AI BASED FERTILIZERS RECOMMENDATION SYSTEM FOR DISEASE PREDICTION (NALAIYA THIRAN)

ON

PROFESSIONAL READINESS FOR INNOVATION , EMPLOYABILITY AND ENTREPRENEURSHIP

Submitted by

TEAM ID: PNT2022TMID38677

TEAM MEMBERS

E.L. Vincy Rashitha	420419205021
S. Dhatchayani	420419205004
K.Saraswathi	420419205015
K.Divya	420419205701

In partial fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

ANNA UNIVERSITY-CHENNAI 600 025

NOVEMBER 2022

BONAFIDE CERTIFICATE

Certified that this report titled **"AI BASED FERTILIZER RECOMMENDATION SYSTEM FOR DISEASE PREDICTION"** is a Bonafide work of **S.DHATCHAYANI(420419205004) , K.SARASWATHI(420419205015) , K.DIVYA(420419205701) , E.L.VINCY RASHITHA(420419205021**). Who carried out the work under my supervision.

SIGNATURE	SIGNATURE	SIGNATURE
Dr.A.BHUVANESWARI M.E., Ph.D.,	**Dr . N. ELAMATHI M.E., Ph.D.,**	**Mr .M. EZHILVENDAN B.Tech., M.E.,**
HOD and SPOC	**Faculty Mentor**	**Internal Evaluator**
Professor	**Assistant professor**	**Assistant professor**
Department of Information Technology	Department of Information Technology	Department of Information Technology
Adhiparasakthi Engineering College Melmaruvathur-603 319	Adhiparasakthi Engineering College Melmaruvathur-603 319	Adhiparasakthi Engineering College Melmaruvathur-603 319

CERTIFICATION OF EVALUATION

College Code/Name : 4204/ Adhiparasakthi Engineering College
Branch/Semester :Information Technology/07

S.NO	NAME OF THE STUDENT AND REGISTER NUMBER	TITLE OF THE PROJECT	NAME OF THE SUPERVISOR WITH DESIGNATION
1. **2.** **3.** **4.**	DHATCHAYANI S (420419205004) SARASWATHI K (420419205015) DIVYA K (420419205701) VINCY RASHITHA E L (420419205021)	AI BASED FERTILIZER RECOMMENDATION SYSTEM FOR DISEASE PREDICTION	Dr .N. ELAMATHI M.E., Ph.D., ASSISTANT PROFESSOR, Department of Information Technology, Adhiparasakthi Engineering College, Melmaruvathur-603319

Submitted for the projectwork and viva-voce held on………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

It is indeed a great pleasure and proud privilege to acknowledge the help and support we received from the positive minds around us in making this endeavor a successful one.

The spiritual blessings of His Holiness **ARULTHIRU AMMA** and the devout guidance **THIRUMATHI**

AMMA have undoubtedly taken us to the path of victory in completing this project.The infrastructural support with all kinds of lab facilities have been a motivating factor in our completion of project work, all because of our **Correspondent Sakthi Dr. G. B. SENTHIL KUMAR** with great pleasure we take this opportunity to thank him. From the academic side the support from **Dr. J. RAJA M.E.,Ph.D., Principal**has encouraged us to work hard and attain this goal of completing the project.

We thank our **Head of the Department and Dr. A.BHUVANESHWARI M.E., Ph.D.,** who has given us both moral and technical support adding experience to the job we have under taken.

With great pleasure we thank our **Faculty Mentor Dr. N.ELAMATHI M.E., Ph.D.**, and **Internal Evaluator, Mr.M.EZHILVENDAN, B.Tech., ME.,** Department of Information Technology who guided and helped us in crossing obstacles in the

path to our glory.

We take immense pleasure in thanking our parents who gave us moral support to complete this project. We also thank other staff members, and non-teaching staff members of computer Center and library, who have given their constant support and motivation in our endeavor.

ABSTRACT

A fertilizer recommendation is the research-based set of guidelines, or management practices, for supplying fertilizer to the crop to achieve yield and quality goals (economic) in a manner that minimizes nutrient losses to the environment.

Agriculture is the most important sector in today’s life , Most plants are affected by a wide variety of bacterial and fungal disease. Diseases on plants placed a major constraint on the production and a major threat to Food security

Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality .In recent years, the number of diseases on plants and the degree of harm caused has increased due to the variation in pathogen varieties, changes in cultivation methods ,and inadequate plant protection techniques.

An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant.

The most efficient technique is used to identify the diseases and suggest the precautions that can be taken for those diseases.

The Two dataset name fruit dataset and vegetable dataset are collected. The collected datasets are trained and tested with deep learning neural network named Convolutional Neural Networks(CNN).

The proposed method uses SVM to classify tree leaves, identify the disease and suggest the fertilizer. The proposed method is compared with the existing CNN based leaf disease prediction. The proposed SVM technique gives a better result when compared to existing CNN.

This technique is used to identify the plant diseases and precaution measures are taken for the current situation to control the disease in the plants and fertilizer is recommended.

This helps in the plant growth and to enrich the soil nutrients and maintain the soil level

Because of the quick and efficient production, this increases harvest yields, making food affordable and reduces the costs of production.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION:

1.1 PROJECT OVERVIEW :

Agriculture is the most important sector in today's life. Most plants are affected by a wide variety of bacterial and fungal diseases. Diseases on plants placed a major constraint on the production and a major threat to food security. Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality. In recent years, the number of diseases on plants and the degree of harm caused has increased due to the variation in pathogen varieties, changes in cultivation methods, and inadequate plant protection techniques.

An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

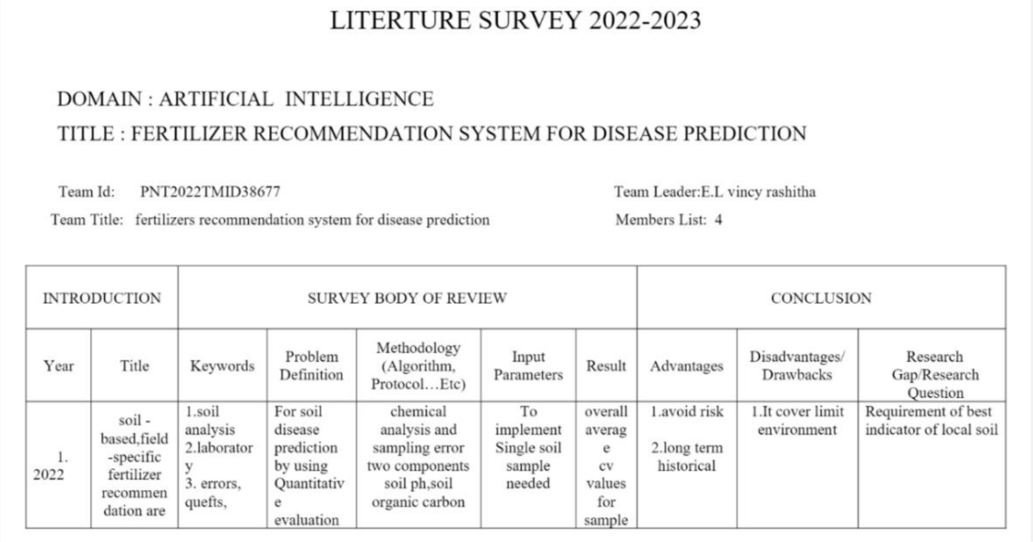
1.2 PUPRPOSE:

- This project is used to test the fruits and vegetables samples and identify the different diseases. Also, this project recommends fertilizers for predicted diseases.
- The disease-based similarity measure is used for fertilizer recommendation.
- Detection and recognition of plant diseases using machine learning are very efficient in providing symptoms of identifying diseases at its earliest.
- The measurements of fertilizers suggested based on disease severity.
- It allows us to predict which crops would be appropriate for a given climate.
- The proposed method uses SVM to classify tree leaves, identify the disease and suggest the fertilizer.
- The proposed method is compared with the existing CNN based leaf disease prediction. The proposed SVM technique gives a better result

2. LITERATURE SURVEY:

2.1EXISTING PROBLEM:

In our case ,when a pathogen that is already present or which invades successfully to plant host tissues and cells results in the plant disease. It is an important to fix the problem because plant diseases reduce the amount of food available to humans by ultimately interfering with crop yields .This can cause inadequate food to humans which result in starvation or death in worst cases.



LITERTURE SURVEY 2022-2023

DOMAIN : ARTIFICIAL INTELLIGENCE

TITLE : FERTILIZER RECOMMENDATION SYSTEM FOR DISEASE PREDICTION

Team Id: PNT2022TMID38677 Team Leader:E.L vincy rashitha

Team Title: fertilizers recommendation system for disease prediction Members List: 4

INTRODUCTION		SURVEY BODY OF REVIEW					CONCLUSION		
Year	Title	Keywords	Problem Definition	Methodology (Algorithm, Protocol…Etc)	Input Parameters	Result	Advantages	Disadvantages/ Drawbacks	Research Gap/Research Question
1. 2022	soil -based,field -specific fertilizer recommendation are	1.soil analysis 2.laboratory 3. errors, quefts,	For soil disease prediction by using Quantitative evaluation	chemical analysis and sampling error two components soil ph,soil organic carbon	To implement Single soil sample needed	overall average cv values for sample	1.avoid risk 2.long term historical	1.It cover limit environment	Requirement of best indicator of local soil

Figure2.1 Literature Survey

2.2 REFERENCES:

[1]Usman Ahmed, Jerry Chun-Wei Lin, Gautam Srivastava” A Nutrient recommendation system for soil fertilization based on evolutionary computation”, 2021.

[2]Tanmay Thorat, B.K. Patle, Sunil Kumar Kashyap,”Intelligent insecticide and fertilizer recommendation system based on TPF - CNN for smart farming ”,2022.

[3]R. Neela , P. Nithya, “Fertilizer recommendation system for disease prediction in tree leave”,2019.

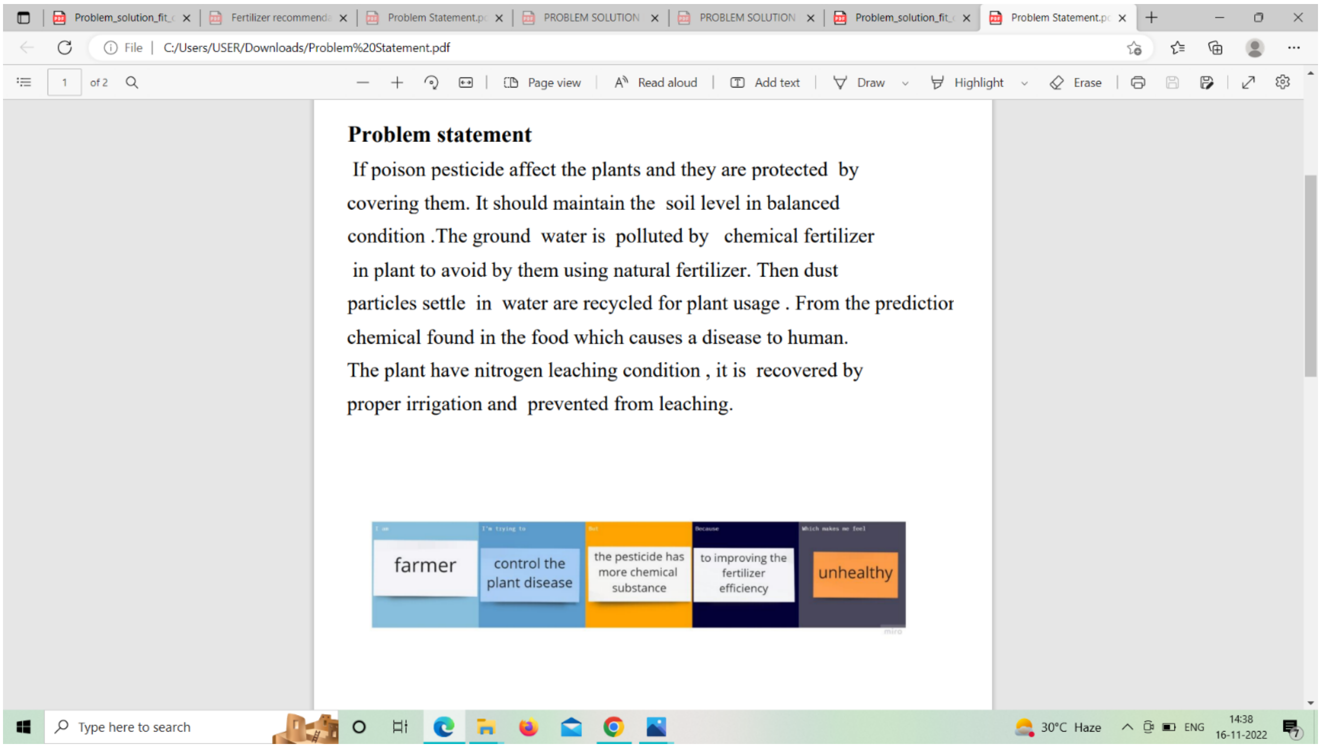
[4] Archana Chougule, Vijay Kumar Jha , Debajyoti Mukhopadhyay, “Crop suitability/fertilizers recommendation using data mining technique”,2019.

[5] Folasade Olubusola Isinkaye Department of Computer Science, Ekiti State University, Ado-Ekiti, Nigeria Emmanuel Damilola Erute Department of Computer Science, Ekiti State University, Ado-Ekiti, Nigeria” A smart phone based plant disease detection and treatment recommendation system”,2022.

2.3 PROBLEM STATEMENT DEFINITION

The Problem statement Comprises set of questions which the project seeks to address .It identifies the current state and future state and any gaps between the two. The Problem raised here in this project is:

1. Where does the problem affect?
2. What is the impact of the issues?
3. What would happen if we didn’t solve the problem?
4. When does the issue occur?
5. Where is the issue occurring ?



Problem statement

If poison pesticide affect the plants and they are protected by covering them. It should maintain the soil level in balanced condition .The ground water is polluted by chemical fertilizer in plant to avoid by them using natural fertilizer. Then dust particles settle in water are recycled for plant usage . From the predictior chemical found in the food which causes a disease to human.
The plant have nitrogen leaching condition , it is recovered by proper irrigation and prevented from leaching.

I am	I'm trying to	But	Because	Which makes me feel
farmer	control the plant disease	the pesticide has more chemical substance	to improving the fertilizer efficiency	unhealthy

Figure 2.3 Problem Statements

To grow , plant required nitrogen compounds from the soil , be provided by fertilizer the release of harmful greenhouse gases into the atmosphere and the eutrophication of our waterways.

1. Cope with climate change , soil erosion and biodiversity lose.
2. Satisfy consumers changing tastes and expectations .
3. Invest in form productivity

3. IDEATION AND PROPOSED SOLUTION:

3.1 EMPATHY MAP:

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers.

1.What do they Think and Feel?

2.What do they See?

3.What do they Say and Do?

4.What do they Hear?

5.Customer Pain?

6.Customer Gain?

"The action of understanding, being aware of, being sensitive to, and vicariously experiencing the feelings, thoughts, and experience “.

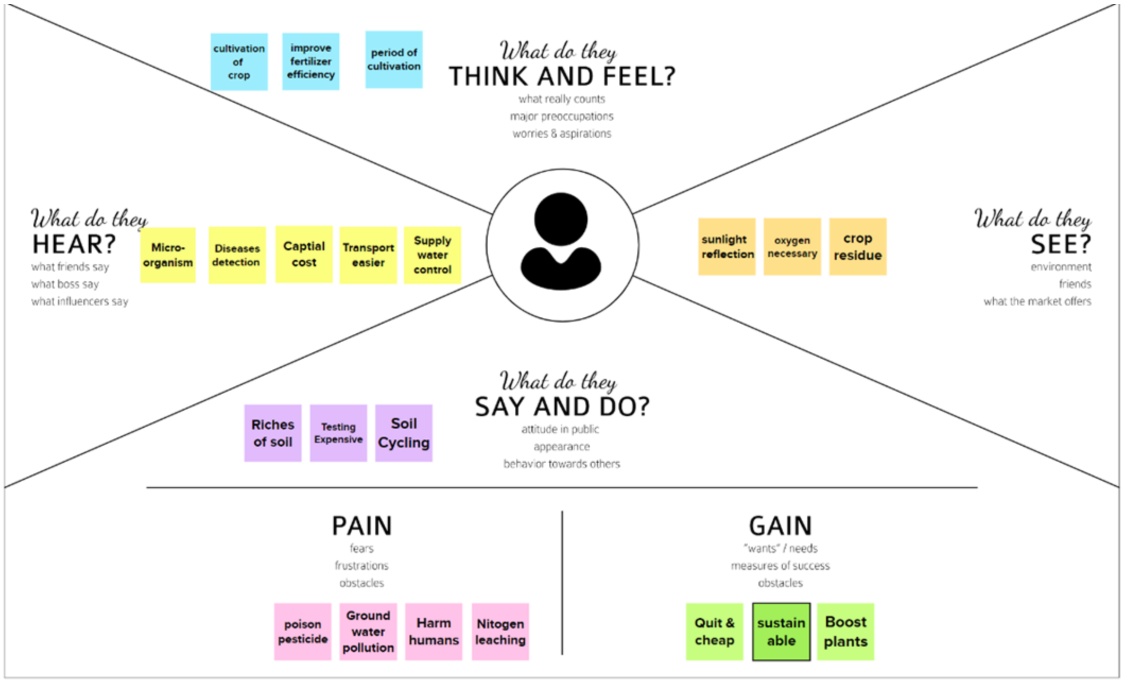
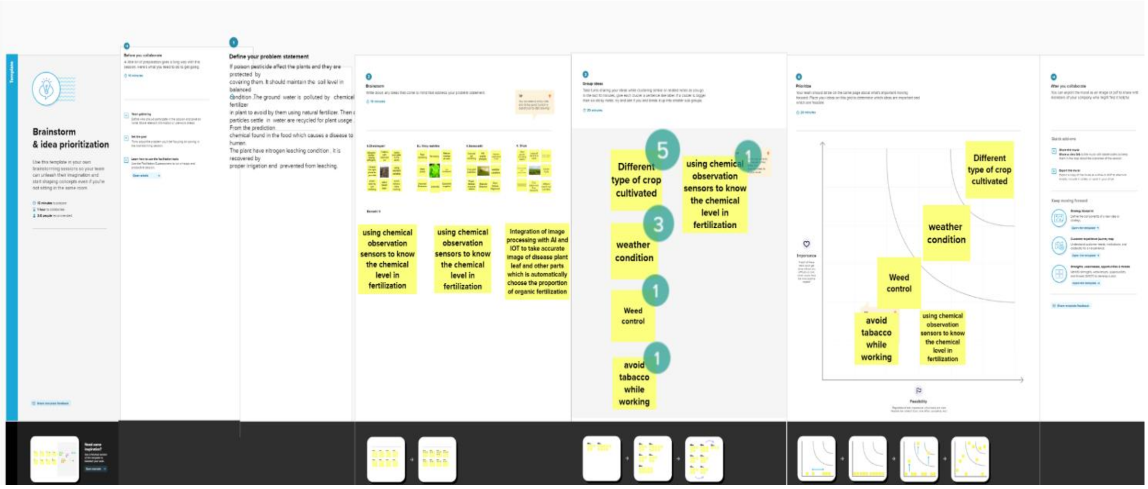


Figure 3.1 Empathy Map

3.2 IDEATION AND PROPOSED SOLUTION:

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.



Brainstorm & idea prioritization

Define your problem statement

If poison pesticide affect the plants and they are protected by covering them. It should maintain the soil level in balanced Condition .The ground water is polluted by chemical fertilizer in plant to avoid by them using natural fertilizer. Then particles settle in water are recycled for plant usage . From the prediction chemical found in the food which causes a disease to human. The plant have nitrogen leaching condition . It is recovered by proper irrigation and prevented from leaching.

using chemical observation sensors to know the chemical level in fertilization

using chemical observation sensors to know the chemical level in fertilization

Integration of image processing with AI and IOT to take accurate image of disease plant leaf and other parts which is automatically choose the proportion of organic fertilization

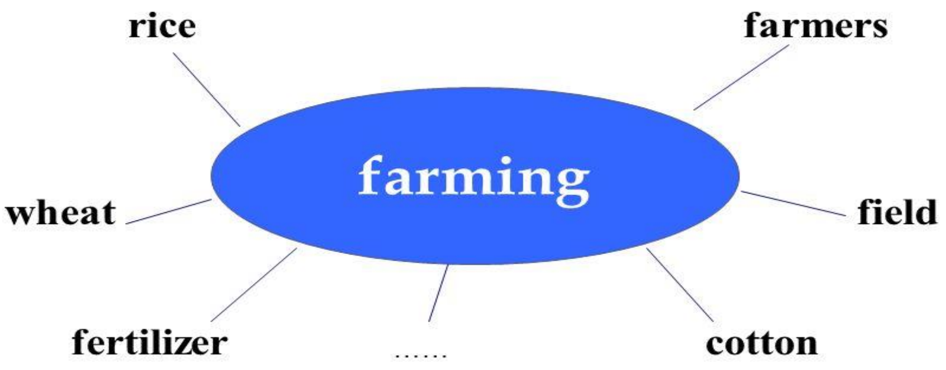
Template
Before you collaborate
Brainstorm
Group ideas
5
Different type of crop cultivated
1
using chemical observation sensors to know the chemical level in fertilization
3
weather condition
1
Weed control
1
avoid tabacco while working
Prioritize
Different type of crop cultivated
weather condition
Weed control
avoid tabacco while working
using chemical observation sensors to know the chemical level in fertilization
Importance
Feasibility
After you collaborate
Need some inspiration?

Figure 3.2 Brainstorming

OVERVIEW OF BRAINSTORMING:

Brainstorming

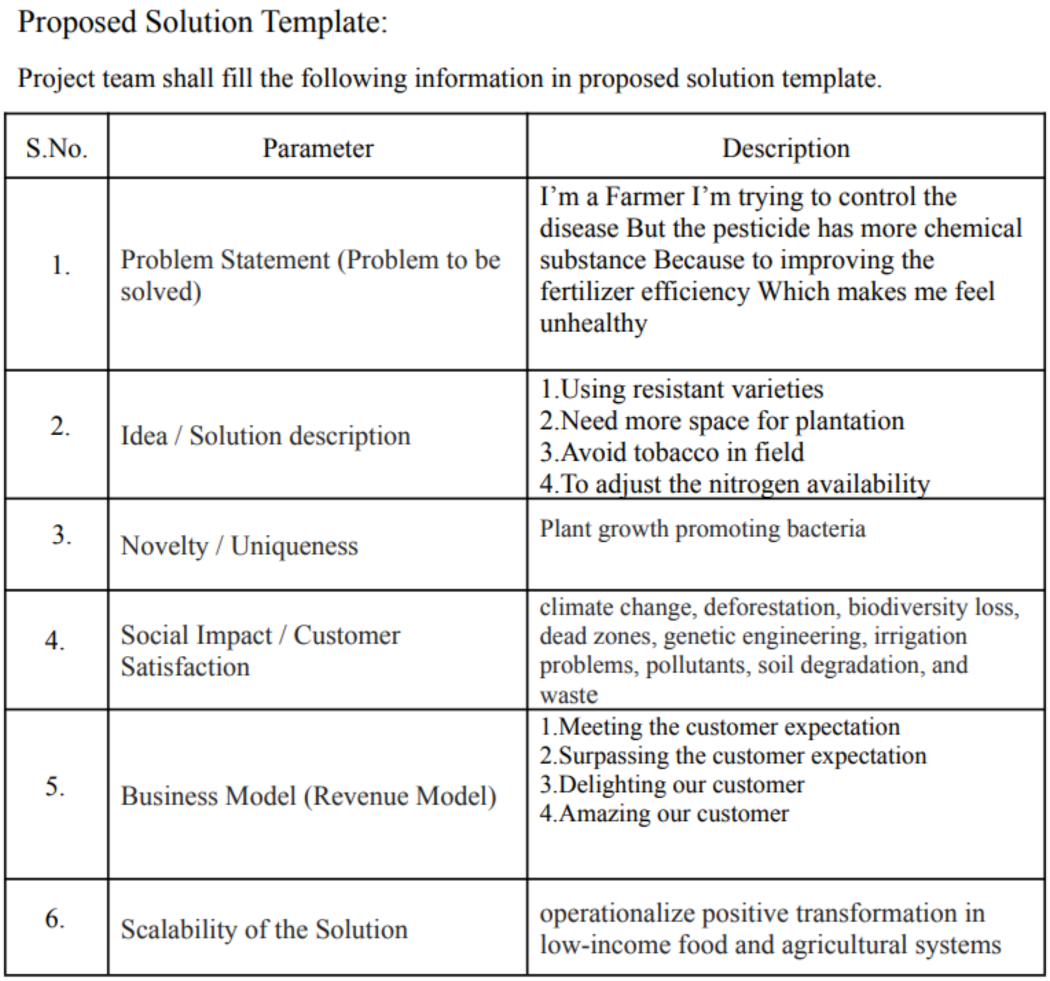
What will you think of when you see the word "*farming*"?



3.3PROPOSED SOLUTION:

The proposed solution should relate the current situation to a desired result and describe the benefits that will accrue when the desired result is achieved. So, begin your proposed solution by briefly describing this desired result. Improve Farming Productivity .

- Implementation of land reforms.
- Smart water management.
- Adopt genetically modified crops
- Develop high-yield crops



Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

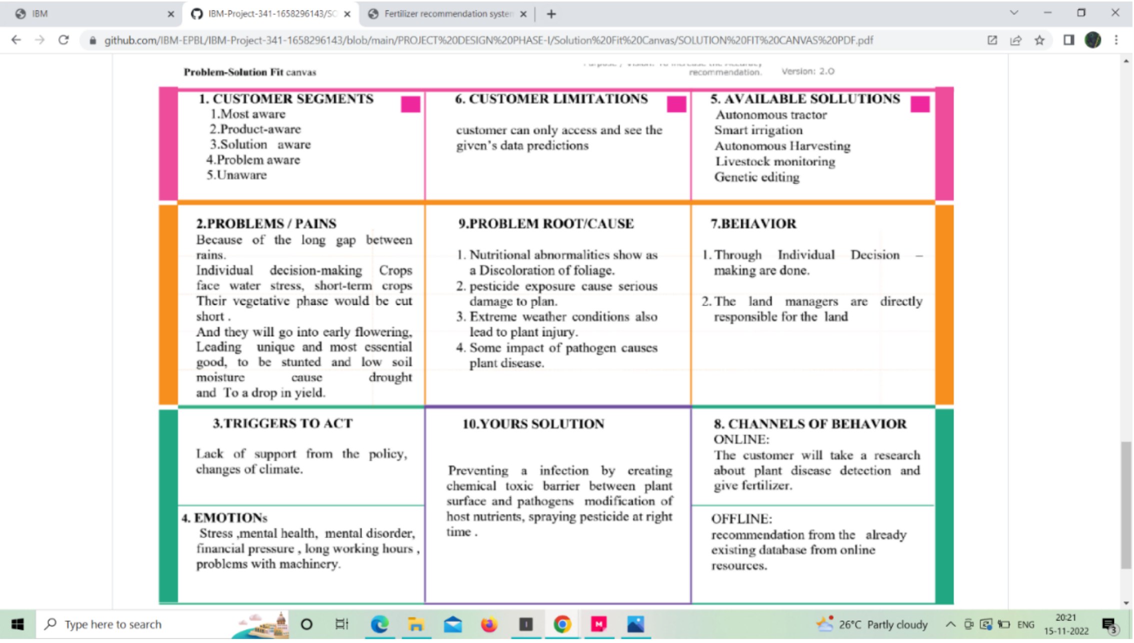
S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	I'm a Farmer I'm trying to control the disease But the pesticide has more chemical substance Because to improving the fertilizer efficiency Which makes me feel unhealthy
2.	Idea / Solution description	1.Using resistant varieties 2.Need more space for plantation 3.Avoid tobacco in field 4.To adjust the nitrogen availability
3.	Novelty / Uniqueness	Plant growth promoting bacteria
4.	Social Impact / Customer Satisfaction	climate change, deforestation, biodiversity loss, dead zones, genetic engineering, irrigation problems, pollutants, soil degradation, and waste
5.	Business Model (Revenue Model)	1.Meeting the customer expectation 2.Surpassing the customer expectation 3.Delighting our customer 4.Amazing our customer
6.	Scalability of the Solution	operationalize positive transformation in low-income food and agricultural systems

Figure 3.3 Proposed Solution

3.4 PROBLEM SOLUTION FIT:

Problem-Solution canvas is a tool for entrepreneurs, marketers and corporate innovators, which helps them identify solutions with higher chances for solution adoption, reduce time spent on solution testing and get a better overview of current situation.

- Customer Segments
- Problems/Pains
- Triggers To Act
- Emotions
- Available Solutions
- Customer Limitations
- Behavior
- Channels of Behavior


IBM
IBM-Project-341-1658296143/SC
Fertilizer recommendation syster
github.com/IBM-EPBL/IBM-Project-341-1658296143/blob/main/PROJECT%20DESIGN%20PHASE-I/Solution%20Fit%20Canvas/SOLUTION%20FIT%20CANVAS%20PDF.pdf
Type here to search
26°C Partly cloudy
ENG
20:21
15-11-2022

Problem-Solution Fit canvas

recommendation. Version: 2.0

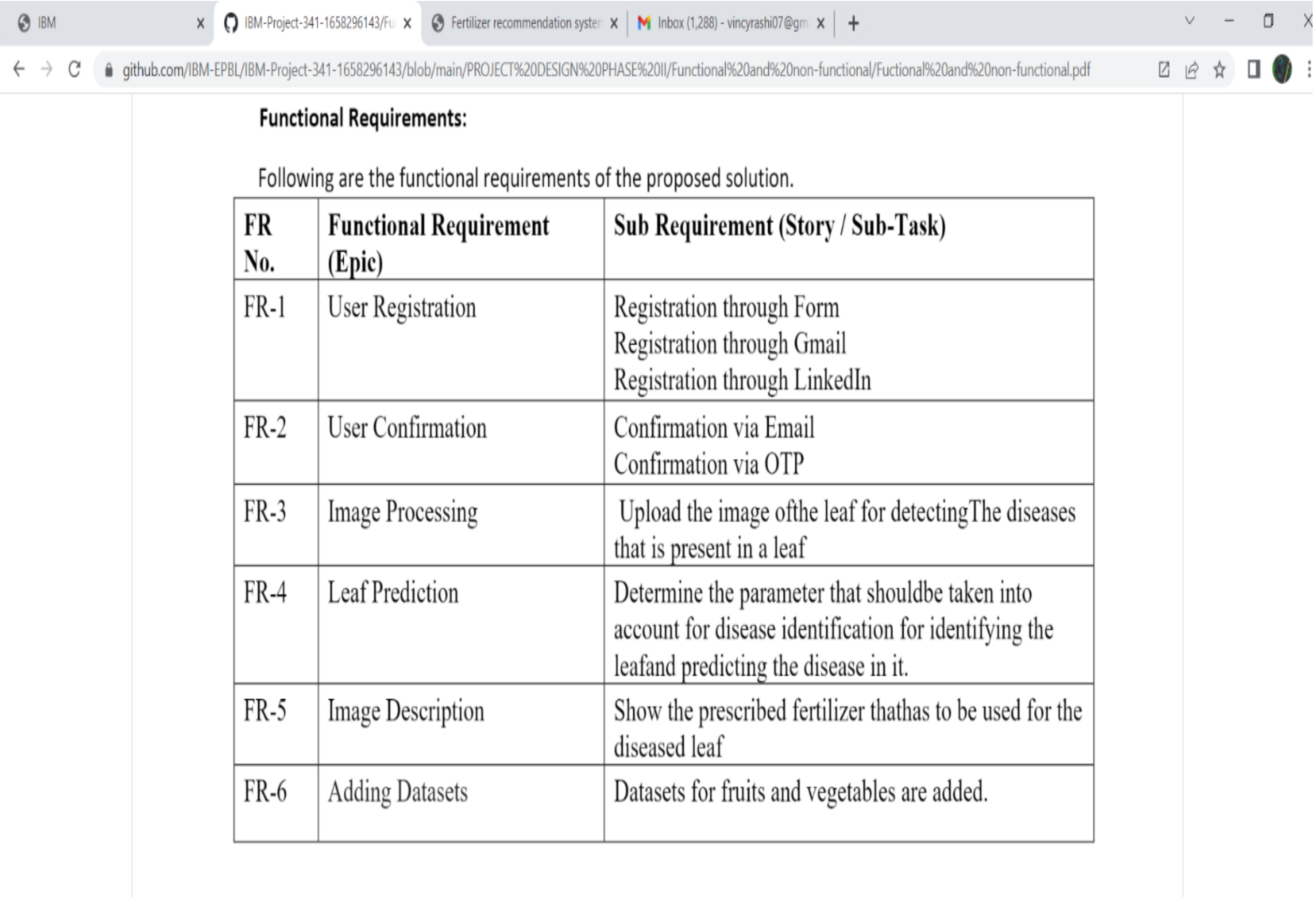
1. CUSTOMER SEGMENTS 1.Most aware 2.Product-aware 3.Solution aware 4.Problem aware 5.Unaware	**6. CUSTOMER LIMITATIONS** customer can only access and see the given's data predictions	**5. AVAILABLE SOLLUTIONS** Autonomous tractor Smart irrigation Autonomous Harvesting Livestock monitoring Genetic editing
2.PROBLEMS / PAINS Because of the long gap between rains. Individual decision-making Crops face water stress, short-term crops Their vegetative phase would be cut short . And they will go into early flowering, Leading unique and most essential good, to be stunted and low soil moisture cause drought and To a drop in yield.	**9.PROBLEM ROOT/CAUSE** 1. Nutritional abnormalities show as a Discoloration of foliage. 2. pesticide exposure cause serious damage to plan. 3. Extreme weather conditions also lead to plant injury. 4. Some impact of pathogen causes plant disease.	**7.BEHAVIOR** 1. Through Individual Decision – making are done. 2. The land managers are directly responsible for the land
3.TRIGGERS TO ACT Lack of support from the policy, changes of climate. **4. EMOTIONs** Stress ,mental health, mental disorder, financial pressure , long working hours , problems with machinery.	**10.YOURS SOLUTION** Preventing a infection by creating chemical toxic barrier between plant surface and pathogens modification of host nutrients, spraying pesticide at right time .	**8. CHANNELS OF BEHAVIOR** ONLINE: The customer will take a research about plant disease detection and give fertilizer. OFFLINE: recommendation from the already existing database from online resources.

Figure 3.4 Problem Solution Fit

4.REQUIREMENT ANALYSIS:

4.1FUNCTIONAL REQUIREMENTS:

Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describe all the cases where the system uses the functional requirements, these are captured in use cases. Generally functional requirements drive the application architecture of a system


IBM
IBM-Project-341-1658296143/Fu
Fertilizer recommendation systen
Inbox (1,288) - vincyrashi07@gm
github.com/IBM-EPBL/IBM-Project-341-1658296143/blob/main/PROJECT%20DESIGN%20PHASE%20II/Functional%20and%20non-functional/Fuctional%20and%20non-functional.pdf

Functional Requirements:

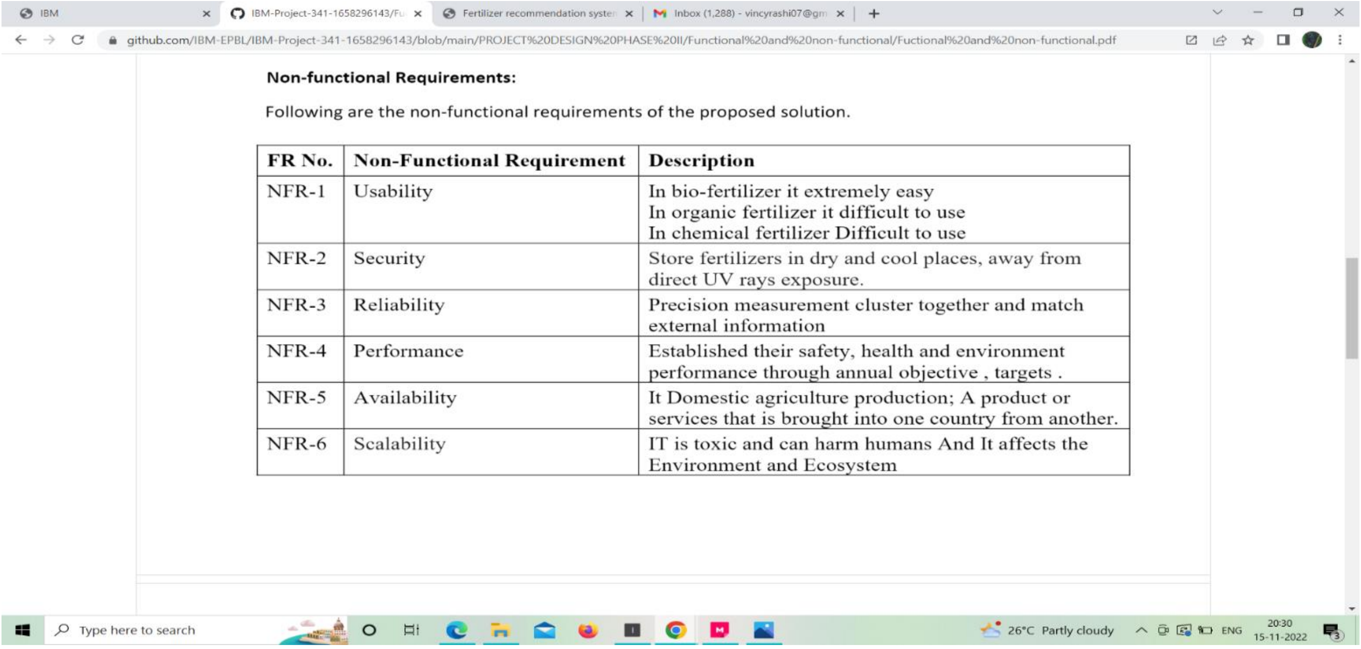
Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Image Processing	Upload the image ofthe leaf for detectingThe diseases that is present in a leaf
FR-4	Leaf Prediction	Determine the parameter that shouldbe taken into account for disease identification for identifying the leafand predicting the disease in it.
FR-5	Image Description	Show the prescribed fertilizer thathas to be used for the diseased leaf
FR-6	Adding Datasets	Datasets for fruits and vegetables are added.

Figure 4.1 Functional Requirement

4.2 NON FUNCTIONAL REQUIREMENTS:

A Non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behavior.


IBM
IBM-Project-341-1658296143/Fu
Fertilizer recommendation syster
Inbox (1,288) - vincyrashi07@gm
github.com/IBM-EPBL/IBM-Project-341-1658296143/blob/main/PROJECT%20DESIGN%20PHASE%20II/Functional%20and%20non-functional/Fuctional%20and%20non-functional.pdf
Type here to search
26°C Partly cloudy
ENG
20:30
15-11-2022

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	In bio-fertilizer it extremely easy In organic fertilizer it difficult to use In chemical fertilizer Difficult to use
NFR-2	Security	Store fertilizers in dry and cool places, away from direct UV rays exposure.
NFR-3	Reliability	Precision measurement cluster together and match external information
NFR-4	Performance	Established their safety, health and environment performance through annual objective , targets .
NFR-5	Availability	It Domestic agriculture production; A product or services that is brought into one country from another.
NFR-6	Scalability	IT is toxic and can harm humans And It affects the Environment and Ecosystem

Figure 4.2 Non – Functional Requirement

5.PROJECT DESIGN:

5.1 DATA FLOW DIAGRAMS:

A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various subprocesses the data moves through. DFDs are built using standardized symbols and notation to describe various entities and their relationships.

Dataflow template allows to easily share your ideas and flow of process for the particular task It is used for processing and enriching stream data for use case such as analysis , machine learning. Dataflow is a easy to understand and it supports both stream and batch processing.

Data Flow Diagrams:

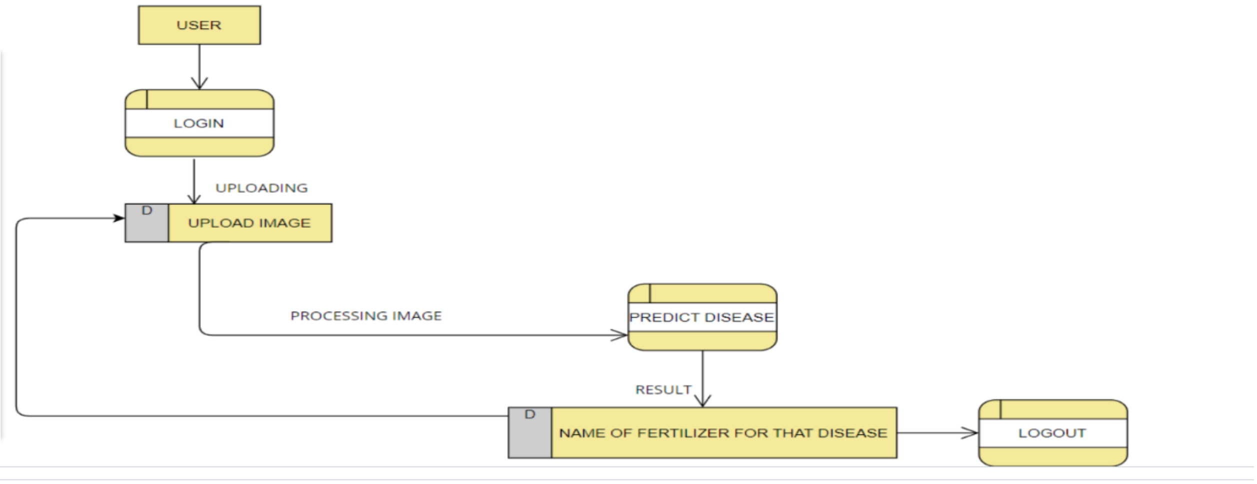


Figure 5.1 Data Flow Diagram

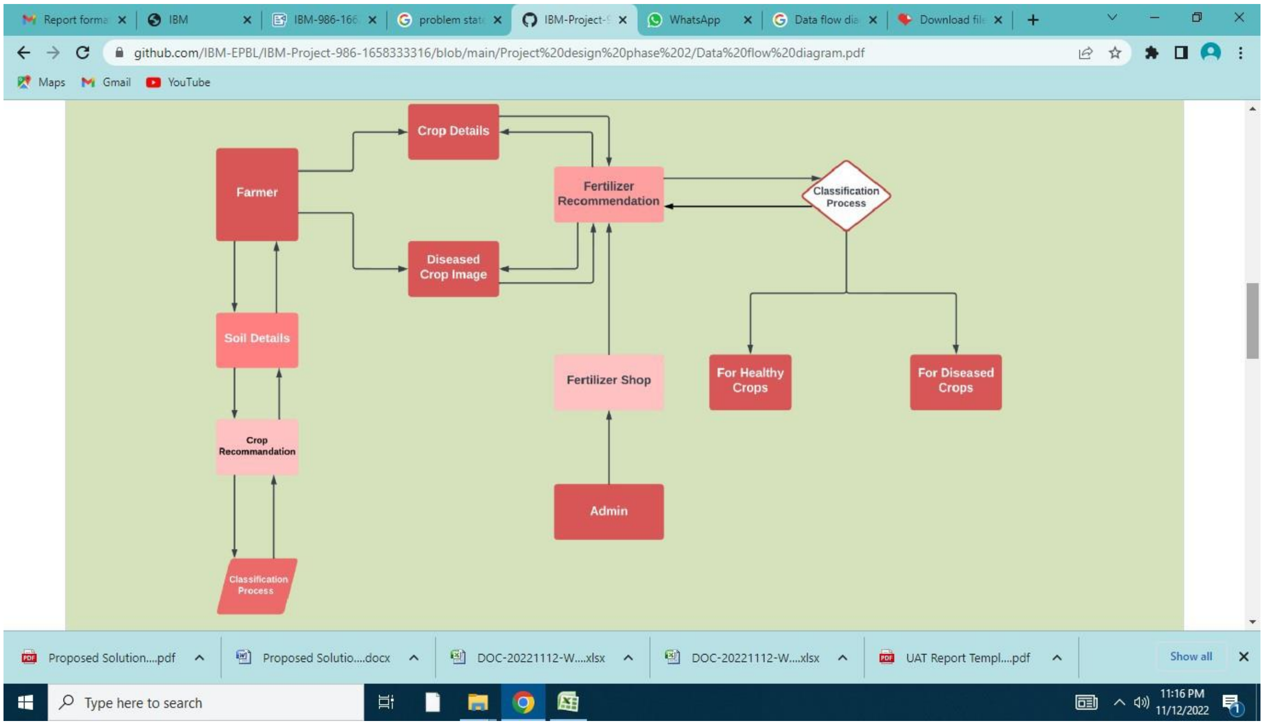


Figure 5.1.1 Data Flow Overview

5.2 SOLUTION AND TECHNICAL ARCHITECTURE:

A solution architecture (SA) is ancestral description archiof a specific solution. SAs combine guidance from different enterprise architecture viewpoints (business, information and technical), as well as from the enterprise solution architecture (ESA).

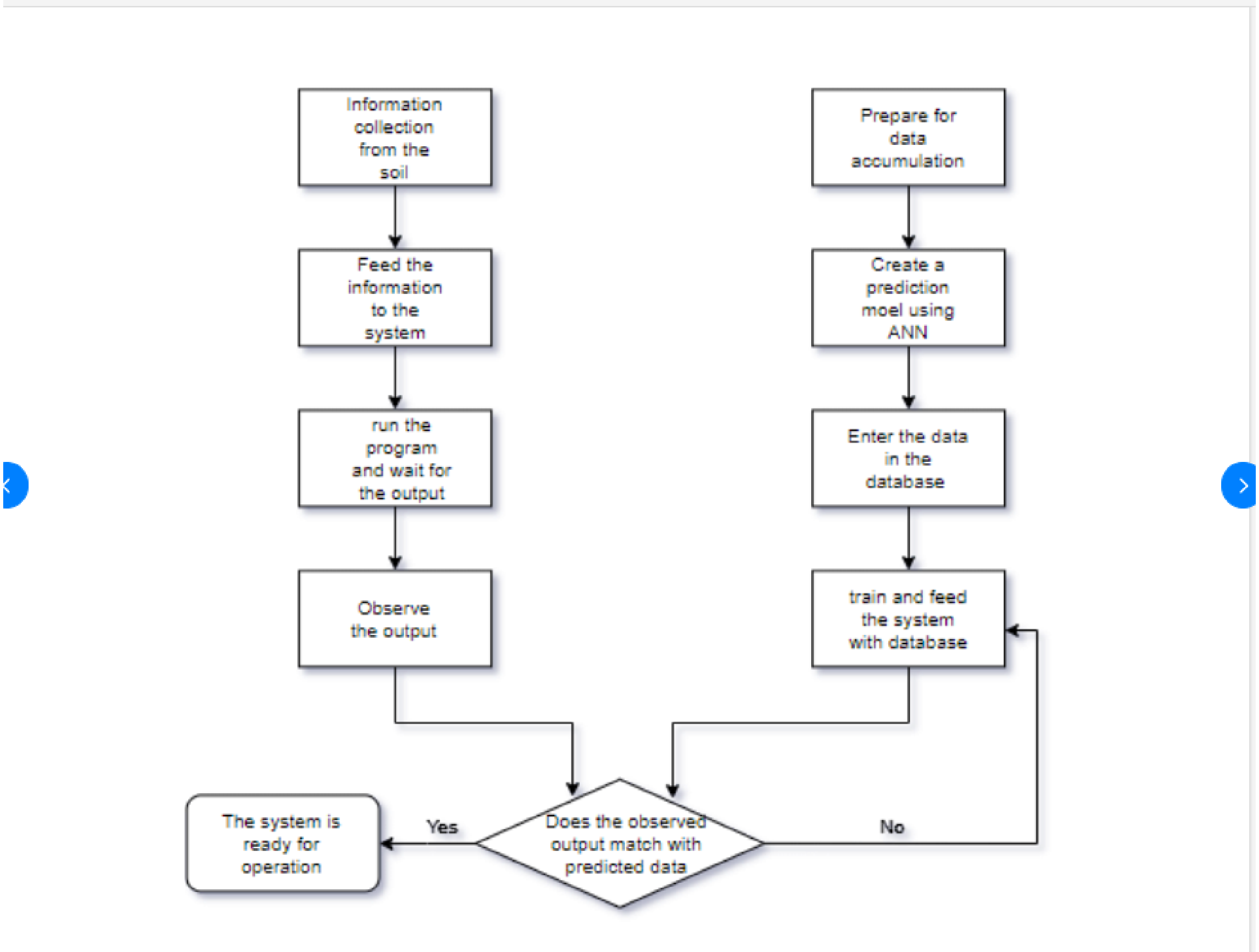

Information collection from the soil
Feed the information to the system
run the program and wait for the output
Observe the output
Prepare for data accumulation
Create a prediction moel using ANN
Enter the data in the database
train and feed the system with database
Does the observed output match with predicted data
Yes
No
The system is ready for operation

Figure 5.2 Solution Architecture

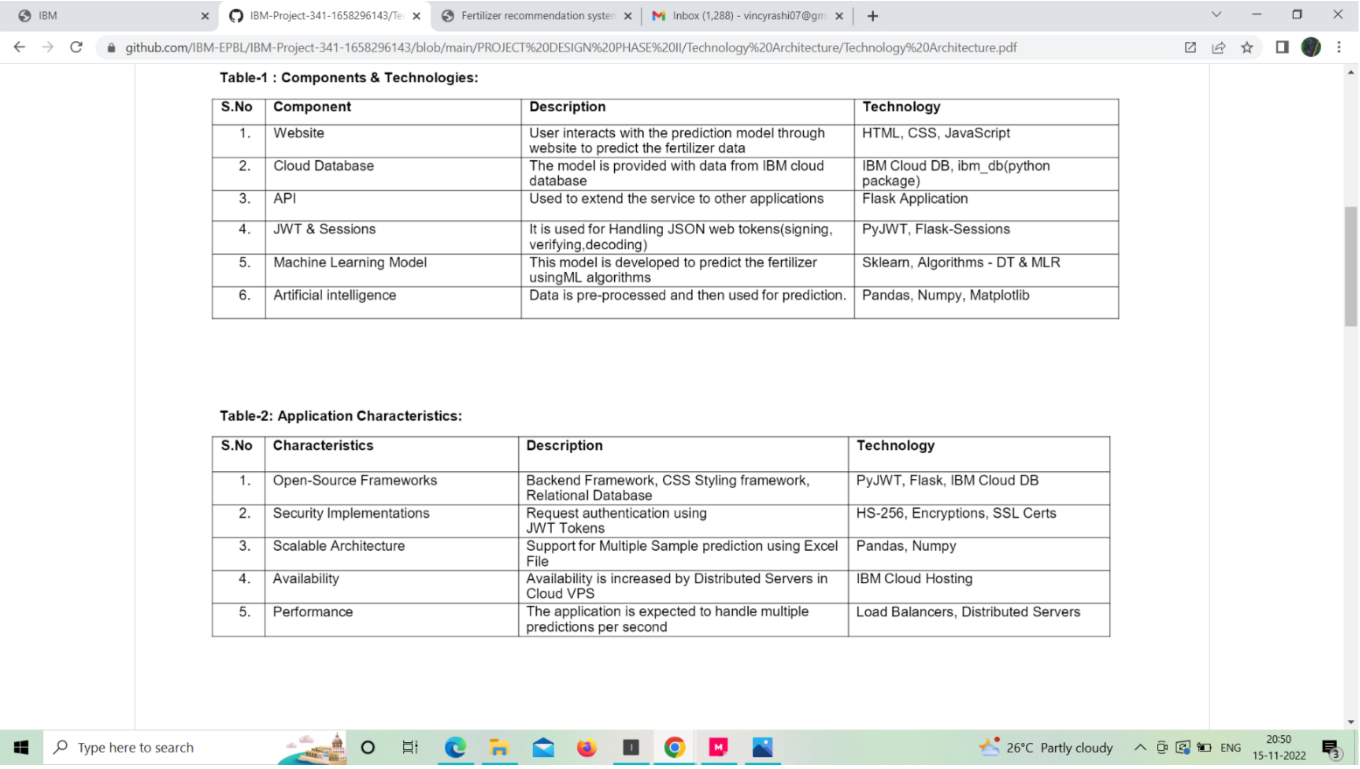

IBM
IBM-Project-341-1658296143/Te
Fertilizer recommendation systen
Inbox (1,288) - vincyrashi07@gm
github.com/IBM-EPBL/IBM-Project-341-1658296143/blob/main/PROJECT%20DESIGN%20PHASE%20II/Technology%20Architecture/Technology%20Architecture.pdf
Type here to search
26°C Partly cloudy
ENG
20:50
15-11-2022

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	Website	User interacts with the prediction model through website to predict the fertilizer data	HTML, CSS, JavaScript
2.	Cloud Database	The model is provided with data from IBM cloud database	IBM Cloud DB, ibm_db(python package)
3.	API	Used to extend the service to other applications	Flask Application
4.	JWT & Sessions	It is used for Handling JSON web tokens(signing, verifying,decoding)	PyJWT, Flask-Sessions
5.	Machine Learning Model	This model is developed to predict the fertilizer usingML algorithms	Sklearn, Algorithms - DT & MLR
6.	Artificial intelligence	Data is pre-processed and then used for prediction.	Pandas, Numpy, Matplotlib

Table-2: Application Characteristics:

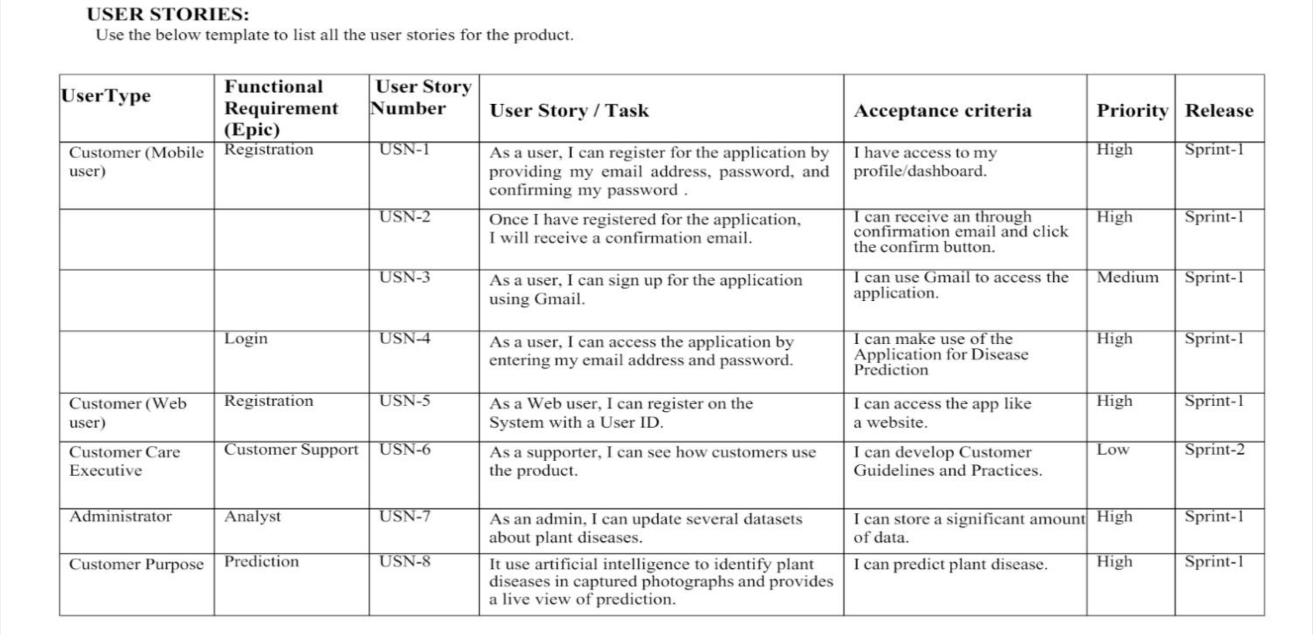
S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Backend Framework, CSS Styling framework, Relational Database	PyJWT, Flask, IBM Cloud DB
2.	Security Implementations	Request authentication using JWT Tokens	HS-256, Encryptions, SSL Certs
3.	Scalable Architecture	Support for Multiple Sample prediction using Excel File	Pandas, Numpy
4.	Availability	Availability is increased by Distributed Servers in Cloud VPS	IBM Cloud Hosting
5.	Performance	The application is expected to handle multiple predictions per second	Load Balancers, Distributed Servers

Figure 5.2 .1 Technical Architecture

5.3 USER STORIES :

A user story is an informal, general explanation of a software feature from the perspective of the end user ,

The purpose of a user story is to articulate how a work will deliver a particular value back to the customer.



USER STORIES:

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

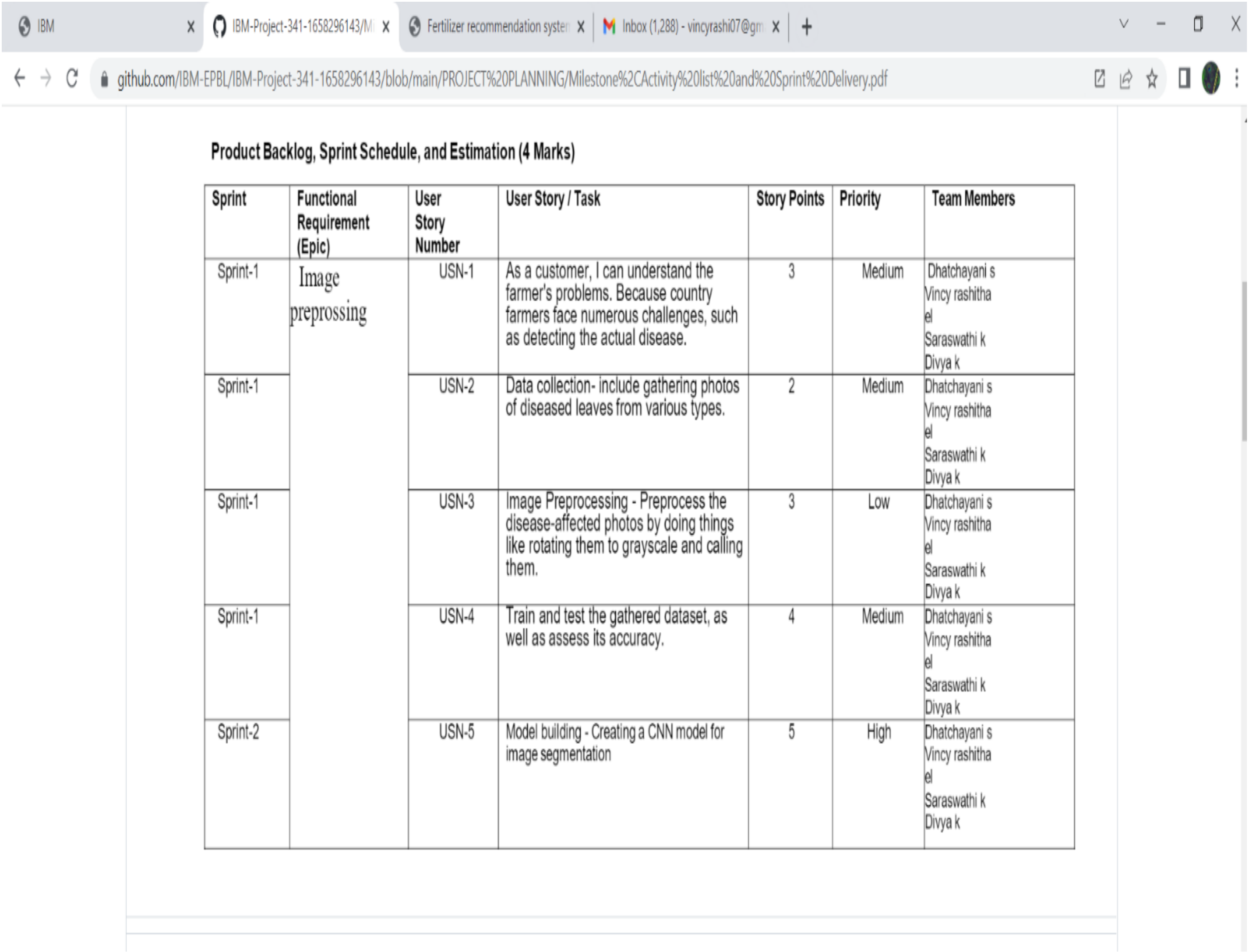
UserType	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by providing my email address, password, and confirming my password .	I have access to my profile/dashboard.	High	Sprint-1
		USN-2	Once I have registered for the application, I will receive a confirmation email.	I can receive an through confirmation email and click the confirm button.	High	Sprint-1
		USN-3	As a user, I can sign up for the application using Gmail.	I can use Gmail to access the application.	Medium	Sprint-1
	Login	USN-4	As a user, I can access the application by entering my email address and password.	I can make use of the Application for Disease Prediction	High	Sprint-1
Customer (Web user)	Registration	USN-5	As a Web user, I can register on the System with a User ID.	I can access the app like a website.	High	Sprint-1
Customer Care Executive	Customer Support	USN-6	As a supporter, I can see how customers use the product.	I can develop Customer Guidelines and Practices.	Low	Sprint-2
Administrator	Analyst	USN-7	As an admin, I can update several datasets about plant diseases.	I can store a significant amount of data.	High	Sprint-1
Customer Purpose	Prediction	USN-8	It use artificial intelligence to identify plant diseases in captured photographs and provides a live view of prediction.	I can predict plant disease.	High	Sprint-1

Figure 5.3 User Stories

6 PROJET PLANNING AND SCHEDULING:

6.1 SPRINT PLANNING AND ESTIMATION:

The objective of the Estimation would be to consider the User Stories for the Sprint by Priority and by the Ability of the team to deliver during the Time Box of the Sprint.


IBM
IBM-Project-341-1658296143/M
Fertilizer recommendation syster
Inbox (1,288) - vincyrashi07@gm
github.com/IBM-EPBL/IBM-Project-341-1658296143/blob/main/PROJECT%20PLANNING/Milestone%2CActivity%20list%20and%20Sprint%20Delivery.pdf

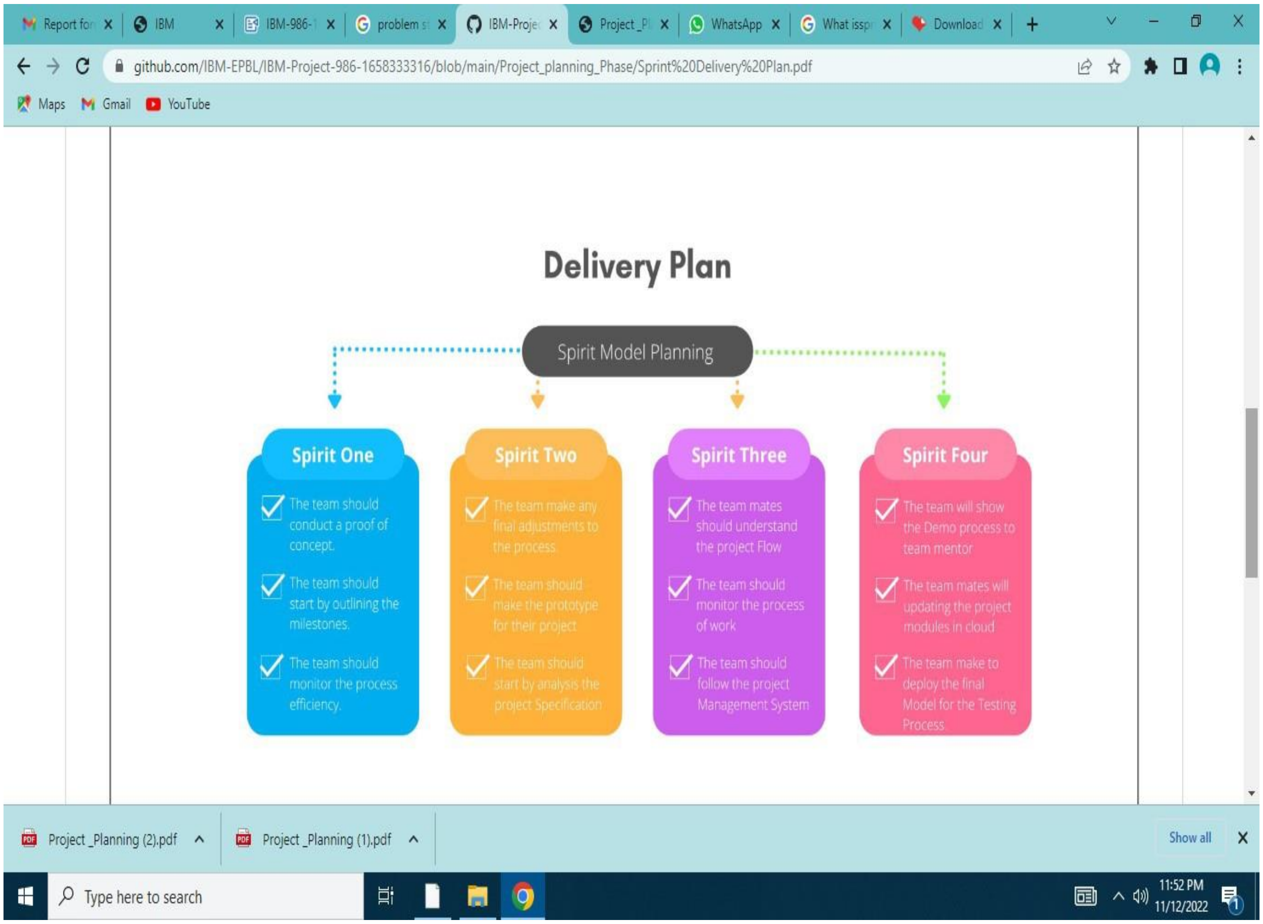
Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Image preprossing	USN-1	As a customer, I can understand the farmer's problems. Because country farmers face numerous challenges, such as detecting the actual disease.	3	Medium	Dhatchayani s Vincy rashitha el Saraswathi k Divya k
Sprint-1		USN-2	Data collection- include gathering photos of diseased leaves from various types.	2	Medium	Dhatchayani s Vincy rashitha el Saraswathi k Divya k
Sprint-1		USN-3	Image Preprocessing - Preprocess the disease-affected photos by doing things like rotating them to grayscale and calling them.	3	Low	Dhatchayani s Vincy rashitha el Saraswathi k Divya k
Sprint-1		USN-4	Train and test the gathered dataset, as well as assess its accuracy.	4	Medium	Dhatchayani s Vincy rashitha el Saraswathi k Divya k
Sprint-2		USN-5	Model building - Creating a CNN model for image segmentation	5	High	Dhatchayani s Vincy rashitha el Saraswathi k Divya k

Figure 6.1 Sprint Planning

6.2 SPRINT DELIVERY SCHEDULE :

The Objectives of the project must have to be must be separated in forms of sprints and separated to all the team members accordingly .Sprint is one timeboxed iteration of a continuous development cycle. Within a Sprint, planned amount of work has to be completed by the team and made ready for review.



Delivery Plan

Spirit Model Planning

Spirit One

- ✓ The team should conduct a proof of concept.
- ✓ The team should start by outlining the milestones.
- ✓ The team should monitor the process efficiency.

Spirit Two

- ✓ The team make any final adjustments to the process.
- ✓ The team should make the prototype for their project
- ✓ The team should start by analysis the project Specification

Spirit Three

- ✓ The team mates should understand the project Flow
- ✓ The team should monitor the process of work
- ✓ The team should follow the project Management System

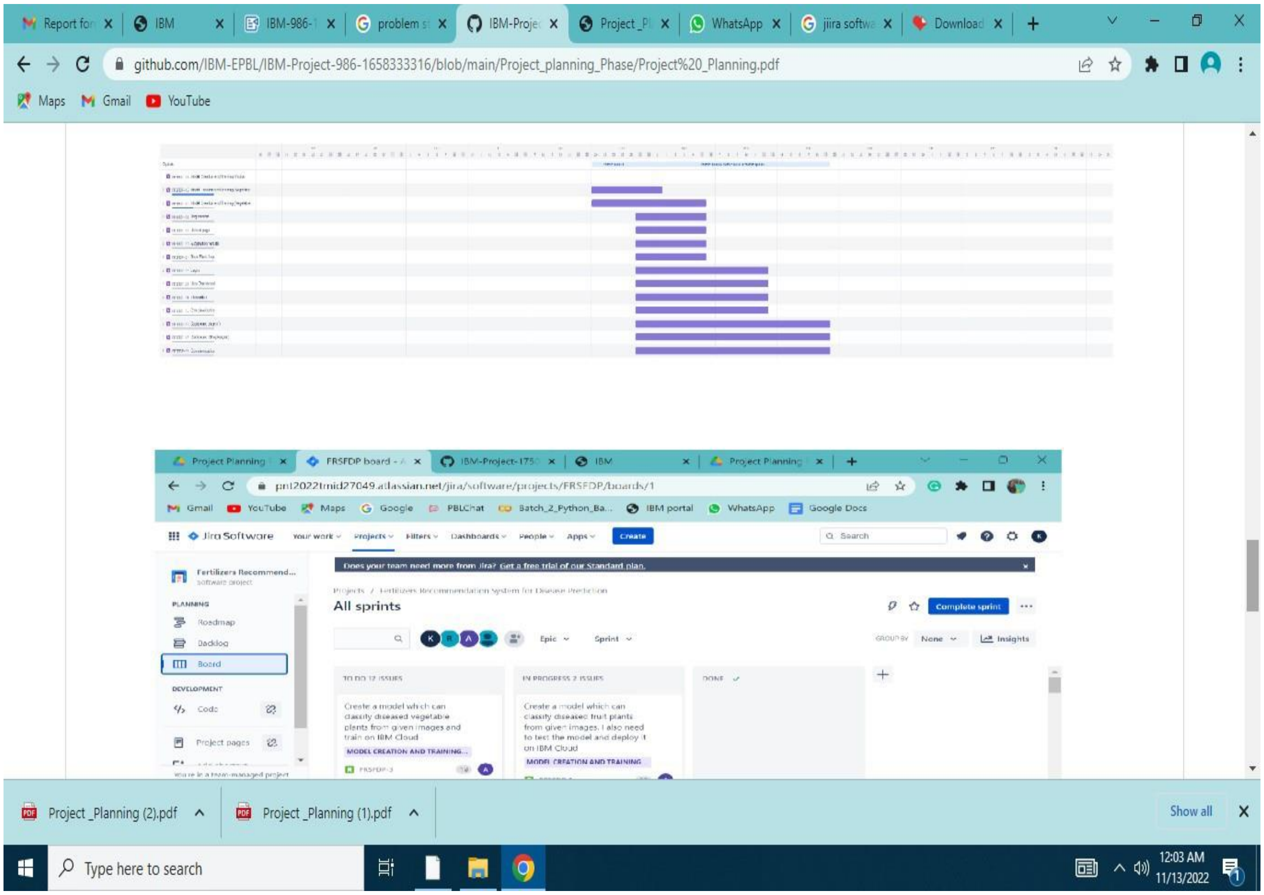
Spirit Four

- ✓ The team will show the Demo process to team mentor
- ✓ The team mates will updating the project modules in cloud
- ✓ The team make to deploy the final Model for the Testing Process

Figure 6.2 Sprint delivery schedule

6.3 REPORTS FROM JIIRA:

Jira Software is part of a family of products designed to help teams of all types manage work. Originally, Jira was designed as a bug and issue tracker. But today, Jira has evolved into a powerful work management tool for all kinds of use cases, from requirements and test case management to agile software development.


github.com/IBM-EPBL/IBM-Project-986-1658333316/blob/main/Project_planning_Phase/Project%20_Planning.pdf
pnt2022tmid27049.atlassian.net/jira/software/projects/FRSFDP/boards/1
Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Does your team need more from Jira? Get a free trial of our Standard plan.
Fertilizers Recommend...
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
You're in a team-managed project
Projects / Fertilizers Recommendation System for Disease Prediction
All sprints
Complete sprint
Epic
Sprint
GROUP BY
None
Insights
TO DO 12 ISSUES
IN PROGRESS 2 ISSUES
DONE ✓
MODEL CREATION AND TRAINING...
FRSFDP-3
MODEL CREATION AND TRAINING
Project _Planning (2).pdf
Project _Planning (1).pdf
Show all
Type here to search
12:03 AM
11/13/2022

Create a model which can classify diseased vegetable plants from given images and train on IBM Cloud

Create a model which can classify diseased fruit plants from given images. I also need to test the model and deploy it on IBM Cloud

Figure 6.3 Reports from Jira

7 CODING AND SOLUTIONING:

HTML FILE

```
<!DOCTYPE html>
<!-- Coding By CodingNepal - youtube.com/codingnepal -->
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>FRSFDP</title>
 <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}" type="text/css">
  </head>
  <body>
    <div class="center">
      <h1>FERTILIZER RECOMMENDATION SYSTEM FOR DISEASE
         PREDICITION</h1>
      <form action="/result" method="post">
        <div class="txt_field">
         <form action = "/predict" method = "POST" >
         <input type = "file" name = "file" />
         <input type = "submit"/>
           </div>
          </form>
          </div>


         </body>
        </html>
```

```
<html>
<body>
<form action = "/predict" method = "POST" >
<input type = "file" name = "file" />
<input type = "submit"/>
</form>
</body>
</html>
<p>{{result}}</p>
<!DOCTYPE html>
<!-- Coding By CodingNepal - youtube.com/codingnepal -->
<html lang="en" dir="ltr">
<head>
<meta charset="utf-8">
<title>Result</title>
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}" type="text/css">
</head>
<body>
<div class="center">
<center>
<P>{{result}}</P>
<img src="{{ url_for('static', filename='00fca0da-2db3-481b-b98a-9b67bb7b105c___RS_HL 7708.jpg')
}}"
```

```
</center>

</div>

</form>

</div>

</body>

</html>

<!DOCTYPE html>

<!-- Coding By CodingNepal - youtube.com/codingnepal -->

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>Result</title>

<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}" type="text/css">

</head>

<body>  <div class="center">

<center>    <P>{{result}}</P>

<img src="{{ url_for('static', filename='00fca0da-2db3-481b-b98a-9b67bb7b105c___RS_HL 7708.jpg')
}}"

</center>

</div>

</form>

</div>

</body> </html
```

CSS

```
.img-preview {
   width: 256px;
   height: 256px;
   position: relative;
   border: 5px solid #F8F8F8;
   box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
   margin-top: 1em;
   margin-bottom: 1em;
}

.img-preview>div {
   width: 100%;
   height: 100%;
   background-size: 256px 256px;
   background-repeat: no-repeat;
   background-position: center;
}

input[type="file"] {
   display: none;
}

.upload-label{
   display: inline-block;
   padding: 12px 30px;
   background: #39D2B4;
   color: #fff;
   font-size: 1em;
   transition: all .4s;
   cursor: pointer;
}

.upload-label:hover{
   background: #34495E;
   color: #39D2B4;
}
.loader {
   border: 8px solid #f3f3f3; /* Light grey */
   border-top: 8px solid #3498db; /* Blue */
   border-radius: 50%;
   width: 50px;
   height: 50px;
   animation: spin 1s linear infinite;
}
```

```
@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }

@import
url('https://fonts.googleapis.com/css2?family=Noto+Sans:wght@700&family=Poppins:wght@400;500;6
00&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Poppins", sans-serif;
}
body{
  margin: 0;
  padding: 0;
  background: linear-gradient(120deg,#2980b9, #8e44ad);
  height: 100vh;
  overflow: hidden;
}
.center{
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  width: 800px;
  background: white;
  border-radius: 10px;
  box-shadow: 10px 10px 15px rgba(0,0,0,0.05);
}
.center h1{
  text-align: center;
  padding: 20px 0;
  border-bottom: 1px solid silver;
}
.center form{
  padding: 0 40px;
  box-sizing: border-box;
```

```
}
form .txt_field{
  position: relative;
  border-bottom: 2px solid #adadad;
  margin: 30px 0;
}
.txt_field input{
  width: 100%;
  padding: 0 5px;
  height: 40px;
  font-size: 16px;
  border: none;
  background: none;
  outline: none;
}
.txt_field label{
  position: absolute;
  top: 50%;
  left: 5px;
  color: #adadad;
  transform: translateY(-50%);
  font-size: 16px;
  pointer-events: none;
  transition: .5s;
}
#head5{
position: absolute;
  top: 20%;
  left: 5px;
  font-size: 16px;
}
#Welcome{
position: absolute;
  top: 9%;
  left: 5px;
  font-size: 16px;
```

```
}
.txt_field span::before{
  content: '';
  position: absolute;
  top: 40px;
  left: 0;
  width: 0%;
  height: 2px;
  background: #2691d9;
  transition: .5s;
}
.txt_field input:focus ~ label,
.txt_field input:valid ~ label{
  top: -5px;
  color: #2691d9;
}
.txt_field input:focus ~ span::before,
.txt_field input:valid ~ span::before{
  width: 100%;
}
.pass{
  margin: -5px 0 20px 5px;
  color: #a6a6a6;
  cursor: pointer;
}
.pass:hover{
  text-decoration: underline;
}
input[type="submit"]{
  width: 100%;
  height: 50px;
  border: 1px solid;
  background: #2691d9;
  border-radius: 25px;
  font-size: 18px;
  color: #e9f4fb;
  font-weight: 700;
  cursor: pointer;
  outline: none;
}
input[type="submit"]:hover{
  border-color: #2691d9;
  transition: .5s;
}
.signup_link{
  margin: 30px 0;
  text-align: center;
  font-size: 16px;
  color: #666666;
}
.signup_link a{
  color: #2691d9;
  text-decoration: none;
}
.signup_link a:hover{
  text-decoration: underline;
}
table, td, th {
  border: 1px solid #ddd;
  text-align: center;
}
table {
  border-collapse: collapse;
  width: 50%;
}
th, td {
  padding: 15px;
}
th{
    background-color:#044154;
    color:white;
}
td{
    background-color:#ffffff;
}
```

PYTHON CODE:

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
# Flask utils
from flask import Flask, redirect, url_for, request, render_template

# Define a flask app
app = Flask(__name__)

def vegitable_predict(file_path):
  model = tf.keras.models.load_model(
    r'vegtable.h5')
  test_datagen_1 = ImageDataGenerator(rescale=1)
  test_generator_1 = test_datagen_1.flow_from_directory(
    test_dir, target_size=(128, 128), batch_size=20, class_mode='categorical')
  img = image.load_img(file_path, target_size=(128, 128))
  x = image.img_to_array(img)
  x = np.expand_dims(x, axis=0)
  y = np.argmax(model.predict(x), axis=1)
  index = [
    'Pepper,_bell___Bacterial_spot', 'Pepper,_bell___healthy',
    'Potato___Early_blight', 'Potato___healthy', 'Potato___Late_blight',
    'Tomato___Bacterial_spot', 'Tomato___Late_blight', 'Tomato___Leaf_Mold',
    'Tomato___Septoria_leaf_spot'
  ]
  return index[y[0]]

def fruit_predict(file_path):
  model = tf.keras.models.load_model(
    r'fruit.h5')
  test_datagen_1 = ImageDataGenerator(rescale=1)
  test_generator_1 = test_datagen_1.flow_from_directory(
    test_dir, target_size=(128, 128), batch_size=20, class_mode='categorical')
  img = image.load_img(file_path, target_size=(128, 128))
  x = image.img_to_array(img)
  x = np.expand_dims(x, axis=0)
  y = np.argmax(model.predict(x), axis=1)
  index = [
    'Apple___Black_rot', 'Apple___healthy', 'Corn_(maize)___healthy',
    'Corn_(maize)___Northern_Leaf_Blight', 'Peach___Bacterial_spot',
    'Peach___healthy'
  ]
  return index[y[0]]

@app.route('/', methods=['GET'])
def index():
  # Main page
  return render_template('home.html')

@app.route('/result', methods=['GET', 'POST'])
def upload():
  if request.method == 'POST':
    # Get the file from post request
    #f = request.files['file']
    # Save the file to ./uploads
    #basepath = os.path.dirname(__file__)
    #file_path = os.path.join(basepath, 'uploads', secure_filenaeme(f.filename))
    #f.save()
    # Make prediction
    #preds = fruit_predict(file_path)
    #result = preds
    return render_template('result.html',result="Apple___Black_rot")

if __name__ == '__main__':
  app.run(port=5001, debug=True
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
# Flask utils
from flask import Flask, redirect, url_for, request, render_template
# Define a flask app
app = Flask(__name__)

def fruit_predict(file_path):
    model = tf.keras.models.load_model(r'/content/drive/MyDrive/Dataset Plant Disease/fruit.h5')
    test_datagen_1=ImageDataGenerator(rescale=1)
    test_generator_1=test_datagen_1.flow_from_directory(
    test_dir,
    target_size=(128,128),
    batch_size=20,
    class_mode='categorical')
    img=image.load_img(file_path,target_size=(128,128))
    x=image.img_to_array(img)
    x=np.expand_dims(x,axis=0)
    y=np.argmax(model.predict(x),axis=1)
    index=['Apple___Black_rot', 'Apple___healthy', 'Corn_(maize)___healthy',
'Corn_(maize)___Northern_Leaf_Blight', 'Peach___Bacterial_spot', 'Peach___healthy']
    return index[y[0]]
@app.route('/', methods=['GET'])
def index():
    # Main page
    return render_template('index.html')


@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['file']
        # Save the file to ./uploads
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)
        # Make prediction
        preds = fruit_predict(file_path)
        #result=preds
        #return result
        return render_template('index.html',result=preds)

if __name__ == '__main__':
    app.run(port=5001,debug=True)

. # -*- coding: utf-8 -*-
"""
Created on Thu Jun 11 22:34:20 2020

@author: Krish Naik
"""

from __future__ import division, print_function
# coding=utf-8
import sys
import os
import glob
import re
import numpy as np
import tensorflow as tf
import tensorflow as tf

from tensorflow.compat.v1 import ConfigProto
from tensorflow.compat.v1 import InteractiveSession

config = ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = 0.2
config.gpu_options.allow_growth = True
session = InteractiveSession(config=config)
# Keras
from tensorflow.keras.applications.resnet50 import preprocess_input
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

# Flask utils
from flask import Flask, redirect, url_for, request, render_template
from werkzeug.utils import secure_filename
#from gevent.pywsgi import WSGIServer

# Define a flask app
app = Flask(__name__)

# Model saved with Keras model.save()
MODEL_PATH ='model_inception.h5'

# Load your trained model
model = load_model(MODEL_PATH)

def model_predict(img_path, model):
    print(img_path)
    img = image.load_img(img_path, target_size=(224, 224))

    # Preprocessing the image
    x = image.img_to_array(img)
    # x = np.true_divide(x, 255)
    ## Scaling
    x=x/255
    x = np.expand_dims(x, axis=0)



    # Be careful how your trained model deals with the input
    # otherwise, it won't make correct prediction!
   # x = preprocess_input(x)
```

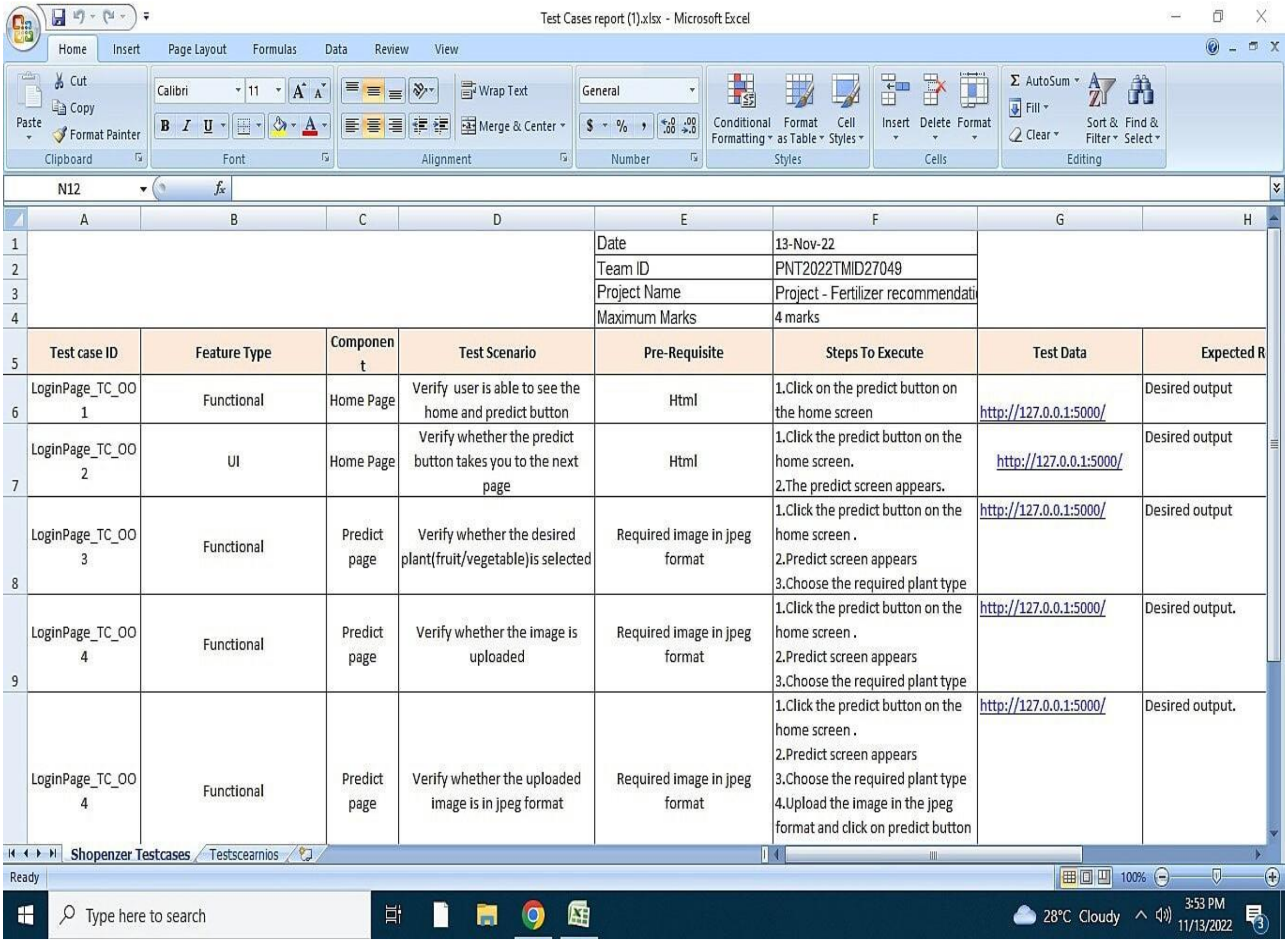
```
    preds = model.predict(x)
    preds=np.argmax(preds, axis=1)
    if preds==0:
        preds="The Disease is Pepper__bell___Bacterial_spot"
    elif preds==1:
        preds="The Disease is Pepper__bell___healthy"
    elif preds==2:
        preds="The Disease is Potato___Early_blight"
    elif preds==3:
        preds="Te Disease is Potato___healthy"
    elif preds==4:
        preds="The Disease is Potato___Late_blight"
    elif preds==5:
        preds="The Disease is Tomato__Tomato_mosaic_virus"
    elif preds==6:
        preds="The Disease is Tomato__Tomato_YellowLeaf__Curl_Virus"
    elif preds==7:
        preds="The Disease is Tomato_Bacterial_spot"
    elif preds==8:
        preds="The Disease is Tomato_Early_blight"
    elif preds==9:
        preds="The Disease is Pepper__bell___Bacterial_spot"
    elif preds==10:
        preds="The Disease is Pepper__bell___Bacterial_spot"
    elif preds==11:
        preds="The Disease is Pepper__bell___Bacterial_spot"
    elif preds==12:
        preds="The Disease is Pepper__bell___Bacterial_spot"
    elif preds==13:
        preds="The Disease is Pepper__bell___Bacterial_spot"
            return preds
@app.route('/', methods=['GET'])
def index():
    # Main page
    return render_template('index.html')
```

```
@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['file']

        # Save the file to ./uploads
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)

        # Make prediction
        preds = model_predict(file_path, model)
        result=preds
        return result
    return None
if __name__ == '__main__':
    app.run(port=5001,debug=True)
```

8. TESTING:

8.1 TEST CASES :

A test is a set of action performed on a system to determine if it satisfies software requirements and function correctly . The purpose of a test case is to determine if different features within a system are performing as expected and to confirm that the system satisfies all related standards , guidelines and customer requirements. The process of writing a test case can also help Reveal errors or defects within the system.



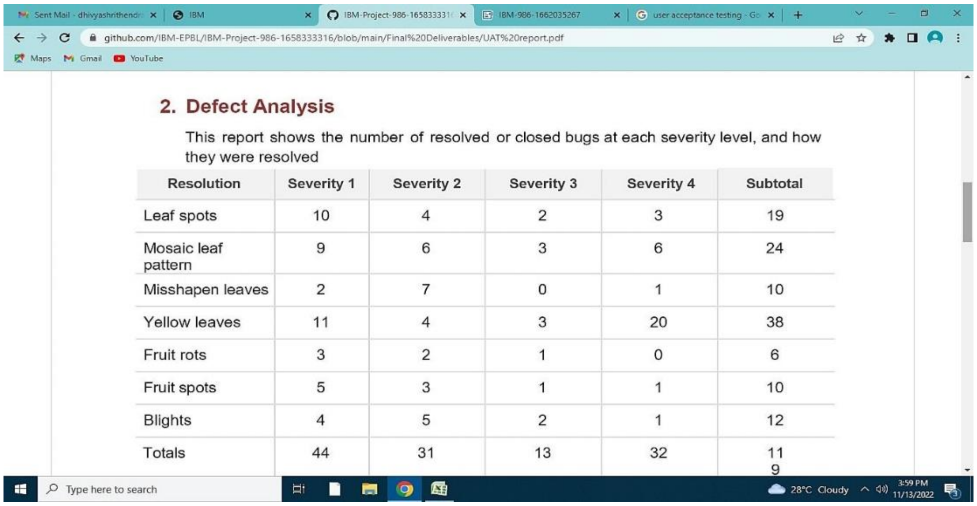
Date	13-Nov-22
Team ID	PNT2022TMID27049
Project Name	Project - Fertilizer recommendati
Maximum Marks	4 marks

Test case ID	Feature Type	Componen t	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected R
LoginPage_TC_OO 1	Functional	Home Page	Verify user is able to see the home and predict button	Html	1.Click on the predict button on the home screen	http://127.0.0.1:5000/	Desired output
LoginPage_TC_OO 2	UI	Home Page	Verify whether the predict button takes you to the next page	Html	1.Click the predict button on the home screen. 2.The predict screen appears.	http://127.0.0.1:5000/	Desired output
LoginPage_TC_OO 3	Functional	Predict page	Verify whether the desired plant(fruit/vegetable)is selected	Required image in jpeg format	1.Click the predict button on the home screen . 2.Predict screen appears 3.Choose the required plant type	http://127.0.0.1:5000/	Desired output
LoginPage_TC_OO 4	Functional	Predict page	Verify whether the image is uploaded	Required image in jpeg format	1.Click the predict button on the home screen . 2.Predict screen appears 3.Choose the required plant type	http://127.0.0.1:5000/	Desired output.
LoginPage_TC_OO 4	Functional	Predict page	Verify whether the uploaded image is in jpeg format	Required image in jpeg format	1.Click the predict button on the home screen . 2.Predict screen appears 3.Choose the required plant type 4.Upload the image in the jpeg format and click on predict button	http://127.0.0.1:5000/	Desired output.

Figure 8.1 Test Case

8.2 USER ACCEPTANCE TESTING:

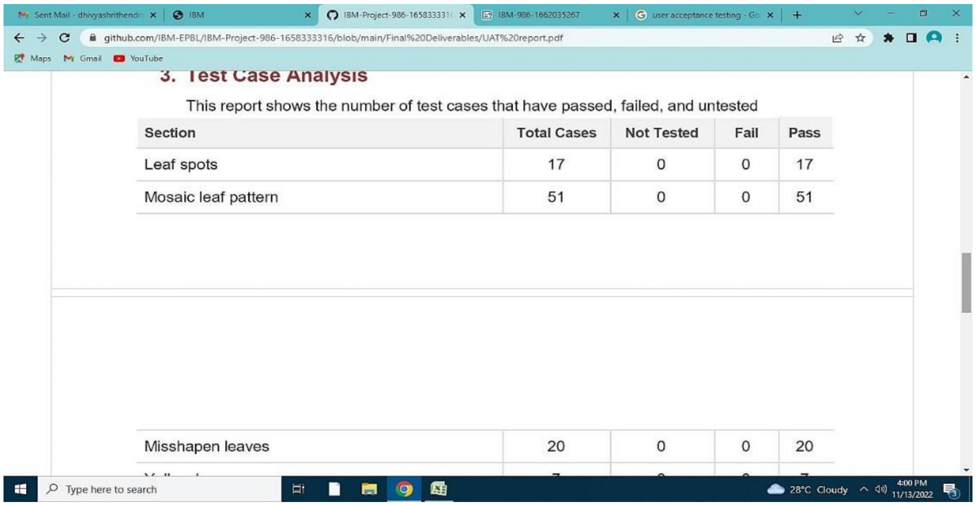
User Acceptance Testing (UAT), which is performed on most UIT projects, sometimes called beta testing or end-user testing, is a phase of software development in which the software is tested in the "real world" by the intended audience or business representative.


Sent Mail - dhivyashrithendr
IBM
IBM-Project-986-165833331
IBM-986-1662035267
user acceptance testing - Go
github.com/IBM-EPBL/IBM-Project-986-1658333316/blob/main/Final%20Deliverables/UAT%20report.pdf
Maps
Gmail
YouTube
Type here to search
28°C Cloudy
3:59 PM
11/13/2022

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
Leaf spots	10	4	2	3	19
Mosaic leaf pattern	9	6	3	6	24
Misshapen leaves	2	7	0	1	10
Yellow leaves	11	4	3	20	38
Fruit rots	3	2	1	0	6
Fruit spots	5	3	1	1	10
Blights	4	5	2	1	12
Totals	44	31	13	32	11 9


Sent Mail - dhivyashrithendr
IBM
IBM-Project-986-165833331
IBM-986-1662035267
user acceptance testing - Go
github.com/IBM-EPBL/IBM-Project-986-1658333316/blob/main/Final%20Deliverables/UAT%20report.pdf
Maps
Gmail
YouTube
Type here to search
28°C Cloudy
4:00 PM
11/13/2022

3. Test Case Analysis

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

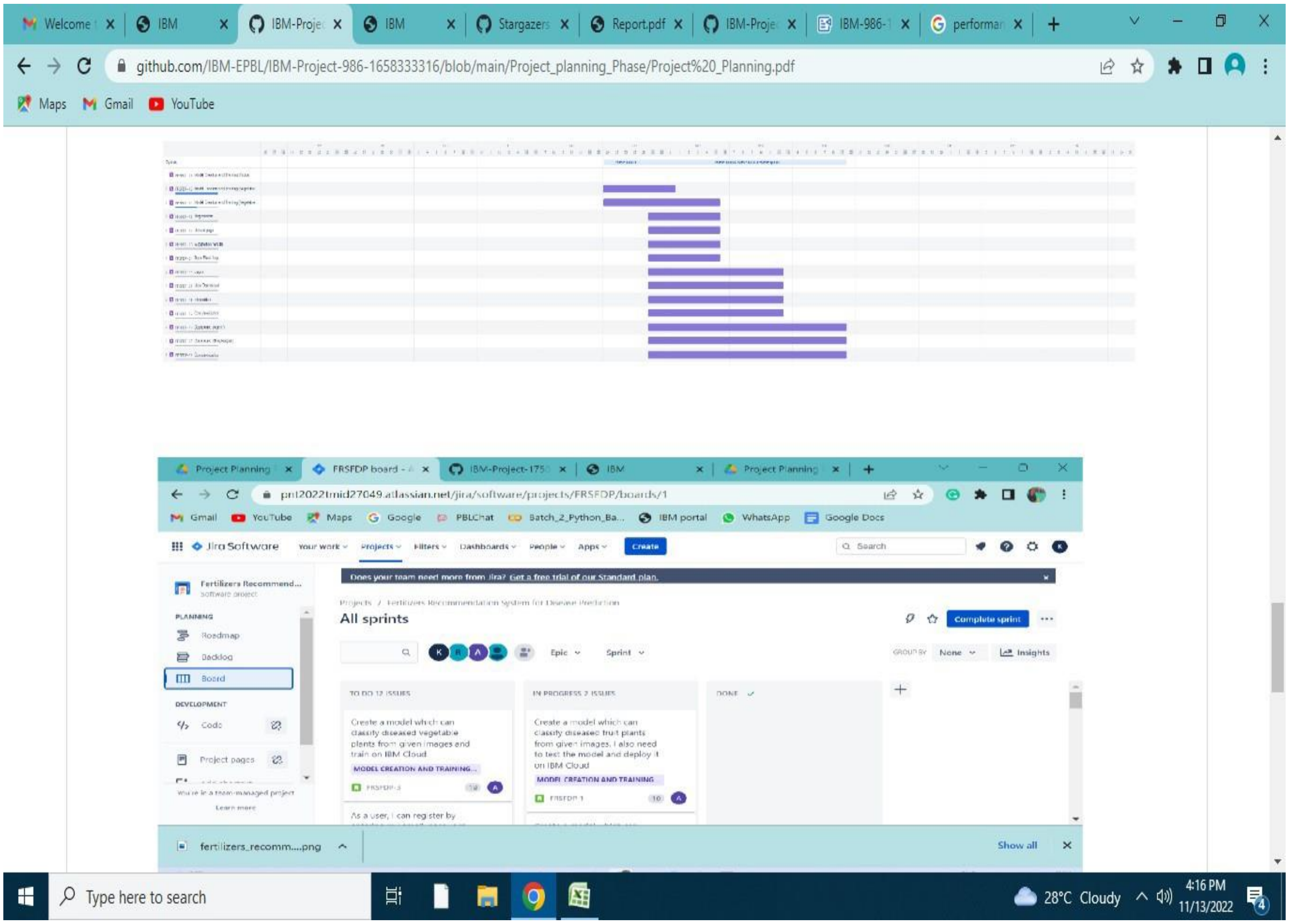
Section	Total Cases	Not Tested	Fail	Pass
Leaf spots	17	0	0	17
Mosaic leaf pattern	51	0	0	51
Misshapen leaves	20	0	0	20

Figure 8.2 User Acceptance Testing

9.RESULT:

9.1 PERFORMANCE METRICS:

Performance metrics are defined as figures and data representative of an organization's actions, abilities, and overall quality. Performance testing can be used to analyze various success factors such as response times and potential errors. With these performance results in hand, you can confidently identify bottlenecks, bugs, and mistakes – and decide how to optimize your application to eliminate the problem(s).



github.com/IBM-EPBL/IBM-Project-986-1658333316/blob/main/Project_planning_Phase/Project%20Planning.pdf

pnt2022tmid27049.atlassian.net/jira/software/projects/FRSFDP/boards/1

Does your team need more from Jira? Get a free trial of our Standard plan.

Projects / Fertilizers Recommendation System for Disease Prediction

All sprints

TO DO 12 ISSUES

Create a model which can classify diseased vegetable plants from given images and train on IBM Cloud

MODEL CREATION AND TRAINING...

FRSFDP-3

As a user, I can register by

IN PROGRESS 2 ISSUES

Create a model which can classify diseased fruit plants from given images. I also need to test the model and deploy it on IBM Cloud

MODEL CREATION AND TRAINING

FRSFDP-1

DONE ✓

Figure 9.1 Performance Metrics

10.ADVANTAGES AND DISADVANTAGES:

ADVANTAGES:

1. Fertilizers provide crops with nutrients like potassium, phosphorus, and nitrogen, which allow crops to grow bigger, faster, and to produce more food. Nitrogen in particular is an essential nutrient for the growth of every organism on Earth. Nitrogen is all around us and makes up about 78% of the air you breathe.

2. Sometimes plants need a quick fix to survive, in this type of cases fertilizers play a vital role to improve plants' health. plants need nutrients that can be absorbed quickly which is fulfilled by fertilizers. They are easily soluble and fatly absorbed by plants and as soon as possible it helps to regain and boost plant health.

3. As the population is increasing, there is a huge demand for food, so good yield is required to fulfill the demand. Here fertilizers become helpful for the good production of crops due to their numerous benefits which promote the fast and healthy growth of plants. For large production, fertilizers become compulsory.

DISADVANTAGES:

4.Fertilizers are man-made so they need production in factories which makes them costlier than naturally made manure. But it is important for plant nutrients so it is in demand and thus it has high value.

5.Fertilizers are used in moderate quantities if we use excessive fertilizers it surely damages the roots of plants and their tissues and thus plants can die. Fertilizers are used according to the need of the plant . Unnecessary use of them can affect the plant's health specially if plants have good fertile soil.

6.There are many types of fertilizers in the market, some of them are many types of fertilizers in the market, some of them are chemically made .These chemical fertilizers are harmful to humans and plants also. Skin irritation , respiratory problems commonly occur due to fertilizers. Can pass harmful chemical in our food which affects.

7.Fertilizers can reduce the quality of soil and can harm microorganisms in the soil. Long-term use disturbs the pH of the soil and also reduces the microbial activities which are naturally good for plants.

11.CONCLUSION:

The authors proposed a new approach for the soil based fertilizers prediction system. The proposed system was able to analyze the soil nutrient type efficiently, kind of leaf disease present in the crop and predict the fertilizer in a proficient manner. The approach was flexible, and can be extended to the needs of the users in a better manner. The proposed method was carried out with five different crops.

12.FUTURE SCOPE:

Different approaches and models of Deep Learning methods were explored and used in this project so that it can detect and classify plant diseases correctly through image processing of leaves of the plants. The procedure starts from collecting the images used for training, testing and validation to image preprocessing and augmentation and finally comparison of different pretrained models over their accuracy. Finally, at the end , our model detects and distinguishes between a healthy plant and different diseases and provides suitable remedies so as to cure the disease .This paper proposed and developed a system which uses plant leaf images to detect different **types** of disease in tomato crops, and also provides appropriate fertilizer suggestions

13.APPENDIX:

The project deliverables are uploaded in GitHub repository dashboard.

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-341-1658296143

Demo Link:
https://drive.google.com/file/d/1T7UM0lt4b9TzDVVEmZzutOvwfZ9JLTCu/view?usp=share_link

Source Code Link:

https://drive.google.com/file/d/1IZ_9cXlKAnqSK6BUhhO9BBw_MdbXnpl2/view?usp=share_link

SOURCE CODE:

```
<html>
<body>
<form action = "/predict" method = "POST" >
<input type = "file" name = "file" />
<input type = "submit"/>
</form>
</body>
</html>

<!DOCTYPE html>
<!-- Coding By CodingNepal - youtube.com/codingnepal -->
<html lang="en" dir="ltr">
<head>
<meta charset="utf-8">
<title>Result</title>
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}" type="text/css">
</head>
<body>
<div class="center">
<center>
<P>{{result}}</P>
</center>
</div>
</form>
</div>
</body>
</html>
```

```
@import
url('https://fonts.googleapis.com/css2?family=Noto+Sans:wght@700&family=Poppins:wght@400;500;6
00&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Poppins", sans-serif;
}
body{
  margin: 0;
  padding: 0;
  background: linear-gradient(120deg,#2980b9, #8e44ad);
  height: 100vh;
  overflow: hidden;
}
.center{
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  width: 800px;
  background: white;
  border-radius: 10px;
  box-shadow: 10px 10px 15px rgba(0,0,0,0.05);
}
.center h1{
  text-align: center;
  padding: 20px 0;
  border-bottom: 1px solid silver;
}
.center form{
  padding: 0 40px;
  box-sizing: border-box;
}
form .txt_field{
  position: relative;
  border-bottom: 2px solid #adadad;
  margin: 30px 0;
}
.txt_field input{
  width: 100%;
  padding: 0 5px;
  height: 40px;
  font-size: 16px;
  border: none;
  background: none;
  outline: none;
}
.txt_field label{
  position: absolute;
  top: 50%;
  left: 5px;
  color: #adadad;
  transform: translateY(-50%);
  font-size: 16px;
  pointer-events: none;
  transition: .5s;
}
#head5{
position: absolute;
  top: 20%;
  left: 5px;
  font-size: 16px;
}
#Welcome{
position: absolute;
  top: 9%;
  left: 5px;
  font-size: 16px;
}
.txt_field span::before{
  content: ";
  position: absolute;
  top: 40px;
  left: 0;
  width: 0%;
  height: 2px;
  background: #2691d9;
  transition: .5s;
}
.txt_field input:focus ~ label,
.txt_field input:valid ~ label{
  top: -5px;
  color: #2691d9;
}
.txt_field input:focus ~ span::before,
.txt_field input:valid ~ span::before{
  width: 100%;
}
.pass{
  margin: -5px 0 20px 5px;
  color: #a6a6a6;
  cursor: pointer;
}
.pass:hover{
  text-decoration: underline;
}
input[type="submit"]{
  width: 100%;
  height: 50px;
  border: 1px solid;
  background: #2691d9;
  border-radius: 25px;
  font-size: 18px;
  color: #e9f4fb;
  font-weight: 700;
  cursor: pointer;
  outline: none;
```

```
}
input[type="submit"]:hover{
  border-color: #2691d9;
  transition: .5s;
}
.signup_link{
  margin: 30px 0;
  text-align: center;
  font-size: 16px;
  color: #666666;
}
.signup_link a{
  color: #2691d9;
  text-decoration: none;
}
.signup_link a:hover{
  text-decoration: underline;
}
table, td, th {
  border: 1px solid #ddd;
  text-align: center;
}
table {
  border-collapse: collapse;
  width: 50%;
}
th, td {
  padding: 15px;
}
th{
    background-color:#044154;
    color:white;
}
td{
    background-color:#ffffff;
}
```

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
# Flask utils
from flask import Flask, redirect, url_for, request, render_template

# Define a flask app
app = Flask(__name__)


def vegitable_predict(file_path):
  model = tf.keras.models.load_model(
    r'vegtable.h5')
  test_datagen_1 = ImageDataGenerator(rescale=1)
  test_generator_1 = test_datagen_1.flow_from_directory(
    test_dir, target_size=(128, 128), batch_size=20, class_mode='categorical')
  img = image.load_img(file_path, target_size=(128, 128))
  x = image.img_to_array(img)
  x = np.expand_dims(x, axis=0)
  y = np.argmax(model.predict(x), axis=1)
  index = [
    'Pepper,_bell___Bacterial_spot', 'Pepper,_bell___healthy',
    'Potato___Early_blight', 'Potato___healthy', 'Potato___Late_blight',
    'Tomato___Bacterial_spot', 'Tomato___Late_blight', 'Tomato___Leaf_Mold',
    'Tomato___Septoria_leaf_spot'
  ]
  return index[y[0]]


def fruit_predict(file_path):
  model = tf.keras.models.load_model(
    r'fruit.h5')
  test_datagen_1 = ImageDataGenerator(rescale=1)
  test_generator_1 = test_datagen_1.flow_from_directory(
    test_dir, target_size=(128, 128), batch_size=20, class_mode='categorical')
  img = image.load_img(file_path, target_size=(128, 128))
  x = image.img_to_array(img)
  x = np.expand_dims(x, axis=0)
  y = np.argmax(model.predict(x), axis=1)
  index = [
    'Apple___Black_rot', 'Apple___healthy', 'Corn_(maize)___healthy',
    'Corn_(maize)___Northern_Leaf_Blight', 'Peach___Bacterial_spot',
    'Peach___healthy'
  ]
  return index[y[0]]


@app.route('/', methods=['GET'])
def index():
  # Main page
  return render_template('home.html')


@app.route('/result', methods=['GET', 'POST'])
def upload():
  if request.method == 'POST':
    # Get the file from post request
    f = request.files['file']
    # Save the file to ./uploads
    basepath = os.path.dirname(__file__)
    file_path = os.path.join(basepath, 'uploads', secure_filenaeme(f.filename))
    f.save()
    # Make prediction
    preds = fruit_predict(file_path)
    #result = preds
    return render_template('result.html',result=pred)

if __name__ == '__main__':
  app.run(port=5001, debug=True)
```