Project Title:

NautiSense - Intelligent Ship Motion Monitoring System

Group Members:

- ➢ **Bakel Bakel Begededum**
- ➢ **Paul Zambrano**
- ➢ **Pedro Henrique de melo Estrella**

Brief Description:

The NautiSense Project short form of Nautical Sense aims to develop a system for sensing and monitoring ship motions in waves. When a ship is bring under large motions due to the incoming waves, it experiences six types of motion: heave, surge, sway, roll, pitch and yaw. Rogue seas can amplify these motions, potentially leading to instability. If the roll angle exceeds the ship's stability limits, it risks capsizing, passengers and crew safety, depending on the wave direction.

When this happen, the system will use statistical methods to define the seaway by it's spectrum, which indicates the amplitudes to it's motions; heave, roll, and pitch motions to assess the impact of different wave headings on a ship's stability and crew safety. Key parameters such as RMS (Root Mean Square), RAO (Response Amplitude Operator), and MSI (Motion Sickness Incidence) will be used to determine:

Resonance conditions that may cause excessive rolling.
Safe sailing speeds based on wave direction and motion response.
Dangerous onboard areas where crew members are at risk of motion sickness due to ship movements.

Deliverables:

1. Hardware System:

a. Towing Tank

This will allow us to validate our motion analysis system in a controlled environment

Objectives:

Test the feasibility of measuring heave, roll, and pitch using IMU sensors.
Validate the response amplitude operator (RAO) calculations in a controlled wave environment.
Establish a correlation between wave heading, ship motion, and safe operational speeds.

b. Vessel Prototype

This phase involves deploying the system on a real (small-scale) vessel for real-world testing.

Prototype Specifications

Length Overall (LOA): ≥ 40 cm (big enough for Arduino Uno & Raspberry Pi).
Hull Type: A scaled model of a monohull ship with stabilizers for controlled motion.

Material: Reinforced 3D-printed plastic.

Electronics & Sensors

1. Motion Measurement:

    IMU (MPU6050)
    Secondary IMU for redundancy
    Magnetometer (HMC5883L) → Determines heading.
    Ultrasonic sensor (HC-SR04) → Measures wave height.

2. Wave & Heading Tracking:

    GPS (u-blox NEO-6M) for speed tracking.
    Compass (HMC5883L) for wave-relative heading.

3. Real-Time Processing & Control:

    Arduino Uno: Captures data from sensors and sends it to Raspberry Pi.
    Raspberry Pi 5: Runs real-time motion calculations & visualization.

4. Power System:

    Battery Pack (Li-Po or 18650 cells)
    Voltage Regulator (5V for Arduino, 3.3V for sensors)

5. A communication module (wired/wireless) to send data for analysis.

2. Software System:

a. Sensor data acquisition and filtering (Ardiuno/C++).

b. Mathematical models for ship motion analysis (Python and Matlab):

RMS for measuring motion intensity.
RAO for resonance detection.
MSI for assessing motion sickness risk.
Graphical interface/dashboard to visualize ship responses in real-time.
Wave direction estimation module to determine the source of waves.

c. Web App Features (React.js, Node.js, Flask, Firebase, Js)
The web app should have the following functionalities:

A. Towing Tank Control

Adjust wave height, frequency, and direction.
Start/stop wave generation.

B. Vessel Monitoring

Live data feed from the ship’s sensors (IMU, wave height, GPS, etc.).
View heave, roll, pitch in real-time.
Show ship heading and velocity.

C. Computational Results

RAO Analysis: Identify resonance conditions.
RMS Motion Visualization: Identify critical motion zones.
MSI Mapping: Highlight discomfort areas on the ship.

D. Visualization & Data Logging

Graphical dashboard for real-time plots.
Download data reports (CSV, JSON, Excel).

2. System Architecture

The web app will communicate with the towing tank control system and the vessel sensors.

A. Hardware Architecture

Communication Flow:

1. User controls towing tank from web app.
2. Web app sends commands to Raspberry Pi (via Flask/Node.js API).
3. Raspberry Pi controls actuators (wave generator, motor).
4. Arduino collects vessel sensors and send real-time data to Raspberry Pi.
5. Raspberry Pi transmits data to web app for visualization.

B. Software Architecture

Component	Technology Stack
Frontend (User Interface)	React.js
Backend API	Flask (Python)
Database	Firebase Realtime DB
Real-time Communication	WebSockets (Socket.io) / MQTT
Embedded Control	Raspberry Pi (Python) + Arduino (C++)
Data Processing	Python (NumPy, SciPy, Pandas, Matplotlib) maybe Matlab

Components school can give

1. Temperature.
2. Relative humidity.
3. RGB LEDs.
4. Flashing LED.
5. Ultrasound.
6. Buzzer.
7. Relay.
8. esp32
9. arduino