**Introduction:**

The NodeMCU Snake Game is a simple, interactive game where you control the classic snake using a web-based controller. The game runs on the NodeMCU, which acts as both an access point (AP) and a server. WebSocket is used for real-time communication between the game display and the controller.  
This project combines web development, networking, and hardware, showcasing the versatility of NodeMCU.

**Features:**

* **Quick Setup:** Just upload the code to the NodeMCU, connect your devices to its Wi-Fi network, and start playing.
* **No External Display Needed:** The game runs directly on your device's browser, so there's no need for an external display or additional setup.
* **Real-Time Control:** The game uses WebSocket for real-time communication between the controller and display, ensuring the action is fast and responsive.
* **Standalone Access Point (AP):** The NodeMCU creates its own Wi-Fi network, allowing you to connect your devices directly to it without needing an internet connection.
* **Low-Cost Hardware:** The entire game logic runs on the NodeMCU, a low-cost microcontroller that offers multiple functionalities. In this project, it acts as both a server and an access point (AP), providing all the necessary resources for the game to run smoothly without the need for expensive hardware

## **Getting Started**

To get the NodeMCU Snake Game up and running, follow these simple steps:

### **Requirements:**

* **Hardware**:
* NodeMCU (ESP8266)
* USB cable to connect the NodeMCU to your computer
* **Software**:
* Arduino IDE (with ESP8266 board support installed)
* Wi-Fi enabled device (like a smartphone or laptop)

### **Steps:**

1. **Install the Arduino IDE**:

* Download and install the Arduino IDE from the official Arduino website.

1. **Set Up NodeMCU in Arduino IDE**:

* Open Arduino IDE and go to **File > Preferences**.
* In the "Additional Boards Manager URLs" field, add: http://arduino.esp8266.com/stable/package\_esp8266com\_index.json
* Go to **Tools > Board > Board Manager**, search for **ESP8266**, and install the package.
* Select the **NodeMCU 1.0 (ESP-12E Module)** board from **Tools > Board**.

1. **Upload the Sketch**:

* Open the provided realtime-nodemcu-snakegame-sketch.ino file in the Arduino IDE.
* Connect your NodeMCU to your computer using a USB cable.
* Select the correct port from **Tools > Port**.
* Click the **Upload** button to upload the sketch to the NodeMCU.

1. **Connect to the NodeMCU Wi-Fi Network**:

* Once the upload is complete, the NodeMCU will create a Wi-Fi network named **MiniProject**.
* Connect your phone or laptop to this network.

1. **Start Playing**:

* Open your browser and visit [**http://192.168.4.1**](http://192.168.4.1/) to see the game display.
* On another device, visit [**http://192.168.4.1/controller**](http://192.168.4.1/controller) to open the controller page.
* Start controlling the snake and enjoy the game!

### **How It Works**

The NodeMCU Snake Game combines smart hardware and web technologies to deliver a fun and interactive gaming experience. Here’s how everything comes together:

1. **NodeMCU as a Wi-Fi Access Point and Server**

* The NodeMCU creates its own Wi-Fi access point, allowing devices to connect directly to it without needing an internet connection.
* It hosts two web pages:
* **Game Display**: Accessed at http://192.168.4.1, this page displays the Snake game on a browser.
* **Controller**: Accessed at http://192.168.4.1/controller, this page provides directional buttons to control the snake.

1. **Real-Time Updates with WebSocket**

* The NodeMCU processes the entire game logic, constantly updating the snake’s position, generating food, and tracking the score.
* WebSocket technology enables instantaneous communication:
* When you press a button on the controller, the command is sent to the NodeMCU in real-time, and the snake moves immediately.
* Changes in the game, such as the snake’s movement or score updates, are reflected instantly on the display.

1. **Game Logic in Action**

* The NodeMCU handles all the game mechanics, including:
* Moving the snake and detecting collisions (with the walls or itself).
* Generating food at random locations and increasing the snake’s length when it eats the food.
* Resetting the game when a collision occurs and updating the final score.

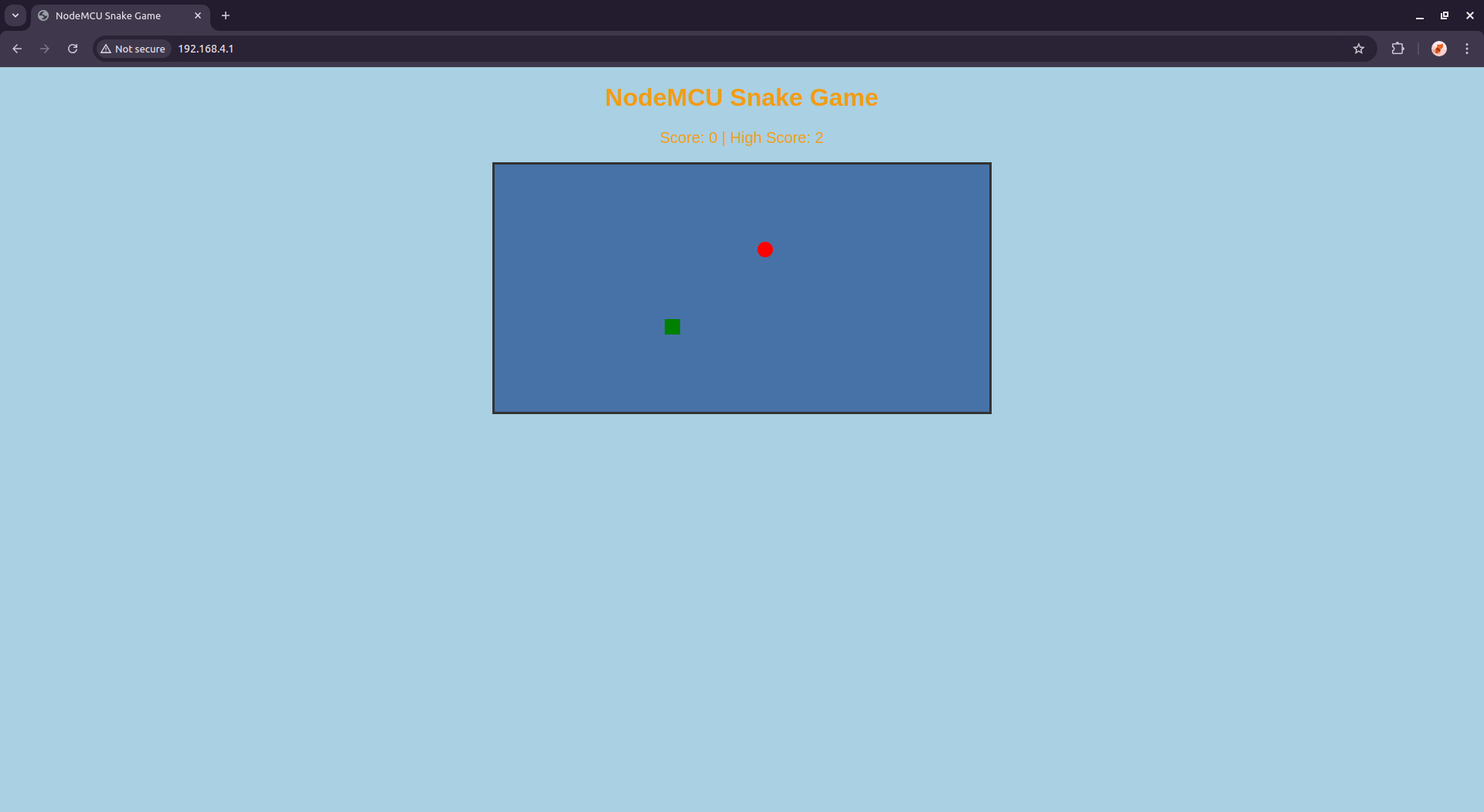
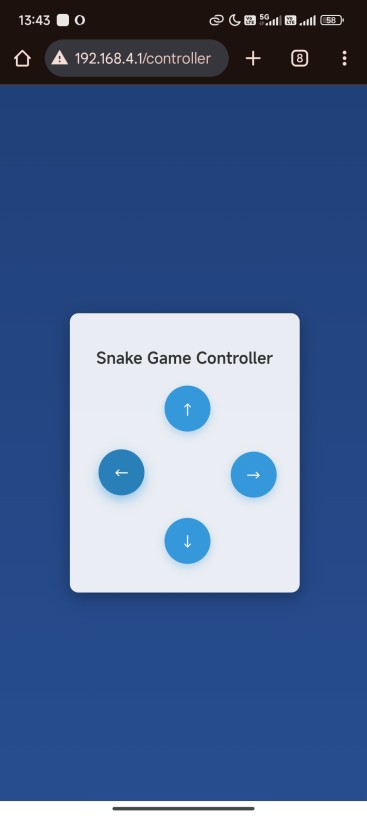
1. **Device Interaction**

* The game requires two devices working together seamlessly:
* One device (e.g., a laptop or tablet) acts as the game display, showing the progress.
* Another device (e.g., a smartphone) acts as the controller, allowing you to steer the snake using directional buttons.

This efficient use of the NodeMCU as a Wi-Fi access point and server highlights its ability to create a standalone, interactive game experience without the need for additional hardware or software dependencies.

**Screenshots :**

**Added these two on the page**



**License:**