VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi – 590 018.

A PROJECT REPORT

on

"IoT BASED SMART CHAIR USING RASPBERRY PI"

Submitted in partial fulfillment of the requirement for the award of the degree

Bachelor of Engineering

in

Computer Science and Engineering

by

DHANUSH B	**:**	**1VI19CS026**
GOWTHAM V	**:**	**1VI19CS032**
HAFSA MOHAMMADI	**:**	**1VI19CS034**
KOMAL BHAT	**:**	**1VI19CS046**

Under the supervision of

Dr. Kantharaju H C
Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VEMANA INSTITUTE OF TECHNOLOGY

BENGALURU – 560 034

2022 - 23

Karnataka ReddyJana Sangha®

VEMANA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi)

Koramangala, Bengaluru-34.

Department of Computer Science and Engineering

Certificate

Certified that the project work entitled "**IoT BASED SMART CHAIR USING RASPBERRY PI**" carried out jointly by **Dhanush B (1VI19CS026), Gowtham V (1VI19CS032), Hafsa Mohammadi (1VI19CS034), Komal Bhat (1VI19CS046),** are bonafide students **of Vemana Institute of Technology** in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** of the **Visvesvaraya Technological University, Belagavi** during the year 2022-23. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the said degree.

Supervisor	**HOD**	**Principal**
Dr. Kantharaju H C	Dr. M. Ramakrishna	Dr. Vijayasimha Reddy. B. G.

Submitted for the university examination (viva-voce) held on ……………………….

External Viva

Internal Examiner	**External Examiner**
- - - - - - - - - - - - - -	- - - - - - - - - - - - - -
- - - - - - - - - - - - - -	- - - - - - - - - - - - - -

DECLARATION BY THE CANDIDATES

We the undersigned solemnly declare that the project report “IoT BASED SMART CHAIR USING RASPBERRY PI” is based on our own work carried out during the course of our study under the supervision of ‘Dr. Kantharaju H C.

We assert the statements made and conclusions drawn are an outcome of our project work. We further certify that,

a. The work contained in the report is original and has been done by us under the general supervision of my supervisor.

b. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.

c. We have followed the guidelines provided by the university in writing the report.

d. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and their details are provided in the references.

Date:

Place:

Project Team Members:

Dhanush B	1VI19CS026
Gowtham V	1VI19CS032
Hafsa Mohammadi	1VI19CS034
Komal Bhat	1VI19CS046

ACKNOWLEDGEMENT

First we would like to thank our parents for their kind help, encouragement and moral support.

We thank **Dr. Vijayasimha Reddy. B. G,** Principal, Vemana Institute of Technology, Bengaluru for providing the necessary support.

We would like to place on record our regards to **Dr. M. Ramakrishna,** Professor and Head, Department of Computer Science and Engineering for his continued support.

We would like to thank our project coordinators **Mrs. Mary Vidya John,** Assistant Professor and **Mrs. J Brundha Elci,** Assistant Professor, Dept. of CSE for their support and coordination.

We would like to thank our project guide **Dr. Kantharaju H C,** Associate Professor, Dept. of CSE for his continuous support, valuable guidance and supervision towards successful completion of the project work.

We also thank all the Teaching and Non-teaching staff of Computer Science and Engineering Department, who have helped us to complete the project in time.

Project Team Members:

DHANUSH B	**(1VI19CS026)**
GOWTHAM V	**(1VI19CS032)**
HAFSA MOHAMMADI	**(1VI19CS034)**
KOMAL BHAT	**(1VI19CS046)**

CONTENTS

ABSTRACT

The Internet of Things (IoT) describes the network of physical objects "things" that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. This project helps us to assess the posture of a seated person in real-time and improve sitting posture. Long term sitting harms the spine and causes chronic problems that need long time therapy. Diseased people have a significant impact on office productivity. Long periods of sitting while studying cause major postural and spinal problems in young people. Our motivation is to help people pay attention to their health and proper sitting in addition to work. Sitting with a straight back and shoulders not only improves a person's physical health but also makes them feel more confident. If a person is sitting straight and taking frequent breaks in between, this will eventually result in healthy habits that will increase productivity. Currently, the problem we are facing is prolonged sitting with improper posture. We use an IoT system with sensors, processing power, and software to make the chair smart so that it can eventually help humans maintain good posture and healthy habits. Making a regular chair into a smart chair using Raspberry Pi, which will continuously detect your sitting posture and send a notification if it is incorrect. Two flex sensors will be fitted onto the backrest of the chair to detect the inclination of the chair. The backrest sensor will detect if the person is sitting straight or not. The humidity and temperature sensor module on the cushion of the seat will detect the temperature and pressure of the user. The goal of this project is to maintain proper posture for improved health and longevity.

Keywords : Posture, Raspberry Pi, Smart Chair, Humidity, Temperature, Flex

LIST OF ABBREVIATIONS

Abbreviation	**Description**
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
ATS	Applicant Tracking System
DHT	Digital Temperature and Humidity
DIY	Do It Yourself
DL	Deep Learning
DSR	Design Science Research
EMN	Echo Memory Network
ERT	Extremely Randomized Trees
ESP	Event Stream Processing
FSR	Force Sensing Resistors
GPIO	General Purpose Input/Output
HDMI	High-Definition Multimedia Interface
IOS	iPhone Operating System
IoT	Internet of Things
I2C	Inter Integrated Circuit
KNN	K Nearest Neighbor
MATLAB	Matrix Laboratory
MCU	Multipoint Control Unit
MEMS	Micro Electro Mechanical System
MLP	Multi-Layer Perception

MQTT	Message Queue Telemetry Transport
OS	Operating System
PCA	Principal Component Analysis
SB	Sedentary Behavior
SBC	Single Board Computer
SCG	Scale Conjugate Gradient
SVM	Support Vector Machine
USB	Universal Serial Buss

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

High levels of sedentary behavior (SB, sitting or lying down with little energy expended while awake) have been associated with negative health consequences, largely regardless of the amount of physical activity. SB is prevalent all day long. SB is particularly common during workplace. In fact, it was discovered that office workers conduct sedentary activities for up to 71% of their working hours. The majority of modern jobs need extensive desk-based sitting. In light of this, workplace SB may pose a risk to one's occupational health. Adults frequently aren't aware of their extended SB. The usage of monitoring SB devices can help raise awareness of excessive sitting. Adults spend a lot of time at work; thus, the workplace is an excellent arena for providing feedback on their prolonged sitting [1].

Authors[2] suggested an Internet of Things (IoT) application that combines a chair with sensors and an integrated IoT device to evaluate posture and alert the user of any changes. Through feedback,users will be able to assess their own sitting patterns and improve the effectiveness of posturecorrection.

Modern humans spend the majority of their time sitting, and bad posture can lead to poor head/neck alignment and increased cervico-thoracic muscle activity, which can lead to health issues, especially for people who work long hours. For instance, maintaining a neutral lumbar posture is crucial for wellness. The curved lower lumbar spine that results from poor sitting posture for comfort, however, may increase health risks. People spend an average of 13 hours a day sitting down, according to a survey. But prolonged sitting can raise your risk of obesity and metabolic disorders [3].

Long-term poor posture has an impact on our personal and professional lives. Research from the University of Washington found that 45% of Americans between the ages of 35 and 55 experience acute back pain annually. Back pain is the leading cause of disability in Americans, posture strains the neck and spine and increases muscular tension as the body tries to make up for the lack of support. [4].

In short, the complications of poor posture include back pain, spinal dysfunction, joint degeneration, rounded shoulders, and a potbelly. Poor posture is dogging people of all ages. It's a common and an important health problem and it can lead to neck pain, back problems, and other aggravating conditions, says Meghan Markowski, a physical therapist at Harvard-affiliated Brigham and Women's Hospital. Turning a normal chair into a smart chair that will continuously detect your sitting posture and sends a notification for wrong postures. We can measure the inclination made by the chair when a person sits on it and our sensor will tell if it is the correct posture and wrong. It can also employ the temperature and humidity module and measure the parameters and send notifications to the app informing that the user has to take a break. Smart chair can detect our postures and send notifications to the app if our postureis wrong and when to take a break [5].

In the era of ever-advancing technology, the concept of IoT has brought aparadigm shift in the way we interact with our surroundings. One of the promising applicationsof IoT is in the development of smart chair, where sensors and electronics are integrated into everyday items to provide enhanced functionality and convenience [6].

Authors[7] explores the development of an IoT-based smart chair using Raspberry Pi, a low-cost and widely popular single-board computer. With the help of smart sensors, actuators, and Raspberry Pi's processing power, this smart chair is capable of collecting and analyzing data related to comfort and health factors, such as sitting posture, temperature, and humidity. The collected data can then be utilized to enhance the chair's functionality, such as adjusting the temperature and backrest to suit individual preferences or alerting users of poor sitting posture to prevent health issues. Overall, this paper not only presents the design and implementation of a prototype smart chair but also highlights the potential of IoT in revolutionizing everyday objects.

Authors[8] suggest that smart chairs are needed because of our poor posture problems and the need for a chair that can actively monitor and adjust to our comfort needs. The development of IoT-based smart chairs has an immense potential to improve the quality of life for people, especially those with mobility or health issues.

Poor posture during office work is a common problem that affects many people who spend long hours sitting in front of a computer. According to a study by the Occupational Safety and Health Administration (OSHA), poor posture can lead to more musculoskeletal disorders (MSDs), which are a major cause of workplace injuries and illnesses [9].

MSDs are a group of conditions that affect the muscles, tendons, ligaments, nerves, and joints, and they can cause pain, discomfort, and disability. One of the most common MSDs associated with poor posture is lower back pain. A study by the National Institute for Occupational Safety and Health (NIOSH) found that workers who spend long hours sitting in front of a computer are at a higher risk of developing lower back pain [10].

Sitting for long periods of time can put pressure on the lower back and cause muscle fatigue, which can lead to pain and discomfort. Poor posture can also cause neck and shoulder pain. According to a study by the American Physical Therapy Association (APTA), sitting with a forward head posture can increase the load on the neck and shoulder muscles, which can cause pain and discomfort [11].

Hunching over a computer can cause the shoulders to round forward, which can lead to tightness and pain in the upper back and neck. Another adverse effect of poor posture during office work is decreased productivity. A study by the University of California, San Francisco found that workers who sit with poor posture are more likely to experience fatigue,which can lead to decreased productivity [12].

Poor posture can cause discomfort and distraction, which can make it difficult to focus on work tasks. Poor posture during office work can have adverse effects on workers' health and productivity. It is important for workers to maintain good posture and take breaks to stretch and move throughout the day. Employers can also help prevent poor posture by providing ergonomic workstations and promoting workplace wellness programs. By addressing poor posture, employers can improve workers' health and well-being, and increase productivity and job satisfaction [13].

1.2 Scope

In this project, we develop an IoT-based smart chair using the Raspberry Pi. It will assess the posture of a seated person in real-time and improve sitting posture. If a person is sitting for more than half an hour, it will notify the user to take a stretch or exercise for a healthy lifestyle. Our project targets people who spend almost all of their working time in a sitting position, as it can lead to neck pain, back problems, and other aggravating conditions. Using an IoT system with sensors, processing power, and software to make the chair smart so that it can eventually help humans maintain good posture and healthy habits. The integration of internet-of-things technology with everyday devices has led to the development of smart devices that can monitor and analyze data for improved functionality and user experience.

One such innovative application can be seen in the development of a smart chair using Raspberry Pi, which has vast potential for various industries and use cases. The IoT-based smart chair can be equipped with sensors to collect data on various aspects such as sitting posture, duration of use, pressure points, and user behavior patterns. With the help of Raspberry Pi's processing power and connectivity features, this data can be analyzed in real- time to provide insights on user comfort levels and potential health risks stemming from prolonged sitting. This smart chair can have multiple potential applications, such as in office environments to improve employee productivity and wellness, in healthcare facilities for patient care, or even in homes for individuals with disabilities or mobility issues. Moreover, an android app which lets us control the chair remotely can be developed using Raspberry Pi's Wi- Fi module, enabling users to monitor their sitting habits and adjust the chair settings accordingly.

1.3 Objective

The main design of the proposed system is to build a system to track posture and be able to know when the user has to take a break, stretch, exercise etc. An application is implemented with the smart chair to know the activities happening in smart chair and get appropriate notifications. The objective of an IoT-based smart chair using Raspberry Pi is to create a connected device that can monitor and improve the sitting habits of its users. The smart chair can collect data on the user's posture, sitting time, and other metrics, and then provide feedback to the user in real- time. This feedback could include alerts when the user is sitting for too long, suggestions for posture improvements, and other personalized recommendations to help the user maintain healthy sitting habits. This can improve the overall user experience and provide added convenience and control over various aspects of the environment. Overall, the purpose of an IoT-based smart chair is to leverage the power of connected technology to help users maintain healthy sitting habits and improve their overall quality of life.

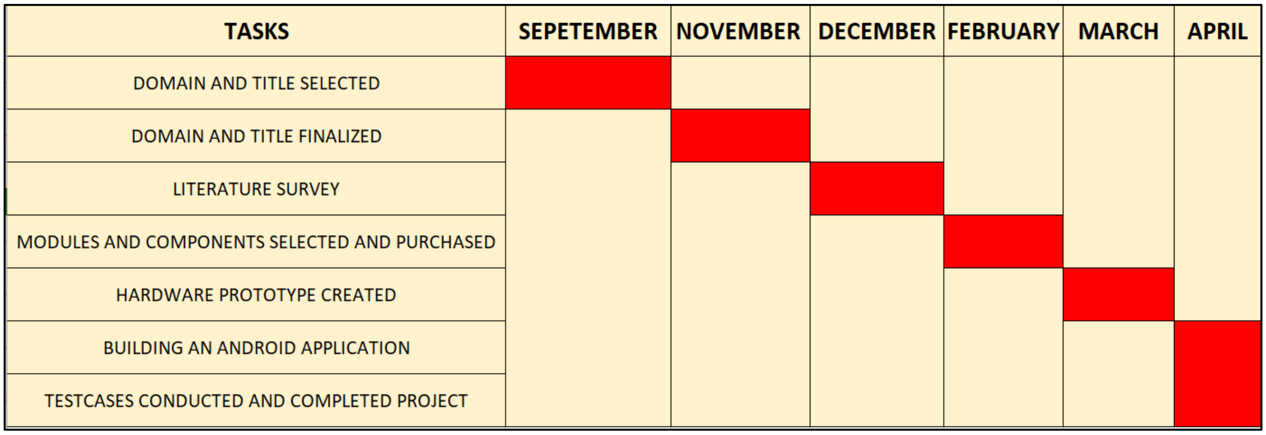
1.4 Organization of the project work

On November 6th 2022, the project domain and title were finalized. After approval by the project guide and the project committee, an extensive literature survey was conducted. In literature survey, papers having similar objectives as those of our project were studied and analyzed. After examining papers from several leading journals, few papers were identified for principal literature survey. The literature survey was followed by identifying the necessary technologies, sensors and implementations of their smart chair. After identifying these resources, principal development of the application started.

1.4.1 Completion timeline

1) In the month of September 2022, the domain of the project and title of the project was discussed.
2) During the month of November, the project title and domain were finalized and approved.
3) In the month of October and November, the literature survey was done based on title of our project.
4) In the month of December, we decided the modules and the components to be used were selected and purchased.
5) In the month of February, we learnt the raspberry pi connections, operating system installations and created a hardware prototype of our project.
6) In the month of March, work on our software, that is the android application began.
7) The project was completed during the month of April and the IoT Based Smart Chair using Raspberry Pi was implemented on the chair and application was created successfully.

Table 1.1 Timeline Chart



TASKS	SEPETEMBER	NOVEMBER	DECEMBER	FEBRUARY	MARCH	APRIL
DOMAIN AND TITLE SELECTED	🟥					
DOMAIN AND TITLE FINALIZED		🟥				
LITERATURE SURVEY			🟥			
MODULES AND COMPONENTS SELECTED AND PURCHASED				🟥		
HARDWARE PROTOTYPE CREATED					🟥	
BUILDING AN ANDROID APPLICATION						🟥
TESTCASES CONDUCTED AND COMPLETED PROJECT						🟥

The Table 1.1 shows the time taken to complete the project which includes, domain and title selection, literature survey, purchasing of components, creating hardware prototype, building an android app and conducting tests.

1.4.2 Outline of the Chapters

CHAPTER 2:

This chapter provides an explanation about the literature survey that was carried out regarding the project. The chapter provides a detailed analysis regarding the methodology utilized in each paper. The individual paper analysis is followed by a tabular comparative analysis of the performance metrics of each paper.

CHAPTER 3:

Chapter 3 examines existing systems and their drawbacks. It also provides a brief explanation regarding the proposed system and analyses its feasibility.

CHAPTER 4:

Chapter 4 examines the system requirements, including the hardware, software, functional and non-functional requirements, and the programming language and the sensors that have been used to develop the IoT Based Smart Chair using Raspberry Pi.

CHAPTER 5:

This chapter provides a description about the project including the system architecture, data-flow diagram and module description.

CHAPTER 6:

This chapter provides an explanation about the implementation of the IoT Based Smart Chair using Raspberry Pi and snapshots of its working.

CHAPTER 7:

This chapter specifies the test cases that were used to determine the working efficiency of the IoT Based Smart Chair using Raspberry Pi.

CHAPTER 8:

This chapter provides concluding remarks about the IoT Based Smart Chair using Raspberry Pi and specifies future enhancements.

CHAPTER 2

LITERATURE SURVEY

2.1 Paper 1

Survey 1: A Smart System for Sitting Posture Detection Based on Force Sensors and Mobile Application [14]

Author: Slavomir Matuska, Martin Paralic, and Robert Hudec

Methodology: Sitting posture detection based on force sensors and mobile applications. Six flexible force sensors, two on the backrest and four on the bottom seat.

Description: This paper presents a smart IoT system for sitting posture detection based on force sensors and mobile applications. Six flexible force sensors, two on the backrest and four on the bottom seat, were embedded in the office chair. Node MCU board was used to measure the sensor' s resistance and sends the data to the cloud using the MQTT protocol. The data are stored and evaluated on the cloud using Node-RED and MongoDB. The user can see the information about sitting posture correctness and other pieces of detailed information in the mobile application.

Advantage: User can see the information about sitting posture correctness and other pieces of detailed information in the mobile application.

Limitation: Implementing the force sensors into the chair. For Apple users, developing an application for IOS or use of the multiplatform framework will become difficult to implement.

2.2 Paper 2

Survey 2: A Sitting Posture Monitoring Instrument to Assess Different Levels of Cognitive Engagement [15]

Author: Daniele Bibbo, Marco Carli, Silvia Conforto and Federica Battisti

Methodology: A set of textile pressure sensors both on the backrest and on the seat of the chair.

Description: In this paper, a new system to evaluate the posture of a seated person is presented. The tests demonstrate that this system can be used in situations whenever different level of engagement, that can drive different levels of stress, occur. In fact, the different levels of engagement produce, in general, a different positioning on the chair of the participants, that moved from the required task. The designed system is based on the use of simple textile sensors and of low-cost electronic equipment that can be easily embedded in the chair structure. This, combined with a wireless transmission, can delineate a final product that could be part of a work environment, in which stressing situations can occur for different and not expectedreasons, thus increasing the level of safety.

Advantage: Use of simple textile sensors and of low-cost electronic equipment that can be easily embedded in the chair structure.

Limitation: The selected sensors were only used as a measure to maximize the reliability of the results.

2.3 Paper 3

Survey 3: Design and Implementation of Smart IoT Chair for Better Health and Productive Work [16]

Author: Timon Padberg, Sojat Sebgaze, Rim Zakar, and Rajeev Kanth

Methodology: The microcontroller used here is the Joy-it Mega2560R3 (Joy-IT), which is like the Arduino Mega.

Description: An overview of the code that is intended to transfer data from Arduino to the USB port. They have two pins that will receive the data in that situation. Then, they have a number of if conditions. For instance, the timer begins to run when the mass of the person sitting in the chair surpasses 20 kg. The second purpose of them is to signal when a person has been seated on a chair for an extended period. The user should be informed that there is already a time limit in that situation. The timer should not be on when the user is not seated on the chair, which is the final function.

Advantage: Cheap and straightforward electronic materials.

Limitation: The notification for the user is not implemented.

2.4 Paper 4

Survey 4: An IoT and Cloud Enabled Smart Chair for Detection and Notification of Wrong Seating Posture [17]

Author: Brijesh Kundaliya, Smit Patel, Jaanvi Patel, Parv Barot, S. K. Hadia

Methodology: Force sensor and flex sensor is used to detect the wrong sitting posture

Description: A smart chair equipped with embedded IoT and cloud technology is developed for detecting and notification of wrong sitting posture. It is identified that only a force sensor is not enough to get the perfect idea of the wrong sitting posture. During backward inclination, the force sensor produced different outputs for the different people having different weights for the same column. To avoid this, the combination of the force sensor and flex sensor is used to detect the wrong sitting posture. The proposed work provides seamless connectivity and a notification mechanism. Not only does it notify the person about the wrong sitting posture but also it maintains the database on the cloud using the Blynk 2.0 platform. The database can be used for medical diagnosis purposes.

Advantage: Computation and data storage has been done on the server rather than locally on user's device which reduces power consumption and memory consumption.

Limitation: If the person is sitting at the edge of the chair, then it is impossible to get reading from the sensor.

2.5 Paper 5

Survey 5: Smart Posture Detector Using IoT [18]

Author: Vidhya. B, Abdul Hayum. A, Saranya. G, Aarthi. M, Aishwarya. V, Dhanashree. K

Methodology: Uses a mems sensor to detect the posture and notifies the user about the same by mentioning the status of it as a bad or worse posture with the help of a voiceover.

Description: This project mainly aims at creating an awareness about the importance of maintaining a good posture to avoid the problems that would take place if it's not been given proper care. The project focus on detecting the user's posture and telling the position of it as ok or worse. Along with the detection on a chair, we have made it wearable by which the kit can be carried anywhere and be detected where it's not necessary that the user should be in a seating position. The chair system is made for the old, aged people by taking their comfort into mind. Also, the data saved in the app by IOT serves as a record for the observations that helps in the cope of improvement for people undergoing physiotherapyor exercises.

Advantage: Along with the detection on a chair, they have made it wearable by which the kit can be carried anywhere and be detected where it's not necessary that the user should be in a seating position.

2.6 Paper 6

Survey 6: Development of a Smart Chair Sensors System and Classification of Sitting Postures with Deep Learning Algorithms [19]

Author: Taraneh Aminosharieh Najafi, Antonio Abramo, Kyandoghere Kyamakya and Antonio Affanni

Methodology: The system consists of eight pressure sensors placed on the chair's sitting cushion and the backrest. The performance of the designed sensors system was evaluated with seven deep learning model

Description: In this research study, a smart chair sensors system was designed, realized, and tested with an experiment involving 40 subjects. A large dataset was created with the acquired data. The performance of the designed sensors system was evaluated with seven deep learning models for eight sitting postures classification and secured by k- fold cross validation. Results of all DL models were compared, and the best average accuracy of 91.68% was achieved by an EMN model, obtained in 5-fold, each fold lasting 27 min for computations and to train the 162,248 trainable parameters. The MLP model, instead, achieved the average accuracy of 90.83%, obtained in 5-fold, each fold lasted 3 min for computations and to train 480 trainable parameters. This second one can be considered an appropriate trade-off model for our application in terms of computational cost vs. accuracy.

Advantage: low number of deployed pressure sensors, the adopted design is easily applicable to many types of chairs and armchair. no cables are required

Limitation: Parameters like emotion, behavior and activity identification were not considered.

2.7 Paper 7

Survey 7: Smart Chair for Monitoring of Sitting Behavior [20]

Author: Mengjie Huang, Ian Gibson, and Rui Yang

Methodology: The sensor array was composed of two layers of polyester films, with the horizontal silver electrode strips and the other vertical on one layer. This orthogonal zebra pattern resulted in a network of silver strips with a resistive unit at each crossing of the sensor array.

Description: Smart chair system was built in this paper to detect sitting posture of human body. The experimental result showed that the overall classification rate of eight sitting postures is high using ANN classifier. The developed smart chair system can monitor the sitting behavior of human body and help in advocating better sitting habits of users. the ANN classifier was trained with the pressure data of known postures. 40 sets of collected pressure data of each position were used as the training sets of ANN classifier. In order to acquire the best overall classification accuracy, various parameters were tested in the first place to find out the optimal settings of the ANN, including the number of neurons in the hidden layer, the transfer function and the network training function. Based on the test results, 20 neurons in the hidden layer, logarithmic sigmoid transfer function and scaled conjugate gradient (SCG) backpropagation network training function were adopted for the ANN classifier in this study. The training performance of the ANN classifier is shown in Figure 5, with mean square error (0.00989) less than the goal (0.01) after 166 epochs. With the properly trained ANN classifier, another 40 sets of pressure data of each position were used for verification of the classifier.

Advantage: With the optimal settings of neural network, the overall classification accuracy of eight sitting postures reached 92.2%. The pressure data of different postures was the input data and the classified positions were the output of the ANN classifier. To avoid the effects of different weights of subjects on the position classification results, the acquired pressure data of eight sitting postures were normalized prior to data processing

Limitation: Hip location on the pressure pattern recognition was not considered to improve the classification rate of sitting postures.

2.8 Comparative analysis

Table 2.1 Comparative Analysis

Ref	Algorithm Technique	Platform used	Performance Metrics	Advantage	Drawback
[14]	Force sensors	Arduino	Six single-zone force sensing resistors FSR402 are used to measure the force.	View information about posture and other info on the mobile application	Implementing force sensors, developing app for IOS users

[15]	Textile pressure	An embedded platform based on a micro controlle runit (MCU)	ATSs pressure trend over time. If the signal is under the threshold, the sensor is pressed.	This system can be used in situations with different level of engagement, different levels of stress.	Not taken the dynamic parameters of the posture
[16]	Load sensors, amplifier, combinator	Arduino Mega micro controller	The load sensors detect weight, and the timer starts to count sitting time	Cheap and straightforward electronic materials.	Notification to users not implemented
[17]	Force sensor, flex sensor	ESP 32	Pressure sensor for recording the data when a person sits on the chair.	Provides connectivity and notification. It also maintains the database on the cloud	Edge of the chair readings impossible to get reading from the sensor.
[18]	IR Sensor, MEMS Sensor	Arduino	MEMS sensor to detect the pressure, IR sensor to detect the posture.	They have made wearable by which the kit can be carried anywhere	
[19]	SVM, PCA, KNN, ANN, Optimization algorithm, EMN	Printed Circuit Board, Arduino	Accuracy of 91.68% by an echo memory network model	Number of pressure sensors are less, no cables required	Only classifies posture as proper or not
[20]	ANN	Arduino and Jupyter Notebook	The pressure data of 5 trials	Accuracy of eight sitting postures reached 92.2%.	Hip location on the pressure pattern recognition was not considered

2.2 Summary of Literature Survey

The papers mentioned above present various studies and applications of smart office chairs with sensors that detect sitting positions and habits. These chairs are intended to monitor the user's posture and provide feedback in order to promote healthy sitting habits and increase workplace productivity. The research involves the design and implementation of smart posture monitoring systems, the classification of sitting postures using deep learning algorithms, and the use of IoT and cloud-enabled smart chairs for detecting and notifying incorrect seating posture. Furthermore, some studies investigate the use of smart chairs to assess different levels of cognitive engagement and promote better workplace health and productivity. Overall, these papers show the potential advantages of incorporating smart chair technology into the workplace to improve sitting posture and overall well-being. While the papers listed above present innovative solutions for improving sitting posture and promoting healthy behaviors in the workplace through the use of smart chairs, they also have some limitations that are common in this field of study. For starters, many of these studies use small sample sizes or are conducted in laboratories, which may not accurately reflect real-world conditions or the diverse range of people who may use these chairs. Second, the accuracy and dependability of the sensors used in smart chairs can be a constraint. Some studies have found problems with the sensors, such as incorrectly detecting sitting positions or failing to detect changes in posture. Third, the cost of implementing smart chair technology may be a barrier to wider adoption, especially in smaller organizations or for individuals. Finally, some studies have emphasized the need for additional research into the long-term effects of smart chairs on user behavior and health outcomes.

In summary, while the papers present innovative solutions for improving sitting posture and promoting healthy behaviors, they also highlight the importance of ongoing research and development to address some of the limitations and challenges associated with smart chair technology.

CHAPTER 3
SYSTEM ANALYSIS

3.1 Existing System

The existing systems on smart chairs vary from simple cushions with vibration feedback to complex chairs with built-in sensors and actuators. Some chairs feature pressure sensors, accelerometers, and gyroscopes to detect the user's posture and movement, while others use a combination of sensors and cameras. The collected data is then analyzed to provide feedback to the user in the form of vibration, sound, or visual cues. However, most of these systems lack the capability to provide continuous feedback and correction over time, and some require the user to wear additional sensors or devices. Moreover, the existing systems may not be cost-effective or easily accessible to the general public. Therefore, there is a need for a smart chair system that can provide continuous posture correction and feedback to the user in a cost-effective and accessible way.

3.1.1 Drawbacks

Some of the drawbacks of existing smart chair systems include limited posture correction capabilities, lack of personalized feedback, high cost, and limited connectivity options. Many of these systems only provide basic posture correction feedback, such as a reminder to sit up straight, and do not offer personalized feedback based on the user's individual posture habits. Additionally, some systems can be expensive and may not be accessible to everyone. Connectivity options may also be limited, with some systems relying on wired connections or proprietary protocols that do not easily integrate with other devices. These limitations can make it difficult for users to fully benefit from the features of the smart chair system.

3.2 Proposed system

Based on the drawbacks of existing systems for posture correction in smart chairs, our proposed system aims to address these limitations by utilizing IoT technology and advanced sensors. Our smart chair system will consist of a Raspberry Pi microcontroller connected to various sensors, including a flex sensor and DHT11 sensor, which will provide real-time feedback on the user's posture and environment. The data collected by the sensors will be analyzed and processed by the Raspberry Pi, and the results will be communicated to the user through a mobile application. The application will provide customized recommendations for posture correction and allow the user to adjust the smart chair settings accordingly. Additionally, the system will utilize Wi-Fi or Bluetooth connectivity to provide seamless integration with other IoT devices, enabling enhanced automation and monitoring of the user's sitting experience. By addressing the limitations of existing systems, our proposed smart chair system aims to provide a more effective and user-friendly solution for posture correction and improved ergonomic sitting.

3.3 Feasibility Study

A feasibility study is necessary to determine whether the proposed system for a smart chair with IoT is feasible in terms of technical, economic, and operational aspects.

3.3.1 Technical Feasibility

From a technical perspective, the use of Raspberry Pi and sensors such as the flex sensor and DHT11 sensor is feasible since these components are readily available and have been proven to work in other similar projects. Additionally, the use of Wi-Fi or Bluetooth for transmitting data to a mobile application is also feasible since these technologies are widely available and can support the required data transfer speeds.

3.3.2 Operational Feasibility

Operationally, the proposed system may require training for users to correctly operate the smart chair and mobile application. Additionally, the reliability and security of the system would need to be thoroughly tested and maintained to ensure the safety and privacy of users.

3.3.3 Economic Feasibility

Economically, the proposed system may require a significant initial investment in terms of hardware and software development costs, but could potentially lead to cost savings in the long term by improving the health and productivity of users. Additionally, the ability to remotely monitor and adjust settings on the smart chair could potentially reduce maintenance costs and increase the lifespan of human beings.

CHAPTER 4

SYSTEM SPECIFICATIONS

4.1 Hardware Requirements:

Table 4.1 Hardware Requirements

Sl. No	Components	Specification	Justification	No. of Units	Cost per Unit	Total Cost
1	Raspberry Pi	Version 4B, 1.4 GHz processor, gigabit Ethernet, 2.4 / 5 GHz dual band 802.11ac Wi-Fi, USB boot and network boot	Huge processing power in a compact board. It supports Python	1	5428	5428
2	Flex Sensor	Flex length - 5.6cm, Total length - 2.2", Life Cycle - >million, Height - 0.43mm, Flat Resistant - 10kΩ, Power rating - 0.5W	A flex sensor in the chair cushion could detect the posture or position of the user by measuring the degree of bend in the sensor	3	325	975
3	DHT11 Humidity and Temperature Sensor Module	Measures temperature from 0°C to 50°C with ±2°C accuracy and humidity from 20% to 90% with ±5% accuracy, operating at 3V to 5.5V DC	Measure temperature and humidity levels to adjust settings for temperature regulation, humidity control	1	275	275

4	16 Bit I2C 4 Channel ADS1115 Module	16 Bit I2C 4 Channel ADS1115 Module: 4 input channels (single-ended/2 differential), 4I2C address options, 2.0-5.5 V operating range,16-bit resolution	Can convert flex sensor's analog signals to 16-bit digital signals, allowing easier analysis of signals to determine user posture and adjust smart chair settings.	1	414	414
5	Jumper wires	Connector type - Male to Male, length of wire - 20cms, weight - 50gm	Provide a flexible and easy way to connect various components in thesmart chair project	1	49	49
6	Office Chair	Designed for use at a desk in an office.	To implement smart chair	1	2000	2000
7	Chair Cover	Cover material - Polyester, closure-Elasticated drawcord	Protect the chair from wear and tearand to maintain hygiene.	1	260	260

4.2 Software Requirements:

Table 4.2 Software Requirements

Sl. No	Tools	Specification	Justification	Open Source / Freeware/ Licensed
1	Operating system Raspbian OS	System: 64-bit · Kernel version: 5.15 · Debian version: 11	Debian Linux- based operating system engineered for use on Pi boards.	Open source
2	Python 3 Compiler	Python 3	Easy to read syntax and fewer lines of code	Open source
3	Programming language: Python	Python	Easy to read syntax and fewer lines of code	Open source
4	MySQL	SQL	To store the data in the database	Open source
5	Android studio	Java	To develop mobile application	Open source
6	Navicat lite for MySQL	SQL	It is a single application that allows us to connect to MySQL and MariaDB databases simultaneously.	Open Source

4.3 Functional Requirements

a) **Posture detection:** The smart chair should be able to detect the user's posture using the flex sensor and other sensors, and analyze the data to determine if the user is sitting in a good or bad posture.

b) **Real-time feedback:** The smart chair should provide real-time feedback to the user about their posture, using visual or auditory cues to prompt the user to adjust their posture.

c) **Posture correction:** The smart chair should be able to adjust its settings, such as seat height, backrest angle, and lumbar support, to correct the user's posture and promote good sitting habits.

d) **Temperature and humidity regulation:** The smart chair should be able to measure the temperature and humidity around the chair using the DHT11 sensor and adjust its settings to maintain a comfortable environment for the user.

e) **User profile and customization:** The smart chair should allow users to create a profile with their preferences for sensor settings, and customize the chair's settings accordingly.

f) **Data tracking and analysis:** The smart chair should be able to collect data on the user's posture, sitting habits, and provide insights and recommendations for improving posture and overall health.

g) **Remote control:** The smart chair will be controlled remotely via a mobile app, allowing users to adjust settings and receive feedback even when not seated in the chair.

h) **Alert system:** The smart chair should have an alert system to notify the user if they are in a bad posture for too long, reminding them to adjust their position.

i) **Mobile app integration:** The smart chair should have a mobile app that allows the user to monitor their posture, customize settings, and receive alerts and notifications on their smartphone.

4.4 Non-Functional requirements

a) **Usability:** The smart chair should be user-friendly and easy to operate, with clear instructions and feedback provided to the user.

b) **Reliability:** The system should be reliable and able to operate without failure or downtime for extended periods of time.

c) **Security:** The smart chair should have appropriate security measures in place to protect user data and prevent unauthorized access.

d) **Scalability:** The system should be designed to accommodate future enhancements and upgrades to the hardware and software components.

e) **Performance:** The smart chair should be able to process data from the sensors in real-time, and respond quickly to user input.

f) **Maintainability:** The system should be easy to maintain, with clear documentation and instructions for troubleshooting and repair.

g) **Portability:** The smart chair should be portable and able to be moved easily, without compromising its functionality or accuracy.

h) **Compatibility:** The system should be compatible with a wide range of devices and platforms, to ensure maximum usability and accessibility.

CHAPTER 5

PROJECT DESCRIPTION

5.1 Problem Definition

The problem we aim to address with our smart chair project is the issue of poor posture and its negative impact on health and well-being. Poor posture, especially during prolonged sitting, can cause various musculoskeletal disorders such as back pain, neck pain, and shoulder pain. It can also affect breathing, digestion, and overall comfort. Traditional solutions such as ergonomic chairs can be expensive and not easily accessible to everyone. Our project aims to provide an affordable and accessible solution by utilizing technology to create a smart chair that can monitor posture and make necessary adjustments to ensure a more comfortable and ergonomic sitting experience.

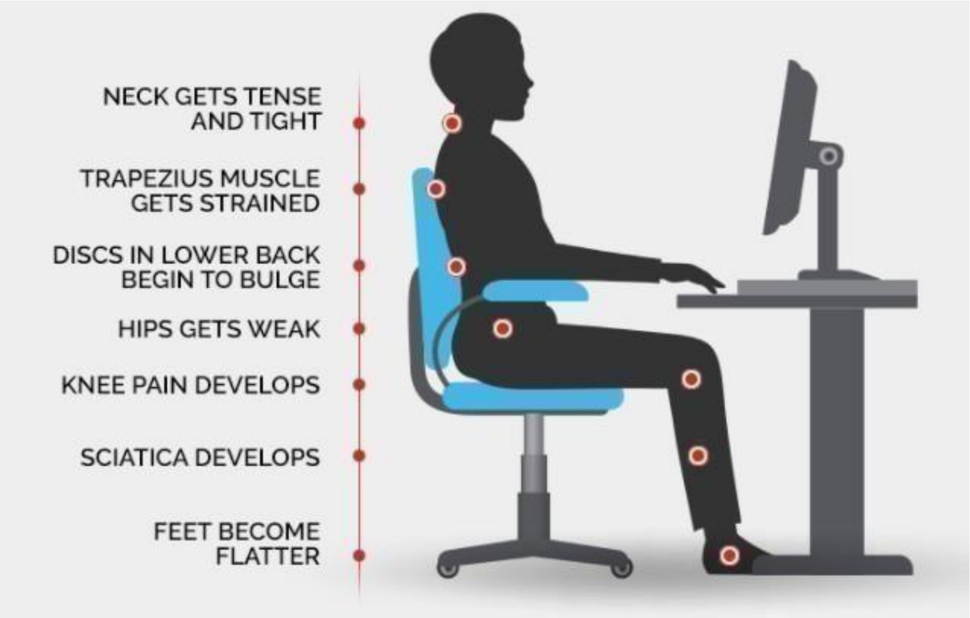

NECK GETS TENSE AND TIGHT
TRAPEZIUS MUSCLE GETS STRAINED
DISCS IN LOWER BACK BEGIN TO BULGE
HIPS GETS WEAK
KNEE PAIN DEVELOPS
SCIATICA DEVELOPS
FEET BECOME FLATTER

Fig 5.1 The sitting posture

Fig 5.1 depicts the problems caused by poor posture, such as muscle tension, sciatica, knee pain, swayback posture, and so on.

Correct seating position is essential for maintaining good posture and preventing long-term health problems such as back pain, neck pain, and spinal issues. A correct seating position involves sitting upright with your back against the chair, your feet flat on the ground, and your knees at a right angle. Your shoulders should be relaxed, and your arms should be comfortably resting on the armrests or on the desk in front of you. Additionally, the chair should provide adequate lumbar support to maintain the natural curve of the spine. Poor seating positions such as slouching, crossing legs, or leaning to one side can cause strain on the muscles and joints, leading to discomfort and pain. Therefore, it is crucial to maintain a correct seating position to prevent long-term health problems and promote comfort and productivity in daily life.

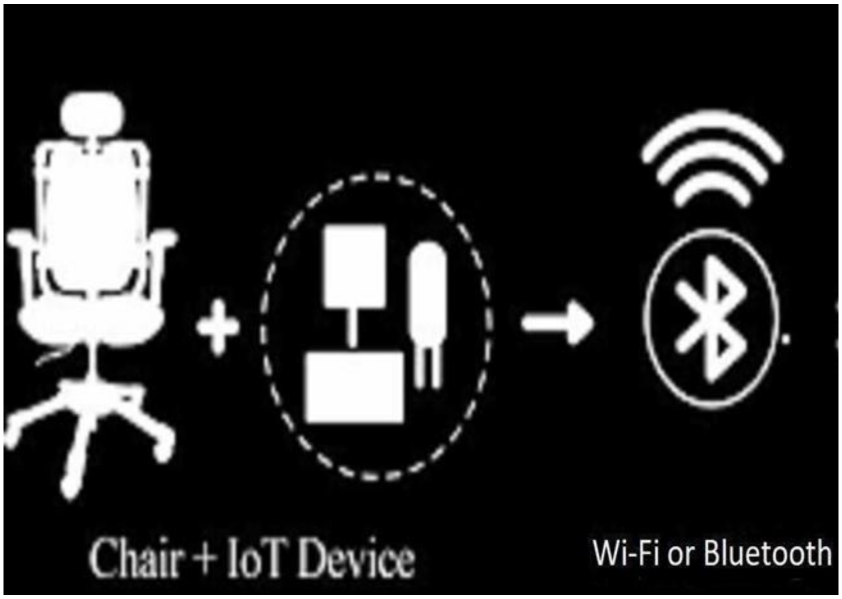


Fig 5.2 IoT Architecture of Smart Chair

Fig 5.2 the smart chair with IoT technology is designed to improve posture and overall health by monitoring the user's seating position and providing feedback through a mobile application. This system consists of various sensors, including a flex sensor and DHT11 sensor, connected to a Raspberry Pi microcontroller. The flex sensor detects the user's posture and bending angle, while the DHT11 sensor measures temperature and humidity levels. The Raspberry Pi processes this data and sends it to the mobile application through Wi-Fi or Bluetooth. The application provides feedback to the user, alerting them when they need to adjust their posture and displaying real-time data on their seating position and environment. This system has the potential to improve user health by promoting correct posture, reducing the risk of musculoskeletal disorders, and providing personalized feedback to encourage healthy habits.

5.2 Overview of the project

Smart chairs are required because prolonged sitting can have negative effects on the body, such as back pain, poor circulation, and decreased productivity. Maintaining proper sitting posture can help to alleviate these issues and improve overall health and well-being. However, maintaining good posture on a consistent basis can be difficult, especially when working long hours in an office setting. Smart chairs can provide real-time feedback to the user on their posture, allowing them to adjust their sitting position as needed. Smart chairs can provide more accurate feedback than traditional chairs by utilizing sensors and data analysis. Smart chairs can also be tailored to the needs of individual users, providing personalized feedback and suggestions for improvement. This can help users develop healthy sitting habits over time, which can lead to better health and well-being.

In a nutshell, smart chair is required to assist users in maintaining proper posture while sitting, improving overall health and wellbeing, and increasing productivity in office settings.

The basic working of the multiple disease prediction system is described through the following steps:

- **Objective:** The objective of the project is to create an IoT based smart chair that can measure the posture of the user and provide feedback to help them maintain a good posture while sitting for extended periods of time.
- **Components Used:** The following components are used in the project:
  - **Raspberry Pi:** Acts as the central processing unit and collects data fromthe sensors.
  - **Flex Sensors:** Measure the bending of the chair back and seat todetermine the posture of the user.
  - **DHT11 Temperature and Humidity Sensor:** Measures the temperature and humidity of the user sitting on the chair.
- **Working Principle:** The smart chair lets the user sit on it, and the flex sensors on the back and below the chair measure the posture of the user. The DHT11 temperature and humidity sensor on the cushion measures the temperature and humidity of the user. The data collected by these sensors is sent to the server, which stores the data. The mobile app displays the data collected by the sensors and provides feedback to the user, such as alerting them to take a break after 30 minutes of sitting. The app also displays the sitting time, good posture, bad posture, graphs, and allows the user to change sensor settings.
- **Hardware Setup:** The flex sensors are attached to the back and below the chair using adhesive. The DHT11 sensor is placed on the cushion of the chair. The Raspberry Pi is connected to the sensors and sends data to the server using Wi-Fi or Ethernet.

- **Data Visualization:** The mobile app displays the data collected by the sensors in a user-friendly way. The app provides graphs and charts that show the user's sitting time, good posture, bad posture, and other relevant data. The app also provides notifications and alerts to the user when they need to take a break or adjust their posture.
- **Importance of Good Posture:** Sitting for extended periods of time can have negative effects on the body, such as back pain, poor circulation, and reduced productivity. Maintaining a good posture can help to alleviate these problems and improve overall health and wellbeing.
- **Benefits of IoT Based Smart Chairs:** IoT based smart chairs can provide personalized feedback to help users maintain good posture and reduce the negative effects of sitting for extended periods of time. By using sensors and data analysis, these chairs can provide more accurate feedback than traditional chairs.
- **Conclusion:** The IoT based smart chair project is a useful and innovative solution to the problem of maintaining good posture while sitting for extended periods of time. By using sensors, data analysis, and a mobile app, this project provides personalized feedback to the user and can help to improve overall health and wellbeing.
- **Future Enhancements:** The project can be enhanced by adding machine learning algorithms to learn the user's sitting habits and provide more accurate feedback on their posture. Additional sensors, such as an accelerometer or a gyroscope, can be added to provide more comprehensive posture analysis. Voice recognition technology can be added to enable users to interact with the smart chair through voice commands. Cloud computing can be used to store and analyze the data collected by the smart chair.

5.3 System Architecture

Smart chairs powered by IoT are intended to improve user comfort and reduce health risks by detecting and monitoring their posture, temperature, and humidity levels. We will use a Raspberry Pi in this project to collect data from a flex sensor that detects the user's posture and a DHT11 sensor that measures temperature and humidity levels. The collected data will be processed and transmitted to a mobile application, which will provide real-time feedback and notifications to the user.

The system architecture of the IoT-based smart chair powered by a Raspberry Pi is made up of three major components: hardware, software, and the architecture itself.

5.3.1 Hardware Components:

Raspberry Pi: We will be using the Raspberry Pi as the main controller to collect, process, and transmit data from the flex sensor and DHT11 sensor to the cloud server. Fig 5.3 represents the pin diagram of the Raspberry Pi which is a series of small single-board computer (SBC) developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom. The Raspberry Pi project originally leaned towards the promotion of teaching basic computer science in schools and in developing countries. The original model became more popular than anticipated, selling outside its target market for uses such as robotics. It is widely used in many areas, such as for weather monitoring, because of its low-cost modularity, and open design. It is typically used by computer and electronic hobbyists, due to its adoption of HDMI and USB devices.

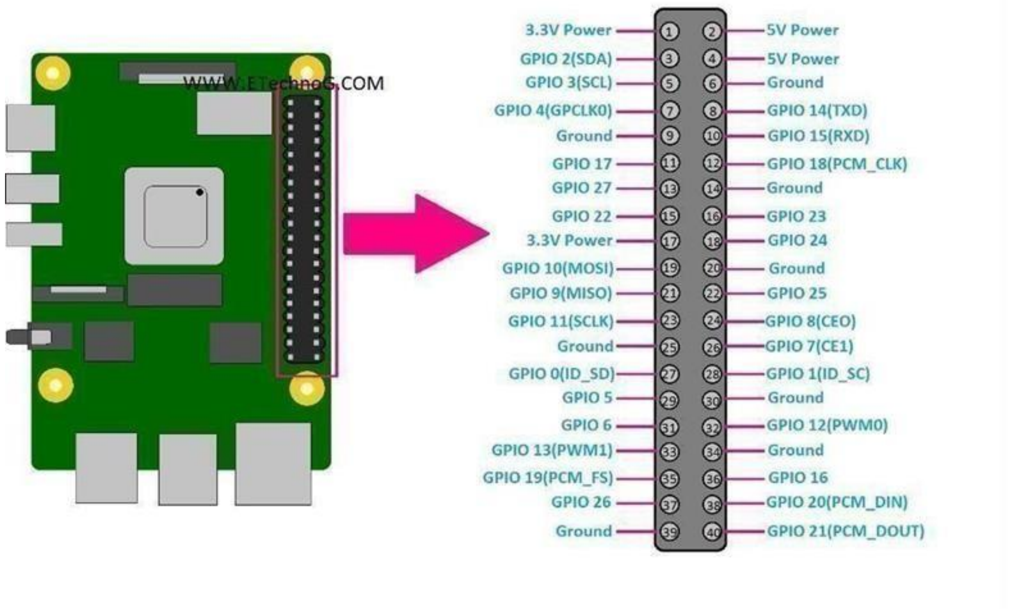

WWW.ETechnoG.COM
3.3V Power 1 2 5V Power
GPIO 2(SDA) 3 4 5V Power
GPIO 3(SCL) 5 6 Ground
GPIO 4(GPCLK0) 7 8 GPIO 14(TXD)
Ground 9 10 GPIO 15(RXD)
GPIO 17 11 12 GPIO 18(PCM_CLK)
GPIO 27 13 14 Ground
GPIO 22 15 16 GPIO 23
3.3V Power 17 18 GPIO 24
GPIO 10(MOSI) 19 20 Ground
GPIO 9(MISO) 21 22 GPIO 25
GPIO 11(SCLK) 23 24 GPIO 8(CEO)
Ground 25 26 GPIO 7(CE1)
GPIO 0(ID_SD) 27 28 GPIO 1(ID_SC)
GPIO 5 29 30 Ground
GPIO 6 31 32 GPIO 12(PWM0)
GPIO 13(PWM1) 33 34 Ground
GPIO 19(PCM_FS) 35 36 GPIO 16
GPIO 26 37 38 GPIO 20(PCM_DIN)
Ground 39 40 GPIO 21(PCM_DOUT)

Fig 5.3 Raspberry Pi

DHT11 Humidity and Temperature Sensor Module: Fig 5.4 depicts the DHT11, which is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin no analog pins needed. It’s fairly simple to use but requires careful timing to grab data. Also, the DHT11 sensor has better humidity measuring range, from 0 to 100% with 2-5% accuracy, while the DHT11 humidity range is from 20 to 80% with 5% accuracy.

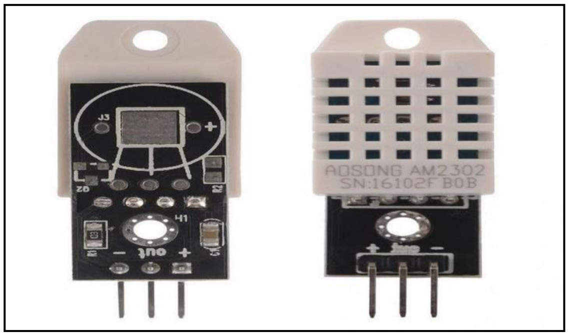


Fig 5.4 DHT11 Humidity and Temperature Module

Flex Sensor: Fig 5.5 depicts the Flex sensors, which are highly versatile sensors that are used to detect bending or flexing movements in various applications. Flex sensors are made of a conductive material, such as carbon or copper, which is coated with a polymer layer. The resistance of a flex sensor varies with the degree of flexing or bending, and the resistance range of a typical flex sensor is between 10KΩ and 50KΩ. Flex sensors are highly sensitive to small changes in bending or flexing movements, making them ideal for detecting subtle movements in robotics or posture correction systems. They come in various sizes and shapes, ranging from a few millimeters to several centimeters in length, with a typical thickness of between 0.2mm and 0.3mm. Flex sensors are designed to withstand repeated bending or flexing movements and can withstand a wide range of temperatures and environmental conditions. Additionally, flex sensors consume very little power, making them ideal for battery-powered devices. They output an analog signal, which can be read by an analog-to-digital converter (ADC) or a microcontroller, and this signal can be used to measure the degree of bending or flexing of the sensor.

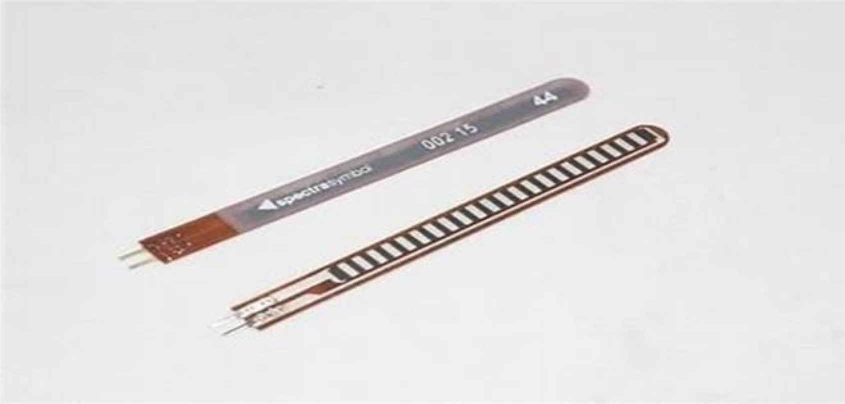


Fig 5.5 Flex Sensor

Power supply: A power supply is required to power the Raspberry Pi and other components.

ADS1115: Fig 5.6 depicts the ADS1115 module. The ADS1115 is a 16-bit Analog-to-Digital Converter (ADC) module that can be used to convert analog signals to digital signals for processing by a microcontroller or computer. It has four input channels that can be used to measure analog voltages with high accuracy and precision. The ADS1115 is particularly useful in projects where high-precision analog-to-digital conversion is required, such as in sensor applications. It can be interfaced with microcontrollers like the Raspberry Pi using I2C communication protocol. This module can be powered using a 3.3V or 5V power supply and can operate at up to 860 samples per second with continuous reading mode.

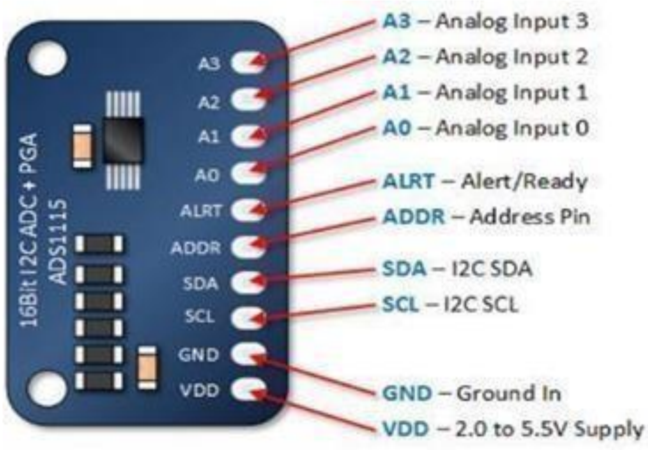


Fig 5.6 ADS1115 module

Jumper Wires: Fig 5.7 depicts the jumper wires which are used to connect the various components in a circuit. They are typically made of insulated wire with metal pins or connectors on each end, allowing them to be plugged into breadboards or other components. Jumper wires come in different lengths, colors, and types to suit different project requirements. When using jumper wires, it's important to ensure that the connections are secure and properly aligned to avoid any loose or faulty connections. It's also important to use jumper wires with appropriate gauges and lengths to minimize any interference or voltage drops in the circuit.

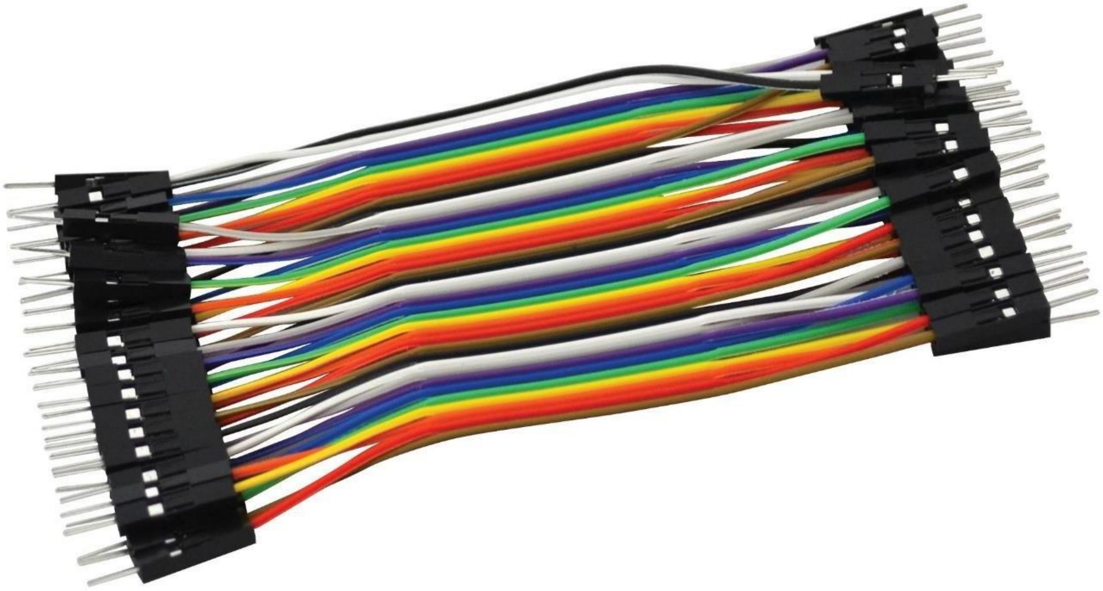


Fig 5.7 Jumper wires

5.3.2 Software Components:

The software components required for this project include the following:

Python programming language: Python is an easy-to-learn, powerful programming language that is widely used in IoT-based projects.

Android Studio: Android Studio is the integrated development environment (IDE) used to develop mobile applications.

MySQL: MySQL is a widely-used open-source relational database management system that provides a powerful set of features for managing databases. It supports a variety of data types and can handle large volumes of data efficiently, making it a popular choice for high-traffic web applications. MySQL is scalable and can handle multiple users and concurrent connections. It can be used with a variety of programming languages and provides a range of tools and utilities for managing and administering databases, including graphical user interfaces and command-line tools. MySQL is known for its versatility, performance, and ease of use, making it a valuable tool for developers and businesses alike.

Raspberry pi imager: Raspberry Pi Imager is a software tool used to install operating systems onto an SD card for use with a Raspberry Pi. The tool is free and available for download on the Raspberry Pi website. Raspberry Pi Imager simplifies the process of installing an operating system onto a Raspberry Pi by providing a user-friendly interface and automatically formatting and writing the image to the SD card. The tool also supports a range of operating systems, including the Raspberry Pi OS, Ubuntu, and various third-party distributions. Overall, Raspberry Pi Imager is a useful tool for anyone looking to set up a Raspberry Pi with an operating system, providing a streamlined and hassle-free installation process.

VNC Viewer: VNC Viewer is a widely used remote desktop client that allows users to access and control remote desktops over a network connection. It is particularly popular in IT support and remote working scenarios, as it allows users to access remote systems and provide technical support without needing to be physically present. VNC Viewer provides a range of features, including support for multiple operating systems, encryption and authentication for secure connections, and the ability to transfer files between remote and local systems. It also provides a range of customization options, allowing users to adjust display settings, keyboard and mouse configurations, and other preferences to suit their needs. Overall, VNC Viewer is a powerful and versatile remote desktop client that is used by individuals and businesses around the world to access and control remote systems.

5.3.3 Architecture:

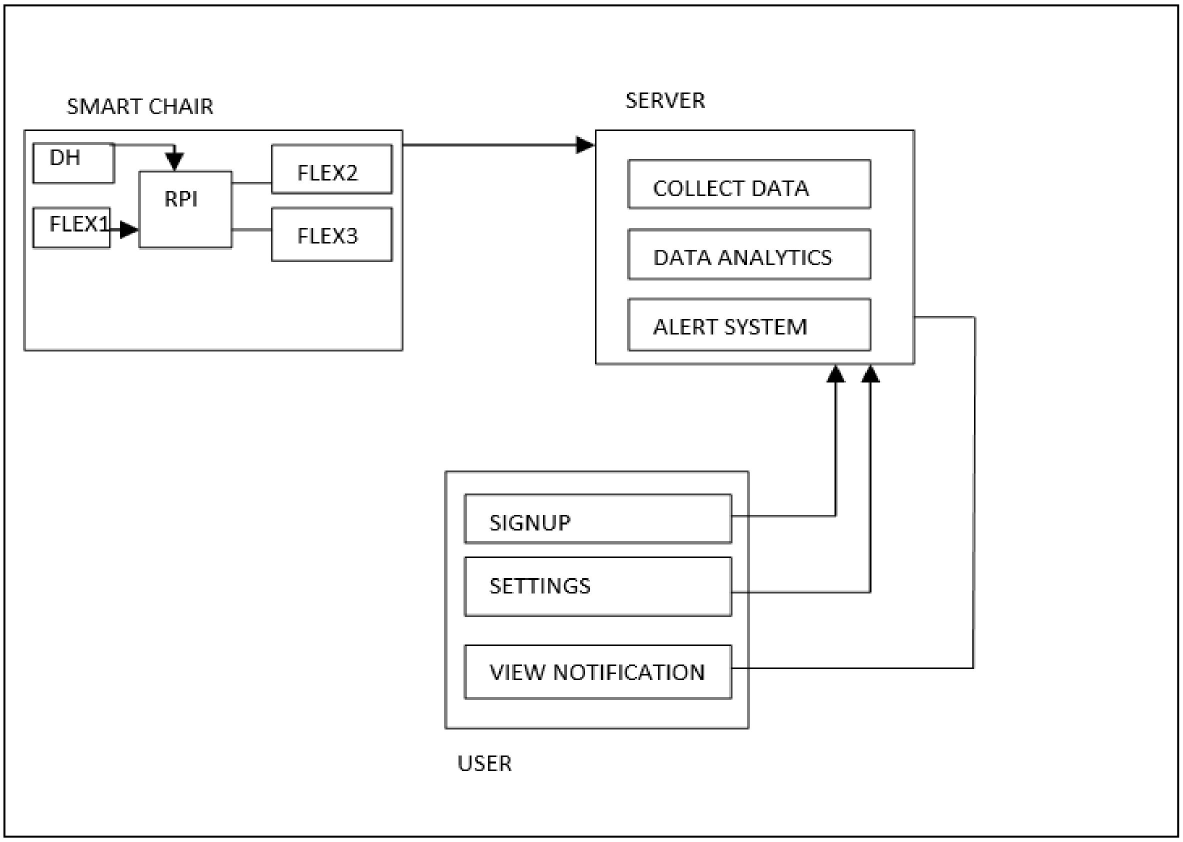


Fig 5.8 System Architecture of smart chair

Fig 5.8 depicts the smart chair's system architecture which is made up of three major components: the smart chair itself, a MySQL server, and a mobile application. The smart chair is outfitted with a number of sensors, including flex sensors and a DHT11 temperature and humidity module, which collect information about the user's posture and temperature conditions. This information is sent to the MySQL server, which stores and manages it. The mobile application allows users to view their posture data as well as other information such as sitting time and graphs through an interface. To retrieve and display data, the mobile application communicates with the MySQL server. Furthermore, after 30 minutes of sitting, the mobile application sends alerts to the user, reminding them to take a break. This system architecture collects and manages data from the smart chair while also providing a user-friendly interface for users to view and interact with their posture data.

The architecture of the system is divided into three main components: the Smart Chair, SQL Server, and Mobile Application.

a) Smart Chair:

The Smart Chair consists of a Raspberry Pi, flex sensors, a DHT11 temperature and humidity module. The flex sensor detects the user's posture and sends the data to the Raspberry Pi, which processes the data and transmits it to the server. The DHT11 sensor measures the temperature and humidity levels, and the Raspberry Pi also processes and sends this data to the server.

b) MySQL Server:

MySQL is a popular data storage and management system for web applications and other software projects. MySQL's SQL support allows it to handle a wide range of data types and efficiently store and retrieve large amounts of data. MySQL has a flexible and scalable data model that allows developers to create tables and relationships between tables that can be queried using SQL. Furthermore, MySQL provides a variety of data management features, such as the ability to insert, update, and delete records, as well as the ability to run complex queries to retrieve data. MySQL also includes security features like user authentication and access control to protect data from unauthorized access. Overall, MySQL is a powerful and dependable data storage and management tool that can be used in a variety of applications.

c) Mobile Application:

The Mobile Application receives data from the server and displays it to the user. The application is developed using Android Studio. The mobile application provides real-time information about the user's posture, temperature, and humidity levels. It also sends notifications to the user if they have been sitting for too long, or if their posture is incorrect for an extended period.

5.4 Module Description

Module 1: Measuring the force readings from Flex sensors

Fig 5.9 depicts the flex sensors, in the smart chair are used to measure the user's posture while sitting. When the user sits in the chair, the flex sensors on the back and beneath the seat bend and measure the amount of flexing. This information is then sent to the Raspberry Pi, which processes it and determines the user's posture. Based on this data, the smart chair can provide the user with alerts and feedback to encourage good sitting posture and reduce the risk of discomfort or injury caused by poor posture. The flex sensors are an important component of the smart chair system, providing valuable data on the user's posture and assisting in the improvement of theirsitting habits.

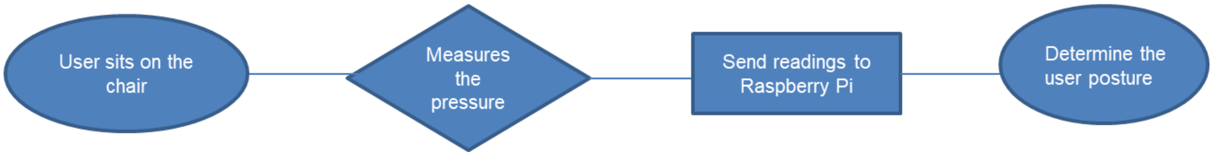


Fig 5.9 Module 1 flowchart

Module 2: Measuring temperature and humidity

Fig 5.10 shows the DHT11 is a digital temperature and humidity sensor that is commonly used in various IoT applications, including smart chairs. In smart chairs, DHT11 sensors are used to measure the temperature and humidity levels of the user. The sensor detects changes in temperature and humidity levels and transmits this data to a microcontroller, such as a Raspberry Pi, which processes the data and provides feedback to the user.

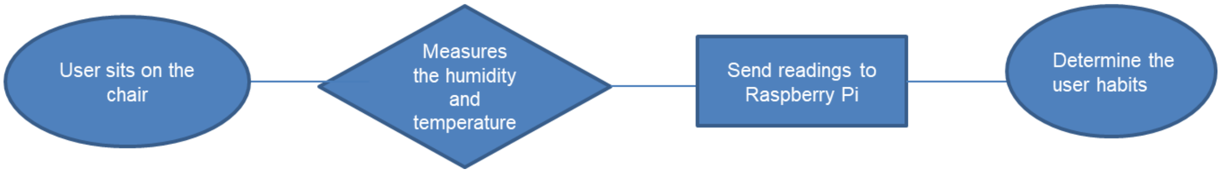

User sits on the chair
Measures the humidity and temperature
Send readings to Raspberry Pi
Determine the user habits

Fig 5.10 Module 2 flowchart

Module 3: Issuing appropriate notifications

Notifications via mobile applications are a common way for smart chairs to alert users about incorrect posture. When a flex sensor attached to the chair detects that the user's posture is incorrect, it sends the data to a microcontroller, which sends a notification to the user's mobile device. If the user has been sitting for more than 30 minutes, a notification is sent to them. The user can then adjust their posture and receive real-time feedback on how well they are doing. Integrating a mobile app with a smart chair improves functionality and usability. Users can change their notification preferences and track their progress over time. This feature is especially beneficial for people who spend long periods of time sitting in chairs, such as office workers or students, and can help prevent or alleviate health problems associated with poor posture, such as back pain or muscle strain. Notifications sent via mobile app in smart chairs are an effective way to alert users to incorrect posture and promote healthy sitting habits. Smart chairs can help improve the user's overall well-being and comfort by providing real-time feedback and customizable settings.

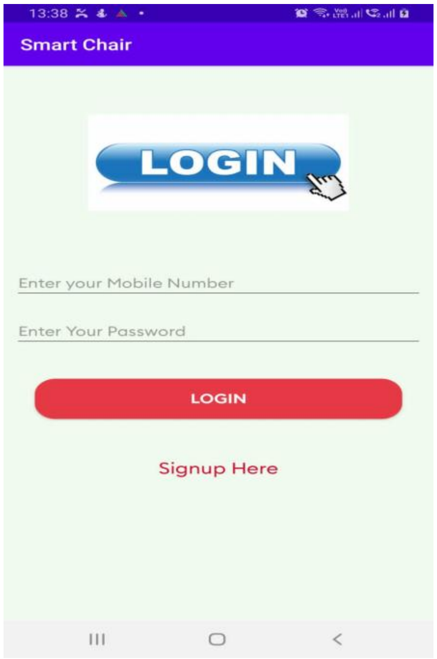


Fig 5.11 Login Page

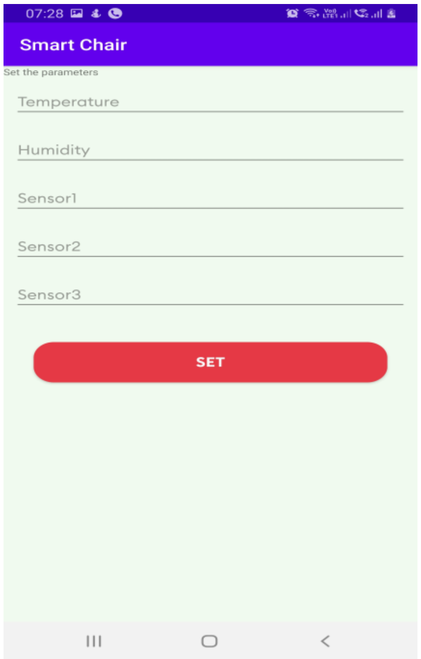


Fig 5.12 User settings Page

Fig 5.11 depicts the login page of the smart chair application where the user can login and see their data, Fig 5.12 depicts the user settings page where the user can change the default settings to its own customizable sensor values.

5.5 Data Flow diagram

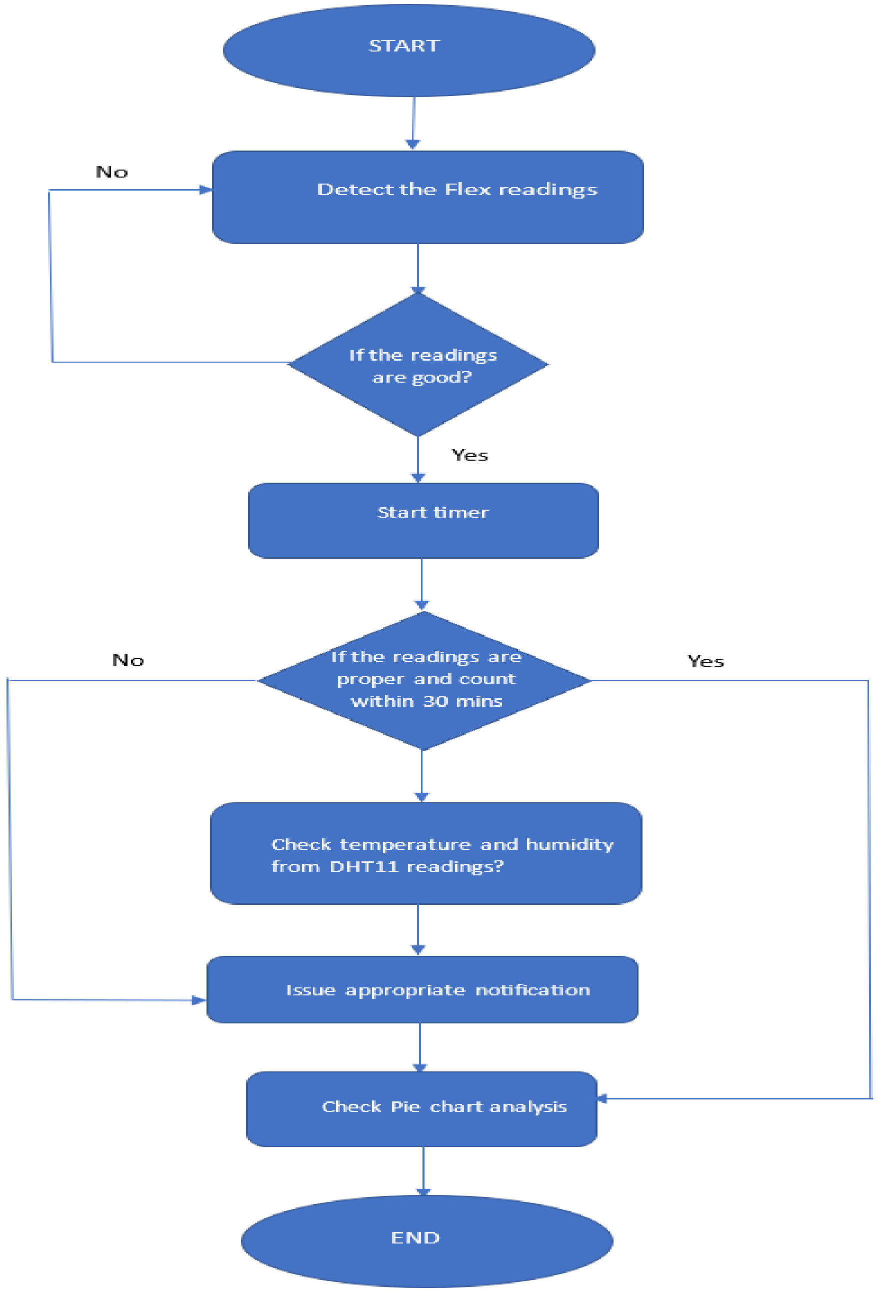

START
No
Detect the Flex readings
If the readings are good?
Yes
Start timer
No
If the readings are proper and count within 30 mins
Yes
Check temperature and humidity from DHT11 readings?
Issue appropriate notification
Check Pie chart analysis
END

Fig 5.13 Data Flow Diagram

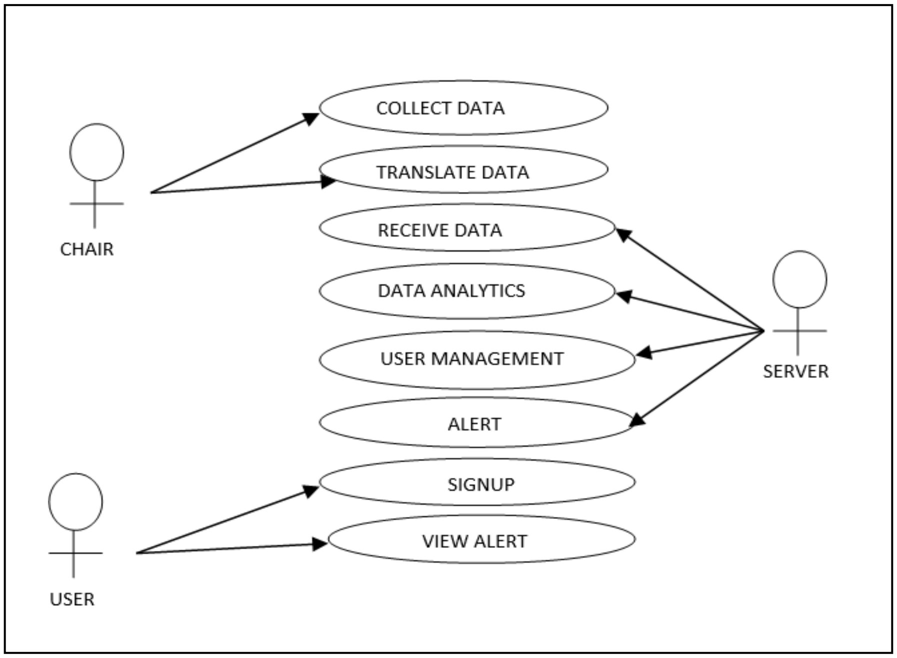
Fig 5.13 In the IoT-based smart chair project, the flex sensor and DHT11 sensor are connected to a Raspberry Pi. The flex sensor detects the user's posture and sends the data to the Raspberry Pi, which processes the data and determines whether the user is sitting in the correct posture or not. The DHT11 sensor detects the temperature and humidity in the user's surroundings, and sends this data to the Raspberry Pi as well. The Raspberry Pi then uses this data to determine if the user is sitting in a comfortable and safe environment. The Raspberry Pi is also connected

to a mobile application through Wi-Fi. When the Raspberry Pi detects that the user is sitting in an incorrect posture or in an uncomfortable environment, it sends a notification to the mobile application. The notification provides information about the issue and suggests corrective actions to the user. The mobile application also displays real-time data from the flex sensor and DHT11 sensor, so the user can monitor their posture and surroundings.

Overall, the data flow diagram of the smart chair project consists of three main components: the sensors, the Raspberry Pi, and the mobile application. The sensors collect data on posture and environmental conditions, which is processed by the Raspberry Pi. The Raspberry Pi then sends notifications to the mobile application, which provides real-time feedback and suggestions to the user.

5.6 Use case diagram

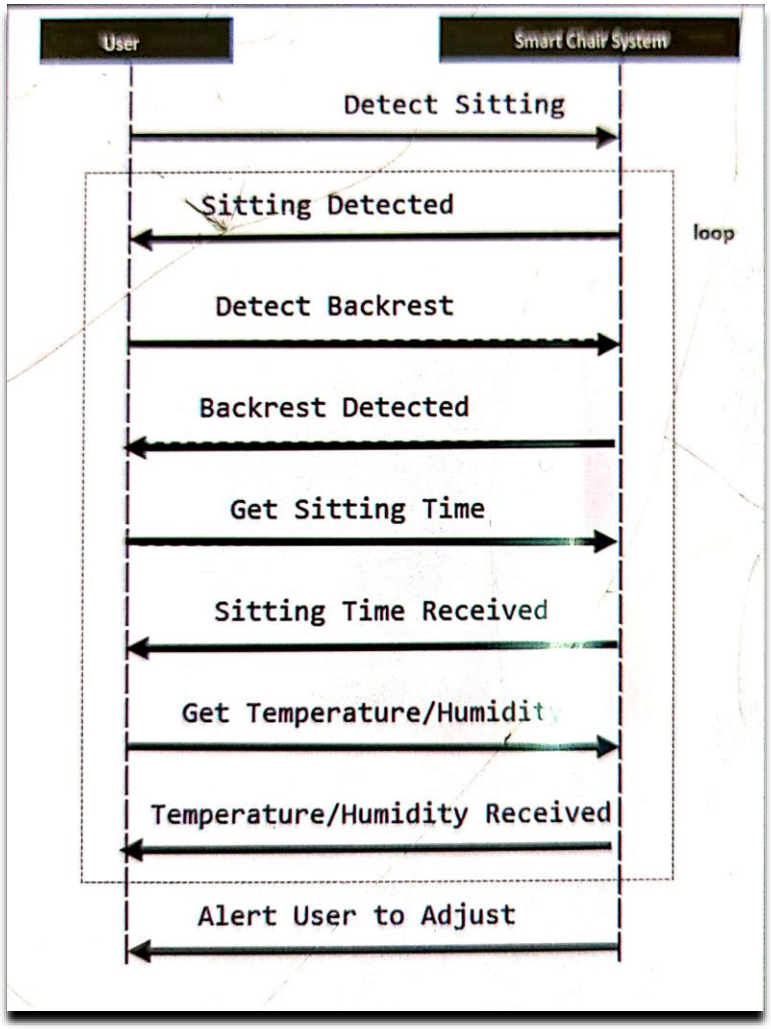
Fig 5.14 depicts the use case diagram. The chair collects the data, which is the three flex sensor readings which is attached to the backrest of the chair and on the seat. It also collects the temperature and humidity of the user. It translates or interprets the data. Server receives the data and stores it. Data analysis is performed on the data, which is mostly visualization of the user’s sitting habits. User data management is done and send an alert to the user, if the user is sitting for more than 30 minutes. The user gets the alert on the smart chair application. User needs to sign up and view the alert and related information.



5.14 Use Case Diagram

5.7 Sequence diagram

Fig 5.15 depicts the sequence diagram. User and smart chair system interaction can be seen. The smart chair project involves several components working together to provide posture correction. The process starts with the user sitting on the chair, and the flex sensor detecting their posture. The flex sensor then sends the posture data to the Raspberry Pi for processing. If the user is sitting in an incorrect posture, the Raspberry Pi sends a notification to the mobile application, which displays it to the user. The user can then adjust their position accordingly. The DHT11 sensor detects the temperature and humidity in the environment and sends the data to the Raspberry Pi. The Raspberry Pi processes the data and sends a notification to the mobile application if the user is sitting in an improper posture and for a long time. The user can then adjust the temperature, humidity, flex sensors as needed.



5.15 Sequence Diagram

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Introduction

The implementation of the smart chair project involves the integration of various hardware and software components to create a functional system. The hardware components include a Raspberry Pi, flex sensor, DHT11 sensor, ADS1115 module. The software components include the Raspbian operating system, Python programming language, and various libraries for sensor data processing and communication. The flex sensors are connected to the ADS1115 module, which converts the analog signals from the sensors into digital signals for processing by the Raspberry Pi. The Raspberry Pi runs a Python program that reads and analyses the sensor data, determines the user's posture and environmental conditions, and adjusts the smart chair's settings accordingly.

The data storage and retrieval system are done using a MySQL server. The sensors attached to the chair will collect data on the user's posture and position, which will be converted into digital signals and sent to the Raspberry Pi. The Android application will retrieve the data from the server and display it to the user in an easy-to-understand format. The application will also send notifications to the user when their posture is incorrect, alerting them to adjust their position to maintain good posture. By using a centralized MySQL database, we can ensure the data is securely stored and easily accessible from the Android application, making it convenient for the user to monitor their posture and maintain a healthy sitting position.

6.1.1 Hardware implementation

Fig 6.1 shows the hardware implementation of the smart chair and its setup. The hardware part of our smart chair project involves the use of a Raspberry Pi to collect data from the sensors, including the flex sensor and DHT11 sensor, and send it to a MySQL server. The Raspberry Pi is connected to the sensors via jumper wires, and the data is collected using the ADS1115 module to convert analog signals to digital signals with high resolution. The Raspberry Pi is powered by a DC power supply and connected to the internet using Wi-Fi. Once the data is sent to the MySQL server, the Android application can access it and use it to determine the posture of the user. If the posture is incorrect, the alert system is triggered and a notification is sent to the user's mobile phone via the Android application. Overall, the hardware implementation of our project is designed to be easy to assemble and maintain, while providing accurate and reliable data collection and analysis for posture correction.

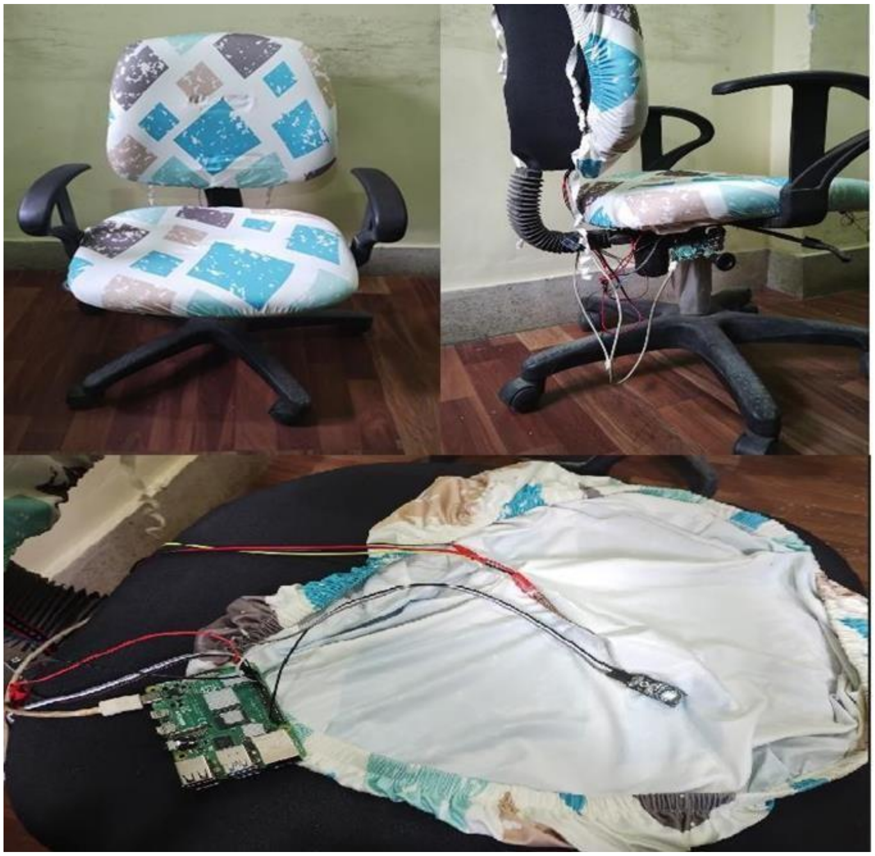
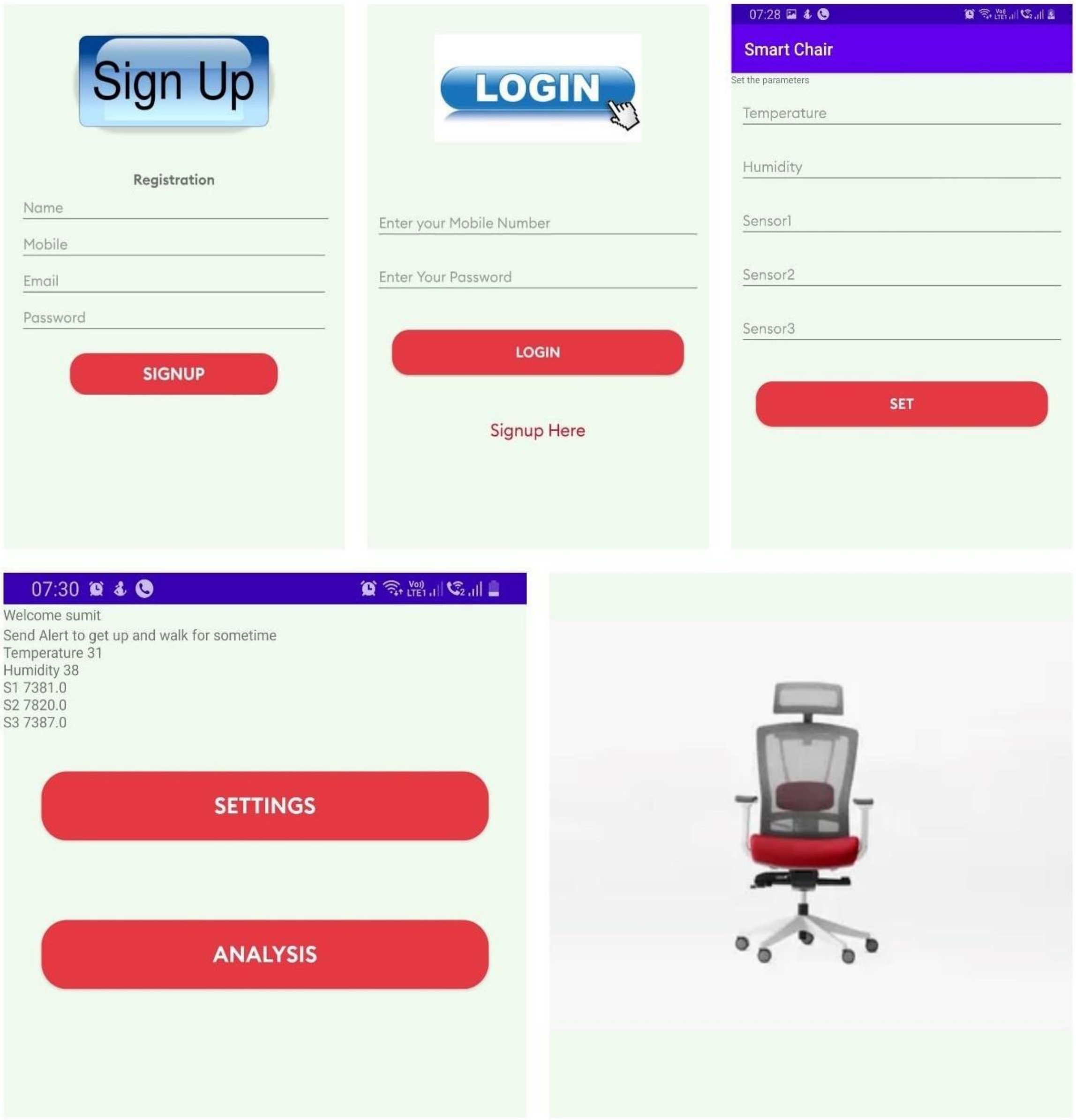


Fig 6.1 Hardware Implementation

6.1.2 Software implementation

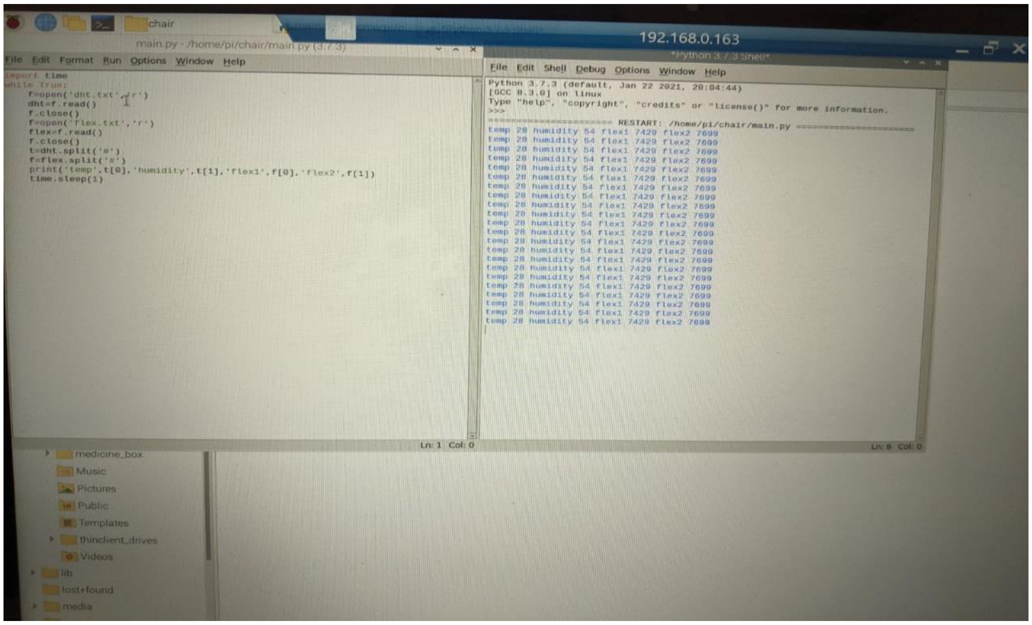
Fig 6.2 shows the software implementation. The software implementation of our project involves using VNC Viewer to remotely connect with the Raspberry Pi and program the sensors and microcontroller using the Thonny IDE, which is a user-friendly Python integrated development environment. We are using Python scripts to read the data from the sensors and store it in the MySQL server. We also developed an Android application using Android Studio, which connects to the MySQL server to retrieve the posture data collected by the Raspberry Pi. The android application is designed with a user- friendly interface that includes a signup and login page. This allows users to create an account, log in, and access the features of the smart chair system. Once logged in, users can view their posture data and receive alerts based on their posture conditions. The application communicates with the MySQL server to retrieve the posture data and display it to the user. The software implementation also includes the integration of user customization features in the Android application. This allows users to adjust the temperature and humidity values according to their preferences, as well as adjust the sensitivity of the flex sensor to their desired level. To achieve this, appropriate user interface elements and functionality have been implemented in the application, allowing users to easily access and modify these settings.



Welcome sumit
Send Alert to get up and walk for sometime
Temperature 31
Humidity 38
S1 7381.0
S2 7820.0
S3 7387.0

Fig 6.2 Software Implementation

6.2 Screenshots

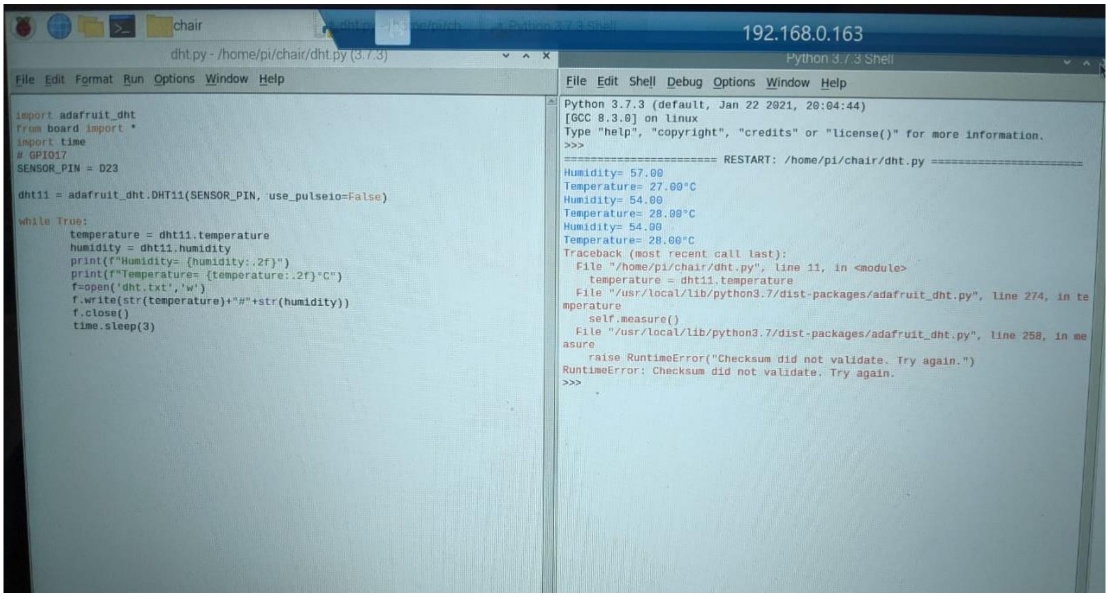


```
import time
while True:
    f=open('dht.txt','r')
    dht=f.read()
    f.close()
    f=open('flex.txt','r')
    flex=f.read()
    f.close()
    t=dht.split('#')
    f=flex.split('#')
    print('temp',t[0],'humidity',t[1],'flex1',f[0],'flex2',f[1])
    time.sleep(1)
```

```
Python 3.7.3 (default, Jan 22 2021, 20:04:44)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===================== RESTART: /home/pi/chair/main.py =====================
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
temp 28 humidity 54 flex1 7429 flex2 7699
```

Fig 6.3 Humidity and temperature checking in raspberry pi

Fig 6.3 depicts the reading in raspberry pi OS, which shows the humidity and temperature readings when the sensor detects changes in humidity and temperature.

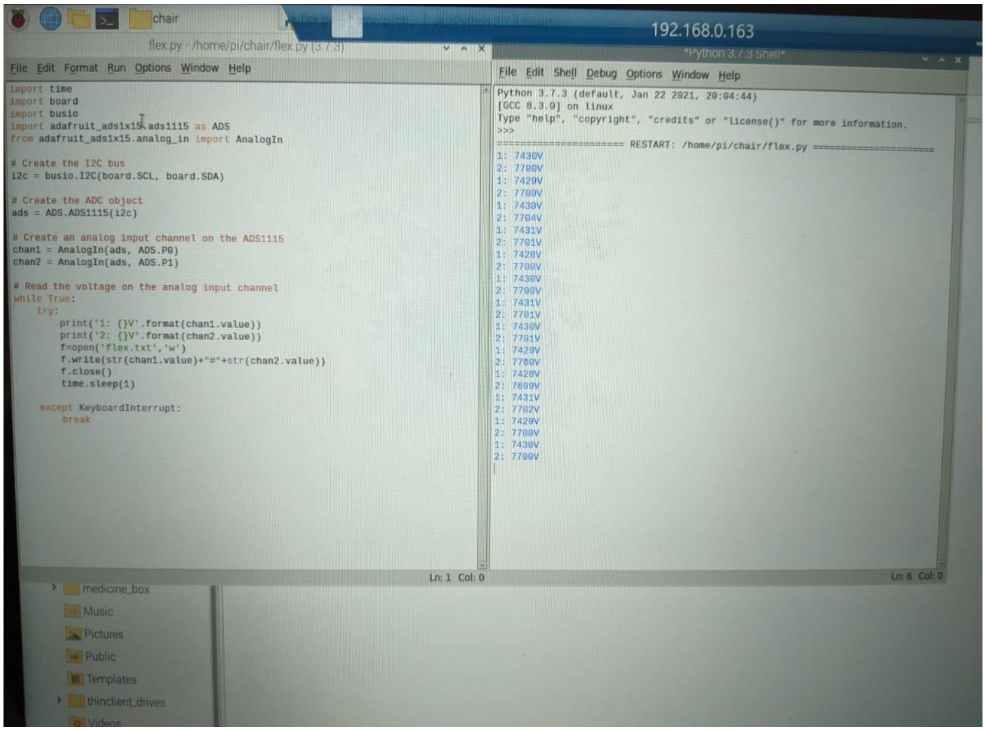

chair
dht.py - /home/pi/chair/dht.py (3.7.3)
File Edit Format Run Options Window Help
192.168.0.163
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help

```
import adafruit_dht
from board import *
import time
# GPIO17
SENSOR_PIN = D23

dht11 = adafruit_dht.DHT11(SENSOR_PIN, use_pulseio=False)

while True:
        temperature = dht11.temperature
        humidity = dht11.humidity
        print(f"Humidity= {humidity:.2f}")
        print(f"Temperature= {temperature:.2f}°C")
        f=open('dht.txt','w')
        f.write(str(temperature)+"#"+str(humidity))
        f.close()
        time.sleep(3)
```

```
Python 3.7.3 (default, Jan 22 2021, 20:04:44)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
====================== RESTART: /home/pi/chair/dht.py ======================
Humidity= 57.00
Temperature= 27.00°C
Humidity= 54.00
Temperature= 28.00°C
Humidity= 54.00
Temperature= 28.00°C
Traceback (most recent call last):
  File "/home/pi/chair/dht.py", line 11, in <module>
    temperature = dht11.temperature
  File "/usr/local/lib/python3.7/dist-packages/adafruit_dht.py", line 274, in te
mperature
    self.measure()
  File "/usr/local/lib/python3.7/dist-packages/adafruit_dht.py", line 258, in me
asure
    raise RuntimeError("Checksum did not validate. Try again.")
RuntimeError: Checksum did not validate. Try again.
>>>
```

Fig 6.4 DHT11 reading integrated

Fig 6.4 depicts the readings from DHT11 Humidity and temperaturemodule, in one screen to get a better understanding of the changes in the readings.



```
import time
import board
import busio
import adafruit_ads1x15.ads1115 as ADS
from adafruit_ads1x15.analog_in import AnalogIn

# Create the I2C bus
i2c = busio.I2C(board.SCL, board.SDA)

# Create the ADC object
ads = ADS.ADS1115(i2c)

# Create an analog input channel on the ADS1115
chan1 = AnalogIn(ads, ADS.P0)
chan2 = AnalogIn(ads, ADS.P1)

# Read the voltage on the analog input channel
while True:
    try:
        print('1: {}V'.format(chan1.value))
        print('2: {}V'.format(chan2.value))
        f=open('flex.txt','w')
        f.write(str(chan1.value)+"#"+str(chan2.value))
        f.close()
        time.sleep(1)

    except KeyboardInterrupt:
        break
```

```
Python 3.7.3 (default, Jan 22 2021, 20:04:44)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===================== RESTART: /home/pi/chair/flex.py =====================
1: 7430V
2: 7700V
1: 7429V
2: 7700V
1: 7430V
2: 7704V
1: 7431V
2: 7701V
1: 7428V
2: 7700V
1: 7430V
2: 7700V
1: 7431V
2: 7701V
1: 7430V
2: 7701V
1: 7429V
2: 7700V
1: 7428V
2: 7699V
1: 7431V
2: 7702V
1: 7429V
2: 7700V
1: 7430V
2: 7700V
```

Fig 6.5 Flex checking in raspberry pi

Fig 6.5 depicts the reading in raspberry pi OS, which shows the flex readings when the sensor detects changes in the pressure.

CHAPTER 7

SYSTEM TESTING

System testing is a crucial part of any project to ensure that it meets the desired requirements and functions correctly. For the smart chair project, there are several components that need to be tested to ensure their proper functioning.

7.1 Tests conducted

7.1.1 Hardware testing

During the hardware testing phase of our project, we checked the connections between the sensors, Raspberry Pi, and other electronic components to ensure that they were properly wired and had secure connections. We tested the flex sensor by bending it to different degrees and verifying that the output values were accurate and within the desired range. We also tested the DHT11 sensor by varying the temperature and humidity levels and verifying that the output values were accurate and within the desired range. In addition, we tested the ADS1115 module by verifying that it was properly converting the analog signals from the flex sensor to digital signals with 16-bit resolution. We tested the alert system by varying the posture and position of the user on the chair and verifying that the correct alerts and notifications were generated on the mobile application. We also verified that the data from the sensors was being correctly stored in the MySQL database and was accessible through the mobile application.

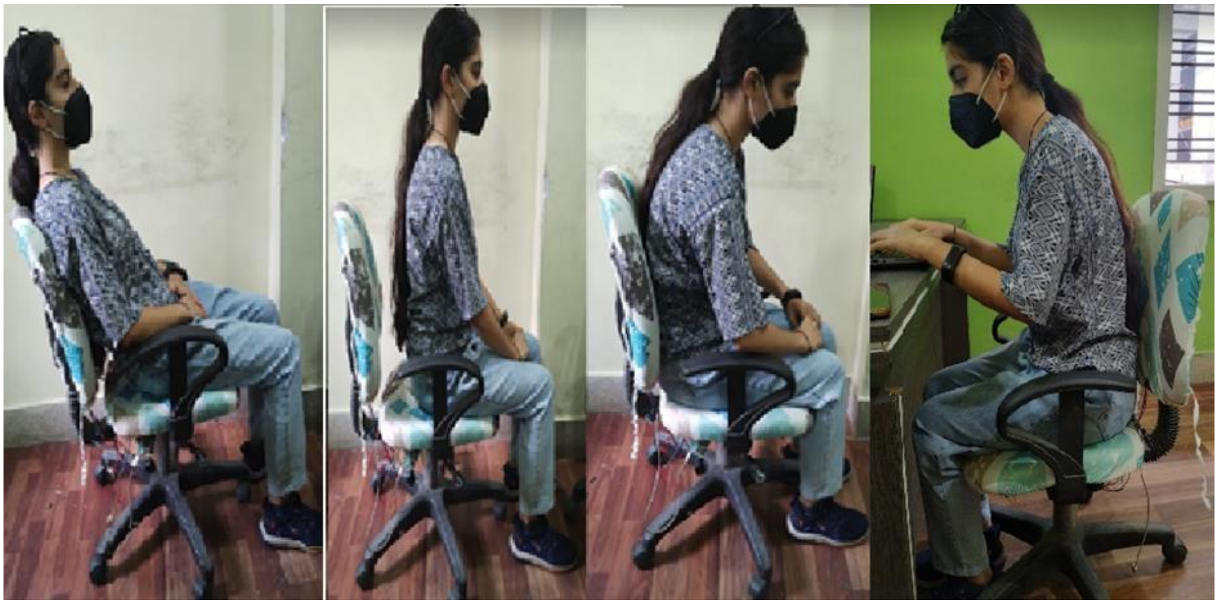


Fig. 7.1 Improper sitting positions

Fig 7.1 depicts the improper sitting positions. A person can minimize the risk of bad posture and back health by avoiding: sitting slumped to one side with the spine bent. keeping the knees, ankles, or arms crossed. dangling or not properly supporting the feet.

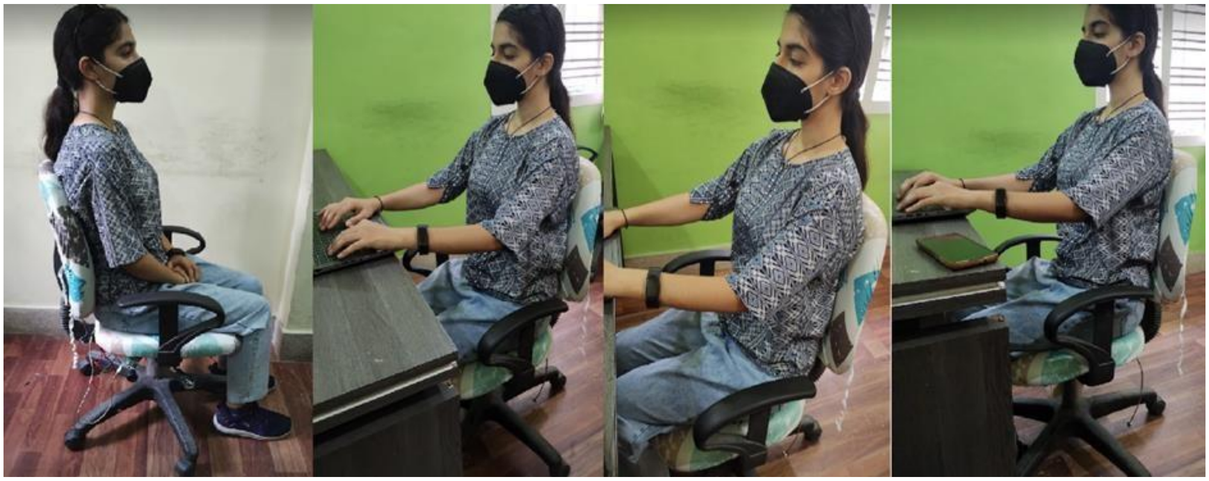


Fig. 7.2 Proper sitting positions

Fig 7.2 depicts the proper sitting posture. Sitting with a straight back and shoulders can help prevent common complaints, such as lower back pain and a stiff neck.

7.1.2 Software testing

Unit testing: We have performed unit testing to ensure that each individual component of the system is functioning correctly. For example, we have tested the flex sensor, DHT11 sensor, and ADS1115 module individually to ensure that they are providing accurate readings.

Integration testing: We have performed integration testing to ensure that all the components are working together as expected. For example, we have tested the communication between the sensors, the Raspberry Pi, and the MySQL server to ensure that data is being transmitted correctly.

Functional testing: We have performed functional testing to ensure that the system is meeting its functional requirements. For example, we have tested the system's ability to detect poor posture and provide alerts and notifications.

Performance testing: We have performed performance testing to ensure that the system is performing optimally. For example, we have tested the system's response time to various thing, stimuli and its ability to handle multiple users simultaneously.

User acceptance testing: We have also conducted user acceptance testing to ensure that the system meets the user's expectations and requirements. This testing involves getting feedback from users and making necessary improvements to the system.

7.2 Test Cases

Table 7.1 Test cases for the Iot based smart chair using raspberry pi

ID	Description	Expected Output	Actual Output	Remark
1	Data Collection	The sensor data to becollected from Raspberry Pi	Same as expected	Pass
2	Transfer data	The collected data tobe sent to the server using HTTP protocol	Same as expected	Pass
3	Signup	User should be ableto sign up using the mobile application	Same as expected	Pass
4	Login	User should be able from the mobile application	Same as expected	Pass
5	Monitor user	Server-side application shouldbe able to monitor the user's sitting pattern	Same as expected	Pass
6	Send alert	Server-side application should be able to send alert to the user's mobileapp	Partial	Pass

Table 7.1 provides the analysis regarding the test cases. It provides a comparison between the actual and expected outputs. It can be said that the IoT based smart chair using raspberry pi performed reasonably well on all the test cases.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 Conclusion

The developed smart chair system will be able to monitor the sitting behavior of human body and help in advocating better sitting habits of users as considering the posture problem we have whether it's the result of sitting at a desk all day, looking down at a smartphone, or lounging on a couch, poor posture is dogging people of all ages. Turning a normal chair into a smart chair that will continuously detect your sitting posture and sends a notification for wrong postures. Flex sensors, DHT11 temperature and humidity sensor and ADS1115 Analog to digital converter module connected with Raspberry Pi will be embedded in the office chair. A notification will tell a person when he/she is sitting for more than 30 minutes. Two flex sensors will be fitted onto the chair cover for the backrest of the chair and one on the bottom seat cover for detecting pressure. The backrest sensors will detect if the person is sitting straight or not based on the pressure or flex on the chair. The sensor on the seat will also be detecting flex based on the pressure of the person sitting, it will also keep detecting the temperature and humidity of the person. If more than 30 minutes has passed an alert is sent through the app which will let the user know that he/she have to take a stretch or exercise for a healthy lifestyle. In future development of the chair features, more advanced sensors can be adopted, as strain gauges-based ones or sensors measuring the rotation of the chair, in order to obtain signals that can be used to extract dynamic parameters proportional to the level of pressure applied or to the swinging movement. This data may also provide information regarding the movement of the participant in different tasks.

8.2 Future Enhancements

Future improvements to an IoT-based smart chair project that uses a Raspberry Pi, flex sensor, DHT11 temperature and humidity module, and an app to measure posture are possible. Among these improvements are:

a) **Adding machine learning:** By incorporating machine learning algorithms into the project, the smart chair can learn the user's sitting habits and provide more accurate feedback on their posture. This can assist users in improving their posture over time.

b) **Integrating Additional Sensors:** To provide a more comprehensive posture analysis, additional sensors such as an accelerometer or a gyroscope can be added to the smart chair. These sensors can aid in movement detection and provide more detailed information about the user's posture.

c) **Providing Customized Feedback:** The smart chair can be programmed to provide the user with customized feedback based on their posture. For example, if the user is slouching, the chair can vibrate to remind them to sit up straight.

d) **Voice Recognition:** By incorporating voice recognition technology, users will be able to interact with the smart chair via voice commands, making it easier to use.

e) **Creating a Mobile App:** A mobile app can be created to allow users to track their progress and monitor their posture over time. To assist users in improving their posture, the app can also provide personalized recommendations and reminders.

REFERENCES

[1] Arrogi A, Boen F, and Seghers J, “Validation of a smart chair and corresponding smartphone app as an objective measure of desk-based sitting”, Journal of Occupational Health, Vol No: 60(1), Pg No: 85-93, doi: 10.1002/1348- 9585.12033, 2018.

[2] Mingyu Park, Younghoon Song, Jaewon Lee, and Jeongyeup Paek, “Design and Implementation of a Smart Chair system for IoT”, In Proceedings of the International Conference on ICT Convergence (ICTC), Pg No: 1062-1064, doi: https://doi.org/10.1109/ICTC.2016.7763406, 2016.

[3] Dr. A. Udhaya Kumar, P. Karuppasamy, J. Kavinkumar, and V. Neelakandan, “A smart chair with posture recognition and temperature monitoring system”, IEEE Sensors Journal, Vol No: 20(14), Pg No: 8117-8124, doi: http://dx.doi.org/10.1109/JSEN.2020.2980207, 2022.

[4] Geonil Kim, Steven Zhou, and Brian Hill, “Posture Sensing Smart Chair”, Pg No: 5-10, 2020.

[5] Tariku Adane Gelaw, Misgina, and Tsighe Hagos, “Posture Prediction for Healthy Sitting usinga Smart Chair”, In Proceedings of the International Conference on Information and Communication Technology for Sustainable Development (ICT4SD 2021) Pg No: 306-315, doi: https://doi.org/10.1007/978-3-030-93709-6_26, 2022.

[6] Cerchecci M, "A Low Power IoT Sensor Node Architecture for Waste Management Within Smart Cities Context", Issue No: 18(4), Vol No: 1282. Pg No: 18, doi: https://doi.org/10.3390/s18041282, 2018.

[7] Contreras D J, García J J, and Diaz D J, "Developing of Industry 4.0 Applications," International Journal of Online Engineering (Ijoe), Pg No: 30, doi: https://doi.org/10.3991/ijoe.v13i10.7331, 2017.

[8] Vazquez-Briseno M, "mHealth Platform and Architectures to Provide Nutritional Guidance to Children", International Journal of Interactive Mobile Technologies (Ijim), Issue No: 7(4), PgNo: 40-54, doi: https://doi.org/10.3991/ijim.v7i4.3083, 2013.

[9] Punnett L, Fine L J, Keyserling W M, Herrin G D, and Chaffin D B, “Back disorders and nonneutral trunk postures of automobile assembly workers", doi: 10.5271/sjweh.681, 2002.

[10] Hedge A, “Effects of ergonomic interventions in computer workplaces” Pg No: 91-105, doi: 10.1518/hfes.46.1.91.30392, 2004.

[11]Karakolis T, and Callaghan J P, “The impact of sit-stand office workstations on worker discomfort and productivity: A review. Applied Ergonomics”, Issue No: 45(5), Pg No: 799-806, doi: 10.1016/j.apergo.2013.10.002, 2014.

[12] O'Sullivan P B, Mitchell T, Bulich P, Waller R, Holte J, and Johnson G, “The relationship between posture and back muscle endurance in industrial workers with flexion-related low back pain”, Pg No: 264-271, doi: 10.1016/j.math.2006.02.004, 2006.

[13] Kim J, Kim W, “Effect of an exercise program with a pelvic cushion on musculoskeletal pain and trunk posture of office workers: A randomized controlled trial”, doi: 10.1002/1348-9585.12206, 2021.

[14] Slavomir Matuska, Martin Paralic, Robert Hudec, “A Smart System for Sitting Posture Detection Based on Force Sensors and Mobile Application”, Pg No: 5, doi: https://doi.org/10.1155/2020/6625797, 2020.

[15] Daniele Bibbo, Marco Carli, Silvia Conforto, Federica Battisti, “A Sitting Posture Monitoring Instrument to Assess Different Levels of Cognitive Engagement”, Pg No: 5, doi: https://doi.org/10.3390%2Fs19030455, 2019.

[16] Timon Padberg, Sojat Sebgaze, Rim Zakar, Rajeev Kanth, “Design and Implementation of Smart IoT Chair for Better Health and Productive Work”, Pg No: 6, doi: 10.13140/RG.2.2.29339.75044, 2022.

[17] Brijesh Kundaliya, Smit Patel, Jaanvi Patel, Parv Barot, and S. K. Hadia, “An IoT and CloudEnabled Smart Chair for Detection and Notification of Wrong Seating Posture”, Pg No: 6, doi: https://doi.org/10.21203/rs.3.rs-1999906/v1, 2022.

[18] Vidhya. B, Abdul Hayum. A, Saranya. G, Aarthi. M, Aishwarya. V, and Dhanashree. K, ‘Smart PostureDetector Using IoT”, Pg No: 7, doi: https://doi.org/10.1109/ICCES51350.2021.9489099, 2021.

[19] Taraneh Aminosharieh Najafi, Antonio Abramo, Kyandoghere Kyamakya, and Antonio Affanni, “Development of a Smart Chair Sensors System and Classification of Sitting Postures with Deep Learning Algorithms”, Pg No: 7, doi: https://doi.org/10.3390/s22155585, 2022.

[20] Mengjie Huang, Ian Gibson, and Rui Yang, “Smart Chair for Monitoring of Sitting Behavior”, Pg No: 8, doi:10.18502/keg.v2i2.626, 2017.

[21] Ana Castillo-Martinez, Jose-Maria Gutierrez-Martinez, Roberto Barchino, Juan José Córdoba Zamora, and Sergio de la Mata Moratilla, “Healthy sitting behavior enhancement usinga smart chair system”, Pg No: 9, 2021.

[22] A Chaitanya Kumar, V G Sridhar, “Design and Analytics of Smart Posture Monitoring System”, J. Phys.: Conf. Ser. Issue No: 2115, Vol No: 012048, Pg No: 10, doi: 10.1088/1742- 6596/2115/1/012048, 2021.

[23] Zoran Vlaović, Marko Jaković, and Danijela Domljan, “Smart Office Chairs with Sensors for Detecting Sitting Positions and Sitting Habits: A Review”, Pg No: 11, doi: https://doi.org/10.5552/drvind.2022.0002, 2022.

APPENDIX A

A.1 Source Code

Httpconnection.java

```
package com.example.chair_app,
import org.json.JSONException,
import org.json.JSONObject,
import java.io.BufferedReader,
import java.io.BufferedWriter,
import java.io.IOException,
import java.io.InputStreamReader,
import java.io.OutputStreamWriter,
import java.net.HttpURLConnection,
import java.net.URL,
import java.net.URLEncoder,
import java.util.Iterator,
/**
 * */
public class HttpConnection {
    public static String getResponse(URL url, JSONObject jsn) throws JSONException,
IOException {
        String strSend = "",
        Iterator<String> itr = jsn.keys(),
        boolean first = true,
        while (itr.hasNext()) {
            String key = itr.next(),
            String value = jsn.get(key).toString(),
            if (first) {
                first = false,
            } else {
                strSend += "&",
            }
            strSend += URLEncoder.encode(key, "UTF-8"),
            strSend += "=",
            strSend += URLEncoder.encode(value, "UTF-8"),
        }
        HttpURLConnection con = (HttpURLConnection) url.openConnection(),
        con.setRequestMethod("POST"),
        BufferedWriter writer = new BufferedWriter(new
OutputStreamWriter(con.getOutputStream())),
        writer.write(strSend),
        writer.close(),
        int code = con.getResponseCode(),
        if (code == 200) {
            BufferedReader reader = new BufferedReader(new
InputStreamReader(con.getInputStream())),
            String strRecv = "",
            String line = reader.readLine(),
            while (line != null) {
                strRecv += line,
                line = reader.readLine(),
            }
            return strRecv,
        } else {
            return "Error Code : " + code,
        }
    }
}
```

Signup.java

```
package com.example.chair_app,
import android.content.Intent,
import  android.os.Bundle,
import android.os.StrictMode,
import android.view.View,
import android.widget.ArrayAdapter,
import  android.widget.Button,
import android.widget.EditText,
import android.widget.Spinner,
import android.widget.Toast,
import androidx.appcompat.app.AppCompatActivity,
import org.json.JSONException,
import org.json.JSONObject,
import java.io.IOException,
import java.net.URL,
public class signup extends AppCompatActivity {
   EditText etName,
   EditText etMob,
   EditText etEmail,
   EditText etPass,
   Spinner sp,
   Button button,
   String name, mobile,  email,password,
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState),
      setContentView(R.layout.activity_signup),
      etName = (EditText) findViewById(R.id.etName),
      etMob = (EditText) findViewById(R.id.etMob),
      etEmail = (EditText) findViewById(R.id.etEmail),
      etPass = (EditText) findViewById(R.id.etPass),
      button = (Button) findViewById(R.id.button),
      button.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            name = etName.getText().toString(),
            mobile = etMob.getText().toString(),
            email = etEmail.getText().toString(),
            password = etPass.getText().toString(),
            try {
                send_data(),
            } catch (IOException e) {
                e.printStackTrace(),
            } catch (JSONException e) {
                e.printStackTrace(),
            }
        }
    }),
}
public void send_data() throws IOException, JSONException {
    URL url = new URL(Global.url + "signup"),
    JSONObject jsn = new JSONObject(),
    jsn.put("name", name),
    jsn.put("mobile", mobile),
    jsn.put("email", email),
    jsn.put("password", password),
      StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().build(),
    StrictMode.setThreadPolicy(policy),
    String response = null,
    response = HttpConnection.getResponse(url, jsn),
    if (response.equalsIgnoreCase("ok")) {
        Intent intennt = new Intent(signup.this, login.class),
        startActivity(intennt),
        Toast.makeText(signup.this, "Registered Successfully",
Toast.LENGTH_SHORT).show(),
    }  else {
        Toast.makeText(signup.this, "Registration Failed", Toast.LENGTH_SHORT).show(),
    }
    }
}
```

Login.java

```
package com.example.chair_app,
import android.content.Intent,
import android.content.SharedPreferences,
import android.os.Bundle,
import android.os.StrictMode,
import android.util.Log,
import android.view.View,
import android.widget.Button,
import android.widget.EditText,
import android.widget.TextView,
import android.widget.Toast,
import androidx.appcompat.app.AppCompatActivity,
import org.json.JSONException,
import org.json.JSONObject,
import java.io.IOException,
import java.net.MalformedURLException,
import java.net.URL,
public class login extends AppCompatActivity {
    public static final String PREFS_NAME = "MyPrefsFile",
    private static final String PREF_USERNAME = "username",
    private static final String PREF_PASSWORD = "password",
    EditText usr,
    EditText pwd,
    TextView studname, studemail,
    Button lbtn,
    Intent intent = null,
    String mob, pass,
    TextView tv,
Intent i1,i2,
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState),
    setContentView(R.layout.activity_login),
    usr = findViewById(R.id.etmob),
    pwd = findViewById(R.id.pw),
    lbtn = findViewById(R.id.loginbtn),
    tv = findViewById(R.id.tv),
    i1=new Intent(this, MainActivity.class),
    i2=new Intent(this,signup.class),
    /*to store username and password*/
    SharedPreferences pref = getSharedPreferences(PREFS_NAME, MODE_PRIVATE),
    String username = pref.getString(PREF_USERNAME, null),
    String password = pref.getString(PREF_PASSWORD, null),
    //Toast.makeText(this, username+password, Toast.LENGTH_SHORT).show(),
    if (username == null || password == null) {
        getSharedPreferences(PREFS_NAME, MODE_PRIVATE)
                .edit()
                .putString(PREF_USERNAME, usr.getText().toString())
                .putString(PREF_PASSWORD, pwd.getText().toString())
                .apply(),
    } else {
        usr.setText(username),
        pwd.setText(password),
    }
    tv.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
        startActivity(i2),
        }
    }),
        //add the function to connect to database
        lbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                getSharedPreferences(PREFS_NAME, MODE_PRIVATE)
                        .edit()
                        .putString(PREF_USERNAME, usr.getText().toString())
                        .putString(PREF_PASSWORD, pwd.getText().toString())
                        .apply(),
                mob = usr.getText().toString().trim(),
                pass = pwd.getText().toString().trim(),
                login_check(),
            }
        }),
    }
        }
        catch (MalformedURLException e) {
            e.printStackTrace(),
        } catch (JSONException e) {
            e.printStackTrace(),
        } catch (IOException e) {
            e.printStackTrace(),
        }
    }
}
```

Mainactivtiy.java

```
package com.example.chair_app,
import android.Manifest,
import android.app.Activity,
import android.app.AlertDialog,
import android.content.Context,
import android.content.DialogInterface,
import android.content.Intent,
import android.content.pm.PackageManager,
import android.hardware.SensorManager,
import android.net.Uri,
import android.os.AsyncTask,
import  android.os.Bundle,
import android.os.Handler,
import android.os.StrictMode,
import android.provider.Settings,
import android.util.Log,
import android.widget.ImageView,
import android.widget.TextView,
import android.widget.Toast,
import androidx.core.app.ActivityCompat,
import org.json.JSONException,
import org.json.JSONObject,
import java.io.IOException,
import java.net.MalformedURLException,
import java.net.URL,
import java.util.Timer,
import java.util.TimerTask,
public class MainActivity extends Activity{// implements SensorEventListener {
    public static String name="",mob="",
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState),
        setContentView(R.layout.activity_main),
        final Handler handler = new Handler(),
        Timer timer = new Timer(),
        TimerTask doAsynchronousTask = new TimerTask() {
            @Override
            public void run() {
                handler.post(new Runnable() {
                    public void run() {
                        try {
                            new PerformBackgroundTask().execute(),
                        } catch (Exception e) {
                        }
                    }
                }),
            }
        },
        timer.schedule(doAsynchronousTask, 0, 5000),
    }
public class PerformBackgroundTask extends AsyncTask<String, Void, String> {
    public void onPreExecute() {
    }
    @Override
    protected void onPostExecute(String status) {
    }
}
}
```

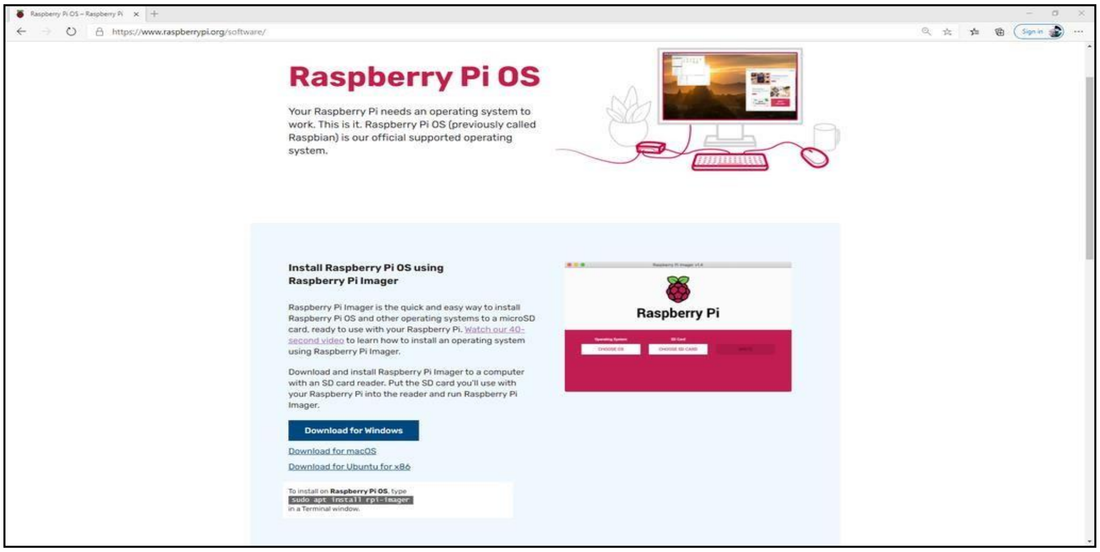
Splashscreen.java

```
package com.example.chair_app,
import android.app.Activity,
import android.content.Intent,
import  android.os.Bundle,
import android.view.Window,
public class Splashscreen extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        this.requestWindowFeature(Window.FEATURE_NO_TITLE),
        super.onCreate(savedInstanceState),
        setContentView(R.layout.activity_splashscreen),
        Thread mythread =new Thread(){
            @Override
            public void run()
            {
                try {
                    sleep(3000),
                    Intent intent=new Intent(getApplicationContext(), login.class),
                    startActivity(intent),
                    finish(),
                } catch (InterruptedException e) {
                    e.printStackTrace(),
                }
            }
        },
        mythread.start(),
    }
}
```

A.2 INSTALLATION PROCEDURE AND DEPLOYMENT

Installing Raspbian on the Raspberry Pi is pretty straightforward. We'll be downloading Raspbian and writing the disc image to a microSD card, then booting the Raspberry Pi to that microSD card. For this project, you'll need a microSD card (go with at least 8 GB), a computer with a slot for it, and, of course, a Raspberry Pi and basic peripherals (a mouse, keyboard, screen, and power source). This isn't the only method for installing Raspbian (more on that in a moment), but it's a useful technique to learn because it can also be used toinstall so many other operating systems on the Raspberry Pi. Once you know how to write a disc image to a microSD card, you open up a lot of options for Raspberry Pi projects.

Step 1: Download and Install Raspberry Pi Imager



Raspberry Pi OS – Raspberry Pi
https://www.raspberrypi.org/software/
Sign in

Raspberry Pi OS

Your Raspberry Pi needs an operating system to work. This is it. Raspberry Pi OS (previously called Raspbian) is our official supported operating system.

Install Raspberry Pi OS using Raspberry Pi Imager

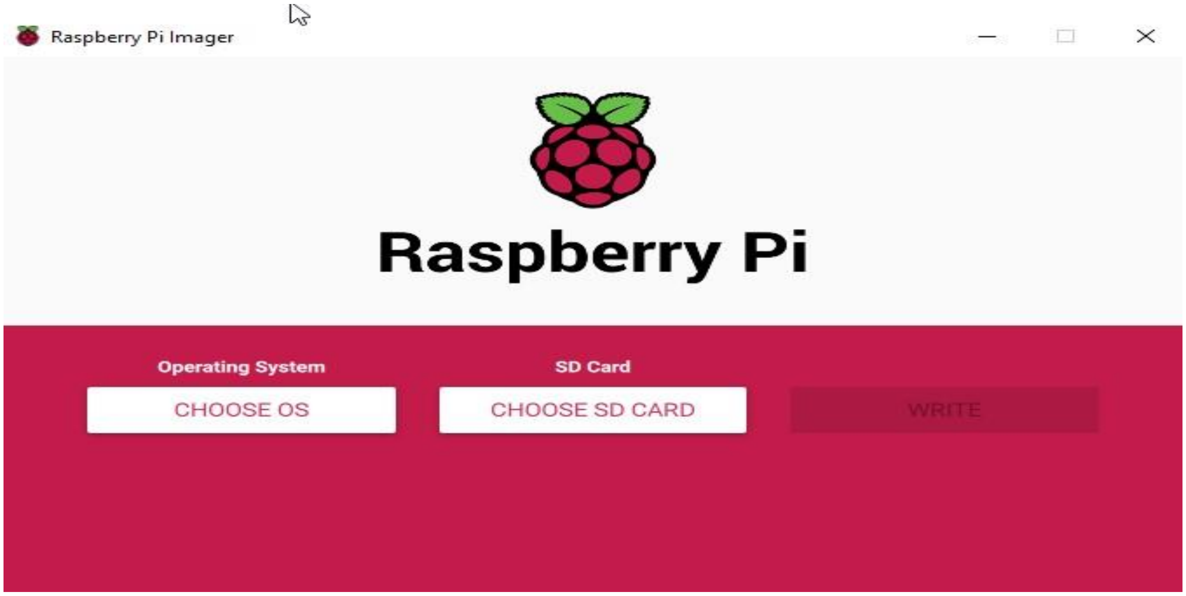
Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. Watch our 40-second video to learn how to install an operating system using Raspberry Pi Imager.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

To install on **Raspberry Pi OS**, type `sudo apt install rpi-imager` in a Terminal window.

Download Raspberry Pi Imager from RaspberryPi.org/software. There are versions for Windows, Mac, Ubuntu, and Raspberry Pi OS itself. Chose the one that works for you, andinstall it just like you would any other program.

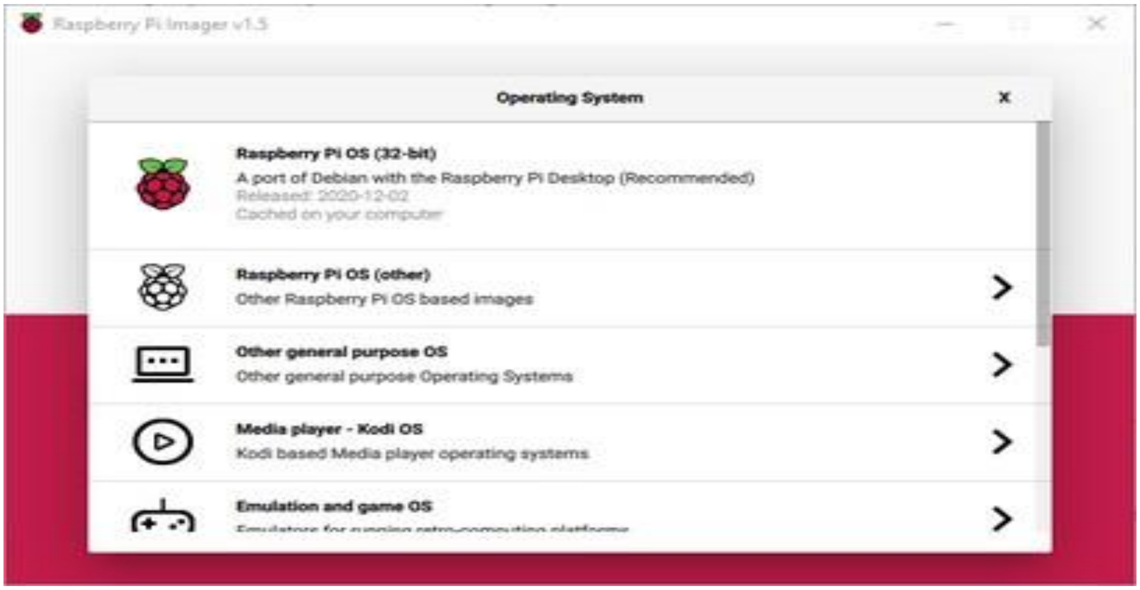
Step 2: Run Raspberry Pi Imager and Format the SD Card



- With the target SD Card inserted into your computer or external card reader, open Raspberry Pi Imager.
- When you run Raspberry Pi Imager you are presented with two options: Choose OS and Choose SD Card.
- Scroll down to the bottom of the list and click on the Erase option.

- Now click the "Choose SD Card" Option and select the target SD Card. Note that any data still on this SD card will be erased forever, and you should ensure that you have a backup of any files you would like to keep.
- With the SD Card Selected the "Write" option becomes available. Click Write and wait for the process to complete. Once the process is complete, a notification window will open letting you know that it's now ok to remove the SD card from thereader. Remove the SD Card and reinsert it into the reader to make it available again.

Step 3: Burn the Raspberry Pi OS Image to the SD Card



- With a freshly formatted SD card, we can now move on to installing the operating system.
- Click the "Choose OS" button, and select one of the available operating systems. For example, Raspberry Pi OS (32-Bit)
- Now select the SD card we just formatted in the previous step.
- The "Write" option will become available. Click "Write" to begin burning the image to the SD card.
- It can take anywhere from a few minutes to upwards of half an hour for the process to complete depending on the quality and speed of the SD card, card reader, and computer.

- Be patient and wait for this process to complete. If you remove the SD card, unplug the card reader, or shut down the computer at any point during this process, the card will become bricked and unusable.
- Once Raspberry Pi Imager has finished writing the files to the SD card, it will verify that the image on the SD card is identical to the image file used to burn the image. This usually takes less than a minute but could take longer.
- When the verification process is complete, a notification window will open letting you know that the write was successful and that it's now safe to remove the SD card.

A.3 EXECUTING THE APPLICATION

A.3.1 Executing the python scripts written in Raspberry pi

Here are the general steps to execute Python scripts in the Raspberry Pi module:

1. Open the Terminal or Command Prompt and connect to the Raspberry Pi through SSH or VNC Viewer.
2. Navigate to the directory where the Python script is located using the "cd" command.
3. Make sure that the script has executable permissions using the "ls -l" command. If the script does not have executable permissions, use the "chmod +x Smart_chair.py" command to add them.
4. Run the Python script using the "Smart_chair.py" command. If the script requires any arguments, pass them after the script name.
5. Monitor the script output and verify that it is running correctly. If there are any errors or issues, debug the script as necessary.

A.3.2 Executing the java scripts in Netbeans IDE

1. Make sure that the MySQL server is running and the database is created with the required tables and fields.
2. Open NetBeans IDE and import the Java MySQL Connector library to connect to the MySQL server.
3. Write the code to retrieve the data from the MySQL server using SQL queries.
4. Set up a timer that runs every 30 seconds to fetch the latest data from the server.

5. Compile and run the Java code in NetBeans IDE to test if the data retrieval is working properly.
6. Export the Java project as a JAR file.
7. Transfer the JAR file to the Android project in Android Studio and import it as a library.
8. Write the code in Android Studio to use the Java library and display the data in the Android application.
9. Compile and run the Android application to test if the data is being fetched correctly every 30 seconds.

Once the application is executed, the user can sign up or log in to the Android application and view the posture data being fetched from the MySQL server. The application will generate alerts and notifications based on the posture data, and the user can customize the temperature, humidity, and flex sensor sensitivity settings as per their preference.

A.3.3 Building a apk file in android studio and installing it

Here is a brief procedure on building an APK file in Android Studio and installing the APK in a mobile device:

1. Open the Android Studio and make sure that the project you want to build an APK file for is open and loaded.
2. In the top menu bar, click on "Build" and select "Generate Signed Bundle / APK".
3. In the dialog box that appears, select "APK" and click on "Next".
4. Select the keystore that you want to use to sign the APK file. If you don't have a keystore, you can create one by clicking on "Create new".
5. Fill in the required details in the keystore creation wizard and click on "OK".
6. Select the key alias that you want to use and click on "Next".
7. Select the build type that you want to use (debug or release) and click on "Next".
8. Choose the destination folder where you want to save the APK file and click on "Finish".
9. Wait for the APK file to be generated. Once it is done, you will see a message that says "APK(s) generated successfully".
10. Transfer the APK file to your mobile device using a USB cable or any other method.
11. On your mobile device, go to the "Settings" app and navigate to "Security".
12. Enable the "Unknown sources" option, which allows you to install apps from sources other than the Google Play Store.

13. Open the file manager app on your mobile device and navigate to the folder where you saved the APK file.
14. Tap on the APK file to start the installation process.
15. Follow the on-screen instructions to install the app on your mobile device.
16. Once the app is installed, you can launch it from your app drawer and start using it.