PROJECT REPORT

in the Title of

IOT BASED SMART CROP PROTECTION FOR AGRICULTURE

TEAM LEAD: Elamathi G

TEAM MEMBER 1: Durgadevi S

TEAM MEMBER 2: Giftson Jebasingh D

TEAM MEMBER 3: Jeyanesh V

TABLE OF CONTENT:

1. INTRODUCTION

1.1 PROJECT OVERVIEW:

The main aim of our project is to protect the crops from damage caused by animal as well as divert the animal without any harm. Crops in farms are many times ravaged by local animals like buffaloes, cows, goats, birds etc and also from the climate change conditions. This leads to huge losses for the farmers. It is not possible for farmers to barricade entire fields or stay on field 24 hours and guard it. So here we propose automatic crop protection system from animals. Animal detection system is designed to detect the presence of animal and offer a warning. This project work contains various sorts of Sensors, Controllers. Different sensors like DHT 11 Humidity & Temperature Sensor, PIR sensor, LDR sensor, HC-SR04 Ultrasonic sensor and cameras are interfaced with the board. It diverts the animal by producing sound and signal further which gives an alert to farmers and forest department immediately.

1.2 PURPOSE:

The main purpose of the project is to prevent the crops from the animals, birds and climate conditions then it will be alerted to the farmer or to relevant person IOT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, Temperature, Humidity using some sensors. Farmers can monitor all the sensor parameter by using a web application even if the farmer is not near his field. Watering the crop is one of the important tasks for the farmers. They can make the decision whether to water the crop or postpone it by monitoring the sensor parameters and control the motor pumps from the mobile application itself.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

The current existing method for measuring the climate conditions and sensing the roaming of animal and birds are done in a separate systems they are not integrated as a single system. Here we need a two different method and hardware design to protect the crop from the animals, birds and sensing the climate scaling parameters , so to avoid that situation and inconvenience of the farmer in his field , we are proposing the new integrated system design using IOT technology . and the important points to execute the systems are:

- In agriculture water is needed for the crops for their growth. If the Soil gets dry it is necessary to supply water. But sometime if the farmer doesn't visit the field, it is not possible to know the condition of soil.
- Sometimes over supply of water or less supply of water affects the growth of crops.
- Sometimes if the weather/temperature changes suddenly it is necessary to take certain actions.
- Specific crops grow better in specific conditions, they may get damaged due to bad weather.

2.2 PROBLEM SOLVING DEFINITION

The proposed method was designed to overcome the problems faced in the existing methodology. Some of the most important solutions are Soil Moisture can be checked by using the sensors that can sense the soil condition and send the moisture content in the soil over the cloud services to the web application.

● The supply of water can be controlled from anywhere by controlling the motor state (ON/OFF), using web application.

● Surrounding temperature can also be sensed by the sensors and displayed on the application.

● Real time weather conditions can also be known by using different weather APIs from different websites and displayed on our application.

2.3 REFERENCES:

1) In 2017 - S. Athani, C. H. Tejeshwar, M. M. Patil, P. Patil and R. Kulkarni, "Soil moisture monitoring using IoT enabled arduino sensors for improving soil management for farmers and predict seasonal rainfall for planning future harvest in North Karnataka — India.
2) In 2019 - P. Venkateswara Rao, Ch Siva Rama Krishna and M Samba Siva Reddy, "A smart crop protection against animal attack".
3) In 2021-R.M. Joany; E. logashanmugam; E. Anna Devi; S. Yogalakshmi; in the title of " IOT based crop protection system during rainy season".
4) In 2019 – S.Gobinath; M. Devi Darshini; K.Durga ; R.Hari Priyanga in the title of " Smart Irrigation With Field Protection and Crop Health Monitoring System ".

3. IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

The Internet of Things (IoT) is an evolving paradigm that seeks to connect different smart physical components for multidomain modernization. To automatically manage and track agricultural lands with minimal human intervention, numerous IoT-based frameworks have been introduced.

This paper presents a rigorous discussion on the major components, new technologies, security issues, challenges and future trends involved in the agriculture domain. An in-depth report on recent advancements has been covered in this paper.

The goal of this survey is to help potential researchers detect relevant IoT problems and, based on the application requirements, adopt suitable technologies. Furthermore, the significance of IoT and Data Analytics for smart agriculture has been highlighted.

3.2 IDEATION AND BRAINSTROM

Soil condition monitoring:

Soil condition is an essential indicator allowing growers to decide on the accurate planting and crop collection time. With IoT-based sensors performing soil condition monitoring, growers get instant warnings of soil salinity and moisture. Other metrics have an air temperature and soil temperature system that allows farmers to schedule watering times and predict the chances of pests. Soil condition monitoring needs a combination of software and hardware systems to work in real-time and pass the warning to users on any changes.

Weather monitoring:

Weather monitoring in farming is the most used application domain for IoT. In crop farming, yields are dependent on environmental conditions, which are naturally volatile. Weather monitoring systems installed directly in the farming field warn farmers of varying weather conditions such as precipitation, temperature, solar radiation, wind speed, and humidity.

Smart Irrigation on Agriculture Land:

In smart irrigation, automated sprinkler systems or intelligent pumps are used. Soil moistures sensors are used in different areas to get the moisture of the soil in agricultural land. Based on the results from the soil moisture sensors, the intelligent pumps or intelligent sprinklers are turned On/Off

Livestock Monitoring:

Internet of Things devices can be used to collect data regarding the location, well-being, and the health of the cattle. This data can be further used for identification of the sick animals so that they can be separated from the others, thereby preventing the spread of diseases. This Live Stock Monitoring also lowers the labour costs with the help of Internet of Things based sensors.

Drone Monitoring:

Drone monitoring is helping large farms to reduce the cost of monitoring, or the use of Geo-positioning sensors can set a stable path. Moreover, the data collected from these drones are sent back to the server where it can be used for analyzing and decision-making.

3.3 PROPOSED SOLUTION

The project primary goals are to safeguard crops from animal damage and safely move animals away from crops. Using latest technology ultrasonic sensors, we can find a solution by detecting the animal's presence and sending a signal to the controller to tell about it and address the issue to get protect.

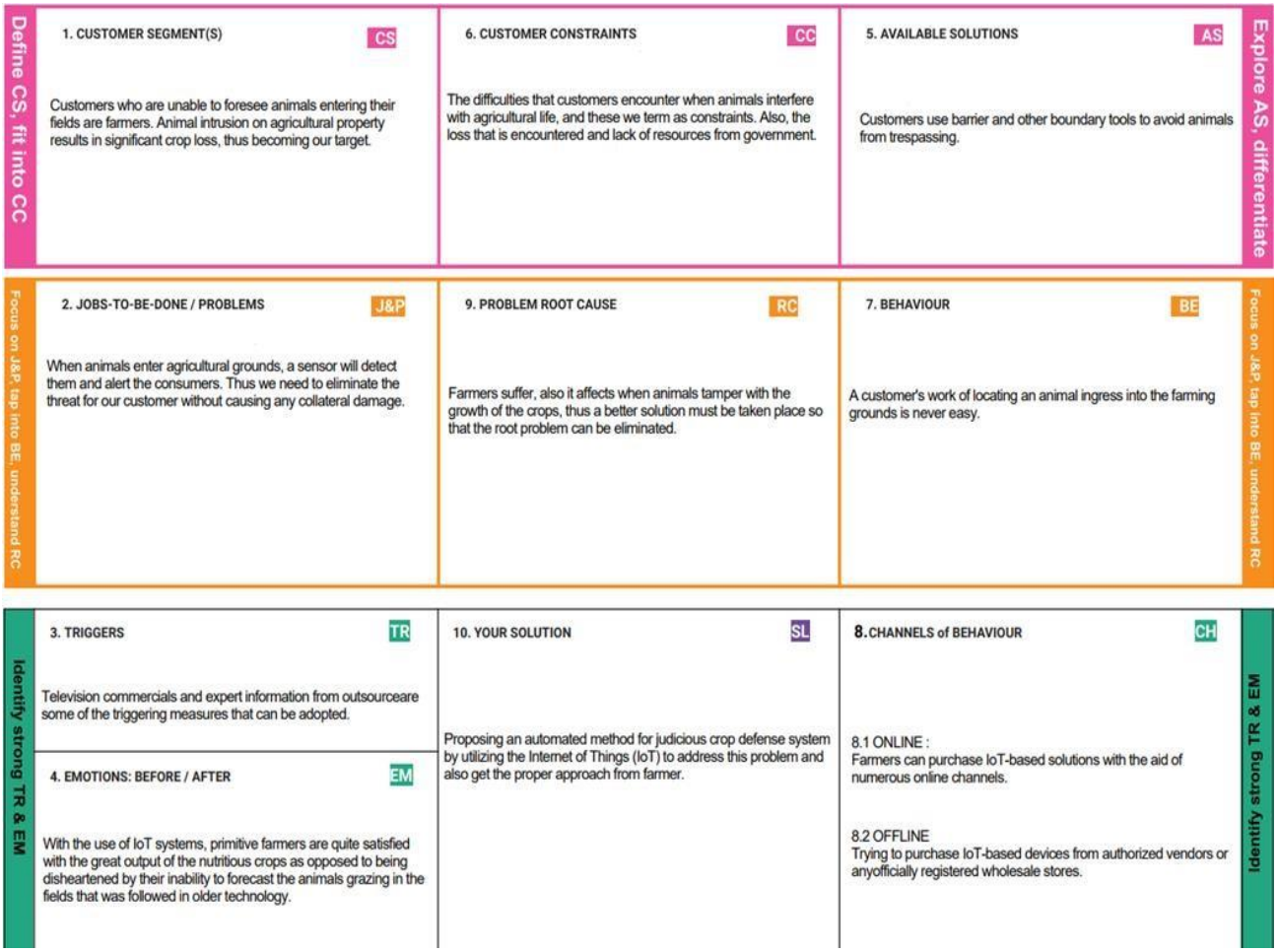
Although the system is already employed by many, but this project promises to give an accurate results and also act unique in many of the regions of operation.

The social impact or the customer satisfaction about this kind of works are to secure a field, farmers cannot barricade off vast tracts of land or stay there all day, they can take immediate action, which results in improved crop yields and increased profitability. Also limited from often visiting the farm and may feel more at peace with this approach.

The scalability of the project solutions are Highly reliable, reduction of cost, and fully/partial automatic that makes the system pretty smarter. The potential to expand because smart farming is a modern emerging technology, minimal soil disturbance, plant diversity, ongoing live plant/root, and livestock integration are examples of soil armor.

The important reason to introduce the new kind of system for the farmers to their cultivating fields are like the integrated system of protection of crops from the animals and birds and the measures of climate conditions , weather reports and intimate the owners of the fields to get maintained on a particular time .

3.4 PROBLEM SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

The functional requirements of the smart crop protection system for agriculture purpose are users visibility, user reception , user understanding and user action. The all functional requirements are tabled below with their sub requirements.

S.NO.	Functional Requirement.	Sub Requirement.
1.	User Visibility	Sends an SMS to the farmer via cloud service when it detects animals approaching the crop field and sounds an alert to entice them away.
2.	User Reception	The values of the Data, the SMS messages are delivered from temperature, humidity, and soil moisture sensors.
3.	User Understanding	To obtain information about the current state offarming land, based on sensor data value.
4.	User Action	Actions that user must take include agricultural residue destruction, deep ploughing, crop rotation, fertilisers, strip cropping, and scheduled planting activities.

4.2 NON – FUNCTIONAL REQUIREMNTS

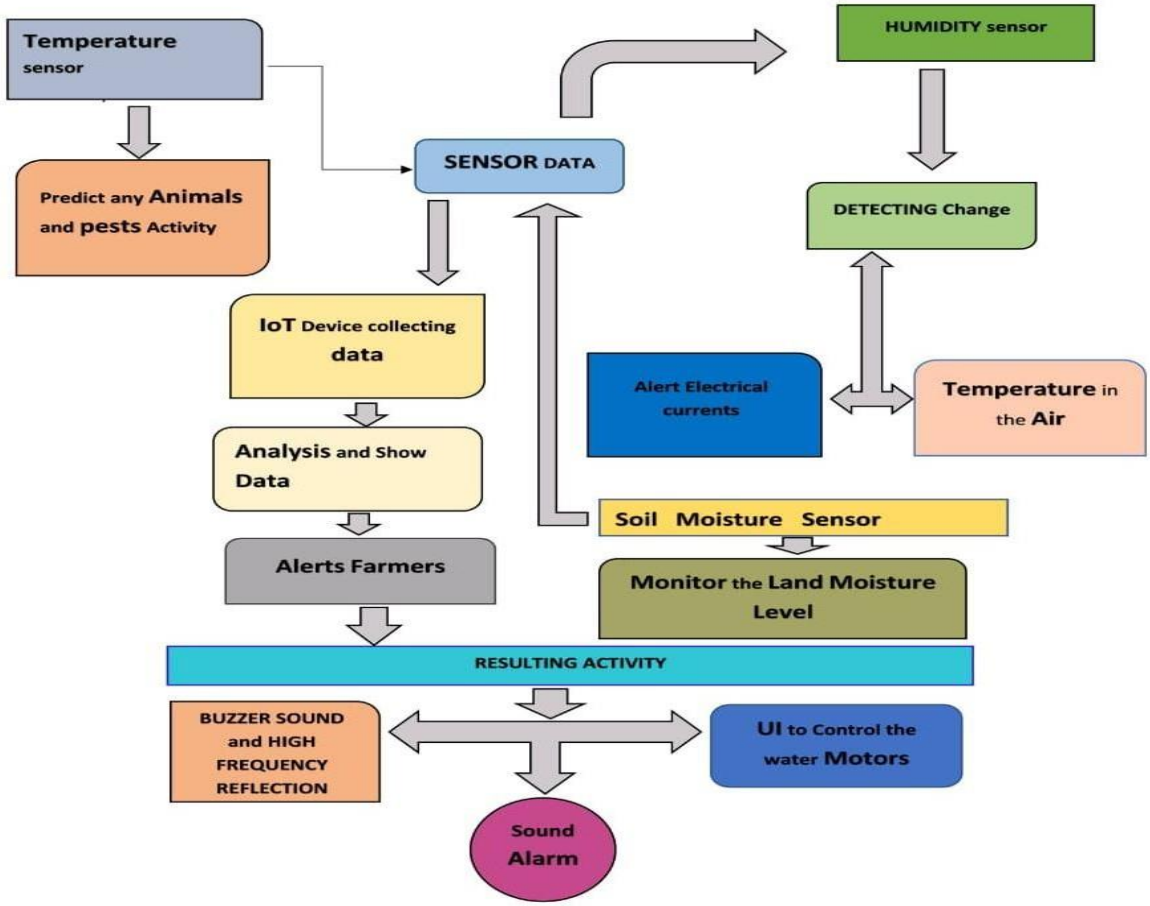
The non – functional requirements of the proposed project works are Usability, security , Reliability, Performance, Availability and Scalability . All the non-functional requirements are tabled below with their description.

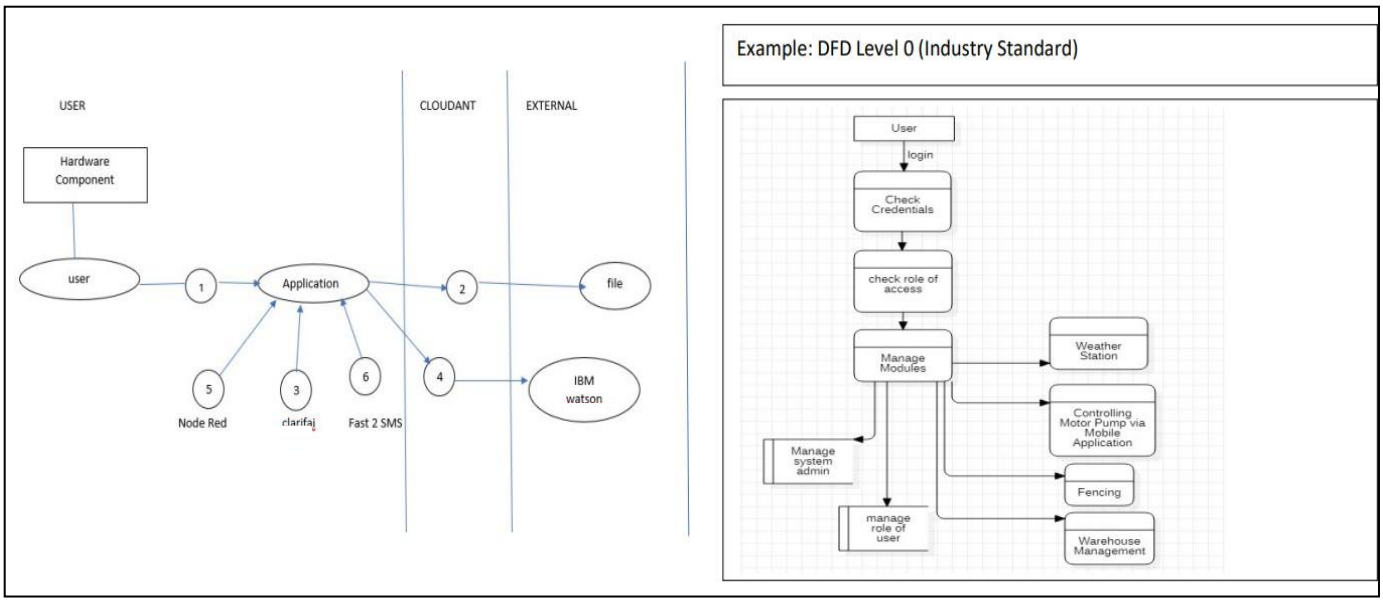
S.NO.	Non-Functional Requirement	Description
1.	Usability	Given the capabilities of mobile devices, Mobile Support Users must be able to interact in the same roles & duties on desktops & mobile devices, if possible.
2.	Security	Authorized users of the system who share information must be able to register and communicate securely ondevices with data that requires secure access.
3.	Reliability	Data could detect disturbances close to the field anddoesn't issue an erroneous warning signal.
4.	Performance	Regardless of the amount of data that is saved and thebackground analytics, it must offer users acceptable response speeds. Communications that are bidirectional and nearly real-time must be supported. The necessity to support industrial and ce protocols at the edge is connected to this requirements.
5.	Availability	Systems with high availability are necessary for IOT solutions and domains to operate around the clock. isnot a vital production application, thus if the IOT solution goes down, neither operations nor productionare affected.
6.	Scalability	The system must be able to handle growing load and data retention requirements based on an upscaling of the solution scope, such as additional buildings and manufacturing facilities.

5. PROJECT DESIGN

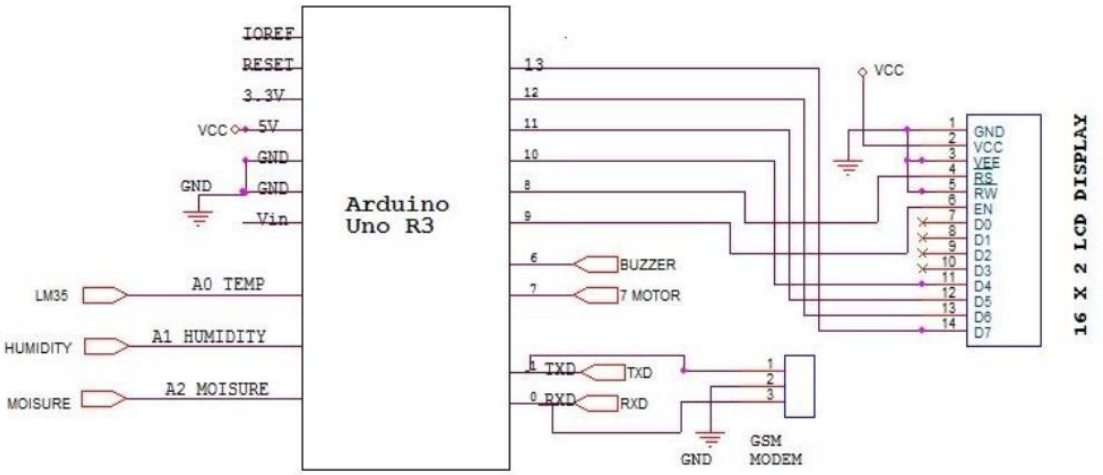
5.1 DATA FLOW DIAGRAM

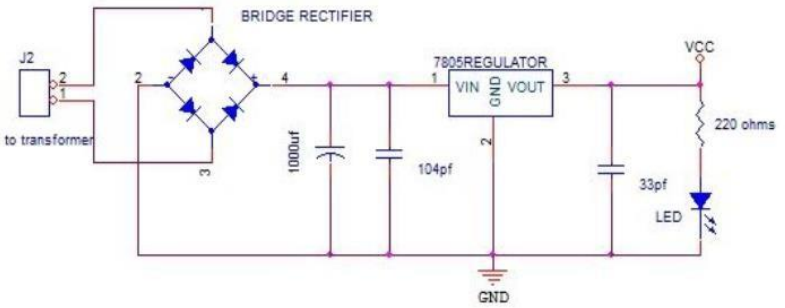
The data flow diagram for the proposed project work .

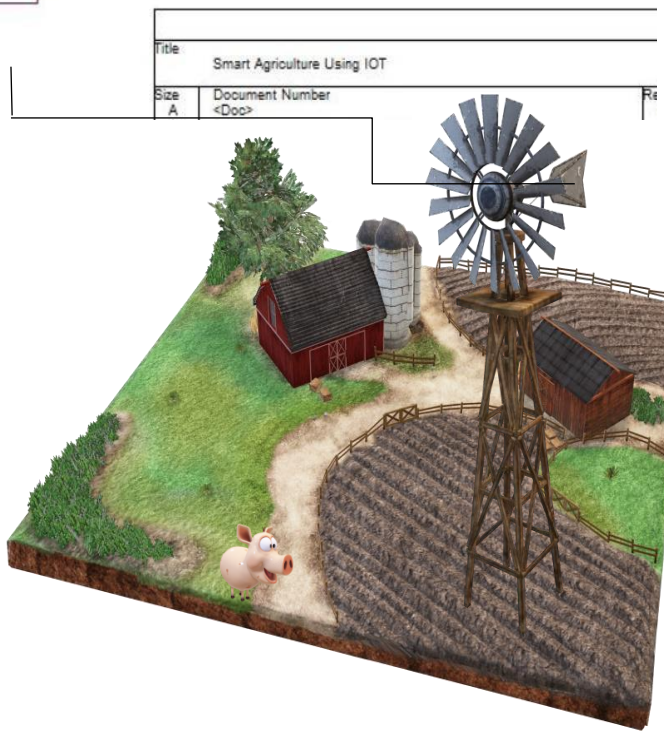




5.2 SOLUTION AND TECHNICAL ARCHITECTURE







5.3 USER STORIES

User Type	Functional requirement (Epic)	User Story number	User Story/Task	Acceptance criteria	Priority	Release
Customer (Mobileuser)	Registration	USN-1	User can enter into the web application	I can access my account /dashboard	High	Sprint 1
		USN-2	User can register their credentials like email id and password	I can receive confirmation email & click confirm	High	Sprint 1
	Login	USN-3	User can log into the application by entering email & password	I can login to my account	High	Sprint 1
	Dashboard	USN-4	User can view the temperature	I can view the data given by the device	High	Sprint 2
		USN-5	User can view the level of sensor monitoring value	I can view the data given by the device	High	Sprint 2
Customer (Web user)	Usage	USN-1	User can view the web page and get the information	I can view the data given by the device	High	Sprint 3
Customer	Working	USN-1	User act according to the alert given by the device	I can get the data work according to it	High	Sprint 3
		USN-2	User turns ON the water motors/Buzzer/Sound Alarm when occur the disturbance on field.	I can get the data work according to it		Sprint 4
Customer care Executive	Action	USN-1	User solve the problem when some faces any usage issues	I can solve the issues when some one fails to understanding the procedure	High	Sprint 4
Administration	Administration	USN-1	User store every information	I can store the gained information	High	Sprint 4

6. PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

In the project planning and scheduling the project backlogs , projects sprint schedule and estimations are tabled .

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points (40)	Priority (Low to High)	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the required dataset by entering my email, password, and confirming my password.	3	High	Giftson Jebasingh D
Sprint-1		USN-2	As a user, I will receive confirmation email andthe SMS once I have registered for the application	2	High	Elamathi G
Sprint-2	Cloud services	USN-3	As a user, I can register for the application through Facebook or any social media	1	Low	Jeyanesh V
Sprint-4		USN-4	As a user, I can register for the application through Gmail/web service	2	Medium	Giftson Jebasingh D
Sprint-3	Login	USN-5	As a user, I can log into the application networkby entering email & password	4	High	Jeyanesh V
Sprint-2	Pre processing	USN-6	As a farmer, the user must be able to find the system easy to access so pre-processes and other task must be perfect.	3	High	Durga Devi S
Sprint-1	Collecting Dataset	USN-7	To collect various sources of animal threats and keep developing a dataset.	3	Medium	Elamathi G
Sprint-4	Integrating	USN-8	To integrate the available dataset and keep improving the accuracy of finding animals	2	High	Durga Devi S
Sprint-3		USN-9	To find and use appropriate compiler to run andtest the data so that we can implement our program	1	Low	Jeyanesh V
Sprint-2		USN-10	Request Saveetha Engineering College to deploy the project in our campus and test	1	Low	Durga Devi S
Sprint-1	Training	USN-11	As programmer, we need to train our data perfectly so that the program runs smoothly	3	High	Giftson Jebasingh D
Sprint-3		USN-12	Train the data using out available services and IBM dataset from server and improve that	2	Medium	Elamathi G
Sprint-4	Coding	USN-13	To modify the code according to our programand improve the efficiency of that code	4	High	Durga Devi S
Sprint-2		USN-13	To improve performance	1	Low	Jeyanesh V
Sprint-2	Record	USN-5	To record the data and plot the graph to showthe characteristics officially	4	High	Durga Devi S
Sprint-1	Planning	USN-4	Plan the programming language and feasibility	3	Medium	Giftson Jebasingh D
Sprint-4		USN-14	Demonstrate the working and improve accuracy overall	2	Low	Durga Devi S

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	5 Days	19 Oct 2022	23 Oct 2022	20	23 Oct 2022
Sprint-2	20	5 Days	24 Oct 2022	28 Oct 2022	20	28 Oct 2022
Sprint-3	20	7 Days	29 Oct 2022	4 Nov 2022	20	4 Nov 2022
Sprint-4	20	7 Days	5 Nov 2022	11 Nov 2022	20	11 Nov 2022

6.2 SPRINT DELIVERY SCHEDULE

ACTIVITY LIST

S. No	Activity Title	Activity Description	Duration
1.	Understanding the Project Requirement	Assign the team members & create the repository in GitHub. Assign the task to each members and teach how to use and open access the GitHub and IBM Career Education.	1 Week
2.	Starting of Project	Advice student to attend classes of IBM portals create and develop an rough diagram based on the project description and gather information of IOT and IBM project.	1 week
3.	Attend classes	Team members & team lead must watch and learn from classes provided by IBM and Nalaiya Thiran and must gain access of MIT License for their project.	4 Week
4.	Budget and scope of the project	Budget & analyse the use of IOT in the project and discuss with the team for budget prediction to predict the favourability of the customer to buy the product for efficient use of the product among the environment.	1 week

7. CODING AND SOLUTIONING

7.1 FEATURE 1

Agriculture is the essential tool for any country as it not only increases the revenue but also is important for one's survival. The system proposed here is an advanced irrigation system approach that can be employed to the cultivation of various crops.

❑ The main objective of this system is to develop a system that can able to smartly operate the irrigation system and also monitor the status of the crop, detect the disease that through the leaf in the crop, that is mainly caused by animals in the fields. Alongside the usage it unites its advantages especially on the sector of execution time rate.

❑ This paper is structured upon Indian agriculture system and Indian weather conditions. This model applies the technology of the Internet of Things to implement real-time analytics of the collected data. Multiple sensors are deployed in the field to create an on-field dataset into the cloud system.

CODE :

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8,9,10,11,12,13);//rs,en,data pins d4 -d7
float TEMP;
int MOISURE,HUM;
const int buzzer=6;
const int motor=7;
int led=4,temp=0,i=0;
char str[30];
int aa=0, dt=0////temp;
int dh=0;///hum
int  s1=0,s3=0;
void setup()  {
lcd.begin(16,2);
```

```
Serial.begin(9600);
pinMode(buzzer, OUTPUT);
pinMode(motor, OUTPUT);
digitalWrite(buzzer, LOW);
digitalWrite(motor, LOW);
lcd.clear();
lcd.setCursor(0,0);lcd.print("IOT Based ");
lcd.setCursor(0,1);lcd.print("Agriculture Crop");
delay(5000);lcd.clear();
lcd.setCursor(0,0);lcd.print("Field Monitoring");
lcd.setCursor(0,1);lcd.print("Irrigation");
delay(5000);lcd.clear();

 lcd.setCursor(0,1);lcd.print("using GPRS ");
 delay(5000);lcd.clear();
 gsm_init();lcd.clear();
 digitalWrite(buzzer, LOW);
 digitalWrite(motor, LOW);
 }
 void loop() {
 aa=aa+1;
 digitalWrite(buzzer, LOW);
 lcd.clear();
 TEMP = analogRead(0);
 TEMP=(TEMP*500)/1023;
 lcd.setCursor(0,0);lcd.print("T:");lcd.setCursor(3,0);lcd.print(TEMP);delay(200);if(TEMP<50){dt=0;delay(100);}
 if(TEMP>50){
 dt=dt+1;
 if(dt==2){
 delay(200);digitalWrite(buzzer, HIGH);delay(200);digitalWrite(buzzer, LOW);
 delay(200);digitalWrite(buzzer, HIGH);delay(200);digitalWrite(buzzer, LOW);lcd.clear();send_gprs();delay(500);
 delay(200);digitalWrite(buzzer, HIGH);delay(200);digitalWrite(buzzer, LOW);
 lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS");lcd.setCursor(0,1);lcd.print("TEMP ALERT");
 Serial.println("AT+CMGF=1");delay(400);
 Serial.println("AT+CMGS=\"9148300815\"");delay(400);
 Serial.println("Over Temperature\n");delay(100);
 Serial.print("Temp=");delay(100);Serial.print(TEMP);delay(500);Serial.write(26);delay(500);
 Serial.print("AT\r\n");delay(1000);Serial.print("AT\r\n");delay(1000);Serial.println("AT+CMGF=1");delay(1000);
```

7.2 FEATURE 2

The cloud system scraps the required data from the meteorological center and compares it with the on-field data set. The analysis is then done using the concepts of data science to determine the amount of water to be released according to each crop type in the field. The proposed system is highly efficient and economically feasible. The system also provides a mobile application which helps the farmer to track the developments occurring throughout the field. In this project we have analysed different image parameters or features to identifying different plant leaves diseases to achieve the best accuracy. Obviously this improves the damage control by insect attacks. Previously plant disease detection is done by visual inspection of the leaves or some chemical processes by experts. For doing so, a large team of experts as well as continuous observation of plant is needed, which costs high when we do with large farms.

CODE:

```
-- phpMyAdmin SQL Dump
-- version 4.9.5
-- https://www.phpmyadmin.net/
--
-- Host: localhost:3306
-- Generation Time: Sep 24, 2022 at 11:38 AM
-- Server version: 10.3.24-MariaDB-cll-lve
-- PHP Version: 7.3.6

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `iotbabyc_agriiot  - IBM Cloud`
```

```
CREATE TABLE `sens` (
  `id` bigint(30) NOT NULL,
  `temp` varchar(120) NOT NULL,
  `hum` varchar(120) NOT NULL,
  `mos` varchar(120) NOT NULL,
  `createdat` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `sens`
--

INSERT INTO `sens` (`id`, `temp`, `hum`, `mos`, `createdat`) VALUES
(1, '30', '40', '120', '2022-09-24 11:38:06'),
(2, '22.48', '35', '1', '2022-09-25 05:32:55'),
(3, '24.44', '35', '1', '2022-09-25 05:33:28'),
(4, '32.26', '35', '1', '2022-09-25 05:34:56'),
(5, '55.72', '35', '1', '2022-09-25 05:35:41'),
(6, '34.70', '34', '1', '2022-09-25 05:36:32'),
(7, '23.95', '34', '1', '2022-09-25 05:37:42'),
(8, '29.81', '34', '182', '2022-09-26 05:38:52'),
(9, '29.33', '33', '172', '2022-09-26 05:40:02'),


--
-- Indexes for table `sens`
--
ALTER TABLE `sens`
  ADD PRIMARY KEY (`id`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `sens`
--
ALTER TABLE `sens`
  MODIFY `id` bigint(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=42;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

8. TESTINGS

8.1 TESTCASES

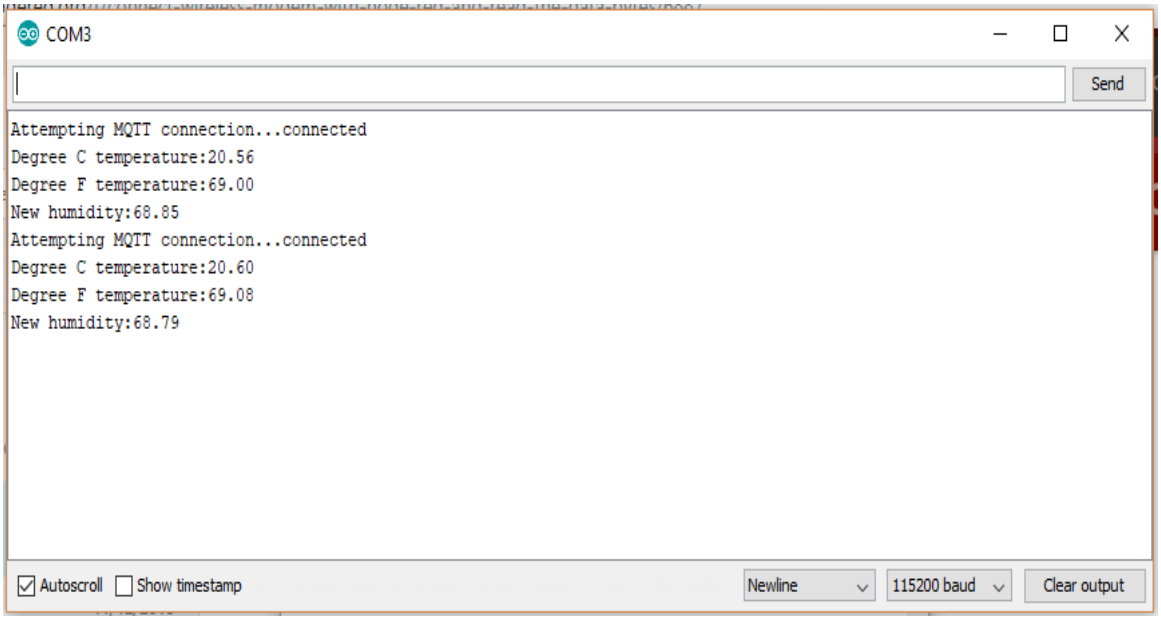
The training output of the source code seen in the above chapter was executed successfully and got the output .

Training Output:

Running output obtained from phpMyAdmin SQL:

```
-- phpMyAdmin SQL Dump
-- version 4.9.5
-- https://www.phpmyadmin.net/
--
-- Host: localhost:3306
-- Generation Time: Sep 29, 2020 at 04:46 PM
-- Server version: 10.3.24-MariaDB-cll-lve
-- PHP Version: 7.3.6

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `iotbabyc_agriiot`
--

-- --------------------------------------------------------

--
-- Table structure for table `bulbs_data`
--

CREATE TABLE `bulbs_data` (
  `id` bigint(30) NOT NULL,
  `motor` varchar(120) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `bulbs_data`
--

INSERT INTO `bulbs_data` (`id`, `motor`) VALUES
(1, 'ON(Manual)');
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

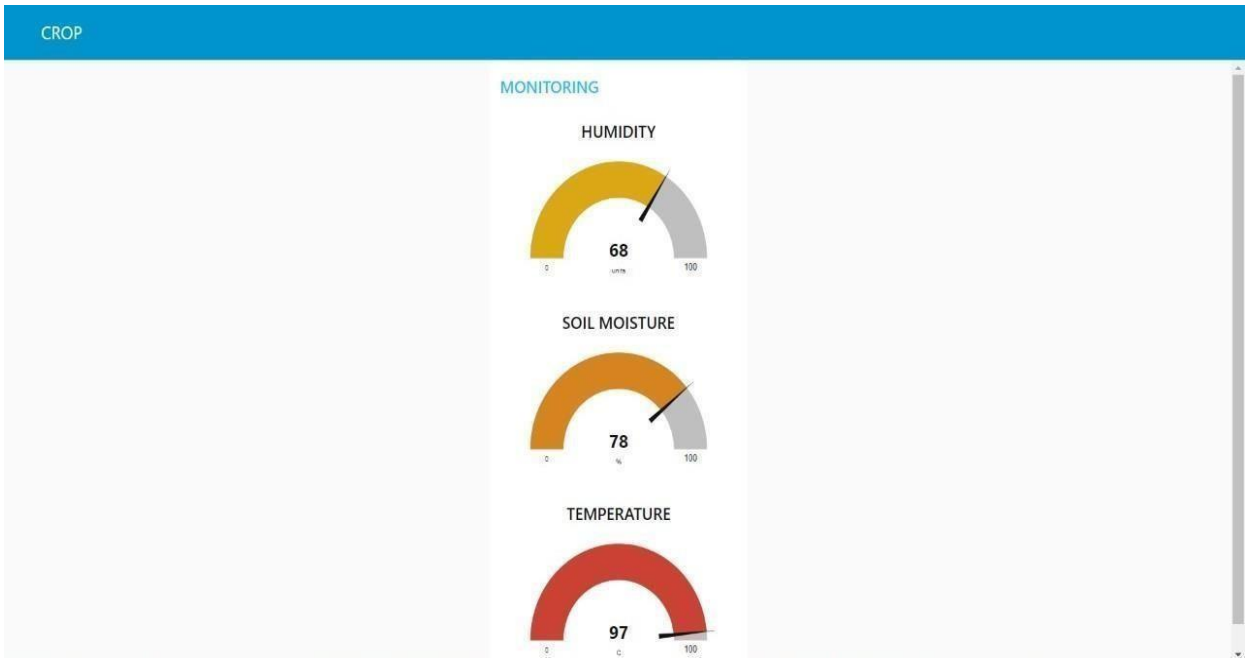
Consoled output based on the approach:



8.2 USER ACCEPTANCE TESTING

The obtained output get stimulated and got the result with the random values.

Display the image and pre-process the level of the Node-RED web UI and display the temperature, humidity, and soil moisture levels. Integrate the buttons in the UI to control the Motors



9. RESULTS

We have successfully completed the project works that the integrated systems of crop protection from the animals and measuring the climate conditions of the field and to alert the farmer using the web application.

10. ADVANTAGES AND DISADVANTAGES

10.1 ADVANTAGES :

- All the data like climatic conditions and changes in them, soil or crop conditions everything can be easily monitored.
- Risk of crop damage can be lowered to a greater extent.
- Many difficult challenges can be avoided making the process automated and the quality of crops can be maintained.
- The process included in farming can be controlled using the web applications from anywhere, anytime.
- Precision Farming that is farming processes can be made more controlled and accurate.
- Live monitoring can be done of all the processes and the conditions on the agricultural field.
- All the controls can be made just on the click.
- Quality can be maintained.

10.2 DISADVANTAGES:

- Smart Agriculture requires internet connectivity continuously, but rural parts cannot 22ulfil this requirement.
- Any faults in the sensors can cause great loss in the agriculture, due to wrong records and the actions of automated processes.
- IoT devices need much money to implement.

11 . CONCLUSION

An IoT Based Web Application is built for smart agricultural system using Watson IoT platform, Watson simulator, IBM cloud and Node-RED. It will be very useful for the farmers to get alerted and to get an field situation when he/she is far from the cultivating field . It will notify the climate conditions at the current state and get alerted and also protect the fields and crops from the animals and birds by get alerting the farmer by the same application .

This was an integrated application system for the farmers that they will get very much easier, eco friendly and also consume very low cost which can be affordable by the farmers.

12. FUTURE SCOPE

In future due to more demand of good and more farming in less time, for betterment of the crops and reducing the usage of extravagant resources like electricity and water IoT can be implemented in most of the places. The system get implemented with some other new technology and with some other advanced features if the need of this system get increased.

13. APPENDIX

13.1 SOURCE CODE

Agri motor :

```
----phpMyAdmin SQL Dump

 -- version 4.9.5

 -- https://www.phpmyadmin.net/

 --

-- Host: localhost:3306

 -- Generation Time: Sep 29, 2020 at 04:46 PM

-- Server version: 10.3.24-MariaDB-cll-lve

 -- PHP Version: 7.3.6


SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";

SET AUTOCOMMIT = 0;

START TRANSACTION;

SET time_zone = "+00:00";

 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
```

```
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;

/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

 /*!40101 SET NAMES utf8mb4 */;

-- Database: `iotbabyc_agriiot`

-- --

 ---------------------------------------------------------- --

-- Table structure for table `bulbs_data`

--

CREATE TABLE `bulbs_data` (

`id` bigint(30) NOT NULL,

`motor` varchar(120) NOT NULL

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

 -- -- Dumping data for table `bulbs_data`

 --

 INSERT INTO `bulbs_data` (`id`, `motor`) VALUES(1, 'ON(Manual)');

 COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Sensor Code:

```
-- phpMyAdmin SQL Dump

-- version 4.9.5

-- https://www.phpmyadmin.net/

 --
```

```
-- Host: localhost:3306

-- Generation Time: Sep 24, 2022 at 11:38 AM

-- Server version: 10.3.24-MariaDB-cll-lve

-- PHP Version: 7.3.6

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";

SET AUTOCOMMIT = 0;

START TRANSACTION;

SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;

/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

/*!40101 SET NAMES utf8mb4 */;

--

-- Database: `iotbabyc_agri6refdhjhiot

- IBM Cloud`

-- --

--------------------------------------------------------

--

-- Table structure for table `sens`

-- CREATE TABLE `sens` (

`id` bigint(30) NOT NULL,

`temp` varchar(120) NOT NULL,

`hum` varchar(120) NOT NULL,

`mos` varchar(120) NOT NULL,
```

```
`createdat` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp() )
ENGINE=MyISAM DEFAULT CHARSET=latin1;

--

 -- Dumping data for table `sens`

-- INSERT INTO `sens` (`id`, `temp`, `hum`, `mos`, `createdat`) VALUES

(1, '30', '40', '120', '2022-09-24 11:38:06'),

(2, '22.48', '35', '1', '2022-09-25 05:32:55'),

(3, '24.44', '35', '1', '2022-09-25 05:33:28'),

(4, '32.26', '35', '1', '2022-09-25 05:34:56'),

(5, '55.72', '35', '1', '2022-09-25 05:35:41'),

(6, '34.70', '34', '1', '2022-09-25 05:36:32'),

(7, '23.95', '34', '1', '2022-09-25 05:37:42'),

(8, '29.81', '34', '182', '2022-09-26 05:38:52'),

(9, '29.33', '33', '172', '2022-09-26 05:40:02'),

(10, '29.81', '33', '171', '2022-09-26 05:41:12'),

(11, '29.33', '33', '209', '2022-09-26 05:42:08'),

(12, '50.34', '33', '201', '2022-09-26 05:43:24'),

(13, '36.17', '40', '200', '2022-09-26 05:44:00'),

(14, '30.79', '36', '200', '2022-09-26 05:45:10'),

(15, '30.30', '34', '199', '2022-09-26 05:46:19'),

(16, '31.77', '32', '199', '2022-09-27 05:47:29'),

(17, '30.30', '31', '199', '2022-09-28 05:48:39'),

(18, '29.33', '33', '196', '2022-09-28 06:07:01'),

(19, '75.76', '33', '196', '2022-09-29 06:07:54'),

(20, '32.75', '33', '195', '2022-09-29 06:08:39'),
```

```
(21, '31.28', '32', '195', '2022-09-29 06:09:52'),

(22, '30.30', '32', '195', '2022-09-29 06:11:04'),

(23, '30.79', '32', '195', '2022-09-29 06:12:17'),

(24, '21.99', '32', '88', '2022-09-29 06:28:48'),

(25, '20.53', '33', '84', '2022-09-29 06:30:16'),

(26, '63.54', '34', '84', '2022-09-30 06:31:04'),

(27, '23.95', '33', '83', '2022-09-30 06:31:55'),

(28, '23.46', '33', '82', '2022-09-30 06:33:10'),

 (29, '26.39', '33', '81', '2022-10-1 06:34:23'),

(30, '21.99', '33', '1', '2022-10-5 07:41:57'),

(31, '63.05', '34', '1', '2022-10-8 07:43:24'),

(32, '63.05', '35', '1', '2022-10-8 07:43:51'),

(33, '30.79', '35', '220', '2022-10-9 07:45:05'),

(34, '30.79', '35', '220', '2022-10-9 07:45:37'),

(35, '30.79', '35', '220', '2022-10-10 07:46:18'),

36, '22.97', '35', '109', '2022-10-13 07:47:32'),

(37, '25.90', '52', '123', '2022-10-15 07:48:32'),

(38, '24.44', '35', '122', '2022-10-17 07:49:10'),

39, '31.28', '33', '115', '2022-10-18 07:50:23'),

(40, '31.28', '32', '137', '2022-10-19 07:51:36'),

(41, '35.68', '29', '1', '2022-10-20 09:08:39');

 -- -- Indexes for dumped tables

-- -- -- Indexes for table `sens`

-- ALTER TABLE `sens`

ADD PRIMARY KEY (`id`);
```

```
--

-- AUTO_INCREMENT for dumped tables

--

--

-- AUTO_INCREMENT for table `sens`

-- ALTER TABLE `sens`

MODIFY `id` bigint(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=42;

COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```