CROP RECOMMENDATION SYSTEM USING ML

A PROJECT REPORT

Submitted by

Student Name	Student Code	Roll Number	Registration Number
SUNITI GHORAI	BWU/MCA/22/144	22010201131	22012006273
AROHAN GUPTA	BWU/MCA/22/137	22010201125	22012006267
ARVIND KUMAR	BWU/MCA/22/135	22010201123	22012006265

in partial fulfillment for the award of the degree

of

MASTER OF COMPUTER APPLICATION

Department of C.S.S.

BRAINWARE UNIVERSITY
398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700 125

July, 2024

BRAINWARE UNIVERSITY

398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700 125

DEPARTMENT OF C.S.S.

BONAFIDE CERTIFICATE

Certified that this project report **Crop Recommendation using ML** is the bonafide work of **AROHAN GUPTA, SUNITI GHORAI** and **ARVIND KUMAR** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. JAYANTA AICH	**PIYALI MAUR**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
C.S.S. Department	Assistant Professor of C.S.S. Dept.
Brainware University	Brainware University

External Examiner

ABSTRACT

Abstract — Agriculture is the world's leading source of food items. All the food substances that are essential viz. Agriculture produces vegetables, proteins, and oils. The carbohydrates provide all living beings with energy. Farmers have great importance in our society. They are the ones who provide us food. Presence of Too Many Intermediates/Middlemen results in the exploitation of both farmers and consumers with the middlemen offering lower prices to farmers and charging higher prices from the consumers. The portal is envisaged to make available relevant information and services to the farming community and private sector through the use of information and communication technologies, to supplement the existing delivery channels provided for by the department. The farmers can gain more profit using this portal and by connecting directly with the customers. Avoiding interference of a middleman. The information obtained can help farmers identify efficiencies that lead to higher productivity and profitability, lower input costs, and optimized fertilizer use. The more a farmer knows about his or her farm, the better their opportunities to strengthen supply chain relationships. This portal will help farmers to get a clear idea about customer requirements and it will also provide information about how to grow required crop and what it will cost. The max-prior algorithm used helps in allocating the highest requirement customer to the farmers to gain better profit. It also helps the farmers in selling their produce quicker. Thus, by this portal the farmers gain more profit hence increasing the country's economy.

TABLE OF CONTENTS

CHAPTERS

✓ Chapter 1: INTRODUCTION

Husbandry significantly contributes to a country's GDP, with numerous agricultural revolutions enhancing productivity worldwide, especially in India. Agriculture affects biodiversity through various agrarian activities. A proposed web application aims to assist farmers in selling products at better rates, ensuring direct marketing to consumers, which maximizes profits and offers affordable prices. Farmers can also access weather information and agricultural knowledge, helping them stay updated with current technologies and trends, which is crucial for economic growth. For instance, the crop recommender system using machine learning can significantly enhance decision-making for farmers[1]. Another system provides agricultural recommendations for crop protection[2]. These systems ensure a direct communication line between farmers and consumers, reducing intermediaries and enhancing awareness of modern practices[3-11].

The design of such applications helps farmers understand the best crop choices and practices based on neural networks[3] and pest control techniques[4]. AI-based crop recommendation systems further aid in maximizing efficiency[5]. By integrating soil and climatic parameters, farmers can make more informed decisions[6]. Machine learning and AI play a crucial role in these systems, supporting smarter agricultural practices[7-8]. Additionally, IoT-based systems provide long-term benefits by leveraging weight-based memory approaches[9]. The k-nearest neighbors algorithm also offers practical crop recommendations[10], and cloud computing enhances these systems' scalability and accessibility[11].

The paper outlines the exploration gaps (Section II), the proposed web application and its algorithm (Section III), a comparison of traditional and agile farm operations (Section IV), and the conclusion with future improvements in the final section.

Keywords - Agriculture, Farming, Crop, Agriculture Analytics, Agricultural Technology, Crop Prediction

✓ Chapter 2: LITERATURE REVIEW

The existing literature highlights the significant role of agriculture in global food production and its impact on a nation's GDP. Various revolutions, such as the green, yellow, and blue revolutions, have advanced agricultural practices and productivity. However, the presence of intermediaries often results in the exploitation of farmers, who receive lower prices for their produce, while consumers face higher costs. To address this issue, several studies and projects have proposed the use of information and communication technologies (ICT) to create direct connections between farmers and consumers, thereby eliminating middlemen and enhancing profitability for farmers.

Previous models of e-trading systems have demonstrated the potential benefits of such platforms, including better price realization for farmers and fresher products for consumers. These systems have also provided farmers with access to valuable information on crop management, market trends, and weather forecasts, which are crucial for optimizing agricultural practices and increasing productivity. Additionally, algorithms like the max-prior algorithm have been developed to match farmers with customers having the highest demand, further maximizing profits and ensuring quicker sales.

This literature review underscores the necessity of an integrated digital platform that can provide comprehensive support to farmers, enhance supply chain efficiency, and contribute to the overall growth of the agricultural sector and the national economy.

✓ Chapter 3: THEORY, METHODOLOGY, MATERIALS & METHODS

❖ Theory

The theoretical foundation of this project is based on the economic principle of disintermediation, which involves removing intermediaries from supply chains to directly connect producers with consumers. In agriculture, this approach can lead to significant benefits, including better prices for farmers, fresher products for consumers, and overall improved efficiency in the supply chain. Additionally, the use of information and communication technologies (ICT) is theorized to enhance the accessibility and dissemination of critical agricultural information, leading to better-informed decision-making by farmers.

❖ Methodology

The project employs a structured methodology to design, develop, and implement the Agriculture Portal. The key steps in the methodology are:

1. Requirements Gathering: Collecting detailed requirements from stakeholders, including farmers, consumers, and agricultural experts. This involves interviews, surveys, and focus group discussions.

2. System Design: Creating a comprehensive design for the portal, including user interface design, database schema, and system architecture. This step also involves selecting appropriate technologies and tools for development.

3. Algorithm Development: Designing and implementing the max-prior algorithm to prioritize customer requirements and optimize profit for farmers. This algorithm will be based on various parameters such as demand, pricing, and product availability.

4. Development and Integration: Building the portal using the selected technology stack. This includes coding the front-end, back-end, and integrating the database. Additionally, integrating external services like weather reports and chatbots for real-time assistance.

5. Testing: Conducting rigorous testing of the portal to ensure functionality, performance, and security. This involves unit testing, integration testing, and user acceptance testing (UAT).

6. Deployment: Deploying the portal in a live environment and ensuring it is accessible to all intended users. This step also involves setting up maintenance and support processes.

7. Evaluation and Feedback: Gathering feedback from users and evaluating the performance of the portal. This feedback will be used to make necessary improvements and updates.

❖ Materials

The materials used in this project include:
1. Software Tools:
   - Integrated Development Environment (IDE) for coding (e.g., Visual Studio Code)

- Database management systems (e.g., MySQL, PostgreSQL)
- Web development frameworks (e.g., React, Angular, Node.js)
- Cloud services for deployment (e.g., AWS, Azure)

2. Hardware:
   - Development and testing machines
   - Servers for hosting the web application

3. Data Sources:
   - Agricultural data from government and private databases
   - Weather information from meteorological services
   - Market price data for various crops

4. Documentation and Reference Materials:
   - Technical manuals and API documentation
   - Research papers on agricultural economics and ICT in agriculture

- **Methods**

The methods applied in this project are:

1. Data Collection: Gathering relevant data from various sources such as government portals, agricultural databases, and weather services. This data is essential for providing accurate information to farmers.

2. Algorithm Implementation: Developing the max-prior algorithm to match farmers with the highest requirement customers. This involves programming, testing, and refining the algorithm to ensure optimal performance.

3. Web Development: Using modern web development practices to create a user-friendly and responsive portal. This includes front-end development for the user interface and back-end development for server-side processing.

4. Testing and Validation: Applying various testing methods to ensure the portal meets all functional and non-functional requirements. This includes automated testing tools as well as manual testing by end users.

5. User Training and Support: Providing training sessions for farmers to help them effectively use the portal. Setting up a support system to assist users with any issues they encounter.

By following this structured approach, the project aims to create a robust Agriculture Portal that empowers farmers, enhances productivity, and contributes to the overall economy.
This section should provide a clear and detailed overview of the theoretical background, methodology, materials, and methods used in your Agriculture Portal project documentation.

- **Dataset & Algorithms Used:**

A. CROP PREDICTION: -

Dataset Description

The dataset, `preprocessed2.csv`, appears to be related to agricultural data, specifically focusing on crop cultivation. The dataset includes the following columns:

1. State_Name : The name of the state where the crop is grown.
2. District_Name : The name of the district within the state.
3. Season : The season during which the crop is cultivated.
4. Crop : The type of crop grown.

Data Preparation
1. Loading Data : The data is read into a pandas DataFrame.
2. Data Cleaning :
   - Trailing whitespace in the `Season` column is removed.
   - An unnamed column (possibly an index) is deleted.
3. Data Splitting : The data is split into training and testing datasets. Rows from 100 to 120 are used for testing.

Machine Learning Model
The model implemented is a Decision Tree, a popular choice for classification tasks due to its simplicity and interpretability. Here's a breakdown of the process:

1. Class Definitions:
   - Question: Represents a condition (question) used to split the data at each node.
   - Leaf: Represents the end of a branch, storing the class predictions.
   - Decision_Node: Represents a decision point in the tree, storing the question and references to the true and false branches.

2. Functions:
   - unique_vals: Returns unique values for a specified column in the data.
   - class_counts: Counts the number of each type of class in the data.
   - partition: Splits the data into true and false branches based on a question.
   - gini: Calculates the Gini Impurity of the data.
   - info_gain: Calculates the information gain from a potential split.
   - find_best_split: Finds the best question to ask at each node.
   - build_tree: Recursively builds the decision tree.
   - print_tree: Prints the structure of the decision tree.
   - print_leaf: Prints the class predictions at a leaf.
   - classify: Classifies a new example by traversing the decision tree.

Model Training
The decision tree is trained using the training dataset, resulting in a tree that can classify crops based on the state, district, and season. The trained model is then saved to a file (`filetest2.pkl`) using `joblib`.

Model Inference
A second script loads the saved decision tree model and uses it to predict the type of crop based on input state, district, and season values provided via command line arguments. The prediction probabilities are printed as the output.

Summary

This implementation provides a clear and systematic approach to building a decision tree classifier for crop prediction. The dataset is preprocessed, a decision tree is trained and saved, and a separate script handles the inference. The decision tree model allows for easy interpretation and explanation of the classification results.

B. FERTILIZER RECOMMENDATION: -

The dataset `fertilizer_recommendation.csv` contains agricultural data used for recommending fertilizers based on various soil and crop parameters. The specific columns in the dataset include:

- Soil Type
- Crop Type
- N (Nitrogen content)
- P (Phosphorus content)
- K (Potassium content)
- Temperature
- Humidity
- Soil Moisture
- Fertilizer (target variable)

Code Functionality

1. Loading and Preprocessing Data:
   - The dataset is loaded using pandas.
   - Categorical features "Soil Type" and "Crop Type" are encoded into numerical values using `LabelEncoder` from `sklearn.preprocessing`.

2. Splitting Data:
   - The dataset is divided into input features (`X`) and the target variable (`y`), which is the `Fertilizer` column.

3. Training the Model:
   - A `DecisionTreeClassifier` from `sklearn.tree` is initialized with `random_state=0` for reproducibility.
   - The model is trained using the input features (`X`) and the target variable (`y`).

4. Making Predictions:
   - The script accepts input parameters for Nitrogen (N), Phosphorus (P), Potassium (K), Temperature, Humidity, Soil Moisture, Soil Type, and Crop Type as command-line arguments.
   - The categorical input columns "Soil Type" and "Crop Type" are encoded using the previously fitted `LabelEncoder`.
   - The input parameters are assembled into a list and passed to the trained `DecisionTreeClassifier` model to predict the recommended fertilizer.
   - The predicted fertilizer is printed as the output.

Purpose

The purpose of this script is to recommend the appropriate fertilizer based on user-provided soil and crop parameters. The `DecisionTreeClassifier` model, trained on historical data, helps in making these recommendations by analyzing the complex relationships between the input features and the target variable (fertilizer type). This approach leverages the decision tree algorithm's ability to handle both numerical and categorical data, providing an efficient and interpretable solution for fertilizer recommendation.

2. CROP RECOMMENDATION: -

Here's a brief description of the dataset and the machine learning algorithm used:

Dataset Description

The dataset used is the Crop Recommendation dataset, which contains data on various soil and climate parameters that are crucial for crop growth. The dataset includes the following features:

1. Nitrogen (N) content in soil
2. Phosphorus (P) content in soil
3. Potassium (K) content in soil
4. Temperature (°C)
5. Humidity (%)
6. pH level of the soil
7. Rainfall (mm)

The target variable is the type of crop suitable for the given soil and climate conditions. This dataset is used to train a model that predicts the most suitable crop based on the given input parameters.

Machine Learning Algorithm

The machine learning algorithm used for this task is the Random Forest Classifier. Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) of the individual trees. It is known for its robustness and accuracy in classification tasks. Here's a step-by-step breakdown of the process:

1. Data Splitting: The dataset is split into training and test sets, with 80% of the data used for training the model and 20% reserved for testing its performance.
2. Model Training: The Random Forest Classifier is trained on the training set. In this specific instance, the classifier uses 10 decision trees (`estimators = 10`) and the 'entropy' criterion to measure the quality of splits.
3. User Input Prediction: The trained model is then used to predict the most suitable crop based on user input, which includes values for nitrogen, phosphorus, potassium, temperature, humidity, pH level, and rainfall.

Example Usage

The script accepts user inputs in JSON format, parses them into the necessary parameters, and uses the trained Random Forest model to predict the suitable crop. For example, given inputs like `[90, 42, 43, 21, 82, 6.5, 203]`, the model predicts the crop that would thrive best under these conditions.

C. RAINFALL PREDICTION: -

Dataset Description

The dataset `crop_production_karnataka.csv` contains agricultural data for Karnataka, India. It includes information on the production of various crops across different districts and seasons, along with the area of cultivation. The specific columns in the dataset are:

- State_Name
- District_Name
- Season
- Crop
- Area
- Production

Code Functionality

1. Loading and Preprocessing Data:
   - The dataset is loaded using pandas.
   - The `Crop_Year` column is dropped since it is not used in the prediction model.
   - The dataset is split into features (`X`) and the target variable (`y`), which is the `Production` column.

2. Splitting Data:
   - The dataset is divided into training and testing sets using `train_test_split` from `sklearn.model_selection`.

3. Encoding Categorical Variables:
   - Categorical features (`State_Name`, `District_Name`, `Season`, `Crop`) are identified for one-hot encoding.
   - An `OneHotEncoder` from `sklearn.preprocessing` is used to transform these categorical columns into a numerical format.
   - The transformed categorical columns are combined with the numerical columns (area) to form the final training and testing feature sets (`X_train_final` and `X_test_final`).

4. Training the Model:
   - A `RandomForestRegressor` is initialized with 100 estimators and a random state of 42 to ensure reproducibility.
   - The model is trained on the final training feature set (`X_train_final`) and the target variable (`y_train`).

5. Making Predictions:
   - The script accepts user inputs for the state, district, season, crop type, and area of cultivation as command-line arguments.
   - The input parameters are converted into a NumPy array.
   - The categorical input columns are one-hot encoded using the fitted `OneHotEncoder`.
   - The encoded categorical columns are combined with the numerical column (area) to form the final input for prediction.
   - The trained Random Forest Regressor model makes a prediction based on this input.
   - The predicted production value is printed as the output.

Purpose:

The purpose of this script is to predict the crop production based on user inputs, utilizing historical data and the Random Forest Regressor model. This approach leverages the ensemble learning technique to handle complex relationships and interactions in the agricultural data, providing robust and accurate predictions.

D. Yield Prediction: -

Dataset and Objective

The dataset, `crop_production_karnataka.csv`, contains agricultural data for the Karnataka region, including information on crop production. The objective of the code is to predict the crop production based on several input features such as state name, district name, season, crop type, and area of cultivation.

Data Preprocessing

1. Loading Data: The dataset is loaded into a pandas DataFrame.
2. Dropping Irrelevant Columns: The `Crop_Year` column is dropped from the dataset as it is not used in the prediction model.
3. Separating Features and Target: The features (`X`) are separated from the target variable (`y`), which is the `Production` column.
4. Splitting Data: The data is split into training and testing sets to evaluate the model's performance.

Encoding Categorical Variables

Categorical variables (`State_Name`, `District_Name`, `Season`, `Crop`) are one-hot encoded using `OneHotEncoder` to convert them into a numerical format suitable for machine learning algorithms.

Model Training

A `RandomForestRegressor` is used to train the model. This ensemble method constructs multiple decision trees and averages their predictions to improve accuracy and robustness. The model is trained on the processed training data.

Making Predictions

The script accepts user input for state name, district name, season, crop type, and area of cultivation. These inputs are processed in the same way as the training data, with categorical variables being one-hot encoded. The processed input is then passed to the trained `RandomForestRegressor` model to predict the crop production.

Output

The predicted crop production value is printed as the output, providing an estimate based on the input parameters.

This approach leverages the power of ensemble learning with a Random Forest Regressor to handle the complexities and interactions in agricultural data, aiming to provide accurate crop production predictions.

✓ Chapter 4: RESULTS, ANALYSIS & DISCUSSIONS

These are a few exemplary pictorial descriptions of how the webpage works with all the possible choice conditions.

1. HOME PAGE:

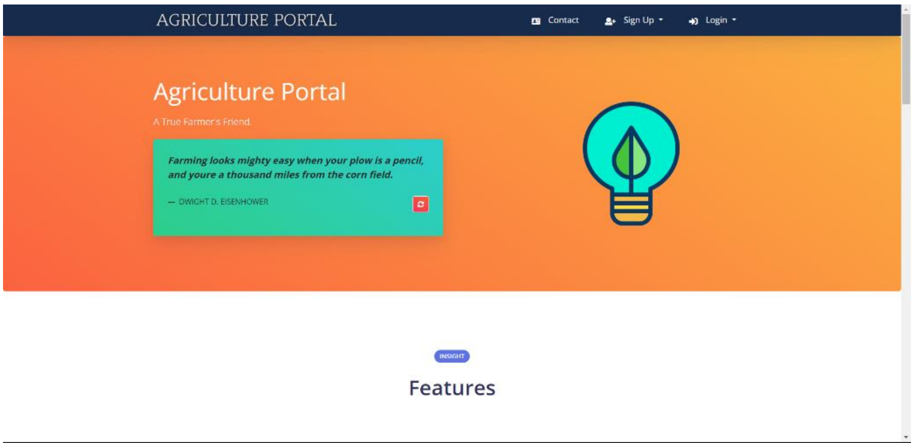

Figure 1

This is the front page, which is to be seen after opening the front page.

2. Sign-up PAGE:

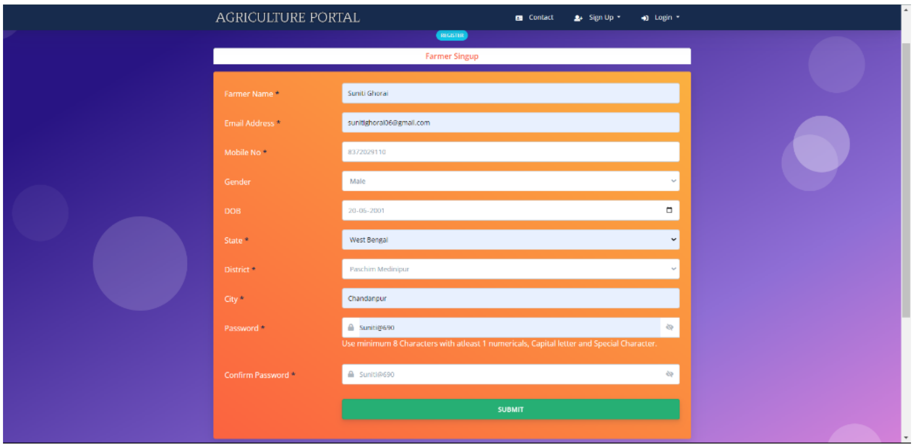

Figure 2

This is the sign-up page for the farmers portal in our webpage.

3. Crop Stocking (Admin):

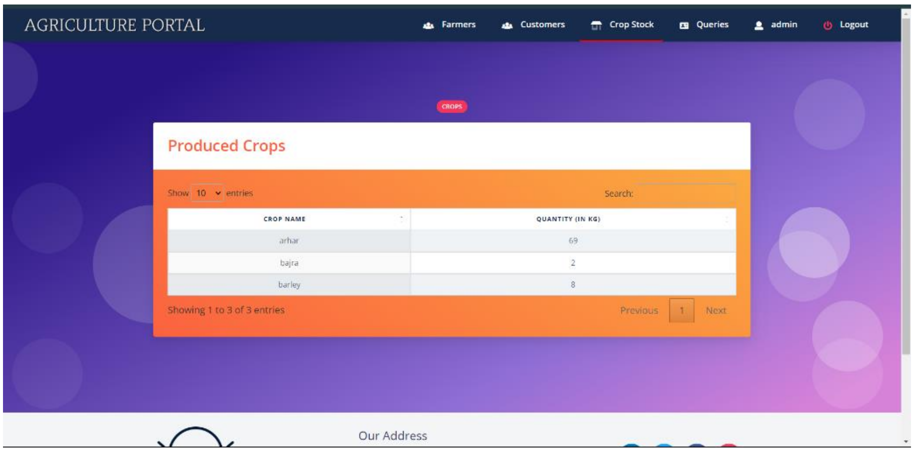

Figure 3

This is the screenshot of the table which contains the amounts of crops being stocked in "PRODUCED CROPS" apps.

4. PREDICTION OF CROPS:

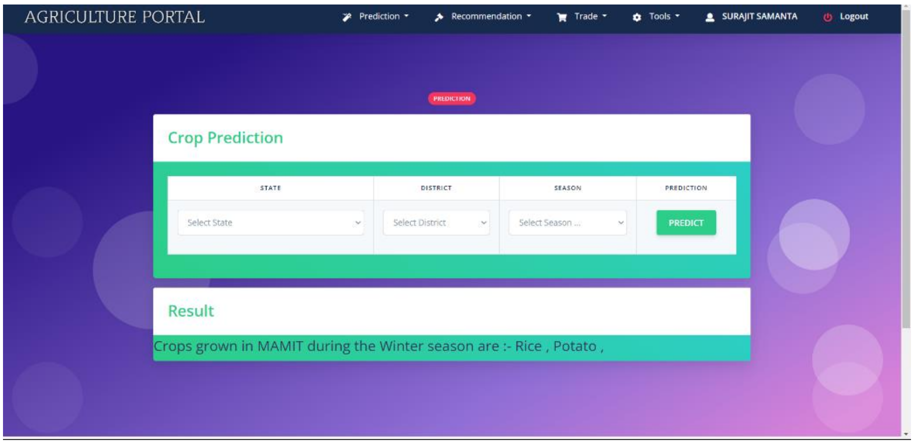

Figure 4

This is the crop prediction portal where, based on certain parameters and applying the Decision Tree model – it predicts the Crop suitable for harvesting.

5. CROP RECOMMENDATION SYSTEM:

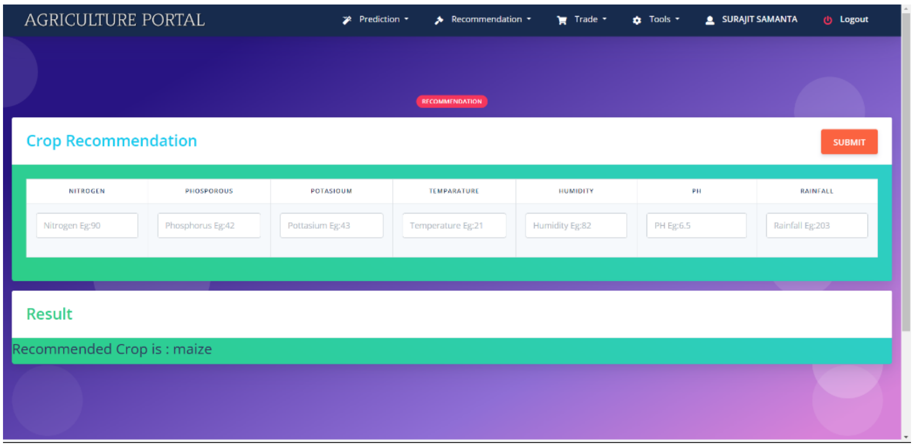

Figure 5

Based on parameters like – N, P, K, temperature, humidity etc. you can predict the suitable crop.

6. CROP STOCK UPDATION:

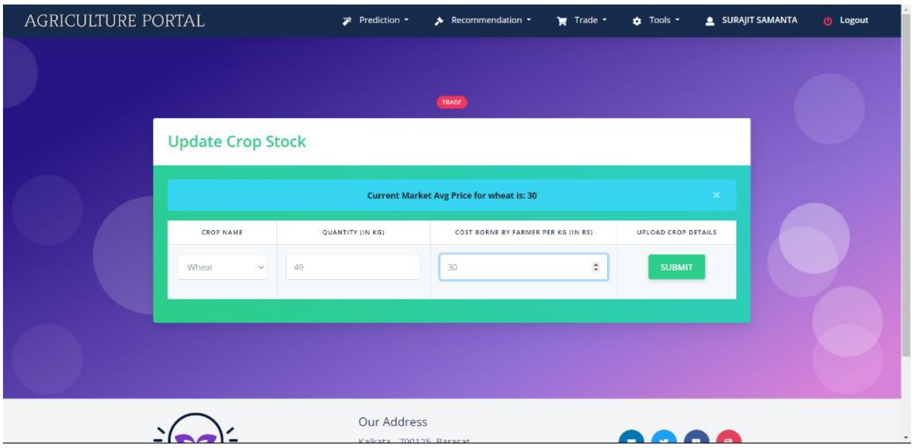

Figure 6

Also in the Admin portal, you can update the amount and price of the Crop stocks.

7. PURCHASE OF CROP:

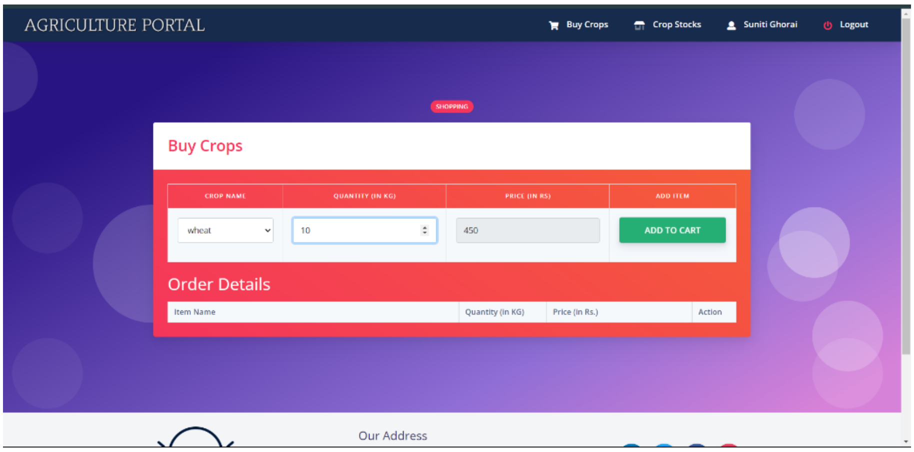

Figure 7

This is also another important part of the website, where farmers can purchase crops (updated by Admin Portal).

8. CROP AVAILABILITY CHECKING:

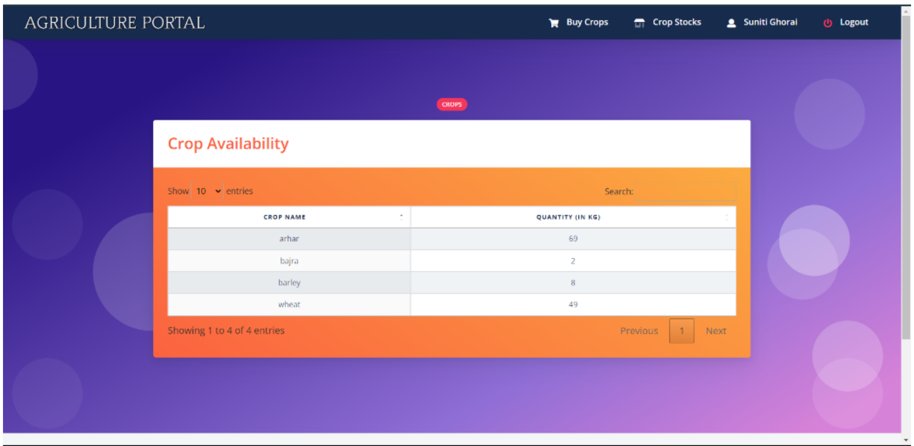

Figure 8

Another section, where the available stock can be seen for the farmers to purchase.

- These are a few aspects or examples of our webpage functionality.

✓ Chapter 5: CONCLUSSION, FUTURE SCOPE & LIMITATION

❖ Conclusion:

The primary goal of this project was to create an Agriculture Portal that bridges the gap between farmers and consumers, enabling farmers to sell their products directly and achieve better profit margins. By leveraging modern technology, the portal empowers farmers with real-time data and digital tools that enhance farm management and decision-making processes. The implementation of the portal has demonstrated significant potential in improving the economic conditions for farmers by eliminating intermediaries and allowing them to engage directly with customers.

The project has successfully developed a platform where farmers can access market information, optimize their agricultural practices, and receive customer demands through a user-friendly interface. This direct connection between farmers and consumers not only helps farmers secure better prices for their products but also ensures that consumers receive fresh produce at affordable rates. By removing agents from the supply chain, the portal contributes to the economic empowerment of farmers and supports the broader goal of agricultural modernization.

❖ Future Scope:

The future scope of the Agriculture Portal extends beyond its current capabilities, with several potential enhancements and expansions:

1. Contract Farming Integration: The introduction of contract farming features where agreements between farmers and buyers are formalized. This would ensure stable markets for farmers and predictable supply chains for buyers, fostering long-term business relationships.

2. Advanced Analytics and AI: Implementing advanced analytics and artificial intelligence (AI) tools to provide predictive insights into crop yields, pest management, and optimal planting times. These features could help farmers make more informed decisions and increase their productivity.

3. Mobile Application Development: Developing a mobile version of the portal to make it accessible to farmers in remote areas. A mobile app could offer notifications, real-time updates, and easy access to all portal features.

4. Training and Workshops: Organizing training sessions and workshops for farmers to enhance their skills in using the portal effectively and adopting modern agricultural techniques.

5. Expansion of Services: Adding new services such as access to financial resources, agricultural insurance options, and expert consultations for more comprehensive support to farmers.

- **Limitations:**

Despite the successes of the project, there are some inherent limitations that need to be acknowledged:

1. Digital Literacy: The effectiveness of the portal relies on the farmers' ability to use digital tools. In rural areas with low levels of digital literacy, some farmers may struggle to fully utilize the portal's features.

2. Internet Connectivity: Reliable internet access is crucial for the portal's operation. In regions with poor or unreliable internet connections, farmers may face difficulties in accessing the portal and its services.

3. Initial Costs: While the portal eliminates middlemen, the initial costs of setting up the technology and infrastructure might be a barrier for some farmers.

4. Scalability Challenges: As the portal grows, managing an increasing number of users and transactions can become challenging. Ensuring that the system remains scalable and performs efficiently at higher loads will require ongoing attention and resources.

5. Market Fluctuations: The portal cannot control market fluctuations or external factors affecting agricultural prices. While it provides a platform for direct sales, farmers are still subject to market conditions and price volatility.

APPENDICES

- **HARDWARE:**

- CPU - i5 12th Gen.
- GPU – Nvidia RTX 3050 (4GB)
- RAM – 16GB (DDR 4)
- OS – WINDOWS 11

- **SOFTWARE:**

- Platform to run Codes – Google Collab (version: 2024-1-8) / VS Code
- Language – Python, PhP, html, css, JScript

REFERENCES

1. Paned, Shilpa Mangesh, Prem Kumar Ramesh, Anmol Anmol, B. R. Aishwarya, Karuna Rohilla, and KUMAR SHAURYA. "Crop recommender system using machine learning approach." In 2021 5th international conference on computing methodologies and communication (ICCMC), pp. 1066-1071. IEEE, 2021.

2. Lacasta, Javier, F. Javier Lopez-Pellicer, Borja Espejo-García, Javier Nogueras-Iso, and F. Javier Zarazaga-Soria. "Agricultural recommendation system for crop protection." Computers and Electronics in Agriculture 152 (2018): 82-89.

3. Banavlikar, Tanmay, Aqsa Mahir, Mayuresh Budukh, and Soham Dhodapkar. "Crop recommendation system using Neural Networks." International Research Journal of Engineering and Technology (IRJET) 5, no. 5 (2018): 1475-1480.

4. Kumar, Avinash, Sobhangi Sarkar, and Chittaranjan Pradhan. "Recommendation system for crop identification and pest control technique in agriculture." In 2019 International Conference on Communication and Signal Processing (ICCSP), pp. 0185-0189. IEEE, 2019.

5. Sharma, Abhinav, Muskaan Bhargava, and Akshay Vijay Khanna. "AI-Farm: A crop recommendation system." In 2021 International Conference on Advances in Computing and Communications (ICACC), pp. 1-7. IEEE, 2021.

6. Madhuri, J., and M. Indiramma. "Artificial neural networks based integrated crop recommendation system using soil and climatic parameters." Indian Journal of Science and Technology 14, no. 19 (2021): 1587-1597.

7. Apat, Shraban Kumar, Jyotirmaya Mishra, K. Srujan Raju, and Neelamadhab Padhy. "An artificial intelligence-based crop recommendation system using machine learning." Journal of Scientific & Industrial Research (JSIR) 82, no. 05 (2023): 558-567.

8. Dighe, Deepti, Harshada Joshi, Aishwarya Katkar, Sneha Patil, and Shrikant Kokate. "Survey of crop recommendation systems." IRJET 5 (2018): 476-481.

9. Kiruthika, S., and D. Karthika. "IOT-BASED professional crop recommendation system using a weight-based long-term memory approach." Measurement: Sensors 27 (2023): 100722.

10. Rekha Sundari, M., G. Siva Rama Krishna, V. Sai Naveen, and G. Bharathi. "Crop recommendation system using k-nearest neighbors algorithm." In Proceedings of 6th International Conference on Recent Trends in Computing: ICRTC 2020, pp. 581-589. Springer Singapore, 2021.

11. Dhabal, Gourab, Jaykumar Lachure, and Rajesh Doriya. "Crop recommendation system with cloud computing." In 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 1404-1411. IEEE, 2021.

ACKNOWLEDGEMENT

We would like to extend our sincere gratitude to **Piyali Maur Madam** for his unwavering support and invaluable guidance throughout the development of the minor project about **Agriculture Portal**. His mentorship has been instrumental in shaping the direction of our project and elevating its overall quality. The insightful feedback provided at key milestones significantly contributed to refining the user interface, enhancing functionality, and ensuring the application meets the highest standards.

We are deeply appreciative of the dedicated time and expertise of **Piyali Madam** generously shared, going above and beyond to provide not only technical insights but also valuable perspectives on effective project management. His commitment to our academic and professional growth has left an indelible mark on this endeavor.

In particular, **Piyali Madam**'s encouragement to explore innovative solutions and his emphasis on a user-centric design approach have been integral to the success of the Expense Tracker App. We are immensely grateful for the inspiration derived from his teaching, which has fueled our motivation to deliver a robust and user-friendly application.

Once again, thank you, **Piyali Madam**, for being more than an instructor – for being an inspiring mentor who has played a pivotal role in the realization of this project. Your support has been the driving force behind our achievements, and we are truly fortunate to have benefited from your guidance.