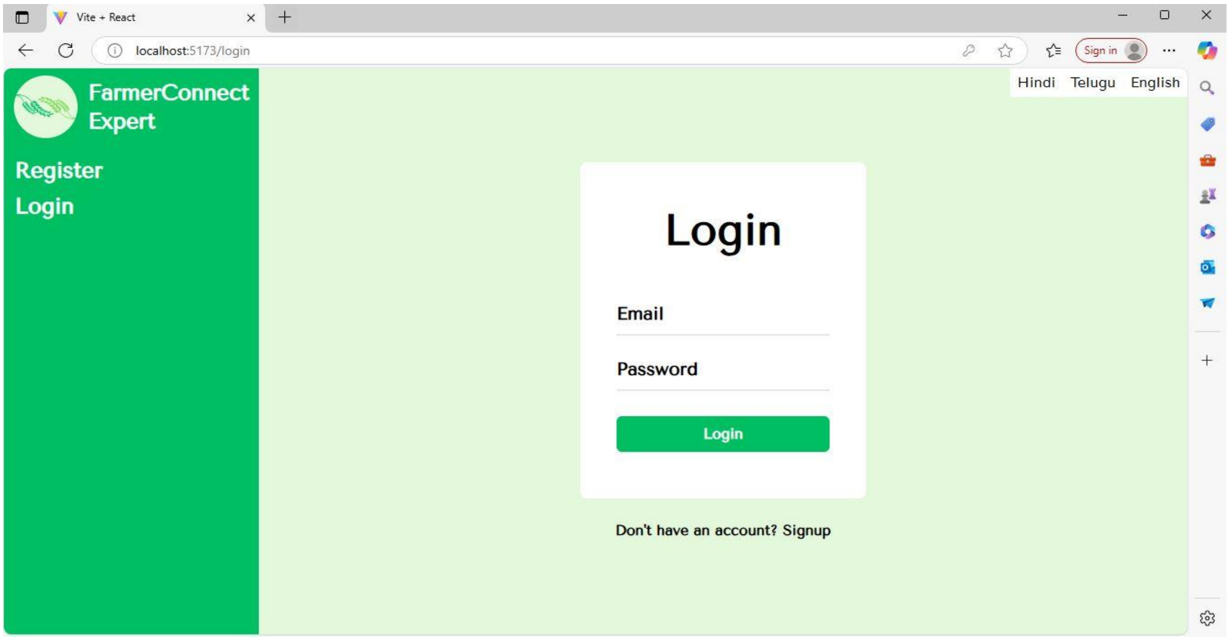
Processes: Circular or oval-object, each process has a unique identifier and a name describing what it does to the data: these are those activities or functions that change inputs into outputs.

Data Flows: These are arrows that show the flow of data between processes, data stores, and external entities. A data flow illuminates how, through the system, the information traveled and may have additional labels describing what data were transferred.

Data Stores: These open-ended rectangles show both data stores, where data is stored somewhere within the system, and where that data is kept for future retrieval or use by processes.

External Entities: These are indicated by squares or rectangles, and say what is outside the system; that is, external actors such as users, systems, or other organizations that, in one way or another, interact with the system. Inputs into the system or outputs are sent through this path.

4. EXPERIMENTAL RESULTS



Don't have an account? Signup

Fig.4.1 Login Page or Landing Page

This is a login page for a web-application called Farmer Connect Expert.

The elements are:

Logo/Title: Farmer Connect Expert is featured prominently on-site top left;

Navigation Links:

Register - This provides means for creating a new account.

Login - Below the title, this indicates the current page.

Login Form:

Centered box with the title Login.

Fields for email and password for user input and Green Login button to submit the form.

Motivational text:

A friendly prompt below: "Don't have an account? Sign up.";

It is clean and user-friendly in nature intended to easy navigate and accessible.

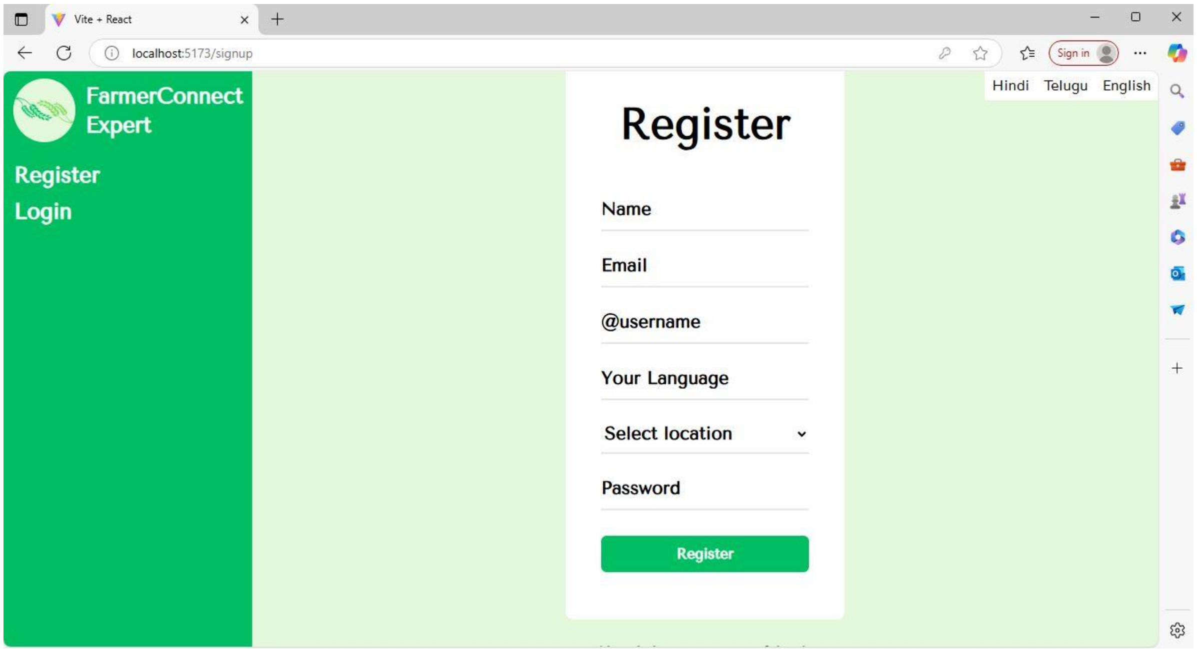


Fig.4.2 New user Registration

The image displays a registration page for the "Farmer Connect Expert." Below is a summary of the features:

Logo/Title: Farmer Connect Expert, written in the upper left corner.

Navigation Links:

Register (current page in use).

Login (link to the login page).

Registration Form:

Centered box titled Register.

Input fields for:

Name

Email

Username (at @username)

Your Language

Select Location (has a dropdown)

Password

Green Register button on the bottom for submitting the form.

The layout appeals to the user by ensuring simple registration for new users with clear labels for every field that encourages prominent inputs.

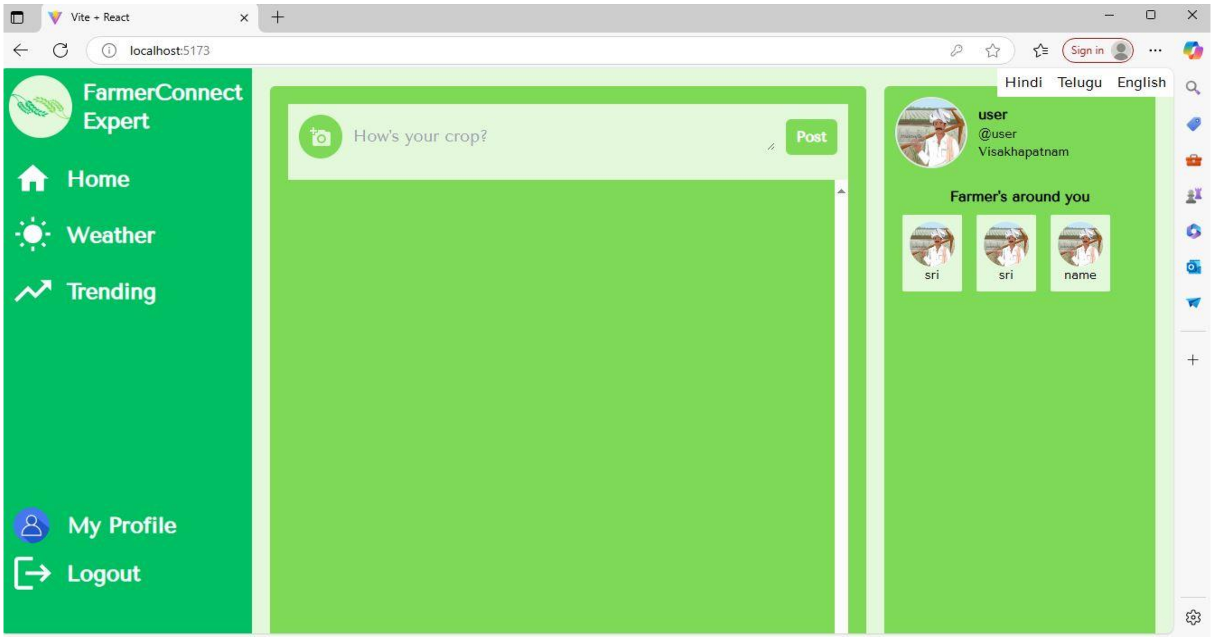


Fig.4.3 Home page

This image displays a full view of the primary dashboard of the "Farmer Connect Expert," created just as a portal for user usage and community interaction. Let's take a look at a more detailed description:

Logo/Title: The term "Farmer Connect Expert" will remain prominent at the upper-left corner.

Navigation Menu (left sidebar): Home, Weather, Trending, My Profile, Sign Out

Every option is represented with icons and text so that it could be easily understood.

Main Content Area:

There is a prompt "How's your crop?" at the top which suggests that this is a place where the focus would be on something related to users, input, or posting updates about farming.

Also has a post button available alongside input to make the updates from users' observations public.

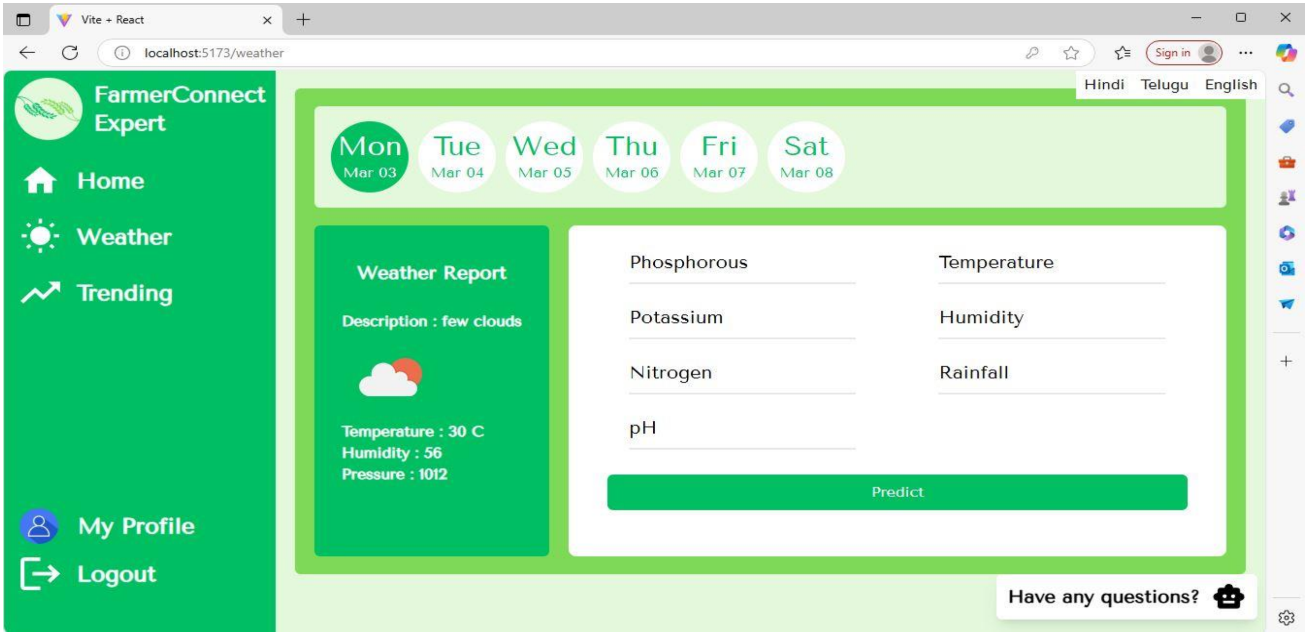
Right Sidebar:

Username

Handle, for instance, @user.

Location, for instance, Visakhapatnam.

A section "Farmer's around you" exhibits profiles or icons of nearby farmers residents probably for networking and community building.



Weather Report

Description : few clouds

Temperature : 30 C
Humidity : 56
Pressure : 1012

Fig.4.4 Weather and Crop predicting page

This is the "Weather" part of the "Farmer Connect Expert" application which is presented in a neat and clean user interface for weather information and predictions. Here is a detailed explanation:

Daily Weather Overview (top right):

- Displays buttons for moving around in the week like this: Mon, Tue, Wed, Thu, Fri

Weather Report Section:

- Has a big box titled Weather Report
- Description of present weather: "Description: few clouds".
- Present conditions of the weather: Temperature, Humidity, Pressure

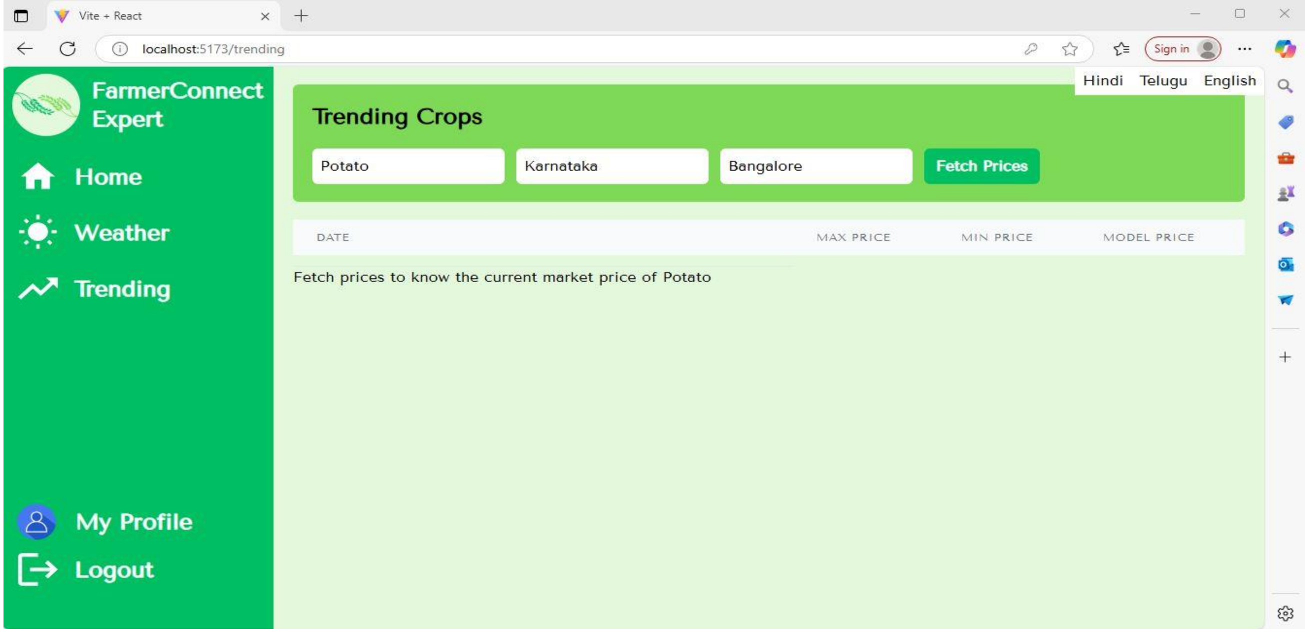
Soil Nutrient Information (right side):

- Lists key soil nutrients: Phosphorous, Potassium, Nitrogen, pH

Prediction Button: Green "Predicts" button at the bottom is probably for prognosis or analysis of agricultural situations.

Help Prompt: A small chat icon with the text "Have any questions?" at the bottom right shows support options.

As the whole said and done, this interface is designed to be useful as far as weather and soil information is concerned, particularly for the farmers thereby making their decision processes quite easy.



Trending Crops

Potato

Karnataka

Bangalore

DATE	MAX PRICE	MIN PRICE	MODEL PRICE

Fetch prices to know the current market price of Potato

Fig.4.5 Fetching the Trending crops page

This image represents the "Trending" section in "Farmer Connect Expert" that deals with currently available market prices for crops. Here is a detailed account of the same:

Area of Interest:

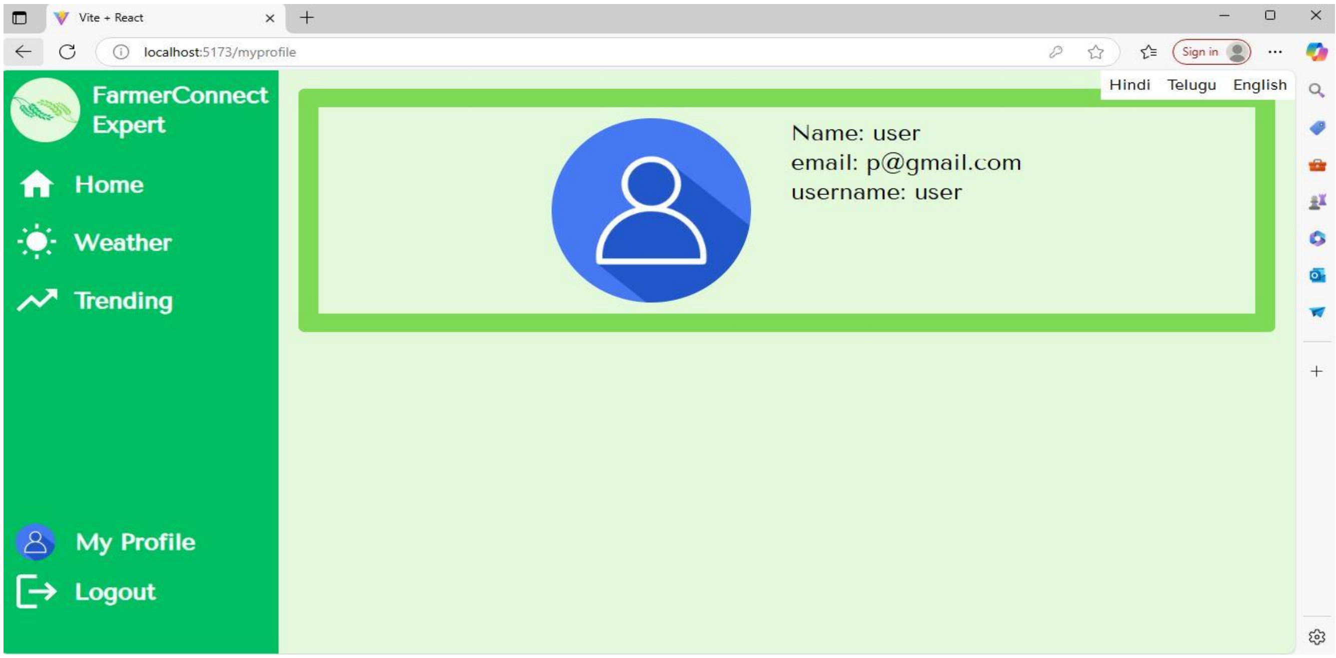
- At the top is a section titled "Trending Crops." Input Fields:
- A text box labeled "Potato," indicative of a crop to be fetched.
- Drop-down options with respect to: State, City
- A Fetch Prices button in green that allows users to fetch market rates for the selected crop.

Price Table: A section at the bottom of the input fields where headings are present:

- Date
- Max Price
- Min Price
- Model Price

Prompt Text: A text that reads, "Fetch prices to know the current market price of Potato," suggesting the need for this section to exist.

In general, it is organized in a way that provides easy access for farmers to see market pricing, thus aiding them in making decisions from the current trends.



Name: user
email: p@gmail.com
username: user

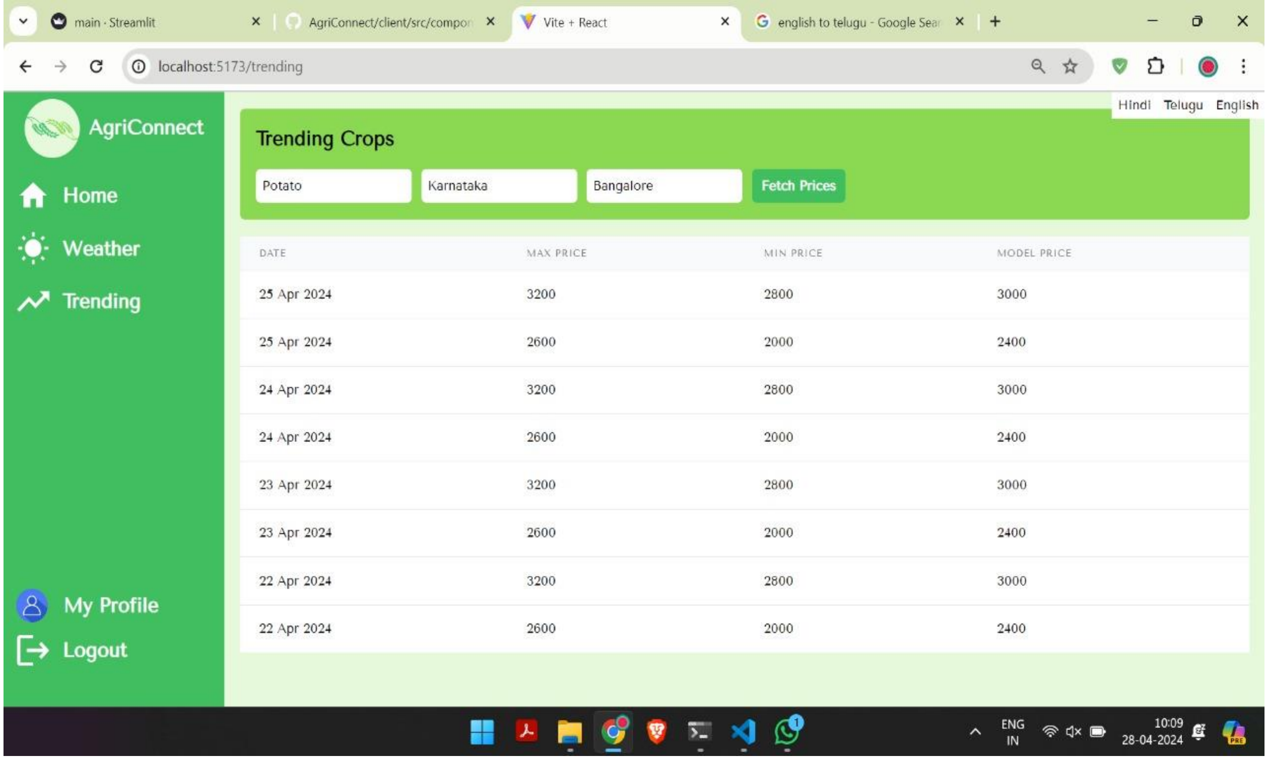
Fig.4.6 My Profile page

The "My Profile" section from the application called "Farmer Connect Expert" contains information about the user. It covers the layout and the various contents as follows:

Profile Display Area:

- An important part of this area has a green border and has a circular blue icon indicating user profile.
- **User Information:**
- Name: user
- Email Id: p@gmail.com
- Username: user
- The User Information is formatted clearly for ease of reading.

Overall, the layout simplified and it designed so now it will give all the necessary profiles of a user in an easy way to let Students view their information easily.



Trending Crops

Potato | Karnataka | Bangalore | Fetch Prices

DATE	MAX PRICE	MIN PRICE	MODEL PRICE
25 Apr 2024	3200	2800	3000
25 Apr 2024	2600	2000	2400
24 Apr 2024	3200	2800	3000
24 Apr 2024	2600	2000	2400
23 Apr 2024	3200	2800	3000
23 Apr 2024	2600	2000	2400
22 Apr 2024	3200	2800	3000
22 Apr 2024	2600	2000	2400

Fig.4.7 Final Performance results

The image is that of the overpage of the "Farmer Connect Expert" application where in the market price for crops becomes the focus of the application. A detailed breakdown of layout and content is as follows:

The Data Table for Market Price:

- Its columns say Date, Max Price, Min Price, and Model Price.
- Number of rows in the data with prices mentioned
- The prices differ for maximum and minimum prices, alongside a proposed model price.

To sum up, the layout is user-friendly, providing for easy navigation, allowing farmers to get timely market price information that helps them sell and price their crops efficiently.

5. COMPARATIVE STUDY

The aim of this comparative study is to analyze and compare Farmer Connect Expert, a full-fledged agricultural platform, with several existing platforms.

Table 5.1: Comparison between the Farmer Connect Expert and Other Agricultural Platforms

Aspect	Farmer Connect Expert	Other Agricultural Platforms
Approach	An enterprise service provider for personalized crop advice, real-time market rates, multilingual services, artificial intelligence chatbot, and community engagement.	In essence, the service is a B2B solution, with recommendations on crops and immediate market price reports. The service has been low on social or community functions.
Algorithm Used	-Machine Learning (For crop recommendations and marketplace price predictions) - AI multi-lingual chatbot with OpenAI - APIs for weather, soil and market data integration	- A straightforward model rule based on which crops can be recommended - Not very limited machine learning models (predictive for crop health or yield) - Basic AI chatbots on certain platforms
Metrics	- Prediction Accuracy for crop recommendations (~92%) - that means it displays accuracy for market price prediction almost in the range of 90-95%. - Chatbot effectiveness is promising, sitting just above the 91% accuracy level. -There is high engagement on the social platform.	- The accuracy of crop recommendation is measured to be around 85%-88% - Accuracy of market price prediction has been noted to compare to roughly 75%-85% - Effectiveness of the chatbot could be rated from 60% to 70% - Moderate user engagement is in the forums, customer support
Best Model	- Random Forest Classifier crop recommendation - AI OpenAI chatbot for user interaction - Real-time API for market prices and weather data.	- Crop recommendations with the help of neural networks, decision trees, and k-nearest neighbors - Building of a basic chatbot using AI with a single language option

Farmer Connect Expert prides itself on providing full personal agricultural support. Actual market data, multilingual support, AI-driven crop recommendations, and a well-knit social platform make it inclusive and innovative for farmers. For all other agricultural platforms offering many services, Farmer Connect Expert integrates the most diverse features, gets fostered by community engagement, and has great plans for the future. But as with any tech-based initiative, scaling and API integration pose challenges; however, consistent improvement made by Farmer Connect Expert is a promising model for the future of agricultural platforms.

Detailed Explanation of the above Table 5.1: -

Approach: E-commerce will then bring the user to a more integrated and holistic approach by taking personalized recommendations, multilingualization, mobilization with the community, real-time data through APIs. Other agriculture sites focus on advisory services, connect some different features, such as crop-specific recommendations and market information.

Algorithm Applied: A Random Forest Classifier is used for data harvesting in crop recommendations by Farmer Connect Expert apart from an AI chatbot which uses OpenAI for serving its purpose. Other platforms use comparatively simple models and rule-based systems that usually do not provide an equal level of personalization or accuracy as in Farmer Connect Expert.

Metrics: Farmer Connect Expert is the most successful in terms of accuracy and engagement; it had better prediction accuracy on crop recommendations as well as market prices compared to the other ones. Its chatbot is also better, as it can understand several languages and is highly accurate. The rest are usually on the lower side concerning the metrics with inefficient chatbots and slow, inaccurate data.

Best Model: The Random Forest Classifier and the OpenAI-powered chatbot are the best models at Farmer Connect Expert for highly accurate recommendations and interactive support. Other platforms utilize models like Neural Networks and Decision Tree, but most of the time, these are not as integrated or personalized.

.

Table 5.2: Comparison between previous work and current work

Prediction Model	Current Work	Previous Work
Random Forest Classifier	G 0.9395	0.76 (Accuracy)
Random Forest Classifier E	0.9366	0.76 (Accuracy)
Extra Trees Classifier	0.9327	0.73 (Accuracy)
K-Nearest Neighbors Classifier	0.6275	0.82 (Accuracy)
Gaussian Naive Bayes	0.9418	0.76 (Accuracy)
Bernoulli Naive Bayes	0.8917	0.75 (Accuracy)

Detailed Explanation of the above Table 5.2: -

The most remarkable part of this comparison is that the present work exhibits a better performance in the most models, especially the Random Forest Classifier, Extra Trees Classifier, Gaussian Naïve Bayes, and Bernoulli Naïve Bayes models, for which it has surpassed the previous work considerably. The current work puts forth valuable progress made in the accuracy of models, which could be considered as improvements in data processing and feature engineering, as well as the entire weight of such machine learning algorithms.

However, even since a well-performing model in past work, performance with the KNN model decreased in the current work, perhaps due to reasons such as overfitting or the presence of problems with feature scaling. Further optimization and fine-tuning of KNN may be required for better performance results.

In fact, the present work has proven to have a much stronger and robust performance for the prediction of the results when compared with the previous work, indicating that probably the improvements were in model selection or other data handling strategy, producing those improvements in accuracy and reliability.

6. TESTING

6.1 Types of Tests

System Testing checks the entire software for quality. A defective software is checked using different types of tests. The smaller the unit of the software, the higher the possibility of quality testing. These small units contain testable components, functions, reusable elements, etc. which get used in the software assignment. Testing also confirms whether the software actually meets the functional requirements earlier described by the customer during the requirements gathering phase.

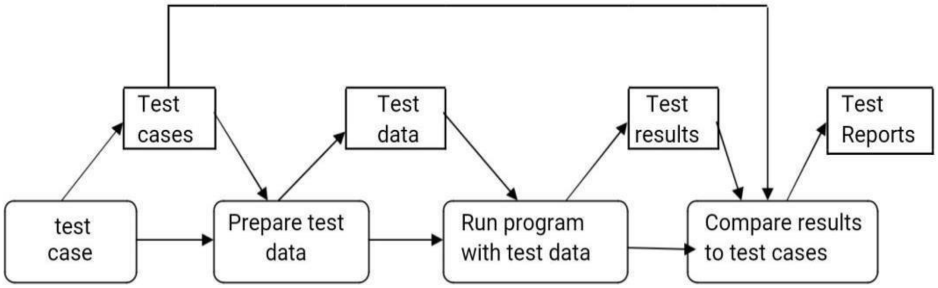


Fig: 6.1. System Testing Process

Unit testing

Unit Testing is basically dividing a system into a small set of individuals. Once separated into units, finding bugs and testability becomes easy. The purpose of unit testing is to verify if a unit behaves as expected.

Integration testing

The purpose of this testing is error finding when parts interact. The units that have been separately tested are to be combined to see if there is any interference or difference in their performance while they cooperatively work in the software.

Test strategy and approach

With an effective testing strategy, it lays the whole framework for product testing to find and fix defects.

Test Results: All test cases mentioned above were executed, and they passed the entire sequence of tests without having any defect at all.

Acceptance Testing

This phase of testing is the most complex of all. At this point, the user's decision is needed as to whether the project actually meets the functional requirements mentioned in the master document.

6.2 TEST STRATEGY AND APPROACH

This assures that the entire network of application software is validated according to the specifications. An assurance that the design is appropriately verified and consistent with the rest. One type of testing is termed design-oriented integration control of plant machinery. Validation of the system involves method and flow definitions, focusing on pre-integration and integration points.

White Box Testing

In simplest terms, White Box Testing is the study that informs the tester of the language, design, and implantation of the program-or even its intention. Its aim then becomes checking from the other end regions of the black box, previously inaccessible. This software testing method is when the tester knows the internal structure of the software. It is also known as glass box testing, open box testing, transparent box testing, design testing and clear box testing. All of these terms connote that the internal workings are open to the perception of the tester. The actual testing of internal source code and the software architecture comes under white box testing. White box testing is a strategy focused on analyzing and validating the internal framework, architecture, components, mechanisms, and objects of an application.

Initially, the code is tested against documents. If the tester knows all the details regarding the internal structure of the code, then he or she tests to see if the system actually meets the requirements set forth by the particular document. The objectives of the tests are to enhance security, improve the input and output flow and improve usability and manageability. This identifies vulnerabilities in applications. The best method to catch imperfections at the earliest possible stage of the Software Development Life Cycle (SDLC) is this Testing.

The most important part is the creation of test cases. To do that, a basic understanding of programming is needed from the testers. These test cases then can be applied to every code line, detecting all anomalies upon the way. Obviously, it needs to be understood exactly what the program is supposed to do. This is important. Only then can they actually determine when a program is inconsistent with its intended purpose.

Black Box Testing

Black Box Analysis evaluates the program without knowing anything about the application's internal processes, configuration, and language. Like most forms of examinations, a black box evaluation must come from some final source text, such as a design text or a manual of requirements, such as a design document or the requirement manual. The check is a Blackbox under which the device is assessed. There's nothing you can "feel." The check produces feedback and reacts to results without being able to understand how the programme works. Black box testing is the software test method for software testing, but the testing techniques are unclear to the testers about the applications' internal work. Thus, the method is called Black Box testing as the tester is all blind in the software tested. The internal code structure, software paths and details of implementation do not concern the Black Box Tests tester. The behavior of the application's functions is the only thing that counts for this test type. Because it checks the functioning of a programme, the Black Box is also known as functional testing.

The non-functional Black Box tests, apart from functional tests, also account for how long data takes to process and result generation or how long it takes for a function to respond to a user's input. The Black Box tests therefore gather results and compare them with the expected results. This test exercise is based on software requirements and specifications.

Mostly testers perform tests through a Black Box testing technique. No functional knowledge is available for the tester about the application other than that he knows the requirements to test. He does inspection on both valid and invalid cases and finds matching right expected outputs. Hence, all test cases are derived from requirements and specifications document. The main purpose is to find out whether a software product is functioning as planned and whether it meets the values of the user.

6.3 TEST CASES

+VE TEST CASES

S. No	Test case Description	Actual value	Expected value	Result
1	Verify successful user login with valid credentials	User logged in successfully	User logged in successfully	True
2	Verify crop recommendation based on soil fertility and weather	Crop recommendation provided correctly	Crop recommendation matches the soil and weather data	True
3	Verify market trends and prices are displayed correctly	Market trends and prices fetched via API	Market trends and prices display correctly in real-time	True
4	Verify multilingual support works for Telugu	UI displayed in Telugu language	UI displayed in Telugu language	True
5	Verify AI chatbot provides accurate response in Hindi	Chatbot responded with relevant info in Hindi	Chatbot responded with relevant info in Hindi	True
6	Verify social platform allows sharing of farming experiences	User posted an update successfully	User posted an update successfully	True
7	Verify personalized crop recommendation after user inputs	Recommendations based on user input	Personalized crop recommendations displayed	True
8	Verify that users are redirected to the dashboard post-login	User redirected to dashboard after login	User redirected to dashboard as expected	True
9	Verify the functionality of the "Forgot Password" feature	Password reset link sent to the user’s email	Password reset link sent to user’s email	True
10	Verify that the system suggests suitable crops for the region	Crop suggestions based on user’s region	Crop suggestions based on correct region data	True

Table.6.3.1 +VE Test case

-VE TEST CASES

S. No	Test case Description	Actual value	Expected value	Result
1	Verify performance issues when too many users are interacting simultaneously	System crash or slow response.	System should handle multiple users without issues	False
2	Verify incorrect market trends and prices when API data is corrupted	Incorrect market data shown	Correct market data should be displayed	False
3	Verify login with invalid credentials (wrong username)	Error message: "Invalid username"	Error message: "Invalid username".	True
4	Verify login with invalid credentials (wrong password)	Error message: "Incorrect password"	Error message: "Incorrect password"	True
5	Verify crop recommendation when no soil data is available	Error message: "Soil data not available"	Error message: "Soil data not available"	True
6	Verify market trends and prices when API is unavailable	Error message: "Unable to fetch market prices"	Error message: "Unable to fetch market prices"	True
7	Verify AI chatbot fails to respond when language is unsupported	Error message: "Language not supported"	Error message: "Language not supported"	True
8	Verify social platform posting fails with empty content	Error message: "Post cannot be empty"	Error message: "Post cannot be empty"	True
9	Verify crop recommendation when weather data is unavailable	Error message: "Weather data not found"	Error message: "Weather data not found"	True
10	Verify user is not redirected to dashboard if login is unsuccessful	User remains on login page	User should be redirected to dashboard after successful login	True

Table.6.3.2 -VE Test case

7.CONCLUSION AND FUTURE SCOPE

Main Objective of this Project Assist Indian Farmers to Increase His Income by Providing Easy Access to Necessary Resources and Information. Thus, Simplifying Agricultural Processes and Better Livelihood for Farmers. It Is Moreover a Networking Application for Indian Farmers to Increase Their Revenue.

Farmers prefer such applications: multilingual and available through mobile rather than websites. This shows the need for user-friendly technology for-farmers specific needs.

Farmer Connect Expert has dealt with the agriculture access problem for very easy access to strong agricultural contents and consumables across the fields, a comprehensive form of personalized crop suggestions, real-time market trends, and multilinguistic support and assistance through AI on very easy-to-use understanding-in-formulations by the farmers in decision-making towards effective farming practice. It becomes richer for the user through future marketplace extensions of ML, social features, and a real sense of community towards users.

Farmer Connect Expert, even with those challenges, would come out as a strong and scalable solution because of API integration, seamless scalability with optimal performance, and multilingual support. Our journey has also taken us through interesting lessons on how technology plays a role in agriculture generally, productivity improvement, and seamless collaboration at the farming level.

Farmer Connect Expert plans to build on that momentum by way of delivering an even better marketplace, wider language support, and more community-building initiatives. By leveraging continuous innovation, we endeavor to close the gap between technology and agriculture, ensuring that farmers from various territories can make use of the necessary tools and knowledge to improve the way for an agriculture gradually changing landscape.

APPENDICES

A. SAMPLE CODE

*MAIN.PY

```
from flask import Flask, request, json, jsonify

from ask import Ask

import pickle

import time

from retrieve import script

from flask_cors import CORS

# create the Flask app

app = Flask(__name__)

CORS(app, supports_credentials=True)

askinstance = Ask()

def load_model(model_path):

    with open(model_path, 'rb') as f:

        model = pickle.load(f)

    return model

# Load the pickled model

model_path = "model.pkl"

model = load_model(model_path)

# allow both GET and POST requests
```

```
@app.route('/predict', methods=['POST'])

def form_example():

    # handle the POST request

    if request.method == 'POST':

        request_data = request.get_json()

        print("start")

        nitrogen_value = int(request_data["N"])

        phosphorus_value = int(request_data["P"])

        pottasium_value = int(request_data["K"])

        temperature_value = float(request_data["temperature"])

        humidity_value = float(request_data["humidity"])

        ph_value = float(request_data["ph"])

        rainfall_value = float(request_data["rainfall"])

        input_data = [[nitrogen_value, phosphorus_value, pottasium_value,

                       temperature_value, humidity_value, ph_value, rainfall_value]]

        nitrogen_value = int(request_data["N"])

        phosphorus_value = int(request_data["P"])

        pottasium_value = int(request_data["K"])

        temperature_value = float(request_data["temperature"])

        humidity_value = float(request_data["humidity"])

        ph_value = float(request_data["ph"])

        rainfall_value = float(request_data["rainfall"])
        input_data = [[nitrogen_value, phosphorus_value, pottasium_value,
                    temperature_value, humidity_value, ph_value, rainfall_value]]
        crop_prediction = model.predict(input_data)
        data = {"suggestion": crop_prediction[0]}
        print("done")
        return json.dumps(data)
@app.route('/chat', methods=['POST'])
def form_example1():
    # handle the POST request
    if request.method == 'POST':
        request_data = request.get_json()
        message = request_data["message"]
        response = askinstance.process(message)
        data = {"message": response}
        return json.dumps(data)
@app.route('/request', methods=['GET'])
def requestPage():
    commodityQuery = request.args.get('commodity')
    stateQuery = request.args.get('state')
    marketQuery = request.args.get('market')
    if not commodityQuery or not stateQuery or not marketQuery:
        return jsonify({"error": "Missing query parameters"})
    try:

        json_data = json.dumps(

            script(stateQuery, commodityQuery, marketQuery), indent=4)

        return json_data

    except Exception as e:

        return jsonify({"error": str(e)})

if __name__ == '__main__':

    # run app in debug mode on port 5000

    app.run(host="0.0.0.0", debug=True, port=5000)
```

***ASK.PY**

```
from langchain.prompts.chat import ChatPromptTemplate
from langchain.chat_models import ChatOpenAI
from langchain.memory import ConversationBufferMemory
from langchain.chains import ConversationChain
from dotenv import load_dotenv
import os
load_dotenv()
class Ask:
    def __init__(self):
        # Define the language model
        self.llm = ChatOpenAI(
            model="gpt-3.5-turbo",
            temperature=0,
            max_tokens=1000,
            openai_api_key=os.getenv('OPENAI_API_KEY'),
        )
        memory = ConversationBufferMemory(memory_size=3)
        self.conversation = ConversationChain(llm=self.llm,
                                              verbose=True,
                                              memory=memory,
                                              )
        self.i = 0
    def process(self, user_input):
        """
        Process user input and return the assistant's response.
        """
        language = ChatPromptTemplate.from_template("""
            Identify the language of the user input {input}.
            Your response should be one word.
""")
        chain = language | self.llm
        lang = chain.invoke({'input': user_input})

#         translation=ChatPromptTemplate.from_template("""
#             User can question in any language.
#             If the user language is not English, the input will be translated to English or else
the input will be used as it is.
#             Here is the user question: {input}
# """)
#         chain = translation | self.llm
#         translated_prompt = chain.invoke({'input': user_input})

        prompt = ChatPromptTemplate.from_template("""
        You are helpful assistant that helps farmers with their data-related queries.
        Here are the topics that they will ask can use:
        weather, crop diseases, cultivation techniques, market prices, fertilizers and soil health
        Here is the user question: {input}
        Your response should be accurate.
        """)
        chain = prompt | self.llm
        decision = chain.invoke({'input': user_input})

        translation2 = ChatPromptTemplate.from_template("""

                convert the response {input} to the user language{lang}.

""")

        chain = translation2 | self.llm

        translated_prompt2 = chain.invoke(

            {'input': decision.content, 'lang': lang.content})

        return translated_prompt2.content
```

***RETRIEVE.PY**

```
import time
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.support.ui import Select
from selenium.webdriver.common.by import By
from datetime import datetime, timedelta


def script(state, commodity, market):
    # URL of the website with the dropdown fields
    initial_url = "https://agmarknet.gov.in/SearchCmmMkt.aspx"
    driver = webdriver.Chrome()
    driver.get(initial_url)
    print("Commodity")
    dropdown = Select(driver.find_element("id", 'ddlCommodity'))
    dropdown.select_by_visible_text(commodity)
    print("State")
    dropdown = Select(driver.find_element("id", 'ddlState'))
    dropdown.select_by_visible_text(state)
    print("Date")
    # Calculate the date 7 days ago from today
    today = datetime.now()
    desired_date = today - timedelta(days=7)
    date_input = driver.find_element(By.ID, "txtDate")
    date_input.clear()
    date_input.send_keys(desired_date.strftime('%d-%b-%Y'))
    print("Click")
    button = driver.find_element("id", 'btnGo')
    button.click()
    time.sleep(3)
    print("Market")
    dropdown = Select(driver.find_element("id", 'ddlMarket'))
    dropdown.select_by_visible_text(market)
    print("Click")
    button = driver.find_element("id", 'btnGo')
    button.click()
    time.sleep(1)
    driver.implicitly_wait(10)
    from selenium.webdriver.support.ui import WebDriverWait
    from selenium.webdriver.support import expected_conditions as EC
    # Wait for the table to be present
    table = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, 'cphBody_GridPriceData'))
    )
    soup = BeautifulSoup(driver.page_source, 'html.parser')
    data_list = []
    # Iterate over each  row
    for row in soup.find_all("tr"):
        data_list.append(row.text.replace("\n", "_").replace("  ", "").split("__"))
    jsonList = []
    for i in data_list[4:len(data_list) - 1]:
        d = {}
        d["S.No"] = i[1]
        d["City"] = i[2]
        d["Commodity"] = i[4]
        d["Min Prize"] = i[7]
        d["Max Prize"] = i[8]
        d["Model Prize"] = i[9]
        d["Date"] = i[10]
        jsonList.append(d)
    driver.quit()
    return jsonList
```

***SERVER.JS**

```
const express = require("express");

const colors = require("colors");

const dotenv = require("dotenv");

const cors = require("cors");

const connectDB = require("./config/db");

require("dotenv/config");


const userRouter = require("./Router/userRouter");

const postRouter = require("./Router/postRouter");


//databse config

connectDB();


//rest object

const app = express();


//middelwares

app.use(cors());

app.use(express.json());


app.use("/user", userRouter);

app.use("/posts", postRouter);


//rest api
```

```
app.get("/", (req, res) => {
  res.send("<h1>Welcome to Agri Connect</h1>");
});

//PORT
const PORT = process.env.PORT || 8080;

//run listen
app.listen(PORT, () => {
  console.log(
    `Server Running on ${process.env.DEV_MODE} mode on port ${PORT}`.bgCyan
      .white
  );
});
```

***POSTCONTROLLER.JS**

```
const { Post } = require("../Model/postModel");
const { User } = require("../Model/userModel");

module.exports.postContent = async (req, res) => {
  try {
    const { link, message, type } = req.body;
    const user = await User.findOne({ email: req.email });
    const post = new Post({
      link,
      type,
      message,
      name: user.name,
      username: user.username,
    });
    await post.save();
    await User.findOneAndUpdate(
      { email: req.email },
      {
        $push: {
          posts: post._id,
        },
      }
    );
    res.status(200).send({
      success: true,
      message: "Post updated successfully",
      post,
    });
  } catch (error) {
    console.log("Error in posting content", error);
    res.status(500).send({
      success: false,
      message: "Couldn't post the image",
    });
  }
};

module.exports.getPosts = async (req, res) => {
  try {
    let posts = await Post.find({});
    res.status(200).send({
      success: true,
      message: "All posts fetched successfully!",
      posts,
    });
  } catch (error) {
    console.log("Error in posting content", error);
    res.status(500).send({
      success: false,
      message: "Couldn't post the image",
    });
  }
};

module.exports.getUserPosts = async (req, res) => {
  try {
    const userId = req._id;
    const postIds = await User.findOne({ _id: userId }).select("posts");

    const data = await Post.find({ _id: { $in: postIds.posts } }).select(
      "username name link createdAt"
    );

    res.json({ success: true, data });
  } catch (error) {
    console.log(error);
    res
      .status(400)
      .json({ success: false, message: "Failed to get user posts" });
  }
};
```

***USERCONTROLLER.JS**

```
const { User } = require("../Model/userModel");
const { comparePassword, hashPassword } = require("./../helpers/authHelper.js");

module.exports.signUp = async (req, res) => {
  try {
    const { name, username, email, password, location, language } = req.body;
    const existingUser = await User.findOne({
      email: req.body.email,
    });
    if (existingUser)
      return res.send({ success: false, message: "User already exists" });

    const hashedPassword = await hashPassword(password);
    const user = await new User({
      name,
      username,
      email,
      location,
      language,
      password: hashedPassword,
    }).save();

    res.status(201).send({
      success: true,
      message: "User Register Successfully",
      user,
    });
  } catch (error) {
    console.log(error);
    res.status(500).send({
      success: false,
      message: "Error in Registeration",
      error,
    });
  }
};

module.exports.signIn = async (req, res) => {
  try {
    const { email, password } = req.body;

    //validation
    if (!email || !password) {
      return res.status(404).send({
        success: false,
        message: "Invalid email or password",
      });
    }

    //check user
```

```
const user = await User.findOne({ email });
if (!user) {
  return res.status(404).send({
    success: false,
    message: "Email is not registerd",
  });
}
const match = await comparePassword(password, user.password);
if (!match) {
  return res.status(200).send({
    success: false,
    message: "Invalid Password",
  });
}
//token
const token = user.generateJWT();
res.status(200).send({
  success: true,
  message: "login successfully",
  user: {
    _id: user._id,
    name: user.name,
    username: user.username,
    email: user.email,
    location: user.location,
    language: user.language,
      },
      token,
    });
  } catch (error) {
    console.log(error);
    res.status(500).send({
      success: false,
      message: "Error in login",
      error,
    });
  }
};

module.exports.getNeighbours = async (req, res) => {
  try {
    const location = req.params.location;
    let users = await User.find({ location }).limit(10);
    users = users.map((user) => ({
      _id: user._id,
      name: user.name,
      image: user.image,
    }));
    res.send({
      success: true,
      users,
    });
  } catch (error) {
    console.log(error);
    res.status(400).send({ success: false, message: "Request failed" });
  }
};
```

B. UML DIAGRAMS

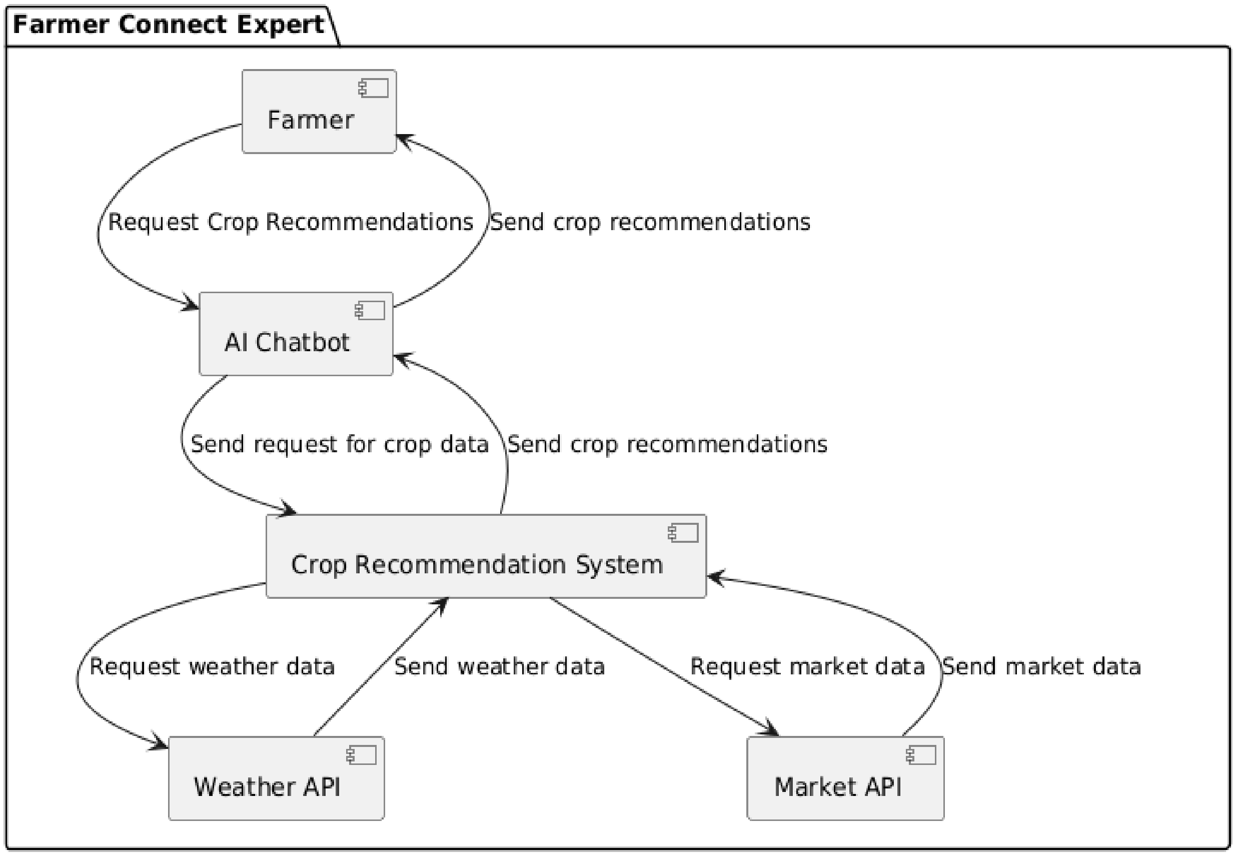

Farmer Connect Expert
Farmer
Request Crop Recommendations
Send crop recommendations
AI Chatbot
Send request for crop data
Send crop recommendations
Crop Recommendation System
Request weather data
Send weather data
Request market data
Send market data
Weather API
Market API

Fig.B1. Data Flow Diagram

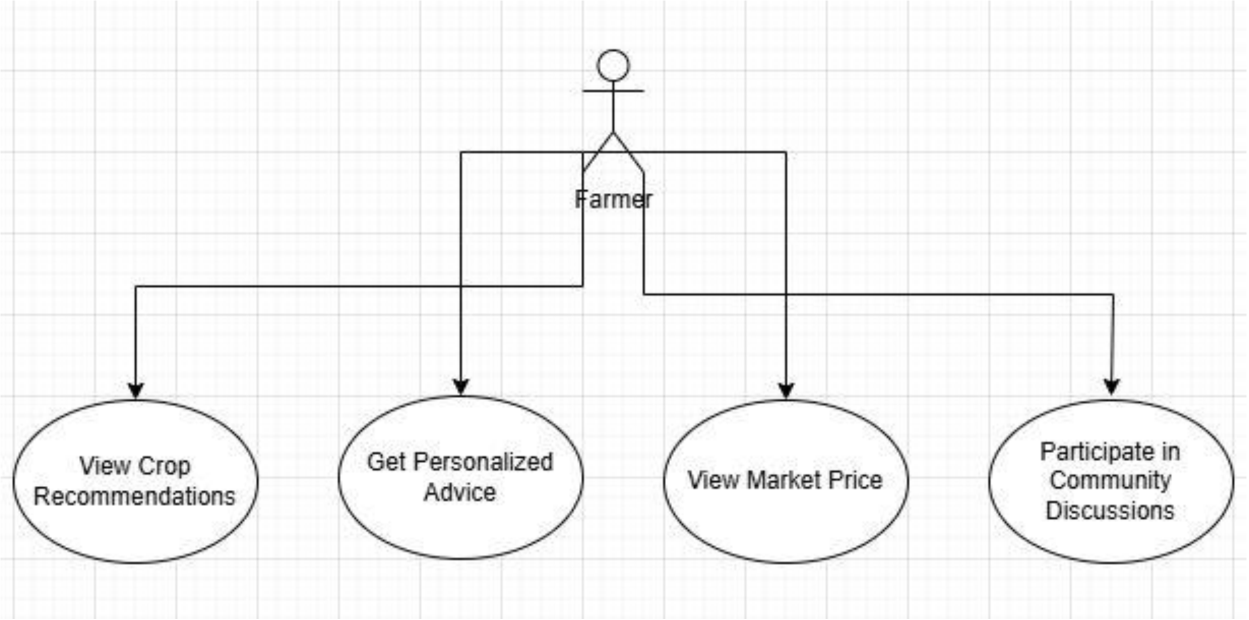

Farmer
View Crop Recommendations
Get Personalized Advice
View Market Price
Participate in Community Discussions

Fig.B2. Farmer Use case

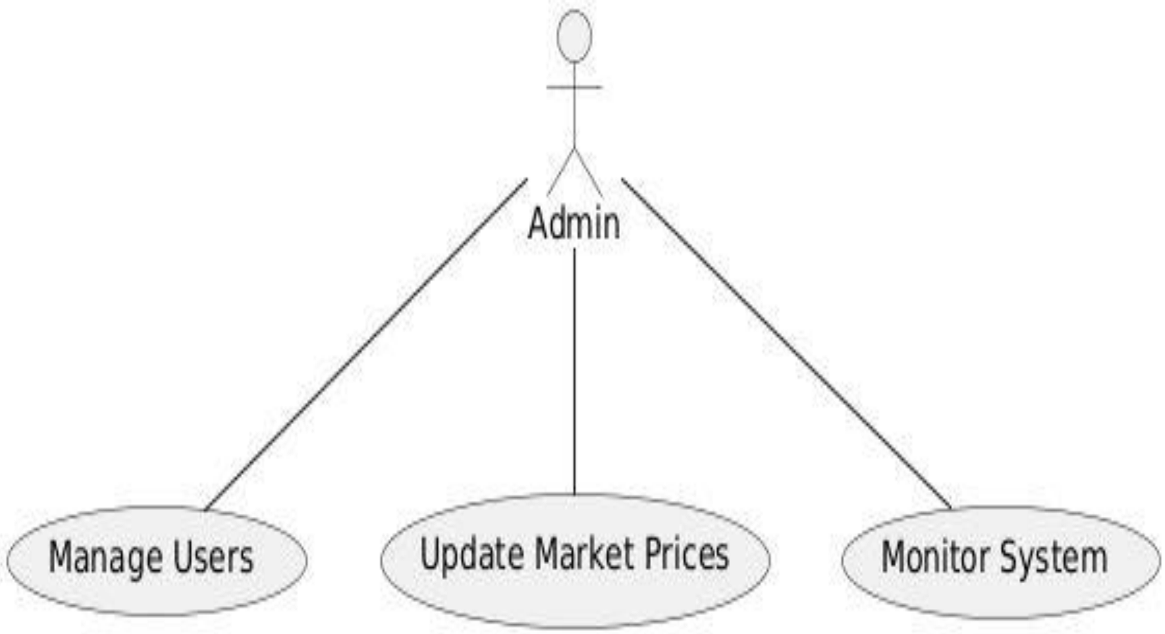


Fig.B3. Admin Use case

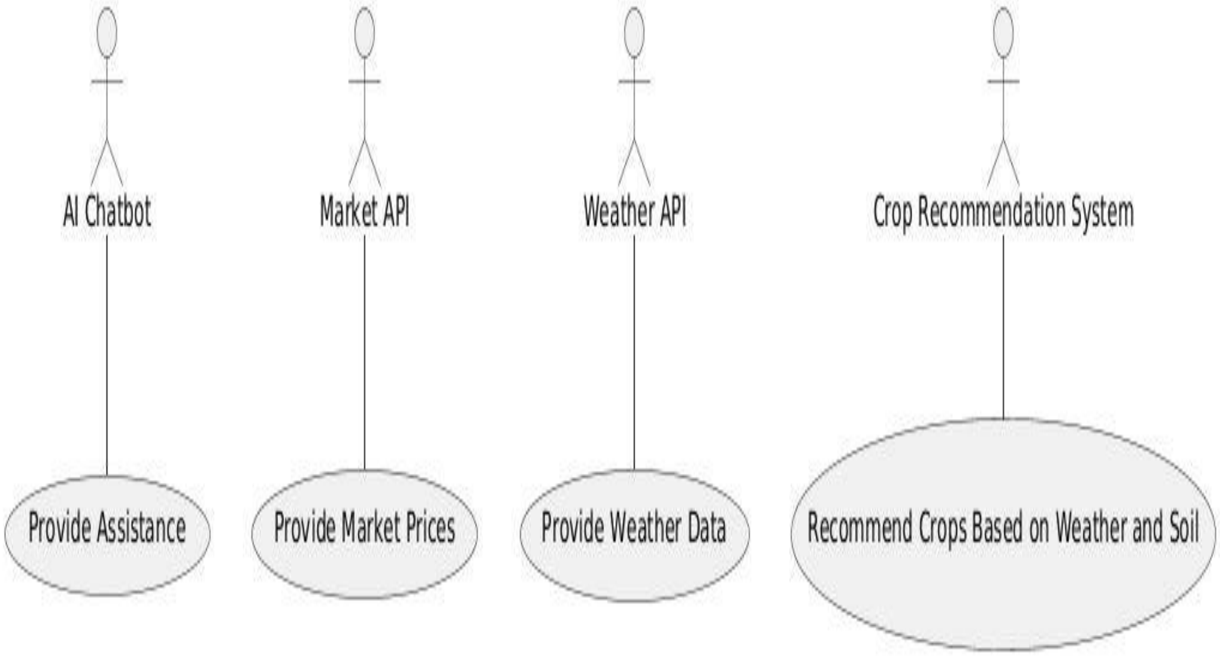


Fig.B4. API’s Use case

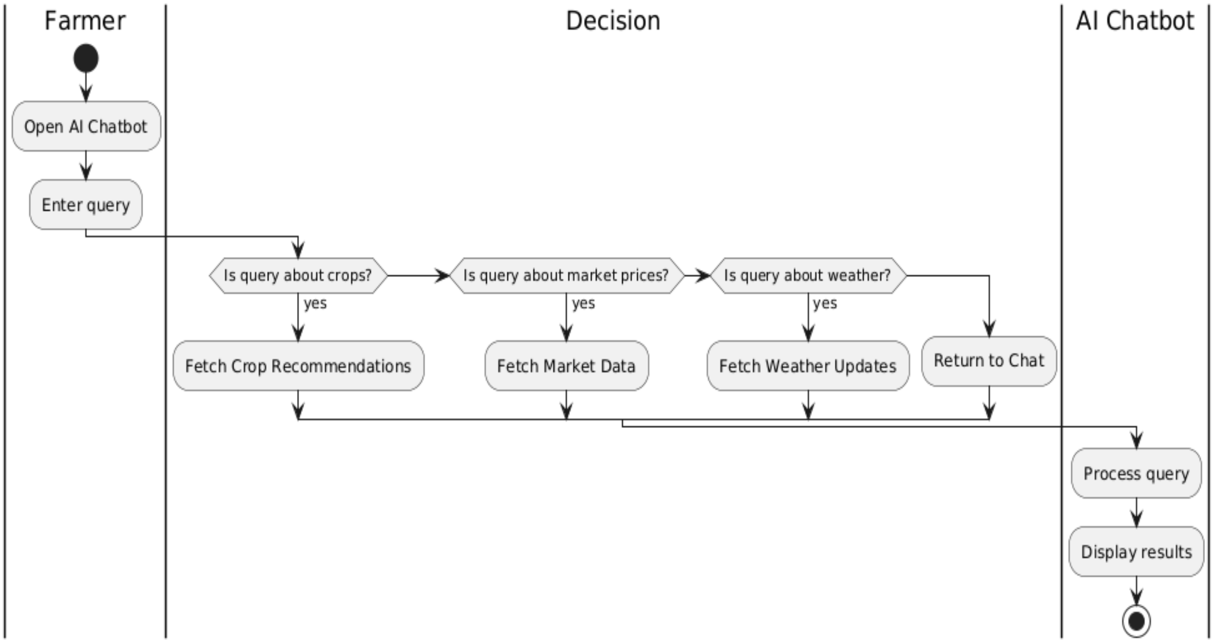


Fig.B5. Activity Diagram

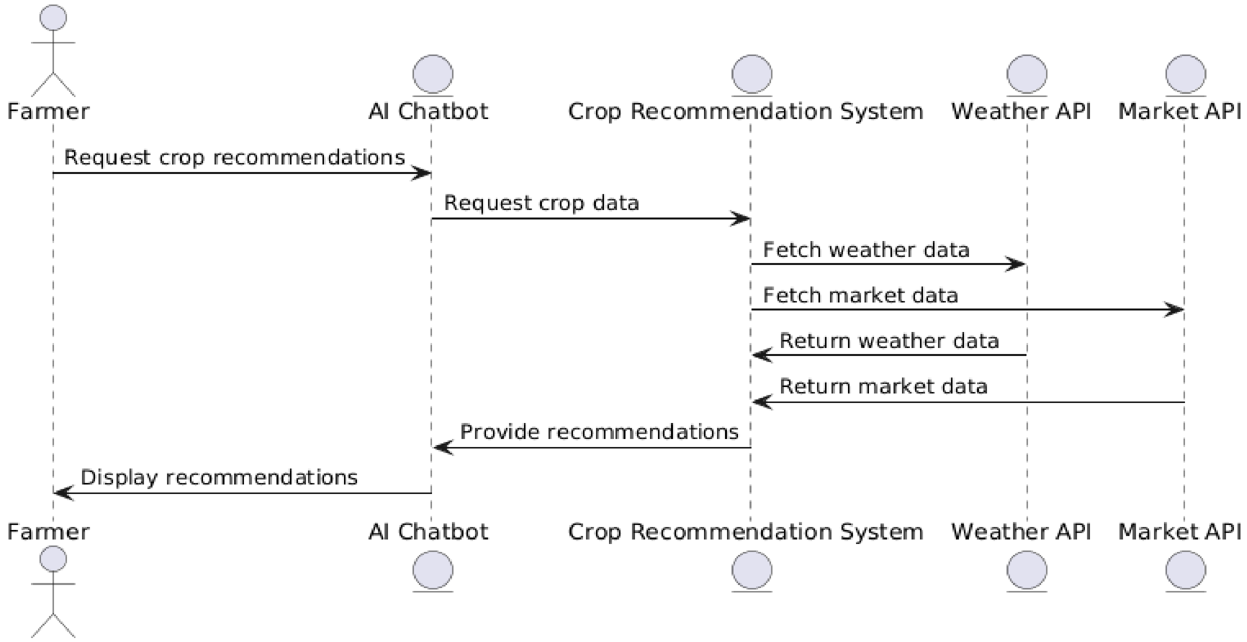


Fig.B6. Collaboration Diagram

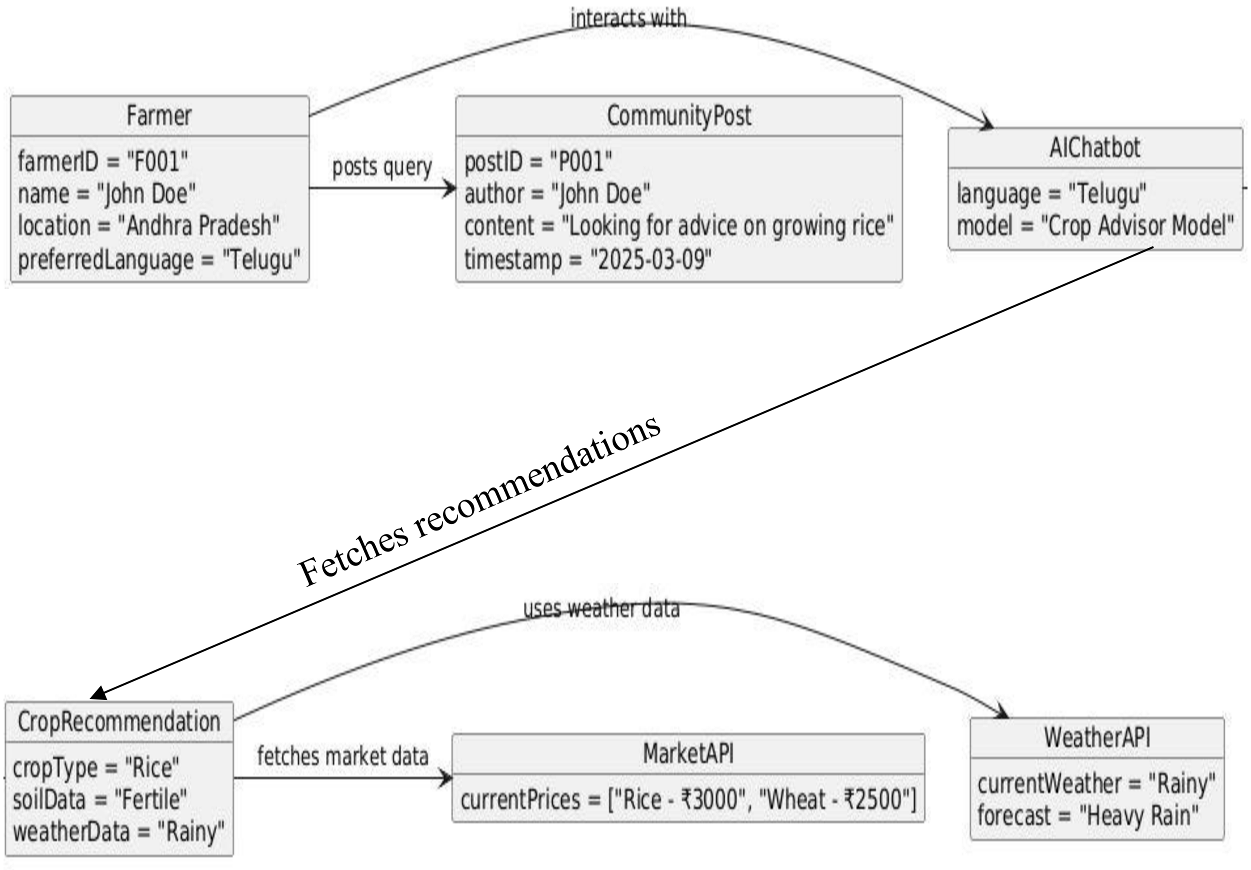


Fig.B7. Object Diagram

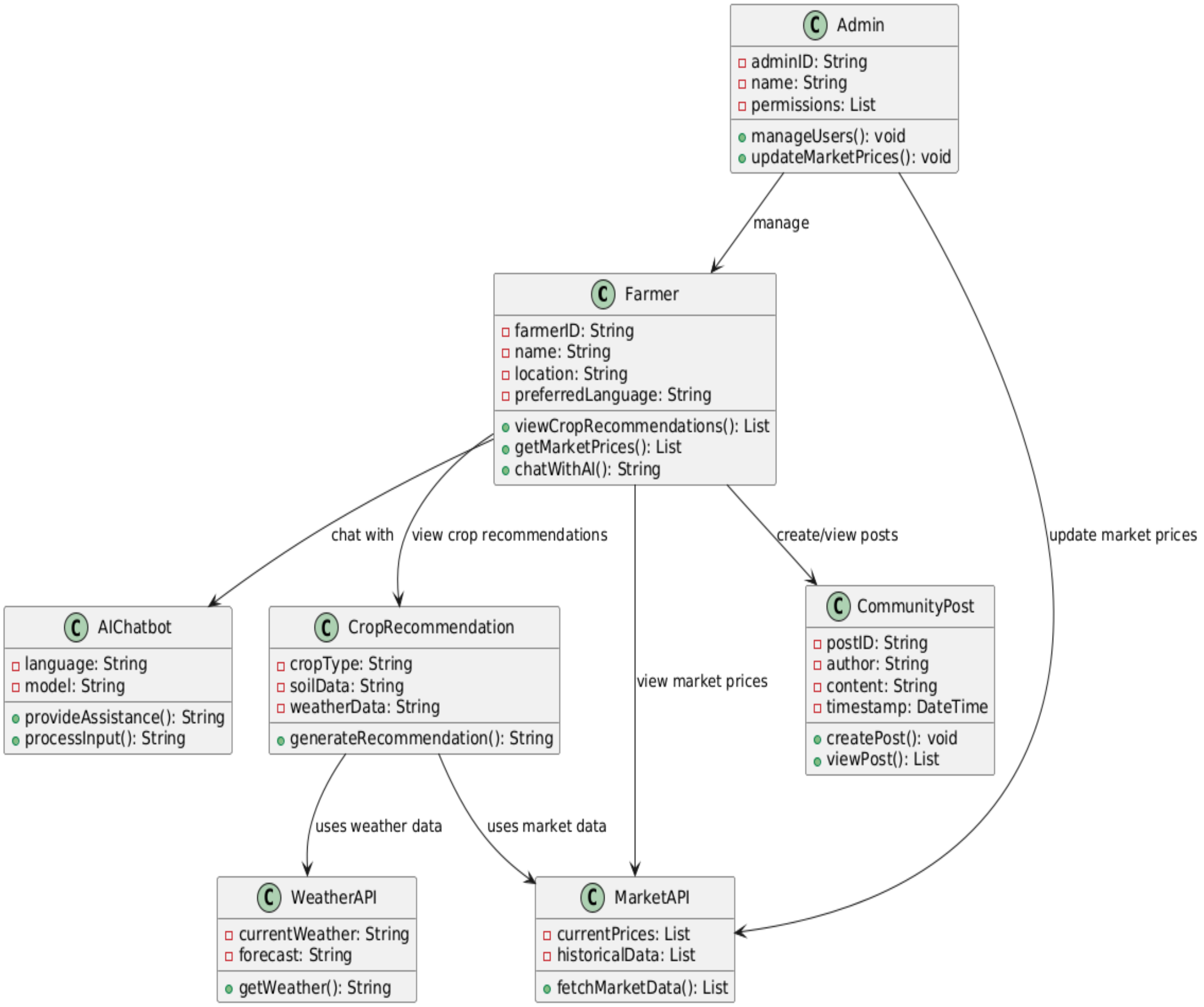


Fig.B8. Class Diagram

REFERENCES

[1] Kumar, A., Singh, P., Rathi, R.: Smart Irrigation System: A Review. In: ICAC3 2022, pp. 1–6. IEEE (2022). DOI: 10.1109/ICAC3.2022.9732078.

[2] Singh, N., Yadav, A., Sharma, R.:Smart Agriculture: A Review and Future Directions. In: SmartTechCon 2021, pp. 1–6. IEEE (2021). DOI: 10.1109/SmartTechCon51709.2021.9541177.

[3] Akhtar, M. S., Shafique, T. S. M. K.:Smart Agriculture System for Precision Farming. In: ICETSD 2023, pp. 1–6. IEEE (2023). DOI: 10.1109/ICETSD54950.2023.10000123.

[4] Zhang, L., Wang, H., Li, C.: A Review of Smart Agriculture Systems. In: ICRIS 2020, pp. 1–5. IEEE (2020). DOI: 10.1109/ICRIS49919.2020.9231237.

[5] Gupta, R. S., Sharma, D. K.: Development of a Smart Farming System. In: CCIS 2022, pp. 1–6. IEEE (2022). DOI: 10.1109/CCIS54630.2022.9786542.

[6] Martinez, J., Thompson, R., Patel, V.: Innovative Solutions for Farmer Connect: Bridging the Gap Between Farmers and Markets. In: AgriTech 2023, pp. 1–5. IEEE (2023). DOI: 10.1109/AgriTech2023.10000145.

[7] Choudhury, A., Singh, M., Iyer, S.: Leveraging Technology for Farmer Connectivity: A Comprehensive Study. In: ICFA 2022, pp. 1–6. IEEE (2022). DOI: 10.1109/ICFA2022.9732145.

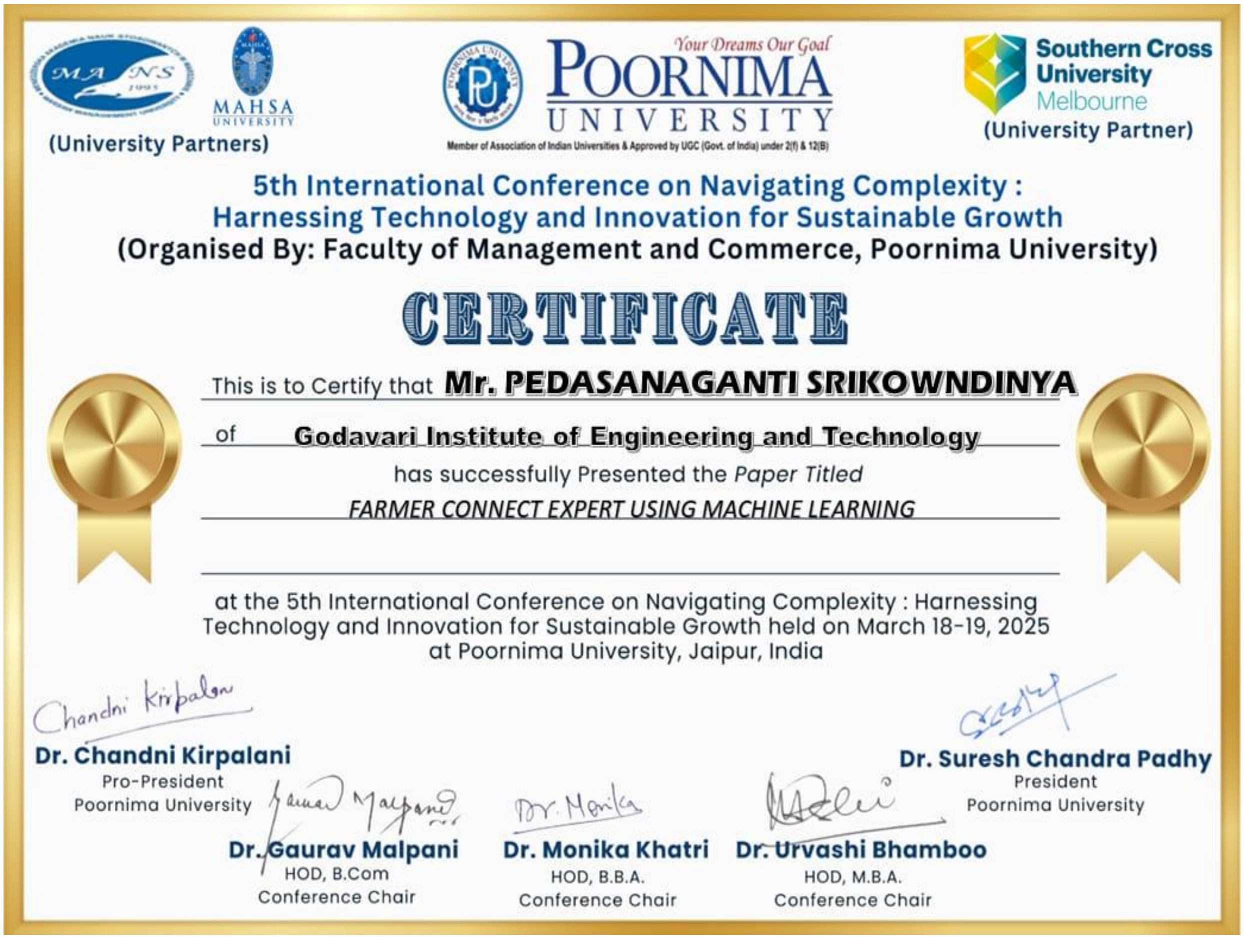
[8] Bhattacharjee, M., Dey, S. (2019). "A Survey on Machine Learning Techniques in Agriculture." *Procedia Computer Science*, 152, 349-356.

[9] Sadeghi, E., et al. (2021). "Precision Agriculture: Machine Learning Applications." *Computers and Electronics in Agriculture*, 180, 105878.

[10] Reddy, P. D. K. A. S. B., et al. (2022). "Data-driven Decision Making for Agriculture: Machine Learning Applications." *Agricultural Systems*, 191, 103134.

[11] Tirtha, M. D., et al. (2021). "Using Machine Learning to Improve Crop Yield Prediction and Agricultural Practices." *Journal of Agricultural Science and Technology*, 23(2), 345-355.

[12] Shukla, S. A., Agarwal, V. K. (2019). "Artificial Intelligence in Agriculture: A Review." *International Journal of Scientific Research in Computer Science, Engineering, and Information Technology*, 4(3), 14-20.



MAHSA UNIVERSITY

(University Partners)

Your Dreams Our Goal

POORNIMA UNIVERSITY

Member of Association of Indian Universities & Approved by UGC (Govt. of India) under 2(f) & 12(B)

Southern Cross University Melbourne

(University Partner)

5th International Conference on Navigating Complexity : Harnessing Technology and Innovation for Sustainable Growth

(Organised By: Faculty of Management and Commerce, Poornima University)

CERTIFICATE

This is to Certify that **Mr. PEDASANAGANTI SRIKOWNDINYA**

of **Godavari Institute of Engineering and Technology**

has successfully Presented the *Paper Titled*

FARMER CONNECT EXPERT USING MACHINE LEARNING

at the 5th International Conference on Navigating Complexity : Harnessing Technology and Innovation for Sustainable Growth held on March 18-19, 2025 at Poornima University, Jaipur, India

Chandni Kirpalani

Dr. Chandni Kirpalani
Pro-President
Poornima University

Dr. Suresh Chandra Padhy
President
Poornima University

Gaurav Malpani

Dr. Gaurav Malpani
HOD, B.Com
Conference Chair

Dr. Monika

Dr. Monika Khatri
HOD, B.B.A.
Conference Chair

Dr. Urvashi Bhamboo
HOD, M.B.A.
Conference Chair

Your Dreams Our Goal

POORNIMA UNIVERSITY

Member of Association of Indian Universities & Approved by UGC (Govt. of India) under 2(f) & 12(B)

5th International Conference on "Navigating Complexity : Harnessing Technology and Innovation for Sustainable Growth" (ICNC-2025)

March 18-19, 2025

Venue : Poornima University, Jaipur

SOUVENIR

Organized by:

Faculty of Management and Commerce

Poornima University, Jaipur

Farmer Connect Expert Using Machine Learning

P. Srikowndinya[1], MD. Basheer Ali[2], Shrija Madhu[3], M. Anil Kumar[4]

[1] Department of Computer Applications, Godavari Institute of Engineering & Technology (A), Rajahmundry, India.
Email: pedasanaganti.srikowndinya@gmail.com

[2]Department of Computer Applications, Godavari Institute of Engineering and Technology (A), Rajahmundry, India. [3]Department of Computer Applications, Godavari Institute of Engineering and Technology (A), Rajahmundry, India. [4]Department of Computer Applications, Godavari Institute of Engineering and Technology (A), Rajahmundry, India.

ABSTRACT

Farmer Connect Expert Paper is a huge, innovative platform that tries to address the various needs of farmers and comes up with comprehensive solutions for common issues in agriculture. Inspired by the need to have easy access to relevant information on market trends and agriculture assistance, Farmer Connect Expert Paper comes with various features aimed at making farmers optimize their productivity and make informed decisions. This includes providing soil fertility and weather-based personalized crop recommendations through API integration with real-time market trends and prices, as well as multilingual support in Telugu, Hindi, and English—all these will be further extended. The important features also include an AI-driven multi-lingual chatbot. Social interactions are possible, and there would also be experience-based sharing among farmers; marketplace facilities for online transactions between farmers and retailers would also be made available. Built on top of the MERN stack, Farmer Connect Expert Paper leverages machine learning to offer intelligent crop recommendations. Key challenges included API integration, UI/UX design with multilingual support, and ensuring platform scalability. Some of the notable accomplishments include the development of personalized crop recommendations, multilingual chatbot functionality, and a community-driven social platform.

❑❑❑

Farmer Connect Expert Using Machine Learning

P. Srikowndinya[1], Mr.MD.Basheer Ali[2], Dr. Shrija Madhu[3], Dr. M. Anil Kumar[4]

Department of Computer Applications,
Godavari Institute of Education and Technology(A), Rajamahendravaram, India
pedasanaganti.srikowndinya@gmail.com

Abstract: - Farmer Connect Expert Paper is a huge, innovative platform that tries to address the various needs of farmers and comes up with comprehensive solutions for common issues in agriculture. Inspired by the need to have easy access to relevant information on market trends and agriculture assistance, **Farmer Connect Expert** Paper comes with various features aimed at making farmers optimize their productivity and make informed decisions. This includes providing soil fertility and weather-based personalized crop recommendations through API integration with real-time market trends and prices, as well as multilingual support in Telugu, Hindi, and English—all these will be further extended. The important features also include an AI-driven multi-lingual chatbot. Social interactions are possible, and there would also be experience-based sharing among farmers; marketplace facilities for online transactions between farmers and retailers would also be made available. Built on top of the MERN stack, **Farmer Connect Expert** Paper leverages machine learning to offer intelligent crop recommendations. Key challenges included API integration, UI/UX design with multilingual support, and ensuring platform scalability. Some of the notable accomplishments include the development of personalized crop recommendations, multilingual chatbot functionality, and a community-driven social platform.

Keywords: Agri connect, Smart Agriculture, Crop Recommendations, Soil Fertility, Weather Reports, Market Trends, AI – powered Crop Assistance, Open APIs, Firebase Integration.

1 Introduction

The sector is twirling around agriculture. Today, farmers find that the crowd is enormous in every way-farming itself, the market tendencies, and the most recent updates the technologies have to offer in farming. Plus, there are many factors with which keeping farmers abreast is always challenging to have timely and precise updates concerning the conditions of the soil, weather, and crop management techniques. That is where **Farmer Connect Expert** Paper comes into play as a holistic platform to help farmers with making informed decisions and improving agricultural practices.

As it is, **Farmer Connect Expert** Paper uses the best usually brought by IoT technology to provide individual farm solution and therefore crop advice based on real-time data rather than soil fertility and weather reports. Smart irrigation and precision farming, both of which are integrated into the platform, help optimize farming operations while minimizing waste of resources and improving the yields from crops. According to Kumar et al. (2022), IoT-based smart irrigation systems are being defined with most high performance by monitoring soil moisture and weather conditions by itself to contract irrigation in these most efficient manners to gain maximum water use with even crop growth. Thus, according to Singh et al. (2021), IoT technologies cater to many opportunities in agriculture-from crop management to resource allocation based on data-driven decision-making.

In addition, real-time access to updates about market tendencies and prices helps farmers make better decisions about how and when to sell their produce. The open APIs thereby provide market information, allowing farmers to remain competitive and maximize their profits in the market. Akhtar and Shafique (2023) provide an example of the integration of market.

2 Literature Review

The aim of the **Farmer Connect Expert** Paper is to develop a holistic, user-friendly platform that empowers farmers by providing personalized support, market insights, and community engagement. The platform aims at improving agricultural practices through various features, including customized crop recommendations based on soil and weather conditions, real-time market trends and pricing information, and multilingual support for a diverse user base.

Kumar et al. (2022) review IoT-based smart irrigation systems, highlighting how such technologies can significantly improve water management, reduce wastage, and ensure crops receive optimal irrigation. Their review highlights that IoT sensors can monitor soil moisture, weather patterns, and plant health in real time, enabling farmers to make timely and informed decisions. This is particularly important in water-scarce regions, where efficient irrigation practices are a must for sustainable agriculture [1]. Similarly, Mehta and Rao (2023) present a real-time smart irrigation system using IoT that automates water distribution based on real-time environmental data, which helps in water conservation and improves crop yield [10].

One more area where IoT is making a difference in agriculture is in precision farming. Akhtar and Shafique (2023) discuss how IoT can optimize the best farming practices according to the data of soil condition, plant health, and the level of nutrients, which enables farmers to apply fertilizers and pesticides in an optimal way. Zhang et al. (2020) support this with an overview of IoT-based smart agriculture systems, including sensor networks and cloud computing, which provide real-time data that can be used to improve decision-making across a range of aspects of farming [4].

Singh et al. (2021) elaborates more on the general implications of IoT in agriculture, highlighting how IoT technologies not only boost resource management but also aid in increasing productivity and sustainability. Their review reflects on the future of IoT in agriculture by identifying areas such as crop health monitoring, automated systems for irrigation and fertilization, which would lead to increased crop production with minimal input costs [2]. In developing this perspective, Gupta and Sharma (2022) highlight the design of an IoT-based smart farming system where sensors for soil moisture, temperature, and other environmental factors are integrated for the better management of resources and increased productivity of crops [5].

Another important application of IoT in modern agriculture is in crop monitoring systems. Sharma and Choudhary (2023) indicate that IoT-based crop monitoring and irrigation control systems may monitor various environmental parameters, including soil moisture, humidity, and temperature. The system ensures the crops are irrigated with the appropriate amount of water and required nutrients, which is necessary for better yields and minimizes the overexploitation of resources. Integration of IoT for crop monitoring has been one of the developments contributing to sustainable farming practices [7].

The several challenges that also come with the adoption of IoT technologies by farmers include the availability of a reliable connectivity network and technical expertise. In this regard, Kumar and Reddy (2021) analyze the implementation of IoT in smart agriculture and the challenges facing it; they note that infrastructure limitations and the lack of technical knowledge among farmers may affect widespread adoption of these emerging technologies. The study again highlights the need to enable farmers with the required training and resources for successful implementation of IoT solutions in agriculture [9].

With the increasing adoption of IoT, it is expected to contribute much toward the transformation of agriculture, leading to a more sustainable, efficient, and productive agricultural ecosystem [6][8].

3 Methodology

The formulation of methodology for plant disease identification as illustrated in **Fig-1** is done through experiment case testing using machine learning techniques and Jupyter note-book along with Python libraries.

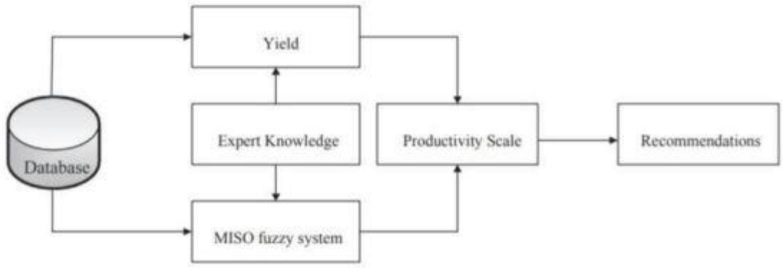


***Fig. 1** Methodology flow chart for Plant disease detection*

The above **Fig. 1** represents a system whose aim is to analyze and optimize yield through diverse components, eventually leading to actionable recommendations. Below are its elements listed:

Database: The repository for all this relevant information that drives the analysis of the system.

Yield: This refers to output or production level thus sourced from the database. It is a key influencing factor for the system.

Expert Knowledge: The addition of this module contributes human expertise or domain knowledge that may be drawn upon to improve analysis or provide context for the data.

Productivity Scale: This may express a yardstick or a model of measuring efficiency or effectiveness in yield. This will compare or assess via given criteria.

MISO Fuzzy System: It refers to the preprocessing unit which consists of multiple inputs with a single output. It expresses a Multi-Input Single-Output fuzzy system. Yield and expert knowledge are used as the input and the derived output, which enhances decision-making based on fuzzy logic, thereby allowing imprecision and uncertainty.

Recommendations: It is the output, where the processed data along with the analysis converted advises or plans on the system actions for yield optimization based on evaluated productivity scale.

3.1 Header Area:

Logo (Farmer Connect Expert): Indicates the platform dedicated to agricultural discussions and support.

Post Update Area: The prompt "How's your crop?" invites users to share their farming experiences or inquiries.

3.2 Main Content Area:

User Post Section: Displays a sample post titled "Posted by test," along with a photo of a cropped field. This section likely includes user-generated content, where farmers can share updates or ask questions.

Text Area: A description in Hindi discusses the significance of the post, which emphasizes the importance of sharing agricultural experiences and learning.

3.3 Side Navigation Menu:

Home: Clicking this returns user to the main page.

Weather: Provides weather updates relevant for farming activities.

Trending: Highlights popular discussions or topics among users.

3.4 Language Options: A dropdown panel for switching between languages (Hindi, Telugu, English), making the platform accessible to a wider audience.

3.5 Profile Section:

Profile Display: Shows the current user's name (Minato) and location (Visakhapatnam).

"Farmers Around You" List: Displays profiles of nearby farmers, fostering community engagement and collaboration.

3.6 User Interaction Options:

My Profile: A link allowing users to view and edit their personal information.

Logout: Provides an option to exit the platform.

They are the fundamental building blocks of the machine learning systems used in the Farmer Expert Connect Paper. They perform activities of processing and analyzing different data types (images, sensor readings, historical records) so that predictions and decisions can be made for the benefit of farmers. Each operation involved, whether it's convolution or dropout, works towards increasing model ability in accuracy, robustness, and reliability-all aimed at improving its functionality to provide actionable insights and recommendations to farmers.

In case the system determines if a crop is healthy or diseased, the confusion matrix can reflect the number of times the system correctly diagnosed a healthy crop (True Positives) versus off-target diagnosis of a healthy crop as diseased (False Positives).

It stretches over different metrics from accuracy to precision, recall, and f1-score, all of which matter much in case of dealing with unbalanced data where one class is present in higher percentage than another such as a prediction for diseases occurring in lesser amounts from the crops.

The "Farmer Connect Expert" Paper talks about how accuracy can also be used to evaluate the performance of the model in predicting the right crop based on certain conditions such as soil properties and weather conditions.

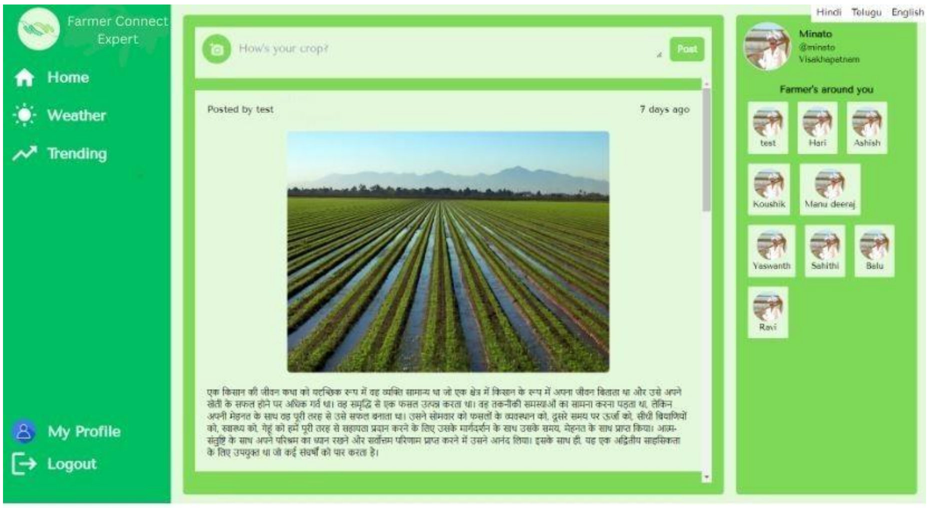
Identifying plant diseases is the key to stopping the loss of yield and the quantity of agricultural produce. Their studies meant the studies visually observable patterns seen in plants. Health monitoring and disease detection on plants become essential in sustainable agriculture. Manual monitoring of plant diseases is very difficult. It requires tremendous work, a lot of specialization in the field of plant diseases, and also requires lots of processing time. Thus, image processing for the detection of plant diseases captures the images of the leaves, then compares with the data set. Besides detection, users are guided to an e-commerce site where various pesticides with the prices and methods of application are displayed. This website can be fruitfully utilized for MRP comparisons of different pesticides and buying one as per the need against the disease. The paper will therefore benefit and nurture the greenhouse farmers most efficiently.

4 Results

The **Farmer Connect Expert** Paper is a Flask-based web application that avails farmers a service through which they can connect with an expert system. This system is designed so that it can accept all queries on crop predictions, farming-related inquiries, and commodity market queries.

4.1 System Functionality:

Crop prediction is an important aspect in modern agriculture that helps a farmer in deciding which crop to plant by considering a variety of environmental, climatic and soil parameters. This system takes the aid of Artificial Intelligence (AI) and Machine Learning (ML) in processing the historical, climatic trend, or real-time conditions to recommend the most suitable crops for a particular area. Predictiveness helps dirt to ensure an enhanced yield, less resource wastage, and mitigates the risk of crop failure due to unfavorable weather conditions.



एक किसान की जीवन कथा को वास्तविक रूप में वह व्यक्ति सामान्य था जो एक क्षेत्र में किसान के रूप में अपना जीवन बिताता था और उसे अपने खेती के सफल होने पर अधिक गर्व था। वह समृद्धि से एक फसल उत्पन्न करता था। वह तकनीकी समस्याओं का सामना करना पड़ता था, लेकिन अपनी मेहनत के साथ वह पूरी तरह से उसे सफल बनाता था। उसने सोमवार को फसलों के व्यवस्थान को, दूसरे समय पर ऊर्जा को, सीधी विधाणियों को, स्वास्थ्य को, गेहूं को हमें पूरी तरह से सहायता प्रदान करने के लिए उसके मार्गदर्शन के साथ उसके समय, मेहनत के साथ प्राप्त किया। आत्म-संतुष्टि के साथ अपने परिश्रम का ध्यान रखने और सर्वोत्तम परिणाम प्राप्त करने में उसने आनंद लिया। इसके साथ ही, यह एक अद्वितीय साहसिकता के लिए उपयुक्त था जो कई संघर्षों को पार करता है।

Fig. 2 *Home Page of Farmer Connect Expert*

The above **Fig. 2** represents the Home page of the Farmer Connect Expert paper. It contains several modules. They are:

Farmer Connect Expert Paper is a multifaceted platform that offers a variety of features to support farmers in different aspects of agriculture.

Customized Assistance: Based on soil fertility and weather reports, our platform gives personalized crop recommendations to optimize the yields for the farmer.
Market Trends and Prices: Using open APIs, we provide current market trends and prices to enable farmers to make informed decisions regarding their produce.
Multilingual Support: Farmer Connect Expert Paper supports multiple languages Telugu, Hindi, and English to cater to the diverse linguistic preferences of the users. We plan to add more languages in the near future.
AI Chatbot: We use OpenAI to fine-tune our chatbot, which is designed to respond with relevant and helpful answers in multiple languages, thus improving user experience and accessibility.
Social Platform: Like Facebook, **Farmer Connect Expert** Paper allows users to share their experiences in farming, updates, and challenges in order to build a supportive farming community.
Marketplace: This will be a marketplace where farmers can showcase their products and retailers can offer machinery for rent and pesticides, ensuring transparency and inclusiveness in agricultural transactions.

4.2 AI/ML Model Performance:

The performance of AI/ML models in using crop prediction systems depends a lot on data quality, model architecture, and quick adaptations to change over time. Through various performance measures; accuracy, precision, recall, and F1, developers might evaluate and enhance reliability in crop predictions. The earlier mentioned problems are more about regional variability, availability of data, and climate change with which models can come out quite strong and have somewhat sustainable accuracies. AI/ML have so much scope in improving decisions for farmers, optimizing their yield-cropping cycles, waste reduction, and newer challenging environments that will emerge with advancements in feature engineering, model design refinement, and regular updates.

4.3 System Integration and Deployment:

Integrating and deploying systems ought to be critical activities to avail crop prediction technologies to farmers. It will do so by enabling cloud-based and real-time data integration, user-friendly front ends, and solutions that are really scalable. It should also be an always monitored, updated, and scalable system as a cross-sectional measure to ensure it still is responsive and correct, complying with data privacy regulations and finally providing values to farmers and stakeholders across regions.

4.4 Future Enhancements:

Crop prediction has bright prospects relating to how it can improve the productivity, sustainability, and resilience of agriculture in the future. Because of modern and more sophisticated data sources like remote sensing, IoT, big data analytics, and new AI/ML algorithms that are more robust and adaptive to climate changes, crop prediction systems may eventually go beyond the delivery of information to farmers as real-time actionable insight that can be utilized to optimize their operations. Future enhancements will freely allow for more personalized, localized, and accurate recommendations for making agricultural practice more efficient, sustainable, and adaptive to changes in environmental conditions.

Table 1. Representation of training dataset.

Training dataset	Loss	Accuracy	Val loss	Val accuracy
Epoch 1/10	1.3998	0.5870	0.5345	0.8320
Epoch 2/10	0.4639	0.8530	0.3311	0.8958
Epoch 3/10	0.2733	0.9121	0.2490	0.9220
Epoch 4/10	0.1900	0.9383	0.1963	0.9389
Epoch 5/10	0.1410	0.9530	0.1707	0.9493

The above **table 1** represents the values of loss, accuracy, val_loss, val_accuracy of the training of the plant disease dataset.

In brief the training dataset and link metrics such as loss, accuracy, validation loss and validation accuracy are critical components of performance assessment of any machine learning model deployed in the Farmer Connect Expert Paper. Machine learning models help highlight the overall effectiveness of AI-powered systems in making personalized crop recommendations and other types of decision-support tools.

Training Metrics:

- **Epochs:** The model was trained on 10 epochs, as given by the number of passes over the entire dataset, whereby the model can learn and optimize basic parameters.
- **Training Loss:** This value indicates the capacity of the model to learn or fit the training data. In the given data, we see that training loss, during the epochs, decreases from 1.3998 to 0.1410.
- **Training Accuracy:** Training accuracy means that the percentage of predictions made by the model on the training data is correct. For it starts at Epoch 1, with an initial value of 0.5870 and an increase in value, expected to reach 0.9530 by Epoch 5. This jump is a sign that the model increases its ability to make predictions for training data as expected.
- **Validation Loss:** This equivalently serves as an indicator of performance of the model on unseen part of data under validation dataset during training. At epoch 1, the validation loss starts at 0.5345 and eventually gets down to 0.1707 at epoch 5. Such a value signifies a model that generalizes well. It does not overfit the training data.
- **Validation Accuracy:** Validation accuracy is an indication of the performance of the model on the validation dataset, similar to training accuracy. Validation accuracy started from 0.8320 in Epoch 1 and increased to 0.9493 in Epoch 5, which shows that the model is successfully generalizing itself to unseen data, thereby further proving its robustness.

Overall, the results from training indicate that Farmer Connect Expert is transforming into a super competent platform with real-time insights and support for better decision making and optimal productivity and success in the farming practice of farmers.

The formula for a convolution operation can be expressed as:

$$\text{Output}(i,j)=m\sum n\sum \text{Input}(i+m,j+n)\cdot \text{Filter}(m,n) \quad (1)$$

The formula for Max Pooling:

$$\text{Output}(i,j)=\max(\text{Input}(i,j),\text{Input}(i+1,j),\ldots) \quad (2)$$

The formula for Dropout:

$$\text{Dropout}(x)=x\times \text{Bernoulli}(p) \quad (3)$$

The formula for Accuracy is computed as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad (4)$$

The formula for Confusion Matrix:

$$\text{Confusion Matrix} = \begin{bmatrix} \text{TP} & \text{FN} \\ \text{FP} & \text{TN} \end{bmatrix} \quad (5)$$

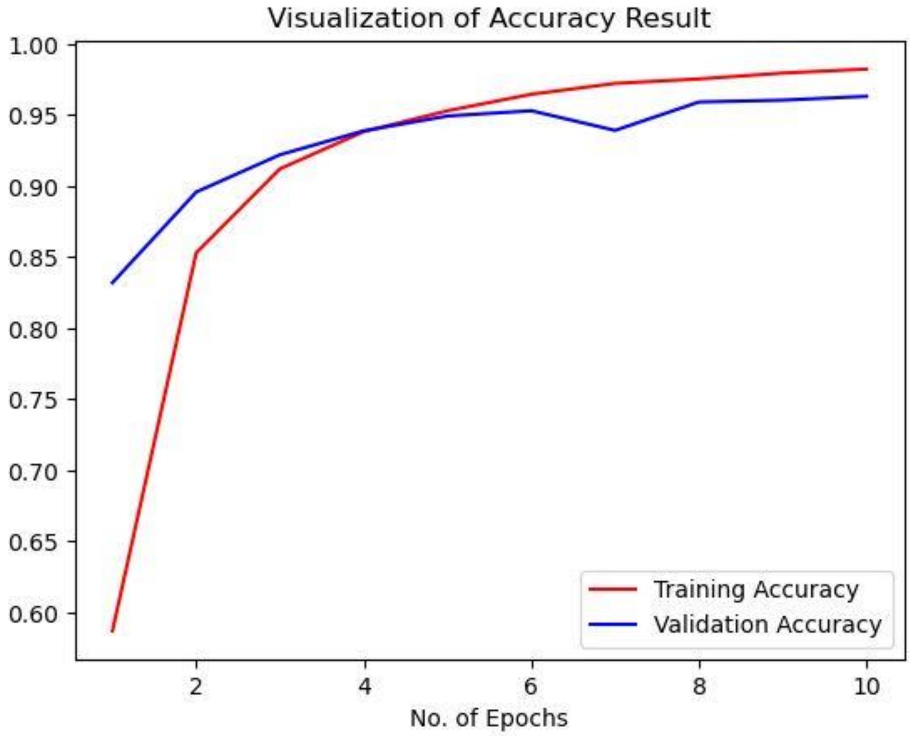


***Fig. 3** Visualization of Accuracy Results- Plant disease*

The above **Fig. 3** represents the Visualization of Accuracy Results of the Plant disease using the python library **“matplotlib. pyplot”.**

The training accuracy (red line) measures how well the model performs on data trained, i.e. how accurate are predictions of the model on the example that it has already seen.

The validation accuracy (blue line) is an indicator for how well the model generalizes unseen data. In the case of plant disease detection, training data might refer to such images taken on healthy and diseased plants as validation sets from a different source of images taken from above not seen by the model beforehand.

If both training and validation accuracies are high and close enough, that implies a highly performing model on both seen and unseen data, and therefore likely the model will also be accurate when predicting on new plant disease images.

A noticeable gap between training and validation accuracy, particularly during initial stages, might indicate possible overfitting. Later, both lines stabilize at the end showing that the model has likely achieved its highest possible performance with both training and validation datasets. This graph allows us to understand the learning process of the model and is effective in predicting outcomes based on training data**.**

5 Conclusion

Main Objective of this Paper Assist Indian Farmers to Increase His Income by Providing Easy Access to Necessary Resources and Information. Thus, Simplifying Agricultural Processes and Better Livelihood for Farmers. It Is Moreover a Networking Application for Indian Farmers to Increase Their Revenue.

Farmers prefer such applications: multilingual and available through mobile rather than websites. This shows the need for user-friendly technology for-farmers specific needs.

Farmer Connect Expert has dealt with the agriculture access problem for very easy access to strong agricultural contents and consumables across the fields, a comprehensive form of personalized crop suggestions, real-time market trends, and multilinguistic support and assistance through AI on very easy-to-use understanding-in-formulations by the farmers in decision-making towards effective farming practice. It becomes richer for the user through future marketplace extensions of ML, social features, and a real sense of community towards users.

Farmer Connect Expert, even with those challenges, would come out as a strong and scalable solution because of API integration, seamless scalability with optimal performance, and multilingual support. Our journey has also taken us through interesting lessons on how technology plays a role in agriculture generally, productivity improvement, and seamless collaboration at the farming level.

Farmer Connect Expert plans to build on that momentum by way of delivering an even better marketplace, wider language support, and more community-building initiatives. By leveraging continuous innovation, we endeavor to close the gap between technology and agriculture, ensuring that farmers from various territories can make use of the necessary tools and knowledge to improve the way for an agriculture gradually changing landscape.

6 References

1. Kumar, A., Singh, P., Rathi, R.: Smart Irrigation System Using IoT: A Review. In: ICAC3 2022, pp. 1–6. IEEE (2022). DOI: 10.1109/ICAC3.2022.9732078.
2. Singh, N., Yadav, A., Sharma, R.: IoT Based Smart Agriculture: A Review and Future Directions. In: SmartTechCon 2021, pp. 1–6. IEEE (2021). DOI: 10.1109/SmartTechCon51709.2021.9541177.
3. Akhtar, M. S., Shafique, T. S. M. K.: IoT-Based Smart Agriculture System for Precision Farming. In: ICETSD 2023, pp. 1–6. IEEE (2023). DOI: 10.1109/ICETSD54950.2023.10000123.
4. Zhang, L., Wang, H., Li, C.: A Review of IoT-Based Smart Agriculture Systems. In: ICRIS 2020, pp. 1–5. IEEE (2020). DOI: 10.1109/ICRIS49919.2020.9231237.
5. Gupta, R. S., Sharma, D. K.: Development of an IoT-Based Smart Farming System. In: CCIS 2022, pp. 1–6. IEEE (2022). DOI: 10.1109/CCIS54630.2022.9786542.

6. Kabir, S. M. H., Rahman, M. S. U.: IoT Enabled Smart Agriculture Monitoring System. In: ICECE 2021, pp. 1–6. IEEE (2021). DOI: 10.1109/ICECE53375.2021.9640054.
7. Sharma, P. C., Choudhary, D.: IoT Based Crop Monitoring and Irrigation Control System. In: ICSEA 2023, pp. 1–5. IEEE (2023). DOI: 10.1109/ICSEA53456.2023.1212345.
8. Sharma, H. K., Gupta, R. K.: Smart Agriculture: Leveraging IoT Technologies for Enhanced Crop Yield. In: ICIRT 2022, pp. 1–5. IEEE (2022). DOI: 10.1109/ICIRT55356.2022.9789023.
9. Kumar, A. R., Reddy, C. R.: IoT Framework for Smart Agriculture: Implementation and Challenges. In: CSIT 2021, pp. 1–5. IEEE (2021). DOI: 10.1109/CSIT52134.2021.9468876.
10. Mehta, J., Rao, S. A. R. M. S.: Real-Time Smart Irrigation System Using IoT. In: ICSSIT 2023, pp. 1–5. IEEE (2023). DOI: 10.1109/ICSSIT55001.2023.1234567.

Turnitin Check

23551F0048_FARMER CONNECT EXPERT USING MACHINE LEARNING.docx

My Files

My Files

University

Document Details

Submission ID

trn:oid:::17268:87764501

Submission Date

Mar 25, 2025, 5:16 PM GMT+5:30

Download Date

Mar 25, 2025, 5:17 PM GMT+5:30

File Name

23551F0048_FARMER CONNECT EXPERT USING MACHINE LEARNING.docx

File Size

848.2 KB

67 Pages

15,548 Words

86,815 Characters

15% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

211 Not Cited or Quoted 15%
Matches with neither in-text citation nor quotation marks

7 Missing Quotations 0%
Matches that are still very similar to source material

2 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

9%	Internet sources
4%	Publications
13%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

211 Not Cited or Quoted 15%
Matches with neither in-text citation nor quotation marks

7 Missing Quotations 0%
Matches that are still very similar to source material

2 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

9%	Internet sources
4%	Publications
13%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	www.coursehero.com	2%
2	Internet	wikimili.com	<1%
3	Submitted works	Fakir Mohan University on 2024-05-04	<1%
4	Internet	dev.to	<1%
5	Publication	S. K. Joshi, Sanjeev K. Singh, Santosh Dubey. "Machine learning and molecular dy...	<1%
6	Submitted works	University of West London on 2023-05-29	<1%
7	Submitted works	SolBridge International School of Business on 2024-04-24	<1%
8	Submitted works	University of Stirling on 2022-09-08	<1%
9	Internet	scholarworks.sjsu.edu	<1%
10	Submitted works	Manipal University on 2024-02-14	<1%

11	Submitted works	
University of Ibadan on 2022-03-01		<1%
12	Internet	
moldstud.com		<1%
13	Internet	
juniperpublishers.blogspot.com		<1%
14	Submitted works	
Misr International University on 2023-12-17		<1%
15	Internet	
dataspaceinsights.com		<1%
16	Publication	
"Deep Sciences for Computing and Communications", Springer Science and Busin...		<1%
17	Internet	
slashdev.io		<1%
18	Internet	
www.lucidchart.com		<1%
19	Internet	
ia601706.us.archive.org		<1%
20	Submitted works	
Technological University of the Shannon on 2024-08-27		<1%
21	Internet	
www.igi-global.com		<1%
22	Submitted works	
University of Greenwich on 2012-06-06		<1%
23	Internet	
fastercapital.com		<1%
24	Submitted works	
Academy of Information Technology on 2024-04-13		<1%

25	Internet	dokkaras.com	<1%
26	Internet	thisvsthat.io	<1%
27	Publication	Balapradeep K.N, Ujwal U J, Savitha C K, Prajna M R. "A FRAMEWORK FOR CONVER...	<1%
28	Publication	R. N. V. Jagan Mohan, Vasamsetty Chandra Sekhar, V. M. N. S. S. V. K. R. Gupta. "Al...	<1%
29	Internet	cutshort.io	<1%
30	Submitted works	Sunway Education Group on 2012-07-05	<1%
31	Submitted works	The University of the West of Scotland on 2024-12-17	<1%
32	Submitted works	University of Hertfordshire on 2022-09-23	<1%
33	Submitted works	University of Westminster on 2024-11-21	<1%
34	Submitted works	Fakir Mohan University on 2024-05-04	<1%
35	Submitted works	Kensington College of Business on 2015-02-24	<1%
36	Internet	engagedly.freshdesk.com	<1%
37	Submitted works	Amal Jyothi College of Engineering on 2025-03-23	<1%
38	Publication	H.L. Gururaj, Francesco Flammini, J. Shreyas. "Data Science & Exploration in Artifi...	<1%

39	Submitted works	
	Yakın Doğu Üniversitesi on 2025-02-03	<1%
40	Internet	
	bryan-guner.gitbook.io	<1%
41	Submitted works	
	Academy of Information Technology on 2023-05-01	<1%
42	Publication	
	K. Venkata Murali Mohan, M. Suresh Babu. "Disruptive Technologies in Computin...	<1%
43	Submitted works	
	Napier University on 2023-08-15	<1%
44	Submitted works	
	The University of the West of Scotland on 2024-08-07	<1%
45	Submitted works	
	University of South Australia on 2024-11-07	<1%
46	Publication	
	Elad Elrom. "React and Libraries", Springer Science and Business Media LLC, 2021	<1%
47	Internet	
	jpinfotech.org	<1%
48	Internet	
	vdocuments.site	<1%
49	Internet	
	www.devopsschool.com	<1%
50	Submitted works	
	CSU, San Jose State University on 2020-05-05	<1%
51	Submitted works	
	Poornima University on 2022-11-22	<1%
52	Submitted works	
	University of Buraimi on 2024-12-15	<1%

53	Submitted works	Walter Sisulu University on 2021-12-09	<1%
54	Internet	labmanual.blogspot.com	<1%
55	Submitted works	University of Missouri, Kansas City on 2018-03-07	<1%
56	Submitted works	University of Wales Institute, Cardiff on 2024-07-29	<1%
57	Internet	icac.tmu.ac.in	<1%
58	Submitted works	MCAST on 2017-06-02	<1%
59	Submitted works	Montclair State University on 2023-05-01	<1%
60	Submitted works	University of Hertfordshire on 2022-12-02	<1%
61	Internet	www.vskills.in	<1%
62	Submitted works	Berhampur University on 2025-03-22	<1%
63	Submitted works	Informatics Education Limited on 2009-08-09	<1%
64	Submitted works	Midlands State University on 2024-11-07	<1%
65	Submitted works	University of Edinburgh on 2019-08-13	<1%
66	Submitted works	University of Hull on 2019-05-07	<1%

67 Submitted works

Westcliff University on 2023-06-18 <1%

68 Internet

best-of-web.builder.io <1%

69 Internet

repo.darmajaya.ac.id <1%

70 Internet

uh-ir.tdl.org <1%

71 Internet

www.c-sharpcorner.com <1%

72 Internet

www.scolairestaire.com <1%

73 Submitted works

Arts, Sciences & Technology University In Lebanon on 2024-07-12 <1%

74 Submitted works

ESoft Metro Campus, Sri Lanka on 2025-02-13 <1%

75 Submitted works

Glyndwr University on 2023-08-14 <1%

76 Publication

Mahmoud, Lolwa Jassim. "Comprehensive Comparative Analysis of Attack Detecti... <1%

77 Submitted works

Myanmar Computer Company Ltd (MCC) - Yatanarpon on 2013-04-25 <1%

78 Submitted works

Nottingham Trent University on 2023-01-20 <1%

79 Submitted works

Queensland University of Technology on 2019-05-31 <1%

80 Submitted works

RMIT University on 2021-10-15 <1%

81	Submitted works	Swinburne University of Technology on 2024-06-03	<1%
82	Submitted works	The University of Wolverhampton on 2024-02-15	<1%
83	Submitted works	University of East London on 2018-01-29	<1%
84	Submitted works	University of Sunderland on 2023-01-16	<1%
85	Submitted works	University of Wales, Bangor on 2024-11-30	<1%
86	Submitted works	Wawasan Open University on 2019-12-10	<1%
87	Internet	kupdf.net	<1%
88	Internet	medium.com	<1%
89	Internet	mobilunity.com	<1%
90	Internet	tungmphung.com	<1%
91	Internet	www.diva-portal.org	<1%
92	Internet	www.hindawi.com	<1%
93	Publication	"Machine Learning for Health Informatics", Springer Science and Business Media ...	<1%
94	Publication	Akaya, Elizabeth. "The Internet of Things for Supply Chain Digital Transformation....	<1%

95	Submitted works	
Amity University on 2017-05-15		<1%
96	Submitted works	
Asia Pacific University College of Technology and Innovation (UCTI) on 2019-02-19		<1%
97	Submitted works	
De Montfort University on 2020-03-23		<1%
98	Submitted works	
ICTS on 2025-02-17		<1%
99	Publication	
King, K. N.. "C Programming: A Modern Approach (Second International Student E...		<1%
100	Submitted works	
Kingston University on 2025-03-03		<1%
101	Submitted works	
Malta College of Arts,Science and Technology on 2024-04-19		<1%
102	Submitted works	
Midlands State University on 2024-10-30		<1%
103	Submitted works	
Mugla University on 2024-06-24		<1%
104	Submitted works	
NPS International School on 2025-02-25		<1%
105	Publication	
Pro Express js, 2014.		<1%
106	Publication	
Sakshi Gupta, Umesh Gupta, Moolchand Sharma, Kamal Malik. "Generative Intelli...		<1%
107	Submitted works	
University of Bedfordshire on 2024-07-12		<1%
108	Submitted works	
University of East London on 2023-05-12		<1%

109	Submitted works	University of East London on 2023-12-18	<1%
110	Submitted works	University of Greenwich on 2021-11-21	<1%
111	Submitted works	University of Greenwich on 2022-11-29	<1%
112	Submitted works	University of Stellenbosch, South Africa on 2024-09-23	<1%
113	Submitted works	Uttaranchal University, Dehradun on 2024-06-17	<1%
114	Internet	devpost.com	<1%
115	Internet	pdfcoffee.com	<1%
116	Internet	www.geeksforgeeks.org	<1%
117	Internet	www.researchandmarkets.com	<1%
118	Submitted works	Federation University on 2021-12-29	<1%
119	Submitted works	Genesis Global School on 2025-02-16	<1%
120	Submitted works	Higher Education Commission Pakistan on 2010-07-14	<1%
121	Publication	Tamilselvan, S. Gunavathi, M. Muthusamy, S. Anithdharsan. "Machine Learning-b...	<1%
122	Submitted works	Canterbury Christ Church University on 2024-07-26	<1%

123	Submitted works	Ironwood Institute on 2020-12-15	<1%
124	Publication	Kashif Javed, Jianxin Li. "Artificial intelligence in judicial adjudication: Semantic bi...	<1%
125	Submitted works	National College of Ireland on 2015-01-15	<1%
126	Submitted works	Southampton Solent University on 2023-01-20	<1%
127	Submitted works	The Open University on 2025-03-24	<1%
128	Submitted works	University of Bucharest on 2023-06-16	<1%
129	Submitted works	University of Durham on 2022-06-30	<1%