

# Web UI for AMR in intralogistics

## Motivation

ROS (Robot Operating System) has become the de facto standard for developing robotic applications in both research and industry. However, ROS typically requires users to be proficient in Ubuntu and comfortable with command-line operations. A web-based user interface (UI) can overcome these limitations by making robot operations more accessible and user-friendly to a wider audience.



Figure 1: Example of web interface

## Advantages

- **Simplified Operation:** Users can control robots through familiar web technologies, which are generally easier to use than terminal commands. This lowers the learning curve and enables operators with minimal technical expertise to work efficiently.
- **Intuitive and Visual Interaction:** A web UI provides a graphical representation of the robot and its environment, allowing users to interact more intuitively.
- **Remote Accessibility:** Operators can control robots remotely from any device with internet connectivity and a standard web browser.
- **Multi-User Collaboration:** Several users can access the interface simultaneously, enabling team collaboration, information sharing, and coordinated operation.
- **Customizability and Extensibility:** Features such as real-time data visualization, task scheduling, and integration with external systems can be added easily, extending the capabilities of the robotic system.

## Summary

A web-based user interface for operating robots in intralogistics—particularly when using ROS 2 (Humble)—offers significant advantages. It simplifies operation, supports remote access and collaboration, and provides opportunities for customization. By eliminating reliance on complex terminal commands, such an interface empowers a broader range of users, thereby enhancing efficiency and productivity in intralogistics environments.

## Tasks

1. **State of the Art Review**
  - Survey existing web UIs developed for ROS.
  - Explore different programming languages and frameworks for web development (e.g., Angular, JavaScript).
2. **Web GUI Development for AMR (fml Chair)**
  - Implement an SSH-based connection to the robot.
  - Enable on-demand visualization of the robot's camera.
3. **SLAM Integration**
  - Launch SLAM algorithms (e.g., ROS 2 SLAM Toolbox).
  - Visualize the generated maps.
  - Save and load maps.
4. **Navigation**
  - Visualize the static map and the position of the robot in the map.
  - Send navigation goals to the robot using Nav2.
  - Visualize the global map and laser readings.
  - Visualize dynamic objects.
5. **ROS Services**
  - Implement functionality to request ROS services via the web interface.


## Required Skills


- Strong oral and written communication in English.
- Interest in robotics.
- Hands-on experience with ROS.
- Programming experience in Python and/or C++.
- Experience in web development.
- Familiarity with SLAM and localization methods.
- Ability to work independently and self-direct.
- Strong attention to detail.

## Contact

Daniel Vidal

Chair of Materials Handling, Material Flow, Logistics (Prof. Fottner)

 +49 (89) 289 - 15955

 Room: 5505.01.590C

 daniel.vidal@tum.de