# WATER FOOTPRINT CALCULATOR

SUBMITTED IN PARTIAL FULFILLMENT FOR THE REQUIREMENT OF THE AWARD OF DEGREE OF

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE**

Submitted by
Shikha Kushwaha (2200290129017)
Shivanshu Srivastava (2100290400061)
Sumit Tiwari (2100290120168)
Suraj Jain (2100290120170)

Supervised by
**Mr. Vivek Kumar Sharma**
Assistant Professor
CS Department

**Department of Computer Science**
**KIET GROUP OF INSTITUTIONS, GHAZIABAD**
**(Affiliated to Dr. A. P. J. Abdul Kalam Technical University, Lucknow, U.P., India)**

**2024-2025**

# DECLARATION

We hereby declare that this submission is our work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person or material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.

Signature:
Name: Shivanshu Srivastava
Roll no.: 2100290400061

Signature:
Name: Shikha Kushwaha
Roll No.: 2200290129017

Signature:
Name: Sumit Tiwari
Roll No.: 2100290120168

Signature:
Name: Suraj Jain
Roll No.: 2100290120170

Date:-

# CERTIFICATE

This is to certify that Project Report entitled “**Water Footprint Calculator**” which is submitted by **Shivanshu Srivastava, Shikha Kushwaha, Sumit Tiwari and Suraj Jain** in partial fulfilment of the requirement for the award of degree B.Tech. in Department of Computer Science of Dr A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

**Date-**

**Supervisor**
Mr.Vivek Kumar Sharma
Assistant Professor
CS Department

# ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the synopsis of the B.Tech. Major Project undertaken during B.Tech. Third Year. We owe a special debt of gratitude to Prof. Vivek Kumar Sharma, Department of Computer Science, KIET Group of Institutions, Delhi-NCR, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen the light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Ajay Kumar Srivastav sir, Head of the Department of Computer Science, KIET Group of Institutions, Delhi-NCR, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution to the completion of the project.

Signature:

Name: Shivanshu Srivastava

Roll No.: 2100290400061

Signature:

Name: Shikha Kushwaha

Roll No.: 2200290129017

Signature:

Name: Sumit Tiwari

Roll No.: 2100290120168

Signature:

Name: Suraj Jain

Roll No.: 2100290120170

# ABSTRACT

The water footprint measures the amount of water used to produce each of the goods and services we use. The water footprint helps us understand for what purposes our limited freshwater resources are being consumed and polluted. The impact of it depends on where the water is taken from and when if it comes from a place where water is already scarce, the consequences can be significant and require action. The increase in the amount of non-available water due to pollution and scarce groundwater level has added more water footprints, at the community as well as at the personal levels. An increased water footprint directly affects the health and future of the citizens.

Preventing severe drought in water-stressed areas is only going to be possible if water is used with more care and efficiency, this can be done if we have readily available data on water footprints. Hence by using digital technologies like AI, Big Data, Blockchain, etc, and computer languages, a user-friendly app or website may be developed which can provide the water footprints of different items/ final products we use in daily life by feeding little inputs or just by scanning through the camera like Google lens. The app should support local languages, this will ensure pan-India usage and sensitize the people about the water footprints of items they use in daily life.

# Alignment with UN Sustainable Development Goals (SDGs)

## SDG 6: Clean Water and Sanitation

The project implements actions directly in line with Target 6.4 that focuses on boosting water-use efficiency together with secure freshwater extraction.

Water measurement enables us to find excessive use then develop conservation plans.

## SDG 12: Responsible Consumption and Production

The program helps achieve Target 12.2 whose focus is sustainable management and efficient resource utilization.

The initiative promotes industries and persons to follow responsible water consumption behaviors.

## SDG 13: Climate Action

Anyone trying to achieve Target 13.1 will find assistance through the project as it helps enhance climate resilience against water-related stress and droughts.

The project delivers climate change data needed to support policy development for addressing water scarcity caused by climate change.

## SDG 9: Industry, Innovation, and Infrastructure

The requirement for examining water footprints drives technological innovating efforts which result in sustainable industrial methods.

## SDG 15: Life on Land

The sustainable implementation of water resources supports freshwater ecosystem protection according to Goal 15.1 through responsible water management and decreased pollution.

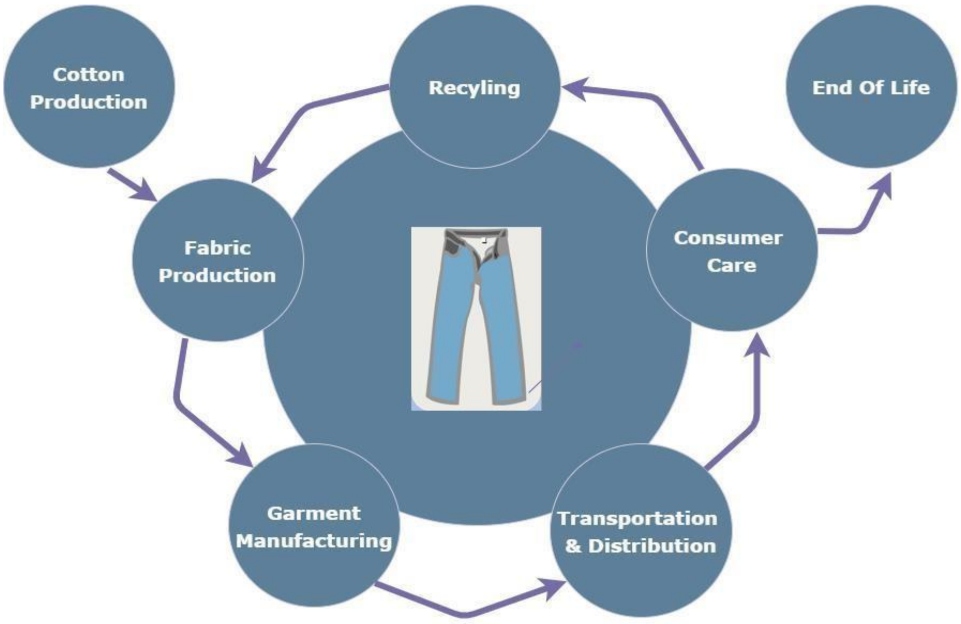
# TABLE OF CONTENTS Page No.

# CHAPTER-1

## 1. INTRODUCTION

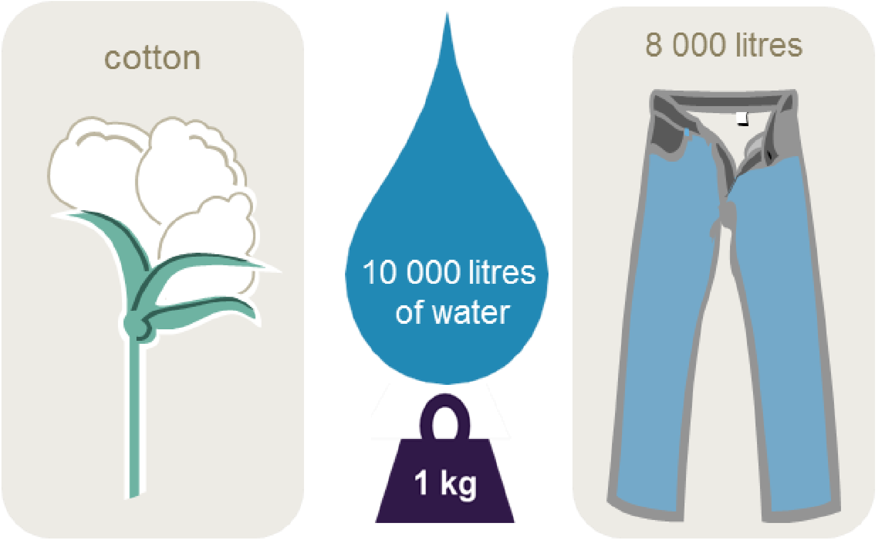
Comprehending the management of water footprints stands as a vital problem statement because these footprints measure the water used for product and service development. The increase of water scarcity as well as pollution requires understanding these footprints for sustainable water management to become effective. Water-stressed areas experience exceptional consequences from this situation which necessitates an immediate solution.

Acceptable insights regarding water footprints become accessible through a user- friendly application or website via digital tools including AI, Big Data, and Blockchain technologies. Users gain power through this solution to understand their choices better while learning sustainable H2O usage habits. The intended solution works to minimize adverse H2O footprint growth while promoting wise freshwater management through awareness programs and proactive actions to achieve sustainability for future generations.



Total consumption of water =
WCP+WFP+WGM+WTD+WCC+

# EXAMPLE OF WATER FOOTPRINT:



## 1.1 OBJECTIVES:

1. **Increase Awareness:** An educational program should teach users about water usage during production along with helping them comprehend their water-related habits and how these affect the environment.

2. **Empower Informed Choices:** Users can make knowledgeable choices by receiving full transparency about product water footprints which lets them select better sustainable alternatives.

3. **Promote Sustainable Behavior:** Users should practice sustainability by understanding how decreasing their water use and pollution actively contributes to water conservation goals.

4. **Facilitate Data Accessibility:** H2O footprint data should be accessible through digital technologies to users for increased transparency of their consumption behavior.

5. **Support Water Management Efforts:** Your expertise in analyzing consumption patterns enables general water management systems to develop policy frameworks along with resource allocation distribution.

## 1.2 LIMITATIONS:

i. **Data Availability and Accuracy** – Lack of reliable water usage data creates accuracy issues because the information is either absent or uncoherent.

ii. **Complexity of Assessment** – The measurement of both direct and indirect water consumption within supply chain systems proves difficult.

iii. **Adoption and Awareness Challenges** – Users alongside industries tend to maintain their current behavior patterns because they lack sufficient reasons or knowledge about the alternatives.

iv. **Technical Challenges** – Implementing AI, Big Data, and Blockchain for real- time analysis requires high computational power.

v. **Lack of Standardized Metrics** – Different methodologies lead to inconsistent water footprint calculations across industries.

# CHAPTER-2

# 2. STUDY OF EXISTING SYSTEM

## 2.1 CASE STUDY

India launched the Water Footprint Calculation Project to measure water utilization within agricultural agriculture industry and domestic spheres. Through their utilization of AI-based technologies and Big Data processing and Blockchain platforms the initiative gathered water usage data and delivered time-sensitive information. Users could check the water footprint of their products through a mobile application or website by either scanning barcodes of products or manually entering product details. Through the tool industries could make their high water-using procedures more efficient and farmers obtained irrigation instructions to reduce water usage. The gathered data showed that agricultural operations consumed 80% of the total water usage focused on rice and sugarcane cultivation. This information helped policy authorities to create water conservation policies while also educating the public. The app feature allowing users to choose their local language provided broad accessibility especially to individuals located in rural regions. Water consumption decreased by 15% in specified regions because of this initiative which created better water sustainability through enhanced resource management and motivated personal along with community water usage awareness.

## 2.2 PROPOSED SYSTEM

Through its Water Footprint Calculation Project the organization intends to build a digital platform which delivers live updates on water utilization levels for all products along with services. The system will gather information by using AI and Big Data and Blockchain to analyze input from industries and agriculture and domestic homes. Users will have access to a simple website and mobile application that enables them to measure product water footprints through barcode scanning and product entry as well as image detection features.

Through the system users will get curated recommendations to minimize water consumption specifically in agricultural and industrial areas. Users across different linguistic regions can access the system because it provides support for multiple local languages. The achieved data serves as essential information for policymakers to establish improved water conservation policies. Real-time system analytics operates as an essential instrument that both monitors water consumption behaviors and stimulates sustainable water preservation advancements. The project works to reduce unnecessary water consumption while raising general awareness about appropriate water usage methods.

# CHAPTER 3

# 3 DATABASE DESIGN

## 3.1 SOFTWARE REQUIREMENTS SPECIFICATION:

### 1. External Interface Requirements:

#### A. User Interfaces:

a. Standard Web Browsers enable users to access the user-friendly web interface of the system.
b. The web interface enables users to start object detection processes while allowing them to view water footprint results through its control panel and support additional system features.
c. A mobile application development for both the iOS and Android operating systems will create interfaces which match web interface functionality.
d. Web interfaces should offer language selection capabilities together with localization features for different users.

#### B. Hardware Interfaces:

a. The system will operate through camera functions available in suitable versions of compatible devices.
b. The system requires smartphone and tablet and computer devices to perform object scanning.
c. The device interfaces need to support camera APIs and drivers which are widely used across different devices.
d. The system runs on any hardware which satisfies the requirements vital to execute the web interface or mobile application.
e. The system operates through web interface and mobile application running on devices.

#### C. Software Interfaces:

a. The system must link with external databases together with APIs in order to retrieve product information.
b. The system must use APIs to acquire information about products alongside production information and water distribution data from external sources.
c. The system needs cloud-based service integration for either data storage or processing needs or web application hosting.

### D. Communications Interfaces:

a. The system needs to use standard HTTP(S) protocols for communication servers and services.
b. Standardized methods need to exist for obtaining data together with processing requirements.
c. The system needs to implement secure transmission protocols based on encryption to safeguard important information.
d. The system requires asynchronous communication patterns which enable concurrent user dealings to operate efficiently.
e. Efficient performance of user requests and data processing assignments is a requirement of the system.

## 2. System Features:

### 1 System Feature 1: Object Detection

a. The system maintains the ability to identify different items using image recognition. recognition technology.
b. Requirements:
c. The system will use trained machine learning models to perform detection of objects. Web interface and mobile application users will find the ability to start object detection operations as a system requirement.
d. application.
e. Real-time or near real-time execution shall be available for object detection because this requirement produces immediate results for users.
f. feedback to users.
g. Users can depend on the system to properly identify detected objects when classification accuracy reaches at least 90% accuracy level.
h. The system detects a varied collection of ordinary home objects among its detection capabilities.
i. food products, beverages, and personal care items.

### 2 System Feature 2: Water Footprint Calculation

a. This feature will compute the water footprint of detected items by using established metrics. predefined metrics and data sources.
b. Requirements:
c. The system should obtain necessary water footprint calculation data by accessing external sources.
d. databases or APIs.
e. Systems should perform water footprint calculations through assessment of water consumption starting from manufacturing operations up to transportation and conclusion phases.
f. transportation, and disposal phases.

g. The calculated water footprint results must use methodologies which have undergone scientific validation as well as industry standard approval and industry standards.
h. Users need to obtain detailed descriptions of water footprint elements that include virtual water usage and geographic allocation information.
i. virtual water usage and geographical distribution.
j. Users require direct access to water footprint calculation procedures for successful verification purposes.

## 3 System Feature 3: Result Presentation

a. The system identifies detected items through its presentation feature which generates their calculated water footprints.
b. clear and user-friendly manner.
c. Requirements:
d. Users can view water footprint results through both web interface and mobile application display.
e. Users will find the complete water footprint value of each detected item within the presented results.
f. Users must have the ability to get detailed separations of water footprint subcomponents. such as blue water, green water, and grey water usage.
g. Visual representations using graphical charts will display water footprint results to all users. graphs, to enhance user understanding.
h. Users will possess the ability to both save and export water footprint reports which they can reference or share at any time.

## 4 System Feature 4: User Management

a. The management of user accounts together with access permissions through this feature falls under its description.
b. permissions within the system.
c. Requirements:
d. Users need to create an account so the system provides access through defined authentication methods.restricted functionalities.
e. Authoritative system users possess the capability to create user profiles and modify these profiles as well as remove inactive accounts.
f. The system should enable administrators to set user roles and access permissions which will limit access to sensitive system data functionalities.
g. The system must apply password policies which will protect user account security through mandatory strength requirements.

## 3.1.2 COLLECTION OF REQUIREMENTS:

### Testing Techniques-

To ensure a smooth and reliable user experience for our Flutter-based frontend, we followed a structured testing approach. Here's how we made sure everything worked perfectly:

**RequirementBased Testing-**
We started by aligning all tests with the project's core requirements. This helped us confirm that every feature not only functioned correctly but also met usability standards and overall project goals.

**Functional Testing-**
Every feature—from input forms and data visualization tools to navigation—was thoroughly checked to ensure it worked exactly as intended. No surprises, just seamless functionality.

**Usability Testing-**
A great app isn't just functional—it's also easy and intuitive to use. We tested the interface with real users to make sure it was accessible, straightforward, and enjoyable to interact with.

**Integration Testing-**
The frontend and backend needed to communicate flawlessly. We rigorously tested how the Flutter app interacted with APIs and backend services to ensure smooth data flow and reliability.

**Performance Testing-**
Speed and stability matter. We put the app through different scenarios to check its responsiveness and performance, making sure it stayed fast and stable under various conditions.

## TESTING METHODOLOGIES:

To build a reliable and user-friendly app, we followed a smart and flexible testing approach. Here’s how we did it:

### Agile Testing-

a. Testing wasn’t an afterthought—it was part of every development sprint.
b. By checking features early and often, we ensured steady progress and high quality at every step.

### Black-Box Testing-

a. We put ourselves in the users’ shoes, testing the app just like they would— without worrying about the code underneath.
b. This helped us focus on real-world usability rather than just technical correctness.

### Exploratory Testing-

a. Sometimes, the best bugs are the ones you don’t expect.
b. We went off-script to hunt down hidden issues that structured tests might miss, making the app more robust.

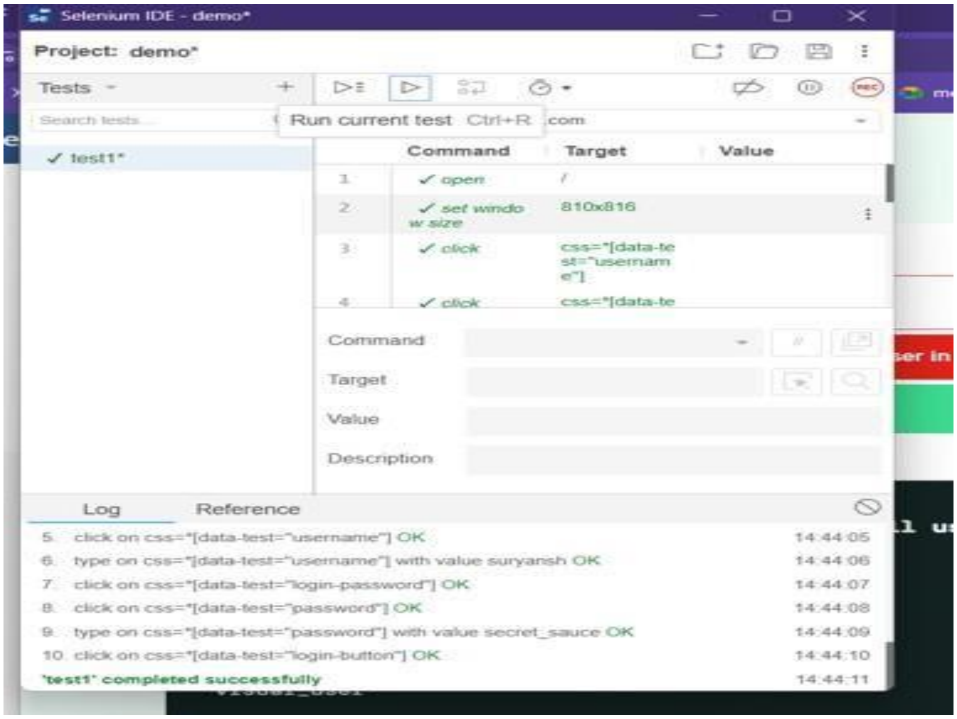
### Cross-Platform Testing-

a. Since Flutter works on both Android and iOS, we made sure the app looked and behaved perfectly on both platforms.
b. No surprises—just a smooth experience, no matter the device.

### Automation Testing-

a. Why waste time on repetitive checks? We used smart tools to handle regression, performance, and other routine tests.
b. Faster results, fewer errors, and more time to focus on what really matters— building great features

# TEST DELIVERABLES:



Selenium IDE - demo*

Project: demo*

Tests

✓ test1*

Run current test Ctrl+R

.com

| | Command | Target | Value |
|---|---|---|---|
| 1 | ✓ open | / | |
| 2 | ✓ set window size | 810x816 | |
| 3 | ✓ click | css="[data-test="username"] | |
| 4 | ✓ click | css="[data-te | |

Command

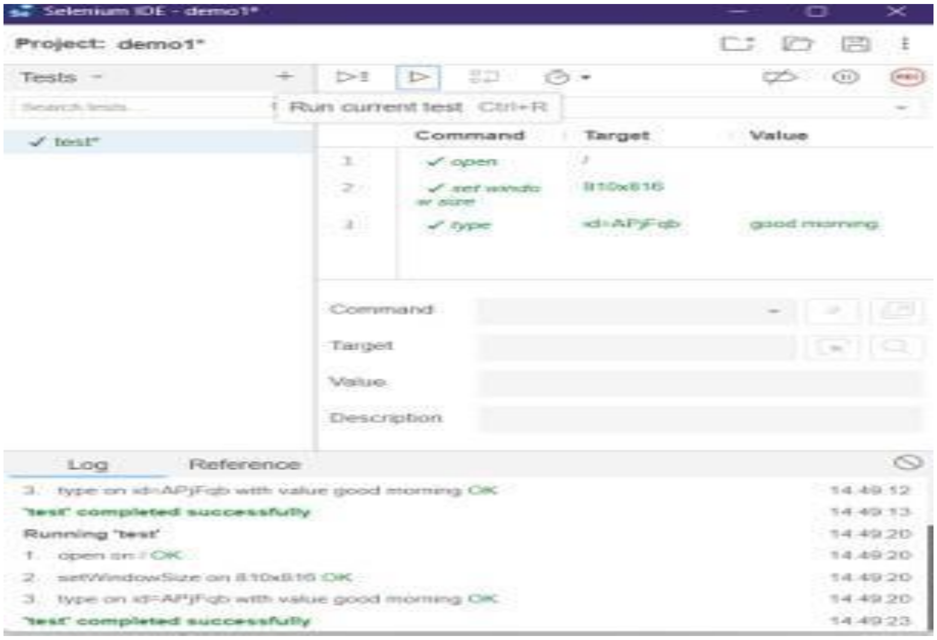
Target

Value

Description

Log Reference

```
5.  click on css="[data-test="username"] OK                           14:44:05
6.  type on css="[data-test="username"] with value suryansh OK         14:44:06
7.  click on css="[data-test="login-password"] OK                     14:44:07
8.  click on css="[data-test="password"] OK                           14:44:08
9.  type on css="[data-test="password"] with value secret_sauce OK     14:44:09
10. click on css="[data-test="login-button"] OK                       14:44:10
'test1' completed successfully                                        14:44:11
```



Selenium IDE - demo1*

Project: demo1*

Tests

✓ test*

Run current test Ctrl+R

| | Command | Target | Value |
|---|---|---|---|
| 1 | ✓ open | / | |
| 2 | ✓ set window size | 810x816 | |
| 3 | ✓ type | id=APjFqb | good morning |

Command

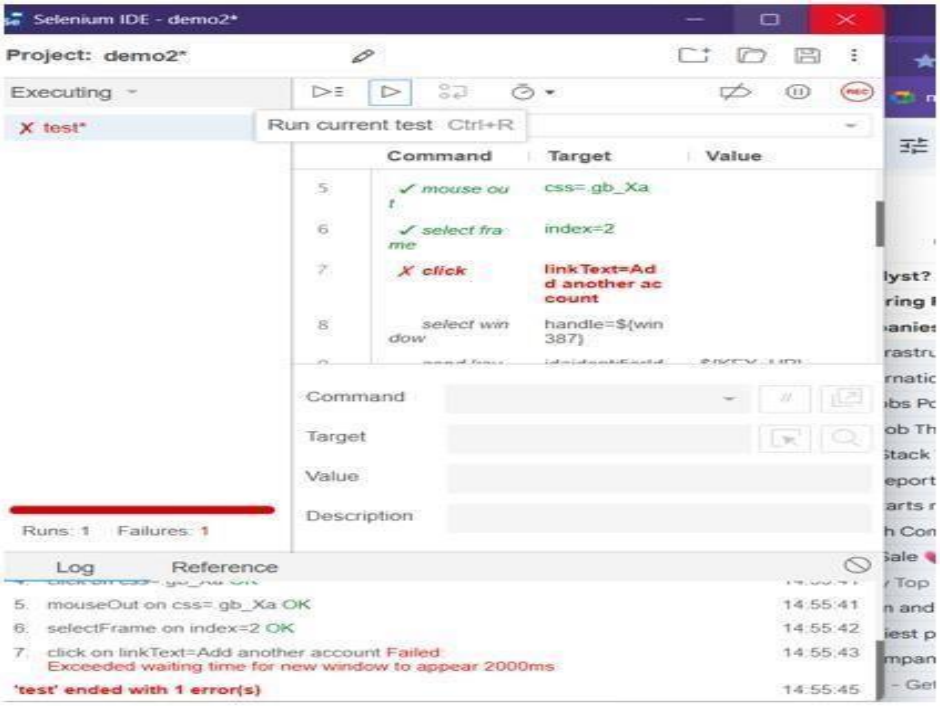
Target

Value

Description

Log Reference

```
3. type on id=APjFqb with value good morning OK        14:49:12
'test' completed successfully                          14:49:13
Running 'test'                                         14:49:20
1. open on / OK                                        14:49:20
2. setWindowSize on 810x816 OK                         14:49:20
3. type on id=APjFqb with value good morning OK        14:49:20
'test' completed successfully                          14:49:23
```



✗ test*

| | Command | Target | Value |
|---|---|---|---|
| 5 | ✓ mouse out | css=.gb_Xa | |
| 6 | ✓ select frame | index=2 | |
| 7 | ✗ click | linkText=Add another account | |
| 8 | select window | handle=${win387} | |

Command

Target

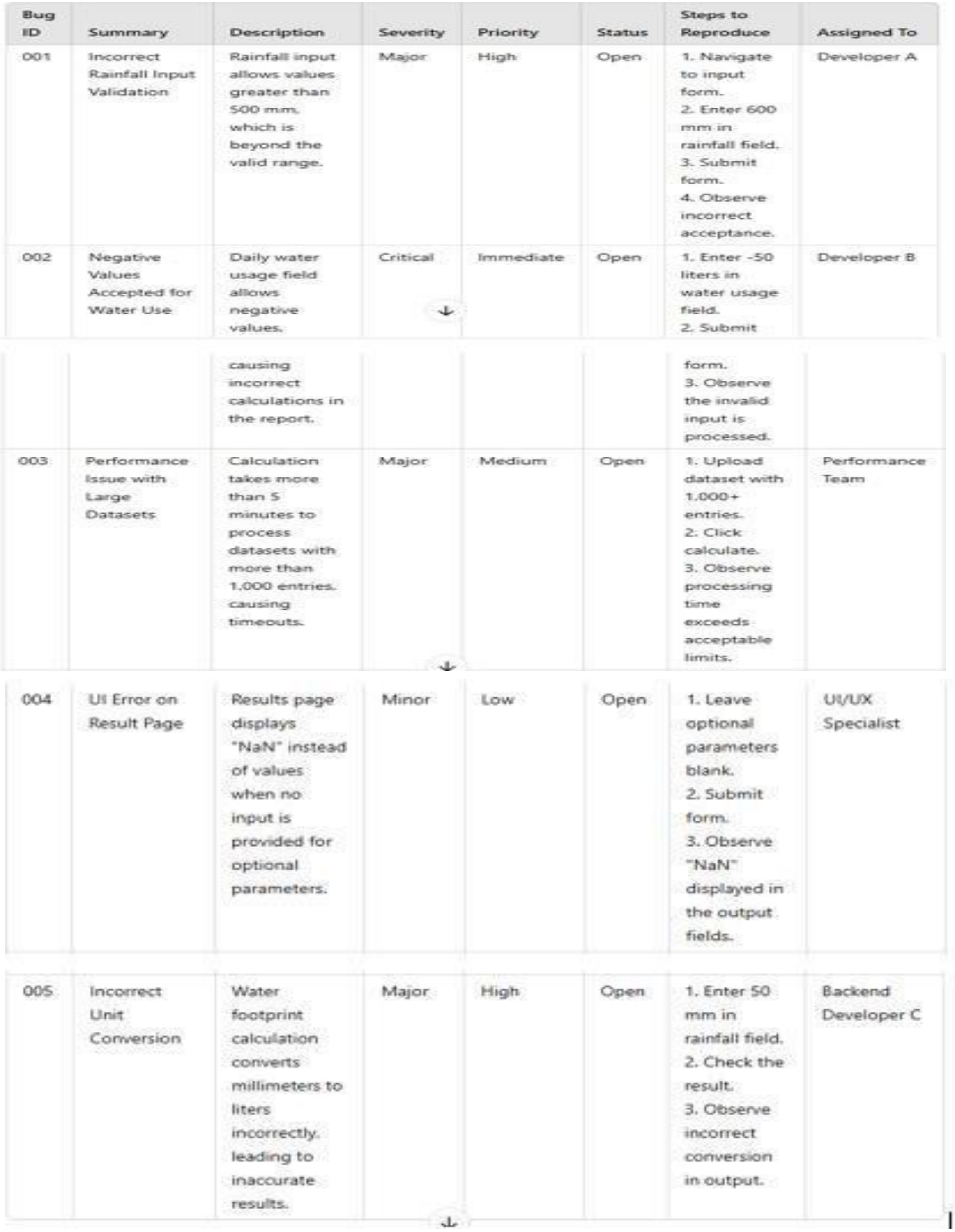
Value

Description

Runs: 1 Failures: 1

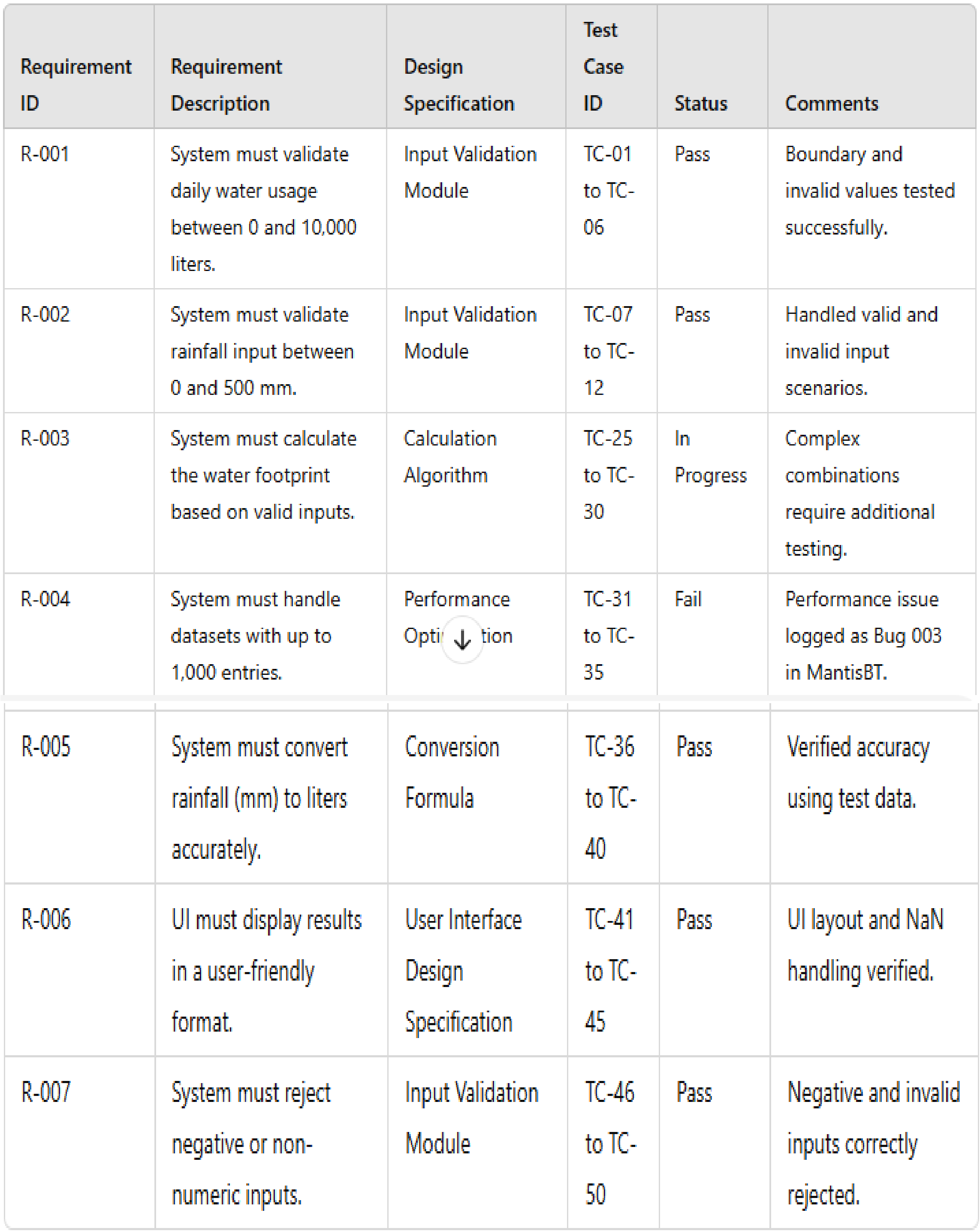
```
5. mouseOut on css=.gb_Xa OK                                      14:55:41
6. selectFrame on index=2 OK                                      14:55:42
7. click on linkText=Add another account Failed:                  14:55:43
   Exceeded waiting time for new window to appear 2000ms
'test' ended with 1 error(s)                                      14:55:45
```

**Bug Report:-** Design a bug report through Mantis BT.



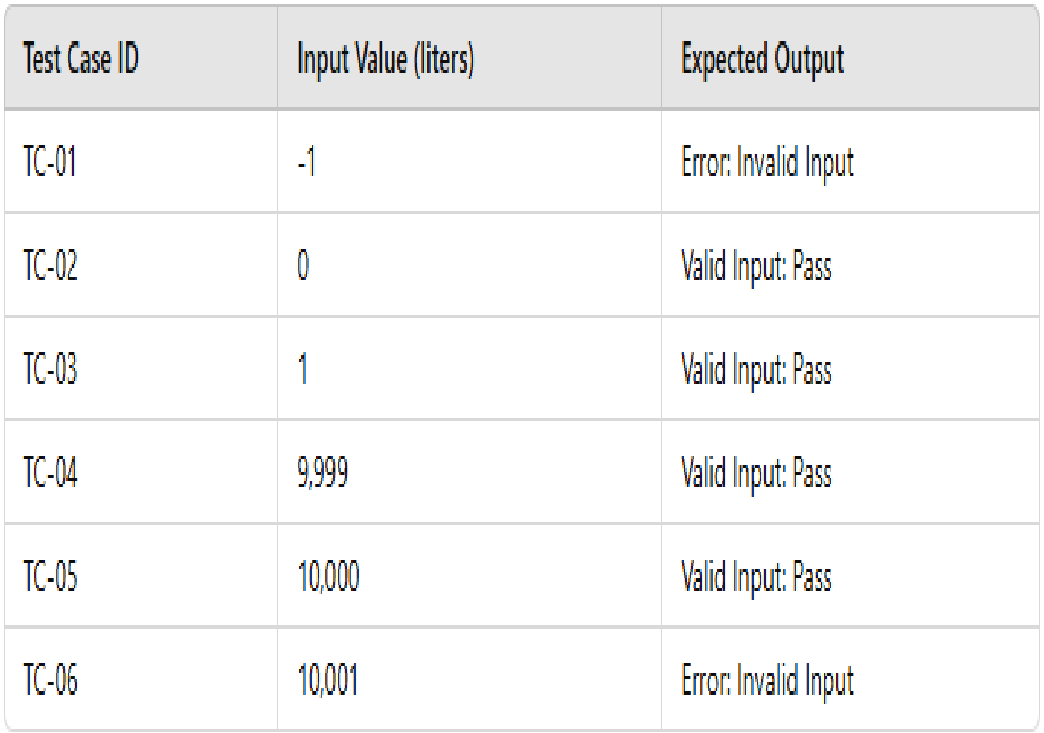
| Bug ID | Summary | Description | Severity | Priority | Status | Steps to Reproduce | Assigned To |
|---|---|---|---|---|---|---|---|
| 001 | Incorrect Rainfall Input Validation | Rainfall input allows values greater than 500 mm, which is beyond the valid range. | Major | High | Open | 1. Navigate to input form.<br>2. Enter 600 mm in rainfall field.<br>3. Submit form.<br>4. Observe incorrect acceptance. | Developer A |
| 002 | Negative Values Accepted for Water Use | Daily water usage field allows negative values, causing incorrect calculations in the report. | Critical | Immediate | Open | 1. Enter -50 liters in water usage field.<br>2. Submit form.<br>3. Observe the invalid input is processed. | Developer B |
| 003 | Performance Issue with Large Datasets | Calculation takes more than 5 minutes to process datasets with more than 1,000 entries, causing timeouts. | Major | Medium | Open | 1. Upload dataset with 1,000+ entries.<br>2. Click calculate.<br>3. Observe processing time exceeds acceptable limits. | Performance Team |
| 004 | UI Error on Result Page | Results page displays "NaN" instead of values when no input is provided for optional parameters. | Minor | Low | Open | 1. Leave optional parameters blank.<br>2. Submit form.<br>3. Observe "NaN" displayed in the output fields. | UI/UX Specialist |
| 005 | Incorrect Unit Conversion | Water footprint calculation converts millimeters to liters incorrectly, leading to inaccurate results. | Major | High | Open | 1. Enter 50 mm in rainfall field.<br>2. Check the result.<br>3. Observe incorrect conversion in output. | Backend Developer C |

## Requirement Traceability Matrix (RTM):-



| Requirement ID | Requirement Description | Design Specification | Test Case ID | Status | Comments |
|---|---|---|---|---|---|
| R-001 | System must validate daily water usage between 0 and 10,000 liters. | Input Validation Module | TC-01 to TC-06 | Pass | Boundary and invalid values tested successfully. |
| R-002 | System must validate rainfall input between 0 and 500 mm. | Input Validation Module | TC-07 to TC-12 | Pass | Handled valid and invalid input scenarios. |
| R-003 | System must calculate the water footprint based on valid inputs. | Calculation Algorithm | TC-25 to TC-30 | In Progress | Complex combinations require additional testing. |
| R-004 | System must handle datasets with up to 1,000 entries. | Performance Opti[illegible]tion | TC-31 to TC-35 | Fail | Performance issue logged as Bug 003 in MantisBT. |
| R-005 | System must convert rainfall (mm) to liters accurately. | Conversion Formula | TC-36 to TC-40 | Pass | Verified accuracy using test data. |
| R-006 | UI must display results in a user-friendly format. | User Interface Design Specification | TC-41 to TC-45 | Pass | UI layout and NaN handling verified. |
| R-007 | System must reject negative or non-numeric inputs. | Input Validation Module | TC-46 to TC-50 | Pass | Negative and invalid inputs correctly rejected. |

## Boundary Value analysis:



| Test Case ID | Input Value (liters) | Expected Output |
|---|---|---|
| TC-01 | -1 | Error: Invalid Input |
| TC-02 | 0 | Valid Input: Pass |
| TC-03 | 1 | Valid Input: Pass |
| TC-04 | 9,999 | Valid Input: Pass |
| TC-05 | 10,000 | Valid Input: Pass |
| TC-06 | 10,001 | Error: Invalid Input |

## 3.1.1 SOFTWARE REQUIREMENTS:

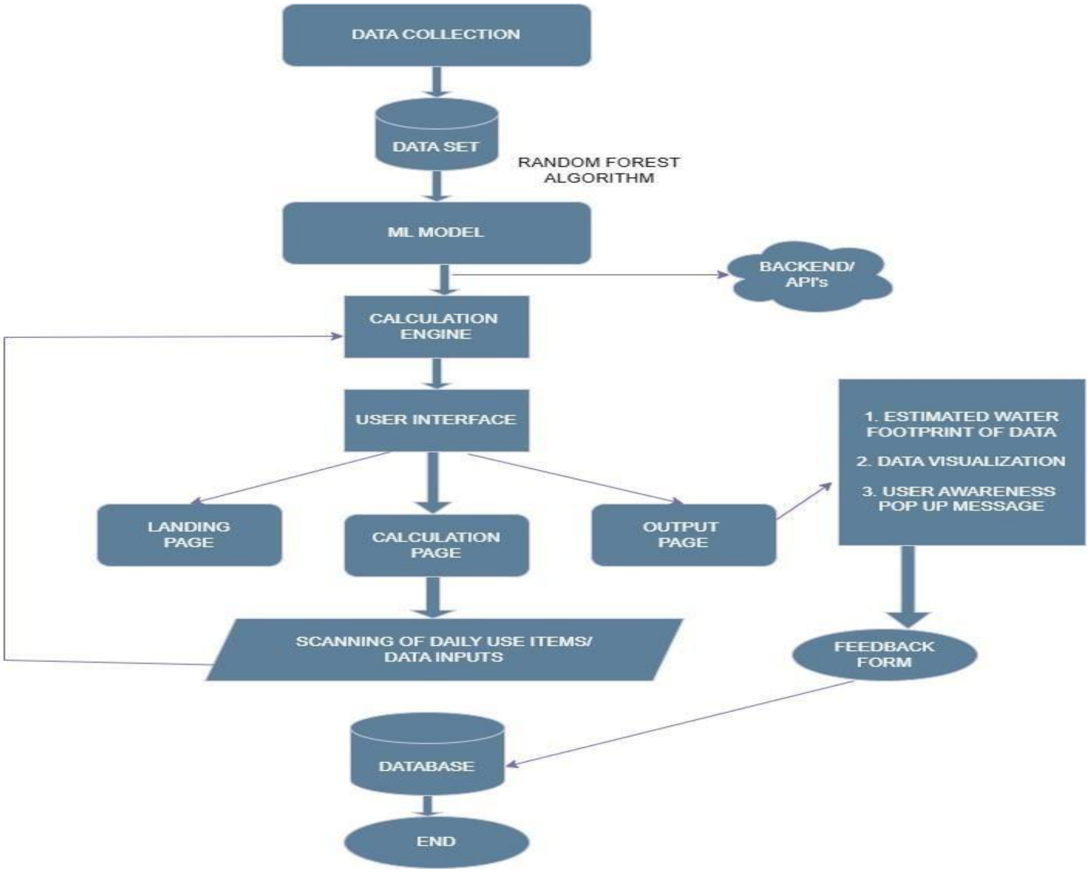
Frontend- HTML, CSS, Java Script, Bootstrap Backend-Python flask (Python 3.7) , SQL Alchemy,

- Operating System: Windows 10
- Google Chrome/Internet Explorer
- XAMPP (Version-3.7)
- Python main editor (user interface): PyCharm Community
- workspace editor: Sublime text 3

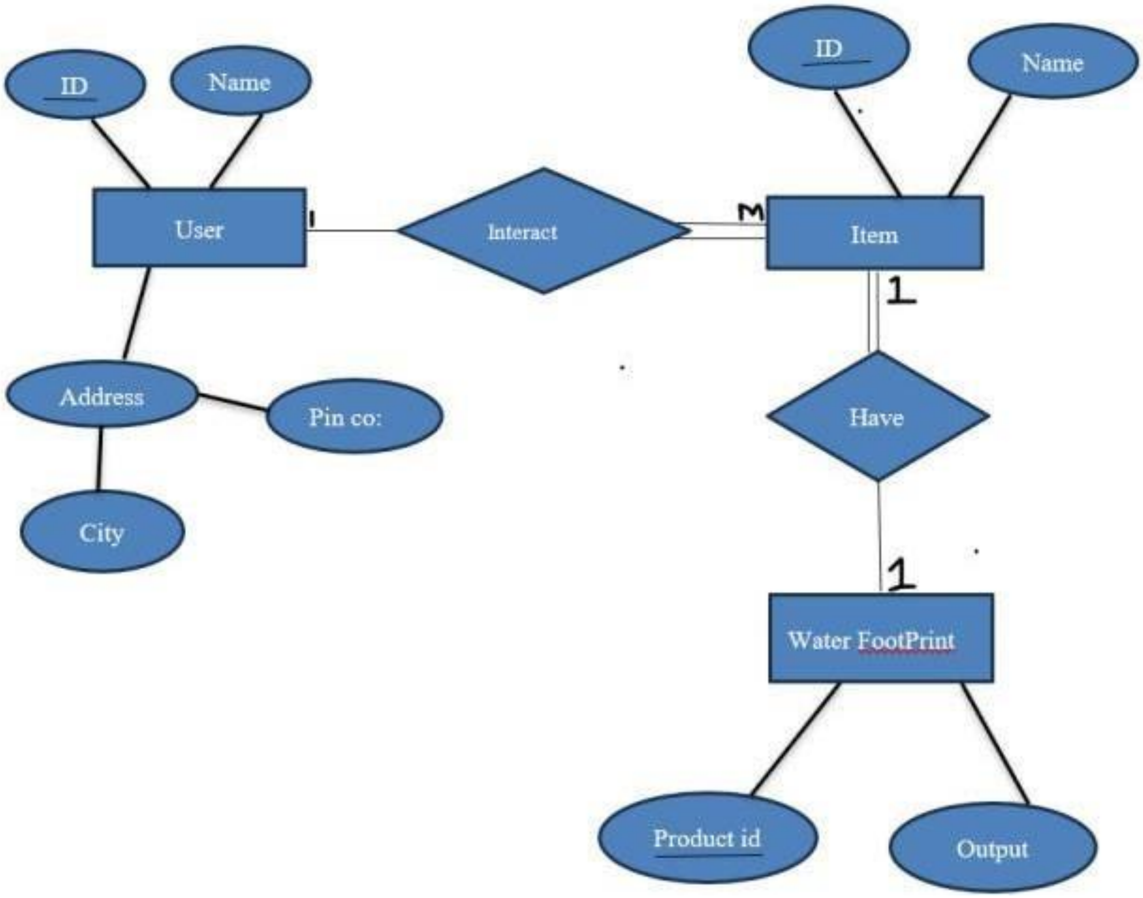
## HARDWARE REQUIREMENTS:

- Computer with a 1.1 GHz or faster processor
- Minimum 2GB of RAM or more
- 2.5 GB of available hard-disk space
- 5400 RPM hard drive
- 1366 × 768 or higher-resolution display
- DVD-ROM drive

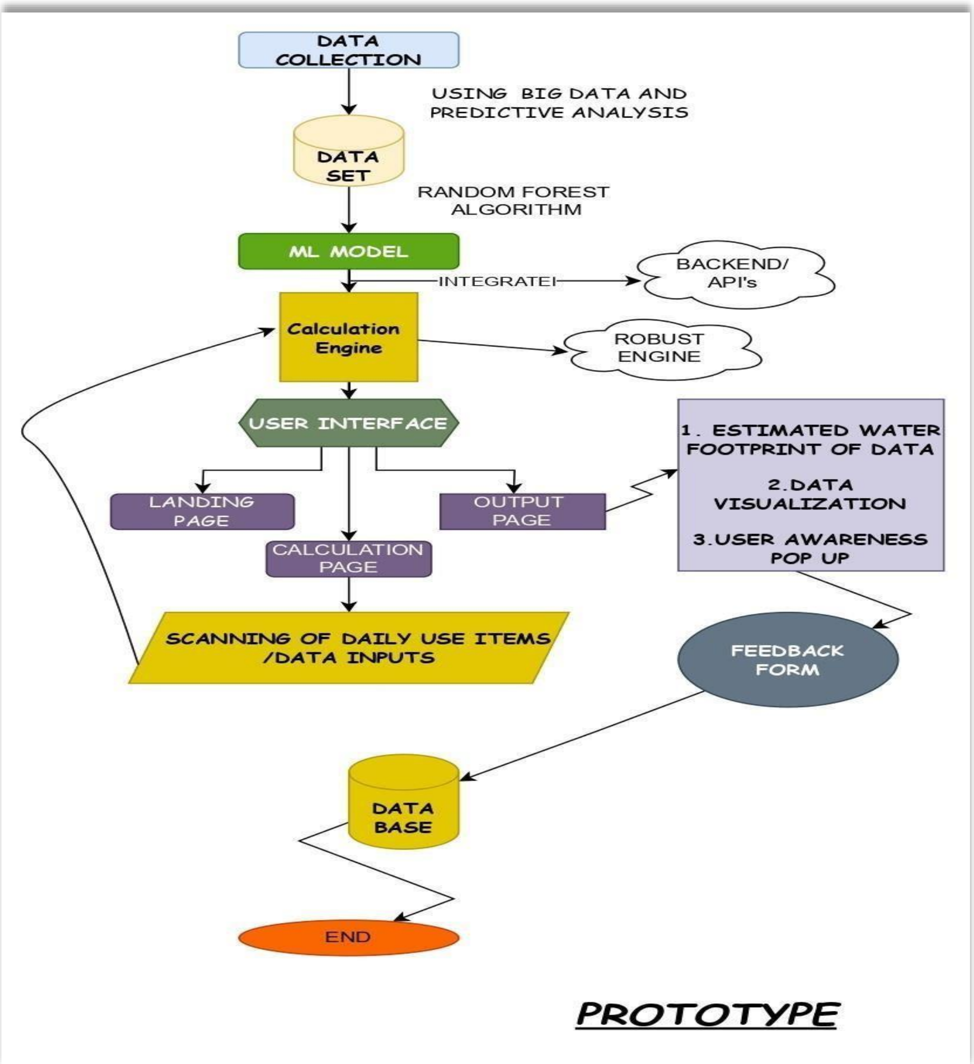
## 3.2 CONCEPTUAL DESIGN:



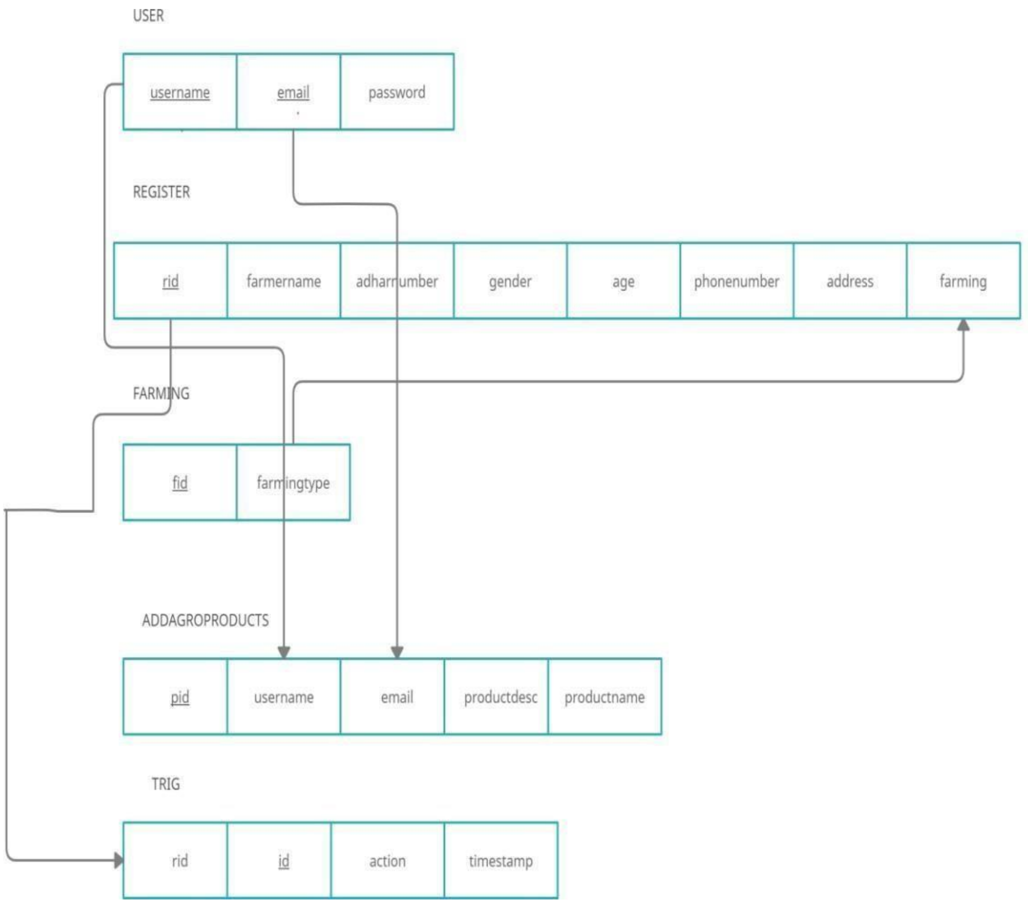
### 3.2.1 E-R DIAGRAM:



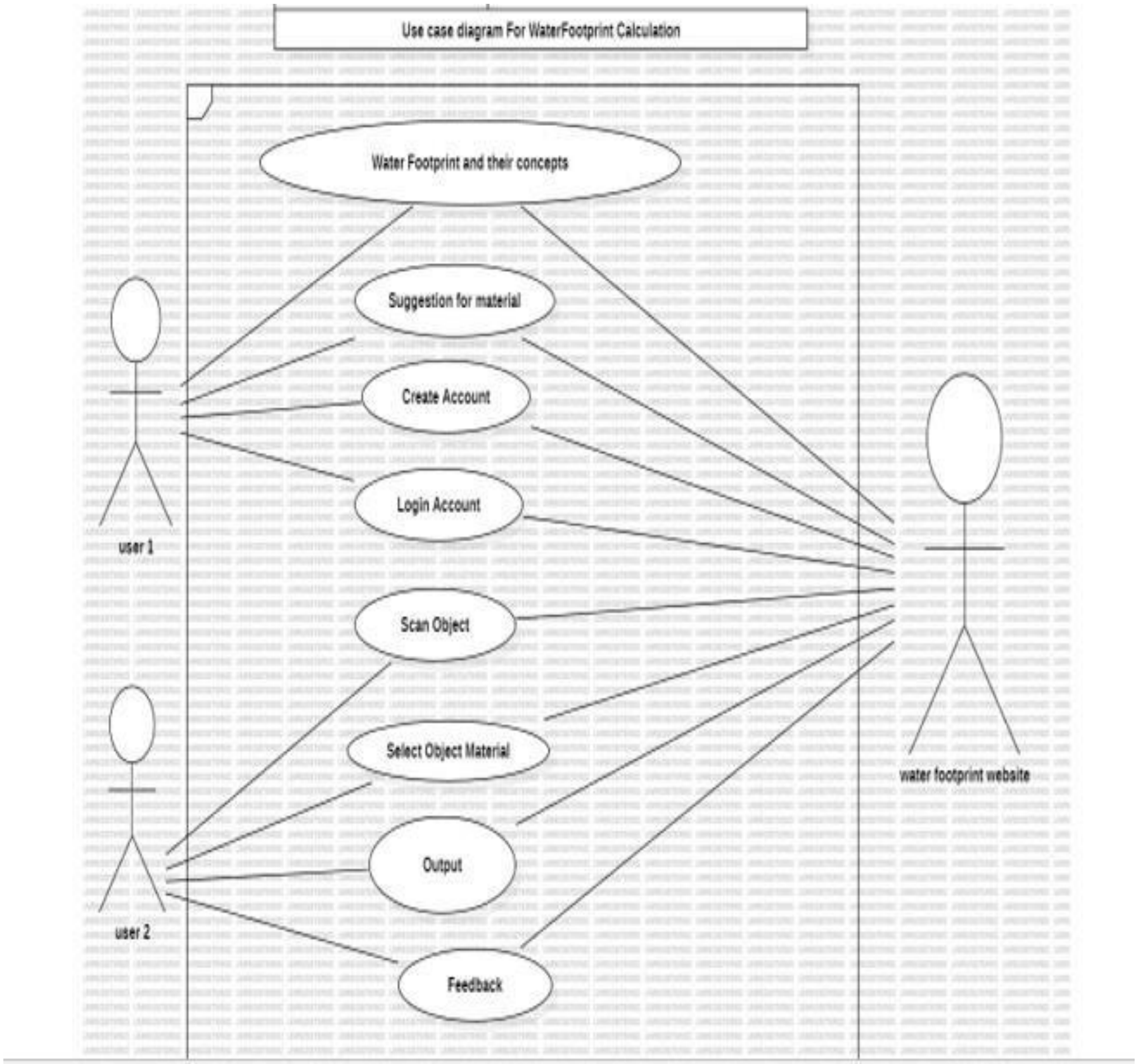
## FLOWCHART:


DATA COLLECTION
USING BIG DATA AND PREDICTIVE ANALYSIS
DATA SET
RANDOM FOREST ALGORITHM
ML MODEL
INTEGRATEI
BACKEND/ API's
Calculation Engine
ROBUST ENGINE
USER INTERFACE
1. ESTIMATED WATER FOOTPRINT OF DATA
2.DATA VISUALIZATION
3.USER AWARENESS POP UP
LANDING PAGE
OUTPUT PAGE
CALCULATION PAGE
SCANNING OF DAILY USE ITEMS /DATA INPUTS
FEEDBACK FORM
DATA BASE
END
PROTOTYPE

### 3.2.3 SCHEMA DIAGRAM:



## USE CASE DIAGRAM:-



# 3.3 IMPLEMENTATION:

An operational system for calculating water consumption needs three main components that consist of a well- organized database together with an accessible user interface and efficient processing capabilities in backend infrastructure. This page includes a structured plan for implementation as follows:

## BackEnd (MySQL)

### Database:

A Database Management System (DBMS) functions as computer software that allows users to manage structural data within databases and operate database requests for many users. Oracle, DB2, Microsoft Access and Microsoft SQL Server and Firebird together with PostgreSQL and MySQL form typical examples of DBMSs. The list continues with SQLite, FileMaker and Sybase Adaptive Server Enterprise. Database administrators rely on DBMSs to build all Database systems. DBMS systems find their typical applications in accounting programs and systems for human resources management and customer service operations. Large-scale companies used to be the only segment with the hardware capacity for managing extensive data sets before DBMSs became mainstream for company back offices.

A database management system represents a sophisticated suite of computer software that directs data organization together with storage and management and retrieval capabilities of database records. A DBMS includes:

Every database inside the DBMS requires a modeling language to establish its schema based on the DBMS data model.

Presently the ad hoc model embedded in SQL remains dominant although relational model purists consider it to be a corrupted implementation which violates multiple foundational relational principles merely for better practicality and higher performance. The Open Database Connectivity API lets programmers access the DBMS through a standard interface that many DBMSs provide.

The data structures use optimized fields, records, files and objects that address large volume data stored on permanent data storage devices with slower retrieval times than main memory. Users access the system through a database query language together with reporting capabilities.

The writer tool enables user interaction with a database for data analysis through inquiry while following user-based permission levels.

The implementation of data security blocks unauthorized users from both viewing and updating databases. The database gives users permission to view the entire system through passwords and also provides access to sub categories of data referred to as sub schemas. An employee database contains complete employee records which particular groups can only see payroll data or work history and student information respectively.

The interactive features which the DBMS provides for entering and updating databases and interrogation serve personal database management needs. The system does not create automated documentation about user activities nor fully support organizational multi- user control requirements. A set of custom-built application programs must exist to enable these controls during data entry and updating activities.

The system requires a transaction mechanism offering full ACID support for maintaining data integrity during simultaneous user interactions and hardware failures.

It also maintains the integrity of the data in the database.

Through its functionality the DBMS protects database integrity by preventing simultaneous updates to the same record by different users. The DBMS enforces unique index constraints to prevent duplicate records from entering the database when customer numbers serve as key fields. The full description of ACID properties explains this method better (Redundancy avoidance).

The adoption of a DBMS enables information systems to adapt more readily because organizations transform their data requirements. Organizations generally run their daily transaction programs through one type of DBMS before transferring details to a different DBMS system for analysis needs. Data administrators together with systems analysts perform the overall systems design. Database administrators carry out the process of detailed database design.

## SQL:

The programming language Structured Query Language (SQL) serves to manipulate relational database systems. The relational model contains a tight connection with Structured Query Language (SQL).

- In the relational model, data is stored in structures called relations or tables.

SQL statements are issued for the purpose of:

- Data definition: Defining tables and structures in the database (DDL used to create, alter and drop schema objects such as tables and indexes)

.

## 4.2 : Stored Procedure

Routine name:
proc Type:
procedur e
Definition: Select * from register;

### 4.3 : Triggers

It is the special kind of stored procedure that automatically executes when an event occurs in the database.

Triggers used :

1: Trigger name: on insert Table:
register

Time: after Event:

insert

INSERT INTO trig VALUES(null,NEW.rid,'Farmer Inserted',NOW())

2: Trigger name: on delete Table:

register Time:

after Event:

delete

Definition: INSERT INTO trig VALUES(null,OLD.rid,'FARMER DELETED',NOW())

3: Trigger name: on update Table:

register Time:

after Event:

update

Definition: INSERT INTO trig VALUES(null,NEW.rid,'FARMER UPDATED',NOW())

# BACKEND PYHTON WITH MYSQL CODE

```
from flask import
Flask,render_template,request,session,redirect,url_for,flash from
flask_sqlalchemy import SQLAlchemy from flask_login import
UserMixin from werkzeug.security

import generate_password_hash,check_password_hash from flask_login
                                                            import
login_user,logout_user,login_manager,LoginManager from
flask_login import login_required,current_user



# MY db
connection
local_server= True
app = Flask( name
        )
app.secret_key='ha
rshithbhaska r'
# this is for getting unique user
                  access
login_manager=LoginMana
ger(app)
login_manager.login_view
='login'

@login_manager.
user_loader def
load_user(user_i d):
return User.query.get(int(user_id))

#
app.config['SQLALCHEMY_DATABASE_URL']='mysql://username:password@localhost/
databas_tabl e_ name'
app.config['SQLALCHEMY_DATABASE_URI']='mysql://root:@localhost/farmers'
```

```
db=SQLAlchemy(app)

# here we will create db models
that is tables class Test(db.Model):
    id=db.Column(db.Integer,primary_key=True)
    name=db.Column(db.String(100))

class Farming(db.Model):
    fid=db.Column(db.Integer,primary_key=
    True)
    farmingtype=db.Column(db.String(100)
 class Addagroproducts(db.Model):
    username=db.Column(db.String(50))
    email=db.Column(db.String(50))
    pid=db.Column(db.Integer,primary_key=
    True)
    productname=db.Column(db.String(100)
    )
    productdesc=db.Column(db.String(300))
    price=db.Column(db.Integer)



class Trig(db.Model):
    id=db.Column(db.Integer,primary_key=T
    rue)        fid=db.Column(db.String(100))
    action=db.Column(db.String(100))
    timestamp=db.Column(db.String(100))
class User(UserMixin,db.Model):
    id=db.Column(db.Integer,primary_key=True)
    username=db.Column(db.String(50))
    email=db.Column(db.String(50),unique=True)
    password=db.Column(db.String(1000))

class Register(db.Model):
    rid=db.Column(db.Integer,primary_key=
    True)
    farmername=db.Column(db.String(50))
    adharnumber=db.Column(db.String(50))
    age=db.Column(db.Integer)
    gender=db.Column(db.String(50))
    phonenumber=db.Column(db.String(50))
    address=db.Column(db.String(50))
    farming=db.Column(db.String(50))
@a
pp.
rou
te('/
')
def
ind
ex(
):
    return render_template('index.html')


@app.route('/farmerdetails')
@login_required
def farmerdetails():
    query=db.engine.execute(f"SELECT   *   FROM
    `register`")                                    return
    render_template('farmerdetails.html',query=quer
    y)


@app.route('/agro
products')        def
agroproducts():
    query=db.engine.execute(f"SELECT          *          FROM
    `addagroproducts`")                                    return
    render_template('agroproducts.html',query=query)


@app.route('/addagroproduct',methods=['PO
  ST','GET']) @login_required def
  addagroproduct():
    if request.method=="POST":
      username=request.form.get('username'
      )
      email=request.form.get('email')
      productname=request.form.get('productname')
      productdesc=request.form.get('productdesc')
      price=request.form.get('price')

  products=Addagroproducts(username=username,email=email,productname=productname,pro
  ductdesc=pr od uctdesc,price=price) db.session.add(products) db.session.commit()
  flash("Product Added","info") return redirect('/agroproducts')

    return render_template('addagroproducts.html')

  @app.route('/triggers'
  )
  @logi
  n_req
  uired
  def
  trigger
  s():
    query=db.engine.execute(f"SELECT * FROM `trig`")
    return render_template('triggers.html',query=query)
@app.route('/addfarming',methods=['POST','GE
  T']) @login_required
  def addfarming():
    if request.method=="POST":
      farmingtype=request.form.get('farming')
      query=Farming.query.filter_by(farmingtype=farming
      type).first() if query:
        flash("Farming Type Already Exist","warning")
        return redirect('/addfarming')
      dep=Farming(farmingtype
        =farmingtype)
        db.session.add(dep)
        db.session.commit()
        flash("Farming
        Addes","success") return
    render_template('farming.html')




@app.route("/delete/<string:rid>",methods=['POST','GET'])
@login_required
def delete(rid):
    db.engine.execute(f"DELETE  FROM  `register`  WHERE  `register`.`rid`={rid}")
    flash("Slot Deleted Successful","danger")
    return redirect('/farmerdetails')


@app.route("/edit/<string:rid>",methods=['P
OST','GET']) @login_required
def edit(rid):
    farming=db.engine.execute("SELECT * FROM
    `farming`")
    posts=Register.query.filter_by(rid=rid).first()                 if request.method=="POST":
        farmername=request.form.get('farmername')
        adharnumber=request.form.get('adharnumber')
        age=request.form.get('age')
        gender=request.form.get('gender')
        phonenumber=request.form.get('phonenumber')
        address=request.form.get('address')
        farmingtype=request.form.get('farmingtype')
        query=db.engine.execute(f"UPDATE `register` SET
`farmername`='{farmername}',`adharnumber`='{adharnumber}',`age`='{age}',`gender`='{gender}',`phone
nu          mber`='{phonenumber}',`address`='{address}',`farming`='{farmingtype}'")
            flash("Slot                                                                    is
Updates","success") return redirect('/farmerdetails')
```

```
    return render_template('edit.html',posts=posts,farming=farming)



@app.route('/signup',methods=['PO
ST','GET']) def signup():
    if request.method == "POST": username=request.form.get('username')
    email=request.form.get('email')
        password=request.form.get('passwo
        rd') print(username,email,password)
        user=User.query.filter_by(email=e
        mail).first() if user:
            flash("Email Already
            Exist","warning") return
            render_template('/signup.h
            tml')
        encpassword=generate_password_hash(password)
        new_user=db.engine.execute(f"INSERT INTO `user` (`username`,`email`,`password`) VALUES
('{username}','{email}','{encpassword}')")

        # this is method 2 to save data in db
        #
                           newuser=User(username=username,email=email,pass
                           word=encpassword) # db.session.add(newuser)
                                                            #
        db.session.commit()
        flash("Signup Succes Please Login","success") return render_template('login.html')



    return
render_template('signup.html')
@app.route('/login',methods=['POS
T','GET']) def login():
    if request.method == "POST":
        email=request.form.get('email')
        password=request.form.get('password'
        )
        user=User.query.filter_by(email=email)
        .first()

        if              user              and
          check_password_hash(user.password,pass
          word): login_user(user)    flash("Login
          Success","primary")      return redirect(url_for('index'))
        else:
          flash("invalid
          credentials","danger")
          return
          render_template('login.ht
          ml')

      return render_template('login.html')


    @app.rou
    te('/logout
    ')
    @login_r
    equired
    def
    logout():
    logout_us
    er()
      flash("Logout
      SuccessFul","warning")
      return
      redirect(url_for('login'))
@app.route('/register',methods=['POST','GET'])
@login_required
    def register():
      farming=db.engine.execute("SELECT      *      FROM
      `farming`") if request.method=="POST":
        farmername=request.form.get('farmername')
        adharnumber=request.form.get('adharnumber')
        age=request.form.get('age')
        gender=request.form.get('gender')
        phonenumber=request.form.get('phonenumber'
        )           address=request.form.get('address')
        farmingtype=request.form.get('farmingtype')
        query=db.engine.execute(f"INSERT      INTO
        `register`
(`farmername`,`adharnumber`,`age`,`gender`,`phonenumber`,`address`,`farming`)
VALUES ('{farmername}','{adharnumber}','{age}','{gender}','{phonenumber}','{address}','{farmingtype}'
        )")    flash("Your    Record    Has    Been    Saved","success")    return
    redirect('/farmerdetails')                                         return
    render_template('farmer.html',farming=farming)


@app.
route('/
test')
def
test():
    try:
        Test.query.all()
        return  'My  database
        is Connected'
    except:
        return 'My db is not Connected'



app.run(debug=True)
```

## 3.3.1 FRONT END CODE

```
<!DOCTYPE html>
<html lang="en">


<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">


  <title>{% block title %}
  {% endblock title %}</title>
  <meta content="" name="description">
  <meta content="" name="keywords">


{% block style %}
{% endblock style %}
  <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,700,7
00i|Raleway: 300,400,50 0,700,800" rel="stylesheet">


  <!-- Vendor CSS Files -->
  <link href="static/assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="static/assets/vendor/venobox/venobox.css" rel="stylesheet">
  <link href="static/assets/vendor/font-awesome/css/font-awesome.min.css" rel="stylesheet">
  <link    href="static/assets/vendor/owl.carousel/assets/owl.carousel.min.css"
  rel="stylesheet">            <link            href="static/assets/vendor/aos/aos.css"
  rel="stylesheet">


  <!-- Template Main CSS File -->
  <link href="static/assets/css/style.css" rel="stylesheet">




</head>
```

```
<body>
  <!-- ======= Header ======= -->
  <header id="header">
    <div class="container">


      <div id="logo" class="pull-left">


        <a href="/" class="scrollto">F.M.S</a>
      </div>


      <nav id="nav-menu-container">
        <ul class="nav-menu">
          <li class="{% block home %}
          {% endblock home %}"><a href="/">Home</a></li>

 <li><a href="/register">Farmer Register</a></li>
 <li><a href="/addfarming">Add Farming</a></li>
 <li><a href="/farmerdetails">Farmer Details</a></li>
 <li><a href="/agroproducts">Agro Products</a></li>
 <li><a href="/triggers">Records</a></li>

{% if current_user.is_authenticated %}
          <li class="buy-tickets"><a href="">Welcome {{current_user.username}}</a></li>
            <li class="buy-tickets"><a href="/logout">Logout</a></li>
          {% else %}
          <li class="buy-tickets"><a href="/signup">Signin</a></li>


          {% endif %}
        </ul>
      </nav><!-- #nav-menu-container -->
      </div>
    </header><!-- End Header -->

    <!-- ======= Intro Section ======= -->
   <section id="intro">
      <div class="intro-container" data-aos="zoom-in" data-aos-delay="100">
   <h1 class="mb-4 pb-0">SELL AGRO PRODUCTS AND BUY </span> </h1>
 <p class="mb-4 pb-0">DBMS Mini Project Using Flask &
 MYSQL</p>

        <a href="/agroproducts" class="about-btn
      scrollto">AGRO PRODUCTS</a> </div>
    </section><!-- End Intro Section -->
    <main id="main">

    {% block body %}

  {% with messages=get_flashed_messages(with_categories=true) %}
  {% if messages %}
  {% for category, message in messages %}

<div class="alert alert-{{category}} alert-dismissible fade show"
role="alert">
      {{message}}

  </div>

    {% endfor %}
    {% endif %}
    {% endwith %}
    {% endblock body %}
  <a href="#" class="back-to-top"><i class="fa fa-angle-up"></i></a>
  <!-- Vendor JS Files -->
  <script src="static/assets/vendor/jquery/jquery.min.js"></script>
  <script   src="static/assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
  <script  src="static/assets/vendor/jquery.easing/jquery.easing.min.js"></script>
  <script   src="static/assets/vendor/php-email-form/validate.js"></script>
  <script  src="static/assets/vendor/venobox/venobox.min.js"></script>
  <script  src="static/assets/vendor/owl.carousel/owl.carousel.min.js"></script>
  <script  src="static/assets/vendor/superfish/superfish.min.js"></script>
  <script  src="static/assets/vendor/hoverIntent/hoverIntent.js"></script>
  <script src="static/assets/vendor/aos/aos.js"></script>



  <!-- Template Main JS File -->
  <script src="static/assets/js/main.js"></script>


</body>
</html> <!DOCTYPE html>
<html lang="en">


<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">


  <title>{% block title %}
  {% endblock title %}</title>
  <meta content="" name="description">
  <meta content="" name="keywords">


{% block style %}
{% endblock style %}
  <link
```

```
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,700,7
00i|Raleway: 300,400,50 0,700,800" rel="stylesheet">
  <!-- Vendor CSS Files -->
  <link href="static/assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="static/assets/vendor/venobox/venobox.css" rel="stylesheet">
  <link href="static/assets/vendor/font-awesome/css/font-awesome.min.css" rel="stylesheet">
  <link href="static/assets/vendor/owl.carousel/assets/owl.carousel.min.css" rel="stylesheet">
  <link href="static/assets/vendor/aos/aos.css" rel="stylesheet">


  <!-- Template Main CSS File -->
  <link href="static/assets/css/style.css" rel="stylesheet">



</head>


<body>


  <!-- ======= Header ======= -->
  <header id="header">
   <div class="container">
    <div id="logo" class="pull-left">


     <a href="/" class="scrollto">F.M.S</a>
    </div>


    <nav id="nav-menu-container">
     <ul class="nav-menu">
      <li class="{% block home %}
      {% endblock home %}"><a href="/">Home</a></li>


 <li><a href="/register">Farmer Register</a></li>
 <li><a href="/addfarming">Add Farming</a></li>
 <li><a href="/farmerdetails">Farmer Details</a></li>
```

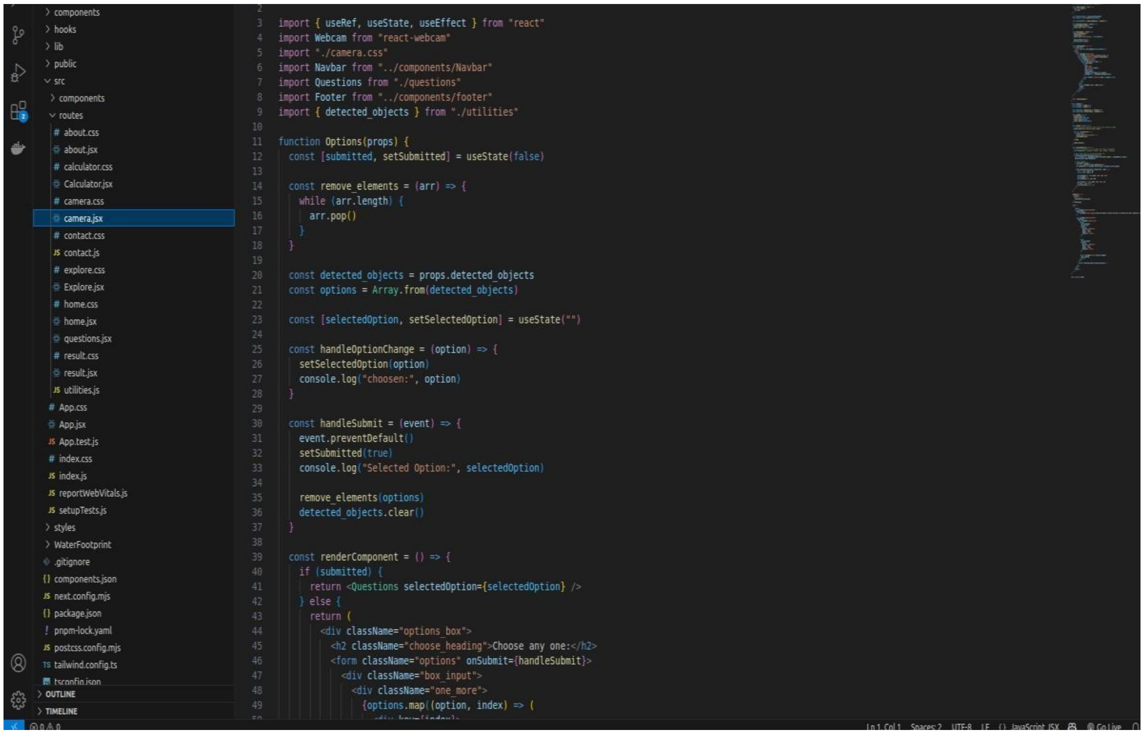
```
{% if current_user.is_authenticated %}
            <li class="buy-tickets"><a href="">Welcome {{current_user.username}}</a></li>
             <li class="buy-tickets"><a href="/logout">Logout</a></li>
            {% else %}
            <li class="buy-tickets"><a href="/signup">Signin</a></li>

            {% endif %}
           </ul>
          </nav><!-- #nav-menu-container -->
         </div>
        </header><!-- End Header -->

        <!-- ======= Intro Section ======= -->
       <section id="intro">
         <div class="intro-container" data-aos="zoom-in" data-aos-delay="100">
          <h1 class="mb-4 pb-0">SELL AGRO PRODUCTS AND BUY </span> </h1>
          <p class="mb-4 pb-0">DBMS Mini Project Using Flask & MYSQL</p>



          <a     href="/agroproducts"     class="about-btn
         scrollto">AGRO PRODUCTS</a> </div>
        </section><!-- End Intro Section -->
        <main id="main">



        {% block body %}

       {% with messages=get_flashed_messages(with_categories=true) %}
       {% if messages %}
       {% for category, message in messages %}

       <div class="alert alert-{{category}} alert-dismissible fade show" role="alert">
```

```
    {{message}}



</div>



  {% endfor %}
  {% endif %}
  {% endwith %}
  {% endblock body %}



  <a href="#" class="back-to-top"><i class="fa fa-angle-up"></i></a>

  <!-- Vendor JS Files -->
  <script src="static/assets/vendor/jquery/jquery.min.js"></script>
  <script   src="static/assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
  <script  src="static/assets/vendor/jquery.easing/jquery.easing.min.js"></script>
  <script   src="static/assets/vendor/php-email-form/validate.js"></script>
  <script  src="static/assets/vendor/venobox/venobox.min.js"></script>
  <script  src="static/assets/vendor/owl.carousel/owl.carousel.min.js"></script>
  <script  src="static/assets/vendor/superfish/superfish.min.js"></script>
  <script  src="static/assets/vendor/hoverIntent/hoverIntent.js"></script>
  <script src="static/assets/vendor/aos/aos.js"></script>



  <!-- Template Main JS File -->
  <script src="static/assets/js/main.js"></script>

</body>


</html>
```

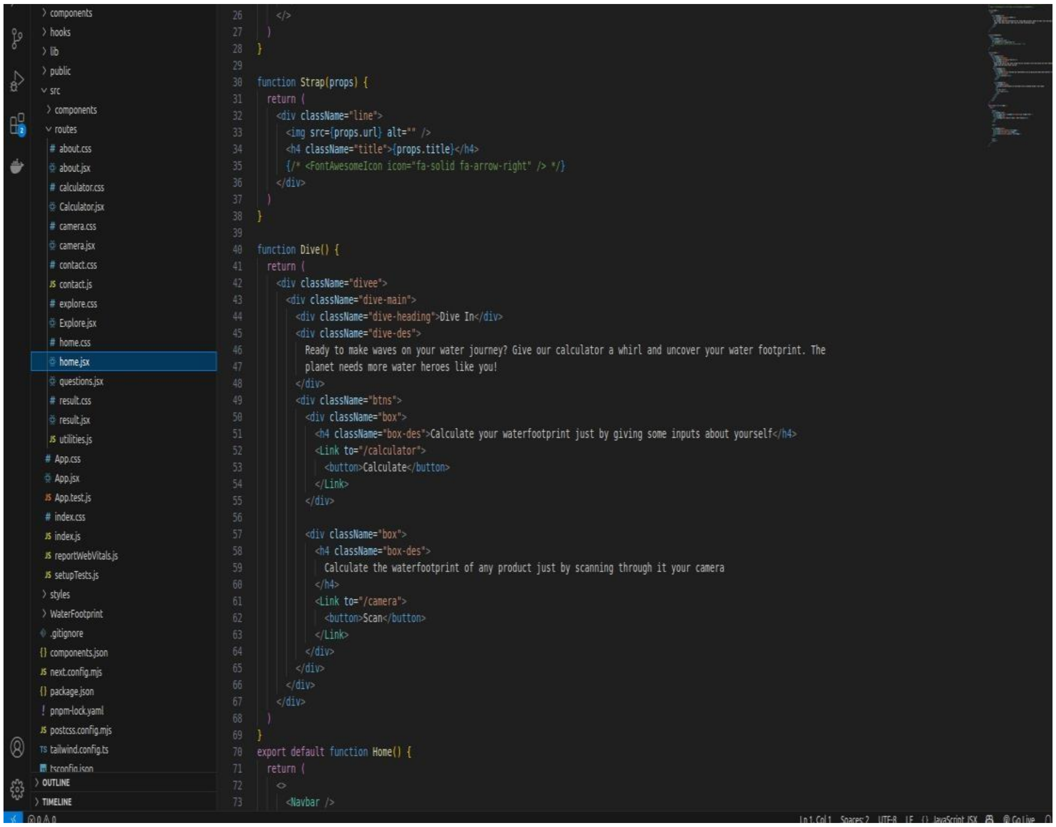
## Farmer.html

```
{% extends 'base.html' %}
{%
block
title  %}
Add
Farming
{% endblock title %}


{% block body %}
 <h3 class="text-center bg-success text-white"><span>Add Farming</span> </h3>


{% with messages=get_flashed_messages(with_categories=true) %}
{% if messages %}
{% for category, message in messages %}


<div class="alert alert-{{category}} alert-dismissible fade show" role="alert">
    {{message}}



</div>
  {% endfor %}
  {% endif %}
  {% endwith %}
<br>
<div class="container">

<div class="row">


<div class="col-md-4"></div>
<div class="col-md-4">
<form action="/addfarming" method="post">
```

```
<input type="text" class="form-control" name="farming" id="farming">
</div>
<br>


  <button type="submit" class="btn btn-success btn-sm btn-block">Add Farming</button>
</form>
<br>
<br>


</div>


<div class="col-md-4"></div>


</div></div>
{% endblock body %}
```

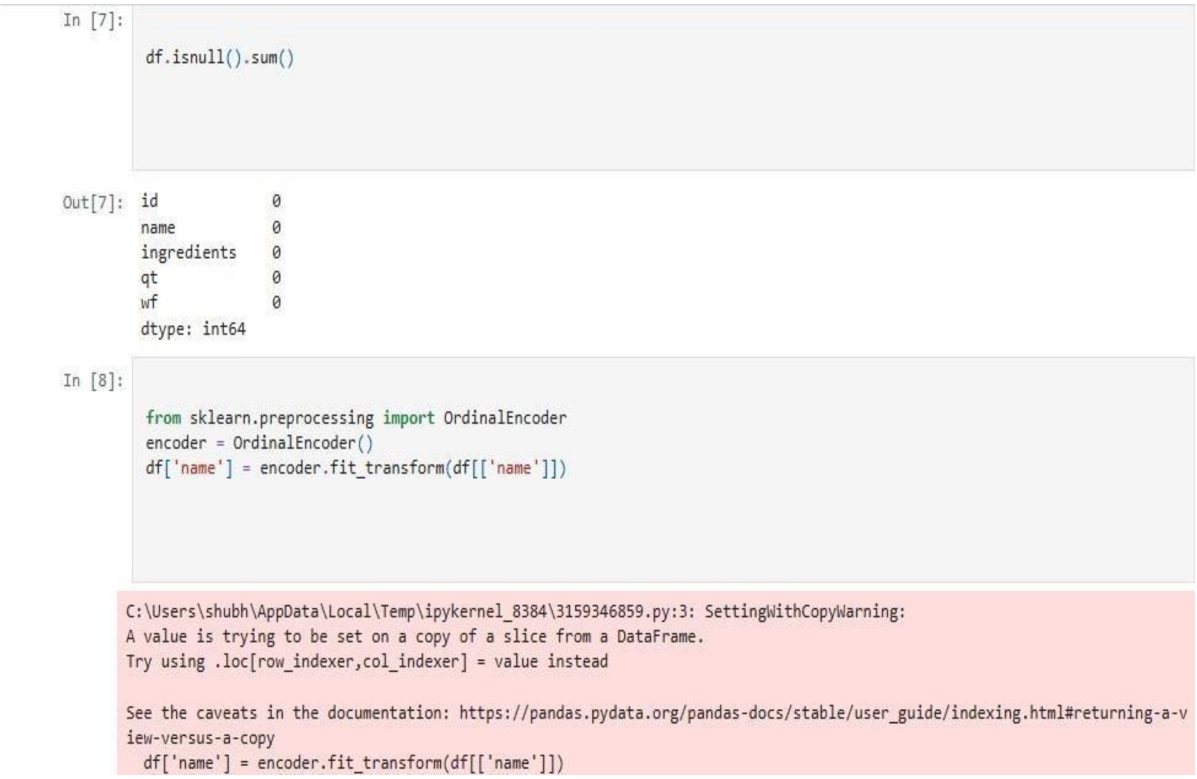


```
import { useRef, useState, useEffect } from "react"
import Webcam from "react-webcam"
import "./camera.css"
import Navbar from "../components/Navbar"
import Questions from "./questions"
import Footer from "../components/footer"
import { detected_objects } from "./utilities"

function Options(props) {
  const [submitted, setSubmitted] = useState(false)

  const remove_elements = (arr) => {
    while (arr.length) {
      arr.pop()
    }
  }

  const detected_objects = props.detected_objects
  const options = Array.from(detected_objects)

  const [selectedOption, setSelectedOption] = useState("")

  const handleOptionChange = (option) => {
    setSelectedOption(option)
    console.log("choosen:", option)
  }

  const handleSubmit = (event) => {
    event.preventDefault()
    setSubmitted(true)
    console.log("Selected Option:", selectedOption)

    remove_elements(options)
    detected_objects.clear()
  }

  const renderComponent = () => {
    if (submitted) {
      return <Questions selectedOption={selectedOption} />
    } else {
      return (
        <div className="options_box">
          <h2 className="choose_heading">Choose any one:</h2>
          <form className="options" onSubmit={handleSubmit}>
            <div className="box_input">
              <div className="one_more">
                {options.map((option, index) => (
                  <div key={index}>
```



```
    </>
  )
}

function Strap(props) {
  return (
    <div className="line">
      <img src={props.url} alt="" />
      <h4 className="title">{props.title}</h4>
      {/* <FontAwesomeIcon icon="fa-solid fa-arrow-right" /> */}
    </div>
  )
}

function Dive() {
  return (
    <div className="divee">
      <div className="dive-main">
        <div className="dive-heading">Dive In</div>
        <div className="dive-des">
          Ready to make waves on your water journey? Give our calculator a whirl and uncover your water footprint. The
          planet needs more water heroes like you!
        </div>
        <div className="btns">
          <div className="box">
            <h4 className="box-des">Calculate your waterfootprint just by giving some inputs about yourself</h4>
            <Link to="/calculator">
              <button>Calculate</button>
            </Link>
          </div>

          <div className="box">
            <h4 className="box-des">
              Calculate the waterfootprint of any product just by scanning through it your camera
            </h4>
            <Link to="/camera">
              <button>Scan</button>
            </Link>
          </div>
        </div>
      </div>
    </div>
  )
}
export default function Home() {
  return (
    <>
      <Navbar />
```

# OBJECT DETECTION MODEL CODE:

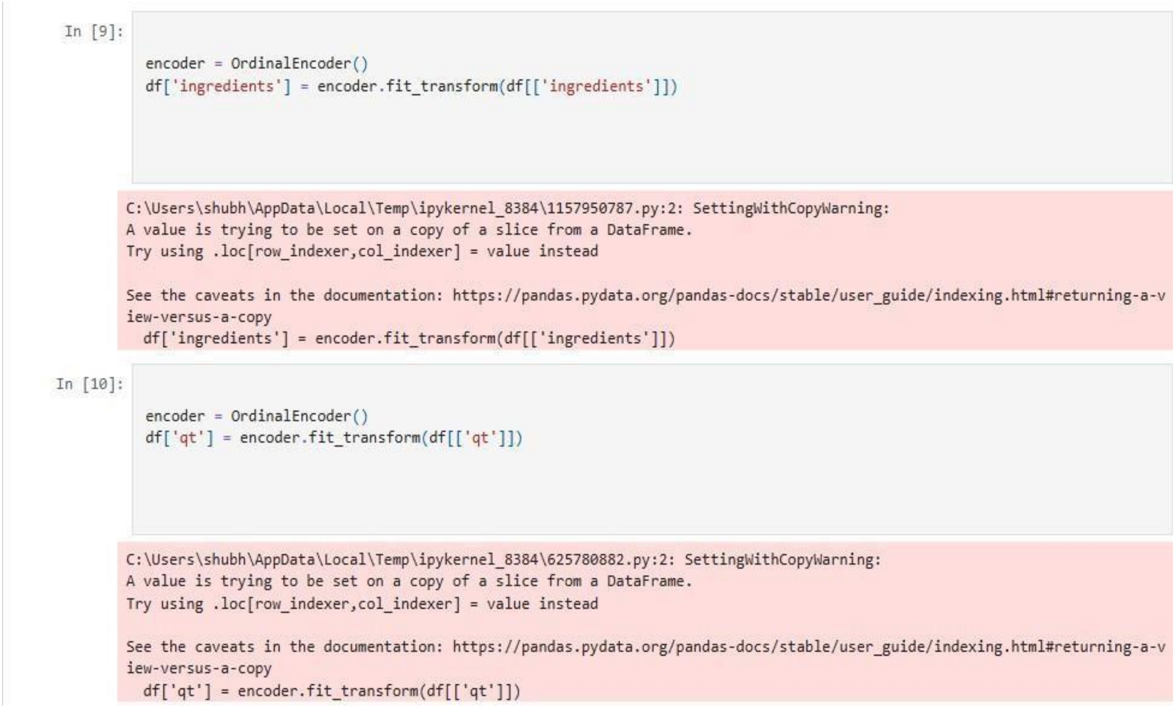


In [7]:
```
df.isnull().sum()
```

Out[7]:
```
id             0
name           0
ingredients    0
qt             0
wf             0
dtype: int64
```

In [8]:
```
from sklearn.preprocessing import OrdinalEncoder
encoder = OrdinalEncoder()
df['name'] = encoder.fit_transform(df[['name']])
```

```
C:\Users\shubh\AppData\Local\Temp\ipykernel_8384\3159346859.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-v
iew-versus-a-copy
  df['name'] = encoder.fit_transform(df[['name']])
```



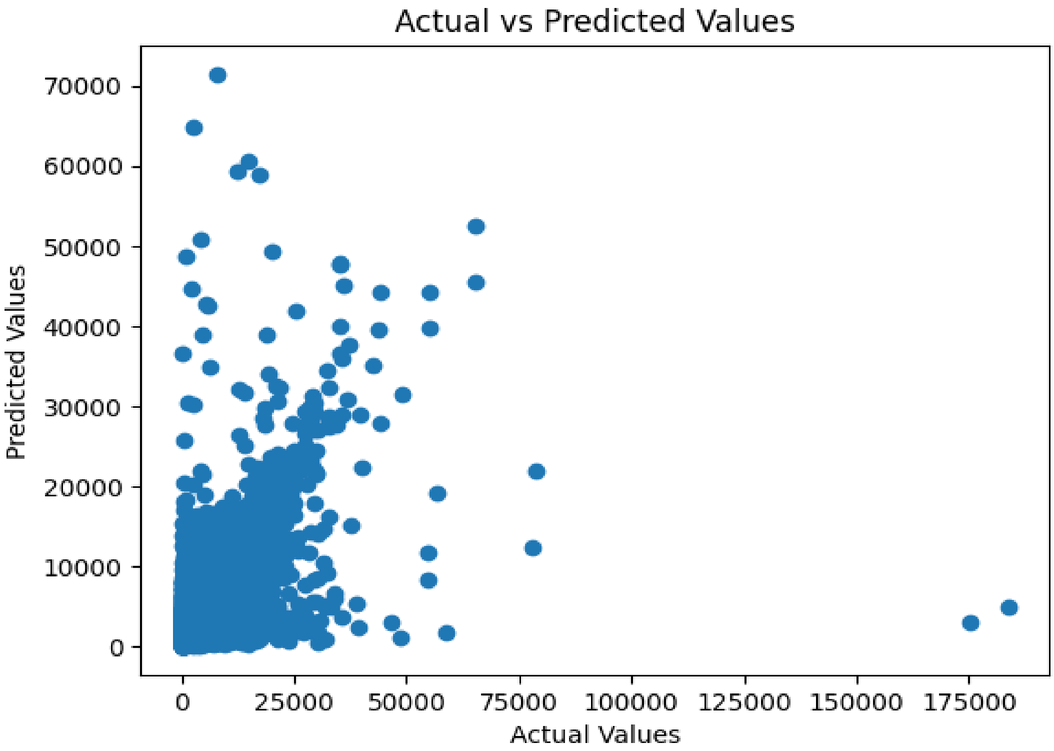
In [9]:
```
encoder = OrdinalEncoder()
df['ingredients'] = encoder.fit_transform(df[['ingredients']])
```

```
C:\Users\shubh\AppData\Local\Temp\ipykernel_8384\1157950787.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-v
iew-versus-a-copy
  df['ingredients'] = encoder.fit_transform(df[['ingredients']])
```

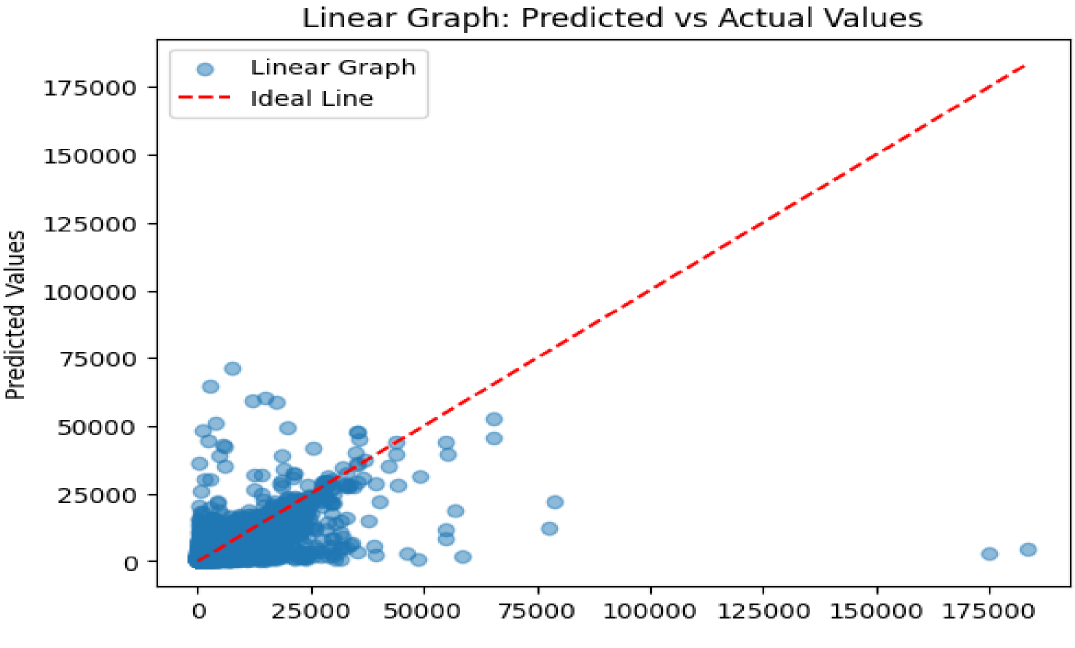
In [10]:
```
encoder = OrdinalEncoder()
df['qt'] = encoder.fit_transform(df[['qt']])
```

```
C:\Users\shubh\AppData\Local\Temp\ipykernel_8384\625780882.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-v
iew-versus-a-copy
  df['qt'] = encoder.fit_transform(df[['qt']])
```

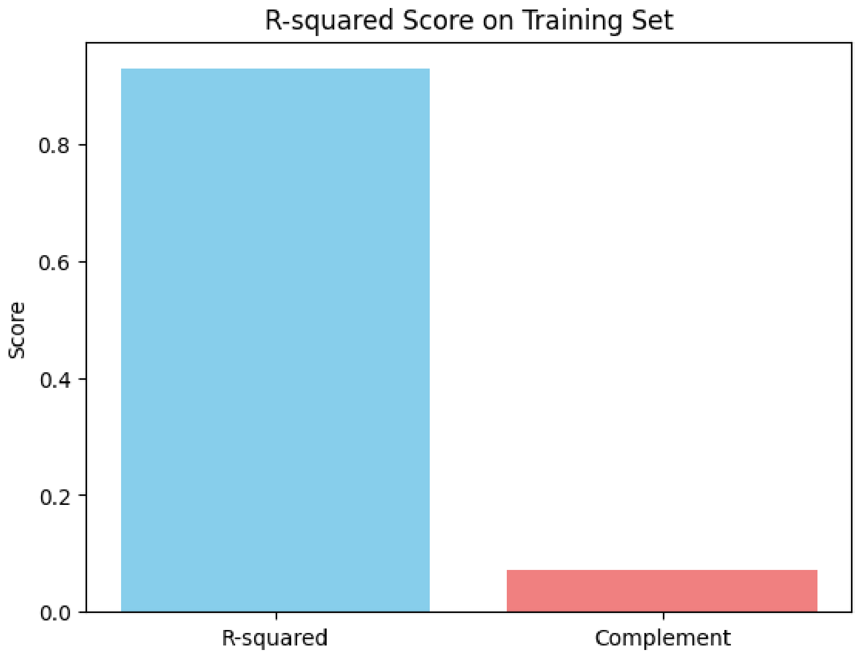
## SCATTER GRAPH:



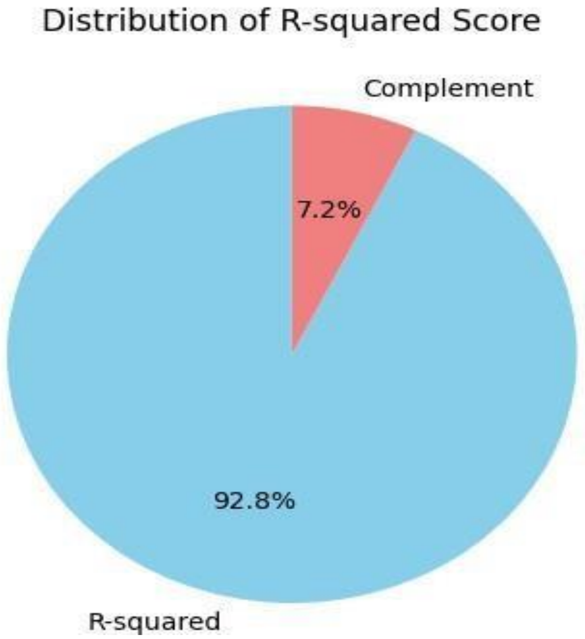
## LINEAR GRAPH:



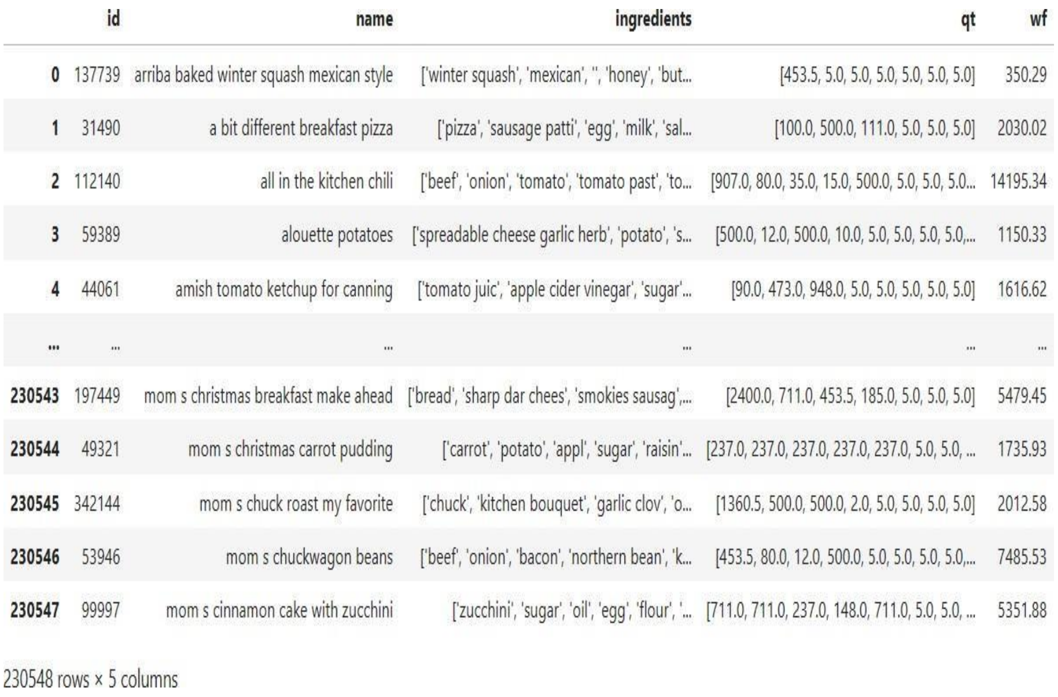
## BAR GRAPH:



## PIE CHART:

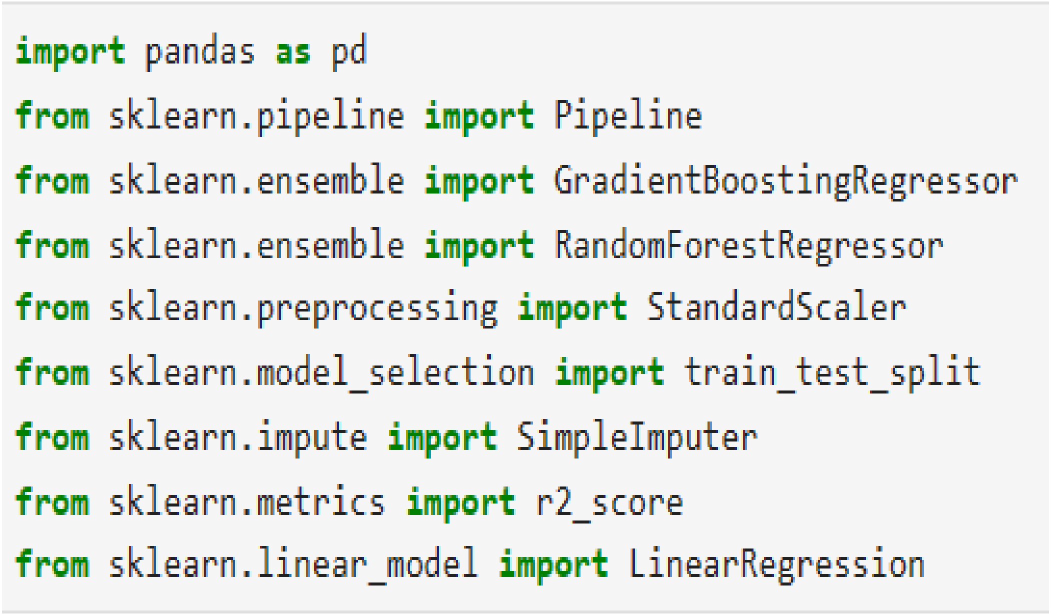


# MODEL WORKING FLOW:

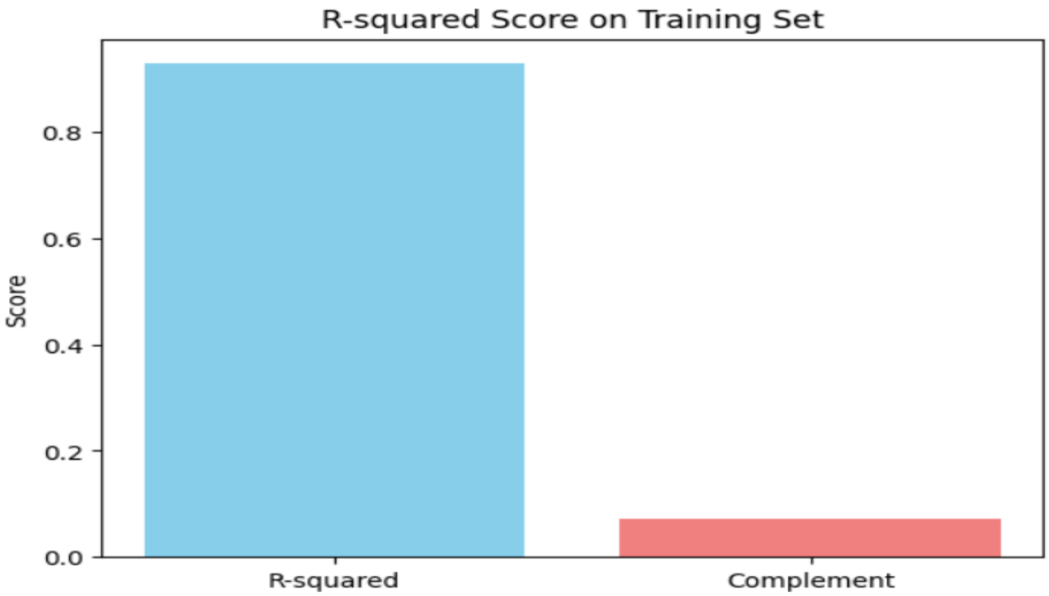


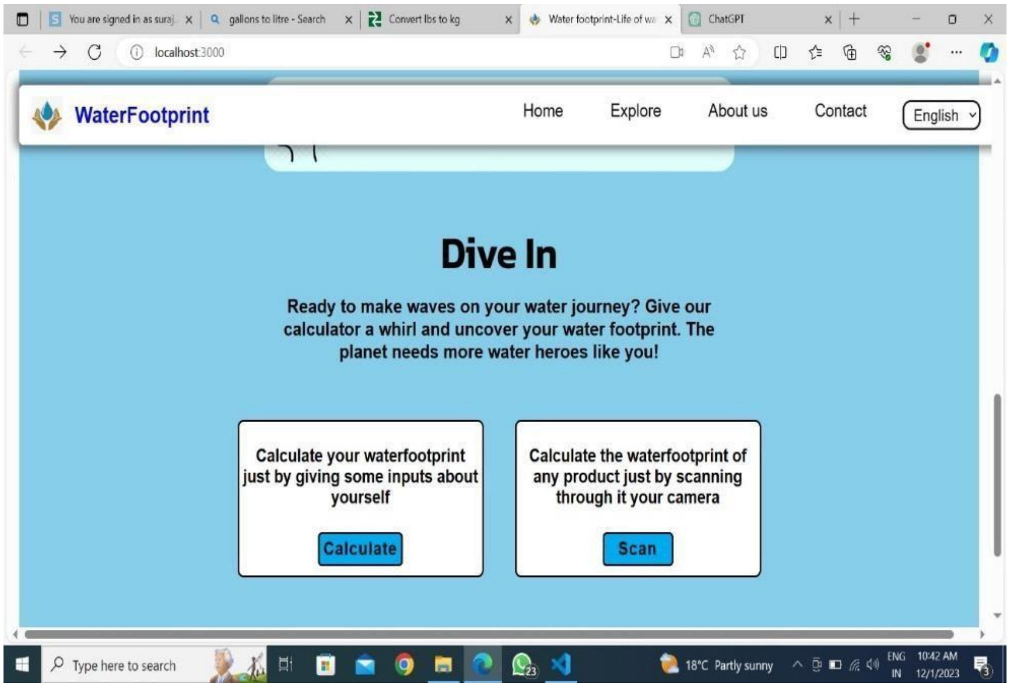
| | id | name | ingredients | qt | wf |
|---|---|---|---|---|---|
| **0** | 137739 | arriba baked winter squash mexican style | ['winter squash', 'mexican', '', 'honey', 'but... | [453.5, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0] | 350.29 |
| **1** | 31490 | a bit different breakfast pizza | ['pizza', 'sausage patti', 'egg', 'milk', 'sal... | [100.0, 500.0, 111.0, 5.0, 5.0, 5.0] | 2030.02 |
| **2** | 112140 | all in the kitchen chili | ['beef', 'onion', 'tomato', 'tomato past', 'to... | [907.0, 80.0, 35.0, 15.0, 500.0, 5.0, 5.0, 5.0... | 14195.34 |
| **3** | 59389 | alouette potatoes | ['spreadable cheese garlic herb', 'potato', 's... | [500.0, 12.0, 500.0, 10.0, 5.0, 5.0, 5.0, 5.0,... | 1150.33 |
| **4** | 44061 | amish tomato ketchup for canning | ['tomato juic', 'apple cider vinegar', 'sugar'... | [90.0, 473.0, 948.0, 5.0, 5.0, 5.0, 5.0, 5.0] | 1616.62 |
| **...** | ... | ... | ... | ... | ... |
| **230543** | 197449 | mom s christmas breakfast make ahead | ['bread', 'sharp dar chees', 'smokies sausag',... | [2400.0, 711.0, 453.5, 185.0, 5.0, 5.0, 5.0] | 5479.45 |
| **230544** | 49321 | mom s christmas carrot pudding | ['carrot', 'potato', 'appl', 'sugar', 'raisin'... | [237.0, 237.0, 237.0, 237.0, 237.0, 5.0, 5.0, ... | 1735.93 |
| **230545** | 342144 | mom s chuck roast my favorite | ['chuck', 'kitchen bouquet', 'garlic clov', 'o... | [1360.5, 500.0, 500.0, 2.0, 5.0, 5.0, 5.0, 5.0] | 2012.58 |
| **230546** | 53946 | mom s chuckwagon beans | ['beef', 'onion', 'bacon', 'northern bean', 'k... | [453.5, 80.0, 12.0, 500.0, 5.0, 5.0, 5.0, 5.0,... | 7485.53 |
| **230547** | 99997 | mom s cinnamon cake with zucchini | ['zucchini', 'sugar', 'oil', 'egg', 'flour', '... | [711.0, 711.0, 237.0, 148.0, 711.0, 5.0, 5.0, ... | 5351.88 |

230548 rows × 5 columns



```
import pandas as pd
from sklearn.pipeline import Pipeline
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
from sklearn.metrics import r2_score
from sklearn.linear_model import LinearRegression
```



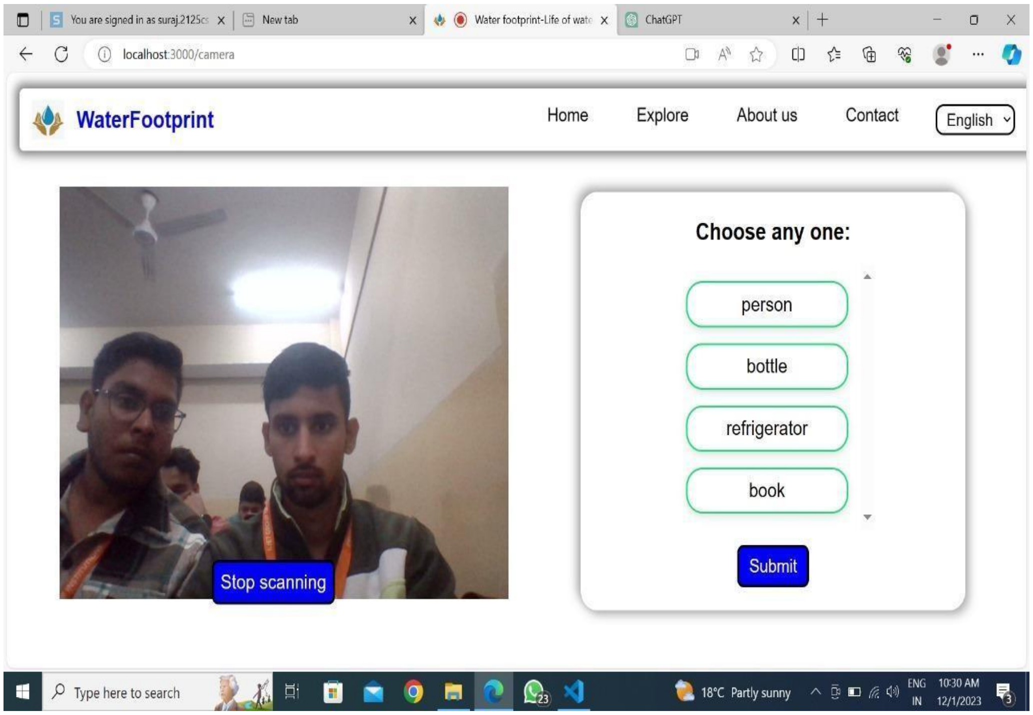


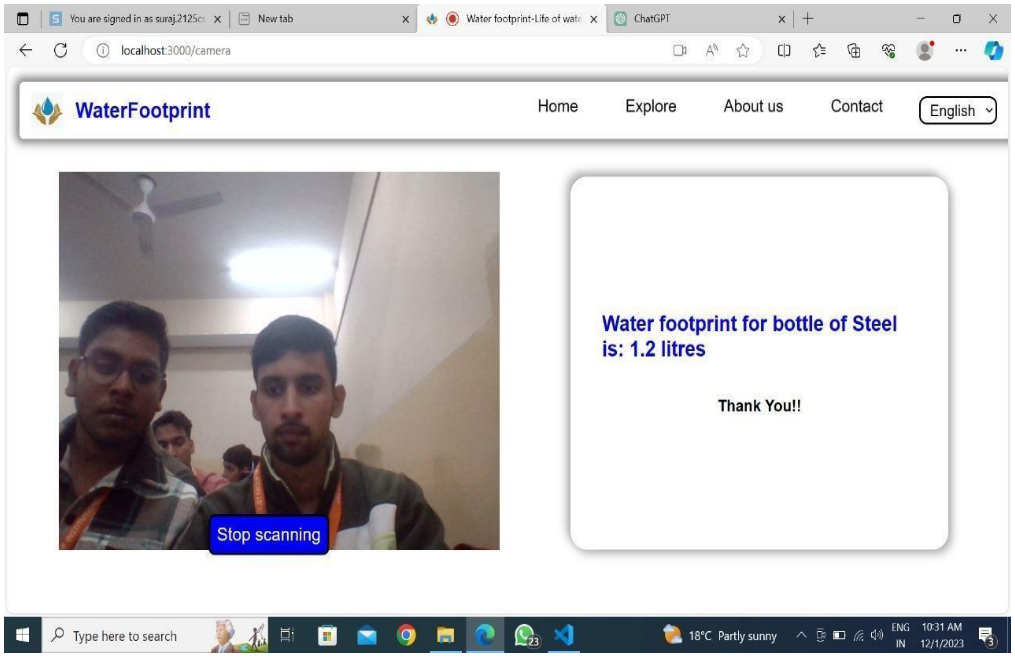
Dive In

Ready to make waves on your water journey? Give our calculator a whirl and uncover your water footprint. The planet needs more water heroes like you!

Calculate your waterfootprint just by giving some inputs about yourself

Calculate the waterfootprint of any product just by scanning through it your camera





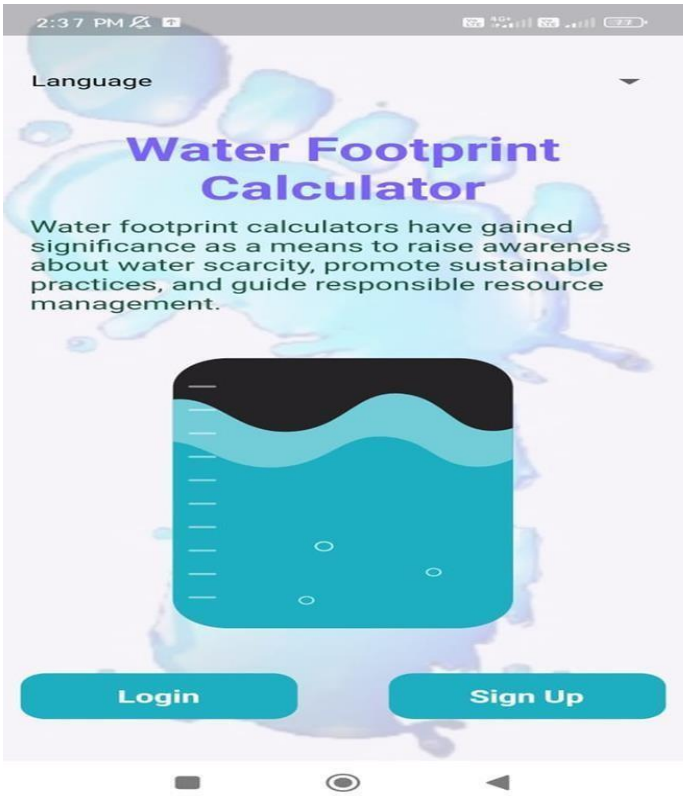
Water footprint for bottle of Steel is: 1.2 litres

Thank You!!

# CHAPTER 4

## 4 USER INTERFACE

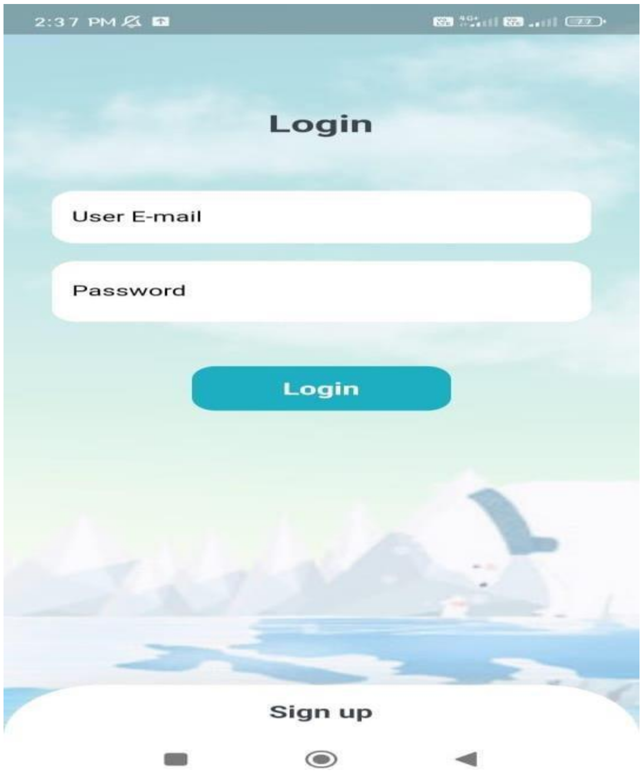
### LANDING PAGE:-

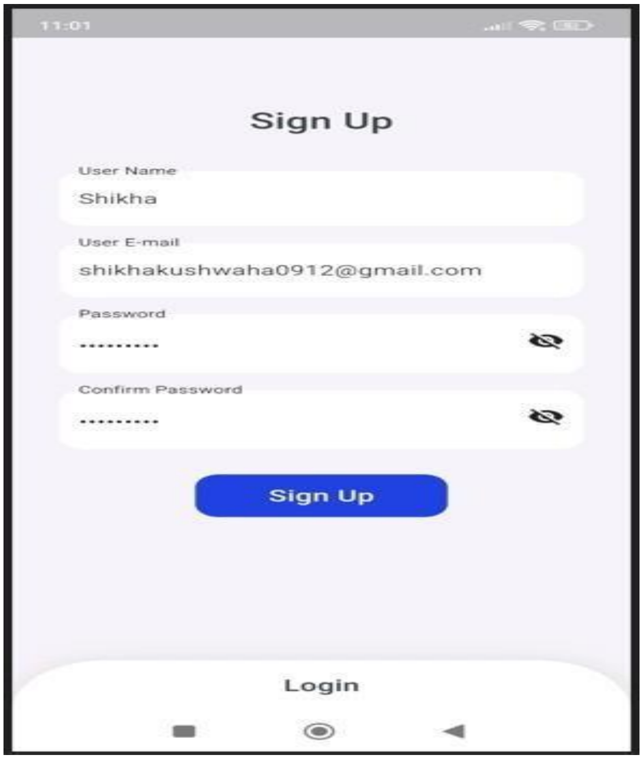


Water Footprint Calculator

Water footprint calculators have gained significance as a means to raise awareness about water scarcity, promote sustainable practices, and guide responsible resource management.

## LOGIN PAGE & SIGNUP PAGE:-





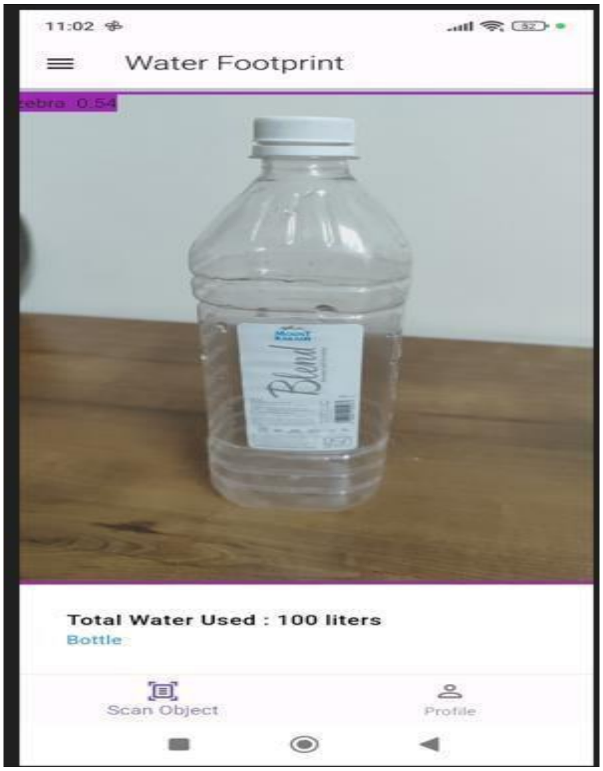
User Name
Shikha

User E-mail
shikhakushwaha0912@gmail.com

Password
•••••••••

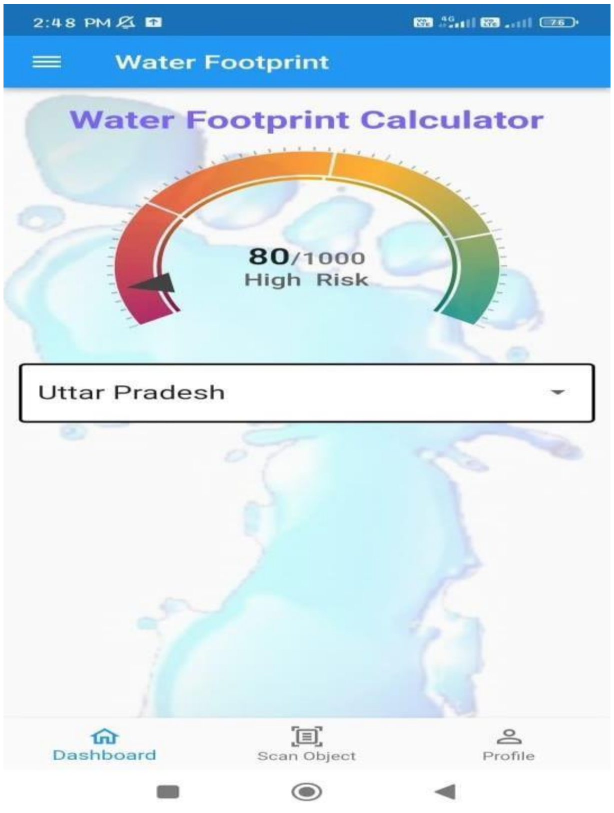
Confirm Password
•••••••••

## CALCULATION PAGE:-



Total Water Used : 100 liters

Bottle



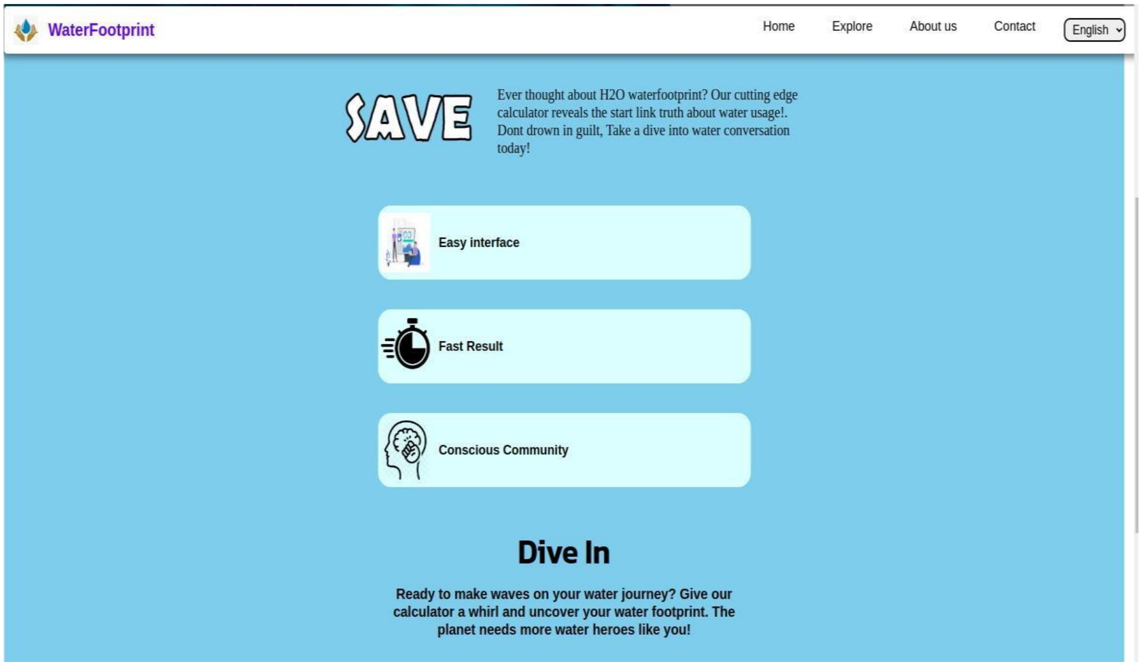
Water Footprint Calculator

80/1000

High Risk

Uttar Pradesh

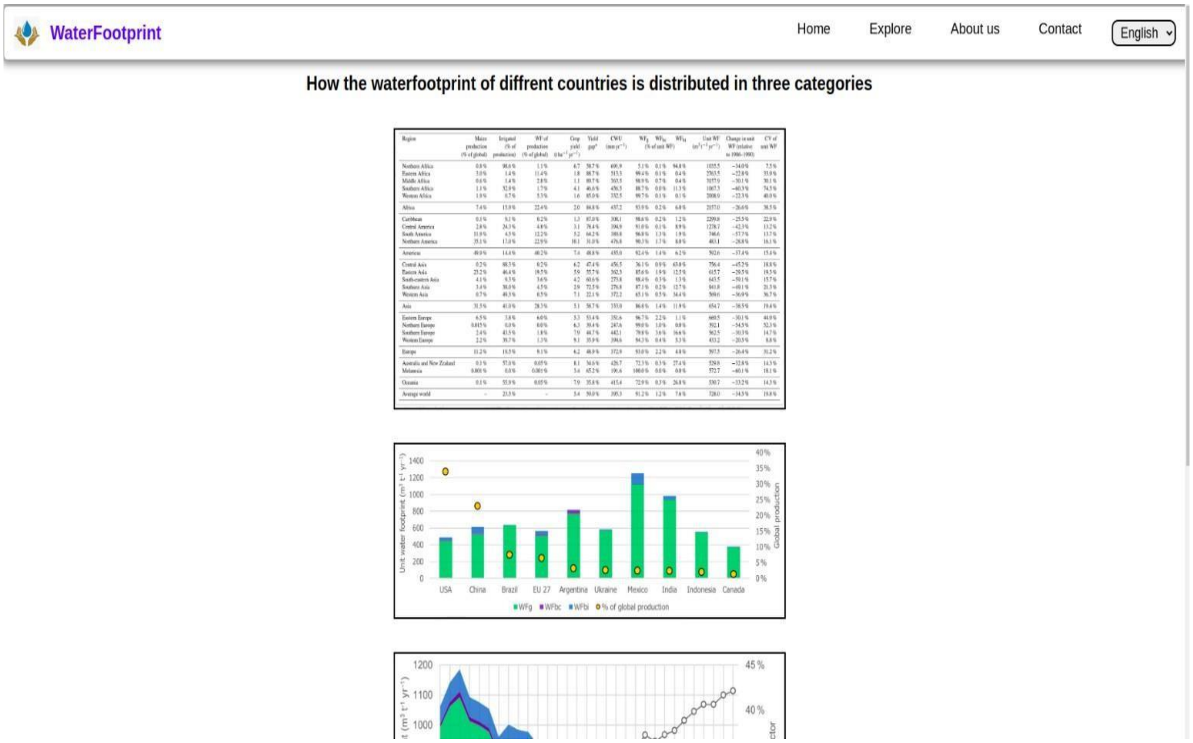
## 4.1 SCREEN SHOTS



Ever thought about H2O waterfootprint? Our cutting edge calculator reveals the start link truth about water usage!. Dont drown in guilt, Take a dive into water conversation today!

Dive In

Ready to make waves on your water journey? Give our calculator a whirl and uncover your water footprint. The planet needs more water heroes like you!



How the waterfootprint of diffrent countries is distributed in three categories

| Region | Maize production (% of global) | Irrigated (% of production) | WF of production (% of global) | Crop yield (t ha$^{-1}$ yr$^{-1}$) | Yield gap* | CWU (mm yr$^{-1}$) | WF$_g$ | WF$_{bc}$ | WF$_{bl}$ | Unit WF (m$^3$ t$^{-1}$ yr$^{-1}$) | Change in unit WF (relative to 1986–1990) | CV of unit WF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | (% of unit WF) | | | | | |
| Northern Africa | 0.8 % | 98.6 % | 1.1 % | 6.7 | 38.7 % | 691.9 | 5.1 % | 0.1 % | 94.8 % | 1035.5 | −34.0 % | 7.5 % |
| Eastern Africa | 3.0 % | 1.4 % | 11.4 % | 1.8 | 88.7 % | 513.3 | 99.4 % | 0.1 % | 0.4 % | 2763.5 | −22.8 % | 33.9 % |
| Middle Africa | 0.6 % | 1.4 % | 2.8 % | 1.1 | 89.7 % | 363.5 | 98.9 % | 0.7 % | 0.4 % | 3157.9 | −30.1 % | 30.1 % |
| Southern Africa | 1.1 % | 32.9 % | 1.7 % | 4.1 | 46.6 % | 456.5 | 88.7 % | 0.0 % | 11.3 % | 1067.3 | −60.3 % | 74.5 % |
| Western Africa | 1.9 % | 0.7 % | 5.3 % | 1.6 | 85.0 % | 332.5 | 99.7 % | 0.1 % | 0.1 % | 2008.9 | −22.3 % | 40.0 % |
| Africa | 7.4 % | 15.9 % | 22.4 % | 2.0 | 84.8 % | 437.2 | 93.9 % | 0.2 % | 6.0 % | 2157.0 | −26.6 % | 38.5 % |
| Caribbean | 0.1 % | 9.1 % | 0.2 % | 1.3 | 87.0 % | 308.1 | 98.6 % | 0.2 % | 1.2 % | 2299.8 | −25.5 % | 22.9 % |
| Central America | 2.8 % | 24.3 % | 4.8 % | 3.1 | 78.4 % | 394.9 | 91.0 % | 0.1 % | 8.9 % | 1278.7 | −42.3 % | 13.2 % |
| South America | 11.9 % | 4.5 % | 12.2 % | 5.2 | 64.2 % | 389.8 | 96.8 % | 1.3 % | 1.9 % | 746.6 | −57.7 % | 13.7 % |
| Northern America | 35.1 % | 17.0 % | 22.9 % | 10.1 | 31.0 % | 476.8 | 90.3 % | 1.7 % | 8.0 % | 483.1 | −28.8 % | 16.1 % |
| Americas | 49.9 % | 14.4 % | 40.2 % | 7.4 | 48.8 % | 435.0 | 92.4 % | 1.4 % | 6.2 % | 592.6 | −37.4 % | 15.4 % |
| Central Asia | 0.2 % | 88.3 % | 0.2 % | 6.2 | 47.4 % | 456.5 | 36.1 % | 0.9 % | 63.0 % | 756.4 | −45.2 % | 18.8 % |
| Eastern Asia | 23.2 % | 46.4 % | 19.5 % | 5.9 | 55.7 % | 362.3 | 85.6 % | 1.9 % | 12.5 % | 615.7 | −29.5 % | 19.3 % |
| South-eastern Asia | 4.1 % | 9.3 % | 3.6 % | 4.2 | 60.6 % | 273.8 | 98.4 % | 0.3 % | 1.3 % | 643.5 | −59.1 % | 15.7 % |
| Southern Asia | 3.4 % | 38.0 % | 4.5 % | 2.9 | 72.5 % | 276.8 | 87.1 % | 0.2 % | 12.7 % | 941.8 | −49.1 % | 21.3 % |
| Western Asia | 0.7 % | 49.3 % | 0.5 % | 7.1 | 22.1 % | 372.2 | 65.1 % | 0.5 % | 34.4 % | 569.6 | −36.9 % | 36.7 % |
| Asia | 31.5 % | 41.0 % | 28.3 % | 5.1 | 58.7 % | 333.0 | 86.8 % | 1.4 % | 11.9 % | 654.7 | −38.5 % | 19.4 % |
| Eastern Europe | 6.5 % | 3.8 % | 6.0 % | 3.3 | 53.4 % | 351.6 | 96.7 % | 2.2 % | 1.1 % | 669.5 | −30.1 % | 44.9 % |
| Northern Europe | 0.015 % | 0.0 % | 0.0 % | 6.3 | 39.4 % | 247.6 | 99.0 % | 1.0 % | 0.0 % | 392.1 | −54.5 % | 52.3 % |
| Southern Europe | 2.4 % | 43.5 % | 1.8 % | 7.9 | 44.7 % | 442.1 | 79.8 % | 3.6 % | 16.6 % | 562.5 | −30.3 % | 14.7 % |
| Western Europe | 2.2 % | 39.7 % | 1.3 % | 9.1 | 35.9 % | 394.6 | 94.3 % | 0.4 % | 5.3 % | 433.2 | −20.5 % | 8.8 % |
| Europe | 11.2 % | 19.5 % | 9.1 % | 6.2 | 48.9 % | 372.9 | 93.0 % | 2.2 % | 4.8 % | 597.5 | −26.4 % | 31.2 % |
| Australia and New Zealand | 0.3 % | 57.0 % | 0.05 % | 8.1 | 34.6 % | 426.7 | 72.3 % | 0.3 % | 27.4 % | 529.8 | −32.8 % | 14.3 % |
| Melanesia | 0.001 % | 0.0 % | 0.001 % | 3.4 | 65.2 % | 191.6 | 100.0 % | 0.0 % | 0.0 % | 572.7 | −60.1 % | 18.1 % |
| Oceania | 0.1 % | 55.9 % | 0.05 % | 7.9 | 35.8 % | 415.4 | 72.9 % | 0.3 % | 26.8 % | 530.7 | −33.2 % | 14.3 % |
| Average world | - | 23.3 % | - | 5.4 | 50.0 % | 395.3 | 91.2 % | 1.2 % | 7.6 % | 728.0 | −34.5 % | 19.8 % |

# CONCLUSION

The creation of a Water Footprint Calculator app builds an essential foundation to tackle water footprint increases and promote sustainable water resource governance. Digital technology combined with this novel solution enables users to obtain easy access to water usage data about different goods and services. Users can protect freshwater supplies by making educated choices while practicing sustainable behaviors through this program which leads to active conservation efforts against water depletion and reduction of water contamination.

The data accessibility features and water management support capabilities through the app demonstrate potential systems-changing capabilities that promote collaboration between stakeholders. The Water Footprint Calculator app maintains its value as a crucial component for our unified work to create a sustainable and resilient future while we deal with environmental and social and economic elements that determine water security. The combination of technology with individual and collective action will lead us toward a future where freshwater resources receive proper judicial and equitable management to benefit current and future generations

# FUTURE ENHANCEMENT

1. **AI-Powered Water Usage Prediction** – Implement machine learning models to predict future water consumption trends and provide recommendations for conservation.

2. **Blockchain-Based Water Transparency** – An implement of blockchain technology enables tamper- proof recording for H2O footprints which allows verifications of sustainability claims.

3. **Gamification for User Engagement** – Providing rewards together with leaderboards along with challenges should motivate people to reduce their water consumption.

4. **AR-Based Water Footprint Visualization** – Users should be able to activate an augmented reality function showing water usage data directly on their item scans for improved understanding.

5. **Integration with Government Databases** – Work with environmental agencies to acquire official water usage data that will enhance accuracy levels.

6. **Automated Water-Saving Tips** – The system should supply recommendations from AI models which adapt to a person's habits to minimize their water usage at home and in industrial processes.

7. **Multilingual Support for Global Reach** – The system should supply recommendations from AI models which adapt to a person's habits to minimize their water usage at home and in industrial processes.

8. **Community-Driven Reporting System** – Users should get the chance to indicate water-intensive business sectors and geographic areas for policymakers to obtain community-generated intelligence.

# REFERENCES

[1] M. K. Mehla, M. Kothari, P. K. Singh, S. R. Bhakar, and K. K. Yadav, “Water footprint assessment and its importance in Indian context: a meta-review,” *Water Supply*, vol. 23, no. 8, pp. 3113–3127, 2023, doi: 10.2166/ws.2023.174.

[2] H. A. Mekonnen MM, “The green, blue and grey water footprint of crops and derived crop products. Hydrology and earth system sciences.,” 2011.

[3] J. B. and M. F. Daniela Lovarelli, “Water Footprint of

crop productions: A review. Science of the total Environment,” 2016.

[4] G.-L. W. Mekonnen MM, “The Water Footprint of Global Food Production. Water,” 2020.

[5] Muratoglu A, “Assessment of wheat’s water footprint and virtual water trade: a case study for Turkey,” 2020, [Online]. Available: https://doi.org/10.1186/s13717- 020-0217-1

[6] P. Firda AA, “Water Footprint Assessment in the Agro- industry: A Case Study of Soy Sauce Production.,” 2nd Int. Conf. Energy, Environ. Inf. Syst. (ICENIS ), 2017, [Online]. Available: https://doi.org/10.1051/e3sconf/201831 08018.

[7] and G. M. Weerasooriya RR. Liyanage LPK, Rathnappriya RHK, Bandara WBMAC, Perera TANT, Jayasinghe GY, “Industrial water conservation by water footprint and sustainable development goals: a review.,” Env. Dev Sustain, 2021.

[8] A. Y. Hoekstra, “Water Footprint Assessment: Evolvement of a New Research Field,” Water Resour. Manag., vol. 31, no. 10, pp. 3061–3081, 2017, doi: 10.1007/s11269-017-1618-5.

[9] M. Egan, The Water Footprint Assessment Manual. Setting the Global Standard, vol. 31, no. 2. 2011. doi: 10.1080/0969160x.2011.593864.

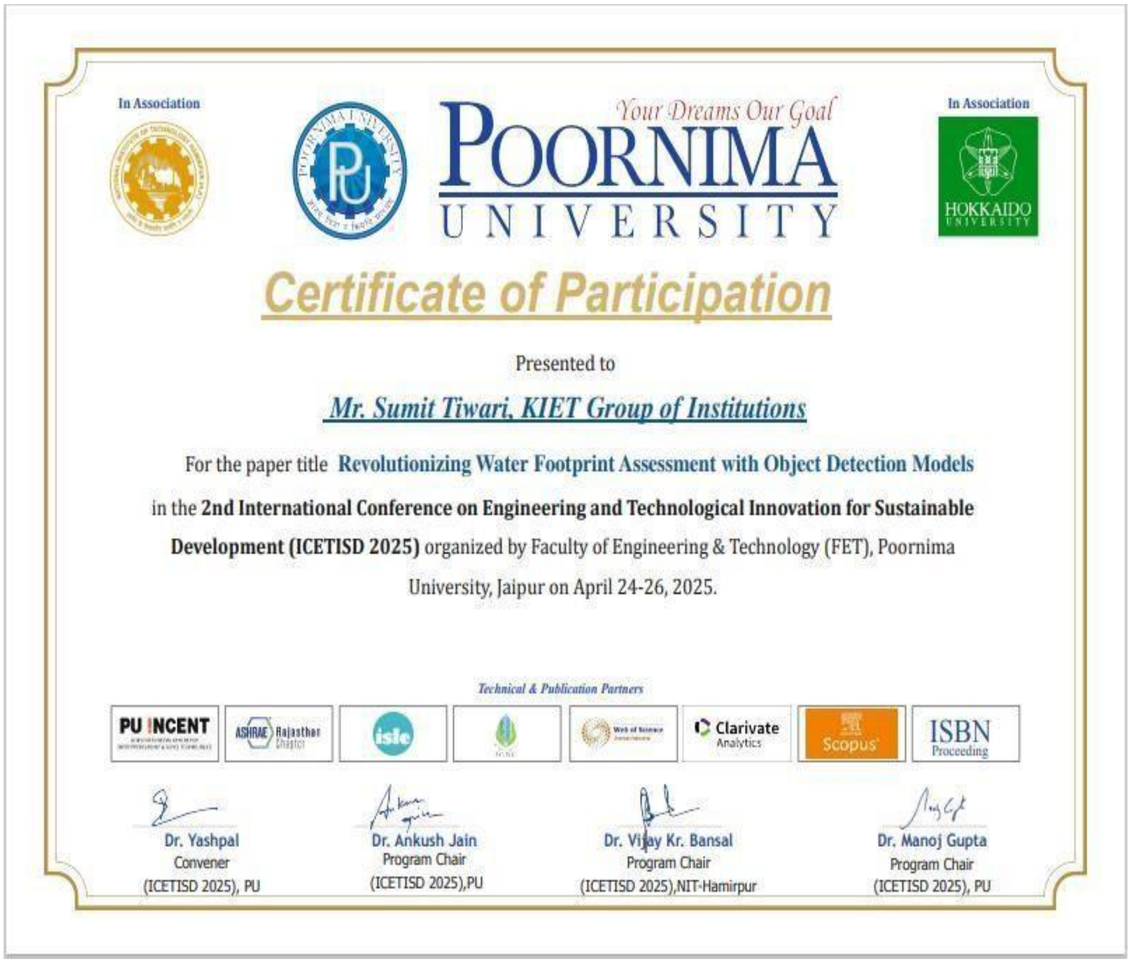
[10] R. J. Hogeboom, “The Water Footprint Concept and Water’s Grand Environmental Challenges,” One Earth, vol. 2, no. 3, pp. 218–222, 2020, doi: 10.1016/j.oneear.2020.02.010.

[11] A. D. Chukalla, M. S. Krol, and A. Y. Hoekstra, “Marginal cost curves for water footprint reduction in irrigated agriculture: Guiding a cost-effective reduction of crop water consumption to a permit or benchmark level,” Hydrol. Earth Syst. Sci., vol. 21, no. 7, pp. 3507–3524, 2017, doi: 10.5194/hess-21-3507-2017.

[12] https://www.waterfootprint.org/time-for- action/what- can-governments-do/.

[13] M. M. Aldaya, P. Martínez-Santos, and M. R. Llamas, “Incorporating the water footprint and virtual water into policy: Reflections from the Mancha Occidental region, Spain,” Water Resour. Manag., vol. 24, no. 5, pp. 941–958, 2010, doi: 10.1007/s11269-009-9480-8.

# RESEARCH PAPER CERTIFICATE:



In Association

Your Dreams Our Goal

POORNIMA UNIVERSITY

In Association

HOKKAIDO UNIVERSITY

## Certificate of Participation

Presented to

***Mr. Sumit Tiwari, KIET Group of Institutions***

For the paper title **Revolutionizing Water Footprint Assessment with Object Detection Models** in the **2nd International Conference on Engineering and Technological Innovation for Sustainable Development (ICETISD 2025)** organized by Faculty of Engineering & Technology (FET), Poornima University, Jaipur on April 24-26, 2025.

*Technical & Publication Partners*

PU INCENT | ASHRAE Rajasthan Chapter | isle | Web of Science | Clarivate Analytics | Scopus | ISBN Proceeding

| | | | |
|---|---|---|---|
| Dr. Yashpal<br>Convener<br>(ICETISD 2025), PU | Dr. Ankush Jain<br>Program Chair<br>(ICETISD 2025),PU | Dr. Vijay Kr. Bansal<br>Program Chair<br>(ICETISD 2025),NIT-Hamirpur | Dr. Manoj Gupta<br>Program Chair<br>(ICETISD 2025), PU |