PES UNIVERSITY, Bangalore

(Established under Karnataka Act No. 16 of 2013)

Department of Computer Science & Engineering

Mobile and Autonomous Robots (UE22CS343BB7)

6th Semester

Mini-Project

Project Title:

Autonomous Garbage Collection Robot System

Team Details:

1. **D Vidya Sagar - PES2UG22CS154**
2. **Darshan Raju Loni - PES2UG22CS158**
3. **Darshan Gowda G L - PES2UG22CS161**
4. **Dhanush Mahendra Hegde - PES2UG22CS174**

Professor Name: Dr. Gokul Kannan Sadasivam

Project Description:

The **Autonomous Garbage Collection System** is designed to automate the collection and disposal of waste bins in indoor and outdoor hospital environments. Hospitals generate various types of medical waste that must be handled with care to avoid contamination and ensure the safety of healthcare workers and patients. Manual waste collection is labor-intensive, time-consuming, and can expose workers to hazardous materials.

This project presents a robotic system that utilizes **ROS 2 (Robot Operating System)**, **OpenCV**, and **Gazebo simulation**, integrated with the **TurtleBot3 platform and OpenMANIPULATOR-X** to autonomously navigate a mapped environment, detect and locate trash bins, pick them up, and place them at a predefined disposal area.

The system uses computer vision for perception, autonomous navigation for path planning, and robotic manipulation for pick-and-place tasks. It is fully simulated to support testing and validation without physical hardware, making it an effective prototype for real-world hospital use.

Project Objectives:

1. Develop a complete simulation-based autonomous robot that performs garbage collection tasks in a hospital-like environment using ROS2 and Gazebo.
2. Implement object detection and localization using computer vision (OpenCV) to identify and locate trash bins.
3. Design autonomous navigation algorithms to enable the robot to move through complex indoor environments and reach target locations efficiently.
4. Enable robotic manipulation to pick up and place waste bins using the OpenMANIPULATOR-X in simulation.
5. Ensure modular and reusable software architecture by separating perception, navigation, and manipulation into distinct modules.
6. Demonstrate safety, accuracy, and efficiency in garbage collection tasks to reduce human involvement in hazardous waste handling.
7. Lay the groundwork for real-world deployment, emphasizing adaptability to real hospital layouts and integration with existing infrastructure.

Methods and Materials:

1. System Design

The system is modular, composed of three ROS2 nodes: **Perception, Navigation,** and **Manipulation.** These nodes interact to complete the garbage collection routine. The robot identifies trash bins using onboard camera input and computer vision, navigates to the bin using the **Nav2 stack** and LiDAR/odometry input, and picks up the bin using a simulated gripper. All operations are run in **Gazebo** using a hospital-like simulation world.

2. Algorithm/Model Development

1. Object Detection: Implemented using contour detection and color segmentation in OpenCV to identify the trash bin.

2. Navigation: Built on **Nav2** stack using Dijkstra for global planning and Dynamic Window Approach (DWA) for local obstacle avoidance.

3. Path Planning: The robot navigates from its current location to the bin location and then to the disposal area.

4. Manipulation:

a. Uses a **pick-place service** mechanism in ROS2.
b. Executes bin pickup by calling **delete_entity** in Gazebo (simulated).
c. Places bin at disposal area using **spawn_entity**.

5. SLAM (Simultaneous Localization and Mapping) for localization.

6. Grid-based search for bin locations.

7. Basic FSM (Finite State Machine) to coordinate bin detection, pick-up and delivery.

3. Implementation Steps

1. World Design: Create a hospital environment using **.world** and **.model** files in Gazebo.

2. Robot Configuration: Integrate TurtleBot3 with OpenMANIPULATOR-X in simulation.

3. Perception Module: Develop and test OpenCV-based detection of bins.

4. Mapping & Navigation:

a. Generate map using SLAM.
b. Launch navigation stack and test path-following to bins and disposal zones.
c. Launch files were used to bring up navigation, mapping, perception, and world setup using ROS2 launch system.

5. Manipulation Module:

d. Pick bin using **delete_entity** service.
e. Place bin using **spawn_entity.**

6. Integration:

f. Connect all modules through launch files.
g. Run the full autonomous routine from bin detection to disposal.

4. Hardware Components (if applicable)

Although the current implementation is simulation-only, the intended real-world hardware includes:

1. TurtleBot3 Burger or Waffle Pi
2. OpenMANIPULATOR-X Arm
3. Intel RealSense Depth Camera (optional for better vision)
4. Raspberry Pi or Jetson Nano for onboard processing
5. Battery power supply and chassis for mobility

The hardware setup is inspired by TurtleBot3 Waffle Pi with OpenMANIPULATOR-X, LiDAR, and an RGB camera. Although the simulation is software-only, this hardware selection supports future physical deployment.

5. Software Tools

Software Tools	Description / Purpose
ROS 2 Foxy Fitzroy	Middleware framework for robotics – used for node communication, lifecycle management
Gazebo	3D simulation environment to test and visualize the robot in a hospital setup
RViz2	Visualizes robot sensors, odometry, and navigation goals in real-time
OpenCV	Image processing library used to detect trash bins using HSV color thresholding
Navigation2(Nav2)	Handles robot path planning, localization, and obstacle avoidance
cv_bridge	Converts between ROS image messages and OpenCV images
C++ (OOP	Main programming language for system modules (navigation, manipulation, perception)
CMake & colcon	Build systems used to compile and structure the ROS2 project
image_transport	Provides efficient ways to publish and subscribe to images

Project Outcome:

1. **Simulation video link (drive link) :**

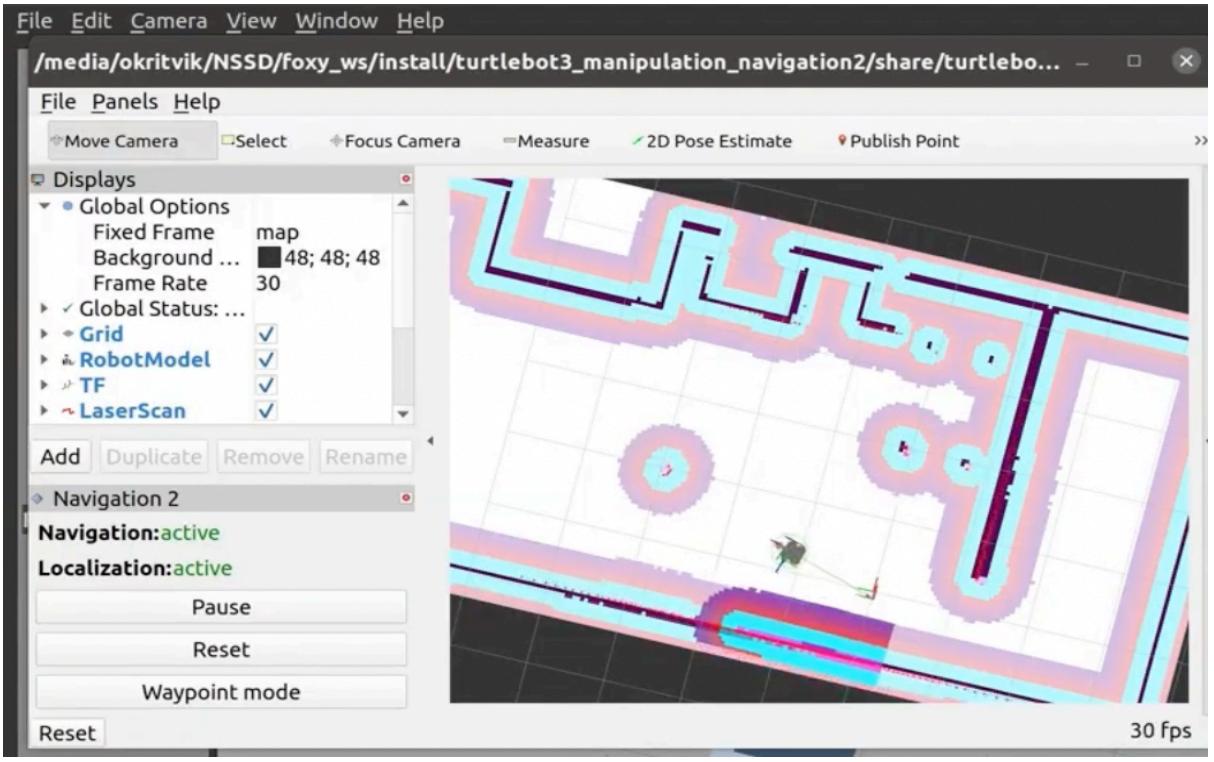
   MARS_TEAM_14_DEMO.mp4

   https://drive.google.com/file/d/163hcycilBACxJOnXSGVSFrwqq90cj3eM/view

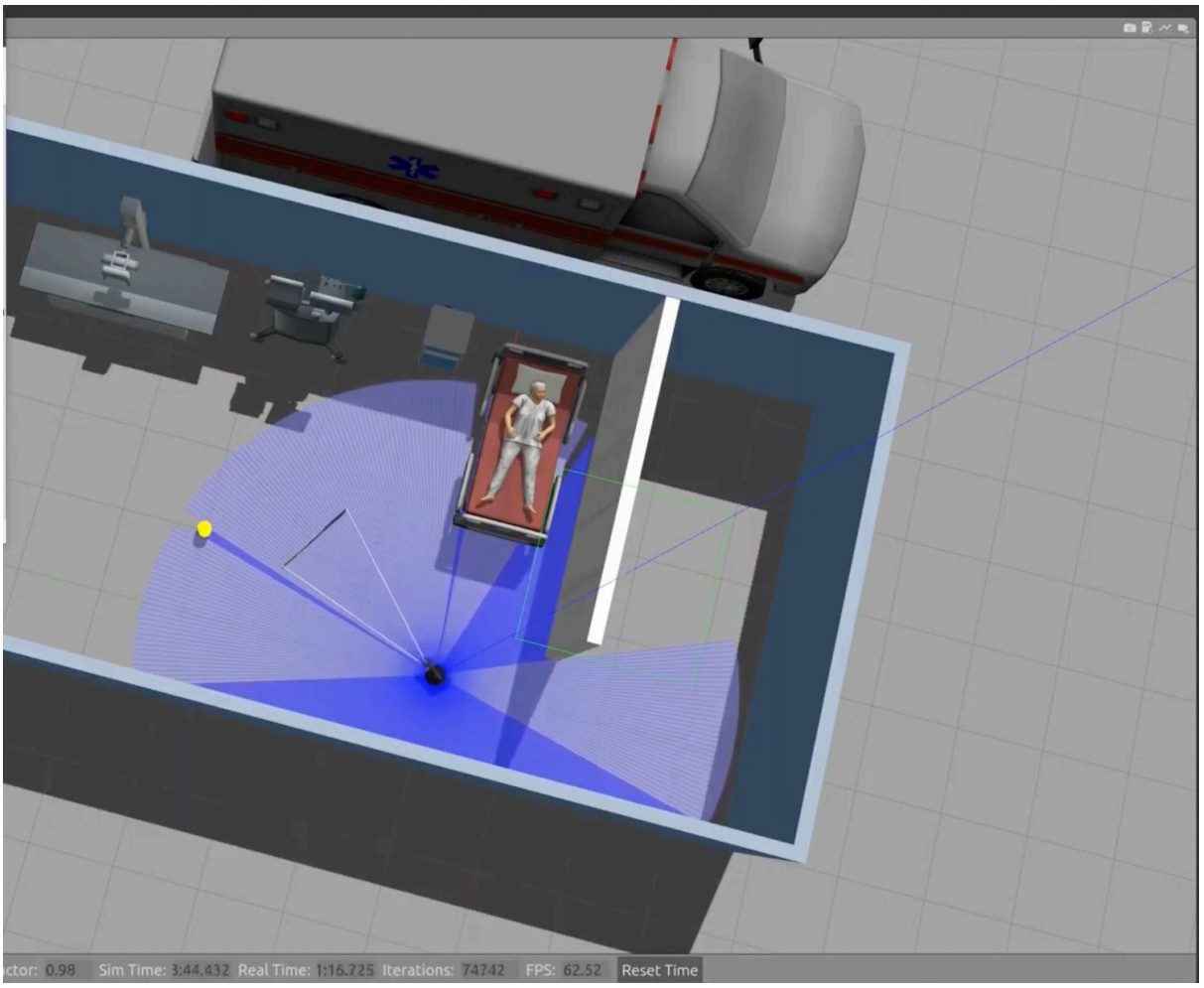
2. **GitHub Repo Link (Source Code):**
   https://github.com/darshandarshugl/Autonomous-Garbage-Collection-Robot-System.git

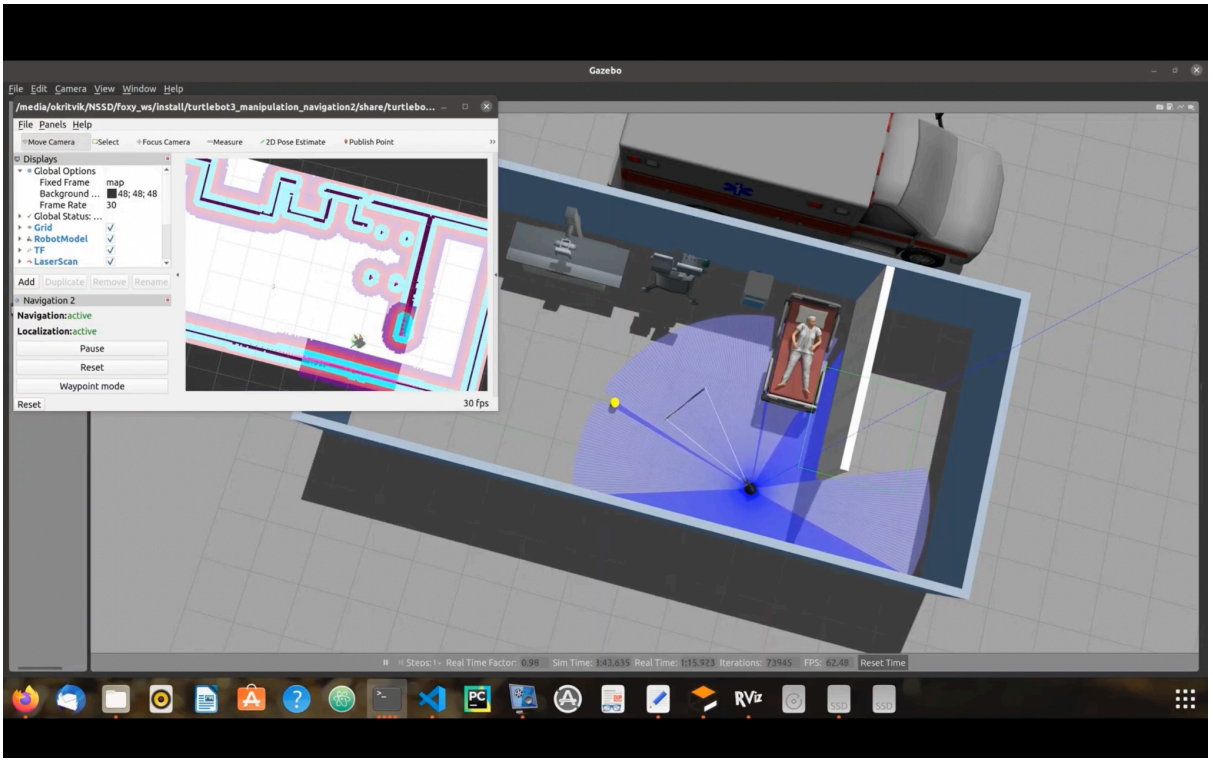
3. **Output Results :**

   Rviz - Visualization



Gazebo - Simulation





References:

1. **Open Robotics – ROS 2 Foxy Fitzroy Documentation**
   https://docs.ros.org/en/foxy/index.html
2. **Gazebo Simulator Documentation – Open Source Robot Simulation**
   https://classic.gazebosim.org/tutorials
3. **OpenCV Library Documentation – Open Source Computer Vision**
   https://docs.opencv.org/master/
4. **Navigation2 Stack – ROS2 Navigation Stack for Autonomous Robots**
   https://navigation.ros.org
5. **cv_bridge Package – ROS Package for Converting ROS Image Messages to OpenCV Images**
   https://github.com/ros-perception/vision_opencv
6. **image_transport Package – ROS Image Streaming Utility**
   https://github.com/ros-perception/image_common
7. **UTM Virtualization Software – Run VMs on Apple Silicon**
   https://mac.getutm.app
8. **Robot Operating System (ROS) Tutorials – Official Beginner Guide**
   https://docs.ros.org/en/foxy/Tutorials.html
9. **"Programming Robots with ROS" – O'Reilly Media, 2015**
   (Author: Morgan Quigley, Brian Gerkey, and William D. Smart)
   ISBN: 978-1-449-36710-7