Setup

1. Initialize WiFi credentials and sensor pin assignments.
2. Initialize light and fan pin assignments and states.
3. Initialize the DHT sensor and WiFi server.
4. Initialize other variables for tracking state and timing.
5. Start the serial communication.
6. Set light and fan pins as outputs and turn them off.
7. Set sensor pins as inputs.
8. Start the WiFi access point.
9. Print the IP address to the serial monitor.
10. Begin the DHT sensor.
11. Start the WiFi server.

Loop

1. Check for a client connection to the server.
2. If a client is connected:
  - Record the current time and reset the previous time.
  - Print a message indicating a new client connection.
  - Initialize an empty string for the current line.
  - While the client is connected and timeout has not occurred:
    - If the client has sent data:
      - Read the data.
      - Print the data to the serial monitor.
      - Append the data to the header string.
      - If a newline character is received:
        - If the current line is empty (indicating the end of the HTTP request):
          - Send the HTTP response header.
          - Call the `sendHTML` function to send the HTML page.
          - If the mode is "manual":
            - If the header contains "GET /light/on":
              - Turn on the light.

- Update `lightstate` to "on".
- Else if the header contains "GET /light/off":
- Turn off the light.
- Update `lightstate` to "off".
- If the header contains "GET /fan/on":
- Turn on the fan.
- Update `fanstate` to "on".
- Else if the header contains "GET /fan/off":
- Turn off the fan.
- Update `fanstate` to "off".
- If the header contains "GET /mode/manual":
- Set the mode to "manual".
- Else if the header contains "GET /mode/auto":
- Set the mode to "auto".
- Break out of the loop.
- Else:
- Clear the current line.
- Else if the character is not a carriage return:
- Append the character to the current line.
- Clear the header string.
- Stop the client connection.
- Print a message indicating the client disconnected.

3. If the mode is "auto":
- Call the `automode` function.

sendHTML

1. Send the HTML structure to the client.

2. Send the CSS styles.

3. Send the body content and buttons for controlling the light, fan, and mode.

4. Update the button labels and styles based on the current states (`lightstate`, `fanstate`, `mode`).

automode

1. Delay for 2 seconds.
2. Read the proximity sensor.
3. If the proximity sensor is triggered:
   - Toggle the `userin` state.
   - If `userin` is true:
     - Delay for 2 seconds.
     - Read the temperature from the DHT sensor.
     - Read the darkness sensor.
     - If the temperature exceeds `templim1`:
       - Turn on the fan.
       - Update `fanstate` to "on".
     - Else if the temperature is below `templim2`:
       - Turn off the fan.
       - Update `fanstate` to "off".
     - If it is dark:
       - Turn on the light.
       - Update `lightstate` to "on".
     - Else:
       - Turn off the light.
       - Update `lightstate` to "off".
   - Else (if `userin` is false):
     - Turn off the light and fan.
     - Update `lightstate` to "off".
     - Update `fanstate` to "off".