Dynamic Path Planning Robot for Crowded Environments

Project Breakdown into Tasks

---

1. Initial Setup and Planning

- **Task 1:** Define project goals and deliverables clearly.
- **Task 2:** Set up the development environment (install ROS 2, Gazebo, Rviz, and Python libraries).
- **Task 3:** Familiarize the team with required tools and technologies (e.g., Dynamic Window Approach (DWA), predictive models).
- **Task 4:** Develop a timeline for each phase of the project.

---

2. Environment Simulation

- **Task 5:** Design a crowded environment in Gazebo (e.g., simulate a train station or mall).
- **Task 6:** Add dynamic obstacles like moving human-like avatars.
- **Task 7:** Configure environmental parameters (e.g., obstacle speeds, sizes, and crowd density).

---

3. Perception System Integration

- **Task 8:** Simulate sensors:
  - Add LiDAR for obstacle detection.
  - Integrate RGB cameras for visual input.
- **Task 9:** Write scripts to process sensor data in ROS 2.

---

4. Dynamic Path Planning

- **Task 10:** Implement the Dynamic Window Approach (DWA):
  - Study and modify Nav2's DWA plugin for dynamic environments.
  - Test with various scenarios in the simulation.
- **Task 11:** Optimize the algorithm for computational efficiency.

---

5. Predictive Models

- **Task 12:** Design predictive models for forecasting the motion of dynamic obstacles:
  - Collect sample movement data from the simulation.
  - Train a model using TensorFlow/PyTorch to predict obstacle trajectories.
- **Task 13:** Integrate the predictive model into the ROS 2 navigation stack.

---

6. Visualization

- **Task 14:** Set up Rviz to:
  - Display the robot's path.
  - Visualize the obstacle map and sensor data.
- **Task 15:** Create a real-time interface for monitoring robot navigation.

---

7. Testing and Debugging

- **Task 16:** Run simulation tests under various crowded scenarios (e.g., high density, random movements).
- **Task 17:** Identify and resolve bugs in:
  - Path planning algorithms.
  - Sensor integration.
  - Predictive models.

---

8. Performance Evaluation

- **Task 18:** Measure the robot’s navigation efficiency (e.g., time to destination, collision rates).
- **Task 19:** Optimize based on test results:
  - Improve response time.
  - Enhance obstacle prediction accuracy.

---

9. Documentation and Reporting

- **Task 20:** Document the implementation process, including:
  - Algorithm design.
  - Challenges and solutions.
  - Performance metrics.
- **Task 21:** Create a final report and presentation.

10. Project Review and Delivery

- **Task 22:** Conduct a final review with stakeholders or instructors.
- **Task 23:** Deliver the project files, documentation, and simulation results.