

Assignment Brief (RQF)

Higher National Certificate/Diploma in _____

Student Name/ID Number:	M.Vihanga Deshan Peiris / M19980106005
Unit Number and Title:	Unit: 13 Computing Research Project
Academic Year:	2019/2020
Unit Assessor:	Mrs. Maduwanthi Uthpala
Assignment Title:	Pearson Set Assignment: Research Project based on Digital Well being)
Issue Date:	20/3/2020
Submission Date:	22/10/2020
Internal Verifier Name:	Buddhini Samarakkody
Date:	

Submission Format:
The submission is in the form of an individual written report. This should be written in a concise, formal business style using single spacing and font size 12. You are required to make use of headings, paragraphs and subsections as appropriate, and all work must be supported with research and referenced using the Harvard referencing system. Please also provide a bibliography using the Harvard referencing system. The recommended word limit is 2,000–2,500 words, although you will not be penalised for exceeding the total word limit.
Unit Learning Outcomes:
LO1 Examine appropriate research methodologies and approaches as part of the research process. LO2 Conduct and analyse research relevant to a computing research project. LO3 Communicate the outcomes of a research project to identified stakeholders. LO4 Reflect on the application of research methodologies and concepts.
Assignment Brief and Guidance:
Scenario Digital Wellbeing is about fashioning and sustaining a healthy relationship with technology. As technology plays a big part in our lives we find ourselves spending an increasing amount of time online and on our devices. Our wellbeing is dependent upon our mental and physical health and thereby our digital wellbeing is influenced by our online interactions and the amount of time we spend on our devices. Whilst technology and the internet can simplify and enhance our lives

they can also be distracting, be a cause of anxiety, and make us feel upset. Being in control of technology enables us to use its full potential and gain all the benefits of it.

This unit will enable students to explore some of the areas of digital wellbeing from the standpoint of a prospective computing professional. It will provide the opportunity for students to investigate digital wellbeing within computing systems and explore the responsibilities and solutions to the problems presented..

The range of topics discussed could cover the following:

- How to find the balance towards a healthy relationship with devices?
- Are tech companies responsible for the health, safety and wellbeing of users?
- What tools and strategies can a company use to develop a system(s) that addresses digital wellbeing for users?
- What impact will future digital tech have on human wellbeing?

You should refer to these instructions as you complete work for this unit.

- Consider the development of a methodical and valid research proposal as the foundation for the project.
- Choose a topic of personal interest in a specialism. The topic chosen should allow a sufficient and suitable degree of research through the existence of adequate background materials.
- A good project proposal title should meet the following criteria:
 - The proposal is one that has an existing body of literature or source material that can be reviewed.
 - The proposal extends a current line of learning that will lend itself to further rigorous exploration.

Research Project Brief

1. Define your research problem or question. This can be stated as a research question, objectives or hypothesis. (LO2)
2. Provide a literature review giving the background and conceptualisation of your proposed area of study. This would provide existing knowledge and benchmarks by which your data can be judged. (LO2)
3. Consider and define your research methodology and research process. Demonstrate understanding of the pitfalls and limitations of the methods chosen and ethical issues that might arise. (LO1)
4. Draw points (1–3, above) together into a research proposal for agreement with your tutor. (LO2)
5. Conduct your research as outlined in your proposal. Keep track of your findings as you work. (LO2)
6. Carry out your research and analyse your findings in relation to your original research question. Draw conclusions. (LO2/LO3)
7. Communicate the outcomes of your research project to the identified audience. (LO3)
8. Reflect on the success of your research project and your performance at the end of the project with the inclusion of a project evaluation and recommendations.(LO4)

Assignment Feedback

Formative Feedback: Assessor to Student

Action Plan

Summative feedback

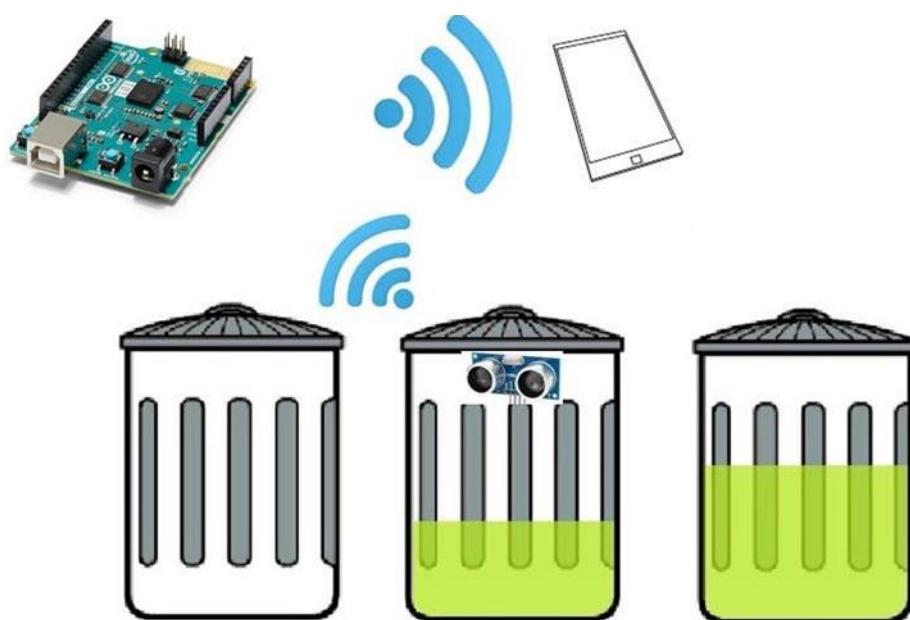
Feedback: Student to Assessor

Assessor Signature		Date	
Student Signature		Date	

End Of the Assignment

Prepared by: Higher Education Qualifications (HEQ) Approved by: HEQ, QDAM (HE)
Version 1.0 – November 2017
DCL 1 – Public (Unclassified)

Garbage Monitoring System for Public Places



By: M.Vihanga Deshan Peiris

Student Number: M19980106005

Supervisor Name: Nimesh Pollwaththage

DECLARATION

I do hereby declare that the work reported in this thesis was exclusively carried out by me under the supervision of Mr. Nimesh Pollwatthage , Lecturer at CINEC Campus, Malabe. It describes the result of my own independent research except where due reference has been made in the text. No part of this project thesis has been submitted earlier or concurrently for the same or any other degree.

M.Vihanga Deshan Peiris



Signature: Date: (22/10/2020)

This is to certify that this thesis is based on the work carried by Vihanga Deshan under my supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

Certified by: Supervisor: Mr. Nimesh Pollwatthage

Signature: Date: (22/10/2020)

Acknowledgment

Hereby I wish to thank all those who help me to success this project. And my precious thank goes to my supervisor lecturer, Mr.Nimesh Pollwatthage, who helped and encouraged me to success this Assignment. In addition, I really appreciate the support given by my batch mates and my family members, friends to success my project. Finally, I like to thank all the mentioned and non-mentioned well-wishers who helped me to success this assignment

Contents

Assignment Brief (RQF)	1
Higher National Certificate/Diploma in	1
Acknowledgment.....	7
List of figures	10
List of tables.....	13
CHAPTER 01.....	15
1. GENERAL INTRODUCTION	15
1.1 Background of the study	15
1.2 Statement of the problem	17
1.3 Aim and Objectives.....	17
1.4 Purpose of the Project.....	18
1.5 Scope of the Project	18
CHAPTER 02.....	19
2. LITERATURE REVIEW.....	19
2.1 Introduction.....	19
2.2 Feature of Garbage Monitoring and Automatic Waste Collection System in Public Places	21
2.3 Advantages of Garbage Monitoring and Automatic Waste Collection System in Public Places.....	21
2.4 Review of problems and challenges	22
CHAPTER 03.....	23
3. System analysis and Design.....	23
3.1 introduction.....	23
3.2 information gathering	23
3.3 Feasibility Analysis.....	27
3.5 Functional Requirement.....	29
3.6 Non-Functional Requirement	29
3.7 Design and implementation.....	29
3.8 Methodology	36
3.9 Advantages and Disadvantages of Waterfall Development Methodology	40
3.10 Work Plan	41
3.11 Diagrams	42

3.12 System Development.....	43
3.13 Final Testing	61
CHAPTER 04.....	63
4. System Implementation and Documentation.....	63
4.1 System implementation	63
CHAPTER 05.....	85
5.1 Summary	85
5.2 Conclusion	85
5.3 Future work	85
References.....	86

List of figures

Figure 1 Smart garbage management system Questionnaire 1	24
Figure 2 Smart garbage management system Questionnaire 2	25
Figure 3 Smart garbage management system Questionnaire 3	25
Figure 4 Smart garbage management system Questionnaire 4	26
Figure 5 Smart garbage management system Questionnaire 5	26
Figure 6 Financial Feasibility Diagram	28
Figure 7 Project diagram	30
Figure 8 VNC software	30
Figure 9 Notepad ++	31
Figure 10 Moba Xterm	31
Figure 11 Windows Server 2012	32
Figure 12 Pm2 software	32
Figure 13 Garbage Container	33
Figure 14 Ultrasonic Sensor	33
Figure 15 GSM module	34
Figure 16 Raspberry Pi board	35
Figure 17 Battery Pack	35
Figure 18 Parallel Development methodology	36
Figure 19 V-model methodology	37
Figure 20 Iterative development methodology	37
Figure 21 Agile Development Methodology	38
Figure 22 Prototype development methodology	38
Figure 23 Waterfall development methodology	39
Figure 24 GSM and Arduino connection	43
Figure 25 Pin define coding 1	44
Figure 26 Pin define coding 2	45
Figure 27 Send message coding	45
Figure 28 Raspberry pi board and ultra-sonic sensor connection diagram	46
Figure 29 Raspberry pi pin define code	46
Figure 30 Server and Client Connection Code –	47
Figure 31 ultrasound transmitter and receiver	47
Figure 32 Garbage bin	48
Figure 33 Distance Calculation Code	49
Figure 34 Bin collection data	49
Figure 35 server side Jason file	50
Figure 36 index.js code 1	50
Figure 37 index.js code 2	51
Figure 38 index.js code 3	51
Figure 39 index.js code 4	52
Figure 40 index.js code 5	52
Figure 41 index.js code 6	53
Figure 42 Index.js code 7	53
Figure 43 index.js code 8	53
Figure 44 index.js code 9	54
Figure 45 index.js code 10	54
Figure 46 index.js code 11	54

Figure 47 index.js code 12	55
Figure 48 style sheet code 1	55
Figure 49 style sheet code 2	56
Figure 50 style sheet code 3	56
Figure 51 style sheet code 4	57
Figure 52 style sheet code 5	57
Figure 53 style sheet code 6	58
Figure 54 index page layout	58
Figure 55 Dashboard layout	58
Figure 56 Server side data.ini file	59
Figure 57 Mobile app internet permission	59
Figure 58 Web view code 1	59
Figure 59 Web view code 2	59
Figure 60 Web view code 3	60
Figure 61 Activity Splash layout	60
Figure 62 Activity Splash layout Activity	60
Figure 63 Testing 1	61
Figure 64 Testing 2	61
Figure 65 Testing 3	62
Figure 66 Cloud server installation 1	63
Figure 67 Cloud server installation 2	63
Figure 68 Cloud server installation 3	64
Figure 69 Cloud server installation 4	64
Figure 70 Cloud server installation 5	65
Figure 71 Cloud server installation 6	65
Figure 72 Cloud server installation 7	66
Figure 73 Cloud server installation 8	66
Figure 74 Cloud server installation 9	66
Figure 75 Create VM RDP Access Key 1	67
Figure 76 Create VM RDP Access Key 2	67
Figure 77 Create VM RDP Access Key 1	68
Figure 78 Create VM RDP Access Key 4	68
Figure 79 Access server using RDP Client 1	69
Figure 80 Access server using RDP Client 2	69
Figure 81 Windows server1	70
Figure 82 Windows server2	70
Figure 83 Node JS installation 1	71
Figure 84 Node JS installation 2	71
Figure 85 Node JS installation 3	72
Figure 86 Node JS installation 1	72
Figure 87 Verify Installation	73
Figure 88 Upload server files and start the server 1	73
Figure 89 Upload server files and start the server 2	74
Figure 90 Upload server files and start the server 3	74
Figure 91 Upload server files and start the server 4	75
Figure 92 Client side Installation 1	76
Figure 93 Client side Installation 2	76
Figure 94 Verify the Node JS Installation	77

Figure 95 Start the node JS app	77
Figure 96 Start the node JS app1	78
Figure 97 Start the node JS app 2	79
Figure 98 Start the node JS app 3	79
Figure 99 Start the node JS app 4	80
Figure 100 Start the node JS app 5	80
Figure 101 Start the node JS app	81
Figure 102 Installation PM2 JS module 1	81
Figure 103 Installation PM2 JS module 2	82
Figure 104 Installation PM2 JS module 3	82
Figure 105 Installation PM2 JS module 4	83
Figure 106 View Pm2 Process log 1	83
Figure 107 View Pm2 Process log 2	84

List of tables

Table 1 - Cost.....	27
Table 2 Advantages and Disadvantages of Waterfall Development Methodology	40
Table 3 Gantt chart.....	41

Abstract

As of now the population of Sri Lanka is 21,391,770 Garbage management is a big challenge in the Sri Lankan Public areas and there are several problems related to garbage management. There are not enough trash bins available to the public and most of the bins available fill up faster than ever. Many of these garbage bins overflow before collection. Garbage collection from these containers was delayed, as information on the fill level in the containers may not be available. The lack of information on empty containers allows the general public to dump trash on and near the street. The overflowing recycle bin becomes a messy street with an unpleasant smell. Sometimes biomedical or hazardous waste is also dumped in the trash on the street. This disastrous situation places great pressure on waste management.

In this situation, we want to implement solution for monitoring garbage bins in public places. We can provide simple solution using IOT (Internet of things) for this issue. Simply this system can be created so that any responsible person will automatically notify us when garbage is full. Also he can monitor using web portal garbage bins trash levels. Then the person concerned can request to dispose of the garbage.

CHAPTER 01

1. GENERAL INTRODUCTION

1.1 Background of the study

Due to the rapid decrease in the natural resources available and the increased costs of production of raw materials, the European Union strategy suggests replacing the natural resources needed Production of products from recycled parts and materials. Technological advancement also speeds up replacement of equipment that generates increased electrical waste and electronic appliances Waste Electrical and Electronic Equipment Directive (WEEE). The recovery and reuse of renewable resources are the key goals of sustainable development for our society as part of the overall effort to minimize natural resource exploitation that leads to a continuous depletion of environmental quality. Looking at the history of separate collection and recycling of waste, one may find that many methods were used in time, but only when a shortage of resources arose did the government and local authorities take action to create rules for separate collection of waste. During the Second World War, the warring countries began campaigns by strongly encouraging individuals to collect and recycle materials like paper, cans, metal and even kitchen fat the complexity of environmental concerns, including waste disposal and the need for recycling and reuse, became increasingly conscious.

However, separate waste management was implemented in legislation only at the beginning of the 1990s, first in Germany and then in EU countries. In this way "the life cycle of the commodity could be closed by making the suppliers accountable for their goods cradle to grave." Orange, blue, brown, yellow and gray bins were available for separate waste collection and changes in the quality and quantity of waste collected for recycling were recorded in a few years. The next major move in waste was the introduction of reverse vending machines (RVM) onto the market. Administration thus the residents were inspired to carry back plastic bottles, glass receptacles and aluminium cans to collect coupons that could be used in supermarkets in return.

Nowadays, hundreds of thousands of RVMs developed by Tomra, Envipco, RVM Systems, Diebold Nixdorf and other RVM manufacturers are used worldwide. The new RVMs include advanced vision systems capable of instantly detecting barcodes and safety marks (allowing users to easily insert containers in a continuous flow) and distinguishing between thousands of different receptors quickly and accurately. New solutions for intelligent waste management are being used recently in combination with the development and deployment of smart city technologies and networks. The smart waste collection technology market is still in an early phase but the use of Smart Bins for waste collection has now begun. Smart storage bins with ultrasonic sensors attached to them can be placed in various areas of the city to measure the empty space. A sensor gateway designed around a wide range.

Area network protocol and cloud systems for data collection, analysis and visualization should be used Theoretical solutions and preliminary findings concerning IoT waste management technologies can be found in the dedicated literature. Zanella and Vangelista present an IoT stage answer for a brilliant city, with a Padova city contextual investigation, where the proposed framework should gather fascinating ecological boundaries, for example, CO level, air temperature

and moistness, vibrations and clamor Zygiaris is introducing a savvy city reference model to be utilized by keen city organizers.

So, characterizing the connection between the different assistance situated models Service Oriented Architectures (SOAs) from every space to be incorporated several papers depict the IoT highlights and attributes yet just a couple of them incorporate outcomes. For instance, in Romania, a vacuum squander assortment framework for the verifiable downtown area of Sibiu was created dependent on a few boundaries: squander volume - which is a wide factor as a result of occasional occasions; organization of waste; source focus; current foundation; good ways from sources and the degree of open worry In Sweden, for instance, the computerized vacuum squander assortment framework can gather as it were four kinds of waste: general waste, natural waste, recyclable paper and recyclable cardboard This framework can't gather Waste Electrical and Electronic Equipment Directive (WEEE)s, batteries, lights and fluorescent lights, huge sheets of cardboard, glass beneficiaries and so forth. The vacuum framework is an inflexible one on the grounds that the funnels must to be masterminded to ensure access without delving into different links and channels, additionally depicted as the "spaghetti impact". Additionally, changing a current system is troublesome and expensive. Arranging and building the system consequently should be done at the absolute first phases of a local location. It is difficult to execute this framework for a whole enormous city like Paris, London, Amsterdam and so forth. Notwithstanding, a completely computerized coordinated waste assortment framework (ready to independently gather various sorts of waste: plastic beneficiaries, glass beneficiaries, aluminium jars, Waste Electrical and Electronic Equipment Directive (WEEE)s, paper, cardboard, batteries, lights and fluorescent lights and so forth.) and a brilliant city stage committed for this sort of framework was not distinguished by the creators in the particular writing and in the examined US and EU licenses. The paper is composed as follows: in part two, insights regarding our exploration venture are introduced along with the mechanized waste assortment framework arrangement came about because of our task.

In chapter the platform solutions for the integration of the automated waste collection system into the smart city infrastructure are presented, while chapter four discusses the conclusion of the research and future work.

- Materials and Methods
- The Smart Collect Project

Romania has a waste collection level well below the EU average and is expected to reach a target of 50 per cent in preparation for waste reuse and recycling by 2020.

Alongside the open's low mindfulness with respect to the need for specific waste assortment, the absence of waste assortment frameworks speaks to a deterrent in accomplishing the objective. Presently, human administrators are as yet utilized in discrete waste gathering frameworks for performing assignments related with Paper and cardboard arranging and gauging. Additionally, the administrators transport the paper or cardboard to the baler, works the parcel press and the came about bunches are put away, and soon;

Emptying holders of plastic and glass beneficiaries or aluminium jars; Large and little Waste Electrical and Electronic Equipment Directive (WEEE) s ID, gauging, transport, move and capacity. To expand the degree of waste assortment, it is important to get a completely robotized squander gathering framework. The different waste assortment arrangements right now executed in Romania are not completely computerized or fit to be coordinated in the keen city foundation.

Our research project began in 2016, in this context. The main objectives of the project are to modernize and automate the existing waste collection system developed by our project partner in the past, and to obtain a fully Automated Waste Collection System (AWCS). The project's first objective is to increase the system's use, improve its productivity and the waste storage capacity that it collects.

1.2 Statement of the problem

This system does is always provide a realistic indication of the level of waste in the trash. You can use that data to optimize waste collection paths and ultimately reduce fuel consumption. Garbage collectors can schedule a daily / weekly selection schedule. Initial Research into sources of information we can see garbage bins are full and the dirt falling to the ground in many places in Sri Lanka. Because of this, we can see the potential for many other countries to manage this garbage. Main inability of current recycle bin collection the system is as follows. The first step is to monitor the accumulation of garbage in a container by using a web portal and to create a system that will automatically close the container once the trash has been collected. Further, the project will enable the person in charge of the area where the trash belongs to be automatically notified when the trash is filled. This web portal can also check all systems installed in the area. The system works as follows: Uses HC-SR04 ultrasonic sensor to detect instantaneous depth or level of trash. This information is then received and processed by the microcontroller, which is mounted on the Raspberry Pi board. Next, based on the processed output, determine the instantaneous level of the recycle bin. Our developed system has four predefined threshold levels. The first level ranges from 0 to 25%, the second level up to 50%, the third level up to 75%, and the fourth level up to 90%. Trash bin depth.

1.3 Aim and Objectives

Aim - The aim of this project is to design Garbage Monitoring System to ease the problems that peoples face while managing waste. In this system, we can monitor entire garbage bin using web portal.

Objective - The objective of this project is to design and implement a cost effective, reliable & efficient Garbage Monitoring System for Public Places. And Improvement of public hygiene conditions. In addition, to create a healthier environment for us.

1.4 Purpose of the Project

The Paper IoT-based waste monitoring system is a very innovative system that helps clean cities. A web page makes it easy for administrators to know the bin storage capacity. An ultrasonic sensor is placed in the trash bin to detect the trash phase and balance the trash intensity with this system. The system sends data using a Raspberry Pi Backend server. Users can use a web page to locate the garbage bin area. An image view of the trash bins is provided on the web page and highlights the accumulated trash in colour to indicate the trash collection phase. This method helps keep the city clean by providing a graphical image of the garbage through the IoT Web development platform and notifying about the garbage level. This system only can be identify the garbage bin trash levels and all garbage bins can view their trash levels using single portal.

1.5 Scope of the Project

The product of the project being produced is a smart garbage system that automates the main operations of the garbage's. The first subsystem is the identifying garbage and collecting them in the system to track authorized the second subsystem is the identifying data separate and send to the web server backend.

The third subsystem is the full of the bin on the led indicator light. Other system is full of the bin send to the message for the area garbage collector or authorized person to notify the garbage bin level. This is the main area that the user needs.

CHAPTER 02

2. LITERATURE REVIEW

2.1 Introduction

In this part we investigate how to assemble venture data for the writing audit stage, a foundation study that incorporates scholarly analysis, key exploration and examination holes. Likewise, the foundation concentrate here is that external exhortation and data must be acquired to actualize this task. Additionally, as per the writing survey of this venture, significant information can be gathered from a few sites and key data can be acquired mostly from the systems administration subjects of the HND program. Simultaneously, my Project Supervisor and Project Plan Implementation and Evaluation Lecturer gave significant guidance and data on this. Likewise the significant examination here can be found in various books with various writers in various translations. Here are a few meanings of a portion of the joined books.

So, we can see garbage bins are full and the dirt falling to the ground in many places in Sri Lanka. Because of this, we can see the potential for many other countries to manage this garbage. Main inability of current recycle bin collection the system is as follows.

- There is no estimation to the amount of solid waste present inside the bin and the surrounding area due to the scattering of waste.
- Lack of information about the collecting time and area
- No any idea about how much garbage is collected in the area per day?

It is at that points, started gathering requirements for Garbage Monitoring and Automatic Waste Collection System. There are many systems to eliminate these problems Developed. Some of these are summarized below.

“Hong, I., Park, S., Lee, B., Lee, J., Jeong, D. and Park, implement IoT-Based Smart Garbage System For Efficient Food Waste Management in 2004. In the research they did how to reduce food waste to reduce, an IoT-based smart waste system was proposed amount of food waste. Battery-based smartphones in this system garbage bins share information with each other wireless Networks, and adds a router and server and analyzing information for service delivery and there are various IoT techniques for user convenience. This project is Solution for IoTBased Smart Garbage System for Efficient Food Waste Management.”

“Priya B. K., T. Lavanya, V. Samyukta Reddy, Yarlagadda Pravallika peoples do the project name called “Bin That Think’s”. They have done this project too smart waste management system. In the research they did radio frequency identification (RFID) and weight sensor to use design a smart waste management system to reduce waste management costs and facilitate waste detection and automate the weighting process. This project is included in “The International Journal of Science and Technoledge. Vol.3, pp 218-223, June 2015”

“Waikhom Reshma, RamKumar Sundaram, M. Rajeev Kumar implement Sensor Unit for Waste Management. In the research they did, an electronic system design is proposed to provide a solution for an inappropriate waste treatment system. Biosensors and weight sensors are used in conjunction with height sensors to detect the extent of trash in the trash and the intensity of contamination caused by unwanted toxic gases from the trash. These sensors are then sent to the controller, which helps the GSM module send notifications about the status of the recycle bin to authorities. The main purpose of this paper is to provide a solution for the proper disposal of waste. An effective waste management system is achieved using the sensors in this paper. Sensor units are used for sensing, microcontrollers are used for control, GSM modules are used for communication, and solar energy is used for power required by the system. This Project is included in A Better Method for Frequent Data Updating System| International Conference on Science, Engineering and Management Research. 2014”

2.2 Feature of Garbage Monitoring and Automatic Waste Collection System in Public Places

- The smart, sensor based dustbin will judge the level of waste in it and send the message directly to the municipal corporation. (Prototype one only can send the one person, we can integrate the SMS gateway and then we can send the multiple corporations)
- It can sense all the type of waste material either it is in the form of solid or liquid. (We need to manually add the waste material type)
- The system is simple.
- If there is any problem with any equipment in the future, that part is easily replaceable with new one without any difficulty and delay.
- Authorized person can monitor garbage bin any time any ware using mobile app or web login.

2.3 Advantages of Garbage Monitoring and Automatic Waste Collection System in Public Places.

- Less time and fuel consumption as trucks only go to loaded containers.
- This system cannot identify the waste type (eg: Paper, Polythene, etc)
- Deployed in the city and service operator. Therefore the focus is only on adding the path-based filling level in the containers.
- The sensors installed in the containers provide real-time information about the filling level.
- This information will help you decide when and where to prioritize collection.
- In this way, both service providers and citizens will benefit from an optimized system, which will reduce major cost savings and urban pollution.
- Reduces service infrastructure (truck, container), operating (fuel) and maintenance costs by up to 30%.
- The application of this technology to the city optimizes management, resources and costs and makes it a "smart city".
- Information about Collections Historical information helps to deploy containers according to the real needs of the city, thus reducing the number of containers that clutter the road and increasing the amount of public parking space.
- It keeps the environment clean and green, pollution free, and emphasizes a healthy environment, keeping cities more beautiful.

2.4 Review of problems and challenges

Waste management in cities is often the most expensive investment item, as it involves both collecting waste and transporting it to an appropriate destination.

In recent years, technologies have been created for the intelligent management of waste in order to improve the collection and disposal of waste. Dumpsters equipped with sensors now provide data on waste disposal, allowing cities to save resources and costs. These are apparently effective ways to manage waste, but the challenges are still plentiful.

The waste management process begins with the disposal of waste in bins supplied close to the point of creation. Then, the garbage is collected by trucks and sent to temporary collection sites. From these places, the garbage is sent for recycling. The implementation of this process seems quite easy; however, it is only a partial solution and creates other problems.

✓ **Misunderstanding of smart sensor operations:**

As it is a new and emerging technology, there is a general misunderstanding about its operations. Many people believe that it is a complicated and expensive method to dispose of waste, which it is not. They are actually very affordable, easy to use, durable and save costs.

✓ **Configuring the smart sensor:**

Although smart sensors are easy to use, you can't just buy one and install it in your trash. There are other steps that need to be taken after purchase to ensure its effectiveness, such as ensuring that there is communication technology for your sensor. He is responsible for collecting information about your trash and sending alerts to the appropriate ports for attention.

✓ **Non-optimized truck routes:**

Non-optimized truck routes lead to excess fuel usage. In addition, some bins may become overfilled and others less as a result. Overfilled trash cans pollute the environment and have low aesthetic quality.

✓ **Recycling:**

Having smart technology is only half the solution. The other half is to ensure that waste is disposed of responsibly and that recyclable waste is properly separated. The method of destination of the garbage from its creation points is generic, that is, all the garbage is in a single trash, therefore the sorting task falls on the garbage management company. The classification method is predominantly manual, so the process is very slow. However, an automated system that manages the life cycle of products could facilitate the process of recycling used products. Technical information about the product, including the materials from which it was obtained, the appropriate recycling method, among other things, would greatly improve the recycling process and make it more economical.

✓ **Non-uniform distribution of waste in Bins:**

Most smart sensors only use distance away to determine the fill levels of the dumps, so if garbage has been unevenly deposited in one part of the bin and the other, the sensors can read that the bin is full when reality is only half full.

CHAPTER 03

3. System analysis and Design

3.1 introduction

The new system has been developed under five phases to develop a more suitable system for waste monitoring in public places and send the notification for authorities. They are initialization, information gathering, analysis and planning, testing and development, and implementation. Also, these five components use different strategies to design a system that is easy to use and accessible. Below are some ideas about the process.

3.2 information gathering

Gathering needs is a very important step in creating a project. We must meet the requirements for waste monitoring in public places and automated waste collection system. Methods of collecting requirements that can be found in the industry. They are as follows,

- Interview
 - Questionnaires
 - Document Analysis
 - Observation
-
- Interview

This is the most commonly used method of collecting usage criteria and users can interview individuals using the program, which classifies employees or managers as high-level and low-level perspectives. Using this approach, data can be collected from the public and the need can be addressed face to face. You will be asked questions about the interviews to meet the requirements here

- Questionnaires

The quiz is a series of questions asked by targeted respondents. This method is another method of search that is often used. We can also ask questions to meet the needs. We can use both open and close questions when designing the questionnaire for data collection. Questionnaires are cheaper than interviews. This can also be done with an electronic and paper based questionnaire.

- Document Analysis

To find out how to use this method we can analyse the documents, check the documents and get the required information. Also, a large number of documents are computer aided to answer questions about the content of a set of documents.

- Observation

In this system, the time analyst has the best systematic observation to check the validity of the existing system and the information collected by the system users. The observation here is a scientific tool for the researcher and a method of data collection. The need for information monitoring should be monitored. Patterns of people, objects, and events are also the process of recording question patterns or their behaviour without communicating.

Therefore, the best method to gather information from them should be selected. I decide to choose a questionnaire and surveys as a method of gathering information from my observations. This is such an easy method. I mainly used Google forms to create a questionnaire. So my questionnaire is as follows.

Smart Garbage Monitoring System for Public Places

Design Garbage Monitoring System to ease the problems that peoples face while managing waste. In this system, we can monitor entire garbage bin using web portal in particular area.

* Required

Email address *

Your email

Full Name: *

Your answer

Figure 1 Smart garbage management system Questionnaire 1

Age: *

13 years – 25 years

26 years – 45 years

Above 50 years

Living District: *

Your answer

Figure 2 Smart garbage management system Questionnaire 2

In your opinion, is the current system of waste management in Sri Lanka is, *

Above Average

Average

Below Average

Should the current system follow for waste management in the country change in the future? *

Yes

No

Figure 3 Smart garbage management system Questionnaire 3

Do you face a problem regarding waste management in your area of residence? *

- Yes
- No

In your opinion, is the central monitoring system of bins in your region, *
*

- Above Average
- Average
- Below Average

Figure 4 Smart garbage management system Questionnaire 4

Do you think E-Waste Management and Monitoring is a more suitable solution for the waste problem in public places in Sri Lanka? *

- Yes
- No

Send me a copy of my responses.

Submit

Figure 5 Smart garbage management system Questionnaire 5

3.3 Feasibility Analysis

While designing a feasibility study is important, there are a number of ways to determine if a feasibility study is a functional system. Three feasibility studies and three tests. There's,

- Technical Feasibility
- Financial Feasibility
- Operational Feasibility

- Technical feasibility

This includes improving the waste monitoring and automated waste collection system in public places, testing the potential for cost reduction, increased efficiency and increased profitability. In addition to economic motivations, improving waste management was also a significant driving factor, which would increase environmental protection. Also, by monitoring the capacity of the containers, it is possible to design the lanes of the vehicles to dispose of the containers completely without having to go to places where there are empty or semi-empty containers unnecessarily, thus reducing fuel consumption and vehicle drag (depreciation) and increasing work efficiency. As well as that of workers and drivers. In addition, the company has information on recyclable waste disposal, and non-profit and non-economic placement courts can be restructured. The purpose of this study was to introduce the concept of IoT to improve the waste management system.

- Financial Feasibility

Through this section we can compare the costs and benefits here and consider whether the project can be done economically or not. The estimated cost of a waste monitoring and automated waste collection system in public places may not be directly linked to making money here, but there should be some benefit in having a system that justifies the cost of building it. Clearly, the benefits are never limitless, and we have a responsibility to build a network that meets the requirements for the least cost.

Item	Price
Raspberry Pi Board	Rs. 6500.00
Arduino Board	Rs. 600.00
GSM Shield for Arduino	Rs. 2000.00
Wires	Rs.100.00
Resistors	Rs.5.00
Total Cost for one garbage bin	Rs. 9205.00

Table 1 - Cost

Here the Raspberry Pi board can connect several ultrasonic sensors at once, so only one system per location is required.

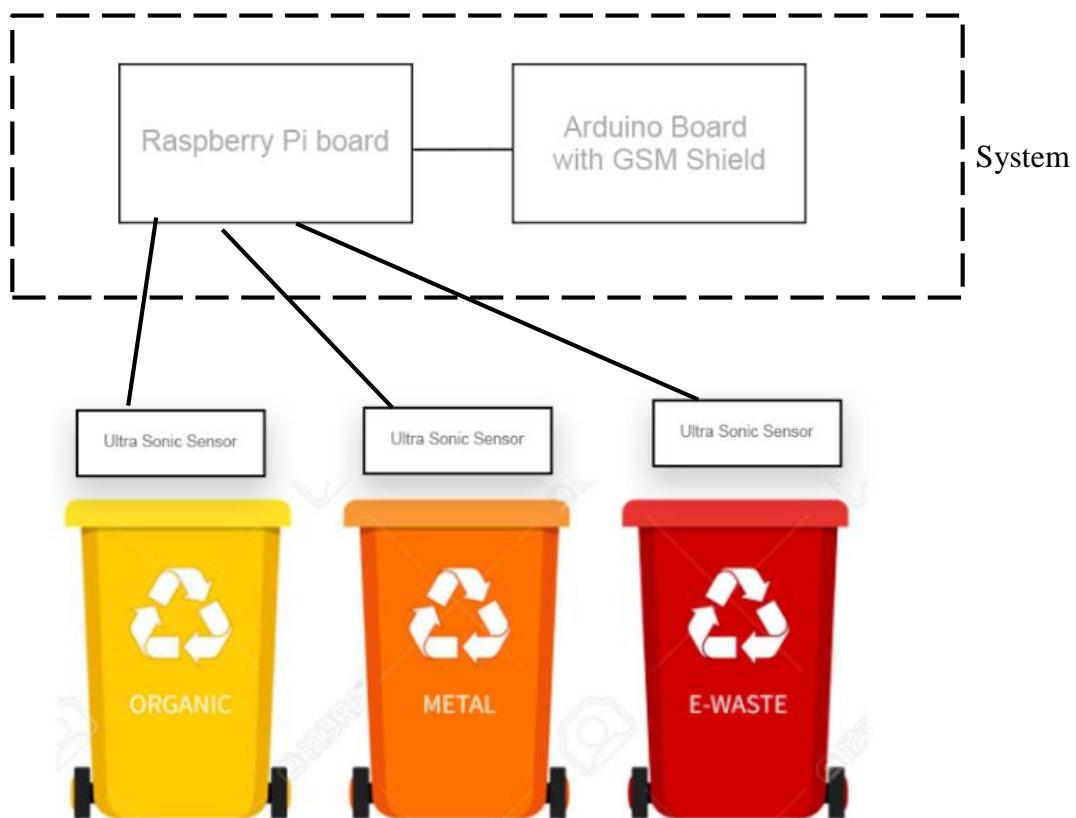


Figure 6 Financial Feasibility Diagram

- Operational feasibility

Here the operational feasibility should be tested. There, the garbage monitoring and automatic waste collection system in public places should be operational without any problems. This means that the system is as active as it gets. Also, this strategic alignment analysis assesses the suitability between the project and the business strategy while reducing the risk as the project increases from an institutional feasibility point of view. Also the system increases product productivity and makes it easier to work with. If we implement intelligent and effective strategies to keep this system safe, we can enjoy the benefits without pain. As well as. User capability for the new system should be considered.

3.5 Functional Requirement

Functional requirements are statement of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situation.

This system provides:-

1. It measure the Level of the Garbage and display it through LED
2. It gives a notification through the speaker not to drop the garbage if bin is full
3. Monitor the all the location garbage bin using single portal
4. Anyone can access system any time anywhere
5. Authorised person can view the data using mobile phone

3.6 Non-Functional Requirement

Non-functional requirements describe user visible aspects of the system that are not directly related with the functional behaviour of the system. But it can support and give more quality for new system to develop. This system works only when there is a network connection between garbage bin location and control room to inform the municipal authorities.

3.7 Design and implementation

Here planning and execution is the fourth stage of the project cycle. It is also useful to set aside this stage for value chain selection, value chain analysis and competitive strategy planning and discussion purposes. It also continues to implement many of the techniques and skills used for practical strategy selection, analysis and development. In addition, these stages of the project cycle are sequential but not linear. We also primarily need software and hardware to design and execute a system for the design of this project. The software requirements and hardware requirements for the proposed system are as follows.

- Designing

In the developed countries of the world, there are formal methods for garbage management. However, there is still no formal system for this in Sri Lanka. Therefore, after studying another project, I decided to launch a project for garbage management.

In the public places want to the garbage management system in this system using ultrasonic sensor identify the garbage's After identified the garbage raspberry board send data Net JS framework through one Send the back end of the server. Normal server path show on the server. Furthermore find out real time how many fulling of the bin and you can know the day when the bin is full.

So we can know the system click on the last selected button we can track details of the last day of the full of the bin and when the day and time of the bin was full.

The bin is in a public place, so people don't know if the bin is full, so when the bin is full, a led indicator light comes on. The massage that needs to go to bin full authorize goes through a GSM Modem to the waste collector who is in charge of the relevant area waste.

Project diagram

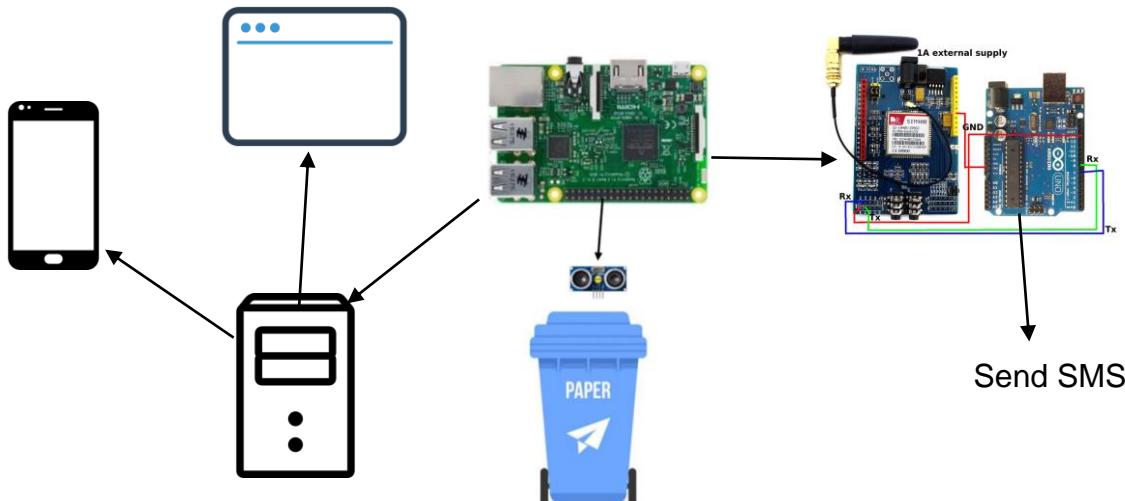


Figure 7 Project diagram

3.7.1 Software requirement

1. Virtual Network Computing (VNC)

Virtual Network Computing (VNC) is a graphical desktop sharing system that uses the Remote Frame Buffer Protocol (RFB) to remotely control another computer. It transmits keyboard and mouse events from one computer to another, and sends graphics-screen updates over the other network. It is also independent of the VNC platform - there are clients and servers for most GUI-based operating systems and Java. Multiple servers can connect to a VNC server at once. Popular uses for this technology include remote technical support and retrieving files from one's work computer from one's home computer or vice versa.



Figure 8 VNC software

2. Notepad ++

Notepad ++ is a free source code editor for "free speech" as well as "free beer" and a notepad replacement for multiple languages. It also operates in the MS Windows environment and is governed by the GNU General Public License. Also based on Cintila, a more powerful editing component, it is written in Notepad ++ C ++ and uses pure Win32 API and STL. Also, Notepad ++ seeks to reduce global carbon dioxide emissions by optimizing activity as much as possible without losing user friendliness. Low CPU

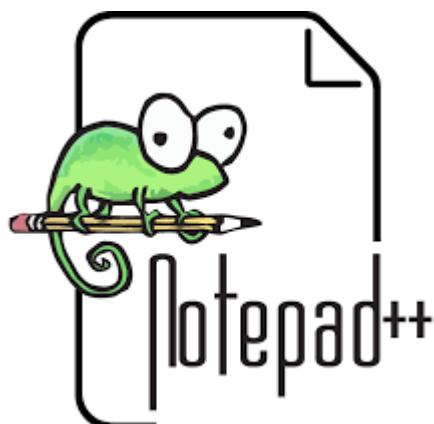


Figure 9 Notepad ++

3. Moba Xterm

Mobaxterm is an X11 server for Windows, a tabbed SSH client, and several other network tools for remote computing. (VNC, RDP, Telnet, Rologin). Mobaxterm also brings essential UNIX commands to the Windows desktop. Exe file running outside the box.



Figure 10 Moba Xterm

4. Window server 2012

Windows Server 2012 is the fifth version of the Windows Server operating system by Microsoft and is part of the Windows NT family of operating systems. It is the server version of Windows based on Windows 8 and the successful Windows 7 Server 2008 R2 released two years ago. Also in development were two pre-release versions, a developer preview and a beta version. Microsoft has been making this software available to the general public since September 4, 2012.



Figure 11 Windows Server 2012

5. Pm2

PM2 or Process Manager 2, is an open source Node.js production process manager that helps developers and developers to manage Node.js applications in the production environment. In comparison to another process manager like Supervisord, Forever, Systemd, some main features of PM2 are automatic application load balancing, declarative application configuration, deployment system and monitoring



Figure 12 Pm2 software

3.7.2 Hardware requirement

1. GARBAGE CONTAINER

A waste container is a container for temporary storage of waste. Usually made of metal or plastic. There are generally three types of garbage cans. Trash cans are containers made of metal or plastic, dumpsters, toss kips, and wheelchairs, or lightweight, usually mobile plastic containers. Also all these collectors will empty these and they will load the content in the trash. It is also excluded to take a truck and take it to a landing, combustion engine or utility facility.



Figure 13 Garbage Container

2. ULTRASONIC SENSOR

It uses a special sonic transmitter for ultrasonic proximity sensors, which allows for alternating transmission and reception of sound waves. The sonic waves emitted by the transmitter are reflected by an object and returned to the transmitter. After the sound waves are emitted, the ultrasonic sensor switches to the receiving mode. The time between emission and receipt is proportional to the distance of the object from the sensor. Also, ultrasonic sensors generate high-frequency sound waves. Estimate the return echo to the sensor, measure the time interval between sending the signal and receiving the echo to determine the distance to an object.

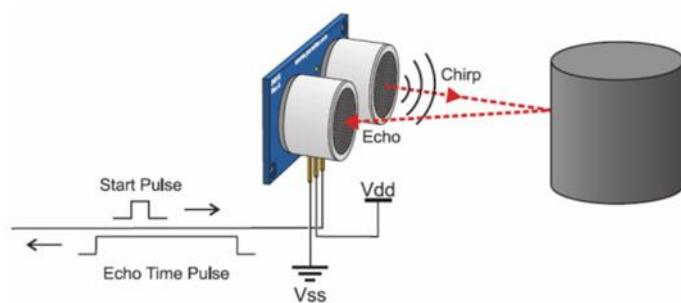


Figure 14 Ultrasonic Sensor

3. GSM module (Additional)

GSM (Global System for Mobile Communication, originally Special Mobile) is a standard set by the European Telecommunications Standards Institute (ETSI). Second generation (2G) digital mobile phones were first introduced in Finland in July 1991. By 2014, it had become the default global standard for mobile communications - operating in more than 219 countries with over 90% market share. GSM networks operate in a number of different carrier frequency bands (2G for 3G and GSM frequency bands for UMTS frequency bands), most 2G GSM networks operate in the 900 MHz or 1800 MHz bands. Where these bands were already reserved, 850 MHz and 1900 MHz bands were used instead. In rare cases, 400 and 450 MHz bands are assigned in some countries because they were previously used for generator systems.



Figure 15 GSM module

4. Raspberry Pi board

Raspberry Pi is a low-cost credit card-sized computer that connects to a computer monitor or TV and uses a standard keyboard and mouse. It is a small device that allows people of all ages to browse computers and learn how to program in languages such as scratch and Python. It also has the ability to do everything you want a desktop computer to do, such as browsing the Internet and playing high-definition videos, making spreadsheets, word processing and playing games.

Moreover, raspberry pie has the ability to interact with the outside world and has been used in a number of digital manufacturing projects to tweet bird houses from music machines and mother detectors to weather stations and from infrared red cameras. We also want to see children around the world use raspberry pie to program and understand how computers work.



Figure 16 Raspberry Pi board

5. Battery Pack

A battery pack is a number of identical batteries or individual battery cells. They can also be configured in series, in parallel or in a mixture of both to provide the desired voltage, capacitance or densities. Also the term battery pack is often used for radio controlled hobby toys and battery electric vehicles. Also, components in battery packs include individual batteries or cells, and there are interconnections between them that provide electrical conductivity. Rechargeable battery packs often contain a temperature sensor that uses a battery charger to detect the end of the charge. The part that connects each cell is also found in the interconnections of the batteries, but batteries are often made only of series threads. Also, when a package contains cell groups in parallel, there are different wiring configurations depending on the electrical balance in the circuit. Battery regulators are sometimes used to keep the voltage of each cell below its maximum value during charging. And then the weak battery allows it to fully charge. For better equilibrium, dynamic equilibrium can be achieved by battery balance devices that can open up energy in real time from strong cells to weak ones. A well-balanced package lasts longer and provides better performance. Also here for a further sorting package, the cells are selected and stacked with the solution between them. The cells are held together and the current pulse generates heat to melt them together and weld all the contacts inside the cell.



Figure 17 Battery Pack

3.8 Methodology

3.8.1 Selection of methodology

For continuation projects, choosing the process model for growth is very important as such design models can be used to manage projects well. Use these methodologies, plan, design, execute, and inspect our projects well. There are,

- Parallel Development Methodology
- V – Model Methodology
- Iterative Development Methodology
- Agile Development Methodology
- Prototype Development Methodology
- Waterfall Development Methodology

- Parallel Development Methodology

Parallel development takes place when the software needs to be developed separately on a code basis relevant to a development project. The software product can also be used by a product development team to deliver a new flagship product to customers, while a product maintenance team can work on debugging and customer patches for shipped products. Starts working on a code basis but the code is inevitably diverted. The code bases commonly used for parallel development efforts should also be integrated at a future date, and the deficiencies provided by the product maintenance team should ensure that the product development team is integrated with the major release in operation.

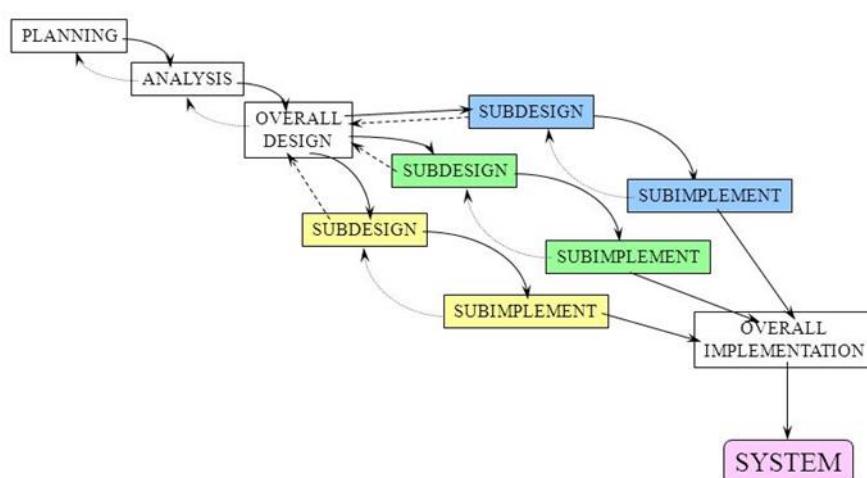


Figure 18 Parallel Development methodology

▪ V – Model Methodology

Here this V-model is another variation of the waterfall design. Also here instead of going down linearly. After the activation and encoding phase, the activation step is bent upwards from the standard V model. Example SDLL format V Process V Format V. It is also known as a template for validation and verification. Also, Model V is an extension of the waterfall approach and is based on each relevant development step in the testing process. And this code is written at the foot of V. Further tests, modules, integration tests and finally acceptance tests are performed on the right side of the upper slope of the platform. Furthermore the difference between this model V-model and the model waterfall is the initial test design of the V model. This model also uses well-defined and well-documented software specifications, as well as well-documented techniques and methods for software design. In this model too.

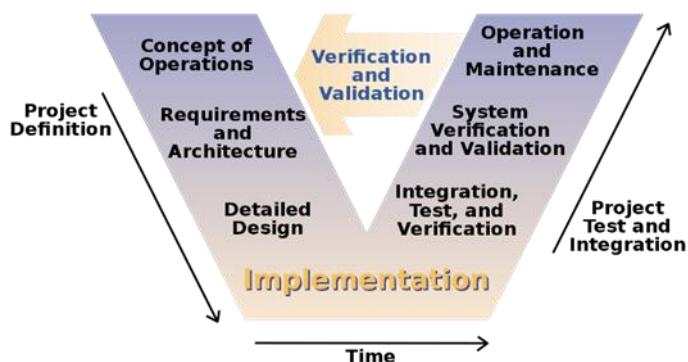


Figure 19 V-model methodology

▪ Iterative Development Methodology

Sequential Development Here the whole project is divided into a series of developed sequences. Also, the first version of the program included many crucial basic specifications, and this version will be rapidly developed and implemented through a small waterfall process. Customers can also find useful inputs for the program's next functionality, and identify additional important needs as users work with the system and include them in later editions.

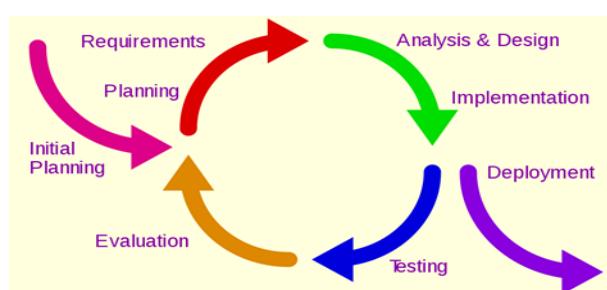


Figure 20 Iterative development methodology

- Agile Development Methodology

Iterative architecture breaks down the entire project into a series of sequences of sequences. Many of the essential basic requirements are packaged into the first iteration of the program, and this iteration is rapidly developed and implemented through a small waterfall process. Customers should have significant feedback to introduce the next version of the program. As users interact with the program, additional essential criteria in the later version can be defined and implemented.



Figure 21 Agile Development Methodology

- Prototype Development Methodology

A prototype model is a system development method that builds, tests, and modifies a prototype until an acceptable prototype can be developed to develop a complete system or product. This model works best when all project requirements are not previously described in detail. It is a process of repetition, trial and error between developers and users.

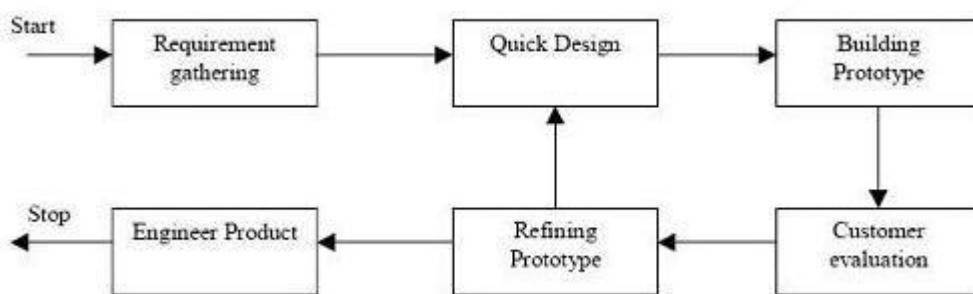


Figure 22 Prototype development methodology

- ❖ **I decided to choose Waterfall Development Methodology for build my project.**

- Waterfall Development Methodology

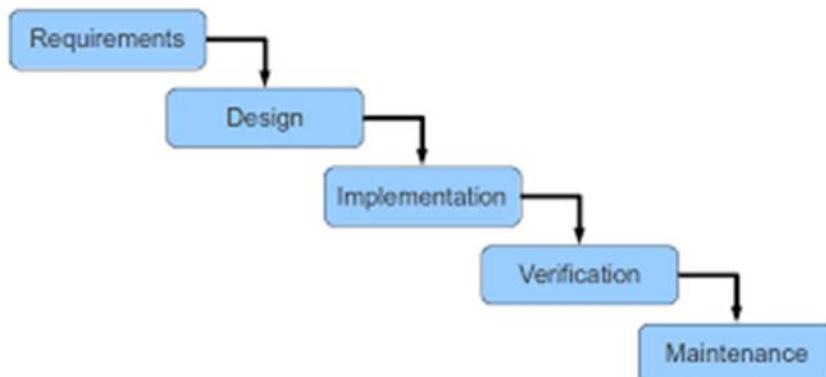


Figure 23 Waterfall development methodology

Here the waterfall model is known as the first process model introduced. This is also known as a linear-sequential life cycle model. This is very simple to understand and use. Also in this waterfall model, each stage must be completed before the next stage can begin and there is no overlap of stages. Also, this waterfall model was the first SDLC approach used for software development. Furthermore, this waterfall model represents the software development process in a linear sequential flow. This means that any phase of the development process begins only when the previous phase is completed.

The sequential phases in Waterfall model are,

- Requirement Gathering and analysis
- System Design
- Implementation
- Integration and Testing
- Deployment of system
- Maintenance

- ✓ **Requirement Gathering and analysis** - All the developmental requirements of the system are captured at this stage and the requirements are documented in a specification document.
- ✓ **System Design** - Requirement specifications from the first stage are studied at this stage and the system design is prepared. This system design helps specify hardware and system requirements and defines the overall system layout.
- ✓ **Implementation** - With system design applications, the system is initially developed by small programs called units, which are integrated into the next stage. Each unit is developed and tested for its functionality, called unit testing.
- ✓ **Integration and Testing** - All units developed during the activation phase are integrated into a system after each unit is tested. Post-integration the entire system is checked for any errors and failures.

- ✓ **Deployment of system** - After performing both active and non-active tests; the product is either deployed in the consumer environment or released to the market.
- ✓ **Maintenance** - There are several issues that arise in the client environment. Patches are released to fix those issues. Several better versions will be released to improve the product. Maintenance is done to make these changes in the customer environment.

3.9 Advantages and Disadvantages of Waterfall Development Methodology

Advantages	Disadvantages
Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.	Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
Works well for smaller projects where requirements are very well understood.	Adjusting scope during the life cycle can end a project.
Simple and easy to understand and use	Not a good model for complex and object-oriented projects.
Phases are processed and completed one at a time.	Poor model for long and ongoing projects
Well understood milestones	Cannot accommodate changing requirements
Process and results are well documented.	High amounts of risk and under
Clearly defined stages.	It is difficult to measure progress within stages.
Process and results are well documented.	No working software is produced until late during the life cycle.

Table 2 Advantages and Disadvantages of Waterfall Development Methodology

3.10 Work Plan

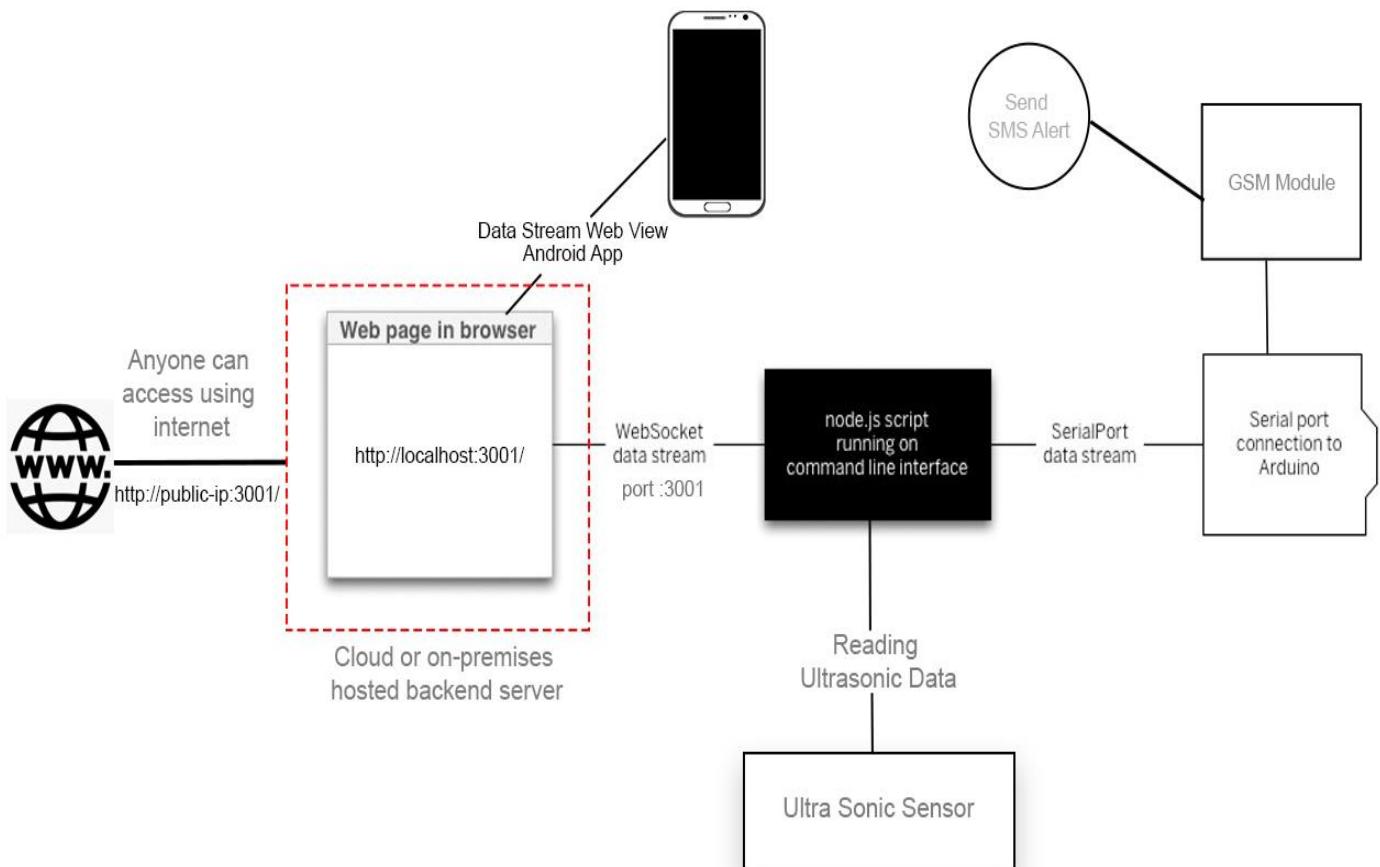
3.10.1 Gantt chart for project

No	Task	Duration	April				May				June				July				August				September			
			week	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
1	Identify Existing Situation	1 weeks																								
	Information gathering existing situation																									
2	Analyze and Design	2 weeks																								
	Prepare the system diagram																									
	Documentation																									
3	Develop System and Network	10 weeks																								
	purchasing new devices and other equipments																									
	Hardware Configuration																									
	Servers installation and configuration																									
	Design Web Portal																									
	Implement IOT Alert Gateway																									
	Network Configuration																									
4	Implementing and Testing	10 weeks																								
	Device Testing																									
	Web Portal and Device Communication Testing																									
	Making mistakes																									
	Implementation Public Area																									
	getting feedback from peoples																									
	making changes according to feedback																									
5	Finalized the project	3 weeks																								
	Documentation																									
	Finalized the project																									

Table 3 Gantt chart

It was expected to be completed on schedule but could not be completed before the scheduled date due to the COVID-19 situation.

3.11 Diagrams



3.12 System Development

Client Side Development

GSM Connection with Arduino micro controller connection diagram

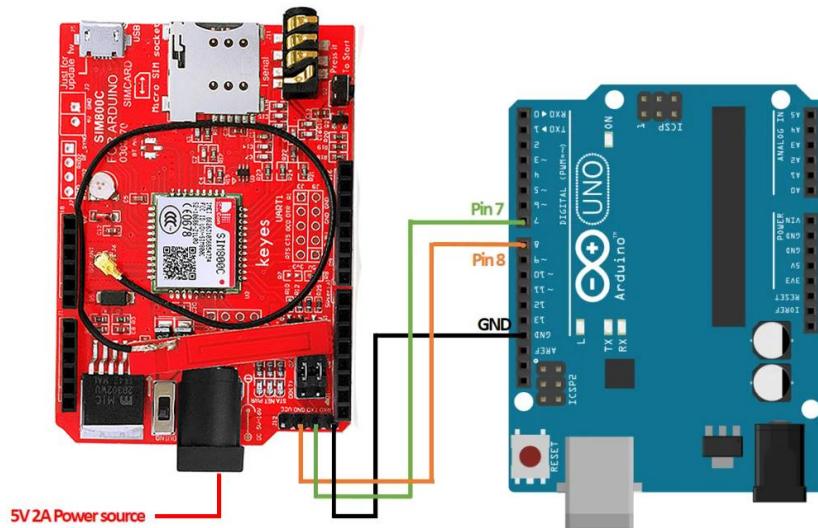


Figure 24 GSM and Arduino connection

GSM stands for **Global System for Mobile Communications** and is the global standard for mobile communications.

GPRS stands for **General Packet Radio Service**. GPRS is a mobile service on the 2G and 3G cellular communication.

Applications:

The GSM GPRS shield is particularly useful as it allows to:

- Connect to the Internet over GPRS network
- Send and receive SMS
- Place and receive phone calls

Its capabilities make it perfect for projects with Arduino like:

- Remote control of electronic appliances – sending an SMS to turn something on;
- Receive notifications – send SMS to your cell phone if movement is detected in your house;
- Receive sensor data – send periodic SMS to your cell phone with daily weather data.

Features

Here's some of the most important features of the shield:

- Compatible with Arduino and clones
- Based on SIM900 module from SIMCOM
- Allows you to send SMS, MMS, GPRS and Audio via UART using AT commands.
- It has 12 GPIOs, 2 PWMs and built-in ADC of the SIM900 module
- Quad Band: 850; 900; 1800 and 1900 MHZ, so it should work in all countries with GSM (2G) networks
- Control via AT commands
- Supports RTC (real time clock) – it has a holder for a 3V CR1220 battery at the back

In this project I'm using SIM800C Arduino shield for connect GSM network and send the SMS to authorities after garbage bin is full.

For the send sms using SIM800C GMS module we can use AT Command,

- set the SIM900 to text mode: **AT+CMGF=1\r**
- send SMS to a number: **AT+CMGS=PHONE_NUMBER** (in international format)
- read the first SMS from the inbox: **AT+CMGR=1\r**
- read the second SMS from the inbox: **AT+CMGR=2\r**
- read all SMS from the inbox: **AT+CMGR=ALL\r**
- call to a number: **ATD+ PHONE NUMBER** (in international format)
- hang up a call: **ATH**
- receive an incoming call: **ATA**

Pin Define

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial (7, 8);

void setup()
{
    mySerial.begin(9600);
    Serial.begin(9600);

    Serial.println("s-Send SMS\n c-Call\n x-Cut Call\n a-Answer Call\n n\n");
}
```

Figure 25 Pin define coding 1

```

void loop()
{
    if (Serial.available() > 0)
        switch (Serial.read())
    {
        case 's':
            SendMessage();
            break;
        case 'c':
            MakeCall();
            break;
        case 'x':
            HangupCall();
            break;
        case 'a':
            ReceiveCall();
            break;
    }

    if (mySerial.available() > 0)
        Serial.write(mySerial.read());
}

```

Figure 26 Pin define coding 2

Send Message using AT Commands

```

void SendMessage()
{
    mySerial.println("AT+CMGF=1");
    delay(1000);
    mySerial.println("AT+CMGS=\"+94764106169\"\r");
    delay(1000);
    mySerial.println("Garbage Bin is Full.");
    delay(100);
    mySerial.println((char)26);
    delay(1000);
}

```

Figure 27 Send message coding

Raspberry pi board and ultra-sonic sensor connection diagram

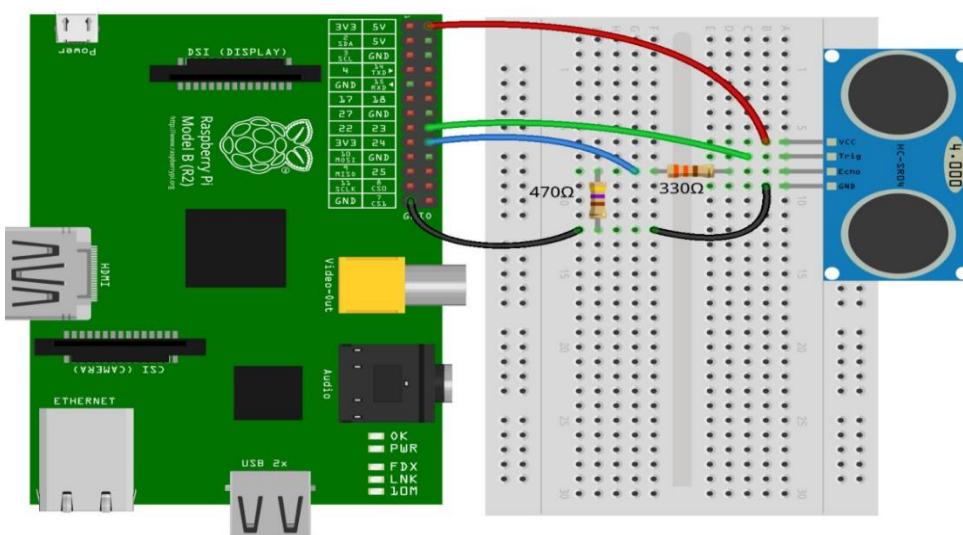


Figure 28 Raspberry pi board and ultra-sonic sensor connection diagram

In this project main thing is get the garbage bin level data real-time. For that requirement I used HC-SR06 Ultra sonic sensor and raspberry pi computer get the data and pass that data into backend server.

Raspberry pi pin define code

```

1  const fs = require('fs');
2
3  eval(fs.readFileSync('include/includeModules.js') + '');
4  eval(fs.readFileSync('include/includeCommonFunctions.js') + '');
5
6
7  const Gpio = require('pigpio').Gpio;
8
9  // The number of microseconds it takes sound to travel lcm at 20 degrees celcius
10 const MICROSECDONDS_PER_CM = 1e6/34321;
11
12 const trigger = new Gpio(11, {mode: Gpio.OUTPUT});
13 const echo = new Gpio(8, {mode: Gpio.INPUT, alert: true});
14 const led = new Gpio(4, {mode: Gpio.OUTPUT});
15
16 trigger.digitalWrite(0); // Make sure trigger is low
17
18 var binbottom=100; //TODO
19 var binfull=10; //TODO
20 var binvalue=0;
21 var ledcount=0;
22
23 var file = 'data/data.ini'
24 var fileObj = fileRead();
25 var count=0;
```

Figure 29 Raspberry pi pin define code

Server and Client Connection Code –

```

26 const
27   io = require("socket.io-client"),
28   //ioClient = io.connect("http://localhost:3001");
29   ioClient = io.connect("http://202.124.183.60:3001");
30
31
32
33 setsetIntervalTimer= setInterval(function () {
34
35   fileObj.binstatus=binvalue;
36
37   // console.log(fileObj);
38
39   ioClient.emit("BIN_DATA",fileObj);
40
41 },1000);

```

Figure 30 Server and Client Connection Code –

How to Calculate Distance?

We will now see how to measure the distance of an object using HC-SR04 Ultrasonic Sensor. In order to send the 40 KHz Ultrasound, the TRIG Pin of the Ultrasonic Sensor must be held HIGH for a minimum duration of 10 μ S.



Figure 31 ultrasound transmitter and receiver

After this, the Ultrasonic Transmitter, will transmits a burst of 8-pulses of ultrasound at 40 KHz. immediately, the control circuit in the sensor will change the state of the ECHO pin to HIGH. This pins stays HIGH until the ultrasound hits an object and returns to the Ultrasonic Receiver.

Based on the Time for which the Echo Pin stays HIGH, you can calculate the distance between the sensor and the object.

For example, if we calculated the time for which ECHO is HIGH as $588\mu\text{s}$, then you can calculate the distance with the help of the speed of sound, which is equal to 340m/s .

Distance = Velocity of Sound / (Time/2) = $340\text{m/s} / (588\mu\text{s} / 2) = 10\text{cm}$.

Following is the how to calculate garbage bin level ultra-sonic sensor

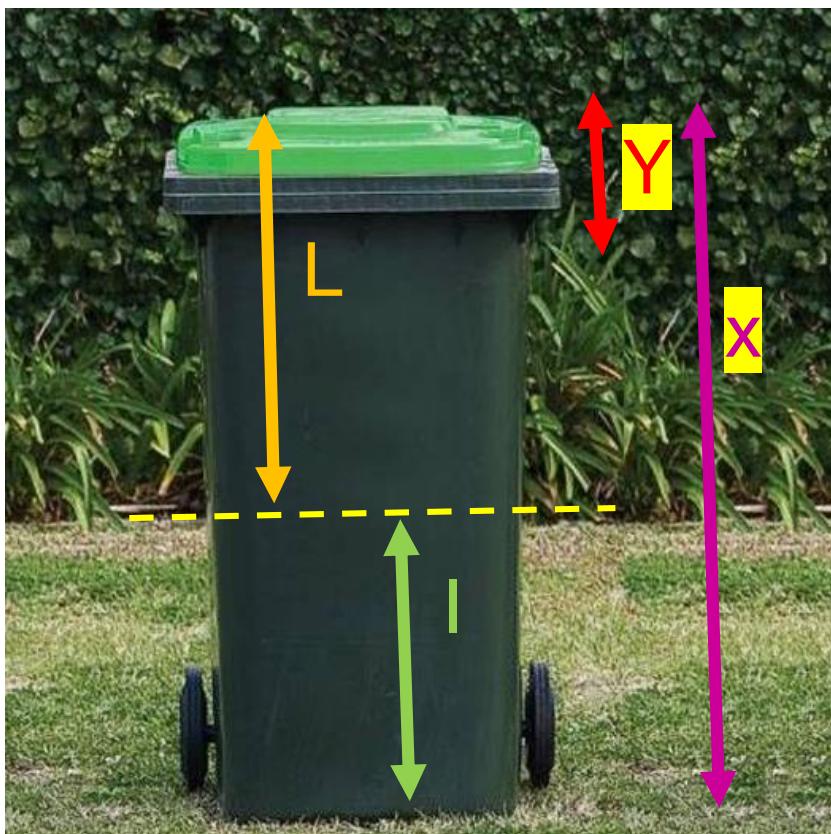


Figure 32 Garbage bin

X = Garbage Bin Height

Y = Garbage Bin Lid Height

L = Bin top to Trash level Height

I = Trash Level

$$\text{Trash Level (I) \%} = \frac{(X-L)}{(X-Y)} * 100$$

Distance Calculation Code –

```

46  const watchHCSR04 = () => {
47    let startTick;
48    echo.on('alert', (level, tick) => {
49      if (level == 1) {
50        startTick = tick;
51        // console.log("start tick");
52      } else {
53        //console.log("dd");
54        const endTick = tick;
55        const diff = (endTick >> 0) - (startTick >> 0); // Unsigned 32 bit arithmetic
56        var temps=(diff / 2 / MICROSECDONDS_PER_CM).toFixed(0);
57        if(temps>binfull*9/10){
58          if(temps<binbottom){
59            binvalue=((binbottom-temps)*100/(binbottom-binfull)).toFixed(1);
60          }else{
61            binvalue=0;
62          }
63        if(binvalue>=98){
64          led.digitalWrite(1);
65          ledcount=0;
66        }else{
67          if(ledcount>5){
68            led.digitalWrite(0);
69          }
70          console.log(temps);
71          console.log("bin value      "+binvalue +" %");
72        }
73      }
74    });
75  });
76 };

```

Figure 33 Distance Calculation Code

All the bin collection data event log goes to data.ini file.

```
23  var file = 'data/data.ini'
```

Figure 34 Bin collection data

Server Side Development

Package.json file

The package.json file is saved to the system libraries.

```
{
  "name": "garbage-system-dashboard",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "dev": "nodemon ./index.js",
    "ui": "browser-sync start --proxy=localhost:8000 --files='**/*.css, **/*.pug, **/*.js' --ignore=node_modules --reload-delay 10 --no-ui --no-notify"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "browser-sync": "^2.26.7",
    "nodemon": "^2.0.4"
  },
  "dependencies": {
    "clone": "^2.1.2",
    "colors": "1.4.0",
    "delay": "4.3.0",
    "express": "4.17.1",
    "mgtt": "4.1.0",
    "nw": "0.46.3",
    "pug": "3.0.0",
    "redis": "3.0.2",
    "request": "2.88.2",
    "screenfull": "5.0.2",
    "shai": "1.1.1",
    "sio": "0.0.4"
  }
}
```

Figure 35 server side Jason file

Index.js

All the system important codes are located in index.js (Eg: init function)

```

1 // index.js
2
3 /**
4  * Required External Modules
5 */
6 const express = require("express");
7 const path = require("path");
8 const screenfull = require('screenfull');
9 const colors = require('colors');
10
11 var http = require('http');
12 const fs = require('fs');
13 eval(fs.readFileSync('./include/Constants.js') + '');
14 eval(fs.readFileSync('./include/CommonFunctions.js') + '');
15
16 var redis = require("redis");
17 var sio = require("socket.io");
18 var jsonfile = require('jsonfile');
19
20 const delay = require('delay');
21 const e = require("express");
22 /**
23  *
24  * App Variables
25 */
26 const app = express();
27 const port = process.env.PORT || "8000";
28 var setsetIntervalTimer;
29 console.log("-----");
30 console.log("| " + getDateTime() + " | Smart Garbage Management System Server init");
31 console.log(colors.bgBlack.green("-----"));
32
33
34
35 var file = 'data/data.ini'
36 var fileObj = fileRead();

```

Figure 36 index.js code 1

```

37  /**
38   * Validate
39   */
40  console.table(fileObj.bins);
41
42  app.set('port', 3001);
43  var server = http.createServer(app).listen(app.get('port'), function () {
44    console.log("Express server listening on port " + app.get('port'));
45  });
46  var io = require('socket.io').listen(server);
47
48
49  /**
50   * App Configuration
51   */
52  app.set("views", path.join(__dirname, "views"));
53  app.set("view engine", "pug");
54  app.use(express.static(path.join(__dirname, "public")));
55

```

Figure 37 index.js code 2

```

74  * Server Activation
75  */
76
77  app.listen(port, () => {
78    console.log(`Listening to requests on http://localhost:\${port}`);
79  });
80
81  io.sockets.on('connection', function (socket) {
82    console.log("Socket connected");
83    console.log(socket.id);
84    //fileWrite();
85
86    socket.on('disconnect', function (data) {
87      console.log("Socket disconnect");
88      console.log(socket.id);
89    });
90
91    socket.on('UI_COMMAND', function (data) {
92      console.log("UI_COMMAND", data);
93
94    });
95    socket.on('BIN_RESET', function (msg) {
96      var data=JSON.parse(msg);
97      console.log("BIN_RESET", data);
98      for (var x = 0; x < fileObj.bins.length; x++) {
99        var bin = fileObj.bins[x];
100       if (bin.id==data["id"]) {
101         bin.status=0;
102         bin.lastCollect=getDateTime();
103         fileObj.bins[x]=bin;
104       }
105     }
106   });

```

Figure 38 index.js code 3

```

104           //  

105       }  

106   }  

107 });
108   socket.on('BIN_DATA', function (data) {
109     console.log("BIN_DATA", data);
110     var newstatus=true;
111     for (var x = 0; x < fileObj.bins.length; x++) {
112       var bin = fileObj.bins[x];
113       if (bin.id==data.id) {
114         newstatus=false;
115         bin.status=data.binstatus;
116         bin.statusUpdate=getDateTime();
117         fileObj.bins[x]=bin;
118         //
119       }
120     }
121     if(newstatus){
122       var newbin={};
123       newbin.id=data.id;
124       newbin.status=data.binstatus;
125       newbin.location=data.location;
126       newbin.name=data.binname;
127       newbin.type=data.type;
128       newbin.lastCollect="";
129       newbin.statusUpdate=getDateTime();
130       fileObj.bins.push(newbin);
131     }
132   //fileWrite();

```

Figure 39 index.js code 4

```

134 });
135   socket.on('INITIAL', function (obj) {
136     var data = JSON.parse(obj);
137     console.log("INITIAL".bgBlack.green);
138     console.log(data);
139   });
140   socket.on('LOGIN', function (obj) {
141     var data = JSON.parse(obj);
142     if(data.username=="vihanga"){
143       if(data.password=="1234"){
144         var responce= { states: "sucess",error_code:0,error:null }
145         io.sockets.emit("INITIAL_RESPONCE", JSON.stringify(responce));
146       }else{
147         var responce= { states: "failed",error_code:2,error:"Invalid Password" }
148         io.sockets.emit("INITIAL_RESPONCE", JSON.stringify(responce));
149       }
150     }else{
151       var responce= { states: "failed",error_code:1,error:"Invalid User Name" }
152       io.sockets.emit("INITIAL_RESPONCE", JSON.stringify(responce));
153     }
154   });
155   clearInterval(setsetIntervalTimer);
156   setsetIntervalTimer= setInterval(function () {
157     sendUIDFVData();
158   },1000);
159 });

```

Figure 40 index.js code 5

```

162  function sendUIDFVData() {
163
164    var jsnObjHeartBeat = {
165      jsnObj: fileObj.bins
166    };
167    io.sockets.emit("UI_DATA", JSON.stringify(jsnObjHeartBeat));
168    console.log("socket emitted");
169    console.table(jsnObjHeartBeat.jsnObj);
170    //fileWrite();
171  }
172

```

Figure 41 index.js code 6

```

1  var elem = document.documentElement;
2  var UI_COMMAND_JSON = { COMMAND: "" };
3
4  var fullScreenStatus = false;
5  var fullScreenGridItem;
6  var socket;
7  $(document).ready(function () {
8    $(".login-div").show();
9    $('#main-div').hide();
10   $('#login-error-msg').hide();
11   connect();
12 });
13

```

Figure 42 Index.js code 7

```

16  function connect() {
17
18    socket = io('http://52.152.216.133:3001/');
19    // socket = io('http://localhost:3001/');
20    socket.on('connect', function(){
21      console.log("Connect to Local Server");
22
23    });
24    socket.on('INITIAL_RESPONSE', function (obj) {
25      var data = JSON.parse(obj);
26      // console.log("INITIAL_RESPONSE");
27      //console.log(data);
28      if(data.error_code==0){
29        $(".login-div").hide();
30        $('#main-div').show();
31        $('#login-error-msg').hide();
32        $('#sign-out').html("<i class='fa fa-sign-out' aria-hidden='true'>logout</i>");
33      }else{
34        $('#login-error-msg').html(data.loginErrorMsg);
35        $('#login-error-msg').show();
36      }
37    });
38    socket.on('UI_DATA', function (obj) {
39      var data = JSON.parse(obj);
40      //console.log(data.jsonObj);
41      tableDraw(data.jsonObj);
42    });

```

Figure 43 index.js code 8

```

44     $("#btn-login").click(function () {
45         if ($("#txt-username").val() != null && $("#txt-username").val() != "" &&
46             $("#txt-password").val() != null && $("#txt-password").val() != "") {
47             var loginJson = { username: $("#txt-username").val(), password: $("#txt-password").val() }
48             console.log("button click debug");
49             socket.emit("LOGIN", JSON.stringify(loginJson));
50         }
51     });
52     $("#sign-out").click(function () {
53         $(".login-div").show();
54         $('#main-div').hide();
55         $('#login-error-msg').hide();
56         $('#txt-username').val("");
57         $('#txt-password').val("");
58     });
59 }
60

```

Figure 44 index.js code 9

```

62     function secondsToHms(date) {
63
64         var startDate = new Date(date);
65         var endDate = new Date();
66         var seconds = (endDate.getTime() - startDate.getTime()) / 1000;
67         d = Number(seconds);
68         var day = Math.floor(d / 86400);
69         var h = Math.floor(d / 3600);
70         var m = Math.floor(d % 3600 / 60);
71         var s = Math.floor(d % 3600 % 60);
72
73         var dayDisplay = day > 0 ? day + (day == 1 ? " day ago" : " days ago") : "";
74         var hDisplay = h > 0 ? h + (h == 1 ? " hour ago" : " hours ago") : "";
75         var mDisplay = m > 0 ? m + (m == 1 ? " minute ago" : " minutes ago") : "";
76         var sDisplay = s > 0 ? s + (s == 1 ? " second ago" : " seconds ago") : "";
77         var txt = "";
78         if (day > 0) {
79             txt = dayDisplay;
80         } else {
81             if (h > 0) {
82                 txt = hDisplay;
83             } else {
84                 if (m > 0) {
85                     txt = mDisplay;
86                 } else {
87                     txt = sDisplay;
88                 }
89             }
90         }
91
92         // return hDisplay + mDisplay + sDisplay;
93         return txt;
94     }

```

Figure 45 index.js code 10

```

95     function tableDraw(data){
96         // console.log(data);
97         var headerdata=<table><tr><th>Id</th><th>Name</th><th>Location</th><th>Type</th><th>Status</th><th>Last Collect Date</th><th></th></tr>;
98         var enddata=</table>';
99         var midData="";
100        for(var i=0;i<data.length;i++){
101            var bin=data[i];
102            midData+="|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| " + bin.id + " | " + bin.name + " | " + bin.location + " | " + bin.type + " | " + bin.status + " | " + bin.lastCollect + 103            "<td><td><i class='fa fa-recycle' onclick='binrest(" + bin.id + ")' aria-hidden='true'></i></td></tr>"; 104        } 105 106        $("#table").html(headerdata+midData+enddata); 107    } |

```

Figure 46 index.js code 11

```

108     function binrest(binId) {
109         var data = { id: binId }
110         ...
111         console.log("BIN_RESET "+binId);
112         socket.emit("BIN_RESET", JSON.stringify(data));

```

Figure 47 index.js code 12

User Interface style sheet

```

2      @font-face {
3          font-family: "Helvetica";
4          src: url("/fonts/Helvetica-Bold.ttf") format("ttf"),
5              url("/fonts/Helvetica Neu Bold.ttf") format("ttf");
6      }
7
8      body {
9          width: 100vw;
10         height: 100vh;
11         display: flex;
12         justify-content: center;
13         align-items: center;
14         flex-direction: column;
15         margin: 0;
16         font-family: 'Roboto', sans-serif;
17
18         &.no-scroll {
19             overflow: hidden;
20         }
21     }
22
23     p{
24         font-size: .7rem;
25
26         margin-left: 2px;
27         margin-bottom: 0;
28
29         /* width */
30         ::-webkit-scrollbar {
31             width: 5px;
32             height: 5px;
33         }
34         .mod-min ::-webkit-scrollbar {
35             width: 0;
36             height: 0;

```

Figure 48 style sheet code 1

```

39  /* Track */
40  ::-webkit-scrollbar-track {
41      background: #f1f1f1;
42  }
43
44  /* Handle */
45  ::-webkit-scrollbar-thumb {
46      background: #888;
47  }
48
49  /* Handle on hover */
50  ::-webkit-scrollbar-thumb:hover {
51      background: #555;
52  }
53
54  .mod-min{
55      /*Animations*/
56  /*     -webkit-transition: all 0.5s ease;
57     -moz-transition: all 0.5s ease;
58     -o-transition: all 0.5s ease;
59     transition: all 0.5s ease; */
60      opacity: 1;
61      justify-content: center;
62      align-items: center;
63      flex-direction: column;
64
65  }

```

Figure 49 style sheet code 2

```

68  .close{
69      margin-right: 10px;
70  }
71
72  .card-body{
73      width: 100%;
74      height: 95%;
75      padding: 0 0 5px 0
76  }
77  .card-body-btn{
78      position: absolute;
79      width: 100%;
80      height: 90%;
81      background: transparent;
82      border: none !important;
83      font-size:0;
84      display:block;
85      cursor:pointer;
86
87  }
88  .card-body-btn:focus {
89      outline: none;
90      box-shadow: none;
91  }

```

Figure 50 style sheet code 3

```

93  .splash {
94      /*background: url("./logo.png"); */
95      width: 360px;
96      height: 360px;
97      margin: auto;
98      padding: 10px;
99      align-items: center;
100 }
101 .splash-div{
102     align-items: center;
103     margin: auto;
104     padding: 10px;
105 }
106 table {
107     border-collapse: collapse;
108     width: 100%;
109     font-size: 2vmin;
110 }

```

Figure 51 style sheet code 4

```

112 th, td {
113     text-align: left;
114     padding: 8px;
115 }
116
117 tr:nth-child(even){background-color: #555}
118
119 th {
120     background-color: #4CAF50;
121     color: white;
122 }
123 .header{
124     background-color: #222;
125     color: white;
126     font-size: 5vmin;
127 }
128 .tabeldiv{
129     position: absolute;
130     top: 8vmin;
131 }
132
133 #main-div{
134     padding: 0%;
135     margin: 0;
136 }
137 .fas:hover ,.fa:hover{
138     color: #fff;
139 }
140 .fas ,.fa{
141     color: #ccc;
142 }
143

```

Figure 52 style sheet code 5

```

144  .fa-sign-out{
145      text-align: right;
146      font-family: 'Roboto', sans-serif;
147      font-size: 2vmin;
148      color: #ccc;
149  }
150  .fa-sign-out:hover{
151      color: #fff;
152  }

```

Figure 53 style sheet code 6

UI Designing using node JS pug dependency

Index page layout

```

1  extends layout
2
3  block layout-content
4      div.login-div
5          div
6              label#login-error-msg Error
7              div.form-group
8                  label Username:
9                  input#txt-username.form-control(name='username', type='text')
10             div.form-group
11                 label Password:
12                 input#txt-password.form-control(name='password', type='password')
13             div.form-actions
14                 input#btn-login.btn.btn-primary(type='submit', value='Login')
15         div#main-div.col-12.row
16             div.col-12.row
17                 div.col-11
18                     span.header Smart Garbage Management System
19                     div.col-1.pull-right#sign-out
20             div.col-12.tabeldiv
21                 div#table
22

```

Figure 54 index page layout

Dashboard Layout

```

1  block variables
2  doctype html
3  html
4      head
5          meta(charset="utf-8")
6          meta(name="viewport", content="width=device-width, initial-scale=1, shrink-to-fit=no")
7          meta(name="theme-color", content="#000000")
8          title #{title} Smart Garbage Management System
9          link(rel="stylesheet" href="/fontawesome/css/all.css")
10         link(rel="stylesheet" href="/bootstrap.min.css")
11         link(rel="stylesheet" href="/style.css")
12         script(src="/jquery.js")
13         script(src="/socket.io-1.4.5.js")
14         script(src="/app.js")
15     body
16         block layout-content

```

Figure 55 Dashboard layout

Server side all the bin data store the data.ini file

```
1 {"bins": [{"id":1,"name":"bin 1","location":"panadura","type":"plastic","status":45,"lastCollect":"2020-07-04","statusUpdate":"2020-07-04"}]}
```

Figure 56 Server side data.ini file

Mobile App Development

```
4
5     <uses-permission android:name="android.permission.INTERNET"/>
6
```

Figure 57 Mobile app internet permission

```
1 package com.example.gmsystem;
2
3 import ...
10
11 public class MainActivity extends AppCompatActivity {
12
13     private WebView webView;
14 }
```

Figure 58 Web view code 1

```
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21
22         webView = findViewById(R.id.webView);
23         webView.setWebViewClient(new WebViewClient());
24         webView.loadUrl("http://3.138.50.14:3001/");
25
26         WebSettings webSettings = webView.getSettings();
27         webSettings.setJavaScriptEnabled(true);
28     }
29 }
```

Figure 59 Web view code 2

```

31     @Override
32     public void onBackPressed() {
33         if(webView.canGoBack()){
34             webView.goBack();
35         } else {
36             super.onBackPressed();
37         }
38     }
39 }
40

```

Figure 60 Web view code 3

Activity Splash Layout

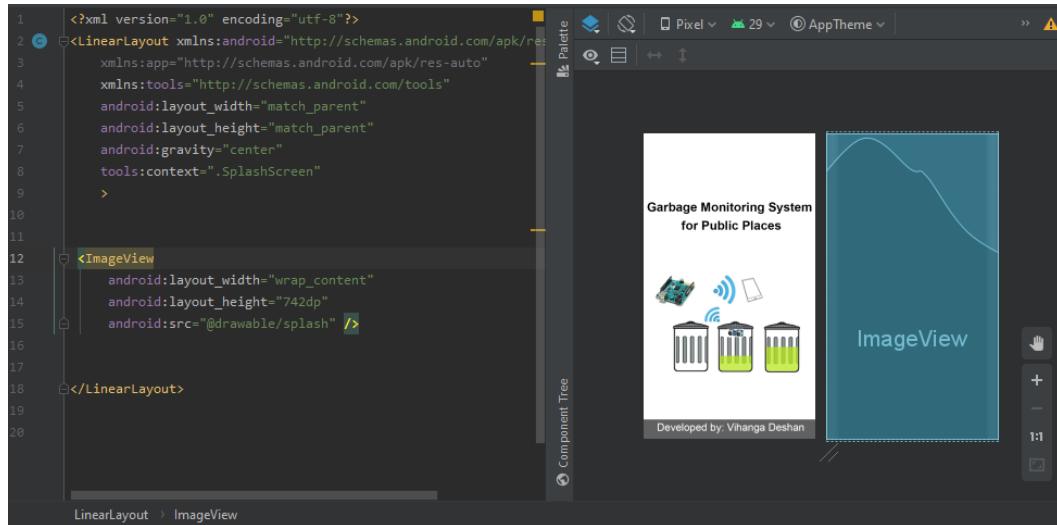


Figure 61 Activity Splash layout

Activity Main Layout

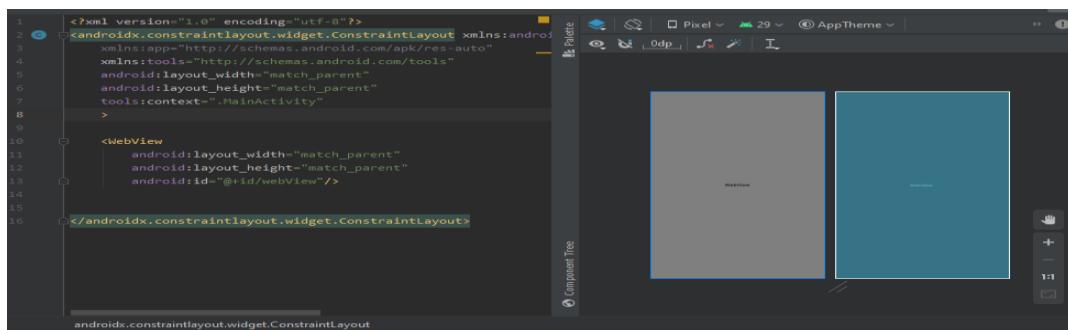


Figure 62 Activity Splash layout Activity

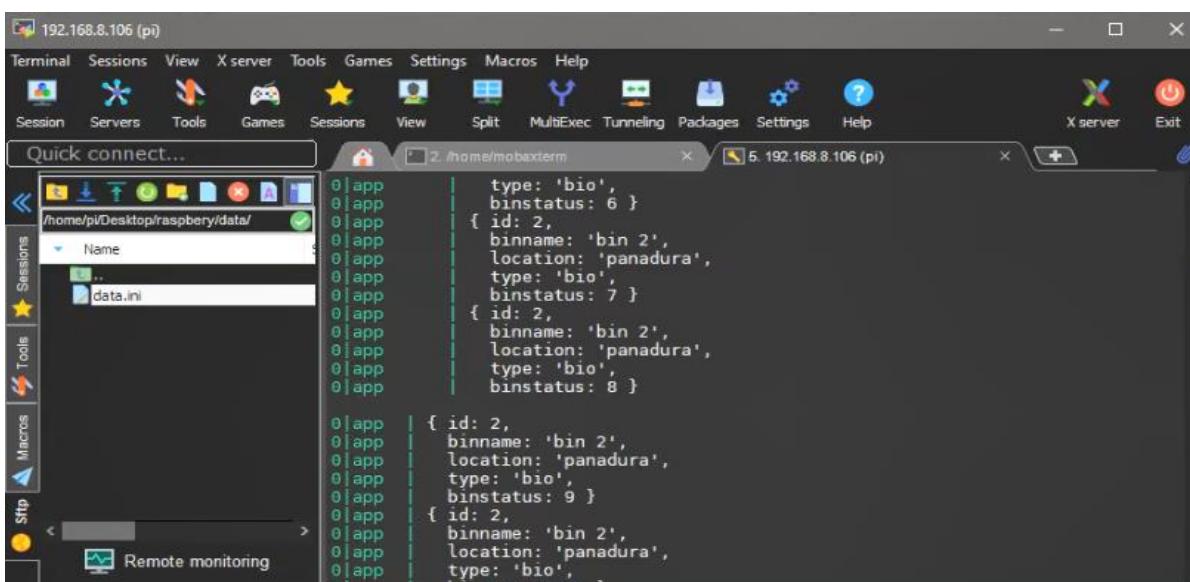
3.13 Final Testing

1. Verify ultrasonic connectivity

```
pi@raspberrypi:~/Desktop/raspberry $ sudo node app.js
file reading...
17
bin value    92.2 %
18
bin value    91.1 %
22
bin value    86.7 %
19
bin value    90.0 %
16
bin value    93.3 %
12
bin value    97.8 %
10
bin value    100.0 %
```

Figure 63 Testing 1

2. Verify ultrasonic connectivity using Pm2 logs



The screenshot shows the MoboTerm application interface. At the top, there is a menu bar with options like Terminal, Sessions, View, X server, Tools, Games, Settings, Macros, Help, Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help. Below the menu is a toolbar with icons for Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help.

The main area of the interface contains two terminal windows. The left terminal window is titled "Quick connect..." and shows a file browser interface with a tree view of files and a list view below it. The list view includes items like ".app", "Name", and "data.ini". The right terminal window is titled "/home/pi/Desktop/raspberry/data/" and displays the contents of a file named "data.ini". The file's content is as follows:

```
type: 'bio',
binstatus: 6 }
{ id: 2,
binname: 'bin 2',
location: 'panadura',
type: 'bio',
binstatus: 7 }
{ id: 2,
binname: 'bin 2',
location: 'panadura',
type: 'bio',
binstatus: 8 }

{ id: 2,
binname: 'bin 2',
location: 'panadura',
type: 'bio',
binstatus: 9 }
{ id: 2,
binname: 'bin 2',
location: 'panadura',
type: 'bio',
binstatus: 10 }
```

At the bottom of the interface, there is a sidebar with icons for Sessions, Tools, Macros, and Ssh, along with a "Remote monitoring" status indicator.

Figure 64 Testing 2

3. Raspberry pi and Server web Socket connectivity testing

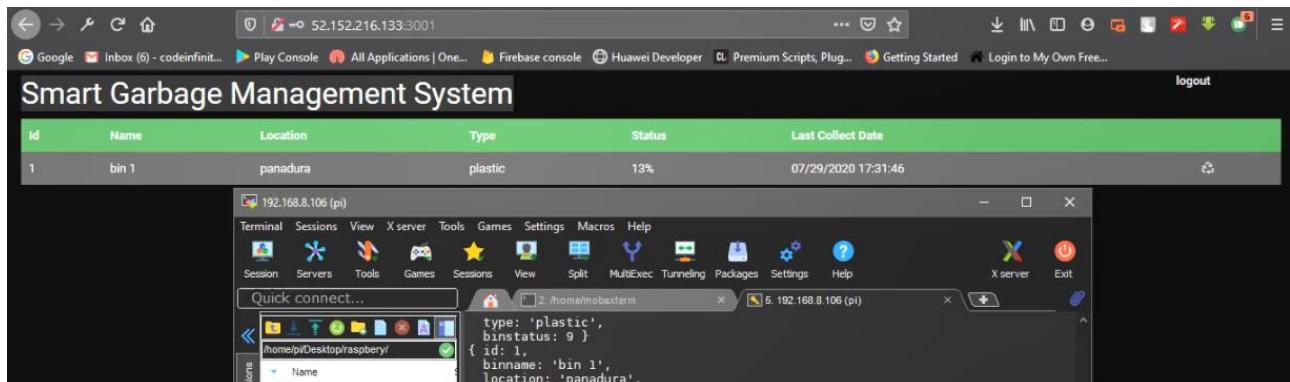
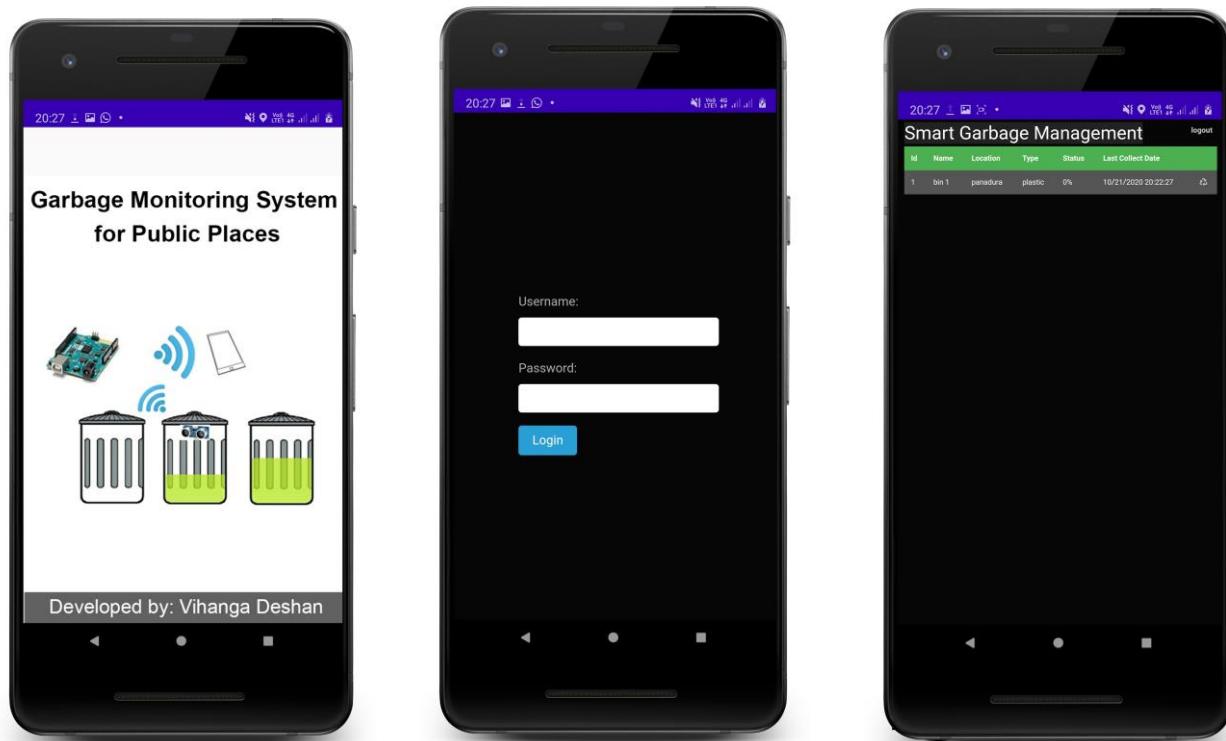
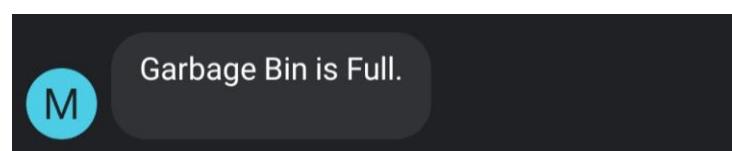


Figure 65 Testing 3

4. Mobile App testing



5. Check if the SMS is received



CHAPTER 04

4. System Implementation and Documentation

4.1 System implementation

Cloud Sever Installation

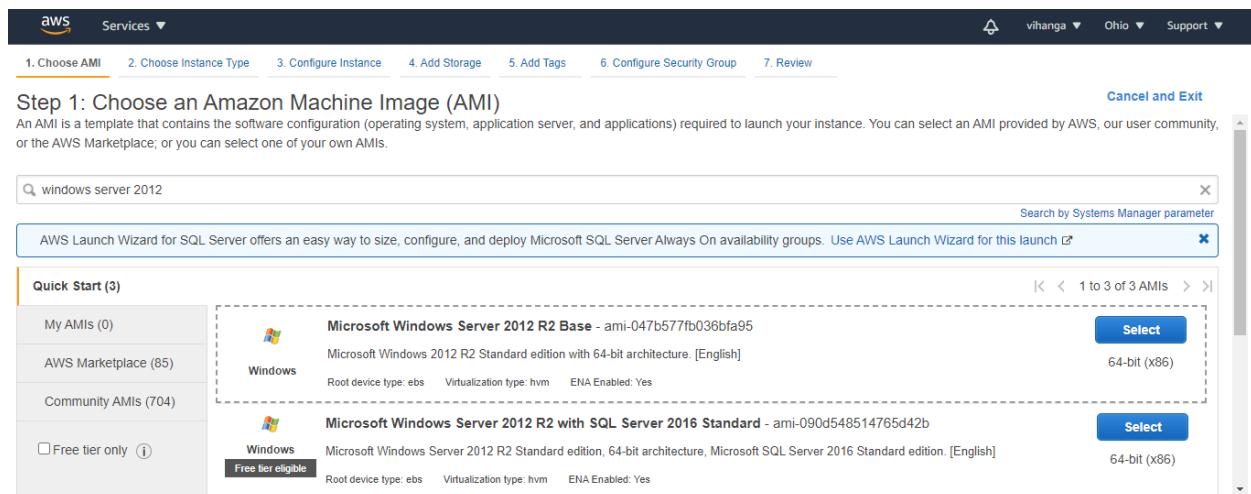


Figure 66 Cloud server installation 1

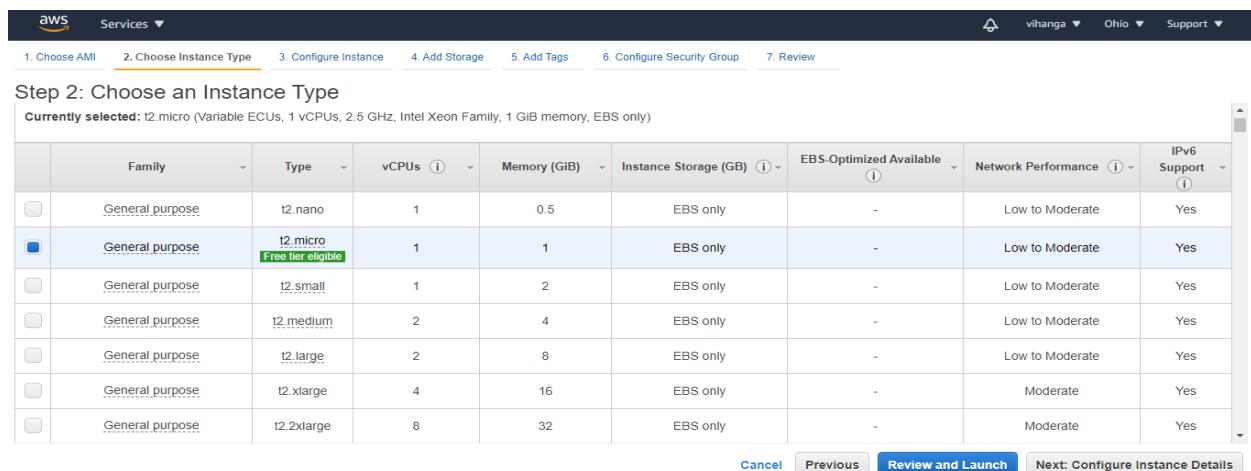


Figure 67 Cloud server installation 2

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	<input type="text" value="1"/>	Launch into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	vpc-84d279ef (default)	<input type="button" value="Create new VPC"/>
Subnet	No preference (default subnet in any Availability Zone)	<input type="button" value="Create new subnet"/>
Auto-assign Public IP	Use subnet setting (Enable)	
Placement group	<input type="checkbox"/> Add instance to placement group	
Capacity Reservation	Open	
Domain join directory	No directory	<input type="button" value="Create new directory"/>
IAM role	None	<input type="button" value="Create new IAM role"/>

Cancel Previous Review and Launch Next: Add Storage

Figure 68 Cloud server installation 3

Step 3: Configure Instance Details

Shutdown behavior	Stop
Stop - Hibernate behavior	<input type="checkbox"/> Enable hibernation as an additional stop behavior
Enable termination protection	<input type="checkbox"/> Protect against accidental termination
Monitoring	<input type="checkbox"/> Enable CloudWatch detailed monitoring <small>Additional charges apply.</small>
Tenancy	Shared - Run a shared hardware instance
Elastic Graphics	<input type="checkbox"/> Add Graphics Acceleration <small>Additional charges apply.</small>
Credit specification	<input type="checkbox"/> Unlimited <small>Additional charges may apply</small>

Advanced Details

Cancel Previous Review and Launch Next: Add Storage

Figure 69 Cloud server installation 4

Step 3: Configure Instance Details

Elastic Graphics Add Graphics Acceleration
Additional charges apply.

Credit specification Unlimited
Additional charges may apply

Advanced Details

Metadata accessible Enabled

Metadata version V1 and V2 (token optional)

Metadata token response hop limit 1

User data As text As file Input is already base64 encoded
(Optional)

Cancel Previous Review and Launch Next: Add Storage

Figure 70 Cloud server installation 5

Define Disk Capacity

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-00a7ef94b2114a177	30	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

Cancel Previous Review and Launch Next: Add Tags

Figure 71 Cloud server installation 6

Configuration Firewall rules

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more about Amazon EC2 security groups.](#)

Assign a security group: Create a new security group
 Select an existing security group

Security group name: launch-wizard-1

Description: launch-wizard-1 created 2020-10-01T21:27:17.462+05:30

Type	Protocol	Port Range	Source	Description
RDP	TCP	3389	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Add Rule

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Buttons: Cancel, Previous, Review and Launch

Figure 72 Cloud server installation 7

Security group rules:

Type	Protocol	Port Range	Source	Description
RDP	TCP	3389	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP	TCP	3001	Custom 0.0.0.0/0	Node Js Client Access
Custom TCP	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Add Rule

Figure 73 Cloud server installation 8

Step 7: Review Instance Launch

User data

- Assign Public IP Use subnet setting (Enable)
- Assign IPv6 IP Use subnet setting (Enable)
- Assign Carrier IP

Storage

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypt
Root	/dev/sda1	snap-00a7ef94b2114a177	30	gp2	100 / 3000	N/A	Yes	Not E...

Tags

Key	Value	Instances	Volumes
This resource currently has no tags			

Buttons: Cancel, Previous, Launch

Figure 74 Cloud server installation 9

Create VM RDP Access Key

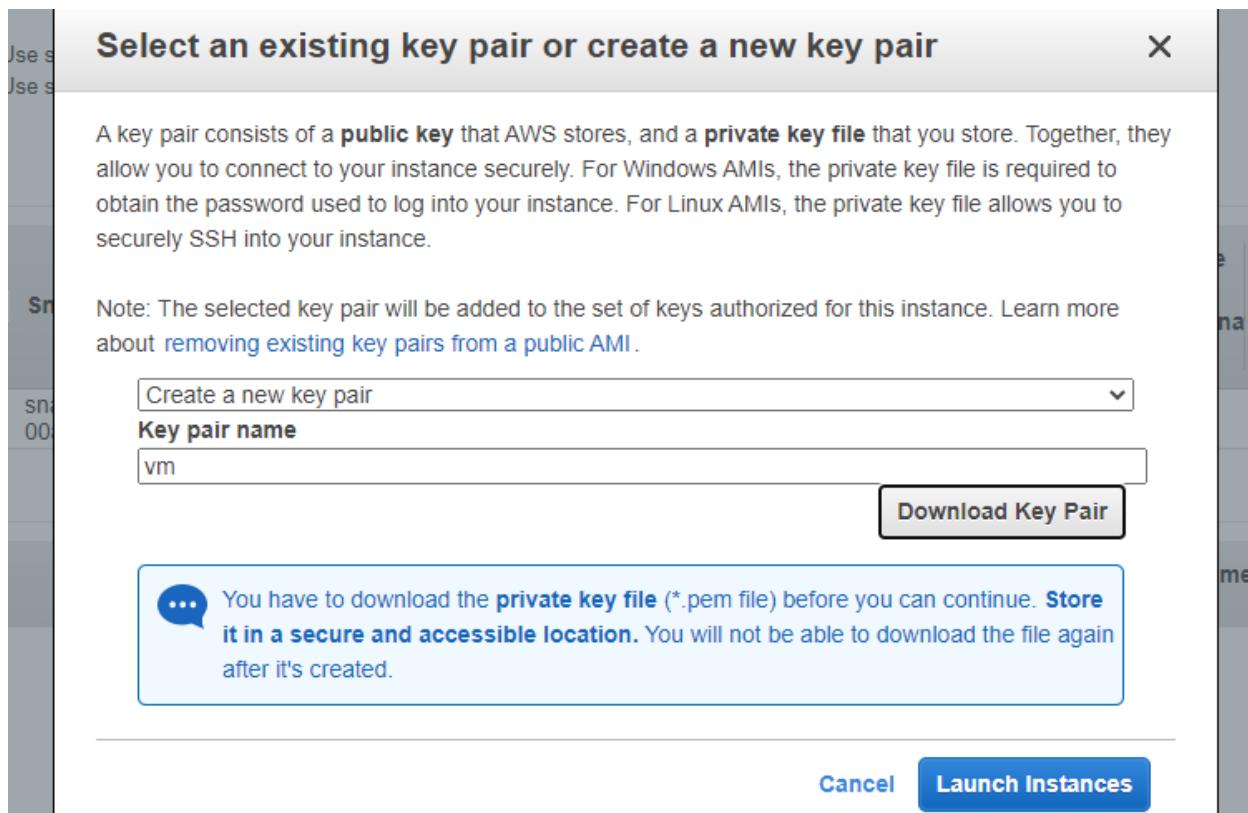


Figure 75 Create VM RDP Access Key 1

Your instances are now launching
The following instance launches have been initiated: i-0afe956afb7eb5d2a [View launch log](#)

Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances
Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.
Click [View Instances](#) to monitor your instances' status. Once your instances are in the **running** state, you can [connect](#) to them from the Instances screen. [Find out](#) how to connect to your instances.

Here are some helpful resources to get you started

- [How to connect to your Windows instance](#)
- [Amazon EC2: User Guide](#)

Figure 76 Create VM RDP Access Key 2

Instance summary for i-0afe956afb7eb5d2a

Instance ID	i-0afe956afb7eb5d2a	Public IPv4 address	52.14.107.220 open address	Private IPv4 addresses	172.31.18.29
Instance state	Running	Public IPv4 DNS	ec2-52-14-107-220.us-east-2.compute.amazonaws.com open address	Private IPv4 DNS	ip-172-31-18-29.us-east-2.compute.internal
Instance type	t2.micro	Elastic IP addresses	-	VPC ID	vpc-84d279ef
IAM Role	-	Subnet ID	subnet-94f7feef		

AWS Compute Optimizer
Opt-in to AWS Compute Optimizer for recommendations. [Learn more](#)

Figure 77 Create VM RDP Access Key 1

Get Administrator Password using decryption key

Key pair associated with this instance
vm

Browse to your key pair:
[Browse](#)

vm.pem
1.7KB

Or copy and paste the contents of the key pair below:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEaselmkF7mVjtc6tzACwjCPRWtHk+a91p9smxBXLeI9xXrH
MvVxz7RjyfRtkm7V8gGev2WxTz76ftarzPw7zXobNLIoOGFmVGPx1aNoJcle
s7b/lIjE+21fwOM/bmaBf8sxk3N0UdolP0mJVZMakstBjuy/zlj63ZqqVa3IKsZ
4pSN3lPTHG9gfVCve55tyleP2c4Nz6j1r0GfV59jMSQaHlmijnwmQa+VMWXwHr
tSRtAWVCPC70Au4/1puJbbyTpwtKWKnkNPFDMDf8GnxgN6Q34DKz2RCQGXPNth
pYM8s424ywjr/uxEx3JLP5n5DW7131EZJy256QIDAQABoIBAD34u9dCDP1vrcoO
gTkUBG3OTLGHTm5uYl8ISnnRnglQ0ystl1Srj4icZy2mPgDi4RHD7dnP3yb4CAu
-----END RSA PRIVATE KEY-----
```

[Cancel](#) [Decrypt Password](#)

When prompted, connect to your instance using the following details:

Public DNS

ec2-52-14-107-220.us-east-2.compute.amazonaws.com

User name

[Administrator](#)

Password [Get password](#)

If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.

Figure 78 Create VM RDP Access Key 4

Access server using RDP Client

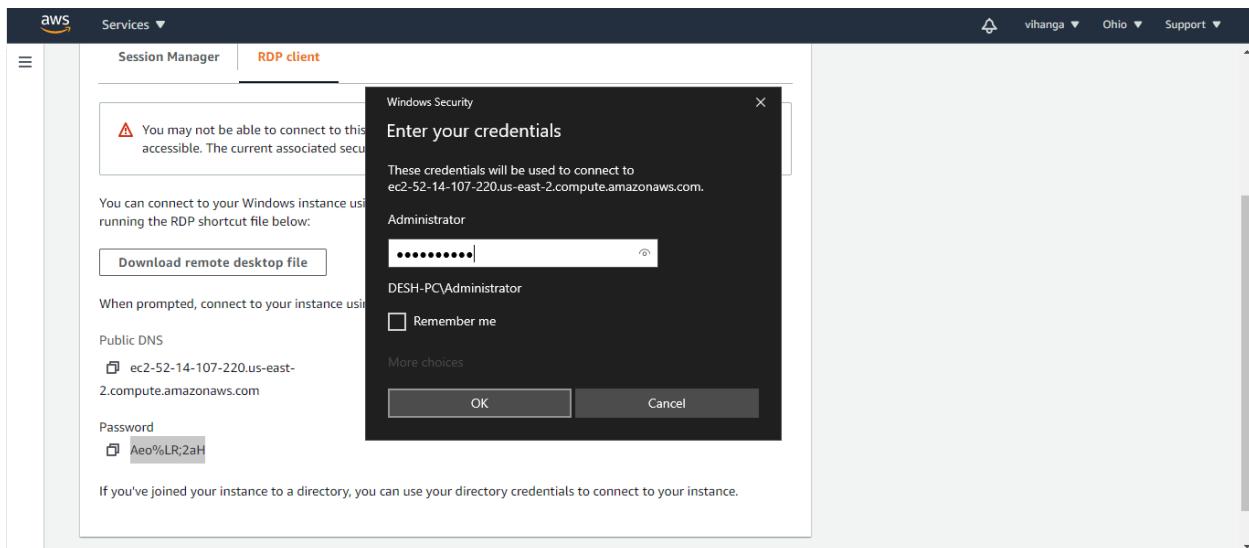


Figure 79 Access server using RDP Client 1

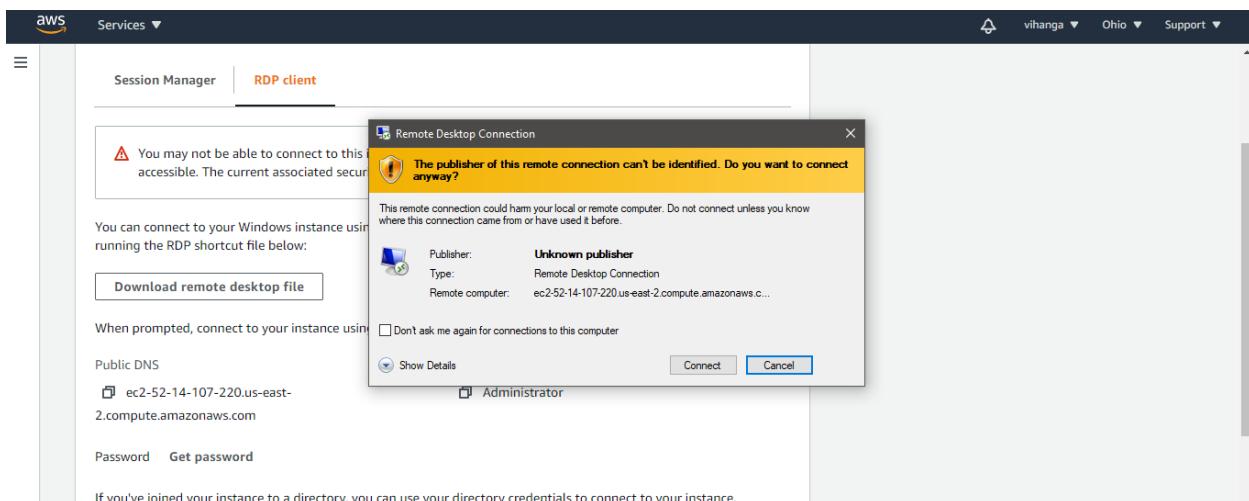


Figure 80 Access server using RDP Client 2

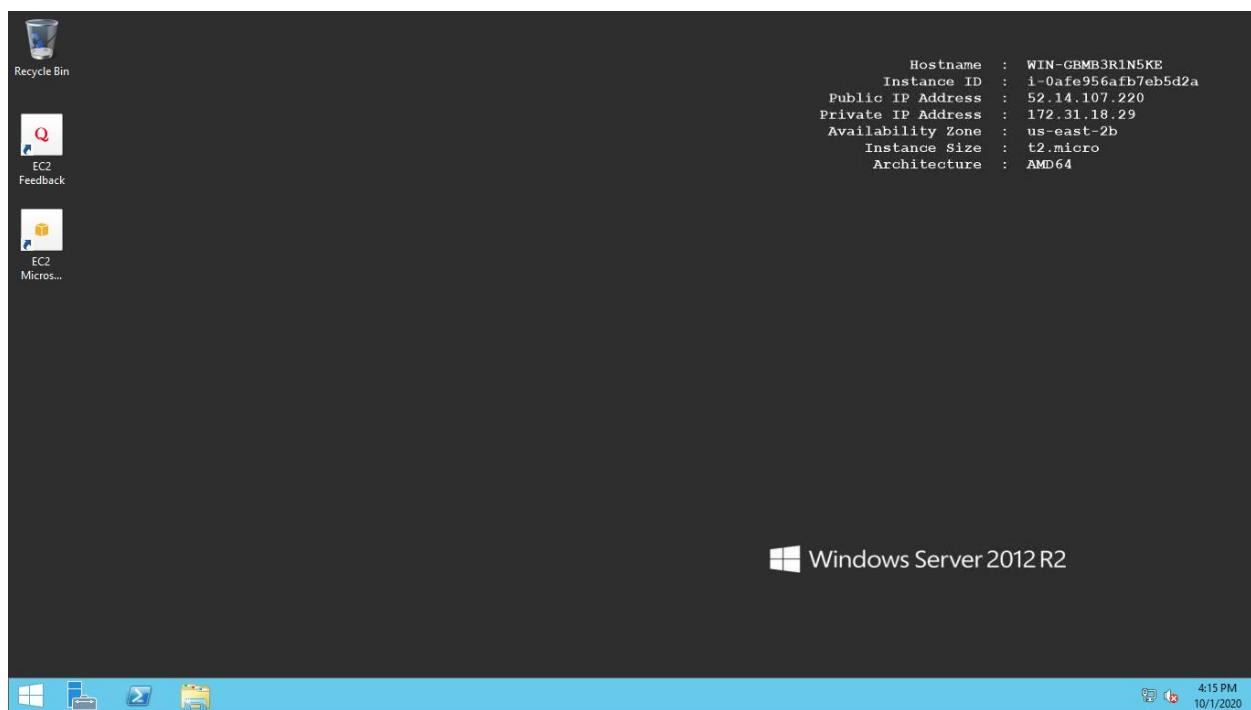


Figure 81 Windows server1

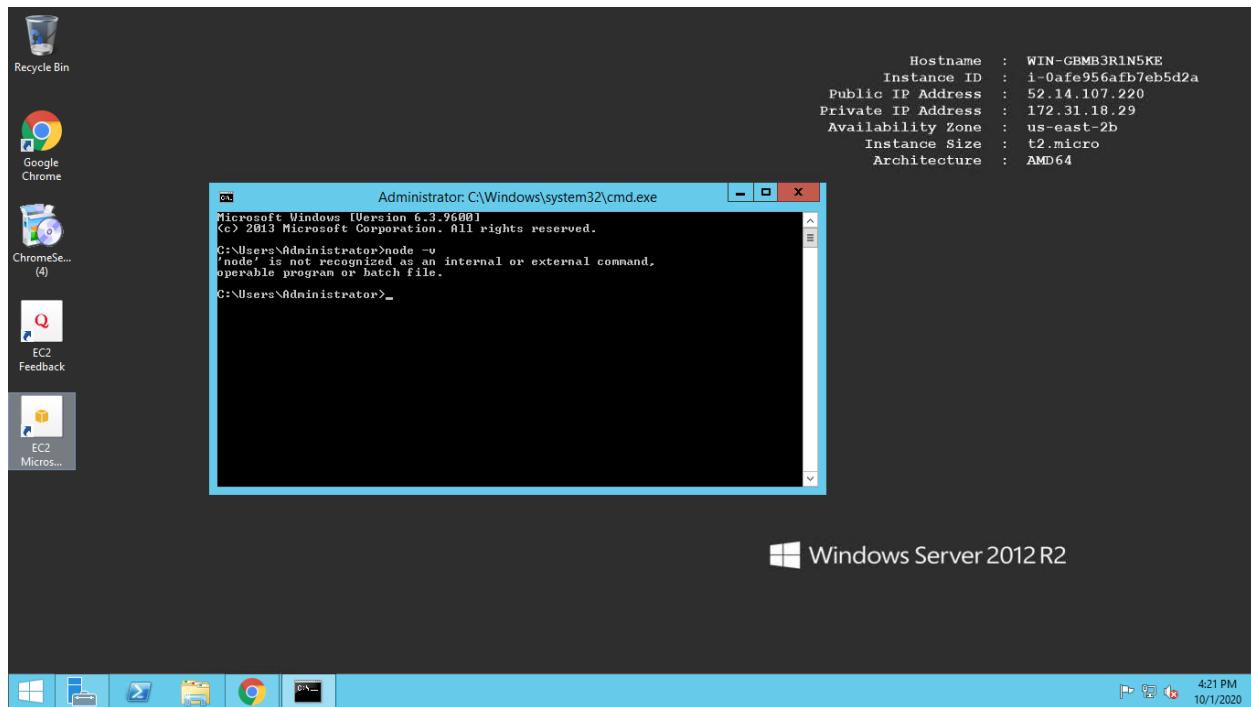


Figure 82 Windows server2

Node JS Installation

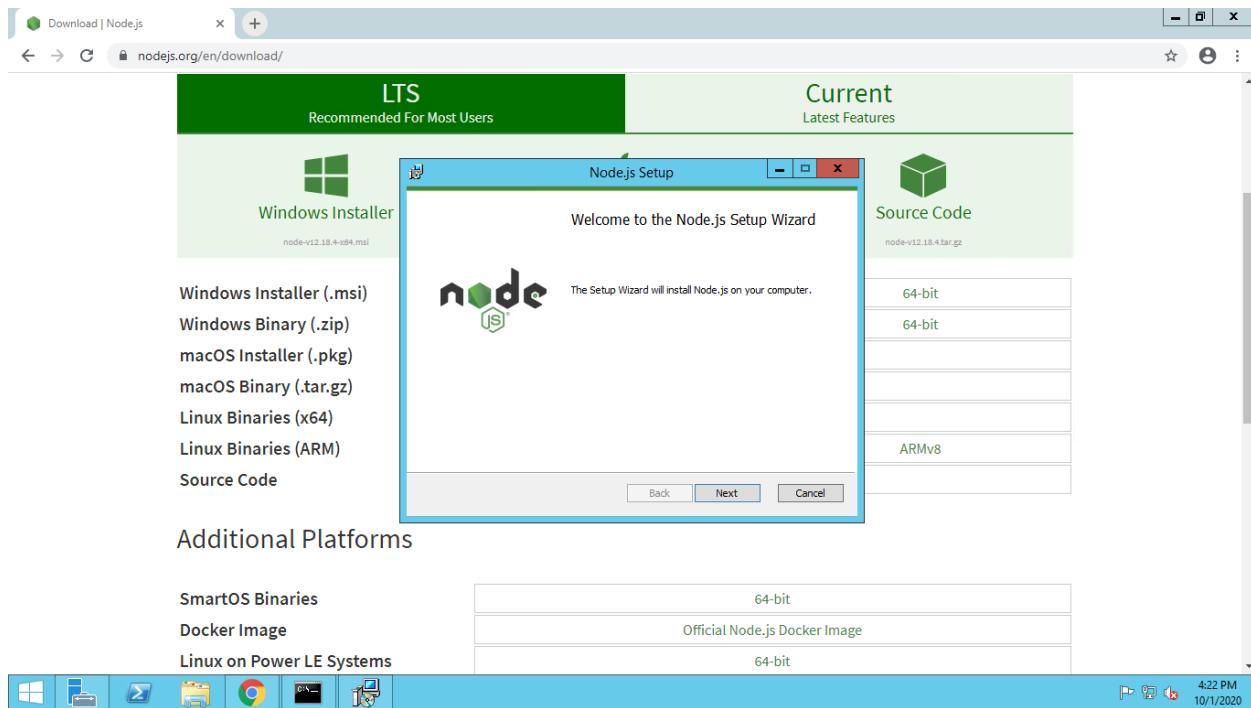


Figure 83 Node JS installation 1

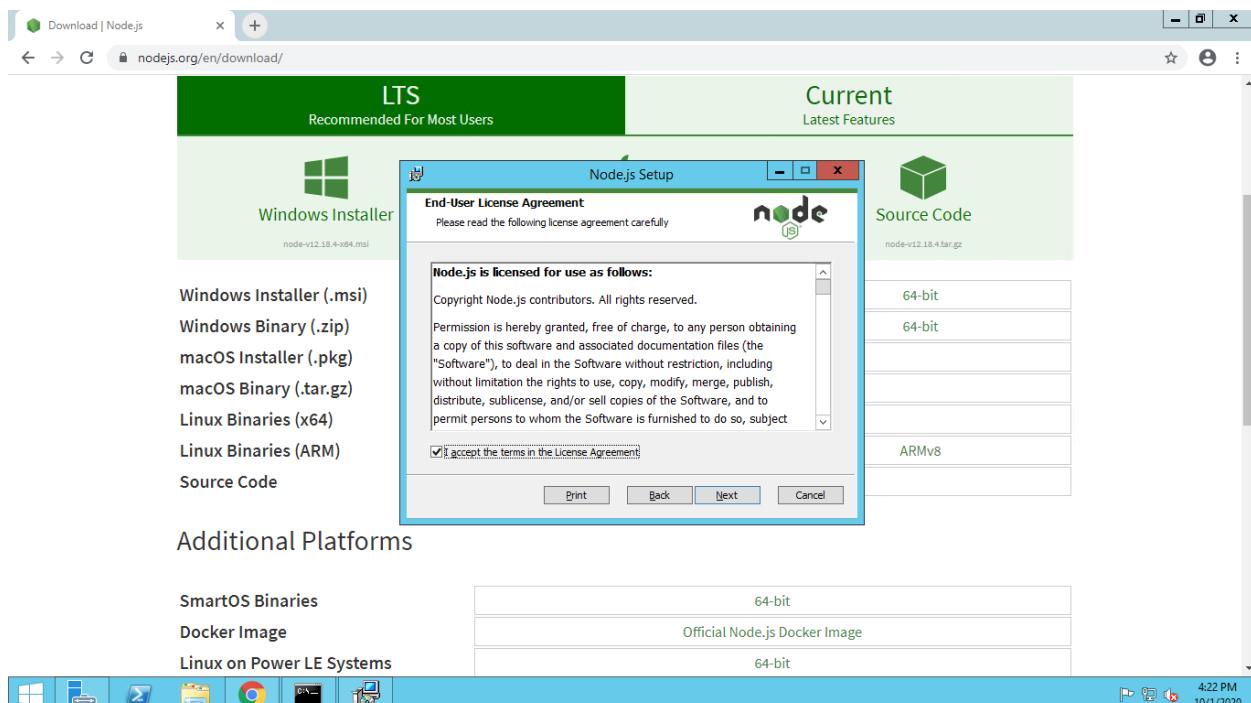


Figure 84 Node JS installation 2

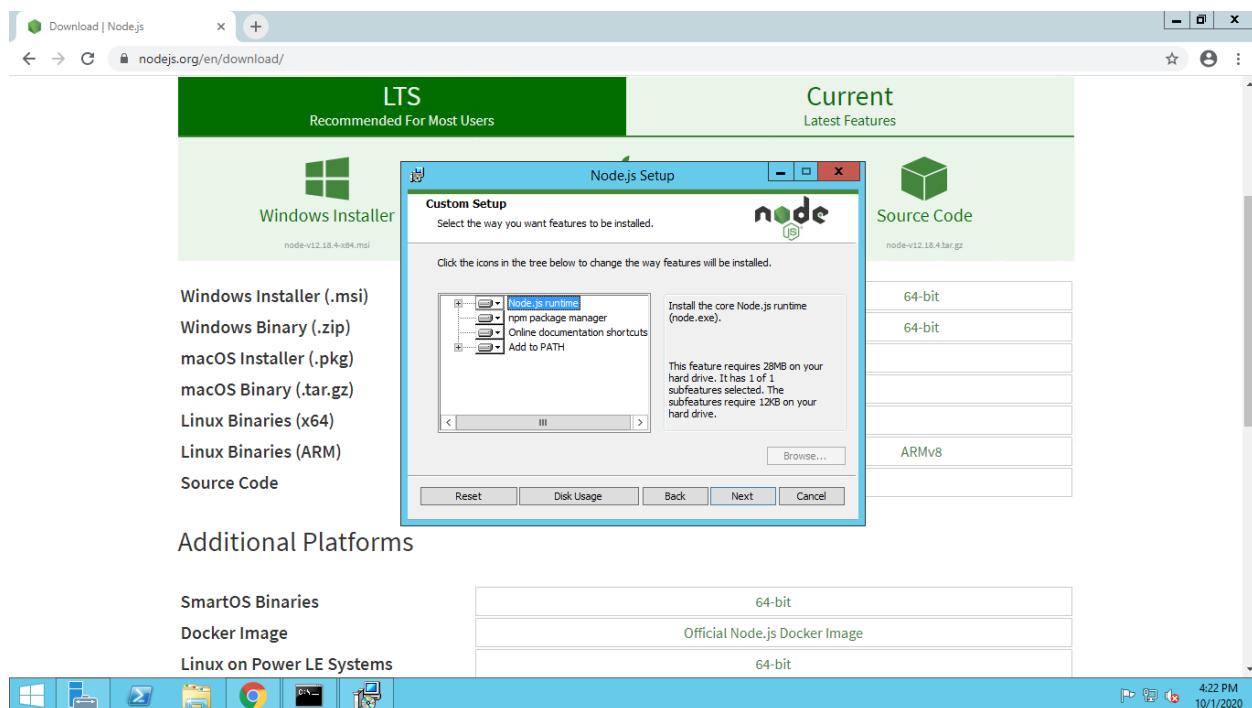


Figure 85 Node JS installation 3

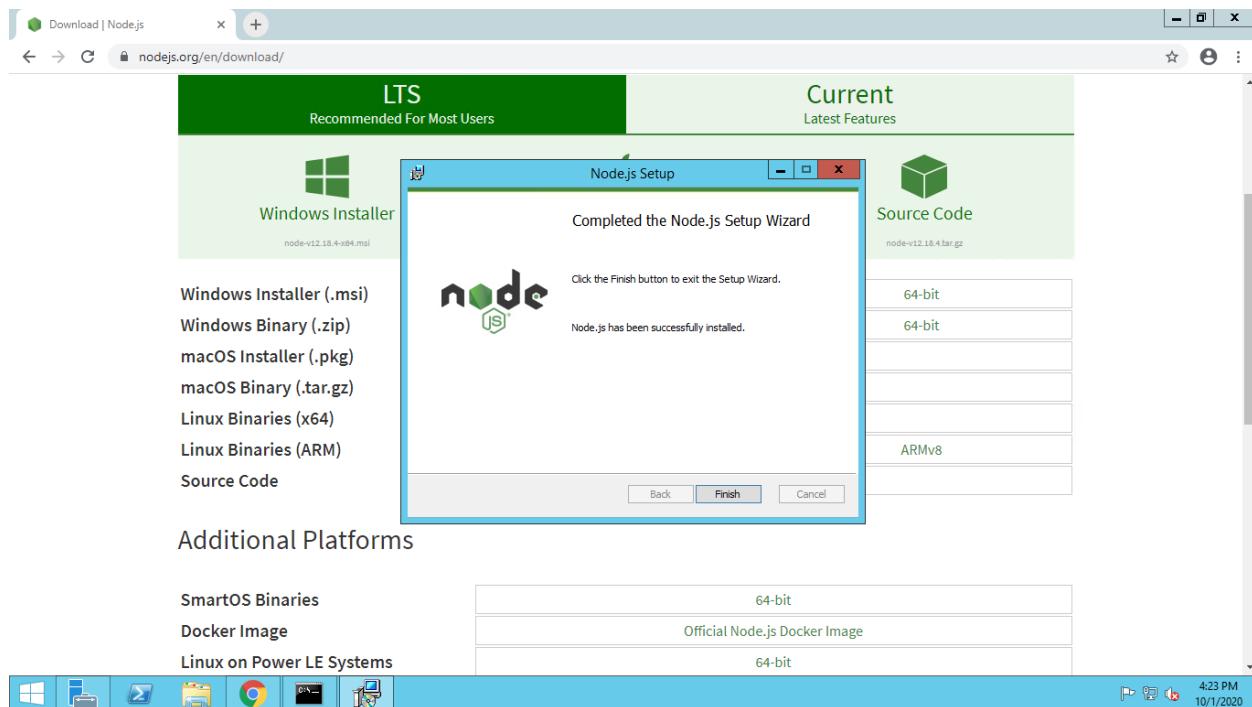


Figure 86 Node JS installation 1

Verify Installation

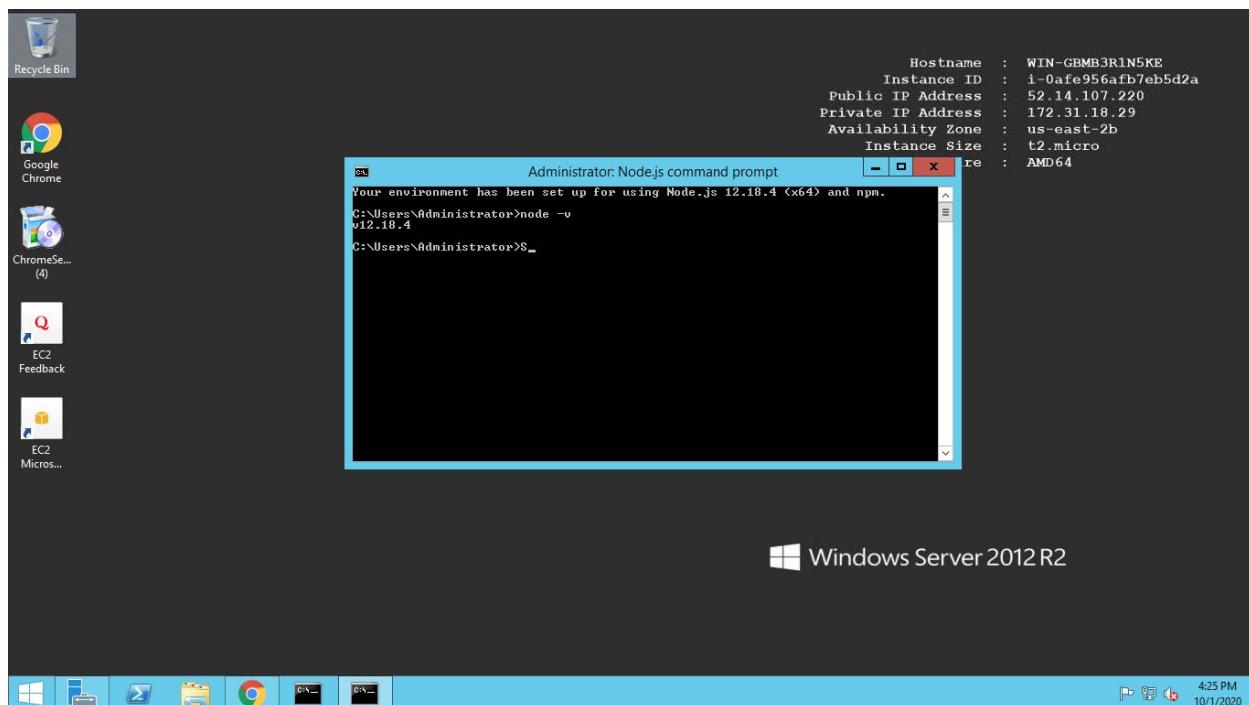


Figure 87 Verify Installation

Upload server files and start the server

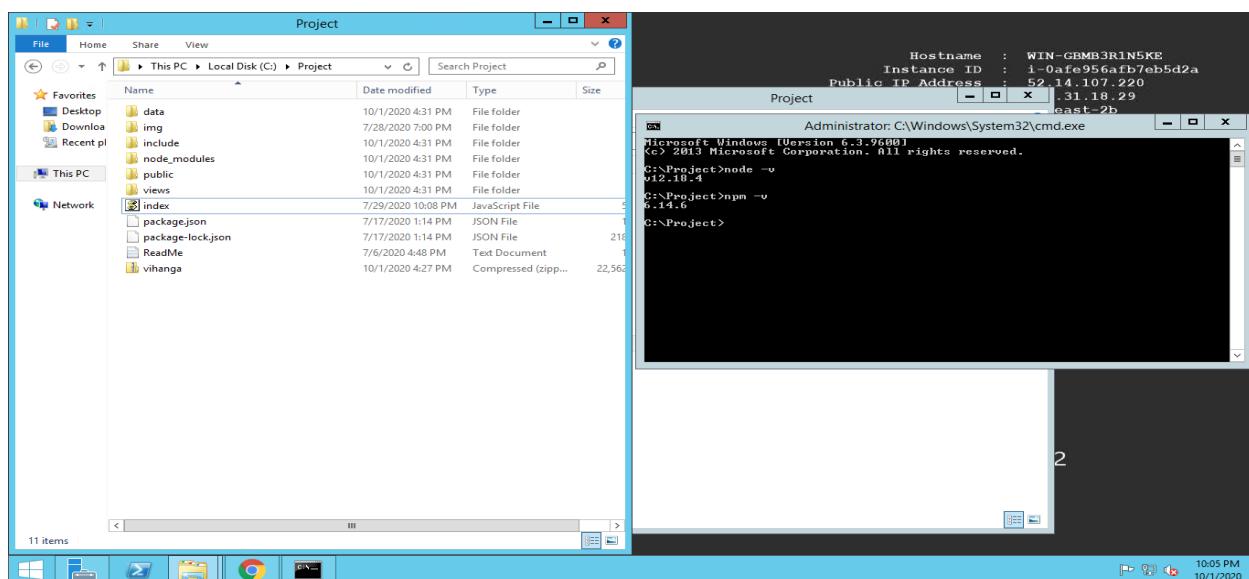


Figure 88 Upload server files and start the server 1

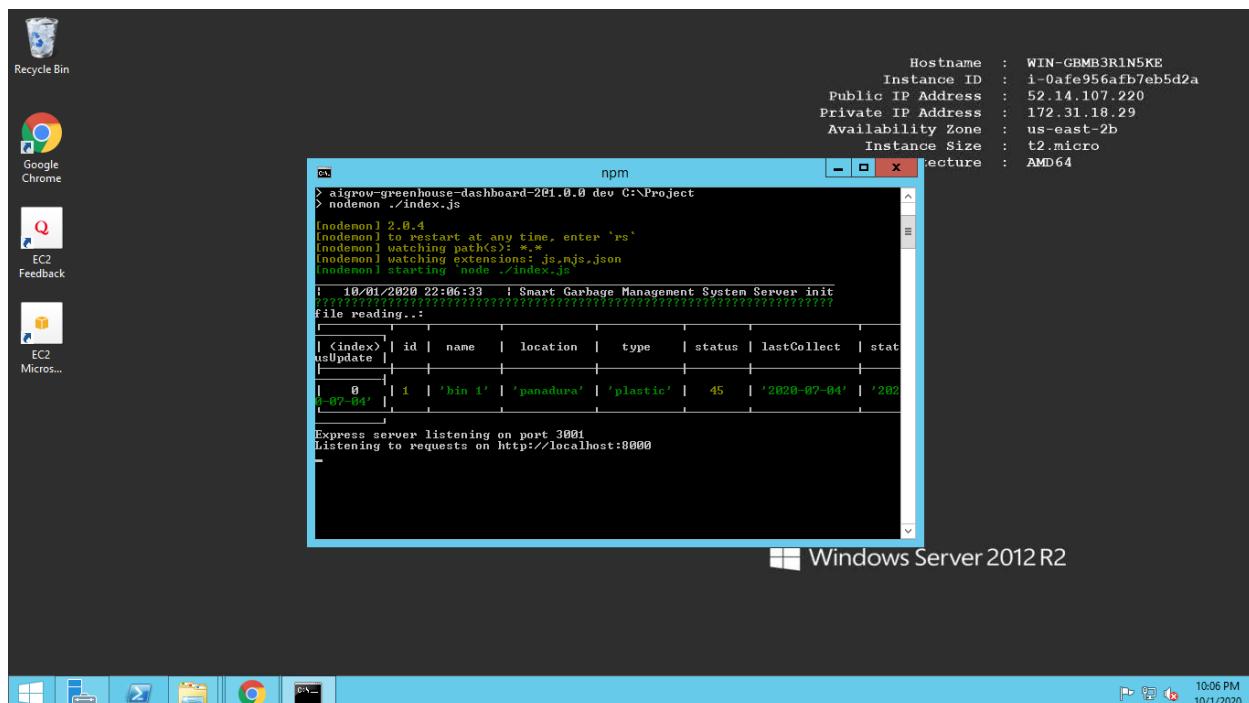


Figure 89 Upload server files and start the server 2

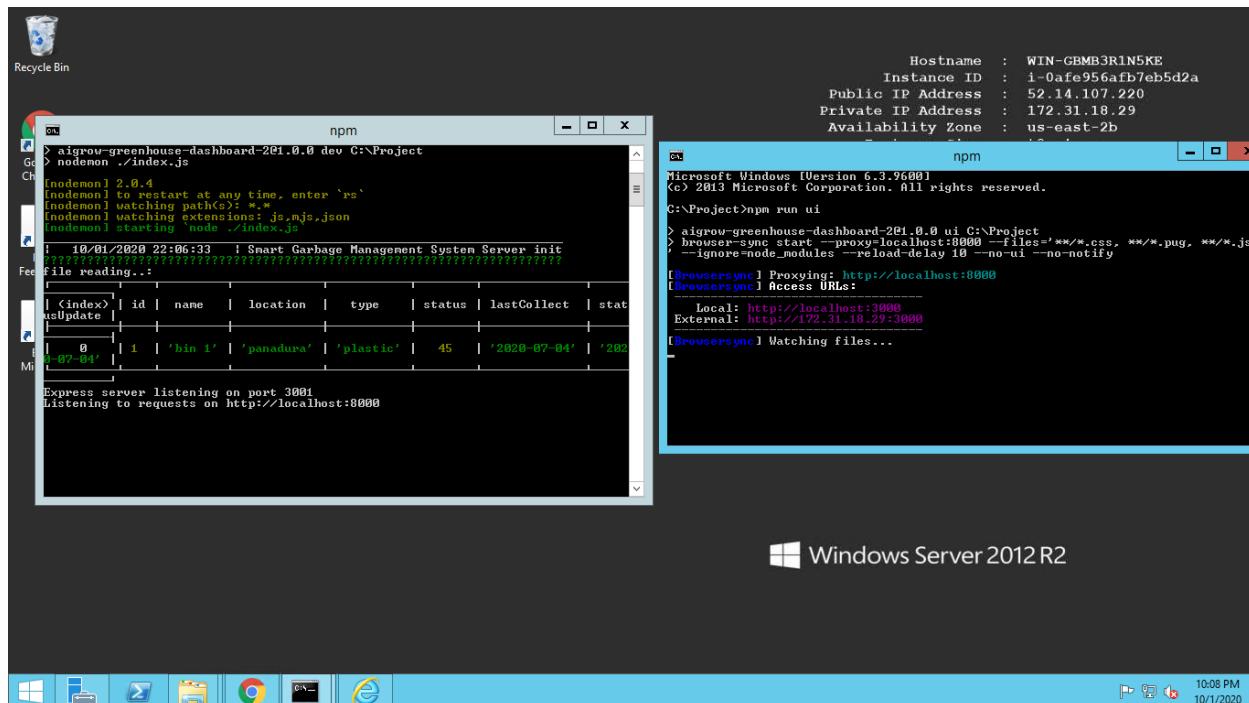


Figure 90 Upload server files and start the server 3

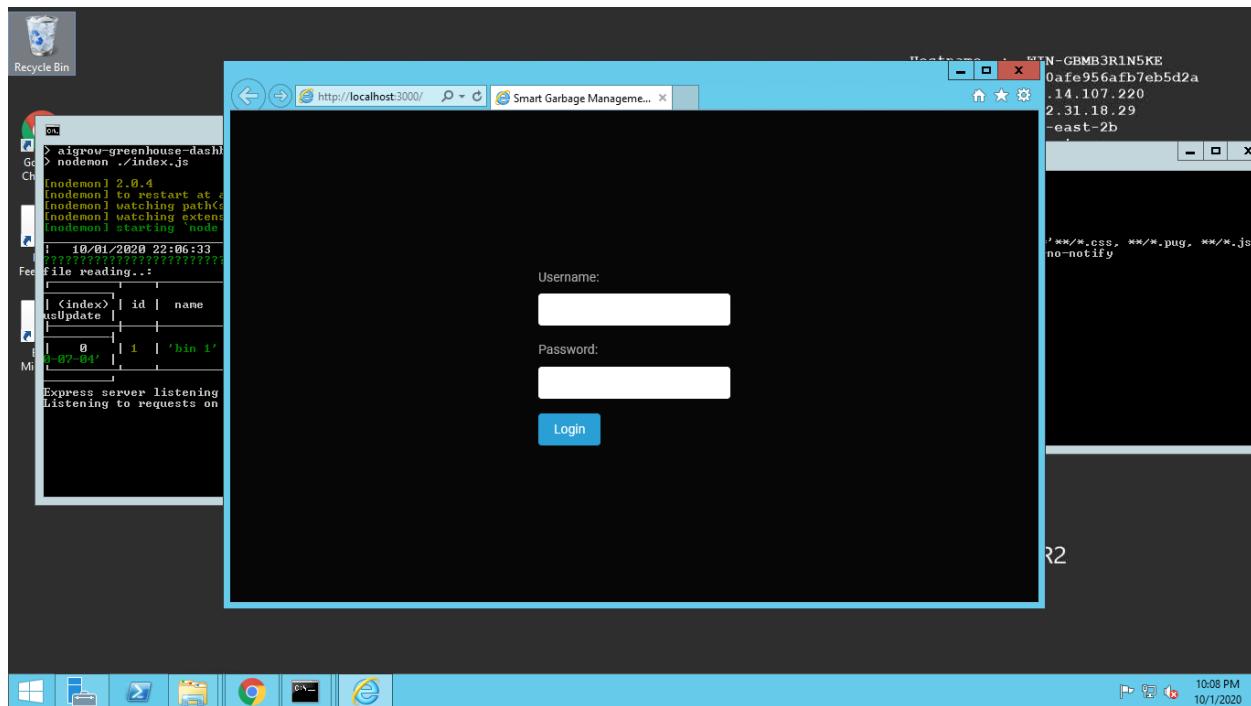


Figure 91 Upload server files and start the server 4

Client side Installation

Install Node.js and npm on Raspberry Pi

```

at eval (eval at <anonymous> (/home/pi/Desktop/raspberry/app.js:3:51), <anonymous>:1:15)
at Object.<anonymous> (/home/pi/Desktop/raspberry/app.js:3:1)
at Module._compile (module.js:409:26)
at Object.Module._extensions..js (module.js:416:10)
at Module.load (module.js:343:32)
at Function.Module._load (module.js:300:12)
pi@raspberrypi:~/Desktop/raspberry $ pm i express
-bash: pm: command not found
pi@raspberrypi:~/Desktop/raspberry $ npm i express
-bash: npm: command not found
pi@raspberrypi:~/Desktop/raspberry $ npm -v
-bash: npm: command not found
pi@raspberrypi:~/Desktop/raspberry $ node -v
v4.8.2
pi@raspberrypi:~/Desktop/raspberry $ sudo apt-get update
Get:1 http://archive.raspberrypi.org/debian stretch InRelease [25.4 kB]
Get:2 http://raspbian.raspberrypi.org/raspbian stretch InRelease [15.0 kB]
Get:3 http://raspbian.raspberrypi.org/raspbian stretch/main armhf Packages [11.7 MB]
Get:4 http://archive.raspberrypi.org/debian stretch/main armhf Packages [192 kB]
Fetched 11.9 MB in 34s (349 kB/s)
Reading package lists... Done
pi@raspberrypi:~/Desktop/raspberry $

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Figure 92 Client side Installation 1

```

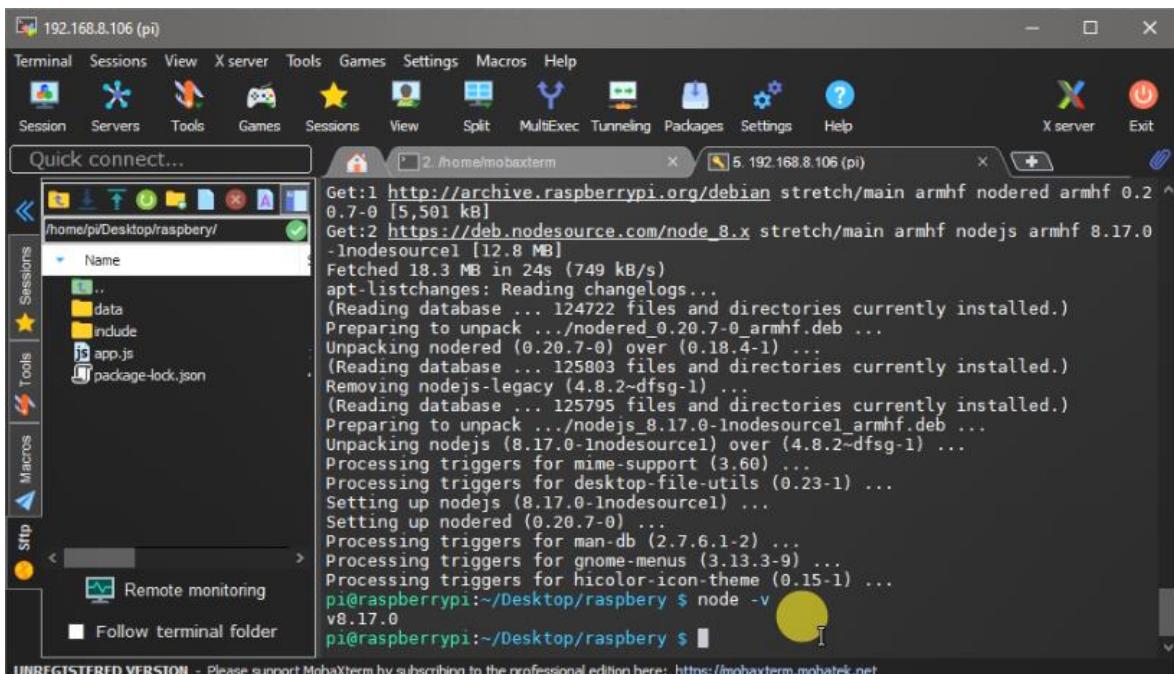
sudo apt-get update && sudo apt-get install yarn
pi@raspberrypi:~/Desktop/raspberry $ sudo apt-get install -y nodejs
Reading package lists... Done
Building dependency tree...
Reading state information... Done
The following package was automatically installed and is no longer required:
  libuv1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  nodered
The following packages will be REMOVED:
  nodejs-legacy
The following packages will be upgraded:
  nodejs nodered
2 upgraded, 0 newly installed, 1 to remove and 346 not upgraded.
Need to get 18.3 MB of archives.
After this operation, 121 MB of additional disk space will be used.
Get:1 http://archive.raspberrypi.org/debian stretch/main armhf nodered armhf 0.2.0-0 [5,501 kB]
Get:2 https://deb.nodesource.com/node_8.x stretch/main armhf nodejs armhf 8.17.0-1nodesource1 [12.8 MB]
22% [1 nodered 2,164 kB/5,501 kB 39%] [2 nodejs 2,866 kB/12.8 MB 22%]

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Figure 93 Client side Installation 2

Verify the Node JS Installation



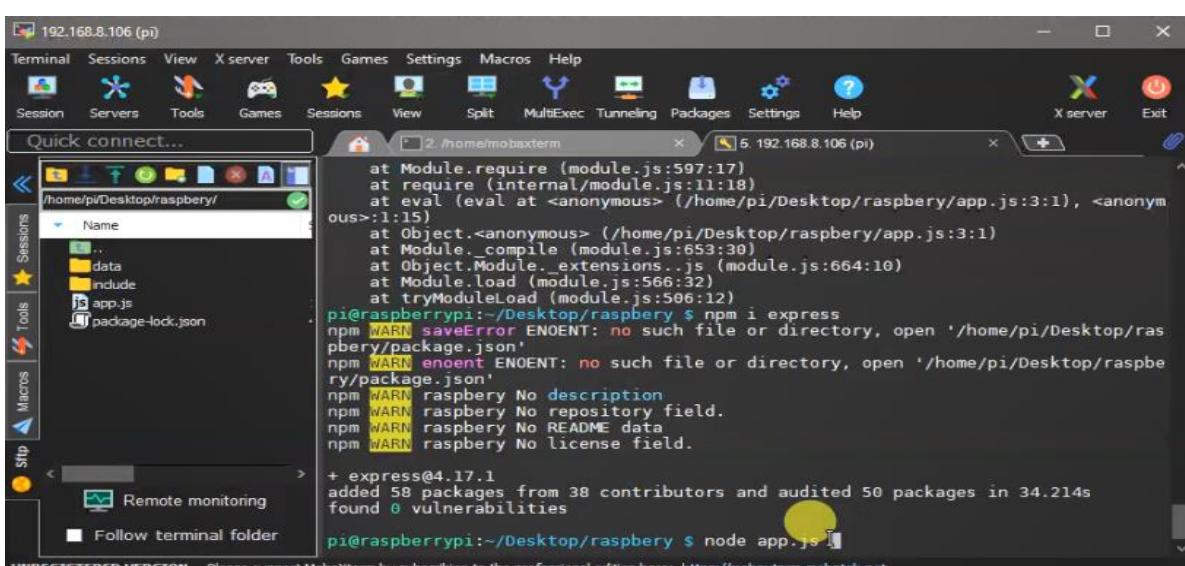
The screenshot shows the MobaXterm interface on a Raspberry Pi. The terminal window displays the output of the command `node -v`, which shows the version as v8.17.0. The left sidebar shows the file structure of the current directory, which contains `..`, `data`, `include`, `app.js`, and `package-lock.json`.

```
Get:1 http://archive.raspberrypi.org/debian stretch/main armhf nodered armhf 0.2+0.7-0 [5,501 kB]
Get:2 https://deb.nodesource.com/node_8.x stretch/main armhf nodejs armhf 8.17.0-1nodesource1 [12.8 MB]
Fetched 18.3 MB in 24s (749 kB/s)
apt-listchanges: Reading changelogs...
(Reading database ... 124722 files and directories currently installed.)
Preparing to unpack .../nodered_0.20.7-0_armhf.deb ...
Unpacking nodered (0.20.7-0) over (0.18.4-1) ...
(Reading database ... 125803 files and directories currently installed.)
Removing nodejs-legacy (4.8.2~dfsg-1) ...
(Reading database ... 125795 files and directories currently installed.)
Preparing to unpack .../nodejs_8.17.0-1nodesource1_armhf.deb ...
Unpacking nodejs (8.17.0-1nodesource1) over (4.8.2~dfsg-1) ...
Processing triggers for mime-support (3.60) ...
Processing triggers for desktop-file-utils (0.23-1) ...
Setting up nodejs (8.17.0-1nodesource1) ...
Setting up nodered (0.20.7-0) ...
Processing triggers for man-db (2.7.6.1-2) ...
Processing triggers for gnome-menus (3.13.3-9) ...
Processing triggers for hicolor-icon-theme (0.15-1) ...
pi@raspberrypi:~/Desktop/raspberry $ node -v
v8.17.0
pi@raspberrypi:~/Desktop/raspberry $
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Figure 94 Verify the Node JS Installation

Start the node JS app



The screenshot shows the MobaXterm interface on a Raspberry Pi. The terminal window displays the output of the command `npm i express`, which installs the Express.js framework. The left sidebar shows the file structure of the current directory, which contains `..`, `data`, `include`, `app.js`, and `package-lock.json`. The command `node app.js` is then run, indicating the start of the application.

```
at Module.require (module.js:597:17)
at require (internal/module.js:11:18)
at eval (eval at <anonymous> (/home/pi/Desktop/raspberry/app.js:3:1), <anonymous>:15)
at Object.<anonymous> (/home/pi/Desktop/raspberry/app.js:3:1)
at Module._compile (module.js:653:30)
at Object.Module._extensions..js (module.js:664:10)
at Module.load (module.js:566:32)
at tryModuleLoad (module.js:506:12)
pi@raspberrypi:~/Desktop/raspberry $ npm i express
npm WARN saveError ENOENT: no such file or directory, open '/home/pi/Desktop/raspberry/package.json'
npm WARN enoent ENOENT: no such file or directory, open '/home/pi/Desktop/raspberry/package.json'
npm WARN raspberry No description
npm WARN raspberry No repository field.
npm WARN raspberry No README data
npm WARN raspberry No license field.

+ express@4.17.1
added 58 packages from 38 contributors and audited 50 packages in 34.214s
found 0 vulnerabilities
pi@raspberrypi:~/Desktop/raspberry $ node app.js
```

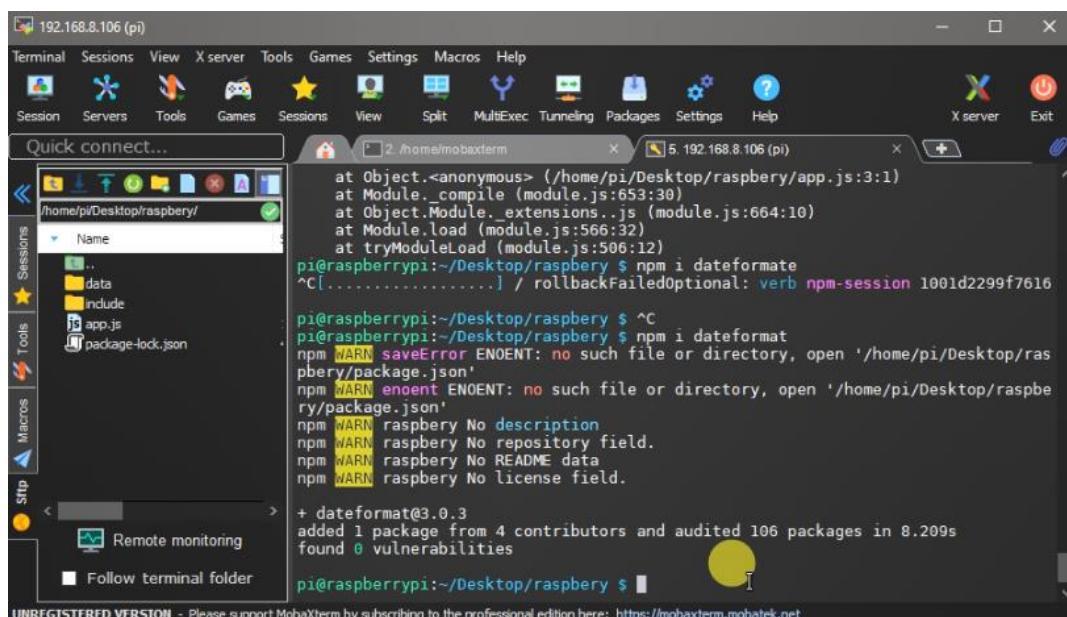
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Figure 95 Start the node JS app

Installation Node JS Packages Installation

Required Packages –

- npm i dateformat
- npm i request
- npm i delay
- npm i jsonfile
- npm i socket.io-client



The screenshot shows a terminal window titled '192.168.8.106 (pi)' running on a Raspberry Pi. The terminal interface includes a menu bar with 'Terminal', 'Sessions', 'View', 'X server', 'Tools', 'Games', 'Settings', 'Macros', 'Help', and various icons for session management. The main window displays a command-line session. The user has run the command `npm i dateformat`. The output shows several warning messages from npm, such as 'ENOENT: no such file or directory' for files like 'package.json', 'README', and 'LICENSE'. It also indicates that the package was added successfully with version 3.0.3, along with audit information.

```

at Object.<anonymous> (/home/pi/Desktop/raspberry/app.js:3:1)
at Module._compile (module.js:653:30)
at Object.Module._extensions..js (module.js:664:10)
at Module.load (module.js:566:32)
at tryModuleLoad (module.js:506:12)
pi@raspberrypi:~/Desktop/raspberry $ npm i dateformat
^C[.....] / rollbackFailedOptional: verb npm-session 1001d2299f7616
pi@raspberrypi:~/Desktop/raspberry $ ^C
pi@raspberrypi:~/Desktop/raspberry $ npm i dateformat
npm WARN saveError ENOENT: no such file or directory, open '/home/pi/Desktop/raspberry/package.json'
npm WARN enoent ENOENT: no such file or directory, open '/home/pi/Desktop/raspberry/package.json'
npm WARN raspberry No description
npm WARN raspberry No repository field.
npm WARN raspberry No README data
npm WARN raspberry No license field.

+ dateformat@3.0.3
added 1 package from 4 contributors and audited 106 packages in 8.209s
found 0 vulnerabilities
pi@raspberrypi:~/Desktop/raspberry $ [REDACTED]

```

Figure 96 Start the node JS app1

The screenshot shows a MobaXterm session titled '192.168.8.106 (pi)'. The terminal window displays the command 'node app.js' being run. The output shows npm errors for 'request' and 'delay' packages, indicating they are deprecated and missing files like README and license. The session also shows a file browser sidebar with files like 'app.js' and 'package-lock.json'.

```

at Module.load (module.js:566:32)
at tryModuleLoad (module.js:506:12)
pi@raspberrypi:~/Desktop/raspberry $ npm i request
npm [WARN] deprecated request@2.88.2: request has been deprecated, see https://git
hub.com/request/request/issues/3142
npm [WARN] saveError ENOENT: no such file or directory, open '/home/pi/Desktop/ras
pberry/package.json'
npm [WARN] enoent ENOENT: no such file or directory, open '/home/pi/Desktop/raspbe
ry/package.json'
npm [WARN] raspberry No description
npm [WARN] raspberry No repository field.
npm [WARN] raspberry No README data
npm [WARN] raspberry No license field.

+ request@2.88.2
added 43 packages from 52 contributors and audited 149 packages in 27.245s

1 package is looking for funding
  run `npm fund` for details

found 0 vulnerabilities

pi@raspberrypi:~/Desktop/raspberry $ node app.js

```

Figure 97 Start the node JS app 2

The screenshot shows a MobaXterm session titled '192.168.8.106 (pi)'. The terminal window displays the command 'node app.js' being run. The output shows npm errors for 'request' and 'delay' packages, indicating they are deprecated and missing files like README and license. The session also shows a file browser sidebar with files like 'app.js' and 'package-lock.json'.

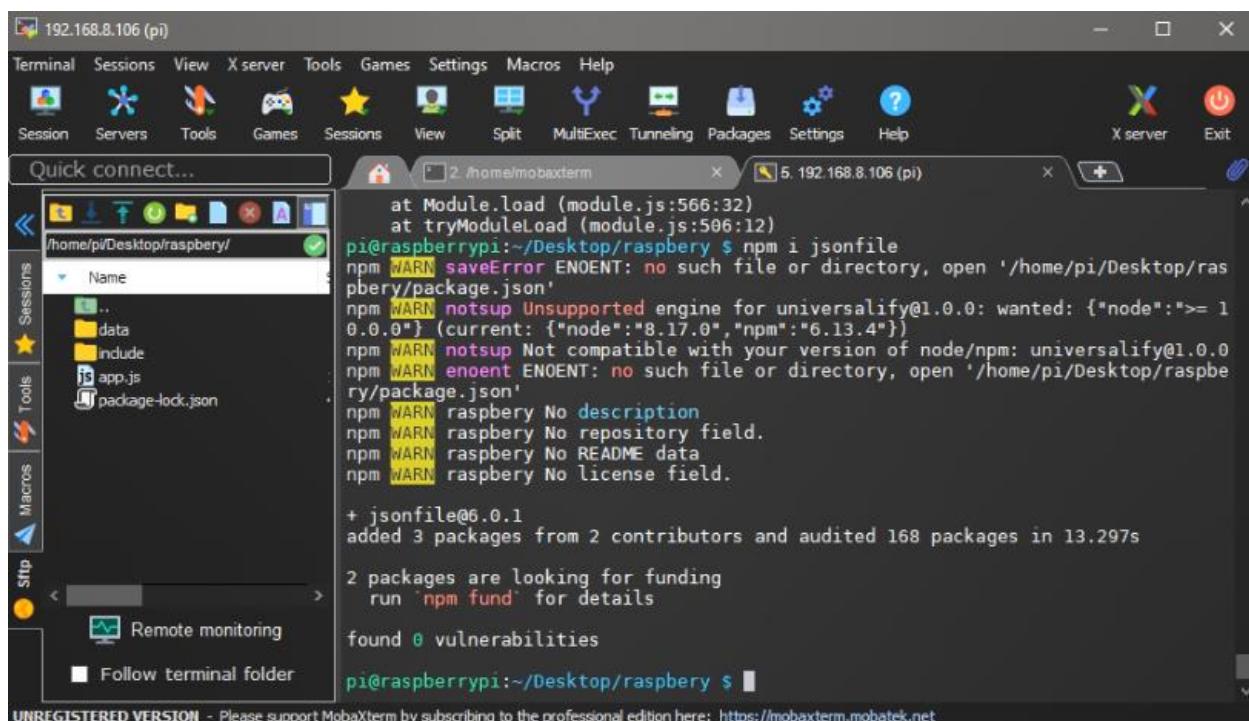
```

at Function.Module._resolveFilename (module.js:548:15)
at Function.Module._load (module.js:475:25)
at Module.require (module.js:597:17)
at require (internal/module.js:11:18)
at eval (eval at <anonymous> (/home/pi/Desktop/raspberry/app.js:3:1), <anonym
ous>:5:15)
at Object.<anonymous> (/home/pi/Desktop/raspberry/app.js:3:1)
at Module._compile (module.js:653:30)
at Object.Module._extensions..js (module.js:664:10)
at Module.load (module.js:566:32)
at tryModuleLoad (module.js:506:12)
pi@raspberrypi:~/Desktop/raspberry $ npm i delay
npm [WARN] saveError ENOENT: no such file or directory, open '/home/pi/Desktop/ras
pberry/package.json'
npm [WARN] enoent ENOENT: no such file or directory, open '/home/pi/Desktop/raspbe
ry/package.json'
npm [WARN] raspberry No description
npm [WARN] raspberry No repository field.
npm [WARN] raspberry No README data
npm [WARN] raspberry No license field.

+ delay@4.4.0
added 1 package from 1 contributor and audited 165 packages in 11.041s

```

Figure 98 Start the node JS app 3



The screenshot shows a MobaXterm window titled '192.168.8.106 (pi)'. The terminal session is running on a Raspberry Pi. The user has navigated to the directory '/home/pi/Desktop/raspberry/' and run the command 'npm i jsonfile'. The output of the command is displayed in the terminal window:

```

at Module._load (module.js:566:32)
at tryModuleLoad (module.js:506:12)
pi@raspberrypi:~/Desktop/raspberry $ npm i jsonfile
npm WARN saveError ENOENT: no such file or directory, open '/home/pi/Desktop/raspberry/package.json'
npm WARN notsup Unsupported engine for universalify@1.0.0: wanted: {"node":">>= 10.0.0"} (current: {"node":"8.17.0","npm":"6.13.4"})
npm WARN notsup Not compatible with your version of node/npm: universalify@1.0.0
npm ERR! enoent ENOENT: no such file or directory, open '/home/pi/Desktop/raspberry/package.json'
npm ERR! raspberry No description
npm ERR! raspberry No repository field.
npm ERR! raspberry No README data
npm ERR! raspberry No license field.

+ jsonfile@6.0.1
added 3 packages from 2 contributors and audited 168 packages in 13.297s

2 packages are looking for funding
  run `npm fund` for details

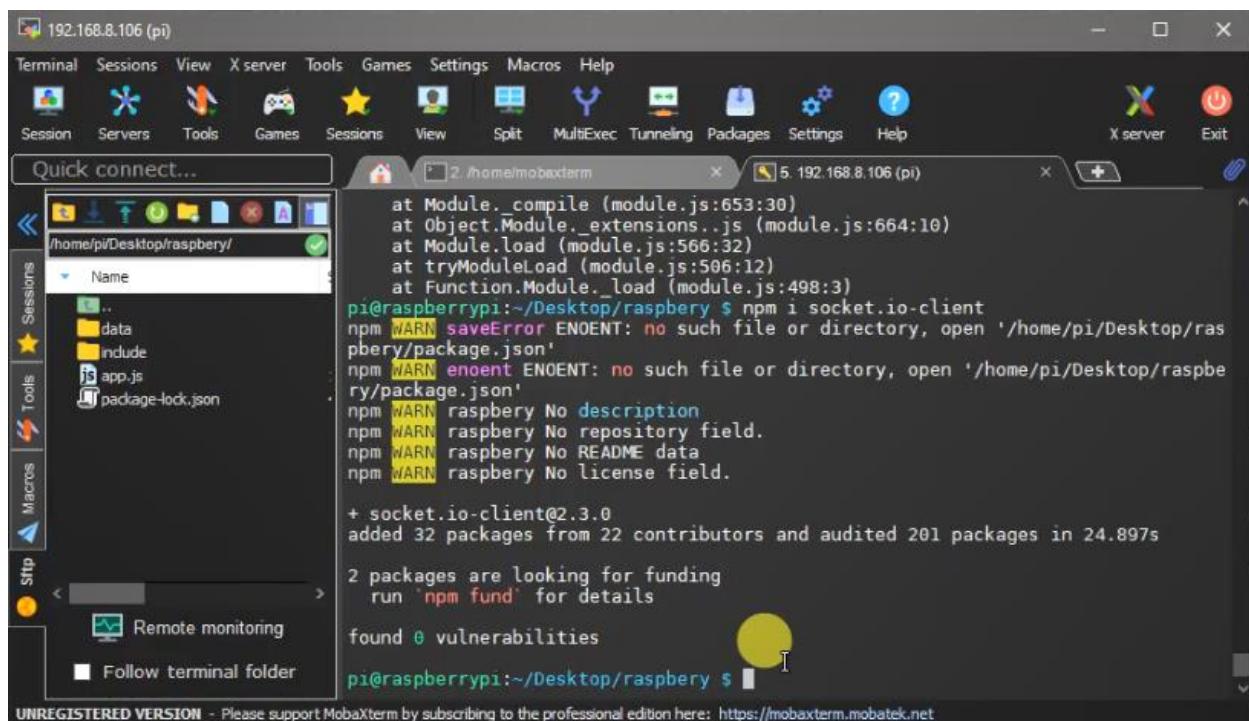
found 0 vulnerabilities

pi@raspberrypi:~/Desktop/raspberry $

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Figure 99 Start the node JS app 4



The screenshot shows a MobaXterm window titled '192.168.8.106 (pi)'. The terminal session is running on a Raspberry Pi. The user has navigated to the directory '/home/pi/Desktop/raspberry/' and run the command 'npm i socket.io-client'. The output of the command is displayed in the terminal window:

```

at Module._compile (module.js:653:30)
at Object.Module._extensions..js (module.js:664:10)
at Module.load (module.js:566:32)
at tryModuleLoad (module.js:506:12)
at Function.Module._load (module.js:498:3)
pi@raspberrypi:~/Desktop/raspberry $ npm i socket.io-client
npm WARN saveError ENOENT: no such file or directory, open '/home/pi/Desktop/raspberry/package.json'
npm WARN enoent ENOENT: no such file or directory, open '/home/pi/Desktop/raspberry/package.json'
npm WARN raspberry No description
npm WARN raspberry No repository field.
npm WARN raspberry No README data
npm WARN raspberry No license field.

+ socket.io-client@2.3.0
added 32 packages from 22 contributors and audited 201 packages in 24.897s

2 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

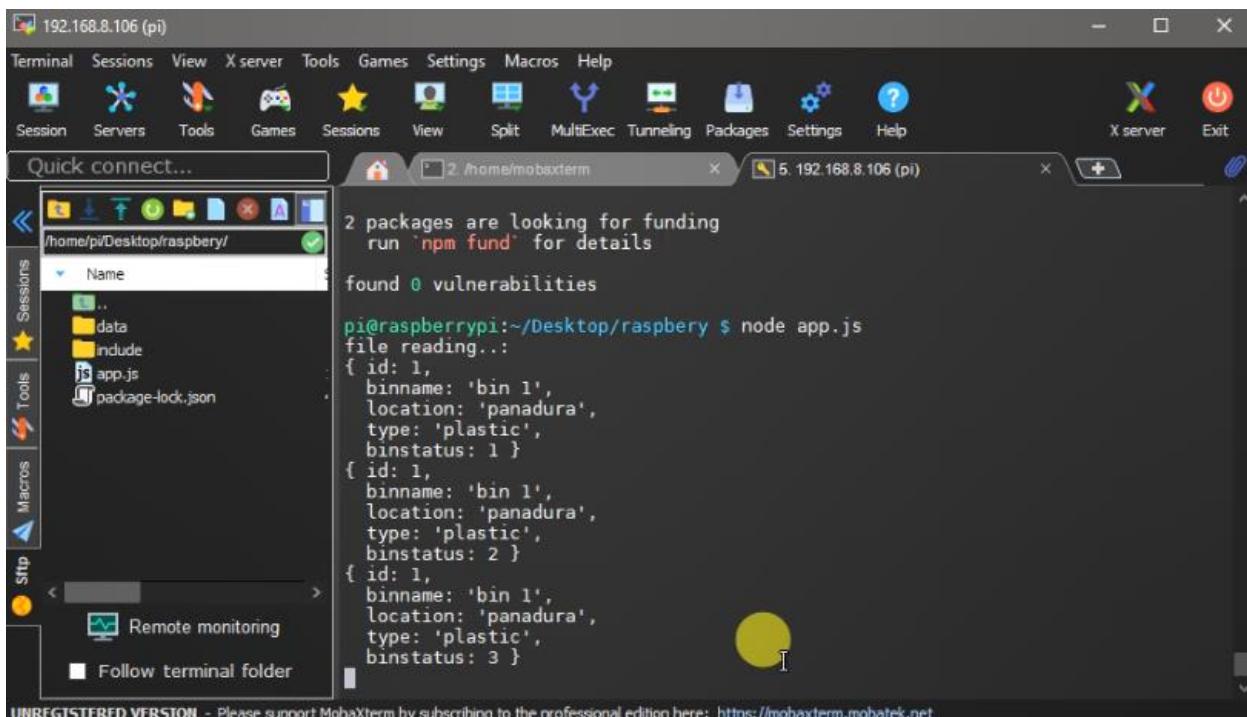
pi@raspberrypi:~/Desktop/raspberry $

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Figure 100 Start the node JS app 5

Start node JS app



The screenshot shows a terminal window in MobaXterm connected to a Raspberry Pi at 192.168.8.106. The command run was `node app.js`. The output shows the reading of a file named `app.js` and its execution. The file contains code to read a file named `data` and log three objects to the console. The objects represent bins with IDs 1, 2, and 3, located in 'panadura' with type 'plastic'.

```

2 packages are looking for funding
  run `npm fund` for details

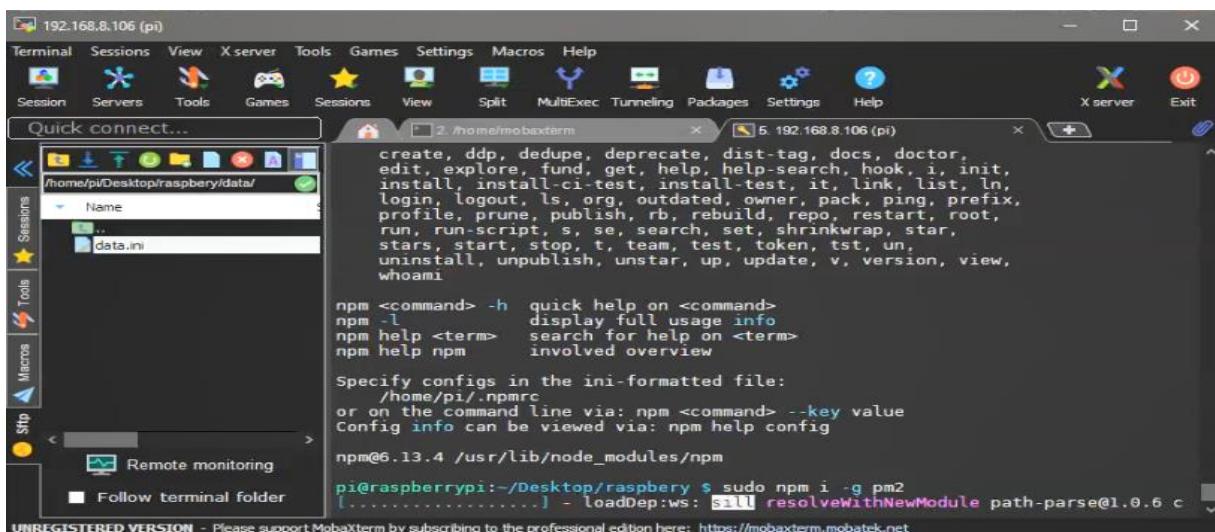
found 0 vulnerabilities

pi@raspberrypi:~/Desktop/raspberry $ node app.js
file reading...
{
  id: 1,
  binname: 'bin 1',
  location: 'panadura',
  type: 'plastic',
  binstatus: 1
}
{
  id: 1,
  binname: 'bin 1',
  location: 'panadura',
  type: 'plastic',
  binstatus: 2
}
{
  id: 1,
  binname: 'bin 1',
  location: 'panadura',
  type: 'plastic',
  binstatus: 3
}

```

Figure 101 Start the node JS app

Installation Pm2 Process management node JS module for when system restart application will automatically start. No need to manual run the commands.



The screenshot shows a terminal window in MobaXterm connected to a Raspberry Pi at 192.168.8.106. The user runs the command `sudo npm i -g pm2` to install the PM2 module. The output shows the installation progress and the creation of a configuration file `data.ini`.

```

create, ddp, dedupe, deprecate, dist-tag, docs, doctor,
edit, explore, fund, get, help, help-search, hook, i, init,
install, install-ci-test, install-test, it, link, list, ln,
login, logout, ls, org, outdated, owner, pack, ping, prefix,
profile, prune, publish, rb, rebuild, repo, restart, root,
run, run-script, s, se, search, set, shrinkwrap, star,
stars, start, stop, t, team, test, token, tst, un,
uninstall, unpublish, unstar, up, update, v, version, view,
whoami

npm <command> -h quick help on <command>
npm -l display full usage info
npm help <term> search for help on <term>
npm help npm involved overview

Specify configs in the ini-formatted file:
  /home/pi/.npmrc
or on the command line via: npm <command> --key value
Config info can be viewed by: npm help config

npm@6.13.4 /usr/lib/node_modules/npm
pi@raspberrypi:~/Desktop/raspberry $ sudo npm i -g pm2
[...]
loadDep:ws: sill resolveWithNewModule path-parse@1.0.6 c

```

Figure 102 Installation PM2 JS module

```

pi@raspberrypi:~/Desktop/raspberry $ sudo npm i -g pm2
/usr/bin/pm2 -> /usr/lib/node_modules/pm2/bin/pm2
/usr/bin/pm2-dev -> /usr/lib/node_modules/pm2/bin/pm2-dev
/usr/bin/pm2-docker -> /usr/lib/node_modules/pm2/bin/pm2-docker
/usr/bin/pm2-runtime -> /usr/lib/node_modules/pm2/bin/pm2-runtime
npm WARN notsup Unsupported engine for mkdirp@1.0.4: wanted: {"node":">>=10"} (current: {"node":"8.17.0","npm":"6.13.4"})
npm WARN notsup Not compatible with your version of node/npm: mkdirp@1.0.4
npm WARN notsup Unsupported engine for semver@7.3.2: wanted: {"node":">>=10"} (current: {"node":"8.17.0","npm":"6.13.4"})
npm WARN notsup Not compatible with your version of node/npm: semver@7.3.2
npm WARN notsup Unsupported engine for semver@7.2.3: wanted: {"node":">>=10"} (current: {"node":"8.17.0","npm":"6.13.4"})
npm WARN notsup Not compatible with your version of node/npm: semver@7.2.3
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.1.2 (node_modules/pm2/node_modules/chokidar/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"arm"})
+ pm2@4.4.0
added 185 packages from 191 contributors in 95.951s
pi@raspberrypi:~/Desktop/raspberry $ pm2 start app.js

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Figure 103 Installation PM2 JS module 2

id	name	mode	status	cpu	memory
0	app	Fork	0	online	0% 20.4mb

```

$ pm2 monitor
Make pm2 auto-boot at server restart:
$ pm2 startup
To go further checkout:
http://pm2.io/

[PM2] Spawning PM2 daemon with pm2_home=/home/pi/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /home/pi/Desktop/raspberry/app.js in fork_mode (1 instance)
[PM2] Done.

      id   name        mode   status    cpu   memory
      0   app        Fork  online   0% 20.4mb

pi@raspberrypi:~/Desktop/raspberry $ pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in /home/pi/.pm2/dump.pm2
pi@raspberrypi:~/Desktop/raspberry $

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Figure 104 Installation PM2 JS module 3

```

[PM2] Spawning PM2 daemon with pm2_home=/home/pi/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /home/pi/Desktop/raspberry/app.js in fork_mode (1 instance)
[PM2] Done.

+-----+
| id | name      | mode | ↴ | status   | cpu    | memory |
|---|---|---|---|---|---|---|
| 0  | app       | fork | 0 | online  | 0%    | 20.4mb |
+-----+
pi@raspberrypi:~/Desktop/raspberry $ pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in /home/pi/.pm2/dump.pm2
pi@raspberrypi:~/Desktop/raspberry $ pm2 startup
[PM2] Init System found: systemd
pi
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/usr/bin /usr/lib/node_modules/pm2/bin/pm2 startup systemd -u pi --hp /home/pi
pi@raspberrypi:~/Desktop/raspberry $ 

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Figure 105 Installation PM2 JS module 4

View Pm2 Process log

```

Smart Garbage Man
+-----+
| id | name      | mode | ↴ | status   | cpu    | memory |
|---|---|---|---|---|---|---|
| 0  | app       | fork | 0 | stopped | 0%    | 35.5mb |
+-----+
pi@raspberrypi:~/Desktop/raspberry $ pm2 list
[PM2] Applying action stopProcessId on app [all](ids: 0)
[PM2] [app|0] +
+-----+
| id | name      | mode | ↴ | status   | cpu    | memory |
|---|---|---|---|---|---|---|
| 0  | app       | fork | 0 | stopped | 0%    | 35.5mb |
+-----+
pi@raspberrypi:~/Desktop/raspberry $ pm2 stop all
[PM2] Applying action restartProcessId on app [all](ids: 0)
[PM2] [app|0]
[PM2] Process successfully started
+-----+
| id | name      | mode | ↴ | status   | cpu    | memory |
|---|---|---|---|---|---|---|
| 0  | app       | fork | 0 | online  | 0%    | 14.4mb |
+-----+
pi@raspberrypi:~/Desktop/raspberry $ 

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Figure 106 View Pm2 Process log 1

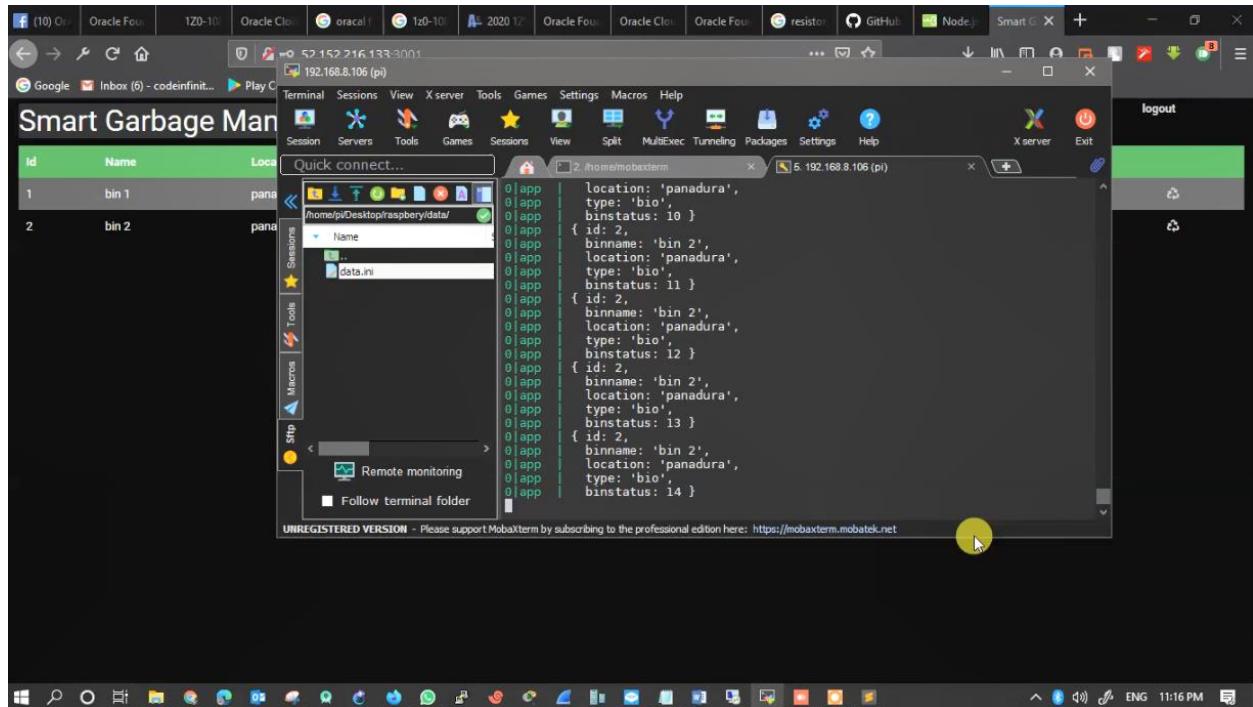


Figure 107 View Pm2 Process log 2

CHAPTER 05

5.1 Summary

With the advancement of technology, the search for life and faster processing has led to the computerization of several processes. A large number of areas of computer technology, especially social activities have been changed and everyone wants to make everything easier so a technology-based waste management system has been developed.

5.2 Conclusion

In this Smart Garbage Management System is developed and addressed in where, ultrasonic sensor embedded on using the garbage's dustbins after that used raspberry pi board through sent the data of the system back end of the server.

for waste level detection, GSM modem is used to send waste level data collected by microcontroller send an MSG of a in the area garbage collector, in this system use the published the public places so people don't know the bin full or no so they have feature of the full of the bin full of the led indicator light is on.

Furthermore, system website has click icon in the click on the system maintainer can know to the day when bin is full.

So, this system in conclusion, an appropriate analysis and evaluation of the designed system allow concluding in a secure way that the system is an efficient, usable and reliable garbage management system. It is working properly and sufficiently meets the minimum expectations that were for it initially. The new system should bring benefits to users and system maintainers and government in terms of efficient use of the garbage system.

5.3 Future work

We hope to add more features to this system in the future

- GEO location tracking, automatically send the SMS notification relevant garbage trucks.
- You can also choose the best way to collect garbage in garbage trucks.
- You can predict how long the garbage bin in the area will be full
- You can also post a website related to this method so that you can check the status of garbage bins in the respective area from home.
- The interface can be made more attractive.

References

1. P. Sukholthaman, K. Shirahada, Proceedings of PICMET '14 Conference: Portland International Center for Management of Engineering and Technology; Infrastructure and Service Integration, (2014)
2. C. K.M. Lee, T. Wu, International Conference on Industrial Engineering and Engineering Management, 798 (2014)
3. A.F. Thompson, A.H. Afolayan, E.O. Ibidunmoye, Information Science, Computing and Telecommunications, 206 (2013)
4. ResearchGate. (n.d.). (PDF) Automated Garbage Monitoring System Using Arduino. [online] Available at: https://www.researchgate.net/publication/332177309_Automated_Garbage_Monitoring_System_Using_Arduino [Accessed 20 Oct. 2020].
5. Instructables (2017). Smart Garbage Monitoring System Using Internet of Things (IOT). [online] Instructables. Available at: <https://www.instructables.com/Smart-Garbage-Monitoring-System-Using-Internet-of-/> [Accessed 20 Oct. 2020].
6. SMART WASTE MANAGEMENT SYSTEM PROJECT REFERENCE NO.: 39S_BE_0321. (n.d.). [online] Available at: http://www.kscst.iisc.ernet.in/spp/39_series/SPP39S/01_Seminar%20Projects/068_39S_BE_0321.pdf.
7. Shirke, S.I., Ithape, S., Lungase, S. and Mohare, M. (2008). Automation of smart waste management using IoT. [online] International Research Journal of Engineering and Technology, p.414. Available at: <https://www.irjet.net/archives/V6/i6/IRJET-V6I6153.pdf>.
8. An Automated Trash Monitoring Systemfor Waste Management using IoT– (I-Bin). (2019). International Journal of Engineering and Advanced Technology, 8(6S3), pp.1134–1137.
9. () *Top 5 Challenges Of Smart Waste Management*, Available at: <https://medium.com/@swachhcoin/top-5-challenges-of-smart-waste-management-59ecf3e24b6d> (Accessed: 05th of August).
10. () *Smart Garbage Monitoring System for Waste Management* , Available at: https://www.matecconferences.org/articles/matecconf/pdf/2017/11/matecconf_etic2017_01098.pdf (Accessed: 05th August 2020).