

A  
Project Report  
on  
**SYSTEM FOR SYSTEMATIC FARMING**

Submitted in Partial Fulfillment of  
the Requirements for the Degree  
of  
**Bachelor of Engineering**  
in  
**Computer Engineering**  
to  
**Kavayitri Bahinabai Chaudhari**  
**North Maharashtra University, Jalgaon**

Submitted by  
**Samadhan Rameshwar Zagare**  
**Saurabh Dhanraj Berad**  
**Yash Suresh Deshmukh**  
**Gaurav Bhagwan Patil**

Under the Guidance of  
**Mr. S.H.Rajput**



DEPARTMENT OF COMPUTER ENGINEERING  
SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY,  
BAMBHORI, JALGAON - 425 001 (MS)  
2022 - 2023

**SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY,  
BAMBHORI, JALGAON - 425 001 (MS)  
DEPARTMENT OF COMPUTER ENGINEERING**

## **CERTIFICATE**

This is to certify that the project entitled *System for Systematic Farming*, submitted by

**Samadhan Rameshwar Zagare  
Saurabh Dhanraj Berad  
Yash Suresh Deshmukh  
Gaurav Bhagwan Patil**

in partial fulfillment of the degree of *Bachelor of Engineering in Computer Engineering* has been satisfactorily carried out under my guidance as per the requirement of Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon.

**Date:** April 18, 2023

**Place:** Jalgaon

**Mr. S.H.Rajput  
Guide**

Dr. Manoj E. Patil  
**Head of Department**

Prof. Dr. G.K.Patnaik  
**Principal**

# Acknowledgements

The success of our project depends largely on the encouragement and help from many others. We take this opportunity to express our thankfulness to the people who have contributed in the successful completion of this project and have been the real guiding force in our project. We would like to show our gratitude to our project mentor **Mr. S.H.Rajput sir** continuously guiding and tremendously helping us throughout this project. We feel motivated and encouraged every time we attend his meeting. Without his encouragement and guidance the completion of this project on **System For Systematic Farming** detection would not have been achieved

Samadhan Rameshwar Zagare  
Saurabh Dhanraj Berad  
Yash Suresh Deshmukh  
Gaurav Bhagwan Patil

# Contents

<b>Acknowledgements</b>	<b>ii</b>
<b>Abstract</b>	<b>1</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Background . . . . .	2
1.2 Motivation . . . . .	3
1.3 Scope . . . . .	3
1.4 Objective . . . . .	5
1.5 Selection of Life cycle model . . . . .	5
1.6 Organization Of Report . . . . .	6
1.7 Summary . . . . .	7
<b>2 Project And Management</b>	<b>8</b>
2.1 Feasibility Study . . . . .	8
2.1.1 Technical Feasibility . . . . .	9
2.1.2 Operational Feasibility . . . . .	9
2.1.3 Economical Feasibility . . . . .	9
2.2 Risk Analysis . . . . .	10
2.3 Project Scheduling . . . . .	10
2.4 Effort Allocation . . . . .	11
2.5 Cost Estimation . . . . .	12
2.6 Summary . . . . .	13
<b>3 Analysis</b>	<b>14</b>
3.1 Represent Requirement Collection And Identification . . . . .	14
3.2 Software Requirement Specification (SRS) . . . . .	15
3.2.1 Production Feature . . . . .	15
3.2.2 Operating Environment . . . . .	15
3.2.3 Assumption . . . . .	16
3.2.4 Functional Requirement . . . . .	16

3.2.5	Non Functional Requirement . . . . .	16
3.2.6	External Interfaces . . . . .	16
3.3	Summary . . . . .	18
<b>4</b>	<b>Design</b>	<b>19</b>
4.1	System Architecture . . . . .	19
4.2	Data Flow Diagram . . . . .	20
4.2.1	Level 0 DFD . . . . .	20
4.2.2	Level 1 DFD . . . . .	21
4.3	UML Diagrams . . . . .	22
4.3.1	UseCase Diagram . . . . .	22
4.3.2	Sequence Diagram . . . . .	22
4.3.3	Collaboration Diagram . . . . .	23
4.3.4	Class Diagram . . . . .	24
4.3.5	State Transition Diagram . . . . .	25
4.3.6	Component Diagram . . . . .	26
4.3.7	Deployment Diagram . . . . .	27
4.3.8	Summary . . . . .	28
<b>5</b>	<b>Coding and Implementation</b>	<b>29</b>
5.1	Define Intents . . . . .	29
5.2	Required Software and Hardware for Development . . . . .	31
5.3	Implementation Environment . . . . .	31
5.3.1	Programming Language . . . . .	31
5.4	Summary . . . . .	32
<b>6</b>	<b>Testing</b>	<b>33</b>
6.1	Black Box Testing . . . . .	33
6.1.1	Functional Testing :- . . . . .	34
6.1.2	Stress Testing :- . . . . .	34
6.1.3	Load Testing :- . . . . .	34
6.1.4	Recovery Testing :- . . . . .	34
6.1.5	User Acceptance Testing :- . . . . .	34
6.1.6	Alpha Testing :- . . . . .	34
6.1.7	Beta Testing :- . . . . .	35
6.2	White Box Testing . . . . .	35
6.2.1	Unit Testing:- . . . . .	35
6.2.2	Branch Coverage:- . . . . .	36

6.2.3	Security Testing:- . . . . .	36
6.2.4	Mutation Testing:- . . . . .	36
6.3	Manual Testing . . . . .	36
6.4	Automated Testing . . . . .	37
6.5	Test Cases Identification and Execution . . . . .	37
6.6	Summary . . . . .	39
<b>7</b>	<b>Result and Discussion</b>	<b>40</b>
7.1	Home Page . . . . .	40
7.2	Translation API . . . . .	40
7.3	Chatbot . . . . .	41
7.3.1	Goverment Scheems . . . . .	41
7.4	Farmer Portal . . . . .	42
7.5	Seeds Information . . . . .	42
7.6	Fertilizer Information . . . . .	43
7.7	Crop Recomendation System . . . . .	43
7.7.1	Crop Recomendation Result . . . . .	43
7.8	Analysis . . . . .	44
7.9	Analysis of Crop Yearwise . . . . .	45
7.10	Analysis of Perticular Crop . . . . .	45
7.11	About us Page . . . . .	46
7.12	Contact us Page . . . . .	46
<b>8</b>	<b>Conclusion</b>	<b>47</b>
<b>Bibliography</b>		<b>48</b>

# **Abstract**

hspace1cmThe agricultural information system provides its users and researches to get online information about, the crop, statistical details and new tendencies. The trends of the crops act so that these will be pretty important to the users who access these via the Internet. The main features of the information system includes information retrieval facilities for users from anywhere in the form of obtaining statistical information about fertilizer, research institutes and researches, land availability, diseases, suitable soil concentration for the corresponding crops, statistical information about exports and etc. In addition this provides individual information about Intercrops related to main crops. The system allows the retrieving facilities but also the updating facilities to the authorized persons in the corresponding institutes. Python was used to create the front end for the system and Mongodb Server was used for the back end. The graphical user interface of the front end use Bootstrap, ASP Interfaces and access the back end Mongodb Server Database using embedded Database Queries for the retrieval and update. The front end and back-end is connected using a Mongodb Server The agricultural information system can be maintained by the Goverment Of Maharastra/India. They will be able to use the this via Internet. Users will be given login name and password so that they can log-in to the database. Mozilla Firefox,internet explorer,chrome,microsoft Browsers can be used to browse the information.

# **Chapter 1**

## **Introduction**

The "Farm Management System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardship faced by the existing system. More ever this system is designed for the particular need of the carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use the system. Thus by this all it proves it is user-friendly. Farm Management System, as described above can lead to error free, secure, reliable, and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organisation in better utilisation of resources.

Every organisation, weather big or small, has challenges to overcome and managing the information of Crops, Farm, Insecticide, Equipments, Pesticides, Every farm Management System has different farm need, therefore we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and will help you to ensure that your organisation is equipped with the right level of information and details of your future goals. Also, for those busy executives who are always on the go. Our system come with remote access features. Which will allow you to manage your workforce, at all times. These systems will ultimately allow you to better manage resources.

### **1.1 Background**

- Indian agriculture began by 9000 BCE as a result of early cultivation of plants, and domestication of crops and animals.

- Settled life soon followed with implements and techniques being developed for agriculture.
- Double monsoons led to two harvests being reaped in one year. Indian products soon reached the world via existing trading networks and foreign crops were introduced to India.
- Plants and animals—considered essential to their survival by the Indians—came to be worshiped and venerated.

## 1.2 Motivation

Given the large-scale implications of a free and fair press in a democracy, the general public needs a more scientific and open approach of discerning baseless news stories from credible, fact-based journalism. We intend to work on a tool that ingests a news article URL and assigns a credibility score to it, indicating whether or not the user should trust it as factual or should conduct further investigation and make their decision. We aim to provide the user with the ability to classify the news as fake or real and also check the authenticity of the website publishing the news.

## 1.3 Scope

It may help collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to Farm Management System. It will be also reduced the cost of collecting the management collection procedure will go on smoothly.

Our project aims at Business process automation, i.e. we have tried to computerize various processes of Farm Management System.

- In computer system the person has to fill the various forms number of copies of the forms can be easily generated at a time.
- In computer system, it is not necessary to create the manifest but we can directly print it, which saves our time.
- To assist the staff in capturing the effort spent on their respective working areas.
- To utilize resources in an efficient manner by increasing their productivity through automation.
- The system generates types of information that can be used for various purposes.
- It satisfy the user requirement

- Be easy to understand by the user and operator
- Be easy to operate
- Have a good user interface
- Delivered on schedule within the budget.
- Be expandable

### **REPORT PROVIDED BY OUR SYSTEM WILL BE:**

- It generates the report on Farm, Crops, Cost Range
- Provide filter reports on Crops, Insecticides, Pesticides
- You can easily export PDF for the Farm Cost Range, Insecticides
- Application also provides excel export for Crops, Crops, Pesticides
- You can also export the report into csv format for Farm, Crops, Pesticides

### **MODULES OF THE SYSTEM**

- Farm Management Module: Used for managing the Farm details.
- Pesticides Module: Used for managing the details of Pesticides
- Cost Range Module: Used for managing the details of Cost Range
- Crops Management Module: Used for managing the information and details of the crop.
- 2Crops Module: Used for managing the Crops details
- Insecticides Module: Used for managing the Insecticides information
- Login Module: Used for managing the login details
- Users Module: Used for managing the users of the system

### **INPUT DATA AND VALIDATION OF PROJECT**

- All the fields such as Farm, Crops, Pesticides are validated and does not take invalid values
- Each form for Farm, Crops, Cost Range can not accept blank value fields
- Avoiding errors in data
- Controlling amount of input
- Integration of all the modules/forms in the system.
- Preparation of the test cases.
- Preparation of the possible test data with all the validation checks.
- Actual testing done manually.
- Recording of all the reproduced errors.
- Modifications done for the errors found during testing.
- Prepared the test result scripts after rectification of the errors.
- Functionality of the entire module/forms.
- Validations for user input.
- Checking of the Coding standards to be maintained during coding.

- Testing the module with all the possible test data.
- Testing of the functionality involving all type of calculations etc.
- Commenting standard in the source files.

### **SQA STRATEGY**

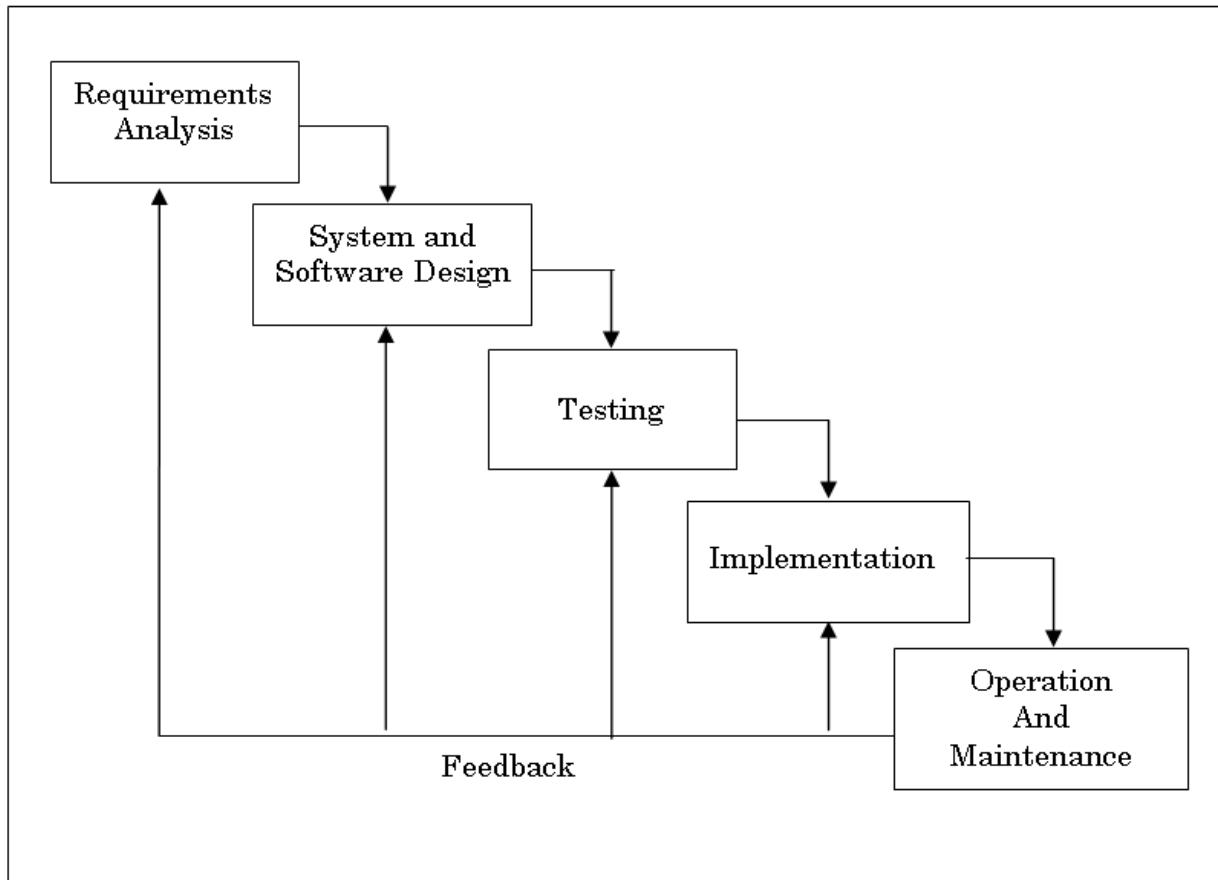
- In the first step, we will select the test factors and rank them. The selected test factors such as reliability, maintainability, portability or etc, will be placed in the matrix according to their ranks.
- The second step is for identifying the phases of the development process. The phase should be recorded in the matrix,
- The third step is that identifying the business risks of the software deliverables. The risks will be ranked into three ranks such as high, medium and low.

## **1.4 Objective**

The main objective of the Project on Farm Management System is to manage the details of Farm, Crops, Crops, Insecticides, Pesticides. It manages all the information about Farm, Cost Range, Pesticides, Farm. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Farm, Crops, Cost Range, Crops. It tracks all the details about the Crops, Insecticides, Pesticides.

## **1.5 Selection of Life cycle model**

The software development life cycle model selected for this project is the Waterfall Model. Waterfall approach was the first SDLC (Software Development Life Cycle) Model to be widely used in software engineering to ensure success of the project. It was developed by Winston W. Royce in 1970. In "The Waterfall" approach, the whole process of software development is divided into separate phases, typically the outcome of one phase acts as the input for the next phase sequentially. All the phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. Requirements for this project are well documented and fixed.



**Figure 1.0 Modified Waterfall Approach Model**

Waterfall Model is best suited model for this project.

1. Because requirements are easily understandable and defined
2. We can define requirements in early stage of development
3. User involvement in all phases is not necessary
4. Limited user's participation

## 1.6 Organization Of Report

**Chapter 1** entitled as Introduction describes the details about Background, Problem Definition, Scope and Objective of the project, Identification of Software Development Process Model and Organization of report.

**Chapter 2** entitled as Project Planning and Management consists of details about the Feasibility Study, Risk Analysis, Project Scheduling, Effort Allocation and Cost Estimation of the project.

**Chapter 3** entitled as Analysis describes in detail, the Requirement Collection and Identification, H/w and S/w Requirements, Functional and Non-Functional Requirements and a Software Requirements Specification(SRS).

**Chapter 4** includes design about System Architecture, Data Flow Diagram and various UML Diagrams.

**Chapter 5** titled conclusion and future work discusses about the overall outcomes and future possibilities of the project.

## 1.7 Summary

As mentioned in above sections, this project shows a System for Systematic Farming. The scopes, objective, etc. are as mentioned above. In the next chapter, project planning and management will be discussed.

# **Chapter 2**

## **Project And Management**

Project planning is a procedural step in project management. It is the practice of initiating, planning, executing, controlling and closing the work team to achieve specific goals. Project planning and management is important because it ensures that the right people do the right things, at the right time. It also ensures the proper project lifecycle. The organization of this chapter is as below. Section 2.1 shows the Feasibility Study of the project. Risk Analysis of the project is represented in Section 2.2 and Project Scheduling is described in Section 2.3. Section 2.4 and 2.5 describe the Effort Allocation and Cost Estimation respectively. The Summary is mentioned in Section 2.6.

### **2.1 Feasibility Study**

A feasibility study is an assessment of the practicality of a proposed project or system. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained. A well-designed feasibility study should provide a historical background of the business or project, a description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation.

A feasibility study evaluates the project's potential for success; therefore, perceived objectivity is an important factor in the credibility of the study for potential investors and lending institutions. It must therefore be conducted with an objective, unbiased approach to provide information upon which decisions can be based. Taking into consideration the technical, operational and economic feasibilities as below, the project can be anticipated as feasible overall. There are few types of feasibility that exists. So, developers should take

care of these feasibility and take them into consideration:

### **2.1.1 Technical Feasibility**

This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project. At this level, the concern is whether the proposal is both technically and legally feasible (assuming moderate cost). It is an evaluation of the hardware and software and how it meets the need of the proposed system.

This project is built upon VS code Editor, a simple Web-Application with Python as the programming language and can be easily hosted on cloud server. Also all the other technologies used are capable of building such a project and serve as well as maintain it for longer period of time. All the required hardware and software are easily available in the market. Hence the project is technically feasible.

### **2.1.2 Operational Feasibility**

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

The operational feasibility assessment focuses on the degree to which the proposed development project fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes. The application is operationally feasible since it is build with the idea for integration with various existing applications and systems.

### **2.1.3 Economical Feasibility**

Describes how much time is available to build the new system, when it can be built, whether it interferes with normal business operations, type and amount of resources required, dependencies, and developmental procedures with company revenue prospectus.

As the necessary hardware and the software are easily available in the market at low cost, the initial investment is the only cost incurred and does not need further enhancement. Hence it is economically feasible.

## **2.2 Risk Analysis**

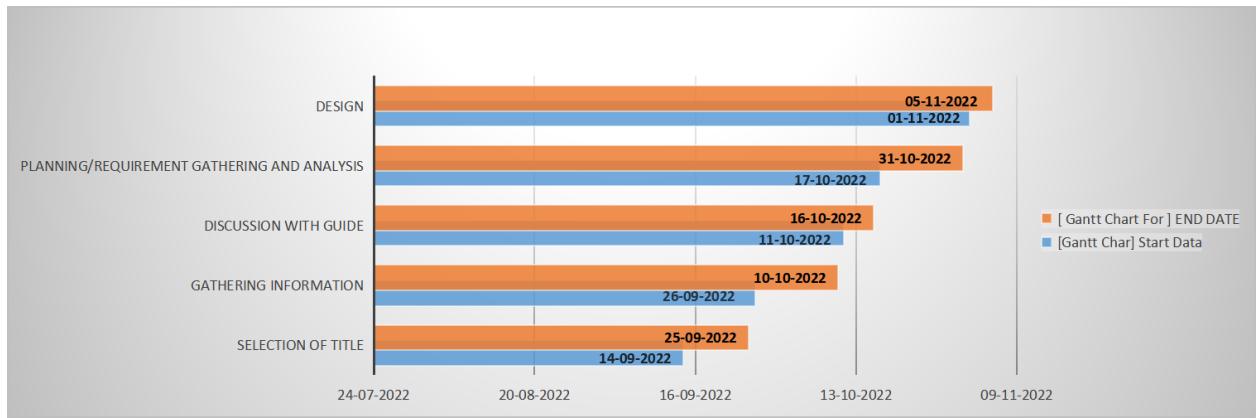
Risk Analysis and Management is a key project management practice to ensure that the least number of surprises occur while your project is underway. While we can never predict the future with certainty, we can apply a simple and streamlined risk management process to predict the uncertainties in the projects and minimize the occurrence or impact of these uncertainties. This project has a very small window for experiencing failures since the real life interaction is less. This improves the chance of successful project completion and reduces the consequences of those risks.

## **2.3 Project Scheduling**

Generally, project scheduling can be stated as the estimated time required for any project from its time of beginning to the end of the project. In detail, for every task, there is a deadline because all the tasks for the completion of project are planned earlier. So that, each task is scheduled to certain time limit. 15 In short, in project management, listing of projects milestones, activities and all from starting to end date, are considered in the project scheduling. A schedule is generally used in the project planning and management of the project with some kind of attributes as budget, task allocation and duration, resource allocation and all.

Sr No.	Tasks	Start Date	Completion Date
1	Selection Of Title	14-Sep-22	25-Sep-22
2	Gathering Information	26-Sep-22	10-Oct-22
3	Discussion With Guide	11-Oct-22	16-Oct-22
4	Planning/Requirement Gathering and Analysis	17-Oct-22	31-Oct-22
5	Coding and Implementation	01-Nov-22	15-Mar-23
6	Testing	16-Mar-23	31-Mar-23
7	Report and Documentation	26-Sep-22	31-Mar-23

**Table :- Task scheduling for the project**



**Gantt Chart For Task Scheduling**

## 2.4 Effort Allocation

Effort Allocation is necessary so every team member can give its best to the project. Project was divided into smaller module and task form, for simplification and easy understanding of project overall. Some modules include every team associate's presence to take advantage of team decision taking skills, and some task include some individual member to work on it with precision.

We divided the project into 6 modules.

1. Gathering of Information
2. Planning / Requirement Analysis
3. Study of included Stack and frameworks
4. Selection of Life cycle Model
5. Planning and Management
6. Analysis & Design UML

		Team Associates			
Sr.no.	Modules	Samadhan Zagare	Saurabh Berad	Gaurav Patil	Yash Deshmukh
1	Gathering of Information	✓	✓	✓	✓
2	Planning/Requirement Analysis	✓	✓	✓	✓
3	Study of included stack and framework	✓	✓	✓	✓
4	Selection of life cycle model		✓		✓
5	Planning and Management	✓			✓
6	Analysis & Design UML	✓		✓	✓
7	Coding And Implementation	✓	✓		✓
8	Testing	✓	✓	✓	✓
9	Report and Documentation	✓	✓		✓

**Table :- Chart of Effort Allocation**

## 2.5 Cost Estimation

Cost Estimation is an important phase for any project. It predicts if the project investment is adequate or there will shortage of capital. It presents the total cost required for development of project. Cost Estimation should be done before initiating the development to prevent loss of efforts and project failure during development. For estimation of cost for this project, we need to consider the server costs for deployment, although the cost is extremely variable since it is dependent on real-time usage. The cost for a machine learning project is generally calculated in three components i.e. data cost, research cost and production cost. Since our project has the required dataset already available, the data cost for the project is zero. The research cost is dependent on the number of people involved in the project for the amount of time required for the project. Assuming the cost for each person who does research to be Rs. 15000 per month, if 4 people work for a timeline of 6 months, the research cost will be

$$\text{Research Cost} = 4 \text{ people} \times 6 \text{ months} \times \text{Rs. } 15000$$

$$= \text{Rs. } 3,60,000$$

According to the Google Cloud Calculator, the cost for deploying Identifier Web Page as a server, 4 vCPU's and Upto 16 GB RAM is Rs. 24,900

$$\text{Production Cost} = 12 \text{ months} \times \text{Rs. } 3000 = \text{Rs. } 36,000$$

Hence the total cost of the project can be calculated as

$$\begin{aligned}\text{Total Cost} &= \text{Research Cost} + \text{Production Cost} \\ &= \text{Rs. } 3,60,000 + \text{Rs. } 24,900 \\ \text{Total Cost} &= \text{Rs. } 3,84,900\end{aligned}$$

## 2.6 Summary

The project, is hence found to be feasible since there is a balance of resources required and the cost incurred. Also the project reduces burden on existing methods required for analysis and maintenance of roads. The project will be able to easily integrate with other required systems

# **Chapter 3**

## **Analysis**

The development of computer-based information system includes the system analysis phase which produces or enhances the data model which itself is to creating or enhancing a database. There are a number of different approaches to system analysis. The analysis is the process which is used to analyze, refine and scrutinize the gathered information of entities in order to make consistence and unambiguous information. Analysis activity provides a graphical view of the entire System. System Analysis is the process of gathering and interpreting facts, diagnosing problems and using the facts to improve the system. System analysis chapter will show overall system analysis of the concept, description of the system, meaning of the system. System analysis is the study of sets of interacting entities, including computer system analysis. The organization of this Chapter is as follows. Section 3.1 represents Requirement Collection and Identification. Software Requirement and Specification are described in the Section 3.2. Section 3.3 describes summary of the chapter.

### **3.1 Represent Requirement Collection And Identification**

Requirement collection is the process which is used to gather, analyze, and documentation and reviews the requirements. Requirements describe what the system will do in place of how. In practical application, most projects will involve some combination of these various methods in order to collect a full set of useful requirements. Requirements collection is initiated when the project need is first identified and the project “solution” is to be proposed. Requirements refinement continues after the project is “selected” and as the scope is defined, aligned and approved.

#### **THE PROPOSED SYSTEM HAS THE FOLLOWING REQUIREMENTS**

- System needs store information about new entry of Farm.
- System needs to help the internal staff to keep information of Crops and find them as per

various queries.

- System need to maintain quantity record.
- System need to keep the record of Crops.
- System need to update and delete the record.
- System also needs a search area.
- It also needs a security system to prevent data.

## 3.2 Software Requirement Specification (SRS)

Software Specification will provide a broad understanding of the requirement specification of this system. Also, understand features of this system along with the requirements.

Software Requirement Specification documents guide the developers in the development process and it will help to reduce the ambiguity of the requirements provided by the end-user. It's used to provide critical information to multiple teams — development, quality assurance, operations, and maintenance. This keeps everyone on the same page.

### 3.2.1 Production Feature

The product features are high level attributes of a software or product such as software performance, user-friendly interface, security portability, etc. These attributes are defined according to the product, in this case, a software product. They are as follows:

- The user will be able to upload the Query to be analysed.
- The user will be able to view the results of submitted Query.
- The user will be able to integrate this system with any working browser Systems.

### 3.2.2 Operating Environment

The software will operate within the following environment:

- Operating System: Standard processor with a speed of 1.6 GHz
- Any browser supporting HTML5 and JavaScript
- Any system with at least 2GB RAM
- System with processor Intel Pentium 4 or later

### **3.2.3 Assumption**

- It is assumed that the web portal will load and render correctly and as expected on the operating machine.
- It is assumed that the user will have a working internet connection with sufficient.
- It is assumed that the user is able to access and register the new Query through web interface.
- It is assumed that the user will viewing information as Proper manner internet speed.

### **3.2.4 Functional Requirement**

Functional requirements are the functions which are expected from the software or platform. Functional requirements along with requirement analysis help identify missing requirements. They help clearly define the expected system service and behavior.

Functional requirements are as follows:

- Search Bar For searching the information about crop on Website.
- Provide information about Crop,Seed,Diseases,Treatment and Wheather.

### **3.2.5 Non Functional Requirement**

Non-functional Requirement is mostly quality requirement. That stipulates how well the portal does, what it has to do. Other than functional requirements in practice, this would entail detail analysis of issues such as availability, security, usability and maintainability.

Non-functional requirements are as follows:

- Provide a Filter Option For Searching A Particular Crop
- Provide chatbot For Website navigation.
- Website Showing Recent Search For Users.
- The results should be comprehensive and detailed.

### **3.2.6 External Interfaces**

- **User Interfaces** User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs. The overall flow of screens and messages is called a dialogue. Guidelines For User Interface Design

1. The system user should always be aware of what to do next.
2. The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area.
3. Message, instructions or information should be displayed long enough to allow the system user to read them.
4. Use display attributes sparingly.
5. Default values for fields and answers to be entered by the user should be specified.
6. A user should not be allowed to proceed without correcting an error.
7. The system user should never get an operating system message or fatal error.

### **Preliminary Product Description**

The first step in the system development life cycle is the preliminary investigation to determine the feasibility of the system. The purpose of the preliminary investigation is to evaluate project requests. It is not a design study nor does it include the collection of details to describe the business system in all respect. Rather, it is the collecting of information that helps committee members to evaluate the merits of the project request and make an informed judgment about the feasibility of the proposed project. Analysts working on the preliminary investigation should accomplish the following objectives :

- Clarify and understand the project request
- Determine the size of the project.
- Assess costs and benefits of alternative approaches.
- Determine the technical and operational feasibility of alternative approaches.
- Report the findings to management, with recommendations outlining the acceptance or rejection of the proposal.

**Benefits To Organisation** The organization will obviously be able to gain benefits such as savings in operating cost, reduction in paperwork, better utilization of human resources and more presentable image increasing goodwill.

**The Initial Cost** The initial cost of setting up the system will include the cost of hardware software (OS, add-on software, utilities) labour (setup maintenance). The same has to bear by the organization. **Running Cost** Besides, the initial cost the long term cost will include the running cost for the system including the AMC, stationary charges, cost for human resources, cost for update/renewal of various related software. **need for training** The users along with the administrator need to be trained

at the time of implementation of the system for smooth running of the system. The client will provide the training site. We talked to the management people who were managing the financial issues of the center, the staff who were keeping the records in lots of registers and the reporting manager regarding their existing system, their requirements and their expectations from the new proposed system. Then, we did the system study of the entire system based on their requirements and the additional features they wanted to incorporate in this system. Reliable, accurate and secure data was also considered to be a complex task without this proposed system. Because there was no such record for keeping track of all the activities, which was done by the Farm Management System on the daily basis. The new system proposed and then developed by me will ease the task of the organization in consideration. It will be helpful in generating the required reports by the staff, which will help them to track their progress and services. Thus it will ease the task of Management to a great extent as activities to be performed, are computerized through this system.

The proposed system has several options for users to interact with. Following are the user interfaces:

- Web-application (GUI)
- Command Line Interface (Terminal based interface)

The web application will be available so that the users will be able to access the system through a simple web-page will be available so that the user will be able to integrate it with other systems easily.

- **Software Interfaces**

The only software interface required for this project is the Application Programming Interface with the VS Code Editor which will then process the Data. This software interface will run on local server along with the VS Code extension live server .

### **3.3 Summary**

In the chapter, Analysis was presented which included the hardware and software requirements, functional and non-functional requirements and the software requirements specification (SRS) as well. In the next chapter, Design is described along with various UML diagrams.

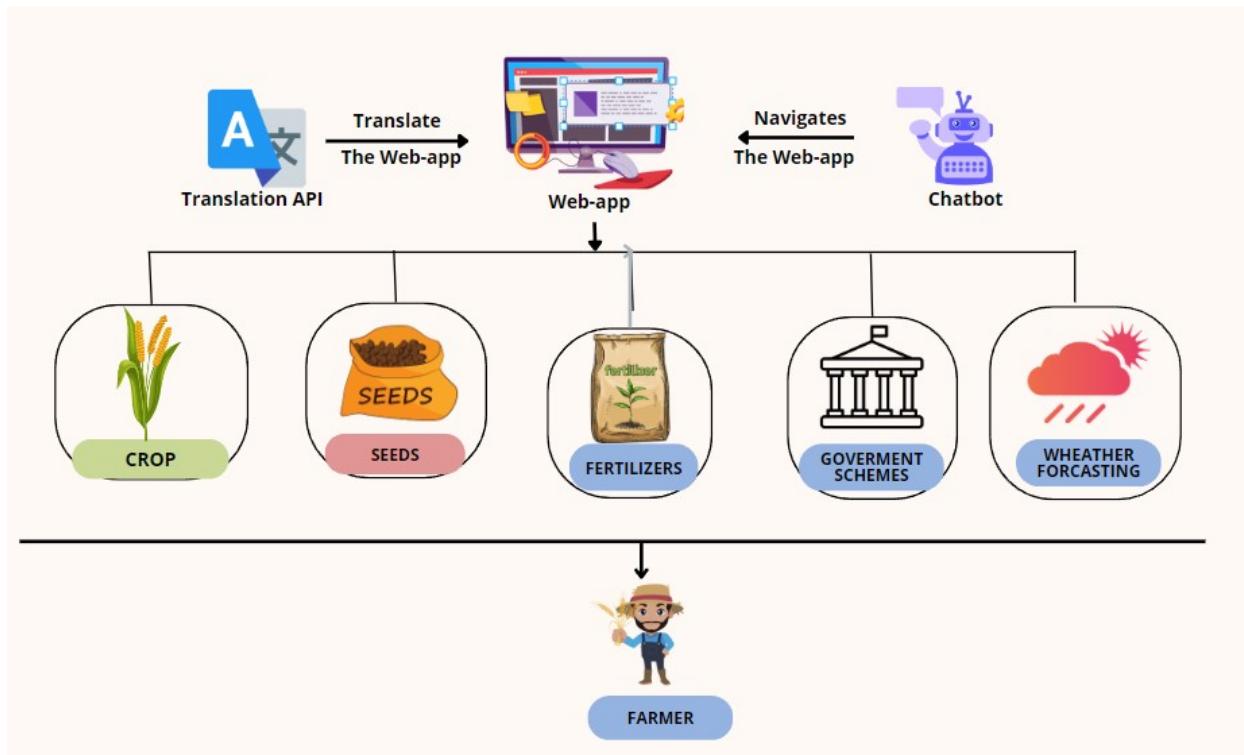
# **Chapter 4**

## **Design**

Design is the activity to design and model the various component of software system. The system design provides the understanding and procedural details necessary for implementing the system. Design is helpful for a better understanding of the project. It contains the UML diagrams, data flow diagrams. UML is a modeling language which is used to document the object-oriented analysis and design. The organization of this Chapter is as follows. Section 4.1 describes the system architecture of the project. DFD of the project are represented in Section 4.2. Section 4.3 represents UML Diagrams (Use case, Class, Sequence, Component, Deployment, State chart, Activity diagram, Class Diagram, Component Diagram, etc.) of the project. Finally, the Summary is described in last Section 4.4.

### **4.1 System Architecture**

Systems Architecture is a generic discipline to handle objects (existing or to be created) called "systems", in a way that supports reasoning about the structural properties of these objects. The system architecture is the conceptual model that defines the structure, behavior and more views of a system. An architecture description is a formal description and representation of a system. It provides broad understanding of the portal. In the system architecture database provide the functionality like get information, select criteria, etc. to users.

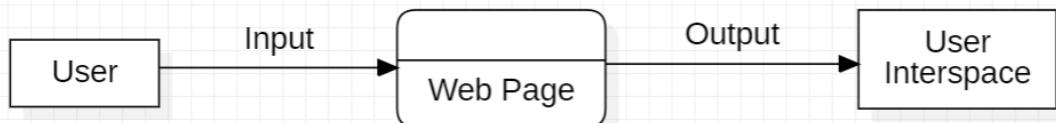


## 4.2 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the ‘flow’ of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

### 4.2.1 Level 0 DFD

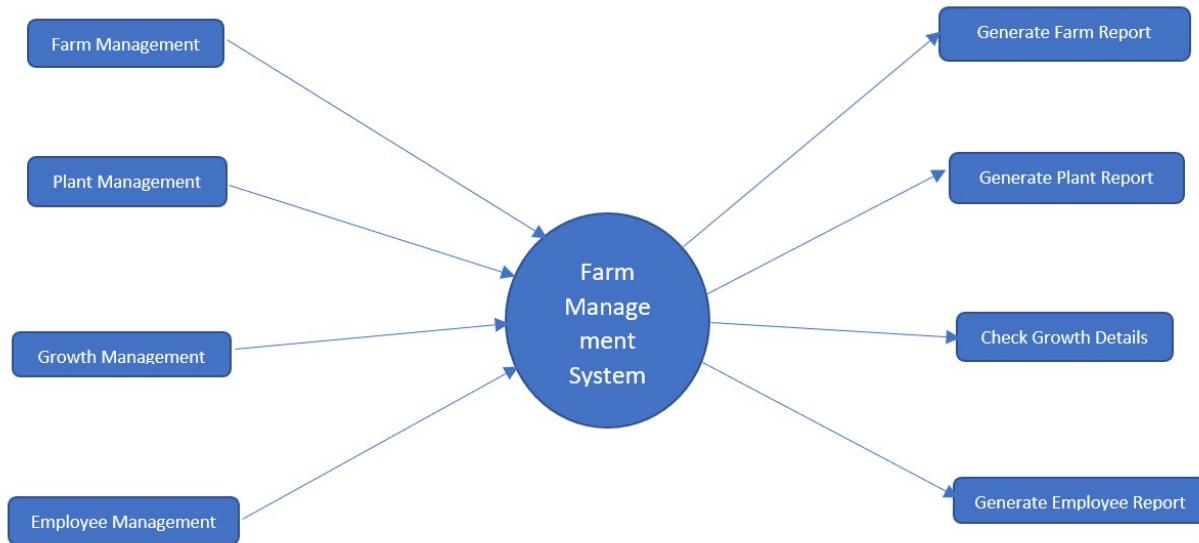
Level 0 contains one input and one output. The system provides information to the user means system is input and the user is output. Following Figure shows Level 0 DFD of project.



**Level 0 Data Flow Diagram**

#### 4.2.2 Level 1 DFD

A level 1 data flow diagram offers a more detailed look at the processes that make up an information system than a level 0 DFD does. It can be used to plan or record the specific makeup of a system. Following Figure shows Level 1 DFD of project.

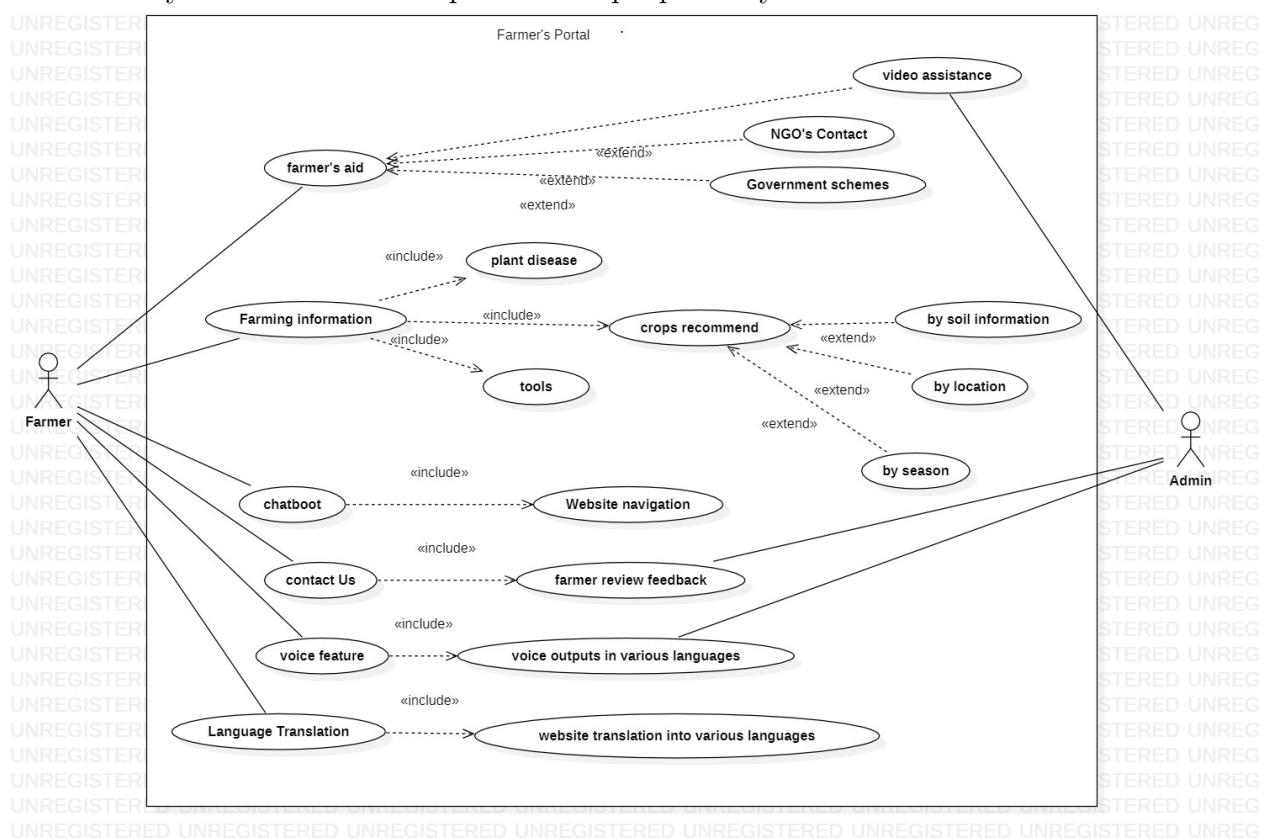


## 4.3 UML Diagrams

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

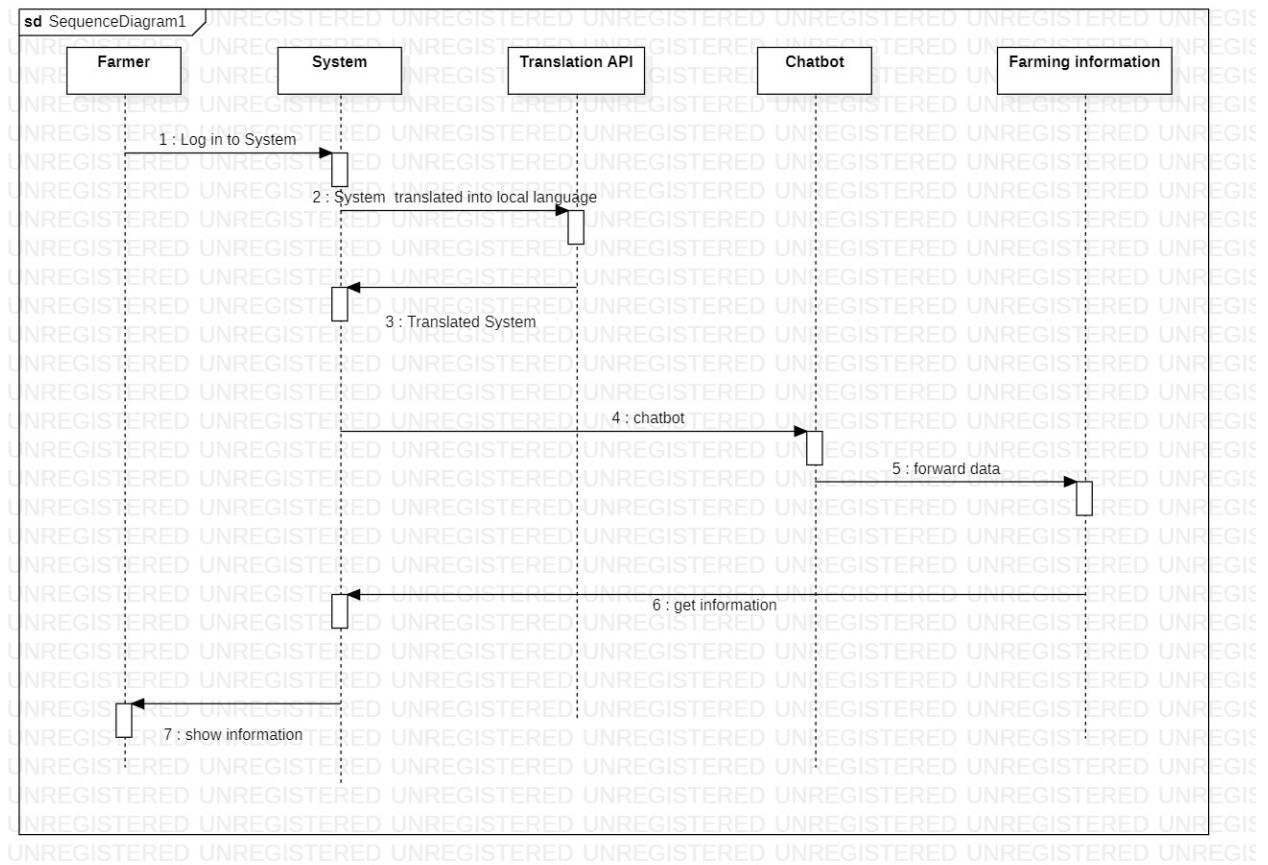
### 4.3.1 UseCase Diagram

Use case diagram shows the interaction between Use case which represents system functionality and actor which represent the people or system.



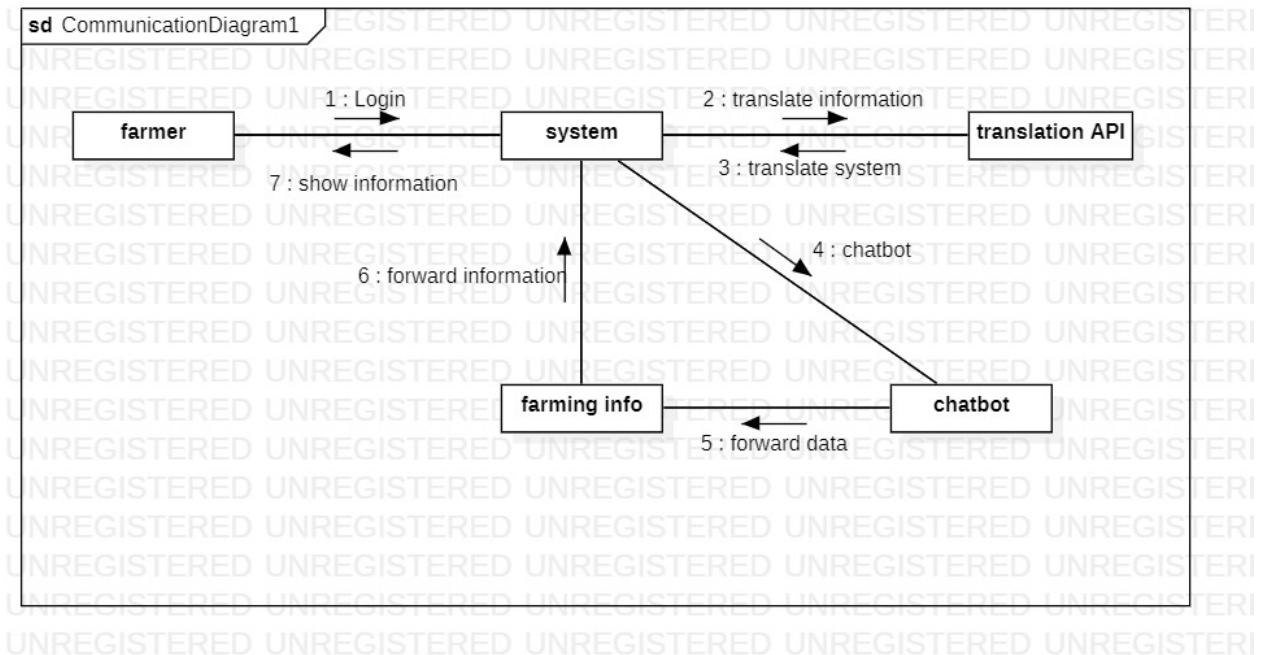
### 4.3.2 Sequence Diagram

The sequence diagram shows the flow of functionality through Use case. A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.



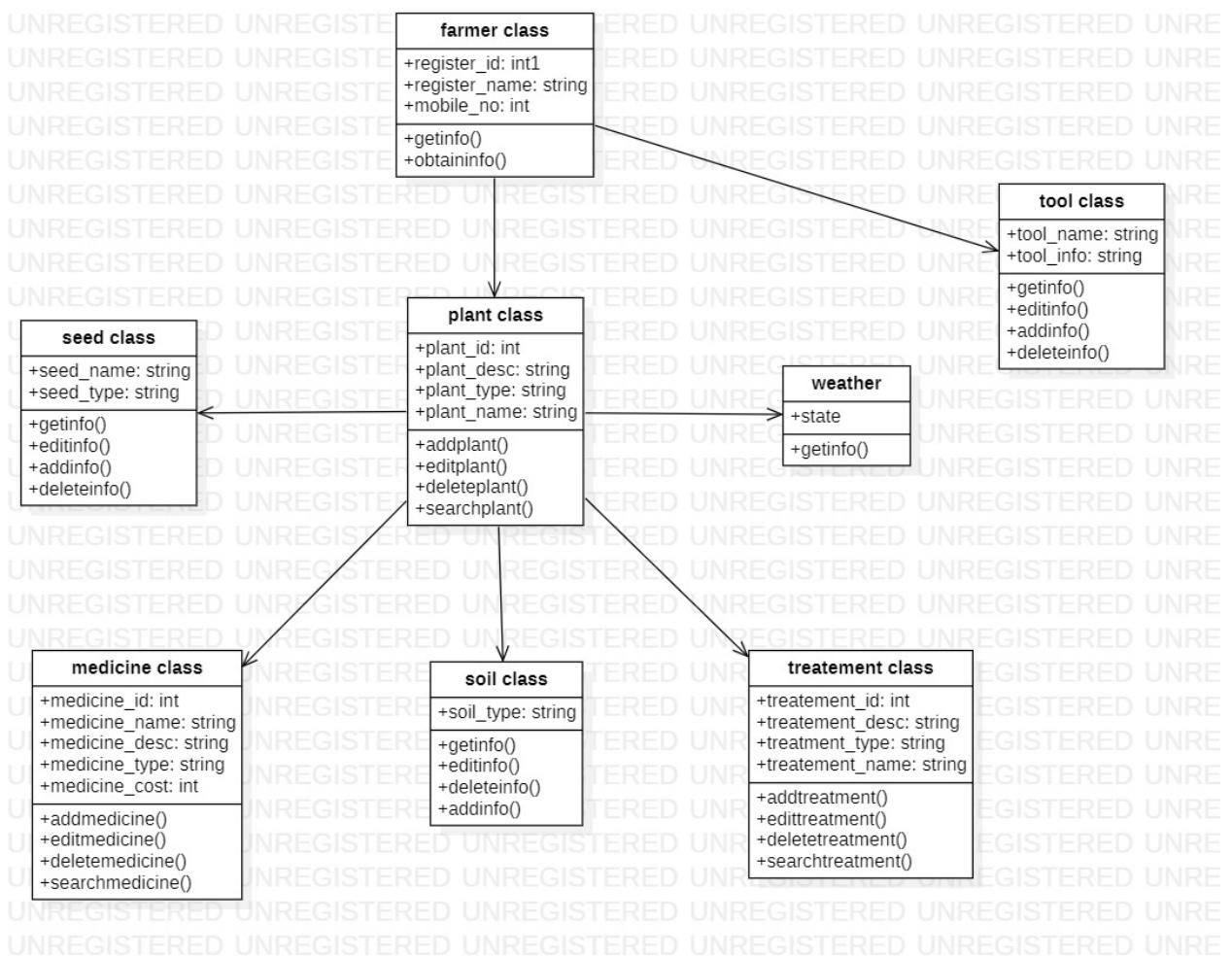
### 4.3.3 Collaboration Diagram

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.



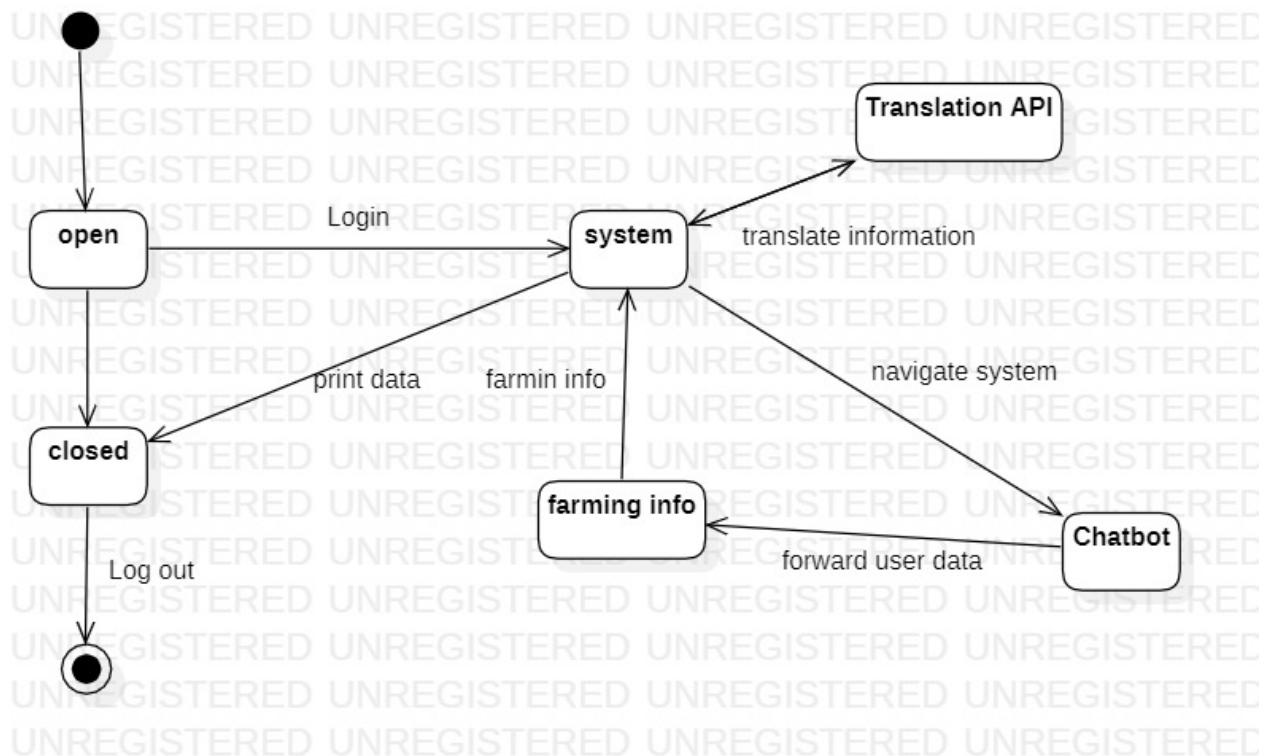
#### 4.3.4 Class Diagram

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.



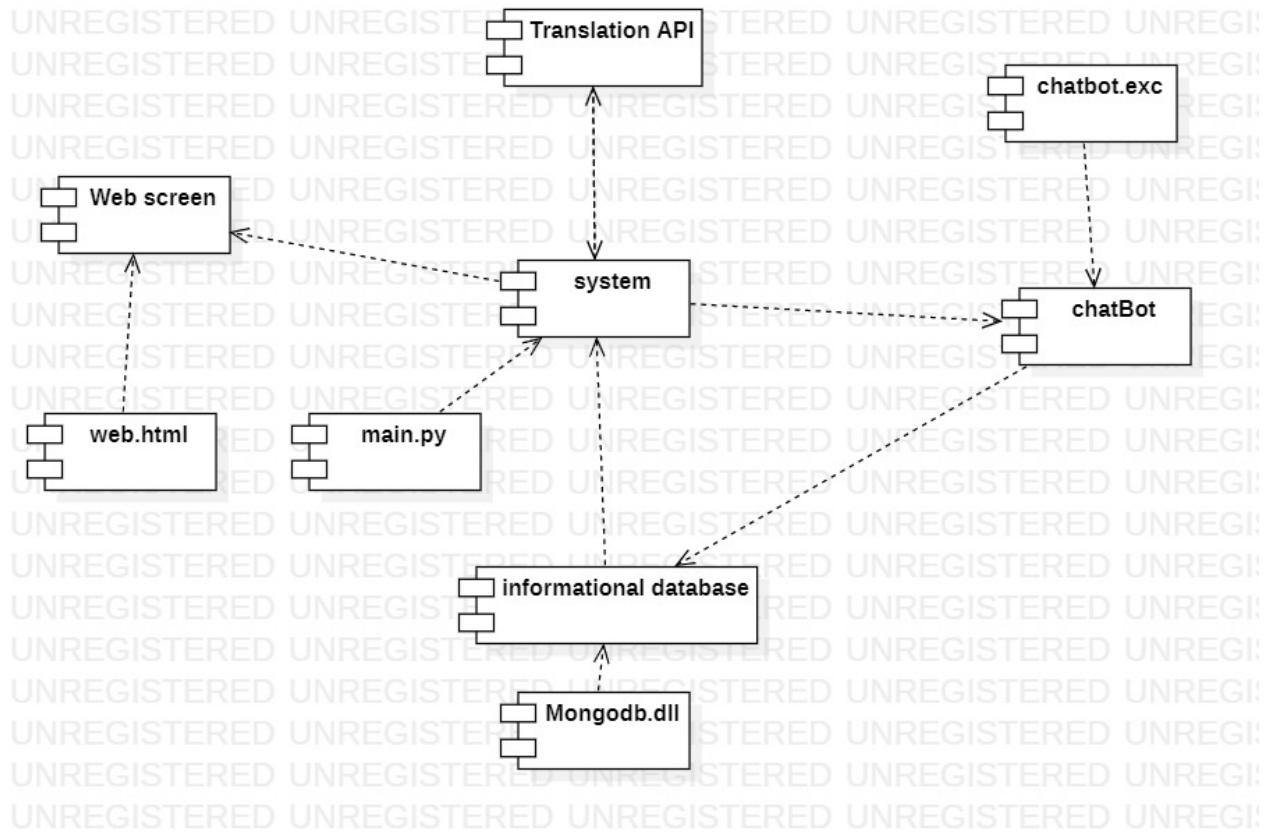
#### 4.3.5 State Transition Diagram

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system. A State Chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.



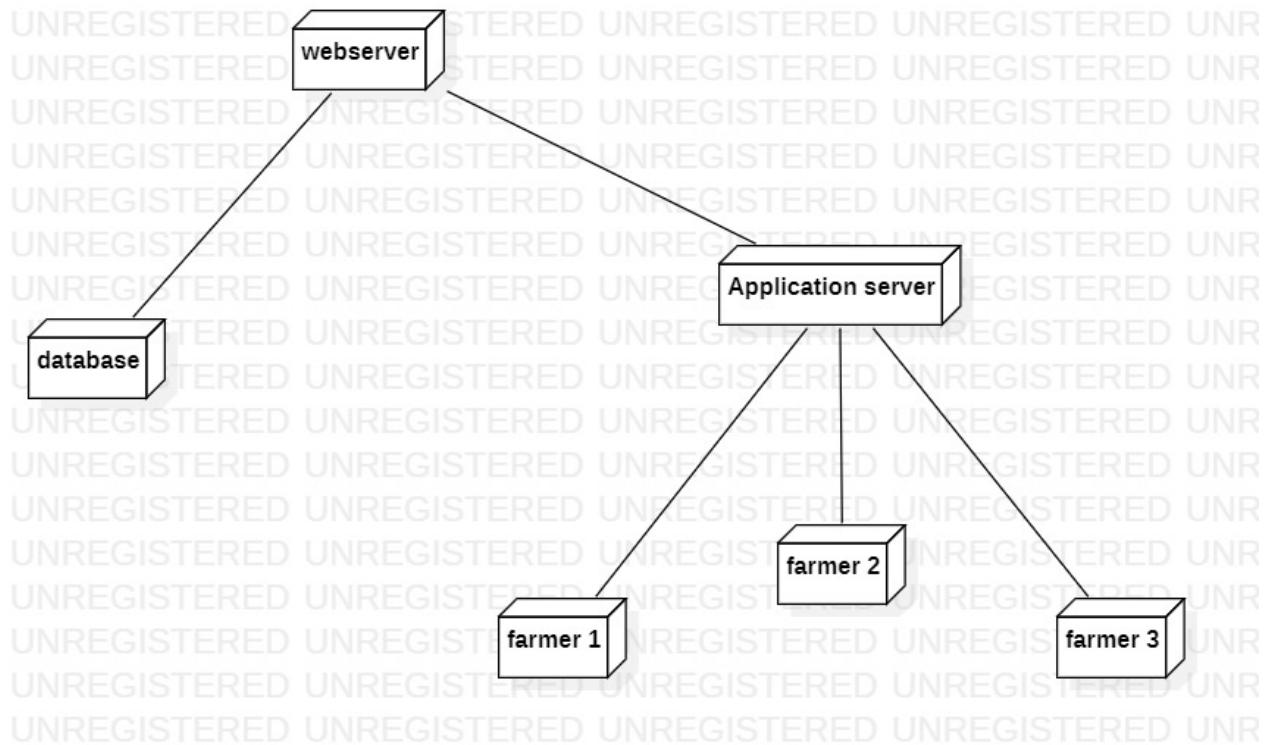
#### 4.3.6 Component Diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development.



#### 4.3.7 Deployment Diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system.



#### 4.3.8 Summary

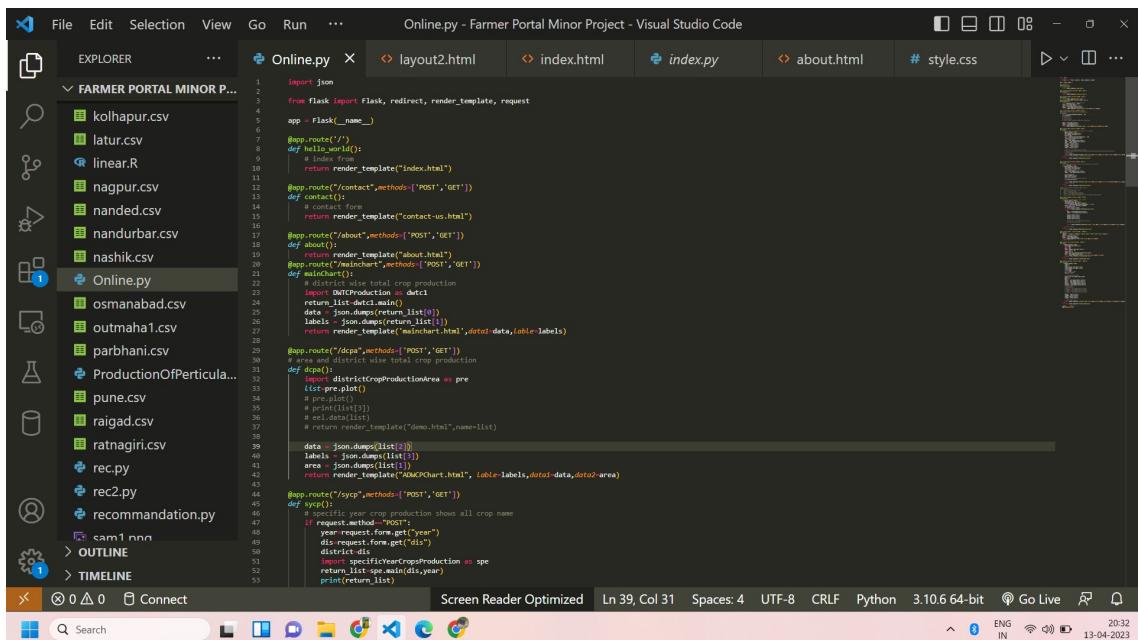
Detailed design of project has been described in this chapter including the Data Flow Diagrams and the UML Diagram explaining all the design details of the project. Conclusion of the project has been explained in the next chapter.

# Chapter 5

## Coding and Implementation

The Systematic Farming Portal is a web-based application, which revolutionizes the way traditional farming. The website provides a flexible and easy-to-use environment on desktops as well as portable devices like smart phones/tablets for the users to achieve their respective objective.

### 5.1 Define Intents



```
File Edit Selection View Go Run ... Online.py - Farmer Portal Minor Project - Visual Studio Code
EXPLORER Online.py layout2.html index.html about.html style.css ...
1 import json
2 from flask import Flask, redirect, render_template, request
3
4 app = Flask(__name__)
5
6 @app.route('/')
7 def index():
8     return render_template("index.html")
9
10 # index from
11
12 @app.route("/contact", methods=['POST', 'GET'])
13 def contact():
14     if request.method == "POST":
15         return render_template("contact-us.html")
16
17 @app.route("/about", methods=['POST', 'GET'])
18 def about():
19     return render_template("about.html")
20
21 @app.route("/districtwise", methods=['POST', 'GET'])
22 def districtwise():
23     # district wise total crop production
24     # get district wise area
25     return list(district.main())
26     data = json.dumps(return_list())
27     labels = json.dumps(return_list())
28
29     render_template("district.html", data=data, labels=labels)
30
31 @app.route("/dcpa", methods=['POST', 'GET'])
32     # area wise / district wise total crop production
33     # get districtCropProductionArea @ pre
34     # list=dcpa()
35     # pre=dcpa()
36     # print(list[1])
37     # print(list[2])
38     # return render_template("demo.html", name=list)
39
40     data = json.dumps(list[1])
41     labels = json.dumps(list[1])
42     areas = json.dumps(list[1])
43
44     render_template("dcpa.html", labels=labels, data=data, areas=areas)
45
46 @app.route("/syp", methods=['POST', 'GET'])
47 def syp():
48     if request.method == "POST":
49         year = request.form.get("year")
50         district_id = request.form.get("dis")
51
52         specificYearCropProduction as spe
53         print(spe)
54         print(district_id)
55         print(year)
56
57         print(return_list)
```

Figure 5.1: Screenshot

```

File Edit Selection View Go Run ... Online.py - Farmer Portal Minor Project - Visual Studio Code
EXPLORER ... Online.py layout2.html index.html index.py about.html style.css
FARMER PORTAL MINOR P...
kolhapur.csv
latur.csv
linear.R
nagpur.csv
nanded.csv
nandurbar.csv
nashik.csv
Online.py
osmanabad.csv
outmaha1.csv
parbhani.csv
raigad.csv
ratnagiri.csv
rec.py
rec2.py
recommendation.py
sam1.pnn
> OUTLINE
> TIMELINE
Screen Reader Optimized Ln 39, Col 31 Spaces: 4 UTF-8 CRLF Python 3.10.6 64-bit Go Live
Search 2033 ENG IN 13-04-2023

```

Figure 5.2: Screenshot

```

File Edit Selection View Go Run ... Online.py - Farmer Portal Minor Project - Visual Studio Code
EXPLORER ... Online.py layout2.html index.html index.py about.html style.css
FARMER PORTAL MINOR P...
kolhapur.csv
latur.csv
linear.R
nagpur.csv
nanded.csv
nandurbar.csv
nashik.csv
Online.py
osmanabad.csv
outmaha1.csv
parbhani.csv
ProductionOfParticularCropYears as POPCIV
return_list = POPCIV.main(dis, cropname)
print(return_list)
else:
    data = json.dumps(return_list[0])
    labels = json.dumps(return_list[1])
    maxpro_return_list[0]
    maxyear_return_list[0]
    minpro_return_list[0]
    minyear_return_list[0]
    # bar_no = json.dumps(return_list[2])
    # print(return_list[2])
    return render_template("POPCIVChart.html", data=data, labels=labels, maxpro=maxpro, maxyear=maxyear, minpro=minpro, minyear=minyear, cropname=cropname)
    # return "Hello"
else:
    return render_template("POPCIVSelective.html")
#app.route("/chart", methods=["GET", "POST"])
def chart():
    months = ["January", "February", "March", "April", "May", "June", "July", "August"]
    values = [10, 5, 8, 7, 9, 5, 12]
    data = json.dumps(values)
    labels = json.dumps(labels)
    return render_template("chart2.html", data=data, labels=labels)
# return "Hello"
#app.route("/rec", methods=["GET", "POST"])
def rec():
    if request.method == "POST":
        dis = request.form.get("dis")
        global dist
        dist = dis
        session["dist"] = dist
        session["season"] = season
        rec2 = rec
        return render_template("recYearProductionCal.html", season=season)
    data = json.dumps(return_list)
    return render_template("recReceiveCrop.html", data=data, dis=dis)
else:
    return render_template("recReceiveOS.html")
#app.route("/recrop", methods=["GET", "POST"])
def recrop():
    if request.method == "POST":
        crop1=>
Screen Reader Optimized Ln 39, Col 31 Spaces: 4 UTF-8 CRLF Python 3.10.6 64-bit Go Live
Search 2034 ENG IN 13-04-2023

```

Figure 5.3: Screenshot

The screenshot shows the Visual Studio Code interface. The title bar reads "Online.py - Farmer Portal Minor Project - Visual Studio Code". The Explorer panel on the left lists several CSV files (kolhapur.csv, latur.csv, linear.R, nagpur.csv, nanded.csv, nandurbar.csv, nashik.csv, Online.py, osmanabad.csv, outmaha1.csv, parbhani.csv, ProductionOfParticulars..., pune.csv, raigad.csv, ratnagiri.csv, rec.py, rec2.py, recommendation.py, sam1.ppt) and Python files (index.py, layout2.html, Online.py, about.html, style.css). The Online.py file is open in the code editor, showing Python code for a web application. The code includes imports like flask, json, and requests, and defines routes for GET and POST requests. The status bar at the bottom shows "Screen Reader Optimized" and various system details.

Figure 5.4: Screenshot

## 5.2 Required Software and Hardware for Development

To convert the system design into a working application, the selection of programming languages and Integrated Development Environments (IDEs) is needed. In this section, we will describe the programming languages and the development environments that have been used during the implementation phase. Programming languages that have been used, are HTML , CSS , Javascript for front end and PHP for back end. We had used MySQL for database . Regarding the development environments, Visual Studio Code have been used for development purposes. This section describes the hardware and Software we required for development of our project.

## 5.3 Implementation Environment

It does this by supporting the user with a wide array of tools such as a debugger, compiler and a text editor. This way a developer can edit the code, see all the different available methods and compile the code into the application, all in one place.

### 5.3.1 Programming Language

This program is written in HTML , CSS and Javascript which are the most commonly used language for making front end of portal. We had used PHP for back end which is also most

popular language for backend as it is open source code so it has an ASCII test file which is easily accessible and support community. PHP is the most popular language for developing a website. We had used MySQL for database which is also open source and easy to use.

## **5.4 Summary**

In this chapter , Implementation is discussed and in next chapter we will discuss about Testing.

# Chapter 6

## Testing

Testing is the process of evaluating a system or its component with the intent to and whether it satisfied requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirement. The testing chapter includes description of Black box/white box testing, manual testing, and test cases identification and execution. Section 6.1 include Black box, white box test description. Manual testing described in section 6.2. and Section 6.3 includes test cases identification and execution

### 6.1 Black Box Testing

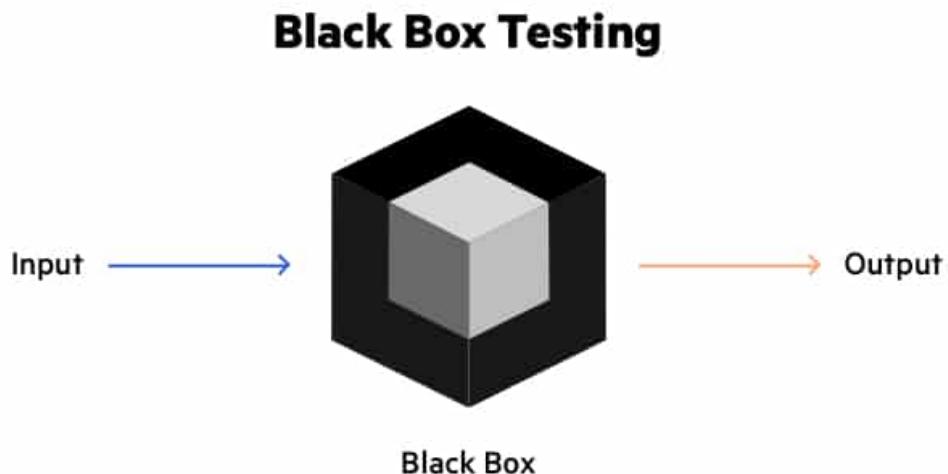


Figure 6.1: Black Box Testing

Black Box Testing is not a type of testing; it instead is a testing strategy, which does not need any knowledge of internal design or code etc. As the name "black box" suggests, no knowledge of internal logic or code structure is required. The types of testing under this strategy are totally based/focused on the testing for requirements and functionality of the work product/software application. The base of the Black box testing strategy lies in the selection of appropriate data as per functionality and testing it against the functional specifications in order to check for normal and abnormal behavior of the system.

#### **6.1.1 Functional Testing :-**

In this type of testing, the software is tested for the functional requirements. The tests are written in order to check if the application behaves as expected.

#### **6.1.2 Stress Testing :-**

The application is tested against heavy load such as complex numerical values, large number of inputs, large number of queries etc. which checks for the stress/load the applications can withstand.

#### **6.1.3 Load Testing :-**

The application is tested against heavy loads or inputs such as testing of web sites in order to find out at what point the web- site/application fails or at what.

#### **6.1.4 Recovery Testing :-**

Recovery testing is basically done in order to check how fast and better the application can recover against any type of crash or hardware failure etc. Type or extent of recovery is specified in the requirement specifications.

#### **6.1.5 User Acceptance Testing :-**

In this type of testing, the software is handed over to the user in order to find out if the software meets the user expectations and works as it is expected to.

#### **6.1.6 Alpha Testing :-**

In this type of testing, the users are invited at the development center where they use the application and the developers note every particular input or action carried out by the user. Any type of abnormal behavior of the system is noted and rectified by the developers.

### **6.1.7 Beta Testing :-**

In this type of testing, the software is distributed as a beta version to the users and users test the application at their sites. As the users explore the software, in case if any exception/defect occurs that is reported to the developers.

## **6.2 White Box Testing**

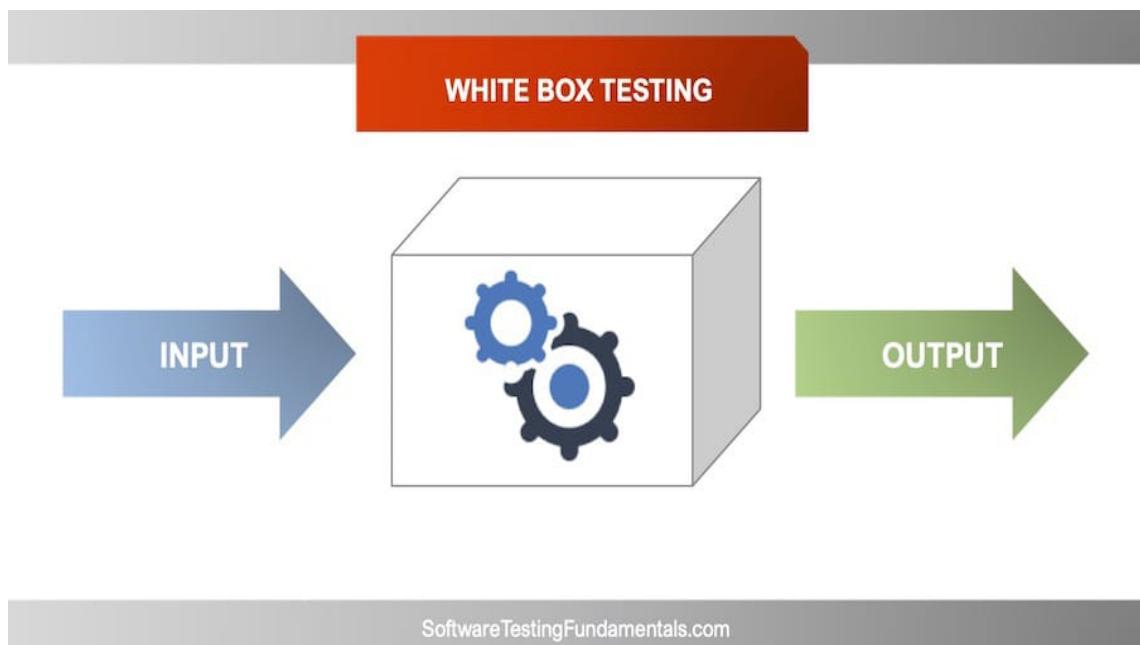


Figure 6.2: White Box Testing

White box testing strategy deals with the internal logic and structure of the code. White box testing is also called as glass, structural, open box or clear box testing. The tests written based on the white box testing strategy incorporate coverage of the code written, branches, paths, statements and internal logic of the code etc. In order to implement white box testing, the tester has to deal with the code and hence is needed to possess knowledge of coding and logic i.e. internal working of the code. White box test also needs the tester to look into the code and find out which unit/statement/chunk of the code is malfunctioning.

### **6.2.1 Unit Testing:-**

The developer carries out unit testing in order to check if the particular module or unit of code is working fine. The Unit Testing comes at the very basic level as it is carried out as and when the unit of the code is developed or a particular functionality is built.

### **6.2.2 Branch Coverage:-**

No software application can be written in a continuous mode of coding, at some point we need to branch out the code in order to perform a particular functionality. Branch coverage testing helps in validating of all the branches in the code and making sure that no branching leads to abnormal behavior of the application.

### **6.2.3 Security Testing:-**

Security Testing is carried out in order to find out how well the system can protect itself from unauthorized access, hacking cracking, any code damage etc. which deals with the code of application. This type of testing needs sophisticated testing.

### **6.2.4 Mutation Testing:-**

A kind of testing in which, the application is tested for the code that was modified after fixing a particular bug/defect. It also helps in finding out which code and which strategy of coding can help in developing the functionality effectively.

## **6.3 Manual Testing**

It is oldest and most rigorous types of testing it is performed by human sitting in front of a computer carefully going through application screens, trying various usage and input combination, comparing the results to be expected behaviour and recording and observations about the project. The manual testing corresponding to project is to test whether that system give accurate classification of dataset. Manual testing is testing of the software where tests are executed manually by a QA analyst. It is performed to discover bugs in software under development. In manual testing, the tester checks all the essential features of the given application or software. In this process, the software testers execute the test cases and generate the test reports without the help of any automation software testing tools. It is a classical method of all testing types and helps and bugs in software systems. It is generally conducted by an experienced tester to accomplish the software testing process.

- The initial investment in the Manual testing is comparatively lower.
- Manual testing is not as accurate because of the possibility of the human errors.
- Manual testing proves useful when the test case only needs to run once or twice.
- No need for programming in Manual Testing.

- While testing a small change, an automation test would require coding which could be time consuming. While you could test manually.

## 6.4 Automated Testing

In Automated Software Testing, testers write code test scripts to automate test execution. Testers use appropriate automation tools to develop the test scripts and validate the software. The goal is to complete test execution in a less amount of time. Automated testing entirely relies on the pre scripted test which runs automatically to compare actual result with the expected results. This helps the tester to determine whether or not an application performs as expected. Automated testing allows you to execute repetitive task and regression test without the intervention of manual tester. Even though all processes are performed automatically, automation requires some manual export to create initial testing scripts.

- The initial investment in the automated testing is higher.
- Automated testing is a reliable method, as it is performed by tools and scripts. There is no testing fatigue.
- Programming knowledge is a must in automation testing.
- Automation testing is useful when frequently executing the same set of test cases.
- Testing coverage can be increased because automation testing tool never forgets to check even the smallest unit.

## 6.5 Test Cases Identification and Execution

### Test Case ID 1

<b>+</b>	
<b>Test Case ID</b>	<b>1</b>
Name	Recommend Crop
Purpose	To check whether the system gives the correct crop recommendation or not.
Input	District name, Name of crop 1 & crop 2.
Expected Output	Best among the two <u>crop</u> should be recommended successfully.
Actual Output	Best among the two crop has been recommended successfully.
Result	Pass

Figure 6.3: Test Case ID 1

### Test Case ID 2

<b>+</b>	
<b>Test Case ID</b>	<b>2</b>
Name	View Analysis
Purpose	To check whether the system gives the correct crop analysis or not.
Input	District name, Name of crop 1 & crop 2, Year, Season.
Expected Output	The crop analysis should be shown successfully.
Actual Output	The crop analysis has been shown successfully.
Result	Pass

Figure 6.4: Test Case ID 2

### Test Case ID 3

<b>Test Case ID</b>	<b>3</b>
Name	Farmers Portal
Purpose	To check whether the system redirects to the farmers portal.
Input	URL of the portal or button.
Expected Output	The system should be redirected to farmers portal successfully.
Actual Output	The system has been redirected to farmers portal successfully.
Result	Pass

Figure 6.5: Test Case ID 3

## 6.6 Summary

In this chapter , Testing is described and in the next chapter Result is given.

# Chapter 7

## Result and Discussion

### 7.1 Home Page

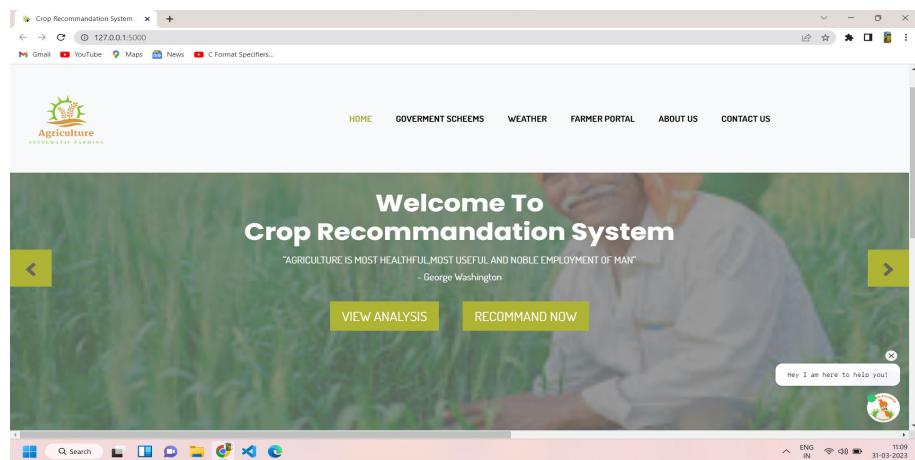


Figure 7.1: Home Page

### 7.2 Translation API

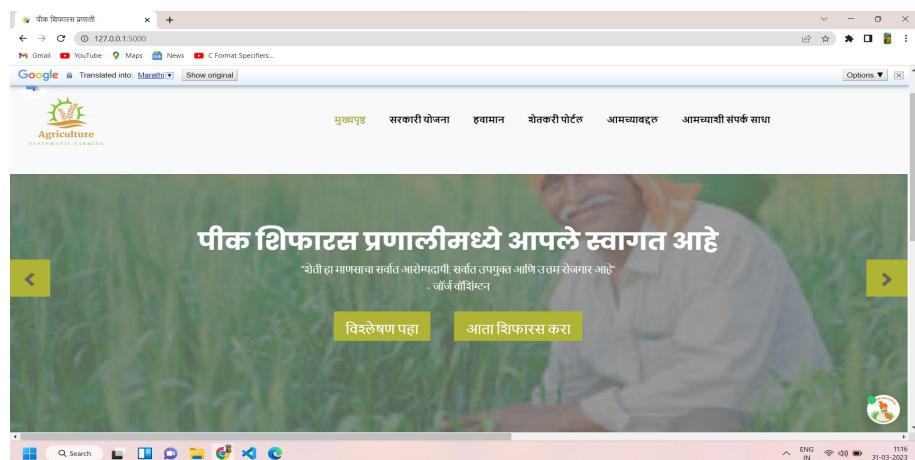


Figure 7.2: Translation API

## 7.3 Chatbot

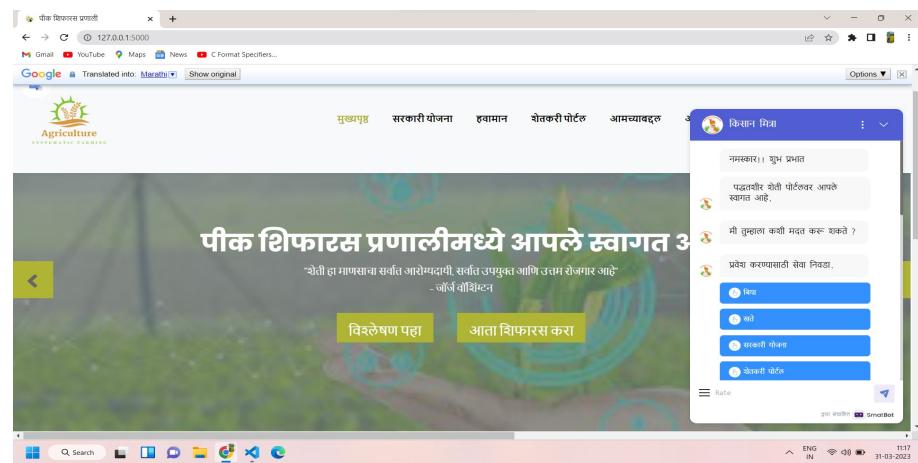


Figure 7.3: Chatbot

### 7.3.1 Goverment Scheems

Following figure show us the Goverment Scheems page.

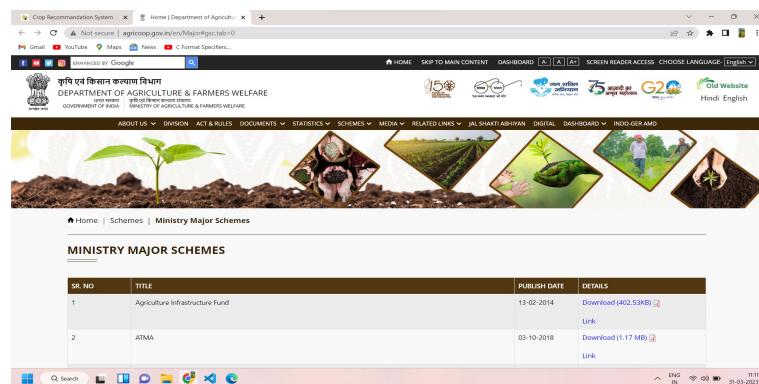


Figure 7.4: Goverment Scheems

## 7.4 Farmer Portal



Figure 7.5: Farmer Portal

## 7.5 Seeds Information

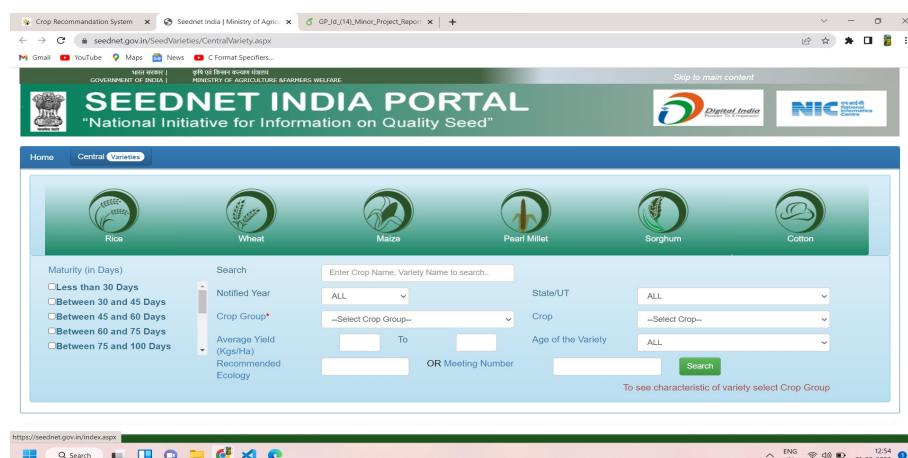


Figure 7.6: Seeds Portal

## 7.6 Fertilizer Information



Figure 7.7: Fertilizer Portal

## 7.7 Crop Recomendation System

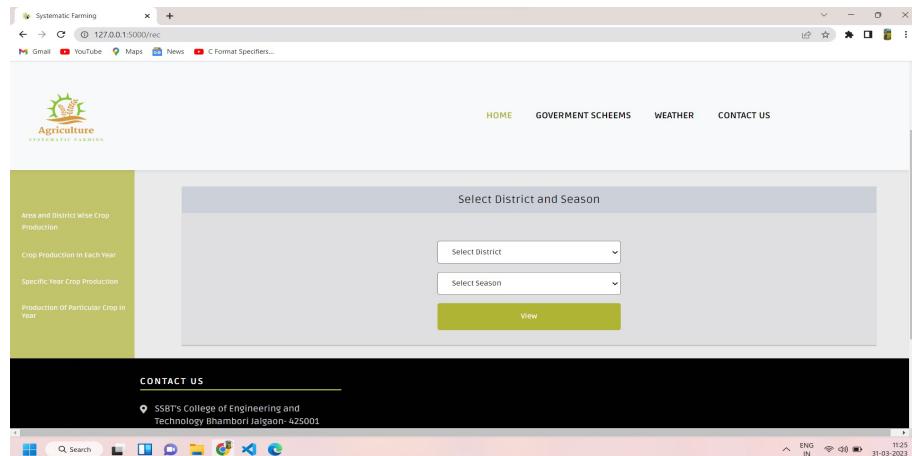


Figure 7.8: Crop Recomendation System

### 7.7.1 Crop Recomendation Result

Following figure show us the page after recomending the crop result.

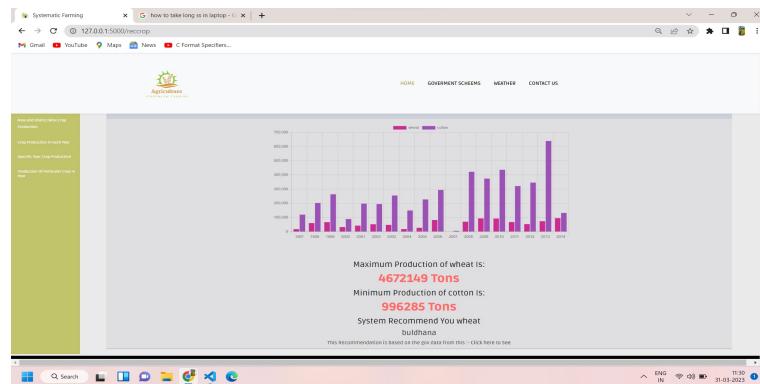


Figure 7.9: Crop Recomendation Result

## 7.8 Analysis

Here we can analysis our crop data disrtictwise,year wise,specific crop Production, etc.

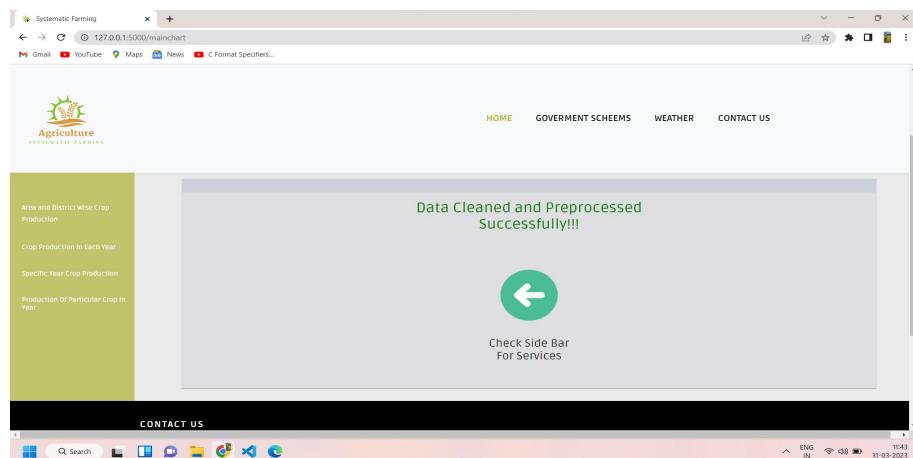


Figure 7.10: Analysis

## 7.9 Analysis of Crop Yearwise

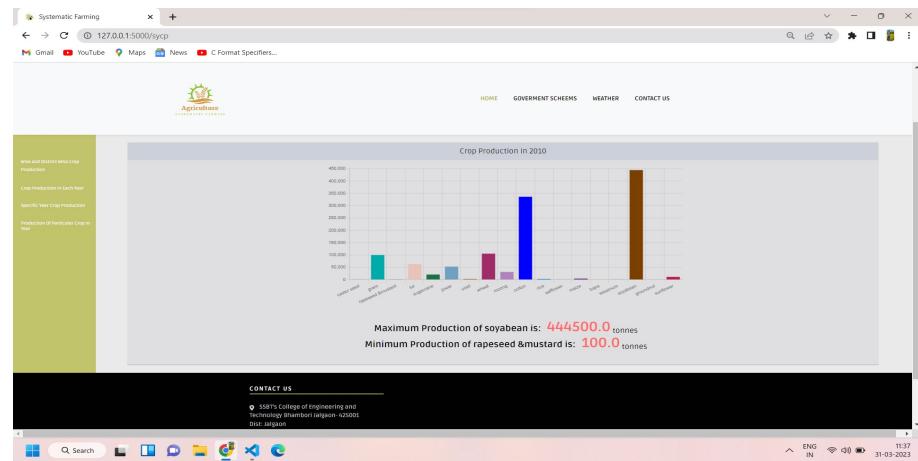


Figure 7.11: Analysis Yearwise

## 7.10 Analysis of Perticular Crop

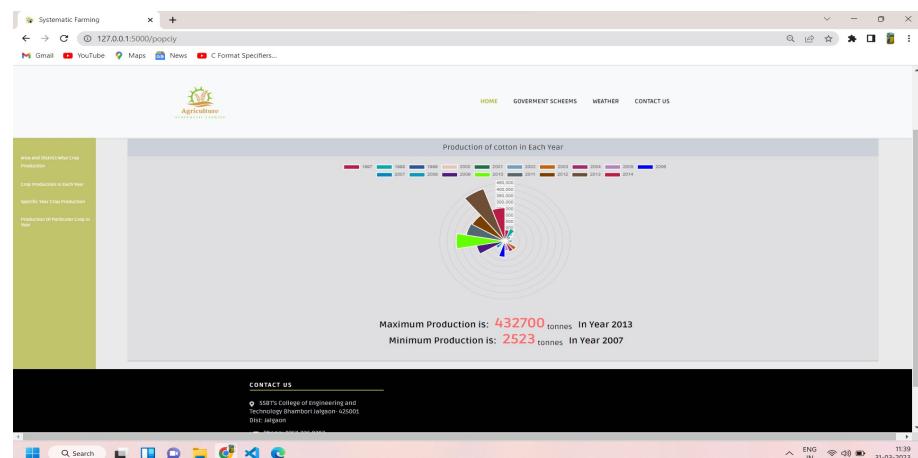


Figure 7.12: Analysis Perticular Crop

## 7.11 About us Page

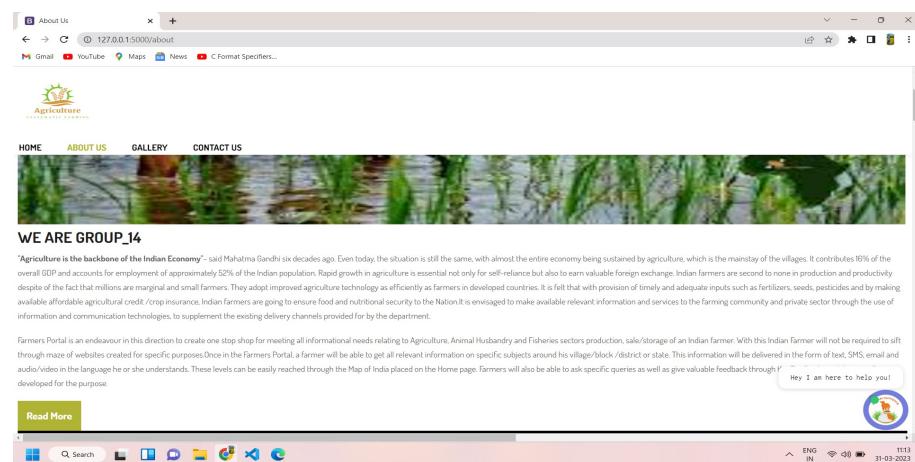


Figure 7.13: About us Page

## 7.12 Contact us Page

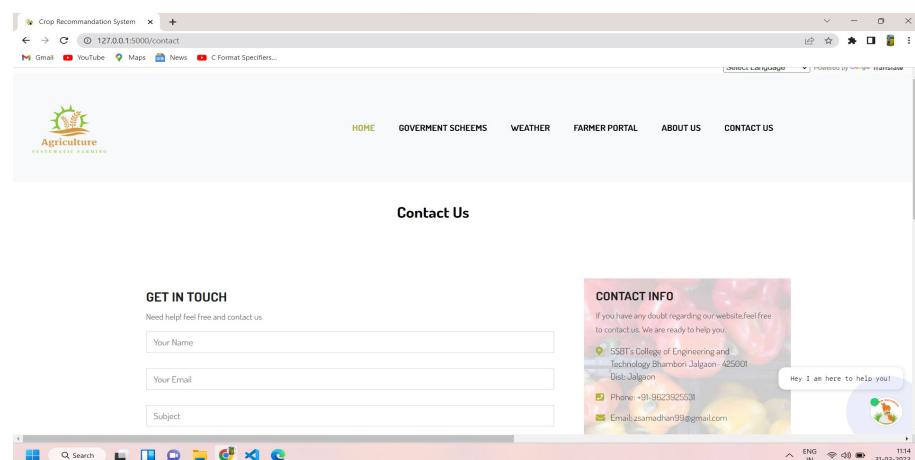


Figure 7.14: Contact us

# **Chapter 8**

## **Conclusion**

This Project had paved the way for an efficient farming. Also the system is supported by cloud computing thereby reducing the price aspect of the system tremendously. This system is proposed to replace the existing system where the farmer has to suffer a lot. Also the main advantage of this project is that it uses Information Technology. The user only needs smartphone or computer with an internet connection. Typically the farmer needs the correct guidance so that there is an increase in the crop production. The farmers need such kind of help so that the loss of farming is reduced, there is more production of crop and they are benefited. Also the main advantage of this project is that it uses Information Technology. The soil type and weather forecasting facilities are provided, which makes the farmer decide which crop should they take at which time.

All these things help the farmers in many aspects in effective and profitable manner. So there is an evolution in old farming system with this digital way. Also, farmers are learning to use new methodologies for farming rather than old techniques.

# Bibliography

1. Aref, S., and M.M. Wander. 1997. Long-term trends of corn yield and soil organic matter in different crop sequences and soil fertility treatments on the Morrow Plots. *Advances in Agronomy* 62: 153–197.
2. Baker, D., and D. Norman. 1990. The Farming Systems Research and Extension Approach to Small Farmer Development. In *Agroecology and Small Farm Development*, ed. M. Altieri and S. Hecht. pp. 91–104. CRC Press: Boca Raton, FL.
3. Carter, M.R., S. Andrews, and L.E. Drinkwater. 2004. Systems Approaches for Improving Soil Quality. In *Managing Soil Quality-Challenges in Modern Agriculture*, ed. P. Schjonning, B.T. Christensen, and S. Elmholz. pp. 261–281. CAB International: Oxford, UK.
4. Chase, L., and V. Grubinger. 2014. *Food, Farms and Community*. University of New Hampshire Press: Durham, NH.
5. Clark, M.S., W.R. Horwath, C. Shennan, K.M. Scow, W.T. Lanini, and H. Ferris. 1999. Nitrogen, weeds and water as yield-limiting factors in conventional, low-input, and organic tomato systems. *Agriculture Ecosystems and Environment* 73: 257–270.
6. Dean, J.M., M.C. Mescher, and C.M. Demoraes. 2009. Plant-rhizobia mutualism influences aphid abundance on soybean. *Plant and Soil* 323(1): 187–196.
7. Drinkwater, L.E. 2002. Cropping systems research: reconsidering agricultural experimental approaches. *HortTechnology* 12(3): 355–361.
8. Drinkwater, L.E. 2009. Ecological Knowledge: Foundation for Sustainable Organic Agriculture. In *Ecology in Organic Farming Systems*, ed. C.A. Francis. American Society of Agronomy: Madison, WI.