

SMART AGRICULTURE USING IOT AND ARTIFICIAL INTELLIGENCE

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering in Computer Science Engineering

By

Pradyum Raj D (37110585)

Pream Kumar P (37110594)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600 119**

MARCH – 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)



Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the Bonafide work of **Pradyum Raj D (37110585)** and **Pream Kumar P (37110594)** who carried out the project entitled "**A SMART AGRICULTURE USING IOT AND ARTIFICIAL INTELLIGENCE**" under my supervision from **August 2020** to **March 2021**.

Internal Guide

Dr. V. Ashok Kumar, M.E., Ph.D.,

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **Pradyum Raj D (37110585), Pream Kumar P (37110594)** hereby declare that the Project Report entitled “**SMART AGRICULTURE USING IOT AND ARTIFICIAL INTELLIGENCE**” is done by me under the guidance of **Dr. V. Ashok Kumar, M.E., Ph.D.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D.,** and **Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. V. Ashok Kumar, M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

CERTIFICATE





ABSTRACT

Most Agriculture is one of the most essential factors in human's life. Nowadays, as the population increases so as the demand grows. There is an adequate need for the government to support farmers, but the bitter truth is farmers are not able to be profited out of this. The ultimate solution for developing this system is to integrate smart technology and agriculture together. In our project we will be trying to integrate smart system into agriculture with the use of IOT devices and Machine learning algorithm. In the end we will use web app for predicting the price of the yielded crops so as to be transparent with the profit cut off the middleman and make farmers live their life in peace. The major use of introducing this project is to increase the food production and reduce the resource getting used like water etc. In this we invent a new technology called IOT and AI. With these technologies the farmers can predict the price of the crop so that without the intermediate the farmers can earn their profit by their own using our application. In India this is a very big problem now with our application all these problems can be solved.

TABLE OF CONTENTS

	ABSTRACT	V
	LIST OF FIGURES	IX
	LIST OF ABBREVIATIONS	X
CHAPTER NO	TITLE	PAGE NO
1.	INTRODUCTION	1
	1.1 OVERVIEW OF IOT	1
	1.2 MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE	2
	1.3 INNOVATION AND AGRICULTURE	2
	1.4 SCOPE OF THE PROJECT	3
2.	LANGUAGE OVERVIEW	4
	2.1 INTRODUCTION TO PYTHON FOR ML	4
	2.1.1 COLLECTION FRAMEWORK	6
	2.2 MY SQL	7
	2.2.1 FLASH	8
	2.2.2 SQL ALCHEMY	9
	2.3 POSTMAN	11
	2.4 SK LEARN	11
3.	LITERATURE SURVEY	12
	3.1 ANALYSIS OF THE LITERATURE	12
	3.2 LITERARY REVIEWS	13

4.	SYSTEM DESIGN AND REQUIREMENTS	17
	4.1 SYSTEM DESIGN	17
	4.1.1 INPUT DESIGN	17
	4.1.2 OUTPUT DESIGN	18
	4.2 SYSTEM REQUIREMENTS	19
	4.2.1 HARDWARE REQUIREMENTS	19
	4.2.2 SOFTWARE REQUIREMENTS	19
	4.3 REQUIREMENT ANALYSIS	
	4.3.1 FUNCTIONAL REQUIREMENTS	
	4.3.2 NON-FUNCTIONAL REQUIREMENTS	
5.	BUILDING SMART AGRICULTURE	22
	5.1 ARCHITECTURAL DESIGN	22
	5.2 MODULE DESCRIPTION	23
	5.3 MAKING UP OF THE SMART AGRICULTURE SYSTEM FRAMEWORK	24 25
	5.3.1 MACHINE LEARNING FOR REDICTION	29 30
	5.3.2 WEB APPLICATION	31
	5.3.3 IOT FRAMEWORK	32
	5.4 CROP PREDICTION	35
	5.5 CROP PRICE PREDICTION	
6.	RESULTS AND DISCUSSIONS	37
7.	CONCLUSION AND FUTURE WORK	38

REFERENCES	39
APPENDIX	
A. SAMPLE CODE	41
B. SCREENSHOTS	55
C. PUBLICATION WITH PLAGIARISM REPORT	60

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
Fig 1.1	IOT and AI function for Smart agriculture and high growth for better yielding	3
Fig 2.1	Python code to run the application	5
Fig 2.2	Collection of app images	6
Fig 2.3	Working Database Engine	8
Fig 2.4	SQL server management studio	9
Fig 2.5	Creating a new DB	9
Fig 2.6	Creating a Web Application Using Postman	11
Fig 5.1	Architectural Diagram	22
Fig 5.2	Program created to predict the crop	25
Fig 5.3	After predicting the crop	26
Fig 5.4	Use case web application screenshot	26
Fig 5.5	sensor screenshot	27
Fig 5.6	Crop prediction screenshot	28

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
IOT	Internet of Things
OS	Operating System
MYSQL	System Query Language
RDBMS	Relational Database Management System
PY	Python Programming Language
PHP	Hypertext Preprocessor
HTML	Hyper Text Machine Language

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF IOT

By implementing the latest sensing and IOT technologies in agriculture practices, every aspect of traditional farming method can be fundamentally changed. Currently, seamless integration of wireless sensors and the IOT in smart agriculture can raise agriculture to levels which were previously unimaginable. By following the practices of smart agriculture, IOT can help to improve the solutions of many traditional farming issues, like drought response, yield optimization, land suitability, irrigation, and pest control. It lists a hierarchy of major applications, services and wireless sensors being used for smart agriculture applications. While, major instances in which the advanced technologies are helping at various stages to enhance overall efficiency are discussed below

The IoT has made remarkable progress in many industries, including farming sectors like poultry, fishing, etc. but when we talk about agriculture, the communication facilities like base stations or Wi-Fi are very limited, which prevents the growth of the IoT in this sector. Such communication infrastructure and related facilities are even worst in developing counties and rural areas, which is one of the major hurdles when introducing the IoT in the agriculture industry. The data acquired through the wireless sensors cannot be transmitted in the absence of reliable communication infrastructure.

Smart farming based on IOT technologies enables growers and farmers to reduce waste and enhance productivity ranging from the quantity of fertilizer utilized to the number of journeys the farm vehicles have made, and enabling efficient utilization of resources such as water, electricity, etc. IOT smart farming solutions is a system that is built for monitoring the crop field with the help of sensors (light, humidity, temperature, soil moisture, crop health, etc.) and automating the irrigation system. The farmers can monitor the field conditions from anywhere. They can also select between manual and automated options for taking necessary actions based on this data. For example, if the soil moisture level decreases, the farmer can deploy sensors to start the irrigation. Smart farming is highly efficient when compared with the conventional approach.

Data collected by smart agriculture sensors, in this approach of farm management, a key component are sensors, control systems, etc. This data can be used to track the state of the business in general as well as to know the performance of the crops, equipment efficiency. The ability to foresee the output of production allows to plan for better product distribution.

1.2 MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

The Machine learning and analytics are used to mine data for trends. In farming, machine learning is used, for example, to predict which genes are best suited for crop production. This has been giving growers all over the world the best seed varieties, those which are highly suitable to respective locations and climate conditions. Machine-learning algorithms, on the other hand, have indicated which products are of high demand and which products are currently unavailable in the market. Thus, for the farmer, this has given valuable clues for future farming. Recent advances in machine learning and analytics will make it possible for farmers to accurately classify their products and weed out fewer desirable crops before they arrive to customers.

Improvement of AI in these days are helpful for developing tools and robots that can easily perform multiple tasks in farming fields. This type of robot is trained to control weeds and harvest crops at a faster pace with higher volumes compared to humans. These types of robots are trained to check the quality of crops and detect weed with picking and packing of crops at the same time. These robots are also capable to fight with challenges faced by agricultural force labor.

1.3 INNOVATION AND AGRICULTURE

Innovative technology has changed the face of agriculture throughout the world. From social networks to cutting-edge tech like self-driving vehicles, farming is going digital. Thereby, giving stakeholders a lot of innovative solutions for effective farming. More focus has been placed by governmental and international bodies to achieve sustainability in agriculture. Creating more sustainable farming practices requires adopting new technologies for crop management, pest control, quality control, and integrated disease

management. The new technologies allow current and future generations of farmers to grow without compromising the needs of the earth.

1.4 SCOPE OF THE PROJECT

The proposed system is a web-based application, which maintains a centralized repository of all scheme related information. This makes it easy to search for the desired information. The main scope of the project is to provide technological solutions to agriculture. The objective is to improve the agricultural yield by introducing IOT (technology) with fewer resources and labor efforts. To provide the necessary support system to agriculture by providing IOT and AI that can handle different activities relating to agriculture and to promote and help an amateur to do agriculture

►

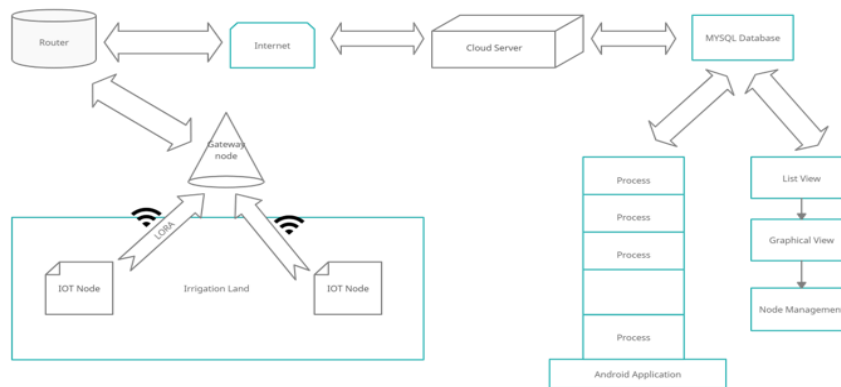


Fig: 1.1 IOT and AI function for Smart agriculture and high growth for better yielding

ADVANTAGES OF AI AND IOT:

- Enhanced Customer Relationship. With the correct implementation, the benefit from the combination of AI and IoT is not confined only to the employees but also helps to enhance the experience of customers
- Cost Effective.
- Increased Operational Efficiency.

- Highly Secured and Safe.
- High growth.
- Focus on New Products and Services

CHAPTER 2

LANGUAGE OVERVIEW

2.1 INTRODUCTION TO PYTHON FOR ML

Python has a simple syntax similar to the English language. Python has syntax that allows developers to write programs with fewer lines than some other programming languages. Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. In this project we preferred python as it is simple and efficient to obtain the desired output.

Application of Python:

Python widely used in every corner of world and of human life. python is not only used in software but is also widely used in designing hardware controlling software components. There are more than 930 million PY downloads each year and 3 billion mobile apk files run python.

Following are some other usages of Python:

- Scientific and Numeric Applications.
- Education programs and training courses
- Mobile Operating System like Android
- Artificial Intelligence and Machine Learning
- Robotics and games etc.
- Language Development.

Features of Python:

The prime reason behind creation of Python was to bring portability and security feature into a computer language. Beside these two major features, there were many other features that played an important role in molding out the final form of this outstanding language. Those features are: -

1) Simple

Python is easy to learn and its syntax is quite simple, clean and easy to understand. The confusing and ambiguous concepts of C++ are either left out in Python or they have been re-implemented in a cleaner way.

2) Object Oriented

In Python everything is Object which has some data and behavior. Python can be easily extended as it is based on Object Model.

3) Robust

Python makes an effort to eliminate error prone codes by emphasizing mainly on compile time error checking and runtime checking. But the main areas which Python improved were Memory Management and mishandled Exceptions by introducing automatic Garbage Collector and Exception Handling.

4) Platform Independent

Unlike other programming languages such as C, C++ etc. which are compiled into platform specific machines. Python is guaranteed to be write-once, run-anywhere language. On compilation python program is compiled into byte code. This byte code is platform independent and can be run on any machine, plus this byte code format also provide security. Any machine with python Runtime Environment can run python Programs.

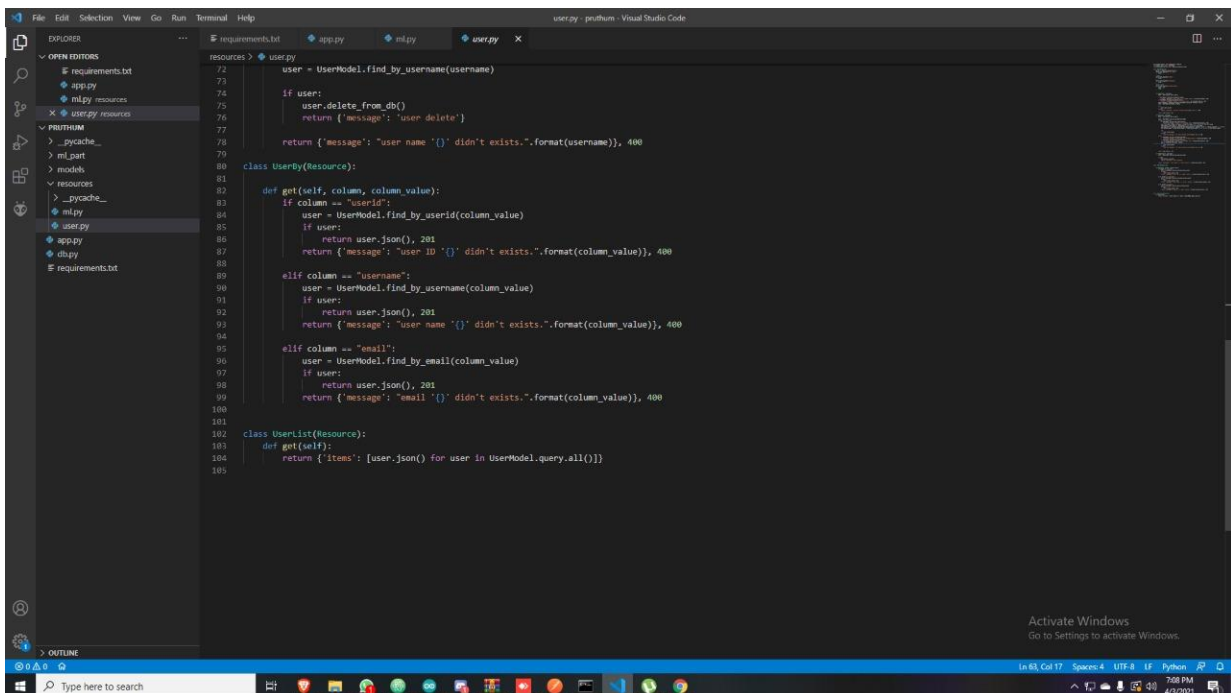


Fig: 2.1 Python code to run the application

5) Secure

When it comes to security, Python is always the first choice. With Python secure features it enables us to develop virus free, temper free system. Python program always runs in Python runtime environment with almost null interaction with system OS, hence it is more secure.

6) Multi-Threading

Python multithreading feature makes it possible to write program that can do many tasks simultaneously. Benefit of multithreading is that it utilizes same memory and other resources to execute multiple threads at the same time, like While typing, grammatical errors are checked along.

7) Architectural Neutral

Compiler generates byte codes, which have nothing to do with a particular computer architecture, hence a Python program is easy to interpret on any machine.

8) Portable

Python Byte code can be carried to any platform. No implementation dependent features. Everything related to storage is predefined, example: size of primitive data types.

10) High Performance

Python is an interpreted language, so it will never be as fast as a compiled language like C or C++. But Python enables high performance with the use of just-in-time compiler.

2.1.1 Collection framework

Collection framework was not part of original Python release. Collections was added to PY 1.2. Prior to Python 2, Python provided adhoc classes such as Dictionary, Vector, Stack and Properties to store and manipulate groups of objects. Collection framework provides many important classes and interfaces to collect and organize an app in the objects.

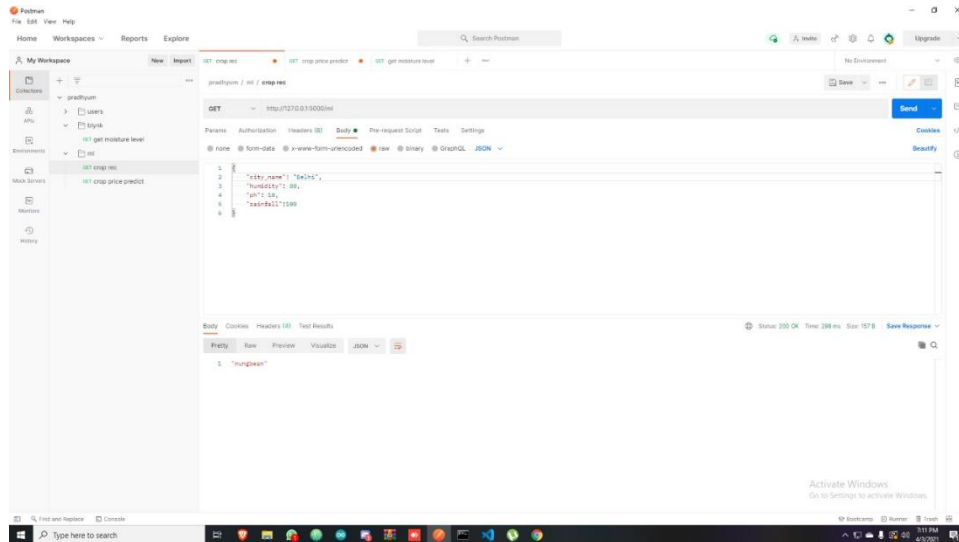


Fig: 2.2 Collection of app images

2.2 MY SQL

MySQL, officially, but also called "My Sequel" is the world's most widely used open-source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases, though SQLite probably has more total embedded deployments. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open-source web application software stack (and other 'AMP' stacks).

T-SQL expands on the SQL standard to include procedural programming, local variables, and various support functions for string processing, data processing, mathematics, etc. SQL Server consists of a Database Engine and a Management Studio (and lots of other stuff which we will not mention here). The Database engine has no graphical interface - it is just a service running in the background of your computer (preferable on the server). The Management Studio is graphical tool for configuring and viewing the information in the database. It can be installed on the server or on the client (or both). SQL Server Management Studio can also be used to create a new database, alter any existing

database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

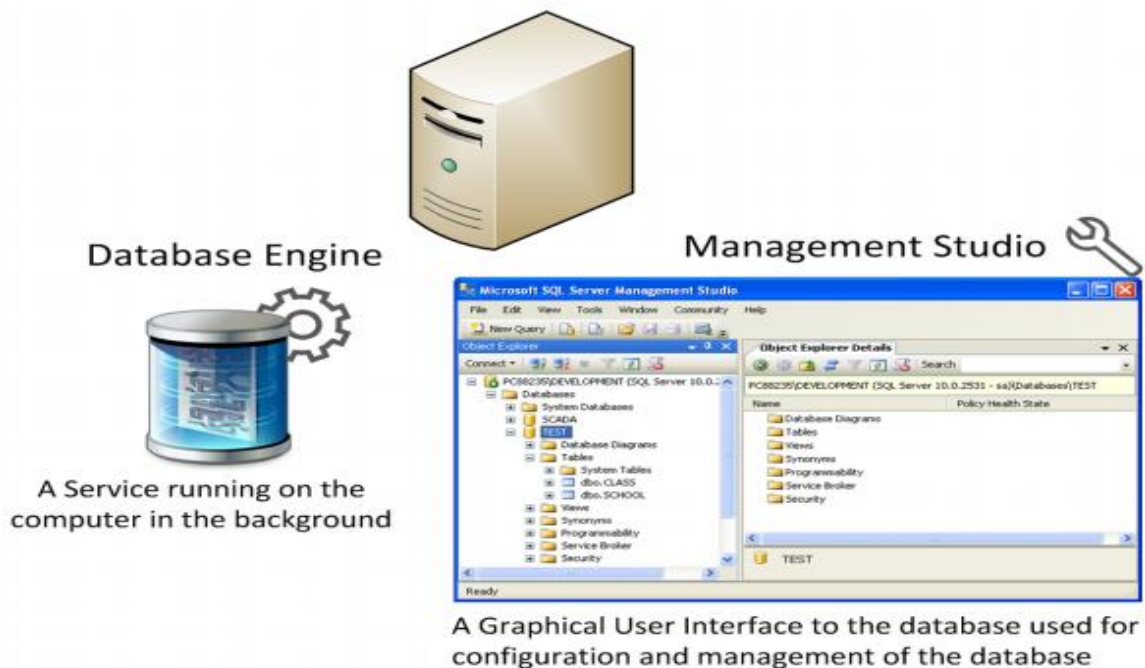


Fig: 2.3 Working Database Engine

2.2.1 Flash

SQLAlchemy is an extension for Flask that adds support for SQLAlchemy to your application. It aims to simplify using SQLAlchemy with Flask by providing useful defaults and extra helpers that make it easier to accomplish common tasks. So with the help of the SQLAlchemy extension we have created the SQLAlchemy extension so by using that the app is created it is the intermediate for the app and the coding language with that we have created the extension.

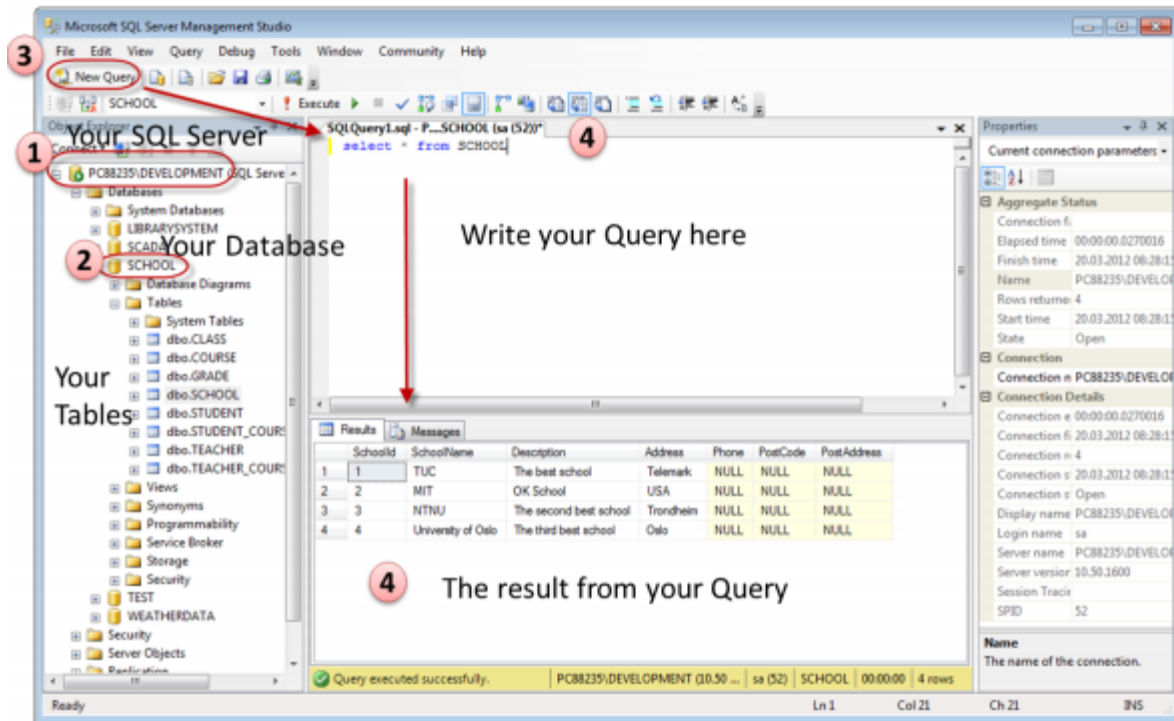


Fig: 2.4 SQL Server management studio

2.2.2 SQL Alchemy

SQLAlchemy is the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL. SQL databases behave less like object collections the more size and performance start to matter; object collections behave less like tables and rows the more abstraction starts to matter. SQLAlchemy aims to accommodate both of these principles. SQLAlchemy considers the database to be a relational algebra engine, not just a collection of tables. Rows can be selected from not only tables but also joins and other select statements; any of these units can be composed into a larger structure. SQLAlchemy's expression language builds on concept.

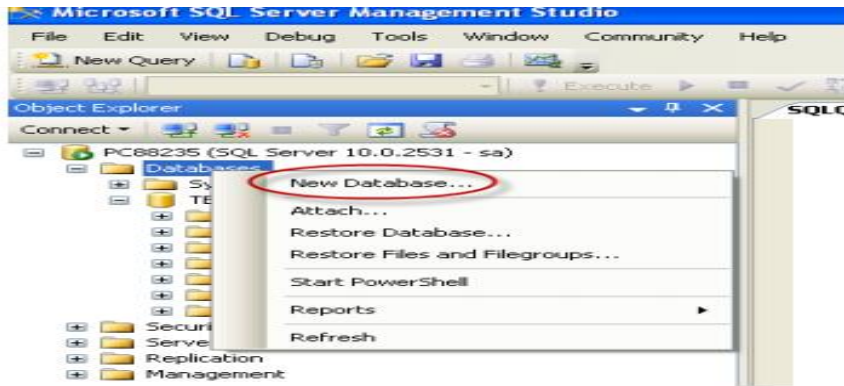


Fig: 2.5 Creating a new DB

The SQL Alchemy is a library that facilitates the communication between Python programs and databases. Most of the times, this library is used as an Object Relational Mapper (ORM) tool that translates Python classes to tables on relational databases and automatically converts function calls to SQL statements. The steps to create a my sql alchemy are

1. INSTALL FLASH
2. IMPORT SQL ALCHEMY
3. CREATE A FLASH APPLICATION
4. CREATE AN OBJECT OF SQL ALCHEMY
5. DATABASE MENTIONED IN URI

INSTALL FLASH

Install Flask-SQL Alchemy extension. The command to install flash is
 pip install flask-sql alchemy

Import SQL Alchemy

Then we need to import SQL Alchemy class from this module.

```
from flask_sqlalchemy import SQL Alchemy
```

Create a FLASH Application

Then we have to create a Flask application object and set URI for the database to be used.

```
App=Flask(__name__)app.config['SQLALCHEMY_DATABASE_URI']  
='SQLite:///students.sqlite3'
```

Create an Object of SQL Alchemy

Then create an object of SQL Alchemy class with application object as the parameter. This object contains helper functions for ORM operations. It also provides a parent Model class using which user defined models are declared. In the snippet below, a student's model is created.

```
db SQLAlchemy(app)class students(db.Model): id = db.Column('student_id', db.Integer,  
primary_key = True) name = db.Column(db.String(100)) city = db.Column(db.String(50))  
addr = db.Column(db.String(200)) pin = db.Column(db.String(10))def __init__(self, name,  
city, addr,pin): self.name = name self.city = city self.addr = addr self.pin = pin
```

Database Mentioned in URI

To create and use the database mentioned in URI, run the `create_all()` method.

`db.create_all()`The Session object of SQL Alchemy manages all persistence operations of ORM object.

2.3 POST MAN

Postman is a scalable API testing tool that quickly integrates into CI/CD pipeline. It started in 2012 as a side project by Abhinav Asthana to simplify API workflow in testing and development. API stands for Application Programming Interface which allows software applications to communicate with each other via API calls.

In output we decided to give this as a final output but then after further brainstorming we decided to create a web app which will make it easier and profession. It will be discussed later.

Then by using the postman testing tool we collaborated the python code and the UI which is designed then at last we gave the final output by using this tool.

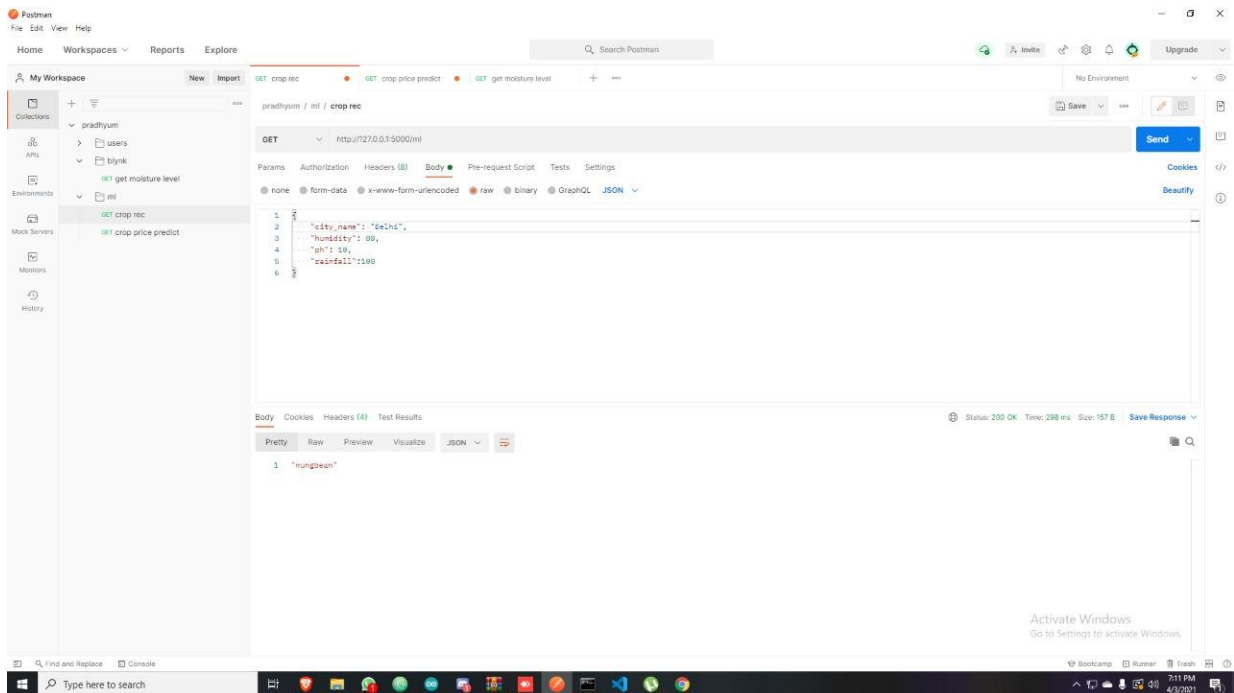


Fig: 2.6 Creating a Web Application Using Postman

2.4 SK LEARN

The SK learn is also called Scikit-learn. The SK learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-Neighbour's, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

So, with the help of the SK learn we united the programming language and the UI to work with the platform called post man.

CHAPTER 3

LITERATURE SURVEY

3.1 ANALYSIS OF THE LITERATURE

Literature survey is the main advance in programming improvement measure. Prior to building up the instrument it is important to decide these things are fulfilled, at that point the subsequent stage is to figure out which working framework and language can be utilized for building up the device. When the developers begin assembling the apparatus the software engineers need parcel of outer help. This help can be gotten from senior developers, from book or from sites. The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Prior to building the framework the above thought is considered for building up the proposed framework. The significant piece of the undertaking advancement area considers and completely survey all the necessary requirements for building up the venture. For each undertaking Literature survey is the main area in programming improvement measure. Prior to building up the instruments and the related planning it is important to decide and survey the time factor, asset prerequisite, labor, economy, and friend's strength. When these things are fulfilled and completely surveyed, at that point the following stage is to decide about the product details in the separate framework, for example, what kind of working framework the venture would require and what are largely the important programming are expected to continue with the subsequent stage like building up the apparatuses, and the related activities. Here we have taken the general surveys of different creators and noted down the fundamental central issues with respect to their work. In this venture literature survey assumes a prevailing part in get assets from different areas and all the connected points that are exceptionally valuable under this segment. The most awesome aspect if this is the manner in which things get all together and encourages us to suite our work according to the current information.

3.2 LITERARY REVIEWS

“L. Zhang, I. K. Dabipi, and W. L. Brown, “Internet of Things applications for agriculture,” in Internet of Things A to Z: Technologies and Applications

In this paper, we learn about the information protection and The history of agriculture starts at least 22,000 years ago when mankind learned to collect wild grains as food. Various crops have been cultivated as earlier as 9500 BC in Levant according to archaeological discoveries (Hillman, 1996; Walsh, 2009). Over tens of thousands of years since then, significant innovations have been made from time to time to increase the agricultural yield and reduce the heavy human labor needed. However, the demand for more foods from the increasing population will never get satisfied. It is predicted that the world’s population will reach 9.7 billion by 2050, which is about 33% more than today (un.org, 2015). Consequently, to keep pace with such population growth, the global production of food has to increase at least 70% to feed the world.

“P. V. Santhi, N. Kapileswar, V. K. R. Chenchela, and C. H. V. S. Prasad, “Sensor and vision based autonomous AGRIBOT for sowing seeds,’smart technologies

In this paper we can learn about how Autonomous agriculture robot is one of the promising solutions for precision agriculture. [...] Key Method Self-awareness of the robot's position is determined by the global and local maps generated from Global Positioning System (GPS) and on-board vision system paired with a personal computer. This paper also presents the proposed sensor-based precision seed metering and sowing mechanism. The proposed robot is a micro planter whose primary task would be to sow seeds at prefixed seeding intervals in the

“Design of web application using the machine learning function written by Nakul Lakhotia”

The author has built up a framework for the administration of wellbeing records, where patients and can create your own machine learning models like regression ,classification,

clustering etc. and deploy them on your web-app. The design of your application completely depends on your Web Development skills.

“A Code of Conduct on Agricultural Data Sharing Signing. Accessed in the machine learning language”

The increase of data produced in agriculture has allowed farmers and other partners in the agri-food chain to move from intuitive to a more data-based decision-making. Nowadays, digital farming makes the collection and exchange of data possible at an unprecedented level, creating value and business opportunities by applying data driven solutions along the agri-food chain

“J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, “A survey on Internet of things: Architecture, enabling technologies, security and privacy, and applications)”

To this end, this paper conducts a comprehensive overview of IoT with respect to system architecture, enabling technologies, security and privacy issues, and present the integration of fog/edge computing and IoT, and applications.

CHAPTER 4

SYSTEM DESIGN AND REQUIREMENTS

4.1 SYSTEM DESIGN

System design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organization. A systemic approach is required for a coherent and well-running system. Bottom-Up or Top-Down approach is required to take into account all related variables of the system. A designer uses the modelling languages to express the information and knowledge in a structure of system that is defined by a consistent set of rules and definitions. Then we used different types of sensors to perform various operations so with the help of those sensors we can find the humidity value it means the soil density and with that further steps are processed. Then by find the value in the humidity sensor we process the value to the app we created at first while doing this app we faced lots of issue and we used machine learning to make the app then we found to do that we need algorithm so we used random forest algorithm so after rectifying with lots of functions we found python program and postman web app development to do our application.

4.1.1 Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data. Then we used SQL Alchemy as the extension by using flash function we did the application in the web with the help of the postman web app development the design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things

4.1.2 Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it first starts from the humidity sensor this humidity sensor will measure the soil fertility from the soil and sends the output to the Arduino board this Arduino board will send the value to the web app we created then the output is shown in the application we created

The output form of an information system should accomplish one or more of the following objectives.

- Humidity sensor
- Node MCU
- Arduino board.

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts. .

4.2 SYSTEM REQUIREMENTS

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently. Failure to meet these requirements can result in wrong values from the sensor so that the whole process will be collapsed. The former may prevent a device or application from getting installed, whereas the latter may cause a product to get wrong values from the sensors or may lead to app crash. The after finding the correct value from the sensors that will feed the value to the web application that we created then with the requirements we created the process will run smoothly so that the farmers can get the nice idea and can sell the product in nice rate.

4.2.1 Hardware Requirements:

- System - Windows 7/10
- Speed - 2.4GHZ
- Hard disk - 40GB
- Monitor - 15VGA color
- RAM - 4GB

4.2.2 Software Requirements:

- Operating System - Windows 10
- Coding language - Python
- IDE - NetBeans
- Backend - MySQL

4.3 REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analyzing, documenting, validating and managing software or system requirements. The requirements should be measurable, testable and traceable related to identified farmers needs or opportunities and define to a level of detail, sufficient for system design.

4.3.1 Functional Requirements

It is a technical specification requirement for the software products. It is the final step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

i. Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

ii. Robustness

It refers to a python program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

iii. Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

iv. Reliability

It is the probability of how often the software fails. The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even capable of working around any failure.

v. Compatibility

It is supported by version above all web browsers. Using any web servers like local host makes the system real-time experience.

vi. Flexibility

The flexibility of the project is provided in such a way that is has the ability to run on different environments with the help of the MySQL alchemy keys and postman app being executed by different users.

vii. Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information.

4.3.2 Non-Functional Requirements

Non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually architecturally significant requirements.

i. Portability

It is the usability of the sensors and python program it can run in different environments. The project can be run in any operating system.

ii. Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

iii. Accuracy

The result of the requesting query is very accurate and high speed of retrieving information.

iv. Maintainability

Project is simple as further updates can be easily done without affecting its stability. That includes the process of weather forecast with that we do daily update our app so that at each and every time if the weather changes it is updated automatically. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyze, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

CHAPTER 5

SOFTWARE DEVELOPMENT METHODOLOGY

5.1 ARCHITECTURAL DESIGN

The architecture of a system describes its major components, their relationships (structures), and how they interact with each other. Software architecture and design includes several contributory factors such as Business strategy, quality attributes, human dynamics, design, and IT environment. Architecture serves as a blueprint for a system. It provides an abstraction to manage the system complexity and establish a communication and coordination mechanism among components.

- It defines a structured solution to meet all the technical and operational requirements, while optimizing the common need of the farmers
- Further, it involves a set of significant decisions about the application related to software development and each of these decisions can have a considerable impact on quality, maintainability, performance, and the overall success of the final product.

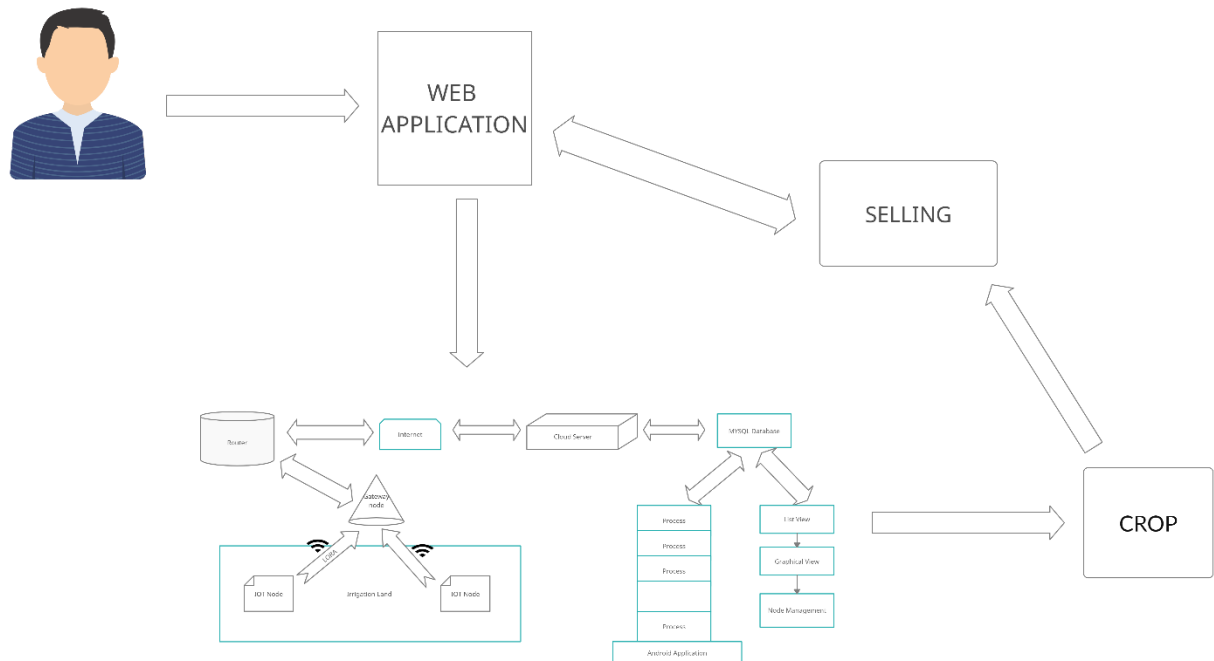


Fig: 5.1 Architecture Diagram

5.2 MODULE DESCRIPTION

Modules used: -

- Web Application Creation
- Soil sensing Module
- Crop Marketing Module
- Result Module

i. Web Application Creation:

Web application module includes wrapping up of all the IOT devices output and the machine learning algorithm working into the single interface. It also includes creation of login credentials, right from creating a small database, encoding and decoding the password till the end application.

ii. Soil Sensing Module:

The soil sensing module includes the feature called measuring the soil content from the soil so with this module we can identify the moisture level in soil. So that the person using this technology can easily identify the moisture level in soil

iii. Crop Marketing Module:

This module provides facility to sell the crop in expected price so with the help of this module we can easily predict the price of the crop and without others help the farmers can sell the crop after harvesting directly to the customer so that there won't be any help from third party

iv. Result Module:

The results module will be providing the end output of the smart agriculture. It includes all the output of the IOT sensors such as humidity sensors, temperature sensor. It also includes the classified crops according to their conditions. In the end it integrates everything into a single interface to provide a hassle-free experience

Other Modules used:

i. Mobile Module:

This module includes a minor phone app that is already available and is integrated with the Arduino board of IOT to give the user reminder alerts to their mobiles for watering the plants in order for making the process easier

ii. Login Module:

In this, user can login to the website by registered login id and a valid password. Only the authenticated user can login and use the website.

iii. Frequent Database module:

We introduce a Pincer-search algorithm for efficient mining of the frequent item sets. The pre-filtering method is applied to the input dataset to remove the item having low variance.

5.3 MAKING UP OF THE SMART AGRICULTURE SYSTEM FRAMEWORK

The progress in the agriculture domain is linked to the recent technological advances. Modern agriculture systems integrate state-of-the-art technological solutions. MYSQLAlchemy is the key concept for python in the agriculture domain. So, after the data further steps are provoked related to the functions added.

5.3.1 Machine learning for prediction

Activity diagrams describe the workflow behavior of a system. Activity diagrams are similar to state diagrams because activities are the state of doing something. The diagrams describe the state of activities by showing the sequence of activities performed. Activity diagrams can show activities that are conditional or parallel. And the images given below shows the project work of predicting the crop price

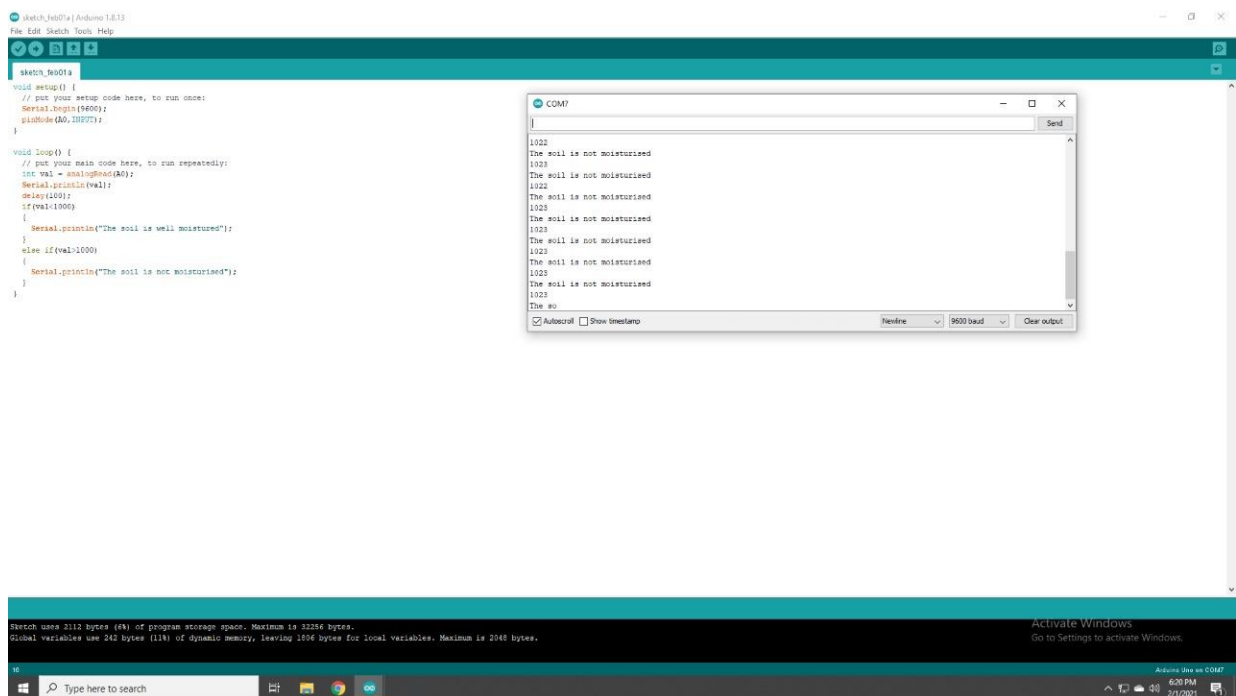


Fig: 5.2 Program created to predict the crop

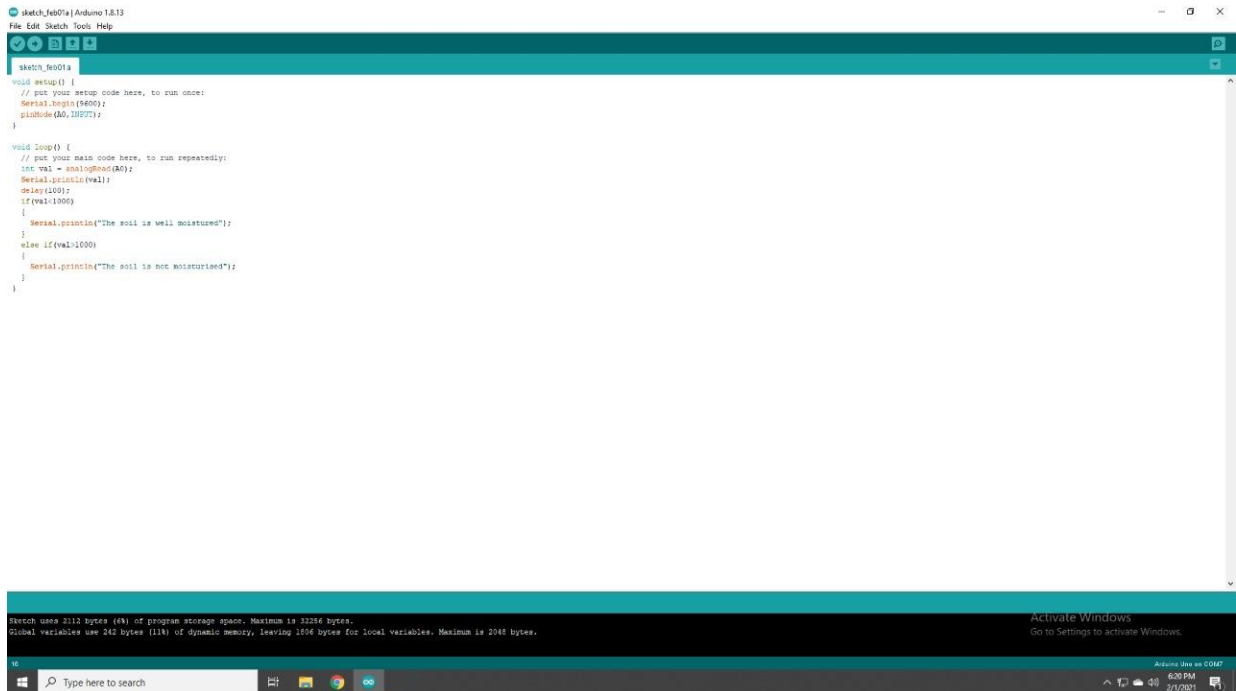


Fig: 5.3 After predicting the crop

5.3.2 Web app creation

The web app creation is a important process in this creation so to create a web app we need keys to perform certain opertions so that by using the python programming language we created keys and with machine learning function we used different algorithms to perform the operations then after creating the keys with the help of the flash function we created the extension file so with that extension file we started digging deep to run the keys then we found the brainstorming idea to run the application it is the postman web application with the help of the postman web service we created the web application

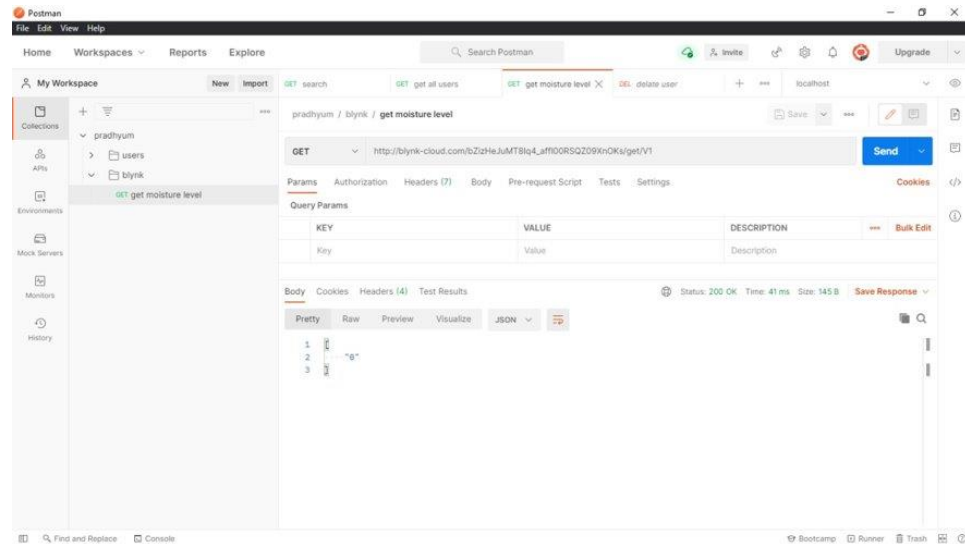


Fig: 5.4 Use case web application screenshot

5.3.3 IOT framework

Class diagrams are widely used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as different sensors. So first the process starts with humidity sensor the humidity sensor maintains the soil fertility of the soil. So by finding the soil value now the farmers can know what crop can be planted for this land. Then by finding the value the humidity sensor send the value to the arduino board then the arduino board send the value to the web application we created with that value our app shows the moisture level and the crop that we can grow and the price of crop so that the farmers can notify everything clearly



Fig: 5.5 sensor screenshot

5.4 Crop prediction

To predict the crop for the land we used different types of algorithm at first with the help of machine learning. Machine learning is an important decision support tool for crop yield prediction, including supporting decisions on what crops to grow and what to do during the growing season of the crops. Several machine learning algorithms have been applied to support crop yield prediction research. Then we found that deep forest algorithm is one of the best algorithms that we ever used with that algorithm we have created the machine learning algorithm to predict crop.

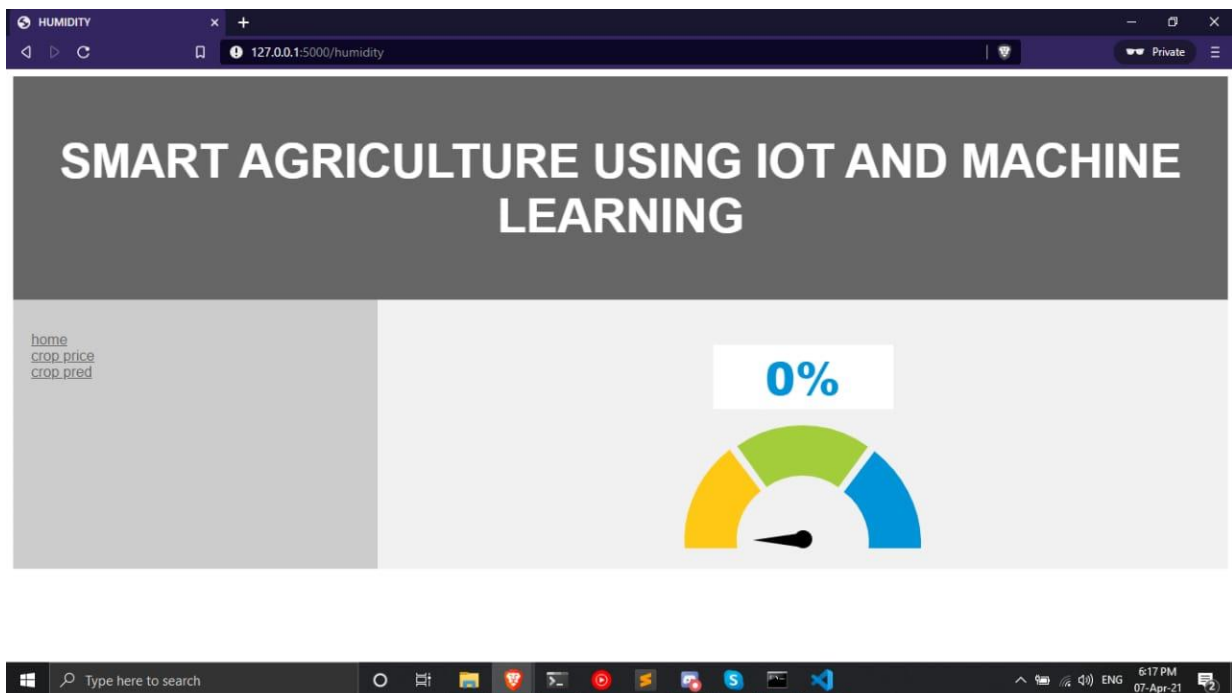


Fig: 5.6 Crop prediction screenshot

5.5 Crop Price Prediction

The crop price prediction is one the most important function that we ever created so to perform that function at first, we used different types of algorithm then we found the best algorithm to predict the crop price is that random forest algorithm. To predict the Price of the Crop we use Naïve Bayes Algorithm which is a Machine Learning Classification technique. The algorithms run in the background and give the Output to the Department

Heads which is in turn conveyed to the Farmers for better pre- knowledge about the outcome

CHAPTER 6

RESULTS AND DISCUSSIONS

As we have been decided that this project will be implemented as an app that includes right from creating credentials for the user and also sensing the soil and right till marketing the crop with machine learning. We are planning to use deep learning algorithm for implementing this approach in this implementation we will be using android studio for app creation, Arduino software for IOT modules and platform for training the machine using deep learning. By using this methodology, we can improvise the yield of the crop and marketing for these crops are made much easier.as the result we will be getting the yielded crops directly from the farmers so that the farmers can get the price of the crop as they expected. And with our app everyone can post their crop that they have so that if any companies needed the crop to do any product, they can buy the yielded crop directly from the farmers so that the farmers and the company both of them can be benefited. The company can buy the crop in low rate and the farmers can sell the crop in the price the needed

CHAPTER 7

CONCLUSION AND FUTURE WORK

In The focus on smarter, better, and more efficient crop growing methodologies is required in order to meet the growing food demand of the increasing world population in the face of the ever-shrinking arable land. The ultimate conclusion is that, with these introductions of technology (IOT) and (AI) we can make sure that there is a very good future for agriculture and to encourage youngsters and also many people to do agriculture in interesting and efficient way. The focus on smarter, better, and more efficient crop growing methodologies is required in order to meet the growing food demand of the increasing world population in the face of the ever-shrinking arable land. The development of new methods of improving crop yield and handling, one can readily see currently: technology-weaned, innovative younger people adopting farming as a profession, agriculture as a means for independence from fossil fuels, tracking the crop growth, safety and nutrition labeling, partnerships between growers, suppliers, and retailers and buyers. This paper considered all these aspects and highlighted the role of various technologies, especially IoT, in order to make the agriculture smarter and more efficient to meet future expectations. For this purpose, wireless sensors, UAVs, Cloud-computing, communication technologies are discussed thoroughly. Furthermore, a deeper insight on recent research efforts is provided. In addition, various IOT-based architectures and platforms are provided with respect to agriculture applications. A summary of current challenges facing the industry and future expectations are listed to provide guidance to researchers and engineers.

REFERENCES

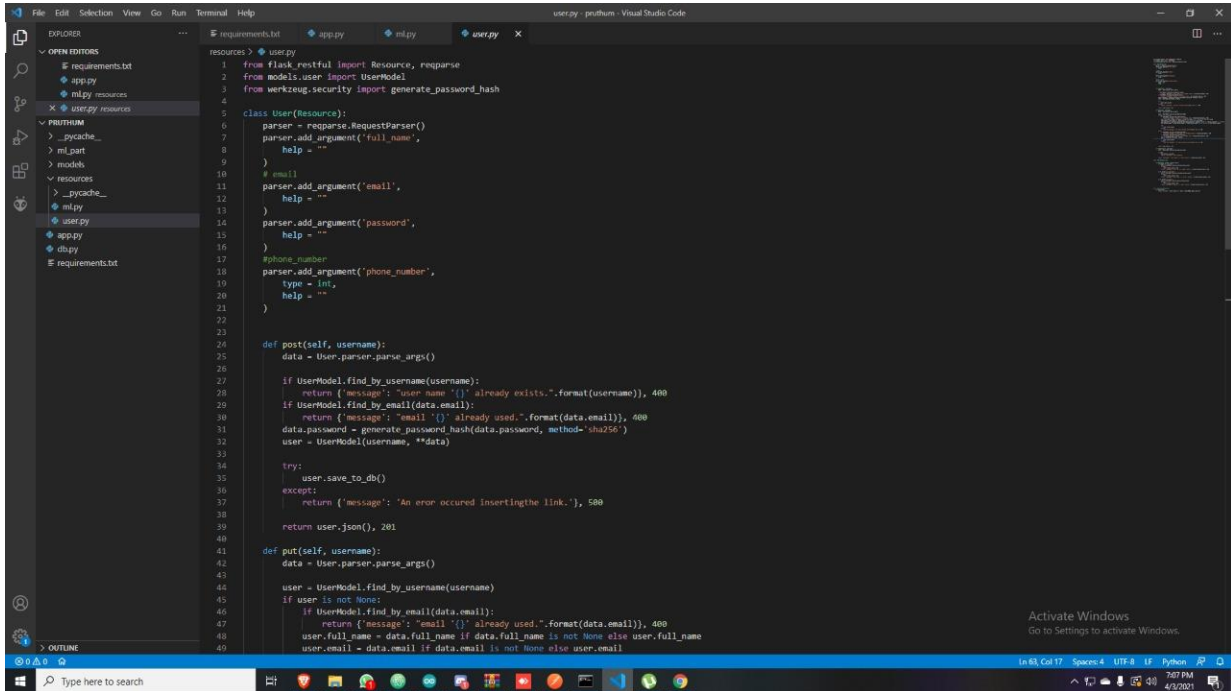
- [1] Code of Conduct on Agricultural Data Sharing Signing. Accessed: Apr. 13, 2019. [Online]. Available: <https://www.ecpa.eu/news/code-conduct-agricultural-data-sharing-signing>
- [2] Khanna and S. Kaur, "Evolution of Internet of Things (IoT) and its significant impact in the field of precision agriculture," *Compute. Electron. Agri cult.*, vol. 157, pp. 218–231, Feb. 2019
- [3] A. Tzounis, N. Katsoulas, T. Bartzanas, and C. Kittas, "Internet of Things in agriculture, recent advances and future challenges," *Biosyst. Eng.*, vol. 164, pp. 31–48, Dec. 2017.
- [4] P. V. Santhi, N. Kapileswar, V. K. R. Chenchela, and C. H. V. S. Prasad, "Sensor and vision based autonomous AGRIBOT for sowing seeds," in *Proc. Int. Conf. Energy, Commun., Data Anal. Soft Comput.*, Chennai, Tamil Nadu, Aug. 2017, pp. 242–245..
- [5] "Food Production Must Double by 2050 to Meet Demand From World's Growing Population. Accessed: Apr. 5, 2019 [Online]. Available: <https://www.un.org/press/en/2009/gae-f3242.doc.htm>
- [6] How to Feed the World in 2050 by FAO. Accessed: Sep. 6, 2019. [Online]. Available: <https://www.fao.org/wsfs/forum2050/wsfs-forum/en/>
- [7] A. D. Tripathi, R. Mishra, K. K. Maurya, R. B. Singh, and D. W. Wilson, "Estimates for world population and global food availability for global health," *The Role of Functional Food Security in Global Health*. 2019, pp. 3–24.
- [8] M. Elder and S. Hayashi, "A regional perspective on biofuels in Asia," in *Biofuels and Sustainability (Science for Sustainable Societies)*. Springer, 2018.
- [9] World Agriculture: Towards 2015/2030 by FAO. Accessed: Apr. 15, 2019. [Online]. Available: <https://www.fao.org/3/a-y4252e.pdf>
- [10] L. Zhang, I. K. Dabipi, and W. L. Brown, "Internet of Things applications for agriculture," in *Internet of Things A to Z: Technologies and Applications*, Q. Hassan, Ed., 2018.

- [11] S. Navulur and M. N. Giri Prasad, "Agricultural management through wireless sensors and Internet of Things," *Int. J. Elect. Comput. Eng.*, vol. 7, no. 6, pp. 3492–3499, 2017.
- [12] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.
- [13] M. Ayaz, M. Ammad-Uddin, I. Baig, and E.-H. M. Aggoune, "Wireless sensor's civil applications, prototypes, and future integration possibilities: A review," *IEEE Sensors J.*, vol. 18, no. 1, pp. 4–30, Jan. 2018.
- [14] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.
- [15] X. Hi, X. An, Q. Zhao, H. Liu, L. Xia, X. Sun, and Y. Guo, "State-of-the-art Internet of Things in protected agriculture," *Sensors*, vol. 19, no. 8, p. 1833, 2019.
- [16] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. N. Hindia, "An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3758–3773, Oct. 2018.
- [17] Code of Conduct on Agricultural Data Sharing Signing. Accessed: Apr. 13, 2019. [Online]. Available: <https://www.ecpa.eu/news/code-conduct-agricultural-data-sharing-signing>
- [18] Industry 4.0 in Agriculture: Focus on IoT Aspects. Accessed: Sep. 5, 2019. [Online]. Available: <https://ec.europa.eu/growth/tools-databases/dem/monitor/content/industry-40-agriculture-focus-iot-aspects>
- [19] K. Thea, C. Martin, M. Jeffrey, E. Gerhard, Z. Dimitrios, M. Edward, and P. Jeremy, "Food safety for food security: Relationship between global megatrends and developments in food safety," *Trends Food Sci. Technol.*, vol. 68, pp. 160–175, Oct. 2017.
- [20] How Blockchain and IoT Tech will Guarantee Food Safety. Accessed: Sep. 6, 2019. [Online]. Available: <https://www.dataversity.net/howblockchain-and-iot-tech-will-guarantee-food-safety/>

APPENDIX

A. SAMPLE CODE

USERS CODE



```
1 from flask_restful import Resource, reqparse
2 from models.user import UserModel
3 from werkzeug.security import generate_password_hash
4
5 class User(Resource):
6     parser = reqparse.RequestParser()
7     parser.add_argument('full_name',
8                         help = ""
9                     )
10    # email
11    parser.add_argument('email',
12                        help = ""
13                    )
14    parser.add_argument('password',
15                        help = ""
16                    )
17    #phone_number
18    parser.add_argument('phone_number',
19                        type = int,
20                        help = ""
21                    )
22
23    def post(self, username):
24        data = User.parser.parse_args()
25
26        if UserModel.find_by_username(username):
27            return {'message': 'User name "{}" already exists.'.format(username)}, 400
28        if UserModel.find_by_email(data.email):
29            return {'message': 'email "{}" already used.'.format(data.email)}, 400
30        data.password = generate_password_hash(data.password, method='sha256')
31        user = UserModel(username, **data)
32
33        try:
34            user.save_to_db()
35        except:
36            return {'message': 'An error occurred inserting the link.'}, 500
37
38        return user.json(), 201
39
40    def put(self, username):
41        data = User.parser.parse_args()
42
43        user = UserModel.find_by_username(username)
44        if user is not None:
45            if UserModel.find_by_email(data.email):
46                return {'message': 'email "{}" already used.'.format(data.email)}, 400
47            user.full_name = data.full_name if data.full_name is not None else user.full_name
48            user.email = data.email if data.email is not None else user.email
```

Fig: User code

The screenshot shows a Visual Studio Code editor with a Python file named `user.py`. The code defines a `user.py` module with a `user` class and a `UserBy(Resource)` class. The `user` class has methods for creating a new user, deleting a user, and finding a user by ID, username, or email. The `UserBy` class is a Flask-RESTful resource that implements the `get` method to return user data as JSON. The code is as follows:

```
48 user.full_name = data.full_name if data.full_name is not None else user.full_name
49 user.email = data.email if data.email is not None else user.email
50 user.password = generate_password_hash(data.password, method='sha256') if data.password is not None else user.password
51 user.phone_number = data.phone_number if data.phone_number is not None else user.phone_number
52
53 try:
54     user.save_to_db()
55 except:
56     return {'message': 'An error occurred inserting the link.'}, 500
57
58 else:
59     if UserModel.find_by_username(username):
60         return {'message': 'user name {} already exists.'.format(username)}, 400
61     if UserModel.find_by_email(email):
62         return {'message': 'email {} already used.'.format(email)}, 400
63     user = UserModel(username, **data)
64     try:
65         user.save_to_db()
66     except:
67         return {'message': 'An error occurred inserting the link.'}, 500
68
69 return user.json(), 201
70
71 def delete(self, username):
72     user = UserModel.find_by_username(username)
73
74     if user:
75         user.delete_from_db()
76         return {'message': 'user delete'}
77
78     return {'message': 'user name {} didn't exists.'.format(username)}, 400
79
80 class UserBy(Resource):
81
82     def get(self, column, column_value):
83         if column == "userid":
84             user = UserModel.find_by_userid(column_value)
85             if user:
86                 return user.json(), 201
87             return {'message': 'user ID {} didn't exists.'.format(column_value)}, 400
88
89         elif column == "username":
90             user = UserModel.find_by_username(column_value)
91             if user:
92                 return user.json(), 201
93             return {'message': 'user name {} didn't exists.'.format(column_value)}, 400
94
95         elif column == "email":
96             user = UserModel.find_by_email(column_value)
97             if user:
98                 return user.json(), 201
99             return {'message': 'email {} didn't exists.'.format(column_value)}, 400
100
101
102 class UserList(Resource):
103     def get(self):
104         return {'items': [user.json() for user in UserModel.query.all()]}
105
```

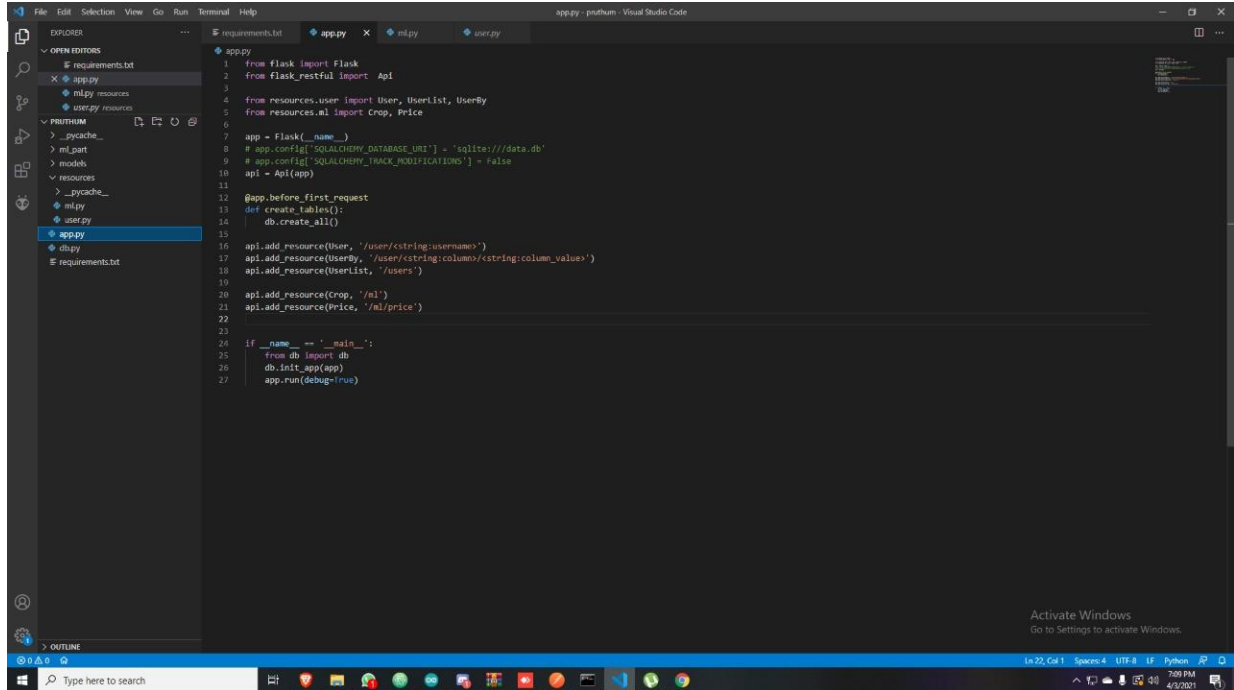
Fig: User code

The screenshot shows a Visual Studio Code editor with a Python file named `user.py`. The code defines a `user.py` module with a `user` class and a `UserBy(Resource)` class. The `user` class has methods for creating a new user, deleting a user, and finding a user by ID, username, or email. The `UserBy` class is a Flask-RESTful resource that implements the `get` method to return user data as JSON. The code is as follows:

```
72 user = UserModel.find_by_username(username)
73
74 if user:
75     user.delete_from_db()
76     return {'message': 'user delete'}
77
78 return {'message': 'user name {} didn't exists.'.format(username)}, 400
79
80 class UserBy(Resource):
81
82     def get(self, column, column_value):
83         if column == "userid":
84             user = UserModel.find_by_userid(column_value)
85             if user:
86                 return user.json(), 201
87             return {'message': 'user ID {} didn't exists.'.format(column_value)}, 400
88
89         elif column == "username":
90             user = UserModel.find_by_username(column_value)
91             if user:
92                 return user.json(), 201
93             return {'message': 'user name {} didn't exists.'.format(column_value)}, 400
94
95         elif column == "email":
96             user = UserModel.find_by_email(column_value)
97             if user:
98                 return user.json(), 201
99             return {'message': 'email {} didn't exists.'.format(column_value)}, 400
100
101
102 class UserList(Resource):
103     def get(self):
104         return {'items': [user.json() for user in UserModel.query.all()]}
105
```

Fig: User code

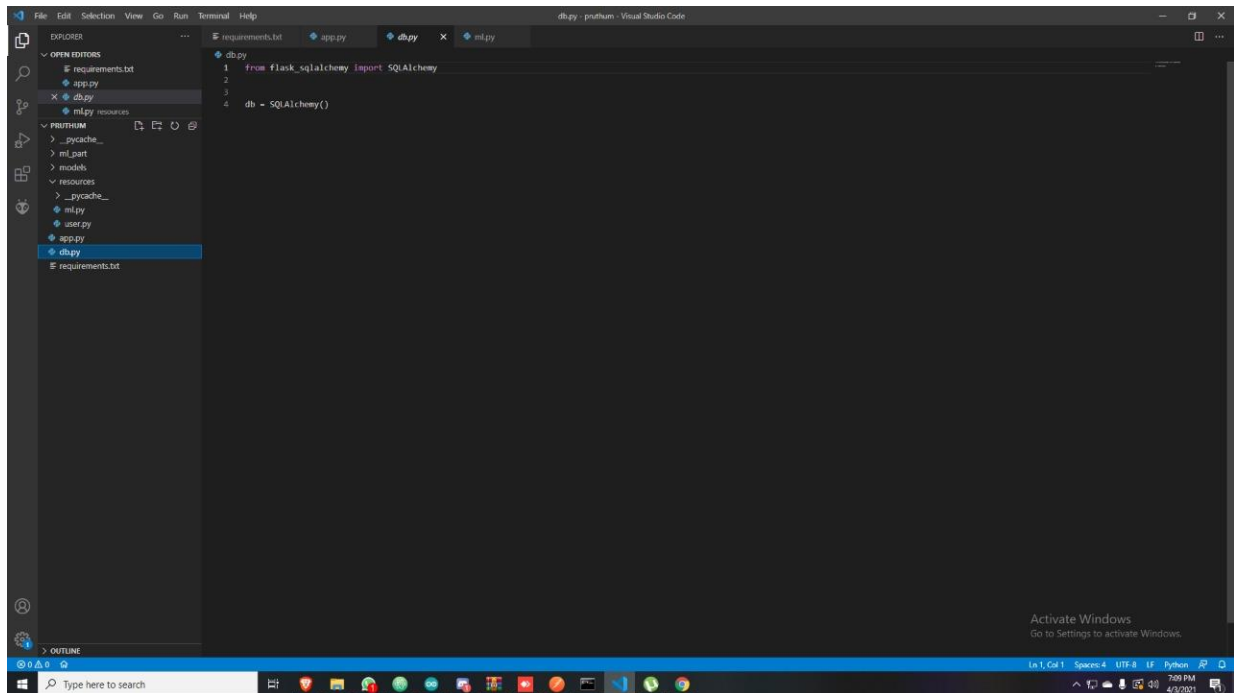
APP CODE



```
1 from flask import Flask
2 from flask_restful import Api
3
4 from resources.user import User, Userlist, UserBy
5 from resources.ml import Crop, Price
6
7 app = Flask(__name__)
8 # app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///data.db'
9 # app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
10 api = Api(app)
11
12 @app.before_first_request
13 def create_tables():
14     db.create_all()
15
16 api.add_resource(User, '/user/<string:username>')
17 api.add_resource(UserBy, '/user/<string:column>/<string:column_value>')
18 api.add_resource(Userlist, '/users')
19
20 api.add_resource(Crop, '/ml')
21 api.add_resource(Price, '/ml/price')
22
23
24
25 if __name__ == '__main__':
26     from db import db
27     db.init_app(app)
28     app.run(debug=True)
```

Fig: App Code

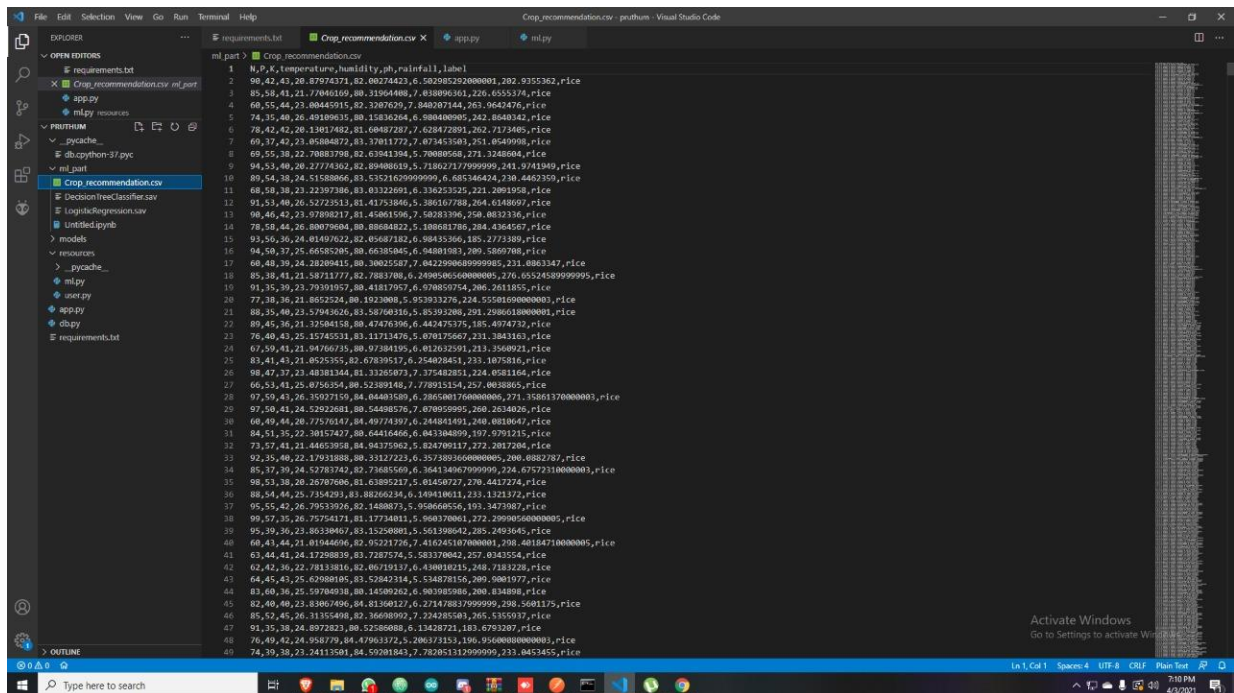
DATABASE CODE



```
1 from flask_sqlalchemy import SQLAlchemy
2
3 db = SQLAlchemy()
```

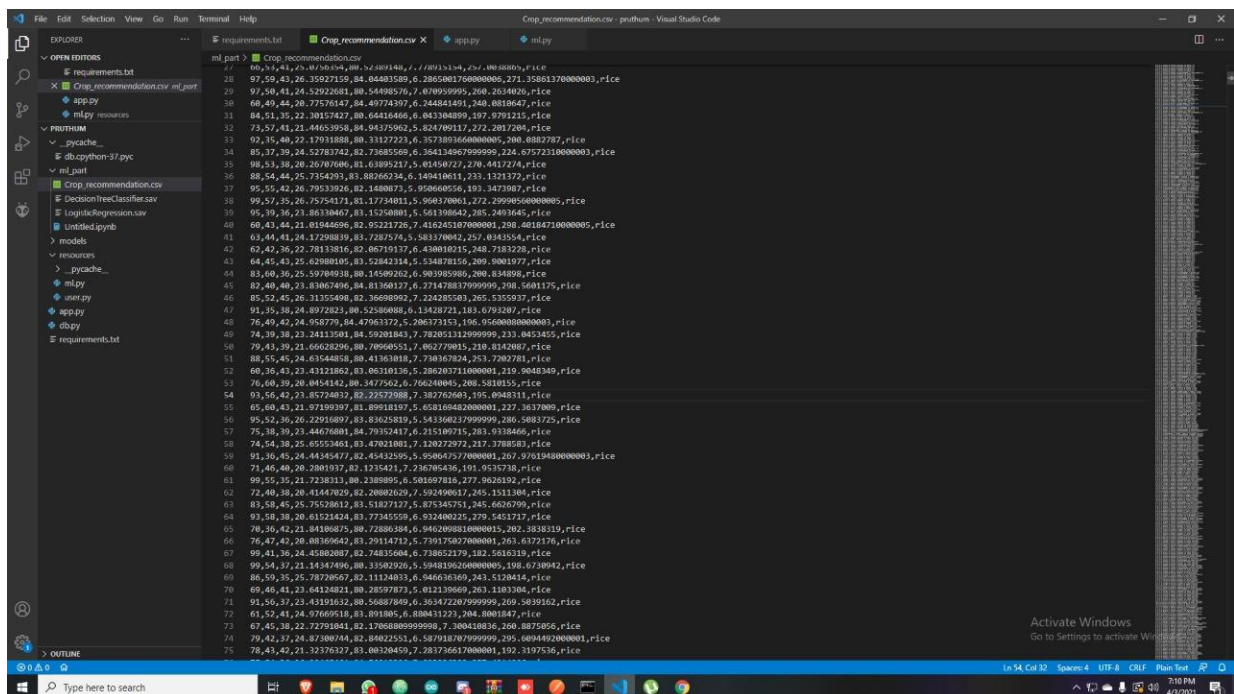
Fig: Database Code

CROP RECOMMENDATION



```
1  N,P,K,temperature,humidity,pH,rainfall,label
2  98.42,43.20,87974371,82.00274423,6.50298520000001,202.935362,rice
3  85.59,41.21,77061109,80.20564067,6.810096301,228.655334,rice
4  66.55,44.23,60445915,82.3207629,7.040207144,263.0642476,rice
5  74.35,40.26,49189635,80.15836264,6.980400905,242.8040342,rice
6  78.42,40.26,13617482,81.60487287,7.628472891,262.7173405,rice
7  69.37,42.23,85680472,83.7081772,7.07145350,251.0549996,rice
8  66.55,38.22,70887398,82.43943305,5.7000508,271.3248004,rice
9  94.53,40.26,27774362,82.89408039,5.71802717999999,241.9741949,rice
10  89.54,38.24,51588066,83.53521629999999,6.685346424,230.4462359,rice
11  68.58,38.23,22397386,83.03322691,6.33625325,221.2091958,rice
12  91.53,40.26,52722933,81.41793845,5.306107708,264.0348607,rice
13  90.46,42.23,9780217,81.45061596,7.50283396,258.0832336,rice
14  78.58,44.26,80079604,80.88684822,5.108681786,284.4364567,rice
15  93.50,36.24,01697622,82.05487182,6.80433366,185.2773389,rice
16  94.50,37.25,6658206,80.6636465,6.94091903,200.586700, rice
17  60.48,35.24,2820415,80.30025587,7.0422990689999995,221.0863347,rice
18  85.38,41.21,58711777,82.7883708,6.2409506500000005,276.6552489999995,rice
19  91.35,39.23,79391957,80.41817057,6.97085974,206.7411855,rice
20  77.38,36.21,8852252,80.1023008,6.93932326,224.25501690000003,rice
21  88.35,40.23,57943626,83.58760310,5.85393280,291.290610000001,rice
22  89.45,36.21,32504158,80.47476396,6.442475375,185.4974732,rice
23  76.40,43.25,15745531,83.13713470,5.070175667,231.3843103,rice
24  67.59,41.21,84766735,80.9738109,6.401632991,211.3506021,rice
25  81.41,41.21,8525351,82.67839517,6.254028451,233.1075816,rice
26  98.47,37.23,48381344,81.33265073,7.375482851,224.0581164,rice
27  66.53,41.25,8750354,80.52389148,7.778915134,267.0038805,rice
28  97.59,43.26,70972159,84.6404509,6.2050817000000003,227.32681370000003,rice
29  57.50,41.24,52822681,80.54488576,7.07095995,260.2034026,rice
30  60.49,44.26,77576147,84.49774397,6.244841491,240.8810647,rice
31  84.51,35.22,38157427,80.64416466,6.843384899,197.5791215,rice
32  73.57,41.21,44659938,84.98259825,6.824709117,277.2037206,rice
33  92.35,40.22,17031888,80.33127223,6.3573936000000005,206.8802787,rice
34  85.37,39.24,52783742,82.73685569,6.364134967999999,224.6752310000003,rice
35  98.53,38.26,26787606,81.63895217,5.61450727,270.4417274,rice
36  80.54,44.25,7354291,81.88260234,6.10418011,233.1323372,rice
37  95.55,42.26,79533926,82.1480871,5.40604056,193.3473987,rice
38  99.57,35.26,75754171,81.17734011,5.968370061,272.29905600000005,rice
39  95.39,36.23,86330467,83.15250801,5.561398642,285.2493645,rice
40  68.44,44.21,01946896,82.9221726,7.416245107000001,208.40184710000005,rice
41  63.44,41.24,17208039,83.7207274,5.583370642,257.043554,rice
42  62.42,36.22,78138116,82.06719137,6.430010215,248.7183228,rice
43  64.45,43.25,62980105,83.52842314,5.534878156,209.9001977,rice
44  83.60,36.25,59784938,80.14580262,6.90395980,206.834896,rice
45  82.40,40.27,83067406,84.11360127,6.274747827999999,208.5601175,rice
46  85.52,45.26,31355408,82.36698992,7.224285503,265.5355937,rice
47  91.35,38.24,8972823,80.52580088,6.13428721,183.6793207,rice
48  76.49,42.24,958779,84.47063372,5.206373153,196.95600000000003,rice
49  74.39,38.23,24123501,84.59280843,7.782051312999999,233.0434575,rice
50  74.39,38.23,24123501,84.59280843,7.782051312999999,233.0434575,rice
```

Fig: Crop Recommendation Dataset



```
1  N,P,K,temperature,humidity,pH,rainfall,label
2  98.42,43.20,87974371,82.00274423,6.50298520000001,202.935362,rice
3  85.59,41.21,77061109,80.20564067,6.810096301,228.655334,rice
4  66.55,44.23,60445915,82.3207629,7.040207144,263.0642476,rice
5  74.35,40.26,49189635,80.15836264,6.980400905,242.8040342,rice
6  78.42,40.26,13617482,81.60487287,7.628472891,262.7173405,rice
7  69.37,42.23,85680472,83.7081772,7.07145350,251.0549996,rice
8  66.55,38.22,70887398,82.43943305,5.7000508,271.3248004,rice
9  94.53,40.26,27774362,82.89408039,5.71802717999999,241.9741949,rice
10  89.54,38.24,51588066,83.53521629999999,6.685346424,230.4462359,rice
11  68.58,38.23,22397386,83.03322691,6.33625325,221.2091958,rice
12  91.53,40.26,52722933,81.41793845,5.306107708,264.0348607,rice
13  90.46,42.23,9780217,81.45061596,7.50283396,258.0832336,rice
14  78.58,44.26,80079604,80.88684822,5.108681786,284.4364567,rice
15  93.50,36.24,01697622,82.05487182,6.80433366,185.2773389,rice
16  94.50,37.25,6658206,80.6636465,6.94091903,200.586700, rice
17  60.48,35.24,2820415,80.30025587,7.0422990689999995,221.0863347,rice
18  85.38,41.21,58711777,82.7883708,6.2409506500000005,276.6552489999995,rice
19  91.35,39.23,79391957,80.41817057,6.97085974,206.7411855,rice
20  77.38,36.21,8852252,80.1023008,6.93932326,224.25501690000003,rice
21  88.35,40.23,57943626,83.58760310,5.85393280,291.290610000001,rice
22  89.45,36.21,32504158,80.47476396,6.442475375,185.4974732,rice
23  76.40,43.25,15745531,83.13713470,5.070175667,231.3843103,rice
24  67.59,41.21,84766735,80.9738109,6.401632991,211.3506021,rice
25  81.41,41.21,8525351,82.67839517,6.254028451,233.1075816,rice
26  98.47,37.23,48381344,81.33265073,7.375482851,224.0581164,rice
27  66.53,41.25,8750354,80.52389148,7.778915134,267.0038805,rice
28  97.59,43.26,70972159,84.6404509,6.2050817000000003,227.32681370000003,rice
29  57.50,41.24,52822681,80.54488576,7.07095995,260.2034026,rice
30  60.49,44.26,77576147,84.49774397,6.244841491,240.8810647,rice
31  84.51,35.22,38157427,80.64416466,6.843384899,197.5791215,rice
32  73.57,41.21,44659938,84.98259825,6.824709117,277.2037206,rice
33  92.35,40.22,17031888,80.33127223,6.3573936000000005,206.8802787,rice
34  85.37,39.24,52783742,82.73685569,6.364134967999999,224.6752310000003,rice
35  98.53,38.26,26787606,81.63895217,5.61450727,270.4417274,rice
36  80.54,44.25,7354291,81.88260234,6.10418011,233.1323372,rice
37  95.55,42.26,79533926,82.1480871,5.40604056,193.3473987,rice
38  99.57,35.26,75754171,81.17734011,5.968370061,272.29905600000005,rice
39  95.39,36.23,86330467,83.15250801,5.561398642,285.2493645,rice
40  68.44,44.21,01946896,82.9221726,7.416245107000001,208.40184710000005,rice
41  63.44,41.24,17208039,83.7207274,5.583370642,257.043554,rice
42  62.42,36.22,78138116,82.06719137,6.430010215,248.7183228,rice
43  64.45,43.25,62980105,83.52842314,5.534878156,209.9001977,rice
44  83.60,36.25,59784938,80.14580262,6.90395980,206.834896,rice
45  82.40,40.27,83067406,84.11360127,6.274747827999999,208.5601175,rice
46  85.52,45.26,31355408,82.36698992,7.224285503,265.5355937,rice
47  91.35,38.24,8972823,80.52580088,6.13428721,183.6793207,rice
48  76.49,42.24,958779,84.47063372,5.206373153,196.95600000000003,rice
49  74.39,38.23,24123501,84.59280843,7.782051312999999,233.0434575,rice
50  74.39,38.23,24123501,84.59280843,7.782051312999999,233.0434575,rice
```

Fig: Crop Recommendation Dataset

B. SCREEN SHOTS

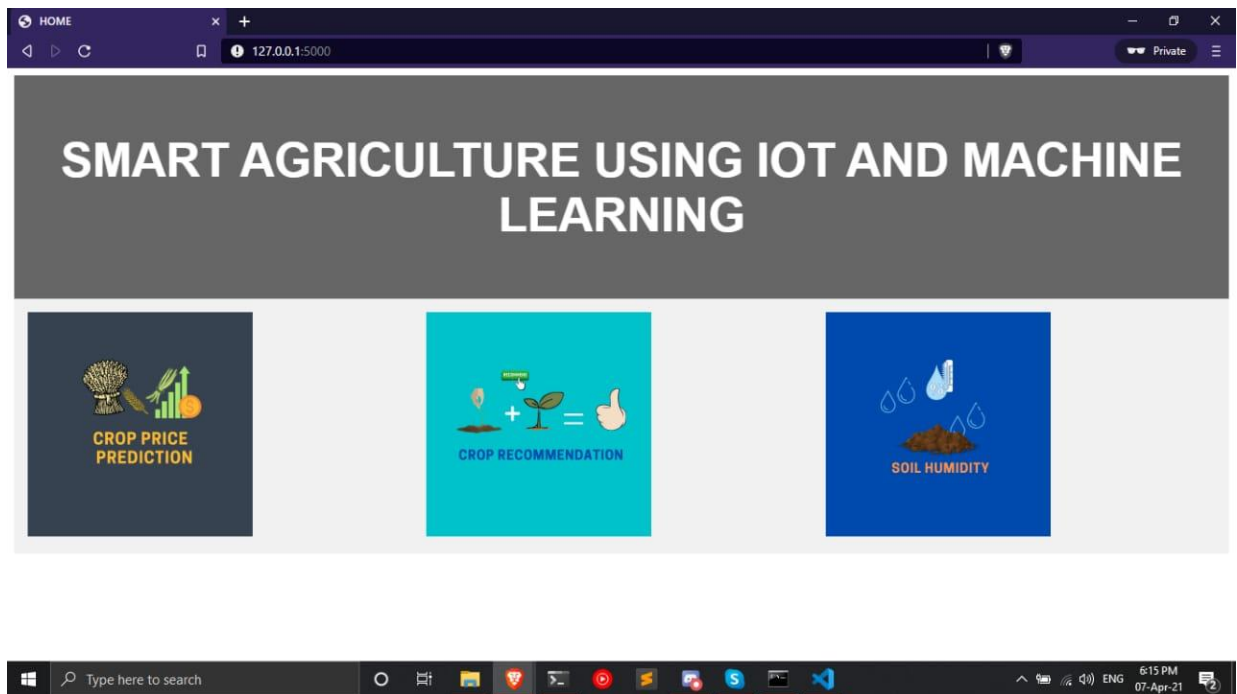


Fig: Web Application Interface

CROP PRICE

127.0.0.1:5000/crop_price

SMART AGRICULTURE USING IOT AND MACHINE LEARNING

HOME
CROP PRICE
CROP PREDICTION

Choose a crop Barley

Choose a year 2021

Choose a month 4

Rainfall value 200 predict

Fig: Crop Price Prediction Form

WPI value

127.0.0.1:5000/price

SMART AGRICULTURE USING IOT AND MACHINE LEARNING

home
crop price
crop pred

Wholesale price index : 117.1

Fig: Predicted Result of a Crop's Price

SMART AGRICULTURE USING IOT AND MACHINE LEARNING

SMART AGRICULTURE USING IOT AND MACHINE LEARNING
CROP PRICE
CROP PREDICTION

City Name

Humidity

PH

RainFall

127.0.0.1:5000/crop_pred

Fig: Crop Prediction Form

SMART AGRICULTURE USING IOT AND MACHINE LEARNING

home
crop price
crop pred

Crop name : rice

127.0.0.1:5000/crop

Fig: Predicted Result of the Appropriate Crop

C. PUBLICATION WITH PLAGIARISM REPORT