MICHIGAN STATE
UNIVERSITY

Project Plan Presentation

Remote Interface for Small-Scale Autonomous Race Cars

The Capstone Experience

Team Michigan State University CSE

Patrick Hogan
Ali Abboodi
Daphne Martin
Skanda Vijaykumar
Christian Wilkins
Zach Estepp

Department of Computer Science and Engineering
Michigan State University

Fall 2025

From Students…
…to Professionals

Project Sponsor Overview

Michigan State University – CSE Department

- MSU – PoliMOVE Accomplishments:
  - Won Indy Autonomous Challenge in 2024
  - First place at the IAC WeatherTech Raceway Laguna Seca event
  - Collab between MSU and Politecnico di Milano
- Sponsors of the Polimove-MSU team:
  - Michigan Office of Future Mobility and Electrification
  - DENSO North America Foundation
  - MSU Research Foundation

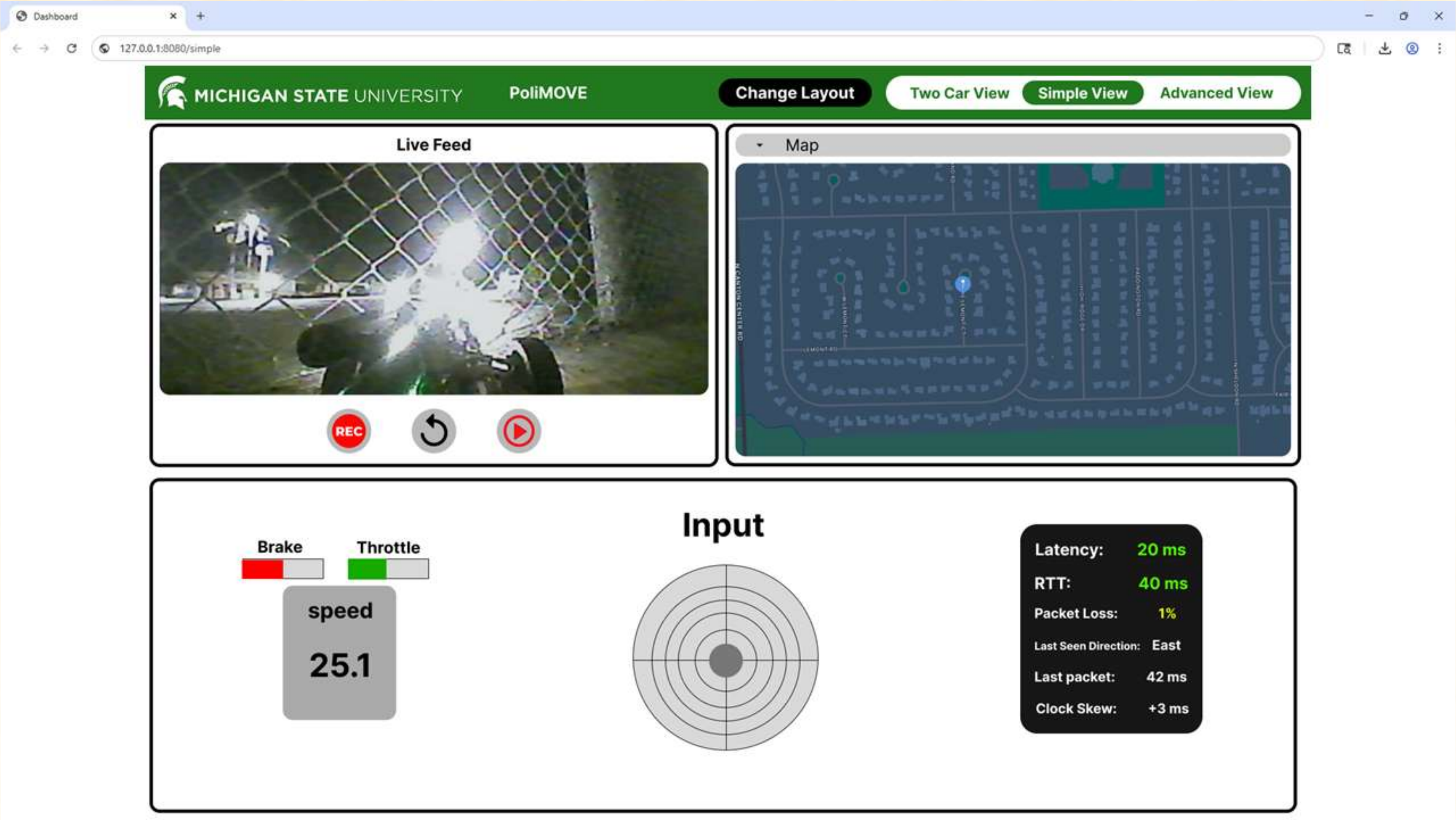
Project Functional Specifications

- **What is your problem?**
  - Indy Autonomous Challenge
  - High Stakes Testing
  - Crashing Full-Scale Car is Costly
  - Subpar Web UI
- **How do you fix the problem?**
  - Small-Scale to capture data
  - Less Cost when Crashing
  - Beautification of the UI
- **What else does it do?**
  - Access to GNSS data when offline
  - Ability for 2 cars to run at once
    - ROS Bags and Unique instances
  - Support Control through Front End UI

Project Design Specifications

- Web-based application for monitoring and interacting with racecars
- Provides basic vehicle control through the interface
- Displays live video feed of the cars in real time
- Shows key data, including:
    - LiDAR
    - GNSS data with charts and maps
    - IMU data with charts
    - Brake and Throttle inputs
    - Steering angle, speed and runtime
- Includes a playback feature to record and replay previous sessions

Screen Mockup: Simple View



speed

25.1

Latency: 20 ms

RTT: 40 ms

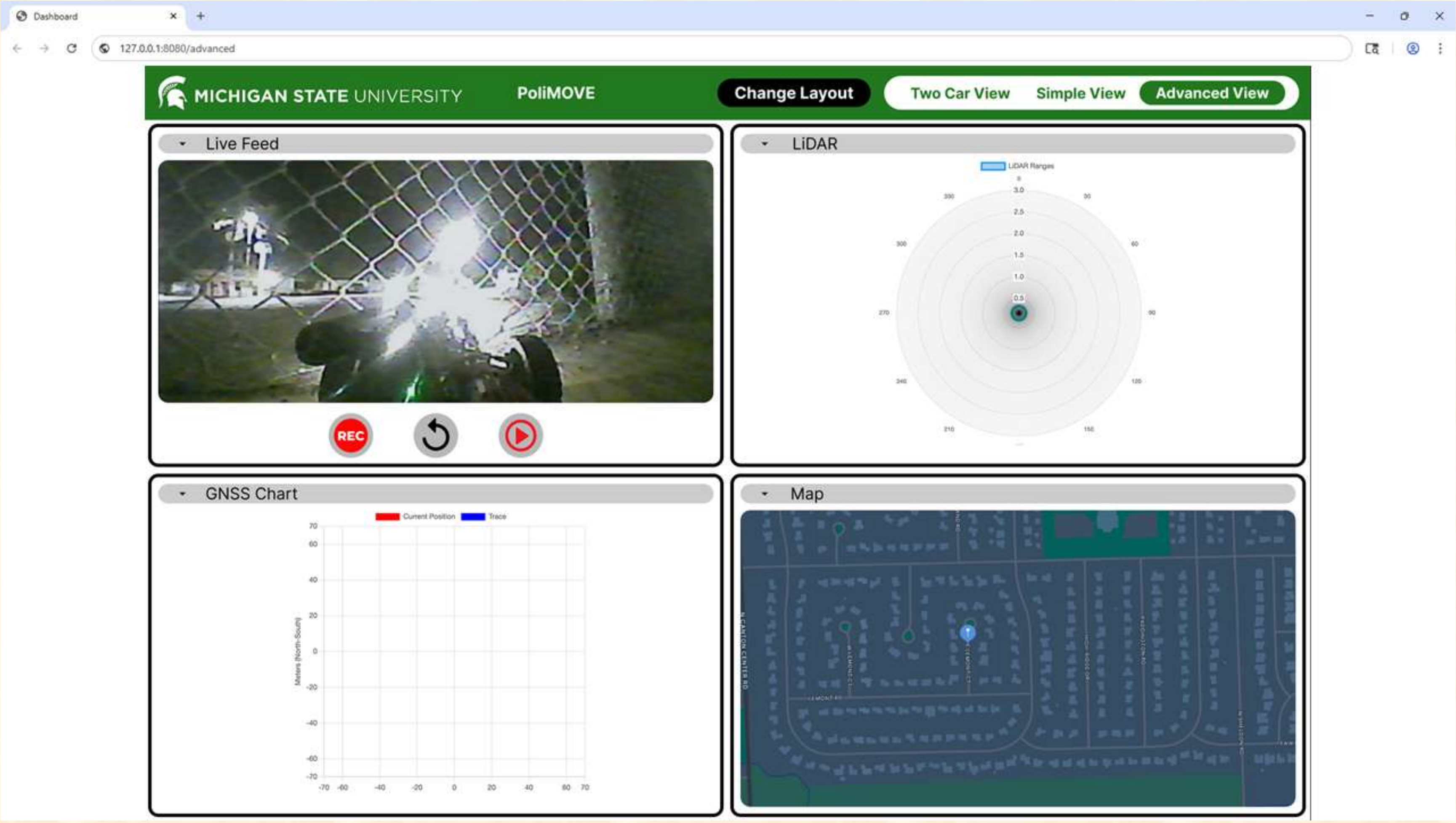
Packet Loss: 1%

Last Seen Direction: East

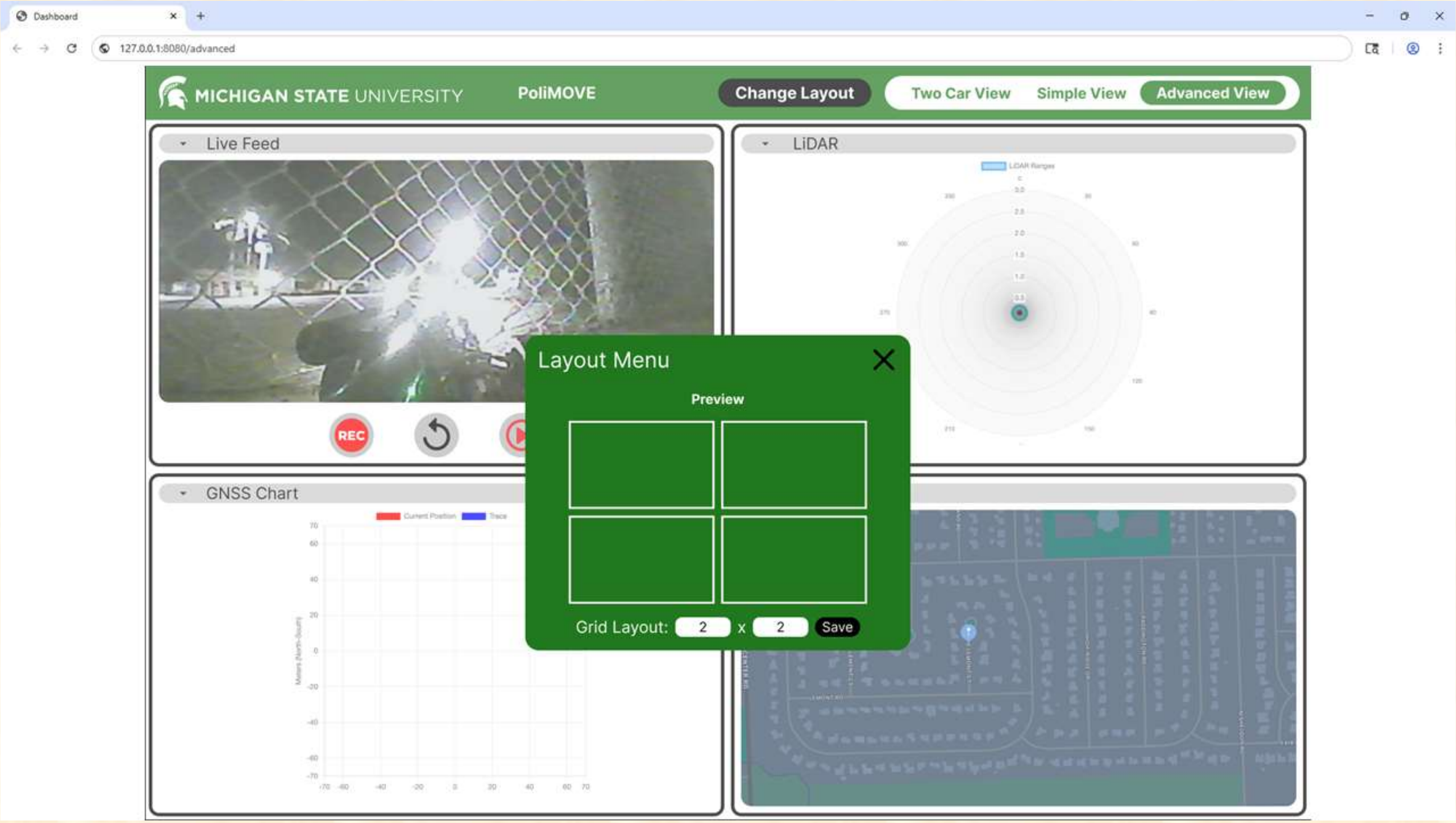
Last packet: 42 ms

Clock Skew: +3 ms

Screen Mockup: Advanced View

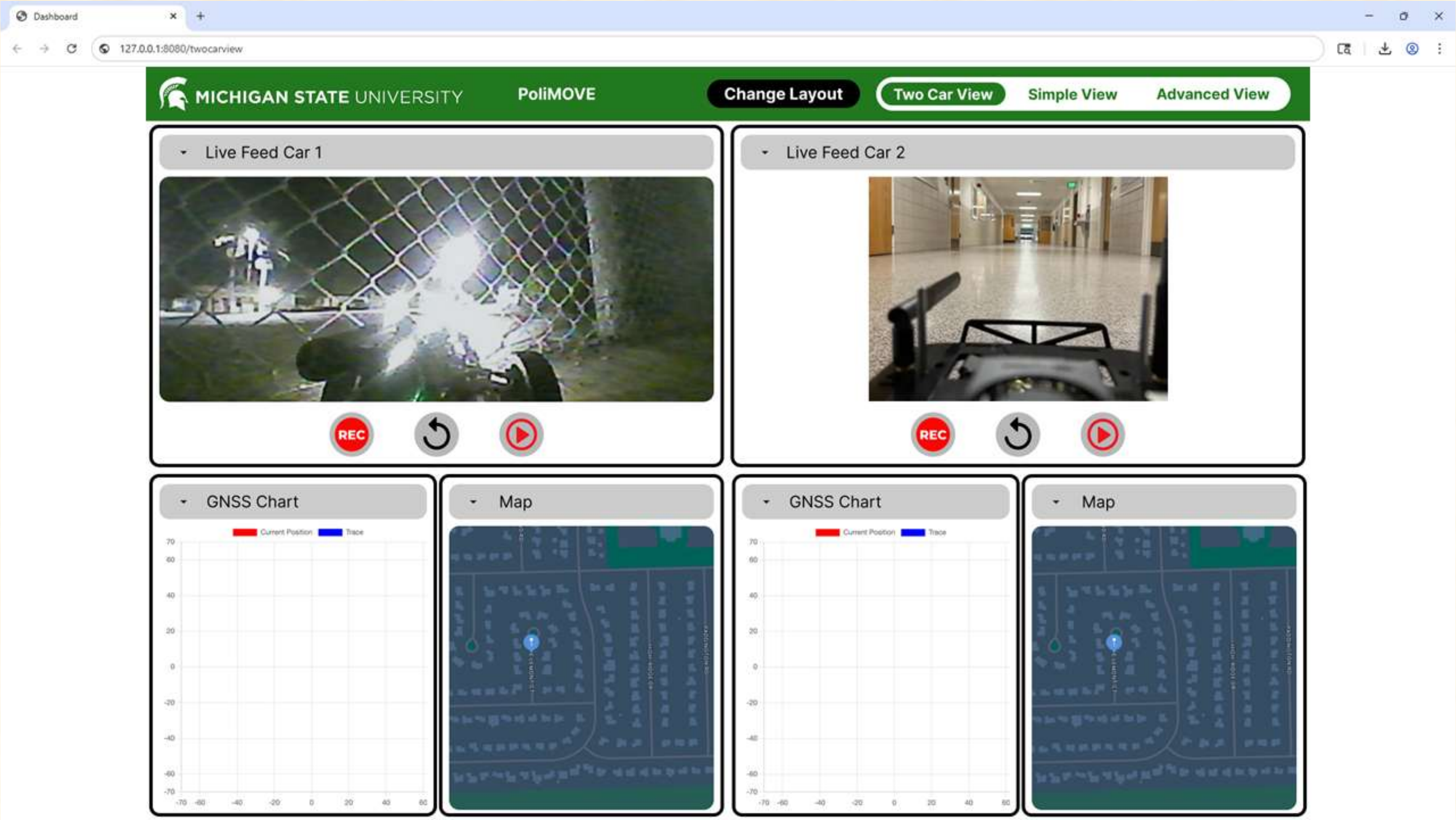


Screen Mockup: Change Layout Menu



Grid Layout: 2 x 2

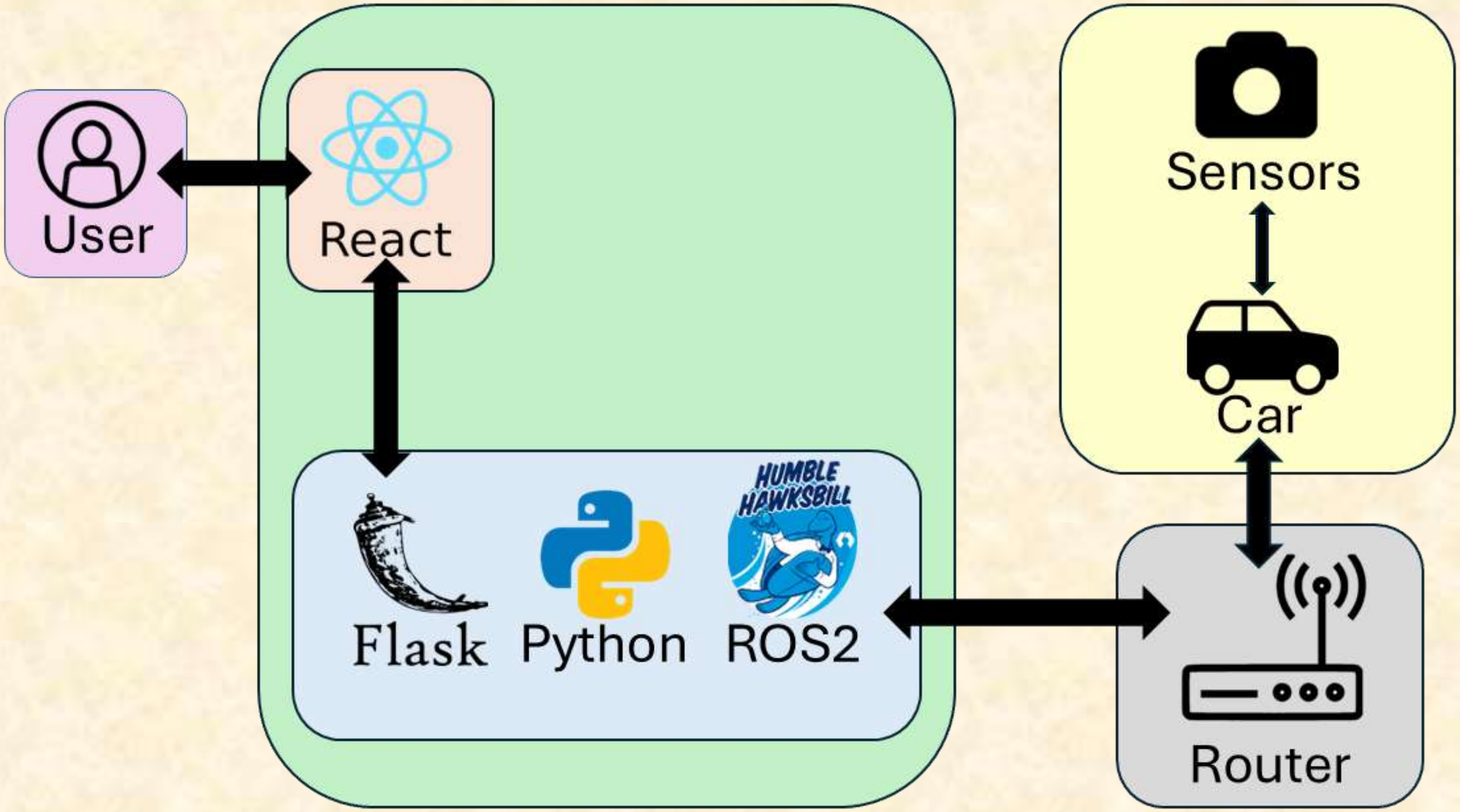
Screen Mockup: Two Car View



Project Technical Specifications

- Easy-to-use and appealing web application
- View sensor data collected by the car in an intuitive fashion
- Ability to drive the car in multiple ways, via controller, steering wheel, and touchscreen
- Ability to cache maps to use while car is offline
- Use application with two cars at the same time
    - One car plays back data, the other is driven

Project System Architecture



Project System Components

- Hardware Platforms
  - Small Remote Operated Car
  - Laptop
  - Wi-Fi Router
- Software Platforms / Technologies
  - Python
  - React
  - Flask
  - Robot Operating System 2 (ROS2)
  - Docker

Project Risks

- Risk 1 - Using and Navigating Cached Maps
  - Description: Accessing map data offline
  - Mitigation: Store map data locally
- Risk 2 - Running 2 cars at once
  - Description: Need to figure out how to run 2 cars at once
  - Mitigation: Communicate with both cars in 2 separate instances
- Risk 3 – Creating an Interactive UI for K-12 use
  - Description: Creating a front end controller for children to use
  - Mitigation: Use web sockets so that front end can communicate with the back end ROS2 subscribers
- Risk 4 – Working with hardware
  - Description: Team has limited experience in robotics/hardware
  - Mitigation: Work hard to become proficient with hardware

Questions?

? ? ? ?

? ?

? ? ?