A PROJECT REPORT-

WATER SUPPLY MANAGEMENT SYSTEM

SUBMITTED BY:
Mr.YUVRAJ BABASAHEB GAIKWAD

SUBJECT-
Programming and problem solving
Using C++
Under the guidance of
Pro .ISHWARI TIRSE

Department of Computer science and Engineering
Sanjivani Rural Education Society's
SANJIVANI UNIVERSITY
KOPARGAON-423603,DIST : AHMEDNAGAR
2024-2025

INDEX

INTRODUCTION -

Water Supply Management Systems

This manegement system covers essential operations like managing customers, tracking water consumption, and billing based on usage. Also you can extend this based on specific needs like adding a GUI, integrating databases, or adding more features.

Code:

```
#include <iostream>
using namespace std;
struct Customer {
    int id;
    string name;
    double consumption;
    double billAmount;
};

class WaterSupplySystem {
private:
    vector<Customer> customers;
    const double RATE_PER_LITER = 0.05;

public:
    void addCustomer(int id, const string &name) {
        Customer newCustomer = {id, name, 0.0, 0.0};
        customers.push_back(newCustomer);
        cout << "Customer " << name << " added successfully!\n";
    }

    void recordConsumption(int id, double liters) {
        for (auto &customer : customers) {
            if (customer.id == id) {
                customer.consumption += liters;
                cout << "Recorded " << liters << " liters for " << customer.name << ".\n";
                return;
            }
        }
        cout << "Customer ID not found.\n";
    }

    void generateBill(int id) {
```

```
        for (auto &customer : customers) {
            if (customer.id == id) {
                customer.billAmount = customer.consumption * RATE_PER_LITER;
                cout << "Bill generated for " << customer.name << ": Rs "
                     << fixed << setprecision(2) << customer.billAmount << "\n";
                return;
            }
        }
        cout << "Customer ID not found.\n";
    }

    void displayAllCustomers() {
        cout << "\n Customer List \n";
        cout << setw(5) << "ID" << setw(15) << "Name"
             << setw(15) << "Consumption" << setw(10) << "Bill\n";
        cout << "\n";

        for (const auto &customer : customers) {
            cout << setw(5) << customer.id
                 << setw(15) << customer.name
                 << setw(15) << customer.consumption << " L"
                 << setw(10) << fixed << setprecision(2) << customer.billAmount << "\n";
        }
    }
};

// Main function
int main() {
    WaterSupplySystem system;
    int choice, id;
    string name;
    double liters;

    do {
        cout << "\n Water Supply Management System\n";
        cout << "1. Add Customer\n";
        cout << "2. Record Water Consumption\n";
        cout << "3. Generate Bill\n";
        cout << "4. Display All Customers\n";
        cout << "5. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                cout << "Enter Customer ID: ";
                cin >> id;
                cout << "Enter Customer Name: ";
```

```
            cin.ignore();
            getline(cin, name);
            system.addCustomer(id, name);
            break;

        case 2:
            cout << "Enter Customer ID: ";
            cin >> id;
            cout << "Enter Water Consumption (in liters): ";
            cin >> liters;
            system.recordConsumption(id, liters);
            break;

        case 3:
            cout << "Enter Customer ID: ";
            cin >> id;
            system.generateBill(id);
            break;

        case 4:
            system.displayAllCustomers();
            break;

        case 5:
            cout << "Exiting system\n";
            break;

        default:
            cout << "Invalid choice! Please try again.\n";
      }
   } while (choice != 5);

   return 0;
}
```

Explanation

1. **Data Structure:**

A Customer struct is used to store the customer's ID, name, water consumption, and bill amount.

2. **Operations**:

Add Customer: Adds a new customer with ID and name.

Record Consumption: Records the amount of water consumed for a particular customer.

Generate Bill: Calculates the bill based on consumption and a fixed rate.

Display All Customers: Shows all customers with their details.

3. **User Input Handling**:

A simple menu-driven interface lets the user choose operations.

cin.ignore() ensures smooth input when switching between numeric and string input.

Output

Water Supply Management System
1. Add Customer
2. Record Water Consumption
3. Generate Bill
4. Display All Customers
5. Exit
Enter your choice: 1
Enter Customer ID: 101
Enter Customer Name: Alice
Customer Alice added successfully!

Enter your choice: 2
Enter Customer ID: 101
Enter Water Consumption (in liters): 500
Recorded 500 liters for Alice.

Enter your choice: 3
Enter Customer ID: 101
Bill generated for Alice: Rs 25.00

Enter your choice: 4

Customer List

ID	Name	Consumption	Bill
101	Alice	500 L	25.00

Conclusion

The Water Supply Management System implemented in C++ provides a simple yet effective solution for managing customer records, tracking water consumption, and generating bills. This system demonstrates key concepts of C++ such as class design, data structures, input handling, and control flow using a menu-driven interface. It helps water service providers efficiently organize data and streamline operations.

While this is a foundational system, it can be extended with advanced features like database connectivity, error handling, and real-time monitoring to enhance usability and scalability. Such a system plays a critical role in promoting sustainable water usage by offering transparent billing and tracking, ensuring that both service providers and customers benefit.