******

*ENGR-UH 1000*

*Computer Programming for Engineers*

**Final project report**

**Correct Posture Electronic (CPE)**

Rayed Khan and Rodrigo Valdivia

Fall 2021

**1.** **Introduction**

Posture can be defined as the position a body has while being sitting, standing or lying down. With good posture, bones are adequately aligned, the strain on ligaments and spine is reduced and muscles are used more effectively (Cleveland Clinic, 2019). Conversely, bad posture has many negative impacts on health. It causes stress on the joints and weakens muscles and ligaments which leads to back pain (Tüzün *et al*, 1999). Around 70% of people will experience some type of medical back pain throughout their lives (Ehrlich, 2003). Back pain has a far-reaching negative, communities, health-care and businesses (Hoy *et al,* 2010). It is exacerbated by sedentary lifestyles, the workplace, low educational status, stress, anxiety, depression, job dissatisfaction and low levels of social support in the workplace (Hoy *et al*, 2010). Daily tasks like sitting or lifting items produce tension that can lead to back pain, and one solution is to improve posture while performing these activities (Harvard Health, 2020). The Canadian Centre for Occupational Health and Safety suggests that the ideal sitting angle has a trunk inclination of 0-30°. This angle will be taken as the range for the project. Bad posture can be treated with different approaches and exercises and technology can be part of the solution.

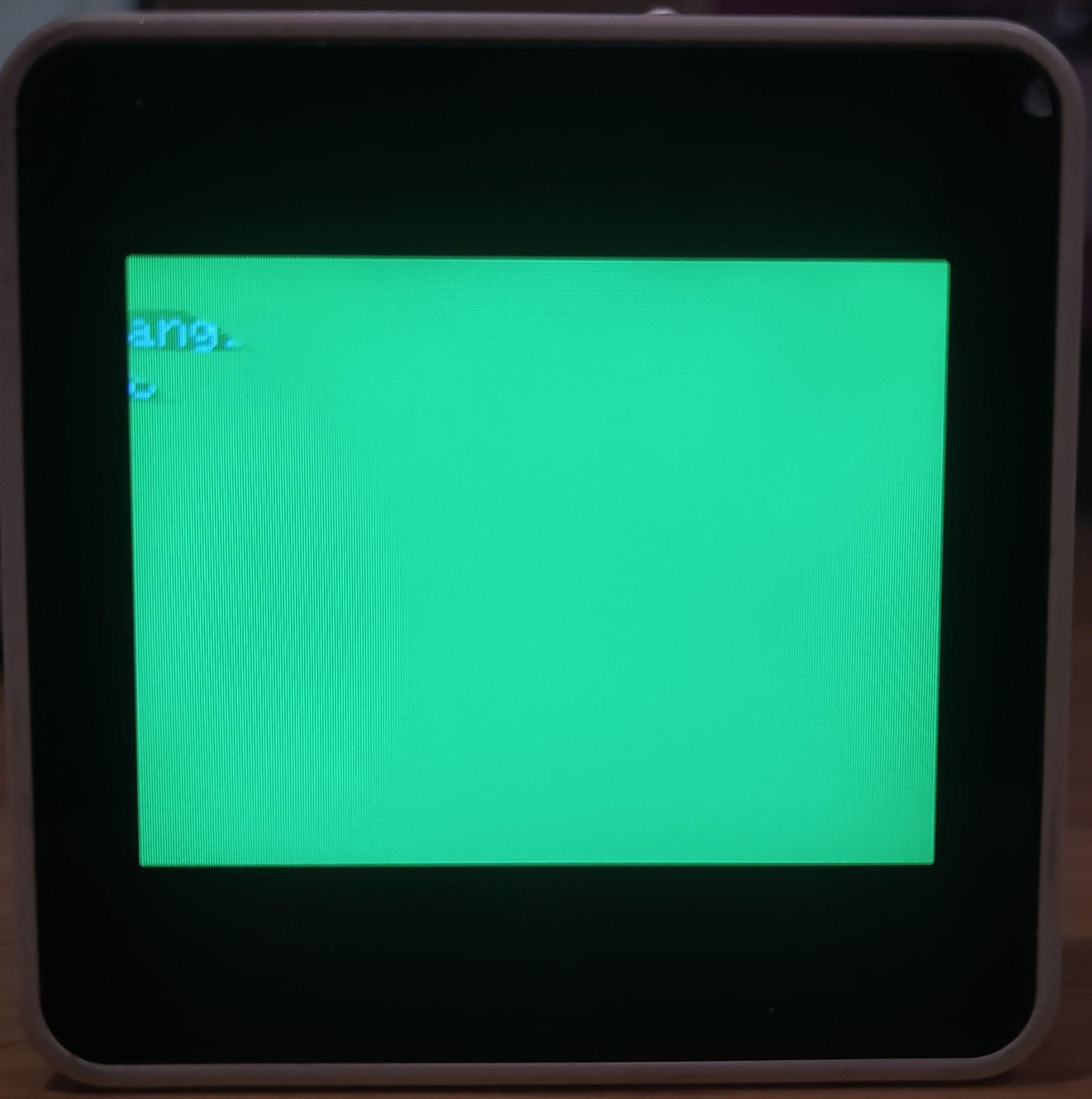
The primary objective of this project is to develop a device that can accurately measure the sitting angle of the user and provide real time feedback to improve their posture. This provides a possible solution to lower back pain which is an issue that is ubiquitous around the world.

**2.** **Project Development**

The device used in the project is the M5 Stack Core 2. It is coupled with the Arduino IDE to use the device libraries and upload and compile code to the program. The functions of the devices that were used for the project are the vibration, speaker, display and IMU.

**Functionality of the device:**

The M5stack is strapped to the chest of the user at the position of the middle of the sternum. After it is turned on, the device asks the user to sit in a position that he considers is comfortable, and records this value. The ideal angle range (0-30°) is then applied to the value of the comfortable sitting position. This ensures that the device can be used by people with different physical characteristics, as the device sits in different angles depending on the structure of the chest of the user. After the value is recorded, the device displays a green screen if the user is sitting in the angle position, and a red screen if the user is sitting beyond the range.



If the user goes beyond the acceptable range for more than 5 seconds, the device vibrates and rings. The 5 seconds allow the user to reach down for things, or perform small movements without the device alerting him that his posture is incorrect.

**Software Algorithm Pseudocode:**

Initialize a standard variable to be of float type and assign a value of 361.0 to it

Perform once when the device is turned on:

Print title “Correct Posture Electronic”

Print appropriate instructions for user to assume ideal/default position and click either button on the M5Stack device

Initialize programs for beep sound output and vibration mechanism

Initialize real-time counter on M5Stack device and set it to zero

Perform continuously in a looping manner:

Input data from IMU sensor of M5Stack device into program

Input data from real-time counter of M5Stack device into program

Print CURRENT roll value of M5Stack device from its IMU sensor under the label “angle”

Check if the value stored within the standard variable is still 361.0, if yes…

Wait for the user to press either button on M5Stack device

When a button is pressed, store corresponding “roll” value from the IMU sensor of M5Stack device into standard variable and print the same

If not…

Check if the standard value is within a dynamic range, if yes…

Display a green screen

Reset real-time counter on M5Stack device

If not…

Display a red screen

Check if number of seconds on real-time counter from M5Stack device is greater than or equal to five, if yes…

Output beep sound and perform vibration of M5Stack device

Reset real-time counter on M5Stack device

**Software Code:**

//inclusion of appropriate libraries

#include <M5Core2.h>

#include <driver/i2s.h>

//setting up of software for beep sound output

extern const unsigned char previewR[120264];

#define CONFIG\_I2S\_BCK\_PIN 12

#define CONFIG\_I2S\_LRCK\_PIN 0

#define CONFIG\_I2S\_DATA\_PIN 2

#define CONFIG\_I2S\_DATA\_IN\_PIN 34

#define Speak\_I2S\_NUMBER I2S\_NUM\_0

#define MODE\_MIC 0

#define MODE\_SPK 1

#define DATA\_SIZE 1024

//setting up of software for IMU input

float pitch = 0.0;

float roll = 0.0;

float yaw = 0.0;

float standard = 361.0;

//declaration of functions required for beep sound output

bool InitI2SSpeakOrMic(int mode){

esp\_err\_t err = ESP\_OK;

i2s\_driver\_uninstall(Speak\_I2S\_NUMBER);

i2s\_config\_t i2s\_config = {

.mode = (i2s\_mode\_t)(I2S\_MODE\_MASTER),

.sample\_rate = 44100,

.bits\_per\_sample = I2S\_BITS\_PER\_SAMPLE\_16BIT,

.channel\_format = I2S\_CHANNEL\_FMT\_ONLY\_RIGHT,

.communication\_format = I2S\_COMM\_FORMAT\_I2S,

.intr\_alloc\_flags = ESP\_INTR\_FLAG\_LEVEL1,

.dma\_buf\_count = 2,

.dma\_buf\_len = 128,

};

if (mode == MODE\_MIC){

i2s\_config.mode = (i2s\_mode\_t)(I2S\_MODE\_MASTER | I2S\_MODE\_RX | I2S\_MODE\_PDM);

}else{

i2s\_config.mode = (i2s\_mode\_t)(I2S\_MODE\_MASTER | I2S\_MODE\_TX);

i2s\_config.use\_apll = false;

i2s\_config.tx\_desc\_auto\_clear = true;

}

err += i2s\_driver\_install(Speak\_I2S\_NUMBER, &i2s\_config, 0, NULL);

i2s\_pin\_config\_t tx\_pin\_config;

tx\_pin\_config.bck\_io\_num = CONFIG\_I2S\_BCK\_PIN;

tx\_pin\_config.ws\_io\_num = CONFIG\_I2S\_LRCK\_PIN;

tx\_pin\_config.data\_out\_num = CONFIG\_I2S\_DATA\_PIN;

tx\_pin\_config.data\_in\_num = CONFIG\_I2S\_DATA\_IN\_PIN;

err += i2s\_set\_pin(Speak\_I2S\_NUMBER, &tx\_pin\_config);

err += i2s\_set\_clk(Speak\_I2S\_NUMBER, 44100, I2S\_BITS\_PER\_SAMPLE\_16BIT, I2S\_CHANNEL\_MONO);

return true;

}

void SpeakInit(void) {

M5.Axp.SetSpkEnable(true);

InitI2SSpeakOrMic(MODE\_SPK);

}

void DingDong(void) {

size\_t bytes\_written = 0;

i2s\_write(Speak\_I2S\_NUMBER, previewR, 120264, &bytes\_written, portMAX\_DELAY);

}

//setting up of software for real-time counter

RTC\_TimeTypeDef TimeStruct;

void setup(){

M5.begin(true, true, true); //Initializes M5Stack device

//formatting of M5Stack display screen

M5.Lcd.fillScreen(BLACK);

M5.Lcd.setTextColor(YELLOW);

M5.Lcd.setTextSize(3);

M5.Lcd.setTextWrap(true, true);

M5.Lcd.setCursor(60,20);

//Printing of title & appropriate instructions to go with project

M5.Lcd.print("Correct Posture");

M5.Lcd.setCursor(70,45);

M5.Lcd.print("Electronic");

M5.Lcd.setTextColor(WHITE);

M5.Lcd.setTextSize(2);

M5.Lcd.setTextWrap(true, true);

M5.Lcd.setCursor(0,120);

M5.Lcd.print("Sit to be in your ideal position and press a button after the beep.");

delay(10000);

SpeakInit();

DingDong();

M5.IMU.Init();

M5.Lcd.fillScreen(BLACK);

M5.Lcd.setTextColor(WHITE, BLACK);

M5.Lcd.setTextSize(2);

//resetting of real-time counter within M5Stack

TimeStruct.Seconds = 0;

M5.Rtc.SetTime(&TimeStruct);

}

void loop() {

M5.IMU.getAhrsData(&pitch,&roll,&yaw); //getting appropriate data from IMU sensor to within the program

M5.Rtc.GetTime(&TimeStruct); //getting appropriate data from real-time counter to within the program

//printing of “roll” value from IMU sensor onto M5Stack display screen

M5.Lcd.setCursor(0, 20);

M5.Lcd.printf("angle");

M5.Lcd.setCursor(0, 40);

M5.Lcd.printf("%5.2f", roll);

//get corresponding “roll” value for default position of user

if (standard == 361) {

M5.update(); //M5Stack ready to detect pressing of either of its buttons

if (M5.BtnA.wasPressed() || M5.BtnB.wasPressed() || M5.BtnC.wasPressed()) {

//store current “roll” value from IMU sensor into “standard” variable and print the same

standard=roll;

M5.Lcd.setCursor(0,100);

M5.Lcd.printf("%5.2f", standard);

delay(5000);

}

}

else {

//if the “roll” value of the user’s current position is within acceptable range

if(roll<(standard+12) && roll>(standard-30)){

M5.Lcd.fillScreen(GREEN);//display a green screen

//resetting of real-time counter within M5Stack

TimeStruct.Seconds = 0;

M5.Rtc.SetTime(&TimeStruct);

}

//if the “roll” value of the user’s current position is outside acceptable range

else{

M5.Lcd.fillScreen(RED); //display a red screen

if (TimeStruct.Seconds >= 5) { //check if number of seconds on real-time counter of M5Stack is equal to five seconds

//perform vibration of M5Stack device and output a beep sound

M5.Axp.SetLDOEnable(3,true);

delay(500);

M5.Axp.SetLDOEnable(3,false);

M5.Axp.SetLDOEnable(3,true);

delay(100);

M5.Axp.SetLDOEnable(3,false);

DingDong();

//resetting of real-time counter within M5Stack

TimeStruct.Seconds = 0;

M5.Rtc.SetTime(&TimeStruct);

}

}

}

delay(10); //ensures delay (of 10ms) between runnings of loop

}

**3.** **Results and Evaluation**

**Results**

The device functionality was tested with 5 different people to ensure that it was adaptable to different physical characteristics. Each person was asked to go beyond the acceptable range first for a few seconds, and then for more than 5. This tested the function that allows the user to go beyond the inclination for smaller movements or reaching out for things. The users were then asked to sit with a purposely slouched position to test if the angle selected was suitable for the project.

The device was capable of adapting to the different chest structures of the people that tried the device, as the tests were successful. The function to delay the sound and vibration of the code for 5 seconds worked, and sitting with a purposely slouched position made the angle go out of the acceptable range which was the main objective of the device. The users thought the device was easy to use.

**Challenges**

There were two major challenges in the development of the project. One related to the software, and the other to the background medical research. After testing the device on ourselves, the product only worked on one person and the ranges had to be hardcoded, so we had to devise a solution for the device to adapt to physical differences. We solved this problem by getting a reference angle for every user’s comfortable sitting angle when the device is first started. Then we applied the ideal angle range to this starting angle. The device then was able to adapt to different chest structures. The other challenge was getting the ideal sitting angle, as many articles included different ranges. We settled for the angle 0° to 30° of trunk inclination after testing the device on different people.

**4.** **Conclusion and Future work**

The project’s result was a device that could accurately measure the sitting angle position and adapt to people’s physical differences to provide real-time feedback about the user’s posture. This device can be used as a solution by people that tend to have postural problems which can lead to lower-back pain.

This project can be further improved by conducting user feedback studies to develop a more comprehensive and user-friendly solution. The opinions that would be collected would be related to the feedback mechanism and the size of the device. The device displays a red screen, vibrates, and beeps when the user is sitting with an incorrect posture, which could lead to interruptions, especially in the workplace. The software could be easily modified to suit these demands, and a good solution would be to allow the user to choose the device’s feedback. The device size could be considered too big for its function, so another device could be explored. Finally, to make the device more valuable for the user, the device could record data throughout the day to report patterns, and times of incorrect body posture. This data could also be helpful for physicians that are addressing their patient’s postural problems.

**5.** **Reflection on Learning**

The main learning outcome of the project was learning how to adapt the c++ logic we learned in class and combine it with functions from the M5Stack library. This showed us that what we learned can be used for other devices, languages and real world applications. We learned that to develop a useful product, especially one that has biofeedback, its reasons and working have to have a scientific base. We arrived at the conclusion that to develop a product you need to take into account user feedback and also use it with different test subjects to ensure that it is suitable for the majority of your target audience.

**References:**

Canadian Centre for Occupational Health and Safety. (2010). Working in a Sitting Position - Good Body Position. <https://www.ccohs.ca/oshanswers/ergonomics/sitting/sitting_position.html>

Ehrlich, G. (2003). *Back Pain.* The Journal of Rheumatology. Vol. 67;26-31

Tüzün, Ç., Yorulmaz, İ., Cindaş, A. *et al.* (1999). *Low Back Pain and Posture*. *Clin Rheumatol*. Vol 18, 308–312. <https://doi.org/10.1007/s100670050107>

Harvard Health. (2020). *4 ways to turn good posture into less back pain.* Harvard Health.

<https://www.health.harvard.edu/pain/4-ways-to-turn-good-posture-into-less-back-pain>

Hoy, D, Brooks, P, Plinth, F, Buchbinder, R. (2010). *The Epidemiology of low back pain.* Best Practice & Research Clinical Rheumatology. Volume 24, Issue 6, 769-781. <https://doi.org/10.1016/j.berh.2010.10.002>.

Will, J. S., Bury, D. C., & Miller, J. A. (2018). Mechanical low back pain. American Family Physician Journal, 98(7), 421–428.