ROS2 Based Tasks – USE ROS2 HUMBLE

Task 1: Simulating a robot by creating a world frame and moving init.

- Create a 2-wheel differential drive robot using URDF.
- Attach the following sensors:
  - LIDAR (mounted on top of the robot)
  - IMU
  - Wheel odometry
- Publish LIDAR and IMU data on their respective topics.
- Visualize the sensor data using RViz.
- Create a custom world in Gazebo and spawn the robot into this world.
- Create a map and integrate Nav2:
  - The robot should be able to navigate to a given goal.
  - Implement obstacle avoidance during navigation.

Task 2: ROS Package and Custom Message/Service Creation

1. Create a ROS package named **`training_interfaces`**.
   - Inside this package, create a custom message **`person.msg`** with the following fields:
     1. `String name`
     2. `Int32 age`
2. Create another ROS package named **`training`**.
3. **Create a publisher and subscriber** in the **`training`** package:
   - Use the `person.msg` message type for communication between the publisher and subscriber.
4. **Create a custom service** named **`value.srv`** inside the **`training_interfaces`** package with the following fields:
   1. `Int32 a`
   2. `Int32 b`

   ---

   3. `Int32 val`

   This service will handle two integer inputs.
5. **Create a ROS service server and client** in the `training` package:
   - In the service server, implement the logic to **add** the two integer values (`a` and `b`) received from the client.
   - Send the sum of these values as the response (`val`).

References:

- **URDF Robot Creation**: https://docs.ros.org/en/humble/Tutorials/Intermediate/URDF/Using-URDF-With-Robot-State-Publisher.html
- **Add LIDAR and IMU to URDF**: https://classic.gazebosim.org/tutorials?tut=ros_imu https://automaticaddison.com/how-to-launch-a-lidar-sensor-in-ros-2-foxy/ *(still applicable for Humble)*
- **Creating Custom Gazebo Worlds**: https://classic.gazebosim.org/tutorials?tut=build_world

- **Nav2 Setup and Integration**:
  https://navigation.ros.org/
  https://docs.ros.org/en/humble/Tutorials/Navigation2/Getting-Started.html
- **ROS Service Sample**:
  https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Writing-A-Simple-Cpp-Service-And-Client.html
- **Custom msg & srv** Implementation of ROS custom messages:
  https://docs.ros.org/en/crystal/Tutorials/Single-Package-Define-And-Use-Interface.html