**Report: Webpage for Controlling the Robot Base Movement**

**Introduction**

This project aims to control the movement of a robot base using a webpage. It involves creating a database to store direction values and building PHP pages to link the interface with the data. In this report, I will explain each step completed in the project with the relevant code.

**1. Designing the Webpage to Control the Robot Base Movement**

A web interface was designed with buttons to control the robot's movement in different directions (Forward, Backward, Left, Right, Stop). When a button is pressed, the data is sent to the database via PHP.

**Code:**

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Robot Control</title>

<style>

body {

text-align: center;

font-family: Arial, sans-serif;

}

.btn {

padding: 15px 30px;

margin: 10px;

font-size: 18px;

cursor: pointer;

}

</style>

</head>

<body>

<h1>Control the Robot</h1>

<button class="btn" onclick="sendDirection('forward')">⬆️ Forward</button> <br>

<button class="btn" onclick="sendDirection('left')">⬅️ Left</button>

<button class="btn" onclick="sendDirection('stop')">🛑 Stop</button>

<button class="btn" onclick="sendDirection('right')">➡️ Right</button> <br>

<button class="btn" onclick="sendDirection('backward')">⬇️ Backward</button>

<script>

function sendDirection(direction) {

fetch('send\_direction.php', {

method: 'POST',

headers: { 'Content-Type': 'application/x-www-form-urlencoded' },

body: 'direction=' + direction

})

.then(response => response.text())

.then(data => alert(data))

.catch(error => console.error('Error:', error));

}

</script>

</body>

</html>

**2. Creating a Database to Store Direction Data**

A robot\_control database was created containing a directions table, where each direction selected by the user is stored along with the time of insertion.

**Code:**

sql

CREATE DATABASE robot\_control;

USE robot\_control;

CREATE TABLE directions (

id INT AUTO\_INCREMENT PRIMARY KEY,

direction VARCHAR(10) NOT NULL,

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**3. Building a PHP Page to Retrieve the Last Value from the Database**

A PHP page (get\_direction.php) was created to retrieve the last direction stored in the database. This page reads the data from the database and sends it via HTTP to be used later.

**Code:**

php

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "robot\_control";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$sql = "SELECT direction FROM directions ORDER BY id DESC LIMIT 1";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

$row = $result->fetch\_assoc();

echo $row["direction"];

} else {

echo "No data";

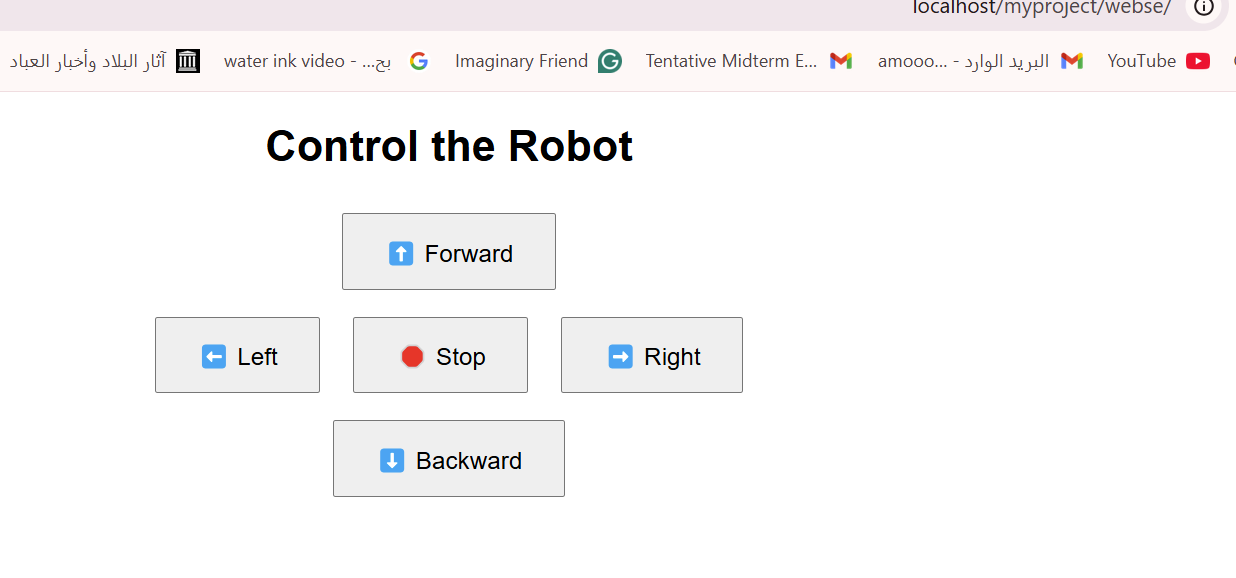
}

$conn->close();

?>

**Conclusion**

All tasks related to designing the webpage for robot control, creating the database for storing directions, and building PHP pages to link the data have been completed. The next steps involve using **ESP32** to receive the data via **WiFi** or **Bluetooth** to control the robot based on this data.



A screenshot of a computer

AI-generated content may be incorrect.